



迁移手册

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 迁移概述
 - 2.1 迁移原则
 - 2.2 迁移范围
 - 2.3 迁移前检查
 - 2.3.1 版本兼容性检查
 - 2.3.2 客户端兼容性检查
- 3 迁移方案
 - 3.1 方案一：双写迁移（推荐）
 - 3.2 方案二：停服迁移
 - 3.3 方案三：灰度迁移
- 4 配置迁移
 - 4.1 导出原 RabbitMQ 配置
 - 4.2 在 ADMQ for RabbitMQ 中创建配置
 - 4.2.1 使用管控台创建
 - 4.2.2 使用命令行创建
- 5 数据迁移
 - 5.1 使用 Shovel 迁移数据
 - 5.1.1 配置 Shovel
 - 5.1.2 创建 Shovel 配置
 - 5.2 使用 Federation 迁移数据
 - 5.3 使用客户端脚本迁移
- 6 客户端迁移
 - 6.1 连接地址变更
 - 6.2 SSL/TLS 配置变更
 - 6.3 常见客户端配置
 - 6.3.1 Java (Spring Boot)
 - 6.3.2 Python (Django)
 - 6.3.3 Go

- 7 迁移验证
 - 7.1 功能验证
 - 7.2 性能验证
 - 7.3 数据一致性验证
- 8 回滚方案
 - 8.1 回滚检查清单
- 9 迁移最佳实践

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）的迁移手册，详细介绍了如何将RabbitMQ迁移到ADMQ for RabbitMQ。

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 迁移概述

2.1 迁移原则

ADMQ for RabbitMQ 完全兼容 RabbitMQ 的 AMQP 0-9-1 协议，迁移过程遵循以下原则：

1. **零代码改造**：客户端无需修改代码，仅需变更连接地址
2. **配置兼容**：支持 RabbitMQ 原生配置参数
3. **数据可迁移**：支持队列数据和配置信息迁移
4. **平滑切换**：支持双写和灰度切换，降低迁移风险

2.2 迁移范围

迁移内容	是否支持	说明
应用代码	是	无需修改，仅需变更连接地址
客户端配置	是	支持 RabbitMQ 原生客户端
Exchange 定义	是	支持所有类型交换机
Queue 定义	是	支持 Classic/Quorum/Stream 队列
Binding 关系	是	支持所有绑定关系
用户权限	是	支持用户和 Vhost 权限迁移
消息数据	是	支持历史消息迁移
策略 (Policy)	是	支持镜像策略等
插件	部分	部分插件可能需要适配

2.3 迁移前检查

2.3.1 版本兼容性检查

原 RabbitMQ 版本	兼容性	说明
3.8.x	完全兼容	推荐迁移版本
3.9.x	完全兼容	推荐迁移版本
3.10.x	完全兼容	推荐迁移版本

3.11.x	完全兼容	推荐迁移版本
3.12.x	完全兼容	推荐迁移版本

2.3.2 客户端兼容性检查

客户端类型	兼容版本	说明
Java (amqp-client)	5.x	完全兼容
Python (pika)	1.x	完全兼容
Go (amqp091-go)	1.x	完全兼容
Node.js (amqplib)	0.10.x	完全兼容
.NET (RabbitMQ.Client)	6.x	完全兼容

3 迁移方案

3.1 方案一：双写迁移（推荐）

适用于不能停机的业务场景，通过双写保证数据一致性。

步骤：

1. 部署 ADMQ for RabbitMQ 集群

- 按照安装手册部署新集群
- 配置网络和防火墙

2. 配置双写

- 在应用中同时连接原 RabbitMQ 和 ADMQ for RabbitMQ
- 发送消息时同时发送到两个集群

3. 同步历史数据

- 使用 Shovel 或 Federation 插件同步队列数据
- 验证数据一致性

4. 切换消费者

- 先切换部分消费者到 ADMQ for RabbitMQ
- 观察业务是否正常

5. 全量切换

- 确认无误后，切换所有消费者
- 停止向原 RabbitMQ 发送消息

6. 下线原集群

- 保留一段时间用于数据查询
- 确认无问题后下线

3.2 方案二：停服迁移

适用于可以短暂停机的业务场景，迁移速度快。

步骤：

1. 停止生产者

- 暂停业务系统向 RabbitMQ 发送消息

2. 导出配置

- 使用 bin/admq rabbitmq admin ctl 导出 Exchange、Queue、Binding、User 等配置

3. 迁移配置到 ADMQ for RabbitMQ

- 在管控台或命令行创建对应的资源

4. 迁移消息数据

- 使用 shovel 导出原队列消息
- 导入到 ADMQ for RabbitMQ

5. 切换连接地址

- 修改应用配置，指向 ADMQ for RabbitMQ

6. 启动生产者

- 恢复业务系统消息发送

3.3 方案三：灰度迁移

适用于大规模系统，逐步迁移不同业务模块。

步骤：

1. 按业务模块划分迁移批次
2. 每批次按双写迁移方案执行
3. 验证通过后迁移下一批次
4. 全部完成后下线原集群

4 配置迁移

4.1 导出原 RabbitMQ 配置

```
# 导出用户和权限
bin/admq rabbitmq admin ctl list_users > users.txt
bin/admq rabbitmq admin ctl list_vhosts > vhosts.txt

# 导出 Exchange
bin/admq rabbitmq admin ctl list_exchanges name type durable
auto_delete internal arguments > exchanges.txt

# 导出 Queue
bin/admq rabbitmq admin ctl list_queues name durable auto_delete
arguments pid slave_pids > queues.txt

# 导出 Binding
bin/admq rabbitmq admin ctl list_bindings source_name source_kind
destination_name destination_kind routing_key arguments >
bindings.txt

# 导出策略
bin/admq rabbitmq admin ctl list_policies > policies.txt
```

4.2 在 ADMQ for RabbitMQ 中创建配置

4.2.1 使用管控台创建

1. 登录管控台
2. 进入「资源管理」页面
3. 按顺序创建：
 - Vhost
 - MQ 用户及权限
 - Exchange
 - Queue

- Binding

4.2.2 使用命令行创建

```
# 创建 Vhost
bin/admq rabbitmq admin ctl add_vhost /test

# 创建用户
bin/admq rabbitmq admin ctl add_user testuser testpass
bin/admq rabbitmq admin ctl set_user_tags testuser management
bin/admq rabbitmq admin ctl set_permissions -p /test testuser ".*"
".*" ".*"

# 创建 Exchange
bin/admq rabbitmq admin ctl eval
'rabbit_exchange:declare(<<"/test">>, <<"test.exchange">>,
<<"direct">>, true, false, false, [], <<"acting_user">>).'

# 创建 Queue
bin/admq rabbitmq admin ctl eval
'rabbit_amqqueue:declare(<<"/test">>, <<"test.queue">>, true, false,
false, [], none, <<"acting_user">>).'
```

5 数据迁移

5.1 使用 Shovel 迁移数据

Shovel 是 RabbitMQ 内置的数据迁移工具。

5.1.1 配置 Shovel

```
# 启用 shovel 插件
rabbitmq-plugins enable rabbitmq_shovel
rabbitmq-plugins enable rabbitmq_shovel_management
```

5.1.2 创建 Shovel 配置

```
{
  "shovels": [
    {
      "name": "migrate-queue1",
      "source": {
        "uri": "amqp://guest:guest@old-rabbitmq:5672/%2F",
        "queue": "queue1"
      },
      "destination": {
        "uri": "amqp://guest:guest@adm-q-rabbitmq:5672/%2F",
        "queue": "queue1"
      },
      "ack-mode": "on-confirm"
    }
  ]
}
```

5.2 使用 Federation 迁移数据

Federation 适用于跨集群的数据同步。

```
# 启用 federation 插件
rabbitmq-plugins enable rabbitmq_federation
rabbitmq-plugins enable rabbitmq_federation_management
```

5.3 使用客户端脚本迁移

对于小规模数据，可以编写客户端脚本迁移：

```
import pika

# 连接原 RabbitMQ
src_conn = pika.BlockingConnection(pika.ConnectionParameters('old-
rabbitmq'))
src_ch = src_conn.channel()

# 连接 ADMQ for RabbitMQ
dst_conn = pika.BlockingConnection(pika.ConnectionParameters('admq-
rabbitmq'))
dst_ch = dst_conn.channel()

# 获取消息并转发
while True:
    method, properties, body = src_ch.basic_get('queue1',
auto_ack=False)
    if method is None:
        break

    dst_ch.basic_publish('', 'queue1', body, properties)
    src_ch.basic_ack(method.delivery_tag)

src_conn.close()
dst_conn.close()
```

6 客户端迁移

6.1 连接地址变更

仅需修改应用配置文件中的连接地址：

```
# 原配置
rabbitmq.host=old-rabbitmq
rabbitmq.port=5672

# 新配置
rabbitmq.host=admq-rabbitmq
rabbitmq.port=5672
```

6.2 SSL/TLS 配置变更

如果启用了 SSL，需要更新证书：

```
# 原配置
rabbitmq.ssl.enabled=true
rabbitmq.ssl.ca-file=/path/to/old-ca.pem

# 新配置
rabbitmq.ssl.enabled=true
rabbitmq.ssl.ca-file=/path/to/admq-ca.pem
rabbitmq.ssl.cert-file=/path/to/client-cert.pem
rabbitmq.ssl.key-file=/path/to/client-key.pem
```

6.3 常见客户端配置

6.3.1 Java (Spring Boot)

```
spring:
  rabbitmq:
```

```
host: admq-rabbitmq
port: 5672
username: guest
password: guest
virtual-host: /
ssl:
  enabled: false
```

6.3.2 Python (Django)

```
BROKER_URL = 'amqp://guest:guest@admq-rabbitmq:5672/'
```

6.3.3 Go

```
conn, err := amqp091.Dial("amqp://guest:guest@admq-rabbitmq:5672/")
```

7 迁移验证

7.1 功能验证

验证项	验证方法
连接测试	使用客户端连接 ADMQ for RabbitMQ，确认能正常建立连接
消息发送	发送测试消息，确认能成功到达 Exchange
消息路由	验证消息能正确路由到 Queue
消息消费	启动消费者，确认能正常接收和处理消息
消息确认	测试手动确认和自动确认功能
持久化	重启服务后验证消息是否保留
集群功能	验证节点故障时是否能自动切换

7.2 性能验证

验证项	验证方法
吞吐量	对比原集群和新集群的消息发送速率
延迟	测量消息从发送到消费的延迟
并发	测试多生产者、多消费者并发场景
稳定性	长时间运行测试，观察是否有异常

7.3 数据一致性验证

```
# 对比原集群和新集群的队列消息数
bin/admq rabbitmq admin ctl list_queues -p /test name messages

# 对比消费者数量
bin/admq rabbitmq admin ctl list_queues -p /test name consumers

# 对比连接数
```

```
bin/admq rabbitmq admin ctl list_connections -p /test peer_host  
state
```

8 回滚方案

如果迁移过程中出现问题，需要回滚到原 RabbitMQ 集群：

1. 停止向 ADMQ for RabbitMQ 发送消息
2. 切换连接地址回原 RabbitMQ
3. 同步 ADMQ for RabbitMQ 中的新消息回原集群（如有需要）
4. 恢复业务系统

8.1 回滚检查清单

- ☐ 原 RabbitMQ 集群仍在运行
- ☐ 原集群数据未被删除
- ☐ 应用配置可快速切换
- ☐ 业务系统支持快速回滚

9 迁移最佳实践

1. **充分测试**：在测试环境完成完整迁移流程验证
2. **备份数据**：迁移前备份原集群配置和数据
3. **监控迁移过程**：实时监控迁移过程中的各项指标
4. **制定回滚计划**：确保在出现问题时能快速回滚
5. **选择低峰期**：在业务低峰期执行迁移操作
6. **逐步切换**：先切换非核心业务，验证无误后再切换核心业务
7. **保留原集群**：迁移完成后保留原集群一段时间，以备查询

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

