

# 尚硅谷大数据技术之 Ranger

(作者：尚硅谷大数据研发部)

版本：V6.0.0

## 第 1 章 Ranger 概述

### 1.1 什么是 Ranger

Apache Ranger 是一个用来在 Hadoop 平台上进行监控，启用服务，以及全方位数据安全访问管理的安全框架。

Ranger 的愿景是在 Apache Hadoop 生态系统中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，这就要求 Hadoop 内的数据安全性需要扩展为同时支持多种不同的需求进行数据访问，同时还需要提供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

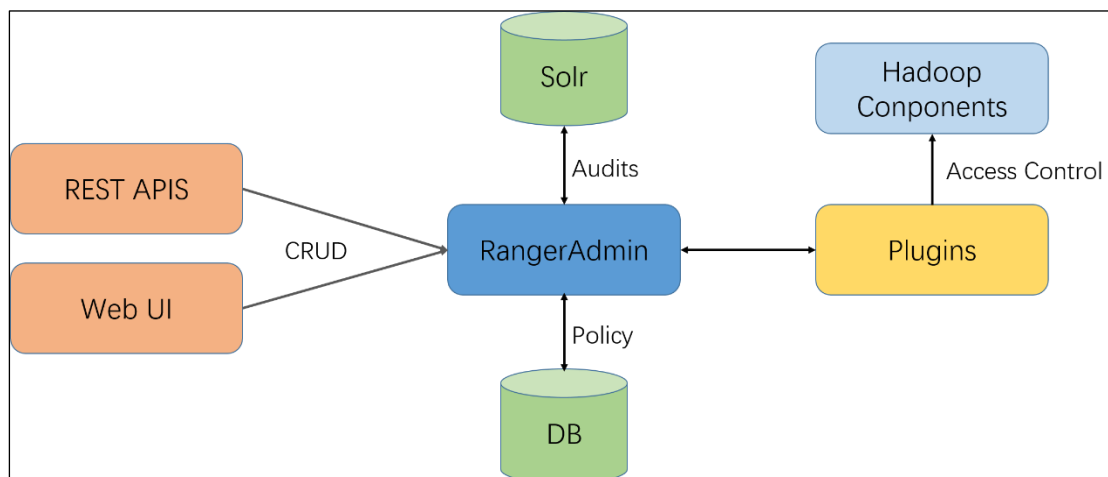
- 允许用户使用 UI 或 REST API 对所有和安全相关的任务进行集中化的管理
- 允许用户使用一个管理工具对操作 Hadoop 体系中的组件和工具的行为进行细粒度的授权
- 支持 Hadoop 体系中各个组件的授权认证标准
- 增强了对不同业务场景需求的授权方法支持，例如基于角色的授权或基于属性的授权
- 支持对 Hadoop 组件所有涉及安全的审计行为的集中化管理

### 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，也成为 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 环境准备

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！

### 2.2 安装 RangerAdmin

#### 2.2.1 数据库环境准备

(1) 登录 MySQL

```
[atguigu@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

```
[atguigu@hadoop102 module]$ mkdir ranger
```

(2) 解压软件

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

(2) 进入 /opt/module/ranger/ranger-2.0.0-admin 路径，对 install.properties 配置

```
[atguigu@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
#其他 ranger admin 需要的用户密码
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
polycmgr_external_url=http://hadoop102:6080
#启动 ranger admin 进程的 linux 用户信息
unix_user=atguigu
unix_user_pwd=atguigu
unix_group=atguigu
#hadoop 的配置文件目录
hadoop_conf=/opt/module/hadoop-3.1.3/etc/hadoop
```

(3) 之后切换到 root 用户，执行安装

```
[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.
```

(4) 创建 ranger 的配置文件软连接到 web 应用下

```
[root@hadoop102 ranger-2.0.0-admin]# ./set_globals.sh
usermod: 无改变
[2020/04/30 13:58:47]: [I] Soft linking /etc/ranger/admin/conf
to ews/webapp/WEB-INF/classes/conf
```

## 2.2.3 启动 RangerAdmin

(1) 配置 RangerAdmin web 应用的配置信息

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

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

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

(2) 启动 ranger

```
[root@hadoop102 conf]# 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！**

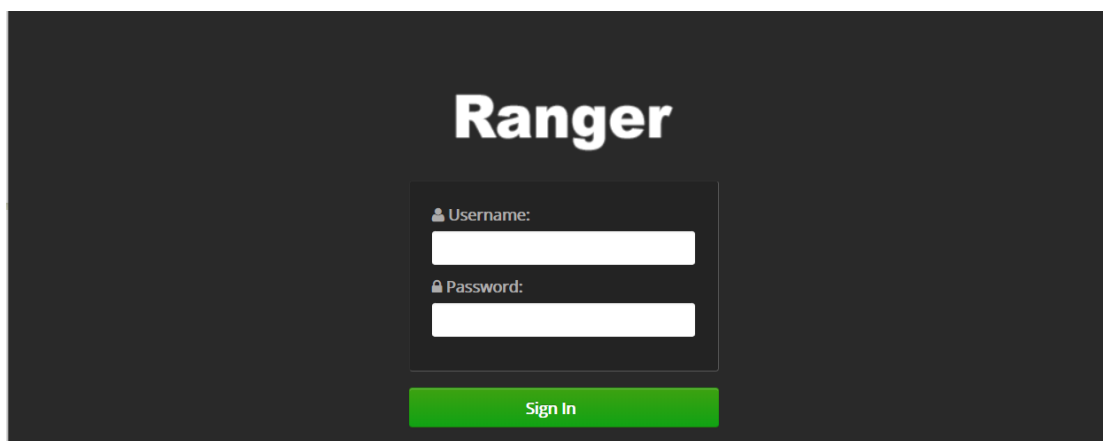
(3) 查看启动后的进程

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

(4) 停止 ranger

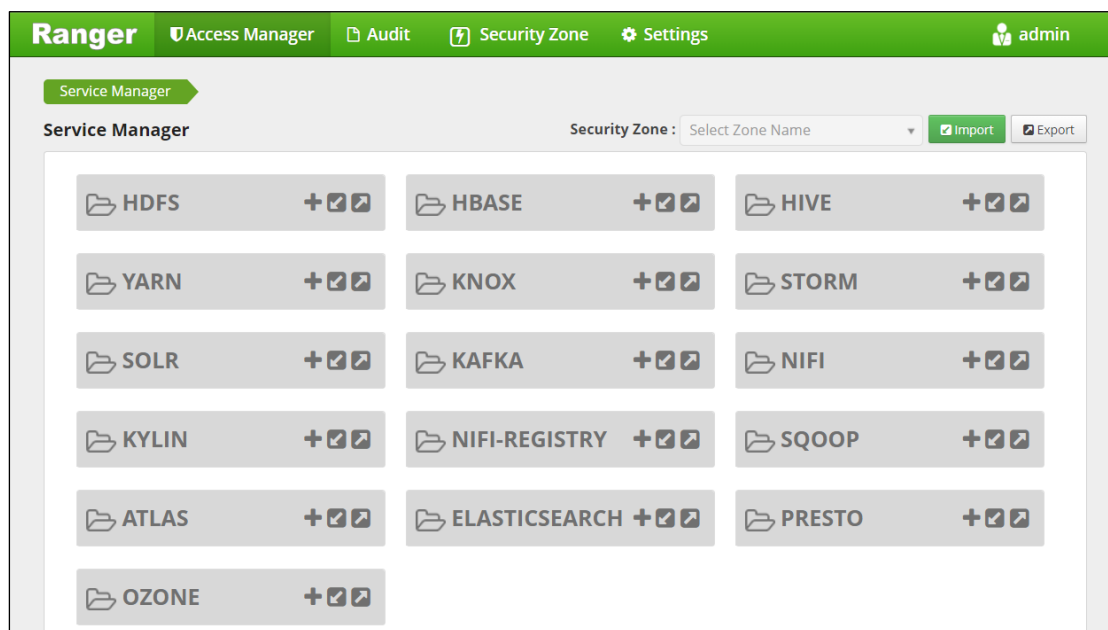
```
[root@hadoop102 conf]# ranger-admin stop
```

之后可以通过访问 <http://hadoop102:6080>，如出现以下界面，说明 ranger-admin 启动完成！



## 2.2.4 登录管理员用户

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

```
[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=atguigu
unix_group=atguigu
#rangerUserSync 的用户密码，参考 rangeradmin 中 install.properties 的
```

配置

```
rangerUsersync_password=atguigu123
```

#hadoop 的配置文件目录

```
hadoop_conf=/opt/module/hadoop-3.1.3/etc/hadoop
```

(3) 使用 root 用户进行安装

```
[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
```

### 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：

Ranger Access Manager Audit Security Zone Settings admin						
Users/Groups/Roles						
Users Groups Roles						
User List						
Search for your users...				Add New User	Set Visibility	
	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) 使用 root 用户启动

```
[root@hadoop102 ranger-2.0.0-usersync]# 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：

Ranger Access Manager Audit Security Zone Settings admin						
Users/Groups/Roles						
Users Groups Roles						
User List						
Search for your users... Add New User Set Visibility						
	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"/>	setrouleshoot		User	External	setrouleshoot	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。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
#组件名称可以自定义
REPOSITORY_NAME=hivedev
#hive 的安装目录
COMPONENT_INSTALL_DIR_NAME=/opt/module/hive
#hive 组件的启动用户
CUSTOM_USER=atguigu
#hive 组件启动用户所属组
CUSTOM_GROUP=atguigu
```

(3) 将 hive 的配置文件作为软连接安装到 Ranger Hive-plugin 目录下

```
[root@hadoop102 ranger-2.0.0-hive-plugin]# ln -s /opt/module/hive/conf/ conf
```

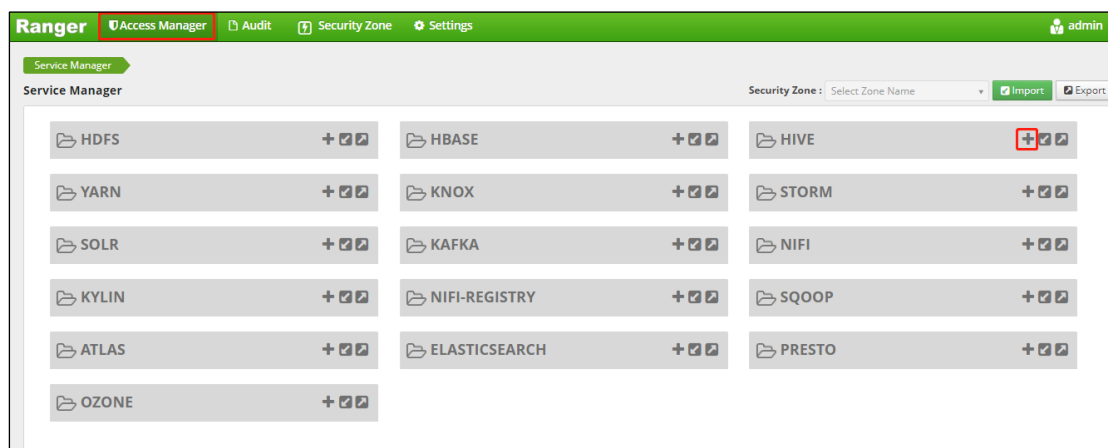
(4) 使用 root 用户启动 Ranger Hive-plugin

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

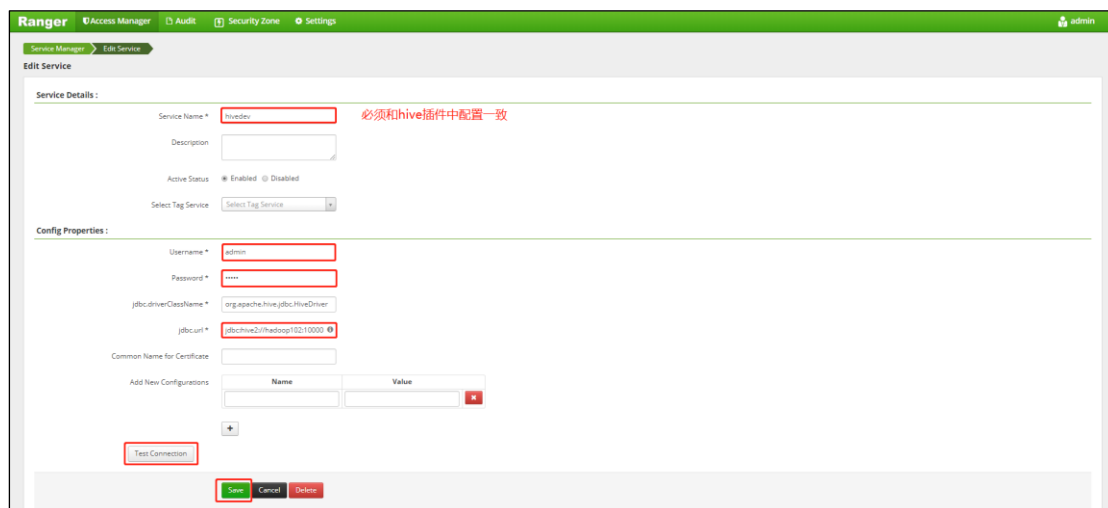
之后需要重启 hive 才能生效！

## 4.3 在 ranger admin 上配置 hive 插件

(1) Access Manager/hive



(2) 配置服务详情



注意：一定要点击 Save 后再执行后续测试。

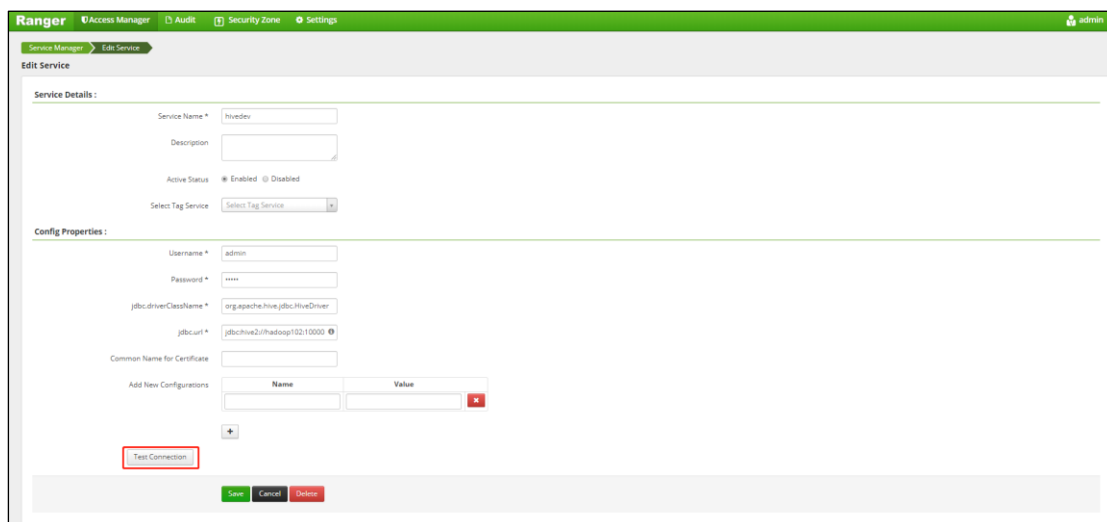
## 4.4 测试连接 hiveserver2

(1) 启动 hiveserver2

```
[atguigu@hadoop102 hive]$ hive --service metastore &
[atguigu@hadoop102 hive]$ hiveserver2
```

(2) 测试插件是否可以连接 hiveserver2





**Ranger** Access Manager Audit Security Zone Settings admin

Service Manager Edit Service

**Service Details:**

Service Name \* hivedev

Description

Active Status ☒ Enabled ☐ Disabled

Select Tag Service Select Tag Service

**Config Properties:**

Username \* admin

Password \* \*\*\*\*

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

jdbcurl \* jdbc:hive2://hadoop102:10000

Common Name for Certificate

Add New Configurations

Name	Value

Test Connection

Save Cancel Delete

(3) 出现以下提示说明连接成功!



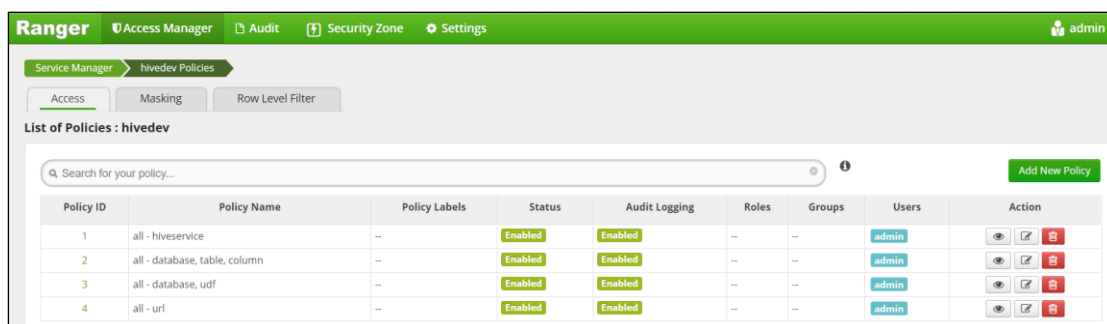
Connected Successfully.

OK

## 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 admin 用户拥有对所有库、表和函数的访问权限



**Ranger** Access Manager Audit Security Zone Settings admin

Service Manager hivedev Policies

Access Masking Row Level Filter

List of Policies: hivedev

Search for your policy...

Add New Policy

Policy ID	Policy Name	Policy Labels	Status	Audit Logging	Roles	Groups	Users	Action
1	all - hiveservice	--	Enabled	Enabled	--	--	admin	
2	all - database, table, column	--	Enabled	Enabled	--	--	admin	
3	all - database, udf	--	Enabled	Enabled	--	--	admin	
4	all - url	--	Enabled	Enabled	--	--	admin	

(2) 验证：使用 atguigu 用户尝试进行登录，登录成功后，执行查询语句

```
[atguigu@hadoop102 ~]$ beeline

beeline> !connect 'jdbc:hive2://hadoop102:10000'

Connecting to jdbc:hive2://hadoop102:10000
Enter username for jdbc:hive2://hadoop102:10000: atguigu
Enter password for jdbc:hive2://hadoop102:10000: ****
Connected to: Apache Hive (version 3.1.2)
Driver: Hive JDBC (version 3.1.2)
Transaction isolation: TRANSACTION_REPEATABLE_READ
0: jdbc:hive2://hadoop102:10000> show tables;
Error: Error while compiling statement: FAILED: HiveAccessControlException
Permission denied: user [atguigu] does not have [USE] privilege on [default]
```

```
(state=42000,code=40000)
0: jdbc:hive2://hadoop102:10000>
```

(3) 之后使用 **admin** 用户进行登录，可以完成 Hive 的所有操作。

```
[atguigu@hadoop102 ~]$ beeline

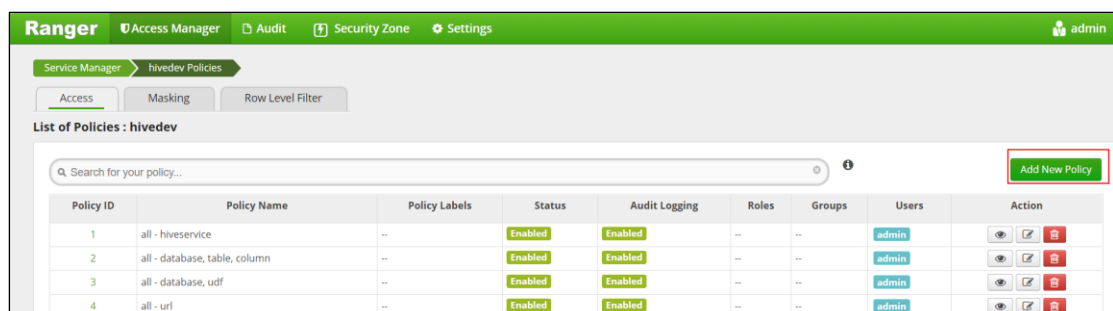
beeline> !connect 'jdbc:hive2://hadoop102:10000'
Connecting to jdbc:hive2://hadoop102:10000
Enter username for jdbc:hive2://hadoop102:10000: admin
Enter password for jdbc:hive2://hadoop102:10000: ****
Connected to: Apache Hive (version 3.1.2)
Driver: Hive JDBC (version 3.1.2)
Transaction isolation: TRANSACTION_REPEATABLE_READ
0: jdbc:hive2://hadoop102:10000> show tables;

+-----+
| tab_name |
+-----+
| student  |
| student1 |
+-----+
```

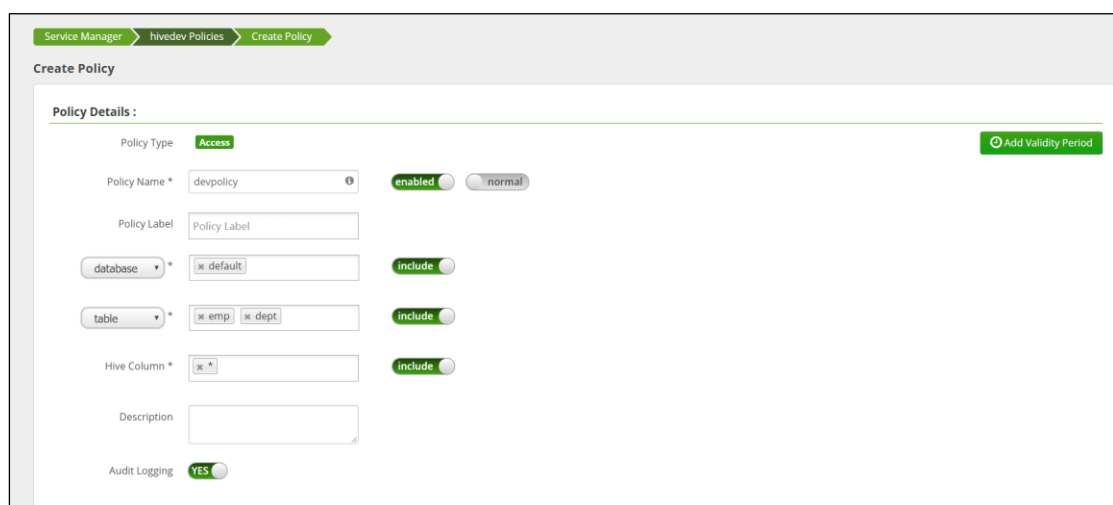
## 5.2 为用户配置权限

例如为 atguigu 用户配置 default 库 emp 和 dept 表的所有列的读权限，为 jack 用户配置 default 库 emp 和 dept 表的所有列的读写权限。

(1) 点击 Add New Policy 按钮



(2) 填写策略名称，以及此策略设计的库、表、列等信息



### (3) 填写设计此策略的允许的用户权限

Allow Conditions :

Select Role	Select Group	Select User	Permissions	Delegate Admin	
Select Roles	Select Groups	atguigu	select	<input type="checkbox"/>	<input type="button" value="x"/>
Select Roles	Select Groups	jack	select update	<input type="checkbox"/>	<input type="button" value="x"/>

### (4) 之后点击 Add 添加按钮，发现在面板上已经添加完成

Service Manager

hivedev Policies

Access Masking Row Level Filter

List of Policies : hivedev

Search for your policy...

Add New Policy

Policy ID	Policy Name	Policy Labels	Status	Audit Logging	Roles	Groups	Users	Action
1	all - hiveservice	--	Enabled	Enabled	--	--	admin	<input type="button" value="eye"/> <input type="button" value="edit"/> <input type="button" value="delete"/>
2	all - database, table, column	--	Enabled	Enabled	--	--	admin	<input type="button" value="eye"/> <input type="button" value="edit"/> <input type="button" value="delete"/>
3	all - database, udf	--	Enabled	Enabled	--	--	admin	<input type="button" value="eye"/> <input type="button" value="edit"/> <input type="button" value="delete"/>
4	all - url	--	Enabled	Enabled	--	--	admin	<input type="button" value="eye"/> <input type="button" value="edit"/> <input type="button" value="delete"/>
5	devpolicy	--	Enabled	Enabled	--	--	atguigu jack	<input type="button" value="eye"/> <input type="button" value="edit"/> <input type="button" value="delete"/>

(5) 测试: beeline 无需重新连接 hiveserver2，再次执行查询，发现 atguigu 用户已经可以进行查询，但是只能查询自己有权限查询的表信息

```

0: jdbc:hive2://hadoop103:10000> show tables;
INFO : Compiling command(queryId=atguigu_20200430155544_0af189ed-9eea-420c-83d8-fba3cb173921): show tables
INFO : Concurrency mode is disabled, not creating a lock manager
INFO : Semantic Analysis Completed (retrial = false)
INFO : Returning Hive schema: Schema(fieldSchemas:[FieldSchema(name:tab_name, type:string, comment:from deserializer)], properties:null)
INFO : Completed compiling command(queryId=atguigu_20200430155544_0af189ed-9eea-420c-83d8-fba3cb173921); Time taken: 0.018 seconds
INFO : Concurrency mode is disabled, not creating a lock manager
INFO : Executing command(queryId=atguigu_20200430155544_0af189ed-9eea-420c-83d8-fba3cb173921): show tables
INFO : Starting task [Stage-0:DDL] in serial mode
INFO : Completed executing command(queryId=atguigu_20200430155544_0af189ed-9eea-420c-83d8-fba3cb173921); Time taken: 0.014 seconds
INFO : OK
INFO : Concurrency mode is disabled, not creating a lock manager
+-----+
| tab_name |
+-----+
| dept    |
| emp     |
+-----+

```

### (6) 对以下两个表，有读权限，没有写权限

```

0: jdbc:hive2://hadoop103:10000> select * from dept;
INFO : Compiling command(queryId=atguigu_20200430160050_cc76d406-4c82-4fec-8f21-cc61f5366821): select * from dept
INFO : Concurrency mode is disabled, not creating a lock manager
INFO : Semantic Analysis Completed (retrial = false)
INFO : Returning Hive schema: Schema(fieldSchemas:[FieldSchema(name:dept.deptno, type:int, comment:null), FieldSchema(name:dept.dname, type:string, comment:null), FieldSchema(name:dept.loc, type:int, comment:null)], properties:null)
INFO : Completed compiling command(queryId=atguigu_20200430160050_cc76d406-4c82-4fec-8f21-cc61f5366821); Time taken: 0.209 seconds
INFO : Concurrency mode is disabled, not creating a lock manager
INFO : Executing command(queryId=atguigu_20200430160050_cc76d406-4c82-4fec-8f21-cc61f5366821): select * from dept
INFO : Completed executing command(queryId=atguigu_20200430160050_cc76d406-4c82-4fec-8f21-cc61f5366821); Time taken: 0.001 seconds
INFO : OK
INFO : Concurrency mode is disabled, not creating a lock manager
+-----+-----+-----+
| dept.deptno | dept.dname | dept.loc |
+-----+-----+-----+
| 10          | ACCOUNTING | 1700     |
| 20          | RESEARCH  | 1800     |
| 30          | SALES      | 1900     |
| 40          | OPERATIONS | 1700     |
+-----+-----+-----+
4 rows selected (0.287 seconds)
0: jdbc:hive2://hadoop103:10000> insert into table dept values(50,'SECURITY',1800);
Error: Error while compiling statement: FAILED: HiveAccessControlException Permission denied: user [atguigu] does not have [UPDATE] privilege on [default/dept] (state=42000,code=40000)

```

### (7) 再次测试 jack 用户，尝试向 dept 表写入数据后查询

```
0: jdbc:hive2://hadoop102:10000> insert into table dept values(50,'SECURITY',1800);
```

```
0: jdbc:hive2://hadoop102:10000> select * from dept;
```

dept.deptno	dept.dname	dept.loc
50	SECURITY	1800
10	ACCOUNTING	1700
20	RESEARCH	1800
30	SALES	1900
40	OPERATIONS	1700

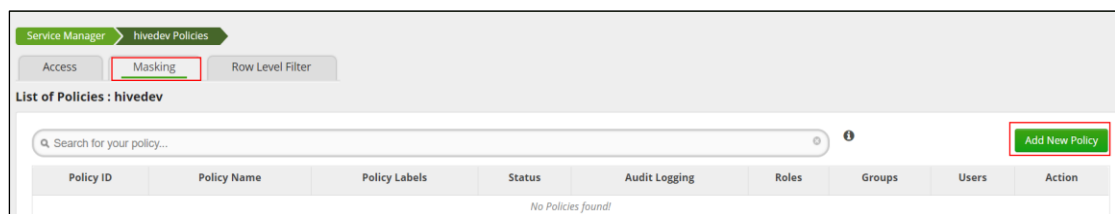
## 5.3 脱敏操作

通过脱敏操作可以限制用户对某一列的访问，将敏感数据不暴露给用户！

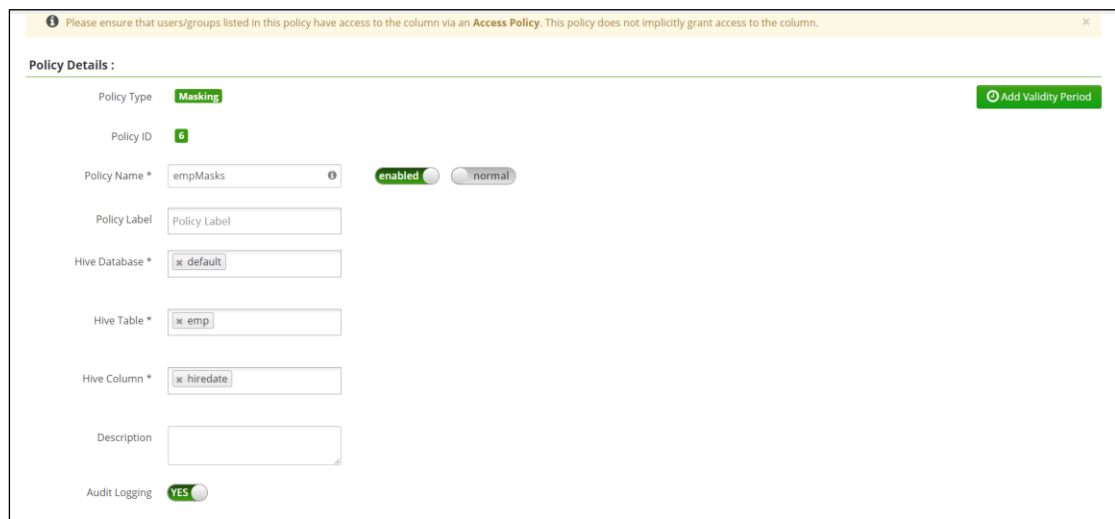
案例：指定 atguigu 用户在查询 emp 表时，对 hiredate 的年月部分脱敏！

首先需要保证用户对指定的列有访问权限，可以参考 5.2 进行配置！

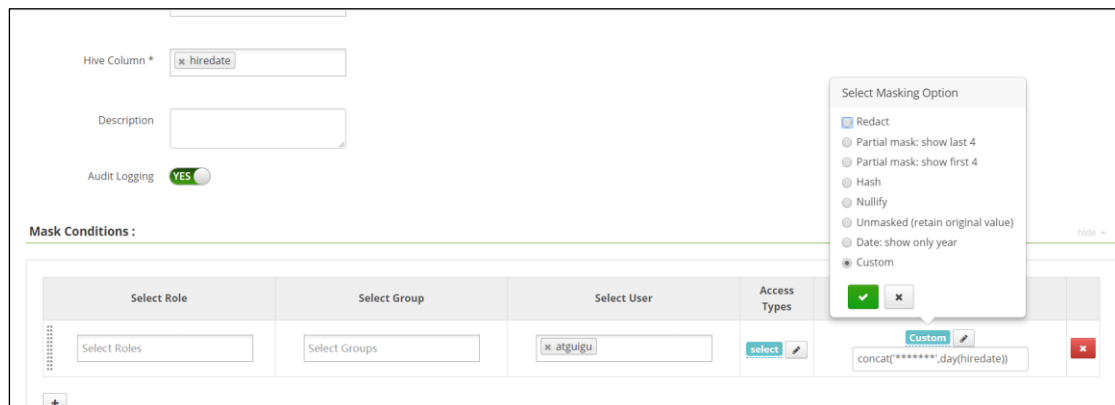
(1) 点击 Masking 标签，再点击 Add New Policy



(2) 指定表和列



(3) 指定用户和脱敏操作



(4) 之后点击 save 按钮！那么只有 atguigu 用户在查询时，会触发此策略！

emp.empno	emp.ename	emp.job	emp.mgr	emp.hiredate	emp.sal	emp.comm	emp.deptno
7369	SMITH	CLERK	7902	*****17	800.0	NULL	20
7499	ALLEN	SALESMAN	7698	*****20	1600.0	300.0	30
7521	WARD	SALESMAN	7698	*****22	1250.0	500.0	30
7566	JONES	MANAGER	7839	*****2	2975.0	NULL	20
7654	MARTIN	SALESMAN	7698	*****28	1250.0	1400.0	30
7698	BLAKE	MANAGER	7839	*****1	2850.0	NULL	30
7782	CLARK	MANAGER	7839	*****9	2450.0	NULL	10
7788	SCOTT	ANALYST	7566	*****19	3000.0	NULL	20
7839	KING	PRESIDENT	NULL	*****17	5000.0	NULL	10
7844	TURNER	SALESMAN	7698	*****8	1500.0	0.0	30
7876	ADAMS	CLERK	7788	*****23	1100.0	NULL	20
7900	JAMES	CLERK	7698	*****3	950.0	NULL	30
7902	FORD	ANALYST	7566	*****3	3000.0	NULL	20
7934	MILLER	CLERK	7782	*****23	1300.0	NULL	10

14 rows selected (0.332 seconds)

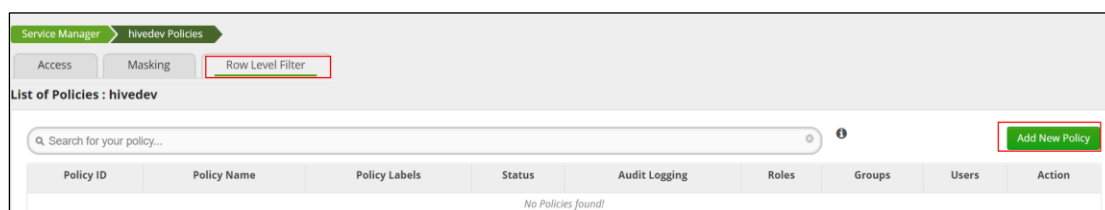
## 5.4 行级别过滤

通过行级别过滤可以将表中的数据进行条件过滤后再暴露给用户！

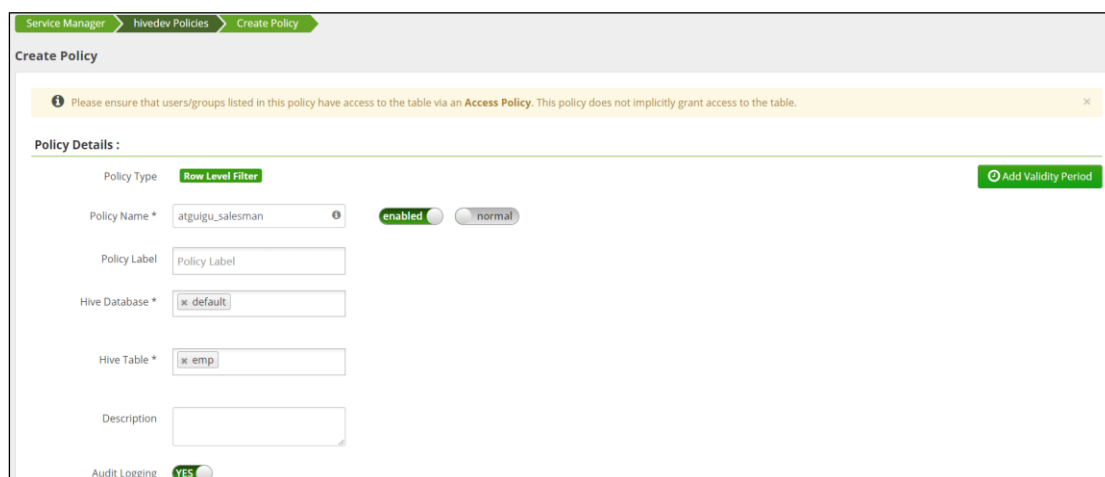
例如：atguigu 用户只允许查询 emp 表中 job 类型为 SALESMAN 的用户信息。

同理，行级别过滤也要求用户对指定表有 access 权限！参考 5.2 的配置！

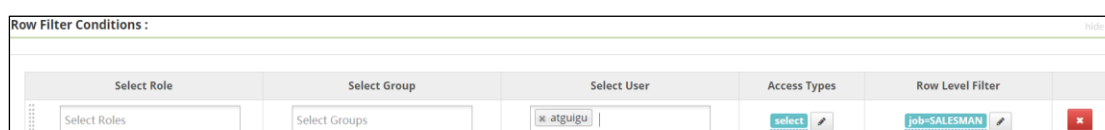
(1) 选择 Row Level Filter 标签，点击 Add New Policy:



(2) 选择对应的库和表:



(3) 添加过滤规则 and 用户



(4) 之后点击 add 按钮！验证。

emp.empno	emp.ename	emp.job	emp.mgr	emp.hiredate	emp.sal	emp.comm	emp.deptno
7499	ALLEN	SALESMAN	7698	*****20	1600.0	300.0	30
7521	WARD	SALESMAN	7698	*****22	1250.0	500.0	30
7654	MARTIN	SALESMAN	7698	*****28	1250.0	1400.0	30
7844	TURNER	SALESMAN	7698	*****8	1500.0	0.0	30

## 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>