

尚硅谷大数据项目之尚品汇（安全环境实战）

(作者：尚硅谷研究院)

版本：V4.0

第 1 章 概述

Hadoop 启用 Kerberos 安全认证之后，之前的非安全环境下的全流程调度脚本和即席查询引擎均会遇到认证问题，故需要对其进行改进。

第 2 章 数仓全流程

2.1 改动说明

此处统一将数仓的全部数据资源的所有者设为 hive 用户，全流程的每步操作均认证为 hive 用户。

2.2 改动实操

2.2.1 用户准备

1.在各节点创建 hive 用户，如已存在则跳过

```
[root@hadoop102 ~]# useradd hive -g hadoop
[root@hadoop102 ~]# echo hive | passwd --stdin hive

[root@hadoop103 ~]# useradd hive -g hadoop
[root@hadoop103 ~]# echo hive | passwd --stdin hive

[root@hadoop104 ~]# useradd hive -g hadoop
[root@hadoop104 ~]# echo hive | passwd --stdin hive
```

2.为 hive 用户创建 Kerberos 主体

1) 创建主体

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

2) 生成 keytab 文件

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

3) 修改 keytab 文件所有者和访问权限

```
[root@hadoop102 ~]# chown hive:hadoop
/etc/security/keytab/hive.keytab
[root@hadoop102 ~]# chmod 440 /etc/security/keytab/hive.keytab
```

4) 分发 keytab 文件

更多 [Java](#) - [大数据](#) - [前端](#) - [python](#) 人工智能资料下载，可百度访问：[尚硅谷官网](#)

```
[root@hadoop102 ~]# xsync /etc/security/keytab/hive.keytab
```

2.2.2 数据采集通道修改

1. 用户行为日志

修改/opt/module/flume/conf/kafka-flume-hdfs.conf 配置文件，增加以下参数

```
[root@hadoop104 ~]# vim /opt/module/flume/conf/kafka-flume-hdfs.conf
a1.sinks.k1.hdfs.kerberosPrincipal=hive@EXAMPLE.COM
a1.sinks.k1.hdfs.kerberosKeytab=/etc/security/keytab/hive.keytab
```

2. 业务数据

修改 sqoop 每日同步脚本/home/atguigu/bin/mysql_to_hdfs.sh，

```
[root@hadoop102 ~]# vim /home/atguigu/bin/mysql_to_hdfs.sh
```

在顶部增加如下认证语句

```
kinit -kt /etc/security/keytab/hive.keytab hive
```

2.2.3 数仓各层脚本修改

数仓各层脚本均需在顶部加入如下认证语句

```
kinit -kt /etc/security/keytab/hive.keytab hive
```

修改语句如下

```
[root@hadoop102 ~]# sed -i '1 a kinit -kt /etc/security/keytab/hive.keytab hive'
hdfs_to_ods_log.sh
[root@hadoop102 ~]# sed -i '1 a kinit -kt /etc/security/keytab/hive.keytab hive'
hdfs_to_ods_db.sh
[root@hadoop102 ~]# sed -i '1 a kinit -kt /etc/security/keytab/hive.keytab hive'
ods_to_dwd_log.sh
[root@hadoop102 ~]# sed -i '1 a kinit -kt /etc/security/keytab/hive.keytab hive'
ods_to_dim_db.sh
[root@hadoop102 ~]# sed -i '1 a kinit -kt /etc/security/keytab/hive.keytab hive'
ods_to_dwd_db.sh
[root@hadoop102 ~]# sed -i '1 a kinit -kt /etc/security/keytab/hive.keytab hive'
dwd_to_dws.sh
[root@hadoop102 ~]# sed -i '1 a kinit -kt /etc/security/keytab/hive.keytab hive'
dws_to_dwt.sh
[root@hadoop102 ~]# sed -i '1 a kinit -kt /etc/security/keytab/hive.keytab hive'
dwt_to_ads.sh
[root@hadoop102 ~]# sed -i '1 a kinit -kt /etc/security/keytab/hive.keytab hive'
hdfs_to_mysql.sh
```

注：

```
sed -i '1 a text' file
```

表示将 text 内容加入到 file 文件的第 1 行之后

2.2.4 修改 HDFS 特定路径所有者

1. 认证为 hdfs 用户，执行以下命令并按提示输入密码

```
[root@hadoop102 ~]# kinit hdfs/hadoop
```

更多 [Java](#) - [大数据](#) - [前端](#) - [python](#) 人工智能资料下载，可百度访问：[尚硅谷官网](#)

2.修改数据采集目标路径

```
[root@hadoop102 ~]# hadoop fs -chown -R hive:hadoop /origin_data
```

3.修改数仓表所在路径

```
[root@hadoop102 ~]# hadoop fs -chown -R hive:hadoop /warehouse
```

4.修改 hive 家目录/user/hive

```
[root@hadoop102 ~]# hadoop fs -chown -R hive:hadoop /user/hive
```

5.修改 spark.eventLog.dir 路径

```
[root@hadoop102 ~]# hadoop fs -chown -R hive:hadoop /spark-history
```

2.2.5 全流程数据准备

1.用户行为日志

1) 启动日志采集通道, 包括 Zookeeper, Kafka, Flume 等

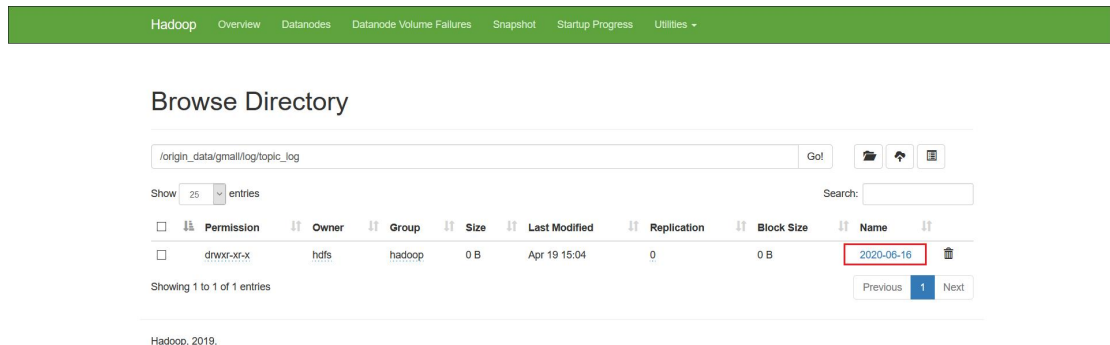
2) 修改 hadoop102, hadoop103 两台节点的/opt/module/applog/application.yml 文件, 将模拟日期改为 2020-06-16 如下

```
#业务日期
mock.date: "2020-06-16"
```

3) 执行生成日志的脚本

```
[root@hadoop102 ~]# lg.sh
```

4) 等待片刻, 观察 HDFS 是否出现 2020-06-16 的日志文件



Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
drwxr-xr-x	hdfs	hadoop	0 B	Apr 19 15:04	0	0 B	2020-06-16

2.业务数据

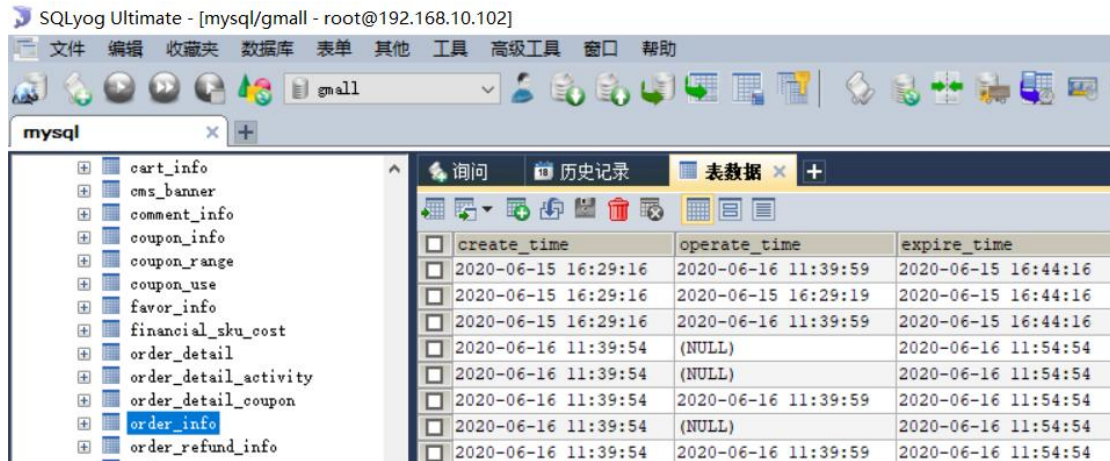
1) 修改/opt/module/db_log/application.properties, 将模拟日期修改为 2020-06-16, 如下

```
#业务日期
mock.date=2020-06-16
```

2) 进入到/opt/module/db_log 路径, 执行模拟生成业务数据的命令, 如下

```
[root@hadoop102 ~]# java -jar gmall2020-mock-db-2021-01-22.jar
```

3) 观察 mysql 的 gmall 数据中是否出现 2020-06-16 的数据



2.2.6 启动 Azkaban

1.在各节点创建 azkaban 用户

```
[root@hadoop102 ~]# useradd azkaban -g hadoop
[root@hadoop102 ~]# echo azkaban | passwd --stdin azkaban

[root@hadoop103 ~]# useradd azkaban -g hadoop
[root@hadoop103 ~]# echo azkaban | passwd --stdin azkaban

[root@hadoop104 ~]# useradd azkaban -g hadoop
[root@hadoop104 ~]# echo azkaban | passwd --stdin azkaban
```

2.将各节点 Azkaban 安装路径所有者改为 azkaban 用户

```
[root@hadoop102 ~]# chown -R azkaban:hadoop /opt/module/azkaban
[root@hadoop103 ~]# chown -R azkaban:hadoop /opt/module/azkaban
[root@hadoop104 ~]# chown -R azkaban:hadoop /opt/module/azkaban
```

3.使用 azkaban 用户启动 Azkaban

1) 启动 Executor Server

在各节点执行以下命令，启动 Executor

```
[root@hadoop102 ~]# sudo -i -u azkaban bash -c "cd /opt/module/azkaban/azkaban-exec;bin/start-exec.sh"
[root@hadoop103 ~]# sudo -i -u azkaban bash -c "cd /opt/module/azkaban/azkaban-exec;bin/start-exec.sh"
[root@hadoop104 ~]# sudo -i -u azkaban bash -c "cd /opt/module/azkaban/azkaban-exec;bin/start-exec.sh"
```

2) 激活 Executor Server，任选一台节点执行以下激活命令即可

```
[root@hadoop102 ~]# curl http://hadoop102:12321/executor?action=activate
[root@hadoop102 ~]# curl http://hadoop103:12321/executor?action=activate
[root@hadoop102 ~]# curl http://hadoop104:12321/executor?action=activate
```

3) 启动 Web Server

```
[root@hadoop102 ~]# sudo -i -u azkaban bash -c "cd /opt/module/azkaban/azkaban-web;bin/start-web.sh"
```

4.修改数仓各层脚本访问权限，确保 azkaban 用户能够访问到

```
[root@hadoop102 ~]# chown -R atguigu:hadoop /home/atguigu
[root@hadoop102 ~]# chmod 770 /home/atguigu

[root@hadoop103 ~]# chown -R atguigu:hadoop /home/atguigu
[root@hadoop103 ~]# chmod 770 /home/atguigu

[root@hadoop104 ~]# chown -R atguigu:hadoop /home/atguigu
[root@hadoop104 ~]# chmod 770 /home/atguigu
```

2.2.7 全流程调度

1) 工作流参数

Execute Flow gmail

[Flow View](#)
[Notification](#)
[Failure Options](#)
[Concurrent](#)
[Flow Parameters](#)

Add temporary flow parameters that are used to override global settings for each job.

Flow Property Override

Name	Value
dt	2020-06-16
useExecutor	4

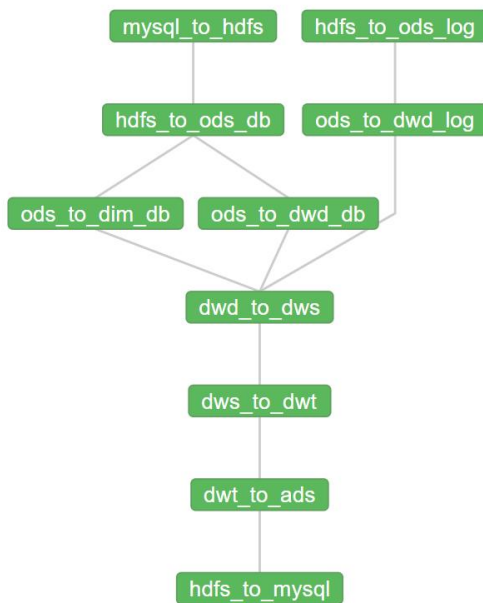
Add Row

Schedule

Cancel

Execute

2) 运行结果



第3章 即席查询之 Presto

3.1 改动说明

Presto 集群开启 Kerberos 认证可只配置 Presto Coordinator 和 Presto Cli 之间进行认证，集群内部通讯可不进行认证。Presto Coordinator 和 Presto Cli 之间的认证要求两者采用更为安全的 HTTPS 协议进行通讯。

若 Presto 对接的是 Hive 数据源，由于其需要访问 Hive 的元数据和 HDFS 上的数据文件，故也需要对 Hive Connector 进行 Kerberos 认证。

3.2 改动实操

3.2.1 用户准备

1. 在所有节点创建 presto 系统用户

```
[root@hadoop102 ~]# useradd presto -g hadoop
[root@hadoop102 ~]# echo presto | passwd --stdin presto

[root@hadoop103 ~]# useradd presto -g hadoop
[root@hadoop103 ~]# echo presto | passwd --stdin presto

[root@hadoop104 ~]# useradd presto -g hadoop
[root@hadoop104 ~]# echo presto | passwd --stdin presto
```

2. 为 Hive Connector 创建 Kerberos 主体

1) 创建 presto 用户的 Kerberos 主体

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

2) 生成 keytab 文件

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

3) 修改 keytab 文件的访问权限

```
[root@hadoop102 ~]# chown presto:hadoop /etc/security/keytab/presto.keytab
```

4) 分发 keytab 文件

```
[root@hadoop102 ~]# xsync /etc/security/keytab/presto.keytab
```

3. 为 Presto Coordinator 创建 Kerberos 主体

1) 创建 presto 用户的 Kerberos 主体

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

2) 生成 keytab 文件

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

3) 修改 keytab 文件的访问权限

```
[root@hadoop102 ~]# chown presto:hadoop /etc/security/keytab/presto.service.keytab
```

3.2.2 创建 HTTPS 协议所需的密钥对

注意:

- (1) alias (别名) 需要和 Presto Coordinator 的 Kerberos 主体名保持一致
- (2) 名字与姓氏 需要填写 Coordinator 所在的主机名

1) 使用 Java 提供的 keytool 工具生成密钥对

```
[root@hadoop102 ~]# keytool -genkeypair -alias presto -keyalg RSA -keystore /etc/security/keytab/keystore.jks
```

输入密钥库口令:

再次输入新口令:

您的名字与姓氏是什么?

[Unknown]: **hadoop102**

您的组织单位名称是什么?

[Unknown]:

您的组织名称是什么?

[Unknown]:

您所在的城市或区域名称是什么?

[Unknown]:

您所在的省/市/自治区名称是什么?

[Unknown]:

该单位的双字母国家/地区代码是什么?

[Unknown]:

CN=hadoop102, OU=Unknown, O=Unknown, L=Unknown, ST=Unknown, C=Unknown 是否正确?

[否]: y

输入 <presto> 的密钥口令

(如果和密钥库口令相同, 按回车):

2) 修改 keystore 文件的所有者和访问权限

```
[root@hadoop102 ~]# chown presto:hadoop /etc/security/keytab/keystore.jks
```

```
[root@hadoop102 ~]# chmod 660 /etc/security/keytab/keystore.jks
```

3.2.3 修改 Presto Coordinator 配置文件

在/opt/module/presto/etc/config.properties 文件中**增加**以下参数

```
[root@hadoop102 ~]# vim /opt/module/presto/etc/config.properties
http-server.authentication.type=KERBEROS

http.server.authentication.krb5.service-name=presto
http.server.authentication.krb5.keytab=/etc/security/keytab/presto.service.keytab
http.authentication.krb5.config=/etc/krb5.conf

http-server.https.enabled=true
http-server.https.port=7778
```

更多 [Java](#) - [大数据](#) - [前端](#) - [python](#) 人工智能资料下载, 可百度访问: [尚硅谷官网](#)

```
http-server.https.keystore.path=/etc/security/keytab/keystore.jks
http-server.https.keystore.key=123456
```

3.2.4 修改 Hive Connector 配置文件

1. 在/opt/module/presto/etc/catalog/hive.properties 中增加以下参数

```
[root@hadoop102 ~]# vim /opt/module/presto/etc/catalog/hive.properties

hive.metastore.authentication.type=KERBEROS
hive.metastore.service.principal=hive/hadoop102@EXAMPLE.COM
hive.metastore.client.principal=presto@EXAMPLE.COM
hive.metastore.client.keytab=/etc/security/keytab/presto.keytab

hive.hdfs.authentication.type=KERBEROS
hive.hdfs.impersonation.enabled=true
hive.hdfs.presto.principal=presto@EXAMPLE.COM
hive.hdfs.presto.keytab=/etc/security/keytab/presto.keytab
hive.config.resources=/opt/module/hadoop-3.1.3/etc/hadoop/core-site.xml,/opt/module/hadoop-3.1.3/etc/hadoop/hdfs-site.xml
```

2. 分发/opt/module/presto/etc/catalog/hive.properties 文件

```
[root@hadoop102 ~]# xsync /opt/module/presto/etc/catalog/hive.properties
```

3.2.5 配置客户端 Kerberos 主体到用户名之间的映射规则

1. 新建/opt/module/presto/etc/access-control.properties 配置文件，内容如下

```
[root@hadoop102 ~]# vim /opt/module/presto/etc/access-control.properties

access-control.name=file
security.config-file=etc/rules.json
```

2. 新建/opt/module/presto/etc/rules.json 文件，内容如下

```
[root@hadoop102 ~]# vim /opt/module/presto/etc/rules.json

{
  "catalogs": [
    {
      "allow": true
    }
  ],
  "user_patterns": [
    "(.*)",
    "([a-zA-Z]+) /?.*@.*"
  ]
}
```

3.2.6 配置 Presto 代理用户

1. 修改 Hadoop 配置文件

修改\$HADOOP_HOME/etc/hadoop/core-site.xml 配置文件，增加如下内容

```
[root@hadoop102 ~]# vim $HADOOP_HOME/etc/hadoop/core-site.xml
```



```
<property>
  <name>hadoop.proxyuser.presto.hosts</name>
  <value>*</value>
</property>

<property>
  <name>hadoop.proxyuser.presto.groups</name>
  <value>*</value>
</property>

<property>
  <name>hadoop.proxyuser.presto.users</name>
  <value>*</value>
</property>
```

2. 分发修改的文件

```
[root@hadoop102 ~]# xsync $HADOOP_HOME/etc/hadoop/core-site.xml
```

3. 重启 Hadoop 集群

```
[root@hadoop102 ~]# stop-dfs.sh
[root@hadoop103 ~]# stop-yarn.sh

[root@hadoop102 ~]# start-dfs.sh
[root@hadoop103 ~]# start-yarn.sh
```

3.2.7 重启 Presto 集群

1. 关闭集群

```
[root@hadoop102 ~]# /opt/module/presto/bin/launcher stop
[root@hadoop103 ~]# /opt/module/presto/bin/launcher stop
[root@hadoop104 ~]# /opt/module/presto/bin/launcher stop
```

2. 修改 Presto 安装路径所有者为 presto

```
[root@hadoop102 ~]# chown -R presto:hadoop /opt/module/presto
[root@hadoop103 ~]# chown -R presto:hadoop /opt/module/presto
[root@hadoop104 ~]# chown -R presto:hadoop /opt/module/presto
```

3. 使用 hive 用户启动 MetaStore 服务

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

4. 使用 presto 用户启动 Presto 集群

```
[root@hadoop102 ~]# sudo -i -u presto /opt/module/presto/bin/launcher
start
[root@hadoop103 ~]# sudo -i -u presto /opt/module/presto/bin/launcher
start
[root@hadoop104 ~]# sudo -i -u presto /opt/module/presto/bin/launcher
start
```

3.2.8 客户端认证访问 Presto 集群

```
[root@hadoop102 presto]# ./prestocli \
--server https://hadoop102:7778 \
--catalog hive \
--schema default \
--enable-authentication \
--krb5-remote-service-name presto \
--krb5-config-path /etc/krb5.conf \
--krb5-principal atguigu@EXAMPLE.COM \
```

更多 [Java](#) - [大数据](#) - [前端](#) - [python](#) 人工智能资料下载，可百度访问：尚硅谷官网

```
--krb5-keytab-path /home/atguigu/atguigu.keytab \  
--keystore-path /etc/security/keytab/keystore.jks \  
--keystore-password 123456 \  
--user atguigu
```

第 4 章 即席查询之 Kylin

4.1 改动说明

从 Kylin 的架构, 可以看出 Kylin 充当只是一个 Hadoop 客户端, 读取 Hive 数据, 利用 MR 或 Spark 进行计算, 将 Cube 存储至 HBase 中。所以在安全的 Hadoop 环境下, Kylin 不需要做额外的配置, 只需要具备一个 Kerberos 主体, 进行常规的认证即可。

但是 Kylin 所依赖的 HBase 需要进行额外的配置, 才能在安全的 Hadoop 环境下正常工作。

4.2 改动实操

4.2.1 HBase 开启 Kerberos 认证

1. 用户准备

1) 在各节点创建 hbase 系统用户

```
[root@hadoop102 ~]# useradd -g hadoop hbase  
[root@hadoop102 ~]# echo hbase | passwd --stdin hbase  
  
[root@hadoop103 ~]# useradd -g hadoop hbase  
[root@hadoop103 ~]# echo hbase | passwd --stdin hbase  
  
[root@hadoop104 ~]# useradd -g hadoop hbase  
[root@hadoop104 ~]# echo hbase | passwd --stdin hbase
```

2) 创建 hbase Kerberos 主体

(1) 在 hadoop102 节点创建主体, 生成密钥文件, 并修改所有者

```
[root@hadoop102 ~]# kadmin -padmin/admin -wadmin -q"addprinc -randkey  
hbase/hadoop102"  
[root@hadoop102 ~]# kadmin -padmin/admin -wadmin -q"xst -k  
/etc/security/keytab/hbase.service.keytab hbase/hadoop102"  
[root@hadoop102 ~]# chown hbase:hadoop  
/etc/security/keytab/hbase.service.keytab
```

(2) 在 hadoop103 节点创建主体, 生成密钥文件, 并修改所有者

```
[root@hadoop103 ~]# kadmin -padmin/admin -wadmin -q"addprinc -randkey  
hbase/hadoop103"  
[root@hadoop103 ~]# kadmin -padmin/admin -wadmin -q"xst -k  
/etc/security/keytab/hbase.service.keytab hbase/hadoop103"  
[root@hadoop103 ~]# chown hbase:hadoop  
/etc/security/keytab/hbase.service.keytab
```

(3) 在 hadoop104 节点创建主体, 生成密钥文件, 并修改所有者

```
[root@hadoop104 ~]# kadmin -padmin/admin -wadmin -q"addprinc -randkey
```

更多 [Java](#) - [大数据](#) - [前端](#) - [python](#) 人工智能资料下载, 可百度访问: [尚硅谷官网](#)

```
hbase/hadoop104"
[root@hadoop104 ~]# kadmin -padmin/admin -wadmin -q"xst -k
/etc/security/keytab/hbase.service.keytab hbase/hadoop104"
[root@hadoop104 ~]# chown hbase:hadoop
/etc/security/keytab/hbase.service.keytab
```

2. 修改 HBase 配置文件

修改\$HBASE_HOME/conf/hbase-site.xml 配置文件，增加以下参数

```
[root@hadoop102 ~]# vim $HBASE_HOME/conf/hbase-site.xml

<property>
  <name>hbase.security.authentication</name>
  <value>kerberos</value>
</property>

<property>
  <name>hbase.master.kerberos.principal</name>
  <value>hbase/_HOST@EXAMPLE.COM</value>
</property>

<property>
<name>hbase.master.keytab.file</name>
<value>/etc/security/keytab/hbase.service.keytab</value>
</property>

<property>
  <name>hbase.regionserver.kerberos.principal</name>
  <value>hbase/_HOST@EXAMPLE.COM</value>
</property>

<property>
  <name>hbase.regionserver.keytab.file</name>
  <value>/etc/security/keytab/hbase.service.keytab</value>
</property>

<property>
  <name>hbase.coprocessor.region.classes</name>

<value>org.apache.hadoop.hbase.security.token.TokenProvider</valu
e>
</property>
```

3. 分发配置文件

```
[root@hadoop102 ~]# xsync $HBASE_HOME/conf/hbase-site.xml
```

4. 修改 hbase.rootdir 路径所有者

1) 使用 hdfs/hadoop 用户进行认证

```
[root@hadoop102 ~]# kinit hdfs/hadoop
```

2) 修改所有者

```
[root@hadoop102 ~]# hadoop fs -chown -R hbase:hadoop /hbase
```

5. 启动 HBase

1) 修改各节点 HBase 安装目录所有者

```
[root@hadoop102 ~]# chown -R hbase:hadoop /opt/module/hbase
```

更多 [Java](#) - [大数据](#) - [前端](#) - [python](#) 人工智能资料下载，可百度访问：尚硅谷官网

```
[root@hadoop103 ~]# chown -R hbase:hadoop /opt/module/hbase
[root@hadoop104 ~]# chown -R hbase:hadoop /opt/module/hbase
```

2) 配置 hbase 用户从主节点 (hadoop102) 到所有节点的 ssh 免密

3) 使用 hbase 用户启动 HBase

```
[root@hadoop102 ~]# sudo -i -u hbase start-hbase.sh
```

6. 停止 HBase

启用 Kerberos 认证之后, 关闭 HBase 时, 需先进行 Kerberos 用户认证, 认证的主体为 hbase。

1) 认证为 hbase 主体

```
[root@hadoop102 ~]# sudo -i -u hbase kinit -kt
/etc/security/keytab/hbase.service.keytab hbase/hadoop102
```

2) 停止 hbase

```
[root@hadoop102 ~]# sudo -i -u hbase stop-hbase.sh
```

4.2.2 Kylin 进行 Kerberos 认证

1. 用户准备

创建 kylin 系统用户

```
[root@hadoop102 ~]# useradd -g hadoop kylin
[root@hadoop102 ~]# echo kylin | passwd --stdin kylin
```

2. 修改 kylin.env.hdfs-working-dir 路径所有者为 kylin

1) 使用 hdfs/hadoop 用户进行认证

```
[root@hadoop102 ~]# kinit hdfs/hadoop
```

2) 修改所有者

```
[root@hadoop102 ~]# hadoop fs -chown -R hive:hadoop /kylin
```

3. 修改/opt/module/kylin 所有者为 kylin

```
[root@hadoop102 ~]# chown -R kylin:hadoop /opt/module/kylin
```

4. 启动 kylin

1) 在 kylin 用户下认证为 hive 主体

```
[root@hadoop102 ~]# sudo -i -u kylin kinit -kt
/etc/security/keytab/hive.keytab hive
```

2) 以 kylin 用户的身份启动 kylin

```
[root@hadoop102 ~]# sudo -i -u kylin /opt/module/kylin/bin/kylin.sh
start
```