**Sentry 1.6.0 移植指南（openEuler 20.03）**

|  |
| --- |
|  |

[1 sentry简介](#_Toc97112731)

[2 环境要求](#_Toc97112732)

[3 配置编译环境](#_Toc97112733)

[4 编译sentry](#_Toc97112739)

[5 测试](#_Toc97112740)

# 1 Sentry简介

Sentry是Hadoop中的一个基于角色的细粒度授权组件。Sentry用于对存储在Hadoop集群上的数据和元数据实施基于角色的细粒度授权，可以在Hadoop集群上对通过身份认证的用户和应用程序控制数据访问权限。本文主要是描述如何将开源版本的Sentry组件适配到openEuler上。 Apache Sentry已于2016年3月成功从孵化器毕业，现在是一个顶级的Apache项目。

更多请参考官网：https://sentry.apache.org/

# 2 环境要求

**硬件要求**

|  |  |
| --- | --- |
| **项目** | **说明** |
| 服务器 | TaiShan服务器 |
| CPU | 鲲鹏920处理器 或 鲲鹏916处理器 |
| 磁盘分区 | 对磁盘分区无要求 |
| 网络 | 可访问外网 |

**软件要求**

|  |  |
| --- | --- |
| **项目** | **版本** |
| OpenJDK | 1.8.0\_222 |
| Hadoop | 2.6.0 |
| Mysql | 5.7.27 |
| Maven | 3.2.5 |
| Hive | 1.1.0 |
| Snetry | 1.6.0 |
| Keberos |  |

**openEuler 系统**

|  |  |
| --- | --- |
| **项目** | **版本** |
| openEuler | 20.03 LTS SP1 |
| OS Kernel | 4.19.90 |

# 3 配置编译环境

## 3.1 安装OpenJDK

**步骤1** 创建java目录，下载openjdk。

cd /

mkdir bigdata

cd bigdata

wget https://github.com/AdoptOpenJDK/openjdk8-binaries/releases/download/jdk8u222-b10/OpenJDK8U-jdk\_aarch64\_linux\_hotspot\_8u222b10.tar.gz

**步骤2** 解压。

tar -zxvf OpenJDK8U-jdk\_aarch64\_linux\_hotspot\_8u222b10.tar.gz

**步骤3** 给解压的包授权。

chown -R root:root jdk8u222-b10/

chmod -R 755 jdk8u222-b10/

**步骤4** 配置环境变量。

vim /etc/profile

#最后添加如下内容

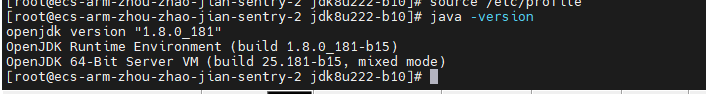
export JAVA\_HOME=/bigdata/jdk8u222-b10

export PATH=$PATH:$JAVA\_HOME/bin

source /etc/profile

检查java 是否安装成功

java –version

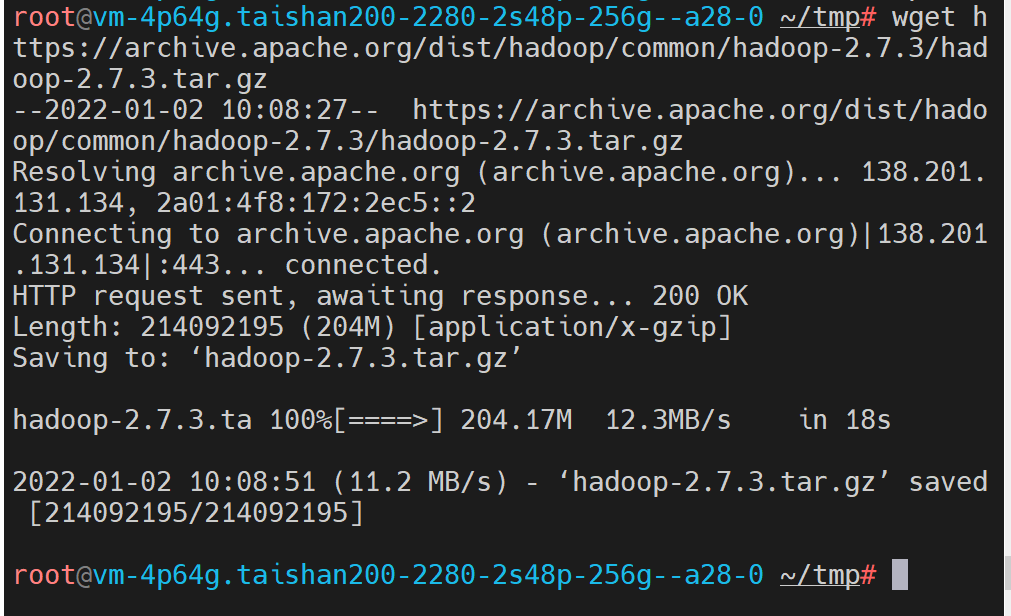


## 3.2 安装maven

**步骤1** 下载maven安装包并放到指定目录

cd /bigdata

wget https://archive.apache.org/dist/maven/maven-3/3.2.5/binaries/apache-maven-3.2.5-bin.tar.gz



**步骤2** 解压

tar -zxvf apache-maven-3.2.5-bin.tar.gz

**步骤3** 配置环境变量

vim /etc/profile

在末尾添加：

export MAVEN\_HOME=/bigdata/apache-maven-3.2.5

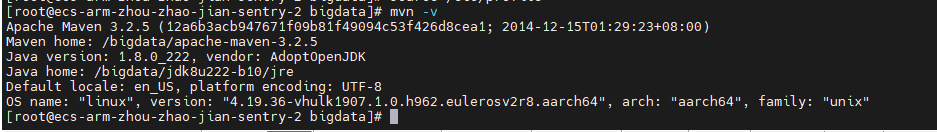
export PATH=$MAVEN\_HOME/bin:$PATH

使配置生效

source /etc/profile

**步骤4** 检查maven环境变量是否正确

mvn -v



**步骤1** 修改maven的settings.xml文件中远程仓库配置

切换目录

cd /bigdata/apache-maven-3.2.5/conf/

vim settings.xml

在 mirrors 内添加如下内容：

<mirror>

<id>huaweimaven</id>

<name>huawei maven</name>

<url>https://mirrors.huaweicloud.com/repository/maven/</url>

<mirrorOf>central</mirrorOf>

</mirror>

:wq 保存并退出

## 3.3 编译Sentry

**步骤1** 下载Sentry1.6.0源码包

cd /bigdata

wget https://github.com/apache/sentry/archive/refs/tags/release-1.6.0-rc0.tar.gz

**步骤2** 解压源码包

tar -zxvf release-1.6.0-rc0.tar.gz

**步骤3** 配置鲲鹏仓库，修改pom.xml文件，保存并退出。

cd /bigdata/sentry-release-1.6.0-rc0

vim pom.xml

在repositories标签内增加鲲鹏maven仓库，鲲鹏仓库一定要放在第一位：

<repository>

<id>Kunpengmaven</id>

<name>Kunpeng maven</name>

<url>https://mirrors.huaweicloud.com/kunpeng/maven/</url>

</repository>

<repository>

<id>huaweicloud.repo</id>

<name>HuaweiCloud Repositories</name>

<url>https://mirrors.huaweicloud.com/repository/maven</url>

</repository>

<repository>

<id>wso2.repo</id>

<url>http://maven.wso2.org/nexus/content/groups/wso2-public/</url>

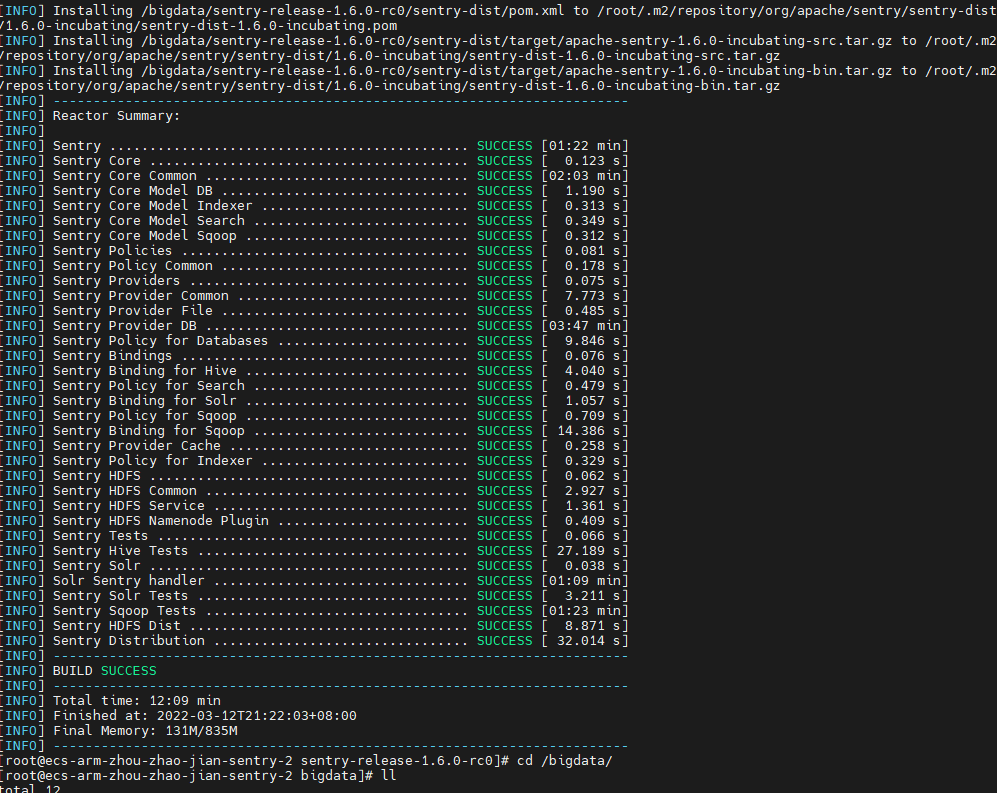
<name>wso2 Repositories</name>

</repository>

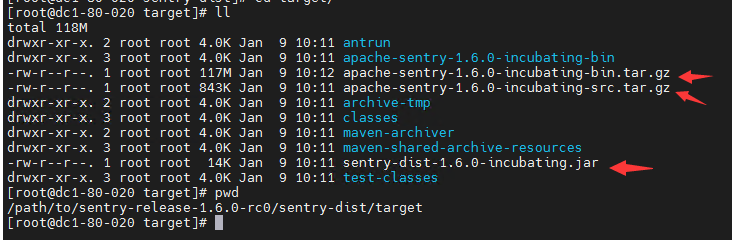
**步骤4** 编译Sentry

mvn clean install -DskipTests

编译结果为“sentry-dist/target/”下的apache-sentry-1.6.0-incubating-bin.tar.gz



编译成功截图：



## 3.4 安装Hadoop

**步骤1** 下载并解压Hadoop的安装包。

cd /bigdata

wget https://archive.apache.org/dist/hadoop/core/hadoop-2.6.0/hadoop-2.6.0.tar.gz

tar -zxvf hadoop-2.6.0.tar.gz

**步骤2** 配置环境变量。

vim /etc/profile

在末尾添加：

export HADOOP\_HOME=/bigdata/hadoop-2.6.0

export PATH=$PATH:$HADOOP\_HOME/bin:$HADOOP\_HOME/sbin

:wq

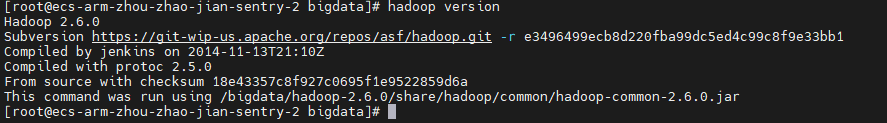
保存退出即时生效：

source /etc/profile

**步骤3** 检查Hadoop环境变量

查看版本

hadoop version



**步骤4** 关闭防火墙

systemctl disable firewalld

systemctl stop firewalld

**步骤5** 修改Hadoop的配置文件

进入配置文件主目录

cd /bigdata/hadoop-2.6.0/etc/hadoop

1. 修改hadoop-env.sh文件

vim hadoop-env.sh

找到 export JAVA\_HOME= 在其后面添加：/usr/java/jdk8u222-b10 ，得到

export JAVA\_HOME= /bigdata/jdk8u222-b10

:wq，保存并退出。

2. 修改 core-site.xml配置文件

vim core-site.xml

在<configuration>中添加如下内容

<property>

<name>hadoop.tmp.dir</name>

<value>file:/usr/local/hadoop/tmp</value>

<description>Abase for other temporary directories.</description>

</property>

<property>

<name>fs.defaultFS</name>

<value>hdfs://localhost:9000</value>

</property>

：wq ，保存并退出

3. 修改hdfs-site.xml配置文件，在<configuration>标签内添加如下内容：

<property>

<name>dfs.replication</name>

<value>1</value>

</property>

<property>

<name>dfs.namenode.name.dir</name>

<value>file:/usr/local/hadoop/tmp/dfs/name</value>

</property>

<property>

<name>dfs.datanode.data.dir</name>

<value>file:/usr/local/hadoop/tmp/dfs/data</value>

</property>

4.添加yarn-site.xml文件

vim yarn-site.xml，添加如下内容

<?xml version="1.0"?>

<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>

<!--

Licensed under the Apache License, Version 2.0 (the "License");

you may not use this file except in compliance with the License.

You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software

distributed under the License is distributed on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.See the License for the specific language governing permissions and limitations under the License. See accompanying LICENSE file.

-->

<!-- Put site-specific property overrides in this file. -->

<configuration>

<property>

<name>mapreduce.framework.name</name>

<value>yarn</value>

</property>

</configuration>

：wq ，保存并退出。

5. 修改yarn-site.xml，

vim yarn-site.xml，添加如下配置

<configuration>

<!-- Site specific YARN configuration properties -->

<property>

<name>yarn.nodemanager.aux-services</name>

<value>mapreduce\_shuffle</value>

</property>

</configuration>

:wq保存并退出

**步骤6** 格式化节点

hdfs namenode -format

**步骤7** 设置 ssh免密登陆

cd /bigdata

ssh-keygen -t rsa （直接回车执行）

cd ~/.ssh

chmod 600 authorized\_keys

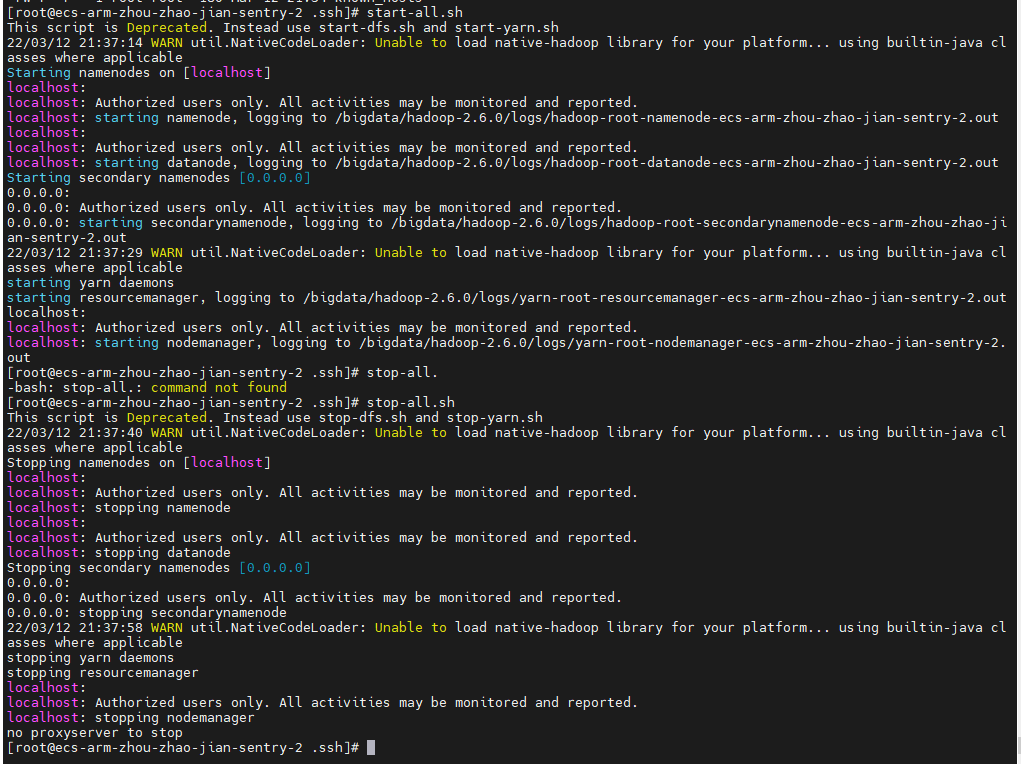
cat id\_rsa.pub >> authorized\_keys

**步骤8** 启动Hadoop

start-all.sh

**步骤9** 验证Hadoop

Jps



## 3.5 安装MySQL

**步骤1** 下载MySql依赖包和源码

cd /bigdata/

安装依赖库

yum install -y libaio\*

wget https://obs.cn-north-4.myhuaweicloud.com/obs-mirror-ftp4/database/mysql-5.7.27-aarch64.tar.gz

**步骤2** 解压MySql并复制

tar -xvf mysql-5.7.27-aarch64.tar.gz -C /usr/local/

**步骤3** 配置MySql

mv /usr/local/mysql-5.7.27-aarch64 /usr/local/mysql

mkdir -p /usr/local/mysql/logs

ln -sf /usr/local/mysql/my.cnf /etc/my.cnf

cp -rf /usr/local/mysql/extra/lib\* /usr/lib64/(yes 2)

mv /usr/lib64/libstdc++.so.6 /usr/lib64/libstdc++.so.6.old

ln -s /usr/lib64/libstdc++.so.6.0.24 /usr/lib64/libstdc++.so.6

groupadd -r mysql && useradd -r -g mysql -s /sbin/nologin -M mysql

chown -R mysql:mysql /usr/local/mysql

cp -rf /usr/local/mysql/support-files/mysql.server /etc/init.d/mysqld

chmod +x /etc/init.d/mysqld

systemctl enable mysqld

**步骤4** 配置环境变量

vim /etc/profile

在末尾添加：

export MYSQL\_HOME=/usr/local/mysql

export PATH=$PATH:$MYSQL\_HOME/bin

立即生效：

source /etc/profile

**步骤5** 初始化mysql

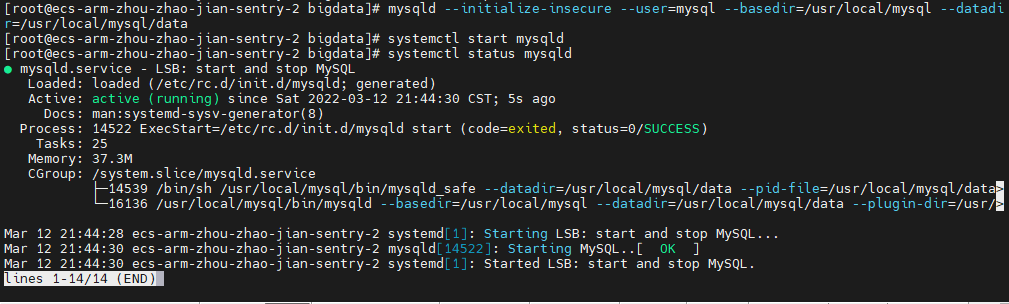
mysqld --initialize-insecure --user=mysql --basedir=/usr/local/mysql --datadir=/usr/local/mysql/data

启动mysql

systemctl start mysqld

查看状态

systemctl status mysqld



**步骤6** 修改root用户密码

mysql -u root

初始密码为空，直接回车登录

输入如下命令修改密码

use mysql;

update user set authentication\_string=password('123456') where user = 'root' and host = 'localhost';

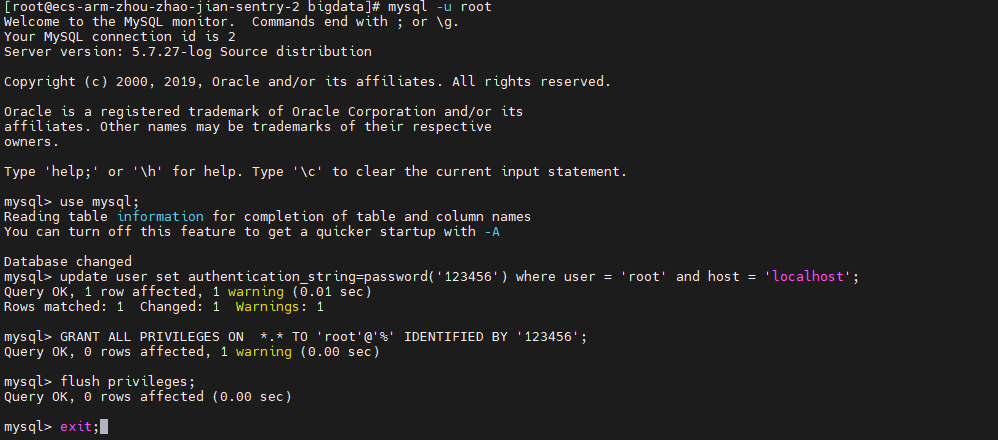
GRANT ALL PRIVILEGES ON \*.\* TO 'root'@'%' IDENTIFIED BY '123456';

# 刷新权限

flush privileges;

退出

exit;



## 3.6 安装Hive

**步骤1** 下载并解压Hive的安装包。

cd /bigdata

wget https://archive.apache.org/dist/hive/hive-1.1.0/apache-hive-1.1.0-bin.tar.gz

tar -zxvf apache-hive-1.1.0-bin.tar.gz

**步骤2** 配置Hive环境变量

vim /etc/profile

在末尾添加如下内容

export HIVE\_HOME=/bigdata/apache-hive-1.1.0-bin

export PATH=$PATH:$HIVE\_HOME/bin

:wq ，保存并退出，立即生效

source /etc/profile

**步骤3** 修改Hive配置文件

cd /bigdata/apache-hive-1.1.0-bin/conf

vim hive-site.xml

添加如下内容：

<configuration>

<property>

<name>javax.jdo.option.ConnectionURL</name>

<value>jdbc:mysql://localhost:3306/hive?createDatabaseIfNotExist=true&amp;useSSL=false</value>

</property>

<property>

<name>javax.jdo.option.ConnectionDriverName</name>

<value>com.mysql.jdbc.Driver</value>

</property>

<property>

<name>javax.jdo.option.ConnectionUserName</name>

<value>root</value>

</property>

<property>

<name>javax.jdo.option.ConnectionPassword</name>

<value>123456</value>

</property>

<property>

<name>hive.server2.thrift.bind.host</name>

<value>localhost</value>

</property>

<property>

<name>hive.server2.thrift.port</name>

<value>10000</value>

</property>

</configuration>

**步骤4** 下载JDBC连接包

cd /bigdata/apache-hive-1.1.0-bin/lib

wget https://repo1.maven.org/maven2/mysql/mysql-connector-java/5.1.47/mysql-connector-java-5.1.47.jar

替换hadoop中jline的jar包

cd /bigdata/apache-hive-1.1.0-bin

cp lib/jline-2.12.jar ../hadoop-2.6.0/share/hadoop/yarn/lib/

rm ${HADOOP\_HOME}/share/hadoop/yarn/lib/jline-0.9.94.jar

cp ${SENTRY\_HOME}/lib/jline-2.12.jar　${HADOOP\_HOME}/share/hadoop/yarn/lib

**步骤５** 初始化Hive原数据库

cd /bigdata/apache-hive-1.1.0-bin/

schematool -dbType mysql -initSchema

# hive 测试

# start-all.sh

# ４Sentry连接数据库

**步骤1** 将编译好的Sentry复制到工作路径并解压

cd /bigdata/sentry-release-1.6.0-rc0/sentry-dist/target

cp apache-sentry-1.6.0-incubating-bin.tar.gz /bigdata

**步骤2** 解压sentry。

cd /bigdata

tar -zxvf apache-sentry-1.6.0-incubating-bin.tar.gz

**步骤3** 下载mysql驱动

cd /bigdata/apache-sentry-1.6.0-incubating-bin/lib

wget https://repo1.maven.org/maven2/mysql/mysql-connector- java/5.1.47/mysql-connector-java-5.1.47.jar

**步骤4** 修改配置文件

vim /bigdata/apache-sentry-1.6.0-incubating-bin/conf/sentry-site.xml

添加如下内容

<?xml version="1.0"?>

<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>

<!--

Licensed to the Apache Software Foundation (ASF) under one or more

contributor license agreements. See the NOTICE file distributed with

this work for additional information regarding copyright ownership.

The ASF licenses this file to You under the Apache License, Version 2.0

(the "License"); you may not use this file except in compliance with

the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software

distributed under the License is distributed on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and

limitations under the License.

-->

<!-- WARNING!!! This file is provided for documentation purposes ONLY! -->

<!-- WARNING!!! You should copy to sentry-site.xml and make modification instead. -->

<configuration>

<property>

<name>sentry.service.allow.connect</name>

<value>hive,admin</value>

<description>comma separated list of users - List of users that are allowed to connect to the service (eg Hive, Impala) </description>

</property>

<property>

<name>sentry.store.jdbc.url</name>

<value>jdbc:mysql://localhost:3306/sentry?createDatabaseIfNotExist=true&amp;useSSL=false</value>

<description>JDBC connection URL for the backed DB</description>

</property>

<property>

<name>sentry.store.jdbc.user</name>

<value>root</value>

<description>Userid for connecting to backend db </description>

</property>

<property>

<name>sentry.store.jdbc.password</name>

<value>123456</value>

<description>Sentry password for backend JDBC user </description>

</property>

<property>

<name>sentry.service.server.rpcport</name>

<value>8038</value>

<description> TCP port number for service</description>

</property>

<property>

<name>sentry.service.server.rpcaddress</name>

<value>0.0.0.0</value>

<description> TCP interface for service to bind to</description>

</property>

<property>

<name>sentry.store.jdbc.driver</name>

<value>com.mysql.jdbc.Driver</value>

<description>Backend JDBC driver - org.apache.derby.jdbc.EmbeddedDriver (only when dbtype = derby) JDBC Driver class for the backed DB</description>

</property>

<property>

<name>sentry.service.admin.group</name>

<value>admin</value>

<description>Comma separates list of groups. List of groups allowed to make policy updates</description>

</property>

<property>

<name>sentry.store.group.mapping</name>

<value>org.apache.sentry.provider.common.HadoopGroupMappingService</value>

<description>

Group mapping class for Sentry service. org.apache.sentry.provider.file.LocalGroupMapping service can be used for local group mapping. </description>

</property>

<property>

<name>sentry.service.security.mode</name>

<value>none</value>

<description>Options: kerberos, none. Authentication mode for Sentry service. Currently supports Kerberos and trusted mode </description>

</property>

<property>

<name>sentry.service.server.principal</name>

<value> </value>

</property>

<property>

<name>sentry.service.server.keytab</name>

<value> </value>

</property>

<property>

<name>sentry.service.reporting</name>

<value>JMX</value>

</property>

<property>

<name>sentry.service.server.rpc-address</name>

<value>localhost</value>

</property>

<property>

<name>sentry.service.server.rpc-port</name>

<value>8038</value>

</property>

<property>

<name>sentry.hive.server</name>

<value>server1</value>

</property>

<property>

<name>sentry.service.web.enable</name>

<value>true</value>

</property>

<property>

<name>sentry.service.web.port</name>

<value>51000</value>

</property>

<property>

<name>sentry.service.web.authentication.type</name>

<value>NONE</value>

</property>

</configuration>

**步骤5** 配置Sentry环境变量

vim /etc/profile

在末尾添加

export SENTRY\_HOME=/bigdata/apache-sentry-1.6.0-incubating-bin

export PATH=$PATH:$SENTRY\_HOME/bin

立即生效

source /etc/profile

**步骤6** 在mysql中创建sentry数据库

cd /bigdata

mysql -uroot -p123456

create database sentry;

CREATE USER sentry IDENTIFIED BY 'sentry';

GRANT all ON sentry.\* TO sentry@'%' IDENTIFIED BY 'sentry';

flush privileges;

exit;

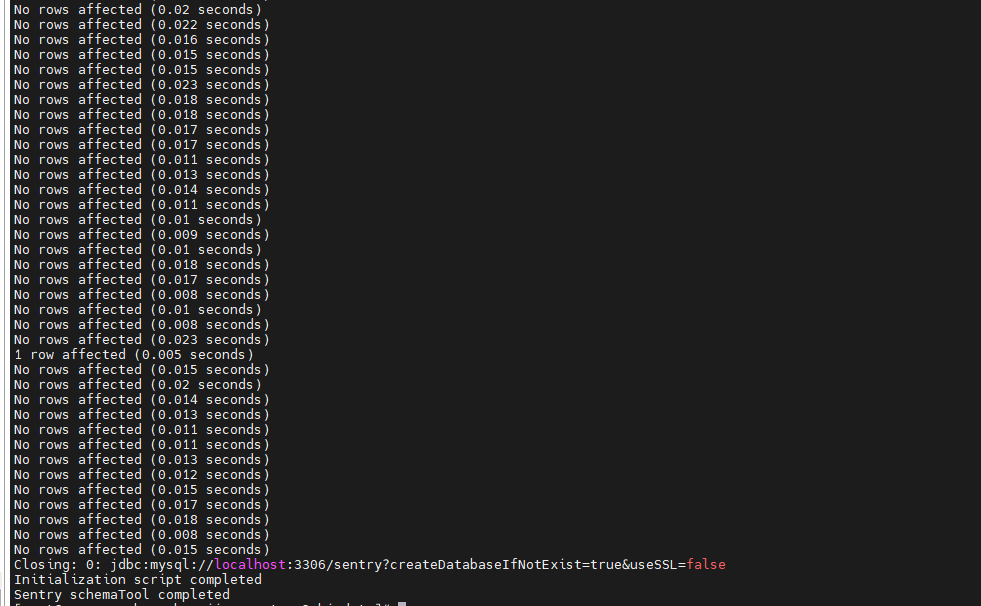
cd /bigdata/

chmod -R 755 /bigdata/

**步骤7** 初始化sentry数据库

cd /bigdata

sentry --command schema-tool --conffile ${SENTRY\_HOME}/conf/sentry-site.xml --dbType mysql –initSchema



创建日志目录

mkdir logs

**步骤8** 启动sentry服务

nohup sentry --command service --conffile ${SENTRY\_HOME}/conf/sentry-site.xml >> /bigdata/logs/sentry.log 2>&1 &

**步骤8** 修改hive-site.xml配置文件

cd /bigdata

vim /bigdata/apache-hive-1.1.0-bin/conf/hive-site.xml

修改为如下内容：

<configuration>

<property>

<name>javax.jdo.option.ConnectionURL</name>

<value>jdbc:mysql://localhost:3306/hive?createDatabaseIfNotExist=true&amp;useSSL=false</value>

</property>

<property>

<name>javax.jdo.option.ConnectionDriverName</name>

<value>com.mysql.jdbc.Driver</value>

</property>

<property>

<name>javax.jdo.option.ConnectionUserName</name>

<value>root</value>

</property>

<property>

<name>javax.jdo.option.ConnectionPassword</name>

<value>123456</value>

</property>

<property>

<name>hive.server2.thrift.bind.host</name>

<value>localhost</value>

</property>

<property>

<name>hive.server2.thrift.port</name>

<value>10000</value>

</property>

<property>

<name>hive.sentry.conf.url</name>

<value>file:///bigdata/apache-hive-1.1.0-bin/conf/sentry-site.xml</value>

</property>

<property>

<name>hive.stats.collect.scancols</name>

<value>true</value>

</property>

<property>

<name>hive.metastore.pre.event.listeners</name>

<value>org.apache.sentry.binding.metastore.MetastoreAuthzBinding</value>

</property>

<property>

<name>hive.metastore.event.listeners</name>

<value>org.apache.sentry.binding.metastore.SentryMetastorePostEventListener</value>

</property>

<property>

<name>hive.server2.session.hook</name>

<value>org.apache.sentry.binding.hive.HiveAuthzBindingSessionHook</value>

</property>

<property>

<name>hive.security.authorization.task.factory</name>

<value>org.apache.sentry.binding.hive.SentryHiveAuthorizationTaskFactoryImpl</value>

</property>

</configuration>

**步骤9** 修改sentry-site.xml配置文件

vim /bigdata/apache-hive-1.1.0-bin/conf/sentry-site.xml

添加如下内容：

<configuration>

<property>

<name>sentry.service.client.server.rpc-address</name>

<value>localhost</value>

</property>

<property>

<name>sentry.service.client.server.rpc-port</name>

<value>8038</value>

</property>

<property>

<name>sentry.service.client.server.rpc-connection-timeout</name>

<value>200000</value>

</property>

<!--配置认证-->

<property>

<name>sentry.service.security.mode</name>

<value>none</value>

</property>

<property>

<name>sentry.service.server.principal</name>

<value></value>

</property>

<property>

<name>sentry.service.server.keytab</name>

<value></value>

</property>

<property>

<name>sentry.provider</name>

<value>org.apache.sentry.provider.file.HadoopGroupResourceAuthorizationProvider</value>

</property>

<property>

<name>sentry.hive.provider.backend</name>

<value>org.apache.sentry.provider.db.SimpleDBProviderBackend</value>

</property>

<property>

<name>sentry.metastore.service.users</name>

<value>hive</value>

<!--queries made by hive user (beeline) skip meta store check-->

</property>

<property>

<name>sentry.hive.server</name>

<value>server1</value>

</property>

<property>

<name>sentry.hive.testing.mode</name>

<value>true</value>

</property>

</configuration>

**步骤10** 复制相关的jar包

cp ${SENTRY\_HOME}/lib/sentry\*.jar ${HIVE\_HOME}/lib

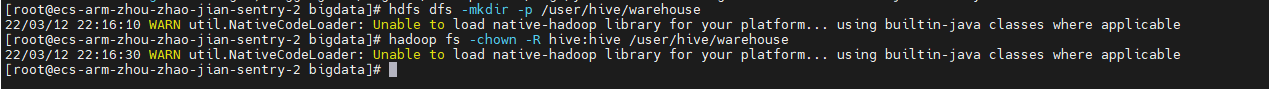
cp ${SENTRY\_HOME}/lib/shiro-\*.jar ${HIVE\_HOME}/lib

**步骤11** 创建warehouse并修改权限

hdfs dfs -mkdir -p /user/hive/warehouse

hadoop fs -chown -R hive:hive /user/hive/warehouse

hadoop fs -chmod -R 777 /user/hive/warehouse



**步骤12** 关闭sentry进程并重启

kill -9 `ps -aux|grep sentry|grep -v "grep"|awk '{print $2}'`

nohup sentry --command service --conffile ${SENTRY\_HOME}/conf/sentry-site.xml >> /bigdata/logs/sentry.log 2>&1 &

启动hive服务

nohup hive --service metastore >> /bigdata/logs/hive-metastore.log 2>&1 &

nohup hive --service hiveserver2 >> /bigdata/logs/hive-server2.log 2>&1 &

# 5 测试sentry

**步骤1** 创建测试数据

cd /bigdata

vim /bigdata/data/events.csv

# 添加如下测试数据

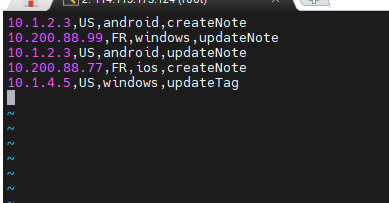
10.1.2.3,US,android,createNote

10.200.88.99,FR,windows,updateNote

10.1.2.3,US,android,updateNote

10.200.88.77,FR,ios,createNote

10.1.4.5,US,windows,updateTag



# 修改文件权限

chmod -R 755 /bigdata/

**步骤2** 添加用户

adduser hive

切换用户

su - hive

**步骤3** 在hive下初始化hive数据

Hive

create database sensitive;

# 创建表

create table sensitive.events (ip STRING, country STRING, client STRING, action STRING) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',';

# 加载本地测试数据

load data local inpath '/bigdata/data/events.csv' overwrite into table sensitive.events;

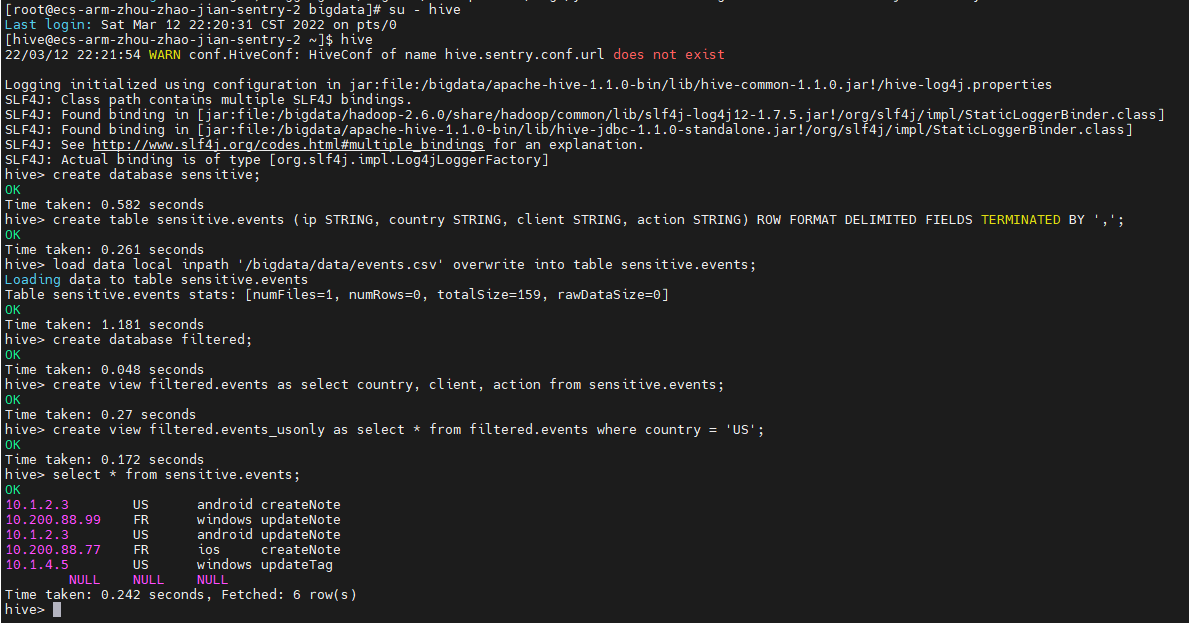
创建数据库和表

create database filtered;

create view filtered.events as select country, client, action from sensitive.events;

create view filtered.events\_usonly as select \* from filtered.events where country = 'US';

退出



exit;

logout

**步骤4** 验证sentry权限控制-添加角色组和赋予不同权限

创建admin 用户

adduser admin

切换用户

su - admin

使用beeline 连接工具

beeline -u "jdbc:hive2://localhost:10000/" -n admin -p admin -d org.apache.hive.jdbc.HiveDriver

create role admin\_role;

GRANT ALL ON SERVER server1 TO ROLE admin\_role;

GRANT ROLE admin\_role TO GROUP admin;

create role test\_role;

GRANT ALL ON DATABASE filtered TO ROLE test\_role;

use sensitive;

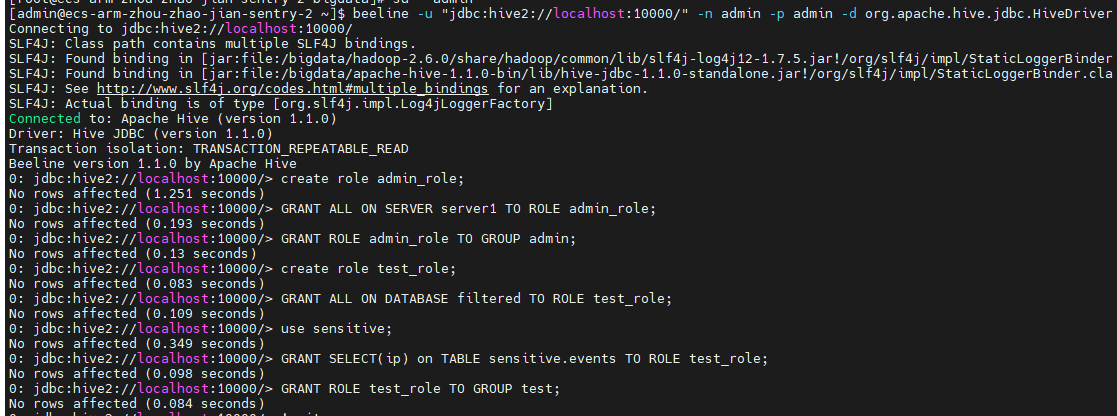
GRANT SELECT(ip) on TABLE sensitive.events TO ROLE test\_role;

GRANT ROLE test\_role TO GROUP test;

# 退出

!exit

# 退出hive命令行重新进入，



**步骤5** 验证admin\_role权限

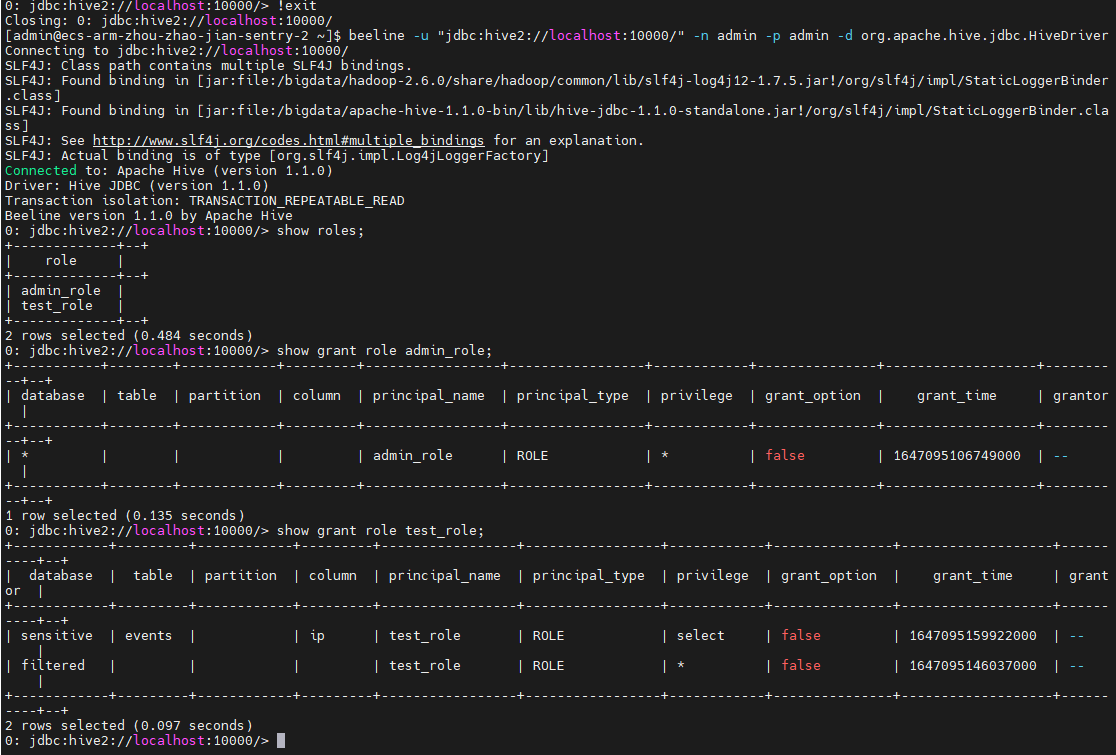
beeline -u "jdbc:hive2://localhost:10000/" -n admin -p admin -d org.apache.hive.jdbc.HiveDriver

# admin 用户 查看角色、权限、数据库等信息

show roles;

show grant role admin\_role;

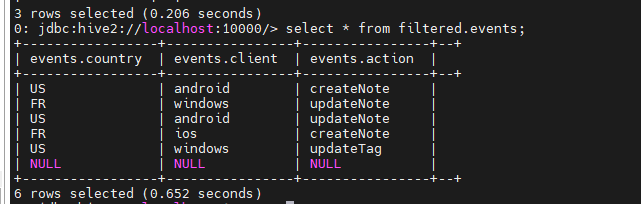
show grant role test\_role;



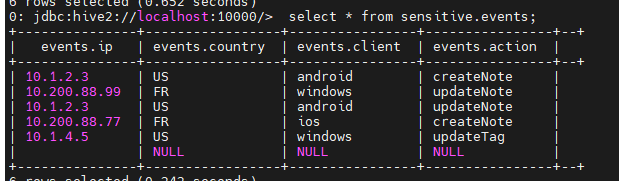
show databases;

# 测试查询操作

select \* from filtered.events;



select \* from sensitive.events;



!exit

logout

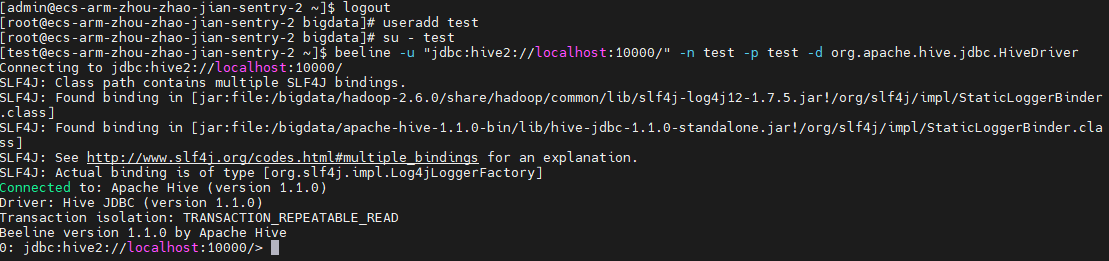
**步骤6** 退出hive命令行之后，验证test用户权限

# 添加test 用户

useradd test

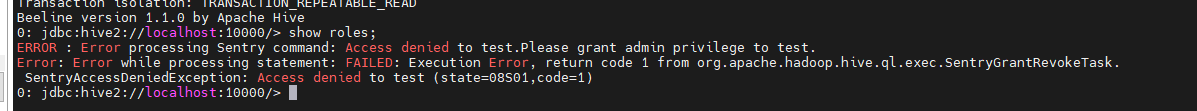
su - test

beeline -u "jdbc:hive2://localhost:10000/" -n test -p test -d org.apache.hive.jdbc.HiveDriver



# test用户不是管理员，是不能查看所有角色的

show roles;



# ERROR : Error processing Sentry command: Access denied to test.Please grant admin privilege to test.

# Error: Error while processing statement: FAILED: Execution Error, return code 1 from org.apache.hadoop.hive.ql.exec.SentryGrantRevokeTask.

# SentryAccessDeniedException: Access denied to test (state=08S01,code=1)

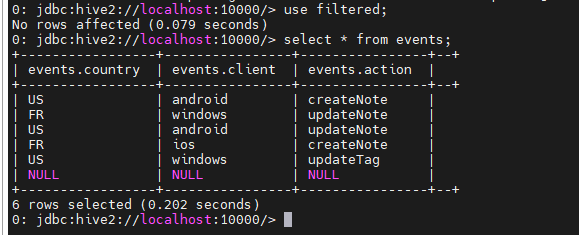
# test用户可以列出所有数据库

show databases;

# test用户拥有filtered库

use filtered;

select \* from events;

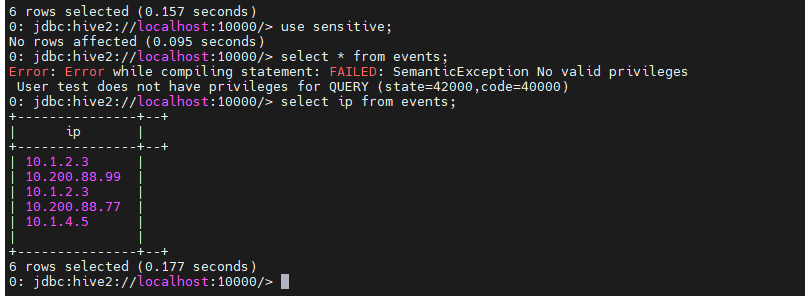


# test用户只能查看sensitive库中events表中的ip字段，其他的都都无法查看

# 可以正常查看 ip字段

use sensitive;

select ip from events;



# 6 安装配置Kerberos

1、yum install -y krb5-libs krb5-server krb5-workstation krb5-devel

2、下载这个网址的zip文件 <https://www.oracle.com/java/technologies/javase-jce8-downloads.html>

解压得到两个jar包，上传到服务器～/

mv local\_policy.jar $JAVA\_HOME/jre/lib/security

mv US\_export\_policy.jar $JAVA\_HOME/jre/lib/security

3、修改配置文件

vim /etc/krb5.conf

[logging]

default = FILE:/var/log/krb5libs.log

kdc = FILE:/var/log/krb5kdc.log

admin\_server = FILE:/var/log/kadmind.log

[libdefaults]

default\_realm = EXAMPLE.COM

dns\_lookup\_kdc = false

dns\_lookup\_realm = false

ticket\_lifetime = 86400

renew\_lifetime = 604800

forwardable = true

default\_tgs\_enctypes = rc4-hmac

default\_tkt\_enctypes = rc4-hmac

permitted\_enctypes = rc4-hmac

udp\_preference\_limit = 1

kdc\_timeout = 3000

[realms]

EXAMPLE.COM = {

kdc = localhost

admin\_server = localhost

}

vim /var/kerberos/krb5kdc/kdc.conf

[kdcdefaults]

kdc\_ports = 88

kdc\_tcp\_ports = 88

[realms]

EXAMPLE.COM = {

#master\_key\_type = aes256-cts

acl\_file = /var/kerberos/krb5kdc/kadm5.acl

dict\_file = /usr/share/dict/words

admin\_keytab = /var/kerberos/krb5kdc/kadm5.keytab

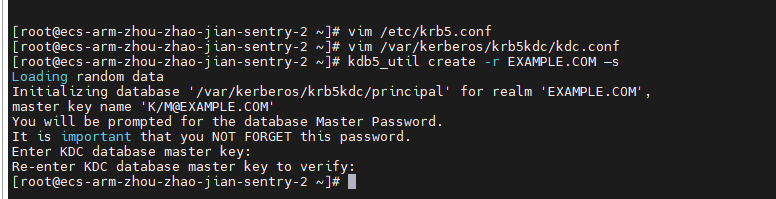
supported\_enctypes = aes256-cts:normal aes128-cts:normal des3-hmac-sha1:normal arcfour-hmac:normal camellia256-cts:normal camellia128-cts:normal des-hmac-sha1:normal des-cbc-md5:normal des-cbc-crc:normal

}

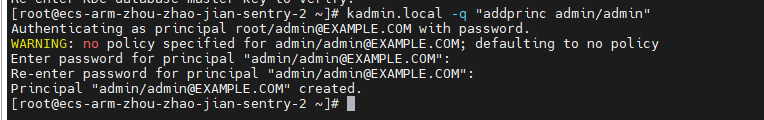
4、生成规则

kdb5\_util create -r EXAMPLE.COM –s

输入密码：admin, 两次

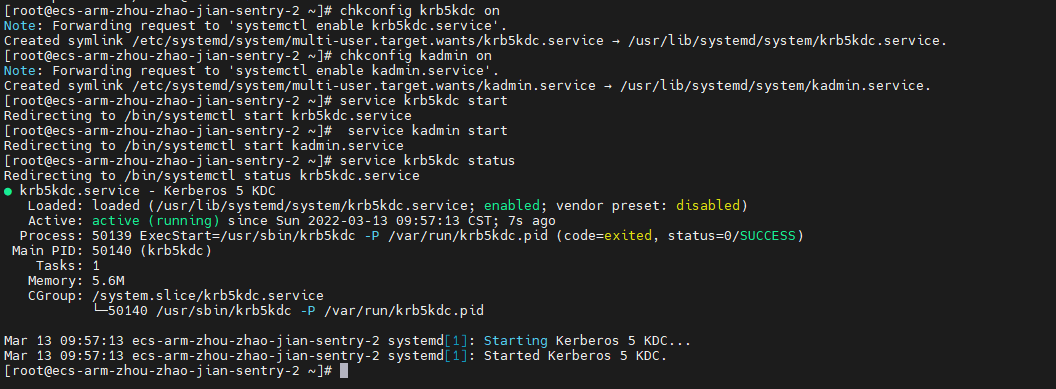


5、创建admin



6、开机自启动

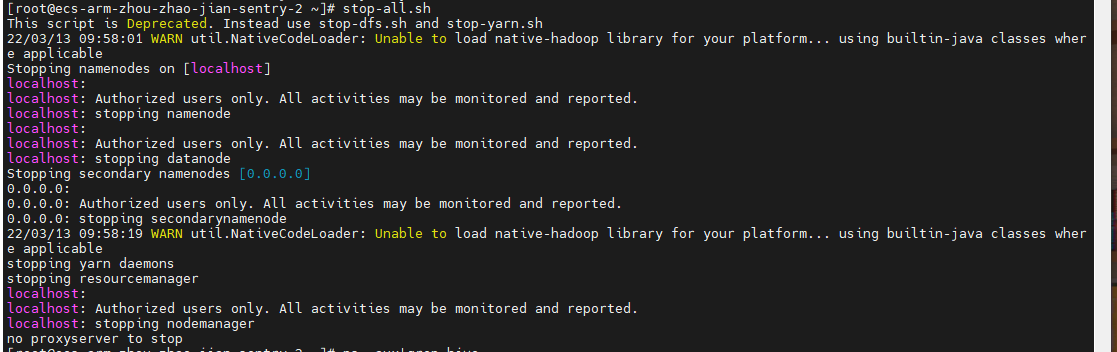
安装并启动成功



整合：Hadoop hive

1、先停止hadoop和hive

stop-all.sh

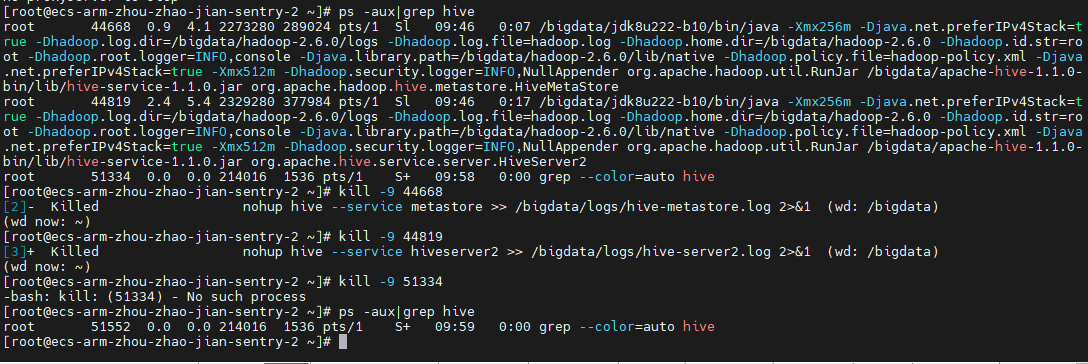


查看hive进程

ps -aux|grep hive

先关闭hive进程再进行配置

kill -9 hive进程号



2、分别创建hive，Hadoop配置文件

mkdir -p /etc/hadoop/conf

修改权限

chmod -R 755 /etc/hadoop/

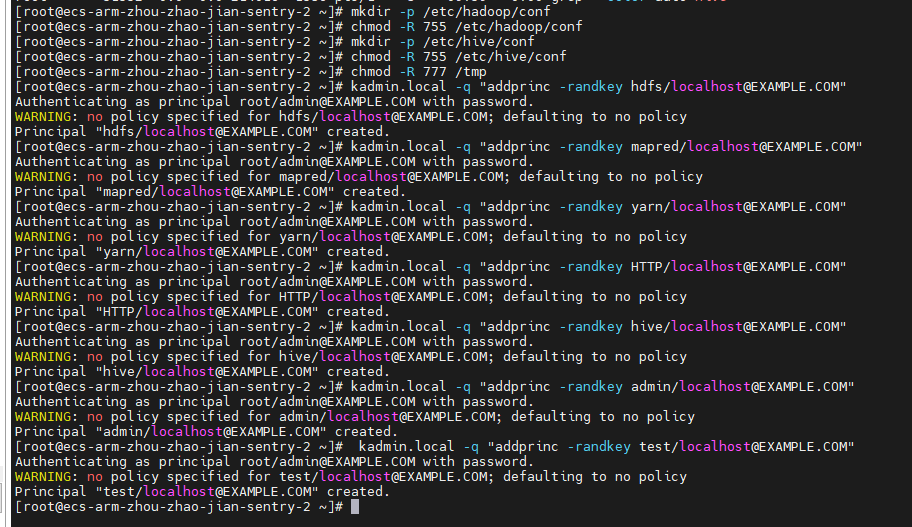
chmod -R 755 /etc/hadoop/conf

mkdir -p /etc/hive/conf

chmod -R 755 /etc/hive/

chmod -R 755 /etc/hive/conf

3、创建Kerberos服务端的规则



4、生成对应的keytab

kadmin.local -q "xst -norandkey -k /etc/hadoop/conf/hdfs.keytab hdfs/localhost@EXAMPLE.COM [HTTP/localhost@EXAMPLE.COM](mailto:HTTP/localhost@EXAMPLE.COM)"

kadmin.local -q "xst -norandkey -k /etc/hadoop/conf/mapred.keytab mapred/localhost@EXAMPLE.COM [HTTP/localhost@EXAMPLE.COM](mailto:HTTP/localhost@EXAMPLE.COM)"

kadmin.local -q "xst -norandkey -k /etc/hadoop/conf/yarn.keytab yarn/localhost@EXAMPLE.COM HTTP/localhost@EXAMPLE.COM "

kadmin.local -q "xst -norandkey -k /etc/hive/conf/hive.keytab hive/localhost@EXAMPLE.COM "

kadmin.local -q "xst -norandkey -k /etc/hive/conf/admin.keytab admin/localhost@EXAMPLE.COM "

kadmin.local -q "xst -norandkey -k /etc/hive/conf/test.keytab test/localhost@EXAMPLE.COM "

5、生成用户Kerberos票据，票据有时间限制

chown hdfs:hadoop /etc/hadoop/conf/hdfs.keytab

chown mapred:hadoop /etc/hadoop/conf/mapred.keytab

chown yarn:hadoop /etc/hadoop/conf/yarn.keytab

chown hive:hadoop /etc/hive/conf/hive.keytab

chown admin:hadoop /etc/hive/conf/admin.keytab

chown test:hadoop /etc/hive/conf/test.keytab

6、修改权限

chmod 400 /etc/hadoop/conf/\*.keytab

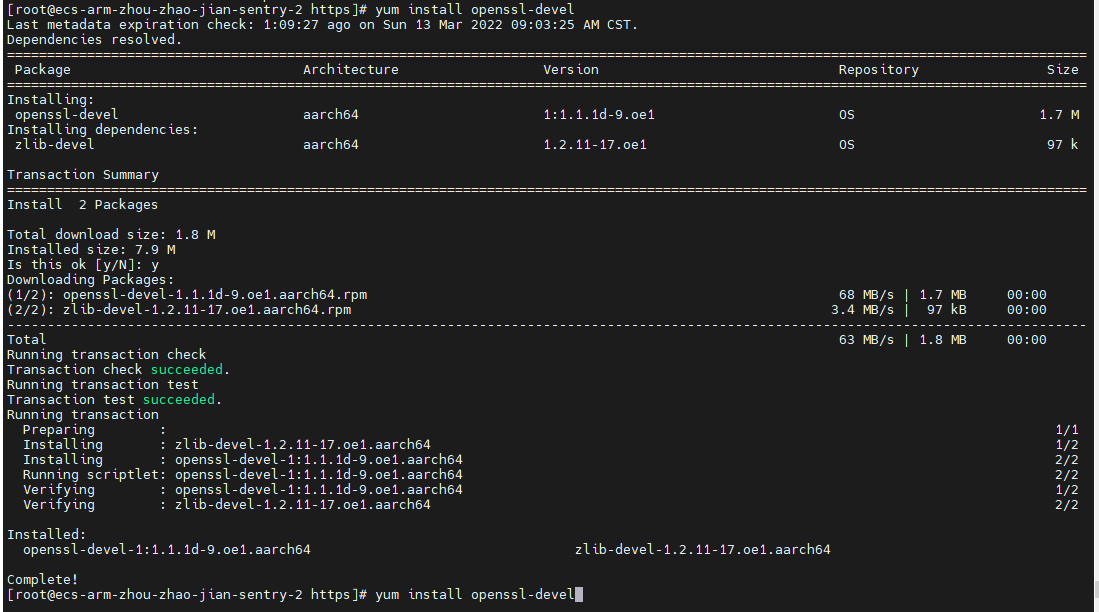
chmod 400 /etc/hive/conf/\*.keytab

6、添加用户组

groupadd hadoop;useradd hdfs -g hadoop -p hdfs;useradd hive -g hadoop -p hive;useradd yarn -g hadoop -p yarn;useradd mapred -g hadoop -p mapred

7、安装依赖

yum install openssl-devel



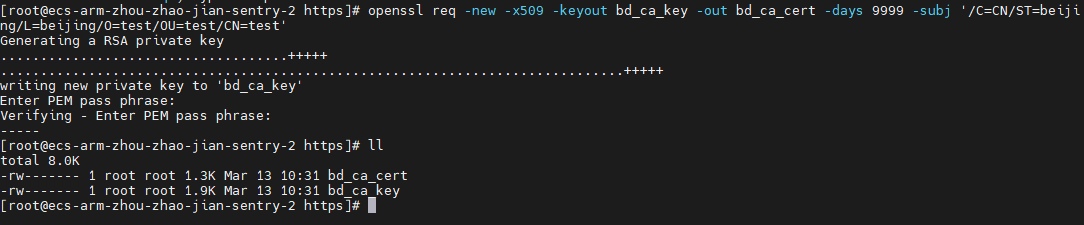
8、创建HTTPS证书

mkdir /etc/hadoop/https

cd /etc/hadoop/https/

openssl req -new -x509 -keyout bd\_ca\_key -out bd\_ca\_cert -days 9999 -subj '/C=CN/ST=beijing/L=beijing/O=test/OU=test/CN=test'

密码 全部 123456

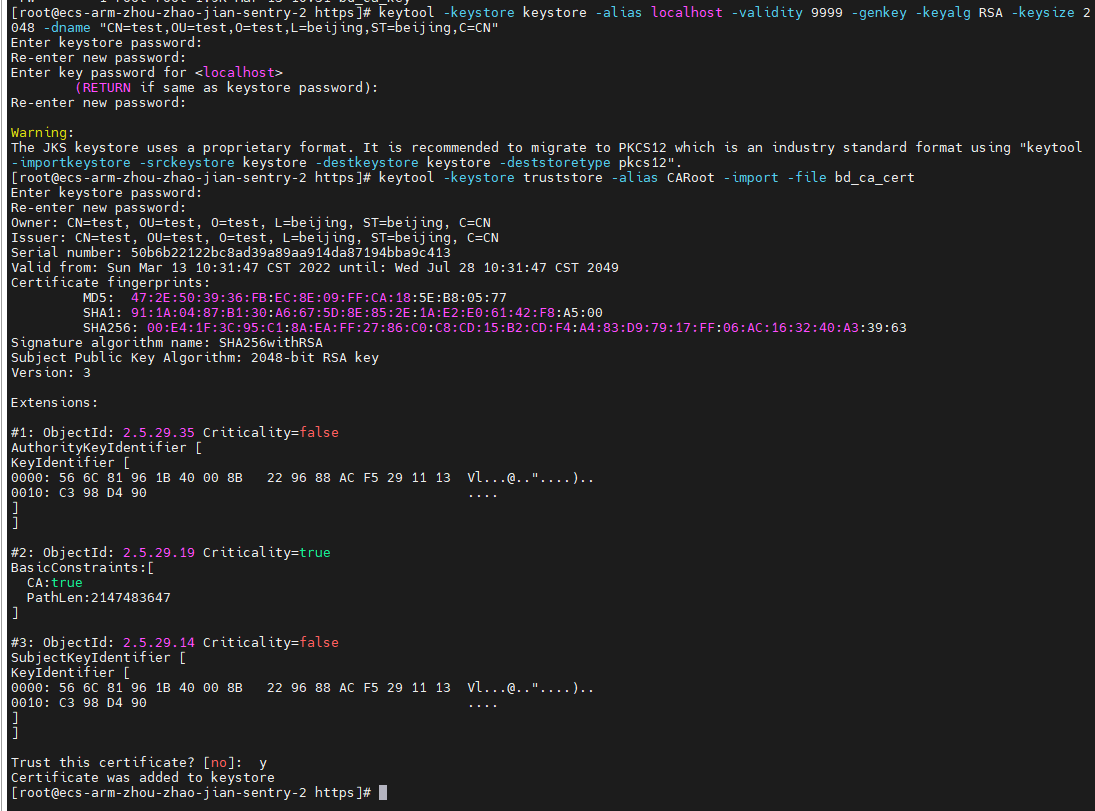


9、保存相关文件

keytool -keystore keystore -alias localhost -validity 9999 -genkey -keyalg RSA -keysize 2048 -dname "CN=test,OU=test,O=test,L=beijing,ST=beijing,C=CN"

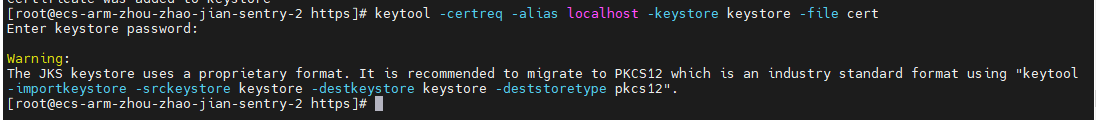
keytool -keystore truststore -alias CARoot -import -file bd\_ca\_cert

密码123456 ，并确定 y



keytool -certreq -alias localhost -keystore keystore -file cert

密码 123456

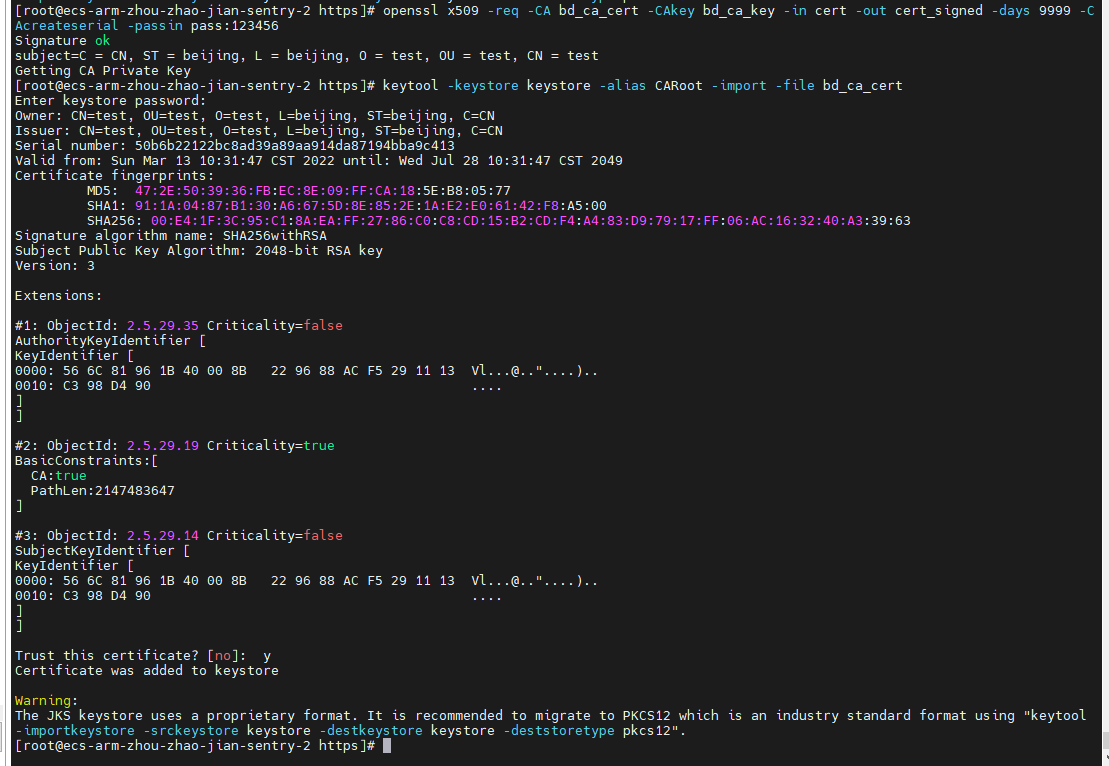


openssl x509 -req -CA bd\_ca\_cert -CAkey bd\_ca\_key -in cert -out cert\_signed -days 9999 -CAcreateserial -passin pass:123456

keytool -keystore keystore -alias CARoot -import -file bd\_ca\_cert

keytool -keystore keystore -alias localhost -import -file cert\_signed

输入密码123456



# 7 整合配置hadoop ,hive、Kerberos

**步骤1** 配置Hadoop

cd /bigdata/hadoop-2.6.0/etc/hadoop/

cp ssl-server.xml.example ssl-server.xml

vim ssl-server.xml

覆盖如下内容：

<?xml version="1.0"?>

<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>

<!--

Licensed to the Apache Software Foundation (ASF) under one or more

contributor license agreements. See the NOTICE file distributed with

this work for additional information regarding copyright ownership.

The ASF licenses this file to You under the Apache License, Version 2.0

(the "License"); you may not use this file except in compliance with

the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software

distributed under the License is distributed on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and

limitations under the License.

-->

<configuration>

<property>

<name>ssl.server.truststore.location</name>

<value>/etc/hadoop/https/truststore</value>

<description>Truststore to be used by NN and DN. Must be specified.

</description>

</property>

<property>

<name>ssl.server.truststore.password</name>

<value>123456</value>

<description>Optional. Default value is "".

</description>

</property>

<property>

<name>ssl.server.truststore.type</name>

<value>jks</value>

<description>Optional. The keystore file format, default value is "jks".

</description>

</property>

<property>

<name>ssl.server.truststore.reload.interval</name>

<value>10000</value>

<description>Truststore reload check interval, in milliseconds.

Default value is 10000 (10 seconds).

</description>

</property>

<property>

<name>ssl.server.keystore.location</name>

<value>/etc/hadoop/https/keystore</value>

<description>Keystore to be used by NN and DN. Must be specified.

</description>

</property>

<property>

<name>ssl.server.keystore.password</name>

<value>123456</value>

<description>Must be specified.

</description>

</property>

<property>

<name>ssl.server.keystore.keypassword</name>

<value>123456</value>

<description>Must be specified.

</description>

</property>

<property>

<name>ssl.server.keystore.type</name>

<value>jks</value>

<description>Optional. The keystore file format, default value is "jks".

</description>

</property>

</configuration>

配置客户端

cd /bigdata/hadoop-2.6.0/etc/hadoop/

cp ssl-client.xml.example ssl-client.xml

vim ssl-client.xml

覆盖

<?xml version="1.0"?>

<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>

<!--

Licensed to the Apache Software Foundation (ASF) under one or more

contributor license agreements. See the NOTICE file distributed with

this work for additional information regarding copyright ownership.

The ASF licenses this file to You under the Apache License, Version 2.0

(the "License"); you may not use this file except in compliance with

the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software

distributed under the License is distributed on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and

limitations under the License.

-->

<configuration>

<property>

<name>ssl.client.truststore.location</name>

<value>/etc/hadoop/https/truststore</value>

<description>Truststore to be used by clients like distcp. Must be

specified.

</description>

</property>

<property>

<name>ssl.client.truststore.password</name>

<value>123456</value>

<description>Optional. Default value is "".

</description>

</property>

<property>

<name>ssl.client.truststore.type</name>

<value>jks</value>

<description>Optional. The keystore file format, default value is "jks".

</description>

</property>

<property>

<name>ssl.client.truststore.reload.interval</name>

<value>10000</value>

<description>Truststore reload check interval, in milliseconds.

Default value is 10000 (10 seconds).

</description>

</property>

<property>

<name>ssl.client.keystore.location</name>

<value>/etc/hadoop/https/keystore</value>

<description>Keystore to be used by clients like distcp. Must be

specified.

</description>

</property>

<property>

<name>ssl.client.keystore.password</name>

<value>123456</value>

<description>Optional. Default value is "".

</description>

</property>

<property>

<name>ssl.client.keystore.keypassword</name>

<value>123456</value>

<description>Optional. Default value is "".

</description>

</property>

<property>

<name>ssl.client.keystore.type</name>

<value>jks</value>

<description>Optional. The keystore file format, default value is "jks".

</description>

</property>

</configuration>

**步骤2** 修改权限

chmod -R 755 /etc/hadoop/https/

cd /bigdata/

chown -R hive:hadoop apache-hive-1.1.0-bin

rm -rf hadoop-2.6.0/logs

mkdir -p hadoop-2.6.0/logs

chmod -R 777 hadoop-2.6.0

修改hadoop配置文件

cd hadoop-2.6.0/etc/hadoop/

vim core-site.xml

<?xml version="1.0" encoding="UTF-8"?>

<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>

<!--

Licensed under the Apache License, Version 2.0 (the "License");

you may not use this file except in compliance with the License.

You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software

distributed under the License is distributed on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and

limitations under the License. See accompanying LICENSE file.

-->

<!-- Put site-specific property overrides in this file. -->

<configuration>

<property>

<name>hadoop.tmp.dir</name>

<value>file:/usr/local/hadoop/tmp</value>

<description>Abase for other temporary directories.</description>

</property>

<property>

<name>fs.defaultFS</name>

<value>hdfs://localhost:9000</value>

</property>

<property>

<name>hadoop.proxyuser.hive.hosts</name>

<value>\*</value>

</property>

<property>

<name>hadoop.proxyuser.hive.groups</name>

<value>\*</value>

</property>

<property>

<name>hadoop.proxyuser.hdfs.hosts</name>

<value>\*</value>

</property>

<property>

<name>hadoop.proxyuser.hdfs.groups</name>

<value>\*</value>

</property>

<property>

<name>hadoop.proxyuser.HTTP.hosts</name>

<value>\*</value>

</property>

<property>

<name>hadoop.proxyuser.HTTP.groups</name>

<value>\*</value>

</property>

<property>

<name>hadoop.proxyuser.yarn.hosts</name>

<value>\*</value>

</property>

<property>

<name>hadoop.proxyuser.yarn.groups</name>

<value>\*</value>

</property>

<property>

<name>hadoop.security.authorization</name>

<value>true</value>

</property>

<property>

<name>hadoop.security.authentication</name>

<value>kerberos</value>

</property>

<property>

<name>hadoop.proxyuser.yarn.hosts</name>

<value>\*</value>

</property>

<property>

<name>hadoop.proxyuser.yarn.groups</name>

<value>\*</value>

</property>

</configuration>

**步骤3** 修改 hdfs-site.xml

<?xml version="1.0" encoding="UTF-8"?>

<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>

<!--

Licensed under the Apache License, Version 2.0 (the "License");

you may not use this file except in compliance with the License.

You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software

distributed under the License is distributed on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and

limitations under the License. See accompanying LICENSE file.

-->

<!-- Put site-specific property overrides in this file. -->

<configuration>

<property>

<name>dfs.replication</name>

<value>1</value>

</property>

<property>

<name>dfs.namenode.name.dir</name>

<value>file:/usr/local/hadoop/tmp/dfs/name</value>

</property>

<property>

<name>dfs.datanode.data.dir</name>

<value>file:/usr/local/hadoop/tmp/dfs/data</value>

</property>

<!-- General HDFS security config -->

<property>

<name>dfs.block.access.token.enable</name>

<value>true</value>

</property>

<!-- NameNode security config -->

<property>

<name>dfs.namenode.keytab.file</name>

<value>/etc/hadoop/conf/hdfs.keytab</value>

</property>

<property>

<name>dfs.namenode.kerberos.principal</name>

<value>hdfs/localhost@EXAMPLE.COM</value>

</property>

<property>

<name>dfs.namenode.kerberos.internal.spnego.principal</name>

<value>HTTP/localhost@EXAMPLE.COM</value>

</property>

<!-- Secondary NameNode security config -->

<property>

<name>dfs.secondary.namenode.keytab.file</name>

<value>/etc/hadoop/conf/hdfs.keytab</value>

</property>

<property>

<name>dfs.secondary.namenode.kerberos.principal</name>

<value>hdfs/localhost@EXAMPLE.COM</value>

</property>

<property>

<name>dfs.secondary.namenode.kerberos.internal.spnego.principal</name>

<value>HTTP/localhost@EXAMPLE.COM</value>

</property>

<!-- DataNode security config -->

<property>

<name>dfs.datanode.data.dir.perm</name>

<value>700</value>

</property>

<!--<property>

<name>dfs.datanode.address</name>

<value>0.0.0.0:1004</value>

</property>

<property>

<name>dfs.datanode.http.address</name>

<value>0.0.0.0:1006</value>

</property>-->

<property>

<name>dfs.datanode.keytab.file</name>

<value>/etc/hadoop/conf/hdfs.keytab</value>

</property>

<property>

<name>dfs.datanode.kerberos.principal</name>

<value>hdfs/localhost@EXAMPLE.COM</value>

</property>

<!-- Web Authentication config -->

<property>

<name>dfs.web.authentication.kerberos.principal</name>

<value>HTTP/localhost@EXAMPLE.COM</value>

</property>

<property>

<name>dfs.journalnode.keytab.file</name>

<value>/etc/hadoop/conf/hdfs.keytab</value>

</property>

<property>

<name>dfs.journalnode.kerberos.principal</name>

<value>hdfs/localhost@EXAMPLE.COM</value>

</property>

<property>

<name>dfs.journalnode.kerberos.internal.spnego.principal</name>

<value>HTTP/localhost@EXAMPLE.COM</value>

</property>

<property>

<name>dfs.data.transfer.protection</name>

<value>integrity</value>

</property>

<property>

　　　　<name>dfs.http.policy</name>

　　　　<value>HTTPS\_ONLY</value>

</property>

</configuration>

**步骤4** 修改 yarn-site.xml

vim yarn-site.xml

完全覆盖

<?xml version="1.0"?>

<!--

Licensed under the Apache License, Version 2.0 (the "License");

you may not use this file except in compliance with the License.

You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software

distributed under the License is distributed on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and

limitations under the License. See accompanying LICENSE file.

-->

<configuration>

<!-- Site specific YARN configuration properties -->

<property>

<name>yarn.nodemanager.aux-services</name>

<value>mapreduce\_shuffle</value>

</property>

<!-- ResourceManager security configs -->

<property>

<name>yarn.resourcemanager.keytab</name>

<value>/etc/hadoop/conf/yarn.keytab</value>

</property>

<property>

<name>yarn.resourcemanager.principal</name>

<value>yarn/localhost@EXAMPLE.COM</value>

</property>

<!-- NodeManager security configs -->

<property>

<name>yarn.nodemanager.keytab</name>

<value>/etc/hadoop/conf/yarn.keytab</value>

</property>

<property>

<name>yarn.nodemanager.principal</name>

<value>yarn/localhost@EXAMPLE.COM</value>

</property>

<property>

<name>yarn.resourcemanager.proxy-user-privileges.enabled</name>

<value>true</value>

</property>

</configuration>

**步骤5** 修改 mapred-site.xml

vim mapred-site.xml

完全覆盖

<?xml version="1.0"?>

<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>

<!--

Licensed under the Apache License, Version 2.0 (the "License");

you may not use this file except in compliance with the License.

You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software

distributed under the License is distributed on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and

limitations under the License. See accompanying LICENSE file.

-->

<!-- Put site-specific property overrides in this file. -->

<configuration>

<property>

<name>mapreduce.framework.name</name>

<value>yarn</value>

</property>

<property>

<name>mapreduce.jobhistory.keytab</name>

<value>/etc/hadoop/conf/mapred.keytab</value>

</property>

<property>

<name>mapreduce.jobhistory.principal</name>

<value>mapred/localhost@EXAMPLE.COM</value>

</property>

<property>

<name>mapreduce.jobhistory.webapp.spnego-principal</name>

<value>HTTP/localhost@EXAMPLE.COM</value>

</property>

<property>

<name>mapreduce.jobhistory.webapp.spnego-keytab-file</name>

<value>/etc/hadoop/conf/mapred.keytab</value>

</property>

</configuration>

**步骤5** 修改/bigdata/apache-hive-1.1.0-bin/conf/hive-site.xml

完全覆盖

<configuration>

<property>

<name>javax.jdo.option.ConnectionURL</name>

<value>jdbc:mysql://localhost:3306/hive?createDatabaseIfNotExist=true&amp;useSSL=false</value>

</property>

<property>

<name>javax.jdo.option.ConnectionDriverName</name>

<value>com.mysql.jdbc.Driver</value>

</property>

<property>

<name>javax.jdo.option.ConnectionUserName</name>

<value>root</value>

</property>

<property>

<name>javax.jdo.option.ConnectionPassword</name>

<value>123456</value>

</property>

<property>

<name>hive.server2.thrift.bind.host</name>

<value>localhost</value>

</property>

<property>

<name>hive.server2.thrift.port</name>

<value>10000</value>

</property>

<property>

<name>hive.sentry.conf.url</name>

<value>file:///bigdata/apache-hive-1.1.0-bin/conf/sentry-site.xml</value>

</property>

<property>

<name>hive.stats.collect.scancols</name>

<value>true</value>

</property>

<property>

<name>hive.metastore.pre.event.listeners</name>

<value>org.apache.sentry.binding.metastore.MetastoreAuthzBinding</value>

</property>

<property>

<name>hive.metastore.event.listeners</name>

<value>org.apache.sentry.binding.metastore.SentryMetastorePostEventListener</value>

</property>

<property>

<name>hive.server2.session.hook</name>

<value>org.apache.sentry.binding.hive.HiveAuthzBindingSessionHook</value>

</property>

<property>

<name>hive.security.authorization.task.factory</name>

<value>org.apache.sentry.binding.hive.SentryHiveAuthorizationTaskFactoryImpl</value>

</property>

<property>

<name>hive.server2.authentication</name>

<value>KERBEROS</value>

</property>

<property>

<name>hive.server2.authentication.kerberos.principal</name>

<value>hive/localhost@EXAMPLE.COM</value>

</property>

<property>

<name>hive.server2.authentication.kerberos.keytab</name>

<value>/etc/hive/conf/hive.keytab</value>

</property>

<property>

<name>hive.metastore.sasl.enabled</name>

<value>true</value>

</property>

<property>

<name>hive.metastore.kerberos.keytab.file</name>

<value>/etc/hive/conf/hive.keytab</value>

</property>

<property>

<name>hive.metastore.kerberos.principal</name>

<value>hive/localhost@EXAMPLE.COM</value>

</property>

</configuration>

**步骤5** 启动Hadoop各个组件

启动hdfs

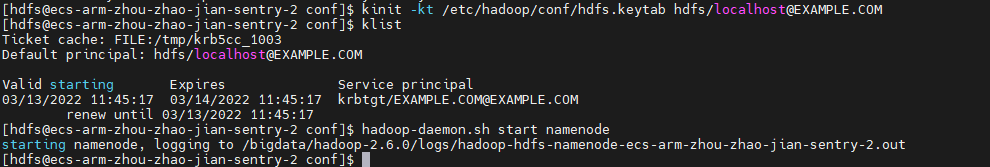
切换用户：su – hdfs

获取Kerberos票据

kinit -kt /etc/hadoop/conf/hdfs.keytab [hdfs/localhost@EXAMPLE.COM](mailto:hdfs/localhost@EXAMPLE.COM)

hadoop-daemon.sh start namenode

正常启动如下：

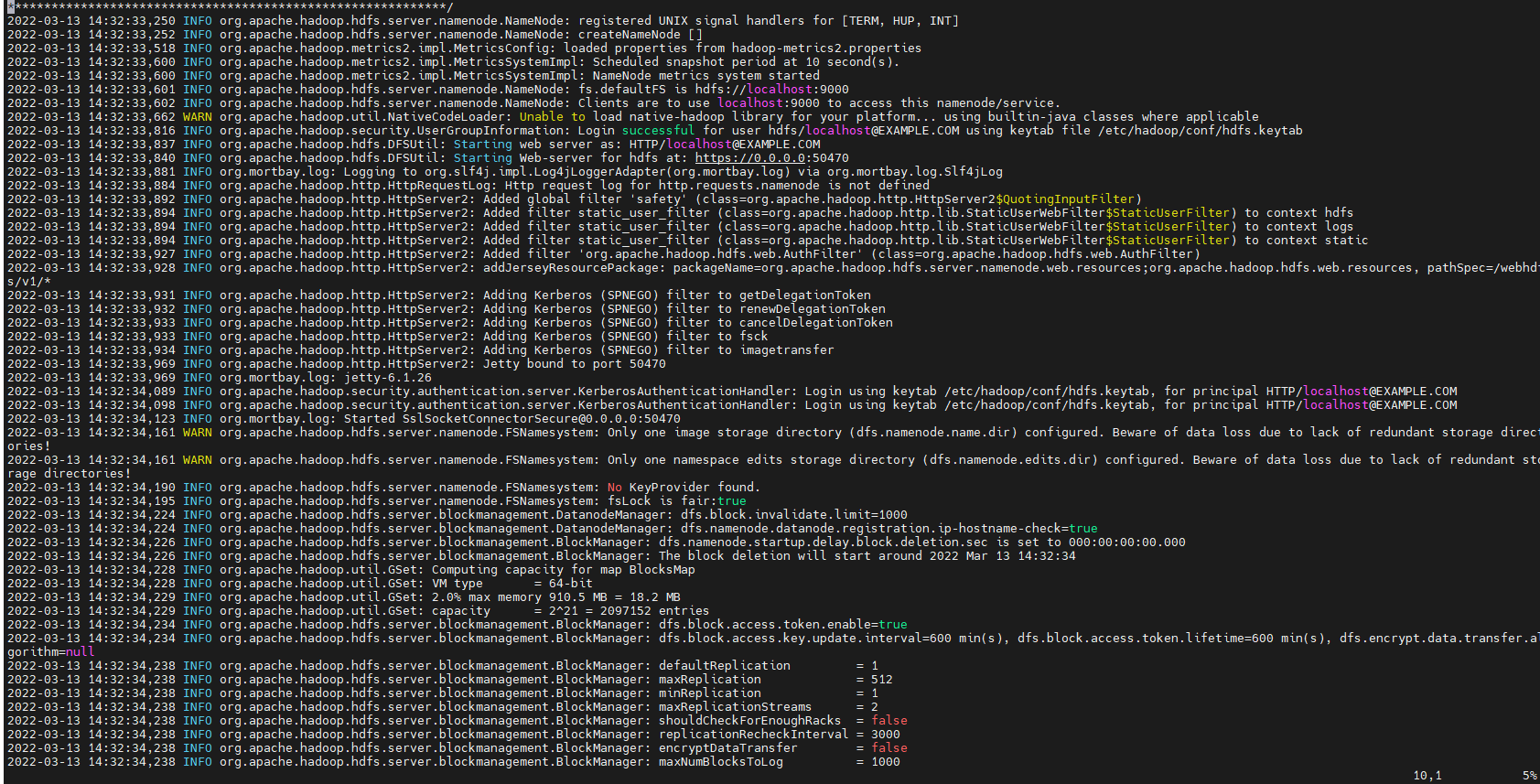


查看启动日志

vim /bigdata/hadoop-2.6.0/logs/hadoop-hdfs-namenode-ecs-arm-zhou-zhao-jian-sentry-2.log

查看namenode

启动成功



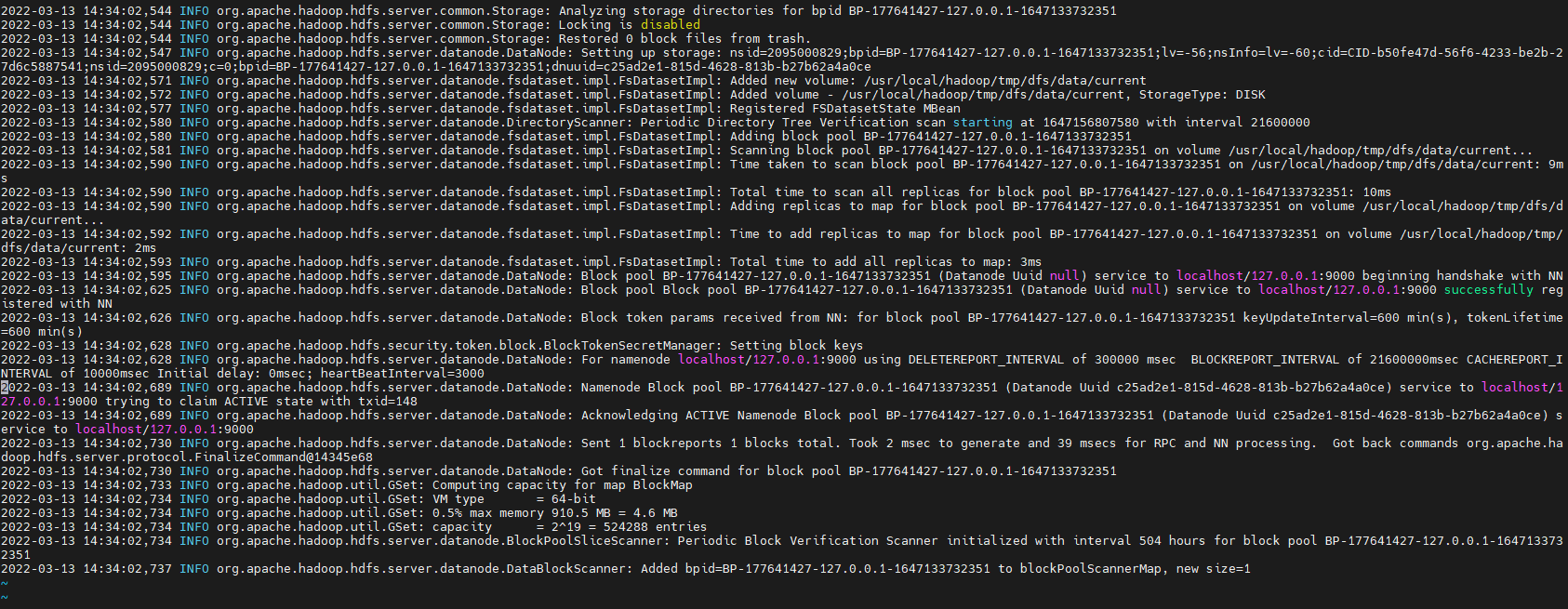
启动datanode

hadoop-daemon.sh start datanode

vim /bigdata/hadoop-2.6.0/logs/hadoop-hdfs-datanode-ecs-arm-zhou-zhao-jian-sentry-2.log

查看datanode

启动成功



退出

exit;

**步骤5** 启动jobhistory

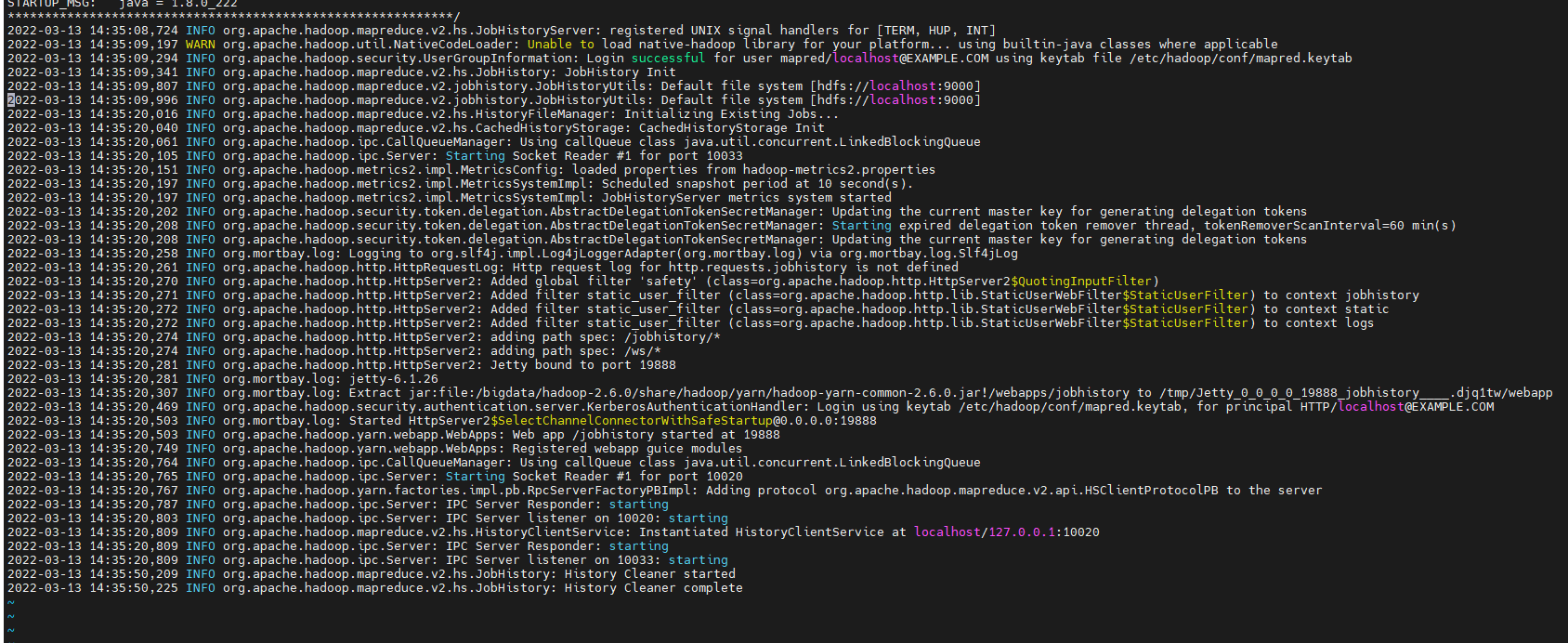
su - mapred

mr-jobhistory-daemon.sh start historyserver

vim /bigdata/hadoop-2.6.0/logs/mapred-mapred-historyserver-ecs-arm-zhou-zhao-jian-sentry-2.log

查看mapred

启动成功



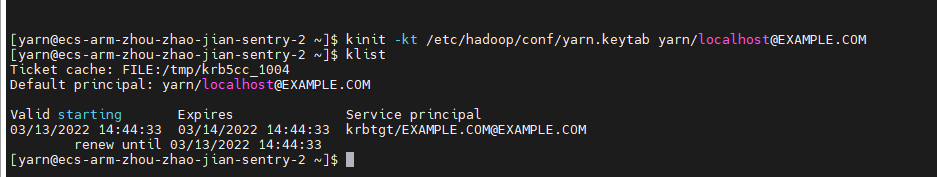
退出

exit;

**步骤6**启动yarn resourcemanager nodemanager

su – yarn

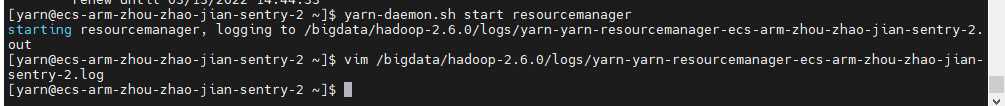
kinit -kt /etc/hadoop/conf/yarn.keytab [yarn/localhost@EXAMPLE.COM](mailto:yarn/localhost@EXAMPLE.COM)

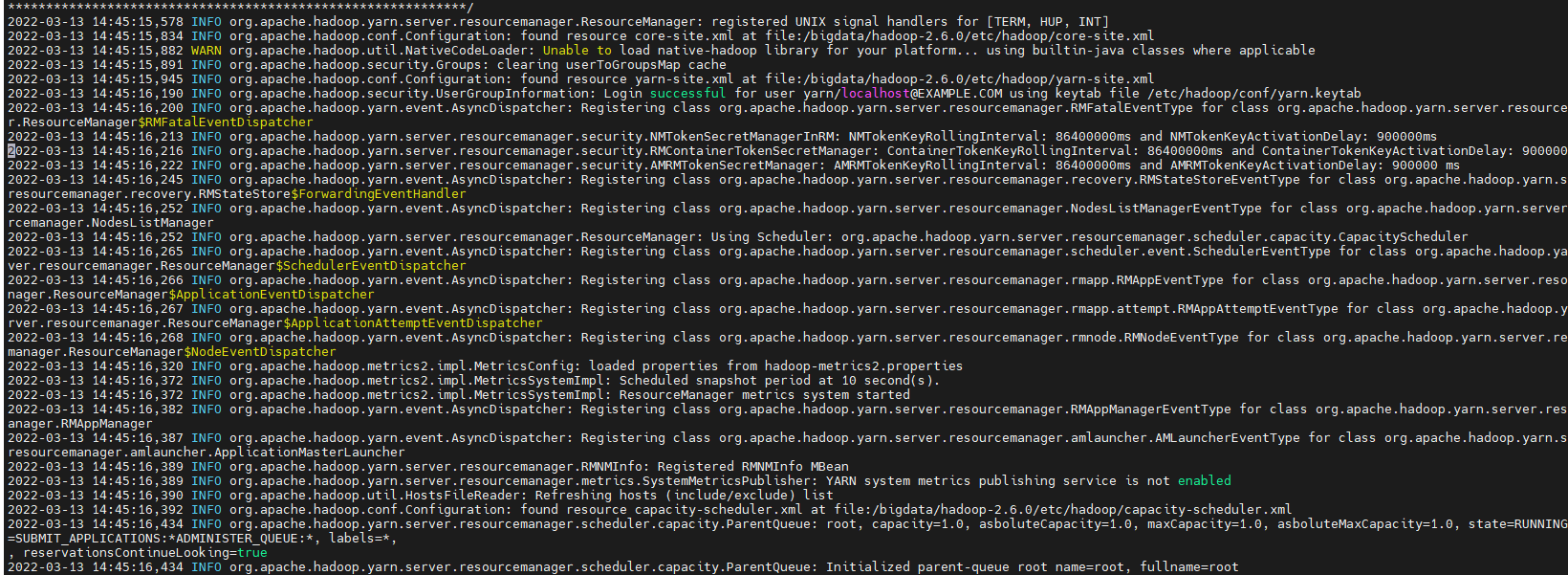


yarn-daemon.sh start resourcemanager

#查看启动日志，是否有报错

vim /bigdata/hadoop-2.6.0/logs/yarn-yarn-resourcemanager-ecs-arm-zhou-zhao-jian-sentry-2.log

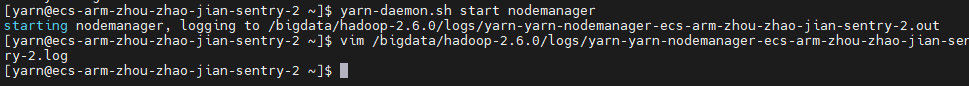


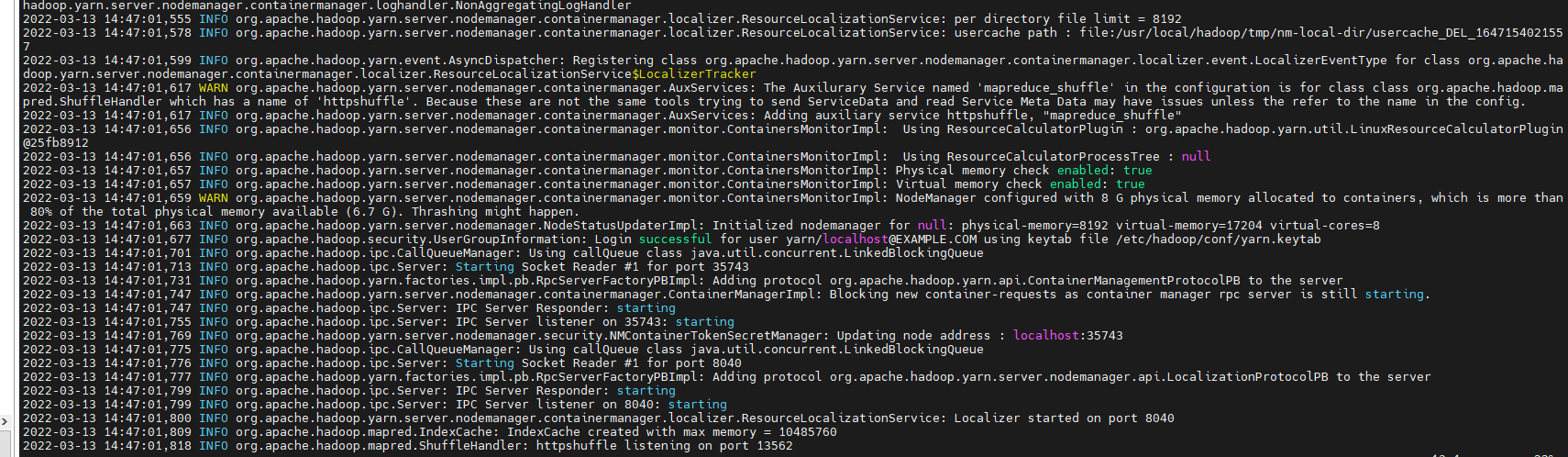


**步骤7** nodemanager

yarn-daemon.sh start nodemanager

vim /bigdata/hadoop-2.6.0/logs/yarn-yarn-nodemanager-ecs-arm-zhou-zhao-jian-sentry-2.log



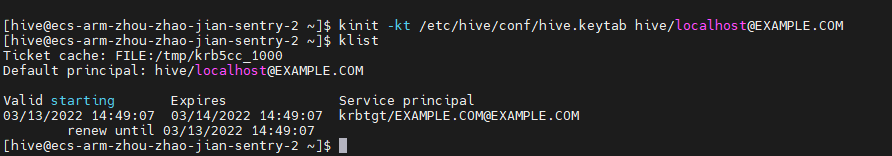


exit;

**步骤8**启动hive

su – hive

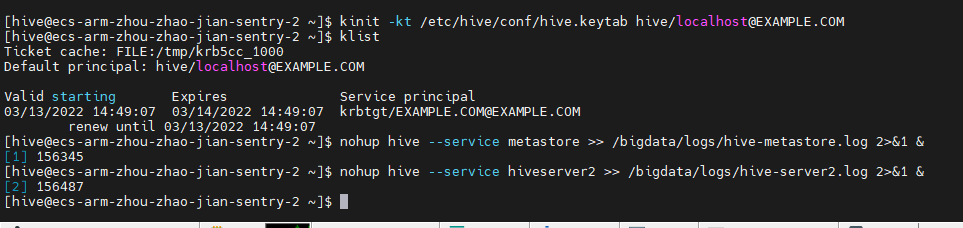
kinit -kt /etc/hive/conf/hive.keytab [hive/localhost@EXAMPLE.COM](mailto:hive/localhost@EXAMPLE.COM)



后台启动hive

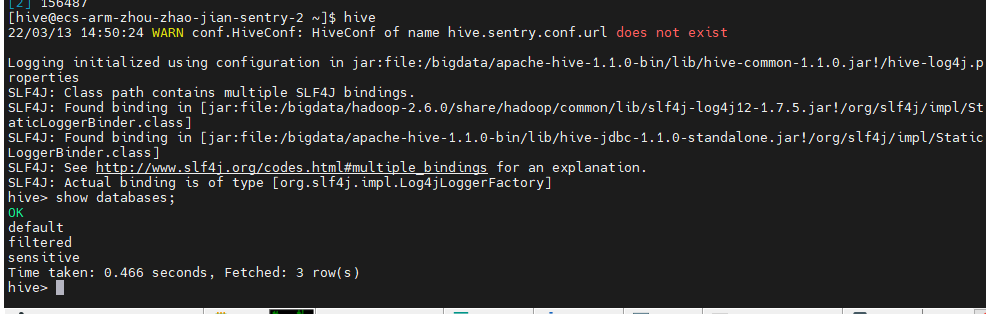
nohup hive --service metastore >> /bigdata/logs/hive-metastore.log 2>&1 &

nohup hive --service hiveserver2 >> /bigdata/logs/hive-server2.log 2>&1 &



查看

show databases;



退出

exit;

# 8整合完毕，验证sentry+Kerberos权限控制

**步骤1: hive 用户权限验证**

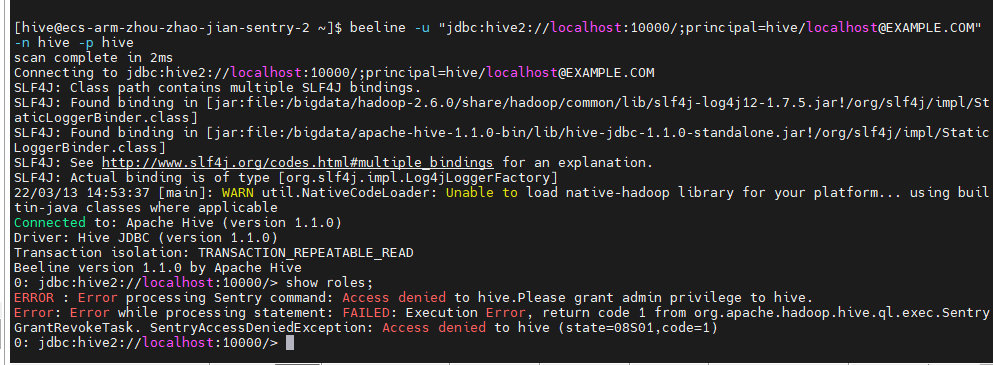
hive用户在sentry上前面是没有任何授权的，理论是没有权限

使用hive账户登录

su – hive

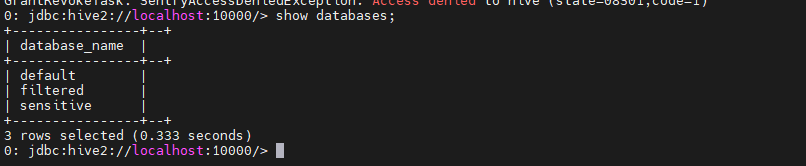
使用beeline连接

beeline -u "jdbc:hive2://localhost:10000/;principal=hive/localhost@EXAMPLE.COM" -n hive -p hive

因为sentry权限控制保护，hive用户不能查看roles 

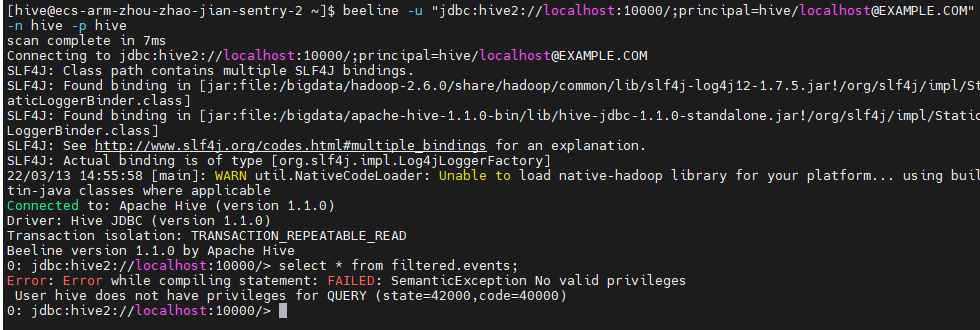
sentry生效

可以查看hive数据库



sentry没有给hive用户select权限

select ip from sensitive.events;



退出hive2 :!quit

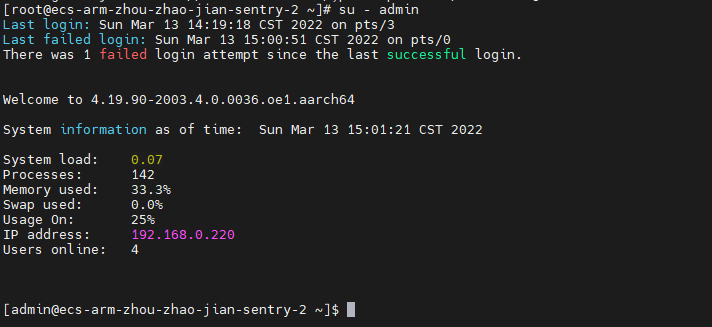
退出hive 用户 logout

**步骤2 admin 用户权限验证**

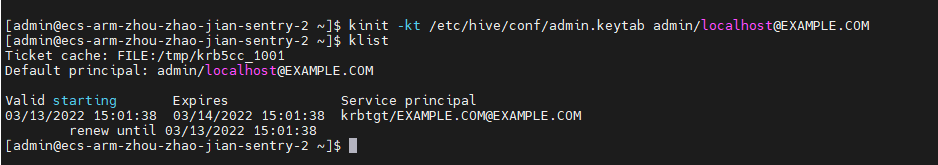
登录admin 用户

cd

su – admin



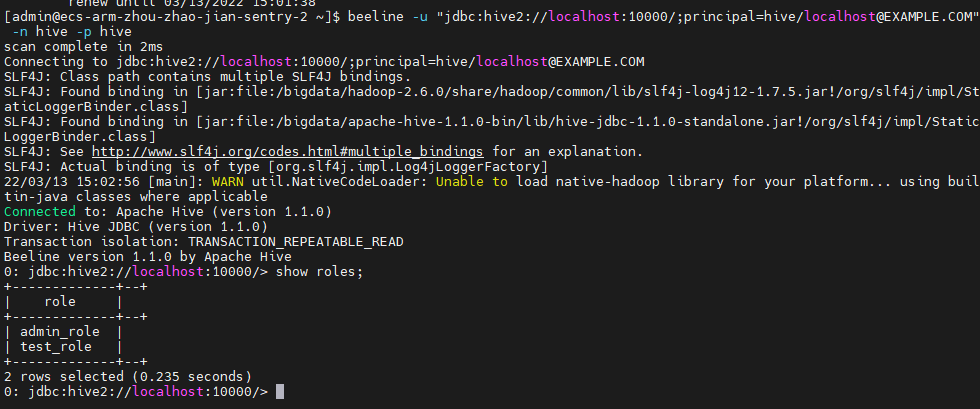
拿到admin用户kerberos票据，验证Kerberos权限，登录



连接hive

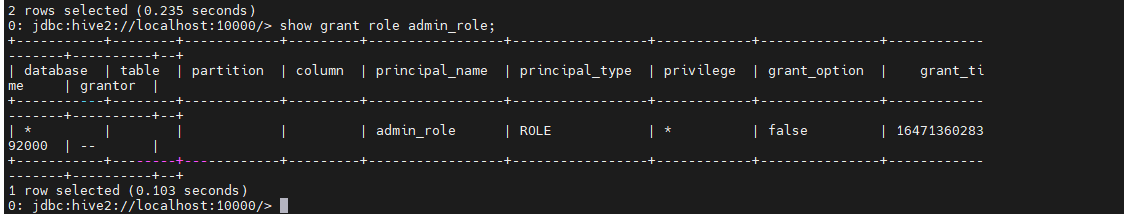
beeline -u "jdbc:hive2://localhost:10000/;principal=hive/localhost@EXAMPLE.COM" -n hive -p hive

连接成功，并显示角色



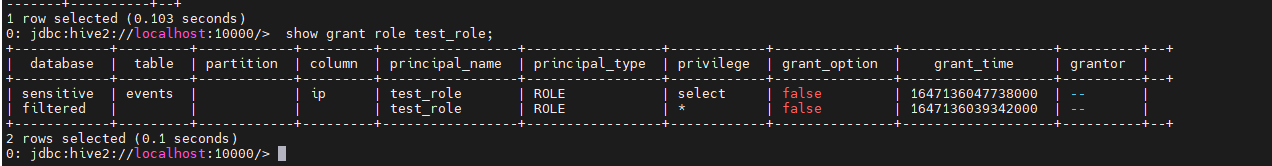
查看admin角色权限

show grant role admin\_role;



查看test用户的权限

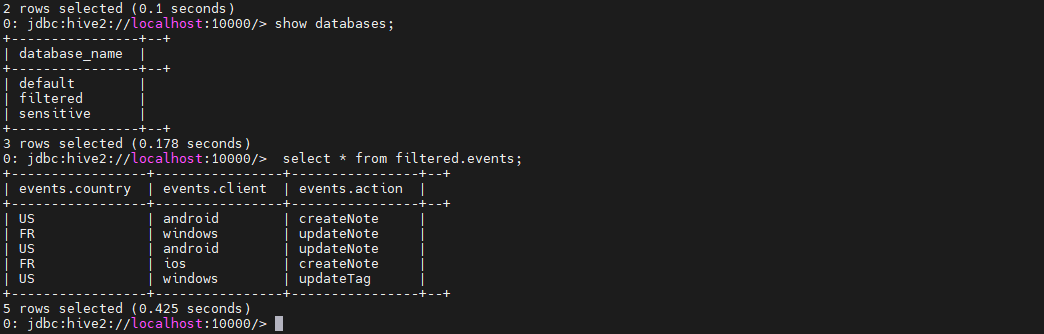
show grant role test\_role;



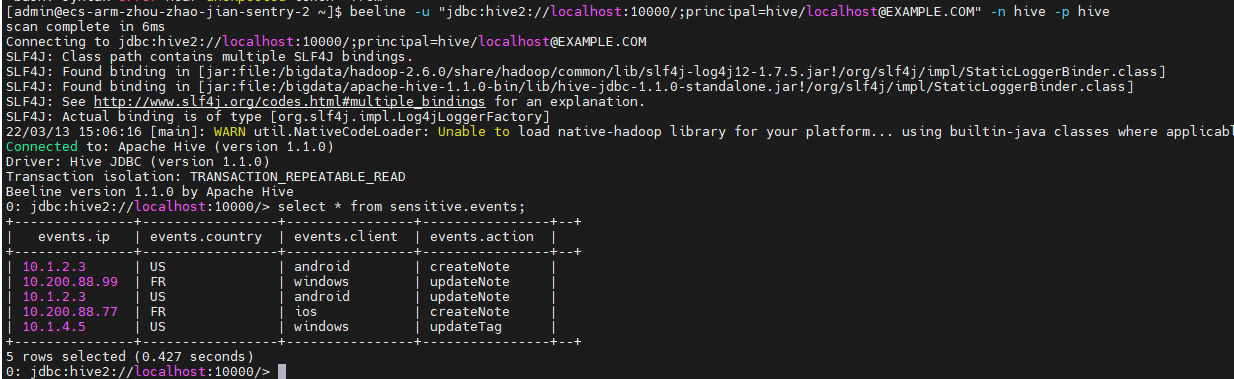
查看数据库并进行查询操作

show databases;

select \* from filtered.events;



select \* from sensitive.events;



admin用户权限验证完成

退出，!quit,并 logout

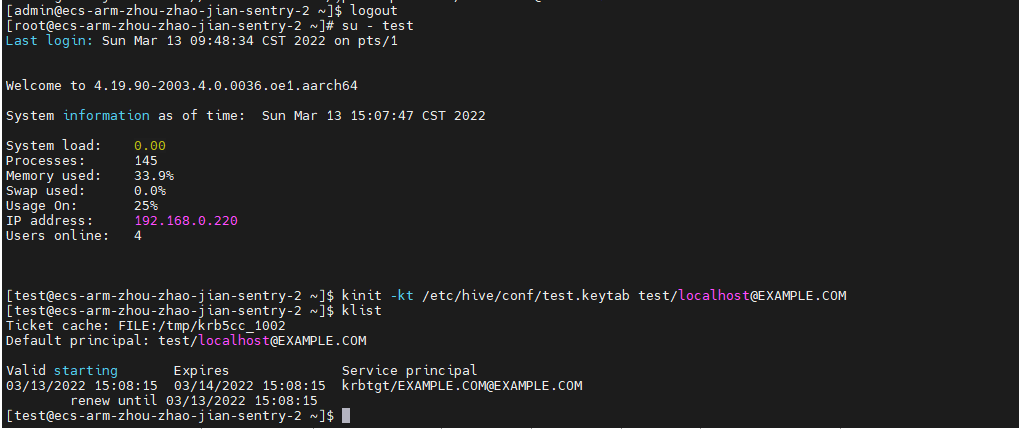
**步骤3 test用户权限验证**

cd

su – test

拿到test用户kerberos票据，验证Kerberos权限，登录

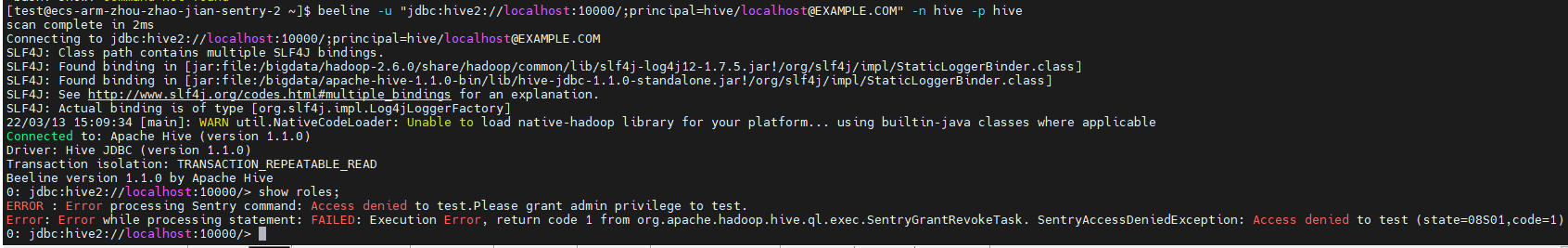
kinit -kt /etc/hive/conf/test.keytab [test/localhost@EXAMPLE.COM](mailto:test/localhost@EXAMPLE.COM)



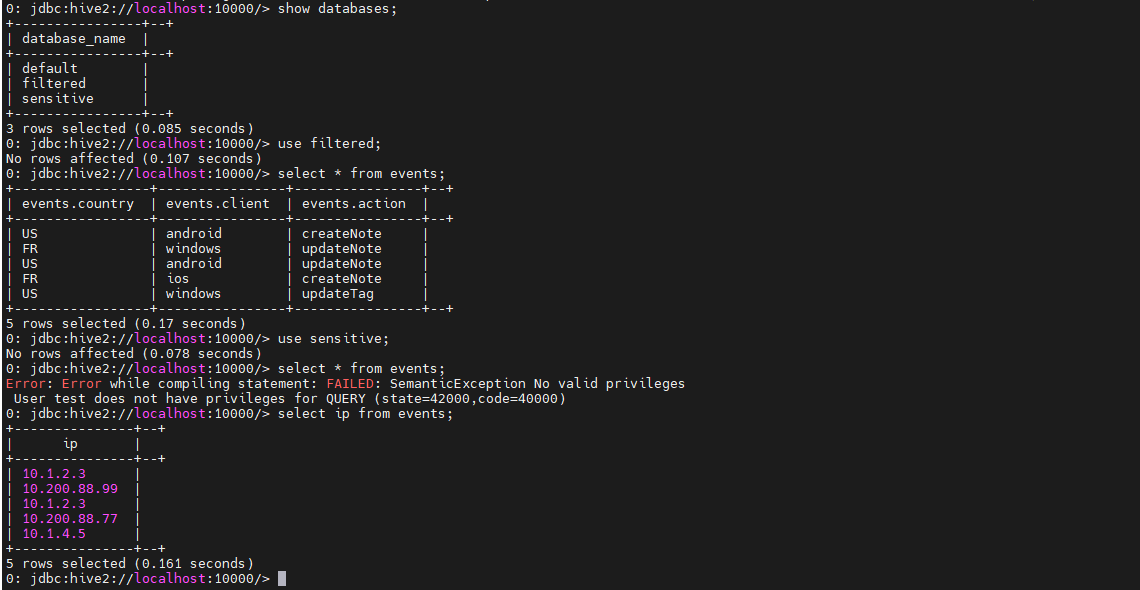
beeline连接

beeline -u "jdbc:hive2://localhost:10000/;principal=hive/localhost@EXAMPLE.COM" -n hive -p hive

test用户不是管理员，是不能查看所有角色的



test用户能查看数据库，能查filtered数据库，只能查sensitive库里面events表的ip字段，其余不能查询。



**步骤4 没有票据用户验证**

添加用户

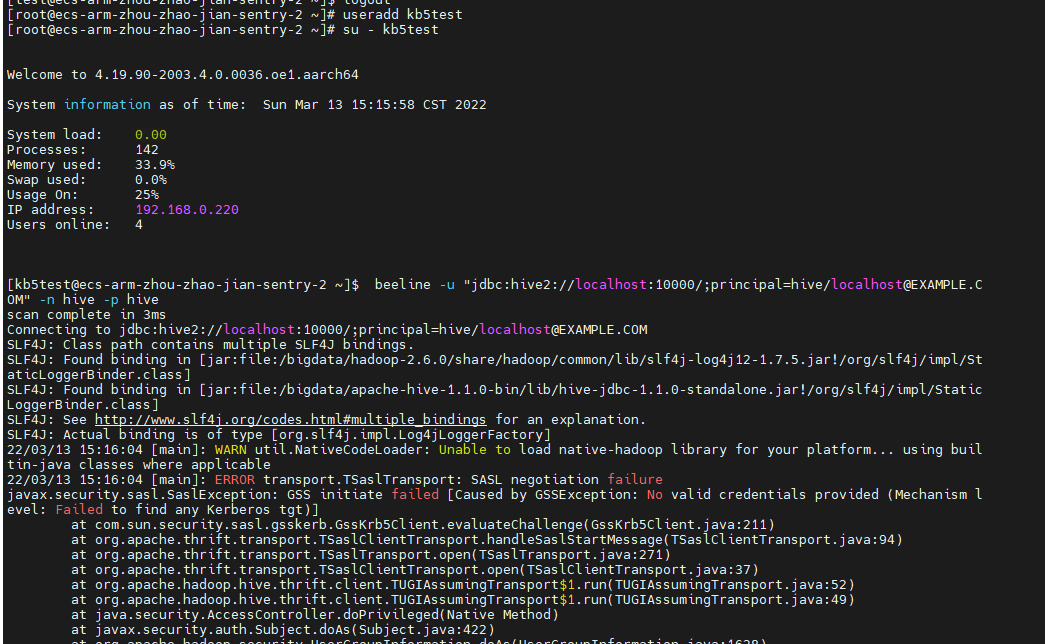
useradd kb5test

切换用户

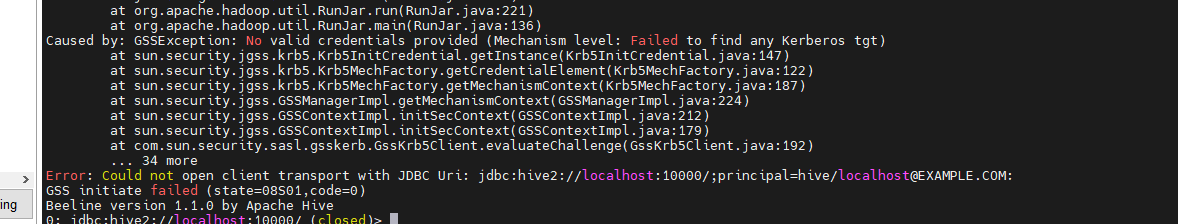
su - kb5test

beeline连接：

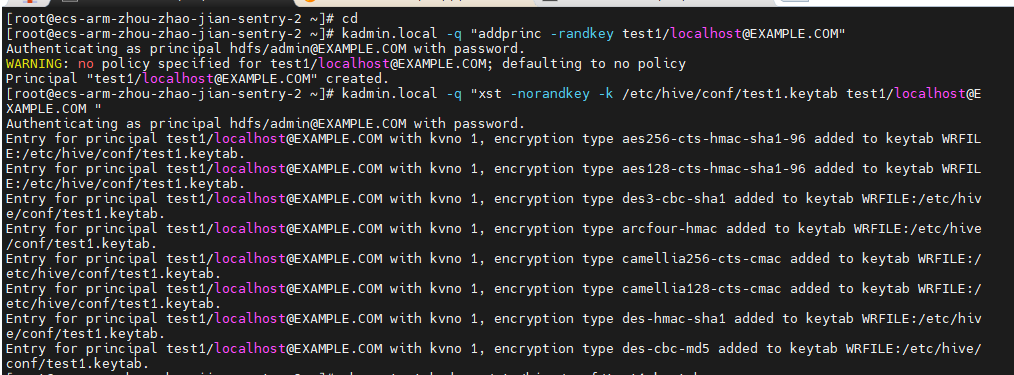
beeline -u "jdbc:hive2://localhost:10000/;principal=hive/localhost@EXAMPLE.COM" -n hive -p hive



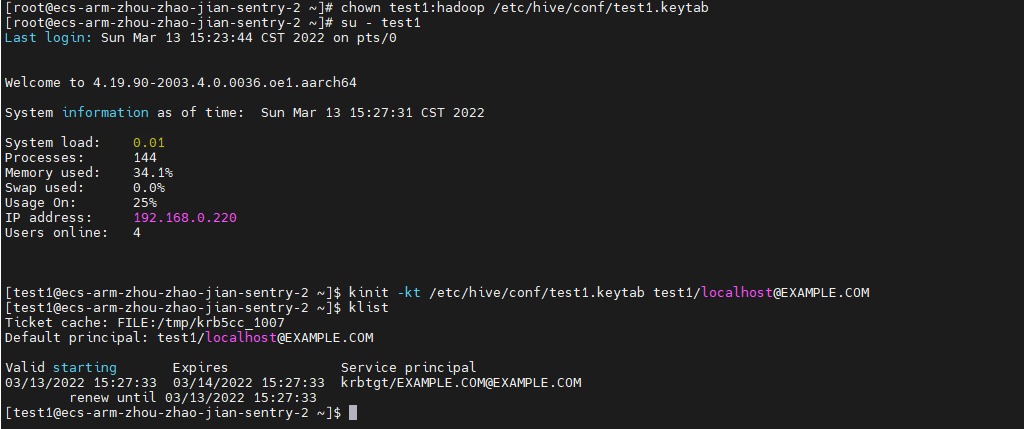
未通过Kerberos验证



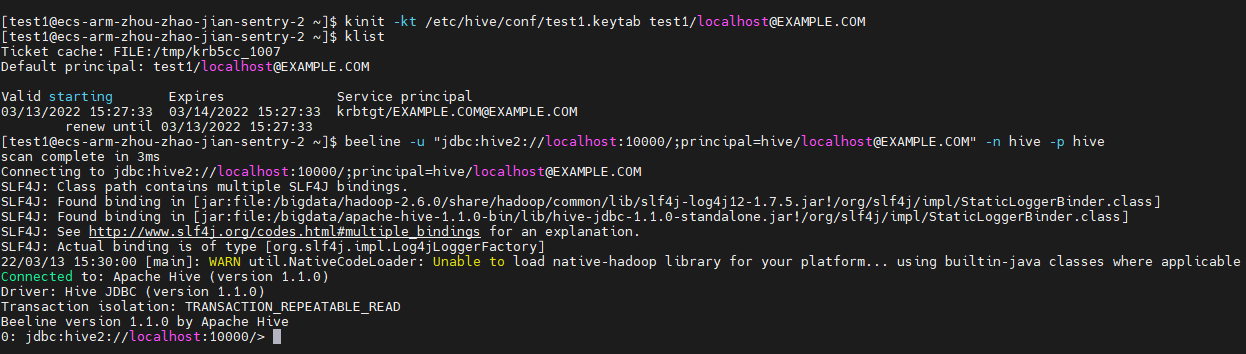
**步骤6 创建新用户，生成Kerberos票据**



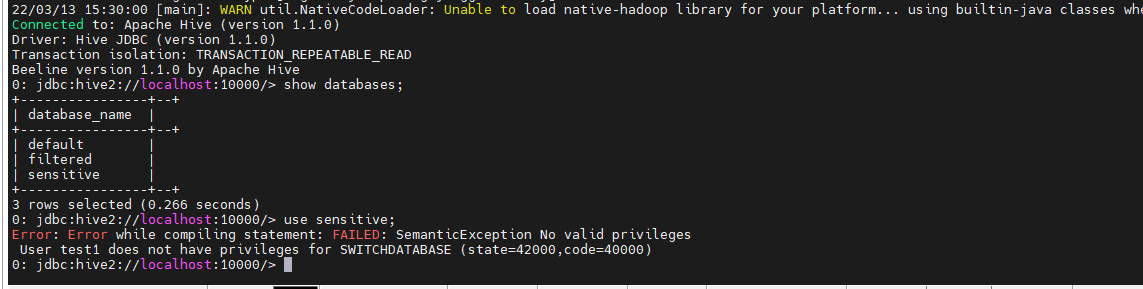
生成票据，并获取



beeline 连接



成功连接，但是未授权，



验证完毕！



rpm包安装成功截图

