



APUSIC
固若长城
睿比世界

安裝手冊

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 部署架構
 - 2.1 整體架構
 - 2.2 元件說明
 - 2.3 部署模式
 - 2.4 架構特點
- 3 環境準備
 - 3.1 系統要求
 - 3.2 網路要求
- 4 單機部署
 - 4.1 建立部署目錄
 - 4.2 下載並解壓軟體包
 - 4.3 目錄結構說明
 - 4.4 配置修改
 - 4.4.1 修改主機名稱
 - 4.4.2 修改監聽埠等資訊（可選）
 - 4.5 授權許可
 - 4.6 啟動服務
 - 4.7 服務驗證
 - 4.7.1 檢查程序狀態
 - 4.7.2 檢查埠監聽
 - 4.7.3 檢視執行日誌
 - 4.7.4 建立使用者
 - 4.7.5 建立虛擬主機
- 5 叢集部署
 - 5.1 部署規劃
 - 5.2 所有節點執行：基礎安裝
 - 5.2.1 建立目錄並解壓
 - 5.2.2 放置授權檔案

- 5.2.3 修改主機名稱
- 5.2.4 修改伺服器 hosts 檔案
- 5.3 配置叢集（可選）
 - 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 啟動各節點服務
- 5.5 建立叢集
- 5.6 映象模式（可選）
- 5.7 叢集驗證與測試
 - 5.7.1 驗證各個節點的埠狀態
 - 5.7.2 檢查叢集狀態
 - 5.7.3 功能測試
- 5.8 叢集故障模擬與維護
 - 5.8.1 停止服務
 - 5.8.2 常見錯誤排查
- 6 常見問題及排查
 - 6.1 服務啟動失敗
 - 6.2 叢集加入失敗
 - 6.3 管控臺無法訪問
 - 6.4 資料來源 403 錯誤

1 前言

本文档為金蝶Apusic分散式訊息佇列for RabbitMQ（Apusic Distributed Message Queue，簡稱：ADMQ for RabbitMQ）的安裝手冊，提供產品各種部署模式（含單機、叢集）在各作業系統、容器環境的安裝部署過程指導以及相關配置說明。

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、運維工程師。

1.2 相關文件

瞭解更多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	簡單介紹瞭如何快速上手使用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、配置、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 管控臺使用者手冊	詳細介紹ADMQ for RabbitMQ管控臺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 運維手冊	詳細介紹ADMQ for RabbitMQ的監控、運維、安全加固等運維說明。
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效能調優的說明。

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：

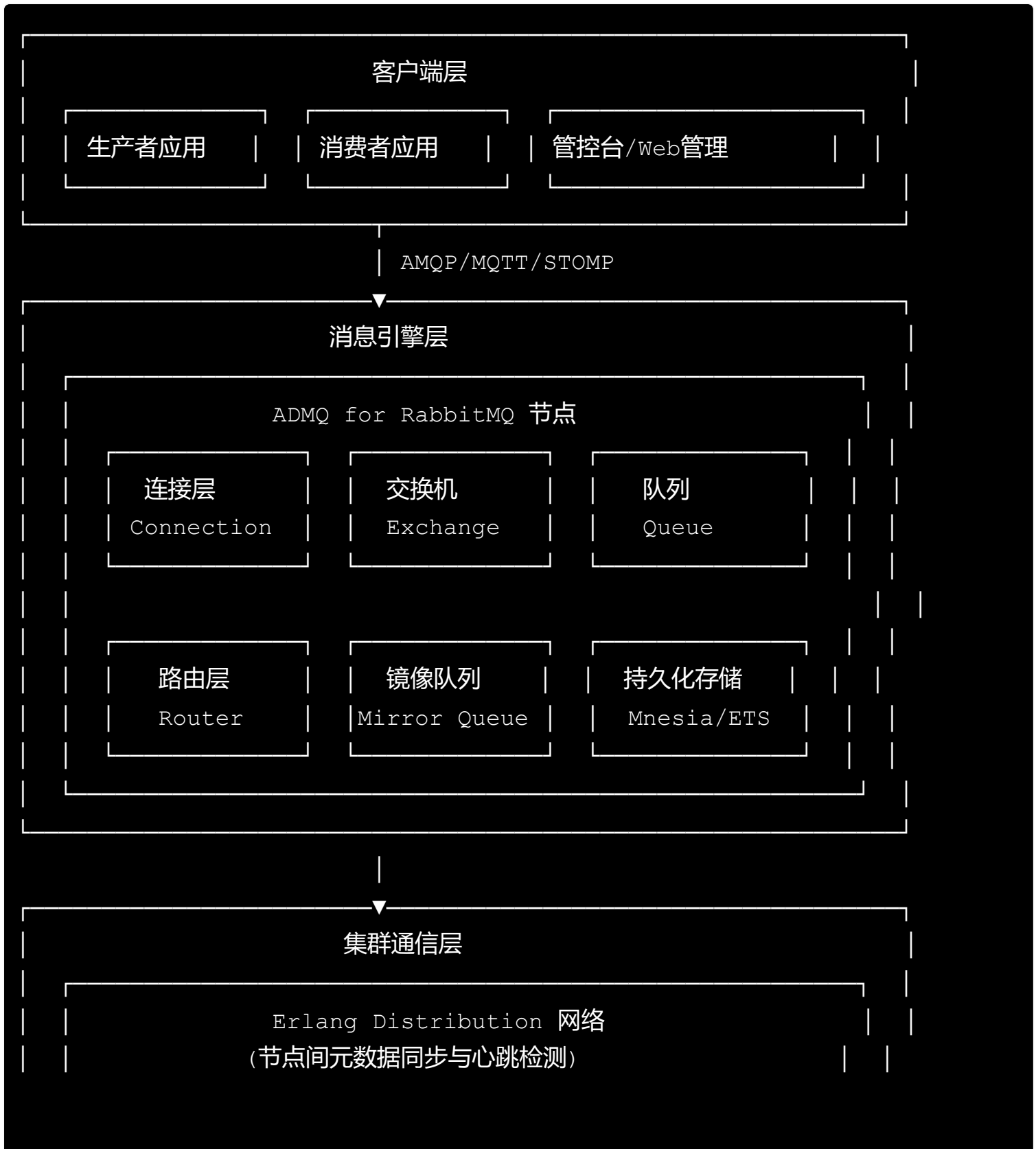
1. 您的姓名
2. 公司資訊與聯絡方式
3. 作業系統及其版本
4. 產品版本號
5. 出現異常及錯誤的日誌、截圖等詳細資訊

ADMQ for Kafka支援透過管控臺部署，請參考《金蝶Apusic分散式訊息佇列V2.0.6_統一管控臺使用者手冊.pdf》，**【快速入門】** 章節。

2 部署架构

2.1 整体架构

ADMQ for RabbitMQ 采用模组化设计，整体架构由以下核心元件组成：



2.2 元件說明

元件	說明	功能
連線層	Connection	處理客戶端連線、協定解析、安全認證
交換機	Exchange	接收訊息並根據路由規則分發到佇列
佇列	Queue	儲存訊息，支援普通佇列和 Quorum Queues
映象佇列	Mirror Queue	叢集模式下同步佇列資料到多個節點
持久化儲存	Mnesia/ETS	儲存元資料、配置資訊和持久化訊息
叢集通訊	Erlang Distribution	節點間通訊，同步元資料和叢集狀態

2.3 部署模式

ADMQ for RabbitMQ 支援以下部署模式：

部署模式	節點數	適用場景	高可用性
單機部署	1 節點	開發測試、低吞吐量場景	無
普通叢集	2+ 節點	生產環境，需要橫向擴充套件	元資料高可用
映象佇列叢集	3+ 節點	生產環境，訊息資料高可用	元資料+資料高可用

2.4 架構特點

1. 分散式設計：支援多節點叢集部署，元資料自動同步
2. 高可用性：支援映象佇列，確保訊息資料在節點故障時不丟失
3. 彈性擴充套件：可動態新增節點擴充套件叢集處理能力
4. 多協定支援：原生支援 AMQP 0-9-1 協定
5. 管控分離：訊息引擎與管控臺可獨立部署

3 環境準備

3.1 系統要求

元件	最低要求	推薦配置
CPU	4 核	8 核及以上
記憶體	8 GB	16 GB 及以上
磁碟	100 GB SSD	500 GB 及以上 SSD
網路	1 Gbps	10 Gbps
作業系統	Linux (CentOS 7+, Ubuntu 18.04+, 麒麟, 統信)	-

3.2 網路要求

埠	用途	說明
5672	AMQP 協定	生產者和消費者連線
5671	AMQP over SSL	SSL 加密連線
15672	Management HTTP	HTTP API 和 Web 管理介面
15671	Management HTTPS	HTTPS API 和 Web 管理介面
15692	Prometheus Metrics	監控採集指標
25672	Erlang Distribution	叢集節點間通訊
12305	管控臺 HTTP	管控臺 Web 介面 (HTTP)
12306	管控臺 HTTPS	管控臺 Web 介面 (HTTPS)

4 單機部署

本文檔指導使用者在 Linux 環境下完成 ADMQ RabbitMQ 的單機模式部署，包含環境準備、安裝配置、授權申請、服務啟動及狀態驗證。

4.1 建立部署目錄

首先建立應用根目錄並進入：

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

4.2 下載並解壓軟體包

從內部倉庫下載指定版本的安裝包並解壓。

```
# 下載安裝包  
# 實施人員提供  
  
# 解壓安裝包  
tar zxvf ADMQ-V2.0.6.31214-RabbitMQ-20260227.tar.gz
```

4.3 目錄結構說明

解壓完成後，當前目錄下將生成 `admq-rabbitmq`。標準目錄結構如下：

```
admq-rabbitmq/  
├── bin/           # 可執行指令碼目錄  
├── config/        # 配置檔案目錄  
├── data/          # 資料儲存目錄  
├── logs/          # 日誌檔案目錄  
├── lib/           # 依賴庫目錄  
└── license.xml    # 授權檔案（需自行放置）
```

4.4 配置修改

4.4.1 修改主機名稱

主機名稱只能包含：

- 字母 (a-z A-Z)
- 數字 (0-9)
- 連字元 (-)

```
# 檢視主機名稱
hostname

# 修改主機名
hostname node84
```

4.4.2 修改監聽埠等資訊（可選）

編輯 `config/rabbitmq.conf`，可以修改監聽埠、使用者名稱密碼等資訊。

```
# 示例：修改監聽埠和預設使用者
listeners.tcp.default = 5672
default_user = admq
default_pass = apusic_123
```

4.5 授權許可

ADMQ 產品需要 License 檔案才能正常執行。

1. **申請授權**：聯絡管理員或透過授權平臺申請對應版本的 `license.xml` 檔案。
2. **放置檔案**：將獲取到的 `license.xml` 檔案上傳至部署目錄根路徑。

```
# 確認檔案位置
ls -l /apusic/admq-rabbitmq/license.xml
```

4.6 啟動服務

啟動訊息收發服務：

```
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 服務驗證

4.7.1 檢查程序狀態

使用 `ps` 命令檢視程序，正常狀態下應能看到相關程序：

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

預期輸出示例：

```
[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      1  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 1 048576 -t 5000000 -stbt db -zdbbl 128000 -sbwt none -sbwtcpu 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-feat ure maybe_expr
root      20355      1  0 14:09 ?        00:00:00 /apusic/admq-rabbitmq/rabbitmq/exec/xdkQuM/erts-14.2.5.11/bin/inet_gethost 4
root      20356      1  0 14:09 ?        00:00:00 /apusic/admq-rabbitmq/rabbitmq/exec/xdkQuM/erts-14.2.5.11/bin/inet_gethost 4
root      20378      1  0 14:09 pts/4    00:00:00 grep --color=auto admq-rabbitmq
[root@localhost admq-rabbitmq]#
```

4.7.2 檢查埠監聽

使用 `netstat` 確認 5672 埠是否處於 `LISTEN` 狀態：

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

預期輸出示例：

```
[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]#
```

如果未看到輸出，請檢查防火牆設定或檢視日誌排查啟動失敗原因。

4.7.3 檢視執行日誌

如果服務啟動異常或需監控執行狀態，請檢視 `logs` 目錄下的日誌檔案：

```
# 檢視實時日誌
tail -f logs/server.log

# 或者檢視目錄結構
ls -lh logs/
```

重點關注日誌中是否有 `ERROR` 級別的資訊。

```
[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 建立使用者

預設已經開啟了認證，客戶端收發訊息需要新增使用者名稱密碼。

建立使用者：test

設定密碼：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 建立虛擬主機

建立虛擬主機：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" ...
```

設定許可權給使用者：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 叢集部署

5.1 部署規劃

假設部署 3 節點 高可用叢集，規劃如下：

前置要求：

1. 確保三臺機器時間同步（NTP）。
2. 關閉防火牆或開放埠： 5672 、 15672 。

5.2 所有節點執行：基礎安裝

在 Node 1、Node 2、Node 3 上分別執行：

5.2.1 建立目錄並解壓

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

5.2.2 放置授權檔案

將 license.xml 上傳至 /apusic/admq-rabbitmq/ 目錄。

5.2.3 修改主機名稱

主機名稱只能包含：

- 字母 (a-z A-Z)
- 數字 (0-9)
- 連字元 (-)

```
# 在 Node 1 執行
hostname node171

# 在 Node 2 執行
```

```
hostname node172

# 在 Node 3 執行
hostname node173
```

5.2.4 修改伺服器 hosts 檔案

需要在 `/etc/hosts` 新增叢集所有節點的資訊。

所有節點都新增如下資訊：

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

5.3 配置叢集（可選）

編輯 `config/rabbitmq.conf`，可以配置監聽埠、預設使用者名稱密碼等資訊。

5.3.1 Node 1 配置 (172.20.140.171)

```
listeners.tcp.default = 5672
# 預設使用者
default_user = admq
default_pass = apusic_123
```

5.3.2 Node 2 配置 (172.20.140.172)

```
listeners.tcp.default = 5672
# 預設使用者
default_user = admq
default_pass = apusic_123
```

5.3.3 Node 3 配置 (172.20.140.173)

```
listeners.tcp.default = 5672
# 預設使用者
default_user = admq
default_pass = apusic_123
```

5.4 啟動各節點服務

依次登入 Node 1、2、3 執行：

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

啟動後日誌如下：

```
[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]#
```

之後需要檢測節點狀態，確認是否啟動成功。可直接檢視埠是否處於監聽狀態。

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

預期輸出：

```
[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 建立叢集

在第一個節點執行叢集建立命令，把另外兩個節點加入到該節點組成叢集：

```
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 映象模式（可選）

叢集模式中，各個節點只能同步元資料，不能同步交換機、佇列下的資料。在沒使用 `Quorum Queues` 模式的情況下可以使用映象佇列同步各個節點的資料。

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

5.7 叢集驗證與測試

5.7.1 驗證各個節點的埠狀態

在每臺伺服器檢查埠監聽狀態：

```
# 檢查 5672 埠是否被監聽
ss -anpl | grep 5672
```

預期：可以看到埠被監聽。

```
[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 檢查叢集狀態

在 Node1 執行：

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

可以看到叢集中三個節點都已經新增完成。

```
[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 功能測試

1. 建立使用者：

建立使用者：test

設定密碼：test123

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

預期輸出示例：

```
[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. 建立虛擬主機：

建立虛擬主機：test

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

預期輸出示例：

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

設定許可權給使用者：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 叢集故障模擬與維護

5.8.1 停止服務

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

5.8.2 常見錯誤排查

叢集節點新增失敗，節點連線失敗

```
[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
```

解決方法：配置 `/etc/hosts`，新增 node172 的資訊，然後重啟該節點。

叢集節點新增失敗，節點停止失敗

```
[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
```

解決方法：重啟需要新增的節點。

6 常見問題及排查

6.1 服務啟動失敗

可能原因	解決方案
埠衝突	檢查 5672、15672、25672 埠是否被佔用
許可權不足	確保 admq-rabbitmq 目錄許可權正確
記憶體不足	檢查系統記憶體，確保滿足最低要求
License 未放置 或過期	確認 license.xml 已放置在部署目錄根路徑且未過期
OpenSSL 版本 不相容	Ubuntu 24.04 等系統預設使用 OpenSSL 3.x，而 Erlang crypto 模組依賴 libcrypto.so.1.1，需安裝 OpenSSL 1.1 相容庫（libssl1.1），安裝完成後重啟服務

OpenSSL 1.1 相容庫安裝示例（Ubuntu/Debian）：

```
# Ubuntu 24.04 需要先安裝 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
```

說明：如果在 logs/rabbitmq-server-*.out 中看到類似 crypto:strong_rand_bytes 或 libcrypto.so.1.1: cannot open shared object file 的錯誤，即為 OpenSSL 相容性問題。

6.2 叢集加入失敗

可能原因	解決方案
主機名不符合規範	確保主機名只包含字母、數字和連字元
網路不通	檢查節點間網路連通性和防火牆設定
主機名解析失敗	確保 /etc/hosts 或 DNS 配置正確
節點未正常啟動	檢查各節點服務是否已正常啟動

6.3 管控臺無法訪問

可能原因	解決方案
埠衝突	修改 config/application.properties 中的 server.port
防火牆阻擋	開放 12305/12306 埠
服務未啟動	檢查管控臺服務狀態

6.4 資料來源 403 錯誤

可能原因	解決方案
RabbitMQ 使用者許可權不足	使用 administrator 標籤使用者，配置 .* 許可權
訊息傳送和接收異常	檢查路由 key 和繫結關係是否正確

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。



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

