# 集群环境

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ip/主机名 | 登录名 | 密码 | 安装的服务 | 用户角色 |
| 192.168.1.21/twin-cloud01 | root/escheduler | admin123 | nameNode、resourceManager、  SparkMaster、HMaster、kafka、zookeeper | root用户，scheduler租户，也就是实际项目的部署用户 |
| 192.168.1.22/twin-cloud02 | root/escheduler | root | DataNode、NodeManager、HbaseRegionserver、kafka、zookeeper | root用户 |
| 192.168.1.23/twin-cloud03 | root/escheduler | root | DataNode、NodeManager、HbaseRegionserver、kafka  、hive、zookeeper | root用户 |

备注：

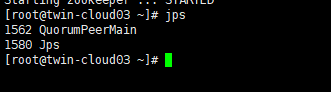
项目基本依赖环境就是zookeeper，使用到大数据计算功能时才会依赖haoop集群，平时只需要启动zookeeper和hadoop集群就可以了。

项目中的上传资源的功能，资源是上传到hdfs路径了，如果需要使用上传资源功能或者hadoop的相关功能，需要先启动hadoop集群。

## Zookeeper和hadoop启动命令

root用户登陆三台机器，分别启动zookeeper服务，启动命令：zkServer.sh start

输入jps有如下表示启动成功：

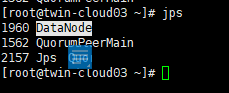
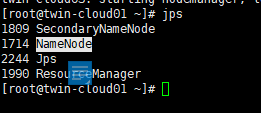


停止命令：zkServer.sh stop

启动hadoop集群的命令，在twin-cloud01机器上执行：

1. start-dfs.sh
2. start-yarn.sh

在twin-cloud01上看到有NameNode另外两台机器上有DataNode就说明启动成功



关机的时候一定要停止集群:

停止zookeeprt，三台机器上分别执行：zkServer.sh stop

停止hadoop机器，任意一台机器执行一次：stop-all.sh

然后jps看不到有进程运行就好了。

Hadoop基本命令的使用和shell差不多，前面加上hadoop fs就好了

hadoop fs -ls /

## Web-ui

hdfs集群UI：[http://192.168.1.21:50070/dfshealth.html#tab-overview](http://192.168.1.21:50070/dfshealth.html" \l "tab-overview)

yarn集群UI：<http://192.168.1.21:8088/cluster>

sparkUI: [http://192.168.1.21:8080/](http://192.168.1.21:8088/cluster)

等。

# 项目部署

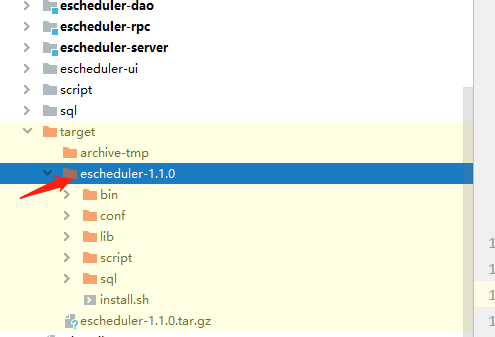
此项目不支持在window上调度task，但是controler层（项目中叫api）可用和前端进行增删改查的交互。

## 部署方法

编译项目：mvn -U clean package assembly:assembly -Dmaven.test.skip=true

（编译项目不成功的换先编译rpc模块，然后在编译项目）

会在项目中target目录生成escheduler-1.1.0文件夹，



把文件夹重命名为escheduler，把install.sh替换为附件中的



把conf/env文件夹下的.escheduler\_env.sh替换为附件中的,附件中的.sh文件是机器的一些配置信息（主要包括zookeeper和hdfs的ip地址）。执行install.sh时候会将源码中的配置信息替换为install中定义的。

然后escheduler用户登陆任意1台机器。

执行： sh install.sh （前提是三台机器要ssh免登录，因为此命令会将本机器文件scp到另外两台）

然后将escheduler上传到linux /data1\_1T/下，然后改变权限： chomd 755 /data1\_1T/escheduler

## 启动服务命令

02上启动api

java -classpath /data1\_1T/escheduler/conf:/data1\_1T/escheduler/lib/\* cn.escheduler.api.ApiApplicationServer

在02 03上启动work

java -classpath /data1\_1T/escheduler/conf:/data1\_1T/escheduler/lib/\* cn.escheduler.server.worker.WorkerServer

java -classpath /data1\_1T/escheduler/conf:/data1\_1T/escheduler/lib/\* cn.escheduler.server.worker.WorkerServer

在01说启动master

java -classpath /data1\_1T/escheduler/conf:/data1\_1T/escheduler/lib/\* cn.escheduler.server.master.MasterServer