目录

[第1章 参考文档 3](#_Toc34314974)

[第2章 概览 3](#_Toc34314975)

[2.1 历史 4](#_Toc34314976)

[2.2 作用 4](#_Toc34314977)

[2.3 角色 4](#_Toc34314978)

[2.4 RocketMQ集群模式架构分析 4](#_Toc34314979)

[2.4.1 单节点 4](#_Toc34314980)

[2.4.2 主从(异步、同步消息) 4](#_Toc34314981)

[2.4.3 双主 4](#_Toc34314982)

[2.4.4 双主双从，多主多从模式（异步复制） 5](#_Toc34314983)

[2.4.5 双主双从，多主多从模式（同步双写） 5](#_Toc34314984)

[2.4.6 详解同步复制异步复制，同步刷盘异步刷盘（一般为同步复制，异步刷盘） 5](#_Toc34314985)

[第3章 系统环境 6](#_Toc34314986)

[第4章 单点安装 6](#_Toc34314987)

[4.1 安装openjdk1.8 6](#_Toc34314988)

[4.2 安装rocketmq（下载带bin的） 6](#_Toc34314989)

[4.2.1 上传到服务器 7](#_Toc34314990)

[4.2.2 解压到/usr/local 7](#_Toc34314991)

[4.2.3 进入bin目录，修改配置(分别修改runserver.sh 以及 runbroker.sh，因为默认配置内存过大，可能导致启动失败) 7](#_Toc34314992)

[4.3 创建RocketMQ存储文件的目录 7](#_Toc34314993)

[4.4 修改配置文件 8](#_Toc34314994)

[4.5 进入conf目录，替换所有xml中的${user.home},保证日志路径正确 10](#_Toc34314995)

[4.6 增加环境变量 11](#_Toc34314996)

[4.7 启动RocketMQ， 11](#_Toc34314997)

[4.8 检查是否启动成功 11](#_Toc34314998)

[4.9 停止方式为：先停止broker，在停止namesrv 12](#_Toc34314999)

[第5章 关于rocketmq不支持jdk11得解决办法 12](#_Toc34315000)

[5.1 第一个runserver.sh脚本 12](#_Toc34315001)

[5.2 第二个脚本runbroker.sh 14](#_Toc34315002)

[5.3 第三个脚本tools.sh 16](#_Toc34315003)

[第6章 增加rocketMQ web管理页面rocketmq-console（基于openjdk1.8，openjdk11启动会报错） 18](#_Toc34315004)

[6.1 安装流程 18](#_Toc34315005)

[6.1.1 安装git 18](#_Toc34315006)

[6.1.2 下载rocketmq-console 18](#_Toc34315007)

[6.1.3 修改配置文件 18](#_Toc34315008)

[6.2 下载安装maven 20](#_Toc34315009)

[6.3 添加环境变量 20](#_Toc34315010)

[6.4 生成jar包 20](#_Toc34315011)

[6.5 启动rocketmq web 20](#_Toc34315012)

[6.6 配置web访问密码 20](#_Toc34315013)

[第7章 搭建rocketmq集群 22](#_Toc34315014)

[7.1 conf目录文件介绍 22](#_Toc34315015)

[7.2 搭建一主一从（同步复制，异步刷盘） 22](#_Toc34315016)

[7.2.1 上传包到主172.26.103.76服务器 23](#_Toc34315017)

[7.2.2 两台 服务器分别配置jdk 23](#_Toc34315018)

[7.2.3 创建RocketMQ存储文件的目录 23](#_Toc34315019)

[7.2.4 更改上传到主服务器的配置文件 24](#_Toc34315020)

[7.2.5 进入conf目录，替换所有xml中的${user.home},保证日志路径正确 27](#_Toc34315021)

[7.2.6 进入bin目录，修改配置(分别修改runserver.sh 以及 runbroker.sh，因为默认配置内存过大，可能导致启动失败) 27](#_Toc34315022)

[7.2.7 把在主上172.26.103.76配置好的整体推送到从服务器172.26.103.77 28](#_Toc34315023)

[7.2.8 主从两台服务器分别增加环境变量 28](#_Toc34315024)

[7.2.9 分别启动主从 28](#_Toc34315025)

[7.2.10 两台机器分别检查是否启动成功 29](#_Toc34315026)

[7.3 两主两从（同步复制，异步刷盘） 30](#_Toc34315027)

[7.3.1 上传包到主172.26.103.76服务器 30](#_Toc34315028)

[7.3.2 四台 服务器分别配置jdk 30](#_Toc34315029)

[7.3.3 创建RocketMQ存储文件的目录(主172.26.103.76上面操作) 31](#_Toc34315030)

[7.3.4 更改上传到主服务器的配置文件 31](#_Toc34315031)

[7.3.1 进入conf目录，替换所有xml中的${user.home},保证日志路径正确（还是在172.26.103.76主服务器上操作） 35](#_Toc34315032)

[7.3.2 进入bin目录，修改配置(分别修改runserver.sh 以及 runbroker.sh，因为默认配置内存过大，可能导致启动失败) 35](#_Toc34315033)

[7.3.3 把在主上172.26.103.76配置好的整体推送到其他三台服务器 36](#_Toc34315034)

[7.3.4 分别在四台服务器上配置环境变量 36](#_Toc34315035)

[7.3.5 四台分别启动 36](#_Toc34315036)

[7.3.6 查看集群状态 37](#_Toc34315037)

[7.3.7 其他的双主配置，以及异步配置，只是配置文件不同，其他都一样的。 38](#_Toc34315038)

[第8章 遇到的问题 38](#_Toc34315039)

[8.1 所有的jdk，不管是yum安装的还是解压缩的，都要添加到环境变量，/bin/tools.sh 会通过环境变量来运行java 38](#_Toc34315040)

[第9章 常用命令 39](#_Toc34315041)

[9.1 查看帮助 39](#_Toc34315042)

[9.2 集群相关命令 39](#_Toc34315043)

# 参考文档

<https://blog.csdn.net/xiaoye319/article/details/102481093>

# 概览

简介

<https://www.jianshu.com/p/16c34951a6c4>

## 历史

初始由阿里创建，后捐献给apache基金会，现属于apache项目

## 作用

应用解耦 流量消峰 消息分发(提供不同消息订阅方使用)

## 角色

NameServer(协调管理) Broker(暂存消息) Producer(消息生产者) Consumer(消息消费者)

## RocketMQ集群模式架构分析

### 单节点

优点：本地开发测试，配置简单，同步刷盘消息一条都不会丢

缺点：不可靠，如果宕机，会导致服务不可用

### 主从(异步、同步消息)

优点：同步双写消息不丢失，异步复制存在少量丢失，主节点宕机，从节点可以对外提供消息的消费，但是不支持写入

缺点：主备有短暂消息延迟，毫秒级，目前不支持自动切换，需要脚本或者其他程序进行监测然后进行停止broker，重启让从节点成为主节点

### 双主

优点：配置简单，可以靠配置RAID磁盘阵列保证消息可靠，异步刷盘丢失少量消息

缺点：master机器宕机期间，未被消费的消息在机器恢复之间不可消息，实时性会受到影响

### 双主双从，多主多从模式（异步复制）

优点：磁盘损坏，消息丢失的非常少，消息实时性不会受影响，Master宕机后，消费者仍然可以从Slave消费

缺点：主备有短暂消息延迟，毫秒级，如果Master宕机，磁盘损坏情况，会丢失少量消息

### 双主双从，多主多从模式（同步双写）

优点：同步双写方式，主备都写成功，向应用才返回成功，服务可用性与数据可用性都非常高

缺点：性能比异步复制模式略低，主宕机后，备机不能自动切换为主机

### 详解同步复制异步复制，同步刷盘异步刷盘（一般为同步复制，异步刷盘）

同步复制、异步复制

　　如果一个broker组有Master和Slave，消息需要从Master复制到Slave上，有同步和异步两种复制方式。

同步复制是等Master和Slave

均写成功后才反馈给客户端写成功状态；异步复制方式是只要Master写成功即可反馈给客户端写成功状态

　　这两种复制方式各有优劣，在异步复制方式下，系统拥有较低的延迟和较高的吞吐量，但是如果Master

出了故障，有些数据因为没有被写入Slave，有可能会丢失；在同步复制方式下，如果Master出故障，Slave

上有全部的备份数据，容易恢复，但是同步复制会增大数据写入延迟，降低系统吞吐量。

　　同步复制和异步复制是通过Broker配置文件里的brokerRole参数进行设置的，这个参数可以被设置成

ASYNC\_MASTER、SYNC\_MASTER、SLAVE三个值中的一个。

　　实际应用中要结合业务场景，合理设置刷盘方式和主从复制方式，尤其是SYNC\_FLUSH方式，由于频繁

的触发写磁盘动作，会明显降低性能。

# 系统环境

2020-03-05

|  |  |
| --- | --- |
| 系统版本 | centos7.7 |
| rocketmq版本 | rocketmq-all-4.5.0-bin-release.zip |
| jdk版本 | openjdk1.8 |
| 硬件 | 2核4G |

下载地址：<http://rocketmq.apache.org/dowloading/releases/>

# 单点安装

## 安装openjdk1.8

yum install java-1.8.0-openjdk java-1.8.0-openjdk-devel -y

所有的jdk，不管是yum安装的还是解压缩的，都要添加到环境变量，/bin/tools.sh 会通过环境变量来运行java

yum安装的openjdk配置环境变量

JAVA\_HOME=/usr/lib/jvm/java

JRE\_HOME=/usr/lib/jvm/java/jre

PATH=$PATH:$JAVA\_HOME/bin:$JRE\_HOME/bin

CLASSPATH=.:$JAVA\_HOME/lib/dt.jar:$JAVA\_HOME/lib/tools.jar:$JRE\_HOME/lib

export JAVA\_HOME JRE\_HOME PATH CLASSPATH

java -version

jps

echo $JAVA\_HOME

## 安装rocketmq（下载带bin的）

wget https://archive.apache.org/dist/rocketmq/4.5.0/rocketmq-all-4.5.0-bin-release.zip

### 上传到服务器

### 解压到/usr/local

### 进入bin目录，修改配置(分别修改runserver.sh 以及 runbroker.sh，因为默认配置内存过大，可能导致启动失败)

RocketMQ对内存的要求比较高，最少1G，如果内存太少，会影响RocketMQ的运行效率和执行性能。我们需要修改bin目录下的runbroker.sh和runserver.sh文件

cd /usr/local/rocketmq-4.5/bin

vim runserver.sh

本来的4g：

JAVA\_OPT="${JAVA\_OPT} -server -Xms4g -Xmx4g -Xmn2g -XX:MetaspaceSize=128m -XX:MaxMetaspaceSize=320m"

按照实际机器配置修改1g：

JAVA\_OPT="${JAVA\_OPT} -server -Xms1g -Xmx1g -Xmn512m -XX:MetaspaceSize=128m -XX:MaxMetaspaceSize=320m"

vim runbroker.sh

本来的8g：

JAVA\_OPT="${JAVA\_OPT} -server -Xms8g -Xmx8g -Xmn4g"

按照实际机器配置修改1g：

JAVA\_OPT="${JAVA\_OPT} -server -Xms1g -Xmx1g -Xmn512m"

## 创建RocketMQ存储文件的目录

cd /usr/local/rocketmq-4.5/

mkdir logs

mkdir store

cd store/

mkdir commitlog

mkdir consumequeue

mkdir index

文件夹说明：

logs:存储RocketMQ日志目录

store:存储RocketMQ数据文件目录

commitlog:存储RocketMQ消息信息

consumequeue、index:存储消息的索引数据

## 修改配置文件

cp broker.conf broker.conf.bak

vim broker.conf

#所属集群名字

brokerClusterName=rocketmq-cluster

#broker名字，注意此处不同的配置文件填写的不一样

brokerName=broker-a

#0 表示Master, > 0 表示slave

brokerId=0

#nameServer 地址，分号分割

namesrvAddr=192.168.0.75:9876;127.0.0.1:9876

#在发送消息时，自动创建服务器不存在的Topic，默认创建的队列数

defaultTopicQueueNums=4

#是否允许Broker 自动创建Topic，建议线下开启，线上关闭

autoCreateTopicEnable=true

#是否允许Broker自动创建订阅组，建议线下开启，线上关闭

autoCreateSubscriptionGroup=true

#Broker 对外服务的监听端口

listenPort=10911

#删除文件时间点，默认是凌晨4点

deleteWhen=04

#文件保留时间，默认48小时

fileReservedTime=120

#commitLog每个文件的大小默认1G

mapedFileSizeCommitLog=1073741824

#ConsumeQueue每个文件默认存30W条，根据业务情况调整

mapedFileSizeConsumeQueue=300000

#destroyMapedFileIntervalForcibly=120000

#redeleteHangedFileInterval=120000

#设置物理文件磁盘最大使用率

#diskMaxUsedSpaceRatio=88

#磁盘空间警戒阈值，超过这个值则停止接受消息，默认值90

#diskSpaceWarningLevelRatio=90

#强制删除文件阈值，默认85

#diskSpaceCleanForciblyRatio=85

#存储路径

storePathRootDir=/usr/local/rocketmq-4.5/store

#commitLog存储路径

storePathCommitLog=/usr/local/rocketmq-4.5/store/commitlog

#消费队列存储路径

storePathConsumeQueue=/usr/local/rocketmq-4.5/store/consumequeue

#消息索引存储路径

storePathIndex=/usr/local/rocketmq-4.5/store/index

#checkpoint 文件存储路径

storeCheckpoint=/usr/local/rocketmq-4.5/store/checkpoint

#abort 文件存储路径

abortFile=/usr/local/rocketmq-4.5/store/abort

#限制的消息大小

maxMessageSize=65536

# flushCommitLogLeastPages=4

# flushConsumeQueueLeastPages=2

# flushCommitLogThoroughInterval=10000

# flushConsumeQueueThoroughInterval=60000

# Broker 的角色（一般为同步复制）

# - ASYNC\_MASTER 异步复制Master

# - SYNC\_MASTER 同步双写Master

# - SLAVE

brokerRole=SYNC\_MASTER

# 刷盘方式（一般为异步刷盘）

# - ASYNC\_FLUSH 异步刷盘

# - SYNC\_FLUSH 同步刷盘

flushDiskType=ASYNC\_FLUSH

#checkTransactionMessageEnable=false

#发消息线程池数量

#sendMessageTreadPoolNums=128

#拉消息线程池数量

#pullMessageTreadPoolNums=128

## 进入conf目录，替换所有xml中的${user.home},保证日志路径正确

sed -i 's#${user.home}#/usr/local/rocketmq-4.5#g' \*.xml

## 增加环境变量

echo 'export PATH=$PATH:/usr/local/rocketmq-4.5/bin' >>/etc/profile

source /etc/profile

## 启动RocketMQ，

注意先启动namesrv再启动broker

必须先cd到bin目录下，不然会报错。

cd /usr/local/rocketmq-4.5/bin/

启动namesrv

nohup sh mqnamesrv &

再启动broker

nohup sh mqbroker -c /usr/local/rocketmq-4.5/conf/broker.conf > /dev/null 2>&1 &

停止RocketMQ的方法

mqshutdown broker

mqshutdown namesrv

## 检查是否启动成功

jps命令为java自带的，如果yum安装没有jps说明未安装jdk-devel包

[root@oracle-01 bin]# jps

15969 BrokerStartup

15762 NamesrvStartup

16861 Jps

有此两个说明运行成功

rocketmq对外端口为9876

broker端口为10911

## 停止方式为：先停止broker，在停止namesrv

mqshutdown broker

mqshutdown namesrv

# 关于rocketmq不支持jdk11得解决办法

<https://www.cnblogs.com/wintersoft/p/10881601.html>

经测试rocketmq4.5.0 rocketmq4.3.2 都可以使用此配置

需要更改三个启动脚本

## 第一个runserver.sh脚本

nameserver:

vim /opt/rocketmq/bin/runserver.sh

删除 -XX:+UseConcMarkSweepGC -XX:+UseCMSCompactAtFullCollection和-XX:-UseParNewGC和-XX:+PrintGCDetails

-Xloggc:改成-Xlog:gc:

删除 JAVA\_OPT="${JAVA\_OPT} -Djava.ext.dirs=${JAVA\_HOME}/jre/lib/ext:${BASE\_DIR}/lib"

#!/bin/sh

# Licensed to the Apache Software Foundation (ASF) under one or more

# contributor license agreements. See the NOTICE file distributed with

# this work for additional information regarding copyright ownership.

# The ASF licenses this file to You under the Apache License, Version 2.0

# (the "License"); you may not use this file except in compliance with

# the License. You may obtain a copy of the License at

#

# http://www.apache.org/licenses/LICENSE-2.0

#

# Unless required by applicable law or agreed to in writing, software

# distributed under the License is distributed on an "AS IS" BASIS,

# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

# See the License for the specific language governing permissions and

# limitations under the License.

#===========================================================================================

# Java Environment Setting

#===========================================================================================

error\_exit ()

{

echo "ERROR: $1 !!"

exit 1

}

[ ! -e "$JAVA\_HOME/bin/java" ] && JAVA\_HOME=$HOME/jdk/java

[ ! -e "$JAVA\_HOME/bin/java" ] && JAVA\_HOME=/usr/java

[ ! -e "$JAVA\_HOME/bin/java" ] && error\_exit "Please set the JAVA\_HOME variable in your environment, We need java(x64)!"

export JAVA\_HOME

export JAVA="$JAVA\_HOME/bin/java"

export BASE\_DIR=$(dirname $0)/..

export CLASSPATH=${BASE\_DIR}/lib/rocketmq-namesrv-4.5.0.jar:${BASE\_DIR}/lib/\*:${BASE\_DIR}/conf:${CLASSPATH}

#export CLASSPATH=.:${BASE\_DIR}/conf:${CLASSPATH}

#echo "BASE\_DIR:$BASE\_DIR"

#echo "CLASSPATH:$CLASSPATH"

#===========================================================================================

# JVM Configuration

#===========================================================================================

JAVA\_OPT="${JAVA\_OPT} -server -Xms1g -Xmx1g -Xmn512m -XX:MetaspaceSize=128m -XX:MaxMetaspaceSize=320m"

#JAVA\_OPT="${JAVA\_OPT} -server -Xms4g -Xmx4g -Xmn2g -XX:MetaspaceSize=128m -XX:MaxMetaspaceSize=320m"

JAVA\_OPT="${JAVA\_OPT} -XX:CMSInitiatingOccupancyFraction=70 -XX:+CMSParallelRemarkEnabled -XX:SoftRefLRUPolicyMSPerMB=0 -XX:+CMSClassUnloadingEnabled -XX:SurvivorRatio=8"

JAVA\_OPT="${JAVA\_OPT} -verbose:gc -Xlog:gc:/dev/shm/rmq\_srv\_gc.log"

JAVA\_OPT="${JAVA\_OPT} -XX:-OmitStackTraceInFastThrow"

JAVA\_OPT="${JAVA\_OPT} -XX:-UseLargePages"

#JAVA\_OPT="${JAVA\_OPT} -Xdebug -Xrunjdwp:transport=dt\_socket,address=9555,server=y,suspend=n"

JAVA\_OPT="${JAVA\_OPT} ${JAVA\_OPT\_EXT}"

JAVA\_OPT="${JAVA\_OPT} -cp ${CLASSPATH}"

$JAVA ${JAVA\_OPT} $@

## 第二个脚本runbroker.sh

broker:

vim /opt/rocketmq/bin/runbroker.sh

如果主从都在同一个服务器上 vim /opt/rocketmqs/bin/runbroker.sh

删除 -XX:+PrintGCDateStamps -XX:+PrintGCApplicationStoppedTime -XX:+PrintAdaptiveSizePolicy 和 JAVA\_OPT="${JAVA\_OPT} -XX:+UseGCLogFileRotation -XX:NumberOfGCLogFiles=5 -XX:GCLogFileSize=30m" 和-XX:+PrintGCDetails

-Xloggc:改成-Xlog:gc:

删除 JAVA\_OPT="${JAVA\_OPT} -Djava.ext.dirs=${JAVA\_HOME}/jre/lib/ext:${BASE\_DIR}/lib"

#!/bin/sh

# Licensed to the Apache Software Foundation (ASF) under one or more

# contributor license agreements. See the NOTICE file distributed with

# this work for additional information regarding copyright ownership.

# The ASF licenses this file to You under the Apache License, Version 2.0

# (the "License"); you may not use this file except in compliance with

# the License. You may obtain a copy of the License at

#

# http://www.apache.org/licenses/LICENSE-2.0

#

# Unless required by applicable law or agreed to in writing, software

# distributed under the License is distributed on an "AS IS" BASIS,

# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

# See the License for the specific language governing permissions and

# limitations under the License.

#===========================================================================================

# Java Environment Setting

#===========================================================================================

error\_exit ()

{

echo "ERROR: $1 !!"

exit 1

}

[ ! -e "$JAVA\_HOME/bin/java" ] && JAVA\_HOME=$HOME/jdk/java

[ ! -e "$JAVA\_HOME/bin/java" ] && JAVA\_HOME=/usr/java

[ ! -e "$JAVA\_HOME/bin/java" ] && error\_exit "Please set the JAVA\_HOME variable in your environment, We need java(x64)!"

export JAVA\_HOME

export JAVA="$JAVA\_HOME/bin/java"

export BASE\_DIR=$(dirname $0)/..

export CLASSPATH=${BASE\_DIR}/lib/rocketmq-broker-4.5.0.jar:${BASE\_DIR}/lib/\*:${BASE\_DIR}/conf:${CLASSPATH}

#export CLASSPATH=.:${BASE\_DIR}/conf:${CLASSPATH}

#echo "BASE\_DIR:$BASE\_DIR"

#echo "CLASSPATH:$CLASSPATH"

#===========================================================================================

# JVM Configuration

#===========================================================================================

JAVA\_OPT="${JAVA\_OPT} -server -Xms1g -Xmx1g -Xmn512m"

#JAVA\_OPT="${JAVA\_OPT} -server -Xms8g -Xmx8g -Xmn4g"

JAVA\_OPT="${JAVA\_OPT} -XX:+UseG1GC -XX:G1HeapRegionSize=16m -XX:G1ReservePercent=25 -XX:InitiatingHeapOccupancyPercent=30 -XX:SoftRefLRUPolicyMSPerMB=0"

JAVA\_OPT="${JAVA\_OPT} -verbose:gc -Xlog:gc:/dev/shm/mq\_gc\_%p.log"

JAVA\_OPT="${JAVA\_OPT} -XX:-OmitStackTraceInFastThrow"

JAVA\_OPT="${JAVA\_OPT} -XX:+AlwaysPreTouch"

JAVA\_OPT="${JAVA\_OPT} -XX:MaxDirectMemorySize=15g"

JAVA\_OPT="${JAVA\_OPT} -XX:-UseLargePages -XX:-UseBiasedLocking"

#JAVA\_OPT="${JAVA\_OPT} -Xdebug -Xrunjdwp:transport=dt\_socket,address=9555,server=y,suspend=n"

JAVA\_OPT="${JAVA\_OPT} ${JAVA\_OPT\_EXT}"

JAVA\_OPT="${JAVA\_OPT} -cp ${CLASSPATH}"

numactl --interleave=all pwd > /dev/null 2>&1

if [ $? -eq 0 ]

then

if [ -z "$RMQ\_NUMA\_NODE" ] ; then

numactl --interleave=all $JAVA ${JAVA\_OPT} $@

else

numactl --cpunodebind=$RMQ\_NUMA\_NODE --membind=$RMQ\_NUMA\_NODE $JAVA ${JAVA\_OPT} $@

fi

else

$JAVA ${JAVA\_OPT} $@

fi

## 第三个脚本tools.sh

tools:

vim /opt/rocketmq/bin/tools.sh

删除 JAVA\_OPT="${JAVA\_OPT} -Djava.ext.dirs=${BASE\_DIR}/lib:${JAVA\_HOME}/jre/lib/ext"

#!/bin/sh

# Licensed to the Apache Software Foundation (ASF) under one or more

# contributor license agreements. See the NOTICE file distributed with

# this work for additional information regarding copyright ownership.

# The ASF licenses this file to You under the Apache License, Version 2.0

# (the "License"); you may not use this file except in compliance with

# the License. You may obtain a copy of the License at

#

# http://www.apache.org/licenses/LICENSE-2.0

#

# Unless required by applicable law or agreed to in writing, software

# distributed under the License is distributed on an "AS IS" BASIS,

# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

# See the License for the specific language governing permissions and

# limitations under the License.

#===========================================================================================

# Java Environment Setting

#===========================================================================================

error\_exit ()

{

echo "ERROR: $1 !!"

exit 1

}

[ ! -e "$JAVA\_HOME/bin/java" ] && JAVA\_HOME=$HOME/jdk/java

[ ! -e "$JAVA\_HOME/bin/java" ] && JAVA\_HOME=/usr/java

[ ! -e "$JAVA\_HOME/bin/java" ] && error\_exit "Please set the JAVA\_HOME variable in your environment, We need java(x64)!"

export JAVA\_HOME

export JAVA="$JAVA\_HOME/bin/java"

export BASE\_DIR=$(dirname $0)/..

export CLASSPATH=${BASE\_DIR}/lib/\*:${BASE\_DIR}/conf:.:${CLASSPATH}

#export CLASSPATH=.:${BASE\_DIR}/conf:${CLASSPATH}

#echo "BASE\_DIR:$BASE\_DIR"

#echo "CLASSPATH:$CLASSPATH"

#===========================================================================================

# JVM Configuration

#===========================================================================================

JAVA\_OPT="${JAVA\_OPT} -server -Xms1g -Xmx1g -Xmn256m -XX:PermSize=128m -XX:MaxPermSize=128m"

JAVA\_OPT="${JAVA\_OPT} -cp ${CLASSPATH}"

$JAVA ${JAVA\_OPT} $@

# 增加rocketMQ web管理页面rocketmq-console（基于openjdk1.8，openjdk11启动会报错）

参考文档

<https://www.cnblogs.com/immense/p/11406399.html>

github配置文档

<https://github.com/apache/rocketmq-externals/blob/master/rocketmq-console/doc/1_0_0/UserGuide_CN.md>

maven下载地址

<https://archive.apache.org/dist/maven/maven-3/>

|  |  |
| --- | --- |
| rocketmq-console版本 | release-rocketmq-console-1.0.0 |
| maven版本 | apache-maven-3.6.0-bin.zip |
| git | yum install git 任意版本 |

## 安装流程

### 安装git

yum install git -y

### 下载rocketmq-console

git clone <https://github.com/apache/rocketmq-externals.git>

mv rocketmq-externals/rocketmq-console/ /usr/local/rocketmq-4.5/

### 修改配置文件

vim /usr/local/rocketmq-4.5/rocketmq-console/src/main/resources/application.properties

实际需要修改的只有

rocketmq.config.dataPath

server.port

rocketmq.config.namesrvAddr

rocketmq.config.loginRequired

###管理后台访问上下文路径，默认为空，如果填写，一定要前面加“/”，后面不要加，否则启动报错

server.contextPath=

###web访问端口

server.port=8181

### SSL setting

#server.ssl.key-store=classpath:rmqcngkeystore.jks

#server.ssl.key-store-password=rocketmq

#server.ssl.keyStoreType=PKCS12

#server.ssl.keyAlias=rmqcngkey

#spring.application.index=true

spring.application.name=rocketmq-console

spring.http.encoding.charset=UTF-8

spring.http.encoding.enabled=true

spring.http.encoding.force=true

logging.config=classpath:logback.xml

####Name Server地址，修改成你自己的服务地址192.168.26.29:9876;192.168.26.206:9876

rocketmq.config.namesrvAddr=

#if you use rocketmq version < 3.5.8, rocketmq.config.isVIPChannel should be false.default true

rocketmq.config.isVIPChannel=

# rocketmq-console数据路径

rocketmq.config.dataPath=/usr/local/rocketmq-4.5/rocketmq-console/data

#set it false if you don't want use dashboard.default true

rocketmq.config.enableDashBoardCollect=true

#set the message track trace topic if you don't want use the default one

rocketmq.config.msgTrackTopicName=

rocketmq.config.ticketKey=ticket

#Must create userInfo file: ${rocketmq.config.dataPath}/users.properties if the login is required

#是否开启密码验证登录，默认不开启 ，默认用户名密码admin/admin

rocketmq.config.loginRequired=true

## 下载安装maven

wget https://archive.apache.org/dist/maven/maven-3/3.6.0/binaries/apache-maven-3.6.0-bin.tar.gz

tar xf apache-maven-3.6.0-bin.tar.gz -C /usr/local/

## 添加环境变量

echo 'export PATH=$PATH:/usr/local/apache-maven-3.6.0/bin' >>/etc/profile

source /etc/profile

测试是否生效

mvn -v

## 生成jar包

cd /usr/local/rocketmq-4.5/rocketmq-console/

mvn clean package -Dmaven.test.skip=true

这段时间会很长，执行完成，会多出一个target目录。在target下生成了需要运行的jar包

## 启动rocketmq web

nohup java -jar target/rocketmq-console-ng-1.0.0.jar &>/dev/null &

<http://39.101.223.167:8181/>

## 配置web访问密码

vim /usr/local/rocketmq-4.5/rocketmq-console/src/main/resources/application.properties

在之前上面的 rocketmq-console配置文件内配置的data路径下创建users.properties文件

vim /usr/local/rocketmq-4.5/rocketmq-console/data/users.properties

把下面的复制进去就好了，密码文件创建后需要重新启动一下rocketmq-console，否则密码文件不生效。

[root@iZ8vb87v6wieobhdkmk73jZ data]# jps

20208 Jps

1441 NamesrvStartup

19909 rocketmq-console-ng-1.0.1.jar

1493 BrokerStartup

[root@iZ8vb87v6wieobhdkmk73jZ data]# kill 19909

[root@iZ8vb87v6wieobhdkmk73jZ rocketmq-console]# cd /usr/local/rocketmq-4.5/rocketmq-console/

[root@iZ8vb87v6wieobhdkmk73jZ rocketmq-console]# nohup java -jar target/rocketmq-console-ng-1.0.0.jar &>/dev/null &

users.properties

# 该文件支持热修改，即添加和修改用户时，不需要重新启动console

# 格式， 每行定义一个用户， username=password[,N] #N是可选项，可以为0 (普通用户)； 1 （管理员）

#定义管理员

admin=test123456,1

#定义普通用户

#user1=user1

#user2=user2

# 搭建rocketmq集群

## conf目录文件介绍

[root@iZ8vb87v6wieobhdkmk73jZ conf]# tree

.

├── 2m-2s-async #两主两从异步用此目录下的配置文件

│   ├── broker-a.properties #a的主配置文件

│   ├── broker-a-s.properties # a的从配置文件

│   ├── broker-b.properties #b的主配置文件

│   └── broker-b-s.properties #b的从配置文件

├── 2m-2s-sync #两主两从同步用此目录下的配置文件

│   ├── broker-a.properties

│   ├── broker-a-s.properties

│   ├── broker-b.properties

│   └── broker-b-s.properties

├── 2m-noslave #双主用此目录下的文件

│   ├── broker-a.properties

│   ├── broker-b.properties

│   └── broker-trace.properties #暂时没用到

├── broker.conf #单点搭建可以用此配置文件

1. 2m-2s-async:2主2从异步
2. 2m-2s-sync :2主2从同步
3. 2m-noslave :2主没有从

## 搭建一主一从（同步复制，异步刷盘）

|  |  |
| --- | --- |
| 两台服务器 | 配置1核2G |
| 172.26.103.76 | 主 |
| 172.26.103.77 | 从 |

### 上传包到主172.26.103.76服务器

rocketmq-all-4.5.0-bin-release.zip

unzip rocketmq-all-4.5.0-bin-release.zip

mv rocketmq-all-4.5.0-bin-release /usr/local/rocketmq-4.5.0

### 两台 服务器分别配置jdk

yum install java-1.8.0-openjdk java-1.8.0-openjdk-devel -y

所有的jdk，不管是yum安装的还是解压缩的，都要添加到环境变量，/bin/tools.sh 会通过环境变量来运行java

yum安装的openjdk配置环境变量

JAVA\_HOME=/usr/lib/jvm/java

JRE\_HOME=/usr/lib/jvm/java/jre

PATH=$PATH:$JAVA\_HOME/bin:$JRE\_HOME/bin

CLASSPATH=.:$JAVA\_HOME/lib/dt.jar:$JAVA\_HOME/lib/tools.jar:$JRE\_HOME/lib

export JAVA\_HOME JRE\_HOME PATH CLASSPATH

java -version

jps

echo $JAVA\_HOME

### 创建RocketMQ存储文件的目录

cd /usr/local/rocketmq-4.5.0/

mkdir logs

mkdir store

cd store/

mkdir commitlog

mkdir consumequeue

mkdir index

文件夹说明：

logs:存储RocketMQ日志目录

store:存储RocketMQ数据文件目录

commitlog:存储RocketMQ消息信息

consumequeue、index:存储消息的索引数据

### 更改上传到主服务器的配置文件

先修改主配置文件

vi /usr/local/rocketmq-4.5.0/conf/2m-2s-sync/broker-a.properties

主要需要注意修改的：

brokerClusterName #所属集群名字

brokerName #broker名字，注意此处不同的配置文件填写的不一样

brokerId #0 表示Master, > 0 表示slave

namesrvAddr #nameServer 地址，分号分割 172.26.103.76:9876;172.26.103.77:9876

storePathRootDir #存储路径

brokerRole # Broker 的角色

flushDiskType #刷盘方式

#所属集群名字

brokerClusterName=rocketmq-cluster

#broker名字，注意此处不同的配置文件填写的不一样

brokerName=broker-a

#0 表示Master, > 0 表示slave

brokerId=0

#nameServer 地址，分号分割

namesrvAddr=172.26.103.76:9876;172.26.103.77:9876

#在发送消息时，自动创建服务器不存在的Topic，默认创建的队列数

defaultTopicQueueNums=4

#是否允许Broker 自动创建Topic，建议线下开启，线上关闭

autoCreateTopicEnable=true

#是否允许Broker自动创建订阅组，建议线下开启，线上关闭

autoCreateSubscriptionGroup=true

#Broker 对外服务的监听端口

listenPort=10911

#删除文件时间点，默认是凌晨4点

deleteWhen=04

#文件保留时间，默认48小时

fileReservedTime=120

#commitLog每个文件的大小默认1G

mapedFileSizeCommitLog=1073741824

#ConsumeQueue每个文件默认存30W条，根据业务情况调整

mapedFileSizeConsumeQueue=300000

#destroyMapedFileIntervalForcibly=120000

#redeleteHangedFileInterval=120000

#设置物理文件磁盘最大使用率

#diskMaxUsedSpaceRatio=88

#磁盘空间警戒阈值，超过这个值则停止接受消息，默认值90

#diskSpaceWarningLevelRatio=90

#强制删除文件阈值，默认85

#diskSpaceCleanForciblyRatio=85

#存储路径

storePathRootDir=/usr/local/rocketmq-4.5/store

#commitLog存储路径

storePathCommitLog=/usr/local/rocketmq-4.5/store/commitlog

#消费队列存储路径

storePathConsumeQueue=/usr/local/rocketmq-4.5/store/consumequeue

#消息索引存储路径

storePathIndex=/usr/local/rocketmq-4.5/store/index

#checkpoint 文件存储路径

storeCheckpoint=/usr/local/rocketmq-4.5/store/checkpoint

#abort 文件存储路径

abortFile=/usr/local/rocketmq-4.5/store/abort

#限制的消息大小

maxMessageSize=65536

# flushCommitLogLeastPages=4

# flushConsumeQueueLeastPages=2

# flushCommitLogThoroughInterval=10000

# flushConsumeQueueThoroughInterval=60000

# Broker 的角色

# - ASYNC\_MASTER 异步复制Master

# - SYNC\_MASTER 同步双写Master

# - SLAVE

brokerRole=SYNC\_MASTER

# 刷盘方式

# - ASYNC\_FLUSH 异步刷盘

# - SYNC\_FLUSH 同步刷盘

flushDiskType=ASYNC\_FLUSH

#checkTransactionMessageEnable=false

#发消息线程池数量

#sendMessageTreadPoolNums=128

#拉消息线程池数量

#pullMessageTreadPoolNums=128

修改从配置文件（还是在主服务器上操作）

cat /usr/local/rocketmq-4.5.0/conf/2m-2s-sync/broker-a.properties >/usr/local/rocketmq-4.5.0/conf/2m-2s-sync/broker-a-s.properties

vim /usr/local/rocketmq-4.5.0/conf/2m-2s-sync/broker-a-s.properties

配置文件与主一样

主要需要注意修改的：

brokerId #0 表示Master, > 0 表示slave，设置为1

brokerRole # 从得Broker 的角色为SLAVE

### 进入conf目录，替换所有xml中的${user.home},保证日志路径正确

sed -i 's#${user.home}#/usr/local/rocketmq-4.5.0#g' \*.xml

### 进入bin目录，修改配置(分别修改runserver.sh 以及 runbroker.sh，因为默认配置内存过大，可能导致启动失败)

RocketMQ对内存的要求比较高，最少1G，如果内存太少，会影响RocketMQ的运行效率和执行性能。我们需要修改bin目录下的runbroker.sh和runserver.sh文件

cd /usr/local/rocketmq-4.5/bin

vim runserver.sh

本来的4g：

JAVA\_OPT="${JAVA\_OPT} -server -Xms4g -Xmx4g -Xmn2g -XX:MetaspaceSize=128m -XX:MaxMetaspaceSize=320m"

按照实际机器配置修改1g：

JAVA\_OPT="${JAVA\_OPT} -server -Xms1g -Xmx1g -Xmn512m -XX:MetaspaceSize=128m -XX:MaxMetaspaceSize=320m"

vim runbroker.sh

本来的8g：

JAVA\_OPT="${JAVA\_OPT} -server -Xms8g -Xmx8g -Xmn4g"

按照实际机器配置修改1g：

JAVA\_OPT="${JAVA\_OPT} -server -Xms1g -Xmx1g -Xmn512m"

### 把在主上172.26.103.76配置好的整体推送到从服务器172.26.103.77

rsync -e "ssh -p 522" -avz /usr/local/rocketmq-4.5.0 172.26.103.77:/usr/local/

### 主从两台服务器分别增加环境变量

echo 'export PATH=$PATH:/usr/local/rocketmq-4.5.0/bin' >>/etc/profile

source /etc/profile

### 分别启动主从

注意先启动namesrv再启动broker

必须先cd到bin目录下，不然会报错。

启动主服务器：

cd /usr/local/rocketmq-4.5.0/bin/

启动namesrv

nohup sh mqnamesrv &

再启动broker

nohup sh mqbroker -c /usr/local/rocketmq-4.5.0/conf/2m-2s-sync/broker-a.properties > /dev/null 2>&1 &

启动从服务器：

cd /usr/local/rocketmq-4.5.0/bin/

启动namesrv

nohup sh mqnamesrv &

再启动broker

nohup sh mqbroker -c /usr/local/rocketmq-4.5.0/conf/2m-2s-sync/broker-a-s.properties > /dev/null 2>&1 &

### 两台机器分别检查是否启动成功

jps

ss -lntup #端口9876 10911

查看集群状态

mqadmin clusterList -n 172.26.103.77:9876

如果有rocketmq-console web管理界面，也可以在这看



## 两主两从（同步复制，异步刷盘）

|  |  |
| --- | --- |
| 四台服务器 | 配置1核2G |
| 172.26.103.76 | a主 |
| 172.26.103.77 | a从 |
| 172.26.103.78 | b主 |
| 172.26.103.79 | b从 |

### 上传包到主172.26.103.76服务器

rocketmq-all-4.5.0-bin-release.zip

unzip rocketmq-all-4.5.0-bin-release.zip

mv rocketmq-all-4.5.0-bin-release /usr/local/rocketmq-4.5.0

### 四台 服务器分别配置jdk

yum install java-1.8.0-openjdk java-1.8.0-openjdk-devel -y

所有的jdk，不管是yum安装的还是解压缩的，都要添加到环境变量，/bin/tools.sh 会通过环境变量来运行java

yum安装的openjdk配置环境变量

JAVA\_HOME=/usr/lib/jvm/java

JRE\_HOME=/usr/lib/jvm/java/jre

PATH=$PATH:$JAVA\_HOME/bin:$JRE\_HOME/bin

CLASSPATH=.:$JAVA\_HOME/lib/dt.jar:$JAVA\_HOME/lib/tools.jar:$JRE\_HOME/lib

export JAVA\_HOME JRE\_HOME PATH CLASSPATH

source /etc/profile

java -version

jps

echo $JAVA\_HOME

### 创建RocketMQ存储文件的目录(主172.26.103.76上面操作)

cd /usr/local/rocketmq-4.5.0/

mkdir logs

mkdir store

cd store/

mkdir commitlog

mkdir consumequeue

mkdir index

文件夹说明：

logs:存储RocketMQ日志目录

store:存储RocketMQ数据文件目录

commitlog:存储RocketMQ消息信息

consumequeue、index:存储消息的索引数据

### 更改上传到主服务器的配置文件

先修改a主配置文件

vi /usr/local/rocketmq-4.5.0/conf/2m-2s-sync/broker-a.properties

主要需要注意修改的：

brokerClusterName #所属集群名字

brokerName #broker名字，同一个主从的名字应相同，双主的应改为broker-a,broker-b

brokerId #0 表示Master, > 0 表示slave

namesrvAddr #nameServer 地址，分号分割 172.26.103.76:9876;172.26.103.77:9876

storePathRootDir #存储路径

brokerRole # Broker 的角色

flushDiskType #刷盘方式

#所属集群名字

brokerClusterName=rocketmq-cluster

#broker名字，注意此处不同的配置文件填写的不一样

brokerName=broker-a

#0 表示Master, > 0 表示slave

brokerId=0

#nameServer 地址，分号分割

namesrvAddr=172.26.103.76:9876;172.26.103.77:9876;172.26.103.78:9876;172.26.103.79:9876

#在发送消息时，自动创建服务器不存在的Topic，默认创建的队列数

defaultTopicQueueNums=4

#是否允许Broker 自动创建Topic，建议线下开启，线上关闭

autoCreateTopicEnable=true

#是否允许Broker自动创建订阅组，建议线下开启，线上关闭

autoCreateSubscriptionGroup=true

#Broker 对外服务的监听端口

listenPort=10911

#删除文件时间点，默认是凌晨4点

deleteWhen=04

#文件保留时间，默认48小时

fileReservedTime=120

#commitLog每个文件的大小默认1G

mapedFileSizeCommitLog=1073741824

#ConsumeQueue每个文件默认存30W条，根据业务情况调整

mapedFileSizeConsumeQueue=300000

#destroyMapedFileIntervalForcibly=120000

#redeleteHangedFileInterval=120000

#设置物理文件磁盘最大使用率

#diskMaxUsedSpaceRatio=88

#磁盘空间警戒阈值，超过这个值则停止接受消息，默认值90

#diskSpaceWarningLevelRatio=90

#强制删除文件阈值，默认85

#diskSpaceCleanForciblyRatio=85

#存储路径

storePathRootDir=/usr/local/rocketmq-4.5/store

#commitLog存储路径

storePathCommitLog=/usr/local/rocketmq-4.5/store/commitlog

#消费队列存储路径

storePathConsumeQueue=/usr/local/rocketmq-4.5/store/consumequeue

#消息索引存储路径

storePathIndex=/usr/local/rocketmq-4.5/store/index

#checkpoint 文件存储路径

storeCheckpoint=/usr/local/rocketmq-4.5/store/checkpoint

#abort 文件存储路径

abortFile=/usr/local/rocketmq-4.5/store/abort

#限制的消息大小

maxMessageSize=65536

# flushCommitLogLeastPages=4

# flushConsumeQueueLeastPages=2

# flushCommitLogThoroughInterval=10000

# flushConsumeQueueThoroughInterval=60000

# Broker 的角色

# - ASYNC\_MASTER 异步复制Master

# - SYNC\_MASTER 同步双写Master

# - SLAVE

brokerRole=SYNC\_MASTER

# 刷盘方式

# - ASYNC\_FLUSH 异步刷盘

# - SYNC\_FLUSH 同步刷盘

flushDiskType=ASYNC\_FLUSH

#checkTransactionMessageEnable=false

#发消息线程池数量

#sendMessageTreadPoolNums=128

#拉消息线程池数量

#pullMessageTreadPoolNums=128

修改a从配置文件（还是在172.26.103.76主服务器上操作）

cat /usr/local/rocketmq-4.5.0/conf/2m-2s-sync/broker-a.properties >/usr/local/rocketmq-4.5.0/conf/2m-2s-sync/broker-a-s.properties

vim /usr/local/rocketmq-4.5.0/conf/2m-2s-sync/broker-a-s.properties

配置文件与主一样

主要需要注意修改的：

brokerId #0 表示Master, > 0 表示slave，设置为1

brokerRole # 从得Broker 的角色为SLAVE

修改b主配置文件（还是在172.26.103.76主服务器上操作）

cat /usr/local/rocketmq-4.5.0/conf/2m-2s-sync/broker-a.properties >/usr/local/rocketmq-4.5.0/conf/2m-2s-sync/broker-b.properties

vim /usr/local/rocketmq-4.5.0/conf/2m-2s-sync/broker-b.properties

配置文件与a主一样

主要需要注意修改的：

brokerId #0 表示Master, > 0 表示slave，

brokerName #broker-a 修改为broker-b

修改b从配置文件（还是在172.26.103.76主服务器上操作）

cat /usr/local/rocketmq-4.5.0/conf/2m-2s-sync/broker-a-s.properties >/usr/local/rocketmq-4.5.0/conf/2m-2s-sync/broker-b-s.properties

vim /usr/local/rocketmq-4.5.0/conf/2m-2s-sync/broker-b-s.properties

配置文件与a从一样

主要需要注意修改的：

brokerName #broker-a 修改为broker-b

### 进入conf目录，替换所有xml中的${user.home},保证日志路径正确（还是在172.26.103.76主服务器上操作）

sed -i 's#${user.home}#/usr/local/rocketmq-4.5.0#g' \*.xml

### 进入bin目录，修改配置(分别修改runserver.sh 以及 runbroker.sh，因为默认配置内存过大，可能导致启动失败)

RocketMQ对内存的要求比较高，最少1G，如果内存太少，会影响RocketMQ的运行效率和执行性能。我们需要修改bin目录下的runbroker.sh和runserver.sh文件

cd /usr/local/rocketmq-4.5/bin

vim runserver.sh

本来的4g：

JAVA\_OPT="${JAVA\_OPT} -server -Xms4g -Xmx4g -Xmn2g -XX:MetaspaceSize=128m -XX:MaxMetaspaceSize=320m"

按照实际机器配置修改1g：

JAVA\_OPT="${JAVA\_OPT} -server -Xms1g -Xmx1g -Xmn512m -XX:MetaspaceSize=128m -XX:MaxMetaspaceSize=320m"

vim runbroker.sh

本来的8g：

JAVA\_OPT="${JAVA\_OPT} -server -Xms8g -Xmx8g -Xmn4g"

按照实际机器配置修改1g：

JAVA\_OPT="${JAVA\_OPT} -server -Xms1g -Xmx1g -Xmn512m"

### 把在主上172.26.103.76配置好的整体推送到其他三台服务器

rsync -e "ssh -p 522" -avz /usr/local/rocketmq-4.5.0 172.26.103.77:/usr/local/

rsync -e "ssh -p 522" -avz /usr/local/rocketmq-4.5.0 172.26.103.78:/usr/local/

rsync -e "ssh -p 522" -avz /usr/local/rocketmq-4.5.0 172.26.103.79:/usr/local/

### 分别在四台服务器上配置环境变量

echo 'export PATH=$PATH:/usr/local/rocketmq-4.5.0/bin' >>/etc/profile

source /etc/profile

### 四台分别启动

注意先启动namesrv再启动broker

必须先cd到bin目录下，不然会报错

172.26.103.76 a主启动

cd /usr/local/rocketmq-4.5.0/bin/

启动namesrv

nohup sh mqnamesrv &

再启动broker

nohup sh mqbroker -c /usr/local/rocketmq-4.5.0/conf/2m-2s-sync/broker-a.properties >>/dev/null 2>&1 &

172.26.103.77 a从启动

cd /usr/local/rocketmq-4.5.0/bin/

启动namesrv

nohup sh mqnamesrv &

再启动broker

nohup sh mqbroker -c /usr/local/rocketmq-4.5.0/conf/2m-2s-sync/broker-a-s.properties >>/dev/null 2>&1 &

172.26.103.78 b主启动

cd /usr/local/rocketmq-4.5.0/bin/

启动namesrv

nohup sh mqnamesrv &

再启动broker

nohup sh mqbroker -c /usr/local/rocketmq-4.5.0/conf/2m-2s-sync/broker-b.properties >>/dev/null 2>&1 &

172.26.103.79 b从启动

cd /usr/local/rocketmq-4.5.0/bin/

启动namesrv

nohup sh mqnamesrv &

再启动broker

nohup sh mqbroker -c /usr/local/rocketmq-4.5.0/conf/2m-2s-sync/broker-b-s.properties >>/dev/null 2>&1 &

### 查看集群状态

mqadmin clusterList -n 172.26.103.79:9876

如果有rocketmq-console web管理界面，也可以在这看



### 其他的双主配置，以及异步配置，只是配置文件不同，其他都一样的。

# 遇到的问题

## 所有的jdk，不管是yum安装的还是解压缩的，都要添加到环境变量，/bin/tools.sh 会通过环境变量来运行java

yum安装的openjdk配置环境变量

JAVA\_HOME=/usr/lib/jvm/java

JRE\_HOME=/usr/lib/jvm/java/jre

PATH=$PATH:$JAVA\_HOME/bin:$JRE\_HOME/bin

CLASSPATH=.:$JAVA\_HOME/lib/dt.jar:$JAVA\_HOME/lib/tools.jar:$JRE\_HOME/lib

export JAVA\_HOME JRE\_HOME PATH CLASSPATH

<https://www.jianshu.com/p/17fb50162d3d>

执行命令报错

## 别人的迁移经历

<https://www.cnblogs.com/zhyg/p/10132598.html>

# 常用命令

<https://blog.csdn.net/gwd1154978352/article/details/80829534>

## 查看帮助

查看帮助：

sh mqadmin help 命令名称

例如，查看updateTopic的使用

sh mqadmin help updateTopic

## 集群相关命令

查看集群信息：

-m 打印更多信息 (增加打印出如下信息 #InTotalYest, #OutTotalYest, #InTotalToday ,#OutTotalToday)

-n nameserve服务地址列表，格式ip:port;ip:port;

-h 打印帮助

mqadmin clusterList -n 10.45.47.168:9876