



Migration Manual

Kingdee Apusic Distributed Message Queue
V2.0.6_for_rocketmq

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 Process Diagram
- 3 Migration Overview
 - 3.1 Migration Principles
 - 3.2 Migration Scope
 - 3.3 Pre-migration Checks
 - 3.3.1 Version Compatibility Check
 - 3.3.2 Client Compatibility Check
- 4 Migration Plan
 - 4.1 Plan 1: Dual-Write Migration (Recommended)
 - 4.2 Plan 2: Service-Stop Migration
- 5 Configuration Migration
 - 5.1 Export Original RocketMQ Configuration
 - 5.2 Create Configuration in ADMQ for RocketMQ
 - 5.2.1 Import Metadata Using the Console
- 6 Data Migration
 - 6.1 Migration Using Message Lookback
 - 6.2 Export and Import Using mqadmin
- 7 Client Migration
 - 7.1 NameServer Address Change
 - 7.2 Common Client Configurations
 - 7.2.1 Java (Spring Boot)
 - 7.2.2 Go
- 8 Migration Verification
 - 8.1 Functional Verification
 - 8.2 Performance Verification
 - 8.3 Data Consistency Verification
- 9 Rollback Plan

1 Preface

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

1.1 Target Audience

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

1.2 Related Documents

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

No.	Document	Description
1	Kingdee Apusic Distributed Message Queue for RocketMQ Quick Start Guide	Briefly introduces how to quickly get started with ADMQ for RocketMQ.
2	Kingdee Apusic Distributed Message Queue for RocketMQ Installation Guide	Details how to install ADMQ for RocketMQ on various operating systems, as well as ADMQ for RocketMQ service start/stop operations.
3	Kingdee Apusic Distributed Message Queue for RocketMQ Message Engine User Guide	Details the usage, configuration, management, and supporting tools of ADMQ for RocketMQ message engine related features.
4	Kingdee Apusic Distributed Message Queue for RocketMQ Console User Guide	Details the usage and operation instructions of the ADMQ for RocketMQ console.
5	Kingdee Apusic Distributed Message Queue for RocketMQ Development Guide	Details the instructions for developing ADMQ for RocketMQ client applications in various development languages.
6	Kingdee Apusic Distributed Message Queue for RocketMQ Migration Guide	Details the instructions for migrating from RocketMQ to ADMQ for RocketMQ.
7	Kingdee Apusic Distributed Message Queue for RocketMQ Operations Guide	Details ADMQ for RocketMQ monitoring, operations, security hardening and other operations instructions.

8	Kingdee Apusic Distributed Message Queue for RocketMQ Performance Optimization Guide	Details the instructions for ADMQ for RocketMQ performance tuning.
---	--	--

1.3 Technical Support

ADMQ for RocketMQ provides comprehensive technical support services. You can obtain technical support through the following methods:

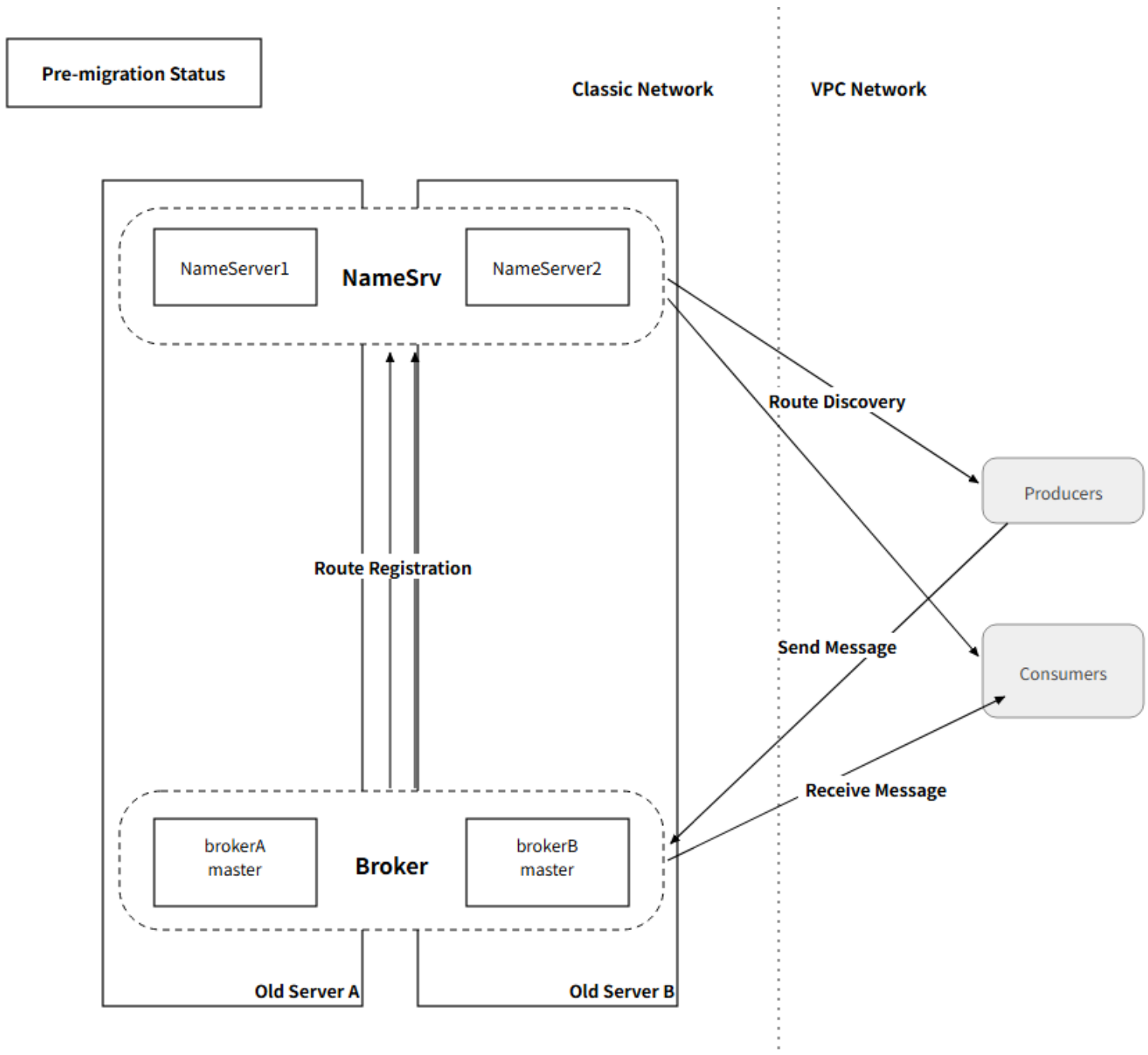
- Website: www.apusic.com
- Phone: 400-855-5800
- Email: support@apusic.com
- Kingdee Cloud Community: <https://vip.kingdee.com/?productId=73&productLineId=14&lang=zh-CN>

When contacting technical support, please provide the following information:

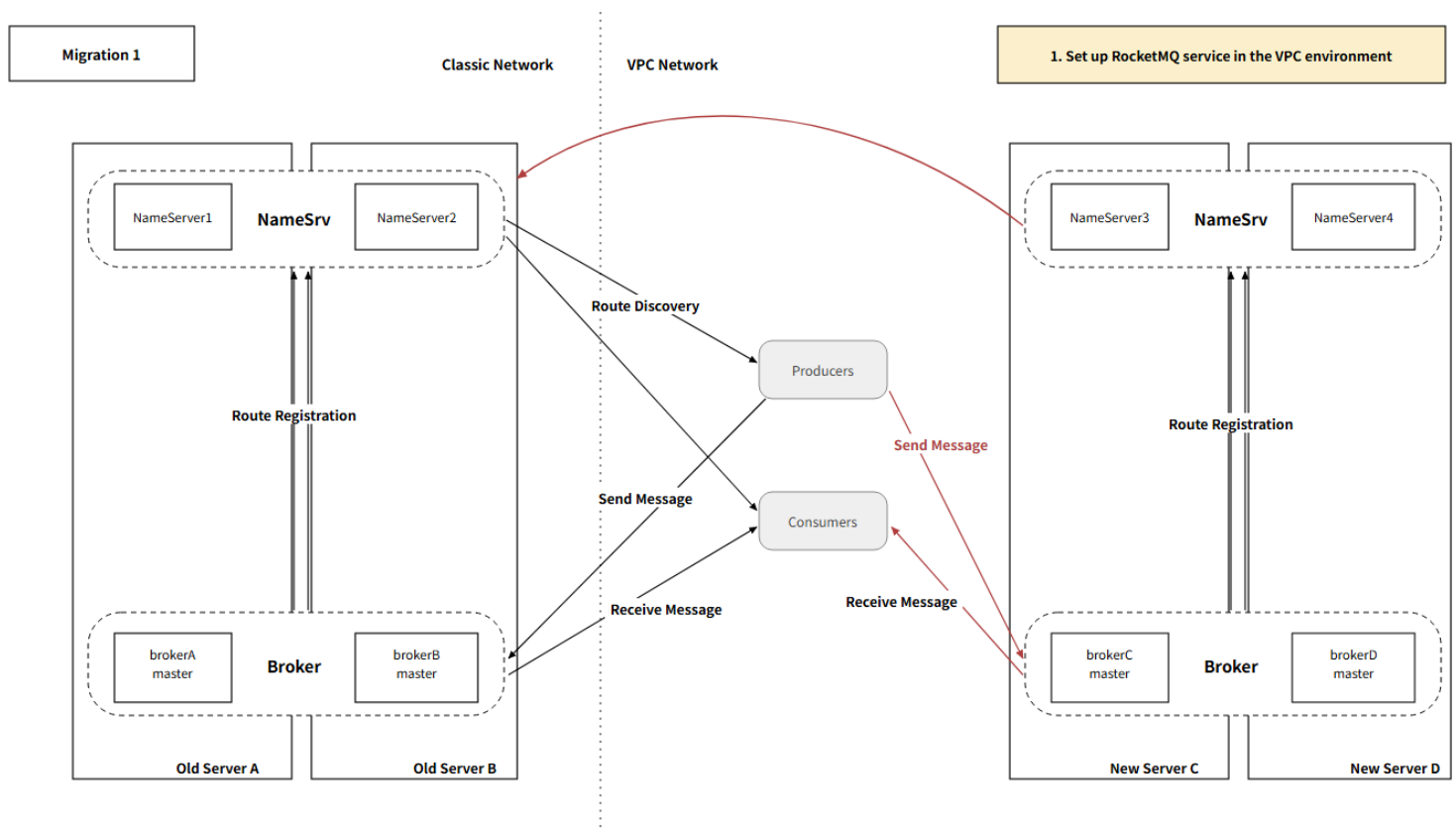
1. Your name
2. Company information and contact details
3. Operating system and its version
4. Product version number
5. Detailed information such as logs, screenshots of exceptions and errors

2 Process Diagram

Pre-migration Status

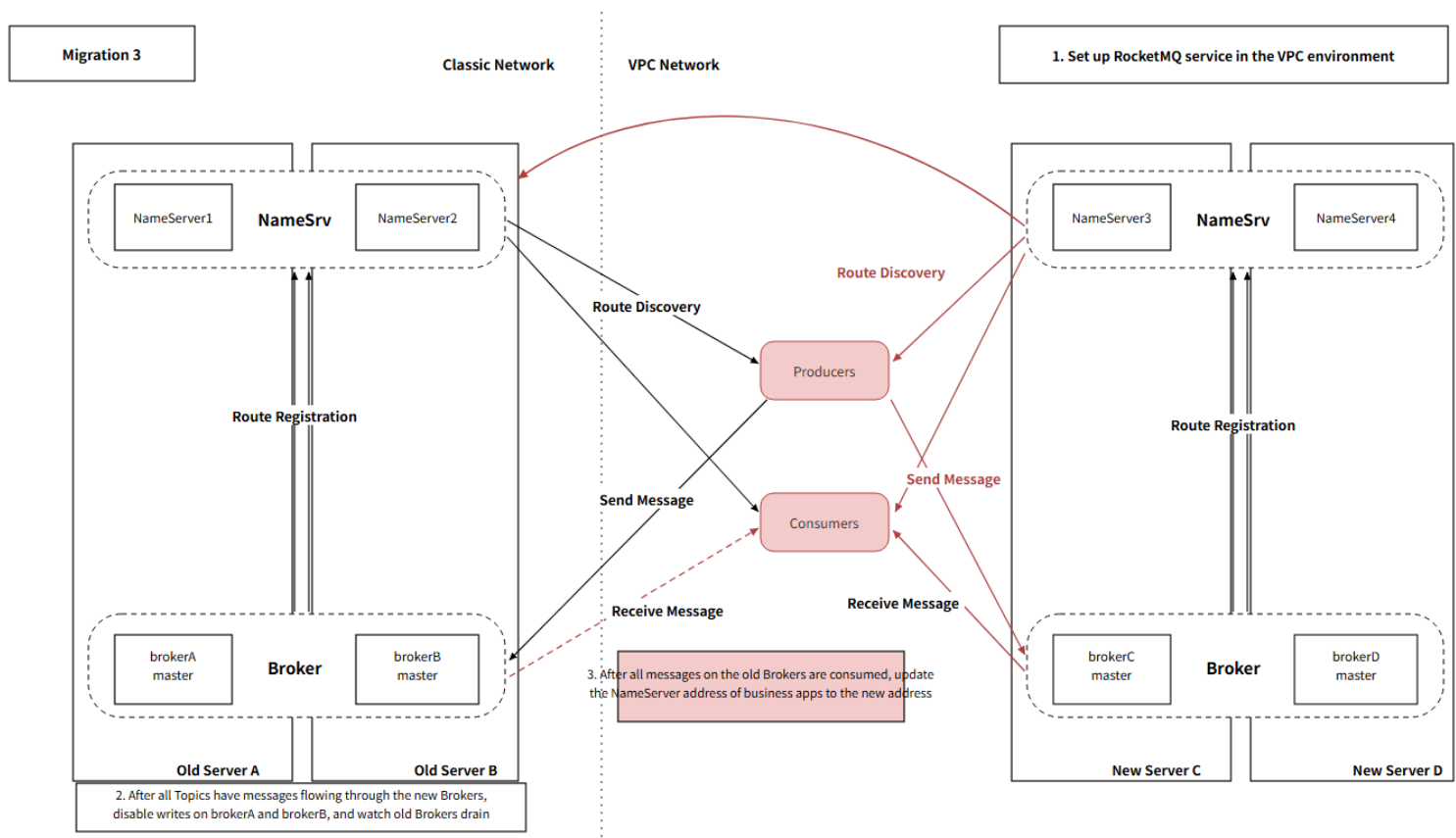


Set up New RocketMQ Service

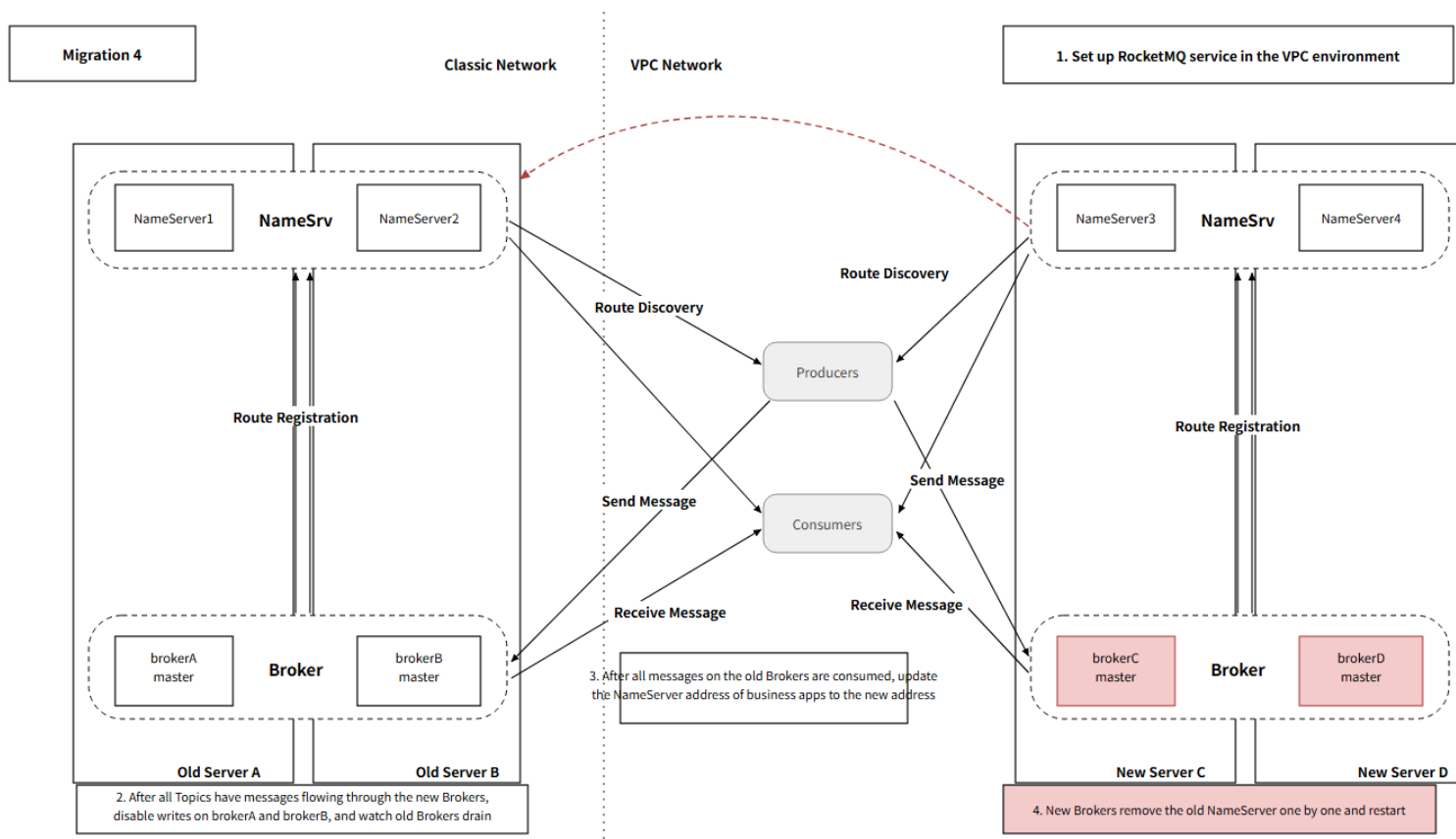


Disable Write Permissions on Old Broker

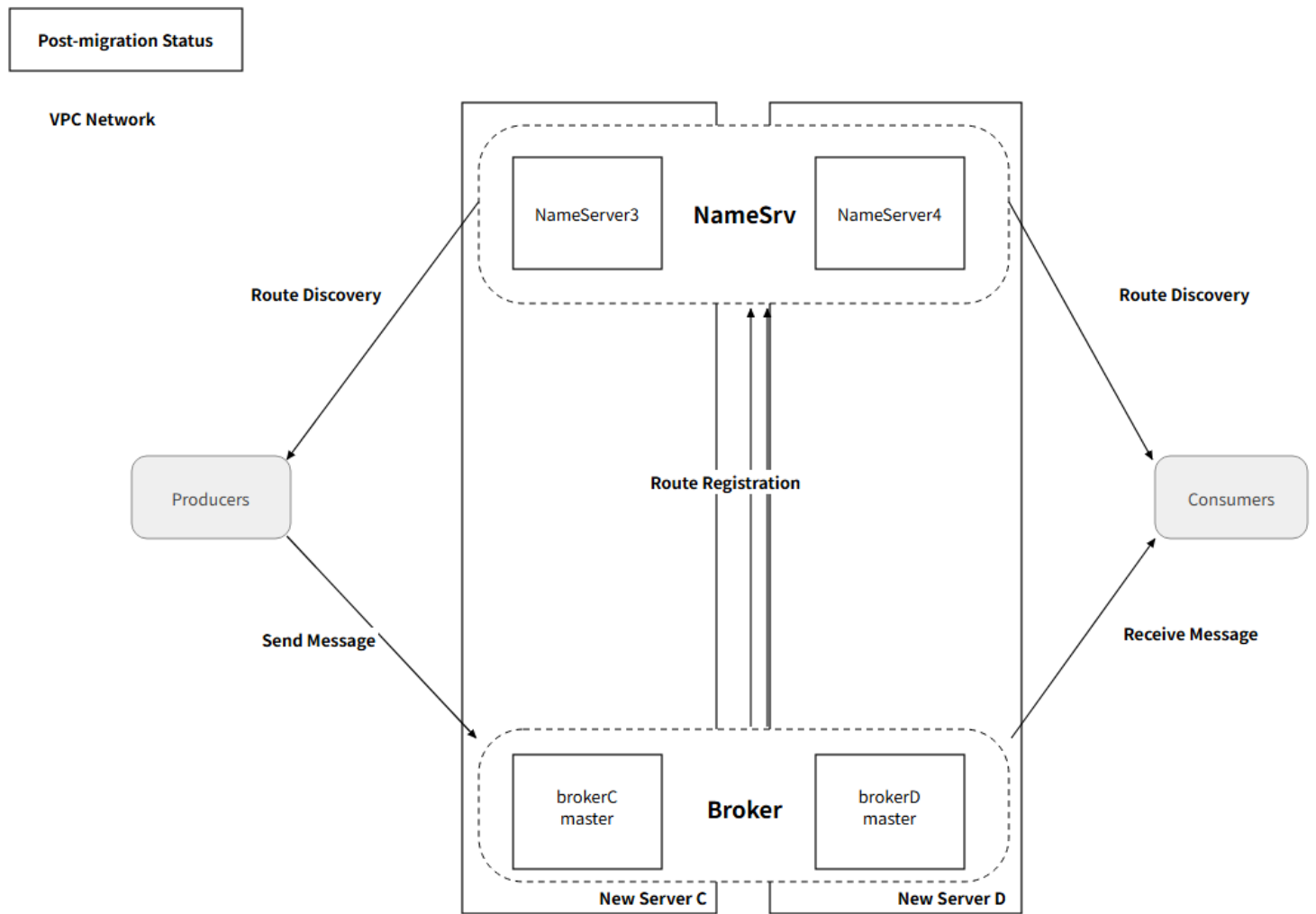
Switch to New NameSrv Address



Remove Old NameSrv Address from New Broker



Post-migration Status



3 Migration Overview

3.1 Migration Principles

ADMQ for RocketMQ is fully compatible with RocketMQ's protocol and clients. The migration process follows these principles:

1. **Zero Code Modification:** Clients require no code changes, only the NameServer address needs to be updated
2. **Configuration Compatibility:** Supports RocketMQ native configuration parameters
3. **Data Migratable:** Supports Topic configuration and message data migration
4. **Smooth Switchover:** Supports dual-write and grayscale switchover to reduce migration risk

3.2 Migration Scope

Migration Content	Supported	Description
Application Code	Yes	No modification needed, only NameServer address needs to be changed
Client Configuration	Yes	Supports RocketMQ native clients
Topic Definition	Yes	Supports Topic creation and configuration migration
Consumer Group	Yes	Supports consumer group configuration migration
User Permissions	Yes	Supports ACL configuration migration
Message Data	Yes	Supports historical message migration
Consumer Offset	Yes	Supports consumption progress migration
Message Trace	Yes	Supports message trace configuration

3.3 Pre-migration Checks

3.3.1 Version Compatibility Check

Original RocketMQ Version	Compatibility	Description
4.9.x	Fully Compatible	Recommended migration version
5.0.x~5.3.x	Fully Compatible	Recommended migration version

3.3.2 Client Compatibility Check

Client Type	Compatible Version	Description
Java	4.9.x - 5.3.x	Fully Compatible
Go	2.x	Fully Compatible
Python	0.4.x	Fully Compatible
C/C++	2.x	Fully Compatible

4 Migration Plan

4.1 Plan 1: Dual-Write Migration (Recommended)

Suitable for business scenarios that cannot tolerate downtime, ensuring data consistency through dual-write.

Steps:

1. Deploy ADMQ for RocketMQ Cluster

- Deploy the new cluster according to the installation guide
- Configure network and firewall

2. Configure Dual-Write

- Connect to both the original RocketMQ and ADMQ for RocketMQ in the application simultaneously
- Send messages to both clusters when sending

3. Sync Historical Data

- Use message回溯 (lookback) functionality to sync historical messages
- Verify data consistency

4. Switch Consumers

- First switch some consumers to ADMQ for RocketMQ
- Observe whether the business operates normally

5. Full Switchover

- After confirming no issues, switch all consumers
- Stop sending messages to the original RocketMQ

6. Decommission Original Cluster

- Retain for a period for data query purposes
- Decommission after confirming no issues

4.2 Plan 2: Service-Stop Migration

Suitable for business scenarios that can tolerate brief downtime, with faster migration speed.

Steps:

1. Stop Producers

- Pause the business system from sending messages to RocketMQ

2. Export Configuration

- Use mqadmin to export Topic, consumer group, and other configurations

3. Migrate Configuration to ADMQ for RocketMQ

- Create corresponding resources in the console or command line

4. Migrate Message Data

- Use message query and sending tools to migrate messages

5. Switch NameServer Address

- Modify application configuration to point to ADMQ for RocketMQ

6. Start Producers

- Resume business system message sending

5 Configuration Migration

5.1 Export Original RocketMQ Configuration

```
# Use the export command to export corresponding configurations and metadata

./bin/mqadmin exportConfigs -n {nameserver address and port} -c {RocketMQ cluster name} -f {path to export metadata file}

./bin/mqadmin exportMetadata -n {nameserver address and port} -c {RocketMQ cluster name} -f {path to export metadata file}
```

5.2 Create Configuration in ADMQ for RocketMQ

5.2.1 Import Metadata Using the Console

1. Log in to the console
2. Go to the "Metadata Import" page
3. Create a metadata import task to import the corresponding metadata

6 Data Migration

6.1 Migration Using Message Lookback

RocketMQ supports looking back to consume messages by time, which can be used for data migration.

```
// Consume historical messages from the original cluster and send to
the new cluster

DefaultLitePullConsumer consumer = new
DefaultLitePullConsumer("migrate-group");
consumer.setNamesrvAddr("old-rocketmq:9876");
consumer.subscribe("test-topic", "*");
consumer.setConsumeFromWhere(ConsumeFromWhere.CONSUME_FROM_FIRST_OFFSET);
consumer.start();

DefaultMQProducer producer = new DefaultMQProducer("migrate-group");
producer.setNamesrvAddr("adm-q-rocketmq:9876");
producer.start();

while (true) {
    List<MessageExt> msgs = consumer.poll();
    for (MessageExt msg : msgs) {
        Message newMsg = new Message(msg.getTopic(), msg.getTags(),
msg.getKeys(), msg.getBody());
        producer.send(newMsg);
    }
}
```

6.2 Export and Import Using mqadmin

```
# Export messages (by time range)
bin/mqadmin printMsg -n localhost:9876 -t test-topic -s "2026-01-
01#00:00:00:00:000" -e "2026-01-02#00:00:00:00:000" > messages.txt
```

```
# Parse and send to the new cluster using a script  
# You need to write your own script to parse the message content and  
send it
```

7 Client Migration

7.1 NameServer Address Change

Only the NameServer address in the application configuration file needs to be modified:

```
# Original configuration
rocketmq.namesrv.addr=old-rocketmq:9876

# New configuration
rocketmq.namesrv.addr=admq-rocketmq:9876
```

7.2 Common Client Configurations

7.2.1 Java (Spring Boot)

```
rocketmq:
  name-server: admq-rocketmq:9876
  producer:
    group: test-group
  consumer:
    group: test-group
```

7.2.2 Go

```
producer, err := golang.NewProducer(
  golang.WithEndpoints("admq-rocketmq:9876"),
  golang.WithConsumerGroup("test-group"),
)
```

8 Migration Verification

8.1 Functional Verification

Verification Item	Verification Method
Connection Test	Use a client to connect to ADMQ for RocketMQ, confirm that the connection can be established normally
Message Sending	Send a test message, confirm it successfully reaches the Broker
Message Consumption	Start a consumer, confirm it can normally receive and process messages
Ordered Messages	Send ordered messages, confirm the consumption order is correct
Delayed Messages	Send delayed messages, confirm timely delivery
Transactional Messages	Send transactional messages, confirm transaction consistency

8.2 Performance Verification

Verification Item	Verification Method
Throughput	Compare the message sending rate of the original cluster and the new cluster
Latency	Measure the latency from message sending to consumption
Concurrency	Test multi-producer, multi-consumer concurrent scenarios
Stability	Run long-duration tests, observe for any anomalies

8.3 Data Consistency Verification

```
# Compare the number of Topic messages between the original cluster
and the new cluster
bin/mqadmin topicStatus -n old-rocketmq:9876 -t test-topic
bin/mqadmin topicStatus -n admq-rocketmq:9876 -t test-topic

# Compare consumer progress
```

```
bin/mqadmin consumerProgress -n old-rocketmq:9876 -g test-group  
bin/mqadmin consumerProgress -n admq-rocketmq:9876 -g test-group
```

9 Rollback Plan

If problems occur during migration, rollback to the original RocketMQ cluster:

1. Stop sending messages to ADMQ for RocketMQ
2. Switch the NameServer address back to the original RocketMQ
3. Sync new messages from ADMQ for RocketMQ back to the original cluster (if needed)
4. Restore the business system

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

