



APUSIC
固若长城
睿比世界

Quick Operation Guide

金蝶Apusic分布式消息队列V2.0.6_for_rabbitmq

Copyright Notice

All intellectual property rights including software copyrights involved in this document have been legally registered and duly owned by Kingdee Apusic Cloud Computing Co., Ltd. Such rights are protected by Copyright Law of the People's Republic of China, Regulations on Protection of Computer Software, Regulations on Intellectual Property Protection, relevant international copyright conventions, laws, regulations and other intellectual property legislations. Unauthorized use is strictly prohibited.

Disclaimer

All copyright information contained herein is legally owned by Kingdee Apusic Cloud Computing Co., Ltd. and protected by applicable laws. The Company shall bear no liability for any information unrelated to Kingdee Apusic contained in this document. You may view, copy and print this document only within the scope permitted by Copyright Law of the People's Republic of China. No entity or individual may use, modify or republish any part or content of this document without written authorization from Kingdee Apusic Cloud Computing Co., Ltd. Any violation shall constitute infringement, and the Company reserves the right to pursue legal liabilities accordingly.

Updates to this document shall not be notified separately. You may consult the Company regarding issues related to this document. All rights not expressly granted by the Company are reserved.

Trademark Notice

 is a registered trademark filed with the National Intellectual Property Administration of PRC by Kingdee Apusic Cloud Computing Co., Ltd. Shenzhen. Exclusive rights to the registered trademark are legally owned by Kingdee Apusic and protected by law. Without written consent of Kingdee Apusic, no entity or individual shall use, copy, modify, disseminate, transcribe, or bundle and sell any part of this trademark with other products in any manner or for any reason. Kingdee Apusic shall pursue legal liabilities against any infringement of its trademark rights. All other trademarks or registered trademarks mentioned in this document belong to their respective owners.

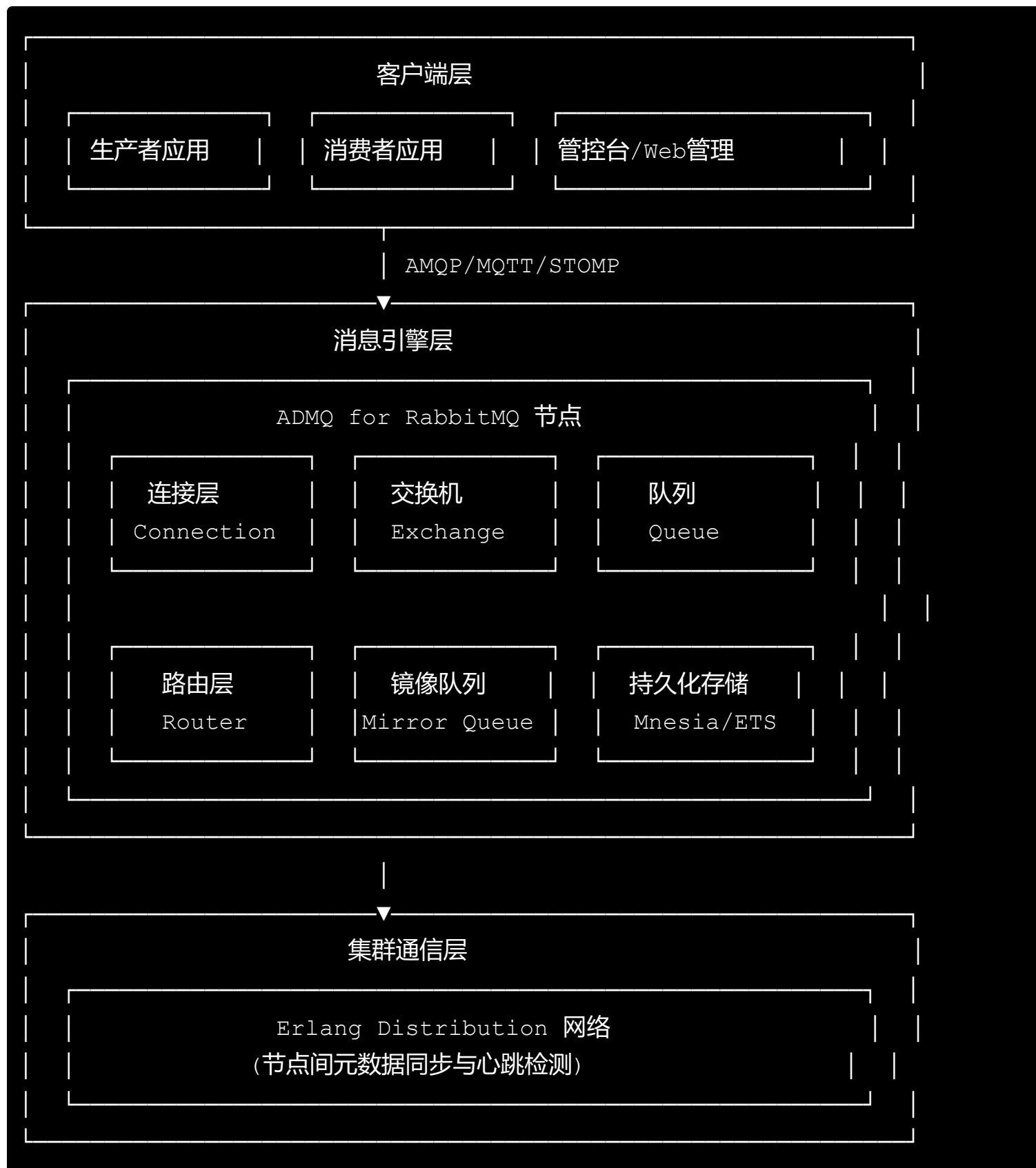
Contents

- 1 1. Deployment Architecture
 - 1.1 Overall Architecture
 - 1.2 Deployment Mode
- 2 2. Quick Installation
 - 2.1 2.1 Environment Preparation
 - 2.2 2.2 Standalone Installation
 - 2.3 2.3 Cluster Installation (3 Nodes)
- 3 3. Core Concepts
 - 3.1 Message Model
 - 3.2 Exchange Types
 - 3.3 Queue Types
- 4 4. Common Command Reference
 - 4.1 Service Management
 - 4.2 Status Inspection
 - 4.3 User Management
 - 4.4 Queue Management
 - 4.5 Cluster Management
 - 4.6 Policy Management
 - 4.7 Backup and Restore
- 5 5. Daily Operations
 - 5.1 Log Locations
 - 5.2 Daily Checklist
 - 5.3 Health Check Script
- 6 6. Configuration Optimization
 - 6.1 rabbitmq.conf Core Configuration
 - 6.2 System Optimization
- 7 7. Troubleshooting
 - 7.1 Service Won't Start
 - 7.2 Message Consumption Anomalies
 - 7.3 Cluster Anomalies
 - 7.4 Diagnostic Commands
- 8 8. Security Hardening

- 8.1 Change Default User
- 8.2 Firewall Configuration
- 8.3 SSL Configuration
- 9 9. Performance Optimization Tips
 - 9.1 Queue Optimization
 - 9.2 Client Optimization
 - 9.3 Monitoring Metrics
- 10 10. Operations Best Practices

1 1. Deployment Architecture

1.1 Overall Architecture



1.2 Deployment Mode

Deployment Mode	Nodes	Use Case	High Availability
Standalone	1 node	Development/Testing	None
Standard Cluster	2+ nodes	Production Environment	Metadata HA
Mirror Queue Cluster	3+ nodes	Production Environment	Metadata + Data HA

2 2. Quick Installation

2.1 2.1 Environment Preparation

Component	Minimum Requirement	Recommended Configuration
CPU	4 cores	8 cores or more
Memory	8 GB	16 GB or more
Disk	100 GB SSD	500 GB or more SSD
OS	CentOS 7+, Ubuntu 18.04+, Kylin, UOS	-

Port Requirements:

Port	Purpose
5672	AMQP protocol
15672	Web management interface
25672	Cluster inter-node communication

2.2 2.2 Standalone Installation

```
# Create directory
mkdir -p /apusic && cd /apusic

# Extract installation package
tar zxvf ADMQ-V2.0.6.xxx.tar.gz
cd admq-rabbitmq

# Place license.xml license file in the root directory
# Modify hostname (only letters, digits, and hyphens allowed)
hostname node1

# Start service
bin/admq-daemon start rabbitmq server
```

2.3 2.3 Cluster Installation (3 Nodes)

Execute on all nodes:

```
mkdir -p /apusic && cd /apusic
tar zxvf ADMQ-V2.0.6.xxx.tar.gz
cd admq-rabbitmq

# Place license.xml
# Modify hostname
hostname node171 # node172, node173

# Configure /etc/hosts
172.20.140.171 node171
172.20.140.172 node172
172.20.140.173 node173

# Start service
bin/admq-daemon start rabbitmq server
```

Execute cluster creation on Node 1:

```
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
```

Enable mirror queue (optional):

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

3 3. Core Concepts

3.1 Message Model

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

3.2 Exchange Types

Type	Routing Method	Use Case
Direct	Exact Routing Key match	Point-to-point messaging
Fanout	Broadcast to all queues	Broadcast notifications
Topic	Pattern matching * #	Log categorization, event routing
Headers	Match based on message headers	Complex conditional routing

3.3 Queue Types

Type	Characteristics	Use Case
Classic Queue	Traditional queue, supports mirroring	General business scenarios
Quorum Queue	Raft-based, automatic multi-replica	High availability scenarios
Stream Queue	Append-only log, supports multiple reads	High throughput, logging scenarios

4 4. Common Command Reference

4.1 Service Management

```
# Start/Stop/Restart
bin/admq-daemon start rabbitmq server
bin/admq-daemon stop rabbitmq server
bin/admq-daemon restart rabbitmq server

# Application start/stop
bin/admq rabbitmq admin ctl start_app
bin/admq rabbitmq admin ctl stop_app
```

4.2 Status Inspection

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

# Cluster status
bin/admq rabbitmq admin ctl cluster_status

# Health check
bin/admq rabbitmq admin diagnostics health_check

# Memory usage
bin/admq rabbitmq admin diagnostics memory_breakdown
```

4.3 User Management

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

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

# Change password
bin/admq rabbitmq admin ctl change_password username newpassword

# Delete user
bin/admq rabbitmq admin ctl delete_user username

# List users
bin/admq rabbitmq admin ctl list_users
```

4.4 Queue Management

```
# List queues
bin/admq rabbitmq admin ctl list_queues
bin/admq rabbitmq admin ctl list_queues name messages consumers

# List exchanges
bin/admq rabbitmq admin ctl list_exchanges name type

# List connections
bin/admq rabbitmq admin ctl list_connections peer_host state

# List channels
bin/admq rabbitmq admin ctl list_channels connection consumer_count
```

4.5 Cluster Management

```
# View cluster status
bin/admq rabbitmq admin ctl cluster_status

# View queue details (including mirror)
bin/admq rabbitmq admin ctl list_queues name pid slave_pids
```

```
synchronised_slave_pids

# Remove failed node
bin/admq rabbitmq admin ctl forget_cluster_node rabbit@node2
```

4.6 Policy Management

```
# List policies
bin/admq rabbitmq admin ctl list_policies

# Set mirror policy
bin/admq rabbitmq admin ctl set_policy ha-all "^" '{"ha-
mode":"all"}' --priority 0 --apply-to queues

# Clear policy
bin/admq rabbitmq admin ctl clear_policy ha-all
```

4.7 Backup and Restore

```
# Export configuration
bin/admq rabbitmq admin ctl export_definitions
/backup/definitions.json

# Import configuration
bin/admq rabbitmq admin ctl import_definitions
/backup/definitions.json

# Backup Mnesia
bin/admq rabbitmq admin ctl eval
'rabbit_mnesia:backup("/backup/mnesia.backup").'
```

5 5. Daily Operations

5.1 Log Locations

Log Type	Path
Service log	./log/rabbitmq/ rabbit@hostname.log
Startup log	./log/rabbitmq/ rabbit@hostname-sasl.log
Console log	./logs/

5.2 Daily Checklist

Check Item	Command	Normal Standard
Node status	<code>bin/admq rabbitmq admin ctl status</code>	Running normally
Cluster status	<code>bin/admq rabbitmq admin ctl cluster_status</code>	All nodes online
Queue backlog	<code>bin/admq rabbitmq admin ctl list_queues messages</code>	No abnormal backlog
Memory usage	<code>bin/admq rabbitmq admin ctl status</code>	Below watermark
Disk space	<code>df -h</code>	Above disk_free_limit

5.3 Health Check Script

```
#!/bin/bash
# Node health check
bin/admq rabbitmq admin diagnostics health_check

# Memory check
bin/admq rabbitmq admin diagnostics memory_breakdown

# Alert queue check (messages > 10000)
bin/admq rabbitmq admin ctl list_queues name messages consumers |
awk '$2 > 10000 {print "WARNING: Queue \"$1\" has \"$2\" messages"}'
```

```
# Connection count check
CONN_COUNT=$(bin/admq rabbitmq admin ctl list_connections | wc -l)
if [ $CONN_COUNT -gt 1000 ]; then
    echo "WARNING: Connection count $CONN_COUNT exceeds threshold"
fi
```

6 6. Configuration Optimization

6.1 rabbitmq.conf Core Configuration

```
# Network
listeners.tcp.default = 5672
listeners.ssl.default = 5671

# Memory (recommended 0.6-0.7)
vm_memory_high_watermark.relative = 0.6
vm_memory_high_watermark.paging_ratio = 0.5

# Disk (recommended absolute value 1GB)
disk_free_limit.relative = 1.5
disk_free_limit.absolute = 1GB

# Connections
channel_max = 2047
heartbeat = 30

# Queues
queue_master_locator = min-masters
```

6.2 System Optimization

```
# File descriptors
ulimit -n 65535

# Kernel parameters (/etc/sysctl.conf)
net.ipv4.ip_local_port_range = 1024 65535
net.ipv4.tcp_tw_reuse = 1
net.core.somaxconn = 65535
vm.swappiness = 10
```

```
# Disk scheduler  
echo deadline > /sys/block/sda/queue/scheduler
```

7 7. Troubleshooting

7.1 Service Won't Start

Symptom	Possible Cause	Solution
Port occupied	5672/15672 occupied	Stop the occupying process or change the port
Cookie mismatch	Cluster node Cookie mismatch	Synchronize .erlang.cookie
Insufficient memory	System memory insufficient	Increase memory or lower watermark
Insufficient disk	Below disk_free_limit	Clean up disk or adjust limit

7.2 Message Consumption Anomalies

Symptom	Possible Cause	Solution
Message backlog	Consumer too slow or offline	Add more consumers or optimize logic
Message loss	Persistence not enabled	Enable delivery_mode=2
Duplicate consumption	Acknowledgement timeout	Optimize processing logic, acknowledge promptly

7.3 Cluster Anomalies

Symptom	Possible Cause	Solution
Node offline	Network failure	Check network, restart service
Split-brain	Network partition	Use forget_cluster_node to clean up
Queue unavailable	Node offline	Use mirror queue or Quorum Queue

7.4 Diagnostic Commands

```
# View detailed information
bin/admq rabbitmq admin ctl status
bin/admq rabbitmq admin ctl cluster_status
```

```
bin/admq rabbitmq admin ctl list_connections peer_host peer_port  
state  
bin/admq rabbitmq admin ctl list_queues name messages consumers  
memory  
  
# View memory/processes  
bin/admq rabbitmq admin diagnostics memory_breakdown  
bin/admq rabbitmq admin diagnostics processes  
bin/admq rabbitmq admin diagnostics connections
```

8 8. Security Hardening

8.1 Change Default User

```
# Delete default guest user
bin/admq rabbitmq admin ctl delete_user guest

# Create new administrator
bin/admq rabbitmq admin ctl add_user admin strongpassword
bin/admq rabbitmq admin ctl set_user_tags admin administrator
bin/admq rabbitmq admin ctl set_permissions -p / admin ".*" ".*"
".*"
```

8.2 Firewall Configuration

```
firewall-cmd --permanent --add-port=5672/tcp
firewall-cmd --permanent --add-port=15672/tcp
firewall-cmd --permanent --add-port=25672/tcp
firewall-cmd --reload
```

8.3 SSL Configuration

```
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
```

9 9. Performance Optimization Tips

9.1 Queue Optimization

```
Map<String, Object> args = new HashMap<>();  
// Queue length limit  
args.put("x-max-length", 100000);  
args.put("x-max-length-bytes", 1073741824); // 1GB  
  
// Message TTL  
args.put("x-message-ttl", 3600000); // 1 hour  
  
// Lazy queue (reduce memory)  
args.put("x-queue-mode", "lazy");
```

9.2 Client Optimization

```
// Connection pool + automatic recovery  
factory.setAutomaticRecoveryEnabled(true);  
factory.setNetworkRecoveryInterval(5000);  
  
// Single connection with multiple channels  
Connection connection = factory.newConnection();  
for (int i = 0; i < 10; i++) {  
    Channel channel = connection.createChannel();  
}  
  
// QoS prefetch  
channel.basicQos(100);  
  
// Batch acknowledgment  
channel.basicAck(deliveryTag, true);
```

9.3 Monitoring Metrics

Metric	Description
rabbitmq_connections	Number of connections
rabbitmq_queue_messages	Total message count
rabbitmq_node_mem_used	Memory usage
rabbitmq_node_disk_free	Available disk space

10 10. Operations Best Practices

1. **Regular Checks:** Establish routine inspection mechanisms
2. **Monitoring and Alerts:** Configure a comprehensive monitoring and alerting system
3. **Backup Strategy:** Regularly back up configurations and data
4. **Capacity Planning:** Evaluate and scale up in advance
5. **Security Updates:** Apply security patches in a timely manner
6. **Least Privilege:** Assign minimum necessary permissions to business users
7. **Disaster Drills:** Conduct regular failure recovery drills

Technical Support:

- Website: www.apusic.com
- Phone: 400-855-5800
- Email: support@apusic.com

UNIFIED SERVICE HOTLINE

4008-555-800

Kingdee Apusic Cloud Computing Co., Ltd. (referred to as “Kingdee Apusic”) was founded in 2000. Its predecessor was “Kingdee Middleware Company.” It is a basic software platform provider under Kingdee Group, a national standards-setting enterprise for cloud computing, and a core software enterprise in the national IT application innovation industry. Kingdee Apusic is an enterprise undertaking the National 863 Key R&D Program and the National Major Science and Technology Project for Core Electronic Components, High-end General-purpose Chips and Fundamental Software, and is also a basic platform provider for the national key projects of “Two Networks, One Website, Four Databases, and Twelve Golden Projects.” Its products are widely used in key industries such as government, military, finance, and energy, and it has cumulatively served more than 100,000 customers.



National standard setter for cloud computing
Basic software enterprise under Kingdee Group
Core enterprise for IT application innovation
Website: www.apusic.com

