



遷移手冊

金蝶Apusic分布式消息队列V2.0.6_for_rabbitmq

版權所有 © 深圳市金蝶天燕雲計算股份有限公司2026。保留所有權利。

版權聲明

本文件所涉及的軟件著作權、版權等知識產權已依法進行了註冊，由金蝶天燕雲計算股份有限公司合法擁有。受《中華人民共和國著作權法》《計算機軟件保護條例》《知識產權保護條例》和相關國際版權條約、法律、法規以及其它知識產權法律和條約的保護。未經授權許可，不得非法使用。

免責聲明

本文件包含的版權資訊由金蝶天燕雲計算股份有限公司合法擁有，受法律的保護，金蝶天燕雲計算股份有限公司對本文件可能涉及到的非金蝶天燕雲計算股份有限公司的資訊不承擔任何責任。在法律允許的範圍內，您可以查閱並僅能夠在《中華人民共和國著作權法》規定的合法範圍內複製和打印本文件。任何單位和個人未經金蝶天燕雲計算股份有限公司書面授權許可，不得使用、修改、再發佈本文件的任何部分和內容，否則將被視為侵權，金蝶天燕雲計算股份有限公司有依法追究其責任的權利。

本文件如有更新，不另行通知。對本文件中的問題您可向金蝶天燕雲計算股份有限公司告知或查詢。未經本公司明確授予的任何權利均予保留。

商標聲明

 是深圳市金蝶天燕雲計算股份有限公司向中華人民共和國國家商標局申請註冊的註冊商標，註冊商標專用權由金蝶天燕合法擁有，受法律保護。未經金蝶天燕的書面許可，任何單位及個人不得以任何方式或理由對該商標的任何部分進行使用、複製、修改、傳播、抄錄或與其它產品捆綁使用銷售。凡侵犯金蝶天燕商標權的，金蝶天燕將依法追究其法律責任。本文件提及的其他所有商標或註冊商標，由各自的所有人擁有。

目錄

- 1 前言
 - 1.1 適用對象
 - 1.2 相關文件
 - 1.3 技術支援
- 2 遷移概述
 - 2.1 遷移原則
 - 2.2 遷移範圍
 - 2.3 遷移前檢查
 - 2.3.1 版本相容性檢查
 - 2.3.2 客戶端相容性檢查
- 3 遷移方案
 - 3.1 方案一：雙寫遷移（推薦）
 - 3.2 方案二：停服遷移
 - 3.3 方案三：灰度遷移
- 4 設定遷移
 - 4.1 匯出原 RabbitMQ 設定
 - 4.2 在 ADMQ for RabbitMQ 中建立設定
 - 4.2.1 使用管控台建立
 - 4.2.2 使用命令列建立
- 5 資料遷移
 - 5.1 使用 Shovel 遷移資料
 - 5.1.1 設定 Shovel
 - 5.1.2 建立 Shovel 設定
 - 5.2 使用 Federation 遷移資料
 - 5.3 使用客戶端指令碼遷移
- 6 客戶端遷移
 - 6.1 連線位址變更
 - 6.2 SSL/TLS 設定變更
 - 6.3 常見客戶端設定
 - 6.3.1 Java (Spring Boot)
 - 6.3.2 Python (Django)
 - 6.3.3 Go

- 7 遷移驗證
 - 7.1 功能驗證
 - 7.2 效能驗證
 - 7.3 資料一致性驗證
- 8 回滾方案
 - 8.1 回滾檢查清單
- 9 遷移最佳實踐

1 前言

本文檔為金蝶Apusic分散式訊息佇列for RabbitMQ（Apusic Distributed Message Queue，簡稱：ADMQ for RabbitMQ）的遷移手冊，詳細介紹了如何將RabbitMQ遷移到ADMQ for RabbitMQ。

1.1 適用對象

本文檔適用於IT資訊化業務負責人、研發經理、軟體專案經理、軟體架構師、維運工程師。

1.2 相關文件

了解更多ADMQ for RabbitMQ產品相關的資訊，請參閱以下ADMQ for RabbitMQ產品手冊文件集：

序號	手冊文件	說明
1	金蝶Apusic分散式訊息佇列for RabbitMQ 快速使用手冊	簡單介紹如何快速上手使用ADMQ for RabbitMQ。
2	金蝶Apusic分散式訊息佇列for RabbitMQ 安裝手冊	詳細介紹如何在各作業系統上安裝ADMQ for RabbitMQ，以及ADMQ for RabbitMQ服務啟停等操作。
3	金蝶Apusic分散式訊息佇列for RabbitMQ 訊息引擎使用者手冊	詳細介紹ADMQ for RabbitMQ訊息引擎相關功能的使用、設定、管理及配套工具的使用方法。
4	金蝶Apusic分散式訊息佇列for RabbitMQ 管控台使用者手冊	詳細介紹ADMQ for RabbitMQ管控台相關功能的使用和操作說明。
5	金蝶Apusic分散式訊息佇列for RabbitMQ 開發手冊	詳細介紹基於各開發語言進行ADMQ for RabbitMQ客戶端應用開發的說明。
6	金蝶Apusic分散式訊息佇列for RabbitMQ 遷移手冊	詳細介紹從RabbitMQ遷移到ADMQ for RabbitMQ的說明。
7	金蝶Apusic分散式訊息佇列for RabbitMQ 維運手冊	詳細介紹ADMQ for RabbitMQ的監控、維運、安全強化等維運說明。
8	金蝶Apusic分散式訊息佇列for RabbitMQ 效能最佳化手冊	詳細介紹ADMQ for RabbitMQ效能調校的說明。

1.3 技術支援

ADMQ for RabbitMQ產品提供全面的技術支援服務，您可以透過以下方式獲得技術支援：

- 網址: www.apusic.com
- 電話: 400-855-5800
- 郵箱: support@apusic.com
- 金蝶雲社群: <https://vip.kingdee.com/?productId=73&productLineId=14&lang=zh-CN>

您在取得技術支援時，請提供如下資訊：

1. 您的姓名
2. 公司資訊與聯絡方式
3. 作業系統及其版本
4. 產品版本號
5. 出現異常及錯誤的日誌、截圖等詳細資訊

2 遷移概述

2.1 遷移原則

ADMQ for RabbitMQ 完全相容 RabbitMQ 的 AMQP 0-9-1 協定，遷移過程遵循以下原則：

1. **零程式碼改造**：客戶端無需修改程式碼，僅需變更連線位址
2. **設定相容**：支援 RabbitMQ 原生設定參數
3. **資料可遷移**：支援佇列資料和設定資訊遷移
4. **平滑切換**：支援雙寫和灰度切換，降低遷移風險

2.2 遷移範圍

遷移內容	是否支援	說明
應用程式碼	是	無需修改，僅需變更連線位址
客戶端設定	是	支援 RabbitMQ 原生客戶端
Exchange 定義	是	支援所有型別交換器
Queue 定義	是	支援 Classic/Quorum/Stream 佇列
Binding 關係	是	支援所有繫結關係
使用者權限	是	支援使用者和 Vhost 權限遷移
訊息資料	是	支援歷史訊息遷移
策略 (Policy)	是	支援映象策略等
外掛	部分	部分外掛可能需要適配

2.3 遷移前檢查

2.3.1 版本相容性檢查

原 RabbitMQ 版本	相容性	說明
3.8.x	完全相容	推薦遷移版本
3.9.x	完全相容	推薦遷移版本
3.10.x	完全相容	推薦遷移版本

3.11.x	完全相容	推薦遷移版本
3.12.x	完全相容	推薦遷移版本

2.3.2 客戶端相容性檢查

客戶端型別	相容版本	說明
Java (amqp-client)	5.x	完全相容
Python (pika)	1.x	完全相容
Go (amqp091-go)	1.x	完全相容
Node.js (amqplib)	0.10.x	完全相容
.NET (RabbitMQ.Client)	6.x	完全相容

3 遷移方案

3.1 方案一：雙寫遷移（推薦）

適用於不能停機的業務場景，透過雙寫保證資料一致性。

步驟：

1. 部署 ADMQ for RabbitMQ 叢集

- 按照安裝手冊部署新叢集
- 設定網路和防火牆

2. 設定雙寫

- 在應用中同時連線原 RabbitMQ 和 ADMQ for RabbitMQ
- 傳送訊息時同時傳送到兩個叢集

3. 同步歷史資料

- 使用 Shovel 或 Federation 外掛同步佇列資料
- 驗證資料一致性

4. 切換消費者

- 先切換部分消費者到 ADMQ for RabbitMQ
- 觀察業務是否正常

5. 全量切換

- 確認無誤後，切換所有消費者
- 停止向原 RabbitMQ 傳送訊息

6. 下線原叢集

- 保留一段時間用於資料查詢
- 確認無問題後下線

3.2 方案二：停服遷移

適用於可以短暫停機的業務場景，遷移速度快。

步驟：

1. 停止生產者

- 暫停業務系統向 RabbitMQ 傳送訊息

2. 匯出設定

- 使用 bin/admq rabbitmq admin ctl 匯出 Exchange、Queue、Binding、User 等設定

3. 遷移設定到 ADMQ for RabbitMQ

- 在管控台或命令列建立對應的資源

4. 遷移訊息資料

- 使用 shovel 匯出原佇列訊息
- 匯入到 ADMQ for RabbitMQ

5. 切換連線位址

- 修改應用設定，指向 ADMQ for RabbitMQ

6. 啟動生產者

- 恢復業務系統訊息傳送

3.3 方案三：灰度遷移

適用於大規模系統，逐步遷移不同業務模組。

步驟：

1. 按業務模組劃分遷移批次
2. 每批次按雙寫遷移方案執行
3. 驗證通過後遷移下一批次
4. 全部完成後下線原叢集

4 設定遷移

4.1 匯出原 RabbitMQ 設定

```
# 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 中建立設定

4.2.1 使用管控台建立

1. 登入管控台
2. 進入「資源管理」頁面
3. 按順序建立：
 - Vhost
 - MQ 使用者及權限
 - Exchange
 - Queue

- Binding

4.2.2 使用命令列建立

```
# 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 資料遷移

5.1 使用 Shovel 遷移資料

Shovel 是 RabbitMQ 內建的資料遷移工具。

5.1.1 設定 Shovel

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

5.1.2 建立 Shovel 設定

```
{
  "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 遷移資料

Federation 適用於跨叢集的資料同步。

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

5.3 使用客戶端指令碼遷移

對於小規模資料，可以編寫客戶端指令碼遷移：

```
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 客戶端遷移

6.1 連線位址變更

僅需修改應用設定檔案中的連線位址：

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

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

6.2 SSL/TLS 設定變更

如果啟用了 SSL，需要更新憑證：

```
# 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 常見客戶端設定

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 遷移驗證

7.1 功能驗證

驗證項	驗證方法
連線測試	使用客戶端連線 ADMQ for RabbitMQ，確認能正常建立連線
訊息傳送	傳送測試訊息，確認能成功到達 Exchange
訊息路由	驗證訊息能正確路由到 Queue
訊息消費	啟動消費者，確認能正常接收和處理訊息
訊息確認	測試手動確認和自動確認功能
持久化	重新啟動服務後驗證訊息是否保留
叢集功能	驗證節點故障時是否能自動切換

7.2 效能驗證

驗證項	驗證方法
吞吐量	對比原叢集和新叢集的訊息傳送速率
延遲	測量訊息從傳送到消費的延遲
並發	測試多生產者、多消費者並發場景
穩定性	長時間執行測試，觀察是否有異常

7.3 資料一致性驗證

```
# 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 回滾方案

如果遷移過程中出現問題，需要回滾到原 RabbitMQ 叢集：

1. 停止向 ADMQ for RabbitMQ 傳送訊息
2. 切換連線位址回原 RabbitMQ
3. 同步 ADMQ for RabbitMQ 中的新訊息回原叢集（如有需要）
4. 恢復業務系統

8.1 回滾檢查清單

- ☐ 原 RabbitMQ 叢集仍在執行
- ☐ 原叢集資料未被刪除
- ☐ 應用設定可快速切換
- ☐ 業務系統支援快速回滾

9 遷移最佳實踐

1. **充分測試**：在測試環境完成完整遷移流程驗證
2. **備份資料**：遷移前備份原叢集設定和資料
3. **監控遷移過程**：即時監控遷移過程中的各項指標
4. **制定回滾計畫**：確保在出現問題時能快速回滾
5. **選擇低峰期**：在業務低峰期執行遷移操作
6. **逐步切換**：先切換非核心業務，驗證無誤後再切換核心業務
7. **保留原叢集**：遷移完成後保留原叢集一段時間，以備查詢

全國統一服務熱線
4008-555-800

金蝶天燕雲計算股份有限公司（簡稱“金蝶天燕”）成立於2000年，前身為“金蝶中間件公司”，是金蝶集團旗下基礎軟件平台提供商，雲計算國家標準制定企業，國家信創產業核心軟件企業。金蝶天燕是國家863重點研發計劃與核高基重大專項承接企業，也是“兩網一站四庫十二金”國家重點工程的基礎平台提供商，產品廣泛應用於政府、軍工、金融、能源等關鍵行業，累計服務客戶總數超過10萬家。

Apusic
金蝶天燕

雲計算國家標準制定企業
金蝶集團旗下基礎軟件企業
資訊技術應用創新核心企業
官網：www.apusic.com

