项目部署

# 一、项目部署主要分以下几步

1.虚拟机准备

2编译打包上传项目

3服务配置

4建表和索引

5服务启动

文档地址：[\\192.168.1.4\组内共享\集群搭建](file:///Z:\集群搭建)

# 二、开源集群准备

分配并搭建虚拟机

三个节点：s103 172.18.18.103

s104 172.18.18.104

s105 172.18.18.105

服务器单节点最低配置：6核、64G内存、500G硬盘；

**三、编译打包上传项目**

## 3.1.下载项目源码到本地

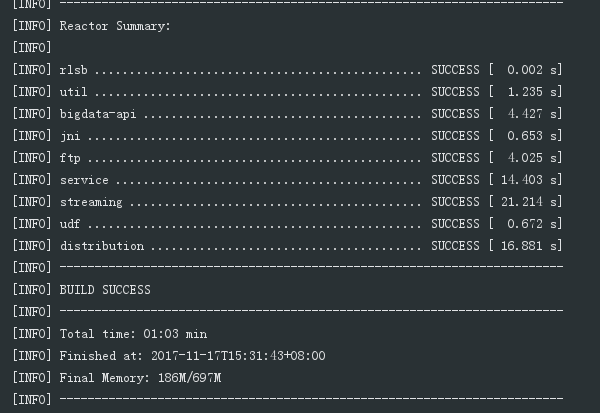
在gitHub上下载master分支最新项目源码：

<https://github.com/Zz897426182/RealTimeFaceCompare>

## 3.2.编译打包

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：

项目打包文件存储路径：

E:\RealTimeFaceCompare\Distribution\target

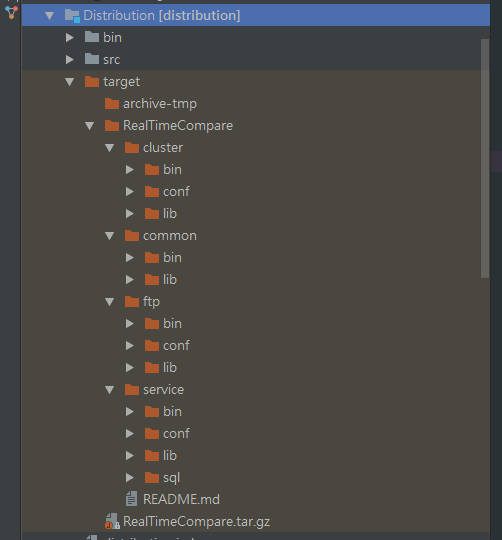
项目中所有建表语句存储路径：

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

各个子模块编译后生成的jar包存储路径：

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

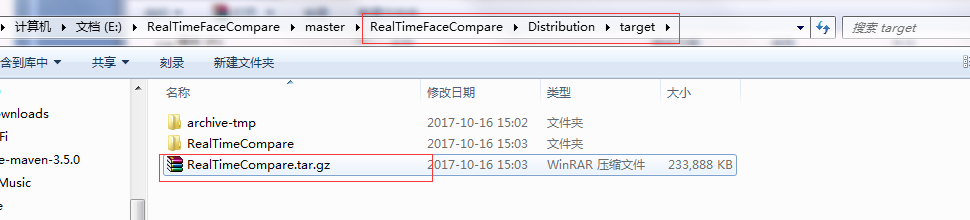
项目打包后结构：



## 3.3.上传项目

项目打包文件存储路径：

E:\RealTimeFaceCompare\Distribution\target



将项目打包文件上传至服务器主节点/opt下

创建RealTimeFaceCompare文件夹：在/opt下mkdir RealTimeFaceCompare

解压项目文件到/opt/RealTimeFaceCompare/下：

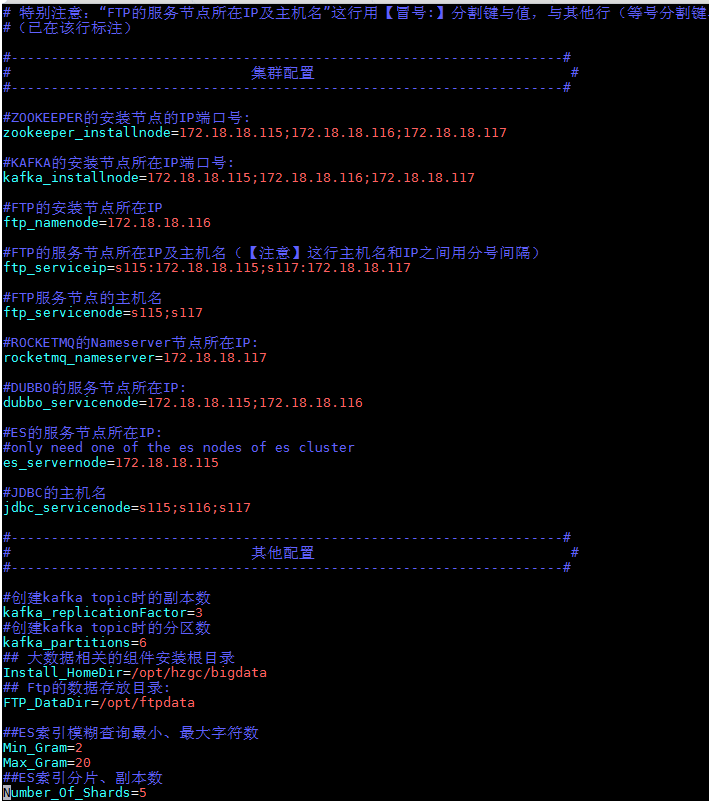
tar -xf RealTimeCompare.tar.gz -C /opt/RealTimeFaceCompare/

# 配置项目配置文件

根据备注提示信息配置项目配置文件

/opt/RealTimeCompare/common/conf/project-conf.properties：

部分配置文件如下：



# 五、一键配置项目配置

一键配置项目配置包括/opt/RealTimeCompare/下的：

1. 配置./common/conf/
2. 配置./ftp/conf/
3. 配置./service/conf/
4. 配置./cluster/conf/

执行步骤：进入/opt/RealTimeCompare/common/bin目录

./common/bin/config-projectconf.sh

# 六、一键创建表：执行create-all-table.sh脚本

脚本功能主要包含：

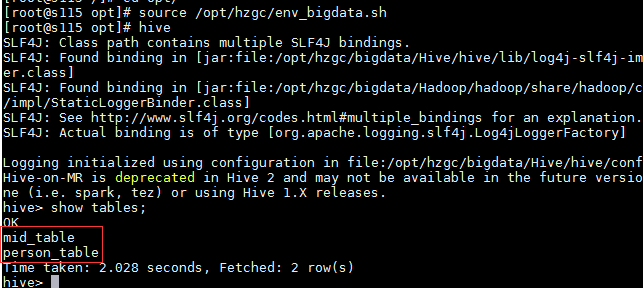
1. ES 索引及其映射关系生成
2. Hive建表
3. Add-udf
4. 创建kafka的Topic
5. 创建hbase静态库、动态库表格

执行步骤：进入/opt/RealTimeCompare/service/bin目录

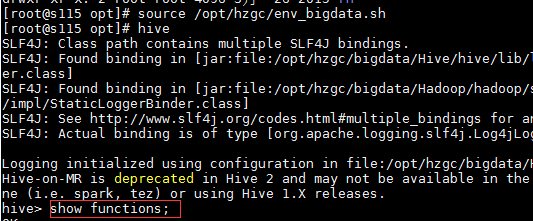
sh create-all-table.sh

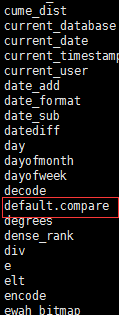
# 七、查看索引、表格、UDF、topic是否创建成功

**7.1查看Hive建表是否成功：**



**7.2查看Add-udf是否成功：**



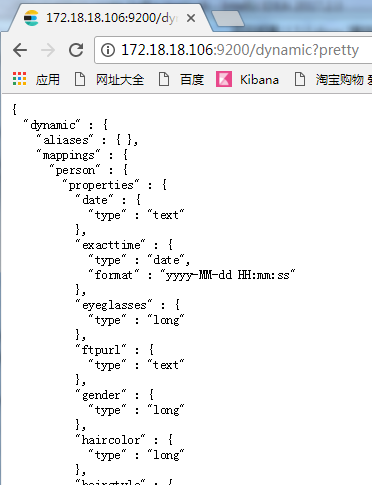
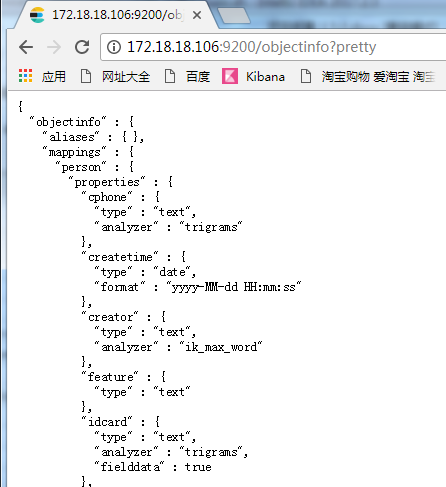


**7.3查看创建es索引是否成功：**

在浏览器中输入

<http://172.18.18.106:9200/objectinfo?pretty>

<http://172.18.18.106:9200/dynamic?pretty>

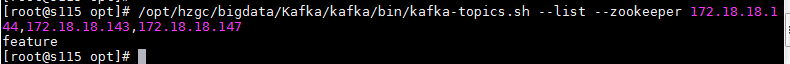


**7.4查看创建kafka topic是否成功：**

执行命令

source/opt/hzgc/env\_bigdata.sh

$KAFKA\_HOME/ bin/kafka-topics.sh --list --zookeeper 172.18.18.106:2181,172.18.18.107:2181,172.18.18.108:2181



**7.5查看hbase表是否成功：**

HBase中需要创建5张表：

设备表： device、objToDevice

动态库建表：upFea、searchRes

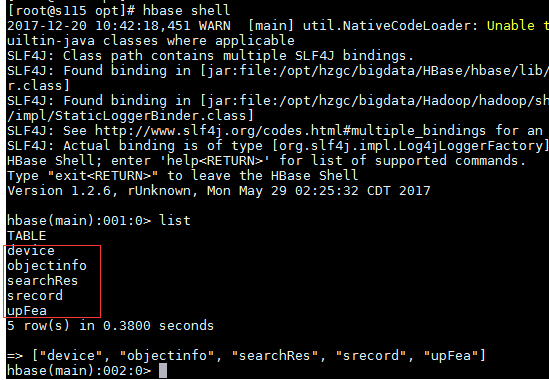
静态库建表：srecord

执行如下命令：

source /opt/hzgc/env\_bigdata.sh

hbase shell

在hbase客户端输入list查看



# 八、服务启动

## 8.1启动FTP服务

## 8.1.1查看FTP进程是否存在

进入到FTP服务节点的机器，jps查看所有进程，查看是否有FTP进程：

或者执行lsof -i:2121查看进程是否存在

若存在使用“kill -9 进程号”将FTP进程杀掉，若没有杀掉，多杀几次就好。

### 8.1.2执行FTP启动脚本：

进入/opt/RealTimeCompare/ftp/bin目录下

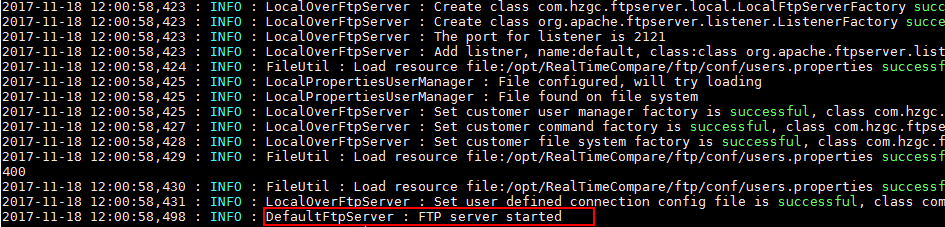
sh start-ftpserver.sh启动FTP服务

### 8.1.3查看FTP日志是否启动成功

进入/opt/RealTimeCompare/ftp/logs目录下

tail -f ftpserver.log

若没有报错信息并显示“FTP server started”，FTP启动成功。



## 8.2.启动Dubbo服务

### 8.2.1.查看Dubbo进程是否存在

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

或者使用jps命令查看是否存在Main进程

若存在kill -9 进程号杀掉Dubbo进程

### 8.2.2. 执行Dubbo启动脚本：

进入/opt/RealTimeCompare/service/bin目录下

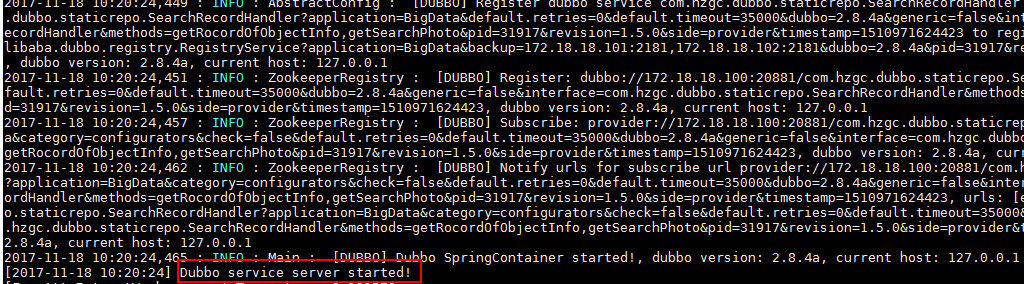
sh start-dubbo.sh启动Dubbo服务

### 8.2.3.查看Dubbo日志是否启动成功

进入/opt/RealTimeCompare/service/logs目录下

tail -f dubbo.log

日志中没有报错信息，出现“Dubbo service server started!”，Dubbo启动成功。



## 8.3.启动consumer服务

### 8.3.1. 执行start-kafka-to-parquet.sh启动脚本

进入/opt/RealTimeCompare/cluster/bin目录下

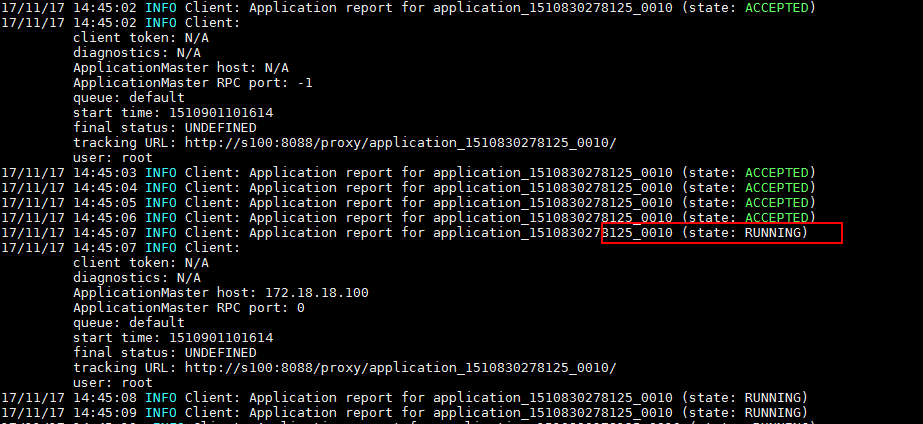
sh start-kafka-to-parquet.sh(先启动thriftserver 删除checkpoint)

### 8.3.2.查看kafkaToParquet.log日志是否启动成功

进入/opt/RealTimeCompare/cluster/logs目录下

tail -f kafkaToParquet.log

日志中没有报错信息，当出现“(state: RUNNING)”，消费服务启动成功。



## 8.4.启动spark任务

### 8.4.1. 识别告警任务启动

进入/opt/RealTimeCompare/cluster/bin目录下

./start-face-recognize-alarm-job.sh

进入/opt/RealTimeCompare/cluster/logs目录下

tail –f sparkFaceRecognizeAlarmJob.log

日志中没有报错信息，当出现“(state: RUNNING)”，识别告警任务启动成功。（如上图）

### 8.4.2. 新增告警任务启动

进入/opt/RealTimeCompare/cluster/bin目录下

./start-face-add-alarm-job.sh

进入/opt/RealTimeCompare/cluster/logs目录下

tail –f sparkFaceAddAlarmJob.log

日志中没有报错信息，当出现“(state: RUNNING)”，新增告警任务启动成功。（如上图）

### 8.4.3. 通过告警任务启动

进入/opt/RealTimeCompare/cluster/bin目录下

./add-offlinejob-crontab.sh

在任意目录下执行crontab -l

当出现下面信息，表示通过告警任务启动成功：

0 0 \* \* \* ${BIN\_DIR}/start-face-offline-alarm-job.sh

## 8.5.启动小文件合并以及守护进程定时任务启动

进入/opt/RealTimeCompare/common/bin目录下：

./smallfile-daemon-crontab.sh