

尚硅谷大数据项目之尚品汇（权限管理）

(作者：尚硅谷研究院)

版本：V4.0.0

第 1 章 Ranger 概述

1.1 什么是 Ranger

Apache Ranger 是一个 Hadoop 平台上的全方位数据安全框架，它可以为整个 Hadoop 生态系统提供全面的安全管理。

随着企业业务的拓展，企业可能在多用户环境中运行多个工作任务，这就需要一个可以对安全策略进行集中管理，配置和监控用户访问的框架。Ranger 由此产生！

Ranger 的官网：<https://ranger.apache.org/>

1.2 Ranger 的目标

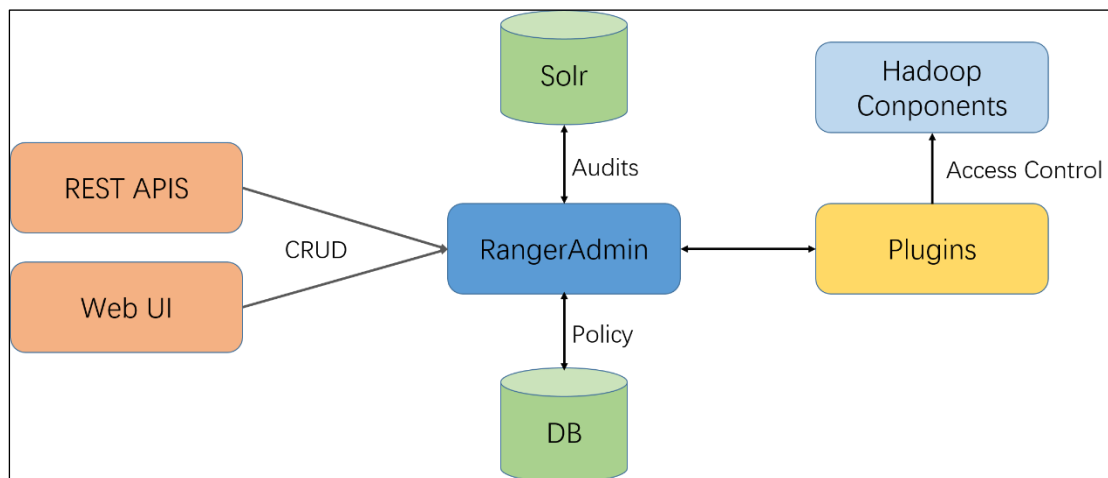
- 允许用户使用 UI 或 REST API 对所有和安全相关的任务进行集中化的管理
- 允许用户使用一个管理工具对操作 Hadoop 体系中的组件和工具的行为进行细粒度的授权
- 支持 Hadoop 体系中各个组件的授权认证标准
- 增强了对不同业务场景需求的授权方法支持，例如基于角色的授权或基于属性的授权
- 支持对 Hadoop 组件所有涉及安全的审计行为的集中化管理

1.3 Ranger 支持的框架

- Apache Hadoop
- Apache Hive
- Apache HBase
- Apache Storm
- Apache Knox
- Apache Solr
- Apache Kafka
- YARN

- NIFI

1.4 Ranger 的架构



1.5 Ranger 的工作原理

Ranger 的核心是 Web 应用程序，也称为 RangerAdmin 模块，此模块由管理策略，审计日志和报告等三部分组成。

管理员角色的用户可以通过 RangerAdmin 提供的 web 界面或 REST APIs 来定制安全策略。这些策略会由 Ranger 提供的轻量级的针对不同 Hadoop 体系中组件的插件来执行。插件会在 Hadoop 的不同组件的核心进程启动后，启动对应的插件进程来进行安全管理！

第 2 章 Ranger 的安装

2.1 环境说明

Ranger2.0 要求对应的 Hadoop 为 3.x 以上，Hive 为 3.x 以上版本，JDK 为 1.8 以上版本。Hadoop 及 Hive 等需开启用户认证功能，本文基于开启 Kerberos 安全认证的 Hadoop 和 Hive 环境。

注：本文中所涉及的 Ranger 相关组件均安装在 hadoop102 节点。

2.2 创建系统用户和 Kerberos 主体

Ranger 的启动和运行需使用特定的用户，故须在 Ranger 所在节点创建所需系统用户并在 Kerberos 中创建所需主体。

1.创建 ranger 系统用户

```
[root@hadoop102 ~]# useradd ranger -G hadoop
[root@hadoop102 ~]# echo ranger | passwd --stdin ranger
```

2.检查 HTTP 主体是否正常（该主体在 Hadoop 开启 Kerberos 时已创建）

1) 使用 keytab 文件认证 HTTP 主体

```
[root@hadoop102 ~]# kinit -kt /etc/security/keytab/spnego.service.keytab HTTP/hadoop102@EXAMPLE.COM
```

2) 查看认证状态，如下图所示，即为正常

```
[root@hadoop102 ~]# klist
[root@hadoop102 logs]# klist
Ticket cache: FILE:/tmp/krb5cc_0
Default principal: HTTP/hadoop102@EXAMPLE.COM

Valid starting    Expires          Service principal
2021-04-10T14:17:01  2021-04-11T14:17:01  krbtgt/EXAMPLE.COM@EXAMPLE.COM
```

3) 注销认证

```
[root@hadoop102 ~]# kdestroy
```

3.创建 rangeradmin 主体

1) 创建主体

```
[root@hadoop102 ~]# kadmin -padmin/admin -wadmin -q"addprinc -randkey rangeradmin/hadoop102"
```

2) 生成 keytab 文件

```
[root@hadoop102 ~]# kadmin -padmin/admin -wadmin -q"xst -k /etc/security/keytab/rangeradmin.keytab rangeradmin/hadoop102"
```

3) 修改 keytab 文件所有者

```
[root@hadoop102 ~]# chown ranger:ranger /etc/security/keytab/rangeradmin.keytab
```

4.创建 rangerlookup 主体

1) 创建主体

```
[root@hadoop102 ~]# kadmin -padmin/admin -wadmin -q"addprinc -randkey rangerlookup/hadoop102"
```

2) 生成 keytab 文件

```
[root@hadoop102 ~]# kadmin -padmin/admin -wadmin -q"xst -k /etc/security/keytab/rangerlookup.keytab rangerlookup/hadoop102"
```

3) 修改 keytab 文件所有者

```
[root@hadoop102 ~]# chown ranger:ranger /etc/security/keytab/rangerlookup.keytab
```

5.创建 rangerusersync 主体

1) 创建主体

```
[root@hadoop102 ~]# kadmin -padmin/admin -wadmin -q"addprinc -randkey rangerusersync/hadoop102"
```

2) 生成 keytab 文件

```
[root@hadoop102 ~]# kadmin -padmin/admin -wadmin -q"xst -k
```

```
/etc/security/keytab/rangerusersync.keytab  
rangerusersync/hadoop102"
```

3) 修改 keytab 文件所有者

```
[root@hadoop102 ~]# chown ranger:ranger  
/etc/security/keytab/rangerusersync.keytab
```

2.2 安装 RangerAdmin

2.2.1 数据库环境准备

(1) 登录 MySQL

```
[root@hadoop102 ~]# mysql -uroot -p000000
```

(2) 在 MySQL 数据库中创建 Ranger 存储数据的数据库

```
mysql> create database ranger;
```

(3) 更改 mysql 密码策略，为了可以采用比较简单的密码

```
mysql> set global validate_password_length=4;  
mysql> set global validate_password_policy=0;
```

(4) 创建用户

```
mysql> grant all privileges on ranger.* to ranger@'%'  
identified by 'ranger';
```

2.2.2 安装 RangerAdmin

1. 在 hadoop102 的 /opt/module 路径上创建一个 ranger

```
[root@hadoop102 ~]# mkdir /opt/module/ranger
```

2. 解压软件

```
[root@hadoop102 software]# tar -zxvf ranger-2.0.0-admin.tar.gz  
-C /opt/module/ranger
```

3. 进入 /opt/module/ranger/ranger-2.0.0-admin 路径，对 install.properties 配置

```
[root@hadoop102 ranger-2.0.0-admin]# vim install.properties
```

修改以下配置内容：

```
#mysql 驱动  
SQL_CONNECTOR_JAR=/opt/software/mysql-connector-java-  
5.1.48.jar  
  
#mysql 的主机名和 root 用户的用户名密码  
db_root_user=root  
db_root_password=000000  
db_host=hadoop102  
  
#ranger 需要的数据库名和用户信息，和 2.2.1 创建的信息要一一对应  
db_name=ranger  
db_user=ranger  
db_password=ranger  
  
#Ranger 各组件的 admin 用户密码  
rangerAdmin_password=atguigu123  
rangerTagsync_password=atguigu123
```

```
rangerUsersync_password=atguigu123
keyadmin_password=atguigu123

#ranger 存储审计日志的路径，默认为 solr，这里为了方便暂不设置
audit_store=

#策略管理器的 url,rangeradmin 安装在哪台机器，主机名就为对应的主机名
policymgr_external_url=http://hadoop102:6080

#启动 ranger admin 进程的 linux 用户信息
unix_user=ranger
unix_user_pwd=ranger
unix_group=ranger

#Kerberos 相关配置
spnego_principal=HTTP/hadoop102@EXAMPLE.COM
spnego_keytab=/etc/security/keytab/spnego.service.keytab
admin_principal=rangeradmin/hadoop102@EXAMPLE.COM
admin_keytab=/etc/security/keytab/rangeradmin.keytab
lookup_principal=rangerlookup/hadoop102@EXAMPLE.COM
lookup_keytab=/etc/security/keytab/rangerlookup.keytab
hadoop_conf=/opt/module/hadoop-3.1.3/etc/hadoop
```

4.在/opt/module/ranger/ranger-2.0.0-admin 目录下执行安装脚本

```
[root@hadoop102 ranger-2.0.0-admin]# ./setup.sh
```

出现以下信息，说明安装完成

```
2020-04-30 13:58:18,051 [I] Ranger all admins default password
change request processed successfully..
Installation of Ranger PolicyManager Web Application is
completed.
```

5.修改/opt/module/ranger/ranger-2.0.0-admin/conf/ranger-admin-site.xml 配置文件中的以下属性。

```
[root@hadoop102 ranger-2.0.0-admin]# vim
/opt/module/ranger/ranger-2.0.0-admin/conf/ranger-admin-
site.xml
```

增加如下参数

```
<property>
  <name>ranger.jpa.jdbc.password</name>
  <value>ranger</value>
  <description />
</property>

<property>
  <name>ranger.service.host</name>
  <value>hadoop102</value>
</property>
```

2.2.3 启动 RangerAdmin

1.启动 ranger-admin（以 ranger 用户启动）

```
[root@hadoop102 ranger-2.0.0-admin]# sudo -i -u ranger ranger-
admin start
Starting Apache Ranger Admin Service
```

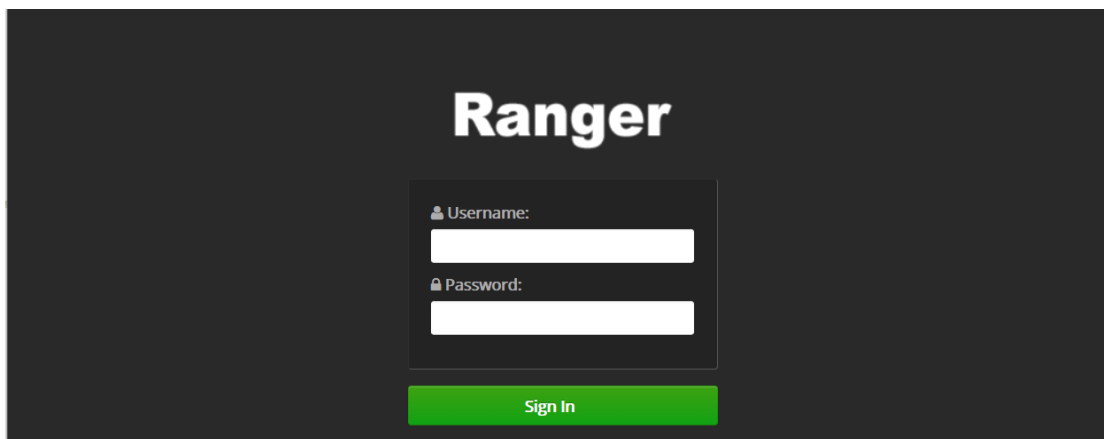
```
Apache Ranger Admin Service with pid 7058 has started.
```

ranger-admin 在安装时已经配设置为开机自启动，因此之后无需再手动启动！

2.查看启动后的进程

```
[root@hadoop102 ranger-2.0.0-admin]# jps
7058 EmbeddedServer
8132 Jps
```

3.访问 Ranger 的 WebUI，地址为：http://hadoop102:6080

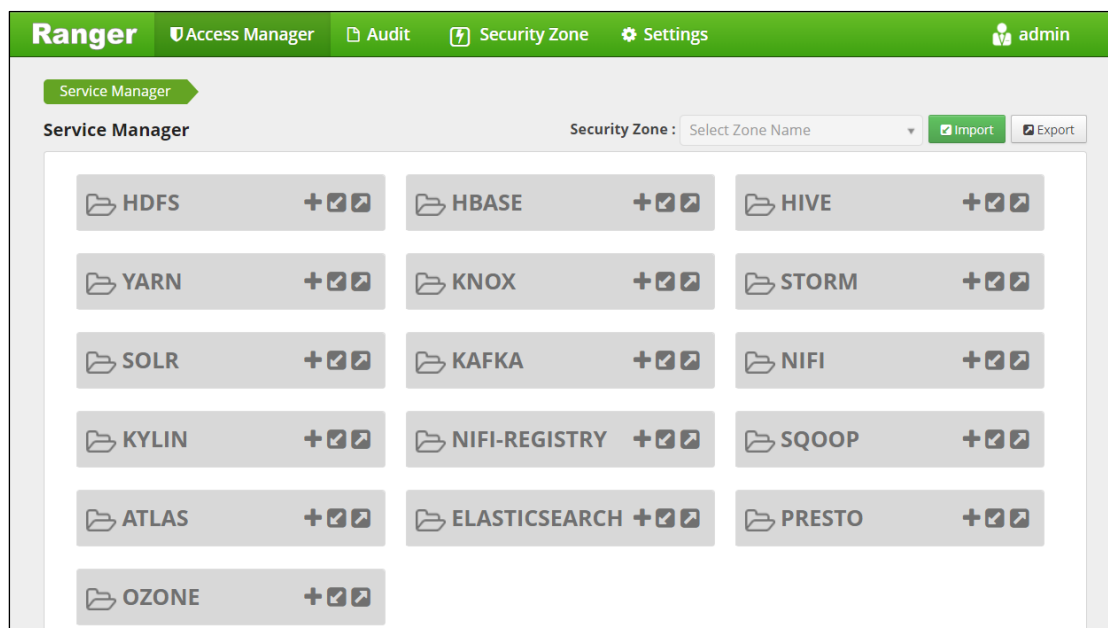


4.停止 ranger（此处不用执行）

```
[root@hadoop102 ranger-2.0.0-admin]# sudo -i -u ranger ranger-admin stop
```

2.2.4 登录 Ranger

默认可以使用用户名：admin，密码为之前配置的 atguigu123 进行登录！登录后界面如下：



第3章 安装 RangerUsersync

3.1 RangerUsersync 简介

RangerUsersync 作为 Ranger 提供的一个管理模块，可以将 Linux 机器上的用户和组信息同步到 RangerAdmin 的数据库中进行管理。

3.2 RangerUsersync 安装

1. 解压软件

```
[root@hadoop102 software]# tar -zxvf ranger-2.0.0-usersync.tar.gz -C /opt/module/ranger/
```

2. 配置软件

在/opt/module/ranger/ranger-2.0.0-usersync 目录下修改以下文件

```
[root@hadoop102 ranger-2.0.0-usersync]# vim install.properties
```

修改以下配置信息

```
#rangeradmin 的 url
POLICY_MGR_URL =http://hadoop102:6080

#同步间隔时间，单位(分钟)
SYNC_INTERVAL = 1

#运行此进程的 linux 用户
unix_user=ranger
unix_group=ranger

#rangerUserSync 用户的密码，参考 rangeradmin 中 install.properties 的配置
rangerUsersync_password=atguigu123

#Kerberos 相关配置
usersync_principal=rangerusersync/hadoop102@EXAMPLE.COM
usersync_keytab=/etc/security/keytab/rangerusersync.keytab
hadoop_conf=/opt/module/hadoop-3.1.3/etc/hadoop
```

3. 在/opt/module/ranger/ranger-2.0.0-usersync 目录下执行安装脚本

```
[root@hadoop102 ranger-2.0.0-usersync]# ./setup.sh
```

出现以下信息，说明安装完成

```
ranger.usersync.policymgr.password has been successfully
created.
Provider
jceks://file/etc/ranger/usersync/conf/rangerusersync.jceks was
updated.
[I] Successfully updated password of rangerusersync user
```

4. 修改/opt/module/ranger/ranger-2.0.0-usersync/conf/ranger-ugsync-site.xml 配置文件中的

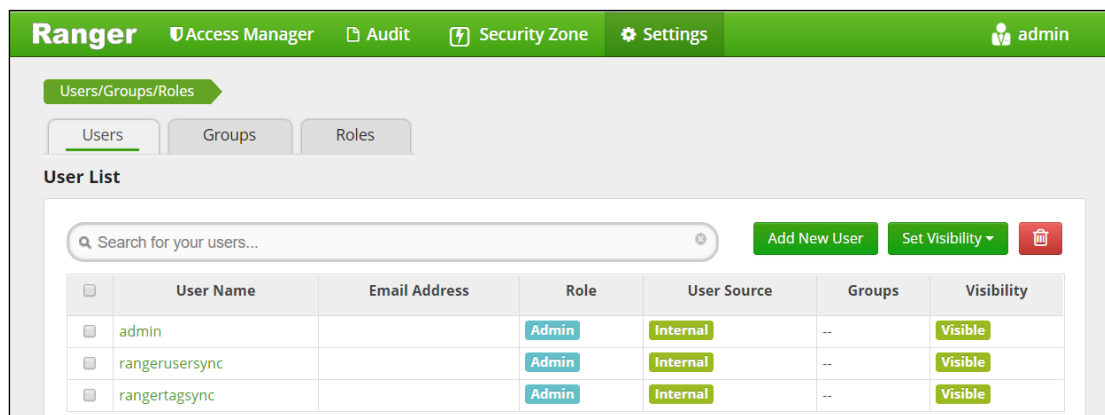
以下参数

```
<property>
```

```
<name>ranger.usersync.enabled</name>
<value>true</value>
</property>
```

3.3 RangerUsersync 启动

1.启动之前，在 ranger admin 的 web-UI 界面，查看用户信息如下：



The screenshot shows the Ranger Admin web-UI. The top navigation bar includes 'Ranger', 'Access Manager', 'Audit', 'Security Zone', 'Settings', and a user profile 'admin'. The main content area is titled 'Users/Groups/Roles' and has tabs for 'Users', 'Groups', and 'Roles'. The 'Users' tab is active, displaying a 'User List' table. Above the table is a search bar and buttons for 'Add New User', 'Set Visibility', and a trash icon. The table lists three users: 'admin', 'rangerusersync', and 'rangertagsync', all with 'Admin' roles and 'Internal' user sources, and are all 'Visible'.

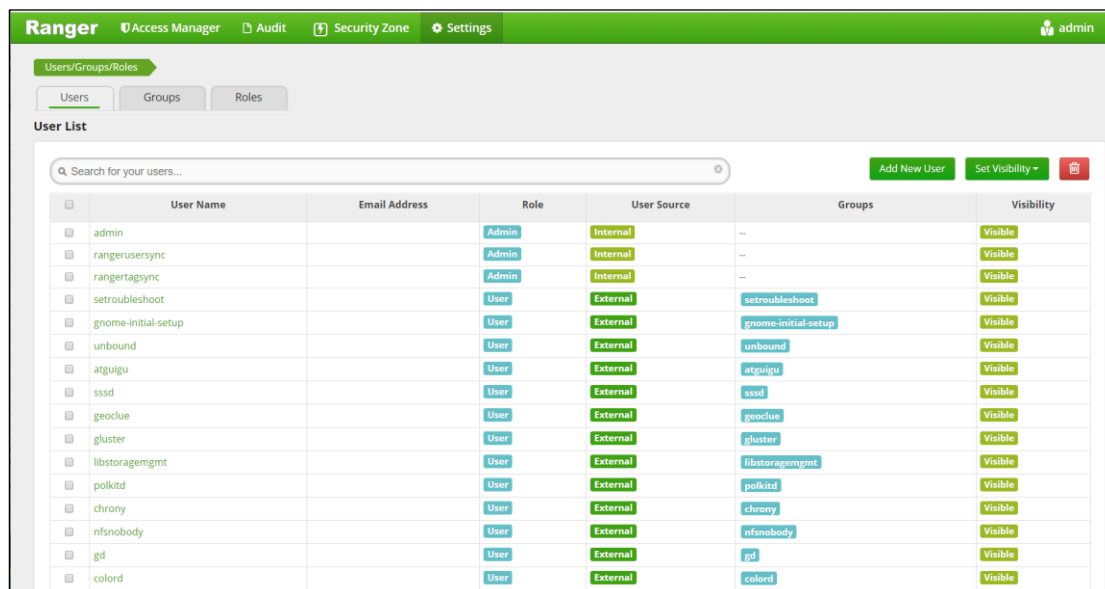
	User Name	Email Address	Role	User Source	Groups	Visibility
<input type="checkbox"/>	admin		Admin	Internal	--	Visible
<input type="checkbox"/>	rangerusersync		Admin	Internal	--	Visible
<input type="checkbox"/>	rangertagsync		Admin	Internal	--	Visible

2.启动 RangerUserSync（使用 ranger 用户启动）

```
[root@hadoop102 ranger-2.0.0-usersync]# sudo -i -u ranger
ranger-usersync start
```

```
Starting Apache Ranger Usersync Service
Apache Ranger Usersync Service with pid 7510 has started.
```

3.启动后，再次查看用户信息：



The screenshot shows the Ranger Admin web-UI after the RangerUserSync service has started. The 'User List' table now contains 20 users. The first three are 'admin', 'rangerusersync', and 'rangertagsync' (Admin roles, Internal sources). The remaining 17 users are 'setroubleshoot', 'gnome-initial-setup', 'unbound', 'atguigu', 'sssd', 'geoclue', 'gluster', 'libstoragemgmt', 'polkitd', 'chrony', 'nfsnobody', 'gd', and 'colord' (all User roles, External sources, and each has a corresponding group listed in the 'Groups' column). All users are 'Visible'.

	User Name	Email Address	Role	User Source	Groups	Visibility
<input type="checkbox"/>	admin		Admin	Internal	--	Visible
<input type="checkbox"/>	rangerusersync		Admin	Internal	--	Visible
<input type="checkbox"/>	rangertagsync		Admin	Internal	--	Visible
<input type="checkbox"/>	setroubleshoot		User	External	setroubleshoot	Visible
<input type="checkbox"/>	gnome-initial-setup		User	External	gnome-initial-setup	Visible
<input type="checkbox"/>	unbound		User	External	unbound	Visible
<input type="checkbox"/>	atguigu		User	External	atguigu	Visible
<input type="checkbox"/>	sssd		User	External	sssd	Visible
<input type="checkbox"/>	geoclue		User	External	geoclue	Visible
<input type="checkbox"/>	gluster		User	External	gluster	Visible
<input type="checkbox"/>	libstoragemgmt		User	External	libstoragemgmt	Visible
<input type="checkbox"/>	polkitd		User	External	polkitd	Visible
<input type="checkbox"/>	chrony		User	External	chrony	Visible
<input type="checkbox"/>	nfsnobody		User	External	nfsnobody	Visible
<input type="checkbox"/>	gd		User	External	gd	Visible
<input type="checkbox"/>	colord		User	External	colord	Visible

说明 ranger-usersync 工作正常！

ranger-usersync 服务也是开机自启动的，因此之后不需要手动启动！

第 4 章 安装 Ranger Hive-plugin

4.1 Ranger Hive-plugin 简介

Ranger Hive-plugin 是 Ranger 对 hive 进行权限管理的插件。需要注意的是, Ranger Hive-plugin 只能对使用 jdbc 方式访问 hive 的请求进行权限管理, hive-cli 并不受限制。

4.2 Ranger Hive-plugin 安装

1. 解压软件

```
[root@hadoop102 software]# tar -zxvf ranger-2.0.0-hive-plugin.tar.gz -C /opt/module/ranger/
```

2. 配置软件

```
[root@hadoop102 ranger-2.0.0-hive-plugin]# vim install.properties
```

修改以下内容

```
#策略管理器的 url 地址
POLICY_MGR_URL=http://hadoop102:6080

#组件名称
REPOSITORY_NAME=hive

#hive 的安装目录
COMPONENT_INSTALL_DIR_NAME=/opt/module/hive

#hive 组件的启动用户
CUSTOM_USER=hive

#hive 组件启动用户所属组
CUSTOM_GROUP=hadoop
```

3. 启用 Ranger Hive-plugin, 在/opt/module/ranger/ranger-2.0.0-hive-plugin 下执行以下命令

```
[root@hadoop102 ranger-2.0.0-hive-plugin]# ./enable-hive-plugin.sh
```

查看\$HIVE_HOME/conf 目录是否出现以下配置文件, 如出现则表示 Hive 插件启用成功。

```
[root@hadoop102 ranger-2.0.0-hive-plugin]# ls $HIVE_HOME/conf | grep -E hiveserver2|ranger
hiveserver2-site.xml
ranger-hive-audit.xml
ranger-hive-security.xml
ranger-policymgr-ssl.xml
ranger-security.xml
```

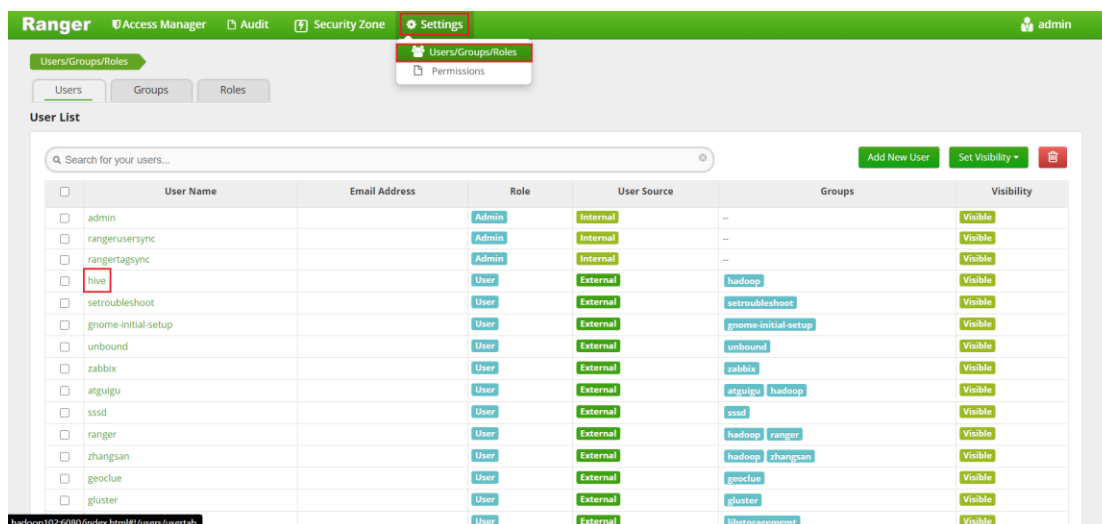
4. 重启 Hiveserver2, 需使用 hive 用户启动。

```
[root@hadoop102 ~]# sudo -i -u hive hiveserver2
```

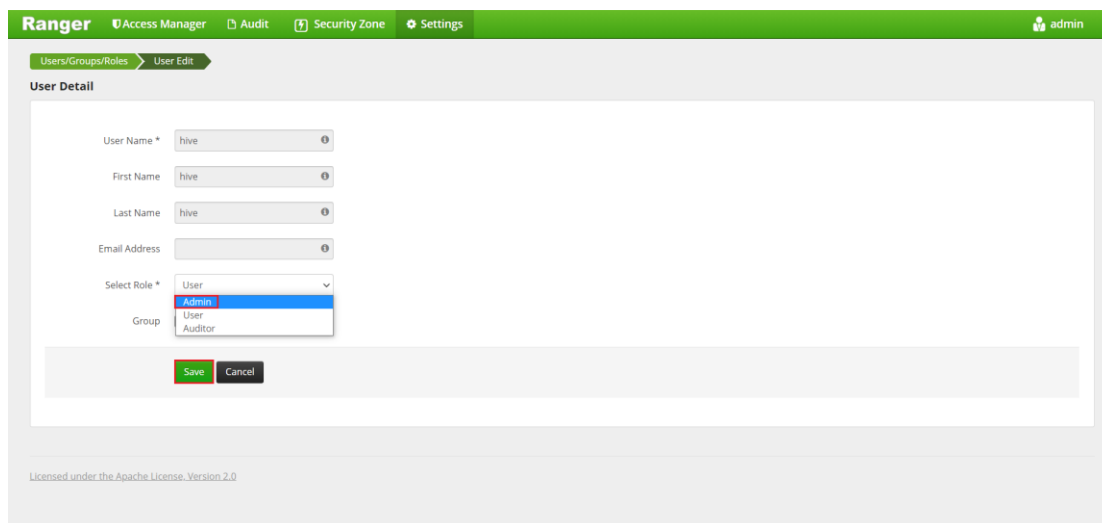
4.3 在 ranger admin 上配置 hive 插件

1.授予 hive 用户在 Ranger 中的 Admin 角色

1) 点击 hive 用户

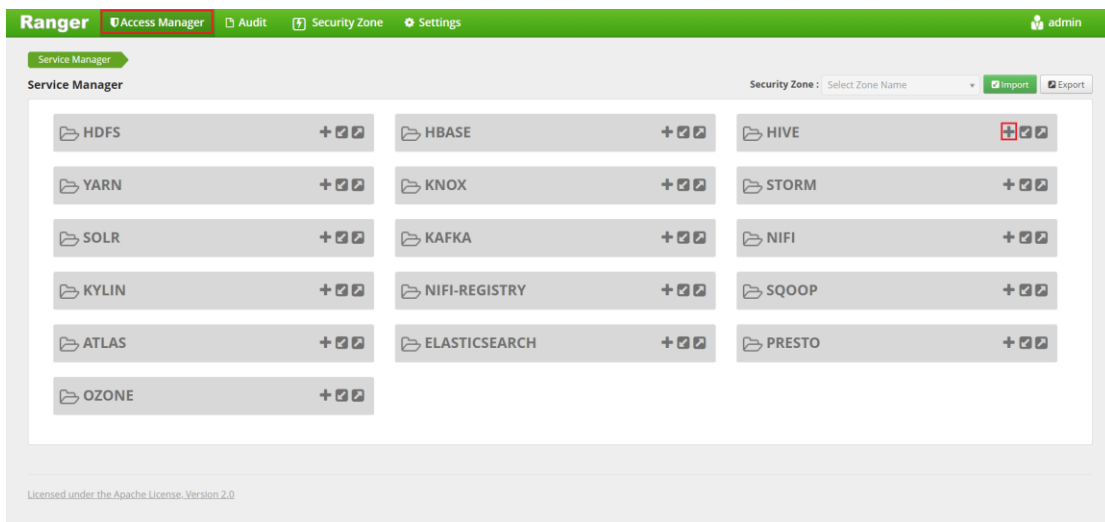


2) 将角色设置为 Admin



2.配置 Hive 插件

1) 点击 Access Manager，添加 Hive Manager。



2) 配置服务详情

Ranger
Access Manager
Audit
Security Zone
Settings
admin

Service Manager
Create Service

Create Service

Service Details :

Service Name *
hive

服务名称，须和hive-plugin的install.properties中的REPOSITORY_NAME参数保持一致

Description

Active Status
☒ Enabled
☐ Disabled

Select Tag Service
Select Tag Service

Config Properties :

Username *
rangerlookup
用户名

Password *
.....
密码，不为空即可

jdbc.driverClassName *
org.apache.hive.jdbc.HiveDriver
Hive JDBC驱动类

jdbc.url *
jdbc:hive2://hadoop102:10000/;
Hive JDBC URL

Common Name for Certificate

Add New Configurations

Name	Value
+	

Test Connection

Add
Cancel

注：

Service Name: hive

Username: rangerlookup

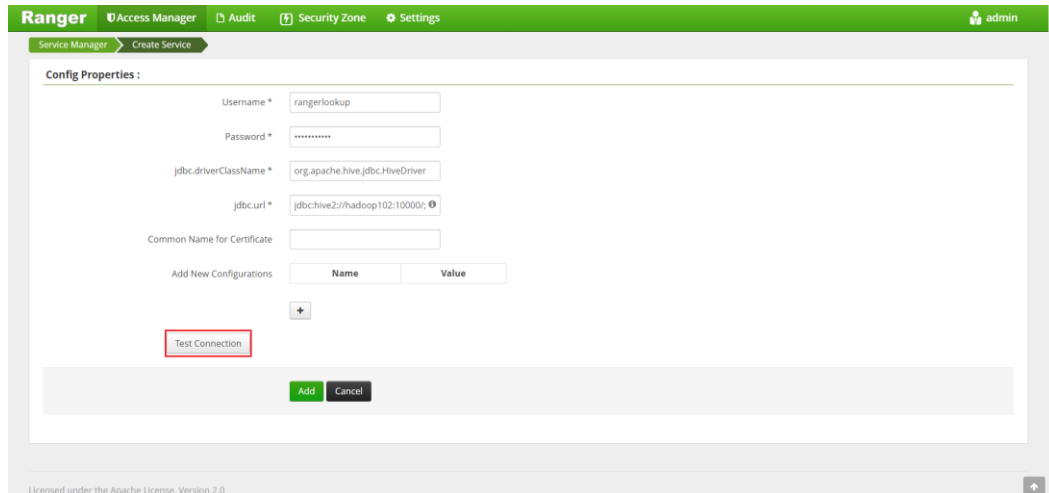
Password: rangerlookup

jdbc.driverClassName: org.apache.hive.jdbc.HiveDriver

jdbc.url: jdbc:hive2://hadoop102:10000/;principal=hive/hadoop102@EXAMPLE.COM

3) 测试连接

点击测试连接



The screenshot shows the 'Create Service' configuration page in the Ranger Service Manager. The 'Config Properties' section contains the following fields:

- Username *: rangerlookup
- Password *: [masked]
- jdbc.driverClassName *: org.apache.hive.jdbc.HiveDriver
- jdbc.url *: jdbc:hive2://hadoop102:10000/
- Common Name for Certificate: [empty]

Below these fields is a table for 'Add New Configurations' with columns 'Name' and 'Value'. A 'Test Connection' button is highlighted with a red box. At the bottom are 'Add' and 'Cancel' buttons.

点击测试连接后会提示连接失败，具体原因是 rangerlookup 用户没有访问 hive 表的权限，这是因为到目前为止，我们还未使用 Ranger 向任何用户赋予任何权限，故此时连接失败为正常现象。

Connection Failed.

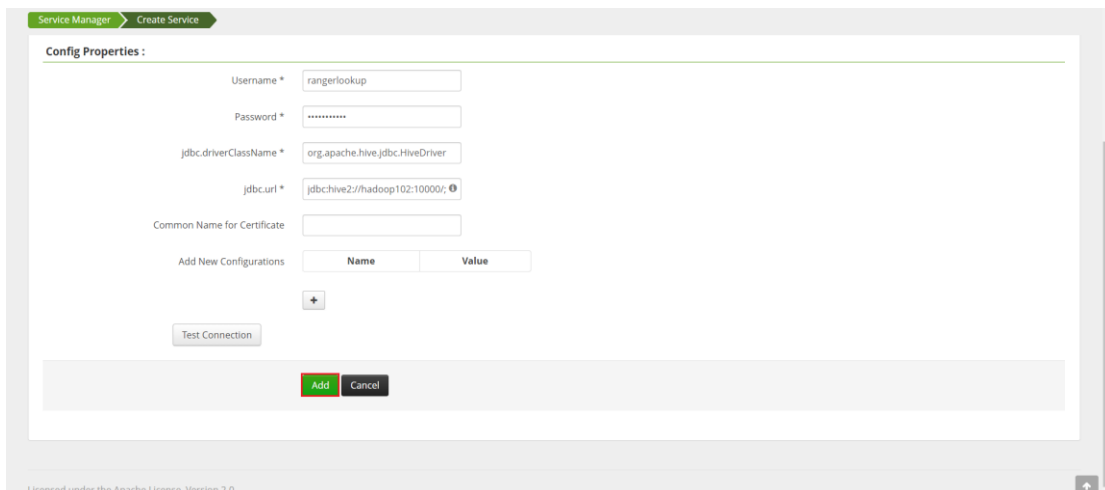
Unable to retrieve any files using given parameters, You can still save the repository and start creating policies, but you would not be able to use autocomplete for resource names. Check ranger_admin.log for more info.

```
org.apache.ranger.plugin.client.HadoopException: Unable to execute SQL [show databases like "*"].  
Unable to execute SQL [show databases like "*"].  
Error while compiling statement: FAILED: HiveAccessControlException Permission denied: user [rangerlookup] does not have [USE] privilege on [Unknown resource!!].  
Permission denied: user [rangerlookup] does not have [USE] privilege on [Unknown resource!!].
```

Show Less..OK

4) 保存 Hive Manager

(1) 点击 Add 按钮



Service Manager > Create Service

Config Properties :

Username * rangerlookup

Password *

jdbc.driverClassName * org.apache.hive.jdbc.HiveDriver

jdbc.url * jdbc:hive2://hadoop102:10000/

Common Name for Certificate

Add New Configurations

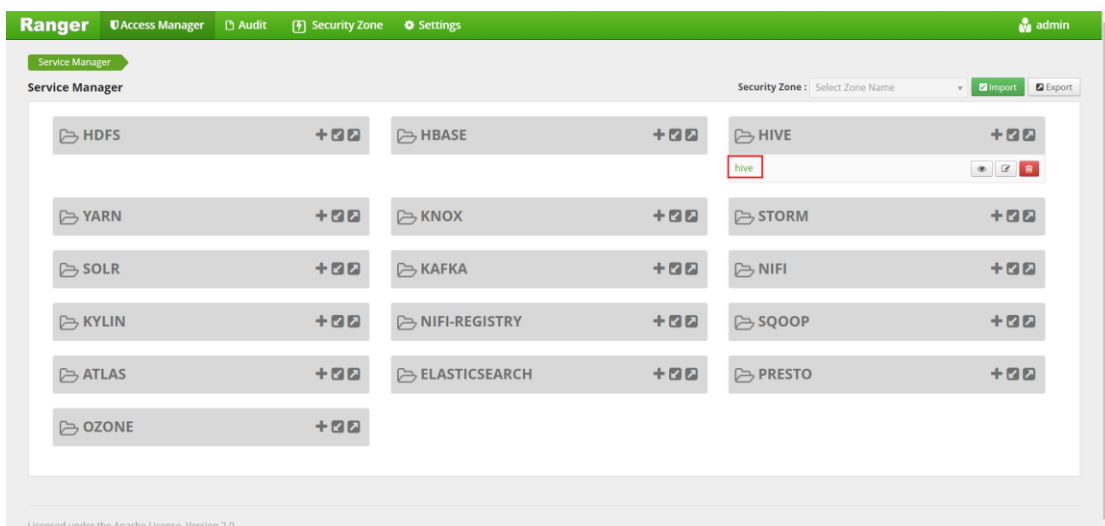
Name	Value

Test Connection

Add Cancel

Licensed under the Apache License, Version 2.0

(2) 点击下图所示 hive 按钮



Ranger Access Manager Audit Security Zone Settings admin

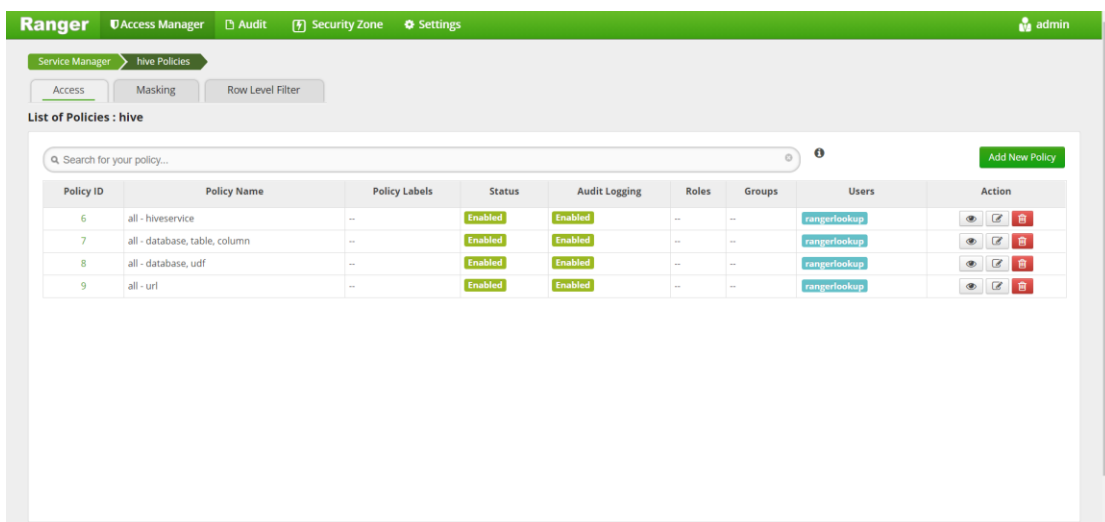
Service Manager

Security Zone: Select Zone Name Import Export

HDFS	HBASE	HIVE
YARN	KNOX	STORM
SOLR	KAFKA	NIFI
KYLIN	NIFI-REGISTRY	SQOOP
ATLAS	ELASTICSEARCH	PRESTO
OZONE		

Licensed under the Apache License, Version 2.0

下图内容表示，目前 rangerlookup 用户已经拥有了 Hive 所有资源的所有权限。



Ranger Access Manager Audit Security Zone Settings admin

Service Manager > hive Policies

Access Masking Row Level Filter

List of Policies : hive

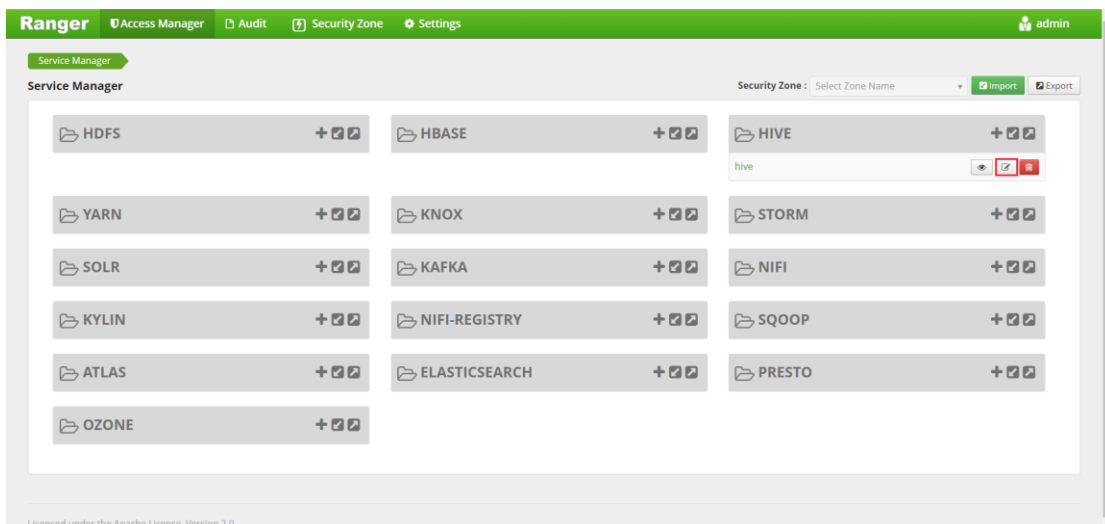
Search for your policy...

Add New Policy

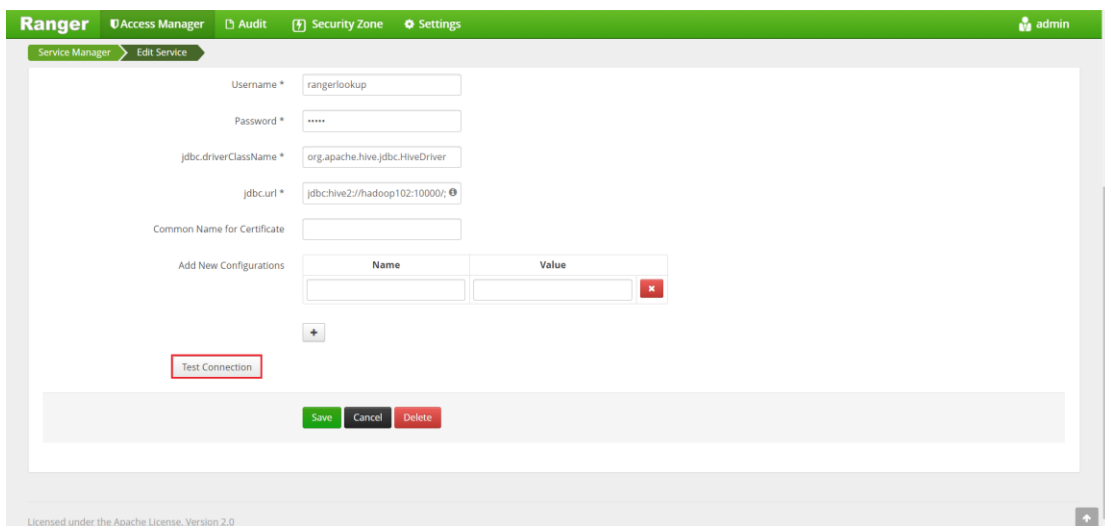
Policy ID	Policy Name	Policy Labels	Status	Audit Logging	Roles	Groups	Users	Action
6	all - hiveservice	--	Enabled	Enabled	--	--	rangerlookup	View Edit Delete
7	all - database, table, column	--	Enabled	Enabled	--	--	rangerlookup	View Edit Delete
8	all - database, udf	--	Enabled	Enabled	--	--	rangerlookup	View Edit Delete
9	all - url	--	Enabled	Enabled	--	--	rangerlookup	View Edit Delete

5) 重新测试连接

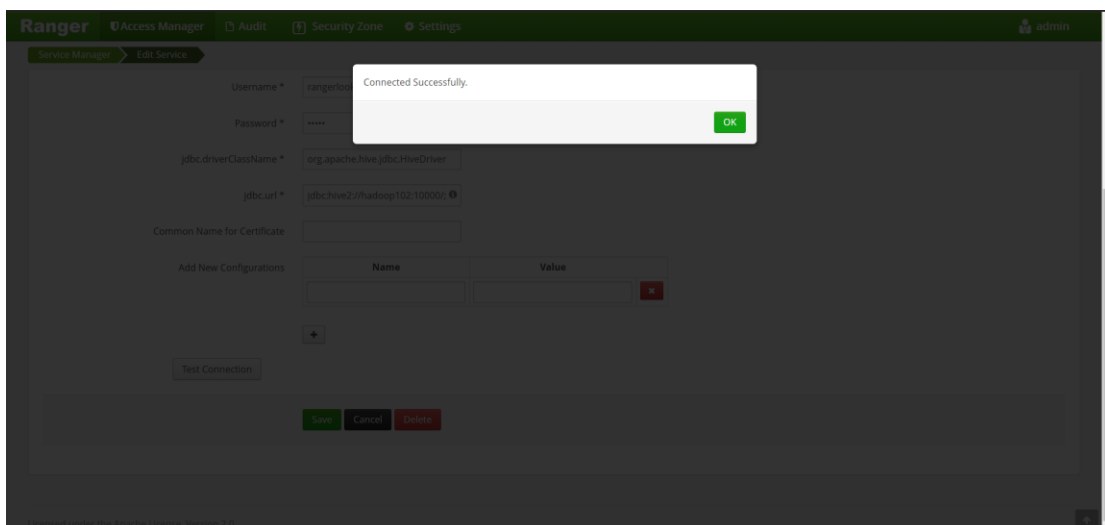
(1) 点击下图编辑按钮



(2) 重新点击 Test Connection



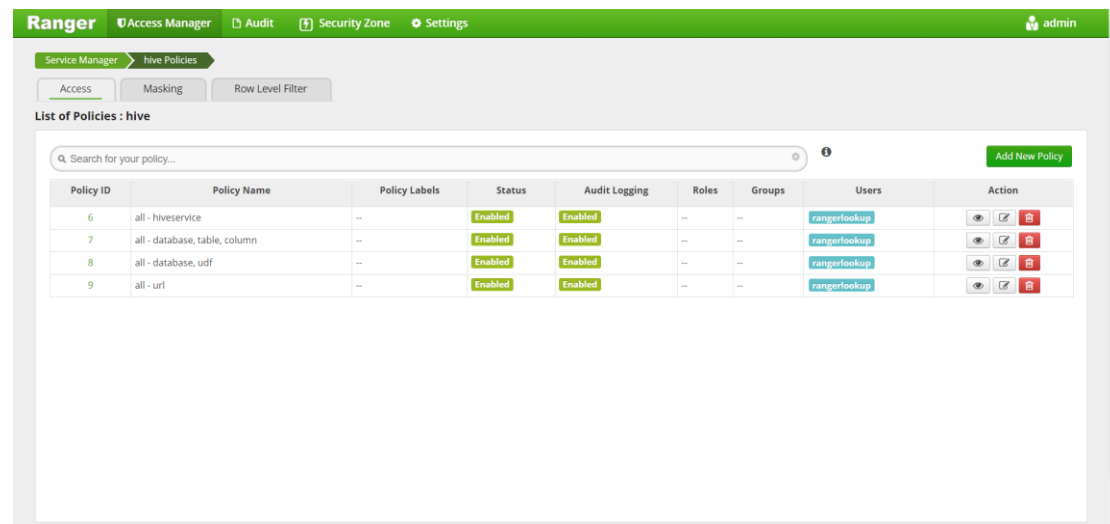
(3) 连接成功



第5章 使用 Ranger 对 Hive 进行权限管理

5.1 权限控制初体验

1. 查看默认的访问策略，此时只有 rangerlookup 用户拥有对所有库、表和函数的访问权限，故理论上其余用户是不能访问任何 Hive 资源的。



2. 验证：使用 atguigu 用户尝试进行认证，认证成功后，使用 beeline 客户端连接 Hiveserver2

1) 使用 atguigu 用户认证，并按照提示输入密码

```
[atguigu@hadoop102 ~]$ kinit atguigu
```

2) 登录 beeline 客户端

```
[atguigu@hadoop102 ~]$ beeline -u "jdbc:hive2://hadoop102:10000/;principal=hive/hadoop102@EXAMPLE.COM"
```

3) 执行以下 sql 语句，验证当前用户为 atguigu

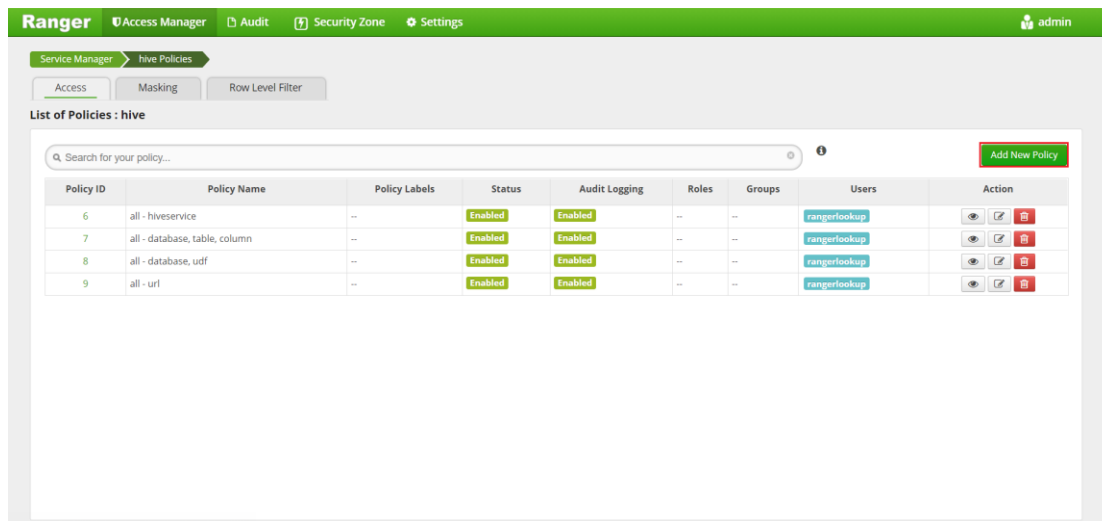
```
0: jdbc:hive2://hadoop102:10000/> select current_user();
INFO : Compiling command(queryId=hive_20210410160551_724802ea-05cc-4b6a-a5e6-f1f6c48cad32): select current_user()
INFO : Concurrency mode is disabled, not creating a lock manager
INFO : Semantic Analysis Completed (retrial = false)
INFO : Returning Hive schema: Schema(fieldSchemas:[FieldSchema(name:_c0, type:string, comment:null)], properties:null)
INFO : Completed compiling command(queryId=hive_20210410160551_724802ea-05cc-4b6a-a5e6-f1f6c48cad32); Time taken: 0.496 seconds
INFO : Concurrency mode is disabled, not creating a lock manager
INFO : Executing command(queryId=hive_20210410160551_724802ea-05cc-4b6a-a5e6-f1f6c48cad32): select current_user()
INFO : Completed executing command(queryId=hive_20210410160551_724802ea-05cc-4b6a-a5e6-f1f6c48cad32); Time taken: 0.0 seconds
INFO : OK
INFO : Concurrency mode is disabled, not creating a lock manager
+-----+
| _c0 |
+-----+
| atguigu |
+-----+
1 row selected (0.556 seconds)
0: jdbc:hive2://hadoop102:10000/>
```

4) 执行 use gmall 语句，结果如图所示，atguigu 用户没有对 gmall 库的使用权限


```
0: jdbc:hive2://hadoop102:10000/> use gmall;
Error: Error while compiling statement: FAILED: HiveAccessControlExce
ption Permission denied: user [atguigu] does not have [USE] privilege
on [gmall] (state=42000,code=40000)
0: jdbc:hive2://hadoop102:10000/>
```

5) 赋予 atguigu 用户对 gmall 数据库的访问权限

1) 点击 Add New Policy



2) 配置授权策略

如下图所示，将 gmall 库的所有表的所有权限均授予给了 atguigu 用户。

Ranger
Access Manager
Audit
Security Zone
Settings
admin

Service Manager
hive Policies
Create Policy

Create Policy

Policy Details :

Policy Type
Access
Add Validity Period

Policy Name *
atguigu 授权策略名称
enabled
no

Policy Label
Policy Label

database *
gmail 目标数据库
include

table *
目标表
include

Hive Column *
目标字段
include

Description

Audit Logging
YES

Allow Conditions : 允许条件
hide

Select Role	Select Group	Select User	Permissions	Delegate Admin	
Select Roles	Select Groups	atguigu	All		

+

Exclude from Allow Conditions :
hide

Select Role	Select Group	Select User	Permissions	Delegate Admin	
+					

Deny All Other Accesses :
False

Deny Conditions : 拒绝条件
hide

Select Role	Select Group	Select User	Permissions	Delegate Admin	
Select Roles	Select Groups	Select Users	Add Permissions		

+

Exclude from Deny Conditions :
hide

Select Role	Select Group	Select User	Permissions	Delegate Admin	
+					

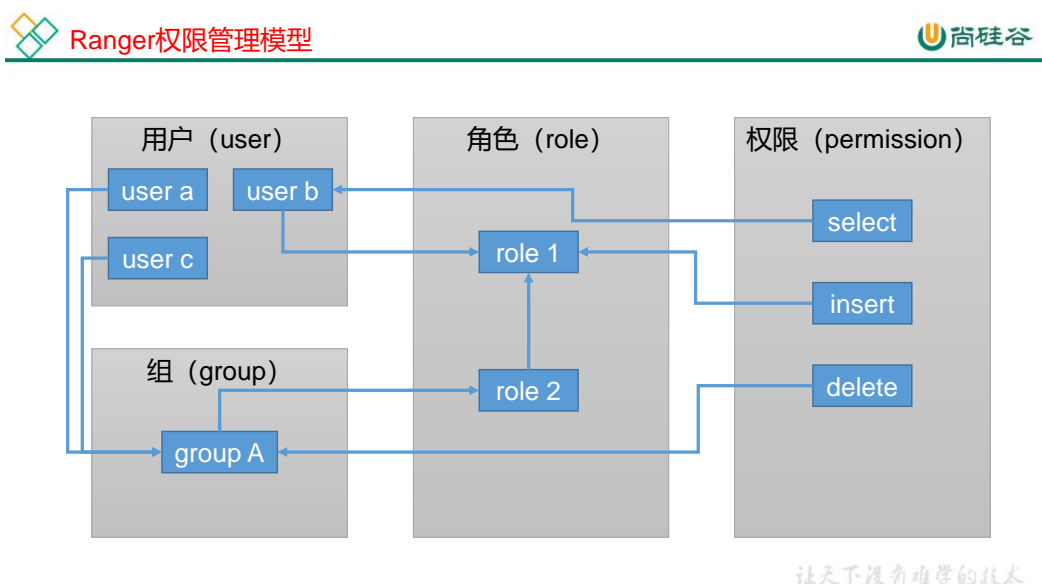
Add
Cancel

3) 等待片刻, 在回到 beeline 客户端, 重新执行 use gmall 语句, 此时 atguigu 用户已经能够使用 gmall 库, 并且可访问 gmall 库下的所有表了。

5.2 Ranger 授权模型

Ranger 所采用的权限管理模型可归类为 RBAC (Role-Based Access Control) 基于角色的访问控制。基础的 RBAC 模型共包含三个实体, 分别是用户 (user)、角色 (role) 和权限 (permission)。用户需划分为某个角色, 权限的授予对象也是角色, 例如用户张三为管理角色, 那他就拥有了管理员角色的所有权限。

Ranger 的权限管理模型比基础的 RBAC 模型要更加灵活, 以下是 Ranger 的权限管理模型。



第 6 章 官网其他权限配置

更多配置, 可以参考官网介绍: <https://cwiki.apache.org/confluence/display/RANGER/Row-level+filtering+and+column-masking+using+Apache+Ranger+policies+in+Apache+Hive>