



Migration 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 Compatible Kafka Versions
 - 2.2 Migration Strategies
- 3 Pre-migration Preparation
 - 3.1 1. Environment Assessment
 - 3.2 2. Data Volume Assessment
 - 3.3 3. Install ADMQ
- 4 Configuration Migration
 - 4.1 Broker Configuration Migration
 - 4.2 Configuration Mapping Example
 - 4.3 JAAS Configuration Migration
- 5 Metadata Migration
 - 5.1 Topic Migration
 - 5.1.1 Method 1: Using kafka-reassign-partitions (Recommended)
 - 5.1.2 Method 2: Export and Import Topic Configuration
 - 5.2 ACL Migration
 - 5.3 Consumer Group Offset Migration
- 6 Data Migration
 - 6.1 Using MirrorMaker 2
 - 6.1.1 Configuring MirrorMaker 2
 - 6.1.2 Starting MirrorMaker 2
 - 6.2 Using Kafka Connect Migration
 - 6.3 Data Verification
- 7 Client Migration
 - 7.1 Switching Client Connection Address
 - 7.2 No Code Changes Required
 - 7.3 Verifying Client Migration
- 8 Migration Steps

- 8.1 Step 1: Deploy ADMQ Cluster
- 8.2 Step 2: Verify Cluster
- 8.3 Step 3: Configuration Migration
- 8.4 Step 4: Metadata Migration
- 8.5 Step 5: Data Migration
- 8.6 Step 6: Client Switchover
- 8.7 Step 7: Verification
- 8.8 Step 8: Cleanup
- 9 Migration Risk Control
 - 9.1 Rollback Plan
 - 9.2 Migration Window
 - 9.3 Precautions
- 10 Frequently Asked Questions
 - 10.1 Topic Creation Failed
 - 10.2 Data Synchronization Latency
 - 10.3 Consumption Progress Lost
 - 10.4 ACL Not Effective

1 Preface

This document is the migration guide for ADMQ for Kafka (Apusic Distributed Message Queue, referred to as ADMQ for Kafka), detailing how to migrate from Kafka to ADMQ for Kafka.

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

This document details the best practices for migrating an existing Kafka cluster to ADMQ for Kafka V2.0.6, including compatible versions, configuration migration, metadata migration, and data migration.

2.1 Compatible Kafka Versions

ADMQ for Kafka V2.0.6 is compatible with the following Kafka versions:

Source Kafka Version	Compatibility	Description
2.4.x	Fully Compatible	Recommended for migration
2.5.x	Fully Compatible	Recommended for migration
2.6.x	Fully Compatible	Recommended for migration
2.7.x	Fully Compatible	Recommended for migration
2.8.x	Fully Compatible	Recommended for migration
3.0.x	Fully Compatible	Recommended for migration
3.1.x	Fully Compatible	Recommended for migration
3.2.x	Fully Compatible	Recommended for migration
3.3.x	Fully Compatible	Recommended for migration
3.4.x	Fully Compatible	Recommended for migration
3.5.x	Fully Compatible	Recommended for migration
3.6.x	Fully Compatible	Recommended for migration
3.7.x	Fully Compatible	Recommended for migration
3.8.x	Fully Compatible	Recommended for migration

2.2 Migration Strategies

Strategy	Use Case	Advantages	Disadvantages
Rolling Upgrade	Minor version upgrade within the same version	No downtime	Not suitable for cross-major-version upgrades

Dual Cluster Migration	Heterogeneous cluster migration	Low migration risk	Requires double resources
Progressive Migration	Production environment migration	Gradual verification	Long migration cycle

3 Pre-migration Preparation

3.1 1. Environment Assessment

```
# Check source cluster status
kafka/bin/kafka-broker-api-versions.sh --bootstrap-server <source-kafka>:9092

# View cluster information
kafka/bin/kafka-topics.sh --list --bootstrap-server <source-kafka>:9092
kafka/bin/kafka-consumer-groups.sh --list --bootstrap-server <source-kafka>:9092
```

3.2 2. Data Volume Assessment

```
# Check Topic count
kafka/bin/kafka-topics.sh --list --bootstrap-server <source-kafka>:9092 | wc -l

# Check total message volume
kafka/bin/kafka-run-class.sh kafka.admin.ConsumerGroupCommand \
  --bootstrap-server <source-kafka>:9092 \
  --list
```

3.3 3. Install ADMQ

Deploy the ADMQ for Kafka cluster according to the installation guide, ensuring:

- Network connectivity with the source cluster
- Configuration compatibility with the source cluster parameters

4 Configuration Migration

4.1 Broker Configuration Migration

Key configurations to migrate:

Configuration Item	Description	Migration Suggestion
<code>broker.id</code>	Broker ID	Keep consistent or reassign
<code>listeners</code>	Listener addresses	Adjust IP according to the new environment
<code>log.dirs</code>	Data directories	Reassign directories
<code>zookeeper.connect</code>	ADCC for zk address	Point to the new ADCC for zk cluster
<code>advertised.listeners</code>	Advertised addresses	Configure with the new cluster address
<code>compression.type</code>	Compression type	Keep original configuration
<code>log.retention.*</code>	Log retention	Adjust as needed
<code>num.partitions</code>	Default partition count	Keep consistent

4.2 Configuration Mapping Example

```
# Source cluster configuration (server.properties)
broker.id=0
listeners=PLAINTEXT://192.168.1.10:9092
log.dirs=/data/kafka
zookeeper.connect=zk1:2181,zk2:2181,zk3:2181/kafka

# ADMQ configuration (kafka-broker.conf)
broker.id=0
listeners=PLAINTEXT://192.168.1.20:9092
log.dirs=/apusic/admq-kafka/data
zookeeper.connect=192.168.1.20:2181,192.168.1.21:2181,192.168.1.22:2181
```

4.3 JAAS Configuration Migration

If using SASL authentication:

```
# Migrate JAAS configuration (passwords below are examples only; use
strong passwords in production)
KafkaServer {
    org.apache.kafka.common.security.plain.PlainLoginModule required
    username="admin"
    password="admin-secret"
    user_admin="admin-secret"
    user_producer="producer-secret"
    user_consumer="consumer-secret";
};
```

5 Metadata Migration

5.1 Topic Migration

5.1.1 Method 1: Using kafka-reassign-partitions (Recommended)

```
# 1. Generate migration plan
kafka/bin/kafka-reassign-partitions.sh \
  --bootstrap-server <source-kafka>:9092 \
  --topics-to-move-json-file topics.json \
  --broker-list "0,1,2" \
  --generate

# topics.json format
{
  "topics": [{"topic": "topic1"}, {"topic": "topic2"}]
}

# 2. Execute migration
kafka/bin/kafka-reassign-partitions.sh \
  --bootstrap-server <source-kafka>:9092 \
  --reassignment-json-file reassign.json \
  --execute

# 3. Verify migration progress
kafka/bin/kafka-reassign-partitions.sh \
  --bootstrap-server <source-kafka>:9092 \
  --reassignment-json-file reassign.json \
  --verify
```

5.1.2 Method 2: Export and Import Topic Configuration

```
# Export Topic configuration
kafka/bin/kafka-configs.sh --bootstrap-server <source-kafka>:9092 \
```

```
--entity-type topics \
--entity-name my-topic \
--describe > topic-config-backup.txt

# Create the same Topic in ADMQ
kafka/bin/kafka-topics.sh --create \
  --bootstrap-server <admq-kafka>:9092 \
  --topic my-topic \
  --partitions 6 \
  --replication-factor 3
```

5.2 ACL Migration

```
# Export ACLs
kafka/bin/kafka-acls.sh --bootstrap-server <source-kafka>:9092 \
  --list > acls-backup.txt

# Create the same ACLs in ADMQ
kafka/bin/kafka-acls.sh --bootstrap-server <admq-kafka>:9092 \
  --add --allow-principal User:producer \
  --operation Write --topic my-topic
```

5.3 Consumer Group Offset Migration

```
# Export consumer group offsets
kafka/bin/kafka-consumer-groups.sh \
  --bootstrap-server <source-kafka>:9092 \
  --group my-group \
  --describe > offset-backup.txt

# Reset offsets (for scenarios where consumption progress needs to
be preserved)
kafka/bin/kafka-consumer-groups.sh \
  --bootstrap-server <admq-kafka>:9092 \
```

```
--group my-group \  
--topic my-topic \  
--reset-offsets --to-offset <offset-number> \  
--execute
```

6 Data Migration

6.1 Using MirrorMaker 2

MirrorMaker 2 is the officially recommended data migration tool for Kafka, supporting:

- Topic data synchronization
- ACL synchronization
- Configuration synchronization
- Consumer group offset synchronization

6.1.1 Configuring MirrorMaker 2

```
# mm2.properties
clusters=source,target
source.bootstrap.servers=<source-kafka>:9092
target.bootstrap.servers=<admq-kafka>:9092

source->target.enabled=true
source->target.topics.regex=.*

# Internal Topic configuration
source->target.emit.syncs.interval.seconds=60
source->target.sync.group.offsets.enabled=true
source->target.sync.group.offsets.interval.seconds=60

# Namespace configuration
source-
>target.rename=sourcePrefixTransformerClass=org.apache.kafka.connect.t
```

6.1.2 Starting MirrorMaker 2

```
# Using connect-mirror-maker.sh
kafka/bin/connect-mirror-maker.sh mm2.properties &
```

```
# Or using distributed mode
kafka/bin/connect-distributed.sh connect-distributed.properties
```

6.2 Using Kafka Connect Migration

```
// source-connector.json
{
  "name": "source-connector",
  "config": {
    "connector.class":
"org.apache.kafka.connect.mirror.MirrorSourceConnector",
    "source.cluster.alias": "source",
    "target.cluster.alias": "target",
    "source.bootstrap.servers": "<source-kafka>:9092",
    "target.bootstrap.servers": "<adm-q-kafka>:9092",
    "topics": ".*",
    "sync.group.offsets.enabled": "true"
  }
}
```

```
# Create Connector
curl -X POST -H "Content-Type: application/json" \
  --data @source-connector.json \
  http://localhost:8083/connectors/
```

6.3 Data Verification

```
# Verify data integrity
kafka/bin/kafka-consumer-groups.sh \
  --bootstrap-server <adm-q-kafka>:9092 \
  --group mirror-consumer-group \
  --describe
```

```
# Compare message counts between source and target clusters
kafka/bin/kafka-run-class.sh kafka.tools.GetOffsetShell \
  --broker-list <source-kafka>:9092 \
  --topic my-topic \
  --time -2 | tail -1

kafka/bin/kafka-run-class.sh kafka.tools.GetOffsetShell \
  --broker-list <admq-kafka>:9092 \
  --topic my-topic \
  --time -2 | tail -1
```

7 Client Migration

7.1 Switching Client Connection Address

Change the `bootstrap.servers` in the client configuration from the source cluster address to the ADMQ address:

```
# Original configuration
bootstrap.servers=192.168.1.10:9092,192.168.1.11:9092

# New configuration (ADMQ)
bootstrap.servers=192.168.1.20:9092,192.168.1.21:9092
```

7.2 No Code Changes Required

Since ADMQ is fully compatible with the Kafka protocol, the following aspects require no modification:

- Producer/consumer code
- Serialization/deserialization methods
- Consumer group management approach
- Topic operation approach

7.3 Verifying Client Migration

```
# Test producer
kafka/bin/kafka-console-producer.sh \
  --bootstrap-server <admq-kafka>:9092 \
  --topic test-migration

# Test consumer
kafka/bin/kafka-console-consumer.sh \
  --bootstrap-server <admq-kafka>:9092 \
  --topic test-migration \
  --from-beginning
```

8 Migration Steps

8.1 Step 1: Deploy ADMQ Cluster

Complete the ADMQ deployment according to the installation guide.

8.2 Step 2: Verify Cluster

```
# Verify ADMQ cluster health
kafka/bin/kafka-broker-api-versions.sh --bootstrap-server <admq-
kafka>:9092
kafka/bin/kafka-topics.sh --create --topic test --partitions 3 --
replication-factor 3
```

8.3 Step 3: Configuration Migration

Migrate the source cluster configuration file parameters to the ADMQ configuration correspondingly.

8.4 Step 4: Metadata Migration

Use tools to synchronize metadata such as Topics, ACLs, and consumer groups.

8.5 Step 5: Data Migration

Use MirrorMaker 2 or Kafka Connect to synchronize data.

8.6 Step 6: Client Switchover

Gradually switch clients to the ADMQ cluster in batches.

8.7 Step 7: Verification

- Verify message integrity
- Verify consumption progress
- Verify business functionality

8.8 Step 8: Cleanup

After confirming the migration is successful, decommission the source cluster.

9 Migration Risk Control

9.1 Rollback Plan

1. Keep the source cluster for a period of time
2. Client configuration supports fast switching
3. Record key configurations and states

9.2 Migration Window

Data Volume	Estimated Migration Time	Description
< 1TB	2-4 hours	Small cluster
1-10TB	4-12 hours	Medium cluster
> 10TB	1-3 days	Large cluster

9.3 Precautions

1. **Network Bandwidth:** Ensure sufficient network bandwidth between the source cluster and the ADMQ cluster
2. **Disk Space:** The ADMQ cluster needs sufficient disk space to store data
3. **Business Off-peak Hours:** It is recommended to perform data migration during business off-peak hours
4. **Monitoring and Alerts:** Enable monitoring and alerts to detect anomalies promptly

10 Frequently Asked Questions

10.1 Topic Creation Failed

Cause: Topic name conflict or partition count exceeds the number of Brokers **Solution:** Check the Topic name and ensure the partition count $\leq$ Broker count

10.2 Data Synchronization Latency

Cause: Insufficient network bandwidth or improper synchronization task configuration **Solution:** Check the network and increase the MirrorMaker concurrency

10.3 Consumption Progress Lost

Cause: Consumer group offsets were not synchronized **Solution:** Enable the `sync.group.offsets.enabled` configuration

10.4 ACL Not Effective

Cause: ACL configuration was not synchronized **Solution:** Manually synchronize the ACL configuration

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

