Sqoop2 integration with Sentry

Installing Sentry

The new feature Sqoop integration will be release 1.6.0. Any user which wants to use this feature should build the Sentry package from trunk.

```
$ git clone https://github.com/apache/incubator-sentry.git
$ mvn clean  install –DskipTests
$ cp sentry-dist/target/apache-sentry-1.6.0-incubating-SNAPSHOT-bin.tar.gz ${yourTestDir}
$ cd ${ yourTestDir}
$ tar –xzvf apache-sentry-1.6.0-incubating-SNAPSHOT-bin.tar.gz
$ cd apache-sentry-1.6.0-incubating-SNAPSHOT-bin
$ export SENTRY_HOME=`pwd`
$ export PATH=${PATH}:${SENTRY_HOME}/bin
```

Initializing database for Sentry Service

The Sentry Service uses Database to store privilege metadatas. Sentry uses JDO layer to decouple the underlying specific database. So the Sentry supports a lot of Database type such as Mysql, Oracle, Postgres, DB2 and Derby. The user should at firstly initialize the Database schema before starting the Sentry Service. The Sentry has already supply the initializing tool for all kinds of Database.

Initialize Derby:

Prepare a configuration file sentry-site.xml before executing the initialize command.

The configuration must contain information about sentry.store.jdbc.url, sentry.store.jdbc.driver, sentry.store.jdbc.user and sentry.store.jdbc.password.

The sentry-site.xml:

```
<configuration>
  <property>
    <name>sentry.store.jdbc.url</name>
    <value>jdbc:derby:;databaseName=sentry_db;create=true</value>
    <description>JDBC connection URL for the backed DB</description>
```

```
  </property>
  <property>
    <name>sentry.store.jdbc.user</name>
    <value>Sentry</value>
    <description>Userid for connecting to backend db </description>
  </property>
  <property>
    <name>sentry.store.jdbc.password</name>
    <value>Sentry</value>
    <description>Sentry password for backend JDBC user </description>
  </property>
  <property>
    <name>sentry.store.jdbc.driver</name>
    <value>org.apache.derby.jdbc.EmbeddedDriver</value>
    <description>Backend JDBC driver - org.apache.derby.jdbc.EmbeddedDriver (only when dbtype = derby) JDBC Driver class for the backed DB</description>
  </property>
</configuration>
```

```
$ sentry --command schema-tool -dbType derby -initSchema -conffile sentry-site.xml
```

Initialize Mysql:

The sentry-site.xml:

```
<configuration>
  <property>
    <name>sentry.store.jdbc.url</name>
    <value>  jdbc:mysql://10.239.47.76:3306/sentry?useUnicode=true&amp;characterEncoding=UTF-8 </value>
    <description>JDBC connection URL for the backed DB</description>
```

```
  </property>
  <property>
    <name>sentry.store.jdbc.user</name>
    <value>root</value>
    <description>Userid for connecting to backend db </description>
  </property>
  <property>
    <name>sentry.store.jdbc.password</name>
    <value>password</value>
    <description>Sentry password for backend JDBC user </description>
  </property>
  <property>
    <name>sentry.store.jdbc.driver</name>
    <value>  com.mysql.jdbc.Driver</value>
    <description>Backend JDBC driver - org.apache.derby.jdbc.EmbeddedDriver (only when dbtype = derby) JDBC Driver class for the backed DB</description>
  </property>
</configuration>
```

```
$ sentry --command schema-tool -dbType mysql -initSchema -conffile sentry-site.xml
```

Initialize Postgres:

The sentry-site.xml:

```
<configuration>
  <property>
    <name>sentry.store.jdbc.url</name>
    <value>jdbc:postgresql://server-582:5432/sentry</value>
    <description>JDBC connection URL for the backed DB</description>
```

```
  </property>
  <property>
    <name>sentry.store.jdbc.user</name>
    <value>sentry</value>
    <description>Userid for connecting to backend db </description>
  </property>
  <property>
    <name>sentry.store.jdbc.password</name>
    <value>sentry</value>
    <description>Sentry password for backend JDBC user </description>
  </property>
  <property>
    <name>sentry.store.jdbc.driver</name>
    <value>org.postgresql.Driver</value>
    <description>Backend JDBC driver - org.apache.derby.jdbc.EmbeddedDriver (only when dbtype = derby) JDBC Driver class for the backed DB</description>
  </property>
</configuration>
```

```
$ sentry --command schema-tool -dbType postgres -initSchema -conffile sentry-site.xml
```

Starting Sentry Service

The Sentry Service runs in a secure environment like Kerberos. So at firstly it needs to generate a principal keytab file.

```
$ kinit admin/admin
$ kadmin
$ addprinc -randkey sentry/server-592.novalocal@HADOOP.COM
$ xst -k sentry.keytab sentry/server-592.novalocal@HADOOP.COM
$ mv sentry.keytab /etc/sentry/conf
```

The Sentry Service needs a configuration file sentry-site.xml to run. The properties sentry.store.jdbc.url, sentry.store.jdbc.driver, sentry.store.jdbc.user and sentry.store.jdbc.password must match the underlying Database in the sentry-site.xml.

The guide uses the Derby as the underlying Database. So the sentry-site.xml as followings:

The sentry-site.xml:

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<configuration>
 <property>
  <name>sentry.verify.schema.version</name>
  <value>false</value>
 </property>
 <property>
  <name>sentry.service.allow.connect</name>
  <value>sqoop2,impala,hive,solr</value>
 </property>
 <property>
  <name>sentry.store.jdbc.url</name>
  <value>jdbc:derby:;databaseName=sentry_db;create=true</value>
 </property>

 <property>
  <name>sentry.store.jdbc.user</name>
  <value>Sentry</value>
 </property>
 <property>
  <name>sentry.store.jdbc.password</name>
  <value>Sentry</value>
 </property>
 <property>
```

```
  <name>sentry.service.server.keytab</name>
  <value>/etc/sentry/conf/sentry.keytab</value>
 </property>
 <property>
  <name>sentry.service.server.rpcport</name>
  <value>8038</value>
 </property>

 <property>
  <name>sentry.service.server.rpcaddress</name>
  <value>0.0.0.0</value>
 </property>

 <property>
  <name>sentry.store.jdbc.driver</name>
  <value>org.apache.derby.jdbc.EmbeddedDriver</value>
 </property>
 <property>
  <name>sentry.service.admin.group</name>
  <value>sqoop2,hive,impala,solr</value>
 </property>
 <property>
  <name>sentry.service.security.mode</name>
  <value>kerberos</value>
 </property>
 <property>
  <name>sentry.service.server.principal</name>
  <value>sentry/server-592.novalocal@HADOOP.COM</value>
 </property>
</configuration>
```

Starting Service:

```
$ sentry --command service --conffile sentry-site.xml
15/05/15 14:17:32 INFO thrift.SentryService: Configured on address /0.0.0.0:8038
15/05/15 14:17:32 INFO thrift.SentryService: Using kerberos principal: sentry/server-592.novalocal@HADOOP.COM
15/05/15 14:17:32 INFO thrift.SentryService: Attempting to start...
15/05/15 14:17:32 INFO thrift.SentryService: Waiting on future.get()
Debug is  true storeKey true useTicketCache true useKeyTab true doNotPrompt true ticketCache is null isInitiator true KeyTab is /root/richard/sentry/apache-sentry-1.6.0-incubating-SNAPSHOT-bin/conf/sentry.keytab refreshKrb5Config is true principal is sentry/server-592.novalocal@HADOOP.COM tryFirstPass is false useFirstPass is false storePass is false clearPass is false
Refreshing Kerberos configuration
Acquire TGT from Cache
Principal is sentry/server-592.novalocal@HADOOP.COM
null credentials from Ticket Cache
principal is sentry/server-592.novalocal@HADOOP.COM
Will use keytab
Commit Succeeded

15/05/15 14:17:32 INFO thrift.SentryKerberosContext: Sentry Ticket renewer thread started
15/05/15 14:17:32 INFO DataNucleus.Persistence: Property datanucleus.cache.level2 unknown - will be ignored
15/05/15 14:17:33 WARN bonecp.BoneCPConfig: Max Connections < 1. Setting to 20
15/05/15 14:17:37 WARN bonecp.BoneCPConfig: Max Connections < 1. Setting to 20
15/05/15 14:17:37 INFO DataNucleus.Persistence: Property datanucleus.cache.level2 unknown - will be ignored
15/05/15 14:17:37 WARN bonecp.BoneCPConfig: Max Connections < 1. Setting to 20
15/05/15 14:17:38 WARN bonecp.BoneCPConfig: Max Connections < 1. Setting to 20
15/05/15 14:17:38 INFO thrift.SentryService: Serving on /0.0.0.0:8038
```

Installing Sqoop2

Download the Sqoop-1.99.6 release and install it.

```
$ wget http://archive.apache.org/dist/sqoop/1.99.6/sqoop-1.99.6-bin-hadoop200.tar.gz
$ tar –xzvf sqoop-1.99.6-bin-hadoop200.tar.gz
$ cd sqoop-1.99.6-bin-hadoop200.tar.gz
$ export SQOOP2_HOME=`pwd`
$ export PATH=${PATH}:${SQOOP2_HOME}/bin
$ export SQOOP2_HOST="server-592.novalocal"
```

Generate Kerberos principals for Sqoop2

The Sentry Service runs in a secure environment like Kerberos. So at firstly it needs to generate a principal keytab file.

```
$ kinit admin/admin
$ kadmin
$ addprinc -randkey sqoop2/server-592.novalocal@HADOOP.COM
$ addprinc –randkey HTTP/server-592.novalocal@HADOOP.COM
$ xst -k sqoop2.keytab sqoop2/server-592.novalocal@HADOOP.COM
$ xst -k sqoop2.keytab HTTP/server-592.novalocal@HADOOP.COM
$ mv sqoop2.keytab /etc/sqoop2/conf
```

Configure Kerberos authentication and Sentry Authorization

The new latest release Sqoop-1.99.6 has support an authorization framework for third-party to integration. Sentry is a security open source product in the Apache Hadoop community that offers role-based authorization control. In this test, we will use the Sentry integration into Sqoop2.Edit the sqoop. properties file to configure Sentry binding authorization.

The default authentication and authorization configuration in the sqoop.properties looks like:

```
$ cat sqoop-1.99.6-bin-hadoop200/server/conf/sqoop.properties
#
# Authentication configuration
#
#org.apache.sqoop.security.authentication.type=SIMPLE
```

```
#org.apache.sqoop.security.authentication.handler=org.apache.sqoop.security.authentication.SimpleAuthenticationHandler

#org.apache.sqoop.security.authentication.anonymous=true

#org.apache.sqoop.security.authentication.type=KERBEROS

#org.apache.sqoop.security.authentication.handler=org.apache.sqoop.security.authentication.
KerberosAuthenticationHandler

#org.apache.sqoop.security.authentication.kerberos.principal=sqoop/_HOST@NOVALOCAL

#org.apache.sqoop.security.authentication.kerberos.keytab=/home/kerberos/sqoop.keytab

#org.apache.sqoop.security.authentication.kerberos.http.principal=HTTP/_HOST@NOVALOCAL

#org.apache.sqoop.security.authentication.kerberos.http.keytab=/home/kerberos/sqoop.keytab

#org.apache.sqoop.security.authentication.enable.doAs=true

#org.apache.sqoop.security.authentication.proxyuser.#USER#.users=*

#org.apache.sqoop.security.authentication.proxyuser.#USER#.groups=*

#org.apache.sqoop.security.authentication.proxyuser.#USER#.hosts=*


#

# Authorization configuration

#

#org.apache.sqoop.security.authorization.handler=org.apache.sqoop.security.authorization.DefaultAuthorizationHandler

#org.apache.sqoop.security.authorization.access_controller=org.apache.sqoop.security.authorization.
DefaultAuthorizationAccessController

#org.apache.sqoop.security.authorization.validator=org.apache.sqoop.security.authorization.DefaultAuthorizationValidator

#org.apache.sqoop.security.authorization.authentication_provider=org.apache.sqoop.security.authorization.
DefaultAuthenticationProvider

#org.apache.sqoop.security.authorization.server_name=SqoopServer1
```

Changing the properties as followings:

```
# Add the sentry packages
```

```
org.apache.sqoop.classpath.extra=${SENTRY_HOME}/lib/sentry-provider-db-1.6.0-incubating-SNAPSHOT.jar:${SENTRY_HOME}/lib/shiro-core/1.2.1/shiro-core-1.2.1.jar:${SENTRY_HOME}/lib/sentry-core-common-1.6.0-incubating-SNAPSHOT.jar:${SENTRY_HOME}/lib/sentry-core-model-db-1.6.0-incubating-SNAPSHOT.jar:${SENTRY_HOME}/lib/sentry-core-model-search-1.6.0-incubating-SNAPSHOT.jar:${SENTRY_HOME}/lib/sentry-core-model-sqoop-1.6.0-incubating-SNAPSHOT.jar:${SENTRY_HOME}/lib/sentry-provider-common-1.6.0-incubating-SNAPSHOT.jar:${SENTRY_HOME}/lib/sentry-policy-common-1.6.0-incubating-SNAPSHOT.jar:${SENTRY_HOME}/lib/libthrift-0.9.2.jar:${SENTRY_HOME}/lib/sentry-provider-file-1.6.0-incubating-SNAPSHOT.jar:${SENTRY_HOME}/lib/sentry-binding-sqoop-1.6.0-incubating-SNAPSHOT.jar:${SENTRY_HOME}/lib/sentry-policy-sqoop-1.6.0-incubating-SNAPSHOT.jar

# Kerberos configuration

org.apache.sqoop.security.authentication.type=KERBEROS

org.apache.sqoop.security.authentication.handler=org.apache.sqoop.security.authentication.KerberosAuthenticationHandler

org.apache.sqoop.security.authentication.kerberos.principal=sqoop2/_HOST@HADOOP.COM

org.apache.sqoop.security.authentication.kerberos.keytab=/etc/sqoop2/conf/sqoop2.keytab

org.apache.sqoop.security.authentication.kerberos.http.principal=HTTP/_HOST@HADOOP.COM

org.apache.sqoop.security.authentication.kerberos.http.keytab=/etc/sqoop2/conf/sqoop2.keytab

org.apache.sqoop.security.authentication.enable.doAs=true

org.apache.sqoop.security.authentication.proxyuser.#USER#.users=*

org.apache.sqoop.security.authentication.proxyuser.#USER#.groups=*

org.apache.sqoop.security.authentication.proxyuser.#USER#.hosts=*

# Sentry Authorization configuration

org.apache.sqoop.security.authorization.handler=org.apache.sentry.sqoop.authz.SentryAuthorizationHander

org.apache.sqoop.security.authorization.access_controller=org.apache.sentry.sqoop.authz.SentryAccessController

org.apache.sqoop.security.authorization.validator=org.apache.sentry.sqoop.authz.SentryAuthorizationValidator

org.apache.sqoop.security.authorization.server_name=SqoopServer1

sentry.sqoop.site.url=file:///etc/sqoop2/conf/sentry-site.xml
```

The /etc/sqoop2/conf/sentry-site.xml:

```
<configuration>
  <property>
    <name>sentry.service.security.mode</name>
    <value>kerberos</value>
  </property>
```

```
<property>
  <name>sentry.service.server.principal</name>
  <value>sentry/server-592.novalocal@HADOOP.COM</value>
</property>
<property>
  <name>sentry.service.client.server.rpc-address</name>
  <value>server-592</value>
</property>
<property>
  <name>sentry.service.client.server.rpc-port</name>
  <value>8038</value>
</property>
<property>
  <name>sentry.sqoop.provider.backend</name>
 <value>org.apache.sentry.sqoop.binding.SqoopProviderBackend</value>
</property>
</configuration>
```

Starting Sqoop2 Service:

```
$ cd ${SQOOP2_HOME}/bin
$ ./sqoop.sh server start
```

Testing Sqoop2 with Sentry Authorization

In the Sentry Authorization component, any user at firstly hasn't privilege on resource

even if the user is an admin. So the first thing is to use the admin to grant privileges to other users which want to access Sqoop resource. The sqoop2 is the admin group

according to Sentry Service configuration:

```
<property>
  <name>sentry.service.admin.group</name>
```

```
  <value>sqoop2,hive,impala,solr</value>
 </property>
```

```
 # sqoop2 as the running sqoop shell
$ kinit sqoop2
$ ./sqoop.sh client
sqoop:000> set option --name verbose --value true
sqoop:000> show connector    #sqoop2 has no privilege
+----+------+---------+-------+----------------------+
| Id | Name | Version | Class | Supported Directions |
+----+------+---------+-------+----------------------+
# sqoop2 as the admin need to grant all privilege to itself
sqoop:000> create role -r sqoop2
sqoop:000> grant role -r sqoop2 --principal-type group --principal
sqoop2
sqoop:000> grant privilege -a all --resource-type server --resource
SqoopServer1 --principal-type role --principal sqoop2
sqoop:000> show role --principal-type group --principal sqoop2
+-----------+
| Role Name |
+-----------+
| sqoop2    |
+-----------+
sqoop:000> show privilege --principal-type role --principal sqoop2
+--------+---------------+---------------+------------+ | Action | Resource Name | Resource Type | With Grant |
+--------+---------------+---------------+------------+
| ALL    | SqoopServer1  | SERVER        | false      |
+--------+---------------+---------------+------------+
# Right now sqoop2 has all privilege on Server scope, so it can do any
operation on any sqoop resource
sqoop:000> show connector
```

```
+----+------------------------+---------------+----------------------
------------------------------+---------------------+
| Id |          Name          |    Version    |
Class                         | Supported Directions |
+----+------------------------+---------------+----------------------
------------------------------+---------------------+
| 1  | sftp-connector         | 2.0.0-SNAPSHOT |
org.apache.sqoop.connector.sftp.SftpConnector        | TO
|
| 2  | kite-connector         | 2.0.0-SNAPSHOT |
org.apache.sqoop.connector.kite.KiteConnector        | FROM/TO
|
| 3  | kafka-connector        | 2.0.0-SNAPSHOT |
org.apache.sqoop.connector.kafka.KafkaConnector      | TO
|
| 4  | hdfs-connector         | 2.0.0-SNAPSHOT |
org.apache.sqoop.connector.hdfs.HdfsConnector        | FROM/TO
|
| 5  | generic-jdbc-connector | 2.0.0-SNAPSHOT |
org.apache.sqoop.connector.jdbc.GenericJdbcConnector | FROM/TO
|
+----+------------------------+---------------+----------------------
------------------------------+---------------------+
# create one HDFS links hdfs1
sqoop:000> create link -c 4
sqoop:000> show link
+----+-------+--------------+----------------+---------+ | Id | Name  | Connector Id | Connector Name | Enabled |
+----+-------+--------------+----------------+---------+
| 1  | hdfs1 | 4            | hdfs-connector | true    |
+----+-------+--------------+----------------+---------+
```

Take four testing users test1, test2, test3 and test4 as for example:

Test1 has privileges as followings:

Read action privilege on connector all

All action privilege on link id 1

Create a new link hdfs2

Test2 has privileges as followings:

Read action privilege on connector all

Read action privilege on link id 1

Read action privilege on job all

Test3 has privileges as followings:

Read action privilege on connector all

Read action privilege on link id 3

All action privilege on job id 1

Test4 has no privilege on any resource.

```
# sqoop2 as the running sqoop shell
$ kinit sqoop2
$ ./sqoop.sh client
# Grant privileges to test1 user
sqoop:000> create role –r test1
sqoop:000> grant role -r test1 --principal-type group --principal
test1
sqoop:000> grant privilege -a read --resource-type connector --
resource all --principal-type role --principal test1
sqoop:000> grant privilege -a read --resource-type link --resource 1 -
-principal-type role --principal test1
sqoop:000> show privilege --principal-type role --principal test1
+--------+---------------+---------------+------------+
| Action | Resource Name | Resource Type | With Grant |
```

```
+--------+---------------+---------------+------------+
| READ   | all           | CONNECTOR     | false      |
| READ   | 1             | LINK          | false      |
+--------+---------------+---------------+------------+
# Grant privileges to test2 user
sqoop:000> create role -r test2
sqoop:000> grant role -r test2 --principal-type group --principal
test2
sqoop:000> grant privilege -a read --resource-type connector --
resource all --principal-type role --principal test2
sqoop:000> grant privilege -a read --resource-type link --resource 1 -
-principal-type role --principal test2
sqoop:000> grant privilege -a read --resource-type job --resource all
--principal-type role --principal test2
sqoop:000> show privilege --principal-type role --principal test2
+--------+---------------+---------------+------------+
| Action | Resource Name | Resource Type | With Grant |
+--------+---------------+---------------+------------+
| READ   | all           | CONNECTOR     | false      |
| READ   | all           | JOB           | false      |
| READ   | 1             | LINK          | false      |
+--------+---------------+---------------+------------+
# Grant privileges to test3 user
sqoop:000> create role -r test3
sqoop:000> grant role -r test3 --principal-type group --principal
test3
sqoop:000> grant privilege -a read --resource-type connector --
resource all --principal-type role --principal test3
sqoop:000> grant privilege -a all --resource-type job --resource 1 --
principal-type role --principal test3
sqoop:000> show privilege --principal-type role --principal test3
+--------+---------------+---------------+------------+
```

```
| Action | Resource Name | Resource Type | With Grant |
+--------+---------------+---------------+------------+
| READ   | all           | CONNECTOR     | false      |
| ALL    | 1             | JOB           | false      |
+--------+---------------+---------------+------------+
# test4 has no privileges
sqoop:000> create role -r test4
sqoop:000> grant role -r test4 --principal-type group --principal
test4
sqoop:000> show privilege --principal-type role --principal test4
+--------+---------------+---------------+------------+
| Action | Resource Name | Resource Type | With Grant |
+--------+---------------+---------------+------------+
+--------+---------------+---------------+------------+
```

Test1 user runs following command:

```
# test1 as the running sqoop shell
$ kinit test1
$ ./sqoop.sh client
# test1 can show privilege on itself
sqoop:000> show privilege --principal-type role --principal test1
+--------+---------------+---------------+------------+
| Action | Resource Name | Resource Type | With Grant |
+--------+---------------+---------------+------------+
| READ   | all           | CONNECTOR     | false      |
| READ   | 1             | LINK          | false      |
+--------+---------------+---------------+------------+
# test1 role can’t allow showing privileges on test2 role
sqoop:000> show privilege --principal-type role --principal test2
Caused by: Exception: java.lang.Throwable Message: Access denied to
test1. Server Stacktrace:
```

```
org.apache.sentry.provider.db.SentryAccessDeniedException: Access
denied to test1
# test1 can show all connector
sqoop:000> show connector
+----+------------------------+----------------+----------------------
--------------------------------+----------------------+
| Id |          Name          |    Version     |
Class                          | Supported Directions |
+----+------------------------+----------------+----------------------
--------------------------------+----------------------+
| 1  | sftp-connector         | 2.0.0-SNAPSHOT |
org.apache.sqoop.connector.sftp.SftpConnector        | TO
|
| 2  | kite-connector         | 2.0.0-SNAPSHOT |
org.apache.sqoop.connector.kite.KiteConnector        | FROM/TO
|
| 3  | kafka-connector        | 2.0.0-SNAPSHOT |
org.apache.sqoop.connector.kafka.KafkaConnector      | TO
|
| 4  | hdfs-connector         | 2.0.0-SNAPSHOT |
org.apache.sqoop.connector.hdfs.HdfsConnector        | FROM/TO
|
| 5  | generic-jdbc-connector | 2.0.0-SNAPSHOT |
org.apache.sqoop.connector.jdbc.GenericJdbcConnector | FROM/TO
|
+----+------------------------+----------------+----------------------
--------------------------------+----------------------+
# test1 can show link id 1
sqoop:000> show link +----+-------+--------------+----------------+---------+
| Id | Name  | Connector Id | Connector Name | Enabled |
+----+-------+--------------+----------------+---------+
| 1  | hdfs1 | 4            | hdfs-connector | true    |
```

```
+----+-------+--------------+----------------+---------+
# test1 can create a new link hdfs2, so the test1 is the owner
# of hdfs2
sqoop:000> create link -c 4
sqoop:000> show link
+----+-------+--------------+----------------+---------+
| Id | Name  | Connector Id | Connector Name | Enabled |
+----+-------+--------------+----------------+---------+
| 1  | hdfs1 | 4            | hdfs-connector | true    |
| 2  | hdfs2 | 4            | hdfs-connector | true    |
+----+-------+--------------+----------------+---------+
# create job 1 using the link id 1 and id 3
sqoop:000> create job -f 1 -t 2
sqoop:000> show job
+----+------+----------------+--------------+---------+
| Id | Name | From Connector | To Connector | Enabled |
+----+------+----------------+--------------+---------+
| 1  | job1 | 4              | 4            | true    |
+----+------+----------------+--------------+---------+
sqoop:000> disable job --jid 1
```

Test2 user runs following command:

```
# test2 as the running sqoop shell
$ kinit test2
$ ./sqoop.sh client
# test2 can show privilege on itself
sqoop:007> show privilege --principal-type role --principal test2
+--------+---------------+---------------+------------+
```

| Action | Resource Name | Resource Type | With Grant |

+--------+---------------+---------------+------------+

| READ | all | CONNECTOR | false |

| READ | all | JOB | false |

| READ | 1 | LINK | false |

+--------+---------------+---------------+------------+ sqoop:007> show role --principal-type group --principal test2

+-----------+

| Role Name |

+-----------+

| test2 |

+-----------+

test2 can’t allow seeing other group belongs to role

sqoop:007> show role --principal-type group --principal test1

Caused by: Exception: java.lang.Throwable Message: Unable to excute

command on sentry server: Access denied to test2. Server Stacktrace:

org.apache.sentry.provider.db.SentryAccessDeniedException: Access

denied to test2

test2 can only show link id 1

sqoop:007> show link

+----+-------+--------------+----------------+---------+

| Id | Name | Connector Id | Connector Name | Enabled |

+----+-------+--------------+----------------+---------+

| 1 | hdfs1 | 4 | hdfs-connector | true |

+----+-------+--------------+----------------+---------+

test2 can show any job

sqoop:007> show job

+----+------+----------------+--------------+---------+

| Id | Name | From Connector | To Connector | Enabled |

+----+------+----------------+--------------+---------+

| 1 | job1 | 4 | 4 | false |

+----+------+----------------+--------------+---------+

test2 can’t update the job id 1

```
sqoop:007> enable job -jid 1
caused by: Exception: java.lang.Throwable Message: User test2 does not
have privileges for : Privilege (Privilege resource: Resource
(Resource name: 1, Resource type: JOB ), Privilege action: WRITE,
Privilege with_grant_option: false )
```

Test3 user runs following command:

```
# test3 as the running sqoop shell
$ kinit test3
$ ./sqoop.sh client
# test3 can show privilege on itself
sqoop:000> show privilege --principal-type role --principal test3
+--------+---------------+---------------+------------+
| Action | Resource Name | Resource Type | With Grant |
+--------+---------------+---------------+------------+
| READ   | all           | CONNECTOR     | false      |
| ALL    | 1             | JOB           | false      |
+--------+---------------+---------------+------------+
# test3 can’t show any link
sqoop:000> show link
+----+-------+--------------+----------------+---------+
| Id | Name  | Connector Id | Connector Name | Enabled |
+----+-------+--------------+----------------+---------+
+----+-------+--------------+----------------+---------+
# test3 has all privilege on job id 1
sqoop:000> show job
+----+------+----------------+--------------+---------+
| Id | Name | From Connector | To Connector | Enabled |
+----+------+----------------+--------------+---------+
| 1  | job1 | 4              | 4            | false   |
```

```
+----+------+---------------+--------------+---------+
sqoop:000> enable job -jid 1
```

Test4 user runs following command:

```
# test4 as the running sqoop shell
$ kinit test4
$ ./sqoop.sh client
# test4 has no privilege
sqoop:000> show privilege --principal-type role --principal test4
+--------+---------------+---------------+------------+
| Action | Resource Name | Resource Type | With Grant |
+--------+---------------+---------------+------------+
+--------+---------------+---------------+------------+
sqoop:000> show connector
+----+------+---------+-------+----------------------+
| Id | Name | Version | Class | Supported Directions |
+----+------+---------+-------+----------------------+
+----+------+---------+-------+----------------------+
sqoop:000> show link
+----+------+--------------+----------------+---------+
| Id | Name | Connector Id | Connector Name | Enabled |
+----+------+--------------+----------------+---------+
+----+------+--------------+----------------+---------+
sqoop:000> show job
+----+------+----------------+--------------+---------+
| Id | Name | From Connector | To Connector | Enabled |
+----+------+----------------+--------------+---------+
+----+------+----------------+--------------+---------+
```

Sqoop2 authorization has ownership on resource object:

When the user creates any resource object like link, job in Sqoop2, he is the owner of the resource. So he has the permanent privilege on that resource until it is deleted. Take the above test case for example. Test1 user has created a link hdfs2, so test1 is the owner of hdfs2. He can show and update link hdfs2 even if the admin user doesn't grant privilege to test1. Test2 can show link hdfs1 because admin user grant read privilege to him.

What if the admin user revoking privilege from test1 and test2, the result shows as the following:

```
#admin user login
# sqoop2 as the running sqoop shell
$ kinit sqoop2
$ ./sqoop.sh client
sqoop:000> show privilege --principal-type role --principal test1
+--------+---------------+---------------+------------+
| Action | Resource Name | Resource Type | With Grant |
+--------+---------------+---------------+------------+
| READ   | all           | CONNECTOR     | false      |
| READ   | 1             | LINK          | false      |
+--------+---------------+---------------+------------+
sqoop:000> revoke privilege -a read --resource-type link --resource 1 --principal-type role --principal test1
sqoop:000> revoke privilege -a read --resource-type link --resource 2 --principal-type role --principal test1
sqoop:000> show privilege --principal-type role --principal test1
+--------+---------------+---------------+------------+
| Action | Resource Name | Resource Type | With Grant |
+--------+---------------+---------------+------------+
| READ   | all           | CONNECTOR     | false      |
+--------+---------------+---------------+------------+
sqoop:000> revoke privilege -a read --resource-type link --resource 1 --principal-type role --principal test2
sqoop:000> revoke privilege -a read --resource-type link --resource 2 --principal-type role --principal test2
sqoop:000> show privilege --principal-type role --principal test2
+--------+---------------+---------------+------------+
| Action | Resource Name | Resource Type | With Grant |
+--------+---------------+---------------+------------+
| READ   | all           | CONNECTOR     | false      |
| READ   | all           | JOB           | false      |
```

```
+--------+---------------+---------------+------------+
# test1 and test2 both have no privilege on link hdfs1 and hdfs2
# test1 login
$ kinit test1
$ ./sqoop.sh client
sqoop:000> show link
+----+-------+--------------+----------------+---------+
| Id | Name  | Connector Id | Connector Name | Enabled |
+----+-------+--------------+----------------+---------+
| 2  | hdfs2 | 4            | hdfs-connector | true    |
# test2 login
$ kinit test2
$ ./sqoop.sh client
sqoop:000> show link
+----+------+--------------+----------------+---------+
| Id | Name | Connector Id | Connector Name | Enabled |
+----+------+--------------+----------------+---------+
+----+------+--------------+----------------+---------+
```