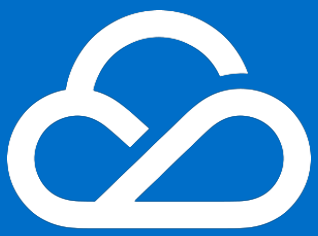
Cloud Message Queue

Message Topic Model

Product Introduction



Tencent Cloud

Copyright Notice

Trademark Notice

Tencent Cloud

All trademarks associated with Tencent Cloud and its services are owned by Tencent Cloud Computing (Beijing) Company Limited and its affiliated companies. Trademarks of third parties referred to in this document are owned by their respective proprietors.

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

Message Topic Model

Tag Matching

Recently, the subscription tag filtering feature has been added to the Topic mode of CMQ. It is similar to the direct_routing mode of rabbitMQ. The topic_pattern mode will be provided at the next iteration. The usage strategy of filter tags is relatively complicated. It will be explained in details below:

Scenario 1:Topic: test1, the subscription publishing API was called at 12:01. This operation is defined "Publish test1", after which the topic delivered 200 messages to each of the three subscribers A, B and C. Assume that A failed to receive 100 of the messages, B failed to received 30 of the messages and C successfully received all of the messages

Scenario analysis:

- Message retention: For the three subscribers A, B and C, we assume there are 110 messages that failed to be delivered (100 failed messages for A, 30 for B, none for C, the failed messages include duplicates). As long as a certain message is associated with one of the subscribers, this message will not be unsubscribed
- Block strategy: Take A as an example. The Topic delivered 200 messages to A, when the 101th message failed, the remaining 99 messages will be blocked. The status is that 100 messages failed to be received
- Retry strategy (backoff retry): Topic-test1 will re-deliver the message to A after every N seconds. The first one of the 100 failed messages will be re-delivered first, this message will be discarded after another three failed retries, after which the topic will retry the next message, so forth
- Retry strategy (exponential decay retry): Take A as an example. The Topic will deliver the 100 messages in a concurrent manner (the order of delivery is not fixed). If the first message failed to be received by the subscriber, the topic will retry from the first message. And if retry fails, this message will continue to be blocked. Newly added messages to be delivered will continue to be blocked
- Message life cycle cannot be prolonged. Assume there are 110 messages retained in topic-test1. Their life cycle is 1 day. Starting from the time point when the producer pushed these messages to the Topic, the messages will be deleted after 1 day has elapsed, no matter how many retries were committed

- Re-push: The producer keeps producing new messages into the Topic. The customer called the subscription publish API again at 12:02. Together with the 110 failed messaged previously retained, we assume that there are now 210 messages in the Topic (110 retained failed messages + 100 newly added messages within the minute), then try to deliver again. In this case, the retry strategy for A and B is exponential decay retry, and the status is "connection lost". The 210 messages will continue to be blocked. C will only receive the newly added 100 messages.

Key point: The ID of each message is the key, and value is the associated subscribers, which indicates whether the subscribers have successfully consumed the message

---

Scenario 2:Topic: test2, the subscription publishing API was called at 12:01. This operation is defined "Publish test2", after which the topic delivered 200 messages to each of the three subscribers A, B and C. Assume that all three of them successfully received all the 200 messages

Scenario analysis:

- For topic-test2, there are only three subscribers A, B and C, all of whom successfully consumed the 200 messages. In this case, the topic will immediately delete these messages

---

Scenario 3:Assume we have four subscribers A, B, C and D. We add tags for these subscribers: The message tag for A is "apple", B is "xiaomi", C doesn't have any configured tags, and D is "imac"+"xiaomi". Now, a producer published 100 messages to the topic, and the tags for message filtering are: apple, imac, iphone, macbook. Then the topic delivers the messages to A/B/C/D.

Scenario analysis:

- For subscriber A, "apple" matches with the message filtering tag "apple", so he/she will receive the 100 messages
- Subscriber B will not receive any messages
- Subscriber C will not receive any messages either
- For subscriber D, one of the tags "imac" matches with the filtering tag, so he/she will receive the 100 messages

---

Scenario 4:Assume there are only four subscribers for Topic: A, B, C and D. None of them have any tags configured. Now, a producer published 100 messages to the topic, and the tags for message filtering are: apple, imac, iphone, macbook. Then the topic delivers the messages to A/B/C/D.

Scenario analysis:

- No tags are configured for the subscribers A, B, C and D, so message filtering is not required when delivering. All of them will receive the 100 messages

---

Scenario 5:Assume there is only one subscriber for Topic: A. The subscription tag "xiaomi" has been configured for A. Now, a producer published 100 messages to the topic, and the tags for message filtering are: apple, imac, iphone, macbook. Then the topic delivers the messages to A

Scenario analysis:

- For subscriber A, the subscription tag does not match, thus A will not receive the 100 messages. In this case, the messages will be immediately discarded, without being retained in CMQ
- When the producer publishes messages to topic, message filtering tags can only be configured for once before publishing. The tags will then bind to the message id and can no longer be modified

---

Summary:

1. If there is no message tags but the subscriber has tags, the subscriber does not have a matching tag and will not receive the message
2. If the message has tags but the subscriber does not, message matching is not required and the subscriber will receive the message
3. If both the message and subscriber has tags, then the subscriber can receive the message only if there is a pair of matching tags. N:M matching is supported. For example, a message has 10 tags, a subscriber has 4 tags, there is a pair of matching tags, then the subscriber will receive the message
4. Neither the message nor the subscribers have tags. All subscribers will receive the message upon delivery

Creating a Topic

When creating a Topic, users need to specify the following attributes:

1) Enter Topic name. Topic name cannot be changed once it has been created. Name can only contain letters, numbers, dashes (-) and underscores (_). The length of the name must be between 3-64 Bytes, the name will be truncated if it goes beyond this limit

2) Configure region attribute of the Topic

3) Configure the maximum length of messages (the MaximumMessageSize attribute). This determines the max length of message bodies that can be sent to this Topic. (Unit: Byte) Value range: 1024 (1 KB)-65536 (64 KB). Default is 65536.

4) Enter notes

5) Message life cycle: The longest time that messages will be kept in this queue. The message will be deleted after this time has elapsed since the message has been sent to this queue, regardless of whether the message has been pulled or not. (Unit: second) Default is 86400s, this value cannot be changed.

6) Message retention: Enabled by default. When the message of a producer has not been triggered and delivered to the subscriber, or the subscriber failed to receive the message, the message will be retained into Topic temporarily

7) Current number of retained messages: You can view the approximate number of retained messages in the details section of the topic

8) The Qps performance of a single topic: The peak production and delivery QPS performance of a single Topic is 5000

Modifying Topic Properties

There are the following restrictions when a user modifies Topic attributes:

1) The name and resource ID of a Topic cannot be changed

2) The region attribute of a Topic that is displayed cannot be changed

3) The maximum length of messages can be changed. This determines the max length of message bodies that can be sent to this Topic. (Unit: Byte) Value range: 1024 (1 KB)-65536 (64 KB). Default is 65536.

4) Creation time and last modification time cannot be changed

Subscribing Topics

You can subscribe a Topic by specifying the following attributes:

1) Enter Topic name

2) Enter Topic resource ID

3) Enter subscription name. This cannot be changed once it has been entered

4) Enter subscription terminal protocol. Options are: Queue message service, URL address

5) Subscription address. Enter URL or Queue name here. Currently, a Topic is only allowed to send messages to Queues under the same account

6) Retry strategy: The NotifyStrategy attribute of subscription. This is the retry strategy in case of errors when pushing messages to the receiving end. This strategy is enabled by default.
You can select one of the two options: a. backoff retry: Message push is retried for three times, the interval between two reties is a random value between 10s and 20s. After three retries, this message will be discarded for this subscriber, with no further retries; b. exponential decay retry: Message push is retried 176 times, the total time for the retries is one day, intervals between retires are 2^0, 2^1, ..., 512, 512, ..., 512 seconds. The

exponential decay retry

strategy is checked by default. The user must check one of the two options

7) Retry verification: (When using HTTPS) a return code of 200 indicates success

8) Add subscriber tag: When adding a subscriber, you can add FilterTag to make the subscriber receive only the messages with this filtertag. Each tag is a string with no more than 16 characters. There can be at most 10 tags for a single subscriber. The subscriber will receive the message delivered by this topic as long as one of the tags matches with the filter tag of the topic. Subscriber cannot receive messages without any tags.

9) Subscriber limit for a single Topic: Up to 100 subscribers can be associated under a single Topic

10) Total number of messages associated with the subscriber: This is an approximate value. It indicates the number of messages that are waiting to be delivered (or waiting for retry) to the subscriber, under a certain Topic

Routing Key Matching

The routing key matching feature of CMQ is similar to the exchange queue of rabbitMQ. It is used to filter messages, and make subscribers receive different messages based on different conditions. You can enable "Routing Key Matching" when creating Topic.

Instructions:

- Binding key and Routing key are used together, they provide a message filtering function which is similar to RabbitMQ. The Routing key used when sending messages is provided by the client when it sends messages. The Binding key used when creating subscription relationships is the binding relationship between Topic and Subscriber.

Restrictions:

1. There can be at most five Binding keys. Binding key is used to indicate the routing path for sending messages, it must be no longer than 64 Bytes, and can contain up to 15 ".", i.e. 16 phrases at most

    1. Routing key consists of a string. Routing key is used to indicate the routing path for sending messages, it must be no longer than 64 Bytes, and can contain up to 15 ".", i.e. 16 phrases at most

Wildcards:

1. Asterisk (*), can be used in place of a word (a string of continuous characters)

1. Pound sign (#), can be used to match with one or multiple characters

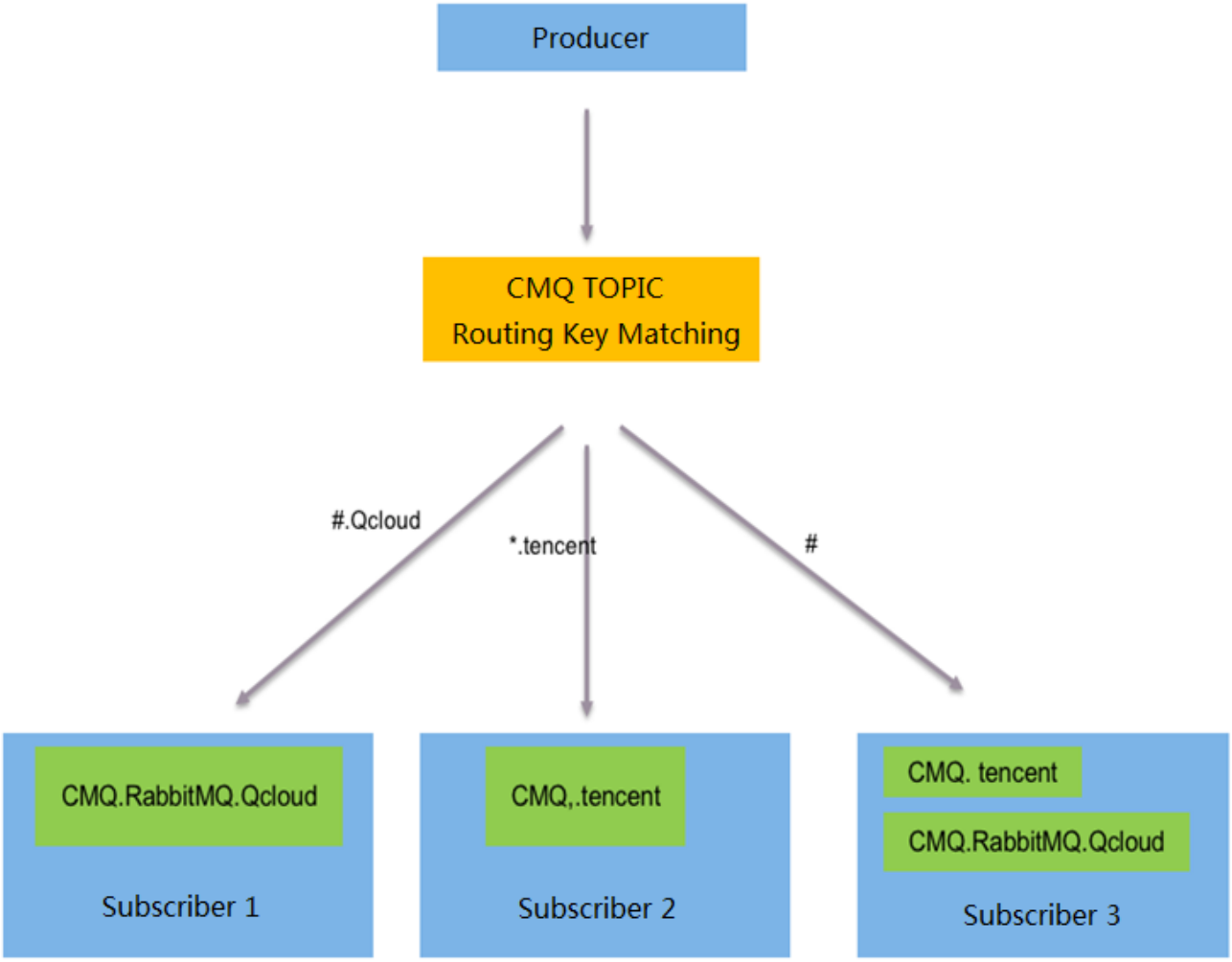
For example:

1. Subscriber is "1.*.0", and message is "1..0", then the subscriber will receive the message

1. Subscriber is "1.#.0", and message is "1.2.3.4.4.2.2.0", then the subscriber will receive

the message (the middle part of the message can be anything)

1. Subscriber is "#", then the subscriber will receive all messages



Publishing Messages to Topics

The producer pushes messages to Topic by specifying the following information:

1) Topic name

2) Topic resource ID

3) Published title. Users may enter the title according to their own needs

4) Published content: Main body of the message, customers may enter the content according to their own needs. CMQ will not encode or modify this content

5) Add message filter tag: Tag, namely message tag, message type, is used to differentiate message categories under the Topic of a certain CMQ. Consumers are allowed to filter messages based on the tags and thus only consume the messages that they're interested in. This feature is disabled by default. In this case, all messages will be sent to all the subscribers. If a subscriber configured a tag, the subscriber will not receive any messages because the tag doesn't match with the tag of the Topic. The message filter tag describes the tag used for message filtering for this subscription (only the messages with consistent tags will be pushed). Each tag is a string with no more than 16 characters. There can be at most 10 tags for a single message

Delivering Messages to Subscribers

The Topic delivers messages to subscribers while following the rules below:

1) The Topic will do its best to deliver messages published by producers (notification) to subscribers

2) If the delivery isn't successful after multiple retries, the message will be retained in Topic and wait for the next delivery. If still not successful, the message will be discarded after the maximum message life cycle (1 day) has elapsed