



APUSIC
固若长城
睿比世界

安装手册

金蝶Apusic分布式消息队列V2.0.6_for_rocketmq

版权声明

本文档所涉及的软件著作权、版权等知识产权已依法进行了注册，由金蝶天燕云计算股份有限公司合法拥有。受《中华人民共和国著作权法》《计算机软件保护条例》《知识产权保护条例》和相关国际版权条约、法律、法规以及其它知识产权法律和条约的保护。未经授权许可，不得非法使用。

免责声明

本文档包含的版权信息由金蝶天燕云计算股份有限公司合法拥有，受法律的保护，金蝶天燕云计算股份有限公司对本文档可能涉及到的非金蝶天燕云计算股份有限公司的信息不承担任何责任。在法律允许的范围内，您可以查阅并仅能够在《中华人民共和国著作权法》规定的合法范围内复制和打印本文档。任何单位和个人未经金蝶天燕云计算股份有限公司书面授权许可，不得使用、修改、再发布本文档的任何部分和内容，否则将被视为侵权，金蝶天燕云计算股份有限公司有依法追究其责任的权利。

本文档如有更新，不另行通知。对本文档中的问题您可向金蝶天燕云计算股份有限公司告知或查询。未经本公司明确授予的任何权利均予保留。

商标声明

 是深圳市金蝶天燕云计算股份有限公司向中华人民共和国国家商标局申请注册的注册商标，注册商标专用权由金蝶天燕合法拥有，受法律保护。未经金蝶天燕的书面许可，任何单位及个人不得以任何方式或理由对该商标的任何部分进行使用、复制、修改、传播、抄录或与其它产品捆绑使用销售。凡侵犯金蝶天燕商标权的，金蝶天燕将依法追究其法律责任。本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

目录

- 1 前言
 - 1.1 适用对象
 - 1.2 相关文档
 - 1.3 技术支持
- 2 部署架构
 - 2.1 ADMQ for RocketMQ网络部署特点
- 3 环境准备
 - 3.1 系统要求
 - 3.2 硬件推荐
 - 3.3 依赖软件
 - 3.4 网络要求
- 4 单机部署
 - 4.1 安装 JDK
 - 4.2 安装 ADMQ for RocketMQ
 - 4.3 启动 NameServer
 - 4.4 启动 Broker
 - 4.5 验证安装
- 5 单机部署（单主部署）
 - 5.1 NameServer 部署
 - 5.2 Broker 部署
- 6 双主部署
 - 6.1 部署架构图
 - 6.2 NameServer 部署
 - 6.3 Broker 部署
- 7 多主多从部署
 - 7.1 部署架构图
 - 7.2 NameServer 部署
 - 7.3 Broker 配置
- 8 管控台部署
 - 8.1 安装管控台
 - 8.2 访问管控台
 - 8.3 初始化集群

• 9 常见问题及排查

- 9.1 NameServer 启动失败
- 9.2 Broker 启动失败
- 9.3 管控台无法访问
- 9.4 消息发送失败

1 前言

本文档为金蝶Apusic分布式消息队列for RocketMQ（Apusic Distributed Message Queue for RocketMQ，简称：ADMQ for RocketMQ）的安装手册，提供产品各种部署模式（含单机、双主、多主多从）在各操作系统、容器环境的安装部署过程指导以及相关配置说明。

1.1 适用对象

本文档适用于IT信息化业务负责人、研发经理、软件项目经理、软件架构师、运维工程师。

1.2 相关文档

了解更多ADMQ for RocketMQ产品相关的信息，请参阅以下ADMQ for RocketMQ产品手册文档集：

序号	手册文档	说明
1	金蝶Apusic分布式消息队列for RocketMQ 快速使用手册	简单介绍了如何快速上手使用ADMQ for RocketMQ。
2	金蝶Apusic分布式消息队列for RocketMQ 安装手册	详细介绍如何在各操作系统上安装ADMQ for RocketMQ，以及ADMQ for RocketMQ服务启停等操作。
3	金蝶Apusic分布式消息队列for RocketMQ 消息引擎用户手册	详细介绍 ADMQ for RocketMQ 消息引擎相关功能的使用、配置、管理及配套工具的使用方法。
4	金蝶Apusic分布式消息队列for RocketMQ 管控台用户手册	详细介绍ADMQ for RocketMQ管控台相关功能的使用和操作说明。
5	金蝶Apusic分布式消息队列for RocketMQ 开发手册	详细介绍基于各开发语言进行ADMQ for RocketMQ客户端应用开发的说明。
6	金蝶Apusic分布式消息队列for RocketMQ 迁移手册	详细介绍从RocketMQ迁移到ADMQ for RocketMQ的说明。
7	金蝶Apusic分布式消息队列for RocketMQ 运维手册	详细介绍ADMQ for RocketMQ的监控、运维、安全加固等运维说明。
8	金蝶Apusic分布式消息队列for RocketMQ 性能优化手册	详细介绍ADMQ for RocketMQ性能调优的说明。

1.3 技术支持

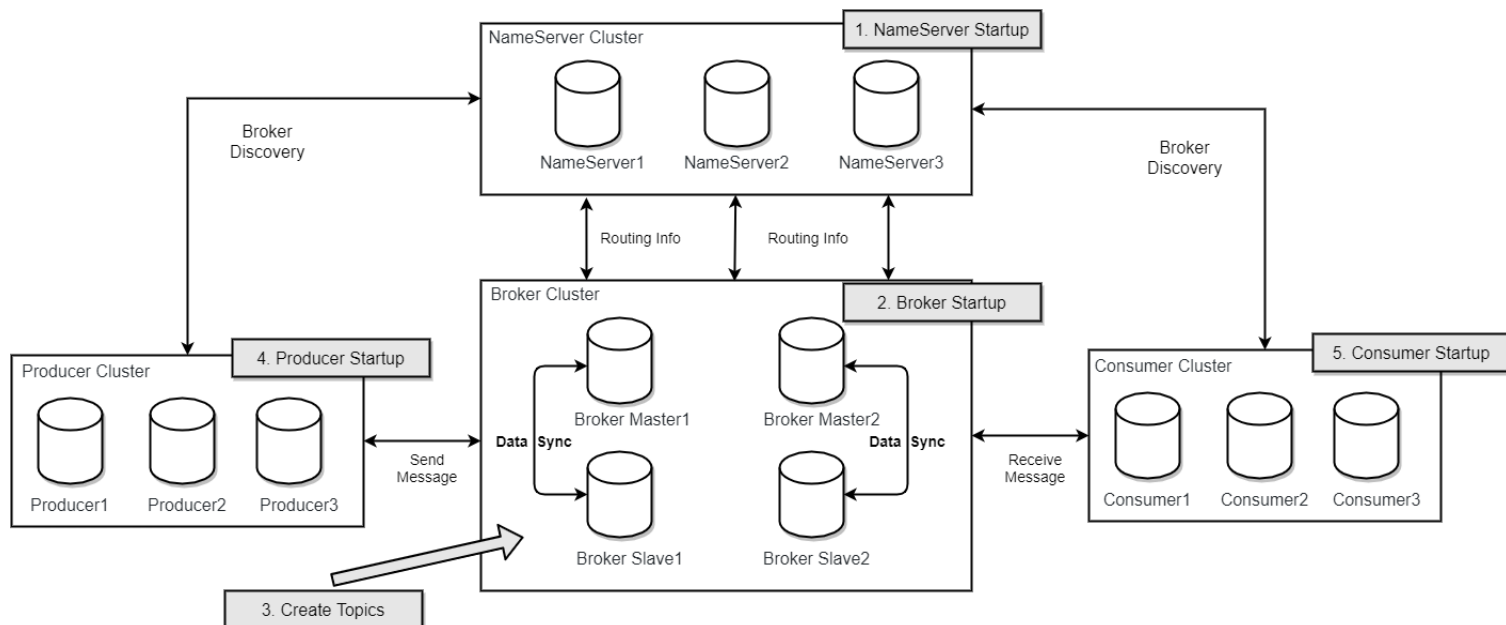
ADMQ for RocketMQ产品提供全面的技术支持服务，您可以通过以下方式获得技术支持：

- 网址：www.apusic.com
- 电话：400-855-5800
- 邮箱：support@apusic.com
- 金蝶云社区：<https://vip.kingdee.com/?productId=73&productLineId=14&lang=zh-CN>

您在取得技术支持时，请提供如下信息：

1. 您的姓名
2. 公司与联系方式
3. 操作系统及其版本
4. 产品版本号
5. 出现异常及错误的日志、截图等详细信息

2 部署架构



2.1 ADMQ for RocketMQ网络部署特点

- NameServer是一个几乎无状态节点，可集群部署，节点之间无任何信息同步。
- Broker部署相对复杂，Broker分为Master与Slave，一个Master可以对应多个Slave，但是一个Slave只能对应一个Master，Master与Slave的对应关系通过指定相同的BrokerName，不同的BrokerId来定义，BrokerId为0表示Master，非0表示Slave。Master也可以部署多个。每个Broker与NameServer集群中的所有节点建立长连接，定时注册Topic信息到所有NameServer。注意：当前RocketMQ版本在部署架构上支持一Master多Slave，但只有BrokerId=1的从服务器才会参与消息的读负载。
- Producer与NameServer集群中的其中一个节点（随机选择）建立长连接，定期从NameServer获取Topic路由信息，并向提供Topic服务的Master建立长连接，且定时向Master发送心跳。Producer完全无状态，可集群部署。
- Consumer与NameServer集群中的其中一个节点（随机选择）建立长连接，定期从NameServer获取Topic路由信息，并向提供Topic服务的Master、Slave建立长连接，且定时向Master、Slave发送心跳。Consumer既可以从Master订阅消息，也可以从Slave订阅消息，消费者在向Master拉取消息时，Master服务器会根据拉取偏移量与最大偏移量的距离（判断是否读老消息，产生读I/O），以及从服务器是否可读等因素建议下一次是从Master还是Slave拉取。

结合部署架构图，描述集群工作流程：

- 启动NameServer，NameServer起来后监听端口，等待Broker、Producer、Consumer连上来，相当于一个路由控制中心。
- Broker启动，跟所有的NameServer保持长连接，定时发送心跳包。心跳包中包含当前Broker信息(IP+端口等)以及存储所有Topic信息。注册成功后，NameServer集群中就有Topic跟Broker的映射关系。
- 收发消息前，先创建Topic，创建Topic时需要指定该Topic要存储在哪些Broker上，也可以在发送消息时自动创建Topic。
- Producer发送消息，启动时先跟NameServer集群中的其中一台建立长连接，并从NameServer中获取当前发送的Topic存在哪些Broker上，轮询从队列列表中选择一个队列，然后与队列所在的Broker建立长连接从而向Broker发消息。
- Consumer跟Producer类似，跟其中一台NameServer建立长连接，获取当前订阅Topic存在哪些Broker上，然后直接跟Broker建立连接通道，开始消费消息。

3 环境准备

3.1 系统要求

在开始部署之前，请确保您的环境满足以下要求：

- **操作系统**：64 位 Linux（推荐 CentOS 7+、Ubuntu 18.04+，国产Kylin v10+、OpenEuler 22.03+）
- **Java 版本**：JDK 8及以上
- **硬件配置**：
 - 最小生产环境需要 7 台服务器（或虚拟机）
 - 3 台运行 Name Server
 - 4 台运行 Broker（Master 和 Slave 各 2 台）
 - 测试环境可以在一台服务器上运行所有组件
- **网络**：所有节点之间可以相互通信，并具有稳定的网络连接
- **DNS**：可选，为所有 Broker 节点配置一个共同的 DNS 名称

3.2 硬件推荐

下表提供了不同规模集群的硬件推荐（每节点）：

组件	小型集群 (250 个主题)	中型集群 (1000 个主题)
Name Server	• CPU: 4 核 • 内存: 16GB • 磁盘: 200GB SSD	• CPU: 8 核 • 内存: 32GB • 磁盘: 500GB SSD
Broker	• CPU: 8 核 • 内存: 32GB • 磁盘: 500GB SSD	• CPU: 16 核 • 内存: 64GB • 磁盘: 1TB SSD

硬件选择建议

- Name Server 只用于元数据存储，可以使用相对较小的服务器
- Broker 计算需要快速的 CPU 和网络，推荐使用 10Gbps 网络接口；存储需要快速的存储设备，推荐使用 SSD 或带电池缓存的 RAID 控制器

3.3 依赖软件

软件	版本要求	说明
JDK	1.8+	Broker 和 NameServer 运行依赖
Java	17+	管控台运行依赖

3.4 网络要求

端口	用途	说明
9876	NameServer	服务注册与发现
10911	Broker	生产者和消费者连接
10909	Broker VIP	VIP 通道
10912	Broker HA	主从复制端口
12305	管控台 HTTP	管控台 Web 界面 (HTTP)
12306	管控台 HTTPS	管控台 Web 界面 (HTTPS)

4 单机部署

4.1 安装 JDK

```
# CentOS/RHEL
sudo yum install -y java-1.8.0-openjdk

# Ubuntu/Debian
sudo apt-get update
sudo apt-get install -y openjdk-8-jdk
```

4.2 安装 ADMQ for RocketMQ

```
mkdir -p /apusic
cd /apusic
# 解压安装包
#tar zxvf ADMQ-V2.0.6-RocketMQ-构建日期.tar.gz -C /apusic/
tar zxvf ADMQ-V2.0.6.534-RocketMQ-20260625.tar.gz -C /apusic/
cd /apusic
# 建议nameserver broker组件文件夹结构分开
unzip rocketmq-V5.3.4-all.zip
mv rocketmq-V5.3.4-all nameserver
cd /apusic/nameserver

unzip rocketmq-V5.3.4-all.zip
mv rocketmq-V5.3.4-all broker
cd /apusic/broker

# 配置环境变量，可以改成对应机器的地址
export NAMESRV_ADDR=localhost:9876

# 配置授权证书，将对应的授权证书放置到对应组件的根目录，假设license.xml文件（文件名需统一license.xml）放置在/opt下
```

```
cp /apusic/license.xml /apusic/nameserver
cp /apusic/license.xml /apusic/broker
```

4.3 启动 NameServer

```
# 启动 NameServer
bin/rocketmq start nameserver

# 验证状态
tail -f logs/namesrv.log
```

4.4 启动 Broker

```
# 配置 Broker, 有些参数是默认的, 可以根据需要修改, 比如集群的brokerId, 根据节点数来设置
vi conf/broker.conf
```

```
# broker.conf
brokerClusterName=DefaultCluster
# (集群节点的brokerName需要区分)
brokerName=broker-a
brokerId=0
brokerRole=ASYNC_MASTER
flushDiskType=ASYNC_FLUSH
# (如果是集群部署, 需要填写集群所有节点的端口ip)
namesrvAddr=localhost:9876
# Promethues开启
enableMultiDispatch=true
enablemq=true
metricsPromExporterHost=0.0.0.0
metricsExporterType=PROM
```

```
# 启动 Broker
bin/rocketmq start broker

# 验证状态
tail -f logs/broker.log
```

4.5 验证安装

```
# 查看集群状态
bin/mqadmin clusterList -n localhost:9876

# 查看 Topic 列表
bin/mqadmin topicList -n localhost:9876
```

5 单机部署（单主部署）

这种方式风险较大，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。

```
+-----+
| NameServer1 |
+-----+
      |
+-----+
|  Broker-a   |
|   Master    |
+-----+
```

5.1 NameServer 部署

在 node1 上启动 NameServer:

```
# node1
nohup bin/rocketmq start nameserver
```

5.2 Broker 部署

Broker-a (node1)

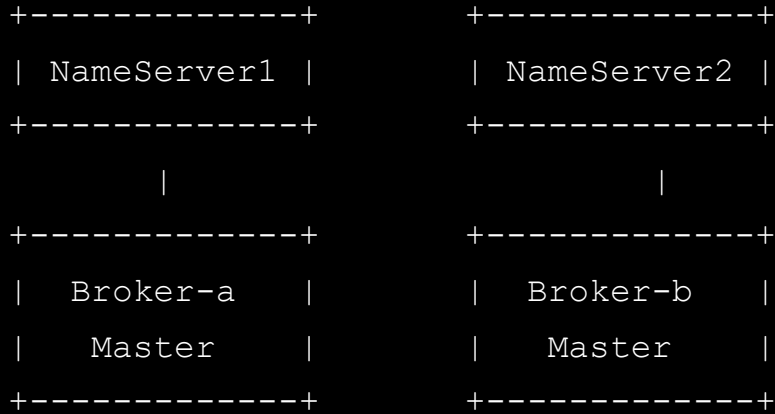
```
# conf/broker-a.properties
brokerClusterName=AdmqCluster
brokerName=broker-a
brokerId=0
brokerRole=SYNC_MASTER
flushDiskType=ASYNC_FLUSH
namesrvAddr=node1:9876
# Promethues开启
enableMultiDispatch=true
enablemq=true
```

```
metricsPromExporterHost=0.0.0.0  
metricsExporterType=PROM
```

```
# node1  
bin/rocketmq start broker
```

6 双主部署

6.1 部署架构图



6.2 NameServer 部署

在 node1 和 node2 上分别启动 NameServer:

```
# node1
bin/rocketmq start nameserver

# node2
bin/rocketmq start nameserver
```

6.3 Broker 部署

Broker-a (node1)

```
# conf/broker-a.properties
brokerClusterName=AdmqCluster
# (集群节点的brokerName需要区分)
brokerName=broker-a
brokerId=0
brokerRole=ASYNC_MASTER
flushDiskType=ASYNC_FLUSH
```

```
# (如果是集群部署, 需要填写集群所有节点的端口ip)
namesrvAddr=node1:9876;node2:9876
# Prometheus开启
enableMultiDispatch=true
enablemq=true
metricsPromExporterHost=0.0.0.0
metricsExporterType=PROM
```

Broker-b (node2)

```
# conf/broker-b.properties
brokerClusterName=AdmqCluster
# (集群节点的brokerName需要区分)
brokerName=broker-b
brokerId=0
brokerRole=ASYNC_MASTER
flushDiskType=ASYNC_FLUSH
# (如果是集群部署, 需要填写集群所有节点的端口ip)
namesrvAddr=node1:9876;node2:9876
# Prometheus开启
enableMultiDispatch=true
enablemq=true
metricsPromExporterHost=0.0.0.0
metricsExporterType=PROM
```

```
# node1
bin/rocketmq start broker

# node2
bin/rocketmq start broker
```

7 多主多从部署

7.1 部署架构图



7.2 NameServer 部署

先启动对应的nameserver组件

```
./bin/rocketmq start nameserver
```

7.3 Broker 配置

Broker-a-Master (node1)

```
brokerClusterName=AdmqCluster
# (集群主节点的brokerName需要区分)
brokerName=broker-a
brokerId=0
brokerRole=ASYNC_MASTER
flushDiskType=ASYNC_FLUSH
# (如果是集群部署, 需要填写集群所有节点的端口ip)
namesrvAddr=node1:9876;node2:9876
# Promethues开启
```

```
enableMultiDispatch=true
enablemq=true
metricsPromExporterHost=0.0.0.0
metricsExporterType=PROM
```

Broker-a-Slave (node3)

```
brokerClusterName=AdmqCluster
brokerName=broker-a
# (从节点的id, 由于主从的名称相同, 所以在同一个组)
brokerId=1
brokerRole=SLAVE
flushDiskType=SYNC_FLUSH
# (如果是集群部署, 需要填写集群所有节点的端口ip)
namesrvAddr=node1:9876;node2:9876
# Prometheus开启
enableMultiDispatch=true
enablemq=true
metricsPromExporterHost=0.0.0.0
metricsExporterType=PROM
```

Broker-b-Master (node2)

```
brokerClusterName=AdmqCluster
# (集群主节点的brokerName需要区分)
brokerName=broker-b
brokerId=0
brokerRole=SYNC_MASTER
flushDiskType=SYNC_FLUSH
# (如果是集群部署, 需要填写集群所有节点的端口ip)
namesrvAddr=node1:9876;node2:9876
# Prometheus开启
enableMultiDispatch=true
enablemq=true
```

```
metricsPromExporterHost=0.0.0.0
metricsExporterType=PROM
```

Broker-b-Slave (node4)

```
brokerClusterName=AdmqCluster
brokerName=broker-b
# (从节点的id, 由于主从的名称相同, 所以在同一个组)
brokerId=1
brokerRole=SLAVE
flushDiskType=SYNC_FLUSH
# (如果是集群部署, 需要填写集群所有节点的端口ip)
namesrvAddr=node1:9876;node2:9876
# Promethues开启
enableMultiDispatch=true
enablemq=true
metricsPromExporterHost=0.0.0.0
metricsExporterType=PROM
```

命令启动

```
./bin/rocketmq start broker
```

8 管控台部署

8.1 安装管控台

```
# 解压管控台安装包
tar zxvf admq-manager-V2.0.6.tar.gz -C /apusic/
cd /apusic/admq-manager-V2.0.6

# 配置 NameServer 地址
vi config/application.properties
```

```
# application.properties
server.port=12306
```

```
# 启动管控台
bin/admq-service start

# 验证状态
bin/admq-service status
```

8.2 访问管控台

地址: `http://<管控台IP>:12305` (`https://<管控台IP>:12306`)
默认账号: `admq / 11111111`

8.3 初始化集群

可以直接通过管控台快速部署集群；对于已经部署的集群，可以通过管控接入的方式接入管理

9 常见问题及排查

9.1 NameServer 启动失败

可能原因	解决方案
端口冲突	检查 9876 端口是否被占用
JDK 未安装	安装 JDK 1.8+
权限不足	确保 admq-rocketmq 目录权限正确

9.2 Broker 启动失败

可能原因	解决方案
NameServer 不可达	检查 namesrvAddr 配置和网络连通性
端口冲突	检查 10911 端口是否被占用
磁盘空间不足	检查磁盘空间，确保满足最低要求
配置错误	检查 broker.conf 配置是否正确

9.3 管控台无法访问

可能原因	解决方案
NameServer 不可达	检查管控台到 NameServer 的网络连通性
端口冲突	修改 config/application.properties 中的 server.port
防火墙阻挡	开放 12305/12306 端口

9.4 消息发送失败

可能原因	解决方案
Topic 未创建	使用 mqadmin 创建 Topic
Broker 不可达	检查 Broker 状态和网络连通性
权限不足	检查 ACL 配置

磁盘已满

清理磁盘空间或调整消息保留策略

全国统一服务热线
4008-555-800



金蝶天燕云计算股份有限公司(简称“金蝶天燕云”)成立于2000年,前身为“金蝶中间件公司”,是金蝶集团旗下新一代软件基础云平台服务商,云计算国家标准制定企业,国家信创产业核心软件企业。金蝶天燕是国家863重点研发计划与核高基重大专项承接企业,也是“两网一站四库十二金”国家重点工程的基础平台提供商,产品广泛应用于政府、军工、金融、能源等关键行业,累计服务客户总数超过10万家。



云计算国家标准制定企业
金蝶集团旗下基础软件企业
信息技术应用创新核心企业
官网: www.apusic.com

