Distributed Message Service for RocketMQ

Getting Started

Issue   01
Date   2023-10-08

Contents

1 Introduction

This document provides instructions for getting started with Distributed Message Service (DMS) for RocketMQ, including creating a RocketMQ instance on the console and connecting to an instance through an Elastic Cloud Server (ECS).

Procedure

Figure 1-1 Procedure for using DMS for RocketMQ
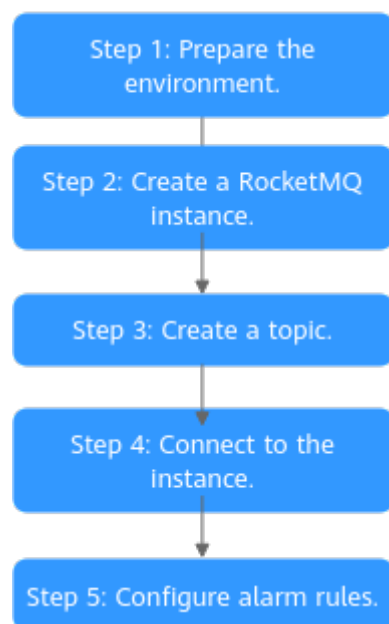
1. **Prepare the environment**.

   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**.

   When creating an instance, you can choose whether to enable SSL. If SSL is enabled, data is encrypted for transmission, improving data security. The SSL setting can be configured only when you create an instance. After an instance is created, the SSL setting cannot be changed.

3. **Create a topic**.

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

4. Connect to the instance.

   A client can connect to an instance **without SSL** or **with SSL**.

5. **Configure alarm rules**.

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

   📖 **NOTE**

   Learn more about the **basic concepts of RocketMQ**.

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 **Creating a VPC**. 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 **Creating a Security Group**. If you already have an available security group, you do not need to create a new one.

Note the following when creating a security group:

- Set **Template** to **Custom**.
- 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	8200	0.0.0.0/0	The port is used for public access to metadata nodes.

Directi on	Protocol	Port	Source	Description
Inboun d	TCP	10100-10 199	0.0.0.0/0	The port is used for accessing service nodes.

📖 **NOTE**

> After a security group is created, retain the default inbound rule that allows communication among ECSs within the security group and the 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

(Optional) EIP

To access a RocketMQ instance over a public network, prepare an elastic IP address (EIP) in advance.

For details, see **Assigning an EIP**.

The EIP must be created in the region the RocketMQ instance is in.

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 **Purchasing 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

3 Step 2: Create a RocketMQ Instance

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

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   Go to the **Buy Instance page**.

Step 2   Select a billing mode.

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   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.   **Specifications**: Select **Default** or **Custom**.

     If you select Default and the version is 4.8.0, specify the flavor, number of brokers, and broker storage space to be supported by the RocketMQ instance based on the site requirements.

     –   **Instance Type**: Currently, only cluster instances are supported.

     –   **CPU Architecture**: Currently, only x86 architecture is supported.

     –   **Broker Flavor**: Select the required flavor.

     –   **Brokers**: Select the required number of brokers.

     –   **Storage Space per Broker**: Specify the disk type and 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 Disk Performance**.

Figure 3-1 Default specifications

Specifications	Default	Custom		
Version	4.8.0			
Instance Type	Cluster			
CPU Architecture	x86			
Broker Flavor	Flavor Name		Maximum Topics per Broker	Maximum Consumer Groups per Broker
	● rocketmq.4u8g.cluster.small		2,000	2,000
	○ rocketmq.4u8g.cluster		4,000	4,000
	○ rocketmq.8u16g.cluster		8,000	8,000
	○ rocketmq.12u24g.cluster		12,000	12,000
	○ rocketmq.16u32g.cluster		16,000	16,000
	Currently Selected rocketmq.4u8g.cluster.small	Maximum Topics per Broker 2,000	Maximum Consumer Groups per Broker 2,000	
	HTTP access unavailable for this flavor.			
Brokers	− 1 +			

If you select Custom, the system calculates the number of brokers and broker storage space for different flavors based on your specified number of topics, peak creation TPS, peak retrieval TPS, average message size, and number of messages created every day. You can select one of the recommended flavors as required.

Figure 3-2 Specification calculation

Specifications	Default	Custom		
Parameters	Topics	20		
	Peak Creation TPS	300		
	Peak Retrieval TPS	500		
	Less			
	Average Message Size (KB)	100		
	Messages Created per Day	2000		
	Calculate			
Recommended Specifications	Flavor Name c6.4u8g.cluster	Flavor Name c6.8u16g.cluster	Flavor Name c6.12u24g.cluster	Flavor Name c6.16u32g.cluster
	Brokers 2	Brokers 2	Brokers 2	Brokers 2
	Storage Space per Broker 600 GB	Storage Space per Broker 600 GB	Storage Space per Broker 600 GB	Storage Space per Broker 600 GB
	Total storage space 1,200 GB	Total storage space 1,200 GB	Total storage space 1,200 GB	Total storage space 1,200 GB

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 Select the required duration.

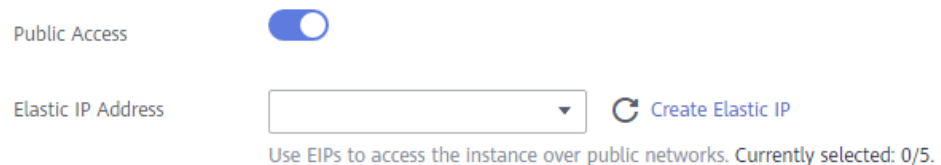This parameter is displayed only if the billing mode is yearly/monthly.

Step 9   Click **More Settings** to configure more parameters.

    1.   Configure public access.

    Public access is disabled by default. You can enable it or keep it disabled as required.

    After public access is enabled, configure an IPv4 EIP for each broker.

Figure 3-3 Configuring public access

Public Access	⬤ (enabled)
Elastic IP Address	[▼] ⟳ Create Elastic IP
	Use EIPs to access the instance over public networks. **Currently selected: 0/5**.

    2.   Configure **SSL**.

    This parameter indicates whether to enable SSL authentication when a client connects to the RocketMQ instance. After SSL is enabled, data will be encrypted before transmission for higher security, but performance will be degraded.

    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.

    3.   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, by usage, owner, or environment).

    –   If you have created predefined tags, select a predefined pair of tag key and value. You can click **View predefined tags** to go to the Tag Management Service (TMS) console and view or create tags.

    –   You can also create new tags by entering **Tag key** and **Tag value**.

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

    4.   Enter a description of the instance.

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 the instance fails to be created, view **Instance Creation Failures**. Delete the instance and create another one. If the instance creation fails again, contact customer service.

    ----End

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 console.

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 4-1**.

Table 4-1 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**.

Parameter	Description
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 4-1 Creating a topic

Step 6 Click **OK**.

----End

5 Step 4: Connect to a RocketMQ Instance

5.1 Without SSL

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 (see **Step 2: Create a RocketMQ Instance**) and you have obtained the metadata connection addresses for intra-VPC access or public network access.
- **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.

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

Parameter description:

- **Connection addresses**: the **metadata connection addresses** for intra-VPC access or public network access to the RocketMQ instance
- **Topic name**: name of the topic created for the RocketMQ instance

In the following example, **100.85.125.89:8200;100.85.119.144:8200** are the metadata connection addresses for public network access to the RocketMQ instance, and **topic-test** is the topic name.

```
sh mqadmin sendMessage -n "100.85.125.89:8200;100.85.119.144:8200" -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 **metadata connection addresses** for intra-VPC access or public network access to the RocketMQ instance
- **Topic name**: name of the topic created for the RocketMQ instance

In the following example, **100.85.125.89:8200;100.85.119.144:8200** are the metadata connection addresses for public network access to the RocketMQ instance, and **topic-test** is the topic name.

```
sh mqadmin consumeMessage -n "100.85.125.89:8200;100.85.119.144:8200" -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 **metadata connection addresses** for intra-VPC access or public network access to the RocketMQ instance
- **Topic name**: name of the topic created for the RocketMQ instance

In the following example, **100.85.125.89:8200;100.85.119.144:8200** are the metadata connection addresses for public network access to the RocketMQ instance, and **topic-test** is the topic name.

```
sh mqadmin sendMessage -n "100.85.125.89:8200;100.85.119.144:8200" -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 **metadata connection addresses** for intra-VPC access or public network access to the RocketMQ instance
- **Topic name**: name of the topic created for the RocketMQ instance

In the following example, **100.85.125.89:8200;100.85.119.144:8200** are the metadata connection addresses for public network access to the RocketMQ instance, and **topic-test** is the topic name.

```
sh mqadmin consumeMessage -n "100.85.125.89:8200;100.85.119.144:8200" -t topic-test -m true
```

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

----End

5.2 With SSL

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 (see **Step 2: Create a RocketMQ Instance**) and you have obtained the metadata connection addresses for intra-VPC access or public network access.

- **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 **metadata connection addresses** for intra-VPC access or public network access to the RocketMQ instance

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

In the following example, **100.85.125.89:8200;100.85.119.144:8200** 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 "100.85.125.89:8200;100.85.119.144:8200" -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 **metadata connection addresses** for intra-VPC access or public network access to the RocketMQ instance
- **Topic name**: name of the topic created for the RocketMQ instance

In the following example, **100.85.125.89:8200;100.85.119.144:8200** 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 "100.85.125.89:8200;100.85.119.144:8200" -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 **metadata connection addresses** for intra-VPC access or public network access to the RocketMQ instance
- **Topic name**: name of the topic created for the RocketMQ instance

In the following example, **100.85.125.89:8200;100.85.119.144:8200** 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 "100.85.125.89:8200;100.85.119.144:8200" -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 **metadata connection addresses** for intra-VPC access or public network access to the RocketMQ instance
- **Topic name**: name of the topic created for the RocketMQ instance

In the following example, **100.85.125.89:8200;100.85.119.144:8200** 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 "100.85.125.89:8200;100.85.119.144:8200" -t
topic-test -m true
```

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

----End

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 6-1 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. For details, see **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. For details, see **Modifying Specifications**. 2. Redeploy existing topics to the new brokers. For details, see **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. For details, see **Modifying Specifications**. 2. Redeploy existing topics to the new brokers. For details, see **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 📈 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 more information about creating 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 6-1 Setting the alarm policy and alarm severity
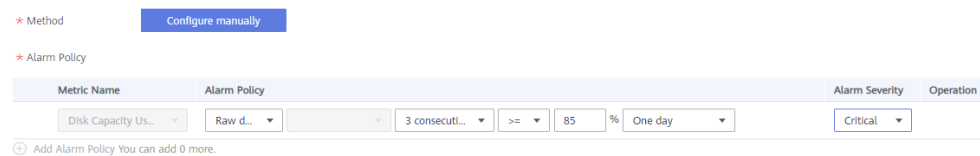
3. Set the alarm notification configurations. If you enable **Alarm Notification**, set the validity period, notification object, and trigger condition.

4. Click **Create**.

----End