hive权限控制

hue里面用sentry来控制，不知道是因为为没有装Kerberos，失败了， 还是用传统的方法

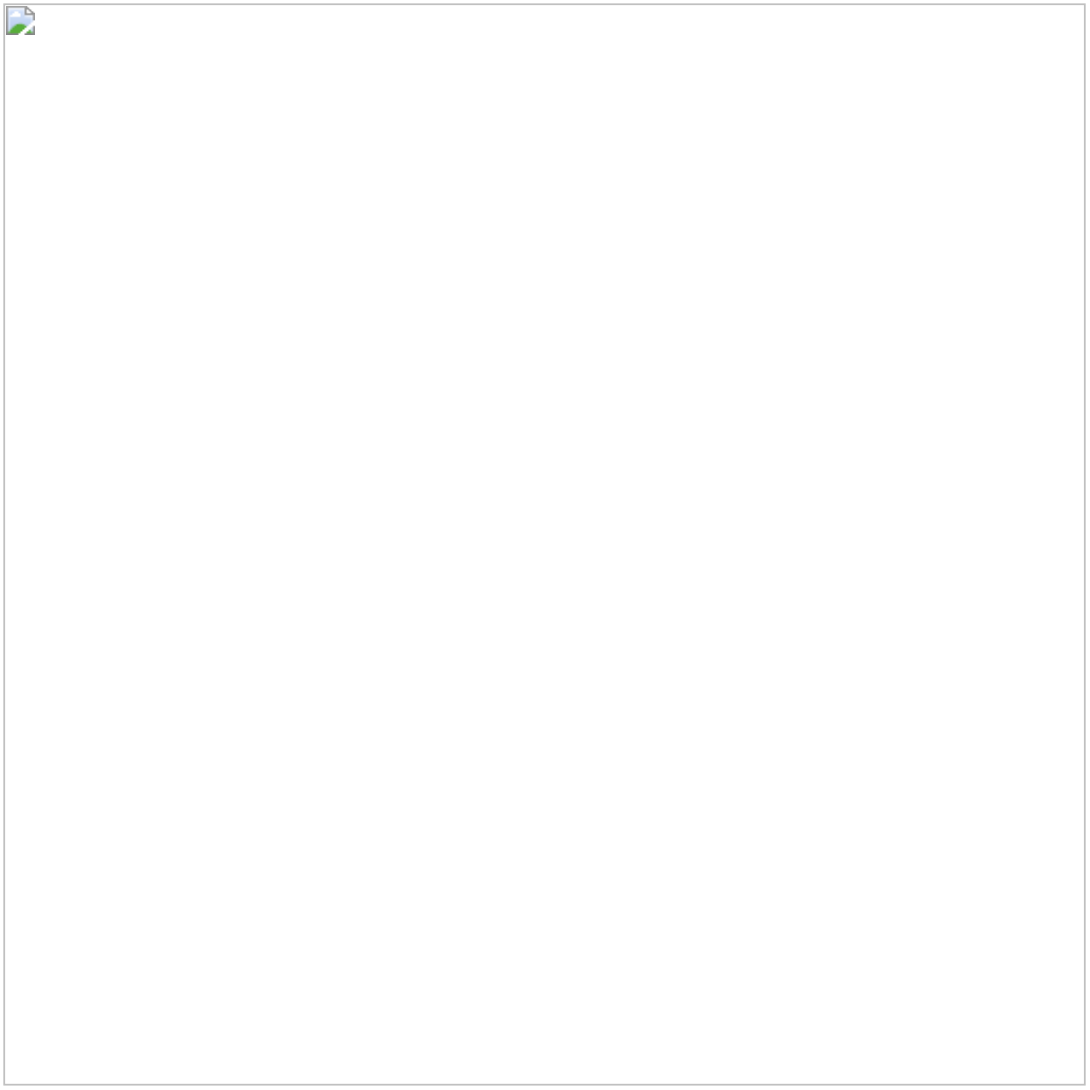
可以通过 beeline 连接进行赋权，或者直接在hue界面也可以
beeline -u "jdbc:hive2://localhost:10000/" -n hive

配置

```
1.开启启身份认证后，任何用户必须被grant privilege才能对实体进行操作。
hive.security.authorization.enabled = true

2.表示创建表时自动赋予一些用户或角色相应的权限
hive.security.authorization.createtable.owner.grants = ALL
hive.security.authorization.createtable.role.grants = admin_role:ALL
hive.security.authorization.createtable.user.grants =
user1,user2:select;user3:create

3.<!-- 假如出现以下错误： Error while compiling statement: FAILED:
SemanticException The current builtin authorization in Hive is incomplete and
disabled. 需要配置下面的属性 -->
hive.security.authorization.task.factory =
org.apache.hadoop.hive.ql.parse.authorization.HiveAuthorizationTaskFactoryImpl
```



角色管理

```
--创建和删除角色
create role role_name;
drop role role_name;
--展示所有roles
show roles
--赋予角色权限
grant select on database db_name to role role_name;
grant select on table t_name to role role_name;
--查看角色权限
show grant role role_name on database db_name;
show grant role role_name on table t_name;
--角色赋予用户
grant role role_name to user user_name
```

```
--回收角色权限
revoke select on database db_name from role role_name;
revoke select on table t_name from role role_name;
--查看某个用户所有角色
show role grant user user_name;
```

超级权限

Hive的权限功能还有一个需要完善的地方，那就是“超级管理员”。

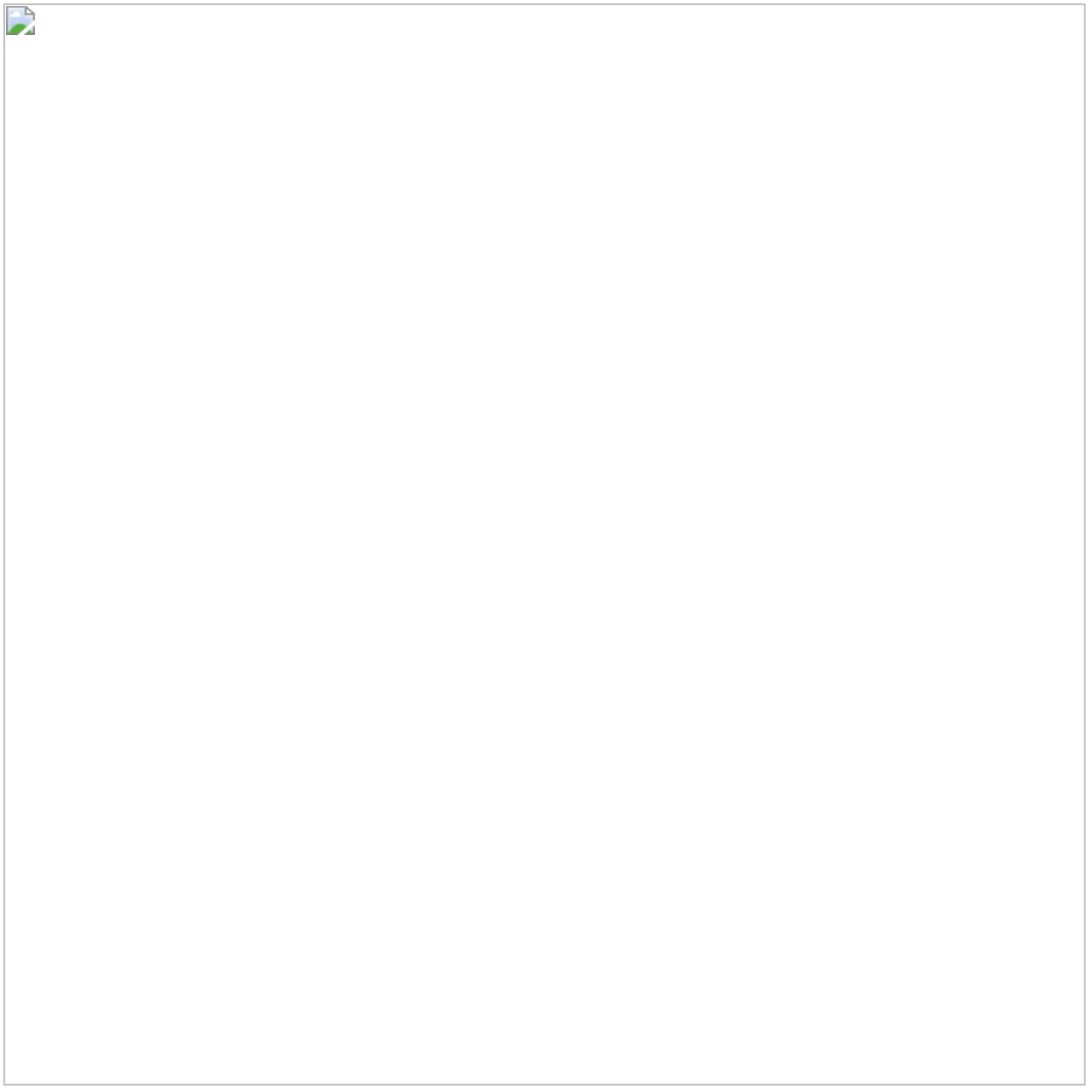
Hive中没有超级管理员，任何用户都可以进行Grant/Revoke操作，为了完善“超级管理员”，必须添加hive.semantic.analyzer.hook配置，并实现自己的权限控制类。

```
hive.semantic.analyzer.hook = com.mycompany.AuthHook
```

1. 编译下面代码(需要导入依赖antlr-runtime-3.4.jar,hive-exec-0.12.0-cdh5.1.2.jar)
2. 打包成jar放置在hive的classpath下(客户端hive shell所在主机的hive-env.sh 中的环境变量：HIVE_AUX_JARS_PATH指向的路径，此配置仅对hive shell生效)

jar包下载地址

1. hive-site.xml中添加参数



2. 最后重启hiveserver。

```
package com.newland;

import org.apache.hadoop.hive.ql.parse.ASTNode;
import org.apache.hadoop.hive.ql.parse.AbstractSemanticAnalyzerHook;
import org.apache.hadoop.hive.ql.parse.HiveParser;
import org.apache.hadoop.hive.ql.parse.HiveSemanticAnalyzerHookContext;
import org.apache.hadoop.hive.ql.parse.SemanticException;
import org.apache.hadoop.hive.ql.session.SessionState;

public class AuthHook extends AbstractSemanticAnalyzerHook {
    private static String[] admin = { "admin", "hive" };

    @Override
    public ASTNode preAnalyze(HiveSemanticAnalyzerHookContext context,
```

```
            ASTNode ast) throws SemanticException {
        switch (ast.getToken().getType()) {
        case HiveParser.TOK_CREATEDATABASE:
        case HiveParser.TOK_DROPDATABASE:
        case HiveParser.TOK_CREATEROLE:
        case HiveParser.TOK_DROPROLE:
        case HiveParser.TOK_GRANT:
        case HiveParser.TOK_REVOKE:
        case HiveParser.TOK_GRANT_ROLE:
        case HiveParser.TOK_REVOKE_ROLE:
            String userName = null;
            if (SessionState.get() != null
                    && SessionState.get().getAuthenticator() != null) {
                userName =
SessionState.get().getAuthenticator().getUserName();
            }
            if (!admin[0].equalsIgnoreCase(userName)
                    && !admin[1].equalsIgnoreCase(userName)) {
                throw new SemanticException(userName
                        + " can't use ADMIN options, except " + admin[0] +
","
                        + admin[1] + ".");
            }
            break;
        default:
            break;
        }
        return ast;
    }

//    public static void main(String[] args) throws SemanticException {
//        String[] admin = { "admin", "root" };
//        String userName = "root";
//        for (String tmp : admin) {
//            System.out.println(tmp);
```

```
//            if (!tmp.equalsIgnoreCase(userName)) {
//                throw new SemanticException(userName
//                        + " can't use ADMIN options, except " + admin[0] +
"," 
//                        + admin[1] + ".");
//            }
//        }
//    }
}
```

权限管理

```
--赋予用户权限
grant [SELECT|...] on [database|table] [db_name|tbl_name ] to user
[username];
grant select(tab_col) on table [tbl_name] to user [username];
--回收用户权限
revoke [ALL|...] on [database|table] [db_name|tbl_name ] from user
[username];
--查看用户权限
show grant user [username] on [database|table] [db_name|tbl_name];
```

HIVE支持以下权限：

```
权限名称
含义
ALL
所有权限
ALTER
允许修改元数据（modify metadata data of object）---表信息数据
UPDATE
允许修改物理数据（modify physical data of object）---实际数据
CREATE
允许进行Create操作
DROP
允许进行DROP操作
```

INDEX
允许建索引（目前还没有实现）
LOCK
当出现并发的使用允许用户进行LOCK和UNLOCK操作
SELECT
允许用户进行SELECT操作
SHOW_DATABASE
允许用户查看可用的数据库

附：

登录hive元数据库，可以发现以下表:

Db_privs:记录了User/Role在DB上的权限

Tbl_privs:记录了User/Role在table上的权限

Tbl_col_privs：记录了User/Role在table column上的权限

Roles：记录了所有创建的role

Role_map：记录了User与Role的对应关系

注意：

admin和hive是超级管理员了，可以给自己赋权，

grant all to user admin; 赋权后就可以 创建数据库和表了。

revoke all from user admin;

查看设置是否生效用set

set hive.security.authorization.enabled