



版本：5.0

部署方式

Apache RocketMQ 5.0 版本完成基本消息收发，包括 NameServer、Broker、Proxy 组件。在 5.0 版本中 Proxy 和 Broker 根据实际诉求可以分为 Local 模式和 Cluster 模式，一般情况下如果没有特殊需求，或者遵循从早期版本平滑升级的思路，可以选用 Local 模式。

- 在 Local 模式下，Broker 和 Proxy 是同进程部署，只是在原有 Broker 的配置基础上新增 Proxy 的简易配置就可以运行。
- 在 Cluster 模式下，Broker 和 Proxy 分别部署，即在原有的集群基础上，额外再部署 Proxy 即可。

Local 模式部署

由于 Local 模式下 Proxy 和 Broker 是同进程部署，Proxy 本身无状态，因此主要的集群配置仍然以 Broker 为基础进行即可。

启动 NameServer

NameServer 需要先于 Broker 启动，且如果在生产环境使用，为了保证高可用，建议一般规模的集群启动 3 个 NameServer，各节点的启动命令相同，如下：

```
### 首先启动Name Server
$ nohup sh mqnamesrv &

### 验证Name Server 是否启动成功
$ tail -f ~/logs/rocketmqlogs/namesrv.log
The Name Server boot success...
```

启动Broker+Proxy

单组节点单副本模式



警告

这种方式风险较大，因为 Broker 只有一个节点，一旦Broker重启或者宕机时，会导致整个服务不可用。不建议线上环境使用,可以用于本地测试。

启动 Broker+Proxy

```
$ nohup sh bin/mqbroker -n localhost:9876 --enable-proxy &
```

验证Broker 是否启动成功，例如Broker的IP为：192.168.1.2，且名称为broker-a

```
$ tail -f ~/logs/rocketmqlogs/broker_default.log
```

```
The broker[xxx,192.169.1.2:10911] boot success...
```

多组节点（集群）单副本模式

一个集群内全部部署 Master 角色，不部署Slave 副本，例如2个Master或者3个Master，这种模式的优缺点如下：

- 优点：配置简单，单个Master宕机或重启维护对应用无影响，在磁盘配置为RAID10时，即使机器宕机不可恢复情况下，由于RAID10磁盘非常可靠，消息也不会丢（异步刷盘丢失少量消息，同步刷盘一条不丢），性能最高；
- 缺点：单台机器宕机期间，这台机器上未被消费的消息在机器恢复之前不可订阅，消息实时性会受到影响。

启动Broker+Proxy集群

在机器A，启动第一个Master，例如NameServer的IP为：192.168.1.1

```
$ nohup sh bin/mqbroker -n 192.168.1.1:9876 -c $ROCKETMQ_HOME/conf/2m-noslave/broker-a.properties --enable-proxy &
```

在机器B，启动第二个Master，例如NameServer的IP为：192.168.1.1

```
$ nohup sh bin/mqbroker -n 192.168.1.1:9876 -c $ROCKETMQ_HOME/conf/2m-noslave/broker-b.properties --enable-proxy &
```

...

① 备注

如上启动命令是在单个NameServer情况下使用的。对于多个NameServer的集群，Broker启动命令中-n后面的地址列表用分号隔开即可，例如 192.168.1.1:9876;192.161.2:9876。

多节点（集群）多副本模式-异步复制

每个Master配置一个Slave，有多组 Master-Slave，HA采用异步复制方式，主备有短暂消息延迟（毫秒级），这种模式的优缺点如下：

- 优点：即使磁盘损坏，消息丢失的非常少，且消息实时性不会受影响，同时Master宕机后，消费者仍然可以从Slave消费，而且此过程对应用透明，不需要人工干预，性能同多Master模式几乎一样；
- 缺点：Master宕机，磁盘损坏情况下会丢失少量消息。

启动Broker+Proxy集群

```
### 在机器A，启动第一个Master，例如NameServer的IP为：192.168.1.1
$ nohup sh bin/mqbroker -n 192.168.1.1:9876 -c $ROCKETMQ_HOME/conf/2m-2s-async/broker-a.properties --enable-proxy &
```

```
### 在机器B，启动第二个Master，例如NameServer的IP为：192.168.1.1
$ nohup sh bin/mqbroker -n 192.168.1.1:9876 -c $ROCKETMQ_HOME/conf/2m-2s-async/broker-b.properties --enable-proxy &
```

```
### 在机器C，启动第一个Slave，例如NameServer的IP为：192.168.1.1
$ nohup sh bin/mqbroker -n 192.168.1.1:9876 -c $ROCKETMQ_HOME/conf/2m-2s-async/broker-a-s.properties --enable-proxy &
```

```
### 在机器D，启动第二个Slave，例如NameServer的IP为：192.168.1.1
$ nohup sh bin/mqbroker -n 192.168.1.1:9876 -c $ROCKETMQ_HOME/conf/2m-2s-async/broker-b-s.properties --enable-proxy &
```

多节点（集群）多副本模式-同步双写

每个Master配置一个Slave，有多对 Master-Slave，HA采用同步双写方式，即只有主备都写成功，才向应用返回成功，这种模式的优缺点如下：

- 优点：数据与服务都无单点故障，Master宕机情况下，消息无延迟，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（大约低10%左右），发送单个消息的RT会略高，且目前版本在主节点宕机后，备机不能自动切换为主机。

启动 Broker+Proxy 集群

在机器A, 启动第一个Master, 例如NameServer的IP为: 192.168.1.1

```
$ nohup sh bin/mqbroker -n 192.168.1.1:9876 -c $ROCKETMQ_HOME/conf/2m-2s-sync/broker-a.properties --enable-proxy &
```

在机器B, 启动第二个Master, 例如NameServer的IP为: 192.168.1.1

```
$ nohup sh bin/mqbroker -n 192.168.1.1:9876 -c $ROCKETMQ_HOME/conf/2m-2s-sync/broker-b.properties --enable-proxy &
```

在机器C, 启动第一个Slave, 例如NameServer的IP为: 192.168.1.1

```
$ nohup sh bin/mqbroker -n 192.168.1.1:9876 -c $ROCKETMQ_HOME/conf/2m-2s-sync/broker-a-s.properties --enable-proxy &
```

在机器D, 启动第二个Slave, 例如NameServer的IP为: 192.168.1.1

```
$ nohup sh bin/mqbroker -n 192.168.1.1:9876 -c $ROCKETMQ_HOME/conf/2m-2s-sync/broker-b-s.properties --enable-proxy &
```

提示

以上 Broker 与 Slave 配对是通过指定相同的 BrokerName 参数来配对, Master 的 BrokerId 必须是 0, Slave 的 BrokerId 必须是大于 0 的数。另外一个 Master 下面可以挂载多个 Slave, 同一 Master 下的多个 Slave 通过指定不同的 BrokerId 来区分。\$ROCKETMQ_HOME指的RocketMQ安装目录, 需要用户自己设置此环境变量。

5.0 HA新模式

提供更具灵活性的HA机制, 让用户更好的平衡成本、服务可用性、数据可靠性, 同时支持业务消息和流存储的场景。 [详见](#)

Cluster模式部署

在 Cluster 模式下, Broker 与 Proxy分别部署, 我可以在 NameServer和 Broker都启动完成之后再部署 Proxy。

在 Cluster模式下, 一个 Proxy集群和 Broker集群为一一对应的关系, 可以在 Proxy的配置文件 `rmq-proxy.json` 中使用 `rocketMQClusterName` 进行配置

启动 NameServer

```
### 首先启动Name Server
```

```
$ nohup sh mqnamesrv &
```

```
### 验证Name Server 是否启动成功
```

```
$ tail -f ~/logs/rocketmqlogs/namesrv.log
```

```
The Name Server boot success...
```

启动 Broker

单组节点单副本模式

警告

这种方式风险较大，因为 Broker 只有一个节点，一旦Broker重启或者宕机时，会导致整个服务不可用。不建议线上环境使用,可以用于本地测试。

```
### 在机器A，启动第一个Master，例如NameServer的IP为：192.168.1.1
```

```
$ nohup sh bin/mqbroker -n 192.168.1.1:9876 &
```

多组节点（集群）单副本模式

一个集群内全部部署 Master 角色，不部署Slave 副本，例如2个Master或者3个Master，这种模式的优缺点如下：

- 优点：配置简单，单个Master宕机或重启维护对应用无影响，在磁盘配置为RAID10时，即使机器宕机不可恢复情况下，由于RAID10磁盘非常可靠，消息也不会丢（异步刷盘丢失少量消息，同步刷盘一条不丢），性能最高；
- 缺点：单台机器宕机期间，这台机器上未被消费的消息在机器恢复之前不可订阅，消息实时性会受到影响。

```
### 在机器A，启动第一个Master，例如NameServer的IP为：192.168.1.1
```

```
$ nohup sh bin/mqbroker -n 192.168.1.1:9876 -c $ROCKETMQ_HOME/conf/2m-noslave/broker-a.properties &
```

```
### 在机器B，启动第二个Master，例如NameServer的IP为：192.168.1.1
```

```
$ nohup sh bin/mqbroker -n 192.168.1.1:9876 -c $ROCKETMQ_HOME/conf/2m-noslave/broker-b.properties &
```

...

备注

如上启动命令是在单个NameServer情况下使用的。对于多个NameServer的集群，Broker启动命令中(-n)后面的地址列表用分号隔开即可，例如 192.168.1.1:9876;192.161.2:9876。

多节点（集群）多副本模式-异步复制

每个Master配置一个Slave，有多组 Master-Slave，HA采用异步复制方式，主备有短暂消息延迟（毫秒级），这种模式的优缺点如下：

- 优点：即使磁盘损坏，消息丢失的非常少，且消息实时性不会受影响，同时Master宕机后，消费者仍然可以从Slave消费，而且此过程对应用透明，不需要人工干预，性能同多Master模式几乎一样；
- 缺点：Master宕机，磁盘损坏情况下会丢失少量消息。

在机器A，启动第一个Master，例如NameServer的IP为：192.168.1.1

```
$ nohup sh bin/mqbroker -n 192.168.1.1:9876 -c $ROCKETMQ_HOME/conf/2m-2s-async/broker-a.properties &
```

在机器B，启动第二个Master，例如NameServer的IP为：192.168.1.1

```
$ nohup sh bin/mqbroker -n 192.168.1.1:9876 -c $ROCKETMQ_HOME/conf/2m-2s-async/broker-b.properties &
```

在机器C，启动第一个Slave，例如NameServer的IP为：192.168.1.1

```
$ nohup sh bin/mqbroker -n 192.168.1.1:9876 -c $ROCKETMQ_HOME/conf/2m-2s-async/broker-a-s.properties &
```

在机器D，启动第二个Slave，例如NameServer的IP为：192.168.1.1

```
$ nohup sh bin/mqbroker -n 192.168.1.1:9876 -c $ROCKETMQ_HOME/conf/2m-2s-async/broker-b-s.properties &
```

多节点（集群）多副本模式-同步双写

每个Master配置一个Slave，有多对 Master-Slave，HA采用同步双写方式，即只有主备都写成功，才向应用返回成功，这种模式的优缺点如下：

- 优点：数据与服务都无单点故障，Master宕机情况下，消息无延迟，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（大约低10%左右），发送单个消息的RT会略高，且目前版本在主节点宕机后，备机不能自动切换为主机。

在机器A，启动第一个Master，例如NameServer的IP为：192.168.1.1

```
$ nohup sh bin/mqbroker -n 192.168.1.1:9876 -c $ROCKETMQ_HOME/conf/2m-2s-sync/broker-a.properties &
```

在机器B，启动第二个Master，例如NameServer的IP为：192.168.1.1

```
$ nohup sh bin/mqbroker -n 192.168.1.1:9876 -c $ROCKETMQ_HOME/conf/2m-2s-sync/broker-b.properties &
```

在机器C，启动第一个Slave，例如NameServer的IP为：192.168.1.1

```
$ nohup sh bin/mqbroker -n 192.168.1.1:9876 -c $ROCKETMQ_HOME/conf/2m-2s-sync/broker-a-s.properties &
```

在机器D，启动第二个Slave，例如NameServer的IP为：192.168.1.1

```
$ nohup sh bin/mqbroker -n 192.168.1.1:9876 -c $ROCKETMQ_HOME/conf/2m-2s-sync/broker-b-s.properties &
```

提示

以上 Broker 与 Slave 配对是通过指定相同的 BrokerName 参数来配对，Master 的 BrokerId 必须是 0，Slave 的 BrokerId 必须是大于 0 的数。另外一个 Master 下面可以挂载多个 Slave，同一 Master 下的多个 Slave 通过指定不同的 BrokerId 来区分。\$ROCKETMQ_HOME指的RocketMQ安装目录，需要用户自己设置此环境变量。

5.0 HA新模式

提供更具灵活性的HA机制，让用户更好的平衡成本、服务可用性、数据可靠性，同时支持业务消息和流存储的场景。[详见](#)

启动 Proxy

可以在多台机器启动多个Proxy

在机器A，启动第一个Proxy，例如NameServer的IP为：192.168.1.1

```
$ nohup sh bin/mqproxy -n 192.168.1.1:9876 &
```

在机器B，启动第二个Proxy，例如NameServer的IP为：192.168.1.1

```
$ nohup sh bin/mqproxy -n 192.168.1.1:9876 &
```

```
### 在机器C, 启动第三个Proxy, 例如NameServer的IP为: 192.168.1.1  
$ nohup sh bin/mqproxy -n 192.168.1.1:9876 &
```

若需要指定配置文件, 可以使用 `-pc` 或者 `--proxyConfigPath` 进行指定

```
### 自定义配置文件  
$ nohup sh bin/mqproxy -n 192.168.1.1:9876 -pc /path/to/proxyConfig.json &
```

 [编辑此页](#)