



消息引擎用户手册

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 核心概念
 - 2.1 AMQP 协议
 - 2.2 消息模型
- 3 交换机 (Exchange)
 - 3.1 交换机类型
 - 3.1.1 Direct Exchange
 - 3.1.2 Fanout Exchange
 - 3.1.3 Topic Exchange
 - 3.1.4 Headers Exchange
 - 3.2 交换机属性
- 4 队列 (Queue)
 - 4.1 队列类型
 - 4.1.1 Classic Queue
 - 4.1.2 Quorum Queue
 - 4.1.3 Stream Queue
 - 4.2 队列属性
 - 4.3 死信队列
- 5 消息机制
 - 5.1 消息确认
 - 5.1.1 自动确认 (Auto Ack)
 - 5.1.2 手动确认 (Manual Ack)
 - 5.2 消息持久化
 - 5.2.1 队列持久化
 - 5.2.2 消息持久化
 - 5.3 消息优先级
 - 5.4 延迟队列
- 6 集群与高可用
 - 6.1 集群模式

- 6.1.1 普通集群
 - 6.1.2 镜像队列
 - 6.1.3 Quorum Queue
- 6.2 集群运维命令
- 7 安全配置
 - 7.1 用户认证
 - 7.1.1 创建用户
 - 7.1.2 配置权限
 - 7.2 SSL/TLS 配置
 - 7.3 访问控制
- 8 性能调优参数
 - 8.1 内存管理
 - 8.2 磁盘管理
 - 8.3 连接与通道
 - 8.4 队列参数
- 9 常用工具
 - 9.1 bin/admq rabbitmq admin ctl
 - 9.2 bin/admq rabbitmq admin diagnostics
 - 9.3 rabbitmq-plugins

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）消息引擎的用户手册，详细介绍了ADMQ for RabbitMQ消息引擎的功能使用、配置方法及管理操作等内容。

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 核心概念

2.1 AMQP 协议

ADMQ for RabbitMQ 完全兼容 AMQP 0-9-1 协议，支持以下核心概念：

概念	说明
Producer	生产者，负责发送消息到 Exchange
Consumer	消费者，负责从 Queue 接收消息
Exchange	交换机，负责接收和路由消息到 Queue
Queue	队列，负责存储消息
Binding	绑定关系，定义 Exchange 到 Queue 的路由规则
Routing Key	路由键，用于消息路由匹配
Channel	通道，轻量级的连接，建立在真实连接之上

2.2 消息模型

```
Producer -> Exchange -> [Binding/Routing Key] -> Queue -> Consumer
```

1. 生产者发送消息到 Exchange
2. Exchange 根据类型和 Routing Key 决定如何路由消息
3. 消息通过 Binding 路由到匹配的 Queue
4. 消费者从 Queue 接收消息

3 交换机 (Exchange)

3.1 交换机类型

3.1.1 Direct Exchange

- 路由方式：精确匹配 Routing Key
- 适用场景：点对点消息、简单路由
- 示例：Routing Key 为 "order.create" 的消息只会路由到绑定键为 "order.create" 的队列

3.1.2 Fanout Exchange

- 路由方式：广播到所有绑定的队列，忽略 Routing Key
- 适用场景：广播通知、群发消息、缓存刷新
- 示例：发送一条消息到 Fanout Exchange，所有绑定的队列都会收到

3.1.3 Topic Exchange

- 路由方式：模式匹配 Routing Key，支持通配符
 - * 匹配一个单词
 - # 匹配零个或多个单词
- 适用场景：日志分类、业务事件路由、多级别订阅
- 示例：
 - 绑定键 `order.*` 匹配 `order.create`、`order.update`
 - 绑定键 `order.#` 匹配 `order.create`、`order.create.success`

3.1.4 Headers Exchange

- 路由方式：基于消息头 (Headers) 进行匹配
- 适用场景：需要复杂条件路由，不依赖 Routing Key
- 示例：匹配 header 中 `x-match=all` 且 `type=order`、`status=paid`

3.2 交换机属性

属性	说明
持久化 (Durable)	交换机是否在服务重启后保留
自动删除 (Auto-delete)	当最后一个队列解绑后自动删除
内部 (Internal)	只允许其他交换机绑定，不允许直接发布消息

4 队列 (Queue)

4.1 队列类型

4.1.1 Classic Queue

- **特点：**传统队列类型，支持镜像模式
- **适用场景：**一般业务场景，对吞吐量要求不是极高的场景
- **配置选项：**
 - 镜像模式：在多个节点间同步队列数据
 - 持久化：消息持久化到磁盘

4.1.2 Quorum Queue

- **特点：**基于 Raft 共识算法的高可用队列，自动实现多副本
- **适用场景：**对高可用和数据安全要求高的场景
- **优势：**
 - 自动故障转移
 - 强一致性保证
 - 无需配置镜像策略
- **限制：**
 - 不支持独占队列
 - 不支持消息 TTL（队列级 TTL 支持）
 - 不支持优先级队列

4.1.3 Stream Queue

- **特点：**只追加日志类型队列，支持多次读取
- **适用场景：**高吞吐量、消息回放、日志收集场景
- **优势：**
 - 极高的写入吞吐量
 - 支持多个消费者独立读取
 - 消息不会自动删除，可配置保留策略
- **特性：**
 - 基于偏移量（offset）消费
 - 支持时间戳定位
 - 消息持久化存储

4.2 队列属性

属性	说明
持久化 (Durable)	队列是否在服务重启后保留
自动删除 (Auto-delete)	当最后一个消费者断开后自动删除
排他性 (Exclusive)	只允许创建者连接使用
消息 TTL	消息在队列中的存活时间
队列 TTL	队列在空闲多久后自动删除
最大长度	队列最大消息数限制
最大字节	队列最大字节数限制
死信交换机 (DLX)	消息被拒绝或过期后转发的交换机
死信路由键 (DLK)	死信消息使用的路由键

4.3 死信队列

死信队列用于处理无法正常消费的消息。

消息成为死信的几种情况：

1. 消息被消费者拒绝 (basic.reject 或 basic.nack) ， 且 requeue=false
2. 消息过期 (TTL 到期)
3. 队列达到最大长度，新消息被挤出

配置死信队列：

1. 创建死信交换机 (DLX)
2. 创建死信队列并绑定到 DLX
3. 在业务队列中设置 `x-dead-letter-exchange` 和 `x-dead-letter-routing-key`

5 消息机制

5.1 消息确认

5.1.1 自动确认 (Auto Ack)

- 消息发送给消费者后立即从队列删除
- 优点：性能高，延迟低
- 缺点：消费者处理失败时消息丢失

5.1.2 手动确认 (Manual Ack)

- 消费者处理完成后发送确认 (basic.ack)
- 支持批量确认 (multiple=true)
- 支持否定确认 (basic.nack) ，可选择是否重新入队
- 优点：保证消息不丢失
- 缺点：性能略低

5.2 消息持久化

5.2.1 队列持久化

创建队列时设置 durable=true，队列元数据会持久化到磁盘。

5.2.2 消息持久化

发送消息时设置 delivery_mode=2，消息内容会持久化到磁盘。

```
channel.basicPublish(EXCHANGE, ROUTING_KEY,  
    MessageProperties.PERSISTENT_TEXT_PLAIN,  
    message.getBytes());
```

5.3 消息优先级

支持为消息设置优先级 (0-255) 。

1. 创建队列时设置 x-max-priority
2. 发送消息时设置 priority

```
Map<String, Object> args = new HashMap<>();
args.put("x-max-priority", 10);
channel.queueDeclare(Queue, true, false, false, args);

AMQP.BasicProperties props = new AMQP.BasicProperties.Builder()
    .priority(5)
    .build();
channel.basicPublish(Exchange, RoutingKey, props,
    message.getBytes());
```

5.4 延迟队列

支持延迟消息投递。

1. 使用死信交换机 + TTL 实现延迟队列
2. 或使用插件支持的延迟交换机 (x-delayed-message)

6 集群与高可用

6.1 集群模式

6.1.1 普通集群

- 节点间共享元数据（Exchange、Binding、Queue 定义）
- 队列数据仅存储在声明队列的节点上
- 优点：资源占用低
- 缺点：节点故障时该节点上的队列不可用

6.1.2 镜像队列

- Classic Queue 支持镜像模式
- 队列数据同步到多个节点
- 配置策略：

```
# 设置所有队列镜像到所有节点
bin/admq rabbitmq admin ctl set_policy ha-all "^" '{"ha-mode":"all"}'

# 设置队列镜像到指定数量节点
bin/admq rabbitmq admin ctl set_policy ha-two "^" '{"ha-mode":"exactly","ha-params":2}'
```

6.1.3 Quorum Queue

- 基于 Raft 共识算法自动实现多副本
- 无需配置镜像策略
- 自动 Leader 选举和故障转移

6.2 集群运维命令

```
# 查看集群状态
bin/admq rabbitmq admin ctl cluster_status

# 添加节点到集群
bin/admq rabbitmq admin ctl join_cluster rabbit@node1
```

移除节点

```
bin/admq rabbitmq admin ctl forget_cluster_node rabbit@node2
```

查看队列详情

```
bin/admq rabbitmq admin ctl list_queues name pid slave_pids  
synchronised_slave_pids
```

查看连接

```
bin/admq rabbitmq admin ctl list_connections peer_host peer_port  
state
```

7 安全配置

7.1 用户认证

7.1.1 创建用户

```
bin/admq rabbitmq admin ctl add_user username password
bin/admq rabbitmq admin ctl set_user_tags username administrator
```

7.1.2 配置权限

```
bin/admq rabbitmq admin ctl set_permissions -p /vhost username ".*"
".*" ".*"
```

权限格式:

- 第一个参数: Configure 权限 (创建/删除资源)
- 第二个参数: Write 权限 (发送消息)
- 第三个参数: Read 权限 (接收消息)

7.2 SSL/TLS 配置

```
# 配置 SSL 监听器
listeners.ssl.default = 5671
ssl_options.cacertfile = /path/to/ca_certificate.pem
ssl_options.certfile = /path/to/server_certificate.pem
ssl_options.keyfile = /path/to/server_key.pem
ssl_options.verify = verify_peer
ssl_options.fail_if_no_peer_cert = false
```

7.3 访问控制

支持基于 Vhost 的资源隔离, 每个用户在不同 Vhost 下可拥有不同权限。

8 性能调优参数

8.1 内存管理

参数	说明	默认值
vm_memory_high_watermark	内存使用上限比例	0.4
vm_memory_high_watermark_paging_ratio	开始分页的内存比例	0.5

8.2 磁盘管理

参数	说明	默认值
disk_free_limit.absolute	磁盘空间下限	50MB
disk_free_limit.relative	磁盘空间下限比例	2.0

8.3 连接与通道

参数	说明	默认值
channel_max	每个连接的最大通道数	2047
connection_max	最大连接数	infinity
heartbeat	心跳间隔（秒）	60

8.4 队列参数

参数	说明	默认值
queue_master_locator	队列 Leader 分配策略	min-masters
lazy_queue_explicit_gc_run_operation_threshold	惰性队列 GC 阈值	1000

9 常用工具

9.1 bin/admq rabbitmq admin ctl

RabbitMQ 命令行管理工具。

服务管理

```
bin/admq rabbitmq admin ctl start_app
```

```
bin/admq rabbitmq admin ctl stop_app
```

```
bin/admq rabbitmq admin ctl status
```

用户管理

```
bin/admq rabbitmq admin ctl list_users
```

```
bin/admq rabbitmq admin ctl add_user user password
```

```
bin/admq rabbitmq admin ctl delete_user user
```

```
bin/admq rabbitmq admin ctl change_password user newpassword
```

队列管理

```
bin/admq rabbitmq admin ctl list_queues
```

```
bin/admq rabbitmq admin ctl list_queues name messages consumers
```

交换机管理

```
bin/admq rabbitmq admin ctl list_exchanges name type
```

连接管理

```
bin/admq rabbitmq admin ctl list_connections peer_host state
```

9.2 bin/admq rabbitmq admin diagnostics

诊断工具。

检查节点健康

```
bin/admq rabbitmq admin diagnostics health_check
```

查看节点状态

```
bin/admq rabbitmq admin diagnostics status

# 查看内存使用
bin/admq rabbitmq admin diagnostics memory_breakdown

# 查看集群状态
bin/admq rabbitmq admin diagnostics cluster_status
```

9.3 rabbitmq-plugins

插件管理工具。

```
# 列出插件
rabbitmq-plugins list

# 启用插件
rabbitmq-plugins enable rabbitmq_management
rabbitmq-plugins enable rabbitmq_shovel
rabbitmq-plugins enable rabbitmq_federation

# 禁用插件
rabbitmq-plugins disable rabbitmq_management
```

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

