

# 1、RocketMQ

## 1.1、官方下载地址

<http://rocketmq.apache.org/downloading/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 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。

### 4.8.0 release

- Released December 21, 2020
  - [Release Notes](#)
  - Source: [rocketmq-all-4.8.0-source-release.zip](#) [PGP] [SHA512]
  - Binary: [rocketmq-all-4.8.0-bin-release.zip](#) [PGP] [SHA512]
- 源码
- 运行版本

## 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)

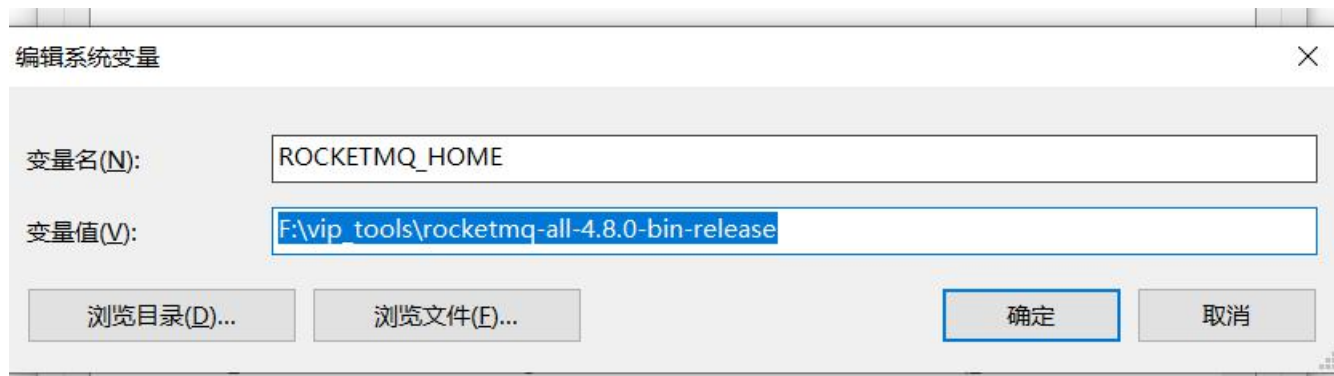
运行前确保: 已经有 jdk1.8

此电脑 > 软件 (F:) > vip_tools > rocketmq-all-4.8.0-bin-release >				
名称	修改日期	类型	大小	
benchmark	2020/12/9 19:46	文件夹		
bin	2020/12/4 14:26	文件夹		
conf	2020/12/4 14:26	文件夹		
lib	2020/12/9 19:46	文件夹		
LICENSE	2020/10/23 10:29	文件	17 KB	
NOTICE	2020/12/4 14:26	文件	2 KB	
README.md	2020/12/4 14:26	Markdown File	6 KB	

## 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

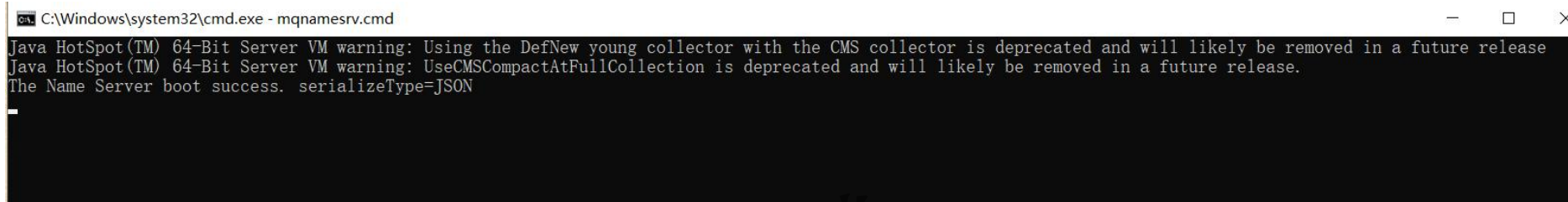
```
JAVA_OPT="${JAVA_OPT} -server -Xms512m -Xmx512m -Xmn256m -XX:MetaspaceSize=128m -XX:MaxMetaspaceSize=320m"
```

## 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

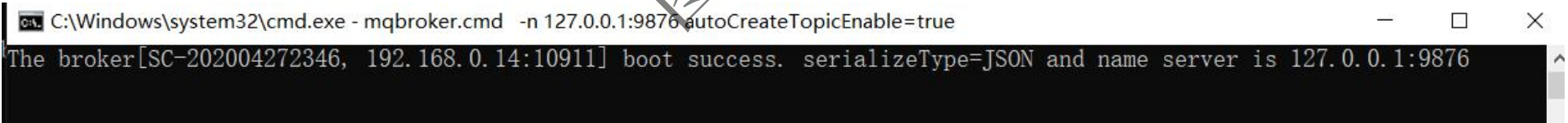
Cmd 命令框执行进入至'MQ 文件夹\bin'下，然后执行'start mqnamesrv.cmd'，启动 NAMESERVER。成功后会弹出提示框，此框勿关闭。



```
C:\Windows\system32\cmd.exe - mqnamesrv.cmd
Java HotSpot(TM) 64-Bit Server VM warning: Using the DefNew young collector with the CMS collector is deprecated and will likely be removed in a future release
Java HotSpot(TM) 64-Bit Server VM warning: UseCMSCompactAtFullCollection is deprecated and will likely be removed in a future release.
The Name Server boot success. serializeType=JSON
```

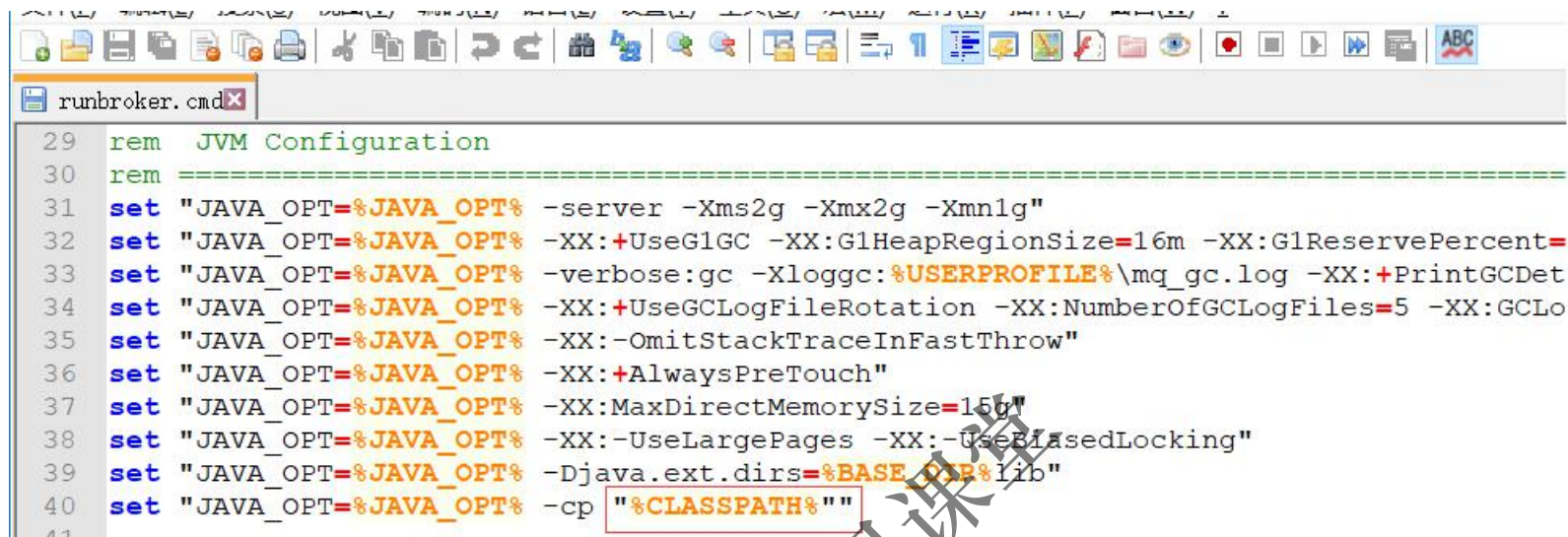
## 2.启动 BROKER

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。



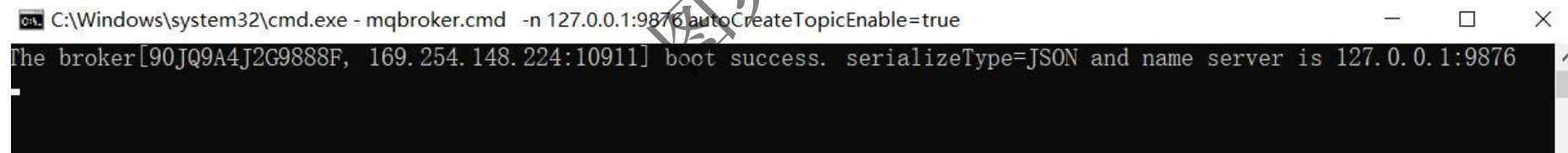
```
C:\Windows\system32\cmd.exe - mqbroker.cmd -n 127.0.0.1:9876 autoCreateTopicEnable=true
The broker[SC-202004272346, 192.168.0.14:10911] boot success. serializeType=JSON and name server is 127.0.0.1:9876
```

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。



```
29 rem JVM Configuration
30 rem =====
31 set "JAVA_OPTS=%JAVA_OPTS% -server -Xms2g -Xmx2g -Xmn1g"
32 set "JAVA_OPTS=%JAVA_OPTS% -XX:+UseG1GC -XX:G1HeapRegionSize=16m -XX:G1ReservePercent=
33 set "JAVA_OPTS=%JAVA_OPTS% -verbose:gc -Xloggc:%USERPROFILE%\mq_gc.log -XX:+PrintGCDet
34 set "JAVA_OPTS=%JAVA_OPTS% -XX:+UseGCLogFileRotation -XX:NumberOfGCLogFiles=5 -XX:GCLog
35 set "JAVA_OPTS=%JAVA_OPTS% -XX:-OmitStackTraceInFastThrow"
36 set "JAVA_OPTS=%JAVA_OPTS% -XX:+AlwaysPreTouch"
37 set "JAVA_OPTS=%JAVA_OPTS% -XX:MaxDirectMemorySize=150M"
38 set "JAVA_OPTS=%JAVA_OPTS% -XX:-UseLargePages -XX:-UseBiasedLocking"
39 set "JAVA_OPTS=%JAVA_OPTS% -Djava.ext.dirs=%BASE_DIR%\lib"
40 set "JAVA_OPTS=%JAVA_OPTS% -cp "%CLASSPATH%" "
```

再次启动



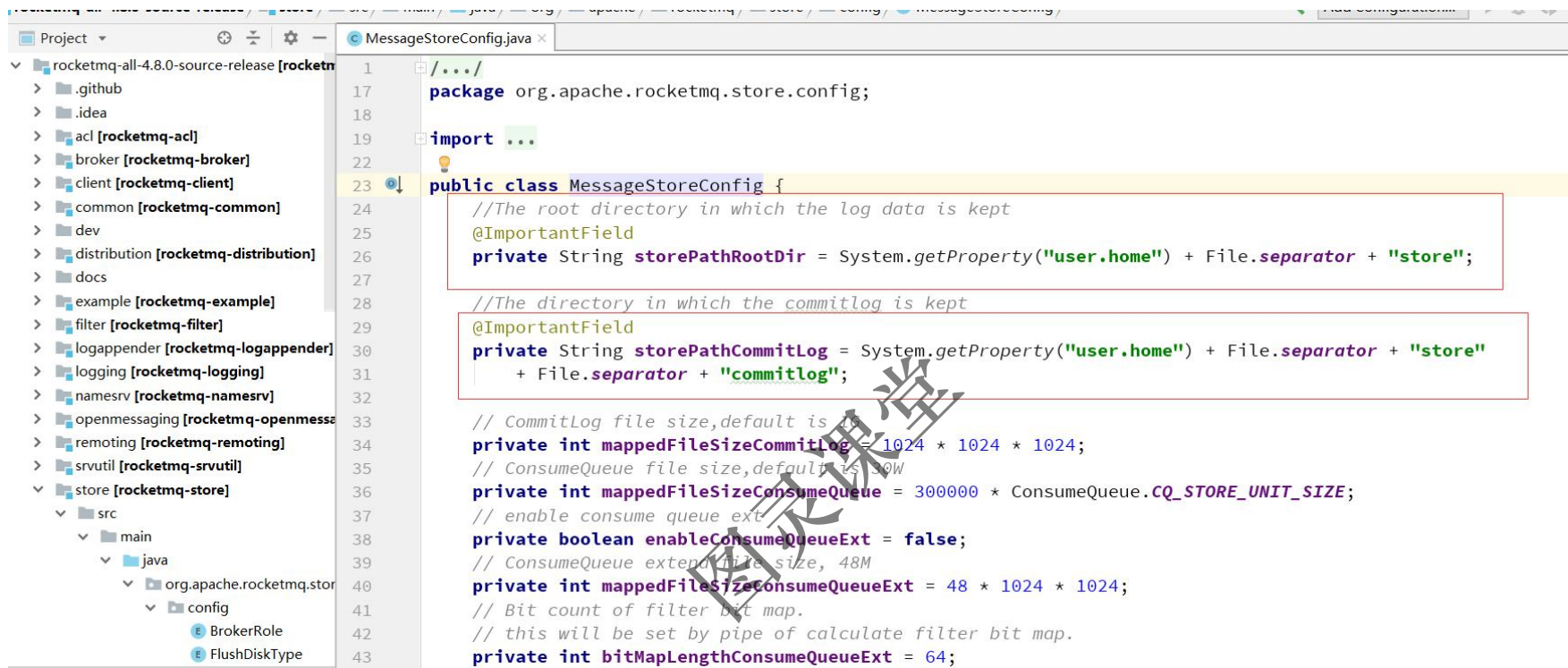
```
C:\Windows\system32\cmd.exe - mqbroker.cmd -n 127.0.0.1:9876 autoCreateTopicEnable=true
The broker[90JQ9A4J2G9888F, 169.254.148.224:10911] boot success. serializeType=JSON and name server is 127.0.0.1:9876
```

## 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。

图灵课堂

此电脑 > 软件 (F:) > vip_tools > rocketmq-externals-master				
	名称	修改日期	类型	大小
	.github	2021/3/11 18:24	文件夹	
	dev	2021/3/11 18:24	文件夹	
	docs	2021/3/11 18:24	文件夹	
	rocketmq-cloudevents-binding	2021/3/11 18:24	文件夹	
	rocketmq-connect	2021/3/11 18:24	文件夹	
	rocketmq-connect-activemq	2021/3/11 18:24	文件夹	
istrator	rocketmq-connect-cassandra	2021/3/11 18:24	文件夹	
	rocketmq-connect-console	2021/3/11 18:24	文件夹	
	rocketmq-connect-es	2021/3/11 18:24	文件夹	
版	rocketmq-connect-jdbc	2021/3/11 18:24	文件夹	
	rocketmq-connect-jms	2021/3/11 18:24	文件夹	
	rocketmq-connect-kafka	2021/3/11 18:24	文件夹	
	rocketmq-connect-mongo	2021/3/11 18:24	文件夹	
	rocketmq-connect-rabbitmq	2021/3/11 18:24	文件夹	
	rocketmq-connect-redis	2021/3/11 18:24	文件夹	
	rocketmq-console	2021/3/11 18:24	文件夹	
	rocketmq-flink	2021/3/11 18:24	文件夹	
	rocketmq-flume	2021/3/11 18:24	文件夹	

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。

D:\vip\_tools\rocketmq-externals-master\rocketmq-console\src\main\resources\application.properties - Notepad++

文件(F) 编辑(E) 搜索(S) 视图(V) 编码(N) 语言(L) 设置(T) 工具(O) 宏(M) 运行(R) 插件(P) 窗口(W) ?

application.properties

```
1 server.contextPath=
2 server.port=8089
3
4 ### SSL setting
5 #server.ssl.key-store=classpath:rmqcnsgkeystore.jks
6 #server.ssl.key-store-password=rocketmq
7 #server.ssl.keyStoreType=PKCS12
8 #server.ssl.keyAlias=rmqcnsgkey
9
10 #spring.application.index=true
11 spring.application.name=rocketmq-console
12 spring.http.encoding.charset=UTF-8
13 spring.http.encoding.enabled=true
14 spring.http.encoding.force=true
15 logging.config=classpath:logback.xml
16 #if this value is empty, use env value rocketmq.config.namesrvAddr NAMESRV_ADDR | now, you can set it
17 rocketmq.config.namesrvAddr=localhost:9876
18 #if you use rocketmq version < 3.5.8, rocketmq.config.isVIPChannel should be false.default true
19 rocketmq.config.isVIPChannel=
20 #rocketmq-console's data path:dashboard/monitor
21 rocketmq.config.dataPath=/tmp/rocketmq-console/data
22 #set it false if you don't want use dashboard.default true
```

控制台端口，可灵活修改，不要和其他端口冲突

RocketMQ服务的地址，这里默认为9876

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

```
[INFO] /F:/vip_tools/rocketmq-externals-master/rocketmq-console/src/main/java/org/apache/rocketmq/console/support/Globa
testfulResponseBodyAdvice.java: 有关详细信息, 请使用 -Xlint:unchecked 重新编译。
[INFO] --- maven-resources-plugin:2.7:testResources (default-testResources) @ rocketmq-console-ng ---
[INFO] Not copying test resources
[INFO] --- maven-compiler-plugin:3.5.1:testCompile (default-testCompile) @ rocketmq-console-ng ---
[INFO] Not compiling test sources
[INFO] --- maven-surefire-plugin:2.19.1:test (default-test) @ rocketmq-console-ng ---
[INFO] Tests are skipped.
[INFO] --- maven-jar-plugin:3.0.0:jar (default-jar) @ rocketmq-console-ng ---
[INFO] Building jar: F:\vip_tools\rocketmq-externals-master\rocketmq-console\target\rocketmq-console-ng-2.0.0.jar
[INFO] --- maven-site-plugin:3.5.1:attach-descriptor (attach-descriptor) @ rocketmq-console-ng ---
[INFO] --- spring-boot-maven-plugin:2.2.2.RELEASE:repackage (default) @ rocketmq-console-ng ---
[INFO] Replacing main artifact with repackaged archive
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 32.707 s
[INFO] Finished at: 2021-03-19T15:13:39+08:00
[INFO] Final Memory: 65M/882M
[INFO] -----
```

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'。

件 (F:) > vip\_tools > rocketmq-externals-master > rocketmq-console > target

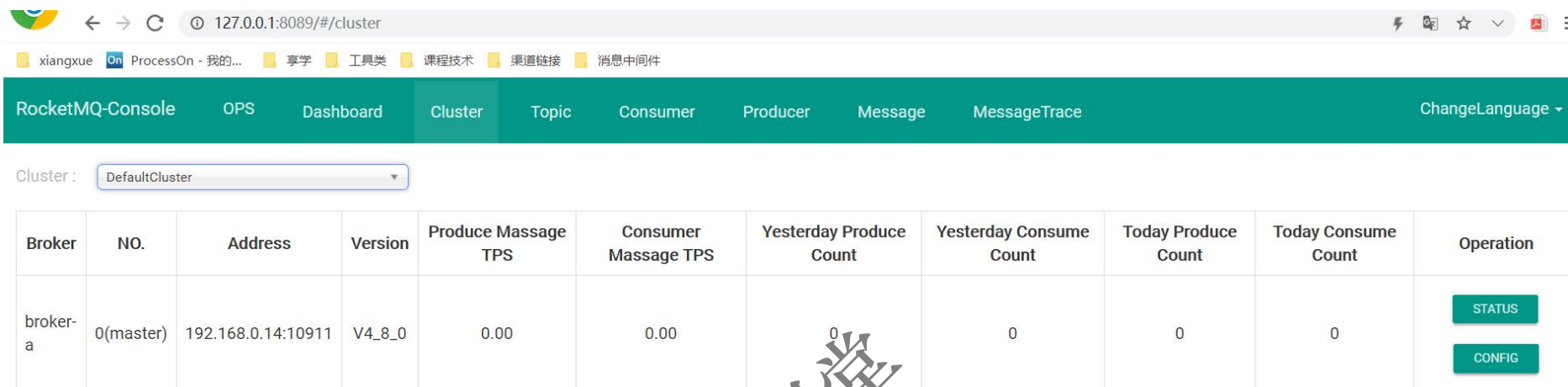
搜索"tar"

名称	修改日期	类型	大小
classes	2021/3/19 15:13	文件夹	
generated-sources	2021/3/19 15:13	文件夹	
maven-archiver	2021/3/19 15:13	文件夹	
maven-shared-archive-resources	2021/3/19 15:13	文件夹	
maven-status	2021/3/19 15:13	文件夹	
.plxarc	2021/3/19 15:13	PLXARC 文件	1 KB
checkstyle-cachefile	2021/3/19 15:13	文件	13 KB
checkstyle-checker.xml	2021/3/19 15:13	XML 文件	6 KB
checkstyle-result.xml	2021/3/19 15:13	XML 文件	13 KB
rocketmq-console-ng-2.0.0.jar	2021/3/19 15:13	Executable Jar File	40,418 KB
rocketmq-console-ng-2.0.0.jar.original	2021/3/19 15:13	ORIGINAL 文件	3,753 KB



```
2021-03-19 15:14:50.882] INFO Initializing ExecutorService 'taskScheduler'
2021-03-19 15:14:50.914] INFO Exposing 2 endpoint(s) beneath base path '/actuator'
2021-03-19 15:14:50.971] INFO Starting ProtocolHandler ["http-nio-0.0.0.0-8089"]
2021-03-19 15:14:51.016] INFO Tomcat started on port(s): 8089 (http) with context path ''
2021-03-19 15:14:51.022] INFO Started App in 5.344 seconds (JVM running for 6.108)
rocketmqLog:WARN No appenders could be found for logger (io.netty.util.internal.PlatformDependent0).
rocketmqLog:WARN Please initialize the logger system properly.
2021-03-19 15:15:02.186] INFO closeChannel: close the connection to remote address[127.0.0.1:9876] result: true
2021-03-19 15:15:31.165] INFO start time: 6.900 μs
2021-03-19 15:15:31.172] WARN Exception caught: mqAdminExt get broker stats data TOPIC_PUT_NUMS failed
2021-03-19 15:15:31.173] WARN Response [CODE: 1 DESC: The stats <TOPIC_PUT_NUMS> <DefaultCluster_REPLY_TOPIC> not exist
or more information, please visit the url, http://rocketmq.apache.org/docs/faq/]
2021-03-19 15:15:31.200] INFO closeChannel: close the connection to remote address[192.168.0.14:10911] result: true
2021-03-19 15:15:31.202] INFO closeChannel: close the connection to remote address[127.0.0.1:9876] result: true
```

浏览器中输入'127.0.0.1:8089'，成功后即可查看。



The screenshot shows the RocketMQ-Console web interface. The top navigation bar includes links for RocketMQ-Console, OPS, Dashboard, Cluster (selected), Topic, Consumer, Producer, Message, and MessageTrace. A 'Cluster' dropdown menu is set to 'DefaultCluster'. Below this is a table with 11 columns: Broker, NO., Address, Version, Produce Message TPS, Consumer Message TPS, Yesterday Produce Count, Yesterday Consume Count, Today Produce Count, Today Consume Count, and Operation. The table contains one row for broker 'broker-a' with address '192.168.0.14:10911' and version 'V4\_8\_0'. The 'Operation' column has two buttons: 'STATUS' and 'CONFIG'.

Broker	NO.	Address	Version	Produce Message TPS	Consumer Message TPS	Yesterday Produce Count	Yesterday Consume Count	Today Produce Count	Today Consume Count	Operation
broker-a	0(master)	192.168.0.14:10911	V4_8_0	0.00	0.00	0	0	0	0	<div>STATUS</div> <div>CONFIG</div>

### 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

```
[root@localhost bin]# tail -f ~/logs/rocketmqlogs/namesrv.log
2019-08-23 13:58:57 INFO main - tls.client.certPath = null
2019-08-23 13:58:57 INFO main - tls.client.authServer = false
2019-08-23 13:58:57 INFO main - tls.client.trustCertPath = null
2019-08-23 13:58:57 INFO main - Using OpenSSL provider
2019-08-23 13:58:57 INFO main - SSLContext created for server
2019-08-23 13:58:57 INFO NettyEventExecutor - NettyEventExecutor service started
2019-08-23 13:58:57 INFO FileWatchService - FileWatchService service started
2019-08-23 13:58:57 INFO main - The Name Server boot success. serializeType=JSON
```

### 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

```
[root@localhost bin]# tail -f ~/logs/rocketmqlogs/broker.log
2019-08-23 14:02:21 WARN main - Load default discard message hook service: DefaultTransactionalMessageCheckListener
2019-08-23 14:02:22 INFO main - The broker dose not enable acl
2019-08-23 14:02:22 INFO FileWatchService - FileWatchService service started
2019-08-23 14:02:22 INFO PullRequestHoldService - PullRequestHoldService service started
2019-08-23 14:02:22 INFO brokerOutApi_thread_1 - register broker to name server localhost:9876 OK
2019-08-23 14:02:22 INFO main - Start transaction service!
2019-08-23 14:02:22 INFO main - The broker[localhost.localdomain, 172.20.0.1:10911] boot success. serializeType=JSON and name server is localhost:9876
2019-08-23 14:02:32 INFO BrokerControllerScheduledThread1 - dispatch behind commit log 0 bytes
2019-08-23 14:02:32 INFO BrokerControllerScheduledThread1 - Slave fall behind master: 0 bytes
2019-08-23 14:02:32 INFO brokerOutApi_thread_2 - register broker to name server localhost:9876 OK
2019-08-23 14:03:02 INFO brokerOutApi_thread_3 - register broker to name server localhost:9876 OK
```

### 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](#)

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/downloading/releases/>

### 4.8.0 release

---

- Released December 21, 2020
- [Release Notes](#)
- Source: [rocketmq-all-4.8.0-source-release.zip](#) [PGP] [SHA512]
- Binary: [rocketmq-all-4.8.0-bin-release.zip](#) [PGP] [SHA512]

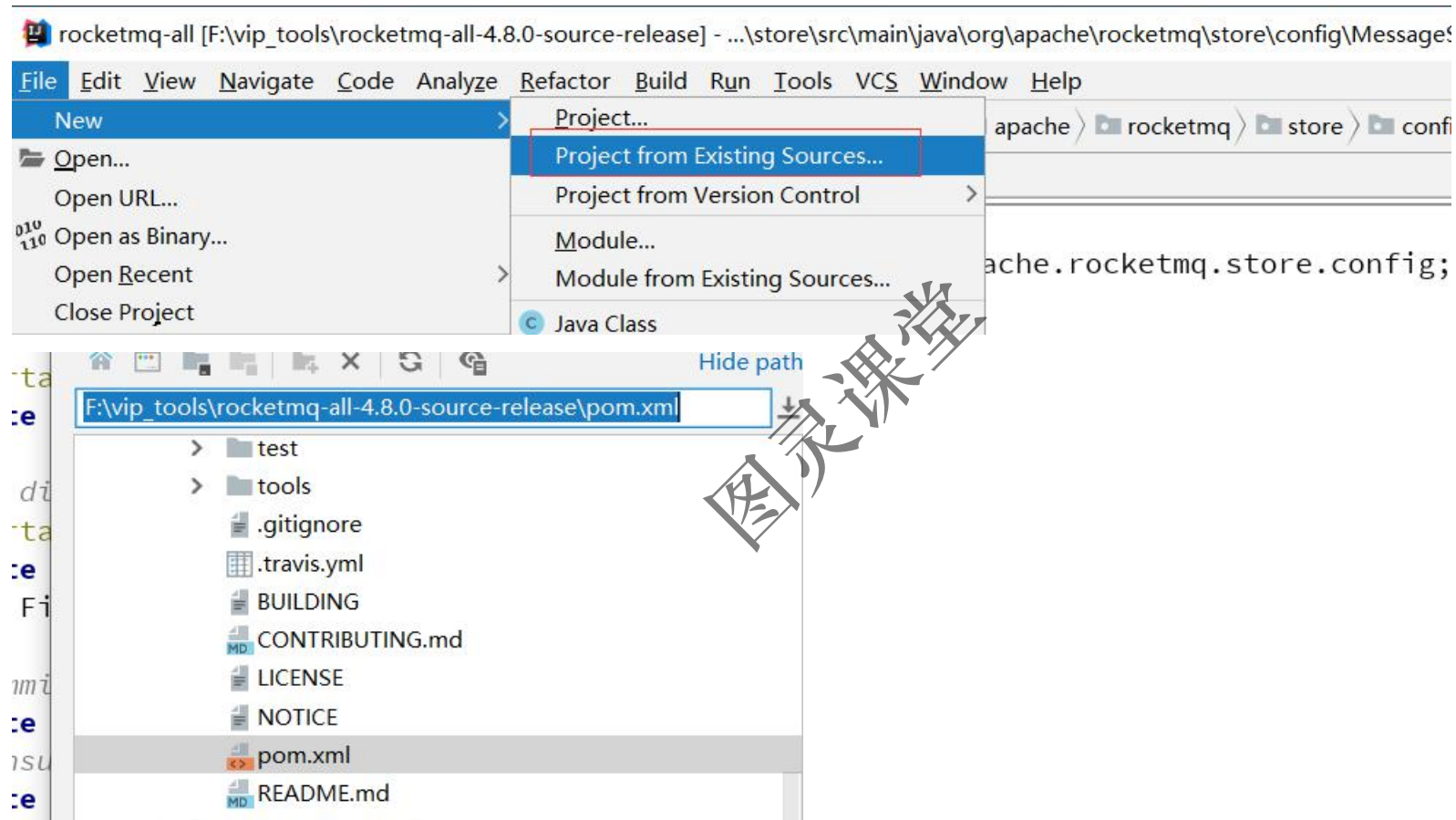
源码

运行版本

## 环境要求

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

## IntelliJ IDEA 导入



Import Project from Maven

Root directory

F:\vip\_tools\rocketmq-all-4.8.0-source-release

☐ Search for projects recursively

Project format: .idea (directory based)

☐ Keep project files in:

☐ Import Maven projects automatically

☒ Create IntelliJ IDEA modules for aggregator projects (with 'pom' packaging)

☐ Create module groups for multi-module Maven projects

☒ Keep source and test folders on reimport

☒ Exclude build directory (%PROJECT\_ROOT%/target)

☒ Use Maven output directories

Generated sources folders: Detect automatically

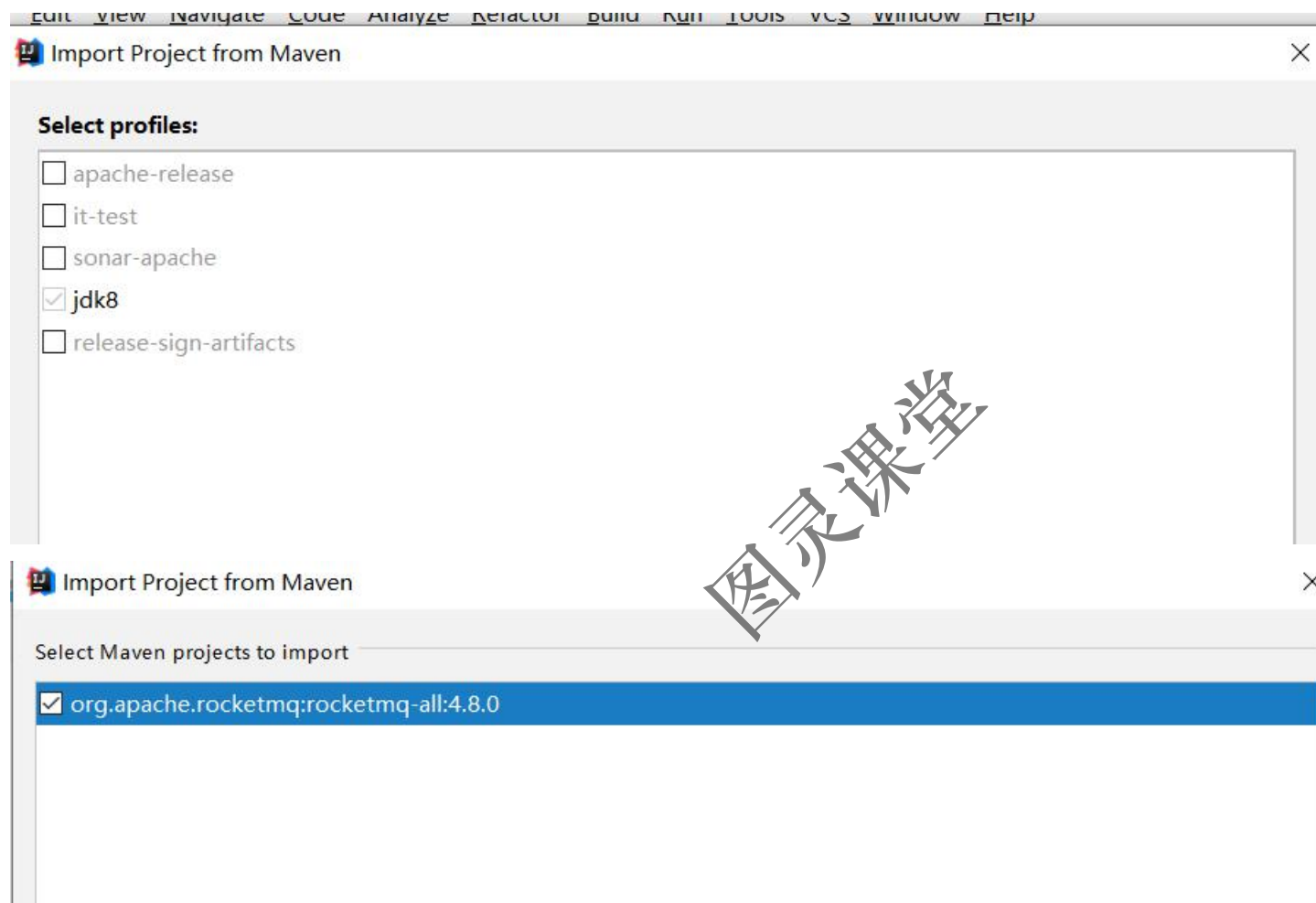
Phase to be used for folders update: process-resources

IDEA needs to execute one of the listed phases in order to discover all source folders that are configured via Maven plugins.  
**Note** that all test-\* phases firstly generate and compile production sources.

Automatically download: ☐ Sources ☐ Documentation

Dependency types: jar, test-jar, maven-plugin, ejb, ejb-client, jboss-har, jboss-sar, war, ear, bundle

Comma separated list of dependency types that should be imported



## Import Project from Maven

Please select project SDK.

This SDK will be used by default by all project modules.

+ -

1.7

1.8

Name: 1.8

JDK home path: C:\Program Files\Java\jdk1.8.0\_101

Classpath

Sourcepath

Annotations

Documentation Paths

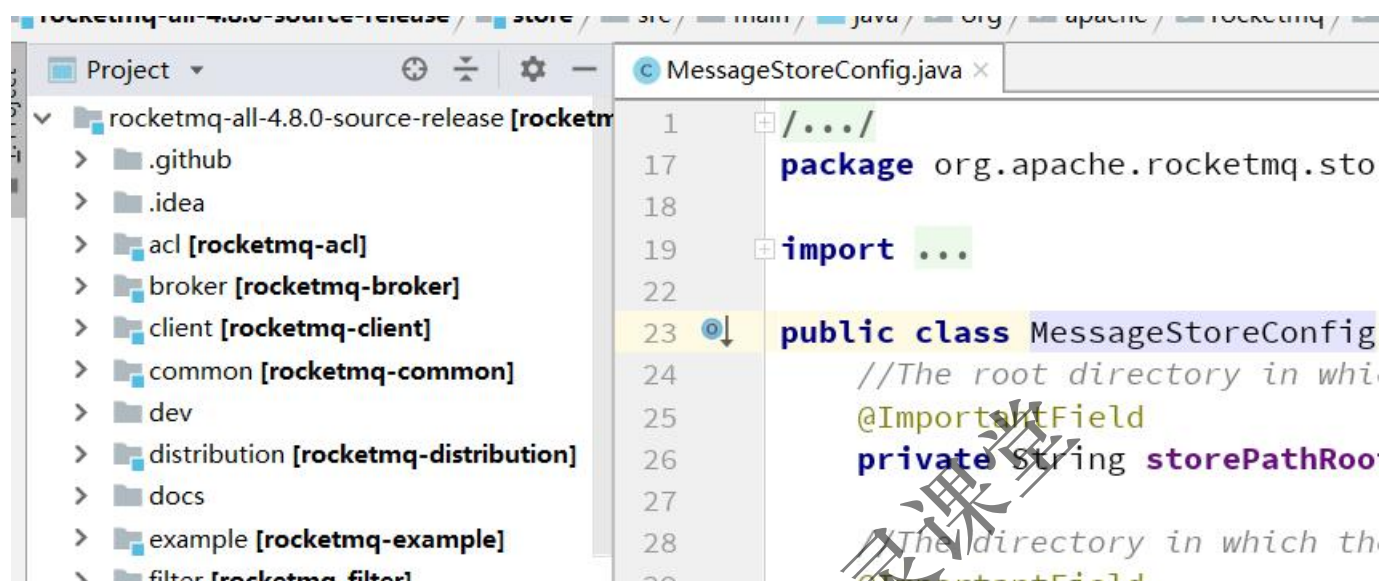
C:\Program Files\Java\jdk1.8.0\_101\jre\lib\charsets.jar

C:\Program Files\Java\jdk1.8.0\_101\jre\lib\deploy.jar

C:\Program Files\Java\jdk1.8.0\_101\jre\lib\ext\access-bridge-64.jar

C:\Program Files\Java\jdk1.8.0\_101\jre\lib\ext\clldata.jar





导入后执行 Maven 命令 install  
mvn install -Dmaven.test.skip=true



terminal

+ [INFO] Reactor Summary:

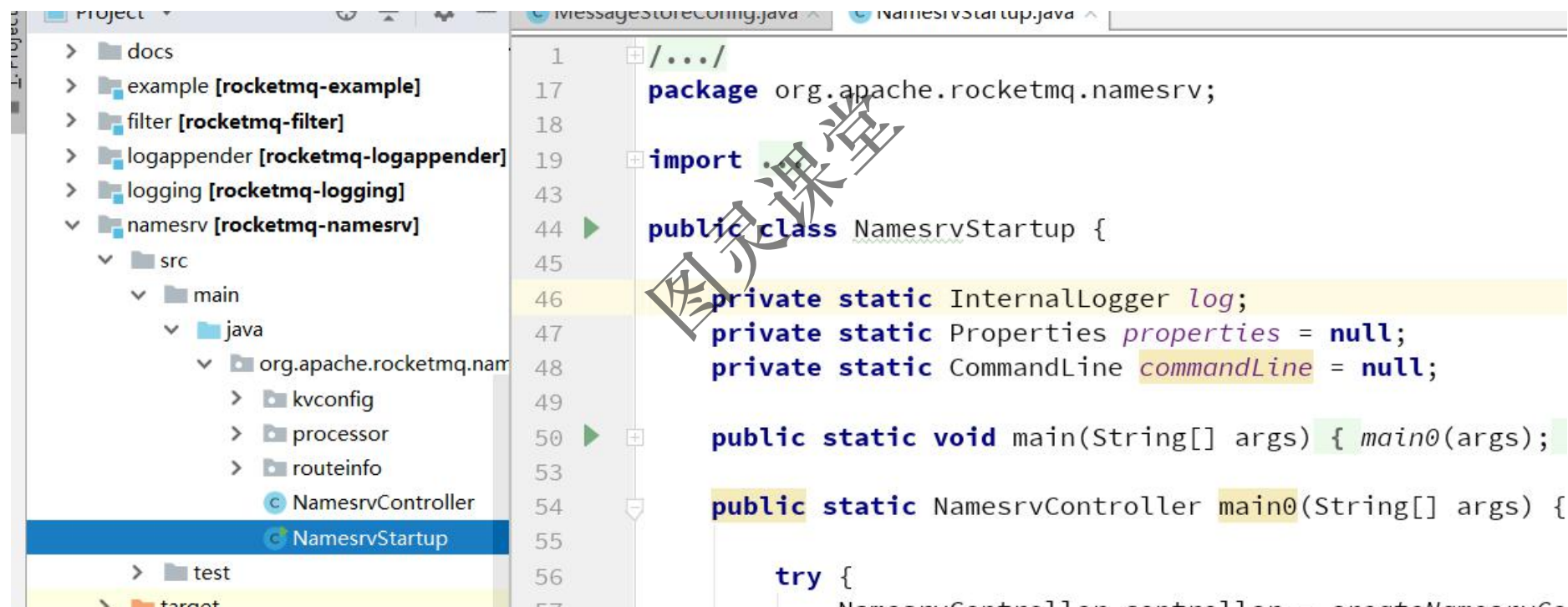
× [INFO]

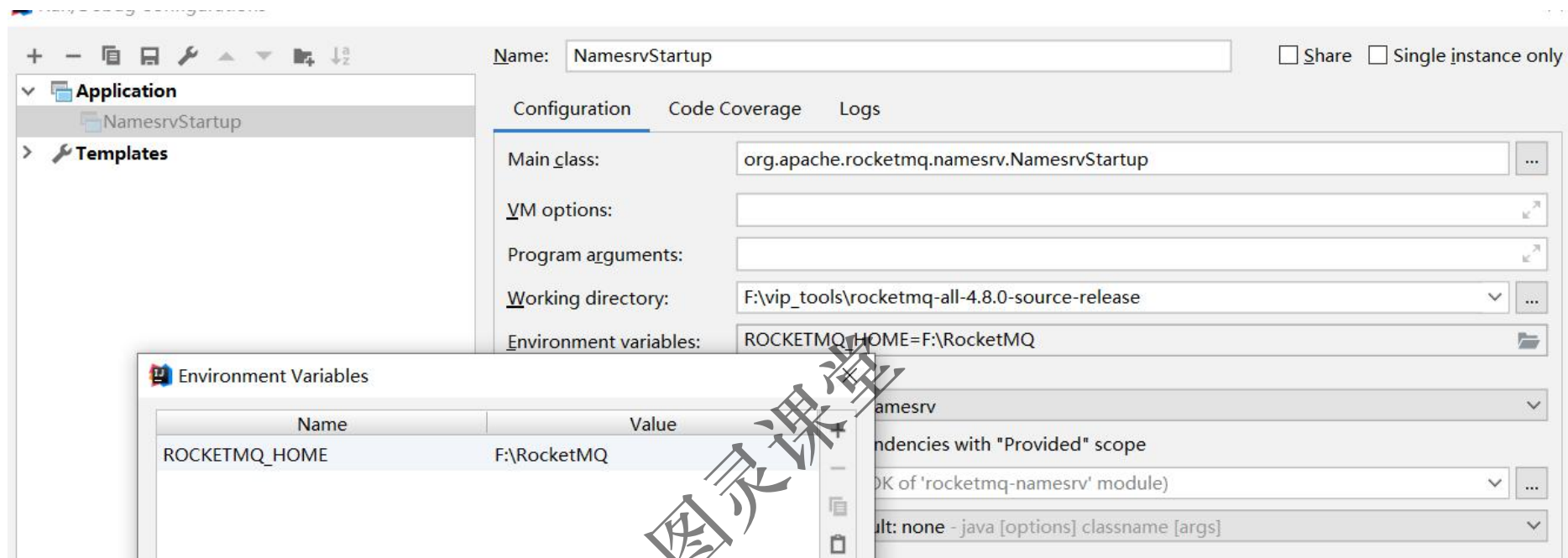
[INFO] Apache RocketMQ 4.8.0 .....	SUCCESS	[ 3.548 s]
[INFO] rocketmq-logging 4.8.0 .....	SUCCESS	[ 2.513 s]
[INFO] rocketmq-remoting 4.8.0 .....	SUCCESS	[ 2.128 s]
[INFO] rocketmq-common 4.8.0 .....	SUCCESS	[ 3.149 s]
[INFO] rocketmq-client 4.8.0 .....	SUCCESS	[ 4.606 s]
[INFO] rocketmq-store 4.8.0 .....	SUCCESS	[ 3.177 s]
[INFO] rocketmq-srvutil 4.8.0 .....	SUCCESS	[ 0.717 s]
[INFO] rocketmq-filter 4.8.0 .....	SUCCESS	[ 1.333 s]
[INFO] rocketmq-acl 4.8.0 .....	SUCCESS	[ 1.220 s]
[INFO] rocketmq-broker 4.8.0 .....	SUCCESS	[ 3.497 s]
[INFO] rocketmq-tools 4.8.0 .....	SUCCESS	[ 2.345 s]
[INFO] rocketmq-namesrv 4.8.0 .....	SUCCESS	[ 1.038 s]
[INFO] rocketmq-logappender 4.8.0 .....	SUCCESS	[ 1.095 s]
[INFO] rocketmq-test 4.8.0 .....	SUCCESS	[ 1.620 s]
[INFO] rocketmq-distribution 4.8.0 .....	SUCCESS	[ 0.198 s]
[INFO] rocketmq-openmessaging 4.8.0 .....	SUCCESS	[ 0.989 s]
[INFO] rocketmq-example 4.8.0 .....	SUCCESS	[ 1.445 s]
[INFO] -----		
[INFO] BUILD SUCCESS		

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

共享 查看

此电脑 > 软件 (F:) > vip_tools > rocketmq-all-4.8.0-source-release > distribution				
名称	修改日期	类型	大小	
benchmark	2020/12/8 18:16	文件夹		
bin	2020/12/8 18:16	文件夹		
conf	2020/12/8 18:16	文件夹		
target	2021/3/19 11:40	文件夹		
LICENSE-BIN	2020/12/8 18:16	文件	17 KB	
NOTICE-BIN	2020/12/8 18:16	文件	2 KB	
pom.xml	2020/12/8 18:16	XML 文档	5 KB	
release.xml	2020/12/8 18:16	XML 文档	3 KB	
release-client.xml	2020/12/8 18:16	XML 文档	3 KB	

此电脑 > 软件 (F:) > RocketMQ > conf				
名称	修改日期	类型	大小	
broker.conf	2020/12/8 18:16	CONF 文件	1 KB	
logback_broker.xml	2020/12/8 18:16	XML 文档	15 KB	
logback_namesrv.xml	2020/12/8 18:16	XML 文档	4 KB	



## 启动 Broker

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



Project ▾

- rocketmq-all-4.8.0-source-release [rocketmq]
- > .github
- > .idea
- > acl [rocketmq-acl]
- ▼ broker [rocketmq-broker]
  - ▼ src
    - ▼ main
      - ▼ java
        - ▼ org.apache.rocketmq.broker
          - > client
          - > dledger
          - > filter
          - > filtersrv
          - > latency
          - > longpolling
          - > mqtrace
          - > offset
          - > out
          - > pagecache
          - > plugin
          - > processor
          - > slave
          - > subscription
          - > topic
          - > transaction
          - > util
          - BrokerController
          - BrokerPathConfigHelper
          - BrokerStartup

resources

BrokerStartup.java ×

```
1  + /.../
17 package org.apache.rocketmq.broker;
18
19 + import ...
50
51 ► public class BrokerStartup {
52     public static Properties properties = null;
53     public static CommandLine commandLine = null;
54     public static String configFile = null;
55     public static InternalLogger log;
56
57 ► + public static void main(String[] args) { start(createBrokerController(args)); }
60
61   public static BrokerController start(BrokerController controller) {
62       try {
63           controller.start();
64
65           String tip = "The broker[" + controller.getBrokerConfig().getBrokerAddr()
66               + controller.getBrokerAddr() + "] boot success. serializeType: " + controller.getBrokerConfig().getSerializeType();
67
68           if (null != controller.getBrokerConfig().getNamesrvAddr()) {
69               tip += " and name server is " + controller.getBrokerConfig().getNamesrvAddr();
70           }
71
72           log.info(tip);
73           System.out.printf("%s\n", tip);
74           return controller;
75       } catch (Throwable e) {
76           log.error("BrokerStartup failed", e);
77       }
78   }
```

需要修改配置文件 broker.conf

图灵课堂

broker.conf

```
10 # Unless required by applicable law or agreed to in writing, software
11 # distributed under the License is distributed on an "AS IS" BASIS,
12 # WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
13 # See the License for the specific language governing permissions and
14 # limitations under the License.
15
16 brokerClusterName = DefaultCluster
17 brokerName = broker-a
18 brokerId = 0
19 deleteWhen = 04
20 fileReservedTime = 48
21 brokerRole = ASYNC_MASTER
22 flushDiskType = ASYNC_FLUSH
23
24 #nameServer
25 namesrvAddr=127.0.0.1:9876
26 autoCreateTopicEnable = true
27
28 storePathRootDir = F:\\RocketMQ\\store
29 #commitLog存储路径
30 storePathCommitLog = F:\\RocketMQ\\store\\commitlog
31 #消费队列存储路径
32 storePathConsumeQueue = F:\\RocketMQ\\store\\consumequeue
33 #消息索引存储路径
34 storePathindex = F:\\RocketMQ\\store\\index
35 #checkpoint文件存储路径
36 storeCheckpoint = F:\\RocketMQ\\store\\checkpoint
37 #abort文件存储路径
38 abortFile = F:\\RocketMQ\\store\\abort
```



#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

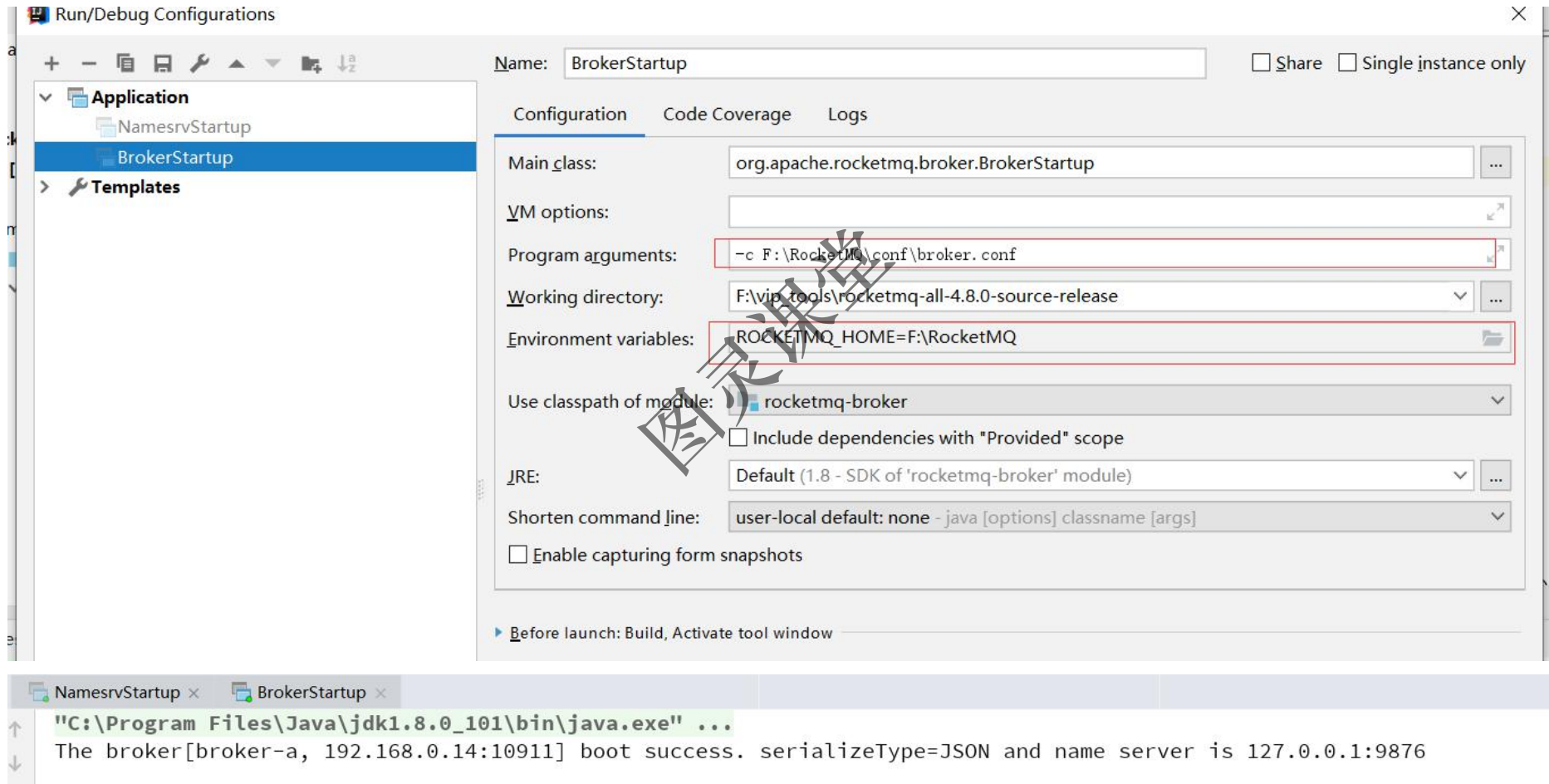
storeCheckpoint = F:\\RocketMQ\\store\\checkpoint

#abort 文件存储路径

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

图灵课堂

## 配置环境变量

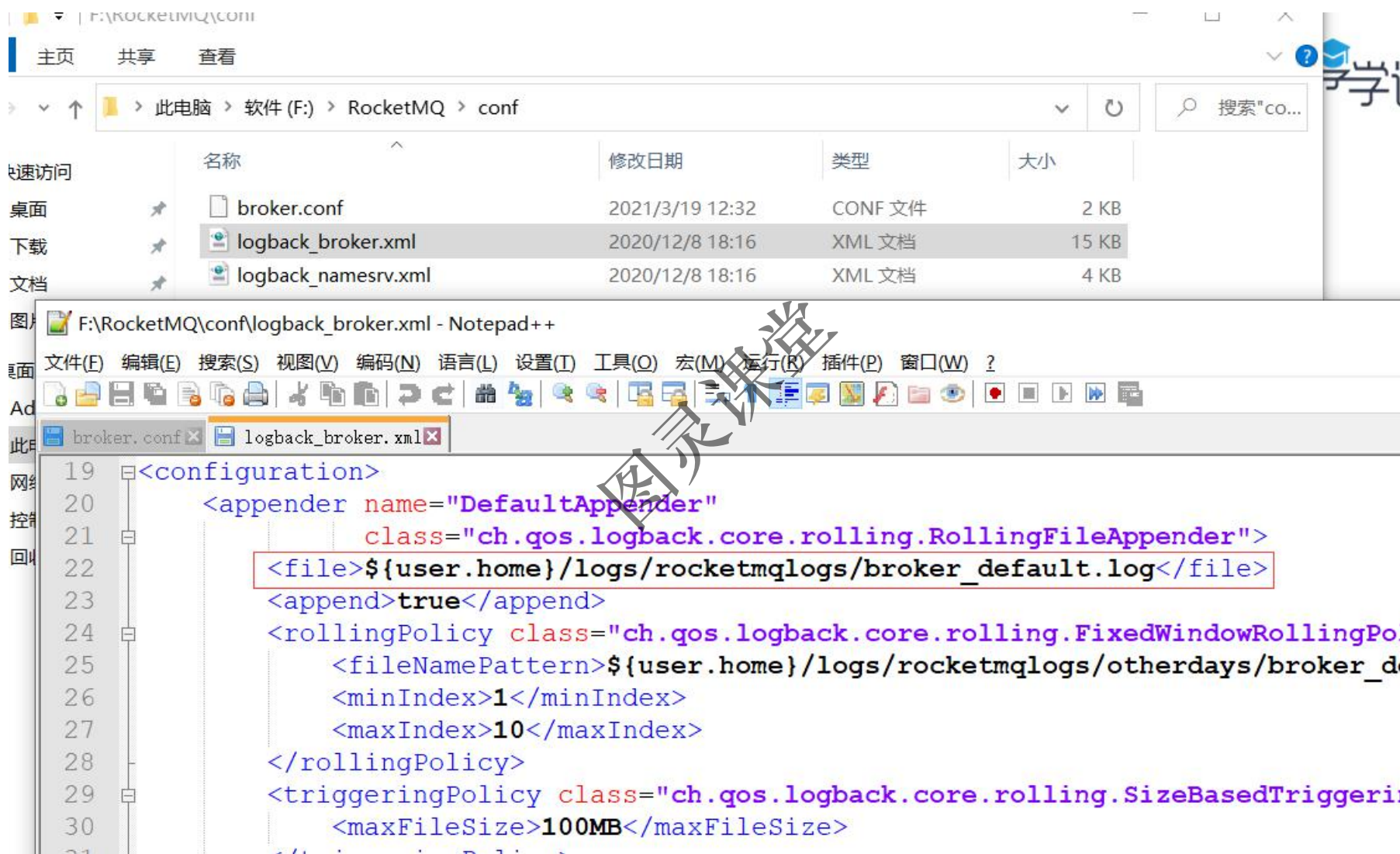


## 最后检查下数据文件

共享		查看	
此电脑		软件 (F:)	RocketMQ > store
名称	修改日期	类型	大小
config	2021/3/19 12:37	文件夹	
abort	2021/3/19 12:36	文件	0 KB
checkpoint	2021/3/19 12:35	文件	4 KB
lock	2021/3/19 12:36	文件	1 KB

## 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



> Administrator > logs > rocketmqlogs

搜索"ro..."

名称	修改日期	类型	大小
 watermark.log	2021/3/19 12:37	文本文档	24,622 KB
 broker.log	2021/3/19 12:37	文本文档	1,261 KB
 store.log	2021/3/19 12:37	文本文档	2,472 KB
 namesrv.log	2021/3/19 12:37	文本文档	361 KB

图灵课堂