



Operations Manual

金蝶Apusic分布式消息队列 V2.0.6_for_kafka

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 Overview
 - 2.1 Basic Concepts
- 3 Common Operations Commands
 - 3.1 Process Management
 - 3.1.1 Start/Stop Kafka ZK Node
 - 3.1.2 Start/Stop Kafka Broker Node
 - 3.2 Topic Management
 - 3.2.1 List Topics
 - 3.2.2 View Topic Details
 - 3.2.3 Create Topic
 - 3.2.4 Delete Topic
 - 3.3 Consumer Management
 - 3.3.1 List Consumer Groups
 - 3.3.2 View Consumption Status
 - 3.4 Produce/Consume
 - 3.4.1 Produce Messages
 - 3.4.2 Consume Messages
- 4 Routine Inspection
 - 4.1 Broker Status
 - 4.2 ISR Status
 - 4.3 Consumer Lag
 - 4.4 Disk Usage
- 5 Common Troubleshooting
 - 5.1 Consumer Backlog
 - 5.2 Broker Down
 - 5.3 ISR Not In Sync
 - 5.4 Topic Has No Leader
 - 5.5 Other Issues

- 6 Kafka Scaling and Maintenance
 - 6.1 Adding Partitions
 - 6.2 Partition Reassignment
- 7 Kafka Performance Tuning
 - 7.1 Broker Parameters
 - 7.2 Producer Optimization
 - 7.3 Consumer Optimization
- 8 Log Management
- 9 Configuration Management
 - 9.1 Default Configuration
 - 9.2 Core Configuration Items
- 10 Security Hardening
 - 10.1 1. Network Security
 - 10.1.1 1.1 Firewall Policy
 - 10.1.2 1.2 Network Isolation
 - 10.2 2. Authentication and Authorization
 - 10.2.1 2.1 Enable SASL Authentication
 - 10.2.2 2.2 Configure User Authentication
 - 10.2.3 2.3 Configure ACL Rules
 - 10.3 3. Data Encryption
 - 10.3.1 3.1 SSL/TLS Configuration
 - 10.3.2 3.2 Certificate Management
 - 10.4 4. Auditing and Monitoring
 - 10.4.1 4.1 Enable Audit Log
 - 10.4.2 4.2 Monitoring and Alert Items
 - 10.5 5. System Security
 - 10.5.1 5.1 Running User
 - 10.5.2 5.2 systemd Security Options
 - 10.6 6. Security Inspection Checklist
- 11 Operations Experience Summary

1 Preface

This document is the operations guide for ADMQ for Kafka (Apusic Distributed Message Queue, referred to as ADMQ for Kafka). It aims to provide middleware operations personnel with standardized operational guidelines for ADMQ for Kafka, covering core scenarios such as installation and deployment, configuration optimization, routine operations, troubleshooting, and security hardening, ensuring stable, efficient, and secure operation of the ADMQ for Kafka service.

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

For more information about the ADMQ for Kafka product, please refer to the following ADMQ for Kafka product documentation set:

No.	Document	Description
1	ADMQ for Kafka Quick Start Guide	A brief introduction on how to quickly get started with ADMQ for Kafka.
2	ADMQ for Kafka Installation Guide	Detailed instructions on installing ADMQ for Kafka on various operating systems, as well as starting and stopping ADMQ for Kafka services.
3	ADMQ for Kafka Message Engine User Guide	Detailed description of the usage, configuration, and management of ADMQ for Kafka message engine features and related tools.
4	ADMQ for Kafka Console User Guide	Detailed description of the usage and operations of the ADMQ for Kafka management console.
5	ADMQ for Kafka Developer Guide	Detailed instructions on developing ADMQ for Kafka client applications using various programming languages.
6	ADMQ for Kafka Migration Guide	Detailed instructions on migrating from Kafka to ADMQ for Kafka.
7	ADMQ for Kafka Operations Guide	Detailed description of monitoring, operations, security hardening, and other operational aspects of ADMQ for Kafka.

8	ADMQ for Kafka Performance Tuning Guide	Detailed instructions on performance tuning for ADMQ for Kafka.
---	--	---

1.3 Technical Support

The ADMQ for Kafka product 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 version
4. Product version number
5. Detailed information such as logs and screenshots of exceptions or errors

2 Overview

ADMQ Kafka is a high-throughput distributed messaging system commonly used for log collection, data pipelines, and real-time stream processing.

2.1 Basic Concepts

Component	Description
Broker	Kafka service node
Topic	Message category
Partition	Partition of a Topic
Producer	Message producer
Consumer	Message consumer
Consumer Group	Consumer group

Simple data flow: Producer → Broker → Topic → Partition → Consumer Features:

- High throughput
- Distributed
- Sequential disk writes
- Replication mechanism ensures reliability

3 Common Operations Commands

Process control commands are located in the admq-kafka/bin directory, and the required configuration files are in the admq-kafka/config directory. Open-source Kafka scripts are supported; you can enter the admq-kafka/kafka/bin directory to execute the corresponding scripts.

3.1 Process Management

3.1.1 Start/Stop Kafka ZK Node

```
bin/admq-daemon start kafka zk
```

```
bin/admq-daemon stop kafka zk
```

3.1.2 Start/Stop Kafka Broker Node

```
bin/admq-daemon start kafka broker 1
```

```
bin/admq-daemon stop kafka broker 2
```

3.2 Topic Management

3.2.1 List Topics

```
kafka-topics.sh --bootstrap-server localhost:9092 --list
```

3.2.2 View Topic Details

```
kafka-topics.sh --bootstrap-server localhost:9092 --describe --topic  
test
```

3.2.3 Create Topic

```
kafka-topics.sh \  
--bootstrap-server localhost:9092 \  
--create \  
--topic test \  
--partitions 3
```

3.2.4 Delete Topic

```
kafka-topics.sh \  
--bootstrap-server localhost:9092 \  
--delete \  
--topic test
```

3.3 Consumer Management

3.3.1 List Consumer Groups

```
kafka-consumer-groups.sh \  
--bootstrap-server localhost:9092 \  
--list
```

3.3.2 View Consumption Status

```
kafka-consumer-groups.sh \  
--bootstrap-server localhost:9092 \  
--describe \  
--group test-group
```

Key metrics:

Metric	Meaning
CURRENT-OFFSET	Current consumption offset
LOG-END-OFFSET	Latest message offset

LAG	Backlog
-----	---------

3.4 Produce/Consume

3.4.1 Produce Messages

```
kafka-console-producer.sh \  
--broker-list localhost:9092 \  
--topic test
```

3.4.2 Consume Messages

```
kafka-console-consumer.sh \  
--bootstrap-server localhost:9092 \  
--topic test \  
--from-beginning
```

4 Routine Inspection

4.1 Broker Status

Check if the broker is online:

```
ps -ef | grep admq-kafka
```

Or

```
netstat -ntlp | grep 9092
```

4.2 ISR Status

Check:

```
kafka-topics.sh --describe
```

Judgment:

```
ISR < Replicas
```

Indicates abnormal replica synchronization.

4.3 Consumer Lag

Check:

```
kafka-consumer-groups.sh --describe
```

Judgment:

```
LAG > 0
```

If it continues to grow, investigation is needed.

4.4 Disk Usage

Check:

```
df -h
```

Recommendations:

Metric	Recommendation
Disk usage	<70%
IO util	<80%

5 Common Troubleshooting

5.1 Consumer Backlog

Symptom:

LAG continues to grow

Causes:

- Insufficient consumer capacity
- Broker performance bottleneck
- Network issues

Solutions:

Method	Description
Add more Consumers	Increase parallelism
Add more Partitions	Improve throughput
Optimize business logic	Reduce processing time

5.2 Broker Down

Symptom:

```
Leader not available
```

Check:

```
ps -ef |grep admq-kafka
```

View logs:

```
cat logs/server.log
```

Resolution:

```
bin/admq-daemon start kafka broker/standalone
```

5.3 ISR Not In Sync

Symptom:

```
ISR < Replicas
```

Causes:

Cause	Description
High disk IO	Follower sync is slow
Network latency	Replica replication is slow
High CPU load	Broker under pressure

Check:

```
iostat -x
```

5.4 Topic Has No Leader

Symptom:

```
Leader: none
```

Resolution:

```
# Execute leader election
kafka-leader-election.sh \
--bootstrap-server localhost:9092 \
--all-topic-partitions \
--election-type preferred
```

5.5 Other Issues

- ADCC for zk startup fails, log reports **Connection refused** or **Session expired**
 - Check if the `myid` file exists and its content is correct.
 - Check if the IP addresses in `server.x` within `zookeeper.properties` are reachable, and if ports 2888/3888 are blocked by the firewall.
- Kafka startup fails, log reports **Broker ID 0 is already registered**
 - Check for residual old ADCC for zk data. For a new deployment, clear the ADCC for zk `dataDir` and Kafka `log.dirs` and retry.
 - Verify that `broker.id` across nodes is unique.
- ISR set incomplete (Isr: 0,1 instead of 0,1,2)
 - Indicates that a Follower is syncing too slowly or the network is unreachable. Check the node's logs and network bandwidth.
- Kafka startup fails, log reports **Failed to bind to port 9092**
 - Check if port 9092 is occupied by another process (use `netstat -tulpn | grep 9092` or `lsof -i:9092`).
 - Check if the port configured in `listeners` in `server.properties` matches the intended port to avoid configuration conflicts.
 - Confirm that the firewall is not blocking port 9092, or that the port has been allowed in the firewall rules.
- Kafka producer fails to send messages, reports **Leader not available**
 - Check if the corresponding Topic exists; create it if it does not (using the `kafka-topics.sh --create` command).
 - Check if the Kafka cluster is running normally and all Broker nodes are online.
 - Check if the Topic's replica configuration is reasonable; if the replication factor exceeds the number of available Brokers, leader election cannot proceed.
- Kafka consumer fails to consume, reports **No offset found for partition**
 - For a new consumer group, having no offset on first consumption is normal. Configure `auto.offset.reset` to `earliest` or `latest`.

- For an existing consumer group, check if offset data in ADCC for zk / Kafka has been manually deleted, or if the consumer group has been inactive for so long that offsets have expired (adjust the `offsets.retention.minutes` configuration).
- After ADCC for zk cluster starts, some nodes cannot join the cluster, log reports `Cannot open channel to x at election address`
 - Check that the `server.x` IP/port configuration in `zoo.cfg` is consistent across all ADCC for zk nodes with no typographical errors.
 - Check that the time is synchronized between nodes (a time difference exceeding 2000ms can cause election failure). Synchronize time using `ntpdate`.
 - Ensure the `dataDir` directory has permissions allowing the ADCC for zk running user to read and write, preventing permission issues from reading the `myid` file.
- Kafka log reports `Not enough replicas`, messages cannot be persisted
 - Check if the Topic's `min.insync.replicas` configuration value is greater than the current number of available in-sync replicas (ISR).
 - For a single Broker deployment, set `min.insync.replicas` to 1 (default is 1; adjust if manually changed).
 - Check if the Follower nodes are synchronizing normally, and investigate issues such as network latency and high disk IO causing slow synchronization.
- After Kafka starts, JVM runs out of memory (OOM), process crashes
 - Check the JVM configuration (`KAFKA_HEAP_OPTS`) in `kafka-server-start.sh`. The default heap memory may be too large or too small.
 - For a single-machine test environment, adjust to `-Xmx512M -Xms512M`. For production, configure based on server memory (recommended not to exceed 50% of physical memory).
 - Check if too many Topics/Partitions are enabled, causing high memory usage. Plan the number of partitions reasonably.
- `SSL handshake failed` when connecting to Kafka
 - If SSL authentication is enabled, check that the SSL certificate path and password configuration are correct on both the client and server sides.
 - Confirm that the certificate has not expired and that the server's `listeners` is configured as `SSL://` instead of `PLAINTEXT://`.
 - Check that the client has correctly configured SSL-related parameters (e.g., `security.protocol=SSL`).

6 Kafka Scaling and Maintenance

6.1 Adding Partitions

```
kafka-topics.sh \  
--bootstrap-server localhost:9092 \  
--alter \  
--topic test \  
--partitions 6
```

Note: After scaling, message ordering may change.

6.2 Partition Reassignment

Generate plan:

```
kafka-reassign-partitions.sh \  
--bootstrap-server localhost:9092 \  
--generate
```

Execute:

```
kafka-reassign-partitions.sh \  
--execute
```

7 Kafka Performance Tuning

7.1 Broker Parameters

Key parameters:

```
num.network.threads=8
num.io.threads=16
log.segment.bytes=1GB
log.retention.hours=168
```

7.2 Producer Optimization

Recommended:

```
acks=all
compression.type=snappy
batch.size=32768
linger.ms=10
```

7.3 Consumer Optimization

```
fetch.min.bytes=1
fetch.max.wait.ms=500
max.poll.records=500
```

8 Log Management

Log path: admq-kafka/logs

Common errors:

```
Leader not available  
NotEnoughReplicas  
Connection refused
```

9 Configuration Management

9.1 Default Configuration

Path: admq-kafka/configs

- `kafka-zk.conf`: Configuration file used by ZK
- `kafka-broker.conf`: Configuration used by broker nodes in cluster deployment
- `kafka-standalone.conf`: Configuration used by broker node in standalone deployment

9.2 Core Configuration Items

```
listeners=PLAINTEXT://192.168.1.10:9092
# For external access, add: EXTERNAL://your.public.ip:9093
advertised.listeners=PLAINTEXT://192.168.1.10:9092
# Security protocol mapping (e.g., when using multiple listeners)
listener.security.protocol.map=PLAINTEXT:PLAINTEXT
# Log retention time (7 days)
log.retention.hours=168
# Or retain by size (e.g., max 10GB per partition)
# log.retention.bytes=10737418240
# Log segment file size (default 1GB)
log.segment.bytes=1073741824
# Background log flush strategy (rely on OS flush for better
efficiency; do not actively flush for HDD)
log.flush.interval.messages=9223372036854775807
log.flush.interval.ms=9223372036854775807
```

10 Security Hardening

10.1 1. Network Security

10.1.1 1.1 Firewall Policy

In production environments, strictly restrict access to Kafka ports:

```
# Allow only internal network access to Kafka port
iptables -A INPUT -p tcp --dport 9092 -s 10.0.0.0/8 -j ACCEPT
iptables -A INPUT -p tcp --dport 9092 -j DROP

# Allow only console management IP access
iptables -A INPUT -p tcp --dport 12305 -s 10.0.0.10 -j ACCEPT
iptables -A INPUT -p tcp --dport 12305 -j DROP
```

10.1.2 1.2 Network Isolation

- Deploy the Kafka cluster in an independent security group/VPC
- Prohibit direct exposure of Kafka ports to the internet
- Use VPN or dedicated lines for cross-datacenter access

10.2 2. Authentication and Authorization

10.2.1 2.1 Enable SASL Authentication

```
# Broker configuration
listeners=SASL_PLAINTEXT://0.0.0.0:9092
security.inter.broker.protocol=SASL_PLAINTEXT
sasl.mechanism.inter.broker.protocol=SCRAM-SHA-512
sasl.enabled.mechanisms=SCRAM-SHA-512

# Enable ACLs
authorizer.class.name=kafka.security.authorizer.AclAuthorizer
allow.everyone.if.no.acl.found=false
```

10.2.2 2.2 Configure User Authentication

```
# Create user and set password
kafka-configs.sh --zookeeper localhost:2181 \
  --alter --add-config 'SCRAM-SHA-512=[password=your-secure-
password]' \
  --entity-type users --entity-name admin

# List users
kafka-configs.sh --zookeeper localhost:2181 \
  --describe --entity-type users
```

10.2.3 2.3 Configure ACL Rules

```
# Production environment access control
kafka-acls.sh --authorizer-properties
zookeeper.connect=localhost:2181 \
  --add --allow-principal User:app-producer \
  --operation Write --topic order-topic

kafka-acls.sh --authorizer-properties
zookeeper.connect=localhost:2181 \
  --add --allow-principal User:app-consumer \
  --operation Read --topic order-topic \
  --group order-consumer-group

# Deny anonymous access
kafka-acls.sh --authorizer-properties
zookeeper.connect=localhost:2181 \
  --add --deny-principal User:ANONYMOUS \
  --operation All --cluster
```

10.3 3. Data Encryption

10.3.1 3.1 SSL/TLS Configuration

```
# Enable SSL
listeners=SSL://0.0.0.0:9093
security.inter.broker.protocol=SSL

# SSL certificate configuration
ssl.keystore.location=/path/to/server.keystore.jks
ssl.keystore.password=your-keystore-password
ssl.key.password=your-key-password
ssl.truststore.location=/path/to/server.truststore.jks
ssl.truststore.password=your-truststore-password
ssl.client.auth=required
```

10.3.2 3.2 Certificate Management

```
# Generate self-signed certificate (test environment)
keytool -keystore server.keystore.jks -alias localhost -validity 365
-genkey

# Export certificate
keytool -keystore server.keystore.jks -alias localhost -export -file
cert.crt

# Import truststore
keytool -keystore server.truststore.jks -alias ca-cert -import -file
ca-cert.crt
```

10.4 4. Auditing and Monitoring

10.4.1 4.1 Enable Audit Log

```
# log4j configuration
log4j.logger.kafka.authorizer.logger=WARN, authorizerAppender
```

```
log4j.additivity.kafka.authorizer.logger=false
log4j.appender.authorizerAppender=org.apache.log4j.DailyRollingFileAppender
log4j.appender.authorizerAppender.File=${log.dir}/kafka-
authorizer.log
log4j.appender.authorizerAppender.layout=org.apache.log4j.PatternLayout
log4j.appender.authorizerAppender.layout.ConversionPattern=[%d] %p
%m (%c)%n
```

10.4.2 4.2 Monitoring and Alert Items

Monitoring Item	Alert Threshold	Description
Authentication failures	> 10 times/min	Possible brute-force attack
Unauthorized access	> 0	Investigate immediately
SSL certificate expiration	< 30 days	Renew in advance
ACL changes	Any change	Requires audit confirmation

10.5 5. System Security

10.5.1 5.1 Running User

```
# Create dedicated user
useradd -r -s /bin/false admq-kafka

# Change file owner
chown -R admq-kafka:admq-kafka /apusic/admq-kafka

# Restrict directory permissions
chmod 750 /apusic/admq-kafka
chmod 600 /apusic/admq-kafka/config/*.conf
chmod 600 /apusic/admq-kafka/license.xml
```

10.5.2 5.2 systemd Security Options

```
[Service]
User=adm-q-kafka
Group=adm-q-kafka
NoNewPrivileges=true
ProtectSystem=strict
ProtectHome=true
PrivateTmp=true
```

10.6 6. Security Inspection Checklist

Check Item	Check Method	Pass Criteria
Firewall rules	<code>iptables -L</code>	Only necessary ports open
Authentication status	View configuration	SASL/SCRAM enabled
ACL configuration	<code>kafka-acls.sh --list</code>	No anonymous access permissions
SSL certificate	<code>keytool -list</code>	Certificate not expired
File permissions	<code>ls -la</code>	Config files 600
Running user	<code>ps -ef</code>	Non-root user
Audit log	<code>ls -la logs/</code>	Authorizer log exists

11 Operations Experience Summary

Kafka failures are mostly concentrated in:

Issue	Percentage
Consumer backlog	50%
Broker down	20%
ISR not in sync	15%
Disk full	10%
Network issues	5%

Operations mantra: First check Broker, then check Leader, then check ISR, then check Lag, finally check IO.

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

