

# **TDMQ for CKafka**

## **FAQs**

### **Product Documentation**



## Copyright Notice

©2013-2025 Tencent Cloud. All rights reserved.

Copyright in this document is exclusively owned by Tencent Cloud. You must not reproduce, modify, copy or distribute in any way, in whole or in part, the contents of this document without Tencent Cloud's the prior written consent.

## Trademark Notice



All trademarks associated with Tencent Cloud and its services are owned by the Tencent corporate group, including its parent, subsidiaries and affiliated companies, as the case may be. 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

## FAQs

Instance

Topic

Consumer Group

Client

Network

Monitoring

Message

General

Connector

Task

Data Reporting Issues

Data Processing Issues

Data Dump

Data Subscription Issues

# FAQs

## Instance

Last updated : 2024-01-09 15:02:48

### How do I select the instance specification?

You can use the specification calculator in the [console](#) to calculate the required production specification.

### Specification Calculator

1 On-Premises Kafka Specification

>

2 Recommended Specification

Kafka Version

0.10.x

Suggestions for CKafka Version Selection

Select 0.10.x if the CKafka version is v0.10 or earlier

Peak Application Bandwidth

40

MB/s

Peak application bandwidth = max(peak production bandwidth \* the number of replicas, peak consumption bandwidth)

Disk

300

GB

Estimate this value based on your disk's current message heap peak

Total Partitions

60

Count

This value indicates the total number of partitions to migrate and is proportional to the number of replicas. For example, a single-replica topic has five partitions while a dual-replica topic has ten. CKafka does not support single-replica topics.

Multi-AZ Deployment

☐ Yes ☒ No

Data Compression

Disabled

CKafka does not support gzip format. For details, see [Data Compression](#)

Next

CKafka instances are divided into Standard Edition and Pro Edition according to their specifications. For the comparison between the two editions, see [Product Specifications](#).

You can select an appropriate product specification as needed. For the billing rules of different specifications, see [Billing Overview](#).

## What are the modes of instance upgrade? What are their differences?

The migration modes of the Standard and Pro Editions are lossless, non-stop, and imperceptible to the client.

Standard Edition uses a shared cluster. If the available resources in the cluster are sufficient, instances will be upgraded by increasing the quota, with no data migration involved; therefore, upgrade can be completed in several minutes normally. If the available resources in the cluster are insufficient, the system will prompt that the resources are insufficient, and you need to submit a ticket for data migration and resource allocation on the backend and then perform normal Standard Edition instance upgrade in the console.

Pro Edition uses a dedicated cluster. When the system algorithm finds that the current dedicated resource pool cannot sustain the resource quota after instance upgrade, it will automatically add resources to the resource pool and migrate instance data. Therefore, Pro Edition has two instance upgrade modes:

When the resources are sufficient, instances will be upgraded by increasing the quota, which can be completed in several minutes.

When the resources are insufficient, data migration will be involved, which may take several minutes to hours subject to the size of heaped data in the cluster.

If data migration is required, you can customize the migration time:

### Upgrade Configuration

Original Configuration

Model	Peak bandwidth (MB/s)	Disk capacity (GB)	Partition Limit
Basic Type	40	500	

Target Model

Basic Type

High Performance Type

Peak Bandwidth

1600

10000

1600

+

MB/s

Disk capacity

10000

100000

10000

+

GB

Topic Limit

Up to 2000 topic(s) available.

Partition Limit

4000

+

Rebalance Time

☒ Execute

☐ Custom (a point in the next 24 hours)

2021-09-10 14:26:31

Upgrade Mode

☒ Stable (about 0 h 35 min)

CKafka will limit the speed of data migration during the downgrade to ensure that instances have optimal bandwidth performance. This option is recommended if you do not want to affect business operations.

☐ High-speed (about 0 h 10 min)

CKafka will not limit the speed of data migration during the upgrade, thus affecting the bandwidth available for data production and consumption of instances. This option is recommended for off-peak hours or when service suspension is allowed.

Total Cost

Submit

Cancel

If data migration is required, you can select a migration mode and set scheduled migration. There are two migration modes:

**Stable:** CKafka will limit the speed of data migration during the upgrade to ensure that instances have optimal bandwidth performance. This option is recommended if you do not want to affect business operations.

**High-speed:** CKafka will not limit the speed of data migration during the upgrade, thus affecting the bandwidth available for data production and consumption of instances. This option is recommended for off-peak hours or when service suspension is allowed.

As migration will use cluster bandwidth and disk resources, you may worry about that the migration will affect the business during peak hours. In this case, you can set scheduled migration; for example, you can set the start time of instance upgrade to midnight. For detailed directions of instance upgrade, see [Upgrading Instance](#).

## Do instances support data compression?

Currently, CKafka supports open-source Snappy and lz4 message compression formats. Gzip compression is not supported currently as it requires high CPU usage. If compression is enabled, it may increase the server pressure in some cases; for example, if the producer protocol version is high, but the consumer protocol version is low, downward version conversion may occur during consumption, increasing the server pressure. Therefore, we recommend you disable message compression before performing a stress test.

Configuration Method: In the configuration file of the producer, set the `compression.type` parameter to `snappy` or `lz4`. The default value is `none`, indicating that the feature is disabled.

We recommend you preferentially select Snappy for compression. After compression is enabled, the server CPU utilization and the production and consumption time may increase. Therefore, you need to enable it with caution.

# Topic

Last updated : 2024-01-09 15:02:48

## How do I choose an appropriate number of CKafka replicas?

We recommend that you select two or three replicas for data storage when creating a topic to ensure data reliability. A created topic has two replicas by default. If your business requires higher availability, you can select three replicas. If you need more replicas, you can [submit a ticket](#) for assistance. When creating a topic, you can select the number of replicas as shown below:

### Create Topic

Name

Please enter topic name

Remarks

Optional. Please enter remarks.

Partition Count ⓘ

1

+

Count

Max number of partitions per topic: 5000

Number of Replicas ⓘ

2 replicas

3

If you select n replicas, up to (n-1) replica(s) are allowed to be down. Supported partition count \* replica count. Replica quota is 60, with 14 used in the quota. You can also create up to 23 partitions with 2 replicas now.

For more partitions, you can upgrade instances. For rule details, please see [Documentation](#).

Allowlisted ⓘ

[Show advanced configuration](#)

Submit

Cancel

To improve data security, CKafka has banned the creation of single-replica topics currently. If you have a single-replica topic in your account, please migrate it as follows:

1. Create a topic, select the same partition parameter, and select "dual-replica".
2. Produce messages in the new topic while the existing single-replica topic continues to be consumed.
3. Modify the consumer configuration after consumption is completed to subscribe to the new topic for consumption.



## Why can messages still be queried after the message retention period set for the topic elapsed?

1. A timestamp field and timestamp types are added to messages. Currently, two timestamp types are supported:

`CreateTime` and `LogAppendTime`. The former indicates the time when the message is created by the producer, and the latter indicates the time when the broker receives the message. If the timestamp data of the time when a client produces a message is invalid, data deletion on the broker server will be affected.

2. If there are too many partitions in the topic and too little message data, and only one log segment file exists in the partition, messages will not be deleted.

3. The log deletion task checks whether the current log size exceeds the set threshold, i.e., 1 GB per segment. If the maximum timestamp data in the log segment is still within the retention period, messages will not be deleted.

## Why is the number of topics (total number of partitions) limited?

A high number of topics (total number of partitions) in Kafka will compromise the cluster performance and stability.

As the number of partitions that a server can sustain is limited, if you want more partitions, you need to add more nodes, which incur more fees. Kafka's storage and coordination mechanisms work by partition. If there are too many partitions, storage fragmentation will be severe, random writes on individual servers will increase, the efficiency of leader switch in the cluster will decrease, controller node failover will slow down, and other problems may occur, which lower the cluster performance and stability.

## What is the relationship between the number of topics and the number of partitions?

CKafka uses partition as an allocation unit.

Total number of partitions = topic A *number of replicas* + topic B *number of replicas* + ... + topic N \* *number of replicas*.

Replicas are also counted into the total number of partitions. For example, if you create 1 topic with 6 partitions and 2 replicas for each partition, then you have a total of 12 partitions (1 6 2).

Partition count: it is a concept in physical partition, where one topic can contain one or more partitions.

Replica Count: The number of partition replicas is used to ensure the high availability of the partition. To ensure data reliability, creating a single-replica topic is not supported. Two replicas are enabled by default.

# Consumer Group

Last updated : 2024-01-09 15:02:48

## How do I set a reasonable number of consumers?

The relationship between consumer group and consumer is as follows:

A consumer can subscribe to multiple topics at the same time.

A topic can contain one or multiple partitions.

A partition can be consumed only by one consumer at a time.

Therefore, the maximum number of consumers in a consumer group is the total number of partitions in all topics.

Consumer refers to a consumer object in programming. There can be multiple consumers on a server. This is because a thread contains a consumer, and if multiple threads are initiated, there will be multiple consumers as shown below:

```
Properties props = new Properties();
props.put(ConsumerConfig.BOOTSTRAP_SERVERS_CONFIG, bootstrap);
KafkaConsumer<String, String> consumer = new KafkaConsumer<>(props);
consumer.subscribe(Arrays.asList(topic));
while (true) {
    ConsumerRecords<String, String> records = consumer.poll(100);
}
```

You can configure the initial number of consumers based on the client resources and configure heap alarms for consumer groups. If messages are heaped, you can add more consumers. For more information on how to configure an alarm, see [Configuring Alarm](#). The configuration page is as shown below:

## Configure Alarm Policies

Alarm Object ⓘ Instance ID ▼ Select object ▼

Trigger Condition ☐ Select template ☒ Configure manually ☒ Use preset trigger conditions ⓘ

## Metric Alarm

When meeting any ▼ of the following metric conditions, the metric will trigger an alarm.

Threshold Type ⓘ ☒ Static ☐ Dynamic ⓘ

▶ If instance disk usa... (statistical period ▼) > ▼ 80 ▼ % at 5 consecutive ▼ then Alarm every 30 min. ▼ ⓘ 🗑️

Threshold Type ⓘ ☒ Static ☐ Dynamic ⓘ

▶ If connect\_percent... (statistical period ▼) > ▼ 80 ▼ % at 5 consecutive ▼ then Alarm every 30 min. ▼ ⓘ 🗑️

Threshold Type ⓘ ☒ Static ☐ Dynamic ⓘ

▶ If produce\_bandwi... (statistical period ▼) > ▼ 80 ▼ % at 5 consecutive ▼ then Alarm every 30 min. ▼ ⓘ 🗑️

Threshold Type ⓘ ☒ Static ☐ Dynamic ⓘ

▶ If consume\_bandw... (statistical period ▼) > ▼ 80 ▼ % at 5 consecutive ▼ then Alarm every 30 min. ▼ ⓘ 🗑️

[Add Metric](#)

# Client

Last updated : 2024-01-09 15:02:48

## What should I do if no messages are displayed in the Kafka console when I test the client?

If the "latest" option is used, the consumer can only get the last messages, and production needs to be ongoing so that the messages can be seen.

Change to the "earliest" option for data consumption.

## What should I do if a production/consumption error occurs after a new client is connected to the service?

Check whether telnet works. It might be a network issue. Check if Kafka and the producer are in the same network.

Check whether the accessed "vip - port" is correctly configured.

Check whether the topic allowlist is enabled. If yes, you need to configure the correct IP for access.

## How can I ensure that messages produced by the client are sequential in the same partition?

If the topic has only one partition, messages will be stored in the order in which they are received by the server. This makes messages sequential in the partition.

If a topic has multiple partitions, you can specify keys for messages of the same type on the producer. If such messages are sent with the same key, CKafka will select a partition to store them using the hash algorithm according to the key. As a partition can be listened on and consumed by only one consumer, messages will be consumed in the sending order.

A single producer produces messages to a single partition sequentially.

## How many connections does a client generally establish to brokers for message production?

For a single client instance (a producer object created with the `new` command), the total number of connections established between it and all servers ranges from one to n (n refers to the number of brokers).

Each Java producer manages TCP connections as follows:

1. The `Sender` thread will be initiated when a `KafkaProducer` instance is created to establish TCP connections to all brokers in `bootstrap.servers`.
2. After the metadata on the `KafkaProducer` instance is updated, TCP connections to all brokers in the cluster will be established again.
3. If no TCP connections are found when a producer sends a message to a broker, a connection will be established immediately.
4. If you set the `connections.max.idle.ms` parameter on the producer to a value above 0, TCP connections established in step 1 will be closed automatically. The parameter value is 9 minutes by default; that is, if no requests

are sent through a TCP connection in 9 minutes, Kafka will automatically close the connection. If you set the parameter to -1, TCP connections established in step 1 cannot be closed and will become "zombie" connections.

## How do I know whether a message is successfully sent from a client?

After sending a message, most clients will return a `Callback` or `Future`. A successful callback indicates that the message is successfully sent.

You can also check whether a message is successfully sent in the console by the following methods:

View the topic partition status to see the number of messages in each partition in real time.

View the topic traffic monitoring data to see the traffic curve of the produced messages.

You can print the partition information returned by the `send` method to check whether the message is successfully sent:

```
Future<RecordMetadata> future = producer.send(new ProducerRecord<>(topic, messageKey));
RecordMetadata recordMetadata = future.get();
log.info("partition: {}", recordMetadata.partition());
log.info("offset: {}", recordMetadata.offset());
```

If the partition and offset information can be printed out, the currently sent message has been correctly saved on the server. At this time, you can use the message query tool to query the information of the relevant offset.

If the partition and offset information cannot be printed out, the message has not been saved on the server, and the client needs to retry.

**Message Query** Global

*Message query will take up the bandwidth resources of the CKafka instance. It is recommended that you try reducing the query range and do not perform frequent operations.*

Instance

ckafka-...

Topic

ckafka-...

Query Type

Query by offsetQuery by time

Partition ID

0

Start Offset

0

Query

Partition ID	Offset	Timestamp	Operation
0	137	2021-05-07 17:24:13	<a href="#">View Message Details</a>
0	138	2021-05-07 17:24:13	<a href="#">View Message Details</a>

## What is leader switch?

When a topic is created, the Kafka broker cluster specifies a leader for each partition, and the topic partitions are evenly distributed to each broker.

As time elapses, the partitions may become unevenly distributed across brokers, and the client may throw exceptions such as `BrokerNotAvailableError` and `NotLeaderForPartitionError` during the production or consumption process.

Generally, such issues occur because the partition leader has been switched as described in the following scenarios:

When the broker where a partition leader resides cannot communicate with the broker controller due to some exceptions such as network disconnections, program crashes, and hardware failures,

**leader switch** will occur, that is, the current topic partition leader will be replaced by a new leader elected from among the follower partitions.

When the Kafka cluster sets `auto.leader.rebalance.enable = true` to automatically trigger rebalancing or when rebalancing is manually triggered by the increase/decrease of the number of brokers,

**leader switch** will also occur due to partition rebalancing.

When the leader is switched due to the broker's accidental disconnection:

If the client has set `ack = all` and `min.insync.replicas > 1`, messages won't get lost as they have been acknowledged by both the leader and follower partitions.

If the client has set `ack = 1` , **some messages may get lost** as they may fail to be synced to the follower partitions within the specified `replica.lag.time.max.ms` .

Messages won't get lost if the broker works normally and the leader is switched due to the rebalancing manually/automatically triggered by instance upgrade, the single- to multi-AZ deployment mode switchover, or instance migration. This is because:

If the client has set `ack = all` and `min.insync.replicas > 1` , messages won't get lost as they have been acknowledged by both the leader and follower partitions.

If the client has set `ack = 1` , messages won't get lost as the partition offset will be automatically synced when the leader is switched.

# Network

Last updated : 2025-05-19 11:13:10

## Does CKafka support public network access?

CKafka currently doesnot support public network access.

## How do I continue accessing CKafka over VPC after SASL is enabled?

If you use the PLAINTEXT method to access CKafka while enabling another SASL access routing, the ACL previously set for the topics will still take effect. If you want PLAINTEXT access to be unaffected, please add the read/write permissions of all users for the topics that PLAINTEXT needs to access.

### **Note:**

When adding an ACL policy, you don't need to select any user, and read/write permissions are added to **all users** by default.

The effect after addition is as follows:



# Monitoring

Last updated : 2024-01-09 15:02:48

## How can I ensure that messages produced by the client are sequential in the same partition?

If the topic has only one partition, messages will be stored in the order in which they are received by the server. This makes messages sequential in the partition.

If a topic has multiple partitions, you can specify keys for messages of the same type on the producer. If such messages are sent with the same key, CKafka will select a partition to store them using the hash algorithm according to the key. As a partition can be listened on and consumed by only one consumer, messages will be consumed in the sending order.

A single producer produces messages to a single partition sequentially.

## How many connections does a client generally establish to brokers for message production?

For a single client instance (a producer object created with the `new` command), the total number of connections established between it and all servers ranges from one to n (n refers to the number of brokers).

Each Java producer manages TCP connections as follows:

1. The `Sender` thread will be initiated when a `KafkaProducer` instance is created to establish TCP connections to all brokers in `bootstrap.servers`.
2. After the metadata on the `KafkaProducer` instance is updated, TCP connections to all brokers in the cluster will be established again.
3. If no TCP connections are found when a producer sends a message to a broker, a connection will be established immediately.
4. If you set the `connections.max.idle.ms` parameter on the producer to a value above 0, TCP connections established in step 1 will be closed automatically. The parameter value is 9 minutes by default; that is, if no requests are sent through a TCP connection in 9 minutes, Kafka will automatically close the connection. If you set the parameter to -1, TCP connections established in step 1 cannot be closed and will become "zombie" connections.

## How do I know whether a message is successfully sent from a client?

After sending a message, most clients will return a `Callback` or `Future`. A successful callback indicates that the message is successfully sent.

You can also check whether a message is successfully sent in the console by the following methods:

View the topic partition status to see the number of messages in each partition in real time.


View the topic traffic monitoring data to see the traffic curve of the produced messages.


You can print the partition information returned by the `send` method to check whether the message is successfully sent:

```
Future<RecordMetadata> future = producer.send(new ProducerRecord<>(topic, messageKey));
RecordMetadata recordMetadata = future.get();
log.info("partition: {}", recordMetadata.partition());
log.info("offset: {}", recordMetadata.offset());
```

If the partition and offset information can be printed out, the currently sent message has been correctly saved on the server. At this time, you can use the message query tool to query the information of the relevant offset.

If the partition and offset information cannot be printed out, the message has not been saved on the server, and the client needs to retry.

**Message Query**  Google Cloud

 Message query will take up the bandwidth resources of the CKafka instance. It is recommended that you try reducing the query range and do not perform frequent operations.

Instance

ckafka-...

Topic

ckafka-...

Query Type

Query by offset

Query by time

Partition ID

0

Start Offset

0

Query

Partition ID	Offset	Timestamp	Operation
0	137	2021-05-07 17:24:13	<a href="#">View Message Details</a>
0	138	2021-05-07 17:24:13	<a href="#">View Message Details</a>

# Message

Last updated : 2024-01-09 15:02:48

## How do I calculate the number of unconsumed messages?

Number of unconsumed messages = maximum offset - submitted offset, as shown below:

▼ eb-connector-9q5fgv4y

Empty

consumer

-

[Offset Settings](#)

[View Consumer Det](#)

[Delete](#)

Enter a topic name

Q

↺

york

Partition Name ⚡	Submitted Offset Position ⚡	Max Offset Position ⚡	Unconsumed Messages ⚡	Operation
partition-0	685238003	685238003	0	<a href="#">View Details</a>
partition-1	203483871	203483871	0	<a href="#">View Details</a>
partition-2	112766871	112766871	0	<a href="#">View Details</a>
partition-3	186934869	186934869	0	<a href="#">View Details</a>
partition-4	203537589	203537589	0	<a href="#">View Details</a>

## Can the message retention period be automatically adjusted?

CKafka allows you to add dynamic message retention policies. After such a policy is set, if the disk utilization reaches the specified percentage, a certain proportion of existing data will automatically expire in case of message surges, helping avoid situations where normal production and consumption stop after the disk space is used up. For detailed directions, please see [Adding Dynamic Message Retention Policy](#).

# General

Last updated : 2024-11-07 15:37:37

## What Are the Differences Between CKafka and CMQ?

CMQ provides financial-level high reliability and high data persistence for message transmission, ensuring strong data consistency.

CKafka is more suitable for scenarios demanding higher throughput but can afford relatively lower reliability requirements (such as log aggregation). Additionally, CKafka is fully compatible with existing Kafka users, allowing for zero migration costs and complete instance exclusivity.

## Which version of open-source Kafka is compatible with CKafka?

Currently, CKafka is compatible with open-source Kafka API 0.9, 0.10, 1.1, 2.4, and 2.8, allowing users to migrate data to the cloud at zero cost.

## Which version of open-source Kafka is the current CKafka based on?

The current CKafka is based on Apache Kafka 0.10, 1.1, 2.4, and 2.8. We recommend you use the SDK of the corresponding version for production and consumption.

## Does CKafka expose ZooKeeper?

CKafka does not expose ZooKeeper or its address.

## Does CKafka support public network access?

Currently, CKafka transfers data over the private network by default. As public network access runs the risk of issues such as delay and network environment security, we do not recommend long-term use of public network transfer. If you have a temporary need for public network transfer, please contact your Tencent Cloud account manager for evaluation and assistance.

## Does CKafka support message compression?

Currently, CKafka supports open-source Snappy and LZ4 message compression formats. Because gzip compression consumes more CPU resources, it is currently not supported.

We recommend you disable message compression when testing.

Configuration Method: in the configuration file of the producer, set the `compression.type` parameter to `snappy` or `lz4`. The default value is `none`, indicating that the feature is disabled.

## Can a Kafka client directly connect to the CKafka service?

CKafka is compatible with open-source Kafka 0.9, 0.10, 1.1, 2.4, and 2.8. You can connect to the message center via a Kafka client and deploy codes to Tencent Cloud services to produce or consume messages.

## What are the restrictions on a CKafka instance?

Different instance specifications have different restrictions on peak throughput, disk capacity, number of topics at the instance level, and number of partitions at the instance level. For more information, please see [Billing Overview](#).

## Will CKafka lose messages?

Open-source Apache Kafka does not guarantee no message loss. As CKafka is optimized for availability, Tencent Cloud promises a CKafka availability of over 99.95%.

CKafka users can enable ACK during production to avoid message loss and improve message reliability.

Cluster changes and upgrades complete in seconds and will not affect user experience.

CKafka is mainly used in big data processing scenarios that require high throughput and high performance but relatively low data reliability. In extreme cases, a small number of messages may be lost. If you require zero message loss and have relatively performance requirements, we recommend TDMQ.

## How does CKafka guarantee security?

CKafka guarantees security by using following features:

Tenant isolation: the network access of instances is isolated among different accounts by default.

Permission control: CKafka adds an additional authentication mechanism based on the source IP allowlist at the application layer and supports [SASL authentication](#).

Security protection: services such as multi-dimensional security protection and anti-DDoS attacks are provided.

# Connector Task

Last updated : 2024-01-09 15:02:48

## What should I do if I fail to create a dump task?

You will get an error message in the console if you fail to create a dump task.

Generally, you can troubleshoot the problem based on the error message:

1. If the error message says "Connection to XXX failed. Check whether the username and password are correct", you can check whether the entered username and password are correct, and if so, recreate the task with correct parameters.
2. If the error message says "Connection failed. Check whether the port is opened", you can check whether the port is opened. You can open the port in the security group as described in [Analysis of Change Logs Tracked by MongoDB Change Streams](#).
3. If the error message says "The table xxx does not exist in the database", you can check whether the table exists in the database.
4. If the error message says "createLink fail", you can check whether the network connection fails, and [contact us](#) for assistance.

If the error message is unclear for you to troubleshoot, [contact us](#) for assistance.

## What should I do if the dump task status is abnormal?

You will get an error message in the console if an exception occurs in the dump task.

Generally, you can troubleshoot the problem based on the error message:

1. If the error message says "The source CKafka instance does not exist" or "The target CKafka instance does not exist", you can check whether the instance is abnormal or has been deleted.
2. If the error message says "the source topic does not exist" or "The target topic does not exist", you can check whether the topic in the CKafka instance is deleted.
3. If the error message says "Check whether the account and password are correct", you can check whether the username and password have been changed, and if so, you need to update them in the task.

If the error message is unclear for you to troubleshoot, [contact us](#) for assistance.

## What is task concurrency?

Concurrency is a concept that exists in all non-data reporting tasks, which is set to 1 by default when a task is created. During task execution, the system automatically detects the data heap.

1. If data heap occurs, the system will automatically expand the concurrency to increase the data throughput.
2. If the amount of data is low, the system will shrink the concurrency to avoid unnecessary resource use.

In data processing and data dump scenarios, the concurrency ranges from 1 to the number of partitions in the CKafka topic.

In the data integration scenario, the concurrency will be adjusted based on the upstream engine in the range of 1 to infinite as needed.

## How does MongoDB data integration work?

MongoDB uses change streams to record data changes. When any data in the database changes, the application can get the changed data through the change stream mechanism. You can consider it as a trigger executed in the application. As for what data the application wants to get in what form, the data can be filtered and transformed through the aggregation framework.

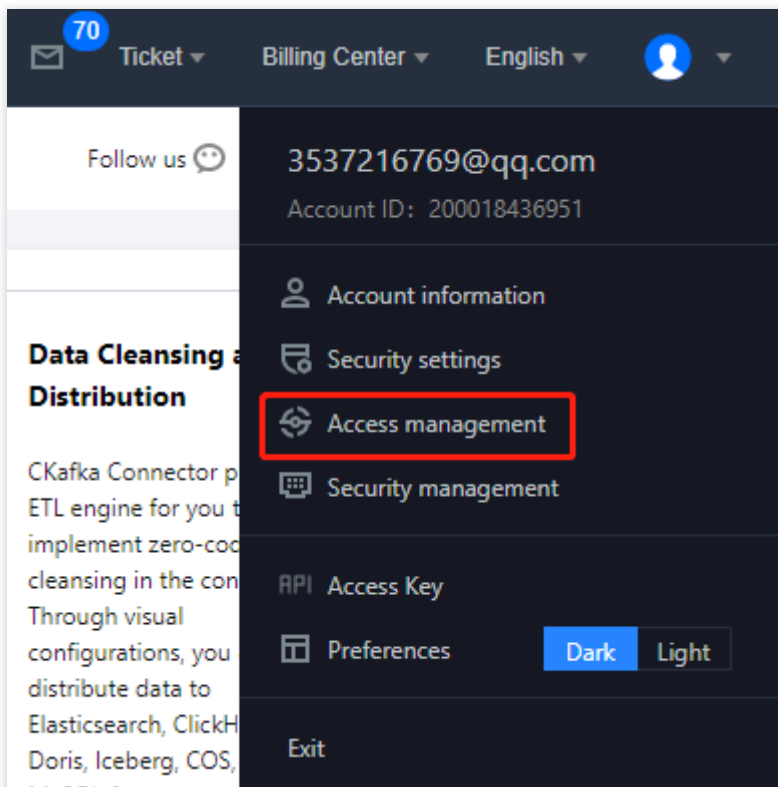
The underlying dependency is the MongoDB's official [Kafka source connector](#). The connector always listens on data changes in the database and stores the changed data in the Kafka topic.

## Is MongoDB data partitionally sequential?

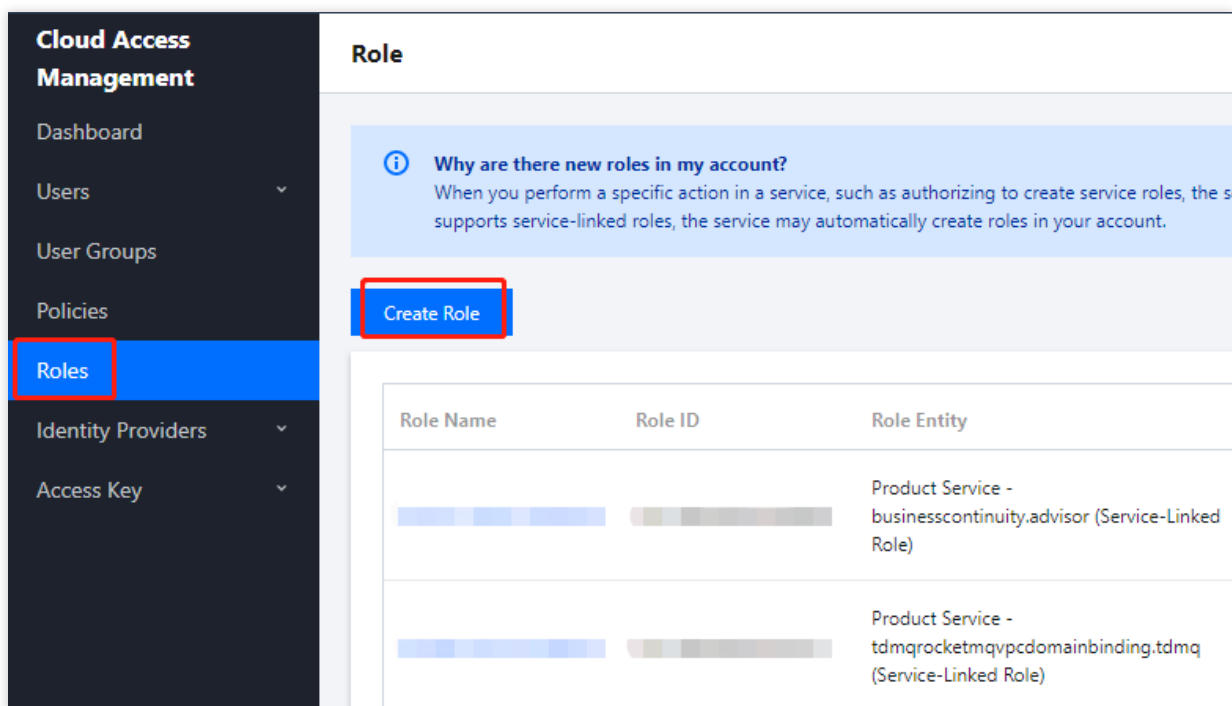
By default, MongoDB data is written to partitions by MongoDB object ID (as the key). Therefore, if the partitions are unchanged, the data can be partitionally sequential. If new partitions are added during subscription, rehashing will occur, and the changed data of the same data record during consumption will become non-sequential for a short time.

## What should I do if a role is missing?

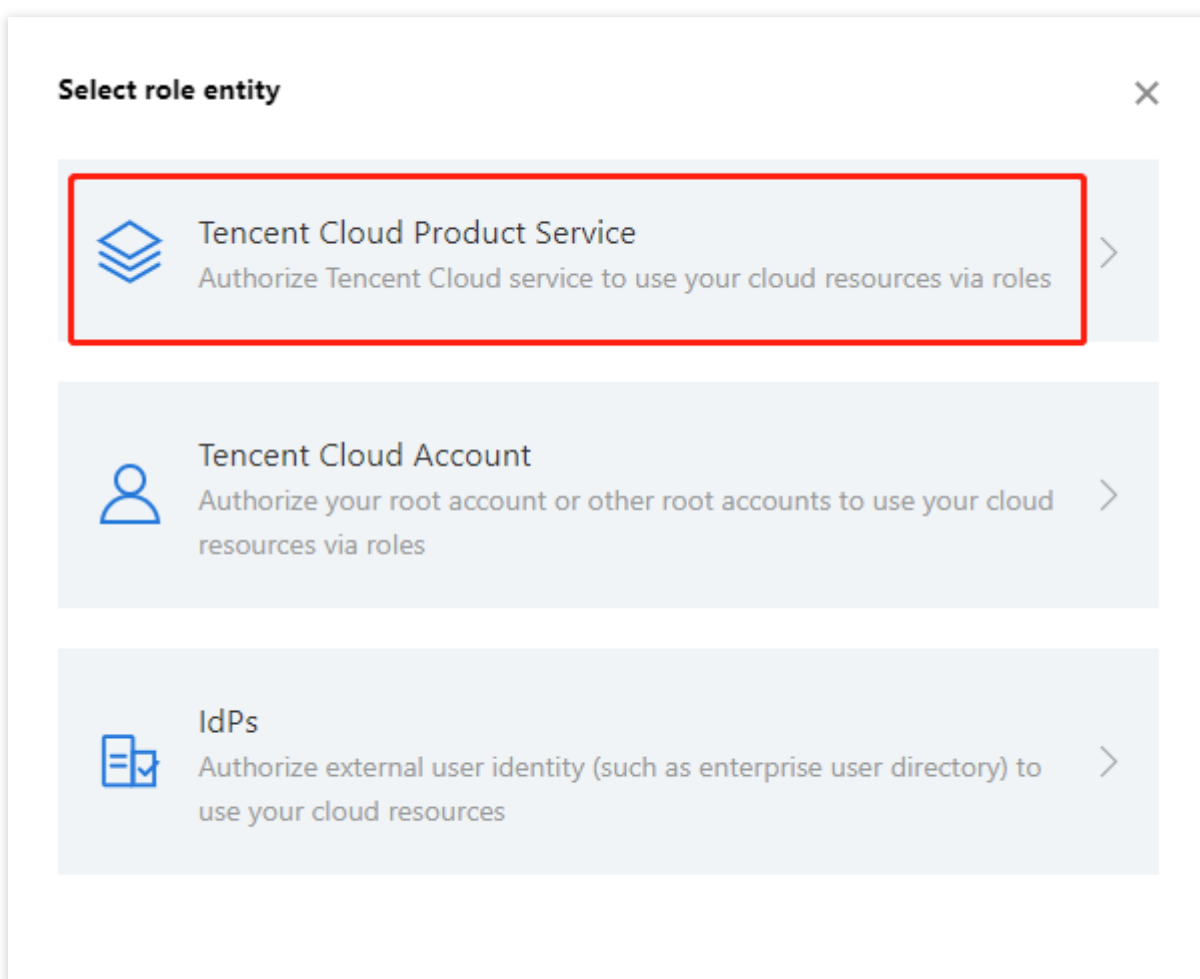
1. Find **Access Management** in the account list.



2. Select **Roles** on the left sidebar and click **Create Role**.



3. Select **Tencent Cloud Product Service** as the **Role Entity**.



4. Find **Message Service (CKafka)** in **Enter Role Entity Info**.



1 Enter Role Entity Info > 2 Configure Role Policy > 3 Set Role Tag > 4 Review

Product Service

<input type="checkbox"/> Tencent Cloud Advisor (advisor)	<input type="checkbox"/> Aegis (aegis)	<input type="checkbox"/> Shenbi Low-Code Application Platform as a Service (shenbi)	<input type="checkbox"/> API Gateway (apigw)	<input type="checkbox"/> Auto Scaling (as)
<input type="checkbox"/> Application Service Workflow (asw)	<input type="checkbox"/> TBaaS (tbaas)	<input type="checkbox"/> Billing (billing)	<input type="checkbox"/> BlueKing (blueking)	<input type="checkbox"/> Cloud Physical Machine (bm)
<input type="checkbox"/> BPaaS (bpaas)	<input type="checkbox"/> Cloud Access Security Broker (casb)	<input type="checkbox"/> Tencent Kubernetes Engine (tke)	<input type="checkbox"/> Cloud Database (cdb)	<input type="checkbox"/> Cloud Data Coffer Service (cdcs)
<input type="checkbox"/> TencentCloud Component Development Tools (cdevops)	<input type="checkbox"/> CDN (cdn)	<input type="checkbox"/> Cloud File Storage (cfs)	<input type="checkbox"/> Cloud Firewall (cfw)	<input type="checkbox"/> Customer Growth Expert (cge)
<input type="checkbox"/> Cloud Infinite (ci)	<input checked="" type="checkbox"/> CKafka (ckafka)	<input type="checkbox"/> Cloud Loader Balance (clb)	<input type="checkbox"/> Cloud Audit (cloudaudit)	<input type="checkbox"/> Cloud Studio (cloudstudio)
<input type="checkbox"/> Cloud Log Service (cls)	<input type="checkbox"/> cloudWaf (waf)	<input type="checkbox"/> Cloud Monitor (monitor)	<input type="checkbox"/> Creativity Platform (cme)	<input type="checkbox"/> CODING DevOps (coding)
<input type="checkbox"/> COS (cos)	<input type="checkbox"/> Cloud Storage Gateway (csg)	<input type="checkbox"/> Cloud Training Platform (ctp)	<input type="checkbox"/> TencentDB for CTSDB (ctsdb)	<input type="checkbox"/> Cloud Virtual Machine (cvm)
<input type="checkbox"/> Cloud Workload Protection (cwp)	<input type="checkbox"/> Tencent Cloud Developer-TDP (devops)	<input type="checkbox"/> DI (di)	<input type="checkbox"/> Data Lake Compute (dlc)	<input type="checkbox"/> Data Security Center (dsgc)
<input type="checkbox"/> Data Transmission Service (dts)	<input type="checkbox"/> EventBridge (eb)	<input type="checkbox"/> App Flow (eis)	<input type="checkbox"/> Elasticsearch MapReduce (emr)	<input type="checkbox"/> faceid (faceid)
<input type="checkbox"/> Game Sever Elastic-scaling (qse)	<input type="checkbox"/> Image AI (facerecognition)	<input type="checkbox"/> IDaaS (idaas)	<input type="checkbox"/> IoT Hub (iothub)	<input type="checkbox"/> IoT Suite (iotsuite)

5. In **Configure Role Policy**, select policies such as `QcloudCLSFullAccess` and `QcloudCOSFullAccess` based on the dump target service and click **Next**.

← Create Custom Role

1 Enter Role Entity Info > 2 Configure Role Policy > 3 Set Role Tag > 4 Review

Select Policies (1 Total)

Policy Name	Policy type
<input checked="" type="checkbox"/> QcloudCOSFullAccess Full read-write access to Cloud Object Storage (COS)	Preset Policy

2 selected

Policy Name	Policy type
QcloudCLSFullAccess Full read-write access to Cloud Log Service (CLS)	Preset Policy
QcloudCOSFullAccess Full read-write access to Cloud Object Storage (COS)	Preset Policy

Support for holding shift key down for multiple selection

Back Next

6. Enter the role name `DIP_QcsRole` on the **Review** page and set the service role **CKafka Connector** as the description. This role will access your Tencent Cloud service resources within the permission scope of the associated policy.

7. Then, you can dump data through CKafka.

# Data Reporting Issues

Last updated : 2024-09-09 21:31:56

## How to Confirm Successful Data Reporting?

Data reporting is considered successful when the TencentCloud API returns the MessageId field. See TencentCloud API call response in [TencentCloud API Return Results](#).

## How to Handle Errors During Data Reporting?

During the data reporting process, common TencentCloud API error responses include:

```
{
  "Response": {
    "Error": {
      "Message": "TopicAuthorizationException: Not authorized to access topic",
      "Code": "FailedOperation"
    },
    "RequestId": "xxxx9bef-52fd-4242-bdc0-8cf5e55xxxxxx"
  }
}
```

1. Based on the error information, check if ACL permissions are set for the Topic (Currently, HTTP data reporting does not support ACL settings.). You need to disable the ACL settings of the previous topic.

```
{
  "Response": {
    "Error": {
      "Message": "task status suspended [datahub-lz7xxxxv]",
      "Code": "FailedOperation.ResourceTaskPaused"
    },
    "RequestId": "xxxxifef-942fd-4242-bdc0-z3f5e55xxxxxx"
  }
}
```

2. According to the error message, the current task status is paused. You just need to resume the task to running status.

```
{
  "Response": {
    "Error": {
      "Message": "limit exceeded dataHubId [datahub-nyxxxxiy]",
      "Code": "RequestLimitExceeded"
    },
    "RequestId": "xxxxqsnrf-942fd-4242-nsdc0-z3f5e55xxxxxx"
  }
}
```

```
}  
}
```

3. The number of requests per second to the interface has reached the limit. You can submit a ticket to adjust the current QPS limit based on business needs.

4. You can also report other error messages with complete error information. You can [Contact Us](#) for assistance.

### What is the Data Reporting Quota, and How Can It Be Increased?

Data reporting is accessed via HTTP Protocol. The default QPS limit for HTTP interfaces is 2,000. If it exceeds 2,000, a request limit error will be triggered.

If a higher QPS is required, you can [Contact Us](#) to increase the QPS limit for a specific task ID.

# Data Processing Issues

Last updated : 2024-09-09 21:33:52

## What Happens to Data that Fails to Be Processed due to Format Issues During Data Processing?

During data processing, data with processing failure due to format issues or does not meet filter rules will be directed to the Failure Handling policy.

Failure Handling policies include:

1. The Discard policy means discarding raw data that failed to process.
2. The Retain policy means retaining the raw data that failed to process and encapsulating the raw data and failure information into a new message body, and delivering it to the target topic.
3. The Dead Letter Queue policy means retaining the raw data that failed to process and encapsulating the raw data and failure information into a new message body, and delivering it to the Dead Letter Queue.

## How to Identify Performance Bottlenecks in Data Processing?

The process of data processing consists of three phases:

1. Consuming data from the source Topic;
2. Processing the raw data;
3. Delivering the processed data to the data target Topic.

We can monitor the performance of phase 1 by tracking the message backlog and consumption speed of the consumer, and for phase 3, by monitoring the production speed of the target Topic.

Generally, we can increase the number of partitions in the source Topic to improve the maximum concurrency of task scalability, thereby enhancing the overall task performance.

# Data Dump

Last updated : 2024-01-09 15:02:48

## How do I check whether data heaps up during dump?

CKafka data dump refers to the process where CKafka data is distributed to other sources such as ES and ClickHouse.

The sync service consumes messages in the CKafka instance, so the corresponding consumer group will be generated, which can be viewed on the consumer group management page in the console. Generally, this consumer group is named `datahub-task-xxx`. After the sync service consumes messages, it will write them to the service of the dump target and then submit the offset position corresponding to the number of written messages. Therefore, to determine whether the dumped data heaps up, you can simply check whether the number of unconsumed messages in the consumer group keeps increasing.

## What should I do if data heaps up?

There are two types of data heap:

If the consumption capacity of the sync service is limited, you can increase task concurrency so that the sync service on the backend can add more consumers. You can also increase the number of topic partitions as needed to improve consumer throughput. If the consumption traffic of the instance reaches the limit and gets throttled, you also need to upgrade the bandwidth specification of the instance.

If the heap problem persists after you increase the CKafka consumption capacity, the problem may be that the rate of writing data to the target is limited, preventing the sync service from quickly completing the process of writing data and submitting the offset. For example, when a large number of writes hit the bottleneck in ES, a lock may be generated to protect the service, which may reject external writes or even cause sync task exceptions; or, if the number of writes per second reaches the upper limit in TDW, writes will be blocked. In this case, you should determine the write bottleneck of the target and increase its write rate.

# Data Subscription Issues

Last updated : 2024-09-09 21:34:48

## MySQL Exception Handling

### Configuration and Startup Errors

The following situations may cause the connector to fail to start, report errors or exceptions in the log, and then stop running:

The configuration of the connector is invalid.

The connector cannot connect to the MySQL server based on the provided configuration parameters.

The connector attempts to recover from the failure point after a restart, but MySQL's binlog has been cleaned and the corresponding historical records are unavailable.

When these situations occur, the error message will provide details of the error and may offer some troubleshooting methods. After troubleshooting, you can try restarting the connector service.

### MySQL Becomes Unavailable

If the MySQL service becomes unavailable, the connector will stop working. You need to restart the connector service when MySQL becomes available. If your MySQL cluster uses the GTIDs protocol, you can immediately restart the connector service; it will connect to another server in the cluster and continue reading the binlog from the last committed transaction.

### Kafka Connect Exceptional Termination

If Kafka Connect is running in distributed mode and the Kafka Connect process stops normally, Kafka Connect will migrate the connector tasks from that process to another Kafka Connect process in the group before shutting down the process. The new connector tasks will accurately continue processing from where the previous tasks stopped. There will be a brief delay in processing when the connector tasks stop normally and restart on the new process.

### Kafka Connect Crash

When Kafka Connect crashes, the process stops immediately and does not have time to commit the most recent offset. In a distributed deployment environment, Kafka Connect will restart a new process, but the new process cannot obtain the latest offset from the crashed process, which may result in duplicate submission of the same event.

However, each change event message includes the connector's metadata information, which you can use to identify duplicate event submissions.

### Kafka Service Becomes Unavailable

When the Kafka service becomes unavailable, the Debezium MySQL connector will pause until it re-establishes connection with the Kafka service.

## MySQL Binlog Cleanup

If the Debezium MySQL connector stops for too long, the MySQL server may clean up the binlog file, causing the latest position read by the connector to be cleaned up. When the connector restarts and finds that the original read position has been cleaned, it will attempt to reinitialize the snapshot. If the snapshot is disabled, the connector will terminate exceptionally.

# PostgreSQL Exception Handling

## Configuration and Startup Errors

The following situations may cause the connector to fail to start, report errors or exceptions in the log, and then stop running:

The configuration of the connector is invalid.

The connector cannot connect to the PostgreSQL server according to the provided configuration parameters.

The connector attempts to resume reading from the record offset at the time of the failure when it restarts, but PostgreSQL has already cleaned up the related records.

When these situations occur, the error message will provide details of the error and may offer some troubleshooting methods. After troubleshooting, you can try restarting the connector service.

## PostgreSQL Becomes Unavailable

If the PostgreSQL service becomes unavailable, the connector will stop working and will need to be restarted once PostgreSQL is available.

## Cluster Failures

Since version 12 is released, PostgreSQL only allows setting up logical replication slots on the primary server. This means you can only point the Debezium PostgreSQL connector to the database cluster's primary server. Moreover, replication slots themselves are not propagated to replicas. If the primary server fails, a new primary server should be elected.

The new primary server should have the [logical decoding plug-in](#) installed, configured for the replication slots used by the plugin, and the database to capture change events should be running. Only then can you reconfigure the connector to connect to the new server and restart the connector.

There are some important warnings during failover: You should pause the Debezium service, and restart the service after verifying that there is a complete replication slot and no data loss. After failover:

Before the application to write to the new primary node is allowed, there should be a process to recreate the Debezium replication slot; otherwise, the application might miss change events.

You may need to verify whether Debezium can read all changes from the slot before the old primary node is terminated.

A reliable method to restore and verify if any change events were lost is to restore the backup of the failed primary node to the position before the failure. Although this might be difficult to execute, it allows you to check if any unconsumed changes remain in the replication slot.

## Kafka Connect Exceptional Termination

If Kafka Connect is running in distributed mode and the Kafka Connect process stops normally, Kafka Connect will migrate the connector tasks from that process to another Kafka Connect process in the group before shutting down. The new connector tasks will accurately continue processing from where the previous tasks stopped. There will be a brief delay in processing when the connector tasks stop normally and restart on the new process.

## Kafka Connect Crash

If the Kafka connector process stops unexpectedly, all the tasks running on the connector will be terminated, and their recently processed offsets will not be logged. When Kafka Connect is running in distributed mode, Kafka Connect will restart these connector tasks on other processes. However, the PostgreSQL connector recovers from the last offset recorded by the earlier process. This means the new replacement tasks may generate some of the same change events processed before the crash. The number of duplicate events depends on the offset refresh cycle and data changes before the crash. During each change event recording, the Debezium connector logs metadata information related to event origin, including PostgreSQL server time and the ID of the server transaction when the event occurred. Consumers can track this information, especially the LSN, to determine if an event is duplicated.

## Kafka Service Becomes Unavailable

When the Kafka service becomes unavailable, the Debezium PostgreSQL connector will pause and retry the connection until it re-establishes with the Kafka service.

## Connector Stopped Service for a Period

If the connector stops normally, the database can continue to be used. Any changes will be recorded in the PostgreSQL WAL. When the connector restarts, it will resume committing changes from where it left off. This means it will generate change event records for all database changes made while the connector was stopped.

A properly configured Kafka cluster can handle data with very high throughput. Kafka Connect is written according to Kafka best practices and, with sufficient resources, the Kafka Connect connector can also handle a large number of database change events. Therefore, after being stopped for a period of time, when the Debezium connector restarts, it is likely to catch up on the changes that occurred in the database during the downtime.