# 1、RocketMQ

## 1.1、官方下载地址

<http://rocketmq.apache.org/dowloading/releases/>

使用最新的4.8的版本。

4.8较之前二期讲的4.4的版本整体上没有变化。

从[Release Notes](http://rocketmq.apache.org/release_notes/release-notes-4.8.0)来看，主要都是一些漏洞的修复和小范围的改进，所以整体上阅读源码上也没多少影响。



## 1.2、环境要求

* Linux64位系统
* JDK1.8(64位)
* 源码安装需要安装Maven 3.2.x

## 1.3、目录介绍

* bin：启动脚本，包括shell脚本和CMD脚本
* conf：实例配置文件 ，包括broker配置文件、logback配置文件等
* lib：依赖jar包，包括Netty、commons-lang、FastJSON等

# 2、RocketMQ的windows下的安装

在windows上安装RocketMQ一般只作为新手入门玩一玩。

因为RocketMQ的存储设计，包括分页设计，设计上都是基于Linux上的。所以生产服务器推荐使用Linux.

## 2.1、解压及运行前环境

使用运行版本(Binary)

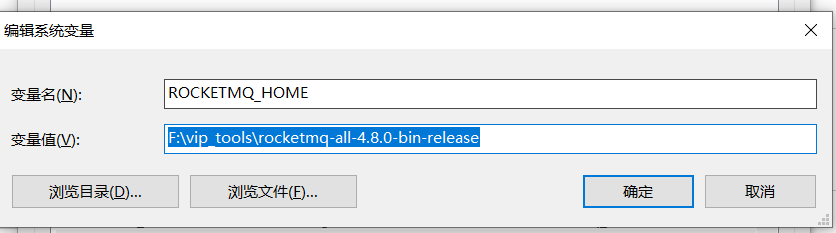
运行前确保：已经有jdk1.8



## 2.2、配置环境变量

变量名：ROCKETMQ\_HOME

 变量值：MQ解压路径\MQ文件夹名



## 2.3、注意事项

RocketMQ默认的虚拟机内存较大，启动Broker如果因为内存不足失败，需要编辑如下两个配置文件，修改JVM内存大小。

但是这个也仅仅是在测试环境中，RocketMQ在生产上最低要求至少8G内存（官方推荐）才能确保RocketMQ的效果

编辑runbroker.sh和runserver.sh修改默认JVM大小（windows上对应cmd文件）

vi runbroker.sh --broker的配置

vi runserver.sh --nameServer的配置

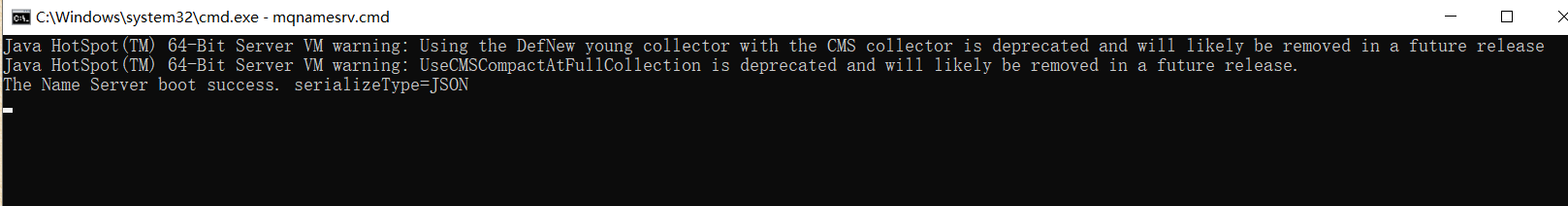
JAVA\_OPT="${JAVA\_OPT} -server -Xms512m -Xmx512m -Xmn256m -XX:MetaspaceSize=128m -XX:MaxMetaspaceSize=320m"

## 2.4、启动

RocketMQ的物理架构中，都是需要先启动NameServer再启动Broker的。所以启动顺序一定不要搞反了。

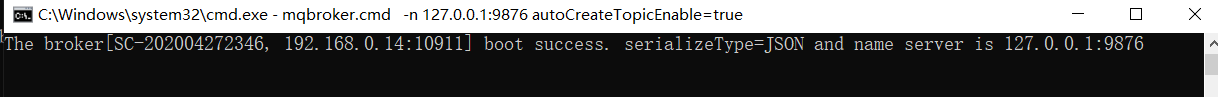
### 1.启动NAMESERVER

     Cmd命令框执行进入至‘MQ文件夹\bin’下，然后执行‘start mqnamesrv.cmd’，启动NAMESERVER。成功后会弹出提示框，此框勿关闭。

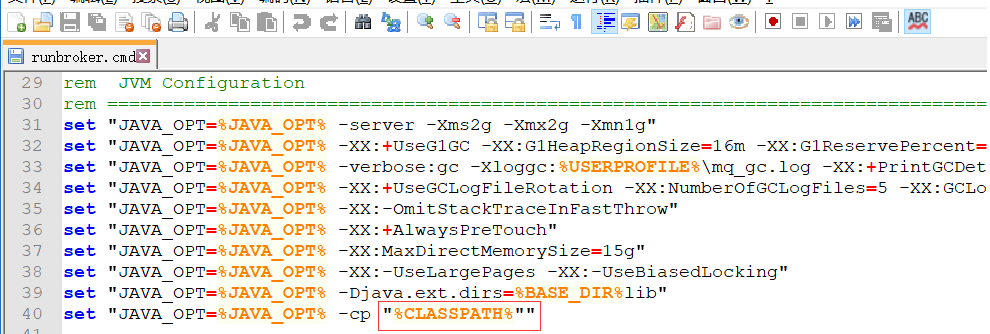


### 2.启动BROKER

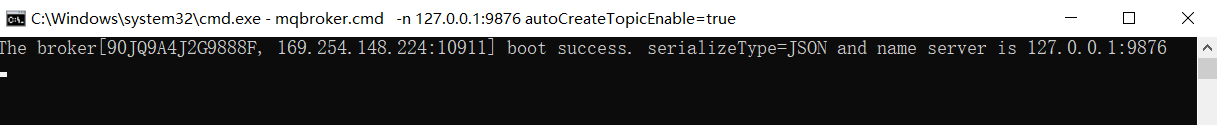
     Cmd命令框执行进入至‘MQ文件夹\bin’下，然后执行‘start mqbroker.cmd -n 127.0.0.1:9876 autoCreateTopicEnable=true’，启动BROKER。成功后会弹出提示框，此框勿关闭。



假如弹出提示框提示‘错误: 找不到或无法加载主类 xxxxxx’。打开runbroker.cmd，然后将‘%CLASSPATH%’加上英文双引号。保存并重新执行start语句。



再次启动

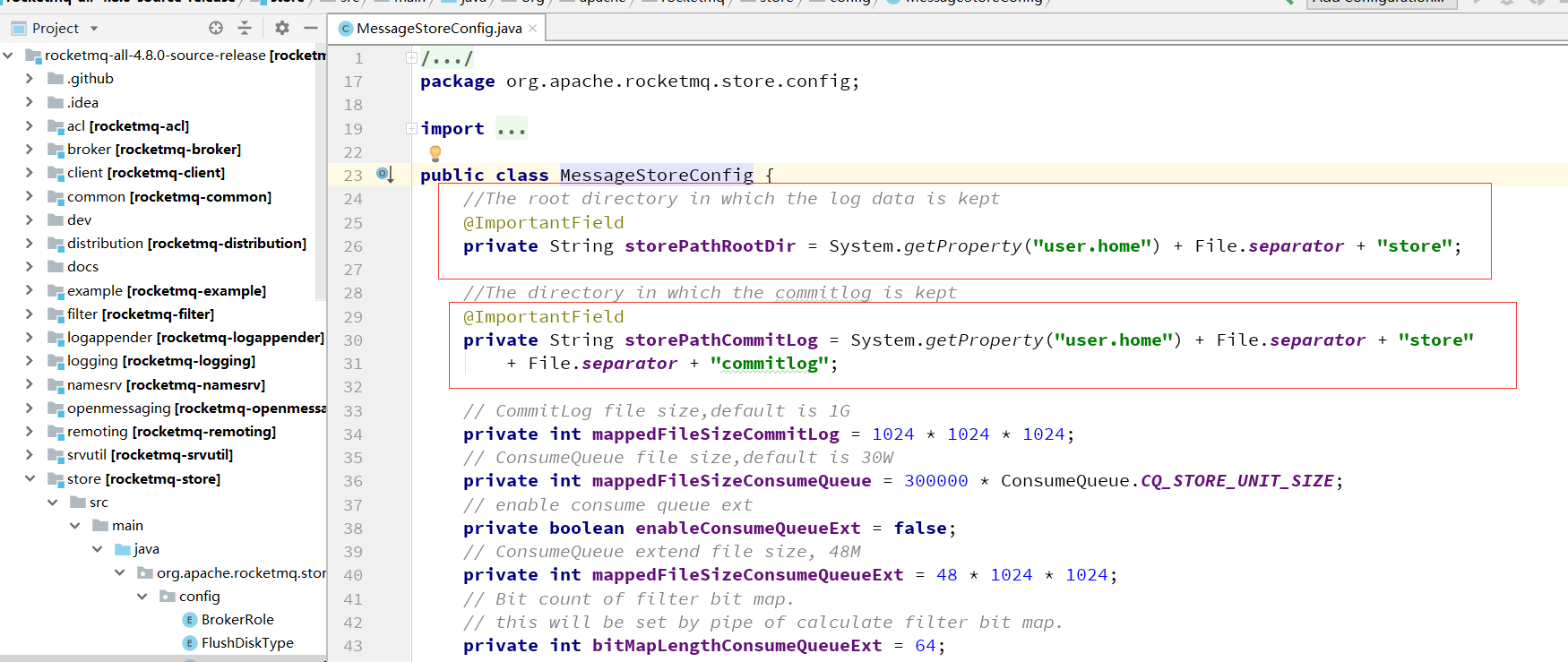


## 2.5、存储

rocketmq取的默认路径是user.home路径，也就是用户的根目录，一般存储放在了路径下的 /store目录。



源码中可以得到验证，如下图：



所以这里会有一个问题，RocketMQ很容易导致C盘空间不够，在使用过程中，创建一个主题默认就是要创建1G的文件，很可能会导致出问题。

所以在windows上容易导致C盘空间吃满。

解决方式有两种：

1、修改源码，比如：全局替换user.home参数为mq.store，然后重新打包

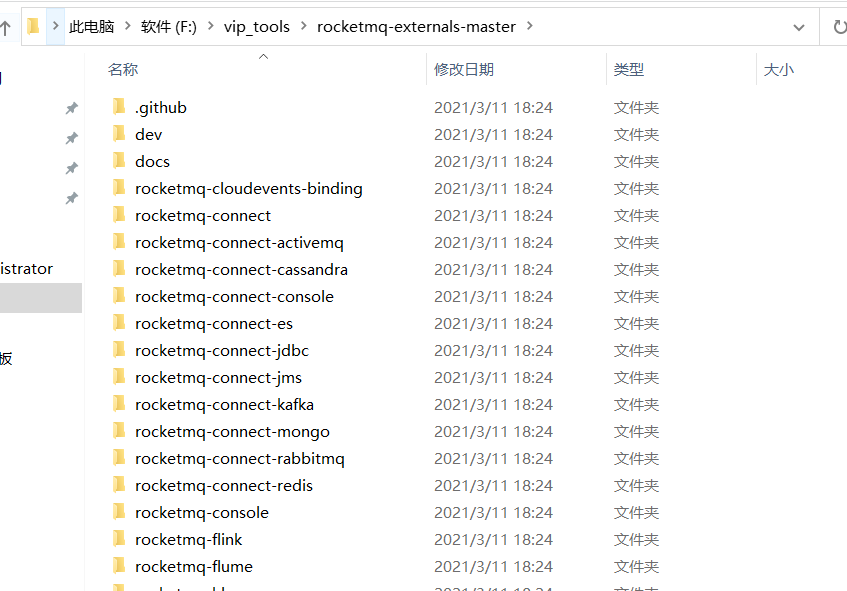
2、使用源码方式启动，源码启动时通过参数设置指定存储位置

## 2.6、安装可视化插件

运行前确保：已经有jdk1.8，Maven(打包需要安装Maven 3.2.x)

下载：<https://codeload.github.com/apache/rocketmq-externals/zip/master>

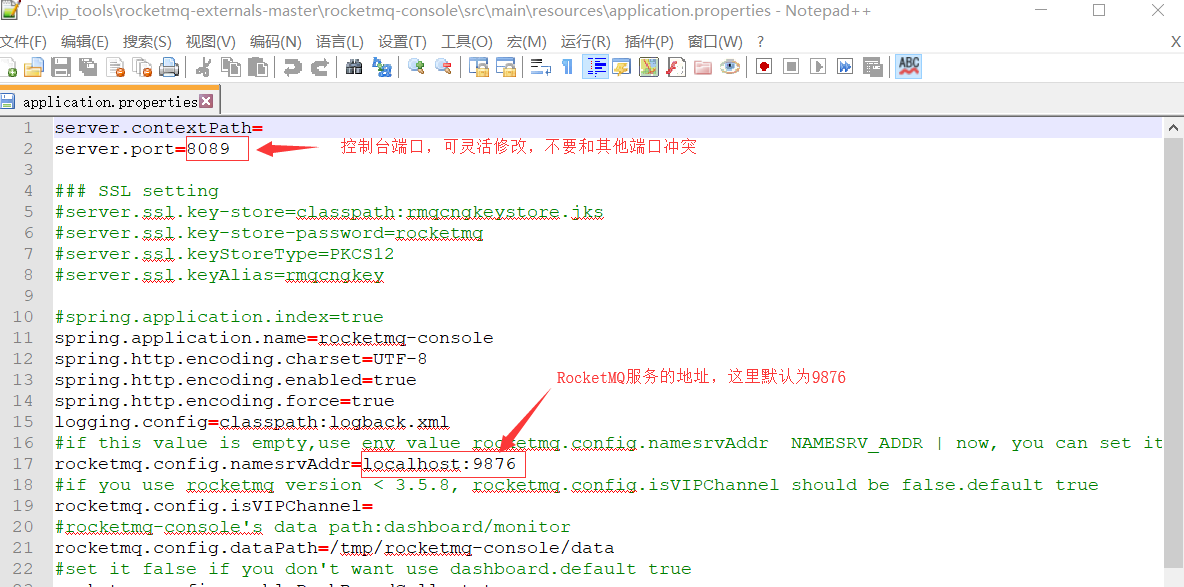
解压： 如果是之前使用的是二期，我也建议同步更新下，使用最新的。



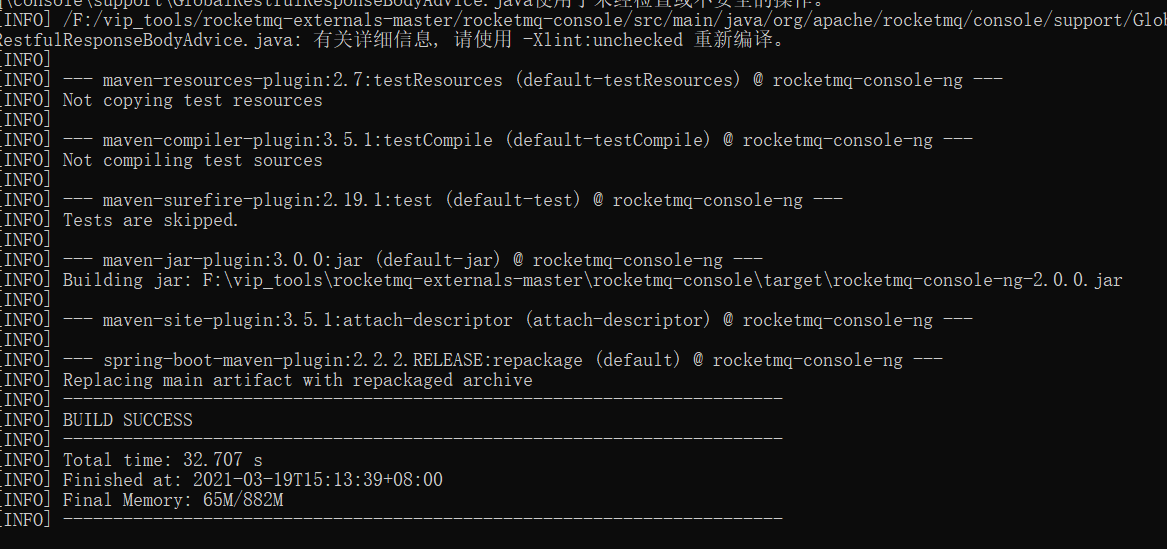
这个包主要包含的是Message Connector，具体详情见<https://rocketmq-1.gitbook.io/rocketmq-connector/>

里面后端管理界面是：rocketmq-console

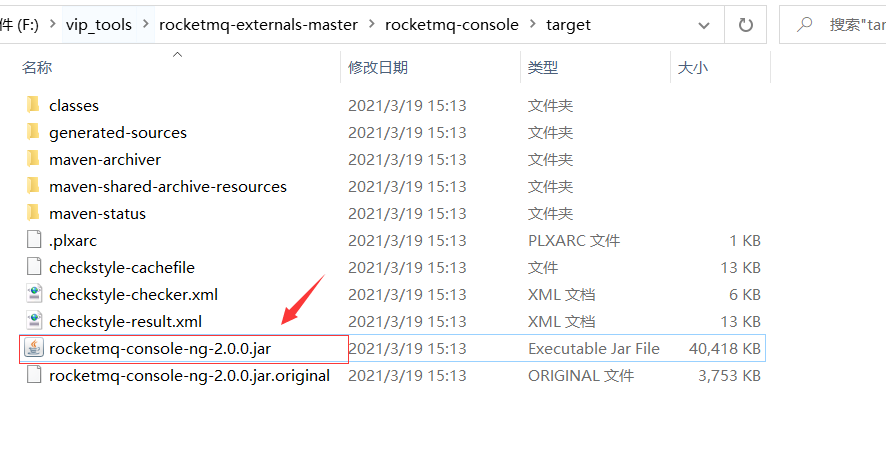
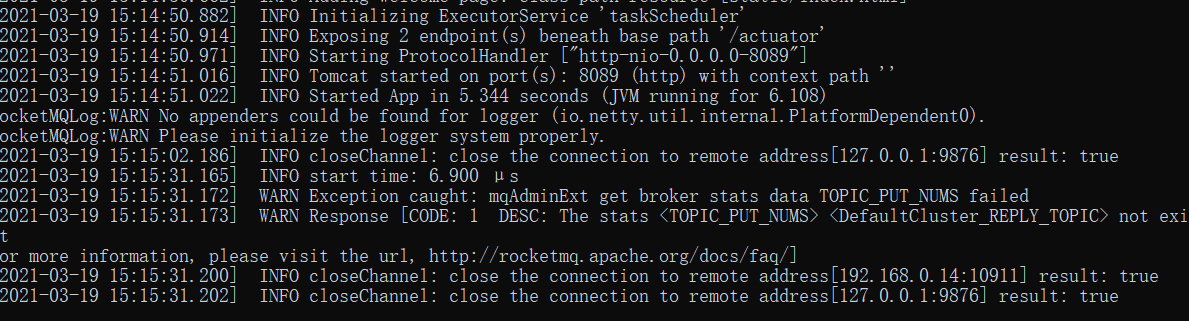
下载完成之后，进入‘\rocketmq-console\src\main\resources’文件夹，打开‘application.properties’进行配置。



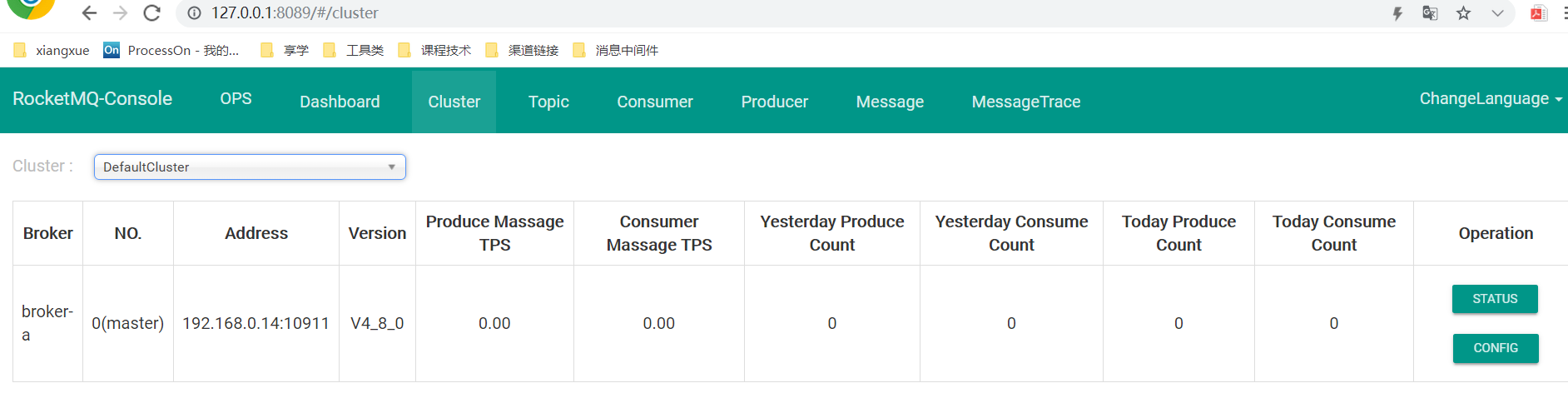
进入‘\rocketmq-externals\rocketmq-console’文件夹，执行‘mvn clean package -Dmaven.test.skip=true’，编译生成。



编译成功之后，Cmd进入‘target’文件夹，执行‘java -jar rocketmq-console-ng-2.0.0.jar’，启动‘rocketmq-console-ng-2.0.0.jar’。

浏览器中输入‘127.0.0.1:8089’，成功后即可查看。



# 3、RocketMQ在Linux下的安装

RocketMQ在Linux安装更加简单

运行前确保：已经有jdk1.8

记得Linux上修改文件权限：命令如下：chmod -R 777 /home/linux

假设服务的外网IP地址：192.168.56.101

## 注意事项

**RocketMQ需要开通的端口：**

rocketMQ自身占用有9876

非vip通道端口:10911

~~vip通道端口:10909(只针对producer 而且4.5以后已经默认不开启了)~~

10909是VIP通道对应的端口，在JAVA中的消费者对象或者是生产者对象中关闭VIP通道即可无需开放10909端口。

VIP通道其实就是多监听一个端口用于接受处理消息，因为默认端口通道可能很多在用，为了防止某些很重要的业务堵塞，就再开一个端口处理。这对于老版本的 RocketMQ 有消息接收队列的时候，作用可能大一点，对于目前的 RocketMQ 的设计，作用没那么大了。所以，这个默认就不开启了，留着只是为了兼容老版本。

RocketMQ默认的虚拟机内存较大，启动Broker如果因为内存不足失败，需要编辑如下两个配置文件，修改JVM内存大小。

但是这个也仅仅是在测试环境中，RocketMQ在生产上最低要求至少8G内存（官方推荐）才能确保RocketMQ的效果

编辑runbroker.sh和runserver.sh修改默认JVM大小（windows上对应cmd文件）

vi runbroker.sh --broker的配置

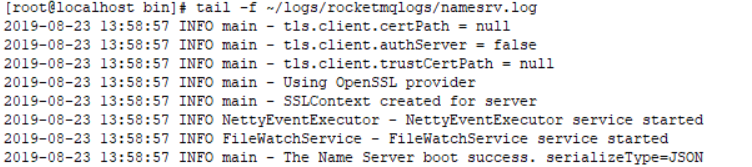
vi runserver.sh --nameServer的配置

JAVA\_OPT="${JAVA\_OPT} -server -Xms1024m -Xmx1024m -Xmn512m -XX:MetaspaceSize=128m -XX:MaxMetaspaceSize=320m"

## 3.1、 启动NAMESERVER

   进入至‘MQ文件夹\bin’下，然后执行‘nohup sh mqnamesrv &’，启动NAMESERVER。

查看日志的命令：tail -f ~/logs/rocketmqlogs/namesrv.log



## 3.2、 启动BROKER

进入至‘MQ文件夹\bin’下，启动BROKER。

修改配置文件增加外网地址(你启动加载哪个配置文件就修改哪个，这里修改broker.conf)

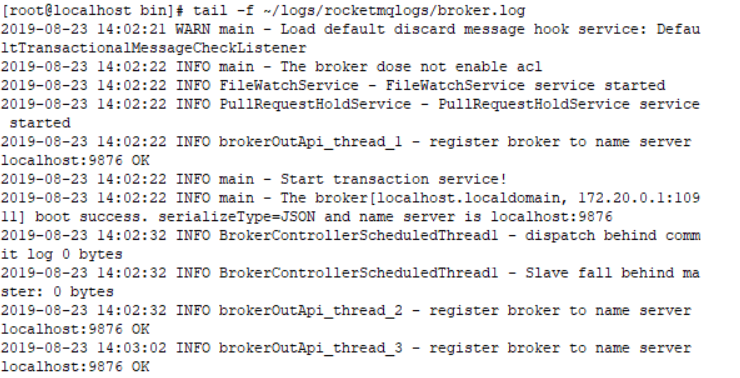
brokerIP1=192.168.56.101

启动命令如下：

nohup sh mqbroker -c ../conf/broker.conf -n 192.168.56.101:9876 autoCreateTopicEnable=true &

这样启动的服务器可以自动创建主题（客户端）,不过生产一般不推荐。

查看日志的命令：tail -f ~/logs/rocketmqlogs/broker.log



## 3.3、安装可视化插件

因为可视化插件是基于java打包的，所以启动过程和windows上是一样，修改端口号和打包可参考[安装可视化插件](#_安装可视化插件)

执行‘nohup java -jar rocketmq-console-ng-1.0.1.jar &’，启动‘rocketmq-console-ng-1.0.1.jar’。

# RocketMQ源码安装与调试

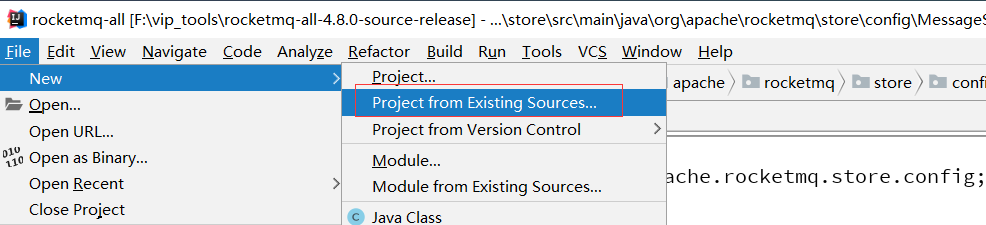
## 官方下载地址

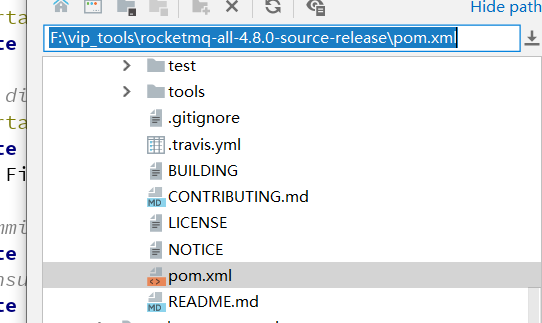
http://rocketmq.apache.org/dowloading/releases/

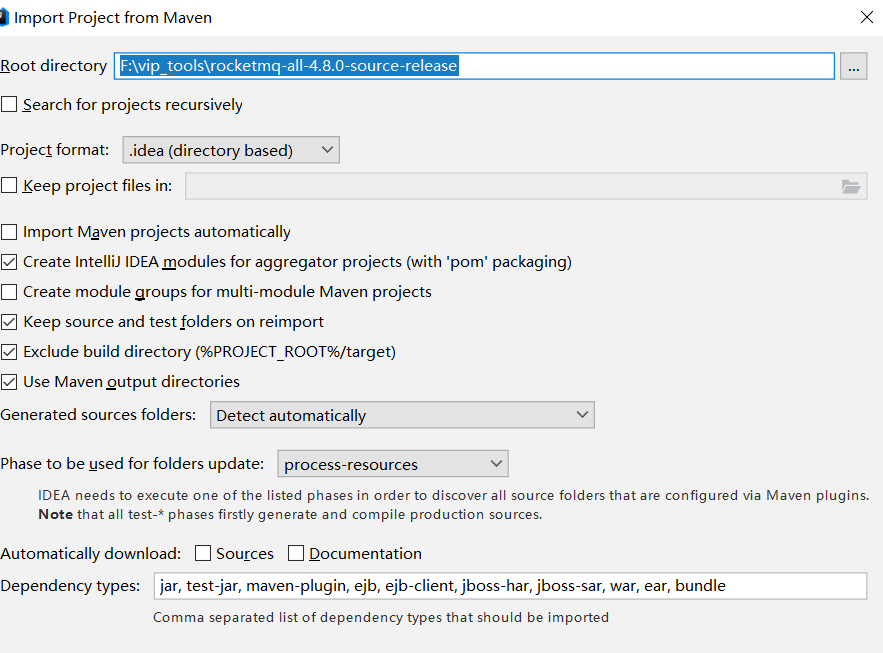
## 环境要求

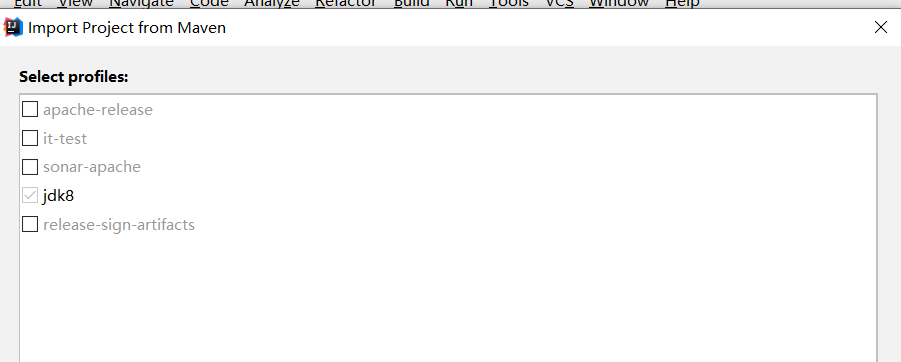
* Linux64位系统
* JDK1.8(64位)
* Maven 3.2.x

## IntelliJ IDEA导入

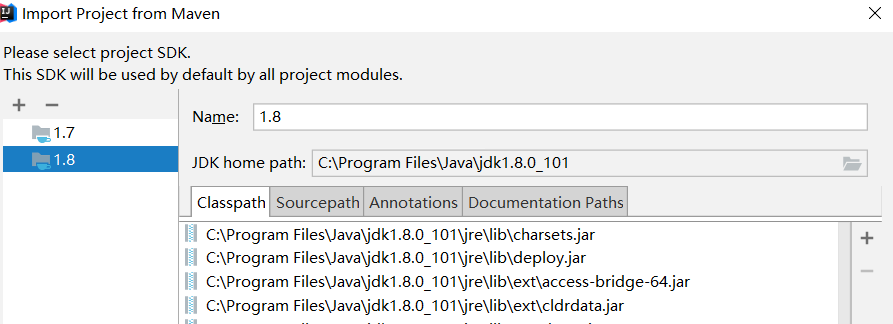


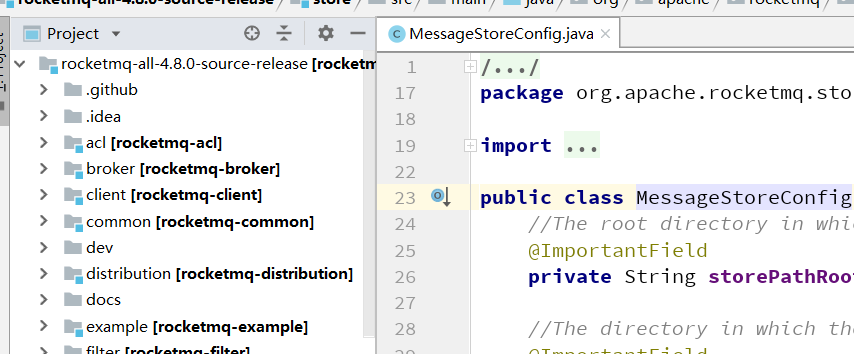






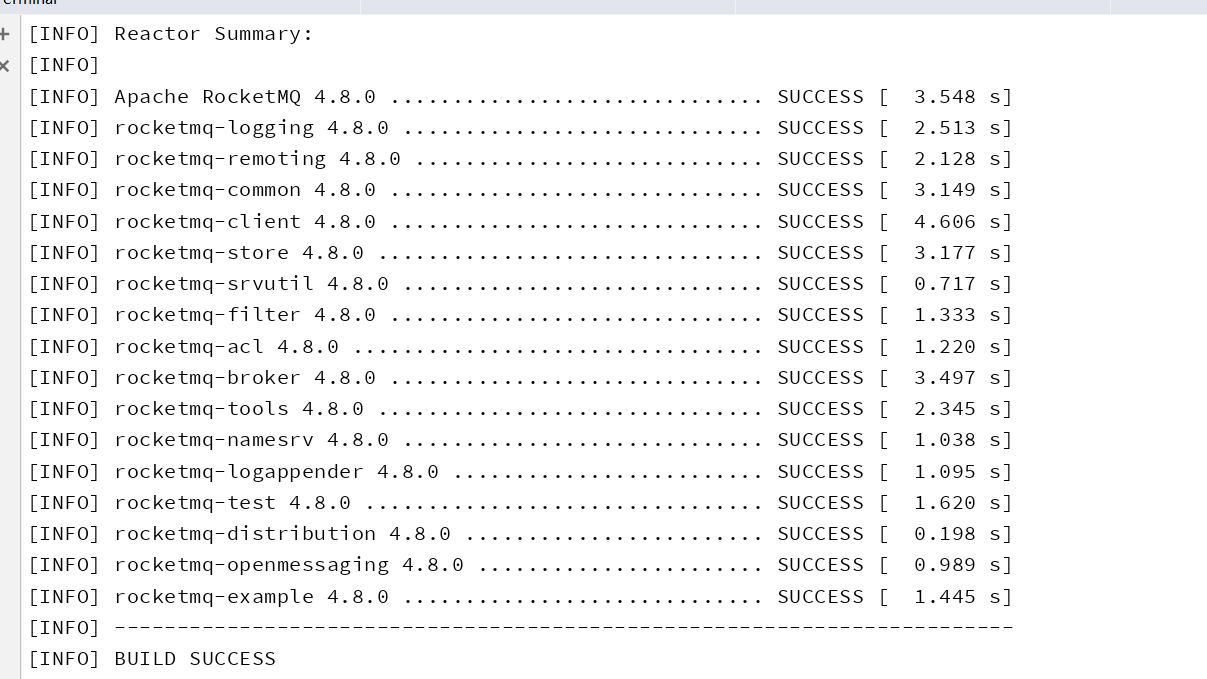






导入后执行Maven命令install

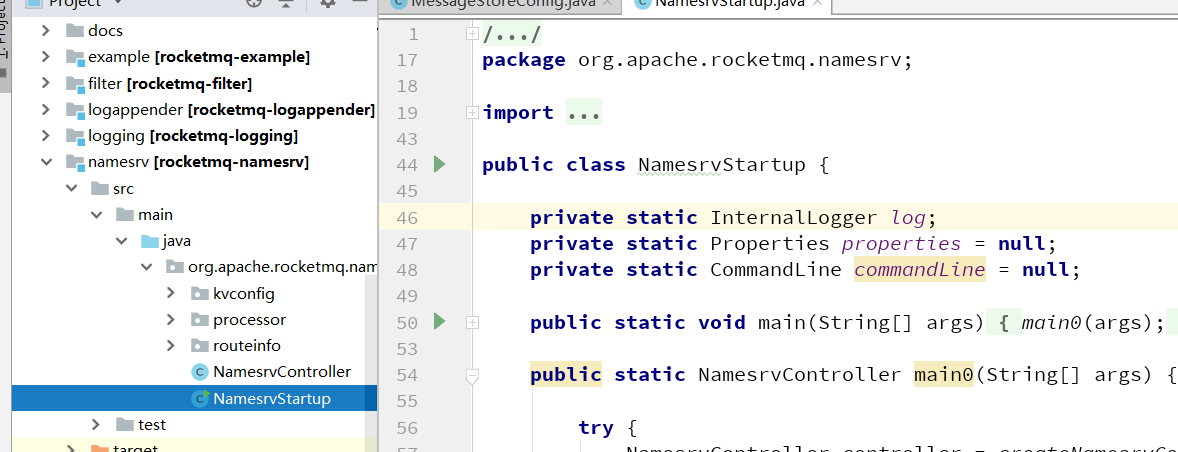
 mvn install -Dmaven.test.skip=true

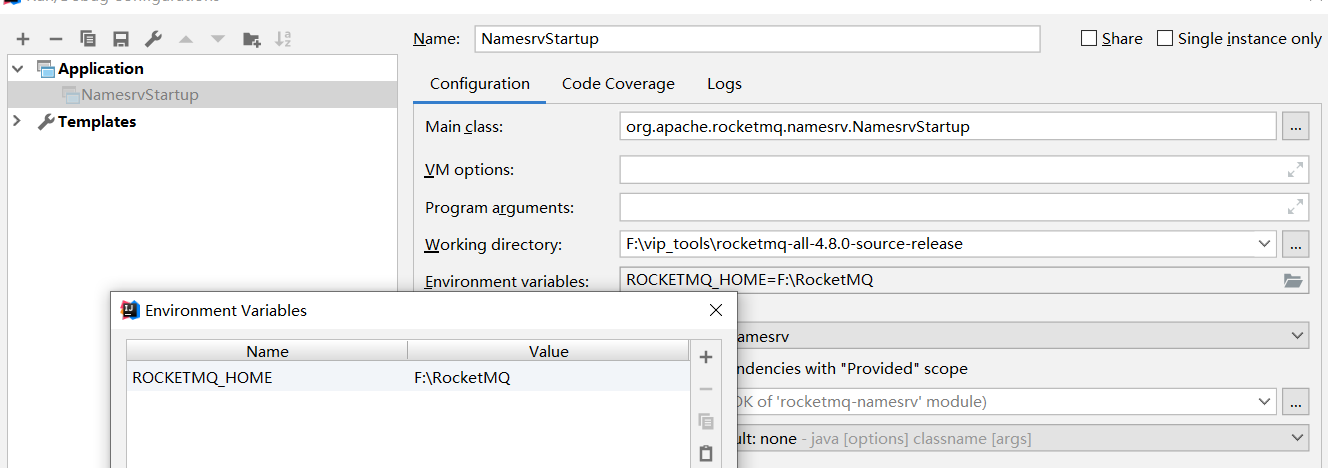


验证下没问题

## 调试RocketMQ源码

### 启动NameServer

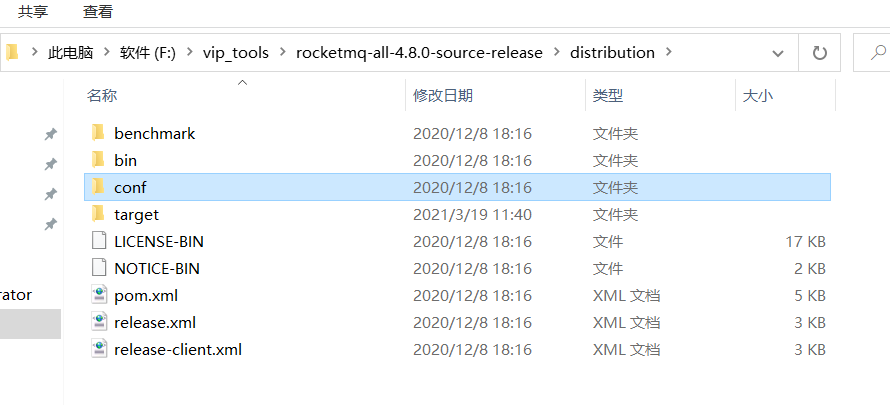


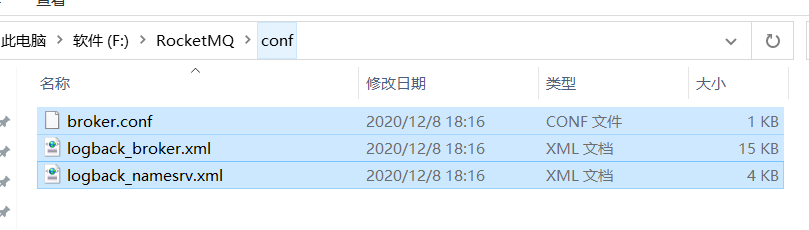


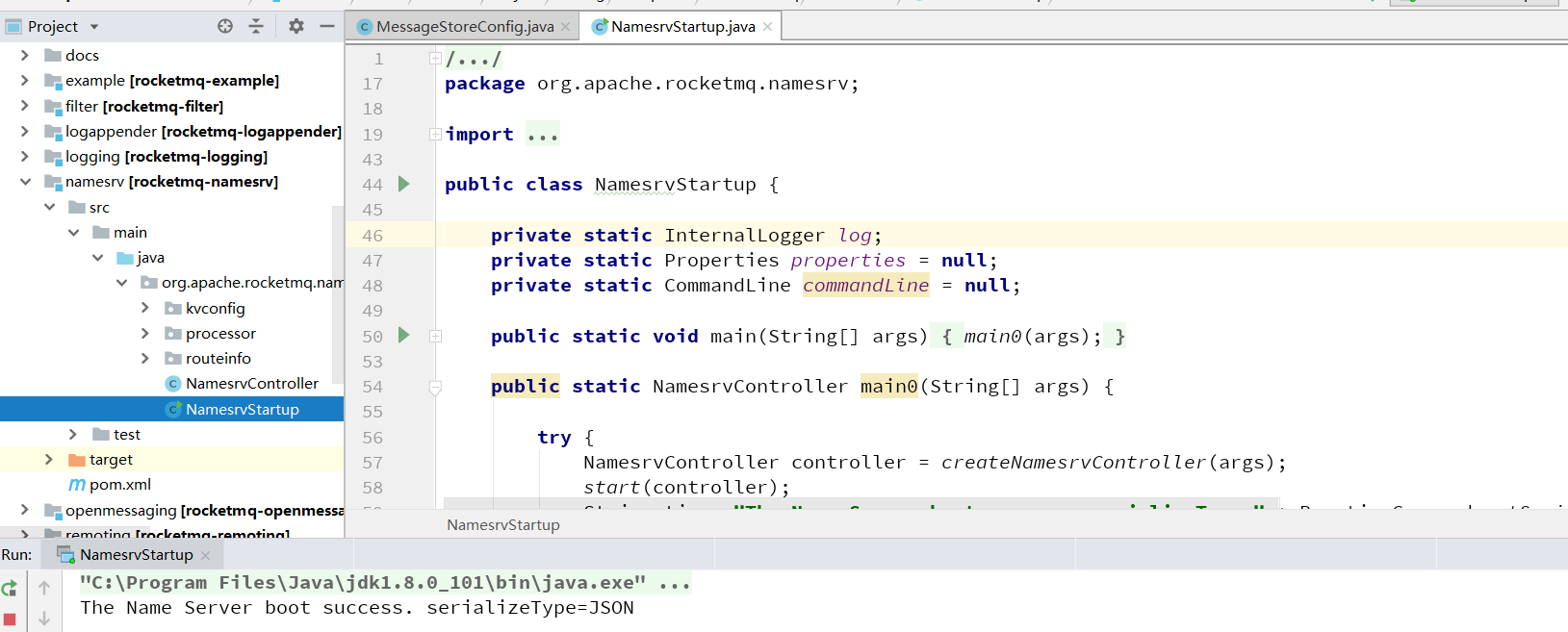
如上图，中Value值是一个Rocket运行主目录（一般这个目录新建）

在Rocket运行主目录中创建conf、logs、store三个文件夹

然后从源码目录中distribution目录下的中将broker.conf、logback\_broker.xml、logback\_namesrv.xml复制到conf目录中

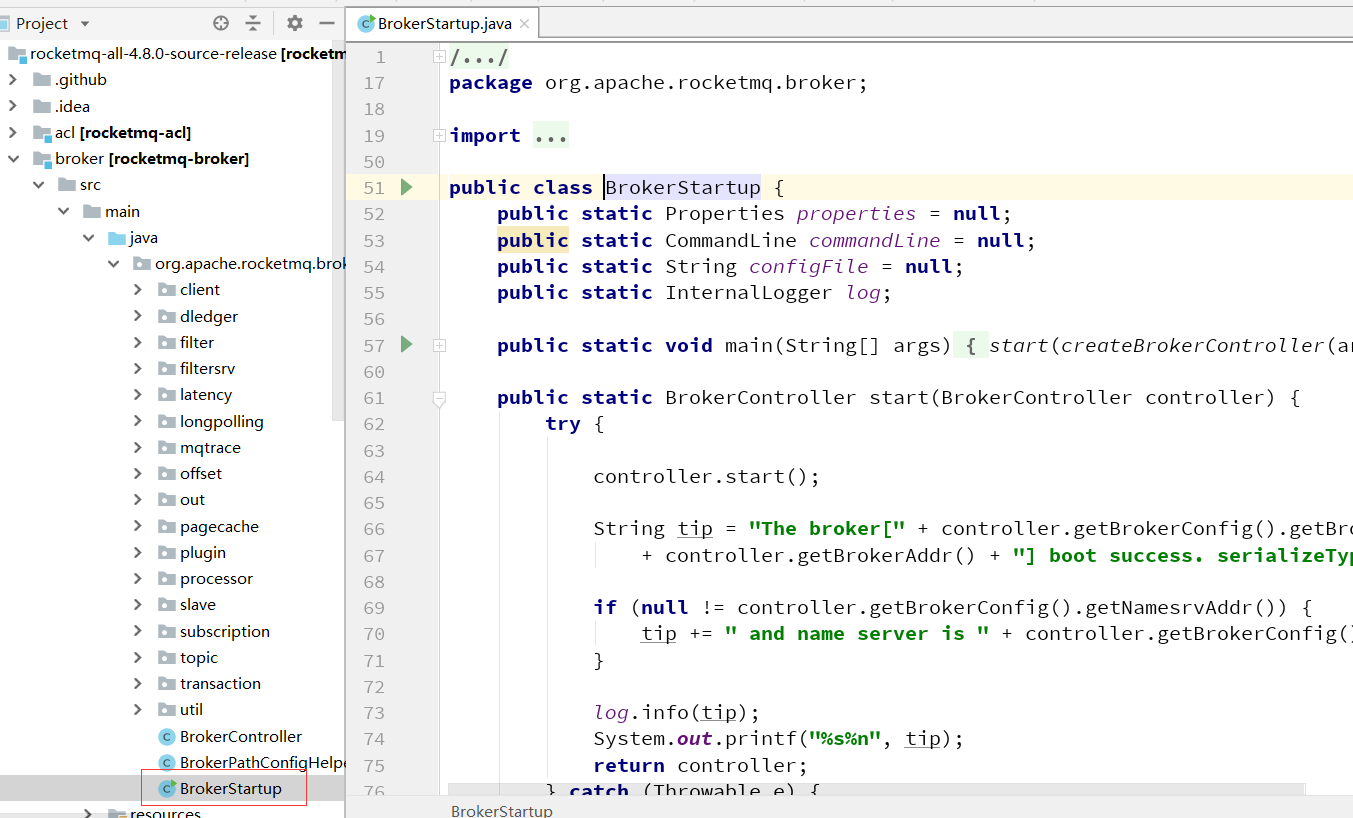




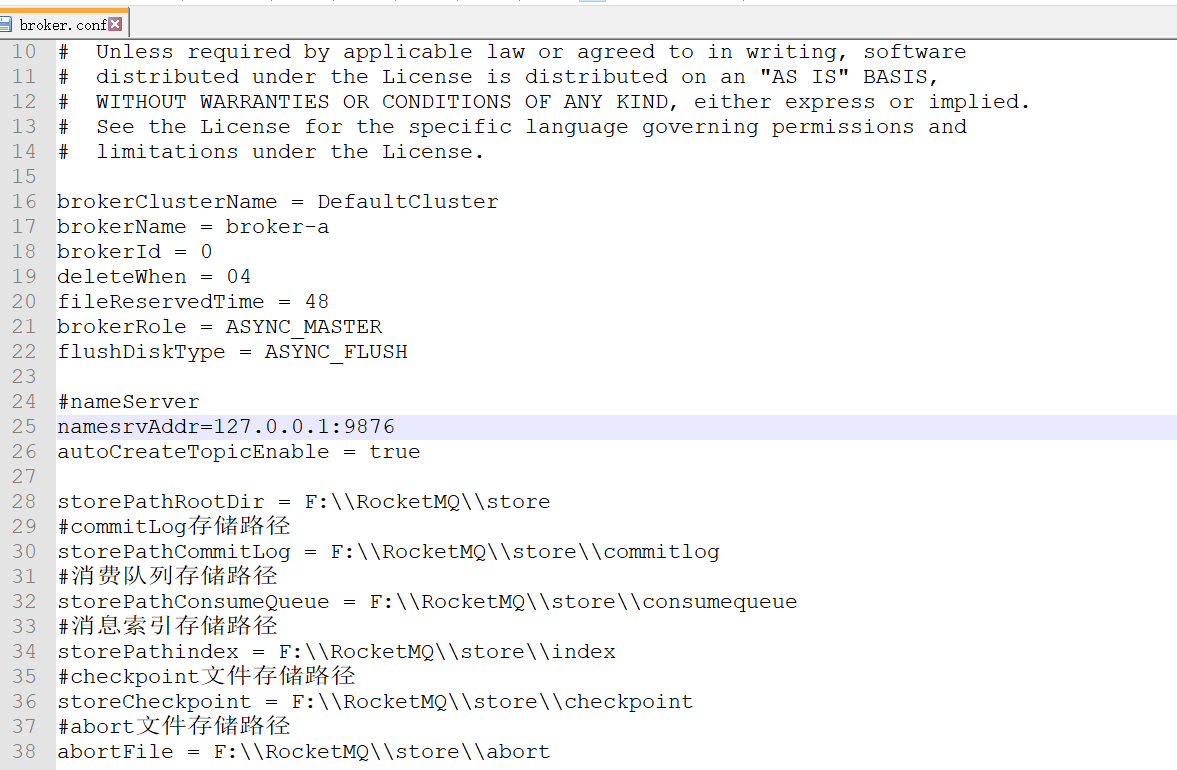


### 启动Broker

在broker模块找到broker模块，同时找到启动类BrokerStartup.java



#### 需要修改配置文件broker.conf



#nameServer

namesrvAddr=127.0.0.1:9876

autoCreateTopicEnable = true

storePathRootDir = F:\\RocketMQ\\store

#commitLog存储路径

storePathCommitLog = F:\\RocketMQ\\store\\commitlog

#消费队列存储路径

storePathConsumeQueue =F:\\RocketMQ\\store\\consumequeue

#消息索引存储路径

storePathindex = F:\\RocketMQ\\store\\index

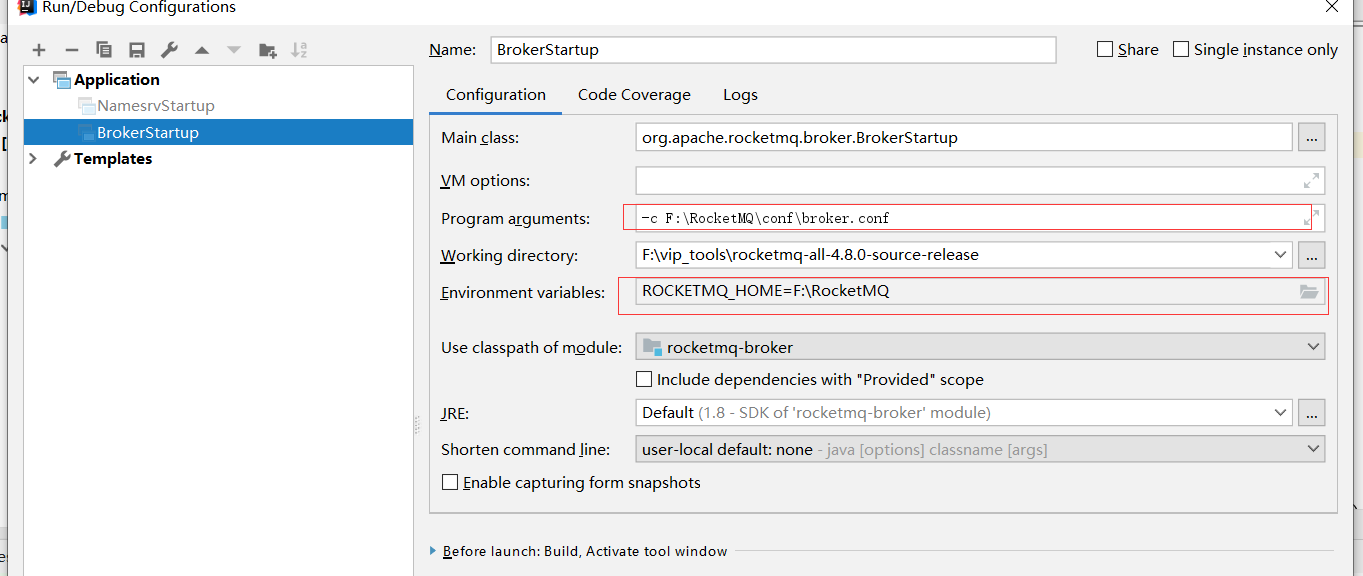
#checkpoint文件存储路径

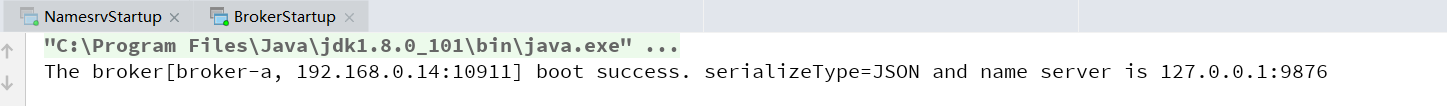
storeCheckpoint = F:\\RocketMQ\\store\\checkpoint

#abort文件存储路径

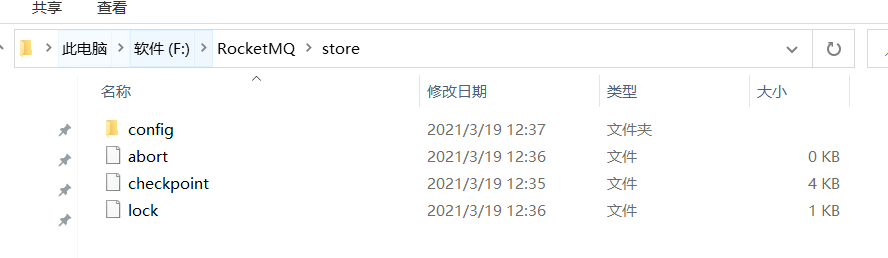
abortFile = F:\\RocketMQ\\store\\abort

#### 配置环境变量





#### 最后检查下数据文件



#### 启动的日志文件目录

启动过程中任何的日志信息已经写入

