# RocketMQ 入门

Wirter：金文浩

需要前置环境： jdk1.8 + mvn

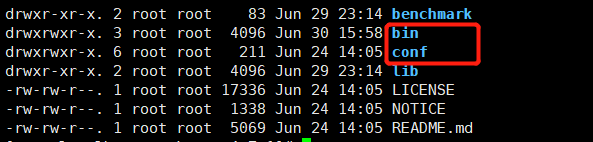
将该压缩包上传至服务器, : rz



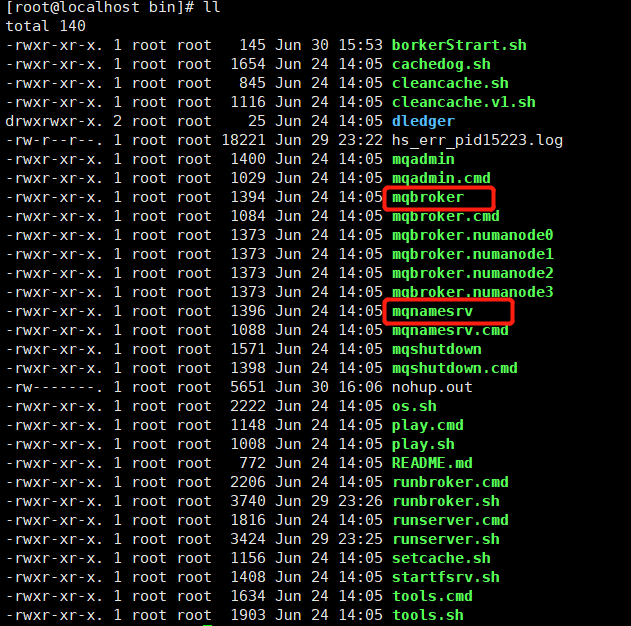
解压 ：unzip

编译安装该项目 ：mvn -Prelease-all -DskipTests clean install -U （过程中报错，看日志，多半是jdk 或者时 mvn的问题）

编译成功后进入目录：distribution/target/apache-rocketmq/bin



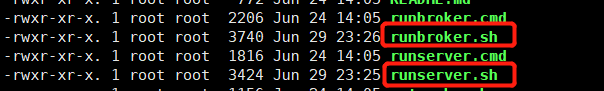
主要还是这两个目录比较重要。



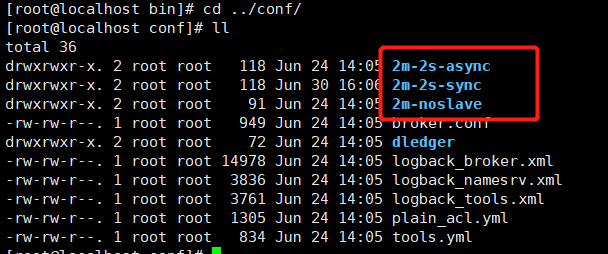
Nameserver broker 启动脚本（先别启动）

nohup sh mqnamesrv & tail -f nohup.out nohup sh mqbroker -c 配置文件 & tail -f nohup.out

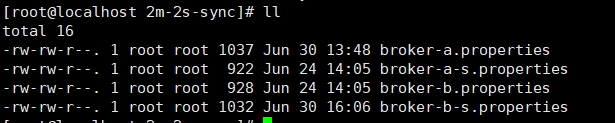
如果启动broker 爆内存不够，需修改，下面两个配置文件。将所有的xms 修改为128m



Rocket集群有以下几种模式：2主2从异步/2主2从同步/2主



本文选择2m-2s-sync，进入目录，有以下文件。见名知意



本文举一个配置例子：

namesrvAddr=192.168.43.142:9876;192.168.43.143:9876

**namesrvAddr：**nameServer实例的ip路径及端口 默认端口为9876，如需改正需增加配置文件 namesrv.properties :listenPort=5432

listenPort=10911

监听端口

storePathRootDir=/home/rocketmq/store-a

数据存储路径

brokerClusterName=DefaultCluster

集群名字

brokerName=broker-a

broker主机名

brokerId=0

0主1从

deleteWhen=04

删除文件时间点

fileReservedTime=48

fileReservedTime=120

brokerRole=SYNC\_MASTER

节点属性，从机配SLAVE

flushDiskType=ASYNC\_FLUSH

刷盘方式 异步/同步

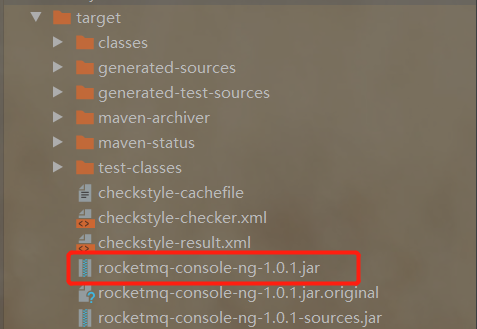
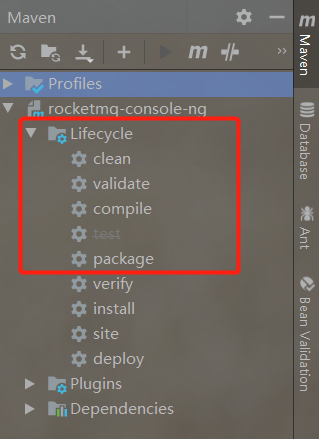
启动集群步骤：

1. 先启动nameserver
2. 在启动 broker 带配置文件启动 nohup sh mqbroker -c ../conf/2m-2s-sync/配置 & tail -f nohup.out

可视化页面搭建：



解压，我这里用的idea ，更改appliacation 的nameserv 配置。

打包成jar包 

启动jar包 java -jar xxxx.jar

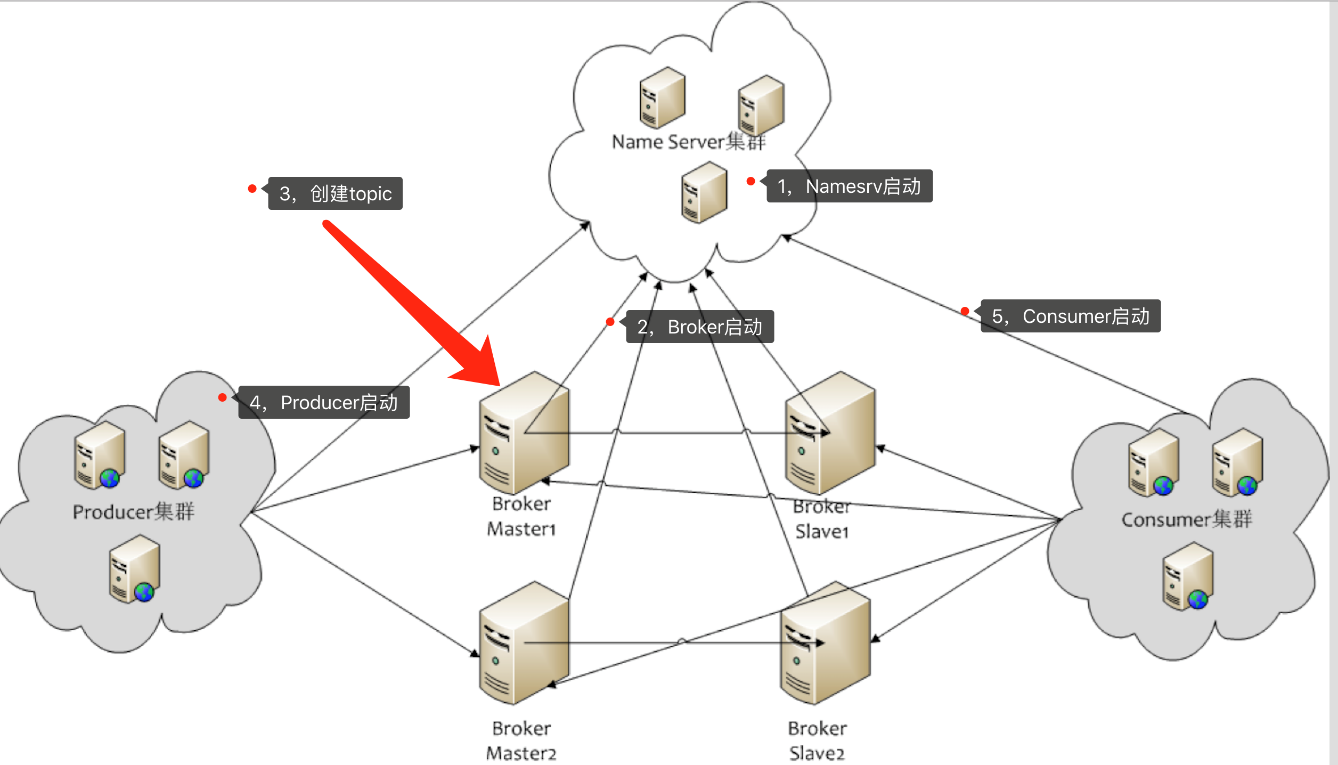


下面推荐一个rocketmq 架构说明的博客，便于理解namesrv,broker,producer等之间的含义及关系：

<https://blog.csdn.net/javahongxi/article/details/72956608> csdn

<http://rocketmq.apache.org/docs/core-concept/> 官网

rocketmq的架构角色：



由上图可知，集群角色分为四类：producer/consumer/name server/broker

关联关系：

Broker 启动将broker及broker上的topic信息注册到name server上

Producer 在发送消息时，需要先通过长连接从name server获取broker消息，来确定自己需要发送到哪个broker上，然后通过长连接推送消息

Consumer 在消费过程中同样需要从name server获取broker信息。另外：由于broker有主动推送消费功能，因此需要consumer与broker建立连接，去主动推送消费。

简单来说：name server相当于broker的注册中心，producer/consumer 需要name server才可以访问broker。

因此：启动时，先要启动name server服务

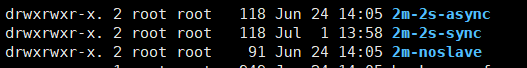
集群模式：

单节点：即1master

多master无slave

多 master多slave 主从同步 （同步即：当producer将消息写入master broker后，直接返回写入成功，通过异步线程进行slave broker 数据同步）

多master 多slave 主从异步（与上述逻辑相反）



发送方式： （demo里都有例子）

1、同步发送

2、异步发送

3、单向发送

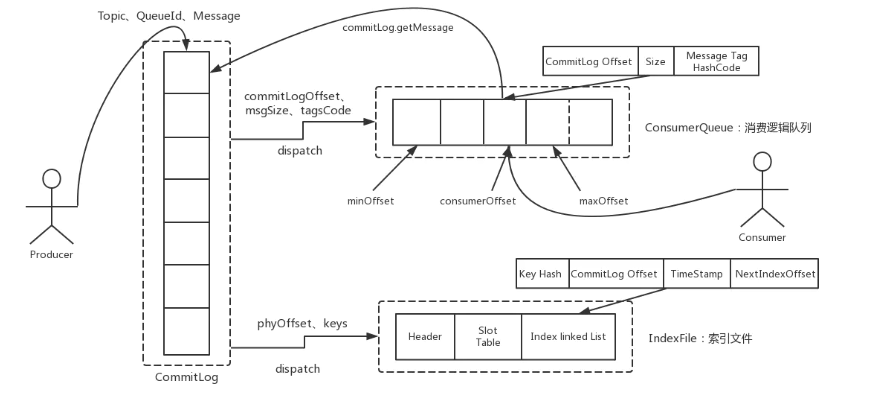
4、批量发送

5、事务发送

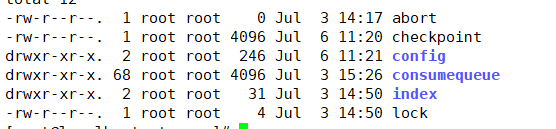
过滤消费方式：

1. tag 过滤消费 2、sql 过滤消费

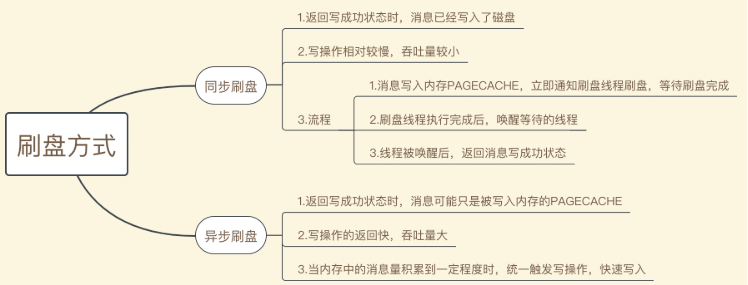
消息存储机制（将数据写到文件：利用顺序写，读：零拷贝提升效率）：



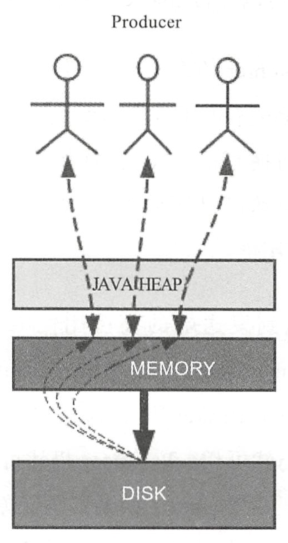
存储数据文件结构：



刷盘方式： （刷盘：即将内存数据写到磁盘文件）



同步刷盘 异步刷盘

主从数据同步方式：

1. 同步
2. 异步