Sentry调研

|  |  |  |  |
| --- | --- | --- | --- |
| 日期 | 版本 | 修订者 | 修订描述 |
| 2018.05.08 | v1 | 林 清 | 初稿 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

[1. Sentry简介 2](#_Toc513642805)

[1.1 架构 2](#_Toc513642806)

[1.2 Sentry的特点 2](#_Toc513642807)

[1.3 Sentry的权限模型 2](#_Toc513642808)

[1.4 为用户授权 3](#_Toc513642809)

[1.5 Sentry对内存的要求 3](#_Toc513642810)

[2. 安装部署Sentry 4](#_Toc513642811)

[2.1 部署前提 4](#_Toc513642812)

[2.2 安装 4](#_Toc513642813)

[3. 权限测试 5](#_Toc513642814)

[3.1 创建测试用例 5](#_Toc513642815)

[3.2 分库访问限制 5](#_Toc513642816)

[3.2.1 创建表 6](#_Toc513642817)

[3.2.2 插入记录 6](#_Toc513642818)

[3.2.3 查找记录 6](#_Toc513642819)

[3.2.4 删除表 7](#_Toc513642820)

[4. grant/revoke语句 7](#_Toc513642821)

[4.1 grant语句 7](#_Toc513642822)

[4.2 revoke语句 7](#_Toc513642823)

[5. 参考文档 7](#_Toc513642824)

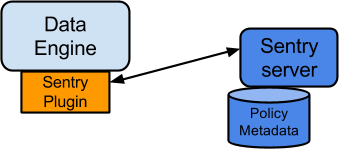
# Sentry简介

Apache Sentry是一个基于角色的授权模块。Sentry提供了在Hadoop集群上控制和实施细粒度级别权限的功能。Sentry目前可以与Apache Hive，Hive Metastore / HCatalog，Apache Solr，Impala和HDFS（限于Hive表数据）一起使用。

## 架构

Sentry主要包括以下三个模块：Sentry Server、Data Engine、Sentry Plugin。

* Sentry Server：管理authorization metadata，提供RPC接口对元数据进行查询以及其他操作。CDH 5.13之后的版本支持部署多个，实现高可用。
* Data Engine：Data Engine指Hive和Impala等需要访问数据的服务。这些服务加载Sentry插件，并拦截所有访问资源的Client请求并将其路由到Sentry插件进行验证。
* Sentry Plugin：它提供了操作Sentry Server中authorization metadata的接口，并根据获取的授权策略对Client进行授权。



## Sentry的特点

* **统一用户验证、用户组映射**：Sentry仅负责为Client授权，而校验功能Impala可以通过Kerberos。Sentry工作时Group信息的来源于Hadoop一致，即可以来自OpenLDAP。
* **统一授权**：在Sentry中一旦定义授权规则，那么在所有组件接入Sentry的组件中都应当遵循该规则。
* **RBAC**：Sentry的授权机制基于“角色的访问控制”。

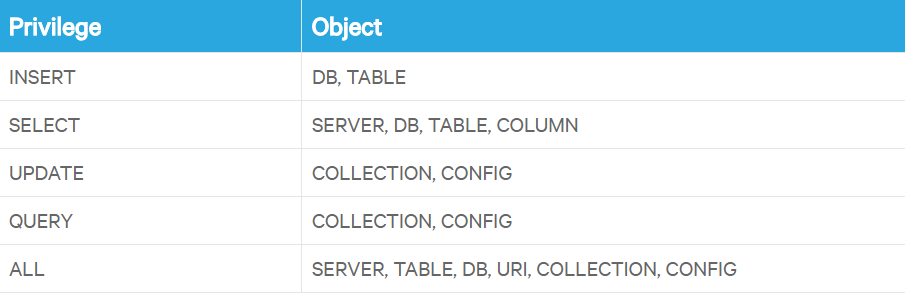
## Sentry的权限模型

Sentry的权限模型和普通关系型数据库类似。当配置Impala、Hive对接Sentry之后，可使用Grant语句为配置用户配置权限。

权限模型中有如下概念：

* Object: 一个对象是受Sentry授权规则保护的实体，可以是server, database, table, URI, collection, config
* Role： 访问特定对象的一组规则
* User：表示通过验证系统验证的实体，可以是一个客户端或者一个用户
* Group：表示一组关联的用户，当前Sentry支持从HDFS获取用户Group信息，或者直接从本地系统获取用户信息

下面的图中列出了可以赋给Object的所有权限，其中和Impala相关的Object有：SERVER（服务器），DB（数据库），TABLE（表），对应的可赋权限包括：INSERT，SELECT，ALL。



用户的权限是向下延伸的，比如Role在某个DB中有Insert权限，那么对所有Table也有Insert权限。

默认情况下允许任何用户执行： show function, desc function, show locks。

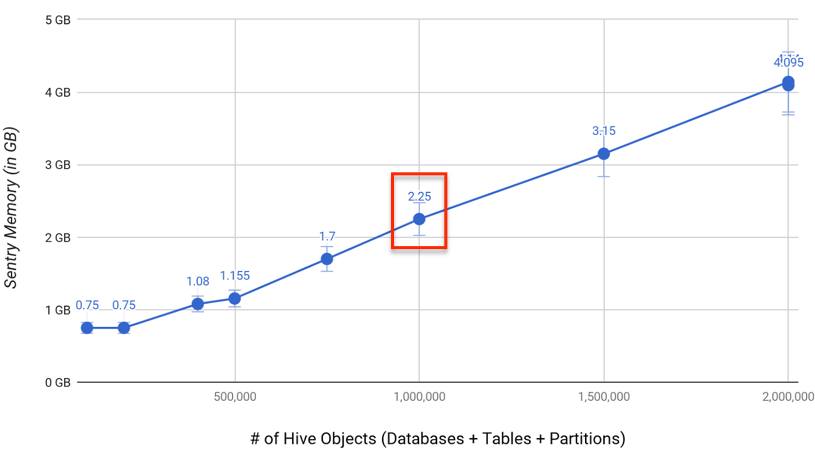
## 为用户授权

当使用Sentry进行授权时，有以下过程：

* 管理员在Sentry中创建若干Role，并且为这些角色定义访问Object的规则（通过Impala Grant语句完成）；
* 管理员将Role和某个Group管理关联（通过Impala Grant语句完成）；
* 管理员将User加入指定Group（操作OpenLDAP完成）；
* 刷新Hadoop用户组映射信息（操作HDFS完成）；

## Sentry对内存的要求

在Sentry中创建大量角色可能会降低Sentry性能，官方建议Sentry中的Roles数目不应该超过5000。同时，每100万个对象需要占用Sentry Server中2.25 GB的内存。



# 安装部署Sentry

## 部署前提

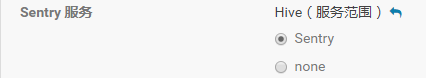
CDH集群启用Kerberos、配置OpenLDAP。

## 安装

* 在CM页面添加Sentry服务，选择部署节点安装和对应的数据库。



* 关闭Hive的配置项hive.sentry.enabled（使用policy文件保存权限规则）；
* 配置Yarn的min.user.id为0（允许1000以下的uid提交Job）；
* 配置Yarn的allowed.system.users，添加HIVE用户；
* 配置HIVE的hadoop.proxyuser.hive.groups，添加hive、hue、sentry（此参数会阻止运行Hive CLI的用户访问Hive Metastore，除非这些用户属于group hue，hive，sentry）；
* 关闭Hive的配置项hive.server2.enable.doAs（模拟 beeline 客户端用户）；
* 配置HIVE、Impala、Hue的Sentry服务范围：



* Sentry配置sentry.service.admin.group，添加hive、Impala、hue、solor、kafka（添加超级权限用户组）；

# 权限测试

## 创建测试用例

登录Kerberos管理员用户，创建以下两个用户：

# impala用户属于Sentry中的超级用户组，有最高的权限

kadmin.local -q "addprinc impala/client" -r CDH.LQING.COM

# 用户的测试数据

kadmin.local -q "addprinc test/client" -r CDH.LQING.COM

将test的用户信息提交到OpenLDAP，并刷新Hadoop的用户映射。

test的用户信息：

**dn: cn=test,ou=Group,dc=ruijie,dc=com**

**objectClass:** posixGroup

**objectClass:** top

**cn:** test

**userPassword:** 123456

**gidNumber:** 9999

**dn: uid=test,ou=People,dc=ruijie,dc=com**

**uid:** test

**cn:** test

**objectClass:** account

**objectClass:** posixAccount

**objectClass:** top

**objectClass:** shadowAccount

**userPassword:** 123456

**loginShell:** /bin/bash

**uidNumber:** 9999

**gidNumber:** 9999

**homeDirectory:** /home/test

**gecos:** test

刷新Hadoop的用户映射

hdfs dfsadmin -refreshUserToGroupsMappings

使用impala-shell登录impala

kinit impala/client

impala-shell –k

虽然Impala用户有最高权限，但是没有为其赋权时，依然没有权限操作数据库。因此，需要使用Impala用户创建一个具有最高权限的角色supervisor，并将其和Impala用户关联。

create role supervisor;

grant all on server to role supervisor;

grant role supervisor to group impala;

## 分库访问限制

使用Impala用户登录创建下面三个数据库

# 创建三个数据库

CREATE DATABASE IF NOT EXISTS db1;

CREATE DATABASE IF NOT EXISTS db2;

CREATE DATABASE IF NOT EXISTS db3;

创建角色role1，并且附上对db1-3的不同权限

# 创建角色，并分别ALL，SELECT，INSERT;

create role role1;

grant ALL on database db1 to role role1;

grant SELECT on database db2 to role role1;

grant INSERT on database db3 to role role1;

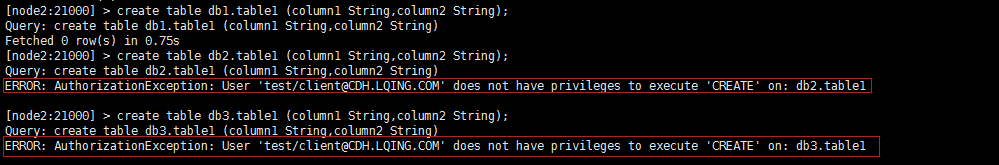
将角色role关联到test组

grant role role1 to group test;

以test用户登录Impala-shell，执行以下操作。

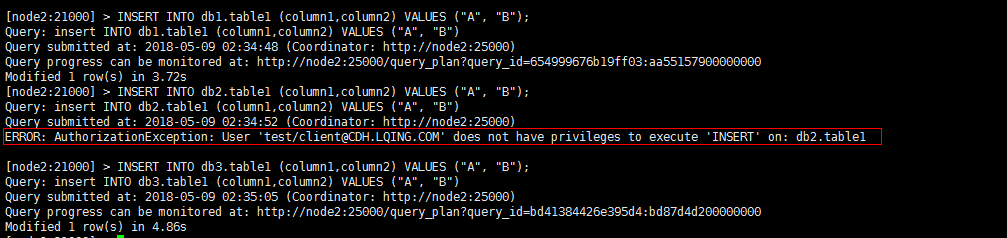
### 创建表

只能在db1创建成功，即建表需要ALL权限；



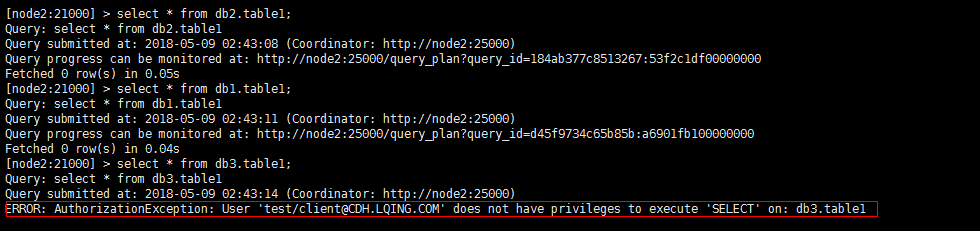
### 插入记录

ALL权限、Insert权限都能完成插入；



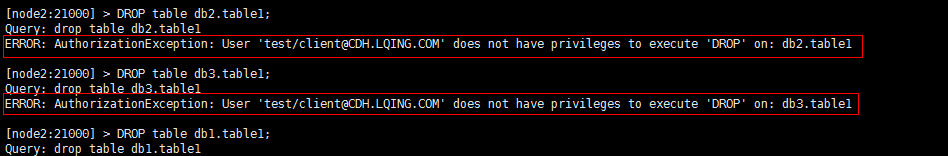
### 查找记录

执行Select语句需要专门的Select权限，Select权限不包含在Insert权限中；



### 删除表

删除表需要ALL权限，Select和Insert都不能删除表；



## 列级权限控制

# 创建角色role2，并且将db1.table2.c1的查询权限授权给他

create role role2;

create table db1.table2 (c1 String,c2 String);

GRANT SELECT(c1) ON TABLE db1.table2 TO ROLE role2;

GRANT SELECT ON TABLE db1.table2 TO ROLE role2;

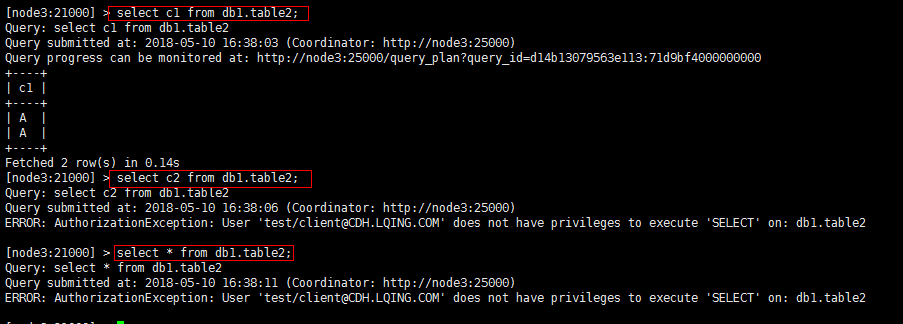
GRANT SELECT ON DATABASE db1 TO ROLE role2;

INSERT INTO db1.table2 (c1,c2) VALUES ("A", "B");

INSERT INTO db1.table2 (c1) VALUES ("A");

grant role role2 to group test;

使用test账号登录，查询表db1.table2：



PS：对于列级别只能授予Select权限，没有ALL和INSERT权限。

# grant/revoke语句

## grant语句

GRANT <权限>

ON <对象类型> <对象名>

TO <Group/Role> <NAME >

PS：

<权限>：ALL、SELECT、INSERT、ROLE

<对象类型>：SERVER、DATABASE、TABLE、COLUMN

## revoke语句

REVOKE <权限>

ON <对象类型> <对象名> FROM <用户>

PS：

# 收回group上分配的角色

REVOKE ROLE <角色名> from GROUP <组名>

# 参考文档

[Apache Impala](https://www.cloudera.com/documentation/enterprise/5-13-x/topics/sg_sentry_overview.html)。

[Impala中不同操作的权限需求](https://www.cloudera.com/documentation/enterprise/5-13-x/topics/cm_sg_sentry_service.html)。