Cloud Message Queue

Migration Description

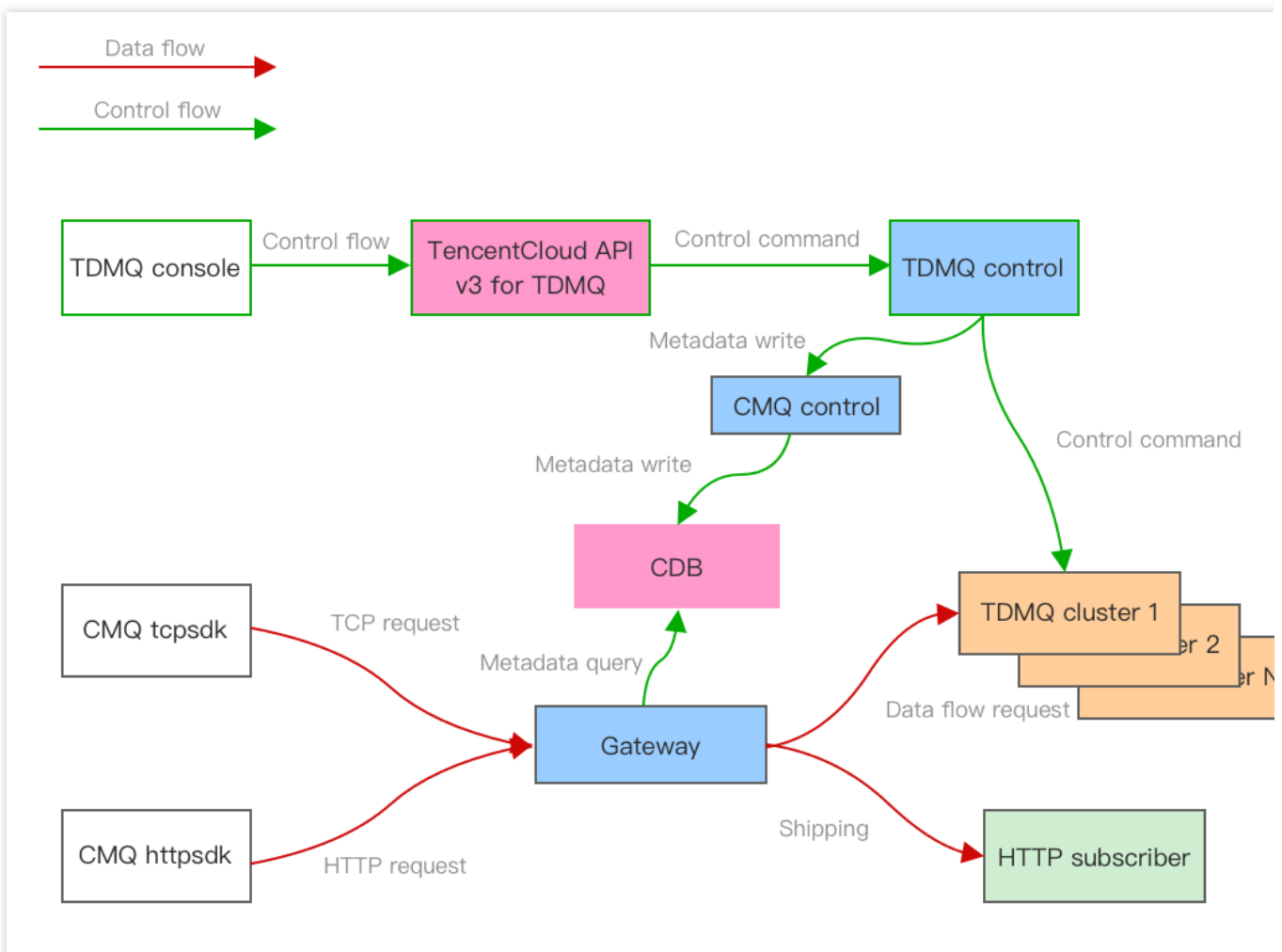# Product Documentation

Contents

Migration Description
CMQ Queue Migration to TDMQ for CMQ

Last updated：2024-12-18 16:08:12

Overview

This document takes the Java client as an example to describe how to migrate a CMQ queue to TDMQ for CMQ.

Migration principle

Scheme overview
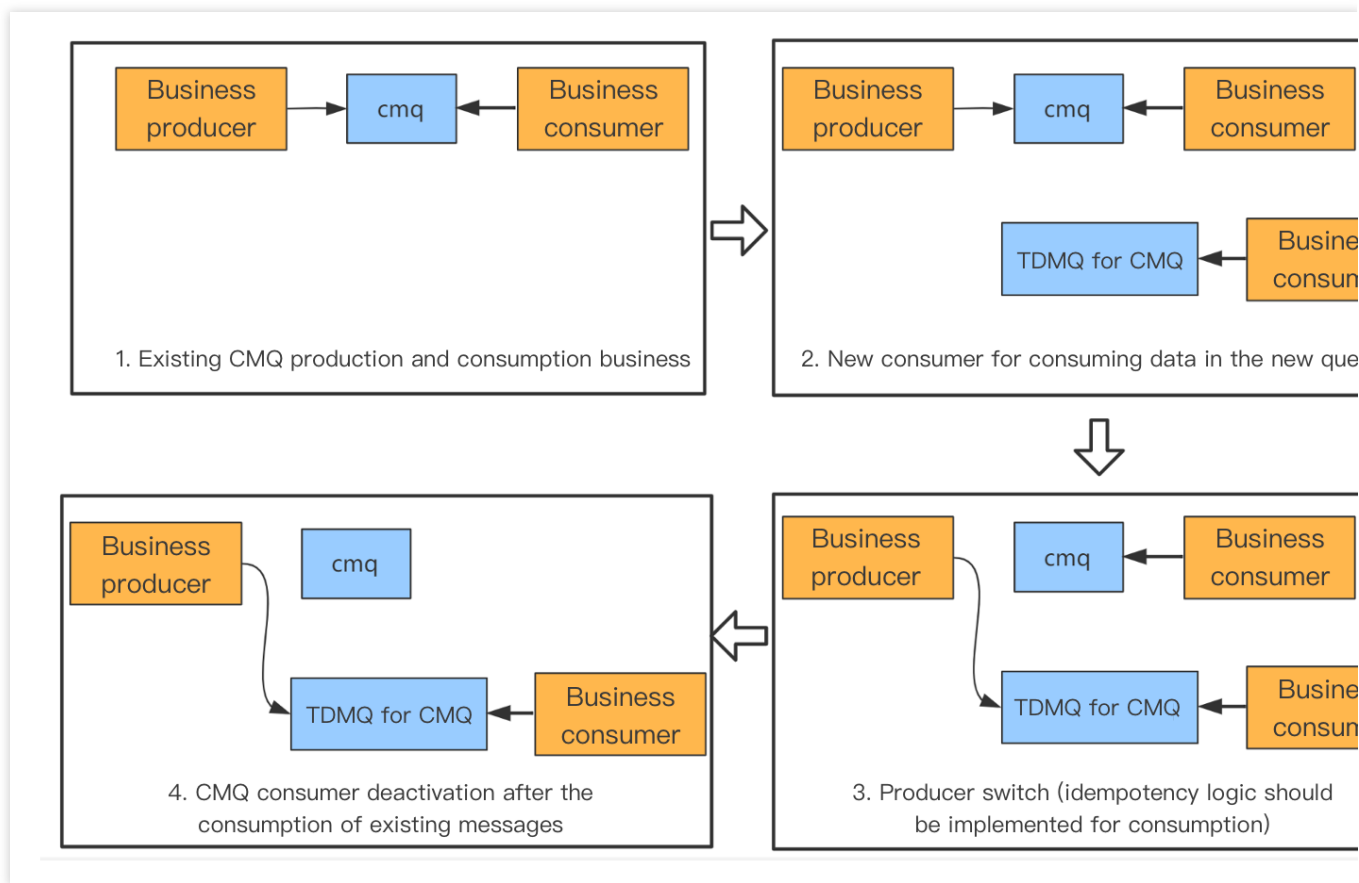
Overall process

Migrate the CMQ queue and topic metadata to TDMQ for CMQ in the CMQ console.

Keep the old consumer unchanged, create a new consumer, and connect it to the TDMQ for CMQ queue.

The producer stops producing messages for the original CMQ queue and switches the production stream to the TDMQ for CMQ queue.

The old CMQ consumer continues consuming existing messages in the original CMQ queue. After the consumption is completed, deactivate it.

Prerequisites

You have deployed CMQ queue producer and consumer services as instructed in SDK for TCP, and they are running normally. The following migration process takes the SDK for TCP as an example.

Directions

Step 1. Migrate the metadata

1. Log in to the CMQ console.

2. Select **Queue** on the left sidebar, select a **Region**, and click **Sync to TDMQ** at the top of the page.

3. In the pop-up window, click **Start** to migrate all queue and topic metadata in the region to TDMQ for CMQ.

4. After the migration is completed, log in to the TDMQ console.

5. Select **Queue Service** on the left sidebar, select the same **Region**, and you can see the queues migrated to TDMQ for CMQ.

Step 2. Create a consumer

1. Create a consumer and connect it to the target TDMQ for CMQ queue.

```
Consumer consumer = new Consumer();
    // VPC address: http://{region}.mqadapter.cmq.tencentyun.com. CVM instances in
    // Public network address: https://cmq-{region}.public.tencenttdmq.com
    consumer.setNameServerAddress("http://****.com");
    // Set `SecretId`, which is required and can be obtained from the console
    consumer.setSecretId("****");
    // Set `SecretKey`, which is required and can be obtained from the console
    consumer.setSecretKey("****");
    // Set the signature algorithm, which is optional and is SHA1 by default
    consumer.setSignMethod(ClientConfig.SIGN_METHOD_SHA256);
    // Set the maximum number of messages that can be pulled in batches, which ran
    consumer.setBatchPullNumber(16);
    // Set the wait timeout period, which is 10s by default. You can pass in the s
    consumer.setPollingWaitSeconds(6);
    // Set the request timeout period, which is 3,000 ms by default
    // If you set the wait timeout period to 6s and request timeout period to 5,00
    consumer.setRequestTimeoutMS(5000);
    // Name of the queue from which messages are pulled
    final String queue = "****";
```

NameServerAddress: API call address, which can be copied in **Queue Service** > **API Request Address** in the TDMQ for CMQ console.

SecretId and SecretKey: TencentCloud API key, which can be copied in **Access Key** > **API Key Management** in the CAM console.

queue: Queue name.

2. Run the code and check whether the consumer service can run normally with no errors reported.

3. In **Queue Service** > **Send Message** in the TDMQ for CMQ console, send a test message to the message recipient to check whether the consumer service can consume normally.

Step 3. Switch the production stream

1. Change the `NameServer` value of the original producer to the access address of the TDMQ for CMQ queue, which can be copied in **Queue Service** > **API Request Address** in the TDMQ for CMQ console.

2. Run the message production program to check whether the producer service can send messages normally.

Step 4. Deactivate the old consumer

After the old CMQ consumer has consumed all existing messages in the original CMQ queue, deactivate it.

On the **Monitoring** page in **Queue Service** > **Queue** in the CMQ console, you can view the number of heaped messages in the CMQ queue. If it is 0, existing messages in the original CMQ queue have all been consumed.

FAQs About Migration

Last updated：2024-12-18 16:08:12

This document provides solutions to common problems during migration from CMQ to TDMQ for CMQ.

Basic Data Migration

Problem

The numbers of subscribers displayed on the basic information pages in the TDMQ for CMQ and CMQ consoles are different.

Cause

The difference lies in the queues manually deleted in CMQ. CMQ doesn't strictly filter out deleted queues when you view subscribers, although these "extra" subscribers have already lost their consumption logic. By contrast, in the subscription logic of TDMQ for CMQ, once a queue is deleted, subscriptions to it will also be deleted, and relevant subscribers won't be displayed.

Solution

To check this problem, search for the subscribed queues that are "not migrated" in the queue list in the CMQ console. If no results are returned, the above reason can explain the problem of missing subscriptions in TDMQ for CMQ.

Longer Consumption Latency

Problem

After switching to TDMQ for CMQ, you find that it takes more time (10–20s) to pull messages. This is a known problem that usually occurs in scenarios with a small number of messages and consumers. In the distributed architecture of TDMQ for CMQ, the randomness of polling nodes to pull messages is more obvious when the number of messages is small, so the underlying layer may poll repeatedly, which increases the latency.

Solution

1. Decrease the value of the **Long Polling Wait Time for Message Receipt** parameter in queue attributes. We recommend you set it to 3 seconds.
2. Clusters specially optimized for this scenario are provided to reduce the polling time. **You can switch the client access points to such clusters to pull messages more quickly**, with no need to create queues again.
Currently, such clusters are available in the following regions:

Region	Access Point
Guangzhou	http://gz.mqadapter.cmq.tencentyun.com:8080 (private network) https://cmq-gz.public.tencenttdmq.com:9443

Shanghai	http://sh.mqadapter.cmq.tencentyun.com:8080 (private network) https://cmq-sh.public.tencenttdmq.com:9443 (public network)
Beijing	http://bj.mqadapter.cmq.tencentyun.com:8080 (private network) https://cmq-bj.public.tencenttdmq.com:8443 (public network)
Hong Kong (China)	http://hk.mqadapter.cmq.tencentyun.com:8080 (private network) https://cmq-hk.public.tencenttdmq.com:8443 (public network)

Incompatibility of Control APIs (for Adding and Viewing Queues)

Problem

Control methods such as `createQueue` in the SDK report errors after the switch.

Cause

TDMQ for CMQ is incompatible with the control operations in the CMQ SDK, such as creating queues and viewing the queue list. We have upgraded TencentCloud APIs from v2 to v3 in the console and no longer provide APIs over the v2 protocol. Therefore, TDMQ for CMQ control APIs need to be transformed based on the latest TencentCloud API protocol. After the transformation, the performance and developer-friendliness will be higher. For more information, see API Documentation.

TDMQ for CMQ separates control flows and data flows for better service. In principle, they have different call scenarios and characteristics. Therefore, their call domain names are also different.

You can get the call addresses of data flows in the console. For the call addresses of control flows, refer to the specific TencentCloud API documentation, such as CreateCmqQueue.

Error with SDK Using TCP Protocol in TDMQ for CMQ

TDMQ for CMQ supports the HTTP protocol only by default. If you use an SDK over the TCP protocol in CMQ and want to seamlessly migrate it, use the following TCP-specific access points:

Region	Access Point
Guangzhou	http://gz.mqadapter.cmq.tencentyun.com:12000 (private network) https://cmq-gz.public.tencenttdmq.com:12000 (public network)
Shanghai	http://sh.mqadapter.cmq.tencentyun.com:12000 (private network) https://cmq-sh.public.tencenttdmq.com:12000 (public network)
Beijing	http://bj.mqadapter.cmq.tencentyun.com:12000 (private network) https://cmq.bj.public.tencenttdmq.com:12000 (public network)

To solve this problem, configure the TCP client to use the above access points.

Difficult Migration of Complicated System

If your system using CMQ is complicated or is hard to maintain and migrate for various reasons, submit a ticket for assistance, and we will provide a backend switch scheme for migration.

Note：

The backend switch will increase the latency.

After the backend switch is completed, do not add or edit items in CMQ.

Parameter Differences

Last updated：2024-12-18 16:08:11

Parameter Differences Between TDMQ for CMQ and CMQ

The usage and syntax of the data flow (messaging) SDK of TDMQ for CMQ remain unchanged, but some parameters and features are different from those of CMQ. Given these differences, TDMQ for CMQ has pertinent parameters configured specially to ensure that the original production and consumption logic will stay the same after migration. However, we recommend you configure new queues and topics based on the logic of TDMQ for CMQ.

Message lifecycle

TDMQ for CMQ adopts the message lifecycle model commonly used in the industry, where time to live (TTL) and retention are used together to avoid high message queue load caused by excessive heap of messages. When the utilization of the capacity for heaped messages reaches 100%, writes will be impossible.

Compared with CMQ, TDMQ for CMQ has a new configuration item of maximum message unacknowledged time ranging from 30 seconds to 12 hours. If the consumer client fails to acknowledge a received message within this time period, the server will automatically acknowledge the message.

Unacknowledged messages will be stored in the message queue and will not be deleted. Acknowledged messages are subject to the message rewindable time and rewindable disk space. If message rewind is not enabled, messages will be deleted directly once acknowledged.

TDMQ for CMQ removes the limit on message lifecycle. By default, messages can no longer be heaped in the queue for a long time. If there are special needs, you can enable message rewind and set the rewindable time range. Only messages with message rewind enabled can be stored in the message queue for more than 12 hours. Once enabled, this feature incurs storage fees as detailed in TDMQ for CMQ's billing overview.

Note:

For queues migrated from CMQ to TDMQ for CMQ, if the lifecycle is set to above 12 hours, the maximum message unacknowledged time will be adjusted to 12 hours; otherwise, it will inherit the lifecycle value in CMQ.

Message heap limit

TDMQ for CMQ removes the limit on the number of heaped messages. In theory, an unlimited number of messages can be heaped as long as the storage capacity is sufficient. However, from the perspective of hardware, we allocate a maximum capacity for heaped message of 10 GB to each queue. You can configure CM alarms based on this value. Generally, about 10 million messages with an average size of 1 KB each can be heaped. If you expect that the heap may exceed the upper limit after migration, submit a ticket for assistance.

Message size

TDMQ for CMQ no longer supports setting the maximum message size. In order not to affect the business migration from CMQ, the original maximum message size of 1,024 KB is set for all new queues.
Note:
 You cannot set the message size for queues migrated from CMQ. If you need to add a limit, create new queues in TDMQ for CMQ.

Long polling wait time for message receipt

The meaning of this parameter remains the same, but its effect is different in TDMQ for CMQ, where we recommend you keep it below 3s.
In TDMQ for CMQ, if the long polling wait time for message receipt is too long, as the underlying layer needs to ensure the semantics of "consumption at least once", the repetition rate of message delivery may increase significantly, causing a severe impact on some downstream services without message deduplication implemented. Therefore, if you want to reduce the probability of message duplication, set this parameter as small as possible (3 seconds is recommended, as basically no repeated delivery will occur under this value).

Capacity for unacknowledged messages

TDMQ for CMQ has a limit on the capacity for unacknowledged messages, ensuring that the memory usage of the message queue server is controlled for guaranteed stability. If the client fails to acknowledge messages in a timely manner, an excessive number of invisible messages are generated. This metric has a monitoring chart. If it surges, check whether the consumer's acknowledgment and deletion logic works properly. If the capacity is insufficient under the surge, submit a ticket for assistance.