TDMQ for RocketMQ

FAQs

Product Documentation

Copyright Notice

Trademark Notice

Tencent Cloud

Service Statement

This document is intended to provide users with general information about Tencent Cloud's products and services only and does not form part of Tencent Cloud's terms and conditions. Tencent Cloud's products or services are subject to change. Specific products and services and the standards applicable to them are exclusively provided for in Tencent Cloud's applicable terms and conditions.

Contents

FAQs

Instance on v4.x

General

Last updated：2023-04-12 11:33:33

No messages can be queried in the TDMQ for RocketMQ console

If all messages cannot be queried， submit a ticket for assistance.

If some messages cannot be queried, there will be a delay of several seconds for message query, and you can only query messages within three days. For other cases, submit a ticket for assistance.

Client Integration

Last updated：2023-04-12 11:34:42

The error message “ `No route info of this topic` ” was prompted

Error details:

`[org.apache.rocketmq.client.exception.MQClientException: No route info of this topic: generation%rocketmq-gab9bb2877z8|generation%video_task`

General possible causes:

Incorrect topic_name. Check whether the namespace is correct. The topic name shoud be formatted as `rocketmq-xxx|[ns]%[topicName]` .

namespace refers to the name of a namespace, which can be copied under the **Namespace** tab on the cluster details page in the console.

topic_name refers to the name of a topic, which can be copied under the `Topic` tab on the cluster details page in the console.

Other possible causes:

Broker prohibits automatic creation of topics, and users do not manually create topics in the console.

Broker is not connected to Name Server correctly. Submit a ticket for assistance.

Producer is not connected to Name Server correctly. Check whether the access point is entered correctly.

The error message “ `Send [3] times, still failed, cost [334]ms` ” was prompted

Error details:

`com.freesky.inspiration.domain.mq.MQProducerManager:sendMessage``[org.apache.rocketmq.client.exception.MQClientException: Send [3] times, still failed, cost [334]ms, Topic: video_task, BrokersSent: [broker-2, broker-0, broker-1]`

General possible causes:

No permissions. Check whether the namespace has been authorized.

Message Sending

Last updated：2023-04-12 11:35:56

The error message “ `ACL exception` ” was prompted when client producer failed to send a message

```
Caused by: org.apache.rocketmq.client.exception.MQBrokerException: CODE: 1  DESC: o
For more information, please visit the url, http://rocketmq.apache.org/docs/faq/
        at org.apache.rocketmq.client.impl.MQClientAPIImpl.processSendResponse(MQCl
        at org.apache.rocketmq.client.impl.MQClientAPIImpl.sendMessageSync(MQClient
        at org.apache.rocketmq.client.impl.MQClientAPIImpl.sendMessage(MQClientAPII
        at org.apache.rocketmq.client.impl.MQClientAPIImpl.sendMessage(MQClientAPII
        at org.apache.rocketmq.client.impl.producer.DefaultMQProducerImpl.sendKerne
        at org.apache.rocketmq.client.impl.producer.DefaultMQProducerImpl.sendDefau
        ... 3 more
```

General possible causes:

The `AclException` is reported if ccess-key/secret-key is not correctly configured.

access-key: Role token in namespace configuration permissions.

secret-key: Role name in namespace configuration permissions.

The error message “ `RemotingConnectException` ” was prompted during client integration

```
Caused by: org.apache.rocketmq.remoting.exception.RemotingConnectException: connect
    at org.apache.rocketmq.remoting.netty.NettyRemotingClient.getAndCreateNameserve
    at org.apache.rocketmq.remoting.netty.NettyRemotingClient.getAndCreateChannel(N
    at org.apache.rocketmq.remoting.netty.NettyRemotingClient.invokeSync(NettyRemot
    at org.apache.rocketmq.client.impl.MQClientAPIImpl.getTopicRouteInfoFromNameSer
    at org.apache.rocketmq.client.impl.MQClientAPIImpl.getTopicRouteInfoFromNameSer
    at org.apache.rocketmq.client.impl.factory.MQClientInstance.updateTopicRouteInf
    ... 6 more
```

The access point is most likely incorrectly entered. The access point can be obtained from the **Network** module on the cluster’s basic information page.

The cluster access address supports VPC by default.

The public network access is disabled by default. If you are using a virtual cluster, you need to submit a ticket for application to enable it. If you are using an exclusive cluster, you can adjust public network bandwidth to enable or disable it. We recommend that you use the public network access only for testing.

Note:

When accessing from VPC, you need to use a CVM machine in the same region or another environment that can successfully access the VPC. You can use ping or telnet to test whether the network is successfully connected to. If the network connection fails, the error message `RemotingConnectException` will be prompted.

Message Consumption

Last updated：2023-10-19 11:07:35

Messages cannot be consumed

Possible causes:

The producer starts before the consumer starts, but `consumeFromWhere` is incorrectly configured on the consumer, causing the consumer to start consumption from the latest offset.

When multiple producers use the same consumer group but subscribe to different tags, message push errors also occur, resulting in messages not being consumed.

Message Pulling Frequency

Last updated：2023-10-19 11:08:40

TDMQ for RocketMQ provides two consumption modes: Push and Pull. The Push mode is implemented based on the Pull mode, wrapping a layer on the Pull mode. It is a variant of the Pull mode rather than a true Push mode.
Therefore, it can be considered that the TDMQ for RocketMQ SDK continuously performs message pulling tasks.
Below are the possible results of each pull:
After the SDK successfully pulls messages, if some messages are filtered out (such as by tag), it will put the message pulling request to the blocking queue to pull messages in a loop; otherwise, the messages will be put to the consumer for consumption while the message pulling request will be put to the blocking queue.
If no new messages are available in the broker or no messages are matched, `pullRequest` will be put to the blocking queue to continue to pull messages in a loop.
That is to say, as long as the consumer is online, it will initiate requests to the server periodically even if no messages are produced. This can ensure the timeliness of message consumption. If you want to reduce the number of pulls, refer to the following section.

How to Control Pulling Frequency

In the Pull mode, you can actively control the message pulling frequency of consumers by using throttling components or waiting for a desired period of time before the next pull.
In the Push mode, you can use `PullInterval` to set the message pulling interval for consumers.

```
// Set the interval to 60s
pushConsumer.setPullInterval(60 * 1000);
```

Note

Modifying the message pulling interval may cause a higher consumption delay. Therefore, proceed with caution.

If a small number of messages are produced, the long polling time of fixed 15s will be readily reached; that is, a broker will initiate at least four pulls for a queue per minute. If there are 10 topics with 3 queues each, and there are 2 brokers on the server, the number of pulls will be 240 (10*3*2*4). In this case, four pulls are initiated per second.

In actual scenarios, retry topics may be generated during the production and consumption processes. As the retry queue also needs to be pulled, the message pulling frequency will increase.