



APUSIC
固若长城
睿比世界

Operations Manual

Kingdee Apusic Distributed Message Queue
V2.0.6_for_mqtt

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 Preface
 - 1.1 Target Audience
 - 1.2 Related Documents
 - 1.3 Technical Support
- 2 Daily Operations
 - 2.1 Service Management
 - 2.1.1 Start Service
 - 2.1.2 Stop Service
 - 2.1.3 Check Service Status
 - 2.1.4 Restart Service
 - 2.2 Log Management
 - 2.2.1 Log Locations
 - 2.2.2 Log Level Configuration
 - 2.2.3 Log Rotation
 - 2.3 Monitoring Checks
 - 2.3.1 Daily Checklist
 - 2.3.2 Health Check Script
- 3 Configuration Management
 - 3.1 Configuration Files
 - 3.1.1 admq-mqtt.conf
 - 3.2 Dynamic Configuration
- 4 Troubleshooting
 - 4.1 Common Issues and Solutions
 - 4.1.1 Service Cannot Start
 - 4.1.2 Client Cannot Connect
 - 4.1.3 Message Delivery Anomalies
 - 4.1.4 Cluster Anomalies
 - 4.2 Troubleshooting Commands
- 5 Backup and Recovery
 - 5.1 Configuration Backup
 - 5.2 Data Backup
 - 5.3 Periodic Backup Script

- 6 Security Hardening
 - 6.1 Network Security
 - 6.1.1 Firewall Configuration
 - 6.1.2 SSL/TLS Configuration
 - 6.2 Access Control
 - 6.2.1 Default Deny Policy
 - 6.2.2 Superuser Management
 - 6.3 Audit Log
- 7 Performance Monitoring
 - 7.1 Prometheus Monitoring
 - 7.1.1 Enable Prometheus Metrics
 - 7.1.2 Monitoring Metrics
 - 7.2 Grafana Dashboard
 - 7.3 Alert Rules
- 8 Upgrade Maintenance
 - 8.1 Version Upgrade
 - 8.1.1 Pre-upgrade Preparation
 - 8.1.2 Upgrade Steps
 - 8.1.3 Cluster Rolling Upgrade
- 9 Operations Best Practices

1 Preface

This document is the operations manual for the Kingdee Apusic Distributed Message Queue for MQTT (Apusic Distributed Message Queue, ADMQ for MQTT), designed to provide middleware operations personnel with standardized operational guidelines covering core scenarios such as installation and deployment, configuration optimization, daily operations, troubleshooting, and security hardening, ensuring stable, efficient, and secure operation of ADMQ for MQTT services.

1.1 Target Audience

This document is intended for IT business leaders, R&D managers, software project managers, software architects, and operations engineers.

1.2 Related Documents

To learn more about ADMQ for MQTT, please refer to the following ADMQ for MQTT product documentation set:

No.	Document	Description
1	ADMQ for MQTT Quick Start Guide	Briefly introduces how to quickly get started with ADMQ for MQTT.
2	ADMQ for MQTT Installation Manual	Details how to install ADMQ for MQTT on various operating systems and how to start/stop ADMQ for MQTT services.
3	ADMQ for MQTT Message Engine User Manual	Details the usage, configuration, management, and supporting tools of ADMQ for MQTT message engine features.
4	ADMQ for MQTT Console User Manual	Details the usage and operation instructions of the ADMQ for MQTT console.
5	ADMQ for MQTT Development Manual	Details ADMQ for MQTT client application development based on various programming languages.
6	ADMQ for MQTT Migration Manual	Details the migration from MQTT Broker to ADMQ for MQTT.
7	ADMQ for MQTT Operations Manual	Details ADMQ for MQTT monitoring, operations, security hardening, and other operational tasks.
8	ADMQ for MQTT Performance Tuning Manual	Details ADMQ for MQTT performance tuning guidelines.

1.3 Technical Support

ADMQ for MQTT provides comprehensive technical support services. You can obtain technical support through the following channels:

- Website: www.apusic.com
- Phone: 400-855-5800
- Email: support@apusic.com
- Kingdee Cloud Community: <https://vip.kingdee.com/?productId=73&productLineId=14&lang=zh-CN>

When contacting technical support, please provide the following information:

1. Your name
2. Company information and contact details
3. Operating system and its version
4. Product version number
5. Detailed information such as logs and screenshots of the exception or error

2 Daily Operations

2.1 Service Management

2.1.1 Start Service

```
# Start MQTT service
bin/admq-service start

# Start application
bin/admq-mqtt start
```

2.1.2 Stop Service

```
# Stop application
bin/admq-mqtt stop

# Stop MQTT service
bin/admq-service stop
```

2.1.3 Check Service Status

```
# Check service status
bin/admq-service status

# Check node status
bin/admq-mqtt ctl status

# Check cluster status
bin/admq-mqtt ctl cluster status
```

2.1.4 Restart Service

```
# Restart service
bin/admq-service restart
```

2.2 Log Management

2.2.1 Log Locations

Log Type	Path	Description
Service log	log/admq-mqtt.log	MQTT Broker main log
Crash log	log/erlang.log.*	Erlang VM crash log
Console log	/opt/admq-manager-V3.0.2/logs/	Console runtime log

2.2.2 Log Level Configuration

```
# View current log level
bin/admq-mqtt ctl log handlers list

# Dynamically adjust log level
bin/admq-mqtt ctl log set-level debug
bin/admq-mqtt ctl log set-level info
bin/admq-mqtt ctl log set-level warning
bin/admq-mqtt ctl log set-level error
```

2.2.3 Log Rotation

```
# etc/admq-mqtt.conf
log.file {
    path = "log/admq-mqtt.log"
    rotation_size = "10MB"
    rotation_count = 5
    level = warning
}
```

2.3 Monitoring Checks

2.3.1 Daily Checklist

Check Item	Check Command	Normal Standard
Node status	admq-mqtt ctl cluster status	All nodes online
Connection count	admq-mqtt ctl stats	Within normal range
Memory usage	admq-mqtt ctl status	Below memory limit
Disk space	df -h	Above minimum requirement
Message throughput	Runtime monitoring	No abnormal fluctuations
Slow subscriptions	Slow subscription management	No large number of slow subscriptions

2.3.2 Health Check Script

```
#!/bin/bash

# Node health check
bin/admq-mqtt ctl status

# Connection count check
CONN_COUNT=$(bin/admq-mqtt ctl stats | grep connections.count | awk '{print $2}')
if [ "$CONN_COUNT" -gt 500000 ]; then
    echo "WARNING: Connection count $CONN_COUNT exceeds threshold"
fi

# Memory usage check
MEM_USAGE=$(bin/admq-mqtt ctl status | grep memory.used | awk '{print $2}')
echo "Memory usage: $MEM_USAGE"
```

3 Configuration Management

3.1 Configuration Files

3.1.1 admq-mqtt.conf

Main configuration file, located at `etc/admq-mqtt.conf` .

```
# Node configuration
node.name = admq@mqtt-node1
node.cookie = your-secret-cookie

# Cluster configuration
cluster.discovery = static
cluster.static.seeds = admq@mqtt-node1,admq@mqtt-node2,admq@mqtt-
node3

# Listener configuration
listeners.tcp.default {
    bind = "0.0.0.0:1883"
    max_connections = 1024000
}

listeners.ssl.default {
    bind = "0.0.0.0:8883"
    ssl_options {
        certfile = "etc/certs/server.crt"
        keyfile = "etc/certs/server.key"
        cacertfile = "etc/certs/ca.crt"
    }
}

# Authenticator configuration
authentication = [
{
    mechanism = password_based
```

```
    backend = built_in_database
    enable = true
  }
]

# Authorization source configuration
authorization {
  sources = [
    {
      type = built_in_database
      enable = true
    }
  ]
}
```

3.2 Dynamic Configuration

```
# View current configuration
bin/admq-mqtt ctl conf show

# Update configuration
bin/admq-mqtt ctl conf update "log.level=info"

# Reset configuration
bin/admq-mqtt ctl conf reset
```

4 Troubleshooting

4.1 Common Issues and Solutions

4.1.1 Service Cannot Start

Phenomenon	Possible Cause	Solution
Port occupied	Port 1883/8883/8083 is occupied	Find and stop the occupying process, or change the listener port
Erlang Cookie mismatch	Cluster node cookies do not match	Synchronize the .erlang.cookie file
Insufficient memory	System memory is insufficient	Increase memory or optimize configuration
Configuration error	admq-mqtt.conf format error	Check configuration file syntax

4.1.2 Client Cannot Connect

Phenomenon	Possible Cause	Solution
Authentication failed	Incorrect username/password	Check user configuration
ACL denied	Insufficient permissions	Check ACL rules
Listener not started	Listener not enabled	Check listener status
Network unreachable	Firewall or network issue	Check network connectivity

4.1.3 Message Delivery Anomalies

Phenomenon	Possible Cause	Solution
Subscriber not receiving messages	Topic mismatch	Check topic and wildcards
Duplicate messages	Normal QoS 1 behavior	Implement idempotent processing on the business side
Message loss	QoS 0 does not guarantee delivery	Use QoS 1/2 instead
Consumption delay	Slow subscription or network latency	Check slow subscriptions and network conditions

4.1.4 Cluster Anomalies

Phenomenon	Possible Cause	Solution
Node offline	Network failure or service exception	Check network connectivity, restart service
Split-brain	Network partition causing cluster split	Check network partition, rebuild cluster if necessary
Session out of sync	Network latency	Wait for automatic synchronization

4.2 Troubleshooting Commands

```
# View node status
bin/admq-mqtt ctl status

# View cluster status
bin/admq-mqtt ctl cluster status

# View online clients
bin/admq-mqtt ctl clients list

# View subscription list
bin/admq-mqtt ctl subscriptions list

# View retained messages
bin/admq-mqtt ctl retained list

# View alarms
bin/admq-mqtt ctl alarms list

# View logs
bin/admq-mqtt ctl log handlers list
```

5 Backup and Recovery

5.1 Configuration Backup

```
# Export current configuration
bin/admq-mqtt ctl conf show > admq-mqtt-config-backup.json

# Backup configuration file
cp etc/admq-mqtt.conf backup/admq-mqtt.conf.$(date +%Y%m%d)
```

5.2 Data Backup

```
# Create data backup
bin/admq-mqtt ctl data backup

# View backup list
bin/admq-mqtt ctl data backup list
```

5.3 Periodic Backup Script

```
#!/bin/bash

BACKUP_DIR="/backup/mqtt/$(date +%Y%m%d) "
mkdir -p $BACKUP_DIR

# Backup configuration
bin/admq-mqtt ctl conf show > $BACKUP_DIR/config.json
cp etc/admq-mqtt.conf $BACKUP_DIR/

# Create data backup
bin/admq-mqtt ctl data backup $BACKUP_DIR/data-backup
```

```
# Compress backup
tar czf $BACKUP_DIR.tar.gz $BACKUP_DIR
rm -rf $BACKUP_DIR

# Keep backups from the last 7 days
find /backup/mqtt -name "*.tar.gz" -mtime +7 -delete
```

6 Security Hardening

6.1 Network Security

6.1.1 Firewall Configuration

```
# Open necessary ports
firewall-cmd --permanent --add-port=1883/tcp      # MQTT TCP
firewall-cmd --permanent --add-port=8883/tcp      # MQTT over SSL
firewall-cmd --permanent --add-port=8083/tcp      # MQTT over
WebSocket
firewall-cmd --permanent --add-port=8084/tcp      # MQTT over WSS
firewall-cmd --permanent --add-port=14567/tcp     # MQTT over QUIC
firewall-cmd --permanent --add-port=18083/tcp     # Dashboard
firewall-cmd --permanent --add-port=12305/tcp     # Console HTTP
firewall-cmd --permanent --add-port=12306/tcp     # Console HTTPS
firewall-cmd --permanent --add-port=4370/tcp      # Erlang
Distribution
firewall-cmd --permanent --add-port=5370/tcp      # Erlang
Distribution SSL
firewall-cmd --reload
```

6.1.2 SSL/TLS Configuration

```
# etc/admq-mqtt.conf
listeners.ssl.default {
    bind = "0.0.0.0:8883"
    ssl_options {
        certfile = "etc/certs/server.crt"
        keyfile = "etc/certs/server.key"
        cacertfile = "etc/certs/ca.crt"
        verify = verify_peer
        fail_if_no_peer_cert = false
        versions = ["tlsv1.2", "tlsv1.3"]
    }
}
```

```
}
}
```

6.2 Access Control

6.2.1 Default Deny Policy

```
# etc/admq-mqtt.conf
authorization {
    no_match = deny
    deny_action = ignore
    sources = [
        {
            type = built_in_database
            enable = true
        }
    ]
}
```

6.2.2 Superuser Management

```
# Create superuser
bin/admq-mqtt ctl admins add admin password

# Use superuser privileges with caution
```

6.3 Audit Log

```
# etc/admq-mqtt.conf
log.audit {
    enable = true
    path = "log/audit.log"
    rotation_size = "10MB"
```

```
rotation_count = 10  
}
```

7 Performance Monitoring

7.1 Prometheus Monitoring

7.1.1 Enable Prometheus Metrics

```
# etc/admq-mqtt.conf

prometheus {
    enable = true
    bind = "0.0.0.0:18083"
    push_gateway_server = "http://localhost:9091"
    interval = "15s"
}
```

7.1.2 Monitoring Metrics

Metric	Description
emqx_connections_count	Current number of connections
emqx_sessions_count	Current number of sessions
emqx_topics_count	Current number of topics
emqx_subscriptions_count	Current number of subscriptions
emqx_messages_received	Number of messages received
emqx_messages_sent	Number of messages sent
emqx_messages_dropped	Number of messages dropped
emqx_bytes_received	Number of bytes received
emqx_bytes_sent	Number of bytes sent

7.2 Grafana Dashboard

Import the official EMQX Grafana dashboard to monitor cluster status.

7.3 Alert Rules

```
# Prometheus alert rules example
groups:
- name: mqtt_alerts
  rules:
    - alert: MQTTNodeDown
      expr: up{job="admq-mqtt"} == 0
      for: 1m
      labels:
        severity: critical
      annotations:
        summary: "MQTT node is down"

    - alert: MQTTHighConnections
      expr: emqx_connections_count > 800000
      for: 5m
      labels:
        severity: warning
      annotations:
        summary: "MQTT connection count is high"

    - alert: MQTTMessagesDropped
      expr: rate(emqx_messages_dropped[5m]) > 100
      for: 5m
      labels:
        severity: warning
      annotations:
        summary: "MQTT messages are being dropped"
```

8 Upgrade Maintenance

8.1 Version Upgrade

8.1.1 Pre-upgrade Preparation

1. Backup configuration files and data
2. Review the version release notes
3. Verify the upgrade process in a test environment

8.1.2 Upgrade Steps

```
# 1. Stop the service
bin/admq-service stop

# 2. Backup data
cp -r data /backup/mqtt-data-$(date +%Y%m%d)

# 3. Upgrade software package
# Download the new version package and extract it
tar zxvf admq-mqtt-V2.0.7.tar.gz -C /opt/

# 4. Migrate configuration
cp etc/admq-mqtt.conf /opt/admq-mqtt-V2.0.7/etc/

# 5. Start the service
cd /opt/admq-mqtt-V2.0.7
bin/admq-service start

# 6. Verify the upgrade
bin/admq-mqtt ctl status
```

8.1.3 Cluster Rolling Upgrade

1. Upgrade one node at a time
2. Confirm the node is functioning normally before upgrading the next
3. Ensure the cluster always has available nodes

9 Operations Best Practices

1. **Regular checks:** Establish a daily inspection mechanism to identify potential issues early
2. **Monitoring and alerts:** Configure a comprehensive monitoring and alerting system
3. **Backup strategy:** Develop a regular backup strategy to ensure data security
4. **Documentation:** Record all operations and changes
5. **Capacity planning:** Regularly evaluate cluster capacity and scale up in advance
6. **Security updates:** Apply security patches promptly
7. **Disaster recovery drills:** Conduct regular failover recovery drills

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

