# 一、部署环境：

## 1.服务器配置：

三个节点：s100 172.18.18.100；s101 172.18.18.101；s102 172.18.18.102；

服务器单节点最低配置：8核、64G内存、500G硬盘

操作系统CentOS-6.7

1.在linux中安装jdk

2.分配并搭建虚拟机

3.安装FI集群

4.安装RocketMQ

5.安装Elasticsearch

文档地址：[\\192.168.1.43\组内共享\集群搭建](file:///\\192.168.1.43\组内共享\集群搭建)

## 2.开发环境Jdk安装：

1.下载并安装jdk：

<http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>

2.设置java环境变量

3.CMD中运行java –version查看是否安装成功

## 3.开发环境Maven安装与配置：

1.下载并安装maven：[http://maven.apache.org/download.cgi#](http://maven.apache.org/download.cgi)

2.设置maven环境变量

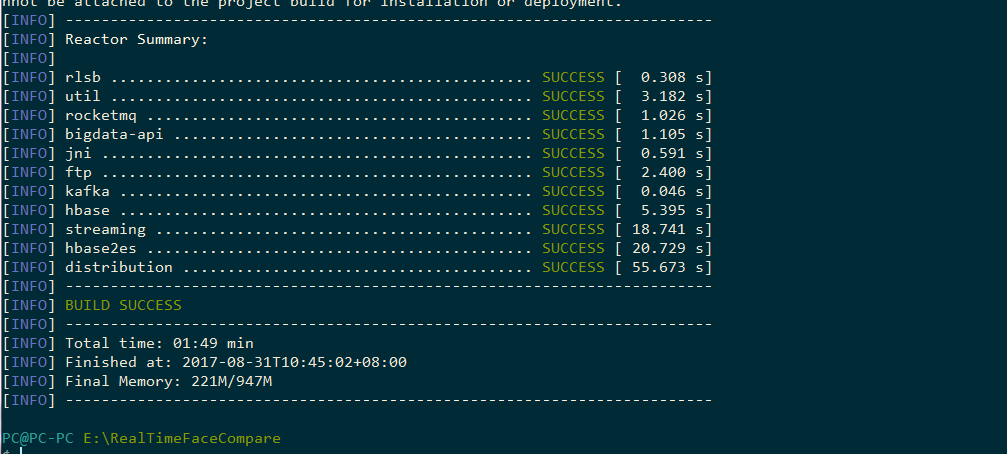
3.在D:\apache-maven-3.5.0\conf中替换项目需要的settings.xml文件，文件获取地址：[\\192.168.1.43\组内共享\Maven配置文件](file:///\\192.168.1.43\组内共享\Maven配置文件)

# 二、项目编译打包：

在gitHub上下载项目源码：<https://github.com/Zz897426182/RealTimeFaceCompare>

打开项目文件，在空白处Shift+鼠标右键点击“在此处打开命令窗口”：

输入编译打包命名：mvn clean package



打包编译后：

项目中所有脚本存储路径：

E:\RealTimeFaceCompare\Distribution\target\RealTimeCompare\bin

项目中所有配置文件存储路径：

E:\RealTimeFaceCompare\Distribution\target\RealTimeCompare\conf

项目中所有编译打包后的jar包与依赖jar包存储路径：

E:\RealTimeFaceCompare\Distribution\target\RealTimeCompare\lib

项目中所有建表语句存储路径：

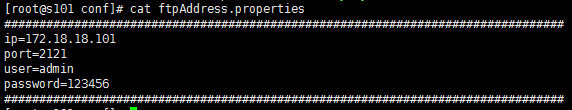
E:\RealTimeFaceCompare\Distribution\target\RealTimeCompare\sql

各个子模块编译后生成的jar包分别存储在各个子模块的\target下

# 三、服务配置：

## 1.FtpServer服务配置：

进入/opt/RealTimeCompare/conf下，根据需要修改ftpAddress.properties文件中的ip地址，如下：

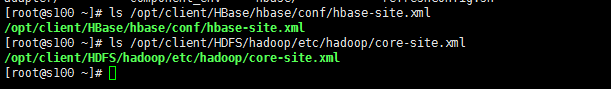


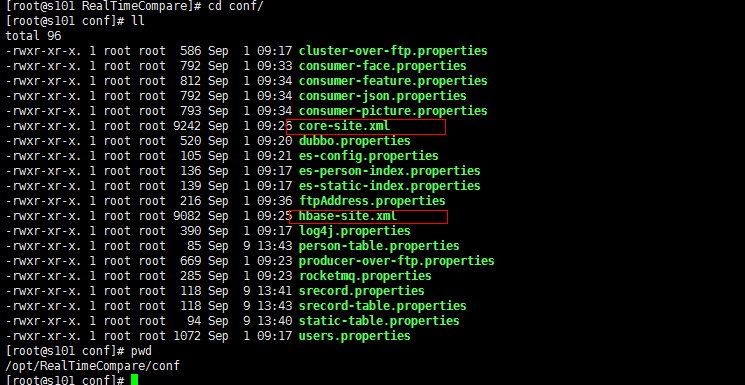
## 2.HBase服务配置：

进入Fi客户端，将下面2个文件copy到s101机器/opt/RealTimeCompare/conf目录下：

/opt/client/HBase/hbase/conf/hbase-site.xml

/opt/client/HDFS/hadoop/etc/hadoop/core-site.xml





## 3. Elasticsearch服务配置：

Elasticsearch配置前先参考《安装elasticSearch搜索引擎》文档完成ES安装，地址：[\\192.168.1.43\组内共享\集群搭建](file:///\\192.168.1.43\组内共享\集群搭建)

### 3.1 ES 索引及其映射关系生成

1.部署ES 索引映射关系，执行以下两个脚本：

index-static.sh.templete

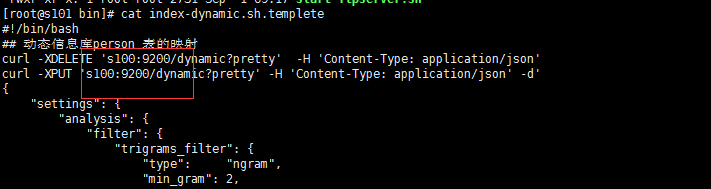
index-dynamic.sh.templete

2.进行如下操作：

cp index-static.sh.templete index-static.sh

cp index-dynamic.sh.templete index-dynamic.sh

3.修改index-static.sh 和 index-dynamic.sh里面的ES 服务器的地址：



比如ES 服务在172.18.18.115 上面：

可以把地址直接换成如下：

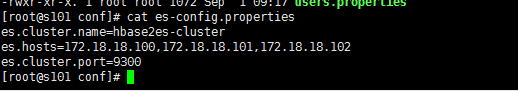
curl –XPUT ' 172.18.18.115:9200/dynamic?pretty' -H 'Content-Type: application/json' -d'

4.执行脚本，生成ES数据类型映射关系，创建索引：

sh index-dynamic.sh

sh index-static.sh

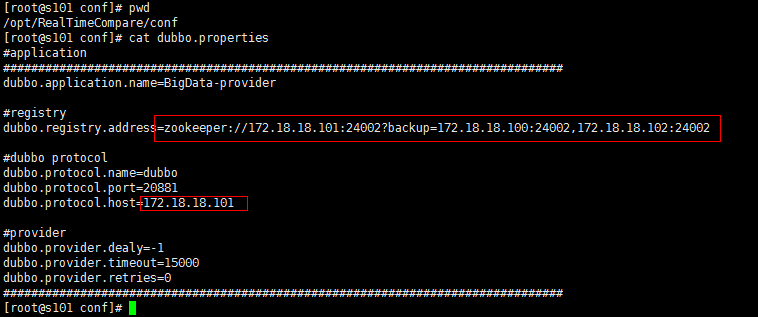
### 3.2 ES IP配置文件修改



修改es.hosts 为ES服务所在的机器IP

## 4.Dubbo服务配置：

进入/opt/RealTimeCompare/conf下，根据需要修改dubbo.properties文件：



## 5. RocketMQ服务配置：

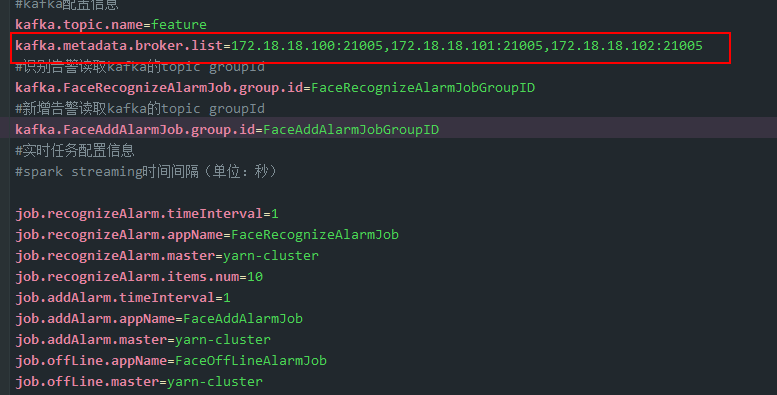
RocketMQ配置前先参考《安装rocketMQ与使用》文档完成RocketMQ安装，地址：[\\192.168.1.43\组内共享\集群搭建](file:///\\192.168.1.43\组内共享\集群搭建)

进入/opt/RealTimeCompare/conf下，根据需要修改rocketmq.properties文件中的address地址为NameServer地址加端口号，如下：



## 6.SparkJob服务配置：

进入/opt/RealTimeCompare/conf下，根据需要修改sparkJob.properties文件中的kafka.metadata.broker.list为集群kafak地址加kafka监听端口号，如下：



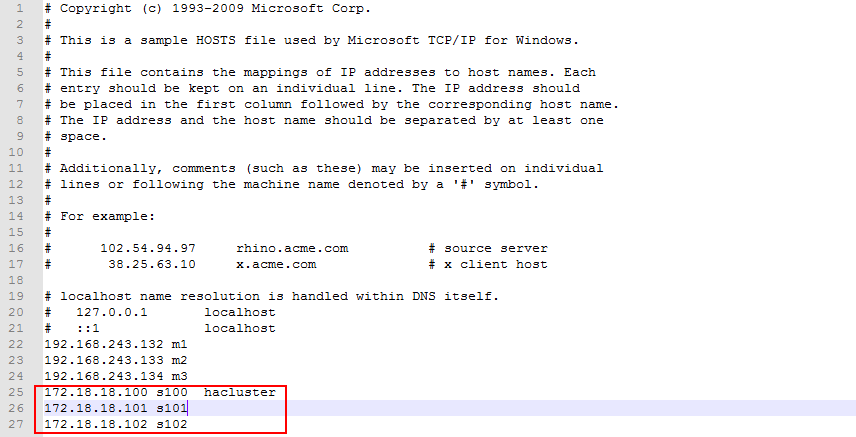
# 四、服务启动：

## 1.连接FI集群：

1.首先修改windows下的hosts文件：

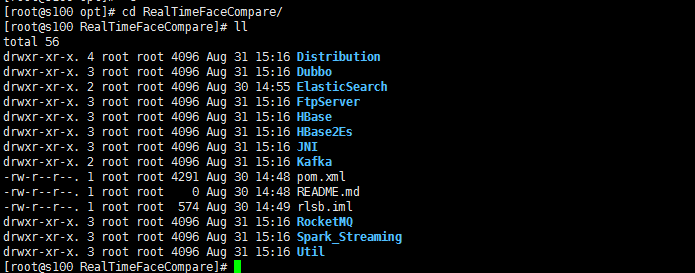
hosts文件路径：[C:\Windows\System32\drivers\etc](C:\\Windows\\System32\\drivers\\etc)

在hosts文件中添加服务器的IP地址与服务器名称，如下所示：

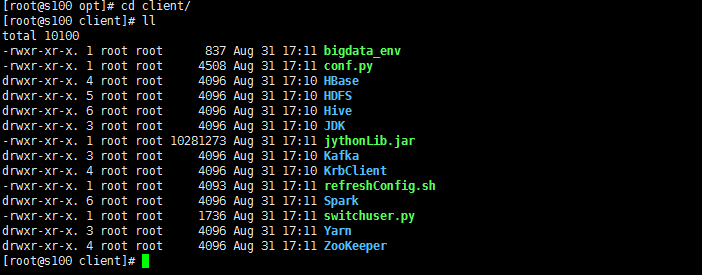


2.使用Xshell或者MobaXterm软件连接FI主节点s100

项目路径：/opt/RealTimeFaceCompare



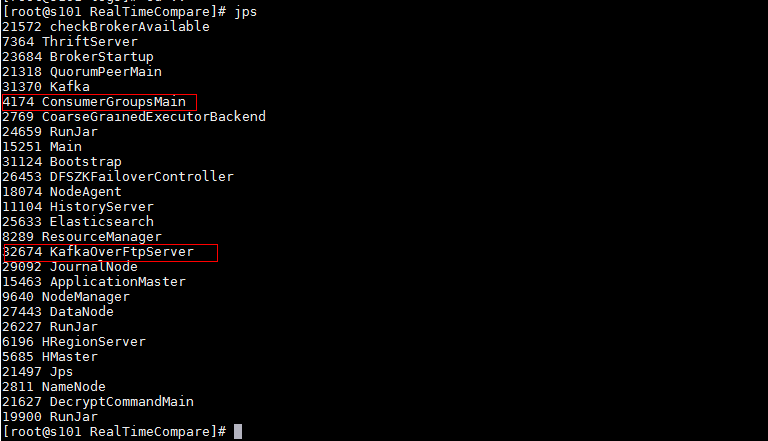
FI客户端路径：/opt/Client



## 2.启动Ftp服务：

进入s101机器下的/opt/RealTimeCompare 路径下：

首先jps查看所有进程，查看是否有ConsumerGroupMain与KafkaOverFtpServer进程：



若没有直接执行脚本启动FtpServer服务与Kafaka consumer消费组；若存在使用“kill -9 进程号”将ConsumerGroupMain与KafkaOverFtpServer进程杀掉，若没有杀掉，多杀几次就好。

./bin/start-ftpserver.sh启动FtpServer服务

./bin/start-consumer.sh启动Kafaka consumer消费组

## 3.启动Dubbo服务：

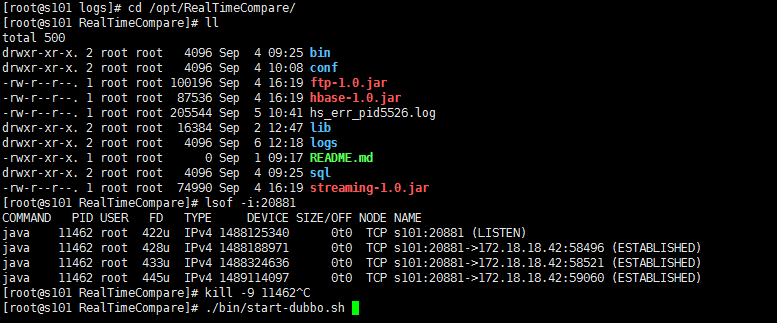
进入s101机器下的/opt/ RealTimeCompare路径下：

lsof -i:20881 （20881为Dubbo端口号）查看Dubbo进程号

kill -9 11462

./bin/start-dubbo.sh

tail -f logs/dubbo.log日志中没有报错信息，出现“Dubbo service server started!”，Dubbo启动成功。



## 4.启动RocketMQ服务：

1.在主节点上启动NameServer

进入 /opt/rocketmq/bin路径下：

nohup ./mqnamesrv –n 172.18.18.102:9876 > ./start.log 2>&1 &

cat start.log

通过jps查看进程中是否启动NameServer

2.在两个Master上启动Broker

进入/opt/rocketmq/bin路径下：

nohup ./mqbroker -n 172.18.18.102:9876 -c ../conf/2m-noslave/broker-a.properties >./start.log 2>&1 &

cat start.log

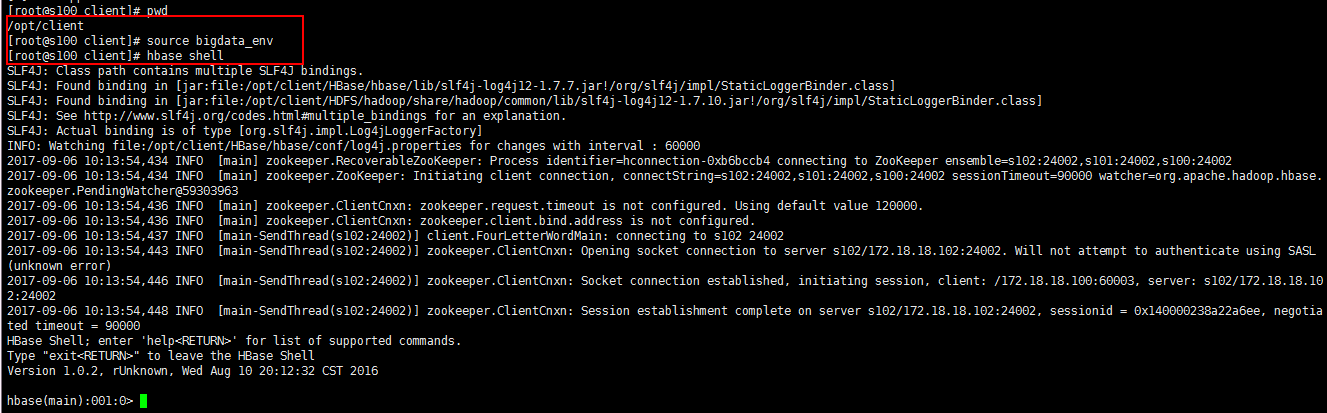
通过jps查看进程中是否启动

## 5.连接HbaseClient:

进入/opt/client路径下：

source bigdata\_env

hbase shell



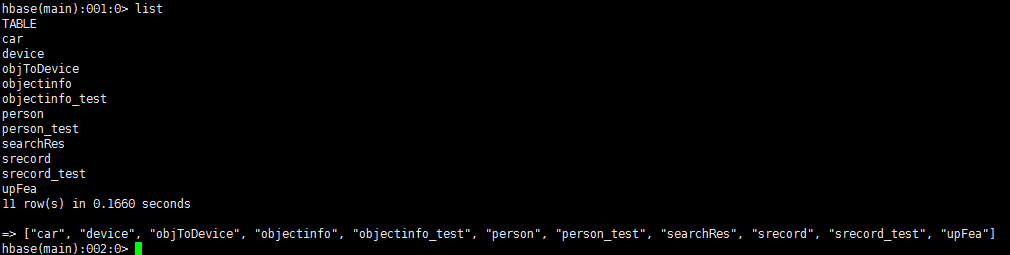
## 6.数据库建表:

设备表：先进入s101机器下的/opt/RealTimeCompare/sql路径下查看并复制device.sql文件中的建表语句，然后连接HBaseClient后粘贴建表语句并回车执行；设备一共有两张表（device、objToDevice）；

静态库建表：进入s101机器下的/opt/RealTimeCompare/bin路径下，执行create-table.sh脚本；静态库一共有两张表（objectinfo、srecord）；

动态库建表：先进入s101机器下的/opt/RealTimeCompare/sql路径下查看并复制dynamic.sql文件中的建表语句，然后连接HBaseClient后粘贴建表语句并回车执行；动态库一共有四张表（person、car、upFea、searchRes）；

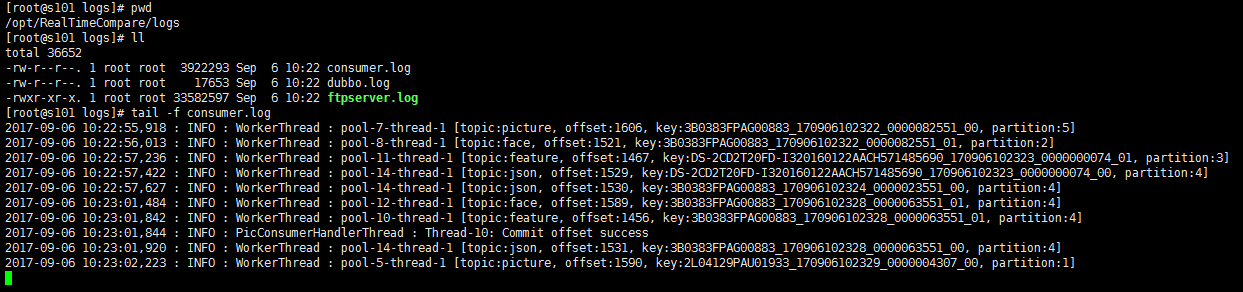
最后使用“list”命令查看是否建表成功。



## 7.查看log文件：

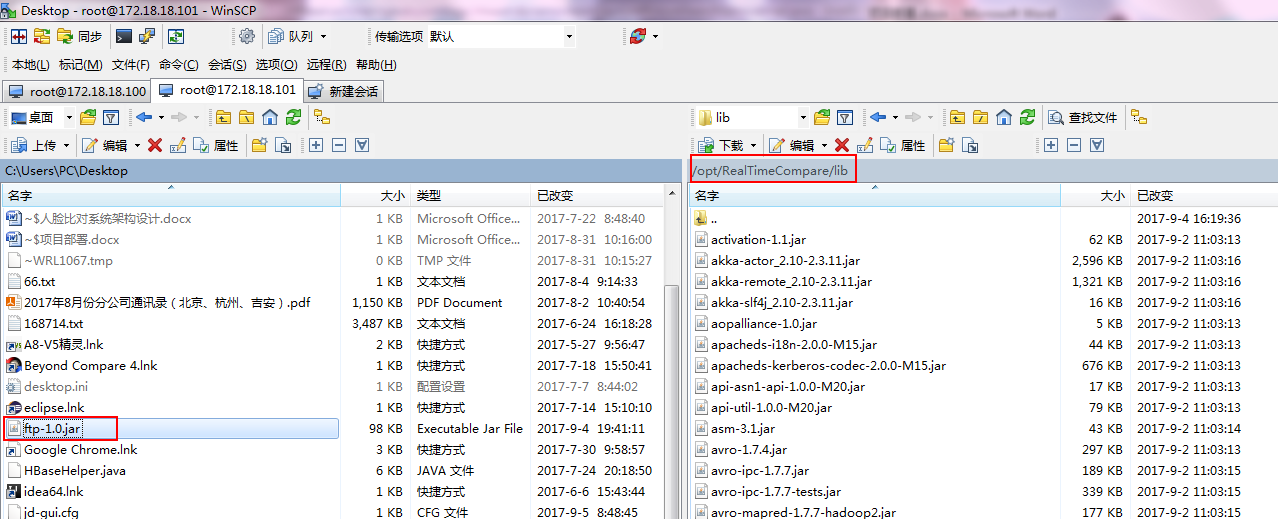
进入s101机器下的/opt/RealTimeCompare/logs路径下：

tail -f filename



## 8.替换项目中jar包：

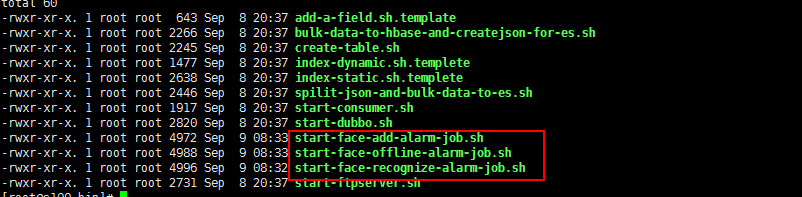
使用winSCP连接s101机器，进入/opt/RealTimeCompare/lib路径下：



选择左面本地机器编译打包完成的jar包，按“F5”拷贝至linux服务器中，或者使用MobaXterm软件在相同路径下替换所需jar包。替换完成后一般只需要重启Dubbo服务就可以了，若是修改了[ftp.jar](ftp://ftp.jar)[包需要先重启ftpserver](ftp://ftp.jar包需要先重启ftpserver)与consumer服务。

## 9.告警任务启动：

进入s101机器下的/opt/RealTimeCompare/bin路径下：



选择对应的告警任务脚本并执行，然后在查看/opt/RealTimeCompare/logs中对应告警任务log日志是否有报错信息。

# 五、修改表字段：

修改DeviceTable（设备）字段：package com.hzgc.hbase.device. DeviceTable

修改DynamicTable（动态库）字段：package com.hzgc.hbase.dynamicrepo. DynamicTable

修改ObjectInfoTable（静态库） 字段：package com.hzgc.dubbo.staticrepo.ObjectInfoTable

修改SrecordTable （静态库）字段：package com.hzgc.dubbo.staticrepo.SrecordTable