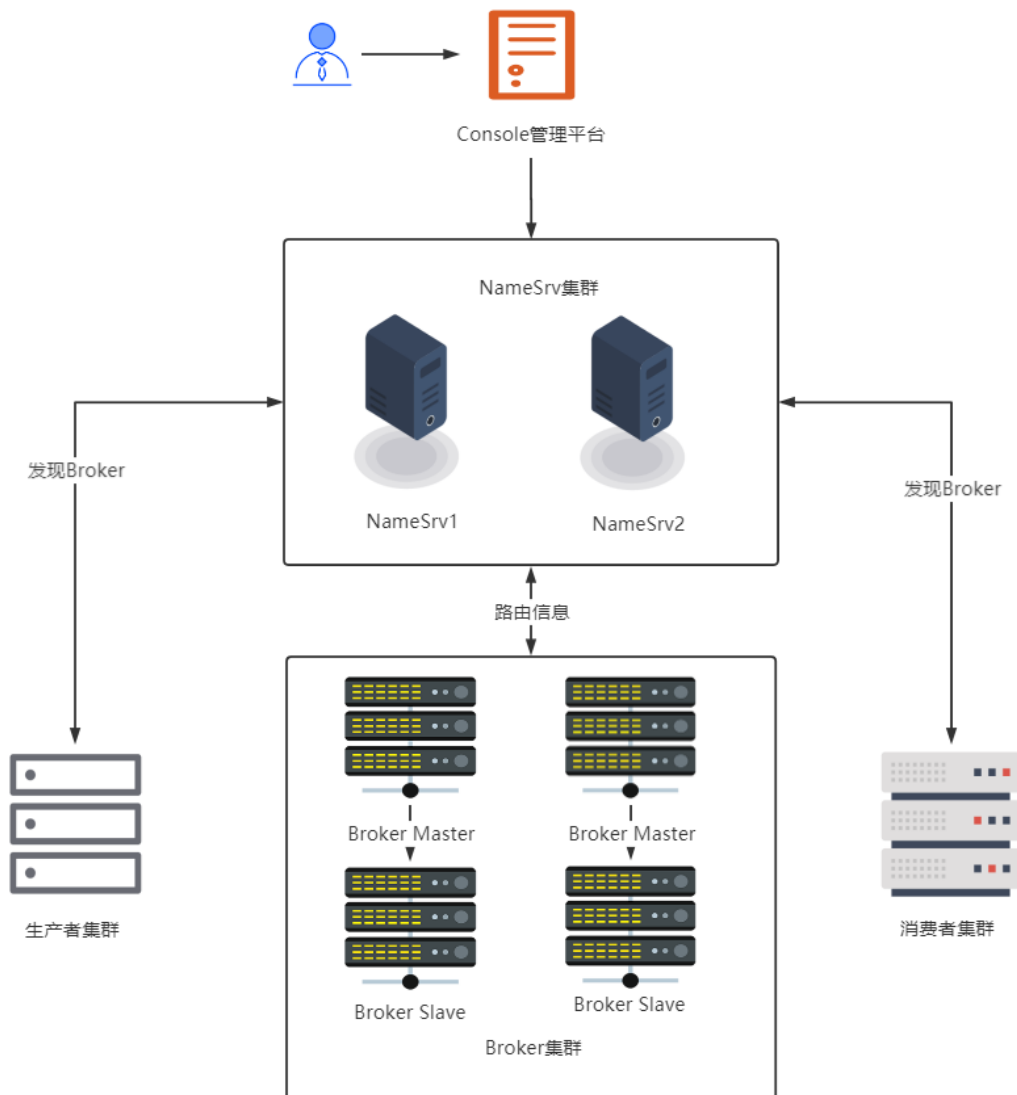


RocketMQ单Master部署

RocketMQ生产拓扑结构



- 使用者：一般是指生产、消费程序的直接研发人员、RocketMQ中间件的维护人员等。
- Console管理平台：管理RocketMQ生产者组、Topic、消费者组和RocketMQ元数据的平台。管理平台可以自研，也可以基于社区提供的 RocketMQ Console二次开发而来，或者直接使用社区提供的RocketMQ Console。
- Namesrv集群：一个无状态的元数据管理，Namesrv相当于RocketMQ的注册中心，提供节点发现。
- Broker 集群：消息中间商或消息代理。主要用于保存消息，处理生产者、消费者的各种请求的代理。包含Master和Slave两种角色，与MySQL中的主从角色类似。
- 生产者集群：消息发送方，通常由一个或多个生产者实例组成。

- 消费者集群：消息接收方，通常由一个或多个消费者实例组成。

RocketMQ的部署模式

Namesrv部署

推荐一个集群并部署2~3个Namesrv节点。

Broker集群部署

- 第一种：单 Master。“集群”中只有一个节点，宕机后不可用。通常用于个人入门学习，比如测试发送消息代码、测试消费消息代码等，建议在生产环境中不要使用这种部署方式。
- 第二种：单 Master，单 Slave。单主从模式，Master 宕机后集群不可写入消息，但可以读取消息。通常用于个人深入学习，比如研究源码、设计原理等，建议在生产环境中不要使用这种部署方式。
- 第三种：多 Master，无 Slave。该种部署方式性能最好，并且当单个 Master 节点宕机时，不影响正常使用。
- 第四种：多Master、多Slave，异步复制。在第三种方式上增加了Slave，当一个Master节点宕机时，该Master不能写入消息，消费可以在其对应的Slave上进行。新消息的生产、消费不受影响。添加Slave后，消费者可以从对应的Slave中读取已发送到宕机Master中的消息。生产环境中可以使用这种部署方式。
- 第五种：多 Master、多 Slave，同步复制。这种部署方式完全解决了第四种部署方式的弊端，虽然由于Master-Slave同步复制导致发送消息耗时增加，集群性能大大下降，但是这仍然是最可靠的部署方式。生产环境中可以使用这种部署方式。

RocketMQ单Master部署

部署NameServer

修改IP地址到103

```
cd /etc/sysconfig/network-scripts
sed -i 's/dhcp/static/g' ifcfg-ens33
sudo cat >> ifcfg-ens33 <<-'EOF'
IPADDR=192.168.31.103
GATEWAY=192.168.31.1
NETMASK=255.255.255.0
DNS1=114.114.114.114
DNS2=8.8.8.8
EOF
service network restart
```

1. 环境准备CentOS 7.x系统，YUM安装OpenJDK 1.8，并配置

```
yum -y install java-1.8.0-openjdk-devel.x86_64
sudo cat >> /etc/profile <<-'EOF'
```

```
export JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk
export JRE_HOME=$JAVA_HOME/jre
export CLASSPATH=$JAVA_HOME/lib:$JRE_HOME/lib:$CLASSPATH
export PATH=$JAVA_HOME/bin:$JRE_HOME/bin:$PATH
EOF
```

```
source /etc/profile
echo $JAVA_HOME
```

```
cd /usr/local
# 官网下的太慢，用我提供的OSS加速下载
# wget --no-check-certificate https://manongbiji.oss-cn-
beijing.aliyuncs.com/ittailkshow/rocketmq/download/rocketmq-all-4.9.2-bin-
release.zip
wget --no-check-certificate https://manongbiji.oss-cn-
beijing.aliyuncs.com/ittailkshow/rocketmq/download/rocketmq-all-4.9.2-bin-
release.zip
```

```
unzip rocketmq-all-4.9.2-bin-release.zip
cd /usr/local/rocketmq-4.9.2
sed -i 's/-Xms4g -Xmx4g -XX:MetaspaceSize=128m/-Xms1g -Xmx1g -
XX:MetaspaceSize=128m/g' ./bin/runserver.sh
firewall-cmd --zone=public --add-port=9876/tcp --permanent
firewall-cmd --reload
# sh ./bin/mqnamesrv
nohup sh ./bin/mqnamesrv > nqnamesrv.log &
```

部署单Master

```
cd /etc/sysconfig/network-scripts
sed -i 's/dhcp/static/g' ifcfg-ens33
sudo cat >> ifcfg-ens33 <<-'EOF'
IPADDR=192.168.31.105
GATEWAY=192.168.31.1
NETMASK=255.255.255.0
DNS1=114.114.114.114
DNS2=8.8.8.8
EOF
service network restart
```

```
yum -y install java-1.8.0-openjdk-devel.x86_64
sudo cat >> /etc/profile <<-'EOF'
export JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk
export JRE_HOME=$JAVA_HOME/jre
export CLASSPATH=$JAVA_HOME/lib:$JRE_HOME/lib:$CLASSPATH
```

```
export PATH=$JAVA_HOME/bin:$JRE_HOME/bin:$PATH
EOF
source /etc/profile
echo $JAVA_HOME
```

```
cd /usr/local
# 官网下的太慢，用我提供的OSS加速下载
# wget --no-check-certificate https://dlcdn.apache.org/rocketmq/4.9.2/rocketmq-
all-4.9.2-bin-release.zip
wget --no-check-certificate https://manongbiji.oss-cn-
beijing.aliyuncs.com/ittailkshow/rocketmq/download/rocketmq-all-4.9.2-bin-
release.zip
unzip rocketmq-all-4.9.2-bin-release.zip
cd /usr/local/rocketmq-4.9.2
sed -i 's/-Xms8g -Xmx8g/-Xms1g -Xmx1g/g' ./bin/runbroker.sh
```

```
cat > single-master.properties <<-'EOF'
#集群名称，同一个集群下的broker要求统一
brokerClusterName=DefaultCluster
```

```
#broker名称
brokerName=broker-a
```

```
#brokerId=0代表主节点，大于零代表从节点
brokerId=0
```

```
#删除日志文件时间点，默认凌晨 4 点
deleteWhen=04
```

```
#日志文件保留时间，默认 48 小时
fileReservedTime=48
```

```
#Broker 的角色
#- ASYNC_MASTER 异步复制Master
#- SYNC_MASTER 同步双写Master
brokerRole=SYNC_MASTER
```

```
#刷盘方式
#- ASYNC_FLUSH 异步刷盘，性能好宕机会丢数
#- SYNC_FLUSH 同步刷盘，性能较差不会丢数
flushDiskType=SYNC_FLUSH
```

```
#末尾追加，NameServer节点列表，使用分号分割
namesrvAddr=192.168.31.103:9876
```

```
autoCreateTopicEnable=true  
EOF
```

```
firewall-cmd --zone=public --add-port=10911/tcp --permanent  
firewall-cmd --reload
```

```
# sh ./bin/mqbroker -c ./single-master.properties  
nohup sh ./bin/mqbroker -c ./single-master.properties > broker.log &
```

```
cd /usr/local/rocketmq-4.9.2/bin  
sh mqadmin clusterList -n 192.168.31.103:9876
```

```
cd /usr/local/rocketmq-4.9.2/bin  
export NAMESRV_ADDR=192.168.31.103:9876  
sh tools.sh org.apache.rocketmq.example.quickstart.Producer
```

```
cd /usr/local/rocketmq-4.9.2/bin  
export NAMESRV_ADDR=192.168.31.103:9876  
sh tools.sh org.apache.rocketmq.example.quickstart.Consumer
```