# 搭建大数据平台

# 一.集群搭建准备（以3 台机器为例）

## 1.1准备机器

|  |  |
| --- | --- |
| 主机名 | IP |
| s115 | 172.18.18.115 |
| s116 | 172.18.18.116 |
| s117 | 172.18.18.117 |

## 1.2下载集群脚本及安装包

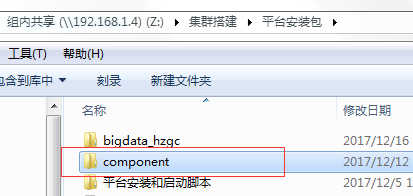
进入到/opt目录，从git 上下载集群脚本文件夹ClusterBuildScripts，并修改权限。执行如下命令：

cd /opt

git clone https://github.com/Zz897426182/ClusterBuildScripts.git

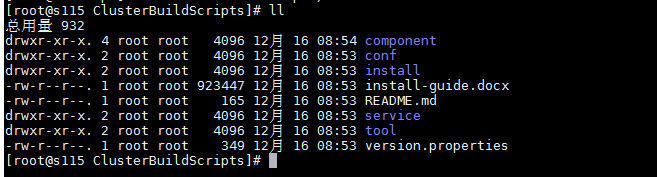
chmod –R 755 ClusterBuildScripts

从共享目录中获取组件安装包（\\192.168.1.4\集群搭建\平台安装包\component），将component文件夹同步到/opt/ClusterBuildScripts目录下。



## 1.3 ClusterBuildScripts目录结构

(1)根目录



(2)component

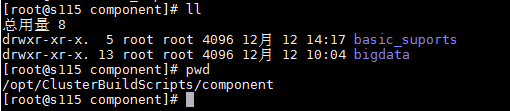
包括/bigdata下：大数据平台安装包

basic\_suports下：

expect工具RPM包

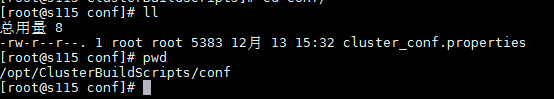
mysql数据库RPM包

dos2unix工具RPM包



(3)conf

大数据平台使用的配置文件cluster\_conf.properties



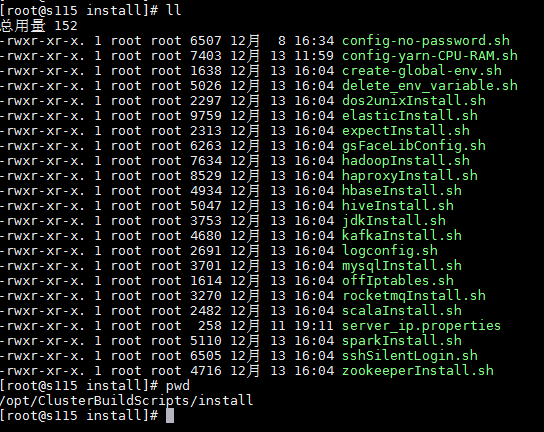
(4)install

大数据平台组件，haproxy，mysql的安装脚本

ssh无密钥登录工具脚本sshSilentLogin.sh

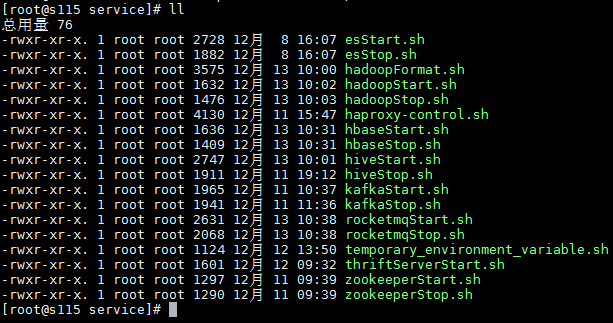
expect工具安装脚本expectInstall.sh

dos2unix工具安装脚本dos2unixInstall.sh



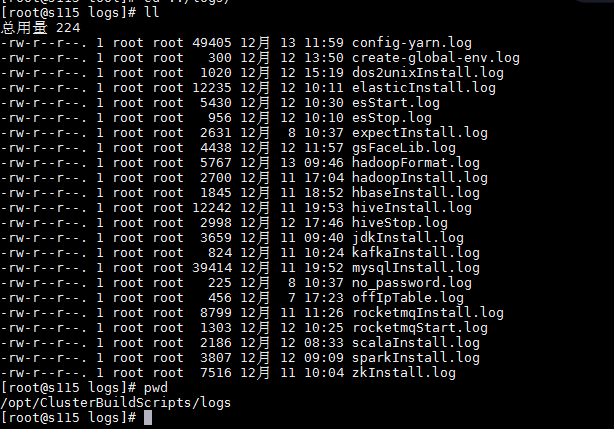
(5)service

大数据平台组件启动和停止脚本



(6)logs

集群安装日志目录

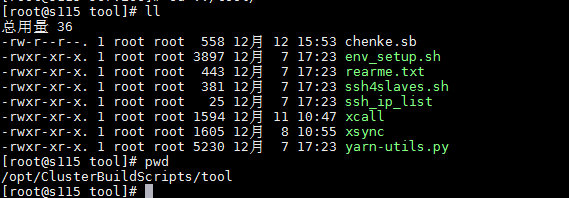


(7)tool

查看集群进程脚本xcall

集群拷贝脚本xsync

配置yarn的内存和cpu脚本yarn-utils.py



(8)version.properties大数据平台组件安装包信息

│ hadoop.tar.gz 版本2.7.2

│ hbase.tar.gz 版本1.2.6

│ hive.tar.gz 版本2.3.0

│ jdk.tar.gz 版本1.8

│ kafka.tgz 版本1.0.0

│ scala.tgz 版本2.11.8

│ spark.tgz 版本2.2.0

│ zookeeper.tar.gz 版本3.4.10

│ elastic.tar.gz 版本5.5.0

│ rocketmq.zip 版本4.1.0

│ haproxy-1.7.9.tar.gz 版本3.4.10

# 二.配置主机名和主机ip映射

## 2.1修改主机s115的/etc/hosts文件（ip和主机名根据实际情况填写）

vim /etc/hosts

写入以下内容，并保存退出(hzgc可配在任意节点下)

172.18.18.115 s115 hzgc

172.18.18.116 s116

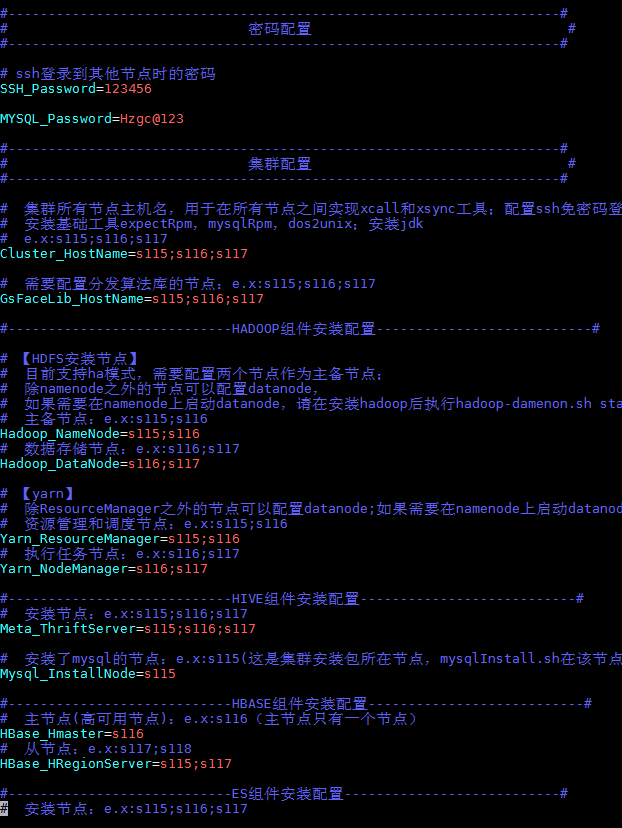
172.18.18.117 s117



**2.2根据配置文件中注释的提示信息修改配置文件**

/opt/ClusterBuildScripts/conf/cluster\_conf.properties

部分配置如下：



**2.3安装dos2unix并转换脚本执行格式**

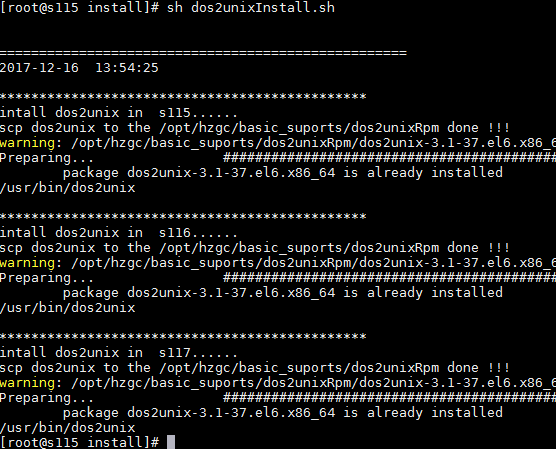
进入到/opt/ClusterBuildScripts/install目录，对dos2unixInstall.sh执行以下操作：

vim dos2unixInstall.sh打开文件

:set ff=unix把脚本从dos格式转化为unix格式

wq保存退出

执行sh dos2unixInstall.sh



转换该文件夹下所有脚本格式：

进入到/opt/ClusterBuildScripts/ conf目录，执行dos2unix \*

进入到/opt/ClusterBuildScripts/ install目录，执行dos2unix \*

进入到/opt/ClusterBuildScripts/ service目录，执行dos2unix \*

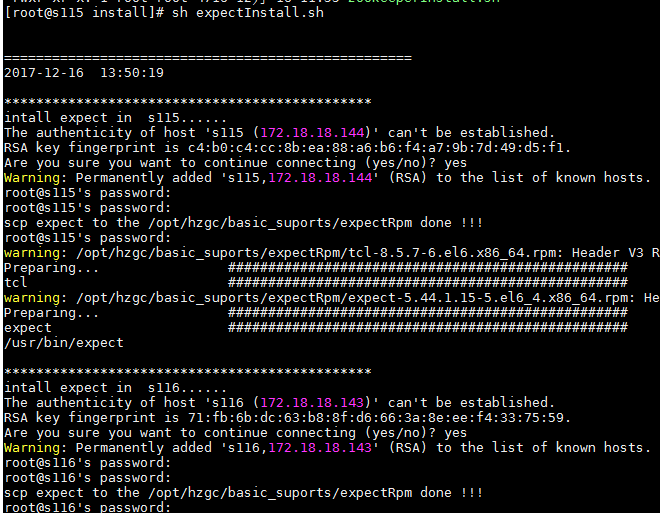
（只对需要转化格式的脚本执行转化格式操作）

**2.4安装expect包（在原集群基础上重装集群，该步骤可省略）**

进入脚本所在目录/opt/ClusterBuildScripts/install，执行sh expectInstall.sh

根据提示输入yes和密码

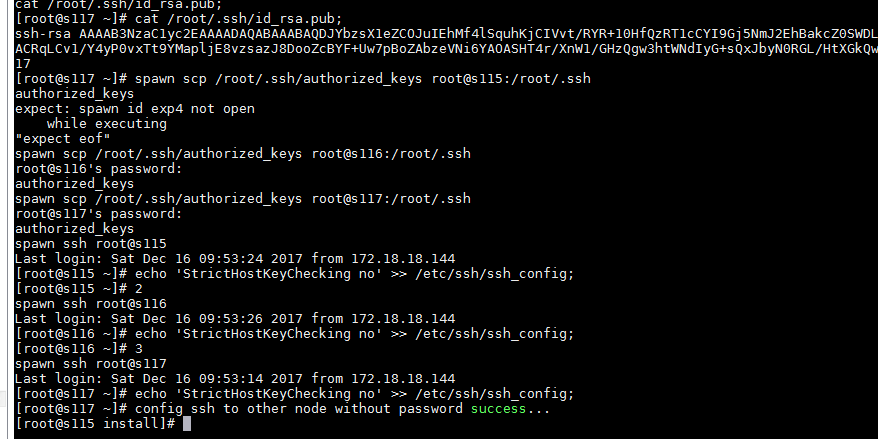
部分执行结果如下：



## 2.5配置ssh免密码登录（在原集群基础上重装集群，该步骤可省略）

进入到/opt/ClusterBuildScripts/install目录，执行sh sshSilentLogin.sh

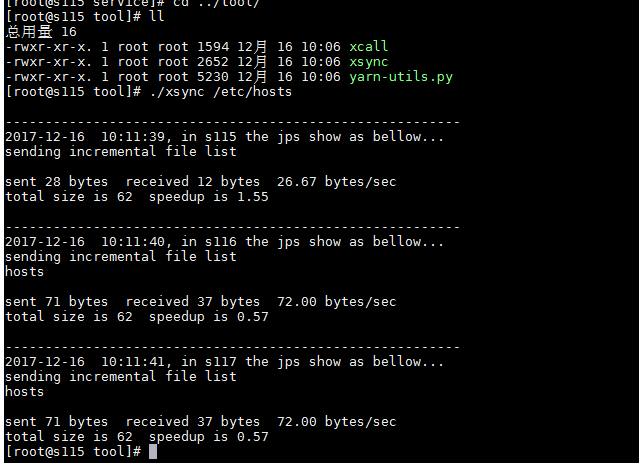
期间不要操作机器，等待配置完成。



**2.6分发hosts文件（在原集群基础上重装集群，该步骤可省略）**

进入/opt/ClusterBuildScripts/tool目录

./xsync /etc/hosts



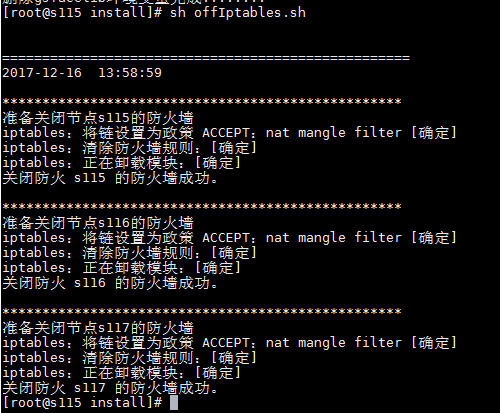
**2.7删除系统环境变量(如果是新建集群可不执行此步骤)**

进入到/opt/ClusterBuildScripts/install目录，执行sh delete\_env\_variable.sh



**2.8关闭防火墙（在原集群基础上重装集群，该步骤可省略）**

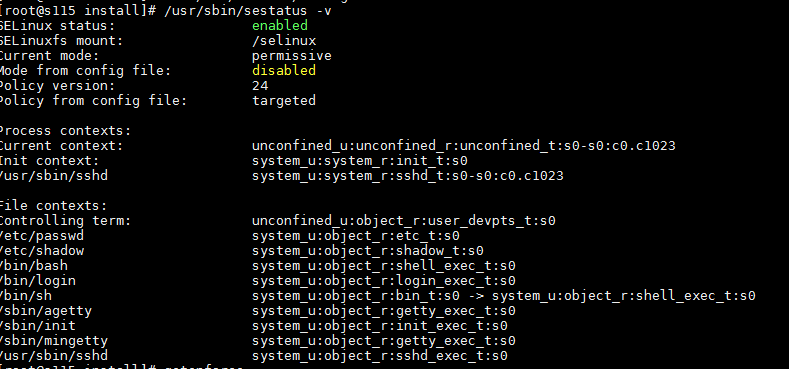
进入到/opt/ClusterBuildScripts/install目录，执行sh offIptables.sh



# 三.安装mysql（在安装包所在节点上安装）

**3.1安装mysql前检查SELinux status状态**

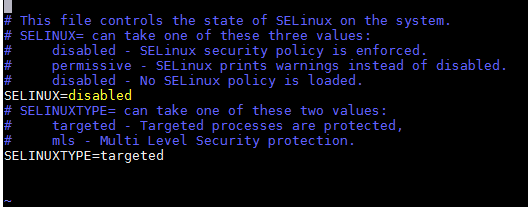
执行/usr/sbin/sestatus –v



若SELinux status为enabled，需要修改将SELinux status为disabled。执行如下命令：

vim /etc/selinux/config

D:\我的文档\Tencent Files\407932000\Image\C2C\]3UW8T$0[JXE{DLM%OV}_B6.png



修改完后需要重启该节点：reboot

重启后查看：/usr/sbin/sestatus –v

D:\我的文档\Tencent Files\407932000\Image\C2C\W258H%T_C$K~F2SV{YLMWRG.png

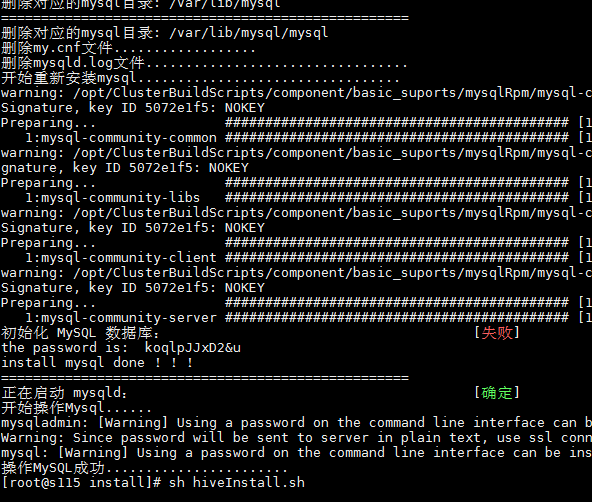
SELinux status为disabled了再安装mysql。

**3.2安装mysql**

脚本所在目录/opt/ClusterBuildScripts/install

执行mysqlInstall.sh

脚本执行结果（涂红处显示失败没有关系）

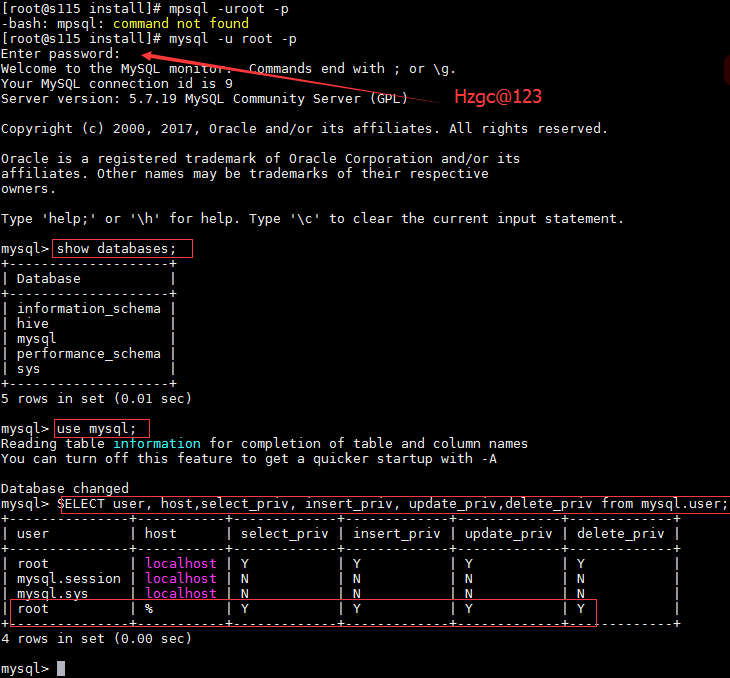


验证是否安装成功：

show databases;

usemysql;

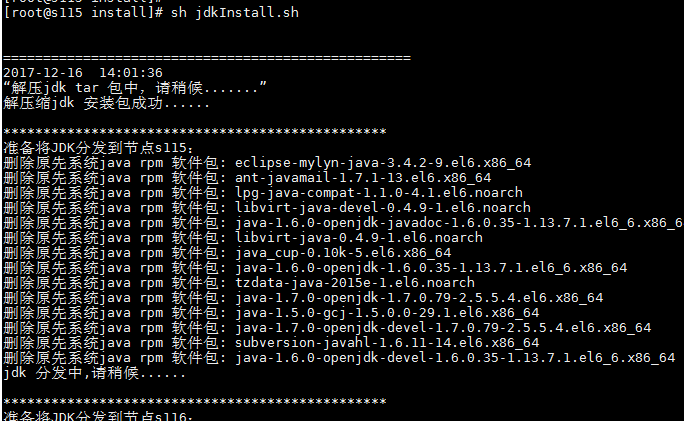
SELECT user, host,select\_priv, insert\_priv, update\_priv,delete\_priv from mysql.user;



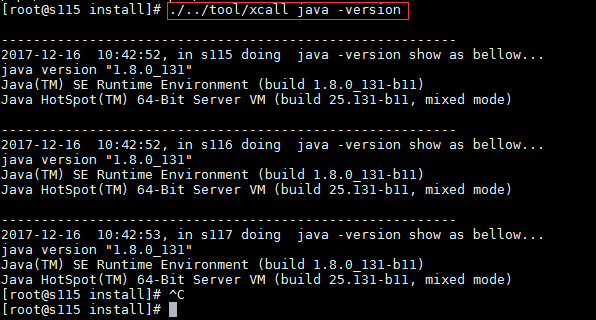
# 四.安装jdk

进入到/opt/ClusterBuildScripts/install目录，执行sh jdkInstall.sh

部分执行结果如下：



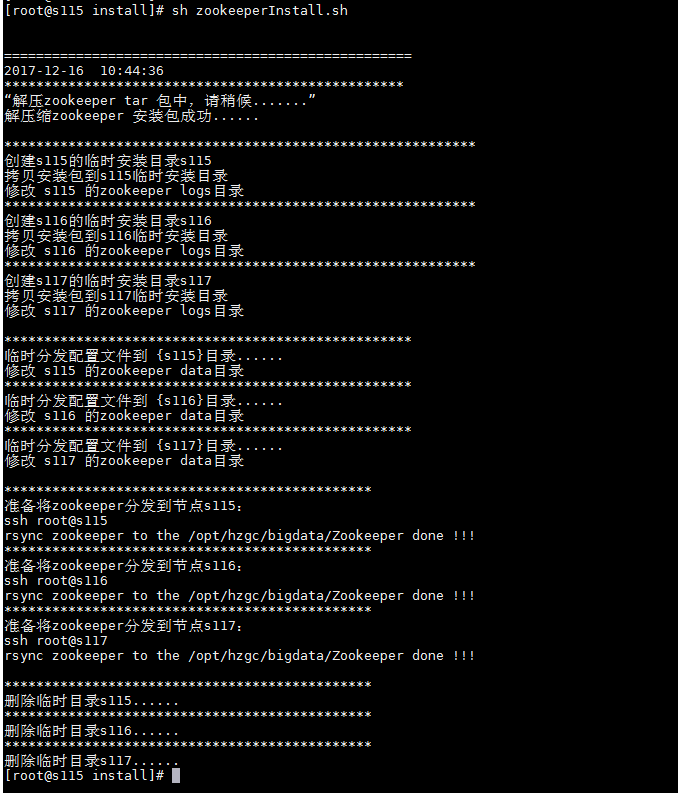
./../tool/xcall java –version



# 五.安装zookeeper

进入到/opt/ClusterBuildScripts/install目录，执行sh zookeeperInstall.sh

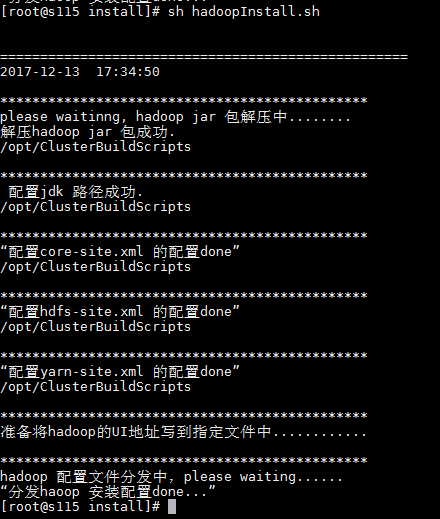
部分执行结果如下：



# 六.安装hadoop

进入到/opt/ClusterBuildScripts/install目录，执行sh hadoopInstall.sh

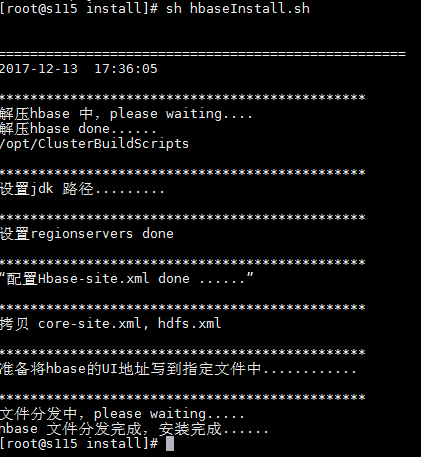
执行结果如下：



# 七.安装hbase

进入到/opt/ClusterBuildScripts/install目录，执行sh hbaseInstall.sh

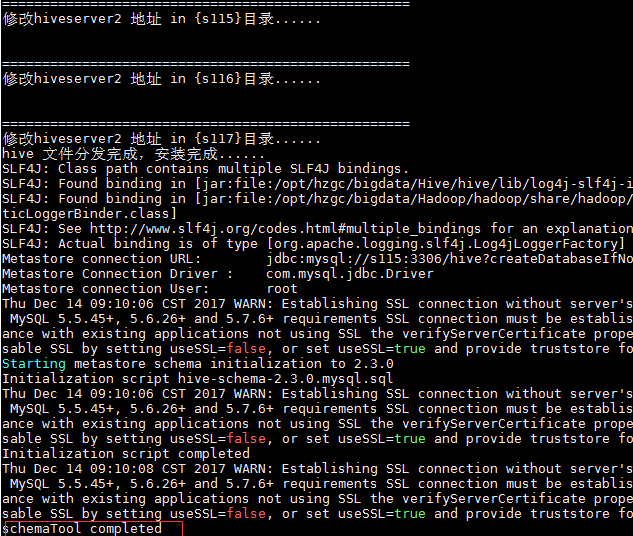
部分执行结果如下：



# 八.安装hive

进入到/opt/ClusterBuildScripts/install目录，执行sh hiveInstall.sh

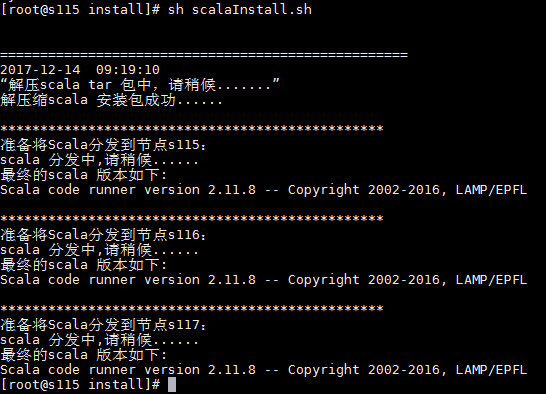
部分执行结果如下：



# 九.安装scala

进入到/opt/ClusterBuildScripts/install目录，执行sh scalaInstall.sh

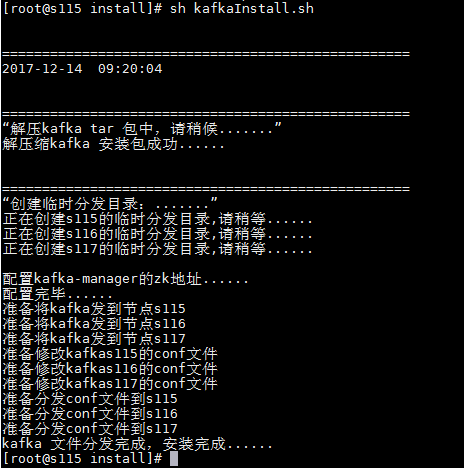
执行结果如下：



# 十.安装kafka

进入到/opt/ClusterBuildScripts/install目录，执行sh kafkaInstall.sh

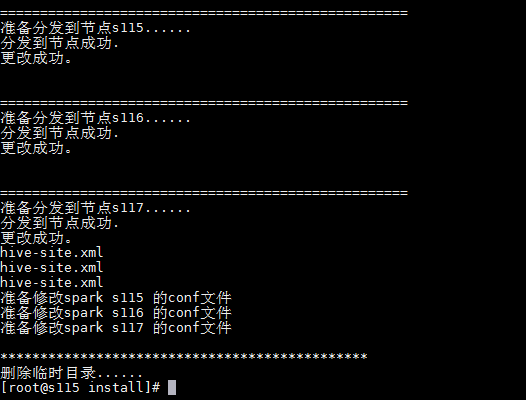
执行结果如下：



# 十一.安装spark

进入到/opt/ClusterBuildScripts/install目录，执行sh sparkInstall.sh

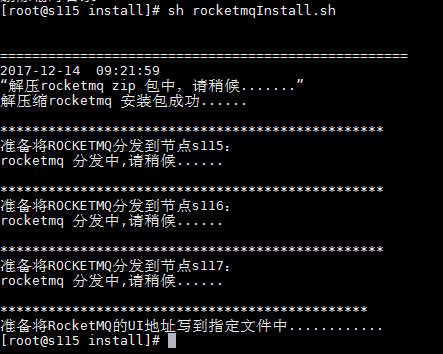
部分执行结果如下：



# 十二.安装roketmq

进入到/opt/ClusterBuildScripts/install目录，执行shrocketmqInstall.sh

执行结果如下：



# 十三.安装Haproxy

进入到/opt/ClusterBuildScripts/install目录，执行sh haproxyInstall.sh

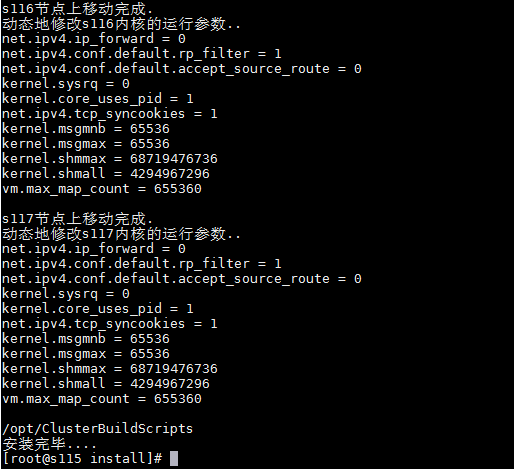
部分执行结果如下：



# 十四.安装ElasticSearch

进入到/opt/ClusterBuildScripts/install目录，执行sh elasticInstall.sh

部分执行结果如下：



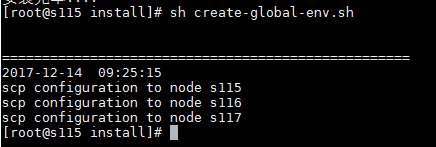
# 十五，安装Phoenix 客户端

进入到/opt/ClusterBuildScripts/install目录，执行sh elasticInstall.sh

# 十六.拷贝环境变量以及相关环境变量到各个目录

进入到/opt/ClusterBuildScripts/install目录，执行sh create-global-env.sh

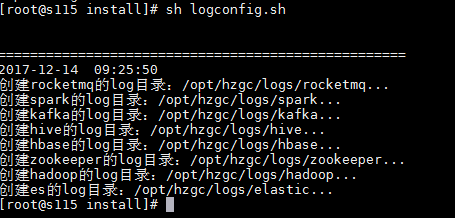
执行结果如下：



# 十七.配置集群组件日志

进入到/opt/ClusterBuildScripts/install目录，执行sh logconfig.sh(需要在执行es启动之前执行)

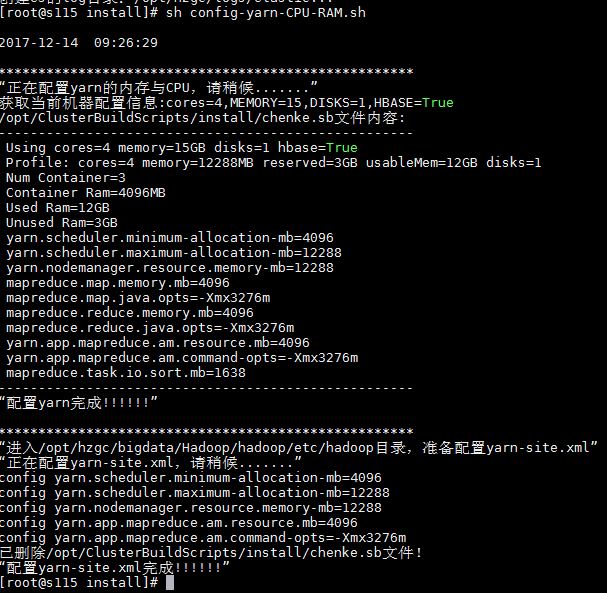
执行结果如下：



# 十八.配置yarn的内存和cpu

进入到/opt/ClusterBuildScripts/install目录，执行sh config-yarn-CPU-RAM.sh

执行结果如下：

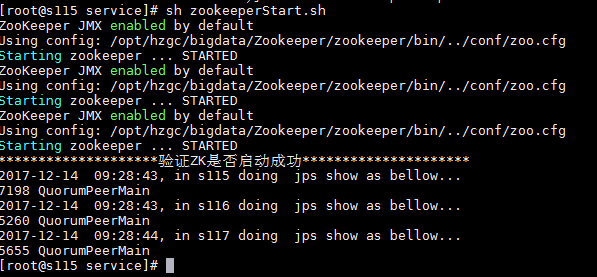


说明：yarn-utils.py这是一个根据输入的参数，自动计算yarn应该如何分配内存和cpu的脚本，算出结果后，将yarn-utils.py脚本所得结果配入yarn的配置文件。执行脚本config-yarn-CPU-RAM.sh就已经涵盖了整个过程，无需手动配置。

# 十九.启动集群各组件

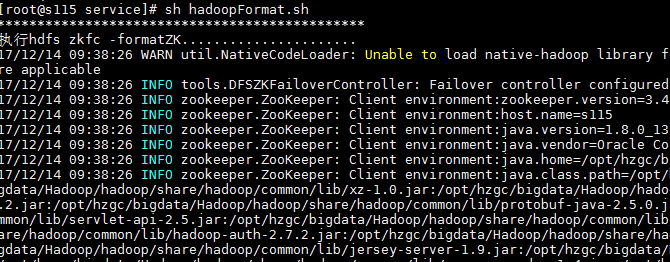
**19.1启动zookeeper**

进入到/opt/ClusterBuildScripts/service目录下执行sh zookeeperStart.sh

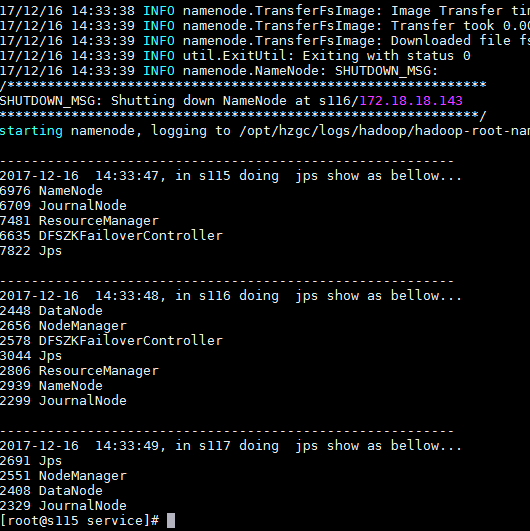


**19.2启动hadoop**

进入到/opt/ClusterBuildScripts/service目录下执行sh hadoopFormat.sh



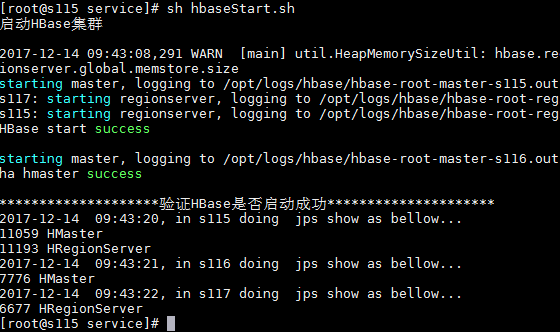
脚本部分执行结果



注意：此处不用执行hadoopStart.sh脚本，hadoopStart.sh用于停止hadoop后再启。

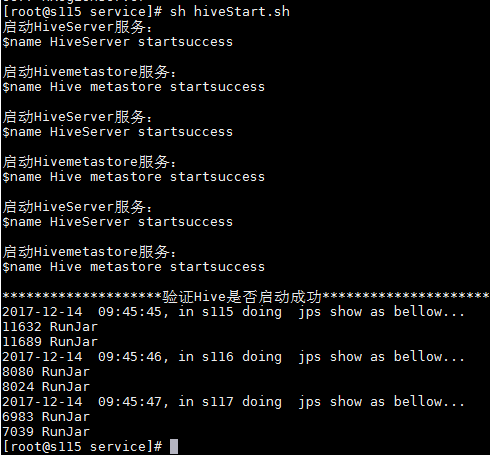
**19.3启动hbase**

进入到/opt/ClusterBuildScripts/service目录下执行sh hbaseStart.sh



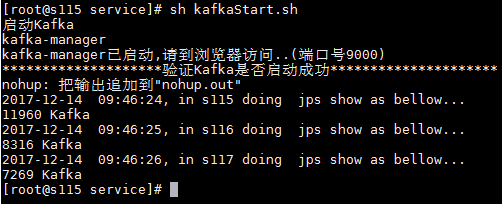
**19.4启动hive（启动后每个节点有两个RunJar进程）**

进入到/opt/ClusterBuildScripts/service目录下执行sh hiveStart.sh



**19.5启动kafka**

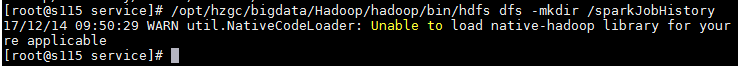
进入到/opt/ClusterBuildScripts/service目录下执行sh kafkaStart.sh



**19.6使用spark jdbc**

第一次使用spark请先在hdfs创建sparkJobHistory

/opt/hzgc/bigdata/Hadoop/hadoop/bin/hdfs dfs -mkdir /sparkJobHistory



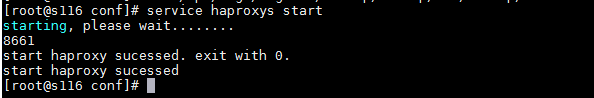
**19.7启动thriftserver服务**

进入到/opt/ClusterBuildScripts/service目录下执行sh thriftServerStart.sh

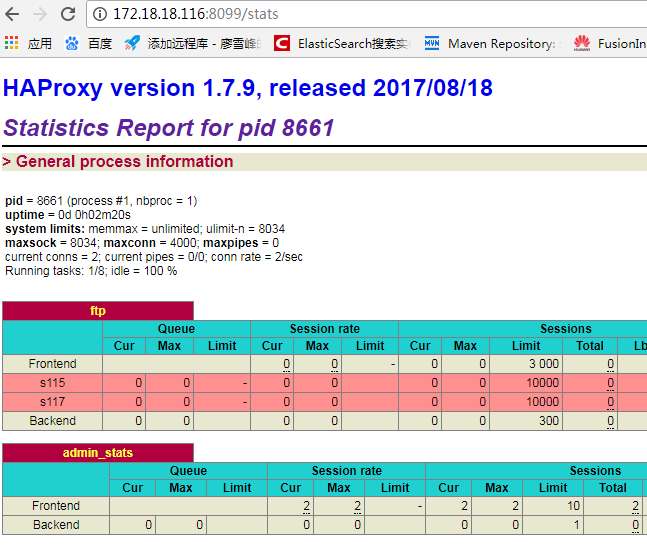
使用方法：sh thriftServerStart.sh1G 1G 1 1 1（请根据实际情况配置），这五个参数分别表示：driver\_memory；executor\_memory；driver\_cores；executor\_cores；num\_executors

**19.8启动Haproxy**

切换至Haproxy代理节点（如s116），（在任意目录下均可）执行service haproxys start

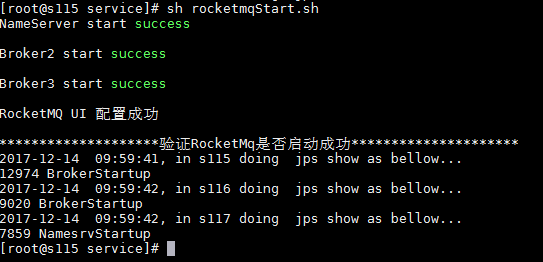


在浏览器上验证是否启动：



**19.9启动rocketmq**

进入到/opt/ClusterBuildScripts/service目录下执行sh rocketmqStart.sh



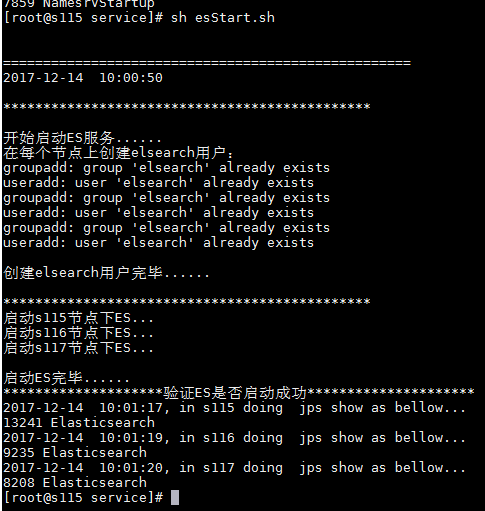
在浏览器上验证是否启动：

ip:8083/#/cluster



**19.10启动elasticsearch**

进入到/opt/ClusterBuildScripts/service目录下执行sh esStart.sh



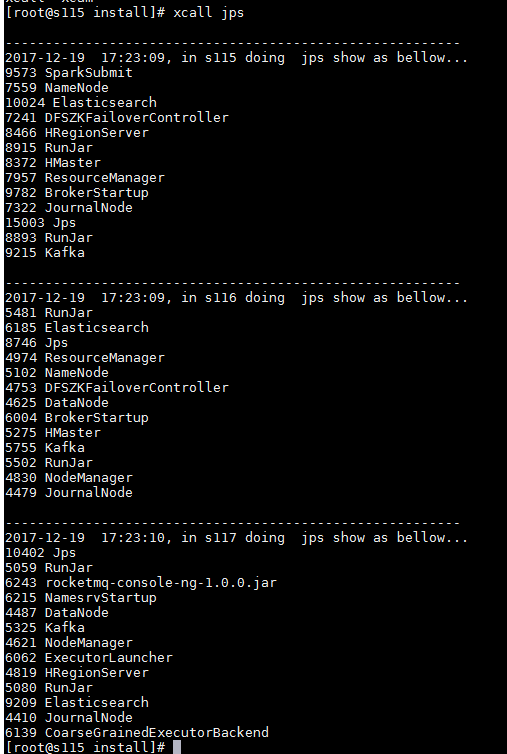
# 二十.服务验证

**20.1 查看每台主机所起的服务**

在任意目录下，执行：

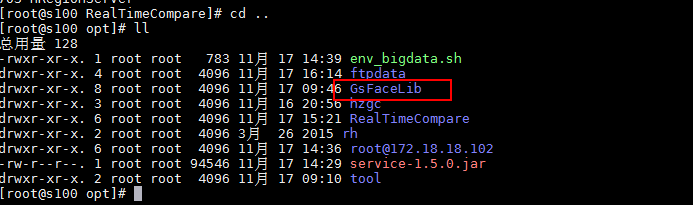
source /opt/hzgc/env\_bigdata.sh

xcall jps



# 二十一. 配置算法库

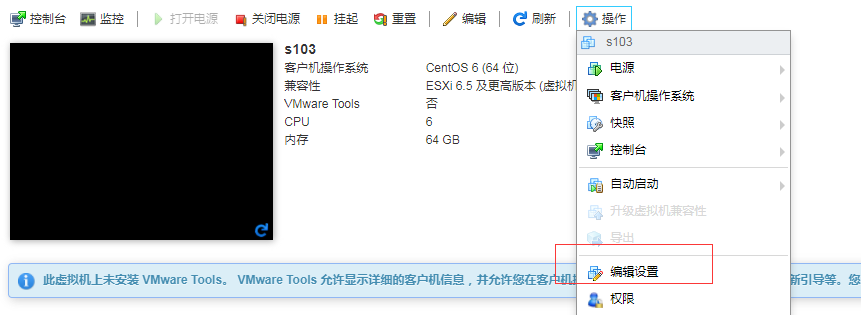
## 21.1.上传GsFaceLib到主节点/opt目录下



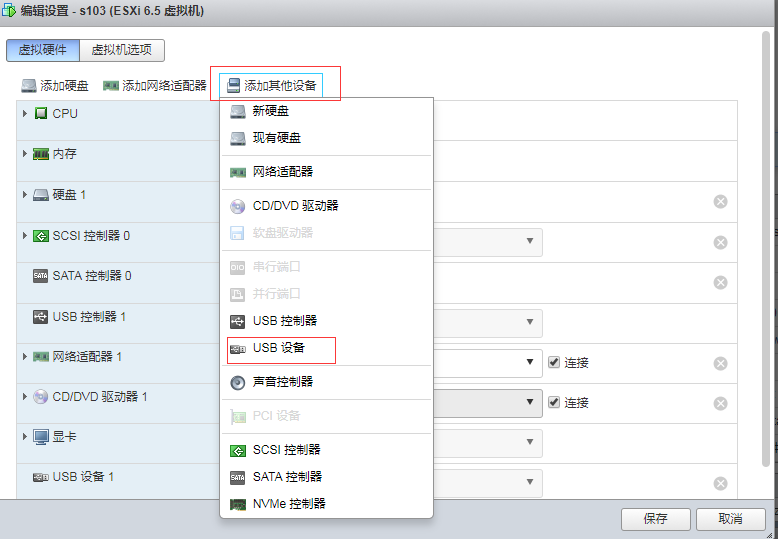
## 21.2.添加加密狗

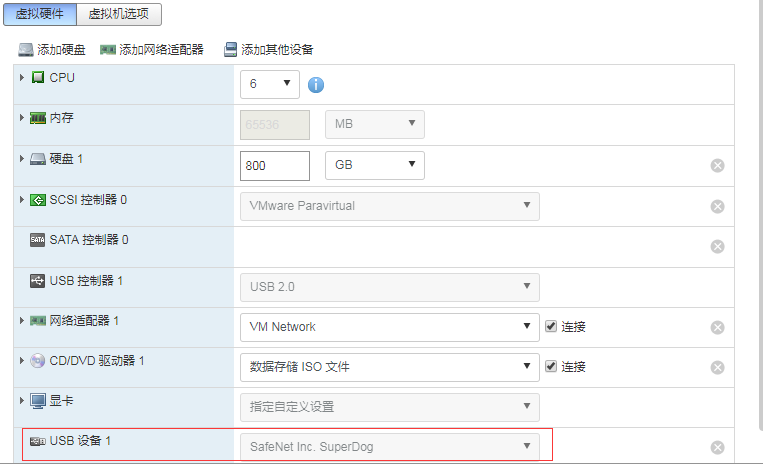
在对应的服务器中添加加密狗

登录对应服务器的vmware，编辑对应虚拟机设置：



添加USB控制器，后选择添加加密狗：





## 21.3.配置算法库

### 21.3.1.配置/etc/profile并删除/opt/GsFaceLib下sn.ini文件

进入到/opt/ClusterBuildScripts/install目录，执行sh gsFaceLibConfig.sh

### 21.3.2.进入/opt/GsFaceLib/example目录下

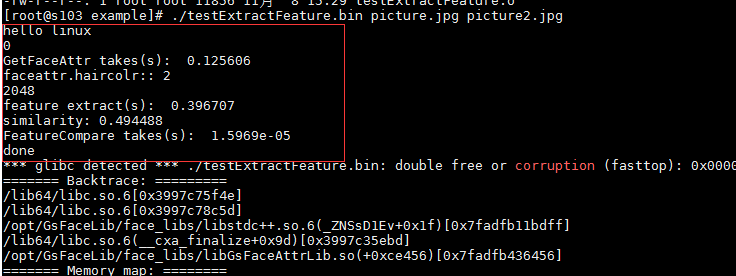
执行./testExtractFeature.bin picture.jpg picture2.jpg（两个图片没有先后顺序）

### 21.3.4.生成新的sn.ini文件

对新产生的sn.ini文件内容发给杨忠桃转成相应的密钥后保存到sn.ini;

### 21.3.5. 继续进入/opt/GsFaceLib/example执行

./testExtractFeature.bin picture.jpg picture2.jpg



# 二十二.重装集群

删除：/opt/hzgc下的所有文件；

建议删除：根目录下的/tmp。

再执行步骤一-二十二。

在原集群基础上重装集群，其中某些步骤可省略（已在对应的步骤中标注）。

# 二十三.常见安装问题

**24.1 hbase错误**

File /hbase/.tmp/hbase.version could only be replicated to 0 nodes instead of minReplica

问题原因：多次执行./hdfsnamenode-format 造成，解决办法手动在各个节点中删除tmp文件夹下data目录内容然后从新启动hdfs服务就可以启动了

**24.2 启动thriftserver**

Exception in thread "main" java.io.FileNotFoundException: File does not exist:

hdfs://hzgc/sparkJobHistory

问题原因：启动前请现在hdfs上创建sparkJobHistory目录，

/opt/client/Hadoop/hadoop/bin/hdfsdfs -mkdir /sparkJobHistory

**24.3 mysql没有被删除**

删除mysql时有好几个步骤：

①卸载rpm包，②删除mysql所有目录，③停止服务，测试时需要mysql+hive一起，看看初始化hive是否成功

(1)mysql没有卸载干净会导致hive初始化schema出错

(2)删除mysql的时候要删除/var/log/mysql.log文件，否则会查出多个密码

**问题：**

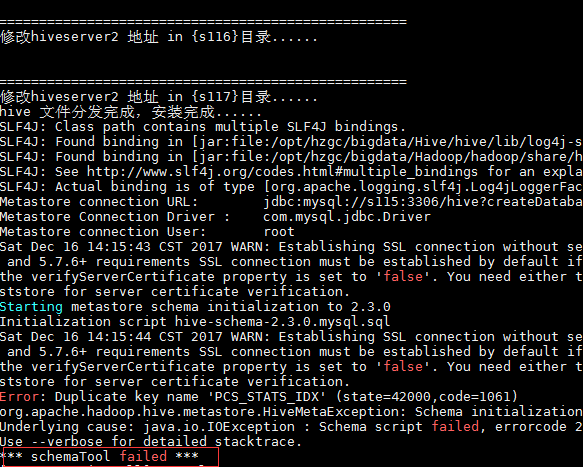
安装Hive时出现：

Error: Duplicate key name 'PCS\_STATS\_IDX' (state=42000,code=1061)

\*\*\* schemaTool failed \*\*\*

**解决方法：**在hive的安装脚本中，初始化元数据步骤${HIVE\_HOME}/bin/schematool -initSchema -dbTypemysql只需在安装了mysql的节点上执行，而不是ssh到每个节点上执行。

**24.4 连续两次执行hiveInstall脚本报错**



**解决方法：**删除mysql中hive数据库，再执行安装脚本