

Hadoop 访问控制技术文档

版 本 号 v1.0

文 档 作 者 苏秋月

撰 写 日 期 2017-06-05

项 目 组 大数据平台项目组

评 审 负 责 人

目 录

1	<u>HADOOP 原生授权机制</u>	<u>5</u>
1.1	HDFS 授权.....	5
1.2	HDFS 扩展 ACLs	5
1.3	服务级别授权	8
2	<u>APACHE RANGER 简介.....</u>	<u>9</u>
2.1	组成框架	10
2.2	实现原理	10
2.2.1	认证	11
2.2.2	授权	11
2.2.3	审计	12
3	<u>RANGER 安装准备</u>	<u>12</u>
3.1	安装 MAVEN	13
3.2	安装 POSTGRESQL.....	13
4	<u>源码编译安装 RANGER</u>	<u>15</u>
4.1	安装配置 RANGER ADMIN	16
4.2	安装配置 RANGER USERSYNC	17
4.3	安装配置 RANGER PLUGINS.....	18
4.4	配置 SOLR 提供 AUDIT 服务.....	19
4.4.1	SOLR 安装配置流程	19
4.4.2	修改 INSTALL.PROPERTIES 文件.....	20
4.4.3	下载 SOLR 安装包	20
4.4.4	启动 SOLR 服务	21
5	<u>使用 AMBARI 部署 RANGER.....</u>	<u>22</u>

5.1	配置数据库.....	22
5.1.1	安装 JDBC 连接驱动.....	22
5.1.2	修改配置文件	22
5.1.3	重启 AMBARI-SERVER.....	23
5.1.4	在环境变量中添加 JDBC 驱动路径.....	23
5.2	安装 SOLR	23
5.3	配置 RANGER.....	23
5.4	配置 RANGER USER SYNC.....	25
5.4.1	配置 LINUX 环境下的用户/组同步映射.....	26
5.4.2	配置文件中用户/组映射.....	26
5.4.3	配置 LDAP/AD 中的用户/组映射	27
5.5	配置 SOLR 提供审计.....	30
5.6	配置 RANGER TAGSYNC	31
5.7	安装配置 ATLAS	32
5.7.1	HIVE 数据导入 ATLAS	32
5.7.2	创建标签	32
5.8	添加 RANGER 插件.....	33
6	<u>使用 RANGER 为 HADOOP 授权管理.....</u>	34
6.1	基于资源的策略管理.....	35
6.1.1	管理 HDFS 策略.....	35
6.1.2	管理 HIVE 策略	38
6.2	管理基于标签的策略.....	41
6.2.1	创建策略存储仓库	41
6.2.2	创建基于标签的策略	42
6.2.3	为组件添加基于标签的策略仓库	43
6.3	审计.....	43
6.3.1	用户访问记录	43
6.3.2	管理员操作记录	44
6.3.3	登录会话	45
6.3.4	插件状态	46
6.4	设置.....	46
6.4.1	用户/组管理.....	46

6.4.2	权限管理	48
6.5	REPORTS 模块管理	48
7	参考文献	50

1 Hadoop 原生授权机制

Hadoop 通过使用 HDFS 文件权限^[1]和服务级别授权^[2]，向认证用户提供授权控制。

- HDFS 使用类似于 UNIX 模型的 POSIX 风格的文件和目录权限模型。每个文件和目录与所有者和组相关联，并具有关联的读取和写入权限。基于这些文件权限和用户的身份和组成员资格，HDFS 控制对分布式文件系统的读写操作。与 UNIX 模型不同，没有“可执行”权限的概念。
- 服务级别授权是一种提供用户访问特定服务的访问控制列表（ACL）的功能。群集可以使用此机制来控制哪些用户和组具有提交作业的权限。
- 执行作业授权，以允许用户只能查看和修改自己的作业，这是通过设置 MapReduce 参数来实现的。

1.1 HDFS 授权

每次尝试访问 HDFS 中的文件或目录时必须先通过授权检查。HDFS 采用与 POSIX 兼容的文件系统通用的授权方案。由三个不同类别的用户管理授权：所有者（owner），组（group）和其他人（other）。每个文件或目录由特定用户拥有，该用户组成对象的所有者类。对象也被分配一个组，该组的所有成员组成对象的组类。所有不属于所有者且不属于分配给该对象的组的所有用户组成其他类。读取，写入和执行权限可以独立授予每个类。

这些权限由单个八进制整数表示，通过将权限值（读取为 4，写入为 2，执行 1）来计算。例如，为了表示一个类具有对目录的读取和执行权限，将分配 5（4+1）的八进制值。在 HDFS 中，如果文件的名称已知，执行位将允许访问文件的内容和元数据信息。分配给所有者、组和其他人的权限可以通过以该顺序连接三个八进制值来表示。例如，获取所有者具有读取和写入权限的文件，并且所有其他用户只有读取权限。该文件的权限将被表示为 644；6 分配给所有者，因为它具有读和写（4+2），并且 4 被分配给组和其他类，因为它们只具有读取权限。对于所有权限授予所有用户的文件，权限将为 777。

无论文件或目录的权限如何，NameNode 运行的用户（通常为 hdfs）和 dfs.permissions.superusergroup（默认为超级组）中定义的组中任何成员都可以读取，写入或删除任何文件和目录。就 HDFS 而言，它们在 Linux 系统上相当于 root。

1.2 HDFS 扩展 ACLs

使用基本 POSIX 权限限制给定文件或目录的访问不总是容易的。如果两个或更多不同的用户组需要访问相同的 HDFS 目录，会发生什么？使用基本的

POSIX 权限，管理员可以使用两个选项：一是可以允许所有人访问文件目录；二是创建一个包含所有需要访问该目录的用户的组，为该组分配访问权限。但是第一种情况可能使数据的使用超过可控的范围，而第二种情况在组管理方面会变得很棘手。当一组用户需要读取访问并且另一组用户需要读取和写入访问时，这个问题变得更加复杂。随着 Hadoop 2.4 的发布，HDFS 现在配备了扩展 ACL。这些 ACL 的工作方式与 Unix 环境中的扩展 ACL 相同。这允许 HDFS 中的文件和目录具有比基本 POSIX 权限更多的权限。

不过要使用 HDFS 扩展 ACL，必须首先在 NameNode 上启用它们。为此，请在 `hdfs-site.xml` 中将配置属性 `dfs.namenode.acls.enabled` 设置为 `true`。ACL 由一组 ACL 条目组成。每个 ACL 条目都会命名特定用户或组，并授予或拒绝该特定用户或组的读取，写入和执行权限。例如：

```
user::rw-
user:bruce:rw-          #effective:r--
group::r-x              #effective:r--
group:sales:rw-         #effective:r--
mask::r--
other::r--
```

ACL 项包含了一个类型，一个可选的名字和一个权限字符串。为了显示的需要，':'用来作为分隔符。在这个例子里，文件的 `owner` 拥有读写权限，文件的 `group` 拥有读和执行权限，其他人拥有读权限。到此，这等价于设置文件的权限 bit 为 654。

除此之外，还有两个扩展的 ACL 项，一个对应特定的用户 `bruce`，一个对应特定的组 `sales`，两者都被授予了完全访问权限。`mask` 是一种特殊的 ACL 项，用来过滤授予特定用户或组，也包含非特定的组的权限。在此例中，`mask` 只有读权限，对应的几个 ACL 项也被相应的过滤，只有读权限。

每条 ACL 必须有一个 `mask`。如果用户不提供 `mask`，那么系统会自动插入一条 `mask`，该 `mask` 是通过 union 所有项的权限作为过滤项计算出来的。对一个拥有 ACL 的文件运行 `chmod` 会导致改变 `mask` 的权限。由于 `mask` 是一个过滤器，这有效的限制了所有扩展的 ACL 项，而不是仅仅改变 `group` 项和可能丢失的其他扩展 ACL 项。

该模型也区分“访问 ACL”和“默认 ACL”，访问 ACL 定义了权限检查时的规则，而默认 ACL 定义了新的子文件或者子目录创建时自动生效的 ACL 项。举例如下：

```
user::rwx
```

```
group::r-x
other::r-x
default:user::rwx
default:user:bruce:rwx      #effective:r-x
default:group::r-x
default:group:sales:rwx     #effective:r-x
default:mask::r-x
default:other::r-x
```

只有目录才会有默认 ACL。当新的文件或子目录创建时，系统自动复制父目录的默认 ACL 到其文件或子目录。子目录同时会把该默认 ACL 作为自己的默认 ACL。在此原则下，默认 ACL 会传递到系统任意深度的目录树结构中。

对于子文件或目录准确的访问 ACL 值是由 mode 参数过滤决定的。假设默认的 umask 是 022，那么对于新的文件来说就是 644，对于新目录就是 755。mode 参数过滤对于未指定的用户（文件 owner），mask 和其他用户的复制来的权限。使用这一特定的示例 ACL，创建一个新的子目录，其权限 mode 是 755，那么 mode 过滤器对于最后的结果不起作用。然而，如果我们假设创建一个文件 mode 是 644，那 mode 过滤器会过滤新文件的 ACL，导致对于未指定用户（文件 owner）拥有读写权限，对于 mask 拥有读权限，对于其他用户拥有读权限。这个 mask 同样作用于已制定名字的 bruce 和 sales，其权限也将是只读。

注意，ACL 的复制发生在新目录或文件创建时。后续的对于父目录的默认 ACL 的变更不会影响已经存在的子文件或目录。

默认 ACL 必须拥有最小的 ACL 项集，包括未指定用户（文件 owner），未指定 group（文件 group）和其他用户。如果用户没有提供上述这些项的设置，那么系统自动复制访问 ACL 中的对应项，或者如果没有访问 ACL，就复制权限位。默认 ACL 必须有 mask。如上所述，如果 mask 未声明，系统自动创建一个 mask，此 mask 通过 union 所有项的权限过滤计算得出。

当一个文件拥有 ACL 时，对于权限的校验算法修改为：

- if 用户名与文件 owner 匹配，那么检测 owner 权限；
- Else if 用户名匹配指定的用户项，那么检测这些匹配的用户权限，这些权限先要经过 mask 过滤；
- Else if 文件的 group 与 group 列表的任意成员匹配，并且经过 mask 过滤的权限授权访问，那么使用这些权限；
- Else if 指定的 group 项匹配 group 列表中的任意成员，并且经过 mask 过滤的权限授权访问，那么使用这些权限；
- Else if 文件 group 或者任意指定的 group 项与 group 列表中的任意成员

匹配，但是没有被授权访问，那么访问拒绝；

- Otherwise 校验文件其他用户的权限。

实现权限控制的最佳实践是利用传统的权限位，同时定义很小的一组 ACLs 来对可能的例外规则增强权限校验。带有 ACL 的文件权限校验相比只用权限位的文件会导致 NameNode 额外的内存消耗。

授权基本命令操作如下：

```
hdfs dfs -getfacl [-R] //获取目录和文件的 ACL 信息
//设置文件和目录的 ACL 信息
hdfs dfs -setfacl [-R] [-b |-k -m |-x ] [--set ]
```

1.3 服务级别授权

服务级别授权是 Hadoop 提供的最原始的授权机制，用于确保只有那些经过授权的客户端才能访问对应的服务。比如管理员可限制只允许若干用户/用户组向 Hadoop 提交作业。服务访问控制是通过控制各个服务之间的通信协议实现的。它通常发生在其他访问控制机制之前，比如文件权限检查、队列权限检查等。为了启用该功能，管理员需在 core-site.xml 中将参数 `hadoop.security.authorization` 置为 `true`，并在 `hadoop-policy.xml` 中为各个通信协议指定具有访问权限的用户或者用户组。目前有 HDFS，MapReduce（MR1）和 YARN（MR2）支持服务级授权。其中 HDFS 服务级别授权配置（ACL）如下：

表1 HDFS 服务级别授权配置

Property name	Description	Suggested value
<code>security.client.protocol.acl</code>	Client to NameNode protocol; used by user code via the <code>DistributedFileSystem</code> class	"yarn,mapred,hadoop-users"
<code>security.client.datanode.protocol.acl</code>	Client to DataNode protocol	"yarn,mapred,hadoop-users"
<code>security.get.user.mappings.protocol.acl</code>	Protocol to retrieve the groups that a user maps to	"yarn,mapred,hadoop-users"
<code>security.datanode.protocol.acl</code>	DataNode to NameNode protocol	"hdfs"
<code>security.inter.datanode.protocol.acl</code>	DataNode to DataNode protocol	"hdfs"
<code>security.namenode.protocol.acl</code>	SecondaryNameNode to NameNode protocol	"hdfs"
<code>security.qjournal.service.protocol.acl</code>	NameNode to JournalNode protocol	"hdfs"
<code>security.zkfc.protocol.acl</code>	Protocol exposed by the <code>ZKFailoverController</code>	"hdfs"
<code>security.ha.service.protocol.acl</code>	Protocol used by the <code>hdfs hadmin</code> command to manage the HA states of the NameNodes	"hdfs,yarn,hadoop-admins"
<code>security.refresh.policy.protocol.acl</code>	Used by the <code>hdfs dfsadmin</code> command to load the latest <code>hadoop-policy.xml</code> file	"hadoop-admins"
<code>security.refresh.user.mappings.protocol.acl</code>	Protocol to refresh the user to group mappings	"hadoop-admins"

MapReduce 和 YARN 都无法控制对数据的访问，谁都可以访问群集资源，

如 CPU，内存，磁盘 I/O 和网络 I/O。由于这些资源是有限的，特别是在多租户环境中，管理员通常将资源分配给特定的用户或组。上面描述的服务级别授权控制对特定协议的访问，例如谁可以和不能将作业提交到集群，但是它们不足以控制对集群资源的访问。为了安全地控制这些资源，Hadoop 支持作业队列上的访问控制列（ACL）。这些 ACL 控制哪些用户可以提交给某些队列以及哪些用户可以管理队列。MapReduce 定义了不同类别的用户，以下这些用户会影响 ACL 的配置：

■ MapReduce / YARN 群集所有者

启动 JobTracker 进程（MR1）或 ResourceManager 进程（YARN）的用户被定义为群集所有者。该用户有权限将作业提交到任何队列，并可以管理任何队列或作业。在大多数情况下，集群所有者在 MapReduce（MR1）上是 mapred 和 YARN 的 yarn。因为让集群的所有者去执行作业是很危险的，LinuxTaskController 默认将用户帐号列入黑名单，使他们无法提交作业。

■ MapReduce 管理员

可以创建具有与群集所有者相同权限的全局 MapReduce 管理员。管理员具有定义特定用户或组的优势，可以审核每个管理员的各个操作。这也可以避免将密码分配到共享帐户，从而增加密码可能被盗用的可能性。

■ 作业所有者

作业的所有者是提交它的用户。作业所有者可以随时管理他们自己的作业，但只能提交作业到他们被授予权限的队列。

■ 队列管理员

可以向用户或组授予对所有作业的管理权限队列。队列管理员还可以将作业提交给他们管理的队列。

2 Apache Ranger 简介

Apache Ranger 是 Hadoop 平台上集中式的安全管理框架，为企业核心安全认证、授权、审计以及数据保护提供了一个安全策略管理中心。Apache Ranger 的主要功能是为 Hadoop 生态圈组件提供细粒度的访问控制，它还具有如下功能：

- 统一的管理界面，包括安全管理员管理、策略管理、日志审计、Ranger KMS 管理、插件管理等；
- 基于策略(Policy-Based)的访问权限模型；
- 通用的策略同步与决策逻辑，方便插件的扩展接入；
- 通用的安全管理员访问日志审计逻辑，可定义的日志存储方式，如 HDFS、Solr 等；

- 支持安全管理员和 LDAP、Unix 系统的安全管理员同步；
- 支持对 HDFS、Hive、HBase、Storm、YARN、Knox、Kafka 等 Hadoop 生态圈组件的权限管控和审计。

2.1 组成框架

Apache Ranger 的组成结构^[3]如下图所示，主要由以下三个模块组成：

- **Ranger admin:** 提供了安全管理员安全管理的 web 界面。安全管理员可以创建和更新策略文件，并将其存储在策略数据库中。每个组件内的插件刷新器定期查询这些策略文件。该门户还包括发送 HDFS 中或在关系数据库中从插件收集存储审计数据的审计服务器。
- **Ranger plugins:** 插件是轻量级的 java 程序，其嵌入在需要安全控制的组件进程中。它提供了两种功能：一是从 Ranger admin 端拉取安全管理员配置的安全策略到本地。当安全管理员提出访问请求时，根据安全策略判断该用户是否有权限访问；二是从本地将用户访问的记录返回给 Ranger 服务进行审计。
- **User group sync:** 提供从 Unix、LDAP/AD 拉取用户和用户组的功能，同步的用户和用户组能够展示在 Ranger admin 的 web 界面中。

Ranger Architecture

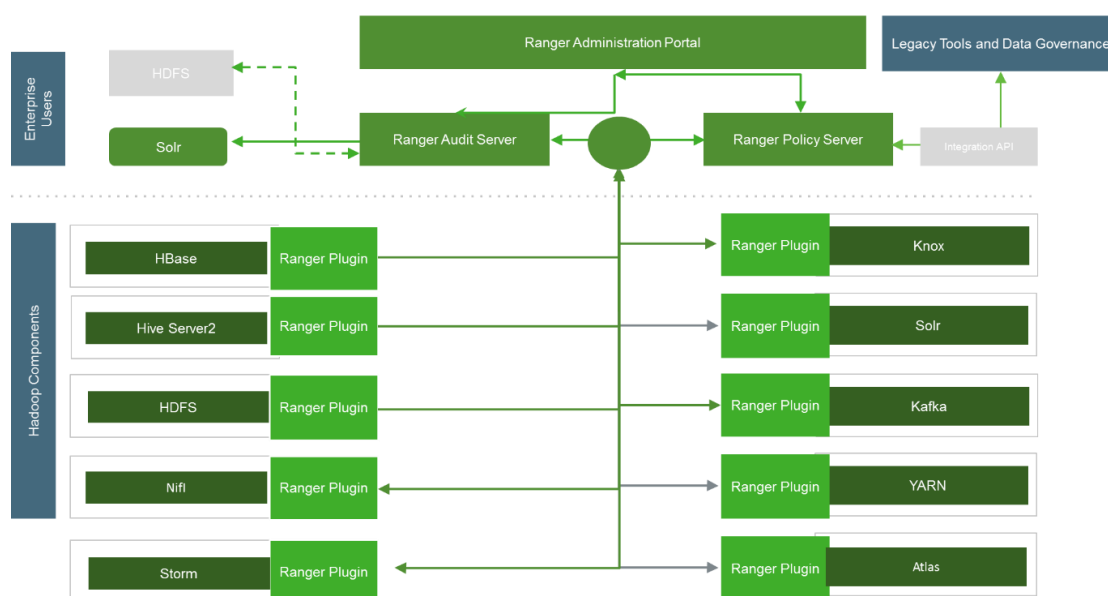


图1 Apache Ranger 框架图

2.2 实现原理

Apache Ranger 这个集中式的安全框架集成了认证、授权、审计的功能，各

功能的实现介绍如下。

2.2.1 认证

Ranger 支持使用 Unix、File 和 LDAP/AD 管理用户、用户组信息，Ranger Usersync 提供了用户同步功能。

2.2.2 授权

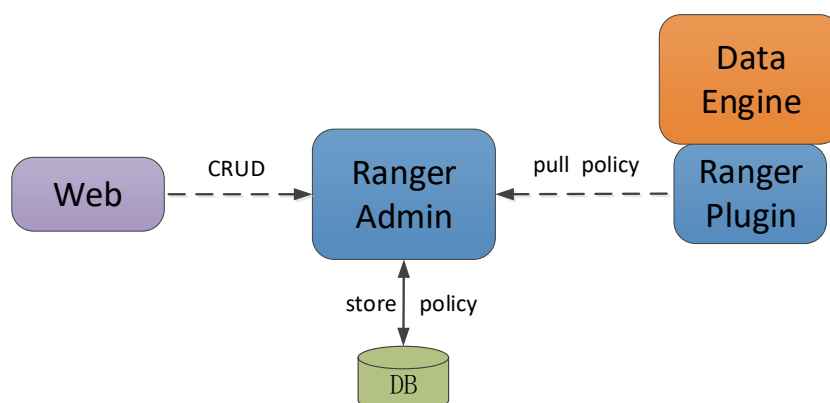


图2 访问控制原理

Ranger 采用 Web UI 界面提供了安全管理员可视化地管理授权策略，由 Ranger Admin 响应 web 端的请求，并将策略存储在数据库中。Ranger 以插件的形式集成到 Hadoop 组件当中，当组件启动时，相应的 ranger 插件也随之初始化，从 Ranger Admin 拉取策略保存到本地，以便对用户访问请求进行判定。以 Ranger 为 Hive 提供权限管理的流程为例，如下图 3 所示：

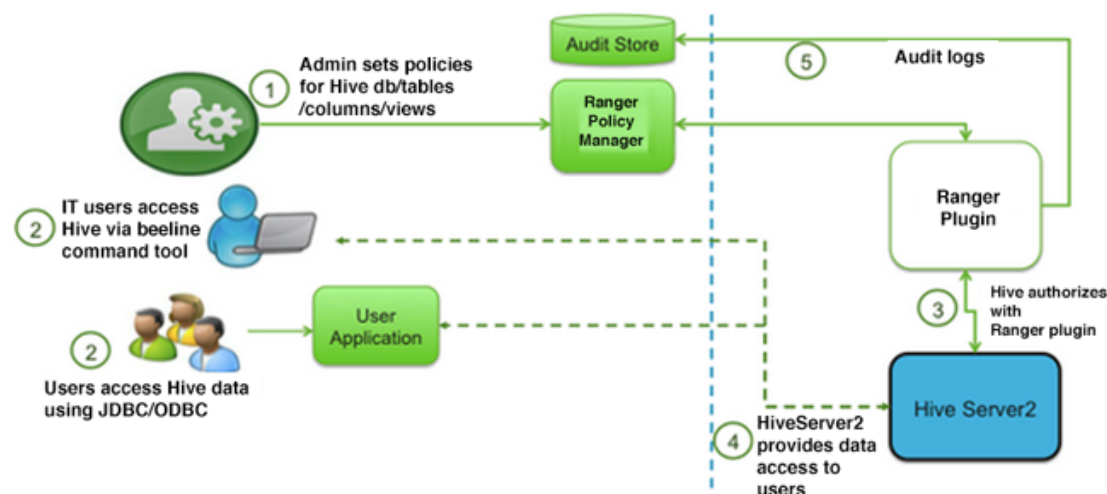


图3 Ranger 对 Hive 的权限管理

Ranger 使用了基于策略的访问控制模型，安全管理员可以通过操作授权策略对用户的访问权限进行管理。策略分为两种类型：基于资源的策略和基于标签

的策略。基于标签的策略，直接将资源以标签名的形式授权给用户，进一步保证了数据的安全。这里标签可以使用文件或 Apache Atlas 来管理，Atlas 提供了将 HDFS、Hive 等资源分类标记的功能。Ranger Tagsync 提供了与标签管理工具中标签同步功能。

Apache Ranger 插件从标签存储中检索标签详细信息，以供在策略评估期间使用。为了最大限度地降低策略评估过程中的性能影响（为资源查找标签），Apache Ranger 插件将标签保存在缓存文件中，并定期轮询标签存储区以进行任何更改。当检测到更改时，插件更新缓存文件。另外，插件将标签细节存储在本地缓存文件中，就像策略存储在本地缓存文件中一样。在组件重启时，如果标签存储不可用，插件将使用本地缓存文件中的标签数据。

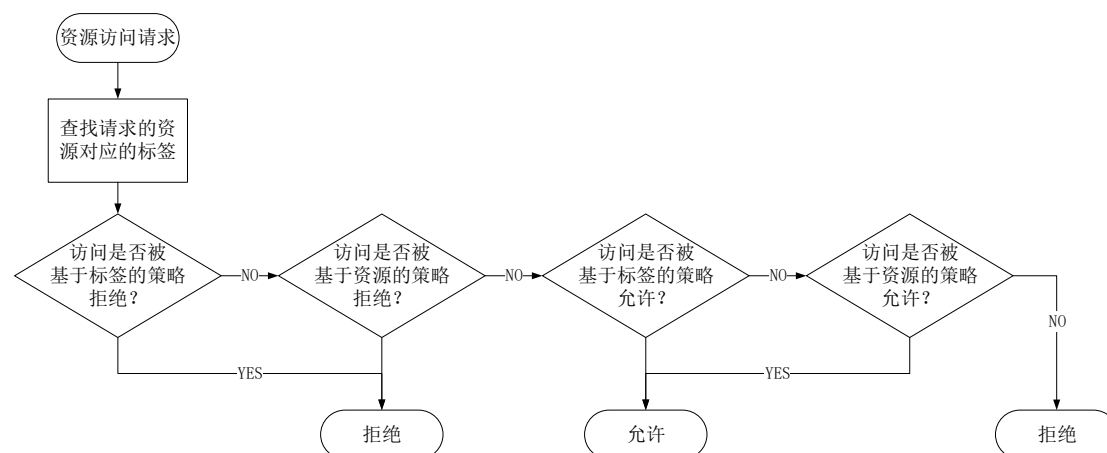


图4 策略评估流程

2.2.3 审计

Ranger 审计对用户访问数据、安全管理员操作策略和服务、安全管理员登录 web UI、以及插件的状态这几方面进行了实时的记录。在 Ranger 0.6 版本以后，只支持将访问日志保存在 HDFS 文件和 Solr 中。其中 Ranger 集成 Solr，可以在 Ranger web 页面上对审计日志进行搜索查询。

3 Ranger 安装准备

Apache Ranger 源码安装^[4]主要包括：Ranger Admin、Ranger UserSync、Component Plugins。在安装这些部件之前，Ranger 的安装环境需求如下：

首先确保集群配置了 java 环境，且 jdk 版本在 1.7 及以上。另外 Apache Ranger 源码编译需要依赖如下 Linux 组件：**maven**、**git**、**gcc**、**数据库**。git 和 gcc 的安装比较简单，直接用 yum install 安装即可，这里就不再多说。

3.1 安装 maven

(1) 下载安装包: <https://maven.apache.org/download.cgi>

(2) 解压:

```
# tar xzvf apache-maven-3.3.9.tar.gz
```

(3) 设置环境变量, 在/etc/profile 中定义 MAVEN_HOME 并追加到 PATH 里

```
export MAVEN_HOME=/home/apache-maven-3.3.9  
export PATH=$PATH:$MAVEN_HOME/bin
```

(4) 查看 maven 版本, 出现以下情况则安装成功:

```
[root@access-master home]# mvn -version  
Apache Maven 3.3.9 (bb52d8502b132ec0a5a3f4c09453c07478323dc5; 2015-11-11T00:41:47+08:00)  
Maven home: /home/apache-maven-3.3.9  
Java version: 1.8.0_112, vendor: Oracle Corporation  
Java home: /usr/java/jdk1.8.0_112/jre  
Default locale: en_US, platform encoding: UTF-8  
OS name: "linux", version: "2.6.32-504.el6.x86_64", arch: "amd64", family: "unix"
```

3.2 安装 postgresql

这里选择 postgresql9.5 版本。

(1) 禁止 iptables

```
关闭服务: service iptables stop  
关闭开机自启动: chkconfig iptables off
```

(2) 配置 yum 安装源

```
yum install  
https://download.postgresql.org/pub/repos/yum/9.5/redhat/rhel-  
7-x86_64/pgdg-centos95-9.5-2.noarch.rpm
```

(3) 安装 postgresql 和 jdbc 驱动

```
yum install postgresql95-server postgresql95-contrib  
postgresql95 (可选)  
yum install postgresql-jdbc*
```

查看安装了哪些 postgresq 服务:

```
rpm -aq |grep postgres
```

(4) 初始化数据库

```
service postgresql-9.5 initdb
```

(5) 设置远程访问数据库

```
修改/var/lib/pgsql/9.5/data/pg_hba.conf
```

```
# "local" is for Unix domain socket connections only
local    all             all                                     peer
# IPv4 local connections:
host     all             all             127.0.0.1/32          ident
# IPv6 local connections:
host     all             all             ::1/128               ident

host all all 10.0.0.45/32 trust
```

以及该目录下的 postgresql.conf 文件

```
# - Connection Settings -

listen_addresses = '*'          # what IP a
                                # (
                                # (
                                #
```

(6) 启动数据库

```
service postgresql-9.5 start
```

(7) 修改 postgres 密码

```
[root@squ-m-45 etc]# su - postgres
-bash-4.1$ psql
psql (9.5.10)
Type "help" for help.

postgres=# \password postgres
Enter new password:
Enter it again:
```

(8) 创建 ranger 用户和数据库

```
postgres=# create user ranger with password '123456';
CREATE ROLE
postgres=# create database rangerdb owner ranger;
CREATE DATABASE
postgres=# grant all privileges on database rangerdb to ranger;
GRANT
postgres=# \l
```

List of databases						
Name	Owner	Encoding	Collation	Ctype	Access privileges	
ambari	postgres	UTF8	en_US.UTF-8	en_US.UTF-8	=Tc/postgres : postgres=Ctc/postgres : ambari=Ctc/postgres	
dbname	postgres	UTF8	en_US.UTF-8	en_US.UTF-8	=Tc/postgres : postgres=Ctc/postgres : rangerdba=Ctc/postgres	
postgres	postgres	UTF8	en_US.UTF-8	en_US.UTF-8		
ranger	rangeradmin	UTF8	en_US.UTF-8	en_US.UTF-8	=Tc/rangeradmin : rangeradmin=Ctc/rangeradmin	
rangerdb	ranger	UTF8	en_US.UTF-8	en_US.UTF-8	=Tc/ranger : ranger=Ctc/ranger	
template0	postgres	UTF8	en_US.UTF-8	en_US.UTF-8	=c/postgres : postgres=Ctc/postgres	
template1	postgres	UTF8	en_US.UTF-8	en_US.UTF-8	=c/postgres : postgres=Ctc/postgres	

(7 rows)

注意：不管是安装 postgresql、mysql 还是 oracle，其目的都是为 Ranger 创建数据库来存储策略和用户相关信息，所以实际安装过程中选择一个安装即可。

4 源码编译安装 Ranger

(1) 解压 ranger 安装包

```
tar -zxvf ranger-0.7.0.tar.gz
```

(2) 编译

```
# cd ranger-0.7.0
# mvn clean compile package assembly:assembly install
```

若编译成功，则在安装包的 target 目录下出现以下压缩包：

```
antrun
archive-tmp
maven-shared-archive-resources
ranger-0.7.0-admin.tar.gz
ranger-0.7.0-admin.zip
ranger-0.7.0-atlas-plugin.tar.gz
ranger-0.7.0-atlas-plugin.zip
ranger-0.7.0-hbase-plugin.tar.gz
ranger-0.7.0-hbase-plugin.zip
ranger-0.7.0-hdfs-plugin.tar.gz
ranger-0.7.0-hdfs-plugin.zip
ranger-0.7.0-hive-plugin.tar.gz
ranger-0.7.0-hive-plugin.zip
ranger-0.7.0-kafka-plugin.tar.gz
ranger-0.7.0-kafka-plugin.zip
ranger-0.7.0-kafka-plugin.zip
ranger-0.7.0-kms.tar.gz
ranger-0.7.0-kms.zip
ranger-0.7.0-knox-plugin.tar.gz
ranger-0.7.0-knox-plugin.zip
ranger-0.7.0-migration-util.tar.gz
ranger-0.7.0-migration-util.zip
ranger-0.7.0-ranger-tools.zip
ranger-0.7.0-ranger-tools.zip
ranger-0.7.0-solr-plugin.tar.gz
ranger-0.7.0-solr-plugin.zip
ranger-0.7.0-solr-plugin.zip
ranger-0.7.0-src.tar.gz
ranger-0.7.0-src.zip
ranger-0.7.0-storm-plugin.tar.gz
ranger-0.7.0-storm-plugin.zip
ranger-0.7.0-storm-plugin.zip
ranger-0.7.0-tagsync.tar.gz
ranger-0.7.0-tagsync.zip
ranger-0.7.0-usersync.tar.gz
ranger-0.7.0-usersync.zip
ranger-0.7.0-yarn-plugin.tar.gz
ranger-0.7.0-yarn-plugin.zip
ranger-hdfs-plugin
ranger-hive-plugin
ranger-usersync
ranger-yarn-plugin
rat.txt
version
```

4.1 安装配置 Ranger Admin

在安装 Ranger Admin 之前，需要先安装 Solr，因为 Ranger 在 solr 里存储日志，Ranger Admin UI 依赖 solr 组件完成审计日志的查询，所以需要先安装和配置好 Solr。

(1) 解压安装包

```
tar -zxvf ranger-0.6.0-admin.tar.gz
```

(2) 修改 install.properties 文件

```
DB_FLAVOR=MYSQL
SQL_CONNECTOR_JAR=/usr/share/java/postgresql-jdbc-8.4.704.jar
#数据库管理员属性
db_root_user=root
db_root_password=123456
db_host=localhost
#ranger 的数据库用户
db_name=ranger
db_user=ranger
db_password=123456
#审计存储设置
audit_store=solr
audit_solr_urls=http://10.0.0.66:6083/solr/ranger_audits
audit_solr_user=solr
audit_solr_password=123456
```

(3) 运行执行脚本

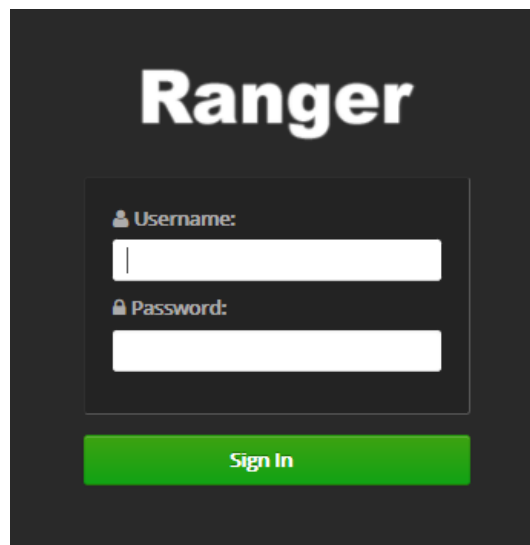
```
./setup.sh
```

(4) 启动服务

```
ranger-admin start
```

(5) 通过使用浏览器访问 http://<host_address>:6080 验证，出现以下页面则表示

安装成功：



初试用户名和密码均是：admin。

4.2 安装配置 Ranger UserSync

Ranger UserSync 的安装配置过程跟 Ranger Admin 一样，我就不再重复了，只是 install.properties 文件配置信息有所不同。

(1) 编辑 install.properties 文件

```
ranger_base_dir = /opt/ranger
hadoop_conf=/home/hadoop-2.7.1/etc/Hadoop
logdir = /var/log/ranger/usersync
```

(2) 安装启动 usersync 服务

```
# ./setup.sh
# ./ranger-usersync-services.sh start
```

若在 Ranger Admin 的用户和用户组管理界面出现 linux 上面的用户即用户源为 external，则表示安装成功。

<input type="checkbox"/>	User Name	Email Address	Role	User Source
<input type="checkbox"/>	admin		Admin	Internal
<input type="checkbox"/>	rangerusersync		Admin	Internal
<input type="checkbox"/>	rangertagsync		Admin	Internal
<input type="checkbox"/>	hive		User	External
<input type="checkbox"/>	nfsnobody		User	External
<input type="checkbox"/>	solr		User	External
<input type="checkbox"/>	ranger		User	External

4.3 安装配置 Ranger Plugins

Apache Ranger 支持的组件很多，安装配置的过程类似，这里选取安装 HDFS 以作示例。

(1) 编辑 install.properties 文件，只列举修改的部分：

```
POLICY_MGR_URL=http://10.0.0.66:6080
SQL_CONNECTOR_JAR=/usr/share/java/postgresql-jdbc-8.4.704.jar
REPOSITORY_NAME=hadoopdev
#审计存储
XAAUDIT.SOLR.ENABLE=true
XAAUDIT.SOLR.URL=http://10.0.0.66:6083/solr/ranger_audits
XAAUDIT.SOLR.USER=solr
XAAUDIT.SOLR.PASSWORD=123456
XAAUDIT.SOLR.ZOOKEEPER=NONE
XAAUDIT.SOLR.FILE_SPOOL_DIR=/var/log/hadoop/hdfs/audit/solr/spool
XAAUDIT.SOLR.IS_ENABLED=true
XAAUDIT.SOLR.MAX_QUEUE_SIZE=1
XAAUDIT.SOLR.MAX_FLUSH_INTERVAL_MS=1000
XAAUDIT.SOLR.SOLR_URL=http://10.0.0.66:6083/solr/ranger_audits
```

(2) 启动 hdfs-plugin

```
# cd /ranger-hdfs-plugin
# ./enable-hdfs-plugin.sh
```

注意：根据脚本文件执行时，在插件安装位置找不到 hadoop conf 和 lib 文件夹，

会导致插件安装失败。报以下两个错误：

- **ERROR:Unable to find the conf directory of component [hadoop]; dir [/opt/hadoop/conf] not found.**

解决方法： 创建一个符号链接作为 hadoop conf 链接到目的目录。

```
ln -s /home/hadoop-2.7.1/etc/hadoop/ /opt/hadoop/conf
```

- **ERROR:Unable to find the lib directory of component [hadoop]; dir [/opt/hadoop/lib] not found.**

解决方法： 将 HDFS Plugin 目录中的 jar 和 Hadoop 中包含 HDFSDE jar 都指向目的目录。

```
# cp ranger-hdfs-plugin/lib/ranger-hdfs-plugin-impl/*.jar
/home/hadoop-2.7.1/share/hadoop/hdfs/lib/

# mkdir /root/hadoop/lib

# ln -s /home/hadoop-2.7.1/share/hadoop/hdfs/lib/
/opt/hadoop/lib/
```

4.4 配置 Solr 提供 Audit 服务

Apache Ranger 使用 Apache Solr 存储审计日志，并通过审计日志提供 UI 搜索^[5]。**注意：Solr 必须安装配置必须在安装 Ranger Admin 或 Ranger Plugins 之前。**

Solr 安装选项：

- Solr -Standalone ， Solr 的单实例易于安装，并且与 Zookeeper 无关。建议使用此选项仅用于测试游侠。
- Solrcloud ，这是 Ranger 的首选设置。SolrCloud 是一种可扩展架构，可以作为单节点或多节点集群运行。它还具有其他功能，如复制和分片，可用于高可用性（HA）和可扩展性。您需要根据群集大小来规划部署。

4.4.1 Solr 安装配置流程

- （1）下载 ranger 安装包：git clone <https://github.com/apache/incubator-ranger.git>;
- （2） 进入 security-admin/contrib/solr_for_audit_setup 目录，根据需要修改 install.properties 文件；
- （3）由于 Apache Ranger 已经写好了安装配置 Solr 的脚本，直接运行 ./setup.sh 下载 slor 安装包。
- （4）打开 \$SOLR_RANGER_HOME/install_notes.txt 根据提示执行命令。

4.4.2 修改 install.properties 文件

以配置 Standalone Solr 为例：

Property Name	Sample values	Description
JAVA_HOME		Provide the path to where you have installed JDK. If it is Hadoop, then you can check /etc/hadoop/conf/hadoop-env.sh for the value of JAVA_HOME. Please note, Solr only support JDK 1.7 and above.
SOLR_USER	solr	The linux user used to run Solr
SOLR_INSTALL_FOLDER	/opt/solr	Location where the Solr is installed. This is the same property used if you want setup.sh to install Solr
SOLR_RANGER_HOME	/opt/solr/ranger_audit_server	This is the location where Ranger related configuration and schema files will be copied
SOLR_RANGER_PORT	6083	The port you want Solr to listen on.
SOLR_DEPLOYMENT	standalone	The value standalone will configure solr to run as standalone.
SOLR_RANGER_DATA_FOLDER	/opt/solr/ranger_audit_server/data	This is the folder where you want the index data to be stored. It is important that the volume for this folder has enough disk space. It is recommended to have at least 1 TB free space for index data. Please take regular backup of this folder.
SOLR_LOG_FOLDER	/var/log/solr/ranger_audits	The folder where where want Solr logs to go. Make sure the volume for this folder has enough disk space. Please delete old log files on regular basis.
SOLR_MAX_MEM	2g	This is the memory assigned for Solr. Make sure you provide adequate memory to the Solr process

SolrCloud 模式修改了两处（注意：若选择该模式，则需先安装 ZooKeeper）：

```
SOLR_DEPLOYMENT = solrcloud  
SOLR_ZK = ${zk_host}:2181/ranger_audits
```

实例：

```
# vim /ranger-0.7.0/security-  
admin/contrib/solr_for_audit_setup/ install.properties  
  
JAVA_HOME=/usr/java/jdk1.8.0_112  
  
SOLR_USER=solr  
  
SOLR_INSTALL=true  
  
SOLR_DOWNLOAD_URL=http://archive.apache.org/dist/lucene/solr/6  
.5.1/solr-6.5.1.tgz  
  
SOLR_INSTALL_FOLDER=/opt/solr  
  
SOLR_RANGER_HOME=/opt/solr/ranger_audit_server  
  
SOLR_DEPLOYMENT=standalone  
  
SOLR_RANGER_DATA_FOLDER=/opt/solr/ranger_audit_server/data
```

4.4.3 下载 Solr 安装包

配置完之后运行命令安装 solr：

```
[root@access-master solr_for_audit_setup]# ./setup.sh
Tue Jun 6 15:54:40 CST 2017|INFO|Downloading solr from http://archive.apache.org/dist/lucene/solr/6.5.1/solr-6.5.1.tgz
--2017-06-06 15:54:41-- http://archive.apache.org/dist/lucene/solr/6.5.1/solr-6.5.1.tgz
Resolving archive.apache.org... 163.172.17.199
Connecting to archive.apache.org|163.172.17.199|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 148699036 (142M) [application/x-gzip]
Saving to: ? . olr-6.5.1.tgz?

100%[=====>] 148,699,036 660K/s in 4m 36s

2017-06-06 15:59:17 (527 KB/s) - ? . olr-6.5.1.tgz? . saved [148699036/148699036]

Tue Jun 6 15:59:20 CST 2017|INFO|Installed Solr in /opt/solr
Tue Jun 6 15:59:20 CST 2017|INFO|Configuring SolrCloud instance
Tue Jun 6 15:59:20 CST 2017|INFO|Creating group solr
Tue Jun 6 15:59:21 CST 2017|INFO|Creating user solr
Tue Jun 6 15:59:21 CST 2017|INFO|Done configuring Solr for Apache Ranger Audit
Tue Jun 6 15:59:21 CST 2017|INFO|Solr HOME for Ranger Audit is /opt/solr/ranger_audit_server
Tue Jun 6 15:59:21 CST 2017|INFO|To start Solr run /opt/solr/ranger_audit_server/scripts/start_solr.sh
Tue Jun 6 15:59:21 CST 2017|INFO|To stop Solr run /opt/solr/ranger_audit_server/scripts/stop_solr.sh
Tue Jun 6 15:59:21 CST 2017|INFO|After starting Solr for RangerAudit, it will listen at 6083. E.g http://access-master:6083
Tue Jun 6 15:59:21 CST 2017|INFO|Configure Ranger to use the following URL http://access-master:6083/solr/ranger_audits
Tue Jun 6 15:59:21 CST 2017|INFO|Please refer to /opt/solr/ranger_audit_server/install_notes.txt for instructions for setting up collections in SolrCloud
Tue Jun 6 15:59:21 CST 2017|INFO| ** NOTE: If Solr is Secured then solrclient JAAS configuration has to be added to Ranger Admin and Ranger Plugin properties
Tue Jun 6 15:59:21 CST 2017|INFO| ** Refer documentation on how to configure Ranger for audit to Secure Solr
##### Done #####
Created file /opt/solr/ranger_audit_server/install_notes.txt with instructions to start and stop
#####
```

4.4.4 启动 Solr 服务

打开/opt/solr/ranger_audit_server/install_notes.txt 文件，里面写好了启动配置步骤：

(1) 将 ranger 审计配置添加进 zookeeper，执行如下命令：

```
#/opt/solr/ranger_audit_server/scripts/add_ranger_audits_conf_
to_zk.sh
```

(2) 启动 solr 服务

```
# /opt/solr/ranger_audit_server/scripts/start_solr.sh
```

后面两个步骤是 SolrCloud 模式模式下使用的。

(3) 创建 ranger 审计 collection：

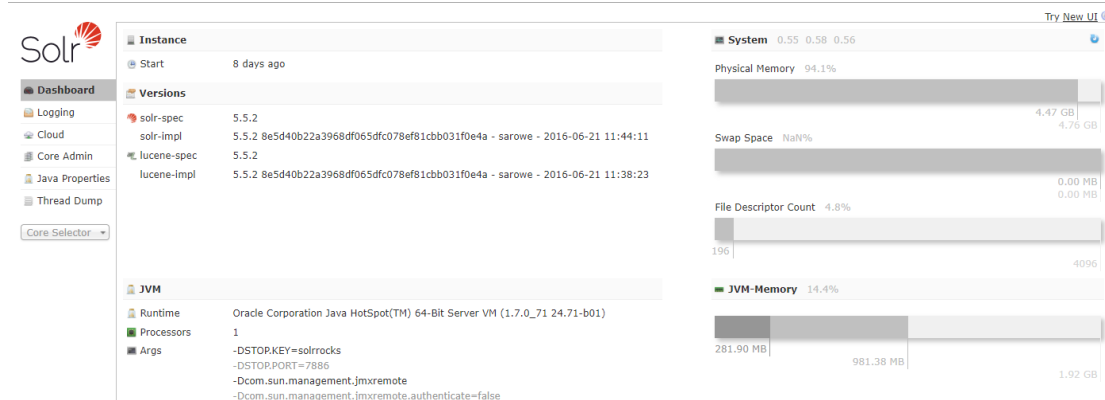
```
vim
/opt/solr/ranger_audit_server/create_ranger_audits_collection.
sh

添加 SOLR_RANGER_PROT=6083
```

(4) 启动命令：

```
./create_ranger_audits_collection.sh
```

(5) 在浏览器上打开 solr: <http://hostname:6083>。



5 使用 Ambari 部署 Ranger

在 Ambari 的 Hadoop 集群上安装 Ranger 相对简单，很多配置工作 Ambari 都帮我们做了，但是在安装 Ranger 之前，有一些工作我们仍然需要提前做好。主要是以下两方面：

- 安装数据库，创建 Ranger 数据库；
- 安装 solr。

5.1 配置数据库

虽然在安装 Ambari 时就已经安装了一个数据库，但这里还是需要进行一些配置^[6]。该步骤与前面手动安装配置数据库还是有些不同。

5.1.1 安装 jdbc 连接驱动

```
yum install postgresql-jdbc*
```

5.1.2 修改配置文件

```
vim /var/lib/pgsql/data/pg_hba.conf
```

```
# TYPE  DATABASE  USER      CIDR-ADDRESS  METHOD
# "local" is for Unix domain socket connections only
local   all   postgres  ident
# IPv4 local connections:
host    all   postgres  127.0.0.1/32  ident
# IPv6 local connections:
host    all   postgres  ::1/128      ident

local   all   ambari,mapred md5
host    all   ambari,mapred 0.0.0.0/0    md5
host    all   ambari,mapred ::/0      md5
host    all   all         10.0.0.45/32 trust
```

重启 mysql 服务使配置生效:

```
service postgresql restart
```

5.1.3 重启 ambari-server

```
ambari-server setup --jdbc-db=postgres --jdbc-driver=/usr/share/java/postgresql.jar
```

5.1.4 在环境变量中添加 jdbc 驱动路径

```
#vim /etc/profile

export
HADOOP_CLASSPATH=${HADOOP_CLASSPATH}:${JAVA_JDBC_LIBS}:/usr/share/java/postgresql-jdbc-8.4.704.jar

#source /etc/profile //生效
```

5.2 安装 solr

- 打开 Ambari, 在 Actions 下点击 Add Service, 在打开的服务列表里勾选 Ambari infra:

Add Service Wizard

<input checked="" type="checkbox"/>	ZooKeeper	3.4.6	Centralized service which provides highly reliable distributed coordination
<input type="checkbox"/>	Falcon	0.10.0	Data management and processing platform
<input type="checkbox"/>	Storm	1.0.1	Apache Hadoop Stream processing framework
<input type="checkbox"/>	Flume	1.5.2	A distributed service for collecting, aggregating, and moving large amounts of streaming data into HDFS
<input type="checkbox"/>	Accumulo	1.7.0	Robust, scalable, high performance distributed key/value store.
<input checked="" type="checkbox"/>	Ambari Infra	0.1.0	Core shared service used by Ambari managed components.
<input type="checkbox"/>	Ambari Metrics	0.1.0	A system for metrics collection that provides storage and retrieval capability for metrics collected from the cluster

- 安装好了启动 solr 服务:

Summary Configs Quick Links ▾

Summary

[Infra Solr Instance](#) ✔ Started No alerts

[Infra Solr Clients](#) 3 Infra Solr Clients Installed

5.3 配置 Ranger

- (1) 打开 Ambari, 在 Actions 下点击 Add Service, 在打开的服务列表里, 勾选 Ranger, 添加 Ranger 服务。

Add Service Wizard

<input type="checkbox"/>	Accumulo	1.7.0	Robust, scalable, high performance distributed key/value store.
<input type="checkbox"/>	Ambari Infra	0.1.0	Core shared service used by Ambari managed components.
<input type="checkbox"/>	Ambari Metrics	0.1.0	A system for metrics collection that provides storage and retrieval capability for metrics collected from the cluster
<input type="checkbox"/>	Atlas	0.7.0	Atlas Metadata and Governance platform
<input type="checkbox"/>	Kafka	0.10.0	A high-throughput distributed messaging system
<input type="checkbox"/>	Knox	0.9.0	Provides a single point of authentication and access for Apache Hadoop services in a cluster
<input type="checkbox"/>	Log Search	0.5.0	Log aggregation, analysis, and visualization for Ambari managed services. This service is Technical Preview .
<input checked="" type="checkbox"/>	Ranger	0.6.0	Comprehensive security for Hadoop
<input type="checkbox"/>	Ranger KMS	0.6.0	Key Management Server
<input type="checkbox"/>	SmartSense	1.3.0.0-22	SmartSense - Hortonworks SmartSense Tool (HST) helps quickly gather configuration, metrics, logs from common HDP services that aids to quickly troubleshoot support cases and receive cluster-specific recommendations.
<input type="checkbox"/>	Spark	1.6.2	Apache Spark is a fast and general engine for large-scale data processing.
<input type="checkbox"/>	Spark2	2.0.0	Apache Spark 2.0 is a fast and general engine for large-scale data processing. This service is Technical Preview .
<input type="checkbox"/>	Zeppelin Notebook	0.6.0	A web-based notebook that enables interactive data analytics. It enables you to make beautiful data-driven, interactive and collaborative documents with SQL, Scala and more.
<input type="checkbox"/>	Mahout	0.9.0	Project of the Apache Software Foundation to produce free implementations of distributed or otherwise scalable machine learning algorithms focused primarily in the areas of collaborative filtering, clustering and classification
<input type="checkbox"/>	Slider	0.91.0	A framework for deploying, managing and monitoring existing distributed applications on YARN.

(2) 配置

目前主要配置这些项目：Ranger Admin、Ranger User Info、Ranger Plugin、Ranger Audit。其中在安装配置页面，Ranger Admin 下，

- DB FLAVOR : 选择 POSTGRES
- Ranger DB host : 选择 Postgres 安装的主机名
- Ranger DB password : 输入在 Postgresql 中 ranger 数据库拥有者的密码，这里即 rangeradmin 的密码；
- DBA username: 相当于数据库超级管理员，这里是 postgres，在 mysql 中是 root；
- DBA password: 输入 postgres 的密码
- JDBC connect string :
jdbc:postgresql://10.0.0.45:5432/postgres

配置完成后，点击 Test Connection。具体如下图所示。

DB FLAVOR <div>POSTGRES</div>	Ranger DB host <div>10.0.0.45</div>
Ranger DB name <div>ranger</div>	Driver class name for a JDBC Ranger database <div>org.postgresql.Driver</div>
Ranger DB username <div>rangeradmin</div>	Ranger DB password <div>.....</div> <div>.....</div>
JDBC connect string for a Ranger database <div>jdbc:postgresql://10.0.0.45:5432/ranger</div>	

Setup Database and Database User

Yes

Database Administrator (DBA) username <div>postgres</div>	Database Administrator (DBA) password <div>.....</div> <div>.....</div>
JDBC connect string for root user <div>jdbc:postgresql://10.0.0.45:5432/postgres</div>	

Test Connection

 Connection OK

在配置完所有项目之后, 开启所有 Ranger 服务, 出现以下情况则启动成功, 可以正常使用 Ranger 了。

Summary		No alerts
Ranger Admin	Started <div>No alerts</div>	
Ranger Usersync	Started <div>No alerts</div>	
Ranger Tagsyncs	1/1 Started	
Ranger HDFS plugin	Enabled	
Ranger YARN plugin	Enabled	

5.4 配置 Ranger User Sync

Ranger 提供了用户/组同步功能, 其中用户数据源支持 UNIX、File、LDAP/AD 三种形式。下面就依次来介绍这三种方式设置用户同步的方法。

Ranger AdminRanger User InfoRanger PluginRanger AuditRanger TagsyncAdvanced

Ranger User Info

Enable User Sync

Yes

Sync Source

UNIX

UNIX

FILE

LDAP/AD

5.4.1 配置 Linux 环境下的用户/组同步映射

若选择 Linux 环境下的用户/组数据源，配置比较简单。首先在 Linux 环境下创建用户和用户组，然后在 Ranger 用户信息配置页面填写 Linux 下保存组的文件路径，如下图所示。

Sync Source

UNIX

Minimum User ID

500

Password File

/etc/passwd

Group File

/etc/group

5.4.2 配置文件中用户/组映射

若选择使用文件作为 Ranger 上用户/组的数据源，则在下面页面中填写用户/组所在的文件，以及文件中分隔符表示：

Sync Source

FILE

File Name

/tmp/usergroup.txt

Delimiter

,

26 / 50

5.4.3 配置 LDAP/AD 中的用户/组映射

若选择 LDAP/AD 作为用户来源，首先要保证 Hadoop 与 LDAP/AD 上用户/组的映射。在确保 LDAP/AD 与 Hadoop 上的用户/组映射成功后，在 Ranger 的配置页面配置三部分信息：通用配置，用户配置和组配置。

Sync Source

LDAP/AD

Common Configs

User Configs

Group Configs

LDAP/AD URL

ldap://10.0.0.45

Bind Anonymous

No

Bind User

cn=Manager,dc=hadoop,dc=apache

Bind User Password

(1) 通用配置

表2 LDAP 通用配置信息

配置项	描述	默认值	示例
LDAP/AD URL	根据 LDAP/AD 同步源添加 URL	ldap://{host}:{port}	ldap:// ldap.example.com:389
Bind Anonymous	若选择是，则不需添加用户名和密码	NO	
Bind User	Linux 服务器上保存用户组所在文件路径		cn=admin, dc=example, dc=com 或 admin@example.com
Bind User Password	绑定用户的密码		

(2) 用户配置

表3 用户配置信息

配置项	描述	默认值	示例
-----	----	-----	----

Username Attribute	LDAP 上用户名称属性		OpenLDAP 上的 cn
User Object Class	识别用户条目的对象类	Person	top,person,user
User Search Base	用户的搜索项		cn=user,dc=example,dc=com
User Search Filter	（可选项）限制选择用于同步用户的过滤器来		cn=*
User Search Scope	设置用于将用户搜索限制选择搜索基础的深度		base,one 或者 sub
User Group Name Attribute	用户条目的属性		memberof, ismemberof
Group User Map Sync	若设置为否，则属于“用户组名称属性”选项中的用户所属的组	NO	

[Common Configs](#) [User Configs](#) [Group Configs](#)

Username Attribute

cn

User Object Class

person

User Search Base

ou=people,dc=hadoop,dc=apache

User Search Filter

cn=*

User Search Scope

sub

User Group Name Attribute

memberof, ismemberof

Group User Map Sync

Yes

(3) 组配置

表4 组配置信息

配置项	描述	默认值	示例
Enable Group Sync	若设置为“NO”，则属于“用户组名称属性”的用户所属组名称； 若设置为“YES”，则使用以下与组相关的属性从 LDAP/AD 中检索用户所属的组。	NO	
Group Member Attribute	LDAP 组成员属性名称		member
Group Name Attribute	LDAP 组名称属性		OpenLDAP 上的 cn
Group Object Class	LDAP 组对象类		group,groupofnames, 或者 posixGroup
Group Search Base	组的搜索项		ou=groups,DC=example,DC=com
Group Search Filter	(可选项) 限制选择同步组的过滤器		cn=*
Enable Group Search First	选择“启用组搜索优先”时，有两种用户检索方式： •如果未选择“启用用户搜索”：将从组的“成员”属性中检索用户。 •如果选择启用用户搜索：通过基于用户配置执行 LDAP 搜索来计算用户成员资格。	NO	

Common Configs

User Configs

Group Configs

Enable Group Sync

Yes

Group Member Attribute

memberUid

Group Name Attribute

cn

Group Object Class

posixGroup

Group Search Base

ou=group,dc=hadoop,dc=apache

Group Search Filter

cn=*

Enable Group Search First

No

5.5 配置 Solr 提供审计

登录 Ambari ，修改 Ranger 的配置如下（目前只修改了红框中的内容）：

Audit to Solr

Audit to Solr

ON

SolrCloud

ON

External SolrCloud

ON

External SolrCloud kerberos

OFF

ranger.audit.solr.zookeepers

10.0.0.66:2181/ranger_audits

ranger.audit.solr.username

ranger_solr

ranger.audit.solr.password

Audit to HDFS

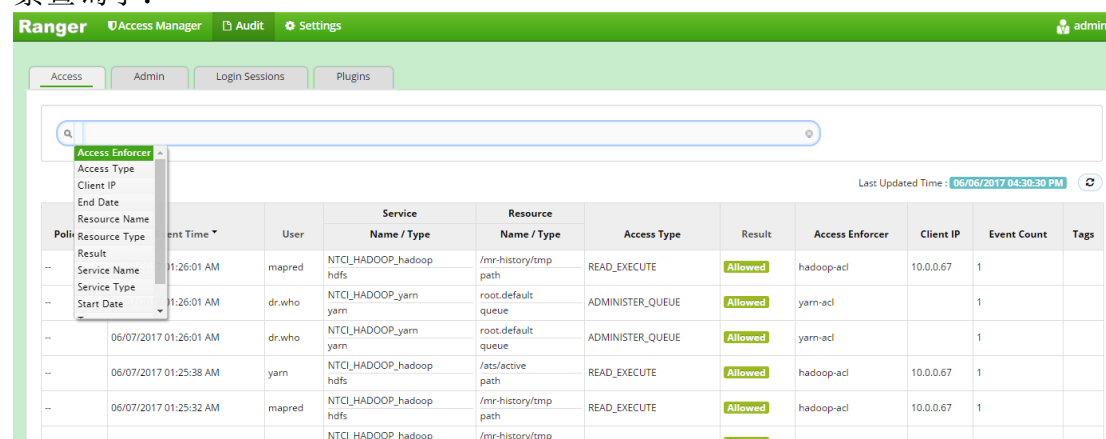
Audit to HDFS

ON

Destination HDFS Directory

hdfs://access-master:8020/ranger/audit

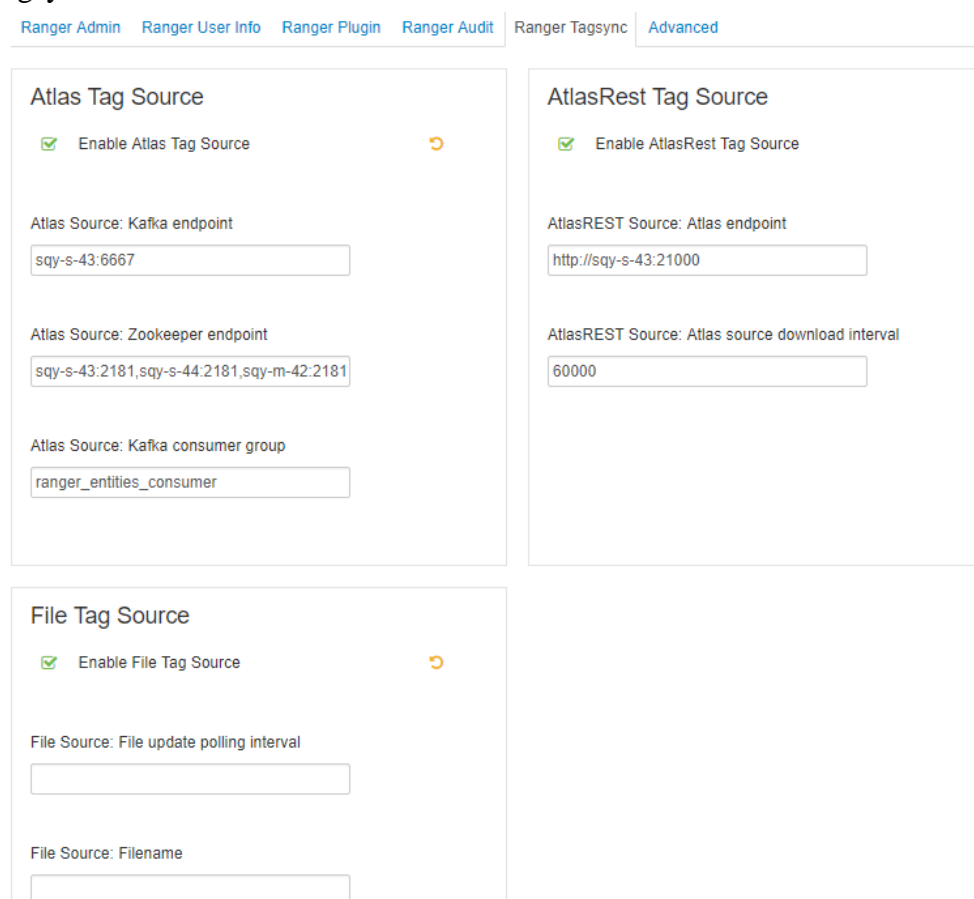
保存了配置之后，登录到 Ranger web 界面审计部分，就能对访问日志进行检索查询了：



Policy	Action	Time	User	Service	Resource	Access Type	Result	Access Enforcer	Client IP	Event Count	Tags
...	...	06/07/2017 01:26:01 AM	mapred	NTCL_HADOOP_hadoop	/mr-history/tmp	READ_EXECUTE	Allowed	hadoop-acl	10.0.0.67	1	
...	...	06/07/2017 01:26:01 AM	dr.who	NTCL_HADOOP_yarn	root.default	ADMINISTER_QUEUE	Allowed	yarn-acl		1	
...	...	06/07/2017 01:26:01 AM	dr.who	NTCL_HADOOP_yarn	root.default	ADMINISTER_QUEUE	Allowed	yarn-acl		1	
...	...	06/07/2017 01:25:38 AM	yarn	NTCL_HADOOP_hadoop	/ats/active	READ_EXECUTE	Allowed	hadoop-acl	10.0.0.67	1	
...	...	06/07/2017 01:25:32 AM	mapred	NTCL_HADOOP_hadoop	/mr-history/tmp	READ_EXECUTE	Allowed	hadoop-acl	10.0.0.67	1	
...	...	06/07/2017 01:25:01 AM	mapred	NTCL_HADOOP_hadoop	/mr-history/tmp	READ_EXECUTE	Allowed	hadoop-acl	10.0.0.67	1	

5.6 配置 Ranger Tagsync

Ranger 支持三种标签来源：File、Atlas 或 AtlasREST。可以选择任意一种作为 Tagsync 源。



Atlas Tag Source
☒ Enable Atlas Tag Source
Atlas Source: Kafka endpoint

Atlas Source: Zookeeper endpoint

Atlas Source: Kafka consumer group

AtlasRest Tag Source
☒ Enable AtlasRest Tag Source
AtlasREST Source: Atlas endpoint

AtlasREST Source: Atlas source download interval

File Tag Source
☒ Enable File Tag Source
File Source: File update polling interval

File Source: Filename

若选择 Atlas 或 AtlasREST 作为标签的来源，则要提前安装配置好 Atlas。下面是 Atlas 的安装配置步骤。

5.7 安装配置 Atlas

Apache Ranger 引入了一种称为“标签”的新服务类型来处理基于标签的策略。使用 Apache Atlas 为资源标记分类，在 Apache Ranger 中为创建基于标签的策略，即只针对标签授权。其中基于标签的策略将资源分类与访问授权相分离，资源被标记后，标签的授权将自动执行，无需创建或更新资源的策略。Ranger 中有一个 Tagsync 模块，可以同步 Apache Atlas 上的标签，用户直接根据标签为用户或用户组访问策略。

5.7.1 Hive 数据导入 Atlas

利用 Ambari 安装 Atlas，在添加 Atlas 服务之前，HBase 和 Kafka 服务需要安装好。这里使用 Atlas 给 Hive 中的数据分类，因此在安装好 Atlas 之后，需要配置将 Hive 的元数据导入 Atlas 中^[7]。具体的配置细节如下：

(1) 配置环境变量：

```
export HADOOP_HOME
export HIVE_CONF_DIR
export HADOOP_CLASSPATH
```

(2) 在 hive-env.sh 文件中添加如下变量：

```
export HIVE_AUX_JARS_PATH=<atlas package>/hook/hive
```

(3) 在<atlas package>/conf/atlas-application.properties 的文件中设置以下参数：

```
atlas.hook.hive.synchronous=true
```

(4) 将 atlas-application.properties 文件拷贝到 hive/conf 文件目录下。

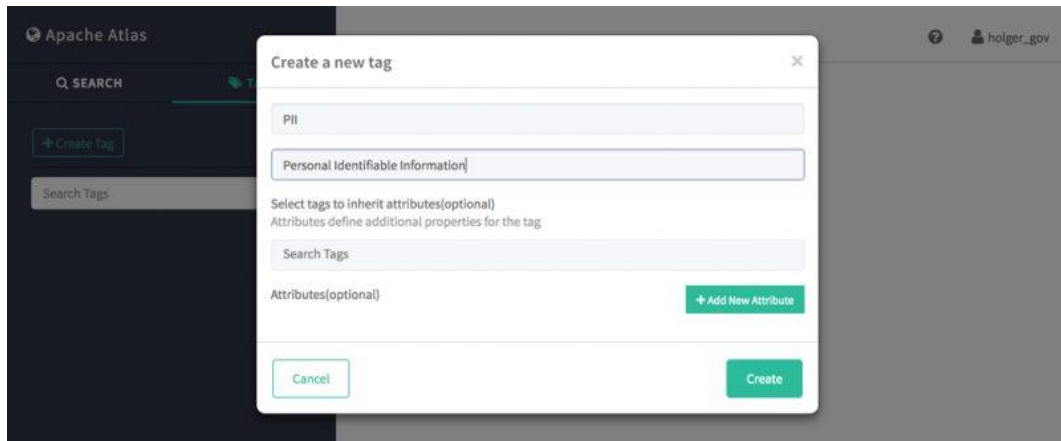
(5) 启动 atlas 安装包下的将 Hive 元数据导入 Atlas 的脚本命令：

```
# cd <atlas package>/hook-bin
# ./import-hive.sh
```

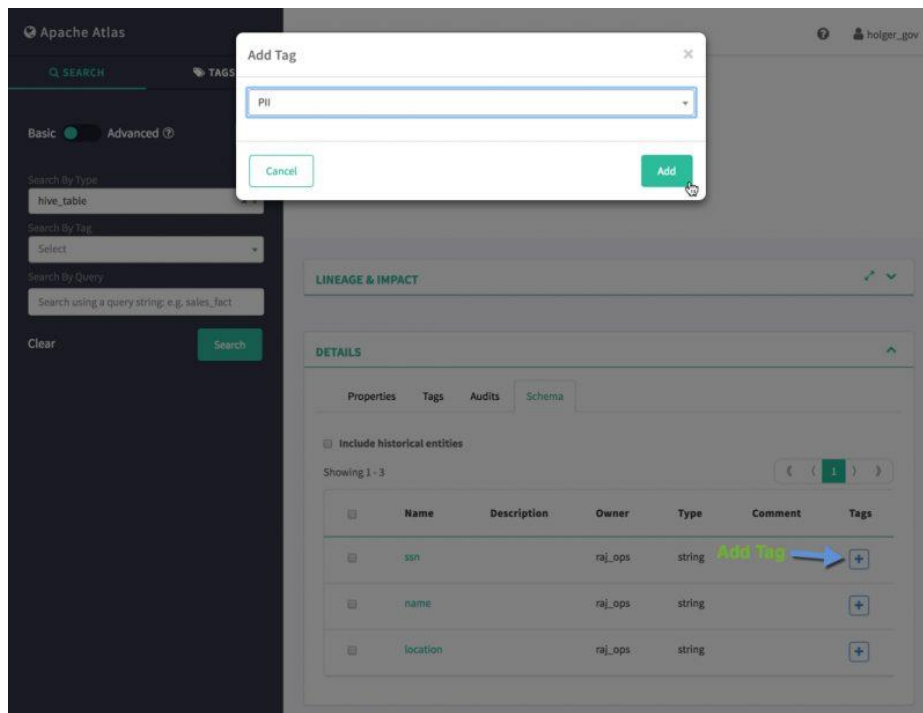
5.7.2 创建标签

使用 Atlas 对数据进行分类，也就是为数据设置标签属性。首先使用 <http://localhost:21000> 登录 Atlas 的 UI 界面。

- 创建一个标签 PII，并为其添加属性；

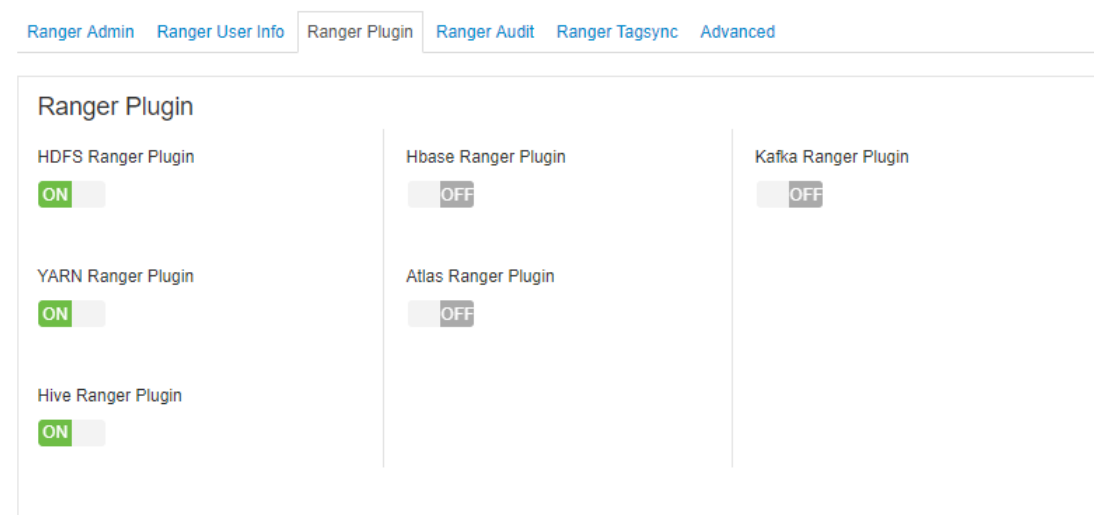


- 为一个实体打上标签 PII，比如 hdfs 文件或文件夹，hive 数据库、表、列等。面的示例是为 hive 表 employee 中的列名为 ssn 赋予标签 PII；

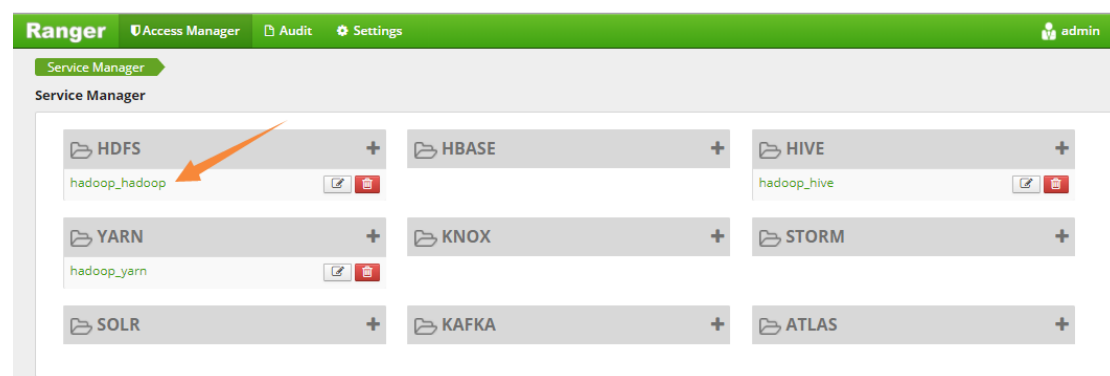


5.8 添加 Ranger 插件

使用 Ambari 上添加 Ranger 插件就简单多了，只需要在 Ranger Plugin 页面选择想要被提供服务的组件即可：

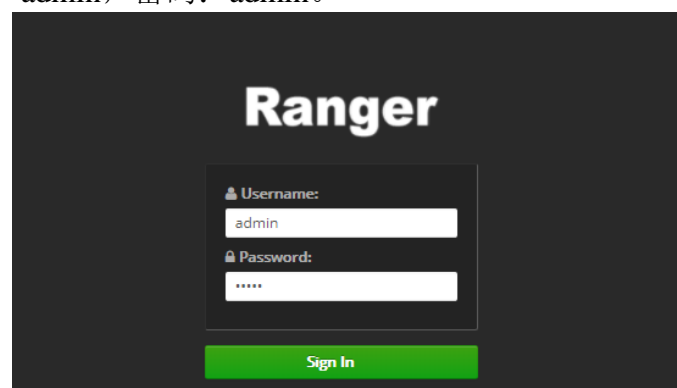


选择了之后保存修改，然后重启相应的组件服务，在 Ranger 管理界面上自动生成了相应组件的策略库。

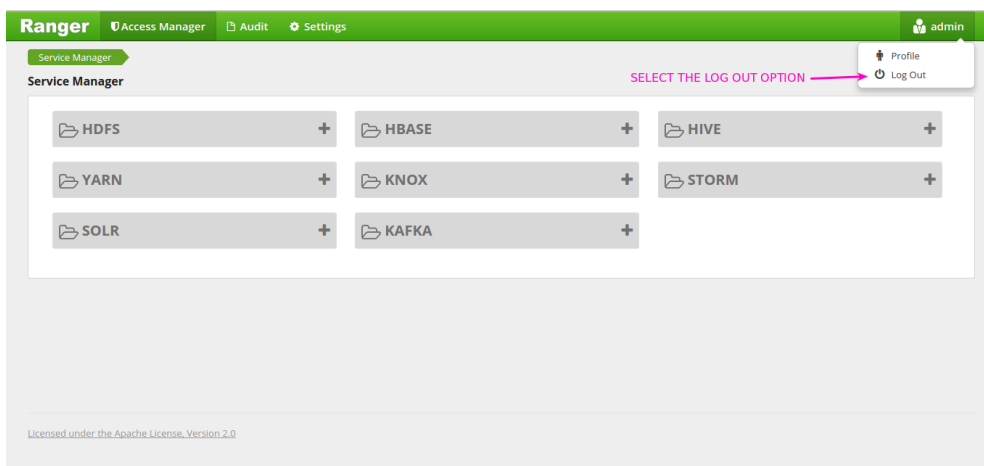


6 使用 Ranger 为 Hadoop 授权管理

当前使用版本 Ranger 0.6.0，Ranger UI 管理页面：<http://localhost:6080>。输入默认用户名：admin，密码：admin。



登录到系统主页，可以通过页面右侧的 Log Out 选项退出登录，Profile 选项修改个人信息。



Ranger 管理主页主要包括有以下几大模块及其子模块：

■ Access Manager

- ✧ **Resource Based Policies:** 提供给安全管理员对各个组件服务基于资源的策略的管理，包括策略的增删改查。
- ✧ **Tag Based Policies:** 提供给安全管理员对基于标签策略的增删改查。
- ✧ **Reports:** 列出了所有组件下的全部策略，并提供根据策略名称、策略类型、组件类型和用户/组这些方式来搜索策略。

■ Audit 记录了用户对资源的访问日志、对策略/服务的操作日志、Ranger UI 的登录日志以及 ranger 插件的状态，并提供了根据不同条件的搜索功能。

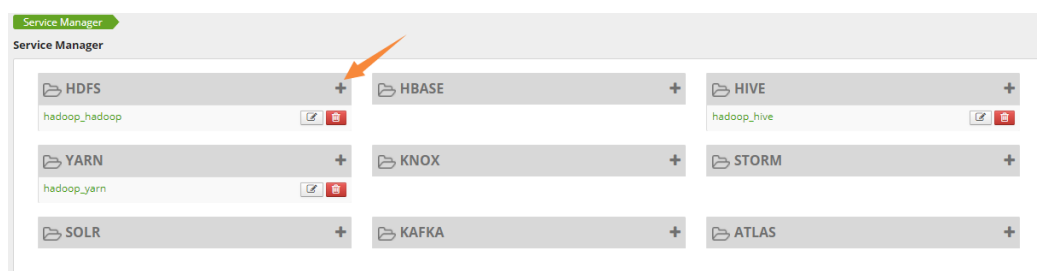
■ Settings

- ✧ **Users/Groups** 管理用户/用户组
- ✧ **Permissions** 管理用户/用户组对 Ranger 界面上各个模块的访问权限。

6.1 基于资源的策略管理

6.1.1 管理 HDFS 策略

6.1.1.1 创建策略存储仓库



进入策略存储仓库编辑页面，主要填写的配置项信息^[2]如下：

- **Service Name:** 存储仓库名称
- **Username:** 可用于连接的终端系统用户名
- **Password:** 用户对应的密码
- **NameNode URL:** 连接 namenode 的 url
- **Authentication Type:** HDFS 的身份认证方式，默认是 simple，若选择 Kerberos 则需要填写相关认证的配置信息。

Ranger Access Manager Audit Settings

Service Details :

Service Name *

Description

Active Status ☒ Enabled ☐ Disabled

Select Tag Service

Config Properties :

Username *

Password *

NameNode URL *

Authorization Enabled

Authentication Type *

hadoop.security.auth_to_local

dfs.datanode.kerberos.principal

dfs.namenode.kerberos.principal

dfs.secondary.namenode.kerberos.principal

RPC Protection Type

Common Name for Certificate

Add New Configurations

Name	Value
<input type="text"/>	<input type="text"/>

6.1.1.2 创建策略

策略存储仓库创建成功了之后，点击 HDFS 策略仓库名，对策略进行操作。

Ranger Access Manager Audit Settings admin

Service Manager > hadoop_hadoop Policies

List of Policies : hadoop_hadoop

Search for your policy...

Policy ID	Policy Name	Status	Audit Logging	Groups	Users	Action
1	all - path	Enabled	Enabled	--	hadoop ambari-qa	<input type="button" value="Edit"/> <input type="button" value="Delete"/>

点击新建策略，进入编辑策略页面：

Ranger
Access Manager
Audit
Settings
admin

Service Manager
hadoop_hadoop Policies
Create Policy

Create Policy

Policy Details :

Policy Type
Access

Policy Name *
enabled

Resource Path *

Recursive
ON

Audit Logging
YES

Description

Allow Conditions :

Select Group

Select User

Add/edit permissions
☒ Read
☐ Write
☐ Execute
☐ Select/Deselect All

Add Permissions

Delegate Admin
☐

上图中策略项的含义如下表所示：

表5 策略项信息

策略项	描述
Policy Type	基于资源的策略主要分三种类型：access、masking、row level filter，后两种主要在 hive 中使用。
Policy Name	策略名称，用于标识当前策略。
Resource path	当前资源类型为 path，定义了文件/文件夹的资源路径，表示要进行访问权限限制的对象。
Description	对当前策略进行描述说明。
Recursive	表示是否指定文件夹下所有文件都在该策略下。
Audit Logging	指定该策略是否被审计。
Group Permissions	从用户组列表中选择用户组，指定该组的权限。
User Permissions	从用户列表中选择用户，指定用户的权限。
Add/edit Permission	HDFS 操作权限，包括读、写、可执行这三种操作。
Delegate Admin	将策略分配给用户或用户组后，指定其是否可以代管该策略，即对该策略进行增删改查。

Enable/disable	默认情况下，策略已启用。也可以禁用策略来限制该策略的用户/组访问权限。
----------------	-------------------------------------

6.1.2 管理 HIVE 策略

6.1.2.1 创建策略存储仓库

与创建 HDFS 策略仓库的步骤类似，但创建 Hive 策略仓库时需要填写连接 hiveserver2 的 url。

Create Service

Service Details :

Service Name *

Description

Active Status ☒ Enabled ☐ Disabled

Select Tag Service

Config Properties :

Username *

Password *

jdbc.driverClassName *

jdbc.url *

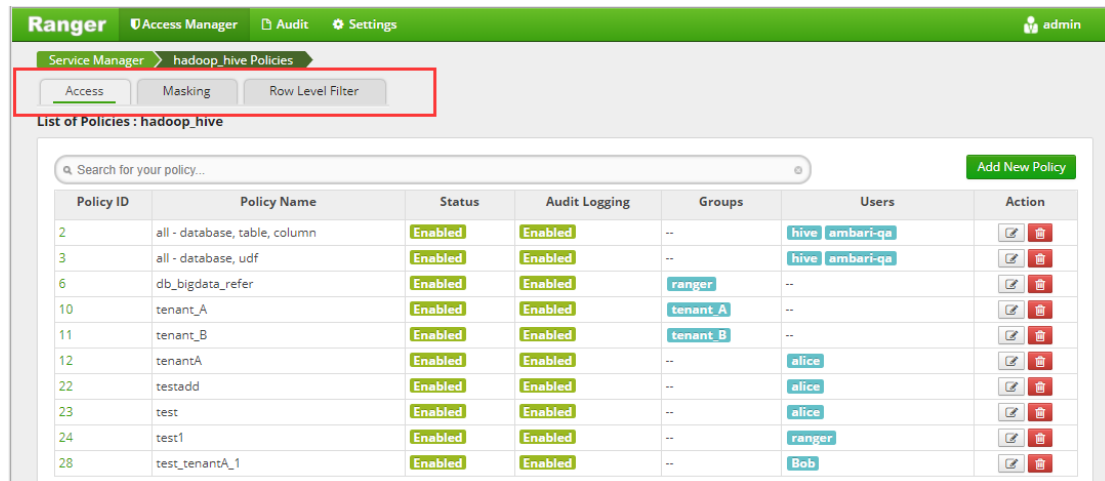
Common Name for Certificate

Add New Configurations

Name	Value
<input type="text"/>	<input type="text"/>

6.1.2.2 创建策略

点击 Hive 策略存储仓库进入策略管理页面，hive 的策略类型相对比较复杂。Ranger 不仅支持对 hive 数据库和表的访问权限控制，还支持对行级过滤、列级数据脱敏。因此，hive 策略类型包括三种：访问、列级脱敏、行级过滤。后两者策略生效的前提是具有访问策略。也就是授权用户首先要有访问权限，才能进一步限制访问权限。下面就根据策略类型来介绍。



一、编辑访问策略

下面列出特定策略项信息：

表6 Hive 策略项

策略项	描述
Hive database	指定授权的资源数据库名
table	对于选定的数据库，指定授权的资源表名
Hive column	对于选定的数据库和表，指定授权的资源列名
Add/edit Permission	指定对数据库、表、列的操作权限
Polidy Condition	指定该策略的上下文条件

Create Policy

Policy Details :

Policy Type
Access

Policy Name *
☐ enabled

Hive Database *
☐ include

table

 *
☐ include

Hive Column *
☐ include

Audit Logging
YES

Description

Add/edit permissions

- ☐ select
- ☐ update
- ☐ Create
- ☐ Drop
- ☐ Alter
- ☐ Index
- ☐ Lock
- ☐ All
- ☐ Select/Deselect All

Allow Conditions :

Select Group	Select User	Policy Condition	Delegate Admin
<input type="text"/>	<input type="text"/>	<input type="checkbox"/> Add Conditions	<input type="checkbox"/> Add Permissions
<input type="button" value="+"/>			<input type="button" value="X"/>

二、编辑列级数据脱敏策略

列级数据脱敏策略与访问策略除了策略类型不同，还需要选择脱敏方式，如下图所示 3.5 所示。

The screenshot shows the 'Create Policy' interface. The 'Policy Details' section has a 'Policy Type' dropdown set to 'Masking'. Below it are fields for 'Policy Name', 'Hive Database', 'Hive Table', and 'Hive Column', each with a text input and a required asterisk. There is an 'enabled' toggle switch and an 'Audit Logging' toggle set to 'YES'. A 'Description' text area is at the bottom. The 'Mask Conditions' section contains a table with columns for 'Select Group', 'Select User', 'Policy Conditions', and 'Access Type'. A 'Select Masking Option' dropdown menu is open, showing options like 'Mask', 'Partial mask: show last 4', 'Partial mask: show first 4', 'Hash', 'NULL', 'No masking', and various date masking options. At the bottom are 'Add' and 'Cancel' buttons.

使用 Ranger 列屏蔽策略来屏蔽在 Hive 查询结果中返回的数据时，以下条件适用：

- 可以使用各种掩码类型，例如显示最后 4 个字符，显示前 4 个字符，散列和日期掩码（仅显示年份）。
- 可以为特定用户，组和条件指定掩蔽类型。
- 不支持通配符匹配。
- 每列应该有自己的掩蔽策略。
- 按策略列出的顺序进行评估。
- 每次将屏蔽策略应用于列时，都会生成审核日志条目。

三、编辑行级过滤策略

行级过滤有助于简化 Hive 查询。通过将访问限制逻辑下移到 Hive 层，Hive 每次尝试访问数据时都会应用访问限制。

Create Policy

Policy Details :

Policy Type **Row Level Filter**

Policy Name * **enabled**

Hive Database *

Hive Table *

Audit Logging **YES**

Description

Row Filter Conditions :

Select Group	Select User	Policy Conditions	Access Type
Select Group	Select User	Add Conditions +	select
		Add Row Filter +	

Enter filter expression

Enter expression

输入过滤表达式，相当于sql中的where子句

不支持通配符

Add Cancel

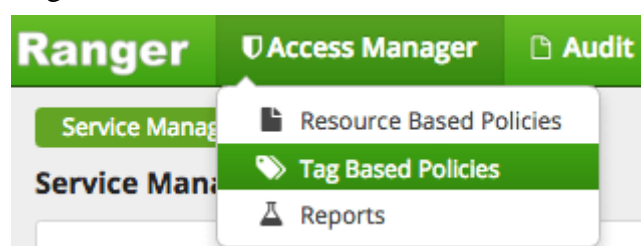
使用行级过滤器时，需满足以下条件：

- 过滤器表达式必须是表或视图的有效 **WHERE** 子句。
- 每个表或视图都应该有自己的行级过滤策略。
- 数据库或表名不支持通配符匹配。
- 过滤器按照策略中列出的顺序进行评估。
- 每次将行级过滤器应用于表或视图时，都会生成审核日志条目。

6.2 管理基于标签的策略

6.2.1 创建策略存储仓库

在 Access Manager 下选择基于标签的策略子模块：



创建基于标签的策略存储仓库，这里没有过多的配置信息需要填写。

Create Service

Service Details :

Service Name *

TagPolicy

Description

Active Status

☒ Enabled
 ☐ Disabled

Add

Cancel

6.2.2 创建基于标签的策略

现在就可以在创建好的标签策略仓库中创建策略了，下面解释几个策略项中比较特别的项目：

- **TAG:** 标签名，授予该标签的权限，表示拥有该标签所对应的所有资源的权限。
- **Component Permissions:** 该策略是全局策略，针对 Ranger 支持的所有组件，不再只针对某一个组件，因此该策略项指定了该条策略的权限范围。

Create Policy

Policy Details :

Policy Type

Access

Policy Name *

TAG *

Audit Logging

YES

Description

Allow Conditions :

hide

Select Group	Select User	Policy Conditions	Component Permissions	
Select Group	Select User	Add Conditions +	Add Permissions +	✕

+

Exclude from Allow Conditions :

show

Deny Conditions :

hide

Select Group	Select User	Policy Conditions	Component Permissions	
Select Group	Select User	Add Conditions +	Add Permissions +	✕

+

Exclude from Deny Conditions :

show

6.2.3 为组件添加基于标签的策略仓库

进入组件的策略存储仓库编辑页面，在 **Select Tag Service** 选项，选择支持的基于标签的策略仓库。以 **Hive** 为例，除了支持基于资源的策略，同时支持基于标签的策略。即基于标签的策略能在用户访问 **Hive** 时生效。

Edit Service

Service Details :

Service Name *

hadoop_hive

Description

hive repo

Active Status

Enabled

Disabled

Select Tag Service

TagPolicy

Config Properties :

Username *

hive

Password *

jdbc.driverClassName *

org.apache.hive.jdbc.HiveDriver

jdbc.url *

jdbc:hive2://sqy-m-42:10000

Common Name for Certificate

Add New Configurations

Name	Value	
policy.grantrevoke.auth.users	hive	✖
tag.download.auth.users	hive	✖
ambari.service.check.user	ambari-qa	✖
policy.download.auth.users	hive	✖

+

6.3 审计

Ranger UI 提供了审计页面，其中以下几个模块，并且每个模块页面都可以根据记录项来搜索相应的日志：

- **Access**：用户访问资源的日志记录；
- **Admin**：管理员操作策略和服务的日志记录；
- **Login Session**：用户登录 Ranger UI 的记录；
- **Plugins**：各个 ranger 插件的状态。

6.3.1 用户访问记录

该模块记录了所有用户访问行为^[8]，日志记录项包括以下这些，也可以根据这些日志记录项来搜索对应的日志。

- **Access Enforcer**：访问请求的决策者，判断访问是允许还是拒绝；

- Access Type: 用户访问操作类型，不同组件访问类型不同；
- Client IP: 试图访问资源的用户客户端 IP 地址；
- Start/End Data: 用户访问的时间范围；
- Resource Name/Type: 用户尝试访问的资源名称或类型；
- Result: 用户访问返回的结果；
- Service Name/Type: 用户尝试访问的服务名称或类型；
- User: 试图访问资源的用户名称

Access											
Admin Login Sessions Plugins											
<input type="text" value="q"/>											
<div> Access Enforcer Access Type Client IP End Date Resource Name Resource Type Result Service Name Service Type Start Date </div>											
Last Updated Time: 12/21/2017 03:17:20 PM											
Policy	Time	User	Service Name / Type	Resource Name / Type	Access Type	Result	Access Enforcer	Client IP	Event Count	Tags	
...	12/21/2017 11:16:43 PM	hbase	hadoop_hadoop_hdfs	/apps/hbase/data/archive path	READ_EXECUTE	Allowed	hadoop-acl	10.0.0.43	1		
...	12/21/2017 11:16:43 PM	hbase	hadoop_hadoop_hdfs	/apps/hbase/data/oldWALS path	READ_EXECUTE	Allowed	hadoop-acl	10.0.0.43	1		
28	12/21/2017 11:16:32 PM	Bob	hadoop_hive_hive		USE	Allowed	ranger-acl	10.0.0.42	1		
...	12/21/2017 11:16:22 PM	hive	hadoop_hadoop_hdfs	/data/tmp/hive/hive/4dad87... path	WRITE	Allowed	hadoop-acl	10.0.0.42	1		
...	12/21/2017 11:16:22 PM	hive	hadoop_hadoop_hdfs	/data/tmp/hive/hive/4dad87... path	WRITE	Allowed	hadoop-acl	10.0.0.42	1		

6.3.2 管理员操作记录

该模块记录了 Ranger web UI 上发生的所有事件，即管理员的所有操作，包括以下：

- Audit Type: 管理员执行操作的对象，包括服务、策略和用户；
- User: 执行操作的用户名；
- Date: 登录时间和日期是为每个会话存储的。日期范围用于过滤该特定日期范围的结果；
- Action: 对资源执行的操作，例如（创建，更新，删除，更改密码等操作）；
- Session Id: 会话 id，每次尝试登录到系统时，会话计数都会增加。

Access Admin Login Sessions Plugins						
<input type="text" value="q"/>						
<div> Actions Audit Type End Date Session Id Start Date User </div>						
Last Updated Time: 12/25/2017 04:58:23 PM						
Policy	Session Id	Audit Type	User	Date (中国标准时间)	Actions	Session Id
Policy updated db_bigdata_China	70925	Ranger Policy	admin	12/20/2017 04:04:08 PM	update	70925
Policy updated db_bigdata_China	70925	Ranger Policy	admin	12/20/2017 04:02:46 PM	update	70925
Policy updated db_bigdata_China	70925	Ranger Policy	admin	12/20/2017 04:01:46 PM	update	70925
Policy updated test_tenantA_1	70925	Ranger Policy	admin	12/20/2017 04:00:22 PM	update	70925
Policy updated db_bigdata_China	70623	Ranger Policy	admin	12/20/2017 11:05:51 AM	update	70623
Policy updated tenant_A	70623	Ranger Policy	admin	12/20/2017 11:05:11 AM	update	70623
Policy updated db_bigdata_China	70623	Ranger Policy	admin	12/20/2017 11:02:55 AM	update	70623

点开操作，可以看到操作的具体细节：

Operation : update

Policy ID : 5

Policy Name : db_bigdata_China

Repository Type : tag

Updated Date : 12/20/2017 11:05:51 AM 中国标准时间

Updated By : Admin

Allow PolicyItems :

Old Value	New Value
<empty>	Groups: <empty>
	Users: qy
	Permissions: hdfs:read , hdfs:write , hdfs:execute , hive:select , hive:update , hive:create , hive:drop ,
	Delegate Admin: disabled

Deny PolicyItems :

Old Value
Groups: <empty>

OK

6.3.3 登录会话

该模块记录了登录 Web UI 的会话相关信息，包括以下：

- Login Id: 登录用户名称；
- Result: 登录返回的结果；
- Login Type: 用户尝试登录的模式；
- IP: 用户登录系统的 IP；
- User Agent: 用户登录系统的编码环境

Access

Admin

Login Sessions

Plugins

Search for your login sessions...

Last Updated Time : 12/25/2017 05:17:54 PM

Session Id	Login Id	Result	Login Type	IP	User Agent	Login Time (中国标准时间)
78885	rangertagsync	Success	Username/Password	10.0.0.43	Java/1.7.0_71	12/25/2017 05:16:34 PM
78884	rangertagsync	Success	Username/Password	10.0.0.43	Java/1.7.0_71	12/25/2017 05:15:33 PM
78883	rangertagsync	Success	Username/Password	10.0.0.43	Java/1.7.0_71	12/25/2017 05:14:32 PM
78882	rangertagsync	Success	Username/Password	10.0.0.43	Java/1.7.0_71	12/25/2017 05:13:31 PM
78881	rangertagsync	Success	Username/Password	10.0.0.43	Java/1.7.0_71	12/25/2017 05:12:30 PM
78880	rangertagsync	Success	Username/Password	10.0.0.43	Java/1.7.0_71	12/25/2017 05:11:29 PM
78879	rangertagsync	Success	Username/Password	10.0.0.43	Java/1.7.0_71	12/25/2017 05:10:28 PM
78878	rangertagsync	Success	Username/Password	10.0.0.43	Java/1.7.0_71	12/25/2017 05:09:27 PM
78877	rangertagsync	Success	Username/Password	10.0.0.43	Java/1.7.0_71	12/25/2017 05:08:26 PM

并且可以根据会话 id，查看会话的详细信息以及具体操作。

Session Detail:	
Login Id	admin
Result	Success
Login Type	Username/Password
IP	10.0.0.42
User Agent	Python-urllib/2.6
Login Time	12/25/2017 05:21:15 PM 中国标准时间

Show Actions

6.3.4 插件状态

该模块显示了 ranger 中各插件的相关信息。

- Service Name: ranger 中策略存储的服务名称;
- Plugin Id: 服务的代理名称;
- Plugin IP: 插件所安装的 IP 地址;
- Http Response Code: ranger 与服务之间通信的 http 状态码。

Access	Admin	Login Sessions	Plugins
--------	-------	----------------	---------

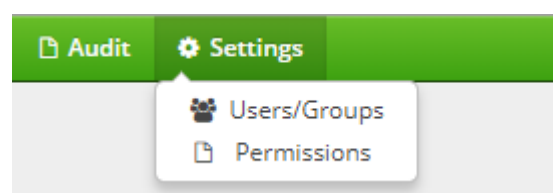
Search for your plugins...

Last Updated Time : 12/25/2017 05:32:03 PM

Export Date (中国标准时间)	Service Name	Plugin Id	Plugin IP	Http Response Code	Status
12/21/2017 09:59:19 AM	hadoop_hive	hiveServer2@sqy-m-42-hadoo...	10.0.0.42	200	Policies synced to plugin
12/20/2017 04:04:12 PM	hadoop_hive	hiveServer2@sqy-m-42-hadoo...	10.0.0.42	200	Policies synced to plugin
12/20/2017 04:04:11 PM	hadoop_hadoop	hdfs@sqy-m-42-hadoop_hado...	10.0.0.42	200	Policies synced to plugin
12/20/2017 04:03:11 PM	hadoop_hive	hiveServer2@sqy-m-42-hadoo...	10.0.0.42	200	Policies synced to plugin
12/20/2017 04:03:10 PM	hadoop_hadoop	hdfs@sqy-m-42-hadoop_hado...	10.0.0.42	200	Policies synced to plugin

6.4 设置

该模块下包括用户/用户组管理和 Web 模块权限管理。



6.4.1 用户/组管理

这里用户和用户组包括 ranger 内部和从外部（UNIX 或 LDAP 等）同步的用户/组。其中 ranger 内部用户/组可以在下面页面创建，但仅作用于 ranger Web 页面。用户具有两种角色，只有 admin 角色的用户可以对服务进行增删改操作。

Users/Groups

UsersGroups

User List

Search for your users...

Add New UserSet Visibility

<input type="checkbox"/>	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"/>	hive		User	External	hadoop	Visible
<input type="checkbox"/>	ambari-qa		User	External	users hadoop	Visible
<input type="checkbox"/>	hdfs		User	External	hdfs hadoop	Visible
<input type="checkbox"/>	ranger		User	External	ranger hadoop	Visible

选择内部用户，如 admin，可以修改用户名和所属组信息：

Users/Groups > User Edit

User Detail

Basic InfoChange Password

User Name *admin

First Name *Admin

Last Name

Email Address

Select Role *Admin

Group Please select

Enter Group Name

publichadoopusers

SaveCancel

选择外部用户，如 hive，只能修改用户所属角色：

Users/Groups > User Edit

User Detail

User Name *hive

First Namehive

Last Namehive

Email Address

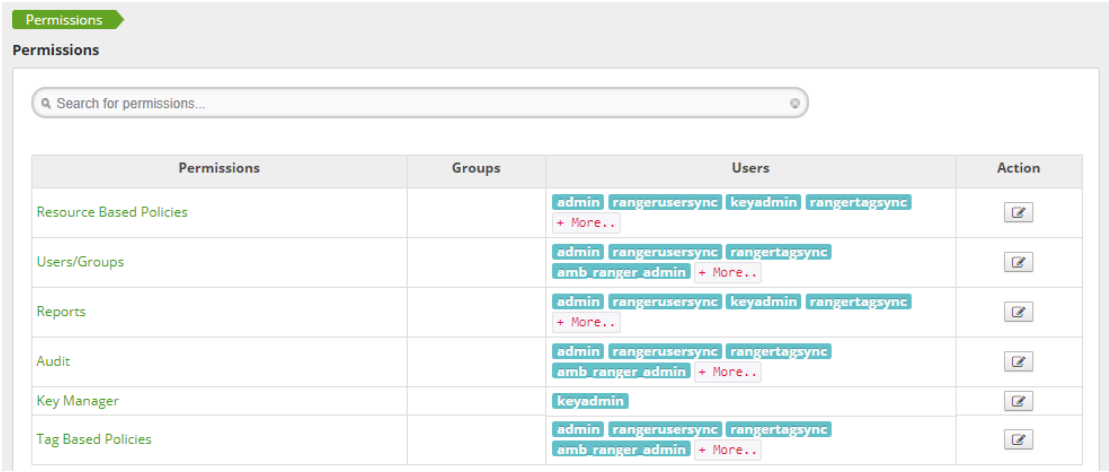
Select Role *User

Grouphadoop

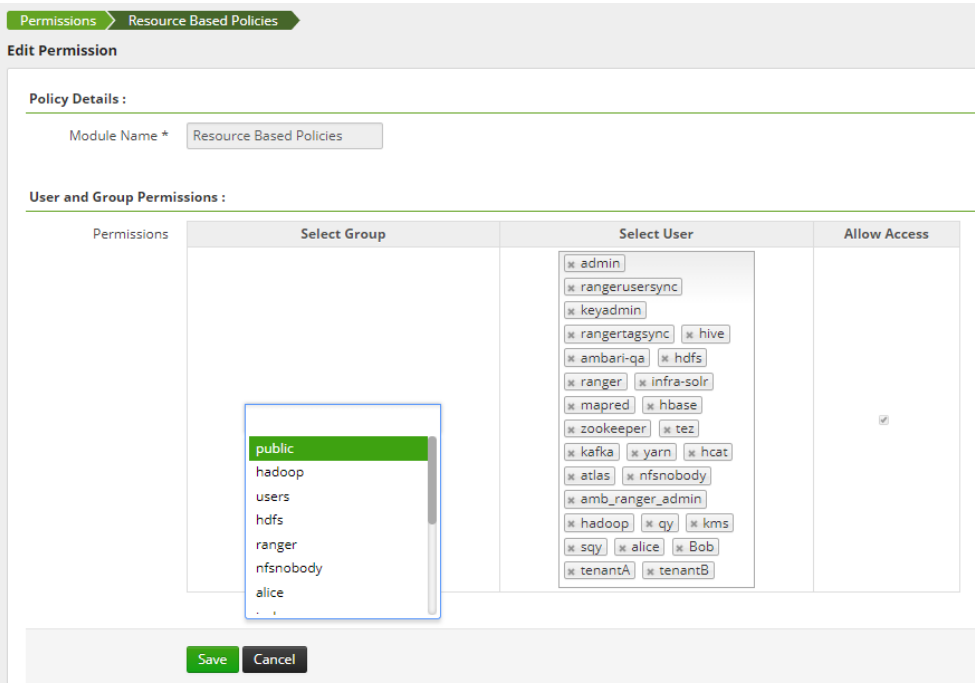
SaveCancel

6.4.2 权限管理

这里的权限是指对 Ranger Web 门户中的模块访问权限，可以选择用户/组对某模块的访问权限。

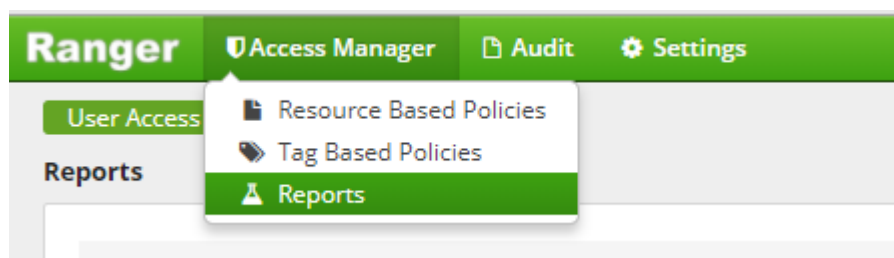


点击想要编辑的权限，出现编辑权限的页面。



6.5 Reports 模块管理

随着策略数量的增加，可以使用“Reports”页面帮助更有效地管理策略。该页面列出了所有 HDFS，HBase，Hive，YARN，Knox，Storm，Solr，Kafka，Atlas 和 tag。



Reports 管理页面提供了对所有策略进行搜索、查看、导出和编辑策的功能。首先可以通过以下搜索项来搜索策略：

- Policy Name: 策略名称；
- Policy Type: 策略类型（访问、列级屏蔽、行级过滤）；
- Component: 策略所属的组件名称（HDFS, HBase, Hive, YARN, Knox, Storm, Solr, Kafka, Atlas 和 tag）；
- Resource: 授权的资源名称。
- Group、user: 被授权用户或用户组

User Access Report

Reports

Search Criteria

hide

Policy Name

Policy Type

Select policy type

Component

Select Component

Resource

Search By

Group

Select Group

Search

Download

HDFS

hide

Policy ID	Policy Name	Resources	Policy Type	Status	Allow Conditions
1	all - path	path:/*	Access	Enabled	+

点击 Download 即可导出所有的策略，导出文件的格式当前版本支持：Excel 和 CSV。点击 Policy ID 即可进入策略编辑页面，编辑策略。

7 参考文献

[1]HDFS Permission Guide.

http://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-hdfs/HdfsPermissionsGuide.html#ACLs_.28Access_Control_Lists.29

[2] Anthony B, Boudnik K, Adams C, et al. Hadoop Security[M]// Professional Hadoop®;.

[3] Apache Ranger.https://zh.hortonworks.com/apache/ranger/#section_2

[4] Apache Ranger 0.5.0 Installation.

<https://cwiki.apache.org/confluence/display/RANGER/Install+and+Configure+Solr+for+Ranger+Audits+-+Apache+Ranger+0.5.0+Installation>

[5] Install and Configure Solr for Ranger Audits.

<https://cwiki.apache.org/confluence/display/RANGER/Install+and+Configure+Solr+for+Ranger+Audits+-+Apache+Ranger+0.5>

[6] Configuring PostgreSQL for Ranger.

https://docs.hortonworks.com/HDPDocuments/HDP2/HDP-2.5.0/bk_security/content/configuring_postgresql_for_ranger.html

[7] Hive Atlas Bridge. <http://atlas.apache.org/Bridge-Hive.html>

[8] Apache Ranger User Guide.

<https://cwiki.apache.org/confluence/display/RANGER/Install+and+Configure+Solr+for+Ranger+Audits+-+Apache+Ranger+0.5.0+Installation>