Distributed Message Service for RocketMQ

Service Overview

Issue 01

Date 2024-12-25

HUAWEI

HUAWEI CLOUD COMPUTING TECHNOLOGIES CO., LTD.

Trademarks and Permissions


and other Huawei trademarks are the property of Huawei Technologies Co., Ltd.
All other trademarks and trade names mentioned in this document are the property of their respective holders.

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 What Is DMS for RocketMQ?.....1
2 Product Advantages.....4
3 Application Scenarios.....5
4 Specifications.....8
5 Comparing RocketMQ, Kafka, and RabbitMQ.....11
6 Comparing DMS for RocketMQ and Open-Source RocketMQ.....14
7 Restrictions.....16
8 Related Services.....19
9 RocketMQ Concepts.....20
10 Permissions Management.....21
11 Billing.....25

1 What Is DMS for RocketMQ?

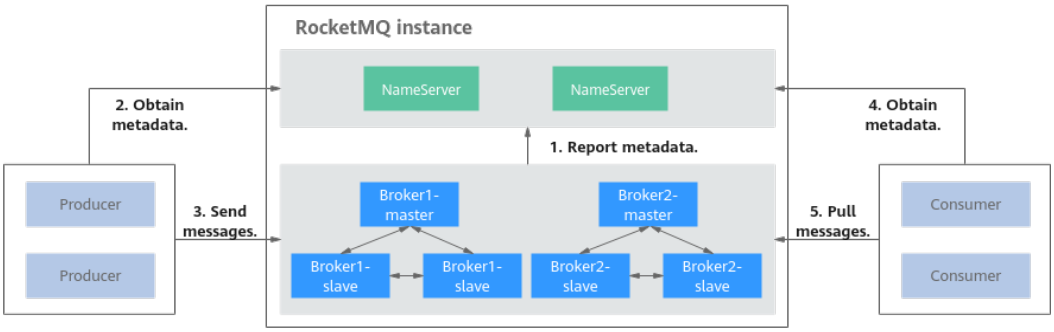
Distributed Message Service (DMS) for RocketMQ is message-oriented middleware that delivers low latency, high flexibility, high throughput, dynamic expansion, easy management, and abundant messaging functions.

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 consumed. If the message consumption 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 **setMaxReconsume-Times** 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.

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 Huawei Cloud, 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.

3 Application Scenarios

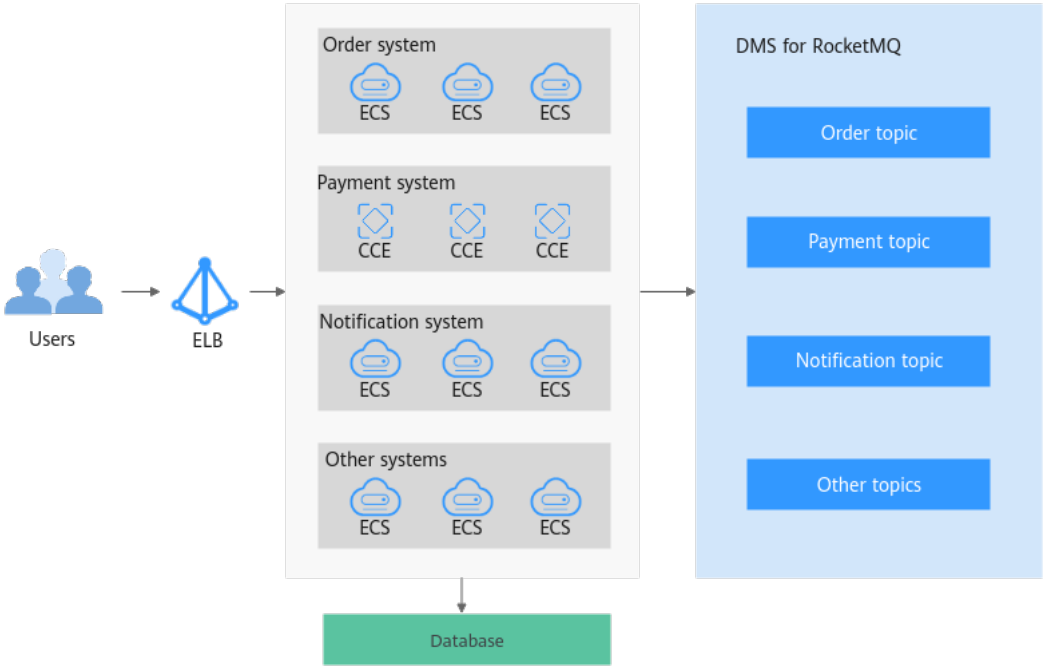
E-Commerce

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.

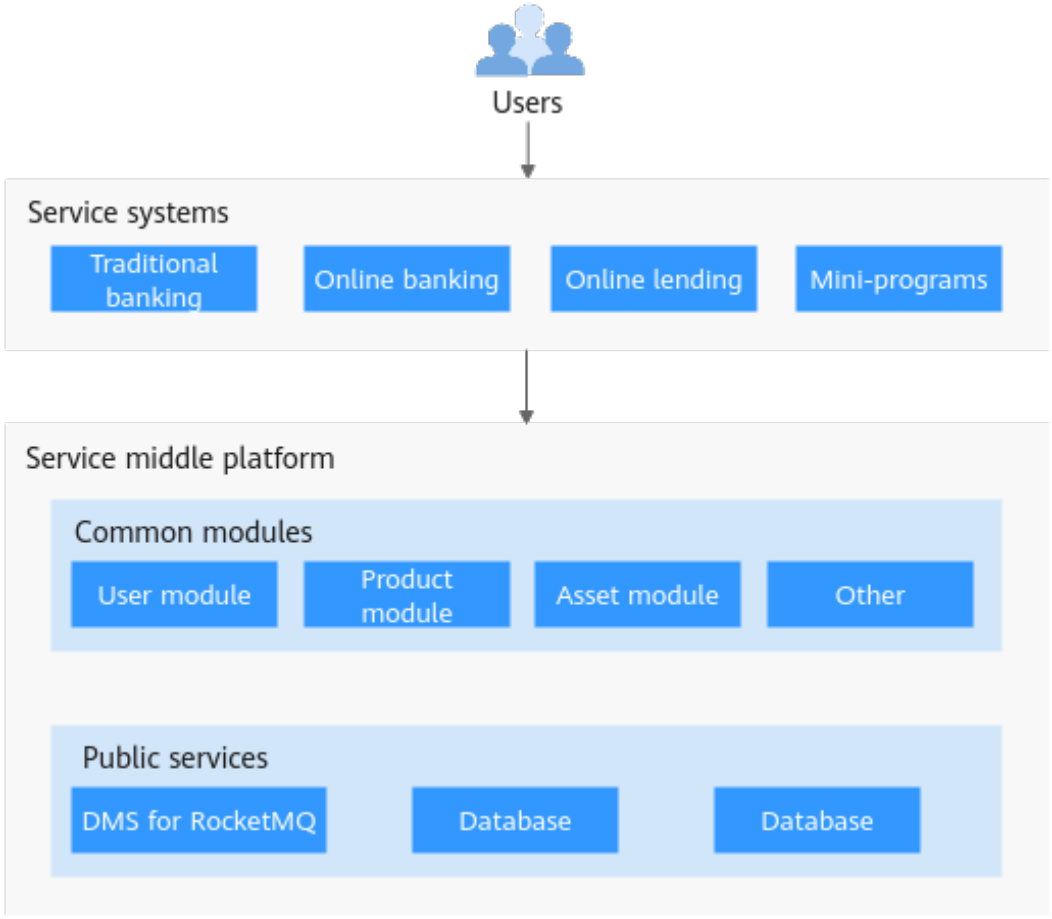
DMS for RocketMQ can help you cope with e-commerce challenges.

- 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.



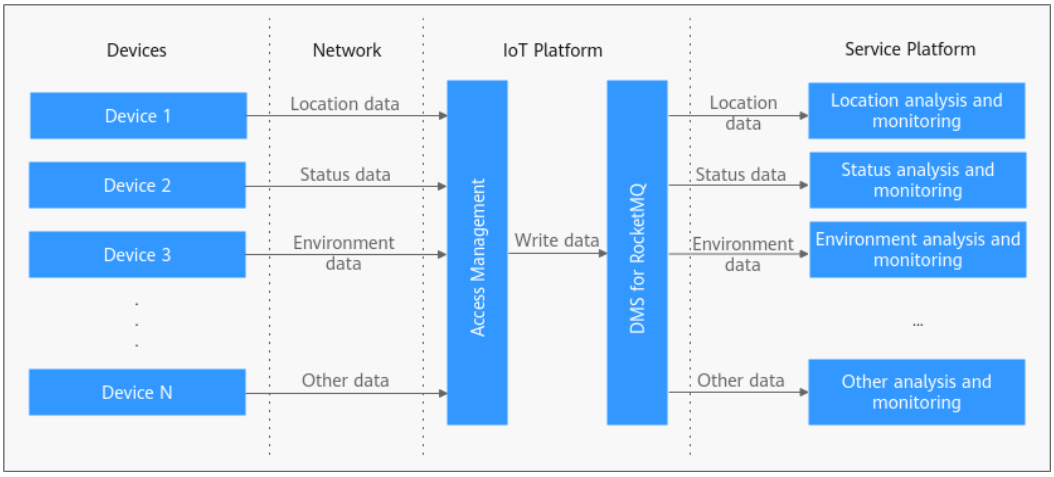
Finance

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.



4 Specifications

DMS for RocketMQ is compatible with open-source RocketMQ 4.8.0 and 5.x. Their specifications are described as follows.

NOTE

RocketMQ process messages, relying on the disk throughput. Ultra-high I/O and higher disks satisfy the performance baseline but high I/O ones may not.

DMS for RocketMQ 4.8.0

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

- ECS flavor: flavor of the ECS used by the instance
- Number of brokers: scale of the instance
- Disk type: defines the storage space. High I/O and ultra-high I/O types are available.
- 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 4-1 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 4-1 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	1–10	300–30,000	15,000	2,000	2,000
rocketmq.4u8g.cluster	1–10	300–60,000	20,000	4,000	4,000
rocketmq.8u16g.cluster	1–10	300–90,000	25,000	8,000	8,000
rocketmq.12u24g.cluster	1–10	300–90,000	28,000	12,000	12,000
rocketmq.16u32g.cluster	1–10	300–90,000	30,000	16,000	16,000

DMS for RocketMQ 5.x Basic Edition

The specifications of a RocketMQ 5.x basic edition instance are defined on six dimensions:

- ECS flavor: flavor of the ECS used by the instance
- Disk type: defines the storage space. High I/O and ultra-high I/O types are available.
- Storage space: storage capacity the instance can use
- TPS: TPS performance of the instance
- Max. topics: number of topics allowed for the instance
- Max. consumer groups: number of consumer groups allowed for the instance

Table 4-2 and Table 4-3 list the flavors of DMS for RocketMQ 5.x basic edition. In the following tables, TPS is calculated using 4 KB normal messages. The TPS of sending scheduled, delayed, or transactional messages is that of sending normal messages multiplied by 5. For receiving these messages, do not do the multiplication. For example, the TPS of sending a normal message per second is 1, and TPS of sending a transactional message per second is 5 (1 x 5).

Table 4-2 Instance flavors (DMS for RocketMQ 5.x basic edition single-node)

Flavor	Storage Space (GB)	TPS	Max. Topics	Max. Consumer Groups
rocketmq.b1.large.1	100–30,000	500	50	500

Table 4-3 Instance flavors (DMS for RocketMQ 5.x basic edition cluster)

Flavor	Storage Space (GB)	TPS	Max. Topics	Max. Consumer Groups
rocketmq.b2.large.4	200–60,000	2,000	100	1000
rocketmq.b2.large.8	200–60,000	4,000	100	1000
rocketmq.b2.large.12	200–60,000	6,000	100	1000

5 Comparing RocketMQ, Kafka, and RabbitMQ

Table 5-1 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.

Feature	RocketMQ	Kafka	RabbitMQ
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.
Message filtering	Supported	Supported	Not supported, but can be encapsulated.
Multi-tenancy	Supported	Supported	Supported
Multi-protocol	Compatible with RocketMQ.	Only supports Apache Kafka.	RabbitMQ is based on AMQP.
Multi-language	Supports clients in multiple programming languages.	Kafka is written in Scala and Java and supports clients in multiple programming languages.	Supports clients in multiple programming languages.
Throttling	RocketMQ 5.x supports traffic control based on instance specifications.	Supports throttling on producer or consumer clients, users, and topics.	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.	Supports FIFO only for single-threaded message queuing without advanced features such as delayed queues or priority queues.

Feature	RocketMQ	Kafka	RabbitMQ
Security	Supports SSL authentication.	Supports SSL and SASL authentication and read/write permissions control.	Supports SSL authentication.
Transactional messages	Supported	Supported	Supported

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 6-1 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

Feature	DMS for RocketMQ	Open-Source RocketMQ
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.
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

7 Restrictions

This section describes the restrictions on using DMS for RocketMQ.

NOTICE

Any instability caused by ignorance of the notes and constraints is not covered by the SLA.

Instance

Table 7-1 Restrictions

Item	Restrictions
Version	Fixed once the instance is created. Use the same version as your client.
Storage space	• The disk type cannot be changed once the instance is created. • Can only be increased. • Can be increased for up to 20 times.
Broker quantity	Can only be increased.
Broker flavor	Can be increased. • If the broker quantity is 1, a master/standby switchover will be performed during an increase, interrupting services for within 30s. • If the broker quantity is greater than 1 and topics are across brokers, intermittent disconnections in seconds will occur during an increase. Ordered messages may not work properly. You are advised to perform an increase during off-peak hours.
VPC/subnet/AZ	Fixed once the instance is created.

Item	Restrictions
Request-Reply	Not supported.
Configuration parameters	Cannot be modified using open-source APIs.

Topic

Table 7-2 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.
Automatic topic creation	Not supported.
Broker faults	When some brokers of an instance are faulty, topics cannot be created, modified, or deleted, but can be queried.

Consumer Group

Table 7-3 Restrictions

Item	Restrictions
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.
Creating consumer groups, consumers, and producers	Consumer groups, consumers, and producers are generated automatically when you use the instance.
Consumer offset reset	You can reset the retrieval start position to any time within two days.
Resetting the consumer offset	Messages may be consumed more than once after the offset is reset.
Broker faults	When some brokers of an instance are faulty, consumer groups cannot be created, modified, or deleted, but can be queried.

Message

Table 7-4 Restrictions

Item	Restrictions
Message retention duration	The default retention period is 48 hours, which can be modified to up to 720 hours. Messages will be automatically deleted after the retention period expires.
Message size	The maximum size of a message is 4 MB. The maximum size of any message attribute (topic name, message type, message ID, or message creation time) is 16 KB. The message size cannot be changed. When this limit is reached, messages will fail to be sent.
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.

User

Table 7-5 Restrictions

Item	Restrictions
Number of users	A maximum of 1000 users can be created for a RocketMQ instance.
Broker faults	When some brokers of an instance are faulty, users cannot be created, modified, or deleted, but can be queried.

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 Huawei Cloudmanagement 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.

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.

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 Huawei 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 HUAWEI ID 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 10-1 lists all the system-defined roles and policies supported by DMS for RocketMQ.

Table 10-1 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
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.

Role/Policy Name	Description	Type	Dependencies
DMSAgencyCheckAccessPolicy	IAM operation permissions to assign to DMS agencies.	System-defined policy	None

Table 10-2 lists the common operations supported by each DMS for RocketMQ system policy or role. Select the policies or roles as required.

Table 10-2 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	√	√	√	×	×

Fine-grained Authorization

To use a custom fine-grained policy, log in to the IAM console as an administrator and select the desired fine-grained permissions for DMS. Table 10-3 describes fine-grained permission dependencies of DMS for RocketMQ.

Table 10-3 Fine-grained permission dependencies of DMS for RocketMQ

Permission	Description	Dependency
dms:instance:list	Viewing the instance list	None
dms:instance:get	Viewing instance details	None

Permission	Description	Dependency
dms:instance:create	Creating an instance	• vpc:vpcs:get • vpc:ports:create • vpc:securityGroups:get • vpc:ports:get • vpc:subnets:get • vpc:vpcs:list • vpc:publicIps:get • vpc:publicIps:list • vpc:ports:update • vpc:publicIps:update • vpc:ports:delete
dms:instance:getBackgroundTask	Viewing background task details	None
dms:instance:deleteBackgroundTask	Deleting a background task	None
dms:instance:modify	Modifying an instance	• vpc:vpcs:get • vpc:ports:create • vpc:securityGroups:get • vpc:ports:get • vpc:subnets:get • vpc:vpcs:list • vpc:publicIps:get • vpc:publicIps:list • vpc:ports:update • vpc:publicIps:update • vpc:ports:delete
dms:instance:delete	Deleting an instance	None

Helpful Links

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

11 Billing

DMS for RocketMQ supports pay-per-use and yearly/monthly billing. For details, see **Pricing Details**.

Billing Items

DMS for RocketMQ is billed based on the instance flavor and storage space.

Table 11-1 Billing items of DMS for RocketMQ

Item	Billing
Instance	• Instances are billed based on the flavors described in **Table 11-2**, **Table 11-3**, and **Table 11-4**. • RocketMQ instances can be billed on a yearly/monthly or pay-per-use (hourly) basis.
Storage space	• Instances are billed based on the storage space. For each type of instance specification, you can choose the high I/O or ultra-high I/O disk type to meet your service requirements. • Storage space can be specified with increments of 100 GB. For details about the storage space range, see **Table 11-2**, **Table 11-3**, and **Table 11-4**. • Storage space can be billed on a yearly/monthly or pay-per-use (hourly) basis.

Table 11-2 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	1–10	300–30,000	15,000	2,000	2,000

Flavor	Brokers	Storage Space (GB per Broker)	TPS per Broker	Max. Topics per Broker	Max. Consumer Groups per Broker
rocketmq.4u8g.cluster	1–10	300–60,000	20,000	4,000	4,000
rocketmq.8u16g.cluster	1–10	300–90,000	25,000	8,000	8,000
rocketmq.12u24g.cluster	1–10	300–90,000	28,000	12,000	12,000
rocketmq.16u32g.cluster	1–10	300–90,000	30,000	16,000	16,000

Table 11-3 Instance flavors (DMS for RocketMQ 5.x basic edition single-node)

Flavor	Storage Space (GB)	TPS	Max. Topics	Max. Consumer Groups
rocketmq.b1.large.1	100–30,000	500	50	500

Table 11-4 Instance flavors (DMS for RocketMQ 5.x basic edition cluster)

Flavor	Storage Space (GB)	TPS	Max. Topics	Max. Consumer Groups
rocketmq.b2.large.4	200–60,000	2,000	100	1000
rocketmq.b2.large.8	200–60,000	4,000	100	1000
rocketmq.b2.large.12	200–60,000	6,000	100	1000

Billing Modes

Two billing modes are available, allowing you to pay less by using more.

- Yearly/Monthly: Provides a larger discount than pay-per-use billing and is recommended for long-term use of resources.
- Pay-per-use (hourly): a flexible billing mode, which allows you to pay for only the resources you actually use. The minimum time unit is one hour. Less than an hour is recorded as an hour.

Renewal

You can renew an instance before it expires, or you can set auto-renewal rules for an instance.