



链滴

Apache RocketMQ 单机部署

作者: [yp](#)

原文链接: <https://ld246.com/article/1545925372968>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



前言

这篇文章以4.3.0版本为标准进行讲述在linux下部署RocketMQ单机实例,在此之前需要已配置JAVA环境。

apache RocketMQ 是阿里巴巴在2016年11月捐赠给了apache基金会并于2017年9月顺利毕业成为apache顶级项目。

下载程序包

直接使用一般就[下载](#)已经编译好的二进制文件就好了,下载好以后

```
<code>> unzip rocketmq-all-4.3.0-bin-release.zip
<code>> cd rocketmq-all-4.3.0-bin-release/
</code>
```

启动name server

```
<code>> nohup sh bin/mqnamesrv &
</code>
```

tail一下日志看看是否已经启动成功

```
<code>> tail -f ~/logs/rocketmqlogs/namesrv.log
The Name Server boot success...
</code>
```

启动Broker

```
<code>&gt; nohup sh bin/mqbroker -n localhost:9876 &
</code>
```

tail一下日志看看是否已经启动成功

```
<code>&gt; tail -f ~/logs/rocketmqlogs/broker.log
The broker[%s, 172.30.30.233:10911] boot success...
</code>
```

发送消息和接收消息

为了简单测试下，在发送/接收消息之前我们需要设置**NAMESRV_ADDR**环境变量

```
<code>&gt; export NAMESRV_ADDR=localhost:9876
</code>
```

发送消息，测试用例中的方式是使用for循环发送了1000条消息

```
<code>&gt; sh bin/tools.sh org.apache.rocketmq.example.quickstart.Producer
SendResult [sendStatus=SEND_OK, msgId= ...
</code>
```

接收消息

```
<code>&gt; sh bin/tools.sh org.apache.rocketmq.example.quickstart.Consumer
SendResult [sendStatus=SEND_OK, msgId= ...
</code>
```

关闭服务

```
<code>&gt; sh bin/mqshutdown broker
The mqbroker(36695) is running...
Send shutdown request to mqbroker(36695) OK
</code>
```

```
<code>&gt; sh bin/mqshutdown namesrv
The mqnamesrv(36664) is running...
Send shutdown request to mqnamesrv(36664) OK
</code>
```

到此，我们已经完成了一个简单的部署以及消息的发送和接收,但是这只是在内网下进行的收发消息。

FAQ

怎么修改name server和broker的启动内存?

一开始按照官网的步骤来操作往往会直接启动不起来，提示内存不足，那是因为name server和broke

默认的启动JVM内存是4G，而内存在不够的情况下自然就启动失败了。

这时候需要修改name server和broker的启动内存参数，

broker的启动内存参数在runbroker.sh下修改，windows下则是runbroker.cmd

name server的启动内存参数是在runserver.sh下修改，windows下则是runserver.cmd

需要配置外网访问要怎么配置呢？

在网上见过很多同学在配置外网访问的时候出现问题，得到的提示大多是

No Topic Route Info

这种提示一般都是连接失败了，

1.broker没有允许自动创建topic，且发送消息对应的topic不存在

2.broker没有正确的连接到name server

3.product没有正确的连接到name server

- 对于情况1，在启动 broker的时候添加autoCreateTopicEnable=true参数就好了。
- 情况2呢，有可能是name server不存在或者是端口冲突，具体可以看日志的报错原因。
- 情况3的话，一般是端口原因，如果在服务器上本地收发消息都正常但是到了用客户端sdk发消息的时候失败了可以首先查看一下端口本地ping服务器的9876端口和10911端口是否正常，如果正常可以关闭ip通道再次尝试，以JAVA客户端的话就是下面这样

生产者关闭VIP通道：

```
producer.setVipChannelEnabled(false);
```

消费者关闭VIP通道：

```
consumer.setVipChannelEnabled(false);
```

关闭VIP通道为什么就可以了呢，是因为VIP通道占用了端口，10909端口，而服务器的10909端口有对外开放，所以导致失败了。

配置文件

这是我的broker.properties配置文件

```
brokerClusterName = DefaultCluster
brokerName = broker-a
brokerId = 0
deleteWhen = 04
fileReservedTime = 48
brokerRole = ASYNC_MASTER
flushDiskType = ASYNC_FLUSH
```

```
brokerIP1=172.17.59.146
brokerIP2=172.17.59.146
namesrvAddr=172.17.59.146:5776
autoCreateTopicEnable=true
```

因为我没有使用默认的9876端口，所以也有一个name server的配置文件，namesrv.conf

listenPort=5776

也可以直接下载我的[broker.properties](#)和[namesrv.conf](#)

使用自己的配置文件启动name server和broker的时候都是添加一个指定文件的参数-c [文件路径](#)

到目前为止，一个简单的apache RocketMQ单机实例已经部署好了，开始愉快的消息队列旅程吧！

更多内容可以查看[我的博客](#)