



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_rabbitmq

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 整体架构
 - 2.2 组件说明
 - 2.3 部署模式
 - 2.4 架构特点
- 3 环境准备
 - 3.1 系统要求
 - 3.2 网络要求
- 4 单机部署
 - 4.1 创建部署目录
 - 4.2 下载并解压软件包
 - 4.3 目录结构说明
 - 4.4 配置修改
 - 4.4.1 修改主机名称
 - 4.4.2 修改监听端口等信息（可选）
 - 4.5 授权许可
 - 4.6 启动服务
 - 4.7 服务验证
 - 4.7.1 检查进程状态
 - 4.7.2 检查端口监听
 - 4.7.3 查看运行日志
 - 4.7.4 创建用户
 - 4.7.5 创建虚拟主机
- 5 集群部署
 - 5.1 部署规划
 - 5.2 所有节点执行：基础安装
 - 5.2.1 创建目录并解压
 - 5.2.2 放置授权文件

- 5.2.3 修改主机名称
- 5.2.4 修改服务器 hosts 文件
- 5.3 配置集群（可选）
 - 5.3.1 Node 1 配置 (172.20.140.171)
 - 5.3.2 Node 2 配置 (172.20.140.172)
 - 5.3.3 Node 3 配置 (172.20.140.173)
- 5.4 启动各节点服务
- 5.5 创建集群
- 5.6 镜像模式（可选）
- 5.7 集群验证与测试
 - 5.7.1 验证各个节点的端口状态
 - 5.7.2 检查集群状态
 - 5.7.3 功能测试
- 5.8 集群故障模拟与维护
 - 5.8.1 停止服务
 - 5.8.2 常见错误排查
- 6 常见问题及排查
 - 6.1 服务启动失败
 - 6.2 集群加入失败
 - 6.3 管控台无法访问
 - 6.4 数据源 403 错误

1 前言

本文档为金蝶Apusic分布式消息队列for RabbitMQ（Apusic Distributed Message Queue，简称：ADMQ for RabbitMQ）的安装手册，提供产品各种部署模式（含单机、集群）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 RabbitMQ产品相关的信息，请参阅以下ADMQ for RabbitMQ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 RabbitMQ 快速使用手册	简单介绍了如何快速上手使用ADMQ for RabbitMQ。
2	金蝶Apusic分布式消息队列for RabbitMQ 安装手册	详细介绍如何在各操作系统上安装ADMQ for RabbitMQ，以及ADMQ for RabbitMQ服务启停等操作。
3	金蝶Apusic分布式消息队列for RabbitMQ 消息引擎用户手册	详细介绍 ADMQ for RabbitMQ 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 RabbitMQ 管控台用户手册	详细介绍ADMQ for RabbitMQ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 RabbitMQ 开发手册	详细介绍基于各开发语言进行ADMQ for RabbitMQ客户端应用开发的说明。
6	金蝶Apusic分布式消息队列for RabbitMQ 迁移手册	详细介绍从RabbitMQ迁移到ADMQ for RabbitMQ的说明。
7	金蝶Apusic分布式消息队列for RabbitMQ 运维手册	详细介绍ADMQ for RabbitMQ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 RabbitMQ 性能优化手册	详细介绍ADMQ for RabbitMQ性能调优的说明。

1.3 技术支持

ADMQ for RabbitMQ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>

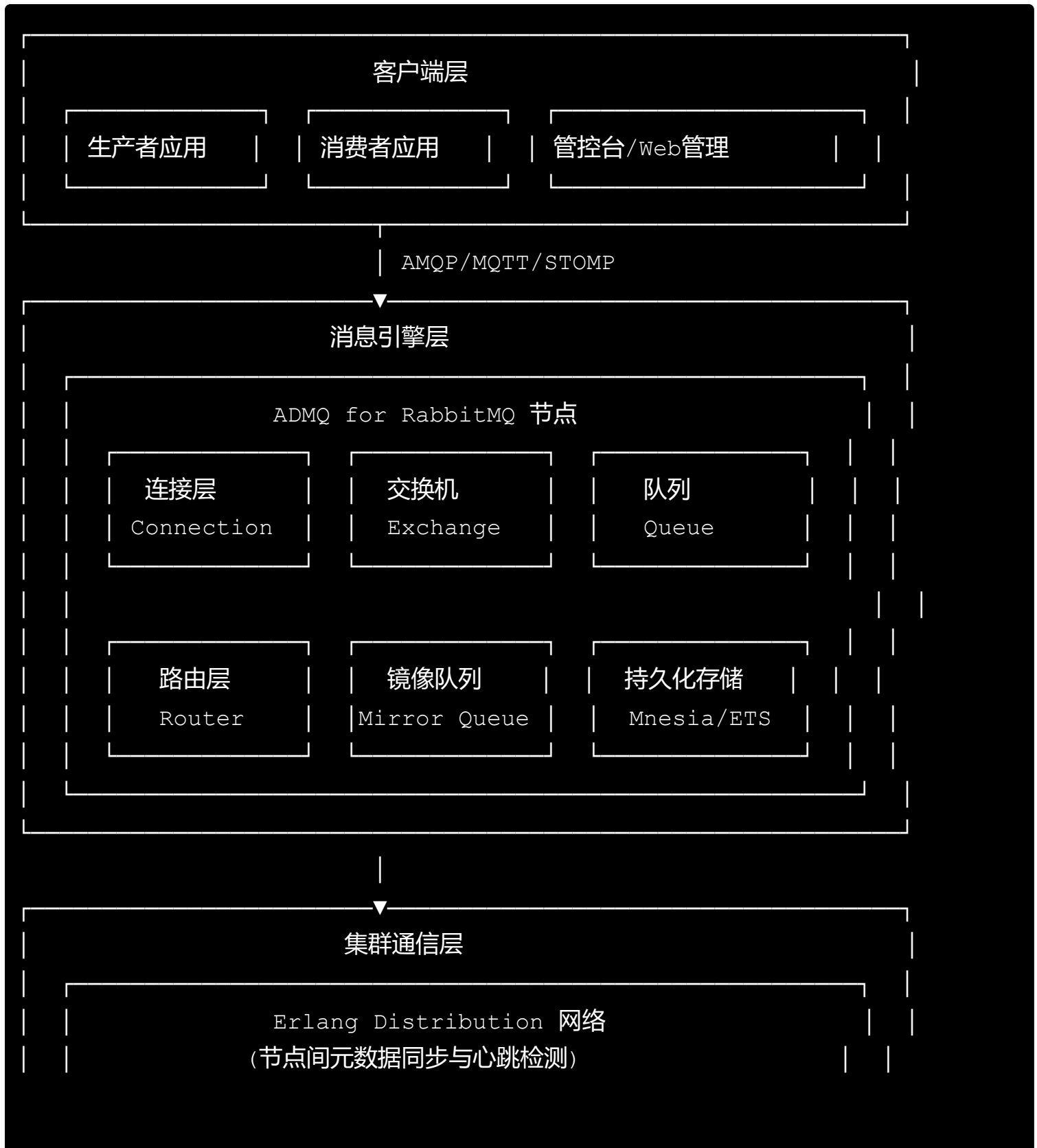
您在取得技术支持时，请提供如下信息：

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 整体架构

ADMQ for RabbitMQ 采用模块化设计，整体架构由以下核心组件组成：





2.2 组件说明

组件	说明	功能
连接层	Connection	处理客户端连接、协议解析、安全认证
交换机	Exchange	接收消息并根据路由规则分发到队列
队列	Queue	存储消息，支持普通队列和 Quorum Queues
镜像队列	Mirror Queue	集群模式下同步队列数据到多个节点
持久化存储	Mnesia/ETS	存储元数据、配置信息和持久化消息
集群通信	Erlang Distribution	节点间通信，同步元数据和集群状态

2.3 部署模式

ADMQ for RabbitMQ 支持以下部署模式：

部署模式	节点数	适用场景	高可用性
单机部署	1 节点	开发测试、低吞吐量场景	无
普通集群	2+ 节点	生产环境，需要横向扩展	元数据高可用
镜像队列集群	3+ 节点	生产环境，消息数据高可用	元数据+数据高可用

2.4 架构特点

- 1. 分布式设计：支持多节点集群部署，元数据自动同步
- 2. 高可用性：支持镜像队列，确保消息数据在节点故障时不丢失
- 3. 弹性扩展：可动态添加节点扩展集群处理能力
- 4. 多协议支持：原生支持 AMQP 0-9-1 协议
- 5. 管控分离：消息引擎与管控台可独立部署

3 环境准备

3.1 系统要求

组件	最低要求	推荐配置
CPU	4 核	8 核及以上
内存	8 GB	16 GB 及以上
磁盘	100 GB SSD	500 GB 及以上 SSD
网络	1 Gbps	10 Gbps
操作系统	Linux (CentOS 7+, Ubuntu 18.04+, 麒麟, 统信)	-

3.2 网络要求

端口	用途	说明
5672	AMQP 协议	生产者和消费者连接
5671	AMQP over SSL	SSL 加密连接
15672	Management HTTP	HTTP API 和 Web 管理界面
15671	Management HTTPS	HTTPS API 和 Web 管理界面
15692	Prometheus Metrics	监控采集指标
25672	Erlang Distribution	集群节点间通信
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

本文档指导用户在 Linux 环境下完成 ADMQ RabbitMQ 的单机模式部署，包含环境准备、安装配置、授权申请、服务启动及状态验证。

4.1 创建部署目录

首先创建应用根目录并进入：

```
mkdir -p /apusic
cd /apusic
```

4.2 下载并解压软件包

从内部仓库下载指定版本的安装包并解压。

```
# 下载安装包
# 实施人员提供

# 解压安装包
tar zxvf ADMQ-V2.0.6.31214-RabbitMQ-20260227.tar.gz
```

4.3 目录结构说明

解压完成后，当前目录下将生成 `adm-q-rabbitmq`。标准目录结构如下：

```
adm-q-rabbitmq/
├── bin/           # 可执行脚本目录
├── config/        # 配置文件目录
├── data/          # 数据存储目录
├── logs/          # 日志文件目录
├── lib/           # 依赖库目录
└── license.xml    # 授权文件（需自行放置）
```

4.4 配置修改

4.4.1 修改主机名称

主机名称只能包含：

- 字母 (a-z A-Z)
- 数字 (0-9)
- 连字符 (-)

```
# 查看主机名称
hostname

# 修改主机名
hostname node84
```

4.4.2 修改监听端口等信息（可选）

编辑 `config/rabbitmq.conf`，可以修改监听端口、用户名密码等信息。

```
# 示例：修改监听端口和默认用户
listeners.tcp.default = 5672
default_user = admq
default_pass = apusic_123
```

4.5 授权许可

ADMQ 产品需要 License 文件才能正常运行。

1. **申请授权**：联系管理员或通过授权平台申请对应版本的 `license.xml` 文件。
2. **放置文件**：将获取到的 `license.xml` 文件上传至部署目录根路径。

```
# 确认文件位置
ls -l /apusic/admq-rabbitmq/license.xml
```

4.6 启动服务

启动消息收发服务：

```
bin/admq-daemon start rabbitmq server
```

```
[root@localhost admq-rabbitmq]# bin/admq-daemon start rabbitmq server
Starting rabbitmq-server, logging to /apusic/admq-rabbitmq/logs
  Base directory: /apusic/admq-rabbitmq/rabbitmq/bin
  Root directory: /apusic/admq-rabbitmq/rabbitmq
  Libs directory: /apusic/admq-rabbitmq/rabbitmq/libs
  Config path: /apusic/admq-rabbitmq/config/rabbitmq.conf
  Starting Apusic RabbitMQ...
[root@localhost admq-rabbitmq]#
```

4.7 服务验证

4.7.1 检查进程状态

使用 `ps` 命令查看进程，正常状态下应能看到相关进程：

```
ps -ef | grep admq-rabbitmq
```

预期输出示例：

```
[root@localhost admq-rabbitmq]# ps -ef | grep admq-rabbitmq
root      19084      1  0 14:01 ?        00:00:00 /apusic/admq-rabbitmq/rabbitmq/exec/xdkQuM/erts-14.2.5.11/bin/epmd -daemon
root      19997      1  5 14:09 pts/4    00:00:01 /apusic/admq-rabbitmq/jdk/aarch64/bin/java -Dconfig.path=/apusic/admq-rabbitmq/config/rabbitmq.conf -cp ./apusic/admq-rabbitmq/rabbitmq/libs/acls-client-1.0.9.jar:/apusic/admq-rabbitmq/rabbitmq/libs/admq-monitor-1.0.0-apusic.jar:/apusic/admq-rabbitmq/rabbitmq/libs/apiguardian-api-1.1.2.jar:/apusic/admq-rabbitmq/rabbitmq/libs/apusic-license-core-1.7.jar:/apusic/admq-rabbitmq/rabbitmq/libs/apusic-license-lib-1.7.jar:/apusic/admq-rabbitmq/rabbitmq/libs/commons-codec-1.11.jar:/apusic/admq-rabbitmq/rabbitmq/libs/commons-lang3-3.18.0.jar:/apusic/admq-rabbitmq/rabbitmq/libs/commons-logging-1.2.jar:/apusic/admq-rabbitmq/rabbitmq/libs/httpclient-4.5.13.jar:/apusic/admq-rabbitmq/rabbitmq/libs/httpcore-4.4.13.jar:/apusic/admq-rabbitmq/rabbitmq/libs/jcommander-1.82.jar:/apusic/admq-rabbitmq/rabbitmq/libs/junit-jupiter-5.10.2.jar:/apusic/admq-rabbitmq/rabbitmq/libs/junit-jupiter-api-5.10.2.jar:/apusic/admq-rabbitmq/rabbitmq/libs/junit-jupiter-engine-5.10.2.jar:/apusic/admq-rabbitmq/rabbitmq/libs/junit-jupiter-params-5.10.2.jar:/apusic/admq-rabbitmq/rabbitmq/libs/junit-platform-commons-1.10.2.jar:/apusic/admq-rabbitmq/rabbitmq/libs/junit-platform-engine-1.10.2.jar:/apusic/admq-rabbitmq/rabbitmq/libs/log4j-api-2.25.3.jar:/apusic/admq-rabbitmq/rabbitmq/libs/log4j-core-2.25.3.jar:/apusic/admq-rabbitmq/rabbitmq/libs/log4j-slf4j2-impl-2.25.3.jar:/apusic/admq-rabbitmq/rabbitmq/libs/lombok-1.18.28.jar:/apusic/admq-rabbitmq/rabbitmq/libs/opentest4j-1.3.0.jar:/apusic/admq-rabbitmq/rabbitmq/libs/slf4j-api-2.0.12.jar:/apusic/admq-rabbitmq/rabbitmq/libs/starter-3.12.14.jar com.apusic.admq.rabbitmq.Starter
root      20034      1  0 14:09 pts/4    00:00:00 /bin/sh /apusic/admq-rabbitmq/rabbitmq/core/sbin/rabbitmq-server
root      20042      0  0 14:09 pts/4    00:00:06 /apusic/admq-rabbitmq/rabbitmq/exec/xdkQuM/erts-14.2.5.11/bin/beam.smp -W w -MBas ageffcbf -MHas ageffcbf -MBlmbcs 512 -MHlmbcs 512 -MMmcs 30 -pc unicode -P 1048576 -t 5000000 -stbt db -zdbbl 128000 -sbwt none -sbwtdcpu none -sbwtio none -B i -- -root /apusic/admq-rabbitmq/rabbitmq/exec/xdkQuM -bindir /apusic/admq-rabbitmq/rabbitmq/exec/xdkQuM/erts-14.2.5.11/bin -p rogname erl -- -home /apusic/admq-rabbitmq -- -pa -noshell -noinput -s rabbit boot -boot start_sasl -syslog logger [] -syslog syslog_error_logger false -kernel prevent_overlapping_partitions false -enable-features maybe_expr
root      20355      0  0 14:09 ?        00:00:00 /apusic/admq-rabbitmq/rabbitmq/exec/xdkQuM/erts-14.2.5.11/bin/inet_gethost 4
root      20356      0  0 14:09 ?        00:00:00 /apusic/admq-rabbitmq/rabbitmq/exec/xdkQuM/erts-14.2.5.11/bin/inet_gethost 4
root      20378      0  0 14:09 pts/4    00:00:00 grep --color=auto admq-rabbitmq
[root@localhost admq-rabbitmq]#
```

4.7.2 检查端口监听

使用 `netstat` 确认 5672 端口是否处于 `LISTEN` 状态：

```
netstat -nltup | grep 5672
```

预期输出示例：

```
[root@localhost admq-rabbitmq]# netstat -nltp | grep 5672
tcp        0      0 0.0.0.0:15672        0.0.0.0:*            LISTEN      20042/beam.smp
tcp        0      0 0.0.0.0:25672        0.0.0.0:*            LISTEN      20042/beam.smp
tcp6       0      0 :::5672              :::*                  LISTEN      20042/beam.smp
[root@localhost admq-rabbitmq]#
```

如果未看到输出，请检查防火墙设置或查看日志排查启动失败原因。

4.7.3 查看运行日志

如果服务启动异常或需监控运行状态，请查看 `logs` 目录下的日志文件：

```
# 查看实时日志
tail -f logs/server.log

# 或者查看目录结构
ls -lh logs/
```

重点关注日志中是否有 `ERROR` 级别的信息。

```
[root@localhost admq-rabbitmq]# tail -f logs/server.log
2026-03-03 14:09:24.850516+08:00 [info] <0.790.0> Management plugin: HTTP (non-TLS) listener started on port 15672
2026-03-03 14:09:24.850715+08:00 [info] <0.818.0> Statistics database started.
2026-03-03 14:09:24.850807+08:00 [info] <0.817.0> Starting worker pool 'management_worker_pool' with 3 processes in it
2026-03-03 14:09:24.851385+08:00 [info] <0.731.0> Ready to start client connection listeners
2026-03-03 14:09:24.913612+08:00 [info] <0.842.0> started TCP listener on [::]:5672
2026-03-03 14:09:25.016642+08:00 [info] <0.731.0> Server startup complete; 3 plugins started.
2026-03-03 14:09:25.016642+08:00 [info] <0.731.0> * rabbitmq_management
2026-03-03 14:09:25.016642+08:00 [info] <0.731.0> * rabbitmq_management_agent
2026-03-03 14:09:25.016642+08:00 [info] <0.731.0> * rabbitmq_web_dispatch
2026-03-03 14:09:25.207324+08:00 [info] <0.9.0> Time to start RabbitMQ: 6005200 us
```

4.7.4 创建用户

默认已经开启了认证，客户端收发消息需要添加用户名密码。

创建用户：test

设置密码：test123

```
bin/admq rabbitmq admin ctl add_user test test123
```

```
[root@node84 admq-rabbitmq]# bin/admq rabbitmq admin ctl add_user test test123
⊗ Executing: rabbitmqctl add_user test test123
Adding user "test" ...
Done. Don't forget to grant the user permissions to some virtual hosts! See 'rabbitmqctl help set_permissions' to learn more.
[root@node84 admq-rabbitmq]#
```

4.7.5 创建虚拟主机

创建虚拟主机：test

```
bin/admq rabbitmq admin ctl add_vhost test
```

```
[root@node84 admq-rabbitmq]# bin/admq rabbitmq admin ctl add_vhost test
⊗ Executing: rabbitmqctl add_vhost test
Adding vhost "test" ...
```

设置权限给用户：test

```
bin/admq rabbitmq admin ctl set_permissions --vhost test test ".*" ".*" ".*"
bin/admq rabbitmq admin ctl set_user_tags test management
```

```
[root@node84 admq-rabbitmq]# bin/admq rabbitmq admin ctl set_permissions --vhost test test ".*" ".*" ".*"
⊗ Executing: rabbitmqctl set_permissions --vhost test test .* .* .*
Setting permissions for user "test" in vhost "test" ...
[root@node84 admq-rabbitmq]#
```

```
[root@node84 admq-rabbitmq]# bin/admq rabbitmq admin ctl set_user_tags test management
⊗ Executing: rabbitmqctl set_user_tags test management
Setting tags for user "test" to [management] ...
[root@node84 admq-rabbitmq]#
```

5 集群部署

5.1 部署规划

假设部署 3 节点 高可用集群，规划如下：

前置要求：

1. 确保三台机器时间同步（NTP）。
2. 关闭防火墙或开放端口： 5672 ， 15672 。

5.2 所有节点执行：基础安装

在 Node 1, Node 2, Node 3 上分别执行：

5.2.1 创建目录并解压

```
mkdir -p /apusic
cd /apusic
tar zxvf ADMQ-V2.0.6.31214-RabbitMQ-20260306.tar.gz
cd admq-rabbitmq
```

5.2.2 放置授权文件

将 `license.xml` 上传至 `/apusic/admq-rabbitmq/` 目录。

5.2.3 修改主机名称

主机名称只能包含：

- 字母 (a-z A-Z)
- 数字 (0-9)
- 连字符 (-)

```
# 在 Node 1 执行
hostname node171

# 在 Node 2 执行
```

```
hostname node172

# 在 Node 3 执行
hostname node173
```

5.2.4 修改服务器 hosts 文件

需要在 `/etc/hosts` 添加集群所有节点的信息。

所有节点都添加如下信息：

```
172.20.140.171 node171
172.20.140.172 node172
172.20.140.173 node173
```

5.3 配置集群（可选）

编辑 `config/rabbitmq.conf`，可以配置监听端口、默认用户名密码等信息。

5.3.1 Node 1 配置 (172.20.140.171)

```
listeners.tcp.default = 5672
# 默认用户
default_user = admq
default_pass = apusic_123
```

5.3.2 Node 2 配置 (172.20.140.172)

```
listeners.tcp.default = 5672
# 默认用户
default_user = admq
default_pass = apusic_123
```

5.3.3 Node 3 配置 (172.20.140.173)

```
listeners.tcp.default = 5672
# 默认用户
default_user = admq
default_pass = apusic_123
```

5.4 启动各节点服务

依次登录 Node 1, 2, 3 执行：

```
cd /apusic/admq-rabbitmq
bin/admq-daemon start rabbitmq server
```

启动后日志如下：

```
[root@localhost admq-rabbitmq]# bin/admq-daemon start rabbitmq server
Starting rabbitmq-server, logging to /apusic/admq-rabbitmq/logs
📁 Base directory: /apusic/admq-rabbitmq/rabbitmq/bin
📁 Root directory: /apusic/admq-rabbitmq/rabbitmq
📁 Libs directory: /apusic/admq-rabbitmq/rabbitmq/libs
📄 Config path: /apusic/admq-rabbitmq/config/rabbitmq.conf
🔧 Starting Apusic RabbitMQ...
[root@localhost admq-rabbitmq]#
```

之后需要检测节点状态，确认是否启动成功。可直接查看端口是否处于监听状态。

```
ss -anpl | grep 5672
```

预期输出：

```
[root@openstack-140-171 admq-rabbitmq]# ss -anp |grep 5672
tcp    LISTEN    0        128      *:25672    *:.*      users:((("beam.smp",pid=28206,fd=18)))
tcp    LISTEN    0        128      *:15672    *:.*      users:((("beam.smp",pid=28206,fd=34)))
tcp    LISTEN    0        128      [::]:5672  [::]:.*   users:((("beam.smp",pid=28206,fd=35)))
```

5.5 创建集群

在第一个节点执行集群创建命令，把另外两个节点加入到该节点组成集群：

```
bin/admq rabbitmq cluster add -s 172.20.140.172:58181 -u monitor -p
apusic_123
bin/admq rabbitmq cluster add -s 172.20.140.173:58181 -u monitor -p
apusic_123
```

```
[root@openstack-140-171 admq-rabbitmq]# bin/admq rabbitmq cluster add -s 172.20.140.172:58181 -u monitor -p apusic_123
🔧 Adding node to cluster...
📌 Target node: 172.20.140.172:58181
👤 Username: monitor
🔄 Step 1: Synchronizing authentication (Cookie)...
✅ Cookie synchronized: Cookie received and saved successfully

🌐 Step 2: Discovering and recording hostnames...
📋 Hosts entry already exists:
✅ Hostnames discovered and recorded

🔄 Step 3: Synchronizing reverse hostnames...
✅ Reverse hostnames synchronized: Hostname already exists in hosts file: 10.0.0.131 node171

⚙️ Step 4: Initializing cluster...
✅ Cluster initialization completed: Successfully joined cluster with master node: rabbit@node171
✅ Node added to cluster successfully!
[root@openstack-140-171 admq-rabbitmq]#
```

```
[root@openstack-140-171 admq-rabbitmq]# bin/admq rabbitmq cluster add -s 172.20.140.173:58181 -u monitor -p apusic_123
🔧 Adding node to cluster...
📌 Target node: 172.20.140.173:58181
👤 Username: monitor
🔄 Step 1: Synchronizing authentication (Cookie)...
✅ Cookie synchronized: Cookie received and saved successfully

🌐 Step 2: Discovering and recording hostnames...
📋 Hosts entry already exists: 10.0.0.6 node173
✅ Hostnames discovered and recorded

🔄 Step 3: Synchronizing reverse hostnames...
{"code":0,"message":"Hostname already exists in hosts file:"}
✅ Reverse hostnames synchronized: Hostname already exists in hosts file:

⚙️ Step 4: Initializing cluster...
✅ Cluster initialization completed: Successfully joined cluster with master node: rabbit@node171
✅ Node added to cluster successfully!
[root@openstack-140-171 admq-rabbitmq]#
```

5.6 镜像模式（可选）

集群模式中，各个节点只能同步元数据，不能同步交换机、队列下的数据。在没使用 `Quorum Queues` 模式的情况下可以使用镜像队列同步各个节点的数据。

```
bin/admq rabbitmq admin ctl set_policy ha-all "^" '{"ha-
mode":"all","ha-sync-mode":"automatic"}'
```

5.7 集群验证与测试

5.7.1 验证各个节点的端口状态

在每台服务器检查端口监听状态：

```
# 检查 5672 端口是否被监听
ss -anpl | grep 5672
```

预期：可以看到端口被监听。

```
[root@node84 admq-rabbitmq]# ss -anpl | grep 5672
tcp    LISTEN 0      1024          0.0.0.0:15672      0.0.0.0:*        users:(("beam.smp",pid=175644,fd=37))
tcp    LISTEN 0      128        172.20.140.84:5672 0.0.0.0:*        users:(("beam.smp",pid=175644,fd=38))
tcp    LISTEN 0      128          0.0.0.0:25672     0.0.0.0:*        users:(("beam.smp",pid=175644,fd=18))
```

5.7.2 检查集群状态

在 Node1 执行：

```
bin/admq rabbitmq cluster status
```

可以看到集群中三个节点都已经添加完成。

```
[root@openstack-140-171 admq-rabbitmq]# bin/admq rabbitmq cluster status
🔍 Checking RabbitMQ cluster status...
🔧 Executing: rabbitmqctl cluster_status
Cluster status of node rabbit@node171 ...

Basics

Cluster name: rabbit@node171
Total CPU cores available cluster-wide: 36


Disk Nodes

rabbit@node171
rabbit@node172
rabbit@node173


Running Nodes

rabbit@node171
rabbit@node172
rabbit@node173


Versions

rabbit@node171: RabbitMQ 3.12.14 on Erlang 26.2.5.15
rabbit@node172: RabbitMQ 3.12.14 on Erlang 26.2.5.15
rabbit@node173: RabbitMQ 3.12.14 on Erlang 26.2.5.15


CPU Cores

Node: rabbit@node171, available CPU cores: 12
Node: rabbit@node172, available CPU cores: 12
Node: rabbit@node173, available CPU cores: 12
```

5.7.3 功能测试

1. 创建用户:

创建用户: test

设置密码: test123

```
bin/admq rabbitmq admin ctl add_user test test123
```

预期输出示例:

```
[root@node84 admq-rabbitmq]# bin/admq rabbitmq admin ctl add_user test test123
⊗ Executing: rabbitmqctl add_user test test123
Adding user "test" ...
Done. Don't forget to grant the user permissions to some virtual hosts! See 'rabbitmqctl help set_permissions' to learn more.
[root@node84 admq-rabbitmq]#
```

2. 创建虚拟主机:

创建虚拟主机: test

```
bin/admq rabbitmq admin ctl add_vhost test
```

预期输出示例:

```
[root@node84 admq-rabbitmq]# bin/admq rabbitmq admin ctl add_vhost test
⊗ Executing: rabbitmqctl add_vhost test
Adding vhost "test" ...
```

设置权限给用户: test

```
bin/admq rabbitmq admin ctl set_permissions --vhost test test ".*" ".*" ".*"
bin/admq rabbitmq admin ctl set_user_tags test management
```

```
[root@node84 admq-rabbitmq]# bin/admq rabbitmq admin ctl set_permissions --vhost test test ".*" ".*" ".*"
⊗ Executing: rabbitmqctl set_permissions --vhost test test .* .* .*
Setting permissions for user "test" in vhost "test" ...
[root@node84 admq-rabbitmq]#
```

```
[root@node84 admq-rabbitmq]# bin/admq rabbitmq admin ctl set_user_tags test management
⊗ Executing: rabbitmqctl set_user_tags test management
Setting tags for user "test" to [management] ...
[root@node84 admq-rabbitmq]#
```

5.8 集群故障模拟与维护

5.8.1 停止服务

```
bin/admq-daemon stop rabbitmq server
```

5.8.2 常见错误排查

集群节点添加失败，节点连接失败

```
[root@openstack-140-173 admq-rabbitmq]# tail -f logs/server.log
2026-03-06 16:57:43.728869+08:00 [info] <0.614.0> Stopping application 'mnesia'
2026-03-06 16:57:43.732265+08:00 [notice] <0.44.0> Application mnesia exited with reason: stopped
2026-03-06 16:57:43.732379+08:00 [info] <0.614.0> Successfully stopped RabbitMQ and its dependencies
2026-03-06 16:57:44.214957+08:00 [notice] <0.44.0> Application mnesia exited with reason: stopped
2026-03-06 16:57:44.215292+08:00 [notice] <0.634.0> Feature flags: checking nodes `rabbit@node173` and `rabbit@node171` compatibility...
2026-03-06 16:57:44.285836+08:00 [warning] <0.662.0> Feature flags: no connection to node `rabbit@node172`; retrying in 5000 milliseconds
2026-03-06 16:57:44.409266+08:00 [warning] <0.664.0> 'global' at rabbit@node173 failed to connect to rabbit@node172
2026-03-06 16:57:49.292214+08:00 [warning] <0.662.0> Feature flags: no connection to node `rabbit@node172`; retrying in 5000 milliseconds
2026-03-06 16:57:54.297877+08:00 [warning] <0.662.0> Feature flags: no connection to node `rabbit@node172`; retrying in 5000 milliseconds
2026-03-06 16:57:59.317362+08:00 [warning] <0.662.0> Feature flags: no connection to node `rabbit@node172`; retrying in 5000 milliseconds
2026-03-06 16:58:04.321727+08:00 [warning] <0.662.0> Feature flags: no connection to node `rabbit@node172`; retrying in 5000 milliseconds
2026-03-06 16:58:09.342998+08:00 [warning] <0.662.0> Feature flags: no connection to node `rabbit@node172`; retrying in 5000 milliseconds
2026-03-06 16:58:14.347594+08:00 [warning] <0.662.0> Feature flags: no connection to node `rabbit@node172`; retrying in 5000 milliseconds
2026-03-06 16:58:19.354276+08:00 [warning] <0.662.0> Feature flags: no connection to node `rabbit@node172`; retrying in 5000 milliseconds
2026-03-06 16:58:24.359365+08:00 [warning] <0.662.0> Feature flags: no connection to node `rabbit@node172`; retrying in 5000 milliseconds
2026-03-06 16:58:29.366830+08:00 [warning] <0.662.0> Feature flags: no connection to node `rabbit@node172`; retrying in 5000 milliseconds
2026-03-06 16:58:34.371249+08:00 [warning] <0.662.0> Feature flags: no connection to node `rabbit@node172`; retrying in 5000 milliseconds
```

解决方法：配置 `/etc/hosts`，添加 node172 的信息，然后重启该节点。

集群节点添加失败，节点停止失败

```
[root@openstack-140-171 admq-rabbitmq]# bin/admq rabbitmq cluster add -s 172.20.140.173:58181 -u monitor -p apusic_123
✎ Adding node to cluster...
✎ Target node: 172.20.140.173:58181
✎ Username: monitor
✎ Step 1: Synchronizing authentication (Cookie)...
☑ Cookie synchronized: Cookie received and saved successfully

✎ Step 2: Discovering and recording hostnames...
☐ Hosts entry already exists: 10.0.0.6 node173
☑ Hostnames discovered and recorded

☐ Step 3: Synchronizing reverse hostnames...
{"code":0,"message":"Hostname received and added to hosts file: \n10.0.0.6 node173\n\n10.0.0.131 node171\n"}
⚠ Failed to sync reverse hostnames: Hostname received and added to hosts file:
10.0.0.6 node173

10.0.0.131 node171

✎ Step 4: Initializing cluster...
✗ Failed to initialize cluster: Error stopping application
```

解决方法：重启需要添加的节点。

6 常见问题及排查

6.1 服务启动失败

可能原因	解决方案
端口冲突	检查 5672、15672、25672 端口是否被占用
权限不足	确保 admq-rabbitmq 目录权限正确
内存不足	检查系统内存，确保满足最低要求
License 未放置或过期	确认 license.xml 已放置在部署目录根路径且未过期
OpenSSL 版本不兼容	Ubuntu 24.04 等系统默认使用 OpenSSL 3.x，而 Erlang crypto 模块依赖 libcrypto.so.1.1，需安装 OpenSSL 1.1 兼容库（libssl1.1），安装完成后重启服务

OpenSSL 1.1 兼容库安装示例（Ubuntu/Debian）：

```
# Ubuntu 24.04 需要先安装 libssl1.1
wget
http://security.ubuntu.com/ubuntu/pool/main/o/openssl/libssl1.1_1.1.1f-1ubuntu2.24_amd64.deb
dpkg -i libssl1.1_1.1.1f-1ubuntu2.24_amd64.deb
```

说明：如果在 logs/rabbitmq-server-*.out 中看到类似 crypto:strong_rand_bytes 或 libcrypto.so.1.1: cannot open shared object file 的错误，即为 OpenSSL 兼容性问题。

6.2 集群加入失败

可能原因	解决方案
主机名不符合规范	确保主机名只包含字母、数字和连字符
网络不通	检查节点间网络连通性和防火墙设置
主机名解析失败	确保 /etc/hosts 或 DNS 配置正确
节点未正常启动	检查各节点服务是否已正常启动

6.3 管控台无法访问

可能原因	解决方案
端口冲突	修改 config/application.properties 中的 server.port
防火墙阻挡	开放 12305/12306 端口
服务未启动	检查管控台服务状态

6.4 数据源 403 错误

可能原因	解决方案
RabbitMQ 用户权限不足	使用 administrator 标签用户，配置 .* 权限
消息发送和接收异常	检查路由 key 和绑定关系是否正确

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

