Distributed Message Service for RocketMQ

User Guide

Issue	01
Date	2024-04-08

HUAWEI CLOUD COMPUTING TECHNOLOGIES CO., LTD.

Trademarks and Permissions

Notice

The purchased products, services and features are stipulated by the contract made between Huawei Cloud and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

Contents

1 Service Overview

1.1 What Is DMS for RocketMQ?

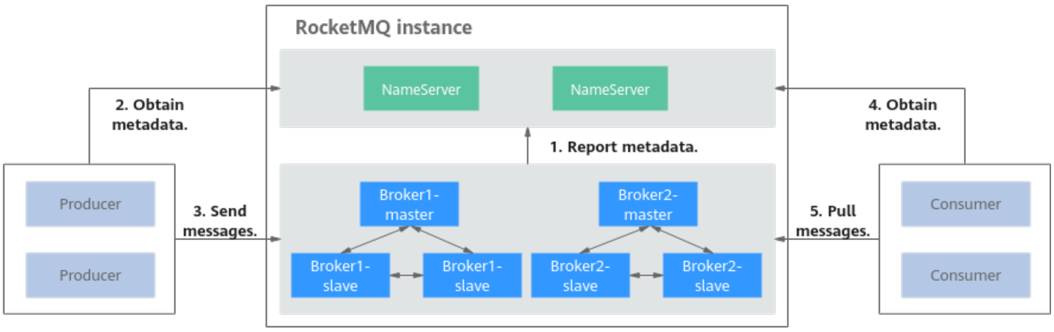
Distributed Message Service (DMS) for RocketMQ is message-oriented middleware that delivers low latency, high flexibility, high throughput, dynamic expansion, easy management, and abundant messaging functions.

DMS for RocketMQ has the following features:

- Compatibility with open-source RocketMQ clients.
- Abundant messaging functions, including ordered message delivery, delayed messages, scheduled messages, message retry, dead letter messages, and transactional messages, which meet diverse needs in e-commerce and finance scenarios.
- Monitoring and analysis functions, including message tracing, message tracking, trace analysis, dead letter message export, monitoring and alarms, which allow you to monitor your services and keep them up and running.

Product Architecture

Figure 1-1 Product architecture



Architecture description:

- Brokers receive and store messages sent by producers or forward messages to consumers. A broker consists of one master and two slave nodes.

- NameServers receive and store broker metadata.
- Producers obtain metadata from NameServers and send messages to brokers.
- Consumers obtain metadata from NameServers and then pull messages from brokers.

Message Types

DMS for RocketMQ supports four messages types.

- Normal messages: Messages that do no have any features of delayed messages, ordered messages, or transactional messages.
- Delayed/Scheduled messages: Messages that are delivered to consumers after a specific period after being sent from producers to DMS for RocketMQ.
- Ordered messages: Messages that are retrieved in the exact order that they are created.
- Transactional message: Messages that achieve eventual consistency, delivering distributed transaction processing similar to X/Open XA.

Advanced Features

DMS for RocketMQ provides four advanced features.

- Message filtering: Consumers can tag messages in the subscribed topics to retrieve only the required messages.
- Message retry: DMS for RocketMQ retries sending messages that fail to be retrieved. If the message retrieval still fails after the maximum number of retries is reached, the messages are sent to the dead letter queue.

  Table 1-1 describes the retry mechanism in DMS for RocketMQ.

Table 1-1 Message retry mechanism

Retrieval Type	Retry Interval	Maximum Retries
Ordered retrieval	The interval is set by the **suspendTimeMillis** parameter. The default interval is 1000 ms, that is, 1s.	The maximum number of retries is set by the **setMaxReconsumeTimes** method on the consumer. If this parameter is not set, there will be unlimited retries.
Normal retrieval	The interval varies by the retry number, as shown in **Table 1-2**.	Set during consumer group creation. Value range: 1–16 times

Table 1-2 Intervals between retries in normal retrievals

Retry No.	Interval	Retry No.	Interval
1	10s	9	7 min
2	30s	10	8 min
3	1 min	11	9 min
4	2 min	12	10 min
5	3 min	13	20 min
6	4 min	14	30 min
7	5 min	15	1 h
8	6 min	16	2 h

- Delayed messages: After being sent from producers to DMS for RocketMQ, messages are delivered to consumers only after a **fixed period**. A producer can specify one of the 18 delay levels listed in **Table 1-3**.

Table 1-3 Delay levels

Delay Level	Delay	Delay Level	Delay
1	1s	10	6 min
2	5s	11	7 min
3	10s	12	8 min
4	30s	13	9 min
5	1 min	14	10 min
6	2 min	15	20 min
7	3 min	16	30 min
8	4 min	17	1 h
9	5 min	18	2 h

- Scheduled messages: After being sent from producers to DMS for RocketMQ, messages are delivered to consumers only after a specified time point. In DMS for RocketMQ, you can schedule messages to be delivered at **any time** within one year. You can also cancel scheduled messages.

1.2 Product Advantages

DMS for RocketMQ has the following advantages:

- Instant availability: You can build your own message service in the cloud in just a few steps. After you create a RocketMQ instance, you can quickly access it at its connection addresses. DMS for RocketMQ is fully compatible with open-source RocketMQ, allowing you to migrate your applications to the cloud with no change to the application code.
- Fully hosted services: DMS for RocketMQ provides automatic deployment, after-sales services, and comprehensive O&M solutions including monitoring and alarms. You can focus on your service development without having to worry about deployment and O&M.
- Low latency: For the deployment on the cloud service platform, intranet access latency is reduced to microseconds.
- Elasticity and high reliability: Raft-based clustering facilitates fault detection and failover, ensuring service continuity and reliability.
- Dynamic scaling: Clusters can be scaled out dynamically as service demand changes.
- Easy management: Monitoring and alarms, link diagnosis, and message tracing facilitate fault locating and routine maintenance.
- Abundant messaging functions: Ordered message delivery, intentional delivery delay, message retry, dead letter messages, message filtering, and transaction messages meet needs in diverse scenarios.

1.3 Application Scenarios

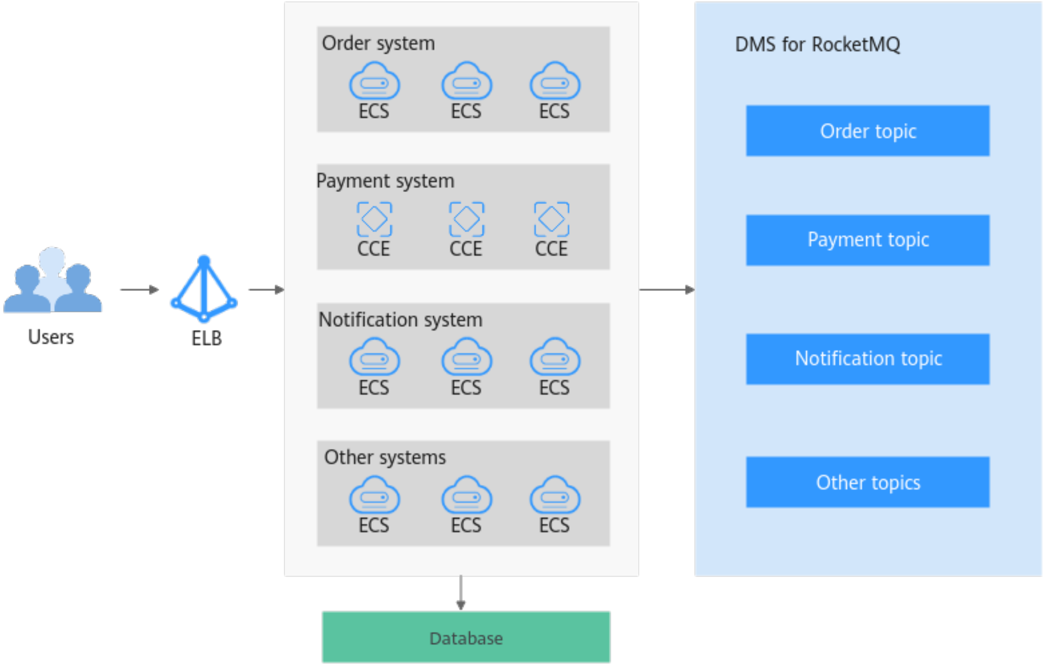
E-Commerce

E-commerce applications face the following challenges:

- E-commerce applications usually involve order placement, payment, and notification processing. Multiple service systems work together to complete a task, and upper-layer service systems depend on lower-layer service systems. If there are a large number of layers, the performance of user-facing services will deteriorate, affecting user experience.
- During e-commerce shopping events, service systems that provide subscription-based notifications are required.

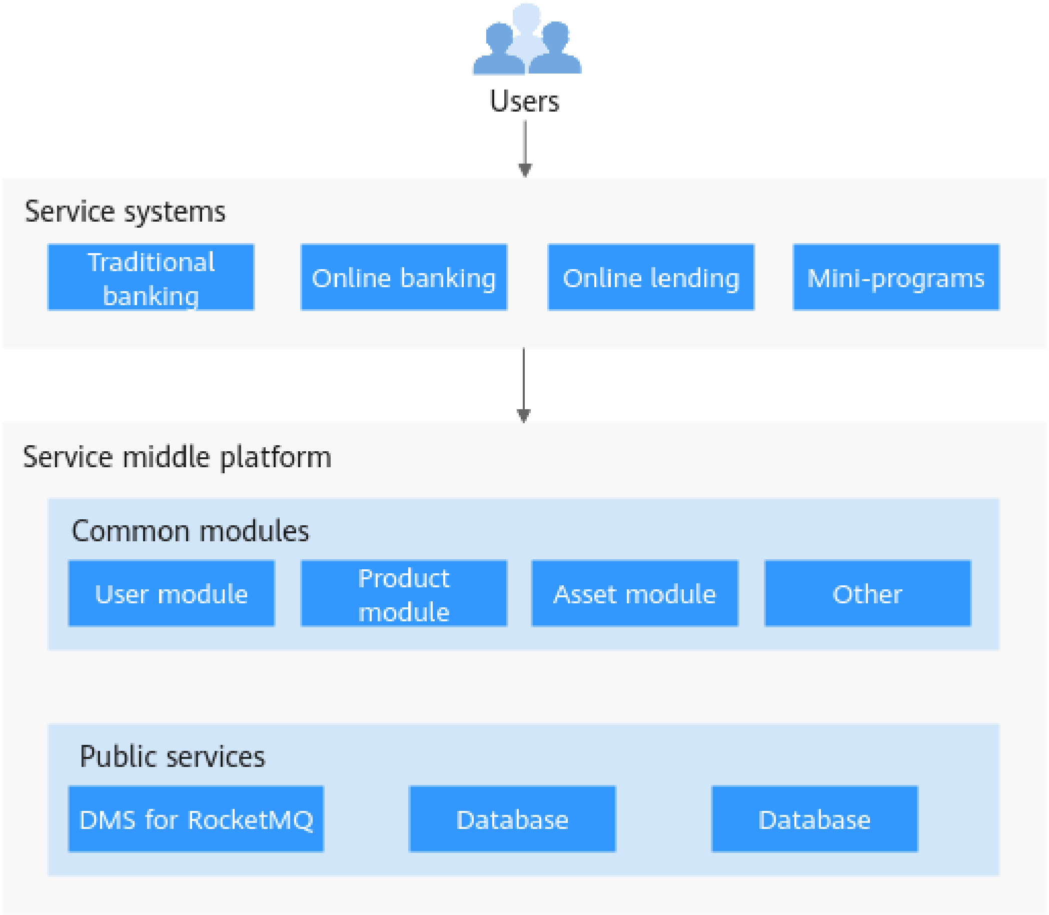
DMS for RocketMQ can help you cope with e-commerce challenges.

- DMS for RocketMQ decouples service systems to improve their processing capabilities and response speeds.
- DMS (for RocketMQ) allows for scheduled and delayed message delivery to meet the requirements for subscription-based notifications.



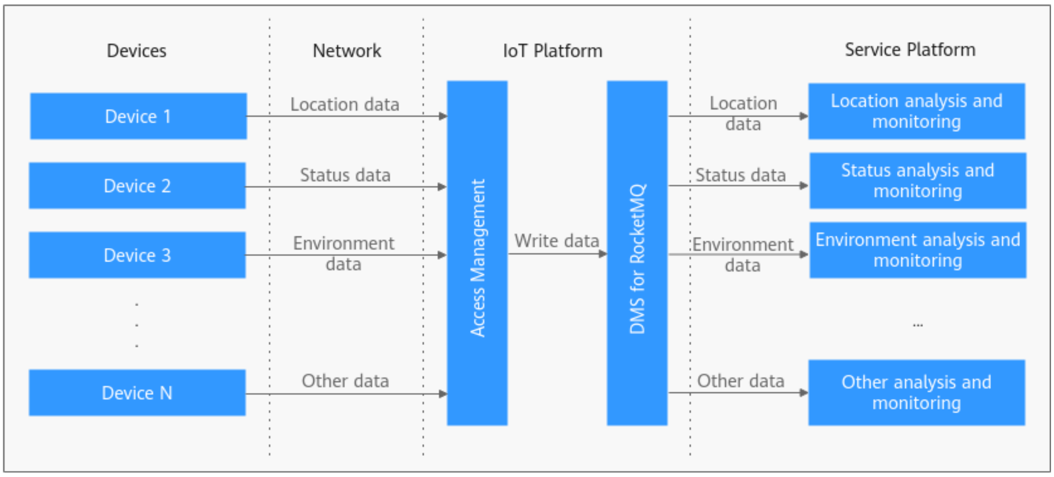
Finance

Unlike traditional finance, Internet finance requires fast responses to the rapidly changing Internet landscape. Common functions in traditional finance are extracted and then packaged into microservice modules to form a middle platform which provides basic capabilities for upper-layer service systems. With its decoupling and linking advantages, DMS for RocketMQ improves the processing capability and response speed of each microservice module, playing an indispensable role in the service middle platform.



IoT

Massive device access is typical in IoT scenarios. A massive number of devices bring in massive amounts of data. However, each service component only desires specific data. Therefore, it is important to quickly identify data that services desire from a large amount of data. To address this challenge, DMS for RocketMQ provides message filtering. When sent from devices, messages are tagged to specify the message type. Services can specify only the desired types of messages to retrieve.



1.4 Specifications

DMS for RocketMQ is compatible with open-source RocketMQ 4.8.0. Their specifications are described as follows.

DMS for RocketMQ 4.8.0

The specifications of a RocketMQ 4.8.0 instance are defined on six dimensions:

- ECS flavor: flavor of the ECS used by the instance
- Number of brokers: scale of the instance
- Storage space: storage capacity available for each broker
- TPS per broker: TPS performance of each broker
- Max. topics per broker: number of topics allowed for each broker
- Max. consumer groups per broker: number of consumer groups allowed for each broker

Table 1-4 lists the flavors of DMS for RocketMQ 4.8.0.

NOTE

TPS (Transaction per second) indicates the total number of produced and consumed messages per second, that is, the total throughput of message production and consumption per second for a flavor.

Table 1-4 Instance flavors (DMS for RocketMQ 4.8.0)

Flavor	Brokers	Storage Space (GB per Broker)	TPS per Broker	Max. Topics per Broker	Max. Consumer Groups per Broker
rocketmq.4u8g.cluster.small	1–2	300–60,000	15,000	2000	2000
rocketmq.4u8g.cluster	1–10	300–600,000	20,000	4000	4000
rocketmq.8u16g.cluster	1–10	300–900,000	25,000	8000	8000
rocketmq.12u24g.cluster	1–10	300–900,000	28,000	12,000	12,000
rocketmq.16u32g.cluster	1–10	300–900,000	30,000	16,000	16,000

1.5 Comparing RocketMQ, Kafka, and RabbitMQ

Table 1-5 Functions

Feature	RocketMQ	Kafka	RabbitMQ
Priority queue	Not supported	Not supported	Supported. It is recommended that the priority be set to 0–10.
Delayed queue	Supported	Not supported	Supported
Dead letter queue	Supported	Not supported	Supported
Message retry	Supported	Not supported	Not supported
Retrieval mode	Pull-based and push-based	Pull-based	Pull-based and push-based
Message broadcasting	Supported	Supported	Supported
Message tracking	Supported	Supports offset and timestamp tracking.	Not supported. Once a message retrieval has been acknowledged, RabbitMQ will be notified that the message can be deleted.
Message accumulation	Supported	Supports higher accumulation performance than RabbitMQ thanks to high throughput.	Supported
Persistence	Supported	Supported	Supported
Message tracing	Supported	Not supported	Supported by the firehose feature or the rabbitmq_tracing plugin. However, rabbitmq_tracing reduces performance and should be used only for troubleshooting.

Feature	RocketMQ	Kafka	RabbitMQ
Message filtering	Supported	Supported	Not supported, but can be encapsulated.
Multi-tenancy	Supported	Not supported	Supported
Multi-protocol	Compatible with RocketMQ.	Only supports Apache Kafka.	RabbitMQ is based on AMQP and supports MQTT and STOMP.
Multi-language	Supports clients in multiple programming languages.	Kafka is written in Scala and Java and supports clients in multiple programming languages.	RabbitMQ is written in Erlang and supports clients in multiple programming languages.
Throttling	Planned	Supports throttling on producer or consumer clients.	Supports credit-based throttling on producers, a mechanism that triggers protection from within.
Ordered message delivery	Message order is maintained within a queue.	Supports partition-level FIFO.	Not supported. Supports FIFO only for single-threaded message queuing without advanced features such as delayed queues or priority queues.
Security	Supports SSL authentication.	Supports SSL and SASL authentication and read/write permissions control.	Supports SSL.
Transactional messages	Supported	Supported	Supported

1.6 Comparing DMS for RocketMQ and Open-Source RocketMQ

DMS for RocketMQ maintains open-source compatibility while supporting or enhancing features in the open-source versions.

Table 1-6 Differences between DMS for RocketMQ and open-source RocketMQ

Feature	DMS for RocketMQ	Open-Source RocketMQ
Delayed/Scheduled messages	● Delayed message: There are 18 delay levels. The longest delay is two hours. ● Scheduled messages: The message delivery delay can be customized. The longest delay is one year.	Only 18 delay levels are supported. The longest delay is two hours.
Ordered messages	Supported	Supported
Message retry	Supported	Supported
Dead letter messages	Supported	Supported
Cluster consumption	Supported	Supported
Message broadcasting	Supported	Supported
Dead letter queues	Supported	Supported
Resetting message retrieval start position	Supported	Supported
Message query	Supported	Supported
Encrypted transmission	Supported	Supported
Message tracing	Supported	Supported
Transactional messages	Supported. The performance is ten times higher than that of open-source RocketMQ when a large number of transactional messages are stacked.	Supported. However, if a large number of transactional messages are stacked, the performance is poor.
Exporting dead letter messages	Supported	Not supported
Data dumping	Planned	Not supported
Instance diagnosis	One-click diagnosis of retrieval issues	Not supported
Instance monitoring	Historical monitoring data can be displayed in graphs. More than 18 metrics can be monitored.	Only current monitoring data can be viewed. Metrics such as CPU and memory cannot be monitored.

Feature	DMS for RocketMQ	Open-Source RocketMQ
ACL	Flexible configuration with a few clicks	Complex configuration
O&M and capacity expansion	Fast expansion with a few clicks	Manual expansion, which is complex

1.7 Restrictions

This section describes the restrictions on using DMS for RocketMQ.

Table 1-7 DMS for RocketMQ restrictions

Item	Restrictions
Maximum number of topics	Depends on the **instance specifications**. This limit cannot be changed. When this limit is reached, no more topics can be created.
Maximum number of consumer groups	Depends on the **instance specifications**. This limit cannot be changed. When this limit is reached, no more consumer groups can be created.
Message size	The maximum size of a message is 4 MB. The maximum size of a message attribute is 16 KB. The message size cannot be changed. When this limit is reached, messages will fail to be sent.
Message retention duration	Messages can be retained for a maximum of two days and will be automatically deleted after two days. The retention period cannot be modified.
Consumer offset reset	You can reset the retrieval start position to any time within two days.
Delay of scheduled messages	The maximum delay is one year. This limit cannot be changed. You can schedule messages to be delivered at any time within one year.
Request-Reply	Not supported.
Configuration parameters	Cannot be modified through open-source APIs.
Broker faults	When some brokers of an instance are faulty, topics, consumer groups, and users cannot be created, modified, or deleted, but can be queried.

1.8 Related Services

- Virtual Private Cloud (VPC)

  RocketMQ instances run in VPCs and use the IP addresses and bandwidth of VPC. Security groups of VPCs enhance the security of network access to the instances.

- Cloud Eye

  Cloud Eye is an open platform that provides monitoring, alarm reporting, and alarm notification on your resources in near real time.

- Cloud Trace Service (CTS)

  Cloud Trace Service (CTS) generates traces to provide you with a history of operations performed on cloud service resources. The traces include operation requests sent using the management console or open APIs, as well as the operation results. You can view all generated traces to query, audit, and backtrack performed operations.

- Elastic Cloud Server (ECS)

  An ECS is a basic computing unit that consists of vCPUs, memory, OS, and EVS disks. RocketMQ instances run on ECSs. A broker corresponds to three ECSs.

- Elastic Volume Service (EVS)

  EVS provides block storage services for ECSs. All RocketMQ data, such as messages, metadata, and logs, is stored in EVS disks.

- Elastic IP (EIP)

  The EIP service provides independent public IP addresses and bandwidth for Internet access. RocketMQ instances bound with EIPs can be accessed over public networks.

- Tag Management Service (TMS)

  TMS is a visualized service for fast and unified cross-region tagging and categorization of cloud services.

  Tags facilitate RocketMQ instance identification and management.

1.9 RocketMQ Concepts

Topic

A topic is a category of messages. It is the basic unit for creating and retrieving messages.

Queue

A topic consists of multiple queues. A larger number of queues indicates higher retrieval concurrency.

Producer

A producer sends messages to the server.

Producer Group

Producers in a group send the same type of messages with the same logic.

Consumer

A consumer obtains messages from the server.

Consumer Group

A consumer group contains consumers that have similar retrieval behavior.

Broker

Brokers are a cluster of ECSs that process services.

NameServer

A lightweight registry that stores metadata. Before producing or consuming messages, producers and consumers must obtain metadata from a NameServer.

1.10 Permissions Management

You can use Identity and Access Management (IAM) to manage DMS for RocketMQ permissions and control access to your resources. IAM provides identity authentication, permissions management, and access control, helping you secure access to your cloud resources.

You can create IAM users for your employees, and assign permissions to these users on a principle of least privilege (PoLP) basis to control their access to specific resource types. For example, you can create IAM users for software developers and assign specific permissions to allow them to use DMS for RocketMQ resources but prevent them from being able to delete resources or perform any high-risk operations.

If your account does not require individual IAM users for permissions management, skip this section.

IAM is free of charge. You pay only for the resources in your account. For more information, see **IAM Service Overview**.

NOTE

DMS for RocketMQ permissions policies are based on DMS. Therefore, when assigning permissions, select DMS permissions policies.

DMS for RocketMQ Permissions

By default, new IAM users do not have any permissions assigned. To assign permissions to these new users, add them to one or more groups, and attach permissions policies or roles to these groups.

DMS for RocketMQ is a project-level service deployed and accessed in specific physical regions. When assigning DMS for RocketMQ permissions to a user group, specify region-specific projects where the permissions will take effect. If you select **All projects**, the permissions will be granted for all region-specific projects. When accessing DMS for RocketMQ, the users need to switch to a region where they have been authorized to use this service.

You can grant users permissions by using roles and policies.

- Roles: A type of coarse-grained authorization mechanism that provides only a limited number of service-level roles. When using roles to grant permissions, you also need to assign dependency roles. However, roles are not an ideal choice for fine-grained authorization and secure access control.
- Policies: A fine-grained authorization strategy that defines permissions required to perform operations on specific cloud resources under certain conditions. This mechanism allows for more flexible policy-based authorization for securer access control. For example, you can grant DMS for RocketMQ users only the permissions for managing DMS for RocketMQ instances. Most policies define permissions based on APIs. For the API actions supported by DMS for RocketMQ, see **Permissions Policies and Supported Actions**.

Table 1-8 lists all the system-defined roles and policies supported by DMS for RocketMQ.

Table 1-8 System-defined roles and policies supported by DMS for RocketMQ

Role/Policy Name	Description	Type	Dependencies
DMS FullAccess	Administrator permissions for DMS. Users granted these permissions can perform all operations on DMS.	System-defined policy	None
DMS UserAccess	Common user permissions for DMS, excluding permissions for creating, modifying, deleting, and scaling up instances.	System-defined policy	None
DMS ReadOnlyAccess	Read-only permissions for DMS. Users granted these permissions can only view DMS data.	System-defined policy	None
DMS VPCAccess	VPC operation permissions to assign to DMS agencies.	System-defined policy	None

Role/Policy Name	Description	Type	Dependencies
DMS KMSAccess	KMS operation permissions to assign to DMS agencies.	System-defined policy	None
DMS Administrator	Administrator permissions for DMS.	System-defined role	This role depends on the **Tenant Guest** and **VPC Administrator** roles.

Table 1-9 lists the common operations supported by each DMS for RocketMQ system policy or role. Select the policies or roles as required.

Table 1-9 Common operations supported by each system-defined policy or role of DMS for RocketMQ

Operation	DMS FullAccess	DMS UserAccess	DMS ReadOnlyAccess	DMS VPCAccess	DMS KMSAccess
Creating an instance	√	×	×	×	×
Modifying instances	√	×	×	×	×
Deleting instances	√	×	×	×	×
Modifying instance specifications	√	×	×	×	×
Querying instance information	√	√	√	×	×

Helpful Links

- **What Is IAM?**
- **Creating a User and Granting DMS for RocketMQ Permissions**
- **Permissions Policies and Supported Actions**

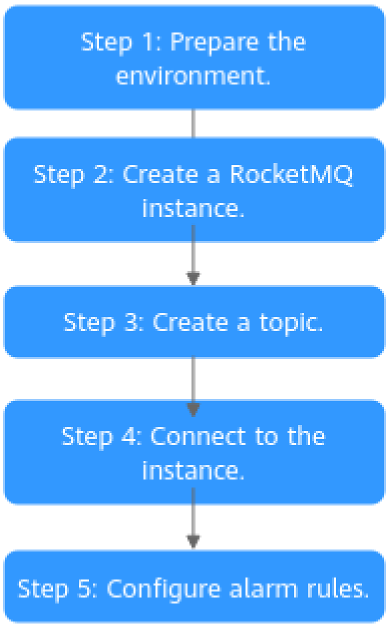
2 Getting Started

2.1 Introduction

This document takes the example of creating and connecting to a RocketMQ instance with SSL enabled to get you quickly started with Distributed Message Service (DMS) for RocketMQ.

Procedure

Figure 2-1 Procedure for using DMS for RocketMQ


Step 1: Prepare the environment.
Step 2: Create a RocketMQ instance.
Step 3: Create a topic.
Step 4: Connect to the instance.
Step 5: Configure alarm rules.

1. **Prepare the environment**.

   A RocketMQ instance runs in a Virtual Private Cloud (VPC). Before creating an instance, ensure that a VPC is available.

2. **Create a RocketMQ instance**.

You can select the specification and quantity and enable SSL when creating a RocketMQ instance. Enabling SSL secures data transmission with encryption.

3. **Create a topic**.

   After an instance is created, create a topic for sending and receiving messages.

4. **Connect to the instance**.

   On the client, connect to the instance and use commands to create and retrieve messages.

5. **Configure alarm rules**.

   Configure alarm rules for the RocketMQ instance to monitor the service running status.

NOTE

For the basic concepts of RocketMQ, see **RocketMQ Concepts**.

2.2 Step 1: Prepare the Environment

VPC

A VPC provides an isolated virtual network for your RocketMQ instances. You can configure and manage the network as required.

Step 1 Before creating a RocketMQ instance, ensure that a VPC and a subnet are available.

For details, see section "Creating a VPC" in *Virtual Private Cloud User Guide*. If you already have an available VPC and subnet, you do not need to create new ones.

Note the following when creating a VPC and subnet:

- The VPC and the RocketMQ instance must be in the same region.
- Use the default settings when creating a VPC and subnet.

Step 2 Before creating a RocketMQ instance, ensure that a security group is available.

For details, see section "Creating a Security Group" in *Virtual Private Cloud User Guide*. If you already have an available security group, you do not need to create a new one.

To use DMS for RocketMQ, add the security group rules described in **Table 2-1**. You can also add other rules based on site requirements.

Table 2-1 Security group rules

Direction	Protocol	Port	Source	Description
Inbound	TCP	8100	0.0.0.0/0	The port is used for intra-VPC access to metadata nodes.
Inbound	TCP	10100-10199	0.0.0.0/0	The port is used for accessing service nodes.

NOTE

After a security group is created, it has a default inbound rule that allows communication among ECSs within the security group and a default outbound rule that allows all outbound traffic. If you access your RocketMQ instance within a VPC, you do not need to add the rules described in **Table 2-1**.

----End

ECS

Before connecting to a RocketMQ instance, ensure that you have purchased an ECS, installed the JDK, and configured environment variables. The following takes a Linux ECS as an example. For more information about how to install JDK and configure the environment variables for a Windows ECS, please search the Internet.

Step 1 Log in to the management console, click ☰ in the upper left corner, click **Elastic Cloud Server** under **Computing**, and then create an ECS.

For details, see **Creating an ECS**. If you already have an available ECS, skip this step.

Step 2 Log in to the ECS.

Step 3 Install JDK or JRE, and add the following contents to **.bash_profile** in the home directory to configure the environment variables **JAVA_HOME** and **PATH**: In this command, **/opt/java/jdk1.8.0_151** is the JDK installation path. Change it to the path where you install JDK or JRE.

```
export JAVA_HOME=/opt/java/jdk1.8.0_151
export PATH=$JAVA_HOME/bin:$PATH
```

Run the **source .bash_profile** command for the modification to take effect.

NOTE

Use Oracle JDK instead of ECS's default JDK (for example, OpenJDK), because ECS's default JDK may not be suitable for the sample project. Obtain Oracle JDK 1.8.111 or later from **Oracle's official website**.

----End

2.3 Step 2: Create a RocketMQ Instance

This section takes the example of creating a RocketMQ v4.8.0 instance to describe how to create a RocketMQ instance on the console.

Prerequisites

Before creating a RocketMQ instance, ensure that a VPC and a security group are available. If no VPC or security group is available, create them by referring to **Step 1: Prepare the Environment**.

Procedure

Step 1 Log in to the DMS for RocketMQ console and click **Buy Instance** in the upper right.

Step 2 Set **Billing Mode** to **Pay-per-use**.

Step 3 Select a region closest to your application to reduce latency and accelerate access.

Step 4 Select a project from the drop-down list.

Step 5 Select one AZ or at least three AZs.

Step 6 Specify the **Instance Name** and the **Enterprise Project**.

Step 7 Set the instance information. For details, see **Table 2-2**.

Table 2-2 Setting instance information

Parameter	Description
Version	Select **4.8.0**. Fixed once the instance is created. Use the same version as your client.
Architecture	Select **Cluster**.
CPU Architecture	Select **x86**.
Broker Flavor	Select **rocketmq.4u8g.cluster**.
Brokers	Enter **1**.
Storage Space per Broker	Select **Ultra-high I/O** and enter **300** GB. Total storage space of an instance = Storage space per broker × Number of brokers

Step 8 Configure the instance network. For details, see **Table 2-3**.

Table 2-3 Configuring instance network

Parameter	Description
VPC	Select the created VPC and subnet. After the RocketMQ instance is created, its VPC and subnet cannot be changed.
Security Group	Select the created security group.
SSL	Enable it.
ACL	Enable it.

Step 9 Click **Advanced Settings**. For more information, see **Table 2-4**.

Table 2-4 Advanced settings

Parameter	Description
Public Access	Do not enable it.
Tags	Skip it.
Description	Skip it.

Step 10 Click **Buy**.

Step 11 Confirm the instance information and submit the request.

Step 12 Return to the instance list and check whether the RocketMQ instance has been created.

It takes 3 to 15 minutes to create an instance. During this period, the instance status is **Creating**.

- If the instance is created successfully, its status changes to **Running**.
- If an instance fails to be created, view it in the **Instance Creation Failures** area and delete it. Then create a new one. If the instance creation fails again, contact customer service.

----End

2.4 Step 3: Create a Topic

A topic is the basic unit for sending and receiving messages. After creating a RocketMQ instance, you must manually create topics before creating and retrieving messages.

Prerequisites

You have **created a RocketMQ instance**.

Procedure

Step 1 Log in to the DMS for RocketMQ consoleconsole.

Step 2 Click a RocketMQ instance to go to the instance details page.

Step 3 In the navigation pane, choose **Topics**.

Step 4 Click **Create Topic**.

Step 5 Configure the topic name and other parameters by referring to **Table 2-5**.

Table 2-5 Topic parameters

Parameter	Description
Topic Name	Name of the topic. The topic name must be unique. Otherwise, the topic cannot be created. Once the topic is created, you cannot modify its name. **NOTE** A percent (%) or vertical bar (|) contained in a topic name will be converted to an underscore (_) by Cloud Eye. For example, if a topic name is **test%01**, it will be displayed as **test_01** on Cloud Eye.
Permission	Topic permission, which can be **publish/subscribe**, **publish**, or **subscribe**.
Brokers	Specify the broker to create the topic on, and the number of queues in the topic. If the instance is deployed on multiple brokers, click **Add** to add more brokers and set the number of queues in the topic.

Step 6 Click **OK**.

----End

2.5 Step 4: Connect to a RocketMQ Instance to Create and Retrieve Messages

This section describes how to use TCP to connect to a RocketMQ instance with SSL in CLI mode.

Prerequisites

- A RocketMQ instance has been created following the instructions in **Step 2: Create a RocketMQ Instance**, and the instance addresses have been recorded.
- **Security group rules** have been configured.
- **A topic has been created** and you have obtained the topic name.
- An ECS has been created. For intra-VPC access, ensure that its VPC, subnet, and security group configurations are the same as those of the RocketMQ instance.
- You have installed the JDK and configured the environment variables. For details, see **Step 1: Prepare the Environment**.

Accessing the Instance with CLI

Step 1 Download the **rocketmq-tutorial** software package.

```
wget https://dms-demo.obs.cn-north-1.myhuaweicloud.com/rocketmq-tutorial.zip
```

Step 2 Decompress the **rocketmq-tutorial** package.

```
unzip rocketmq-tutorial.zip
```

Step 3 Go to the **rocketmq-tutorial/bin** directory.

```
cd rocketmq-tutorial/bin
```

Step 4 Create normal messages using the sample project.

```
JAVA_OPT=-Dtls.enable=true sh mqadmin sendMessage -n "${Connection addresses}" -t ${Topic name} -p "hello rocketmq"
```

Parameter description:

- **Connection addresses**: the RocketMQ instance address.
- **Topic name**: name of the topic created for the RocketMQ instance

In the following example, **100.xxx.xxx.89:8200;100.xxx.xxx.144:8200** are the connection addresses to the RocketMQ instance, and **topic-test** is the topic name.

```
JAVA_OPT=-Dtls.enable=true sh mqadmin sendMessage -n "100.xxx.xxx.89:8200;100.xxx.xxx.144:8200" -t topic-test -p "hello rocketmq"
```

Press **Ctrl**+**C** to exit.

Step 5 Retrieve normal messages using the sample project.

```
JAVA_OPT=-Dtls.enable=true sh mqadmin consumeMessage -n "${Connection addresses}" -t ${Topic name}
```

Parameter description:

- **Connection addresses**: the RocketMQ instance address.
- **Topic name**: name of the topic created for the RocketMQ instance

In the following example, **100.xxx.xxx.89:8200;100.xxx.xxx.144:8200** are the connection addresses to the RocketMQ instance, and **topic-test** is the topic name.

```
JAVA_OPT=-Dtls.enable=true sh mqadmin consumeMessage -n "100.xxx.xxx.89:8200;100.xxx.xxx.144:8200" -t topic-test
```

To stop consuming messages, press **Ctrl**+**C** to exit.

Step 6 Create messages with traces using the sample project.

```
JAVA_OPT=-Dtls.enable=true sh mqadmin sendMessage -n "${Connection addresses}" -t ${Topic name} -p "hello rocketmq" -m true
```

Parameter description:

- **Connection addresses**: the RocketMQ instance address.
- **Topic name**: name of the topic created for the RocketMQ instance

In the following example, **100.xxx.xxx.89:8200;100.xxx.xxx.144:8200** are the connection addresses to the RocketMQ instance, and **topic-test** is the topic name.

```
JAVA_OPT=-Dtls.enable=true sh mqadmin sendMessage -n "100.xxx.xxx.89:8200;100.xxx.xxx.144:8200" -t topic-test -p "hello rocketmq" -m true
```

Press **Ctrl**+**C** to exit.

Step 7 Retrieve messages and send the message traces using the sample project.

```
JAVA_OPT=-Dtls.enable=true sh mqadmin consumeMessage -n "${Connection addresses}" -t ${Topic name} -m true
```

Parameter description:

- **Connection addresses**: the RocketMQ instance address.

- **Topic name**: name of the topic created for the RocketMQ instance

In the following example, **100.xxx.xxx.89:8200;100.xxx.xxx.144:8200** are the connection addresses to the RocketMQ instance, and **topic-test** is the topic name.

```
JAVA_OPT=-Dtls.enable=true sh mqadmin consumeMessage -n "100.xxx.xxx.89:8200;100.xxx.xxx.144:8200" -
t topic-test -m true
```

Press **Ctrl**+**C** to exit.

----End

2.6 Step 5: Configure Alarm Rules

This section describes the alarm policies of some metrics and how to configure them. In actual services, you are advised to configure alarm rules for metrics based on the following alarm policies.

NOTE

Approach Upper Limit in the following table indicates whether the threshold is close to the upper limit of the performance supported by current resources. If the threshold is close to the upper limit and usage continues to rise, services may be abnormal.

Table 2-6 RocketMQ instance metrics to configure alarm rules for

Metric Name	Normal Range	Alarm Policy	Approach Upper Limit	Metric Description and Alarm Handling Suggestions
Disk Capacity Usage	0–100	Alarm threshold: Raw data > 85 Number of consecutive periods: 3 Alarm severity: Critical	Yes	**Metric description:** disk usage of the RocketMQ VM. Unit: % **Handling suggestion**: If an alarm is generated for this metric, the current instance specifications are insufficient to carry services. The **storage space** needs to be expanded by referring to **Modifying Specifications**.

Metric Name	Normal Range	Alarm Policy	Approach Upper Limit	Metric Description and Alarm Handling Suggestions
Average Load per CPU Core	0–2	Alarm threshold: Raw data > 1.5 Number of consecutive periods: 3 Alarm severity: Major	Yes	**Metric description:** average load of each CPU core of the RocketMQ VM. **Handling suggestion**: If an alarm is generated for this metric, perform the following operations: 1. Add brokers by referring to **Modifying Specifications**. 2. Redeploy existing topics to the new brokers by referring to **Modifying Topic Parameters**.
Memory Usage	0–100	Alarm threshold: Raw data > 85 Number of consecutive periods: 3 Alarm severity: Critical	Yes	**Metric description:** memory usage of the RocketMQ VM. Unit: % **Handling suggestion**: If an alarm is generated for this metric, perform the following operations: 1. Add brokers by referring to **Modifying Specifications**. 2. Redeploy existing topics to the new brokers by referring to **Modifying Topic Parameters**.

Procedure

Step 1 Log in to the console.

Step 2 Click 📍 in the upper left corner to select a region.

NOTE

Select the same region as your application service.

Step 3 Click ☰ and choose **Application** > **Distributed Message Service (for RocketMQ)** to open the console of DMS for RocketMQ.

Step 4 Click **View Metric** next to the RocketMQ instance name to go to the instance monitoring page of the Cloud Eye console.

Step 5 Hover the mouse pointer over a metric and click + to create an alarm rule for the metric.

The **Create Alarm Rule** page is displayed.

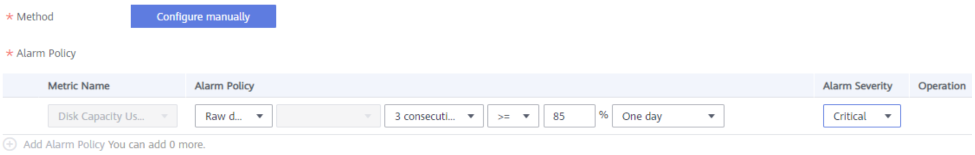
Step 6 Specify the alarm rule details.

For details about how to create alarm rules, see **Creating an Alarm Rule**.

1. Enter the alarm name and description.
2. Specify the alarm policy and alarm severity.

   As shown in the following figure, if the original disk capacity usage exceeds 85% for three consecutive periods, an alarm is generated. If the alarm is not handled on time, an alarm notification is sent.

   Figure 2-2 Setting the alarm policy and alarm severity

   * Method: Configure manually

   * Alarm Policy

Metric Name	Alarm Policy						Alarm Severity	Operation
Disk Capacity Us...	Raw d...		3 consecuti...	>=	85 %	One day	Critical	

   ⊕ Add Alarm Policy You can add 0 more.

3. Set the alarm notification configurations. If you enable **Alarm Notification**, set the validity period, notification object, and trigger condition.
4. Click **Create**.

----End

3 Permissions Management

3.1 Creating a User and Granting DMS for RocketMQ Permissions

This section describes how to use Identity and Access Management (IAM) for fine-grained permissions control for your Distributed Message Service (DMS) for RocketMQ resources. With IAM, you can:

- Create IAM users for personnel based on your enterprise's organizational structure. Each IAM user has their own identity credentials for accessing DMS for RocketMQ resources.
- Grant users only the permissions required to perform a given task based on their job responsibilities.
- Entrust an account or a cloud service to perform efficient O&M on your DMS for RocketMQ resources.

If your account meets your permissions requirements, you can skip this section.

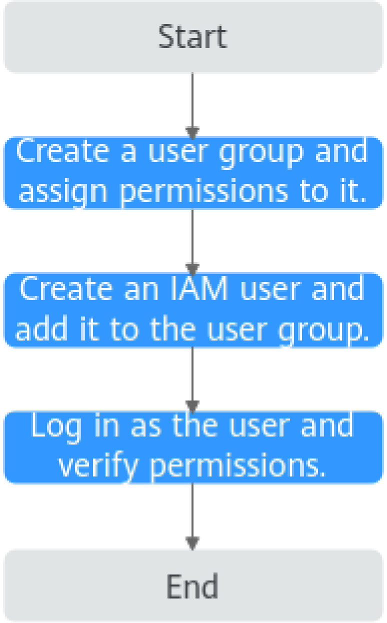
This section describes the procedure for granting permissions (see **Figure 3-1**).

Prerequisites

Learn about the permissions (see **Permissions Management**) supported by DMS for RocketMQ and choose policies or roles according to your requirements. For the system policies of other services, see **System Permissions**.

Process Flow

Figure 3-1 Process for granting DMS for RocketMQ permissions


Start
Create a user group and assign permissions to it.
Create an IAM user and add it to the user group.
Log in as the user and verify permissions.
End

1. On the IAM console, **create a user group and grant it permissions**.

   DMS ReadOnlyAccess is used as an example.

2. **Create an IAM user and add it to the created user group**.
3. **Log in as the IAM user** and verify permissions.

   In the authorized region, perform the following operations:

   - Choose **Service List** > **Distributed Message Service for RocketMQ**. Then click **Buy Instance** on the console of DMS for RocketMQ. If a message appears indicating that you have insufficient permissions to perform the operation, the **DMS ReadOnlyAccess** policy is in effect.
   - Choose **Service List** > **Elastic Volume Service**. If a message appears indicating that you have insufficient permissions to access the service, the **DMS ReadOnlyAccess** policy is in effect.

3.2 DMS for RocketMQ Custom Policies

Custom policies can be created to supplement the system-defined policies of DMS for RocketMQ. For the actions that can be added to custom policies, see **Permissions Policies and Supported Actions**.

You can create custom policies in either of the following ways:

- Visual editor: Select cloud services, actions, resources, and request conditions. This does not require knowledge of policy syntax.
- JSON: Edit JSON policies from scratch or based on an existing policy.

For details, see **Creating a Custom Policy**. The following section contains examples of common DMS for RocketMQ custom policies.

NOTE

DMS for RocketMQ permissions policies are based on DMS. Therefore, when assigning permissions, select DMS permissions policies.

Example Custom Policies

- Example 1: Allowing users to delete and restart instances

```
{
    "Version": "1.1",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "dms:instance:modifyStatus",
                "dms:instance:delete"
            ]
        }
    ]
}
```

- Example 2: Denying instance deletion

  A policy with only "Deny" permissions must be used in conjunction with other policies to take effect. If the permissions assigned to a user include both "Allow" and "Deny", the "Deny" permissions take precedence over the "Allow" permissions.

  The following method can be used if you need to assign permissions of the **DMS FullAccess** policy to a user but you want to prevent the user from deleting instances. Create a custom policy for denying instance deletion, and attach both policies to the group to which the user belongs. Then, the user can perform all operations on DMS for RocketMQ except deleting instances. The following is an example of a deny policy:

```
{
        "Version": "1.1",
        "Statement": [
                {
                        "Effect": "Deny",
                        "Action": [
                                "dms:instance:delete"
                        ]
                }
        ]
}
```

4 Preparing Required Resources

Overview

Before purchasing a RocketMQ instance, prepare the required resources, including a virtual private cloud (VPC), subnet, and security group with proper rules. Each RocketMQ instance is deployed in a VPC and bound to a specific subnet and security group, which provide an isolated virtual network environment and allow you to easily configure and manage security protection policies.

Required Resources

Table 4-1 lists the resources required by a RocketMQ instance.

Table 4-1 RocketMQ resources

Resource	Requirement	Operations
VPC and subnet	Different RocketMQ instances can use the same or different VPCs and subnets based on site requirements. Note the following when creating a VPC and subnet: ● The VPC and the RocketMQ instance must be in the same region. ● Use the default settings when creating a VPC and subnet.	For details about how to create a VPC and subnet, see the *Virtual Private Cloud User Guide*.

Resource	Requirement	Operations
Security group	Different RocketMQ instances can use the same security group or different security groups. To use RocketMQ instances, add the security group rules described in **Table 4-2**. You can also add other rules based on site requirements. **NOTE** After a security group is created, its default inbound rule allows communication among ECSs within the security group and its default outbound rule allows all outbound traffic. In this case, you can access a RocketMQ instance within a VPC, and do not need to add rules according to **Table 4-2**.	For details about how to create a security group and configure security group rules, see the *Virtual Private Cloud User Guide*.
EIP	Note the following when creating EIPs: ● The EIPs must be created in the same region as the RocketMQ instance. ● **The RocketMQ console cannot identify IPv6 EIPs.**	For details about how to create an EIP, see "Assigning an EIP" in *Elastic IP User Guide*.

Table 4-2 Security group rules

Direction	Protocol	Port	Source	Description
Inbound	TCP	8100	0.0.0.0/0	The port is used for intra-VPC access to metadata nodes.
Inbound	TCP	8200	0.0.0.0/0	The port is used for public access to metadata nodes.
Inbound	TCP	10100-10199	0.0.0.0/0	The port is used for accessing service nodes.

5 Buying an Instance

Scenario

Your RocketMQ instance will be deployed in physical isolation and exclusively used by you. You can customize the computing capabilities and storage space of an instance based on service requirements.

Prerequisites

You have **prepared required resources**.

Procedure

Step 1 Log in to the management console.

Step 2 Click 📍 in the upper left corner to select a region.

> NOTE
>
> Select the region your application is in.

Step 3 Click ☰ and choose **Application** > **Distributed Message Service for RocketMQ** to open the console of DMS for RocketMQ.

Step 4 Click **Buy RocketMQ Instance** in the upper right corner of the page.

Step 5 Specify **Billing Mode**, **Region**, **Project**, and **AZ**.

Step 6 Configure the following instance parameters:

1. **Instance Name**: Enter a name for the instance.
2. **Enterprise Project**: Select an enterprise project based on service requirements.
3. **Version**: Only 4.8.0 is available.
4. **Architecture**: Retain the default value.
5. **CPU Architecture**: Currently, only x86 architecture is supported. Retain the default value.
6. **Broker Flavor**: Select the required flavor.

7. **Brokers**: Select the required number of brokers.
8. **Storage Space per Broker**: Disk type and total storage space of each broker.

   The disk type can be high I/O or ultra-high I/O. For details on how to select a disk type, see "Disk Types and Performance" in *Elastic Volume Service User Guide*.

Step 7 Configure the instance network parameters.

1. Select a VPC and a subnet.

   A VPC provides an isolated virtual network for your RocketMQ instances. You can configure and manage the network.

   NOTE

   After the RocketMQ instance is created, its VPC and subnet cannot be changed.

2. Select a security group.

   A security group is a set of rules for accessing a RocketMQ instance.

Step 8 Configure SSL.

This parameter indicates whether to enable SSL authentication when a client connects to the RocketMQ instance. If **SSL** is enabled, data will be encrypted before transmission for enhanced security.

This setting cannot be changed after the instance is created. If you want to use a different SSL setting after the instance is created, you must create another instance.

Step 9 Configure ACL.

After ACL is enabled, the permissions for each user are exclusive.

Step 10 Click **More Settings** to configure more parameters.

- Configure **Public Access**.

  Public access is disabled by default. You can enable or disable it as required. After public access is enabled, configure an IPv4 EIP for each broker.

- Specify tags.

  Tags are used to identify cloud resources. When you have many cloud resources of the same type, you can use tags to classify cloud resources by dimension (for example, usage, owner, or environment).

  - If you have created predefined tags, select a predefined pair of tag key and value. To view or create predefined tags, click **View predefined tags** on the right. You will then be directed to the TMS console.
  - You can also create new tags by entering **Tag key** and **Tag value**.

  Up to 20 tags can be added to each RocketMQ instance..

- Enter a description of the instance.

Step 11 Click **Buy**.

Step 12 Confirm the instance information, and click **Submit**.

Step 13 Return to the instance list and check whether the instance has been created.

It takes 3 to 15 minutes to create an instance. During this period, the instance status is **Creating**.

- If the instance is created successfully, its status changes to **Running**.
- If an instance fails to be created, view it in the **Instance Creation Failures** area, and delete it by referring to **Deleting an Instance** and then create a new one. If the instance creation fails again, contact customer service.

----End

6 Accessing an Instance

6.1 Accessing a RocketMQ Instance Without SSL Enabled (TCP)

This section describes how to use TCP to connect to a RocketMQ instance without SSL in CLI mode.

Intra-VPC access and public access differ only in the connection IP addresses and ports. For intra-VPC access, use port 8100. For public access, use port 8200.

The following describes only the procedure for public access. For intra-VPC access, replace the IP addresses with the actual ones.

Prerequisites

- A RocketMQ instance has been created and you have obtained the connection addresses for intra-VPC access or public network access.
- **Security group rules** have been configured.
- **A topic** has been created.
- An ECS has been created. For intra-VPC access, ensure that its VPC, subnet, and security group configurations are the same as those of the RocketMQ instance.
- **JDK v1.8.111 or later** has been installed, and related environment variables have been configured.

Accessing the Instance with CLI

Step 1 Download the **rocketmq-tutorial** software package.

```
wget https://dms-demo.obs.cn-north-1.myhuaweicloud.com/rocketmq-tutorial.zip
```

Step 2 Decompress the **rocketmq-tutorial** package.

```
unzip rocketmq-tutorial.zip
```

Step 3 Go to the **rocketmq-tutorial/bin** directory.

```
cd rocketmq-tutorial/bin
```

Step 4 Create normal messages using the sample project.

```
sh mqadmin sendMessage -n "${Connection addresses}" -t ${Topic name} -p "hello rocketmq"
```

Parameter description:

- **Connection addresses**: the **Instance Address** for private network access or **Instance Address (Public Network)** for public network access.
- **Topic name**: name of the topic created for the RocketMQ instance

In the following example, **100.xxx.xxx.89:8200;100.xxx.xxx.144:8200** are the metadata connection addresses for public network access to the RocketMQ instance, and **topic-test** is the topic name.

```
sh mqadmin sendMessage -n "100.xxx.xxx.89:8200;100.xxx.xxx.144:8200" -t topic-test -p "hello rocketmq"
```

Press **Ctrl**+**C** to exit.

Step 5 Retrieve normal messages using the sample project.

```
sh mqadmin consumeMessage -n "${Connection addresses}" -t ${Topic name}
```

Parameter description:

- **Connection addresses**: the **Instance Address** for private network access or **Instance Address (Public Network)** for public network access.
- **Topic name**: name of the topic created for the RocketMQ instance

In the following example, **100.xxx.xxx.89:8200;100.xxx.xxx.144:8200** are the metadata connection addresses for public network access to the RocketMQ instance, and **topic-test** is the topic name.

```
sh mqadmin consumeMessage -n "100.xxx.xxx.89:8200;100.xxx.xxx.144:8200" -t topic-test
```

To stop consuming messages, press **Ctrl**+**C** to exit.

Step 6 Create messages with traces using the sample project.

```
sh mqadmin sendMessage -n "${Connection addresses}" -t ${Topic name} -p "hello rocketmq" -m true
```

Parameter description:

- **Connection addresses**: the **Instance Address** for private network access or **Instance Address (Public Network)** for public network access.
- **Topic name**: name of the topic created for the RocketMQ instance

In the following example, **100.xxx.xxx.89:8200;100.xxx.xxx.144:8200** are the metadata connection addresses for public network access to the RocketMQ instance, and **topic-test** is the topic name.

```
sh mqadmin sendMessage -n "100.xxx.xxx.89:8200;100.xxx.xxx.144:8200" -t topic-test -p "hello rocketmq" -
m true
```

Press **Ctrl**+**C** to exit.

Step 7 Retrieve messages and send the message traces using the sample project.

```
sh mqadmin consumeMessage -n "${Connection addresses}" -t ${Topic name} -m true
```

Parameter description:

- **Connection addresses**: the **Instance Address** for private network access or **Instance Address (Public Network)** for public network access.
- **Topic name**: name of the topic created for the RocketMQ instance

In the following example, **100.xxx.xxx.89:8200;100.xxx.xxx.144:8200** are the metadata connection addresses for public network access to the RocketMQ instance, and **topic-test** is the topic name.

```
sh mqadmin consumeMessage -n "100.xxx.xxx.89:8200;100.xxx.xxx.144:8200" -t topic-test -m true
```

Press **Ctrl**+**C** to exit.

----End

6.2 Accessing a RocketMQ Instance with SSL Enabled (TCP)

If SSL is enabled, data will be encrypted before transmission for enhanced security. This section describes how to use TCP to connect to a RocketMQ instance with SSL in CLI mode.

Intra-VPC access and public access differ only in the connection IP addresses and ports. For intra-VPC access, use port 8100. For public access, use port 8200.

The following describes only the procedure for public access. For intra-VPC access, replace the IP addresses with the actual ones.

Prerequisites

- A RocketMQ instance has been created and you have obtained the connection addresses for intra-VPC access or public network access.
- **Security group rules** have been configured.
- **A topic** has been created.
- An ECS has been created. For intra-VPC access, ensure that its VPC, subnet, and security group configurations are the same as those of the RocketMQ instance.
- **JDK v1.8.111 or later** has been installed, and related environment variables have been configured.

Accessing the Instance with CLI

Step 1 Download the **rocketmq-tutorial** software package.

```
wget https://dms-demo.obs.cn-north-1.myhuaweicloud.com/rocketmq-tutorial.zip
```

Step 2 Decompress the **rocketmq-tutorial** package.

```
unzip rocketmq-tutorial.zip
```

Step 3 Go to the **rocketmq-tutorial/bin** directory.

```
cd rocketmq-tutorial/bin
```

Step 4 Create normal messages using the sample project.

```
JAVA_OPT=-Dtls.enable=true sh mqadmin sendMessage -n "${Connection addresses}" -t ${Topic name} -p
"hello rocketmq"
```

Parameter description:

- **Connection addresses**: the **Instance Address** for private network access or **Instance Address (Public Network)** for public network access.

- **Topic name**: name of the topic created for the RocketMQ instance

In the following example, **100.xxx.xxx.89:8200;100.xxx.xxx.144:8200** are the metadata connection addresses for public network access to the RocketMQ instance, and **topic-test** is the topic name.

```
JAVA_OPT=-Dtls.enable=true sh mqadmin sendMessage -n "100.xxx.xxx.89:8200;100.xxx.xxx.144:8200" -t
topic-test -p "hello rocketmq"
```

Press **Ctrl**+**C** to exit.

Step 5 Retrieve normal messages using the sample project.

```
JAVA_OPT=-Dtls.enable=true sh mqadmin consumeMessage -n "${Connection addresses}" -t ${Topic name}
```

Parameter description:

- **Connection addresses**: the **Instance Address** for private network access or **Instance Address (Public Network)** for public network access.
- **Topic name**: name of the topic created for the RocketMQ instance

In the following example, **100.xxx.xxx.89:8200;100.xxx.xxx.144:8200** are the metadata connection addresses for public network access to the RocketMQ instance, and **topic-test** is the topic name.

```
JAVA_OPT=-Dtls.enable=true sh mqadmin consumeMessage -n "100.xxx.xxx.89:8200;100.xxx.xxx.144:8200" -
t topic-test
```

To stop consuming messages, press **Ctrl**+**C** to exit.

Step 6 Create messages with traces using the sample project.

```
JAVA_OPT=-Dtls.enable=true sh mqadmin sendMessage -n "${Connection addresses}" -t ${Topic name} -p
"hello rocketmq" -m true
```

Parameter description:

- **Connection addresses**: the **Instance Address** for private network access or **Instance Address (Public Network)** for public network access.
- **Topic name**: name of the topic created for the RocketMQ instance

In the following example, **100.xxx.xxx.89:8200;100.xxx.xxx.144:8200** are the metadata connection addresses for public network access to the RocketMQ instance, and **topic-test** is the topic name.

```
JAVA_OPT=-Dtls.enable=true sh mqadmin sendMessage -n "100.xxx.xxx.89:8200;100.xxx.xxx.144:8200" -t
topic-test -p "hello rocketmq" -m true
```

Press **Ctrl**+**C** to exit.

Step 7 Retrieve messages and send the message traces using the sample project.

```
JAVA_OPT=-Dtls.enable=true sh mqadmin consumeMessage -n "${Connection addresses}" -t ${Topic name}
-m true
```

Parameter description:

- **Connection addresses**: the **Instance Address** for private network access or **Instance Address (Public Network)** for public network access.
- **Topic name**: name of the topic created for the RocketMQ instance

In the following example, **100.xxx.xxx.89:8200;100.xxx.xxx.144:8200** are the metadata connection addresses for public network access to the RocketMQ instance, and **topic-test** is the topic name.

```
JAVA_OPT=-Dtls.enable=true sh mqadmin consumeMessage -n "100.xxx.xxx.89:8200;100.xxx.xxx.144:8200" -t topic-test -m true
```

Press **Ctrl**+**C** to exit.

----End

7 Managing Instances

7.1 Modifying Specifications

Scenario

After creating a RocketMQ instance, you can increase its specifications. **Table 7-1** lists available modification options.

Table 7-1 Specification modification options

Modified Object	Increase	Decrease
Broker quantity	√	×
Storage space	√	×
Broker flavor	√	×

Constraints

- You can expand the storage space 20 times.
- When brokers are added, the storage space is proportionally expanded based on the current disk space. For example, assume that the original number of brokers of an instance is 1 and the disk size of each broker is 300 GB. If the broker quantity changes to 2 and the disk size of each broker is still 300 GB, the total disk size becomes 600 GB.
- **rocketmq.4u8g.cluster.small** does not support broker flavor increase.

Prerequisites

A RocketMQ instance has been created and is in the **Running** state.

Procedure

Step 1 Log in to the management console.

Step 2 Click 📍 in the upper left corner to select a region.

NOTE

Select the region where your RocketMQ instance is located.

Step 3 Click ☰ and choose **Application** > **Distributed Message Service for RocketMQ** to open the console of DMS for RocketMQ.

Step 4 Modify the instance specifications using either of the following methods:

- Locate the row that contains the desired instance, choose **More** > **Modify Specifications**.
- Click a RocketMQ instance to go to the instance details page. In the upper right corner, choose **More** > **Modify Specifications**.

Step 5 Specify the required storage space, broker quantity, or broker flavor.

- Expand the storage space.

  For **Modify By**, select **Storage**. For **Storage Space per Broker**, specify a new storage space, and click **Next**. Confirm the configurations and click **Submit**.

  View the new storage space (Storage space per broker x Number of brokers) in the **Used/Available Storage Space (GB)** column in the instance list.

  NOTE

  – Storage space expansion does not affect services.

  – Available storage space = Actual storage space – Storage space for storing logs – Disk formatting loss

    For example, if the storage space is expanded to 700 GB, the storage space for storing logs is 100 GB, and the disk formatting loss is 7 GB, then the available storage space after capacity expansion will be 593 GB.

- Add brokers.

  For **Modify By**, select **Brokers**. For **Brokers**, specify a new quantity, and click **Next**. Confirm the configurations and click **Submit**.

  View the number of brokers in the **Specifications** column in the instance list.

  NOTE

  – Adding brokers does not affect the original brokers or services.

  – If public access is enabled and EIPs are configured for the instance, configure EIPs for the new brokers.

- Increase the broker flavor.

  For **Modify By**, select **Broker Flavor**. Then, select a new flavor and click **Next**. Confirm the configurations and click **Submit**.

  View the broker flavor in the **Flavor** column of the instance list.

NOTE

Increasing the broker flavor will interrupt the instance for about one minute. If the production TPS is high during the change, the change may fail. To prevent this from happening, add brokers first.

----End

7.2 Viewing an Instance

Scenario

View detailed information about a RocketMQ instance on the console, for example, the IP address and port number for accessing the RocketMQ instance.

Procedure

Step 1 Log in to the management console.

Step 2 Click 📍 in the upper left corner to select a region.

NOTE

Select the region where your RocketMQ instance is located.

Step 3 Click ☰ and choose **Application** > **Distributed Message Service for RocketMQ** to open the console of DMS for RocketMQ.

Step 4 Search for a RocketMQ instance by specifying the filters. Current filters include the status, name, version, instance type, specification, used/available storage space, billing mode, enterprise project, and tag. For RocketMQ instance statuses, see **Table 7-2**.

Table 7-2 RocketMQ instance status description

Status	Description
Creating	The instance is being created.
Running	The instance is running properly. Only instances in the **Running** state can provide services.
Faulty	The instance is not running properly.
Changing	The public access configurations or instance specifications are being modified.
Change failed	The public access configurations or instance specifications failed to be modified.

Step 5 Click the name of the chosen RocketMQ instance and view the instance details on the page that is displayed.

Parameters for connecting to an instance are listed in **Table 7-3**. For details about other parameters, see the **Basic Information** tab page of the instance on the console.

Table 7-3 Connection information

Parameter	Description
Instance Address	Address for connecting to the instance when public access is disabled.
SSL	SSL status of the instance.
Public Access	Indicates whether public access is enabled. Click ✎ to enable or disable public access.
Instance Address (Public Network)	This parameter is displayed only when public access is enabled. Address for connecting to the instance when public access is enabled.
ACL	ACL status of the instance. Click ○ to enable ACL.

----End

7.3 Deleting an Instance

Scenario

With a few clicks on the console, you can delete one or multiple RocketMQ instances that have been created.

NOTICE

After a RocketMQ instance is deleted, the instance data will be deleted without backup.

Prerequisites

The RocketMQ instances you want to delete are in the **Running** or **Faulty** state.

Deleting a RocketMQ Instance

Step 1 Log in to the management console.

Step 2 Click 📍 in the upper left corner to select a region.

NOTE

Select the region where your RocketMQ instance is located.

Step 3 Click ☰ and choose **Application** > **Distributed Message Service for RocketMQ** to open the console of DMS for RocketMQ.

Step 4 Delete instances using any of the following methods:

- In the row containing the RocketMQ instance you want to delete, choose **More** > **Delete**.
- Click a RocketMQ instance to go to the instance details page. In the upper right corner, choose **More** > **Delete**.
- To delete multiple instances at a time, select the instances, and click **Delete** above the instance list.

NOTE

RocketMQ instances in the **Creating** state cannot be deleted.

Step 5 Click **Yes**.

It takes 1 to 60 seconds to delete a RocketMQ instance.

----End

7.4 Modifying Instance Information

Scenario

After purchasing a RocketMQ instance, you can modify parameters of the instance based on your service requirements, including the instance name, enterprise project, description, and security group.

Procedure

Step 1 Log in to the management console.

Step 2 Click 📍 in the upper left corner to select a region.

NOTE

Select the region where your RocketMQ instance is located.

Step 3 Click ☰ and choose **Application** > **Distributed Message Service for RocketMQ** to open the console of DMS for RocketMQ.

Step 4 Click a RocketMQ instance to go to the instance details page.

Step 5 Modify the following parameters if needed:

- Instance name
- Enterprise project
- Description
- Public access (To change the configuration, see **Configuring Public Access**.)
- Security group
- ACL

- Message aging time (To change the configuration, see **Changing Message Retention Period**.)

After the parameters are modified, view the modification result in the following ways:

After you change **Instance Name**, **Description**, **Public Access**, **ACL**, **Message Aging Time**, **Enterprise Project**, and **Security Group**, the result is displayed in the upper right corner.

----End

7.5 Configuring Public Access

To access a RocketMQ instance over a public network, enable public access and configure EIPs for the instance. If you no longer need public access to the instance, disable it.

Prerequisites

You can change the public access setting only when the RocketMQ instance is in the **Running** state.

Enabling Public Access

Step 1 Log in to the management console.

Step 2 Click in the upper left corner to select a region.

NOTE

Select the region where your RocketMQ instance is located.

Step 3 Click ☰ and choose **Application** > **Distributed Message Service for RocketMQ** to open the console of DMS for RocketMQ.

Step 4 Click a RocketMQ instance to go to the instance details page.

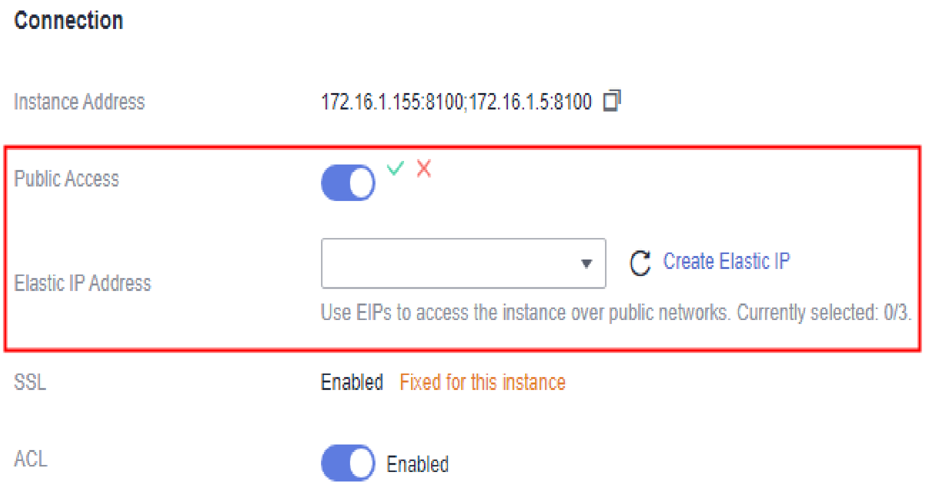
Step 5 Click ✎ next to **Public Access**.

Step 6 Click , in the **Elastic IP Address** area, select the required number of IP addresses, and click ✓.

NOTE

- Only IPv4 EIPs can be bound to RocketMQ instances.
- If no EIP exists in the **Elastic IP Address** drop-down list box, or the EIPs are insufficient, click **Create Elastic IP** to create an EIP on the page that is displayed. After the EIP is created, return to the RocketMQ console, click ⟳ next to **Elastic IP Address**, and select the new EIP from the drop-down list.

Figure 7-1 Enabling public access



Connection

Instance Address 172.16.1.155:8100;172.16.1.5:8100

Public Access ✓ ✗

Elastic IP Address Create Elastic IP

Use EIPs to access the instance over public networks. Currently selected: 0/3.

SSL Enabled Fixed for this instance

ACL Enabled

After public access is enabled, **Instance Address (Public Network)** is displayed for instances. .

After public access is enabled, modify security group rules before attempting to access the RocketMQ instance.

Table 7-4 Security group rules

Direction	Protocol	Port	Source	Description
Inbound	TCP	8200	0.0.0.0/0	The port is used for public access to metadata nodes.
Inbound	TCP	10100-10199	0.0.0.0/0	The port is used for accessing service nodes.

----End

Disabling Public Access

Step 1 Log in to the management console.

Step 2 Click in the upper left corner to select a region.

NOTE

Select the region where your RocketMQ instance is located.

Step 3 Click ☰ and choose **Application** > **Distributed Message Service for RocketMQ** to open the console of DMS for RocketMQ.

Step 4 Click a RocketMQ instance to go to the instance details page.

Step 5 Click ✎ next to **Public Access**.

Step 6 Click ● and then ✓ to disable public access.

After public access is disabled, modify security group rules before attempting to access the RocketMQ instance over the private network.

Table 7-5 Security group rules

Direction	Protocol	Port	Source	Description
Inbound	TCP	8100	0.0.0.0/0	The port is used for intra-VPC access to metadata nodes.
Inbound	TCP	10100-10199	0.0.0.0/0	The port is used for accessing service nodes.

----End

7.6 Restarting Brokers

Scenario

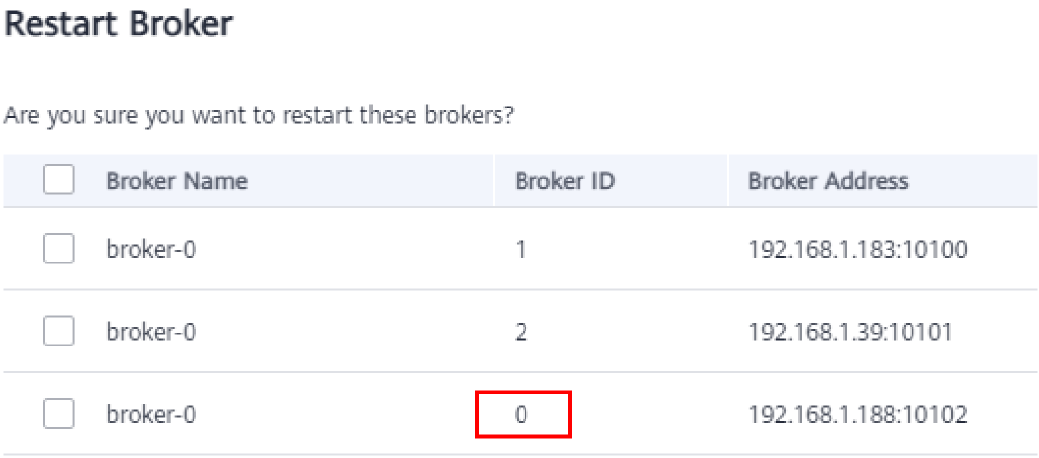
Restart one or more brokers of a RocketMQ instance at a time on the DMS for RocketMQ console.

During the restart, client requests for message retrieval and creation will be rejected for up to 30s. Restart brokers during off-peak hours. Restarting slave brokers does not affect services.

How do I know which broker is the master?

The **Broker ID** of the master broker is **0**.

Figure 7-2 Restarting brokers



Restart Broker

Are you sure you want to restart these brokers?

□	Broker Name	Broker ID	Broker Address
□	broker-0	1	192.168.1.183:10100
□	broker-0	2	192.168.1.39:10101
□	broker-0	0	192.168.1.188:10102

Prerequisite

The RocketMQ instance must be running.

Procedure

Step 1 Log in to the management console.

Step 2 Click 📍 in the upper left corner to select a region.

NOTE

Select the region where your RocketMQ instance is located.

Step 3 Click ☰ and choose **Application** > **Distributed Message Service for RocketMQ** to open the console of DMS for RocketMQ.

Step 4 In the row containing the desired RocketMQ instance, click **Restart Broker**.

Step 5 Select the brokers to be restarted and click **Yes**.

It takes 3 to 15 minutes to restart the brokers.

----End

7.7 Managing Instance Tags

Tags facilitate RocketMQ instance identification and management.

You can add tags to a RocketMQ instance when creating the instance or add tags on the **Tags** tab page of the created instance. Up to 20 tags can be added to an instance. Tags can be deleted.

A tag consists of a tag key and a tag value. Table 7-6 lists the tag key and value requirements.

Table 7-6 Tag key and value requirements

Parameter	Requirement
Tag key	● Cannot be left blank. ● Must be unique for the same instance. ● Can contain 1 to 128 characters. ● Can contain letters, digits, spaces, and special characters _.:=+-@ ● Cannot start or end with a space. ● Cannot start with **_sys_**.
Tag value	● Can contain 0 to 255 characters. ● Can contain letters, digits, spaces, and special characters _.:=+-@ ● Cannot start or end with a space.

Procedure

Step 1 Log in to the management console.

Step 2 Click 📍 in the upper left corner to select a region.

NOTE

Select the region where your RocketMQ instance is located.

Step 3 Click ☰ and choose **Application** > **Distributed Message Service for RocketMQ** to open the console of DMS for RocketMQ.

Step 4 Click the name of an instance.

Step 5 In the navigation pane on the left, choose **Tags**.

View the tags of the instance.

Step 6 Perform the following operations as required:

- Add a tag
  a. Click **Create/Delete Tag**.
  b. Enter a tag key and a tag value, and click **Add**.
     If you have predefined tags, select a predefined pair of tag key and value, and click **Add**.
  c. Click **OK**.
- Delete a tag
  Delete a tag using either of the following methods:
  – In the row containing the tag to be deleted, click **Delete**. Click **OK**.
  – Click **Create/Delete Tag**. In the dialog box that is displayed, click ⊗ next to the tag to be deleted and click **OK**.

----End

7.8 Exporting Instance Information

Scenario

Export the instance list from the console.

Procedure

Step 1 Log in to the management console.

Step 2 Click 📍 in the upper left corner to select a region.

NOTE

Select the region where your RocketMQ instance is located.

Step 3 Click ☰ and choose **Application** > **Distributed Message Service for RocketMQ** to open the console of DMS for RocketMQ.

Step 4 Click **Export** to export the instance list.

----End

8 Managing Topics

8.1 Creating a Topic

Scenario

A topic is the basic unit for sending and receiving messages. After purchasing a RocketMQ instance, you must manually create topics before creating and retrieving messages.

Prerequisites

A RocketMQ instance has been purchased.

Procedure

Step 1 Log in to the management console.

Step 2 Click 📍 in the upper left corner to select a region.

> **NOTE**
>
> Select the region where your RocketMQ instance is located.

Step 3 Click ☰ and choose **Application** > **Distributed Message Service for RocketMQ** to open the console of DMS for RocketMQ.

Step 4 Click a RocketMQ instance to go to the instance details page.

Step 5 In the navigation pane, choose **Topics**.

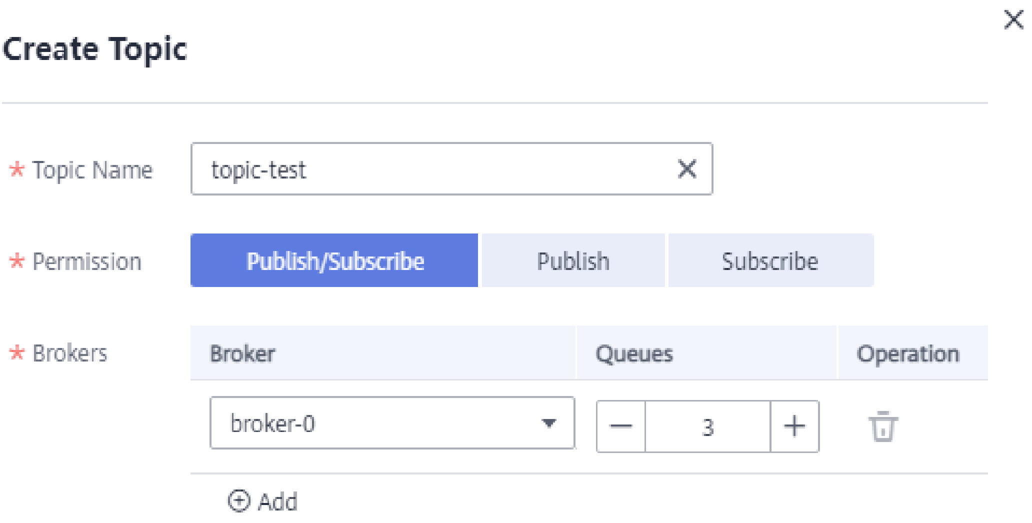
Step 6 Click **Create Topic**.

Step 7 Configure the topic name and other parameters by referring to **Table 8-1**.

Table 8-1 Topic parameters

Parameter	Description
Topic Name	Name of the topic. The topic name must be unique. Otherwise, the topic cannot be created. Once the topic is created, you cannot modify its name. **NOTE** A percent (%) or vertical bar (|) contained in a topic name will be converted to an underscore (_) by Cloud Eye. For example, if a topic name is **test%01**, it will be displayed as **test_01** on Cloud Eye.
Permission	Topic permission, which can be **publish/subscribe**, **publish**, or **subscribe**.
Brokers	Specify the broker to create the topic on, and the number of queues in the topic. If the instance is deployed on multiple brokers, click **Add** to add more brokers and set the number of queues in the topic.

Figure 8-1 Creating a topic



Create Topic

* Topic Name: topic-test

* Permission: Publish/Subscribe (selected) | Publish | Subscribe

* Brokers:

Broker	Queues	Operation
broker-0	3	

Step 8 Click **OK**.

----End

8.2 Querying Topic Details

Scenario

After a topic is created, you can query its configuration and status.

Procedure

Step 1 Log in to the management console.

Step 2 Click in the upper left corner to select a region.

NOTE

Select the region where your RocketMQ instance is located.

Step 3 Click and choose **Application** > **Distributed Message Service for RocketMQ** to open the console of DMS for RocketMQ.

Step 4 Click a RocketMQ instance to go to the instance details page.

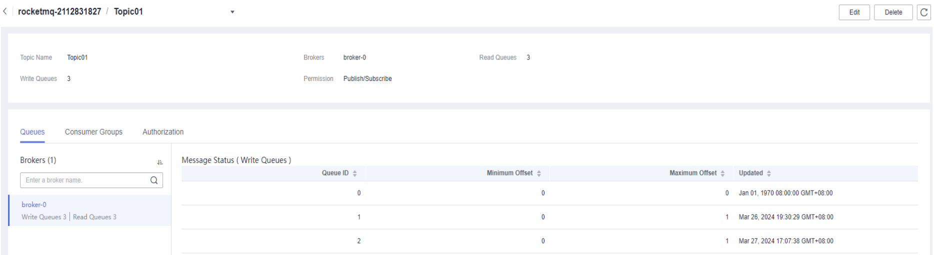
Step 5 In the navigation pane, choose **Topics**.

Step 6 Click the name of the topic to be queried, or locate the row that contains the desired topic, and choose **More** > **View Detail**. The topic details page is displayed.

The upper part of the topic details page shows the topic name, permissions, the number of brokers, read queues, and write queues.

The lower part of the topic details page shows queues on each broker, including the queue ID, minimum offset, maximum offset, and message update time. You can also view information about consumer groups that retrieve messages from the topic, including the consumer group name, maximum number of retries, and whether message broadcasting is enabled.

Figure 8-2 Topic details



Topic Name Topic01
Write Queues 3
Brokers broker-0
Permission Publish/Subscribe
Read Queues 3

broker-0
Write Queues 3 | Read Queues 3

Message Status (Write Queues)

Queue ID	Minimum Offset	Maximum Offset	Updated
0	0	0	Jan 01, 1970 08:00:00 GMT+08:00
1	0	1	Mar 26, 2024 19:30:29 GMT+08:00
2	0	1	Mar 27, 2024 17:07:38 GMT+08:00

----End

8.3 Modifying Topic Parameters

Scenario

After a topic is created, you can modify the following parameters based on service requirements: the number of read queues, number of write queues, topic permission, and associated brokers.

Procedure

Step 1 Log in to the management console.

Step 2 Click 📍 in the upper left corner to select a region.

NOTE

Select the region where your RocketMQ instance is located.

Step 3 Click ☰ and choose **Application** > **Distributed Message Service for RocketMQ** to open the console of DMS for RocketMQ.

Step 4 Click a RocketMQ instance to go to the instance details page.

Step 5 In the navigation pane, choose **Topics**.

Step 6 Modify topic parameters using either of the following methods:

- In the row containing the topic whose parameters you want to modify, click **Edit**.
- Click a topic to go to the topic details page and then click **Edit** in the upper right corner.

Step 7 Modify topic parameters by referring to **Table 8-2**.

Table 8-2 Topic parameters

Parameter	Description
Permission	Topic permission, which can be **publish/subscribe**, **publish**, or **subscribe**.
Brokers	Change the number of read queues or write queues. ● Read queues: total number of available queues in the topic for reading data. ● Write queues: total number of available queues in the topic for writing data. If an existing topic is not associated with all brokers during creation, click **Add** to distribute the topic to more brokers and set the number of read queues and write queues.

Step 8 Click **OK**.

----End

8.4 Deleting a Topic

Scenario

Delete topics if they are no longer needed.

Prerequisites

Topics can be deleted only when the instance is in the **Running** state.

Procedure

Step 1 Log in to the management console.

Step 2 Click ⌖ in the upper left corner to select a region.

NOTE

Select the region where your RocketMQ instance is located.

Step 3 Click ☰ and choose **Application** > **Distributed Message Service for RocketMQ** to open the console of DMS for RocketMQ.

Step 4 Click a RocketMQ instance to go to the instance details page.

Step 5 In the navigation pane, choose **Topics**.

Step 6 Delete topics using any of the following methods:

- In the row containing the topic you want to delete, choose **More** > **Delete**.
- Click a topic to go to the topic details page and then click **Delete** in the upper right corner.
- To delete multiple topics at a time, select the topics, and click **Delete Topic** above the topic list.

NOTICE

Exercise caution when deleting topics because data will be lost.

Step 7 Click **OK**.

----End

9 Managing Messages

9.1 Querying Messages

Scenario

This section describes how to view the content of a message and check whether the message is successfully created or retrieved.

On the DMS for RocketMQ console, you can query messages by topic, message ID, or message key.

- By topic: All messages of a specified topic within a specified period will be queried. This is range query and may find a large number of messages.
- By message ID: The message with the specified message ID will be queried. This is exact search and can quickly find a specific message.
- By message key: The message with the specified message key will be queried. This is exact search and can quickly find a specific message.

Message Deletion Mechanism

RocketMQ messages are retained for 48 hours (not modifiable), regardless of whether they have been consumed. RocketMQ messages are stored in CommitLog files. Each CommitLog file is 1 GB. When a CommitLog file is full, a new CommitLog file is generated. Message deletion in RocketMQ means to delete the CommitLog files, instead of individual messages. CommitLog files are written in sequence. A CommitLog file expires when the last message written in it expires. CommitLog files are deleted in the following scenarios:

- Expired files are cleared at 04:00 every day. Earlier instances that do not have a time zone are cleared at 12:00 every day.
- Expired files are deleted immediately when the disk usage reaches 70%.
- The earliest files are deleted, regardless of whether they have expired, when the disk usage reaches 85%.

Prerequisites

- A RocketMQ instance and topics have been created.
- To query messages by message ID, you need the name of the topic to which the message belongs and the message ID.
- To query messages by message key, you need the name of the topic to which the message belongs and the message key.
- To query message traces, you must first **enable message tracing**.

Querying the Message Content

Step 1 Log in to the management console.

Step 2 Click 📍 in the upper left corner to select a region.

NOTE

Select the region where your RocketMQ instance is located.

Step 3 Click ☰ and choose **Application** > **Distributed Message Service for RocketMQ** to open the console of DMS for RocketMQ.

Step 4 Click a RocketMQ instance to go to the instance details page.

Step 5 In the navigation pane, choose **Message Query**.

Step 6 Query messages in either of the following ways:

- By topic: Select the topic to be queried from the **Topic** drop-down list and the queue to the queried from the **Queue** drop-down list. For **Stored**, select a time period.
- By message ID: Select the name of the topic to be queried from the **Topic** drop-down list, enter the ID of the message to be queried, and click **Search**.
- By message key: Select the name of the topic to be queried from the **Topic** drop-down list, enter the key of the message to be queried, and click **Search**.

Step 7 In the row that contains the desired message, click **View Details** to view the message content.

NOTE

If the message body is greater than 4096 bytes, some content may not be displayed on the console. In this case, click **Download** to view the message body in the downloaded JSON file.

----End

Viewing the Message Trace

Step 1 Locate the row that contains the message to be queried. Click **Message Trace**.

Step 2 View the message trace to check whether the message is successfully produced or consumed.

Table 9-1 describes message trace parameters.

Table 9-1 Message trace parameters

Parameter	Description
Producer status	A producer can be in one of the following states: ● Sent: The message is sent successfully, and the server has successfully stored the message. ● Committed: The message can be retrieved by consumers. ● Rollback: The message will be discarded and cannot be retrieved by consumers. ● Unknown: The status of the message cannot be determined. After a period of time, the server initiates a check request to the producer.
Creation duration	Time taken to send the message by the producer.
Producer address	IP address and port of the producer.
Consumer status	A consumer can be in one of the following states: ● Retrieved ● Retrieval timed out ● Abnormal retrieval ● NULL returned ● Retrieval failed
Retrieved	Time when the message is retrieved.
Retrieval duration	Time taken to retrieve the message by the consumer.
Consumer address	IP address and port of the consumer.

----End

9.2 Enabling Message Tracing

Scenario

To query message traces, you must first enable message tracing on clients.

The following procedures describe how to enable message tracing in Java and Go.

Prerequisites

- Transactional message tracing is supported only if the producer Java client is v4.9.0 or later. If your client is earlier than v4.9.0, upgrade it to a later version.
- For RocketMQ instances with SSL enabled, message tracing is supported only when the producer and consumer Java clients are v4.9.2 or later. If the version does not meet the requirement, upgrade it first.

Procedure (Java)

Do as follows to enable message tracing on clients:

- Enabling message tracing on a producer client (**tracing messages other than transactional messages**)

  Set **enableMsgTrace** of the constructor to **true**. For example:

  ```
  DefaultMQProducer producer = new DefaultMQProducer("ProducerGroupName", true);
  ```

- Enabling message tracing on a producer client (**tracing transactional messages**)

  Set **enableMsgTrace** of the constructor to **true**. For example:

  ```
  TransactionMQProducer producer = new TransactionMQProducer(null, "ProducerGroupName", null,
  true, null);
  ```

- Enabling message tracing on a consumer

  Set **enableMsgTrace** of the constructor to **true**. For example:

  ```
  DefaultMQPushConsumer consumer = new DefaultMQPushConsumer("ConsumerGroupName", true);
  ```

Procedure (Go)

Do as follows to enable message tracing on clients:

Step 1 Run the following command to check whether Go has been installed:

```
go version
```

If the following information is displayed, Go has been installed.

```
[root@ecs-test sarama]# go version
go version go1.16.5 linux/amd64
```

If Go is not installed, **download** and install it.

Step 2 Create a **go.mod** file and add the following code to it to add the dependency:

```
module rocketmq-example-go

go 1.13

require (
    github.com/apache/rocketmq-client-go/v2 v2.1.0
)
```

Step 3 Enable message tracing on the producer. Replace the information in bold with the actual values.

```
package main

import (
    "context"
    "fmt"
    "os"
    "time"

    "github.com/apache/rocketmq-client-go/v2"
    "github.com/apache/rocketmq-client-go/v2/primitive"
    "github.com/apache/rocketmq-client-go/v2/producer"
)

func main() {
    namesrvs := []string{"192.168.0.1:8100"}
    traceCfg := &primitive.TraceConfig{
        Access:   primitive.Local,
        Resolver: primitive.NewPassthroughResolver(namesrvs),
```

```
    }

    p, _ := rocketmq.NewProducer(
        producer.WithNsResolver(primitive.NewPassthroughResolver([]string{"192.168.0.1:8100"})),
        producer.WithRetry(2),
        producer.WithTrace(traceCfg))
    err := p.Start()
    if err != nil {
        fmt.Printf("start producer error: %s", err.Error())
        os.Exit(1)
    }
    res, err := p.SendSync(context.Background(), primitive.NewMessage("topic1",
        []byte("Hello RocketMQ Go Client!")))

    if err != nil {
        fmt.Printf("send message error: %s\n", err)
    } else {
        fmt.Printf("send message success: result=%s\n", res.String())
    }

    time.Sleep(10 * time.Second)

    err = p.Shutdown()
    if err != nil {
        fmt.Printf("shutdown producer error: %s", err.Error())
    }
}
```

Step 4 Enable message tracing on the consumer. Replace the information in bold with the actual values.

```
package main

import (
    "context"
    "fmt"
    "os"
    "time"

    "github.com/apache/rocketmq-client-go/v2"
    "github.com/apache/rocketmq-client-go/v2/consumer"
    "github.com/apache/rocketmq-client-go/v2/primitive"
)

func main() {
    namesrvs := []string{"192.168.0.1:8100"}
    traceCfg := &primitive.TraceConfig{
        Access:   primitive.Local,
        Resolver: primitive.NewPassthroughResolver(namesrvs),
    }

    c, _ := rocketmq.NewPushConsumer(
        consumer.WithGroupName("testGroup"),
        consumer.WithNsResolver(primitive.NewPassthroughResolver([]string{"192.168.0.1:8100"})),
        consumer.WithTrace(traceCfg),
    )
    err := c.Subscribe("TopicTest", consumer.MessageSelector{}, func(ctx context.Context,
        msgs ...*primitive.MessageExt) (consumer.ConsumeResult, error) {
        fmt.Printf("subscribe callback: %v \n", msgs)
        return consumer.ConsumeSuccess, nil
    })
    if err != nil {
        fmt.Println(err.Error())
    }
    // Note: start after subscribe
    err = c.Start()
    if err != nil {
        fmt.Println(err.Error())
        os.Exit(-1)
    }
    time.Sleep(time.Hour)
    err = c.Shutdown()
    if err != nil {
        fmt.Printf("shutdown Consumer error: %s", err.Error())
    }
}
```

----End

9.3 Verifying Consumption

Scenario

Consumption verification is to resend messages to a specified online client to verify that the consumer can receive messages. Consumption verification causes repeated message consumption.

Prerequisites

- Consumption can be verified only when an instance is in the **Running** state.
- Ensure that the client for which consumption is to be verified is online and the consumer has subscribed to the topic whose message is to be resent.

Procedure

Step 1 Log in to the management console.

Step 2 Click 📍 in the upper left corner to select a region.

📖 **NOTE**

Select the region where your RocketMQ instance is located.

Step 3 Click ☰ and choose **Application** > **Distributed Message Service for RocketMQ** to open the console of DMS for RocketMQ.

Step 4 Click a RocketMQ instance to go to the details page.

Step 5 In the navigation pane, choose **Message Query**.

Step 6 Set search criteria on the top of the page.

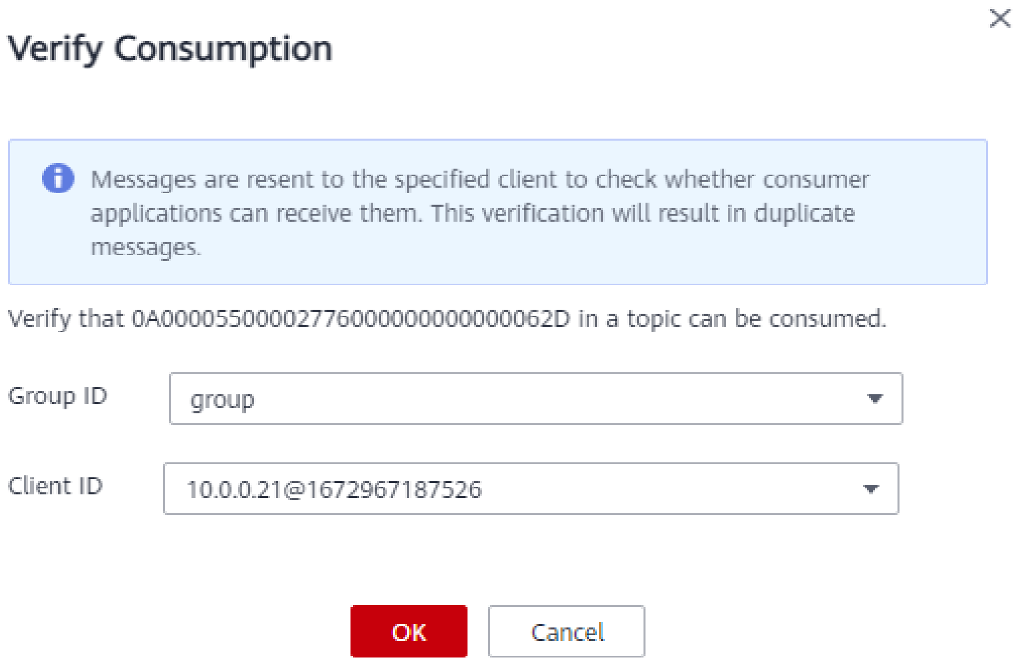
Step 7 In the row that contains the message to be resent, click **Verify Consumption**. The **Verify Consumption** dialog box is displayed.

Step 8 Set parameters by referring to **Table 9-2**.

Table 9-2 Consumption verification parameters

Parameter	Description
Group ID	Name of the consumer group for which consumption is to be verified.
Client ID	ID of the client for which consumption is to be verified.

Figure 9-1 Verifying consumption



Verify Consumption

Messages are resent to the specified client to check whether consumer applications can receive them. This verification will result in duplicate messages.

Verify that 0A000055000027760000000000000062D in a topic can be consumed.

Group ID: group

Client ID: 10.0.0.21@1672967187526

Step 9 Click **OK**.

After the message is successfully resent, "Verified" is displayed on the top of the **Verify Consumption** dialog box. The client should have received the resent message.

NOTE

Consumption verification causes repeated message consumption.

----End

9.4 Changing Message Retention Period

Scenario

Message retention period is a period that messages in a topic are retained for. Consumers must retrieve messages before this period ends. Otherwise, the messages will be deleted and can no longer be retrieved.

The retention period of normal messages and dead letter messages is controlled by and changes with the message retention period.

After creating an instance, you can modify its message retention period based on service requirements. Changing the retention period does not restart the instance. The default message retention period is 48 hours.

Procedure

Step 1 Log in to the management console.

Step 2 Click 📍 in the upper left corner to select a region.

NOTE

Select the region where your RocketMQ instance is located.

Step 3 Click ☰ and choose **Application** > **Distributed Message Service for RocketMQ** to open the console of DMS for RocketMQ.

Step 4 Click a RocketMQ instance to go to the details page.

Step 5 Click ✎ next to **Message Retention**, enter a value in the range from 1 to 720, and click ✓.

The modification result is displayed in the upper right corner of the page.

----End

9.5 Producing a Message

Scenario

This section describes how to produce a specified message for a RocketMQ instance to verify service logic.

Prerequisites

Messages can be produced only when the instance is in the **Running** state.

Procedure

Step 1 Log in to the management console.

Step 2 Click 📍 in the upper left corner to select a region.

NOTE

Select the region where your RocketMQ instance is located.

Step 3 Click ☰ and choose **Application** > **Distributed Message Service for RocketMQ** to open the console of DMS for RocketMQ.

Step 4 Click a RocketMQ instance to go to the details page.

Step 5 In the navigation pane, choose **Topics**.

Step 6 Click **Produce Message** in the row that contains the desired topic. The **Produce Message** dialog box is displayed.

Step 7 Enter the message body, key, and tag (**Message Key** and **Message Tag** are optional) and click **OK**.

When "Message sent." is displayed in the upper right corner, the message is sent successfully. You can view the message on the **Message Query** page.

----End

10 Managing Consumer Groups

10.1 Creating a Consumer Group

Scenario

A consumer group is a group of consumers with the same behavior and is used to manage and maintain message retrieval.

Prerequisites

A RocketMQ instance has been purchased.

Procedure

Step 1 Log in to the management console.

Step 2 Click 📍 in the upper left corner to select a region.

> **NOTE**
>
> Select the region where your RocketMQ instance is located.

Step 3 Click ☰ and choose **Application** > **Distributed Message Service for RocketMQ** to open the console of DMS for RocketMQ.

Step 4 Click a RocketMQ instance to go to the instance details page.

Step 5 In the navigation pane, choose **Consumer Groups**.

Step 6 Click **Create Consumer Group**.

Step 7 Configure the consumer group name and other parameters by referring to **Table 10-1**.

Table 10-1 Consumer group parameters

Parameter	Description
Consumer Group Name	Name of the consumer group. Once the consumer group is created, you cannot modify its name.
Brokers	Select one or more brokers on which the consumer group is to be created. Once the consumer group is created, you cannot modify its brokers.
Maximum Retries	Maximum number of retry attempts allowed for normal messages. For ordered retrievals, this limit can be configured by using the **setMaxReconsumeTimes** method.
Broadcast	Indicates whether messages are broadcast. If this option is enabled, each message is retrieved by all consumers in the consumer group. If this option is disabled, each message is retrieved by only one consumer in the consumer group.

Step 8 Click **OK**.

----End

10.2 Querying Consumer Group Details

Scenario

After a consumer group is created, you can query its configuration and status.

Procedure

Step 1 Log in to the management console.

Step 2 Click 📍 in the upper left corner to select a region.

NOTE

Select the region where your RocketMQ instance is located.

Step 3 Click ☰ and choose **Application** > **Distributed Message Service for RocketMQ** to open the console of DMS for RocketMQ.

Step 4 Click a RocketMQ instance to go to the instance details page.

Step 5 In the navigation pane, choose **Consumer Groups**.

Step 6 Click the name of the desired consumer group, or choose **More** > **View Detail** in the row that contains the desired consumer group to go to the details page.

In the upper part of the page, you can view the consumer group name, brokers, the maximum number of retries, and whether messages are broadcast.

In the lower part of the page, you can view all topics subscribed by the consumer group, the authentication information, and the subscription relationships.

----End

10.3 Modifying Consumer Group Parameters

Scenario

After a consumer group is created, you can modify the maximum number of retries, and specify whether to enable message broadcasting.

Procedure

Step 1 Log in to the management console.

Step 2 Click 📍 in the upper left corner to select a region.

NOTE

Select the region where your RocketMQ instance is located.

Step 3 Click ☰ and choose **Application** > **Distributed Message Service for RocketMQ** to open the console of DMS for RocketMQ.

Step 4 Click a RocketMQ instance to go to the instance details page.

Step 5 In the navigation pane, choose **Consumer Groups**.

Step 6 Modify consumer group parameters using either of the following methods:

- In the row containing the consumer group whose parameters you want to modify, click **Edit**.
- Click a consumer group and then click **Edit** in the upper right corner of the consumer group details page.

Step 7 Modify consumer group parameters by referring to **Table 10-2**.

Table 10-2 Consumer group parameters

Parameter	Description
Maximum Retries	Maximum number of times that messages are resent upon retrieval failures.
Broadcast	Indicates whether messages are broadcast. If this option is enabled, each message is retrieved by all consumers in the consumer group. If this option is disabled, each message is retrieved by only one consumer in the consumer group.

Step 8 Click **OK**.

----End

10.4 Deleting a Consumer Group

Scenario

Delete consumer groups if they are no longer needed.

Procedure

Step 1 Log in to the management console.

Step 2 Click 📍 in the upper left corner to select a region.

NOTE

Select the region where your RocketMQ instance is located.

Step 3 Click ☰ and choose **Application** > **Distributed Message Service for RocketMQ** to open the console of DMS for RocketMQ.

Step 4 Click a RocketMQ instance to go to the instance details page.

Step 5 In the navigation pane, choose **Consumer Groups**.

Step 6 Delete consumer groups using any of the following methods:

- In the row containing the consumer group you want to delete, click **Delete**.
- Click a consumer group to go to the consumer group details page and then click **Delete** in the upper right corner.
- To delete multiple consumer groups at a time, select the consumer groups, and click **Delete Consumer Group** above the consumer group list.

Step 7 Click **OK**.

----End

10.5 Viewing Consumer Connection Addresses

Scenario

View consumer connection addresses on the DMS for RocketMQ console.

NOTE

A consumer's connection addresses can be viewed only when the consumer is connected to a RocketMQ instance.

Procedure

Step 1 Log in to the management console.

Step 2 Click in the upper left corner to select a region.

NOTE

Select the region where your RocketMQ instance is located.

Step 3 Click and choose **Application** > **Distributed Message Service for RocketMQ** to open the console of DMS for RocketMQ.

Step 4 Click a RocketMQ instance to go to the instance details page.

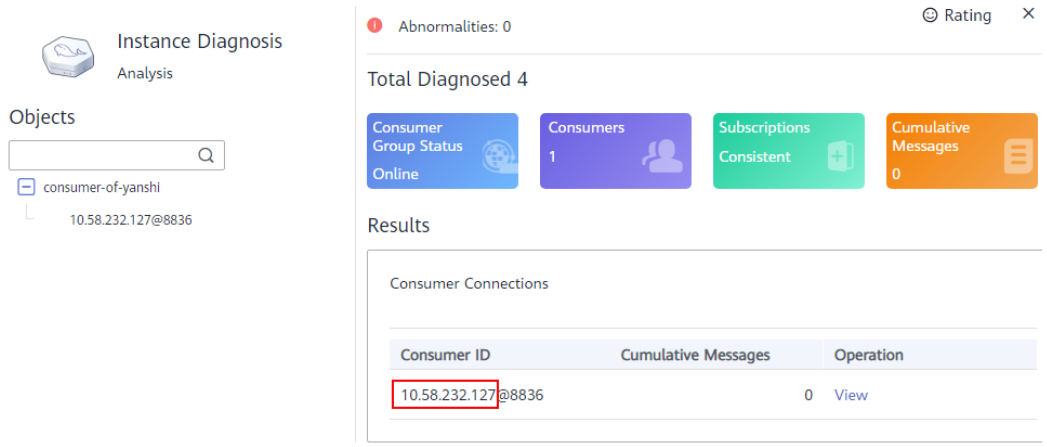
Step 5 In the navigation pane on the left, choose **Instance Diagnosis**.

Step 6 For **Consumer Group**, select the group whose consumer connection address you want to view. Then click **Start Diagnosis**.

Step 7 Locate the row that contains the target diagnosis record, and click **View**.

Step 8 In the **Results** area, view the consumer connection address.

Figure 10-1 Consumer connection addresses



Consumer Connections

Consumer ID	Cumulative Messages	Operation
10.58.232.127@8836	0	View

----End

10.6 Resetting the Consumer Offset

Scenario

Resetting the consumer offset is to change the retrieval position of a consumer.

NOTICE

Messages may be retrieved more than once after the offset is reset. Exercise caution when performing this operation.

Procedure

Step 1 Log in to the management console.

Step 2 Click 📍 in the upper left corner to select a region.

NOTE

Select the region where your RocketMQ instance is located.

Step 3 Click ☰ and choose **Application** > **Distributed Message Service for RocketMQ** to open the console of DMS for RocketMQ.

Step 4 Click a RocketMQ instance to go to the details page.

Step 5 In the navigation pane, choose **Consumer Groups**.

Step 6 In the row that contains the desired consumer group, choose **More** > **Reset Consumer Offset**.

Step 7 Set parameters by referring to **Table 10-3**.

Table 10-3 Parameters for resetting the consumer offset

Parameter	Description
Topic Name	Select the topic whose consumer offset is to be reset.
Time	Select a time point. After the reset is complete, retrieval starts from this time point. ● **Custom**: a custom time point ● **Earliest**: the earliest offset ● **Latest**: the latest offset

Step 8 Click **OK**.

Step 9 Click **Yes** in the confirmation dialog box. The consumer offset is reset.

----End

11 Managing Users

11.1 Creating a User

Scenario

RocketMQ instances support ACL-based permission control. You can create multiple users and assign different topic and consumer group permissions to them.

Prerequisites

A RocketMQ instance has been purchased.

Step 1: Enable ACL

Step 1 Log in to the management console.

Step 2 Click 📍 in the upper left corner to select a region.

NOTE

Select the region where your RocketMQ instance is located.

Step 3 Click ☰ and choose **Application** > **Distributed Message Service for RocketMQ** to open the console of DMS for RocketMQ.

Step 4 Click a RocketMQ instance to go to the instance details page.

Step 5 In the **Connection** area, click ○ next to **ACL** to enable ACL.

NOTICE

Enabling ACL will disconnect clients without authentication configuration.

----End

Step 2: Create a User

Step 1 In the navigation pane, choose **Users**.

Step 2 Click **Create User**.

Step 3 Configure the user's name and other parameters by referring to **Table 11-1**.

Table 11-1 User parameters

Parameter	Description
Name	Name of the user. The name cannot be changed after the user is created.
IP Whitelist	Users from whitelisted IP addresses have publish/subscribe permissions for all topics and consumer groups, and their secret keys will not be verified. The IP whitelist can be set to specific IP addresses or network segments. Example: **192.168.1.2,192.168.2.3** or **192.*.*.***
Administrator	A user configured as the administrator will have publish/subscribe permissions for all topics and consumer groups.
Default Topic Permissions	The user's default permissions for topics. The default permissions will be overwritten by the permissions configured for specific topics, if any. For example, if **Default Topic Permissions** is set to **Subscribe**, but a topic is configured with the **Publish/Subscribe** permissions, the topic's actual permissions will be **Publish/Subscribe**.
Default Consumer Group Permissions	The user's default permissions for consumer groups. The default permissions will be overwritten by the permissions configured for specific consumer groups, if any. For example, if a consumer group is configured with the **None** permissions, the user will not have permissions for the consumer group, even if **Default Consumer Group Permissions** is set to **Subscribe**.
Secret Key	The user's secret key.

Step 4 Click **OK**.

----End

(Optional) Step 3: Configure Permissions for a Specific Topic or Consumer Group

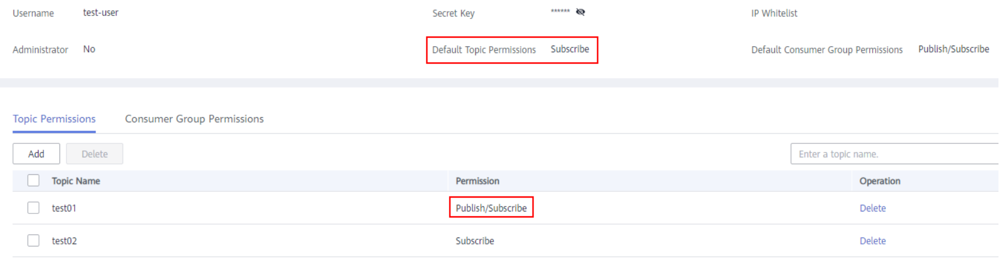
Step 1 Click a user to go to the user details page.

Step 2 On the **Topic Permissions** or **Consumer Group Permissions** tab page, click **Add**.

Step 3 Select desired topics or consumer groups, select the required permissions, and click **OK**.

These permissions overwrite the default permissions. For example, in **Figure 11-1**, users finally have publish/subscribe permissions for topic **test01**.

Figure 11-1 User details page



Username test-user

Secret Key ******

IP Whitelist

Administrator No

Default Topic Permissions Subscribe

Default Consumer Group Permissions Publish/Subscribe

	Topic Name	Permission	Operation
□	test01	Publish/Subscribe	Delete
□	test02	Subscribe	Delete

----End

Step 4: Access the Server as a User

After ACL is enabled for an instance, user authentication information must be added to both the producer and consumer configurations. For details, see the following instructions:

- Section "Java" > "Controlling Access with ACL" in *Distributed Message Service for RocketMQ Developer Guide*
- Section "Go" > "Controlling Access with ACL" in *Distributed Message Service for RocketMQ Developer Guide*
- Section "Python" > "Controlling Access with ACL" in *Distributed Message Service for RocketMQ Developer Guide*

11.2 Modifying User Information

Scenario

After a user is created, you can modify its IP whitelist, administrator option, default topic and consumer group permissions, and secret key.

Procedure

Step 1 Log in to the management console.

Step 2 Click 📍 in the upper left corner to select a region.

NOTE

Select the region where your RocketMQ instance is located.

Step 3 Click ☰ and choose **Application** > **Distributed Message Service for RocketMQ** to open the console of DMS for RocketMQ.

Step 4 Click a RocketMQ instance to go to the instance details page.

Step 5 In the navigation pane, choose **Users**.

Step 6 In the row containing the user to edit, click **Edit**.

Step 7 Modify the information shown in **Table 11-2** as required.

Table 11-2 User parameters

Parameter	Description
IP Whitelist	Users from whitelisted IP addresses have publish/subscribe permissions for all topics and consumer groups, and their secret keys will not be verified. The IP whitelist can be set to specific IP addresses or network segments. Example: **192.168.1.2,192.168.2.3** or **192.*.*.***
Administrator	A user configured as the administrator will have publish/ subscribe permissions for all topics and consumer groups.
Default Topic Permissions	The user's default permissions for topics. The default permissions will be overwritten by the permissions configured for specific topics, if any. For example, if **Default Topic Permissions** is set to **Subscribe**, but a topic is configured with the **Publish/Subscribe** permissions, the topic's actual permissions will be **Publish/Subscribe**.
Default Consumer Group Permissions	The user's default permissions for consumer groups. The default permissions will be overwritten by the permissions configured for specific consumer groups, if any. For example, if a consumer group is configured with the **None** permissions, the user will not have permissions for the consumer group, even if **Default Consumer Group Permissions** is set to **Subscribe**.
Secret Key	The user's secret key.

Step 8 Click **OK**.

----End

11.3 Deleting a User

Scenario

Delete users if they are no longer needed.

NOTICE

Deleting a user will remove its authorization relationship and disconnect it from the instance.

Procedure

Step 1 Log in to the management console.

Step 2 Click 📍 in the upper left corner to select a region.

NOTE

Select the region where your RocketMQ instance is located.

Step 3 Click ☰ and choose **Application** > **Distributed Message Service for RocketMQ** to open the console of DMS for RocketMQ.

Step 4 Click a RocketMQ instance to go to the instance details page.

Step 5 In the navigation pane, choose **Users**.

Step 6 In the row containing the user to be deleted, click **Delete**.

Step 7 Click **OK**.

----End

12 Managing Dead Letter Queues

Introduction

Dead letter queues store messages that cannot be correctly retrieved.

If a message fails to be retrieved, the system will retry sending it. After the maximum number of retries (which is 16 by default and configurable on the client) is reached and the message retrieval still fails, the message becomes a dead letter message and is delivered to the dead letter queue that corresponds to the consumer.

Dead letter messages:

- Cannot be retrieved.
- By default, dead letter messages are retained for 48 hours before being deleted automatically.

Scenario

This section describes how to query, export, and resend dead letter messages.

On the DMS for RocketMQ console, you can query dead letter messages by group, message ID, and message key.

- By group: All dead letter messages of a specified consumer group within a specified period will be queried. This is range query and may find a large number of dead letter messages.
- By message ID: The message with the specified message ID will be queried. This is exact search and can quickly find a specific dead letter message.
- By message key: The message with the specified message key will be queried. This is exact search and can quickly find a specific dead letter message.

Prerequisites

- A RocketMQ instance and consumer groups have been created.
- To query by message ID, you need the name of the consumer group to which the message belongs and the message ID.
- To query by message key, you need the name of the consumer group to which the message belongs and the message key.

Querying Dead Letter Messages

Step 1 Log in to the management console.

Step 2 Click 📍 in the upper left corner to select a region.

NOTE

Select the region where your RocketMQ instance is located.

Step 3 Click ☰ and choose **Application** > **Distributed Message Service for RocketMQ** to open the console of DMS for RocketMQ.

Step 4 Click a RocketMQ instance to go to the instance details page.

Step 5 In the navigation pane, choose **Dead Letter Queues**.

Step 6 Query dead letter messages in either of the following ways:

- By group: Select the name of the consumer group to be queried from the **Group** drop-down list. For **Stored**, select a time period.
- By message ID: Select the name of the consumer group to be queried from the **Group** drop-down list, enter the message ID of the dead letter message to be queried, and click **Search**.
- By message key: Select the name of the consumer group to be queried from the **Group** drop-down list, enter the message key of the dead letter message to be queried, and click **Search**.

----End

Exporting Dead Letter Messages

Step 1 Log in to the management console.

Step 2 Click 📍 in the upper left corner to select a region.

NOTE

Select the region where your RocketMQ instance is located.

Step 3 Click ☰ and choose **Application** > **Distributed Message Service for RocketMQ** to open the console of DMS for RocketMQ.

Step 4 Click a RocketMQ instance to go to the instance details page.

Step 5 In the navigation pane, choose **Dead Letter Queues**.

Step 6 Click **Export Message** in the row containing the desired message.

The message will be exported in JSON format.

NOTE

To export multiple dead letter messages at a time, select them and click **Export Message** above the message list.

Table 12-1 describes the fields of an exported message.

Table 12-1 Message fields

Parameter	Description
msg_id	Message ID.
instance_id	Instance ID.
topic	Topic name.
store_timestamp	Time when the message is stored.
born_timestamp	Time when the message is generated.
reconsume_times	Number of retry times.
body	Message body.
body_crc	Message body checksum.
store_size	Storage size.
property_list	Message attribute list. ● **name**: attribute name. ● **value**: attribute value.
born_host	IP address of the host that generates the message.
store_host	IP address of the host that stores the message.
queue_id	Queue ID.
queue_offset	Offset in the queue.

----End

Resending a Dead Letter Message

Dead letter messages cannot be retrieved by consumers. Locate and rectify the fault, and then resend dead letter messages on the console.

Step 1 Log in to the management console.

Step 2 Click 📍 in the upper left corner to select a region.

NOTE

Select the region where your RocketMQ instance is located.

Step 3 Click ☰ and choose **Application** > **Distributed Message Service for RocketMQ** to open the console of DMS for RocketMQ.

Step 4 Click a RocketMQ instance to go to the instance details page.

Step 5 In the navigation pane, choose **Dead Letter Queues**.

Step 6 Resend dead letter messages in either of the following ways:

- In the row containing the dead letter message to be resent, click **Resend**.
- Select multiple dead letter messages to be resent and click **Resend**.

NOTICE

After a dead letter message is successfully resent, it still exists in the dead letter queue and will not be deleted. To avoid repeated retrieval, do not resend messages repeatedly.

----End

13 Diagnosing an Instance

Scenario

You can diagnose an instance to quickly locate faults when message retrieval is slow or fails.

Prerequisites

- **A consumer group has been created**, and there are consumers in the group.
- **A topic** has been created.

Procedure

Step 1 Log in to the management console.

Step 2 Click 📍 in the upper left corner to select a region.

NOTE

Select the region where your RocketMQ instance is located.

Step 3 Click ☰ and choose **Application** > **Distributed Message Service for RocketMQ** to open the console of DMS for RocketMQ.

Step 4 Click a RocketMQ instance to go to the instance details page.

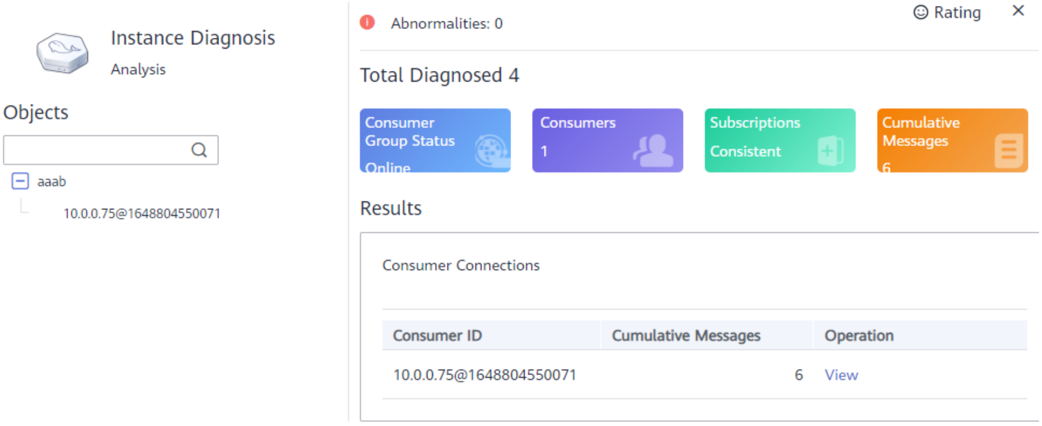
Step 5 In the navigation pane on the left, choose **Instance Diagnosis**.

Step 6 Select a consumer group and click **Start Diagnose**.

After the diagnosis is complete, a diagnosis record is displayed in the **Diagnosis Reports** area.

Step 7 Click **View** to view the diagnosis result.

Figure 13-1 Diagnosis result



Consumer ID	Cumulative Messages	Operation
10.0.0.75@1648804550071	6	View

NOTE

If the consumer group is not online, the diagnosis is successful, but one abnormality is displayed.

----End

Deleting Diagnosis Records

Step 1 Log in to the management console.

Step 2 Click 📍 in the upper left corner to select a region.

NOTE

Select the region where your RocketMQ instance is located.

Step 3 Click ☰ and choose **Application** > **Distributed Message Service for RocketMQ** to open the console of DMS for RocketMQ.

Step 4 Click a RocketMQ instance to go to the instance details page.

Step 5 In the navigation pane on the left, choose **Instance Diagnosis**.

Step 6 Use either of the following methods to delete diagnosis records:

- In the row containing the record you want to delete, click **Delete**.
- Select multiple records you want to delete and click **Delete** above the record list.

Step 7 In the confirmation dialog box, click **Yes**.

----End

14 Migrating Metadata

14.1 Migrating RocketMQ Metadata from Another Cloud or Self-hosted RocketMQ

Scenario

Migrate metadata from a third-party or self-hosted RocketMQ instance to DMS for RocketMQ.

You can use one of the following migration methods as required:

- Method 1: Run the **mqadmin** command to export the source instance metadata and then create a migration task in DMS for RocketMQ.
- Method 2: Export the source topics and consumer groups and then import them to DMS for RocketMQ using scripts. (Use this method when metadata cannot be exported using the **mqadmin** command.)

Prerequisites

- A RocketMQ instance has been purchased.
- A Linux host is available, **JDK v1.8.111 or later** has been installed on the host, and related environment variables have been configured.

Method 1: Run the mqadmin Command to Export the Source Instance Metadata and Then Create a Migration Task in DMS for RocketMQ

Obtain the RocketMQ metadata from another cloud or self-hosted RocketMQ.

Step 1 Log in to the host and download the RocketMQ software package.

```
wget https://archive.apache.org/dist/rocketmq/4.9.4/rocketmq-all-4.9.4-bin-release.zip
```

Step 2 Decompress the software package.

```
unzip rocketmq-all-4.9.4-bin-release.zip
```

Step 3 (Optional) If ACL is enabled for the RocketMQ instance, authentication is required when you run the **mqadmin** command.

Switch to the directory where the decompressed software package is stored and add the following content to the **conf/tools.yml** file:

```
accessKey:*******
secretKey:*******
```

accessKey and **secretKey** are the username and secret key set on the **Users** page of the console.

Step 4 Go to the directory where the decompressed software package is stored and run the following command to query the cluster name:

```
sh ./bin/mqadmin clusterList -n {nameserver address and port number}
```

For example, if the nameserver address and port number are **192.168.0.65:8100**, run the following command:

```
sh ./bin/mqadmin clusterList -n 192.168.0.65:8100
```

Step 5 Run the following command to export metadata:

- If SSL is disabled, run the following command:

  ```
  sh ./bin/mqadmin exportMetadata -n {nameserver address and port number} -c {RocketMQ cluster name} -f {Path for storing the exported metadata file}
  ```

  For example, if the nameserver address and port number are **192.168.0.65:8100**, the RocketMQ cluster name is **DmsCluster**, and the path for storing exported metadata files is **/tmp/rocketmq/export**, run the following command:

  ```
  sh ./bin/mqadmin exportMetadata -n 192.168.0.65:8100 -c DmsCluster -f /tmp/rocketmq/export
  ```

- If SSL is enabled, run the following command:

  ```
  JAVA_OPT=-Dtls.enable=true sh ./bin/mqadmin exportMetadata -n {nameserver address and port number} -c {RocketMQ cluster name} -f {path for storing the exported metadata file}
  ```

  For example, if the nameserver address and port number are **192.168.0.65:8100**, the RocketMQ cluster name is **DmsCluster**, and the path for storing exported metadata files is **/tmp/rocketmq/export**, run the following command:

  ```
  JAVA_OPT=-Dtls.enable=true sh ./bin/mqadmin exportMetadata -n 192.168.0.65:8100 -c DmsCluster -f /tmp/rocketmq/export
  ```

----End

Migrate metadata on the console.

Step 1 Log in to the management console.

Step 2 Click 📍 in the upper left corner to select a region.

NOTE

Select the region where your RocketMQ instance is located.

Step 3 Click ☰ and choose **Application** > **Distributed Message Service for RocketMQ** to open the console of DMS for RocketMQ.

Step 4 Click a RocketMQ instance to go to the instance details page.

Step 5 In the navigation pane, choose **Metadata Migration**.

Step 6 Click **Create Migration Task**.

Step 7 Configure the migration task by referring to **Table 14-1**.

Table 14-1 Migration task parameters

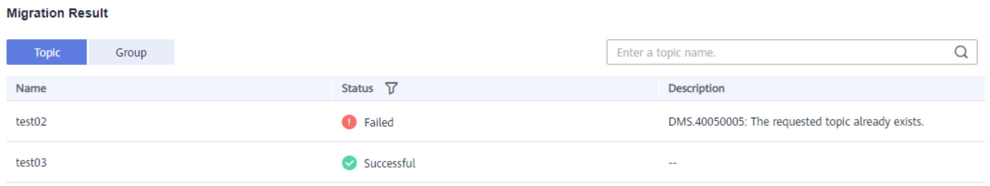
Parameter	Description
Task Type	Select **From self-built RocketMQ to cloud.**
Task Name	Unique name of the migration task.
Overwrite	● If this option is enabled, configurations in the metadata file with the same name as the uploaded file will be modified. Assume that Topic01 on the source instance has three read queues, and Topic01 on the DMS instance has two read queues. If **Overwrite** is enabled, Topic01 on the DMS instance will have three read queues after migration. ● If this option is disabled, migration of the metadata file with the same name as the uploaded file will fail. Assume that the source instance has Topic01 and Topic02, and the DMS instance has Topic01 and Topic03. If **Overwrite** is disabled, migration of the source Topic01 will fail.
Metadata	Upload the **RocketMQ metadata obtained from another cloud or self-hosted RocketMQ**.

Step 8 Click **OK**.

After the migration is complete, view **Task Status** in the migration task list.

- If **Task Status** is **Complete**, all metadata has been successfully migrated.
- If **Task Status** is **Failed**, some or all metadata fails to be migrated. Click the migration task name to go to the migration task details page. In the **Migration Result** area, view the name of the topic or consumer group that fails to be migrated and the failure cause.

Figure 14-1 Migration result



Migration Result

Name	Status	Description
test02	Failed	DMS.40050005: The requested topic already exists.
test03	Successful	--

----End

Method 2: Export the Source Topics and Consumer Groups and Import Them to DMS for RocketMQ Using Scripts

Step 1 Log in to the console of another vendor and export the lists of source topics and consumer groups.

Step 2 Create the **topics.txt** and **groups.txt** files and add the source topic list and consumer group list to the files respectively. Each line contains a topic or consumer group name. For example:

```
topic-01
topic-02
...
topic-n
```

Note: The **groups.txt** file cannot contain blank lines (for example, a newline character at the end of a consumer group name). Otherwise, consumer groups with empty names will be created when the lists are imported to the RocketMQ instance.

Step 3 Log in to the host and download the RocketMQ software package.

```
wget https://archive.apache.org/dist/rocketmq/4.9.4/rocketmq-all-4.9.4-bin-release.zip
```

Step 4 Decompress the software package.

```
unzip rocketmq-all-4.9.4-bin-release.zip
```

Step 5 (Optional) If ACL is enabled for the RocketMQ instance, authentication is required when you run the **mqadmin** command.

Switch to the directory where the decompressed software package is stored and add the following content to the **conf/tools.yml** file:

```
accessKey:*******
secretKey:*******
```

accessKey and **secretKey** are the username and secret key set on the **Users** page of the console.

Step 6 Go to the **bin** directory of the decompressed software package and upload **topics.txt** and **groups.txt** to this directory.

Step 7 Run the following script to import the source topics and consumer groups to DMS for RocketMQ:

```
#!/bin/bash

# Read groups from groups.txt file
groups=()
while read -r group; do
  groups+=("$group")
done < "groups.txt"

# Read topics from topic.txt file
topics=()
while read -r topic; do
  topics+=("$topic")
done < "topics.txt"

# Add topics
for topic in "${topics[@]}"; do
  echo "Adding topic: $topic"
  sh mqadmin updateTopic -n <namesrvIp:8100> -c DmsCluster -t "$topic"
done

# Add consumer groups
for group in "${groups[@]}"; do
  echo "Adding consumer group: $group"
  sh mqadmin updateSubGroup -n <namesrvIp:8100> -c DmsCluster -g "$group"
done
```

namesrvIp:8100 indicates the address of the RocketMQ instance.

Step 8 Log in to DMS for RocketMQ console. Go to the **Topics** and **Consumer Groups** pages and check whether the topics and consumer groups are successfully imported.

----End

14.2 Migrating RabbitMQ Metadata

Scenario

On the console, you can migrate RabbitMQ metadata to your DMS RocketMQ instance.

Prerequisites

- A RocketMQ instance has been purchased.
- **RabbitMQ plug-ins** have been enabled.

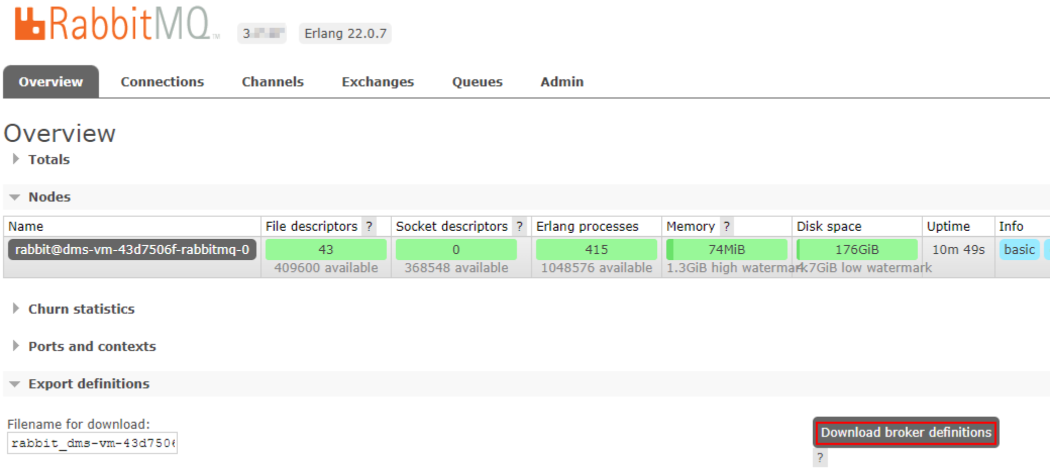
Step 1: Obtain RabbitMQ Metadata

Step 1 Enter **http://***IP address of your RabbitMQ***:15672/** in the address box of the browser to access the open-source RabbitMQ console.

Step 2 On the login page, enter the username and password, and click **Login**.

Step 3 On the **Overview** tab page, click **Download broker definitions** to export the metadata.

Figure 14-2 Exporting metadata



Name	File descriptors ?	Socket descriptors ?	Erlang processes	Memory ?	Disk space	Uptime	Info
rabbit@dms-vm-43d7506f-rabbitmq-0	43 409600 available	0 368548 available	415 1048576 available	74MiB 1.3GiB high watermark	176GiB 4.7GiB low watermark	10m 49s	basic

Filename for download:
rabbit_dms-vm-43d750(

----End

Step 2: Migrate Metadata on the Console

Step 1 Log in to the management console.

Step 2 Click in the upper left corner to select a region.

NOTE

Select the region where your RocketMQ instance is located.

Step 3 Click ☰ and choose **Application** > **Distributed Message Service for RocketMQ** to open the console of DMS for RocketMQ.

Step 4 Click a RocketMQ instance to go to the details page.

Step 5 In the navigation pane, choose **Metadata Migration**.

Step 6 Click **Create Migration Task**.

Step 7 Configure the migration task by referring to **Table 14-2**.

Table 14-2 Migration task parameters

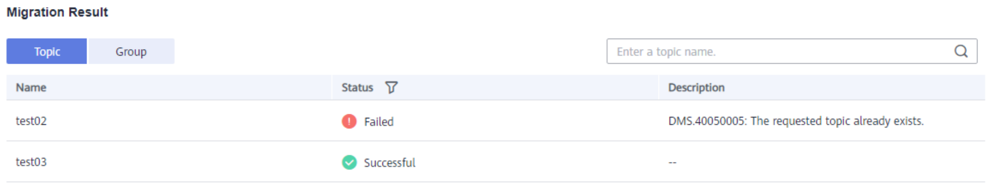
Parameter	Description
Task Type	Select **From RabbitMQ to RocketMQ**.
Task Name	Unique name of the migration task.
Overwrite	● If this option is enabled, configurations in the metadata file with the same name as the uploaded file will be modified. Assume that Topic01 on the source instance has three read queues, and Topic01 on the DMS instance has two read queues. If **Overwrite** is enabled, Topic01 on the DMS instance will have three read queues after migration. ● If this option is disabled, migration of the metadata file with the same name as the uploaded file will fail. Assume that the source instance has Topic01 and Topic02, and the DMS instance has Topic01 and Topic03. If **Overwrite** is disabled, migration of the source Topic01 will fail.
Metadata	Upload the **RabbitMQ metadata**.

Step 8 Click **OK**.

After the migration is complete, view **Task Status** in the migration task list.

- If **Task Status** is **Complete**, all metadata has been successfully migrated.
- If **Task Status** is **Failed**, some or all metadata fails to be migrated. Click the migration task name to go to the migration task details page. In the **Migration Result** area, view the name of the topic or consumer group that fails to be migrated and the failure cause.

Figure 14-3 Migration result (migrating RabbitMQ metadata)



Migration Result

Name	Status	Description
test02	● Failed	DMS.40050005: The requested topic already exists.
test03	✓ Successful	--

----End

15 Monitoring

15.1 DMS for RocketMQ Metrics

Introduction

This section describes the metrics that DMS for RocketMQ reports to Cloud Eye. You can view metrics and alarms by using the DMS for RocketMQ console.

Namespace

SYS.DMS

Instance Metrics

Table 15-1 Instance metrics

Metric ID	Metric Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
instance_produce_msg	Created Messages	Number of messages received by the instance per minute Unit: count	> 0	RocketMQ instance	1 minute
instance_consume_msg	Retrieved Messages	Number of messages retrieved from the instance per minute Unit: count	> 0	RocketMQ instance	1 minute

Broker Metrics

Table 15-2 Broker metrics

Metric ID	Metric Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
broker_produce_msg	Created Messages	Number of messages received by the broker per minute Unit: count	> 0	RocketMQ instance broker	1 minute
broker_consume_msg	Retrieved Messages	Number of messages retrieved from the broker per minute Unit: count	> 0	RocketMQ instance broker	1 minute
broker_produce_rate	Message Creation Rate	Number of messages received by the broker per second Unit: count/s	> 0	RocketMQ instance broker	1 minute
broker_consume_rate	Message Retrieval Rate	Number of messages retrieved from the broker per second Unit: count/s	> 0	RocketMQ instance broker	1 minute
broker_total_bytes_in_rate	Inbound Traffic	Inbound traffic to the broker per second Unit: byte/s	> 0	RocketMQ instance broker	1 minute
broker_total_bytes_out_rate	Outbound Traffic	Outbound traffic from the broker per second Unit: byte/s	> 0	RocketMQ instance broker	1 minute
broker_cpu_core_load	Average Load per CPU Core	Average load of each CPU core in the broker VM	> 0	RocketMQ instance broker	1 minute

Metric ID	Metric Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
broker_disk_usage	Disk Capacity Usage	Disk usage of the broker VM Unit: %	0–100	RocketMQ instance broker	1 minute
broker_memory_usage	Memory Usage	Memory usage of the broker VM Unit: %	0–100	RocketMQ instance broker	1 minute

Topic Metrics

Table 15-3 Topic metrics

Metric ID	Metric Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
topic_produce_msg	Created Messages	Number of messages received by the topic per minute Unit: count	> 0	RocketMQ instance queue	1 minute
topic_consume_msg	Retrieved Messages	Number of messages retrieved from the topic per minute Unit: count	> 0	RocketMQ instance queue	1 minute
topic_produce_rate	Message Creation Rate	Number of messages received by the topic per second Unit: count/s	> 0	RocketMQ instance queue	1 minute
topic_consume_rate	Message Retrieval Rate	Number of messages retrieved from the topic per second Unit: count/s	> 0	RocketMQ instance queue	1 minute

Consumer Group Metrics

Table 15-4 Consumer group metrics

Metric ID	Metric Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
group_consume_msg	Retrieved Messages	Number of messages retrieved by the consumer group per minute Unit: count **NOTE** This metric is available only when **Topic** is set to **All topics** on the **By Consumer Group** tab page.	> 0	Consumer group of a RocketMQ instance	1 minute
group_accumulation	Available Messages	Number of messages accumulated in the consumer group Unit: count **NOTE** This metric is available only when **Topic** is set to **All topics** on the **By Consumer Group** tab page.	> 0	Consumer group of a RocketMQ instance	1 minute

Metric ID	Metric Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
group_topic_consume_msg	Retrieved Messages	Number of messages in the specified topic that are retrieved by the consumer group per minute Unit: count **NOTE** This metric is available only when **Topic** is set to a specified topic on the **By Consumer Group** tab page.	> 0	Consumer group of a RocketMQ instance	1 minute
group_topic_consume_rate	Message Retrieval Rate	Number of messages in the specified topic that are retrieved by the consumer group per second Unit: count/s **NOTE** This metric is available only when **Topic** is set to a specified topic on the **By Consumer Group** tab page.	> 0	Consumer group of a RocketMQ instance	1 minute

Metric ID	Metric Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
group_topic_accumulation	Available Messages	Number of messages in a specific topic Unit: count **NOTE** This metric is available only when **Topic** is set to a specified topic on the **By Consumer Group** tab page.	> 0	Consumer group of a RocketMQ instance	1 minute

Dead Letter Queue Metrics

Table 15-5 Dead letter queue metrics

Metric ID	Metric Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
dlq_accumulation	Messages in the Dead Letter Queue	Number of messages in the dead letter queue Unit: count	> 0	Dead letter queue of a RocketMQ instance	1 minute

Dimensions

Key	Value
reliablemq_instance_id	RocketMQ instance
reliablemq_broker	RocketMQ instance broker
reliablemq_topics	RocketMQ instance queue
reliablemq_groups	Consumer group of a RocketMQ instance
reliablemq_groups_topics	Consumer group of a RocketMQ instance queue
reliablemq_dlq_topics	Dead letter queue of a RocketMQ instance

15.2 Configuring Alarm Rules for Critical Metrics

This section describes the alarm policies of some metrics and how to configure them. In actual services, you are advised to configure alarm rules for metrics based on the following alarm policies.

NOTE

Approach Upper Limit in the following table indicates whether the threshold is close to the upper limit of the performance supported by current resources. If the threshold is close to the upper limit and usage continues to rise, services may be abnormal.

Table 15-6 RocketMQ instance metrics to configure alarm rules for

Metric Name	Normal Range	Alarm Policy	Approach Upper Limit	Metric Description and Alarm Handling Suggestions
Disk Capacity Usage	0–100	Alarm threshold: Raw data > 85 Number of consecutive periods: 3 Alarm severity: Critical	Yes	**Metric description:** disk usage of the RocketMQ VM. Unit: % **Handling suggestion**: If an alarm is generated for this metric, the current instance specifications are insufficient to carry services. The **storage space** needs to be expanded by referring to **Modifying Specifications**.
Average Load per CPU Core	0–2	Alarm threshold: Raw data > 1.5 Number of consecutive periods: 3 Alarm severity: Major	Yes	**Metric description:** average load of each CPU core of the RocketMQ VM. **Handling suggestion**: If an alarm is generated for this metric, perform the following operations: 1. Add brokers by referring to **Modifying Specifications**. 2. Redeploy existing topics to the new brokers by referring to **Modifying Topic Parameters**.

Metric Name	Normal Range	Alarm Policy	Approach Upper Limit	Metric Description and Alarm Handling Suggestions
Memory Usage	0–100	Alarm threshold: Raw data > 85 Number of consecutive periods: 3 Alarm severity: Critical	Yes	**Metric description:** memory usage of the RocketMQ VM. Unit: % **Handling suggestion**: If an alarm is generated for this metric, perform the following operations: 1. Add brokers by referring to **Modifying Specifications**. 2. Redeploy existing topics to the new brokers by referring to **Modifying Topic Parameters**.

Procedure

Step 1 Log in to the management console.

Step 2 Click 📍 in the upper left corner to select a region.

NOTE

Select the region where your RocketMQ instance is located.

Step 3 Click ☰ and choose **Application** > **Distributed Message Service for RocketMQ** to open the console of DMS for RocketMQ.

Step 4 In the row containing the desired instance, click 🖥.

You are redirected to the metric monitoring page on the Cloud Eye console.

Step 5 Hover the mouse pointer over a metric and click + to create an alarm rule for the metric.

The **Create Alarm Rule** page is displayed.

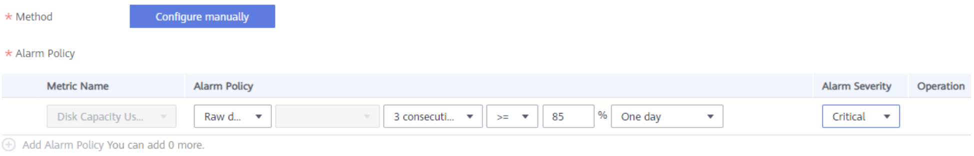
Step 6 Specify the alarm rule details.

For details about how to create alarm rules, see **Creating an Alarm Rule**.

1. Enter the alarm name and description.
2. Specify the alarm policy and alarm severity.

   As shown in the following figure, if the original disk capacity usage exceeds 85% for three consecutive periods, an alarm is generated. If the alarm is not handled on time, an alarm notification is sent.

Figure 15-1 Setting the alarm policy and alarm severity



* Method: Configure manually

* Alarm Policy

Metric Name	Alarm Policy	Alarm Severity	Operation
Disk Capacity Us...	Raw d... 3 consecuti... >= 85 % One day	Critical	

3. Set the alarm notification configurations. If you enable **Alarm Notification**, set the validity period, notification object, and trigger condition.
4. Click **Create**.

----End

15.3 Viewing Metrics

Scenario

Cloud Eye monitors RocketMQ instance metrics in real time. You can view these metrics on the console.

Prerequisites

A RocketMQ instance has been purchased.

Procedure

Step 1 Log in to the management console.

Step 2 Click in the upper left corner to select a region.

NOTE

Select the region where your RocketMQ instance is located.

Step 3 Click ☰ and choose **Application** > **Distributed Message Service for RocketMQ** to open the console of DMS for RocketMQ.

Step 4 View the instance metrics using either of the following methods:

- Locate the row that contains the instance to be checked, click **View Metrics**.
- Click a RocketMQ instance to go to the instance details page. In the navigation pane on the left, choose **Monitoring**.

Step 5 View the monitoring data of the instance, brokers, topics, consumer groups, and dead letter queues.

Rollup is the process in which Cloud Eye calculates the maximum, minimum, average, sum, or variance value of raw data sampled in different periods and persists the results. A calculation period is called a rollup period.

Step 6 (Optional) You can select the metrics to display on the page and rearrange the order of selected metrics.

For example, if you only need to view the **Created Messages** and **Retrieved Messages** metrics on the **Brokers** tab page, do as follows:

1. On the **Brokers** tab page, click **Select Metric**.
2. Select **Created Messages** and **Retrieved Messages**, and click **OK**.
3. Drag and drop the selected metrics to desired locations to rearrange the order.

----End

16 Auditing

16.1 Operations Logged by CTS

With Cloud Trace Service (CTS), you can record DMS for RocketMQ operations for later query, audit, and backtrack operations.

Table 16-1 DMS for RocketMQ operations that can be recorded by CTS

Operation	Resource Type	Trace Name
Successfully creating a topic	reliability	RocketMQ_Topic_CreationSuccess
Failing to create a topic	reliability	RocketMQ_Topic_CreationFailure
Successfully modifying a topic	reliability	RocketMQ_Topic_ModifySuccess
Failing to modify a topic	reliability	RocketMQ_Topic_ModifyFailure
Successfully deleting a topic	reliability	RocketMQ_Topic_DeletionSuccess
Failing to delete a topic	reliability	RocketMQ_Topic_DeletionFailure
Successfully deleting topics in batches	reliability	RocketMQ_Batch_Topic_DeletionSuccess
Failing to delete topics in batches	reliability	RocketMQ_Batch_Topic_DeletionFailure
Successfully modifying topics in batches	reliability	RocketMQ_Batch_Topic_ModifySuccess
Failing to modify topics in batches	reliability	RocketMQ_Batch_Topic_ModifyFailure

Operation	Resource Type	Trace Name
Successfully creating a consumer group	reliability	RocketMQ_Create_GroupSuccess
Failing to create a consumer group	reliability	RocketMQ_Create_GroupFailure
Successfully editing a consumer group	reliability	RocketMQ_Group_ModifySuccess
Failing to edit a consumer group	reliability	RocketMQ_Group_ModifyFailure
Successfully deleting a consumer group	reliability	RocketMQ_Group_DeletionSuccess
Failing to delete a consumer group	reliability	RocketMQ_Group_DeletionFailure
Successfully deleting consumer groups in batches	reliability	RocketMQ_Batch_Group_DeletionSuccess
Failing to delete consumer groups in batches	reliability	RocketMQ_Batch_Group_DeletionFailure
Successfully editing consumer groups in batches	reliability	RocketMQ_Batch_Group_Modify Success
Failing to edit consumer groups in batches	reliability	RocketMQ_Batch_Group_Modify Failure
Successfully restarting an instance node	reliability	restartInstanceBrokerSuccess
Failing to restart an instance node	reliability	restartInstanceBrokerFailure
Successfully deleting a background task	reliability	deleteDMSBackendJobSuccess
Failing to delete a background task	reliability	deleteDMSBackendJobFailure
Successfully creating an order for creating an instance	reliability	createDMSInstanceOrderSuccess
Failing to create an order for creating an instance	reliability	createDMSInstanceOrderFailure

Operation	Resource Type	Trace Name
Successfully submitting a request to modify an instance order	reliability	modifyDMSInstanceOrderSuccess
Failing to submit a request to modify an instance order	reliability	modifyDMSInstanceOrderFailure
Successfully submitting a request to scale up an instance	reliability	extendDMSInstanceSuccess
Failing to submit a request to scale up an instance	reliability	extendDMSInstanceFailure
Successfully deleting an instance that failed to be created	reliability	deleteDMSCreateFailureInstancesSuccess
Failing to delete an instance that failed to be created	reliability	deleteDMSCreateFailureInstancesFailure
Successfully submitting a request to delete multiple instances at a time	reliability	batchDeleteDMSInstanceSuccess
Failing to submit a request to delete multiple instances at a time	reliability	batchDeleteDMSInstanceFailure
Successfully submitting a request to modify instance information	reliability	modifyDMSInstanceInfoSuccess
Failing to submit a request to modify instance information	reliability	modifyDMSInstanceInfoFailure
Deleting multiple instance tasks at a time	reliability	batchDeleteDMSInstanceTask
Successfully deleting an instance	reliability	deleteDMSInstanceTaskSuccess
Failing to delete an instance	reliability	deleteDMSInstanceTaskFailure

Operation	Resource Type	Trace Name
Successfully creating an instance	reliability	createDMSInstanceTaskSuccess
Failing to create an instance	reliability	createDMSInstanceTaskFailure
Successfully scaling up an instance	reliability	extendDMSInstanceTaskSuccess
Failing to scale up an instance	reliability	extendDMSInstanceTaskFailure
Successfully modifying instance information	reliability	modifyDMSInstanceInfoTaskSuccess
Failing to modify instance information	reliability	modifyDMSInstanceInfoTaskFailure

16.2 Querying Real-Time Traces

Scenarios

After you enable CTS and the management tracker is created, CTS starts recording operations on cloud resources. After a data tracker is created, the system starts recording operations on data in OBS buckets. CTS stores operation records generated in the last seven days.

This section describes how to query and export operation records of the last seven days on the CTS console.

- **Viewing Real-Time Traces in the Trace List of the New Edition**
- **Viewing Real-Time Traces in the Trace List of the Old Edition**

Viewing Real-Time Traces in the Trace List of the New Edition

1. Log in to the management console.
2. Click ☰ in the upper left corner and choose **Management & Deployment** > **Cloud Trace Service**. The CTS console is displayed.
3. Choose **Trace List** in the navigation pane on the left.
4. On the **Trace List** page, use advanced search to query traces. You can combine one or more filters.
   - **Trace Name**: Enter a trace name.
   - **Trace ID**: Enter a trace ID.
   - **Resource Name**: Enter a resource name. If the cloud resource involved in the trace does not have a resource name or the corresponding API operation does not involve the resource name parameter, leave this field empty.

- **Resource ID**: Enter a resource ID. Leave this field empty if the resource has no resource ID or if resource creation failed.
- **Trace Source**: Select a cloud service name from the drop-down list.
- **Resource Type**: Select a resource type from the drop-down list.
- **Operator**: Select one or more operators from the drop-down list.
- **Trace Status**: Select **normal**, **warning**, or **incident**.
  - **normal**: The operation succeeded.
  - **warning**: The operation failed.
  - **incident**: The operation caused a fault that is more serious than the operation failure, for example, causing other faults.
- Time range: Select **Last 1 hour**, **Last 1 day**, or **Last 1 week**, or specify a custom time range.

5. On the **Trace List** page, you can also export and refresh the trace list, and customize the list display settings.
   - Enter any keyword in the search box and click 🔍 to filter desired traces.
   - Click **Export** to export all traces in the query result as an .xlsx file. The file can contain up to 5000 records.
   - Click ↻ to view the latest information about traces.
   - Click ⚙ to customize the information to be displayed in the trace list. If **Auto wrapping** is enabled (●), excess text will move down to the next line; otherwise, the text will be truncated. By default, this function is disabled.
6. For details about key fields in the trace structure, see section "Trace References" > "Trace Structure" and section "Trace References" > "Example Traces".
7. (Optional) On the **Trace List** page of the new edition, click **Go to Old Edition** in the upper right corner to switch to the **Trace List** page of the old edition.

Viewing Real-Time Traces in the Trace List of the Old Edition

1. Log in to the management console.
2. Click ☰ in the upper left corner and choose **Management & Deployment** > **Cloud Trace Service**. The CTS console is displayed.
3. Choose **Trace List** in the navigation pane on the left.
4. Each time you log in to the CTS console, the new edition is displayed by default. Click **Go to Old Edition** in the upper right corner to switch to the trace list of the old edition.
5. Set filters to search for your desired traces. The following filters are available:
   - **Trace Type**, **Trace Source**, **Resource Type**, and **Search By**: Select a filter from the drop-down list.
     - If you select **Resource ID** for **Search By**, specify a resource ID.

- If you select **Trace name** for **Search By**, specify a trace name.
- If you select **Resource name** for **Search By**, specify a resource name.

– **Operator**: Select a user.

– **Trace Status**: Select **All trace statuses**, **Normal**, **Warning**, or **Incident**.

– Time range: You can query traces generated during any time range in the last seven days.

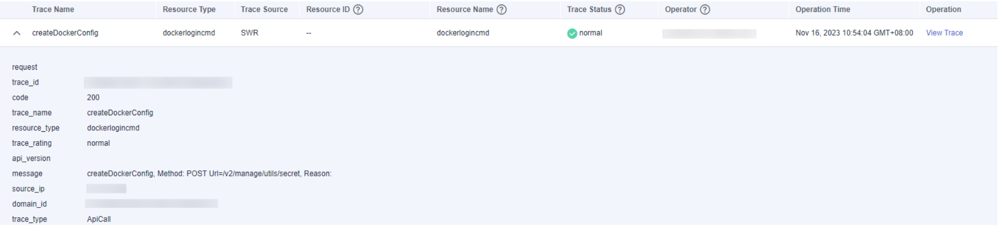
– Click **Export** to export all traces in the query result as a CSV file. The file can contain up to 5000 records.

6. Click **Query**.
7. On the **Trace List** page, you can also export and refresh the trace list.

   – Click **Export** to export all traces in the query result as a CSV file. The file can contain up to 5000 records.

   – Click ↻ to view the latest information about traces.

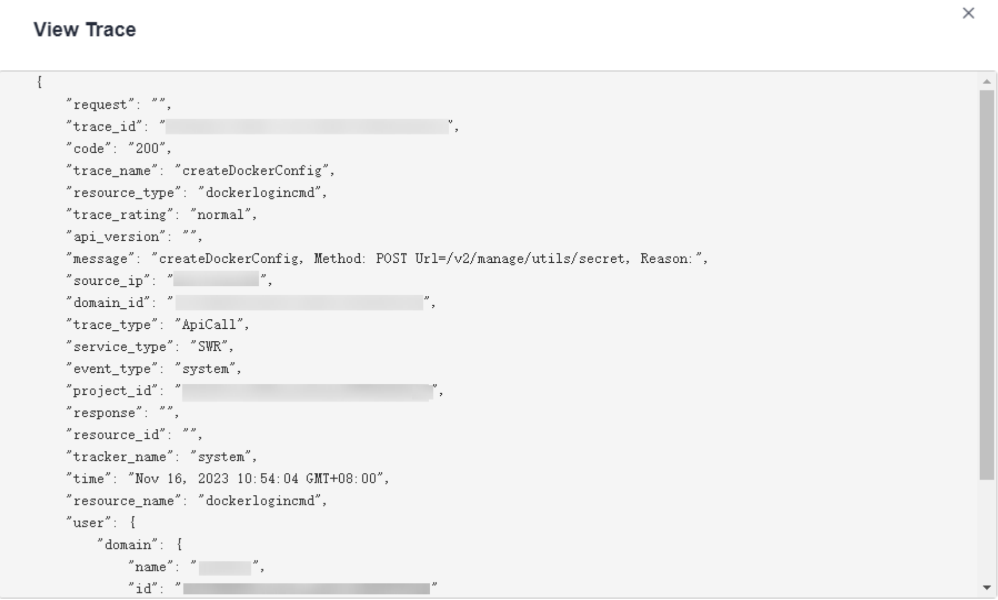
8. Click ⌄ on the left of a trace to expand its details.



	Trace Name	Resource Type	Trace Source	Resource ID ⓘ	Resource Name ⓘ	Trace Status ⓘ	Operator ⓘ	Operation Time	Operation
^	createDockerConfig	dockerlogincmd	SWR	--	dockerlogincmd	✓ normal	[redacted]	Nov 16, 2023 10:54:04 GMT+08:00	View Trace

request	
trace_id	[redacted]
code	200
trace_name	createDockerConfig
resource_type	dockerlogincmd
trace_rating	normal
api_version	
message	createDockerConfig, Method: POST Url=/v2/manage/utils/secret, Reason:
source_ip	[redacted]
domain_id	[redacted]
trace_type	ApiCall

9. Click **View Trace** in the **Operation** column. The trace details are displayed.



View Trace

```
{
    "request": "",
    "trace_id": "[redacted]",
    "code": "200",
    "trace_name": "createDockerConfig",
    "resource_type": "dockerlogincmd",
    "trace_rating": "normal",
    "api_version": "",
    "message": "createDockerConfig, Method: POST Url=/v2/manage/utils/secret, Reason:",
    "source_ip": "[redacted]",
    "domain_id": "[redacted]",
    "trace_type": "ApiCall",
    "service_type": "SWR",
    "event_type": "system",
    "project_id": "[redacted]",
    "response": "",
    "resource_id": "",
    "tracker_name": "system",
    "time": "Nov 16, 2023 10:54:04 GMT+08:00",
    "resource_name": "dockerlogincmd",
    "user": {
        "domain": {
            "name": "[redacted]",
            "id": "[redacted]"
```

10. For details about key fields in the trace structure, see section "Trace References" > "Trace Structure" and section "Trace References" > "Example Traces".
11. (Optional) On the **Trace List** page of the old edition, click **New Edition** in the upper right corner to switch to the **Trace List** page of the new edition.

17 FAQs

17.1 Instance FAQs

17.1.1 Which RocketMQ Versions Are RocketMQ Instances Compatible with?

RocketMQ instances are compatible with open-source RocketMQ 4.8.0. 4.8.0 and later clients are recommended.

17.1.2 How Is Instance Reliability Ensured?

RocketMQ instance reliability is ensured through:

- Data storage in one master and two slaves
- Cross-AZ deployment

17.1.3 Why Can't I Select Two AZs?

To improve the reliability of a RocketMQ instance, you are advised to select three AZs or more when creating the instance. An instance deployed across two AZs may become unavailable when one of the AZs is faulty.

17.1.4 Can I Change the SSL Setting of a RocketMQ Instance?

No. Once the instance has been created, you cannot enable or disable SSL.

17.2 Connection FAQs

17.2.1 Does RocketMQ Allow Self-built Brokers to Be Connected to NameServers on the Cloud?

No. If a self-built broker is connected to the NameServer on the cloud, management plane functions are affected. As a result, topics cannot be queried.

17.2.2 How Do I Handle Instance Connection Failure "RemotingTooMuchRequestException: sendDefaultImpl call timeout"

Symptom

Connecting to a RocketMQ instance failed. The error message is "RemotingTooMuchRequestException: sendDefaultImpl call timeout".

Possible Cause

The security group does not allow the service port.

Solution

1. On the instance details page of the RocketMQ console, view the port next to **Allow Access To** in the **Network** area.
2. Click the security group name to view details.
3. Add an inbound rule to allow the port obtained in **1**.

17.3 Topic FAQs

17.3.1 How Many Topics Are Supported?

As shown in **Table 17-1**, the number of topics supported varies depending on the instance flavor.

Table 17-1 Instance flavors (DMS for RocketMQ 4.8.0)

Flavor	Brokers	Storage Space (GB per Broker)	TPS per Broker	Max. Topics per Broker	Max. Consumer Groups per Broker
rocketmq.4u8g.cluster.small	1–2	300–60,000	15,000	2000	2000
rocketmq.4u8g.cluster	1–10	300–600,000	20,000	4000	4000
rocketmq.8u16g.cluster	1–10	300–900,000	25,000	8000	8000
rocketmq.12u24g.cluster	1–10	300–900,000	28,000	12,000	12,000
rocketmq.16u32g.cluster	1–10	300–900,000	30,000	16,000	16,000

17.3.2 Is Automatic Topic Creation Supported?

No. RocketMQ does not support automatic topic creation.

17.4 Consumer Group FAQs

17.4.1 How Many Consumer Groups Are Supported?

As shown in **Table 17-2**, the number of consumer groups supported varies depending on the instance flavor.

Table 17-2 Instance flavors (DMS for RocketMQ 4.8.0)

Flavor	Brokers	Storage Space (GB per Broker)	TPS per Broker	Max. Topics per Broker	Max. Consumer Groups per Broker
rocketmq.4u8g.cluster.small	1–2	300–60,000	15,000	2000	2000
rocketmq.4u8g.cluster	1–10	300–600,000	20,000	4000	4000
rocketmq.8u16g.cluster	1–10	300–900,000	25,000	8000	8000
rocketmq.12u24g.cluster	1–10	300–900,000	28,000	12,000	12,000
rocketmq.16u32g.cluster	1–10	300–900,000	30,000	16,000	16,000

17.5 Message FAQs

17.5.1 What Is the Message Retention Period of RocketMQ? Can I Modify It?

The message retention period for RocketMQ instances is 48 hours by default and the upper limit is 720 hours. For details about how to change message retention period, see **Changing Message Retention Period**.

17.5.2 What Is the Maximum Message Size Supported RocketMQ? Can I Modify It?

The maximum message size supported by RocketMQ is 4 MB. This limit cannot be modified.

17.5.3 What Is the Replication Mode of RocketMQ? Can I Modify It?

A RocketMQ message is stored on three replicas, one master and two slaves. This mode cannot be modified.

17.5.4 Where Is Message Creation Time Set?

The message creation time is set by the producer during message production.

17.6 Monitoring & Alarms

17.6.1 Why Can't I View Topic Monitoring Data?

A percent (%) or vertical bar (|) contained in a topic name will be converted to an underscore (_) by Cloud Eye. For example, if a topic name is **test%01**, it will be displayed as **test_01** on Cloud Eye.

A Change History

Released On	Description
2023-03-23	This issue is the first official release.