# 高创集群&项目部署指导书

# 分布式集群搭建

## 机器准备

准备机器（以3 台机器为例）

|  |  |
| --- | --- |
| 主机名（hostname） | IP |
| s181 | 172.18.18.181 |
| s182 | 172.18.18.182 |
| s183 | 172.18.18.183 |

**注意**：本指导书中所有与ip相关的配置均为此3台机器，部署时请根据实际情况配置

## 准备集群部署脚本及安装包

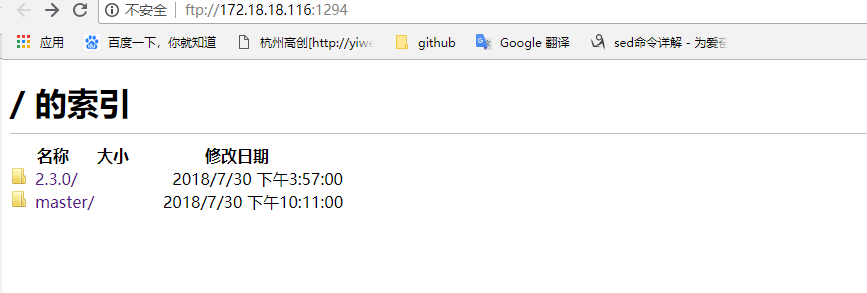
### 使用GoSunBigData.tar.gz安装

登录服务器/虚拟机，从出包目录获取安装包

当前有2个分支（master和2.3.0），文件夹名为出包的时间

根据出的版本和时间自取

浏览器登录出包目录，地址：<ftp://172.18.18.116:1294/>





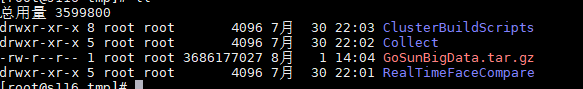
到/opt下执行取包命令

取包命令：wget <ftp://172.18.18.116:1294/分支名/出包时间/>GoSunBigData.tar.gz

解压tar包，执行解压命令

解压命令：tar -zxvf GoSunBigData.tar.gz

解压后目录结构如下：



## 准备集群安装环境

### 配置主机名和主机ip映射

**注意：系统自带的hosts配置不要删除！！**

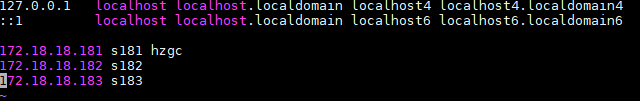
执行命令：vim /etc/hosts

在文件结尾添加以下内容，并保存退出(hzgc可配在任意节点下)

172.18.18.181 s181 hzgc

172.18.18.182 s182

172.18.18.183 s183



### 修改集群安装配置文件

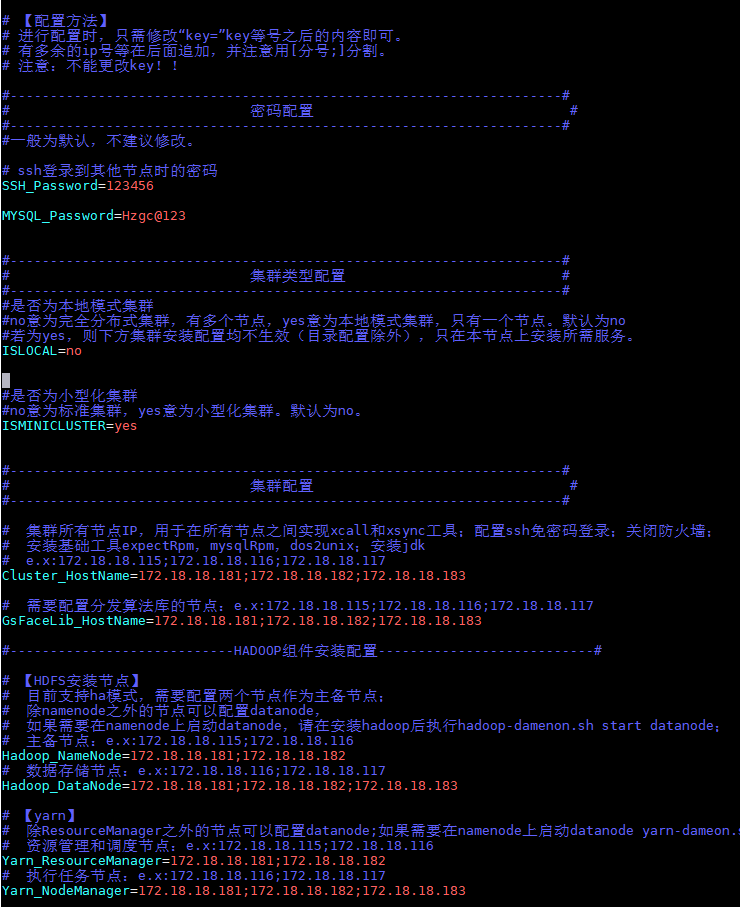
进入/ClusterBuildScripts/conf目录，修改cluster\_conf.properties

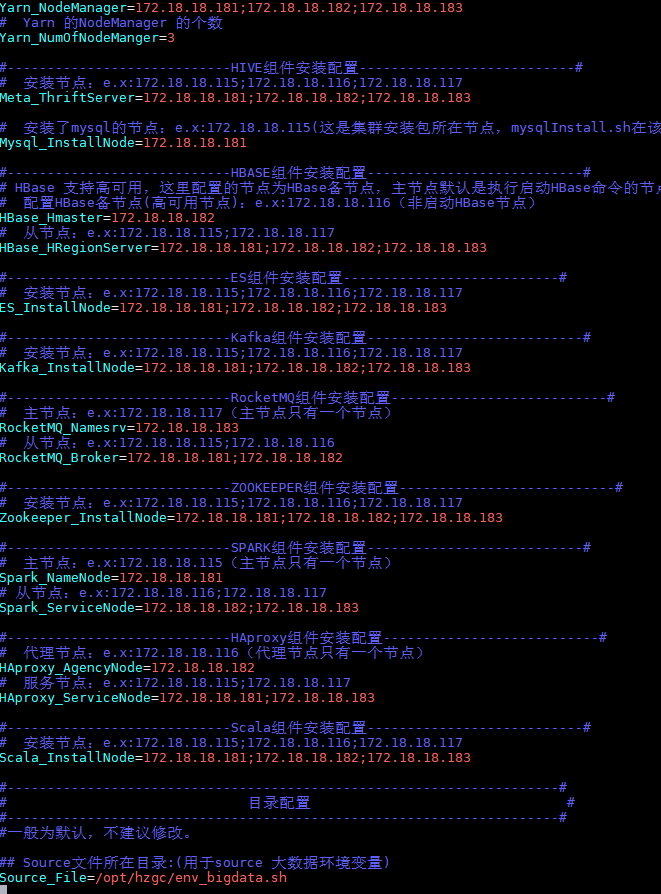
**注意：分布式集群， ISLOCAL选项必须为no**

**本地模式集群，ISLOCAL选项必须为yes**

**虚拟机集群，根据虚拟机配置决定ISMINICLUSTER选项是否为yes**

vim cluster\_conf.properties

部分配置如下：



### 环境部署

进入install目录

cd /opt/ClusterBuildScripts/install

执行环境部署脚本

sh installEnv.sh

installEnv.sh包含：

1. 在当前节点上安装sshpass，dos2unix工具
2. 所有节点上安装expect工具
3. 配置所有节点（集群节点和ftp节点）间的免密登录
4. 分发/etc/hosts文件
5. 删除原有环境变量
6. 关闭所有节点的防火墙

**注意：**若是脚本安装出错，根据报错情况解决后重新执行脚本即可

## 集群组件安装

进入install目录

cd /opt/ClusterBuildScripts/install

执行组件安装脚本

sh installCompent.sh

installCompent.sh包含：

1. 检查selinux状态，若为enabled，会提示用户需改为disabled，并中断脚本执行

重启机器后再执行installCompent.sh即可

1. 安装并配置mysql
2. 安装并配置jdk
3. 安装并配置zookeeper，hadoop，hbase，hive，scala，kafka，spark，rocketmq，haproxy，elastic，azkaban，kibana，phoenix
4. 配置环境变量
5. 配置组件的日志目录
6. 根据集群类型和机器配置调整yarn的相关配置项

**注意：**若是安装出错请根据报错信息，解决后重新执行命令

## 启动停止集群服务

进入服务脚本目录，执行对应服务的启动停止脚本

### 启动集群服务

#### 启动zookeeper

cd /opt/ClusterBuildScripts/service

sh zookeeperStart.sh

脚本输出如下：



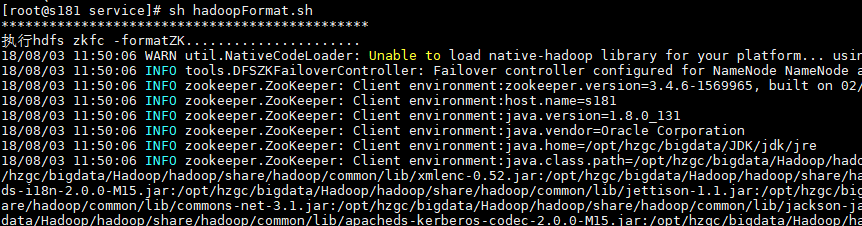
#### 启动hadoop

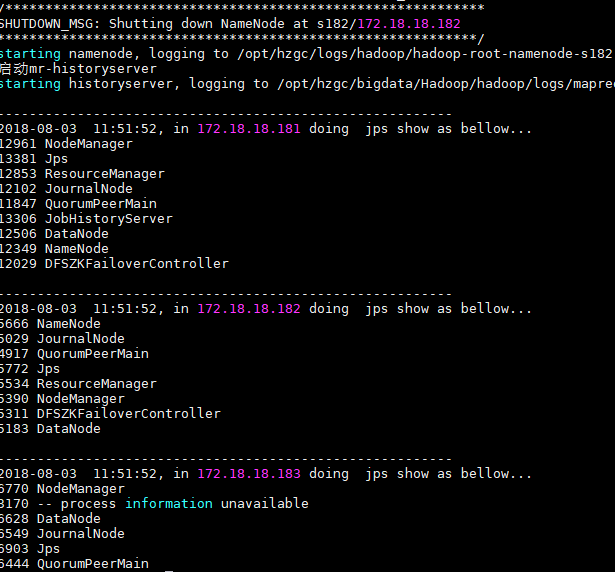
cd /opt/ClusterBuildScripts/service

第一次启动时执行hadoopFormat.sh，停止后启动执行hadoopStart.sh

sh hadoopFormat.sh

脚本输出如下：



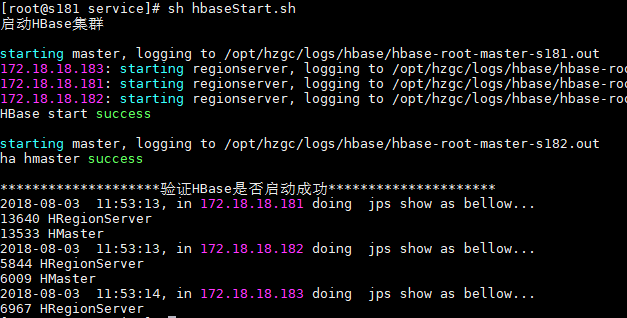


#### 启动hbase

cd /opt/ClusterBuildScripts/service

sh hbaseStart.sh

脚本输出如下：

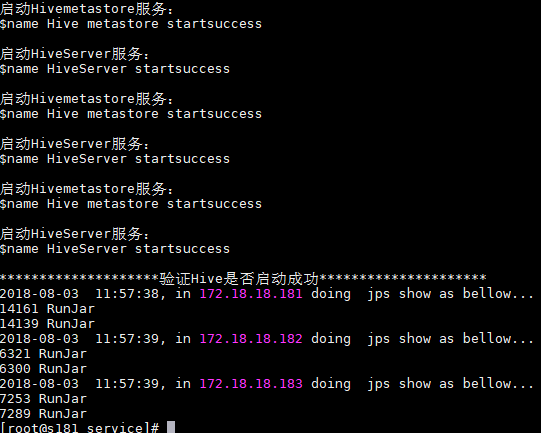


#### 启动hive

cd /opt/ClusterBuildScripts/service

sh hiveStart.sh

脚本输出如下：

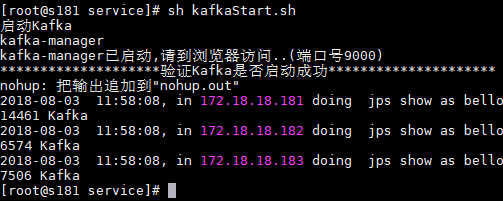


#### 启动kafka

cd /opt/ClusterBuildScripts/service

sh kafkaStart.sh

脚本输出如下：



#### 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

命令输出如下：



#### 启动thriftserver服务

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

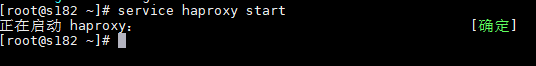
使用方法：sh thriftServerStart.sh1G 1G 1 1 1（请根据实际情况配置），这五个参数分别表示：driver\_memory；executor\_memory；driver\_cores；executor\_cores；num\_executors

#### 启动Haproxy

切换至Haproxy代理节点（一般为第二个节点，如s182），（在任意目录下均可）

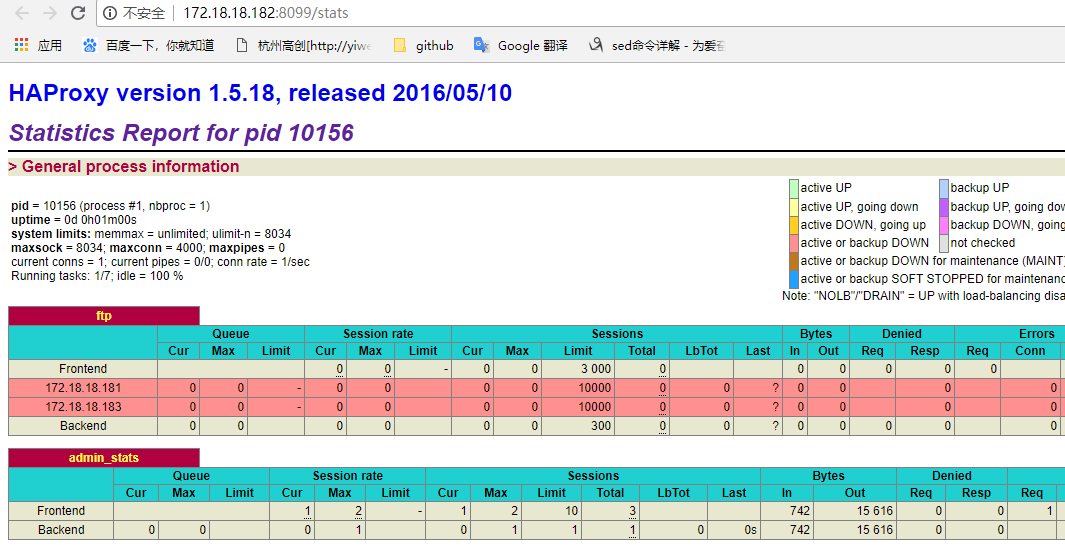
service haproxy start

输出如下：



在浏览器上验证是否启动：

172.18.18.106:8099/stats

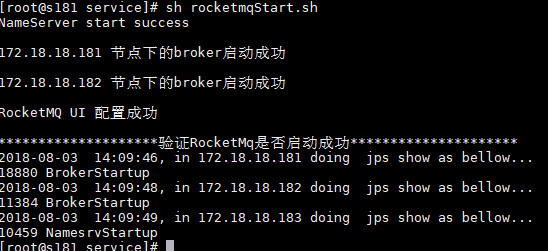


#### 启动rocketmq

cd /opt/ClusterBuildScripts/service

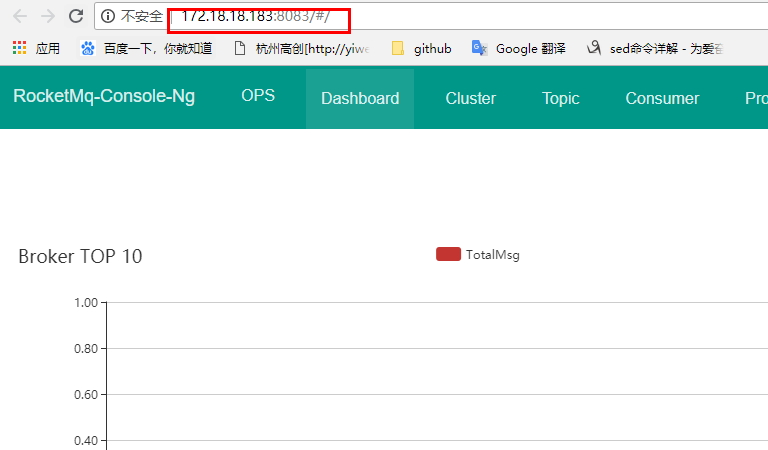
sh rocketmqStart.sh

脚本输出如下：



在浏览器上验证是否启动：

172.18.18.183:8083

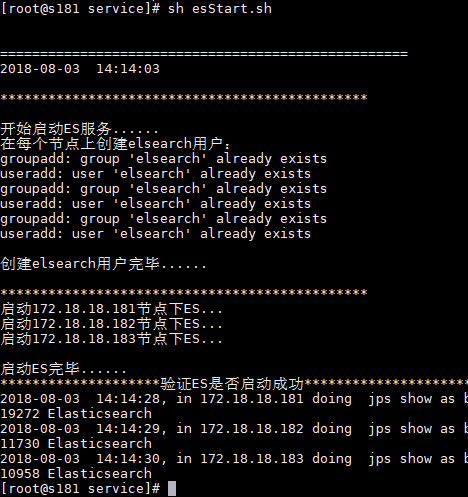


#### 启动elasticsearch

cd /opt/ClusterBuildScripts/service

sh esStart.sh

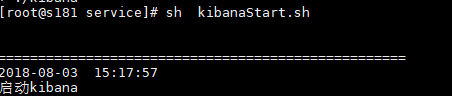
脚本输出如下：



#### 启动kibana

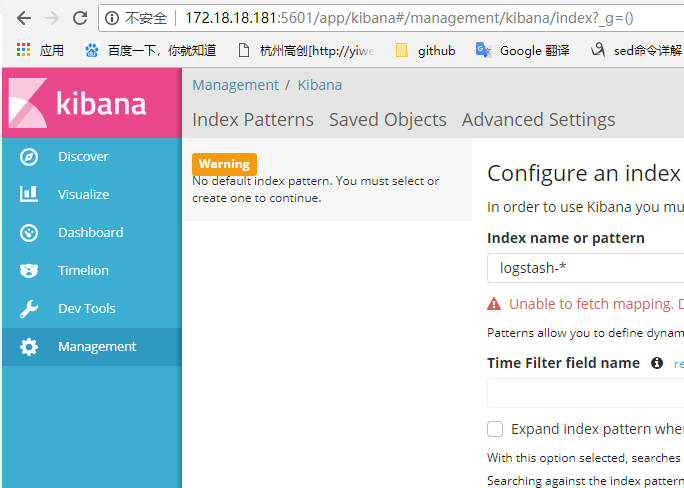
cd /opt/ClusterBuildScripts/service

sh kibanaStart.sh



在浏览器中验证是否启动成功：

<http://172.18.18.181:5601/>

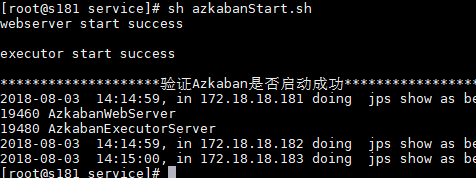


#### 启动Azkaban(在安装mysql所在节点启动)

cd /opt/ClusterBuildScripts/service

sh azkabanStart.sh

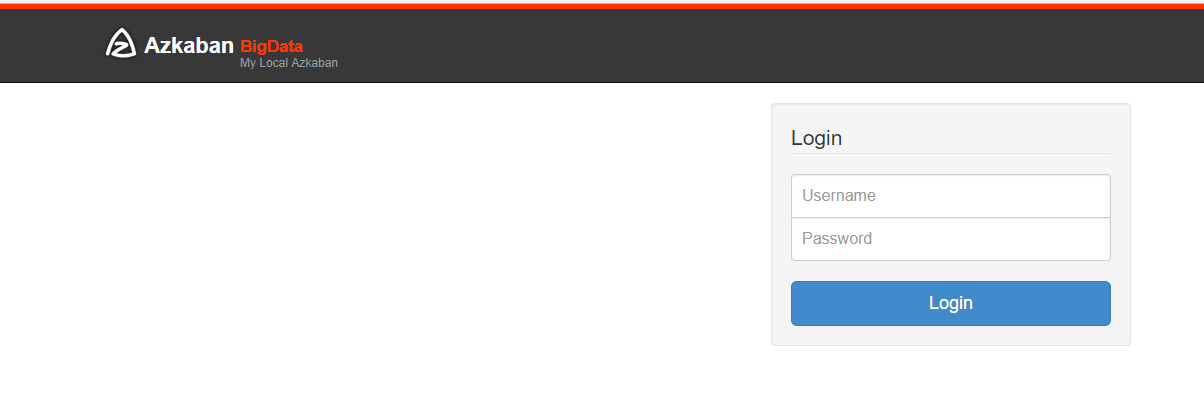
脚本输出如下：



在浏览器中验证是否启动成功：

[https://172.18.18.181:8443/stats](https://172.18.18.105:8443/stats)

用户名：admin 密码：admin

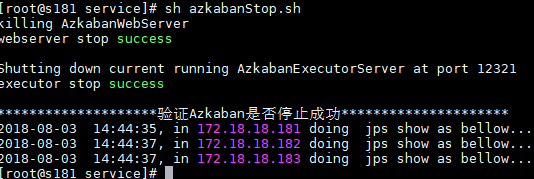


### 停止集群服务

#### 停止azkaban

cd /opt/ClusterBuildScripts/service

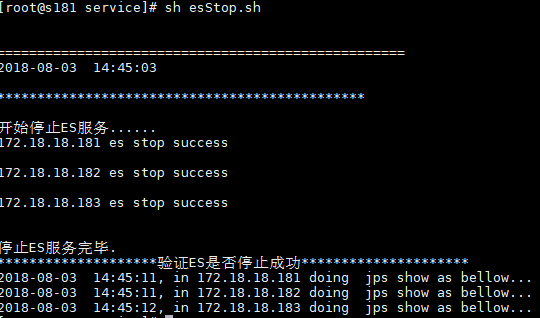
sh azkabanStop.sh



#### 停止elasticsearch

cd /opt/ClusterBuildScripts/service

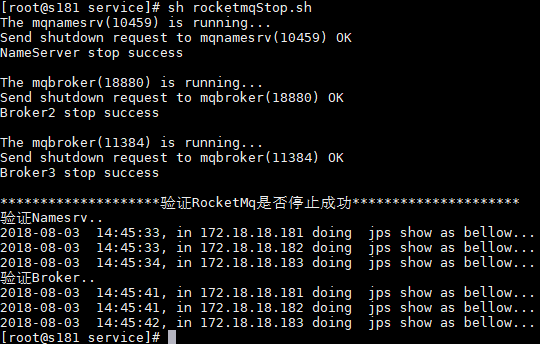
sh esStop.sh



#### 停止rocketmq

cd /opt/ClusterBuildScripts/service

sh rocketmqStop.sh



#### 停止kafka

cd /opt/ClusterBuildScripts/service

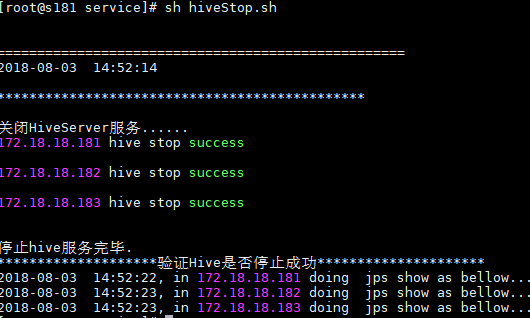
sh kafkaStop.sh



#### 停止hive

cd /opt/ClusterBuildScripts/service

sh hiveStop.sh



#### 停止hbase

cd /opt/ClusterBuildScripts/service

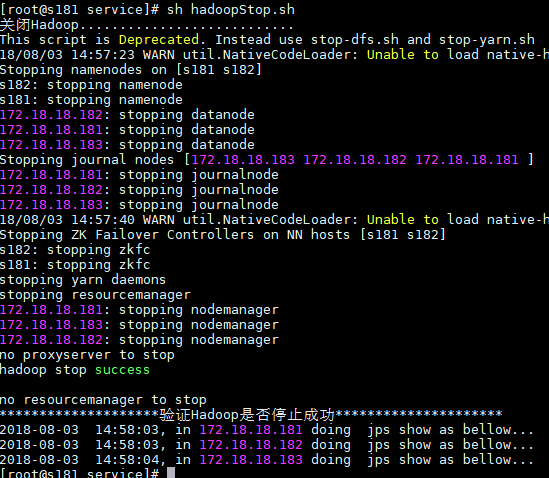
sh hbaseStop.sh



#### 停止hadoop

cd /opt/ClusterBuildScripts/service

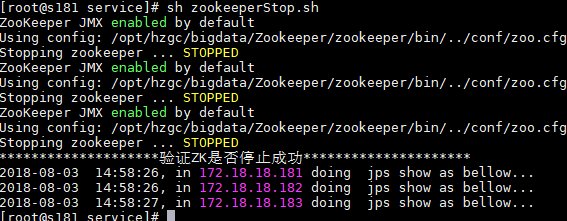
sh hadoopStop.sh



#### 停止zookeeper

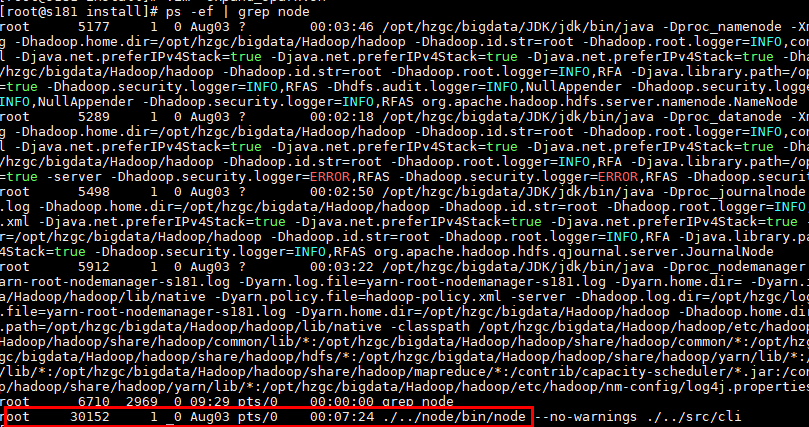
cd /opt/ClusterBuildScripts/service

sh zookeeperStop.sh



#### 停止kibana

ps -ef | grep node



Kill -9 pid(进程号)

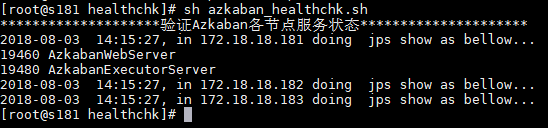
## 集群服务检查

进入服务检查目录，执行对应服务的进程检查脚本

### Azkaban服务检查

cd /opt/ClusterBuildScripts/healthchk

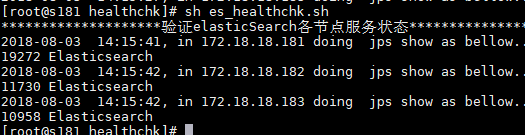
sh azkaban\_healthchk.sh



### Elastic服务检查

cd /opt/ClusterBuildScripts/healthchk

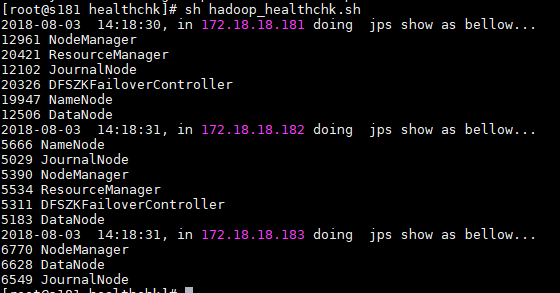
sh es\_healthchk.sh



### Hadoop服务检查

cd /opt/ClusterBuildScripts/healthchk

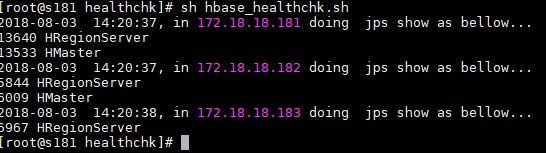
sh hadoop\_healthchk.sh



### HBase服务检查

cd /opt/ClusterBuildScripts/healthchk

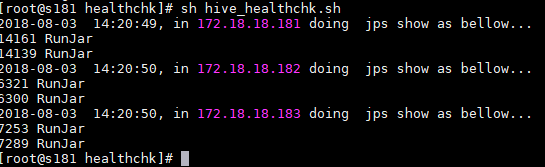
sh hbase\_healthchk.sh



### Hive服务检查

cd /opt/ClusterBuildScripts/healthchk

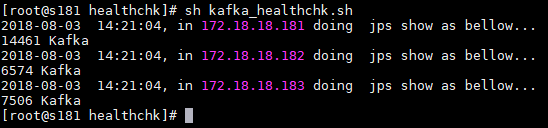
sh hive\_healthchk.sh



### Kafka服务检查

cd /opt/ClusterBuildScripts/healthchk

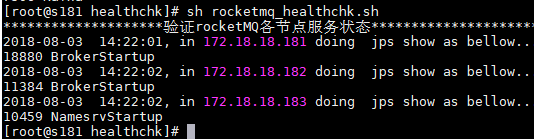
sh kafka\_healthchk.sh



### Rocketmq服务检查

cd /opt/ClusterBuildScripts/healthchk

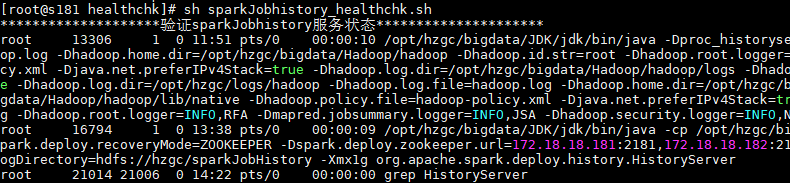
sh rocketmq\_healthchk.sh



### Sparkjobhistory服务检查

cd /opt/ClusterBuildScripts/healthchk

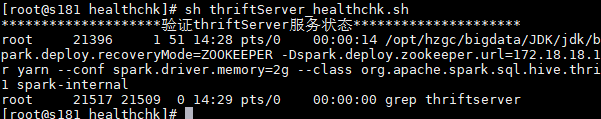
sh sparkJobhistory\_healthchk.sh



### Thriftserver服务检查

cd /opt/ClusterBuildScripts/healthchk

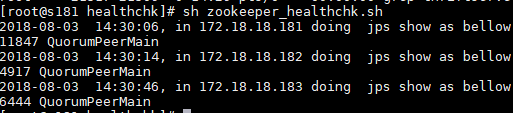
sh thriftServer\_healthchk.sh



### Zookeeper服务检查

cd /opt/ClusterBuildScripts/healthchk

sh zookeeper\_healthchk.sh



## 集群扩展

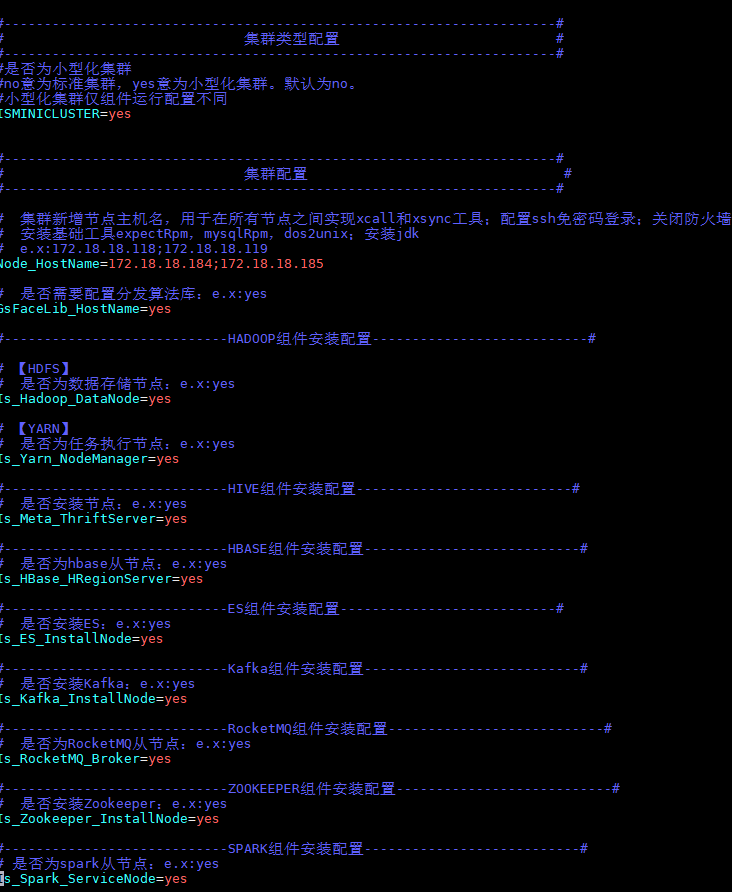
进入集群扩展模块目录，修改配置文件，依次执行扩展脚本，服务启动脚本（部分服务不支持动态扩展，需重启整个集群后才生效）。

### 修改扩展配置文件

cd /opt/ClusterBuildScripts/expand/conf

vim expand\_conf.properties

将需要安装的组件的key改为yes即可，不需要的改为no，详细说明见配置文件注释



### 配置主机名和主机ip映射

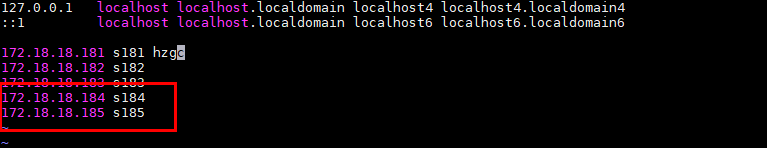
修改/etc/hosts文件

执行命令：vim /etc/hosts

在文件结尾添加以下内容，并保存退出(hzgc可配在任意节点下)

172.18.18.184 s184

172.18.18.185 s185



### 执行扩展节点环境部署脚本

cd /opt/ClusterBuildScripts/expand/install

执行expandEnv.sh

expandEnv.sh包含：

1. 扩展节点安装dos2unix，expect
2. 配置原有节点和新增节点的免密登录
3. 分发新的host文件
4. 删除扩展节点上的原有环境变量
5. 关闭扩展节点防火墙
6. 在扩展节点上安装jdk

### 执行扩展节点服务部署脚本

cd /opt/ClusterBuildScripts/expand/install

执行expandCompent.sh

expendCompent.sh包含：

1. 安装hadoop中的datanode，resourcemanager服务
2. 安装zookeeper服务
3. 安装hbase中的regionserver服务
4. 安装hive，kafka，scala，spark，rocketmq，es服务
5. 配置环境变量和服务启动目录
6. 将节点上服务安装情况同步到主配置文件cluster\_conf.properties中

### 启动扩展节点服务

进入扩展节点服务启动目录，执行服务启动脚本（若在目录下没有找到对应服务的启动脚本，则说明该服务需要集群重启生效）

cd /opt/ClusterBuildScripts/expand/service

#### 启动datanode

sh start\_datanode.sh

#### 启动elasticSearch

sh start\_es.sh

#### 启动hive

sh start\_hive.sh

#### 启动nodemanager

sh start\_nodemanager.sh

#### 启动rocketmq

sh start\_rocketmq.sh

# 小型化集群搭建

## 机器准备

准备一个机器

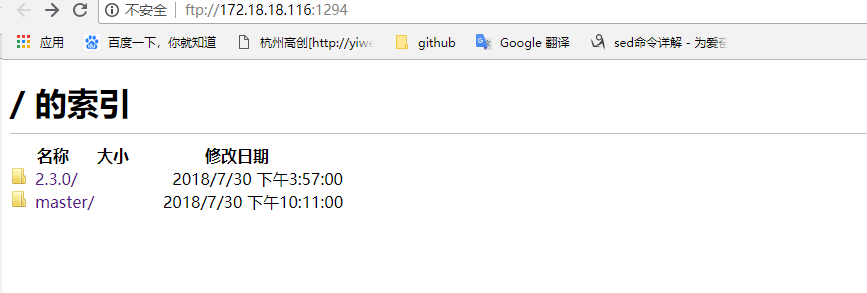
|  |  |
| --- | --- |
| 主机名(Hostname) | IP |
| s186 | 172.18.18.186 |

## 准备集群部署脚本及安装包

### 使用GoSunBigData.tar.gz安装

登录服务器/虚拟机，从出包目录获取安装包，根据出的版本和时间自取

浏览器登录出包目录，地址：<ftp://172.18.18.116:1294/>





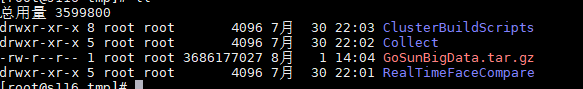
到/opt下执行取包命令

取包命令：wget <ftp://172.18.18.116:1294/分支名/出包时间/>GoSunBigData.tar.gz

解压tar包，执行解压命令

解压命令：tar -zxvf GoSunBigData.tar.gz

解压后目录结构如下：



## 准备集群安装环境

进入bin目录，执行环境部署脚本

cd /opt/ClusterBuildScripts/local/bin

执行命令：

sh envInstall\_local.sh

envInstall\_local.sh包含：

1. 本节点上安装sshpass，dos2unix，expect
2. 配置免密登录
3. 删除环境变量
4. 关闭防火墙
5. 创建日志目录

## 集群组件安装

进入bin目录，执行集群部署脚本

cd /opt/ClusterBuildScripts/local/bin

执行命令：

sh compentInstall\_local.sh

compentInstall\_local.sh包含：

1. 安装JDK，scala，mysql
2. 安装Hadoop，HBase，Zookeeper，Spark，Phoenix
3. 安装Rocketmq，azkaban，ElasticSearch，Kinaba

## 启动停止集群服务

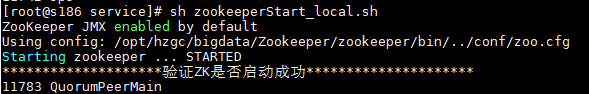
进入单节点组件启停目录

### 启动集群服务

#### 启动zookeeper

cd /opt/ClusterBuildScripts/local/service

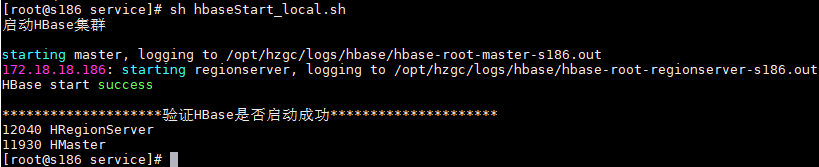
sh zookeeperStart\_local.sh



#### 启动hbase

cd /opt/ClusterBuildScripts/local/service

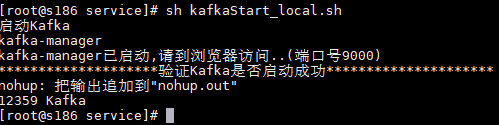
sh hbaseStart\_local.sh



#### 启动kafka

cd /opt/ClusterBuildScripts/local/service

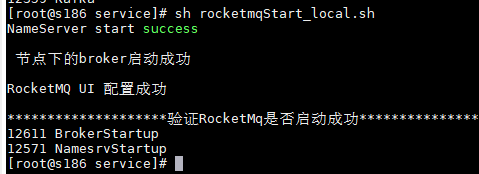
sh kafkaStart\_local.sh



#### 启动rocketmq

cd /opt/ClusterBuildScripts/local/service

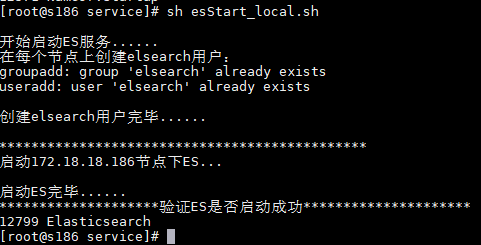
sh rocketmqStart\_local.sh



#### 启动elasticserach

cd /opt/ClusterBuildScripts/local/service

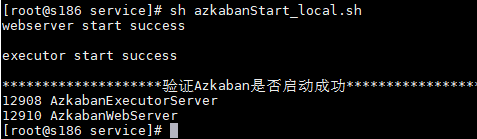
sh esStart\_local.sh



#### 启动azkaban

cd /opt/ClusterBuildScripts/local/service

sh azkabanStart\_local.sh



### 停止集群服务

#### 停止zookeeper

cd /opt/ClusterBuildScripts/local/service

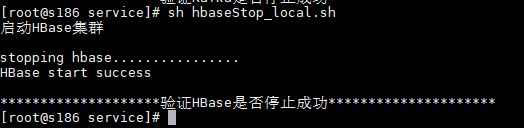
sh zookeeperStop\_local.sh



#### 停止hbase

cd /opt/ClusterBuildScripts/local/service

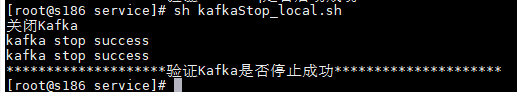
sh hbaseStop\_local.sh



#### 停止kafka

cd /opt/ClusterBuildScripts/local/service

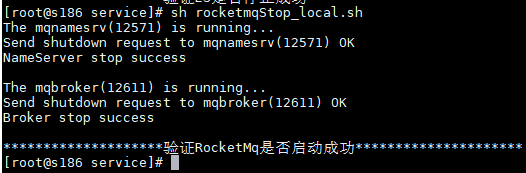
sh kafkaStop\_local.sh



#### 停止rocketmq

cd /opt/ClusterBuildScripts/local/service

sh rocketmqStop\_local.sh



#### 停止elasticserach

cd /opt/ClusterBuildScripts/local/service

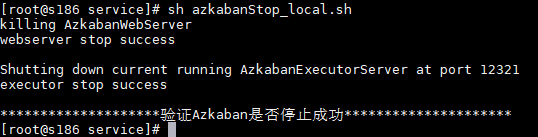
sh esStop\_local.sh



#### 停止azkaban

cd /opt/ClusterBuildScripts/local/service

sh azkabanStop\_local.sh



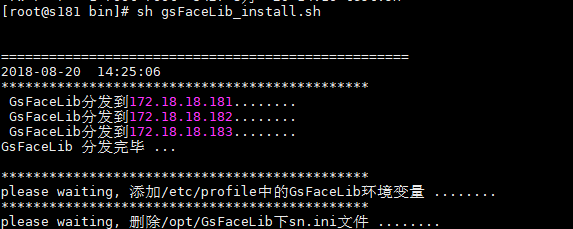
# 项目部署

## 部署算法库

### 分发配置

cd /opt/GoSunBigDataDeploy/project/bin/

sh gsFaceLib\_install.sh

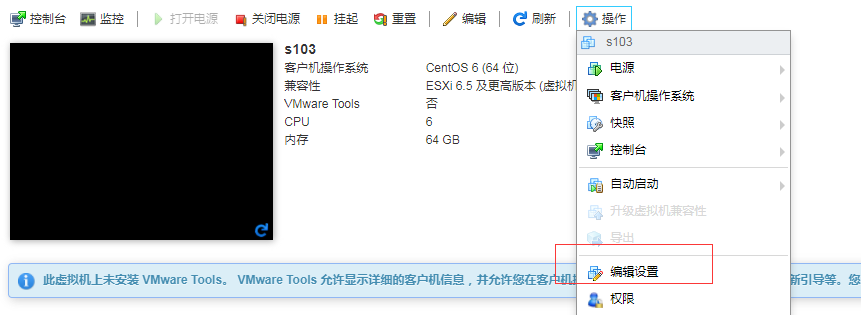


### 添加加密狗

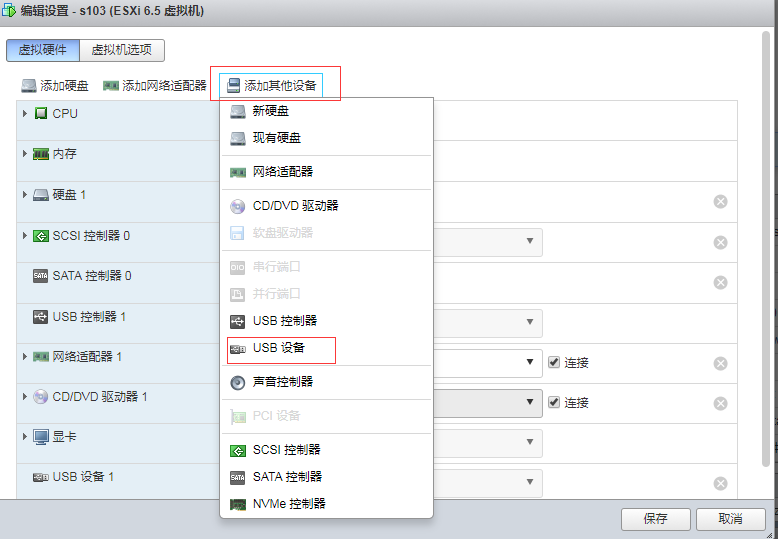
### 添加加密狗驱动（vmware情况下执行）

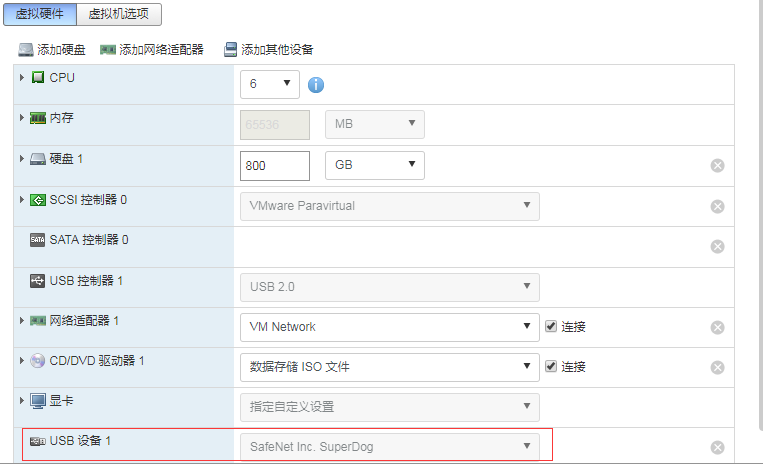
在对应的服务器中添加加密狗

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：





### 添加加密狗驱动（每台需要配置算法的服务器下执行）

cd /opt/GsFaceLib/driver/

rpm -ivh superdog-1.0-1.i386.rpm



### 配置算法库

#### 修改图片解析方式

cd /opt/GsFaceLib/

vim facelib.config

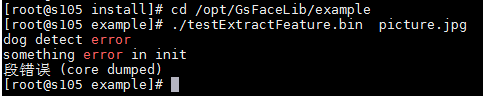
修改为device=CPU



#### 进入/opt/GsFaceLib/example目录下

cd /opt/GsFaceLib/example

./testExtractFeature.bin picture.jpg



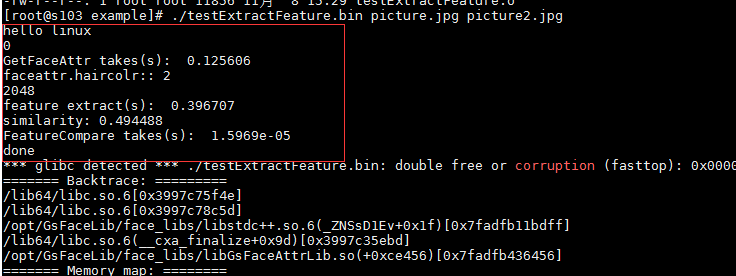
#### 生成新的sn.ini文件

将新产生的sn.ini文件内容发给**算法组 周琛辉** 转成相应的密钥后保存到sn.ini

#### 继续进入/opt/GsFaceLib/example执行

cd /opt/GsFaceLib/example

./testExtractFeature.bin picture.jpg



## 部署Collect

### 修改FTP配置文件

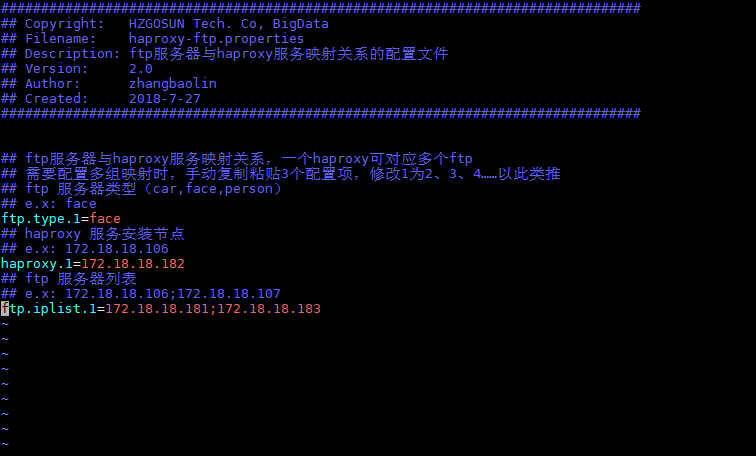
vim /opt/GoSunBigDataDeploy/project/conf/haproxy-ftp.properties

参数说明：

ftp.type 服务器存储数据类型：face（人脸），person（行人），car（汽车）

haproxy 代理节点ip

ftp.iplist ftp服务器列表



### 启动FTP

cd /opt/Collect

sh bin/start-ftpserver.sh

查看Ftp是否启动成功

tailf logs/ftpserver.log

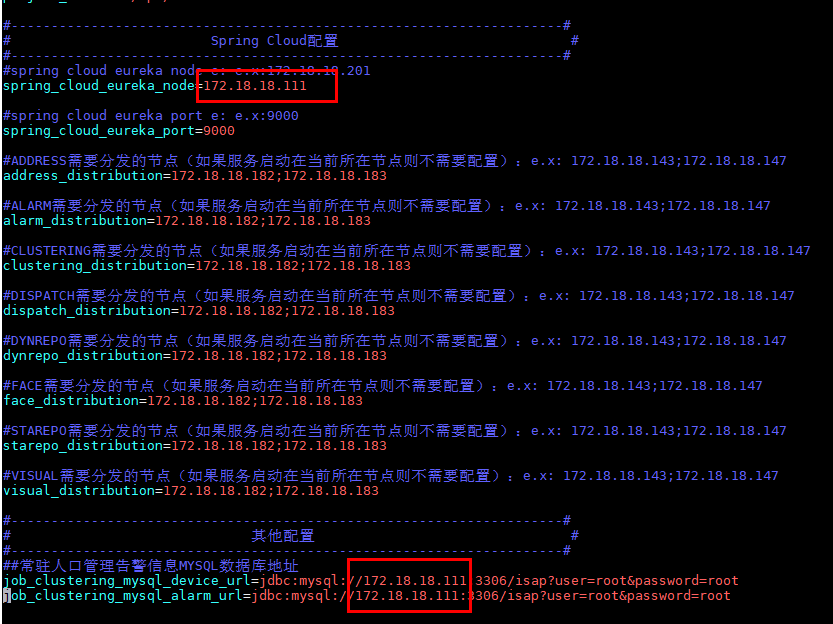


## 部署GoSunBigData

### 修改配置文件

vim /opt/GoSunBigDataDeploy/project/conf/haproxy-ftp.properties

Spring\_cloud的ip和端口、mysql的ip向平台组同事获取



### 执行部署脚本

cd /opt/GoSunBigDataDeploy/project/bin/

sh project\_distribute.sh

### CLuster模块（建表，验证，启动job）

执行建表脚本

cd /opt/GoSunBigDataDeploy/project/bin/

sh create-all-table.sh

#### 验证Es索引

查看es索引是否创建成功

在浏览器中输入http://172.18.18.181:9200/dynamic?pretty

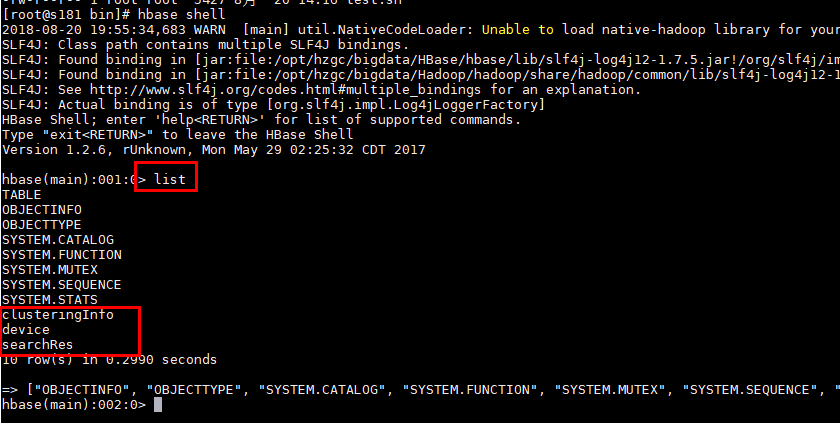


#### 验证Hbase表

执行下面命令查看hbase中表数据

source /opt/hzgc/env\_bigdata.sh

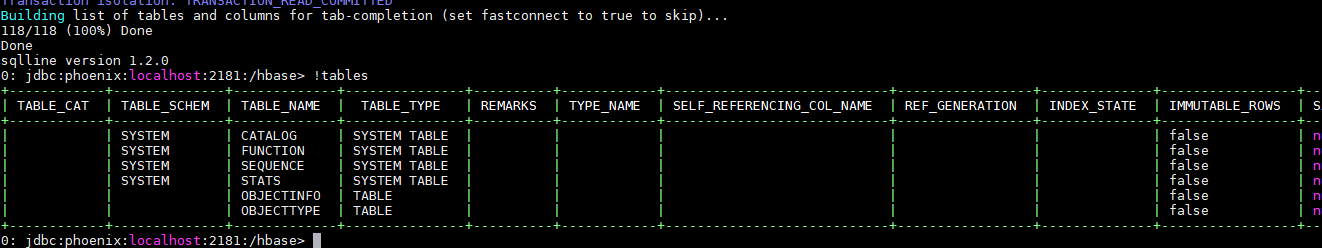
hbase shell



#### 验证Phoenix表

查看phoenix下的表

sqlline.py



#### 验证Spark

查看spark-udf

输入hive

show functions



#### 启动start-face-alarm-job.sh

cd /opt/GoSunBigData/Cluster/spark/bin

执行命令 sh start-face-alarm-job.sh

#### 启动start-kafka-to-parquet.sh

cd /opt/GoSunBigData/Cluster/spark/bin

执行命令 sh start-kafka-to-parquet.sh

#### 启动start-kmeans-clustering-job.sh

cd /opt/GoSunBigData/Cluster/spark/bin

执行命令 sh start-kmeans-clustering-job.sh

#### 启动azkaban任务

cd /opt/GoSunBigData/Cluster/azkaban/bin

sh create-schedule-job-to-zip.sh

cd zip 进入压缩包目录

ll 查看压缩包

TIM图片20180621100436

把压缩包拷贝到本地

进入Azkaban网页 <https://172.18.18.103:8443启动定时任务>

点击Create Project



填写Name和Description然后点击Create Project



点击Upload上传对应的三个zip包，上传成功后点击Upload





点击Execute Flow然后点击Schedule





填写对应的Time(可以往后填一点)和Date(选择当天就ok)，Recurrence中Days根据ZIP包的名称进行选择，然后点击Schedule，然后点击Continue就ok了



### 微服务模块

common目录：address，face，visual

face目录：alarm，peoplemanager，dispatch，dynRepo，staRepo

服务启动方法以address为例：

启动address服务

cd /opt/GoSunBigData/Service/common

cd address

sh bin/start-address.sh

tailf log/2018-06-22.address.0.log 查看服务是否启动成功



alarm，peoplemanager，dispatch，dynRepo，face，staRepo，visual，person，car服务启动与上面步骤类似(服务启动完成后jps查看下对应的服务进程是否启动成功)