

RocketMQ安装

一、单实例安装&启动

1、安装-目录规划

```
1 cd /opt/soft
2 # 上传RocketMQ二进制压缩包，到/opt/soft目录下 或者直接wget下载
3 wget https://archive.apache.org/dist/rocketmq/4.7.1/rocketmq-all-4.7.1-bin-release.zip
4 # 安装unzip解压
5 yum install -y unzip
6 # 解压
7 unzip rocketmq-all-4.7.1-bin-release.zip
8 # 重命名
9 mv rocketmq-all-4.7.1-bin-release rocketmq-all-4.7.1
10 # 为了替换版本方便，创建软链
11 ln -s /opt/soft/rocketmq-all-4.7.1 /usr/local/rocketmq
```

2、启动&停止NameServer、Broker

先启动NameServer，然后启动Broker，并向 NameServer 注册自己

```
1 ##### NameServer #####
2 # 启动之前根据本机内存大小，看看是否需要修改 JAVA_OPT 的内存设置
3 cd /usr/local/rocketmq/bin
4 sed -i 's/-Xms4g -Xmx4g -Xmn2g/-Xms1g -Xmx1g -Xmn512m/g' ./runserver.sh
5
6 # 启动 NameServer
7 nohup /usr/local/rocketmq/bin/mqnamesrv &
8 # 查看 NameServer启动日志
9 tail -f ~/logs/rocketmqlogs/namesrv.log
10 # 关闭 NameServer
11 /usr/local/rocketmq/bin/mqshutdown namesrv
12
13 ##### Broker #####
14 # 启动之前根据本机内存大小，看看是否需要修改 JAVA_OPT 的内存设置
15 cd /usr/local/rocketmq/bin
16 sed -i 's/-Xms8g -Xmx8g -Xmn4g/-Xms1g -Xmx1g -Xmn512m/g' ./runbroker.sh
17
18 # 启动 Broker
19 nohup /usr/local/rocketmq/bin/mqbroker -n localhost:9876 &
20 # 我的测试版本4.7.1，需要加上autoCreateTopicEnable=true 参数
21 # 否则在测试发送消息的时候会报异常
22 nohup /usr/local/rocketmq/bin/mqbroker -n localhost:9876
   autoCreateTopicEnable=true &
23
24 # 查看 Broker 启动日志
25 tail -f ~/logs/rocketmqlogs/broker.log
26
27 ### 2.关闭Broker
```

3、验证

查看JAVA进程，是否存在NamesrvStartup、BrokerStartup，如果没有则查看LOG日志查找问题。

```
1 | jps
```

```
2612 Jps
2503 BrokerStartup
2456 NamesrvStartup
```

发送和接收消息测试

```
1 | cd /usr/local/rocketmq/bin
2 | # 创建topic
3 | sh mqadmin updateTopic -n localhost:9876 -b localhost:10911 -t TopicTest
4 | # 设置临时环境变量
5 | export NAMESRV_ADDR=localhost:9876
6 | # 开启自带的生产者
7 | sh tools.sh org.apache.rocketmq.example.quickstart.Producer
8 | # 开启自带的消费者
9 | sh tools.sh org.apache.rocketmq.example.quickstart.Consumer
```

二、RocketMQ 排坑：设置broker的ip地址

先上结论

需要在启动的时候导入配置文件

```
1 | # 进入rocketmq根目录
2 | cd /usr/local/rocketmq
3 | #启动 mqnamesrv
4 | nohup sh bin/mqnamesrv &
5 |
6 | # 编写配置文件，并写好配置
7 | echo "brokerIP1=192.168.18.24" >> conf/broker.properties
8 |
9 | #重点: mrbroker 启动时通过 -c 加载配置文件
10 | nohup sh bin/mqbroker -n 192.168.18.24:9876 -c /usr/local/rocketmq/conf/broker.conf &
```

原理：

RocketMQ 的broker 启动类

```
1 | org.apache.rocketmq.broker.BrokerStartup
```

启动的时候会读代码中的默认配置，关于 broker的配置在

```
1 | org.apache.rocketmq.common.BrokerConfig
```

中，根据源代码可以得知，broker使用的默认IP为本机ip

```
1 | brokerIP1 = RemotingUtil.getLocalAddress();
```

得到选取ip的思路是，遍历本地的所有网卡ip，过滤掉“127.0”和“192.168”开头的ip地址然后得到第一个ip，为本机ip。

问题：

当这台机器有很多别的网卡（如：安装docker后），broker使用的ip，就可能会导致我们的客户端无法连接。

解决：

RocketMQ 在加载默认配置后，根据启动时是否包含 -c 参数

```
1 | if (commandLine.hasOption('c'))
```

确定是否执行代码加载额外配置文件。加载时，通过反射的方式，根据配置文件中的键值对，赋值到BrokerConfig 中对应的属性中。

最后：

RocketMQ 启动时会打印出所有“重要的配置”（被@ImportantField所注解的属性），通过打印的brokerIP1 的属性，就可以知道，配置是否成功了。