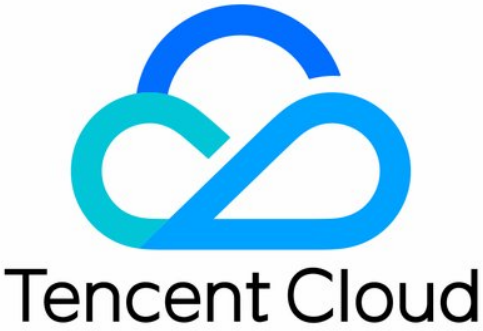
TDMQ for CKafka

FAQs

Product Documentation



©2013-2022 Tencent Cloud. All rights reserved.

Copyright Notice

©2013-2024 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

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

FAQs
- Instance
- Topic
- Consumer Group
- Client
- Network
- Traffic Throttling
- Monitoring
- Message
- General
- Connector
  - Task
  - Data Dump

©2013-2022 Tencent Cloud. All rights reserved.

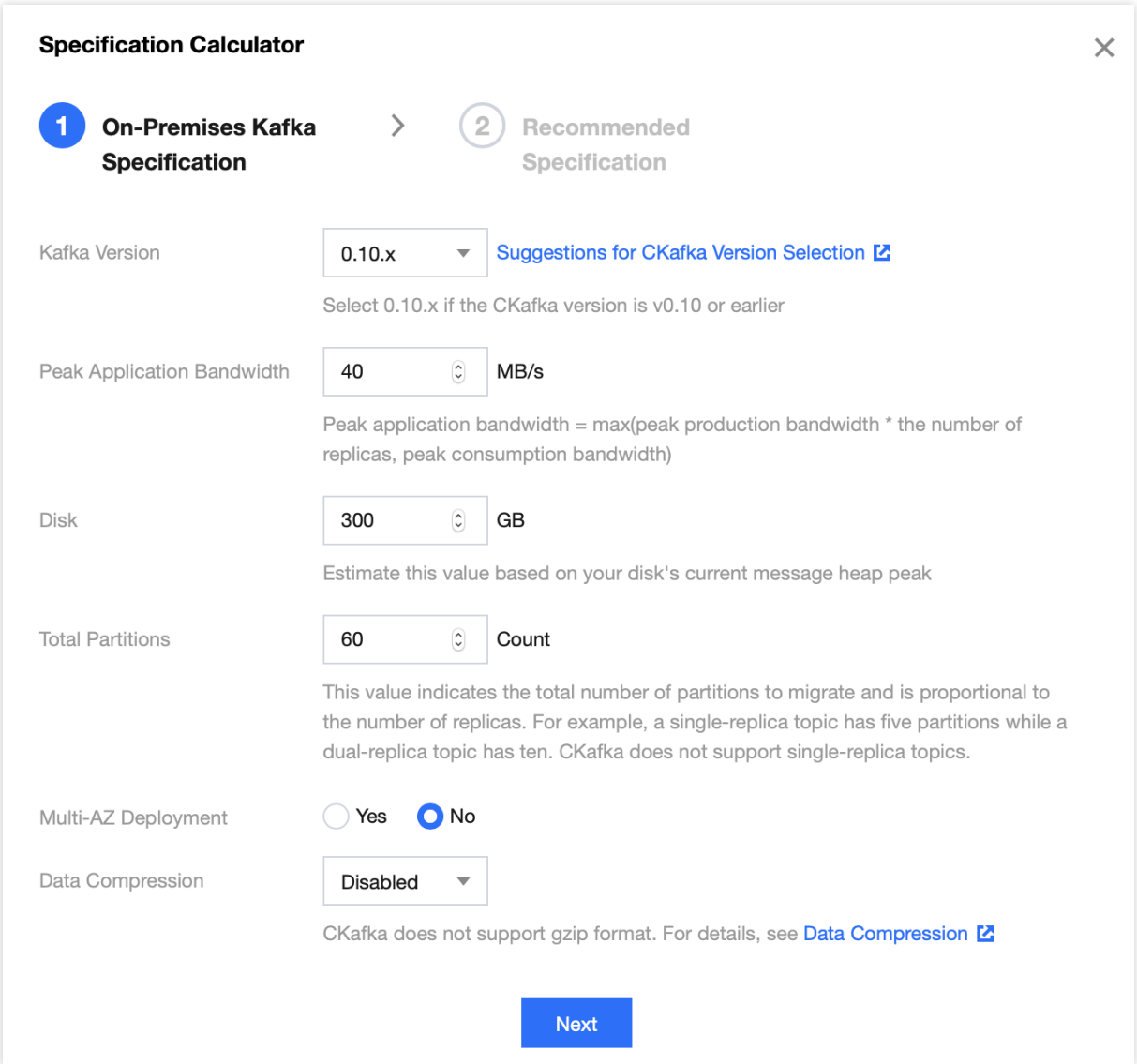
FAQs

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.



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

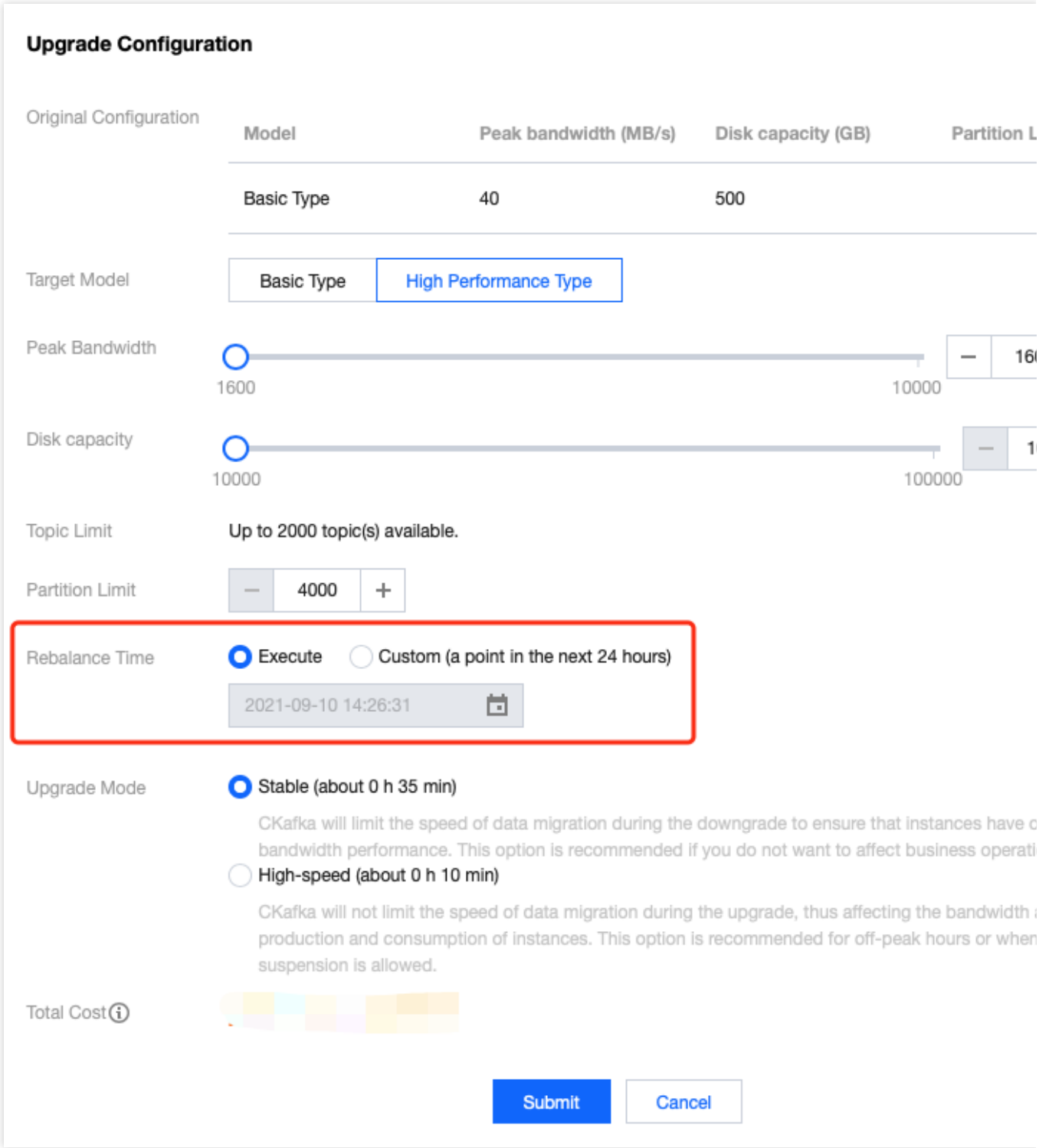
You can select an appropriate product specification as needed. For the billing rules of different specifications, see Billing Overview.

©2013-2022 Tencent Cloud. All rights reserved.

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:

©2013-2022 Tencent Cloud. All rights reserved.



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.

©2013-2022 Tencent Cloud. All rights reserved.

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.

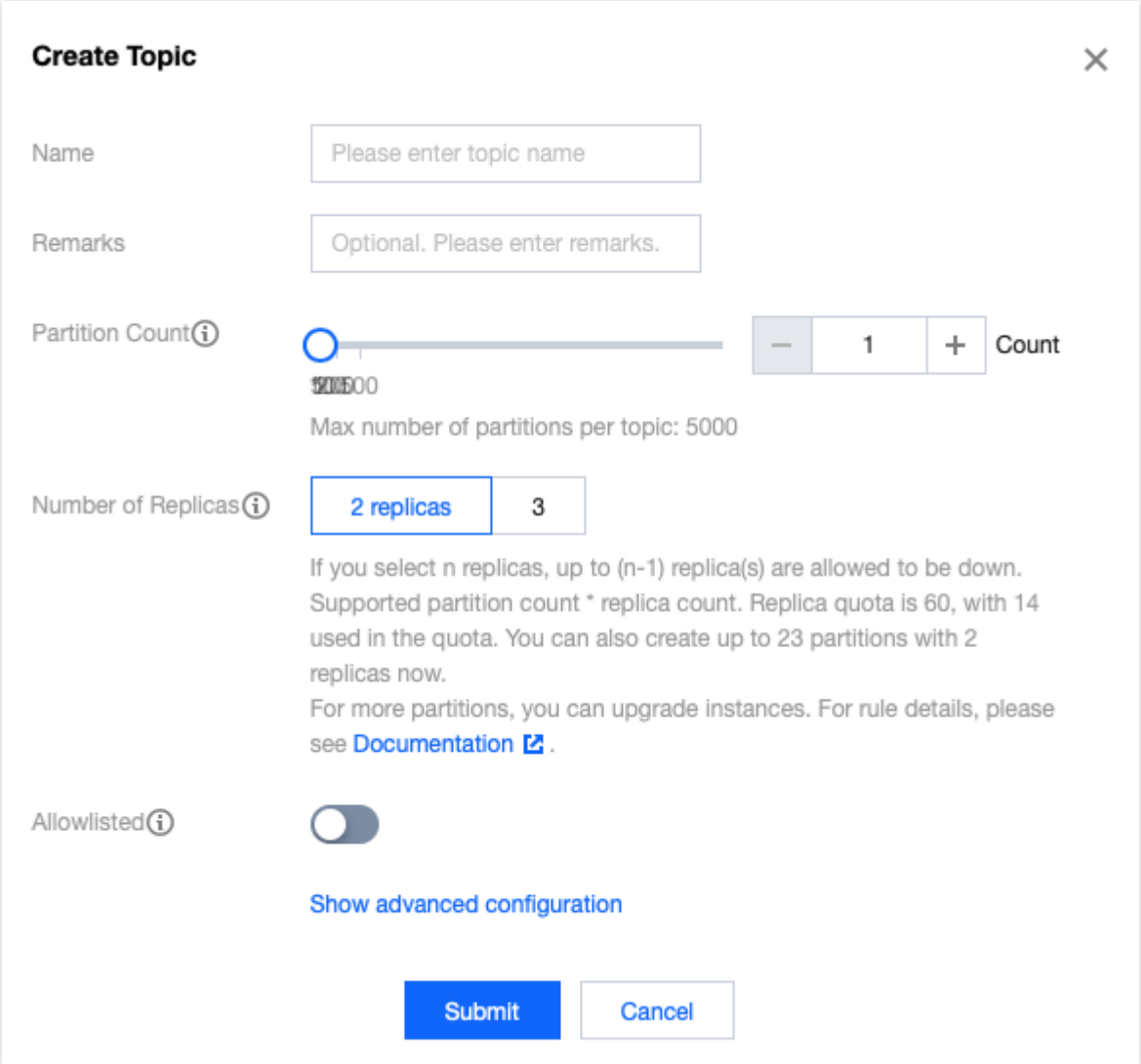
©2013-2022 Tencent Cloud. All rights reserved.

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:



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.

©2013-2022 Tencent Cloud. All rights reserved.

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.

©2013-2022 Tencent Cloud. All rights reserved.

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.

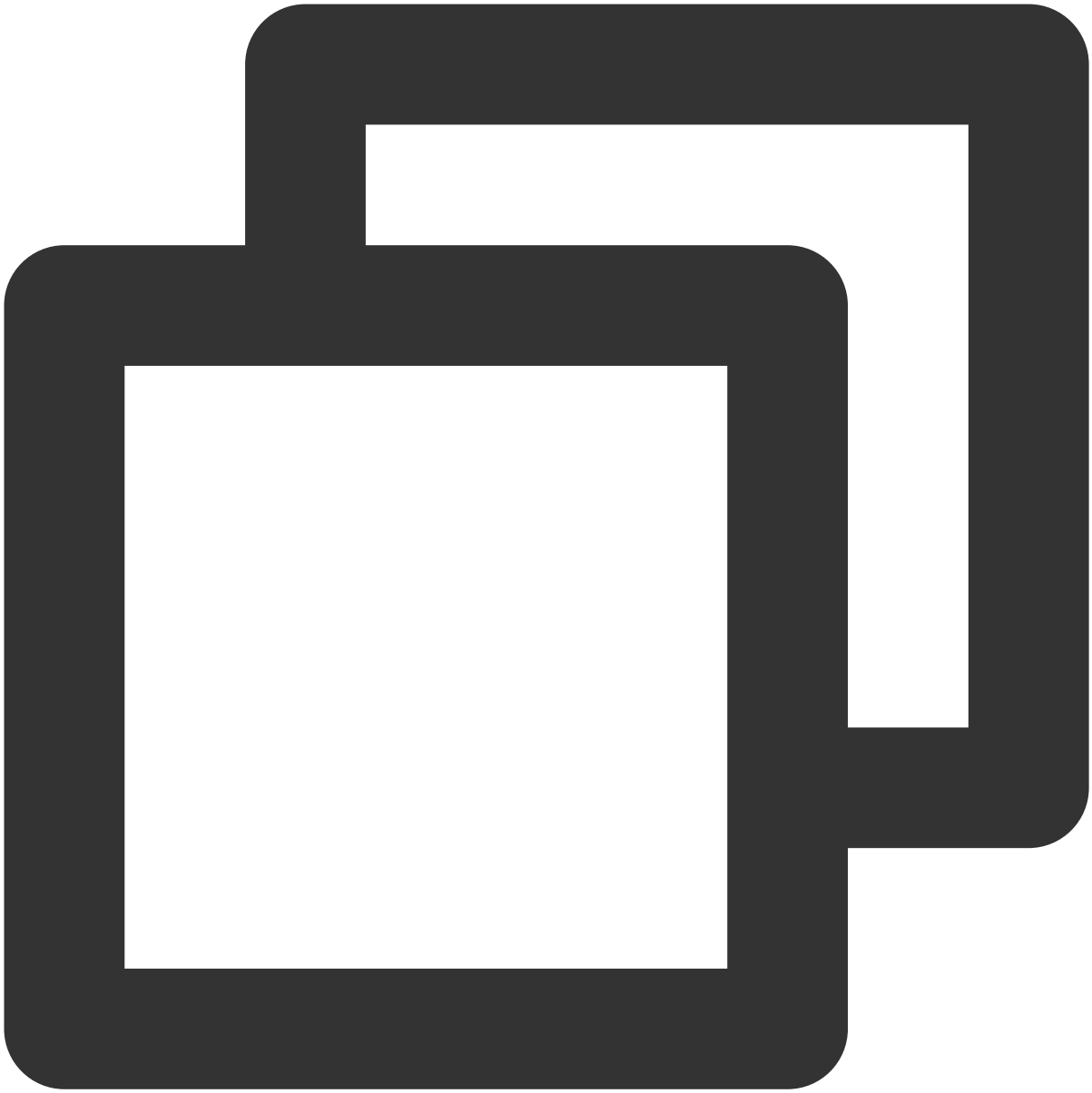
A topic can contain one or multiple partitions.

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:

©2013-2022 Tencent Cloud. All rights reserved.

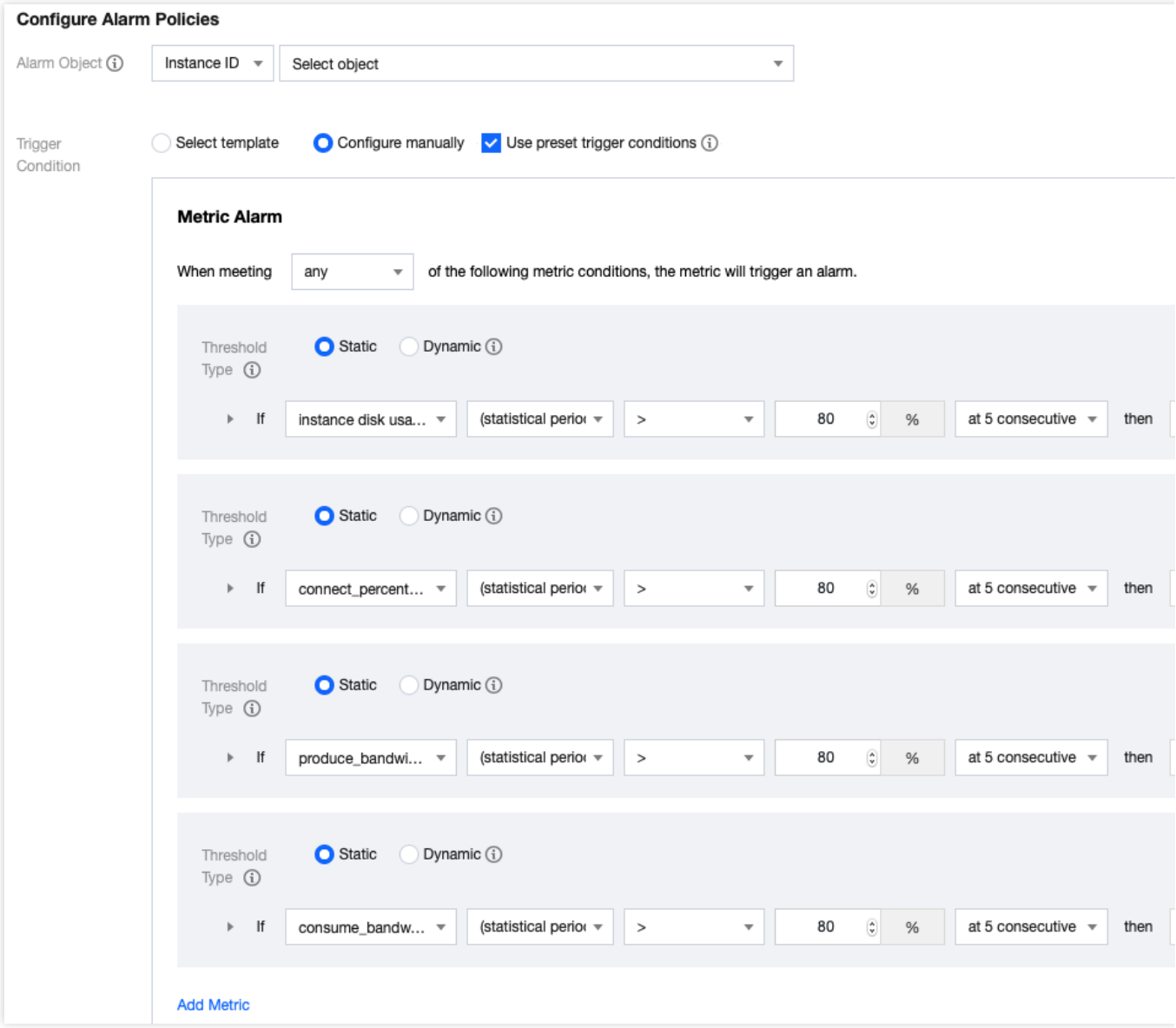


```
Properties props = new Properties();
props.put(ConsumerConfig.BOOTSTRAP_SERVERS_CONFIG, bootstrap);
KafkaConsumer<String, String> consumer = new KafkaConsumer<>(props);
consumer.subscribe(Arrays.asList(topic));
while (true) {
            ConsumerRecords<String, String> records = consumer.poll(100);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

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

©2013-2022 Tencent Cloud. All rights reserved.

an alarm, see Configuring Alarm. The configuration page is as shown below:



©2013-2022 Tencent Cloud. All rights reserved.

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

©2013-2022 Tencent Cloud. All rights reserved.

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.

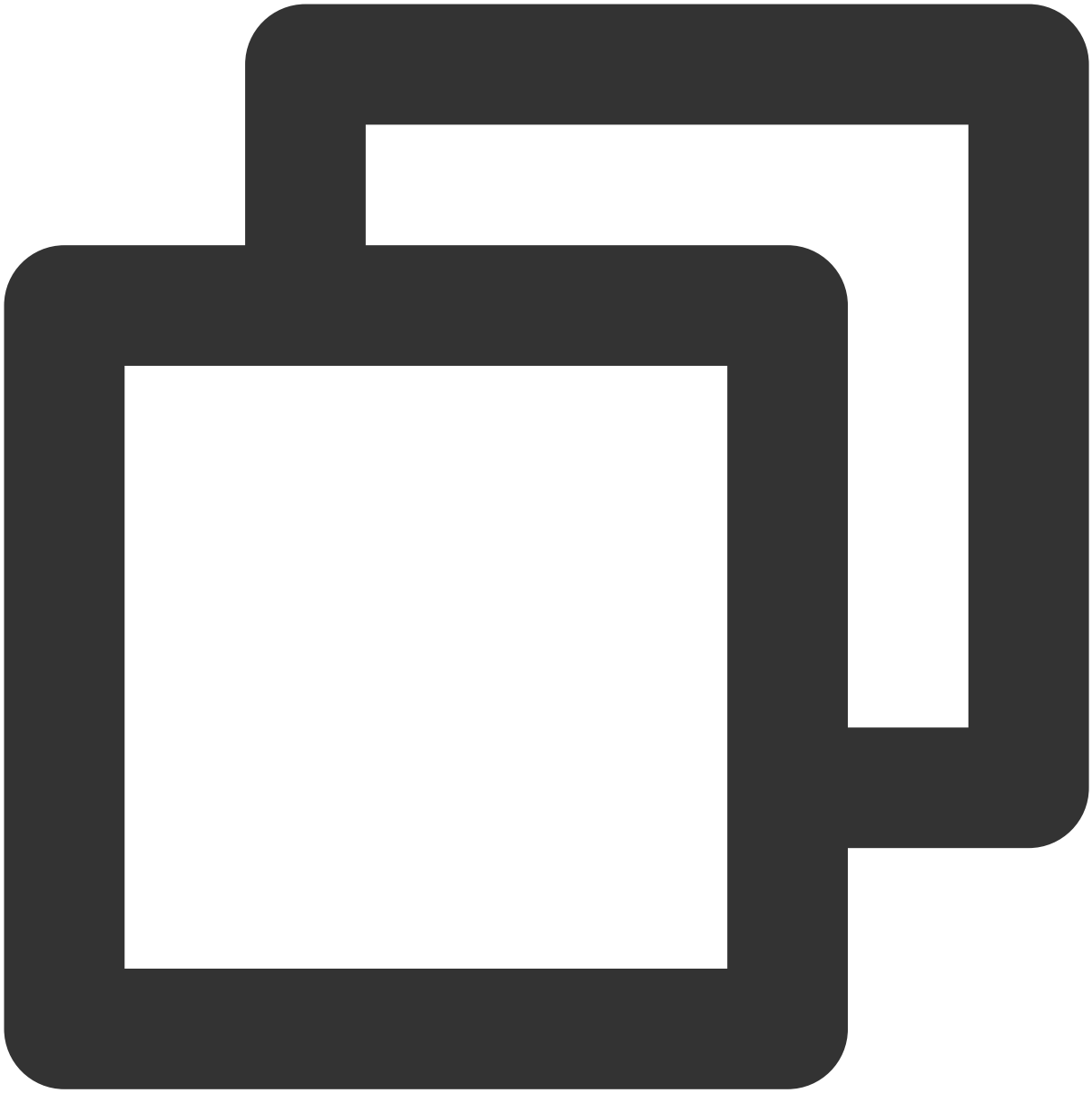
You can also check whether a message is successfully sent in the console by the following methods:

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:

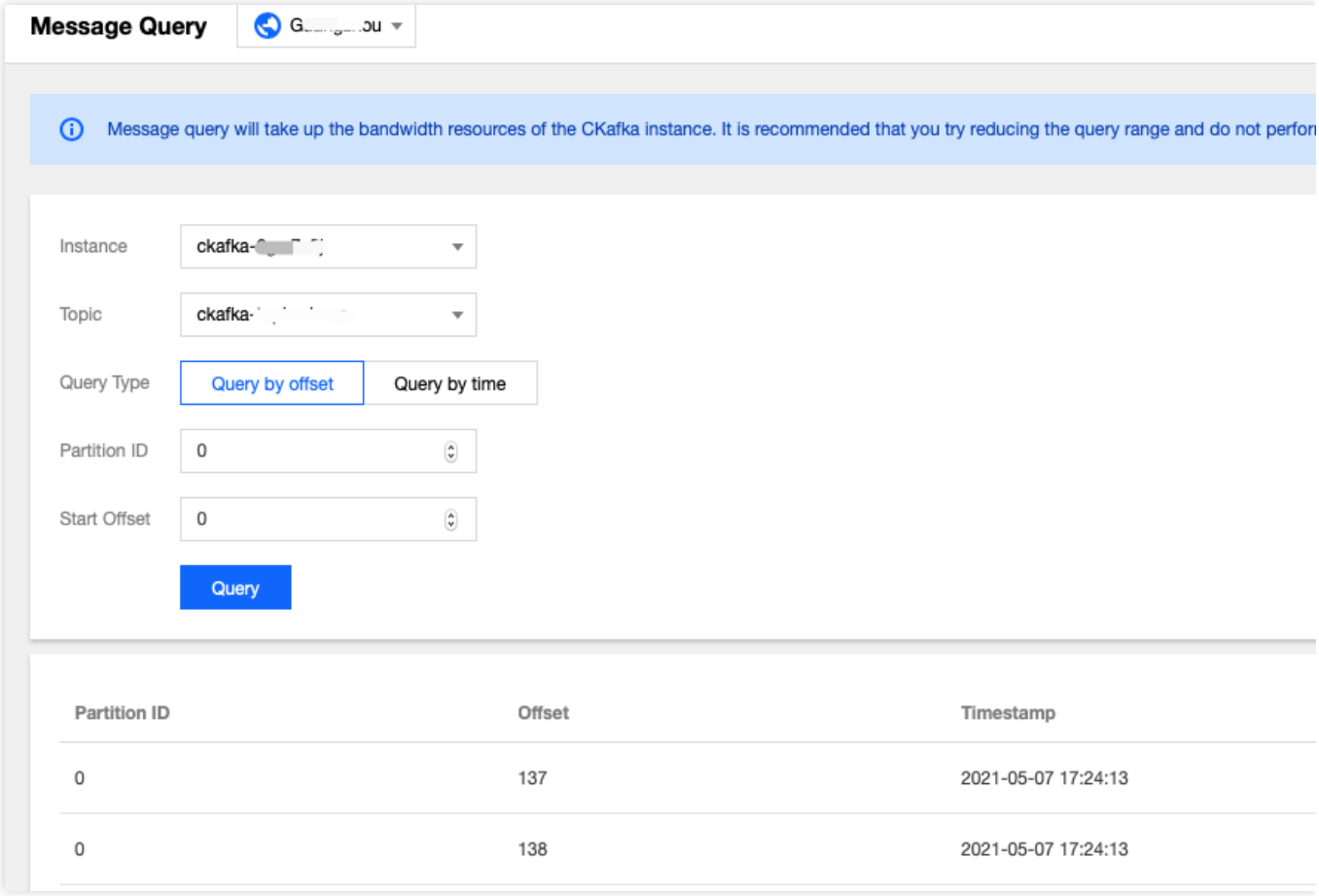
©2013-2022 Tencent Cloud. All rights reserved.



```
Future<RecordMetadata> future = producer.send(new ProducerRecord<>(topic, messageKe
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.

©2013-2022 Tencent Cloud. All rights reserved.



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.

©2013-2022 Tencent Cloud. All rights reserved.

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.

©2013-2022 Tencent Cloud. All rights reserved.

Network

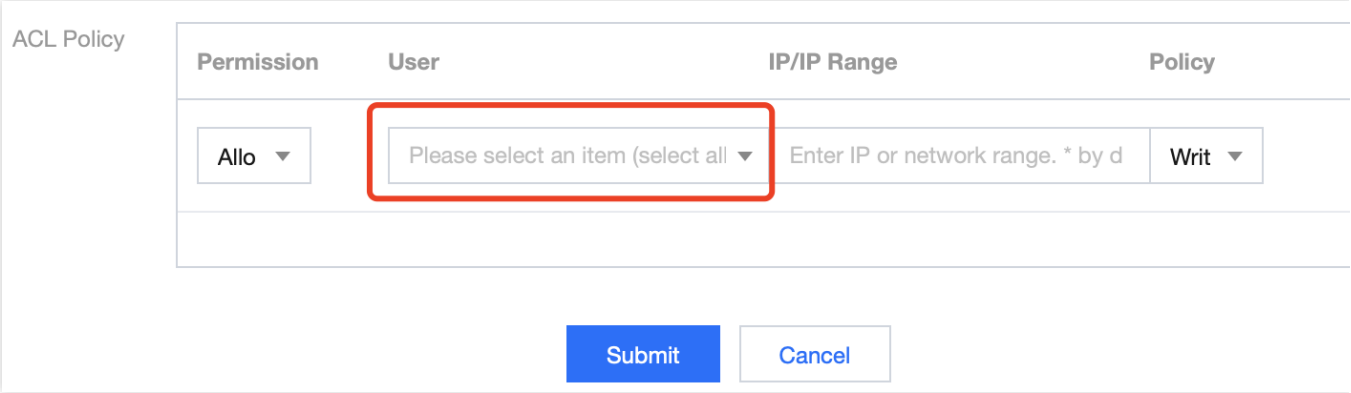
Last updated：2024-01-09 15:02:48

Does CKafka support public network access?

CKafka transfers data over the private network by default. To access over the public network, you need to activate a public network route separately. For detailed directions, please see Adding Routing Policy. Currently, a broker provides 3 Mbps public network bandwidth free of charge by default.
CKafka Pro Edition supports increasing the public network bandwidth to up to 198 Mbps for a fee. For the specific prices, please see Billing Overview.
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.

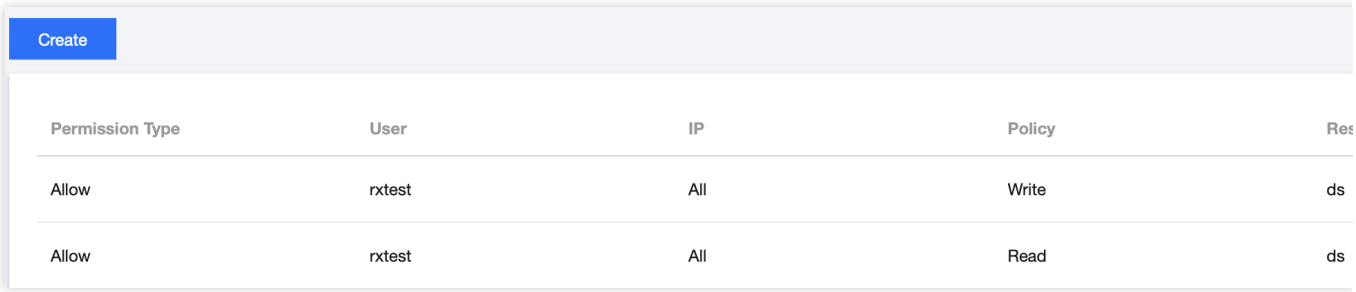
How do I continue accessing CKafka over VPC after SASL is enabled for the public network?

If you use the PLAINTEXT method to access CKafka while enabling public network 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:

©2013-2022 Tencent Cloud. All rights reserved.



©2013-2022 Tencent Cloud. All rights reserved.

Traffic Throttling

Last updated：2024-01-09 15:02:47

Traffic Throttling Mechanism Description

Kafka's traffic throttling mechanism is soft throttling; that is, when the user's traffic exceeds the quota, Kafka uses the delayed response method for processing instead of returning an error to the client.

Take API traffic throttling as an example:

Hard traffic throttling: If the call rate is 100 times/sec, when the client calls more than 100 times per second, the server will return an error, and the client needs to process it according to the business logic.

Soft traffic throttling: If the call rate is 100 times/sec, and the normal duration is 10 ms, when the client calls more than 100 times per second:

If it is 110 times, this request will take 20 ms.

If it is 200 times, this request will take 50 ms. In this case, throttling is friendly to the client, no error alarms will be triggered due to traffic surges or fluctuations, and the business can continue normally.

Therefore, in high-traffic scenarios such as Kafka, soft traffic throttling is better for a smooth user experience.

Relationship between purchased bandwidth and production/consumption bandwidth:

Maximum production bandwidth (per second) = purchased bandwidth/number of replicas

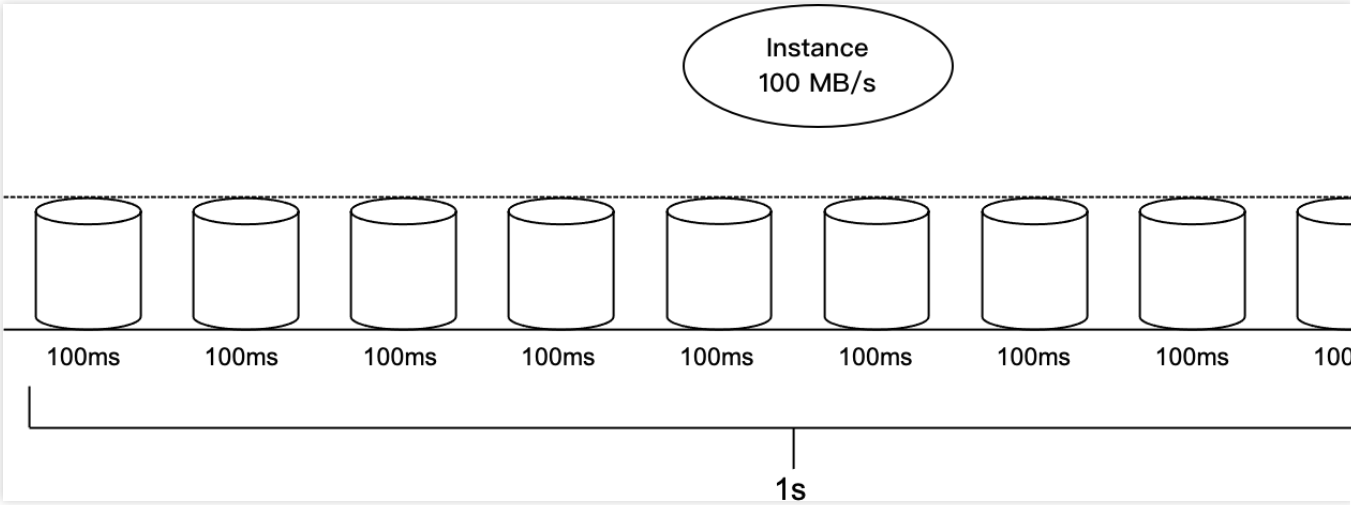
Maximum consumption bandwidth (per second) = purchased bandwidth

How Delayed Response Works

The underlying traffic throttling mechanism of a CKafka instance is implemented based on token bucket. Each second is divided into multiple time buckets measured in milliseconds.

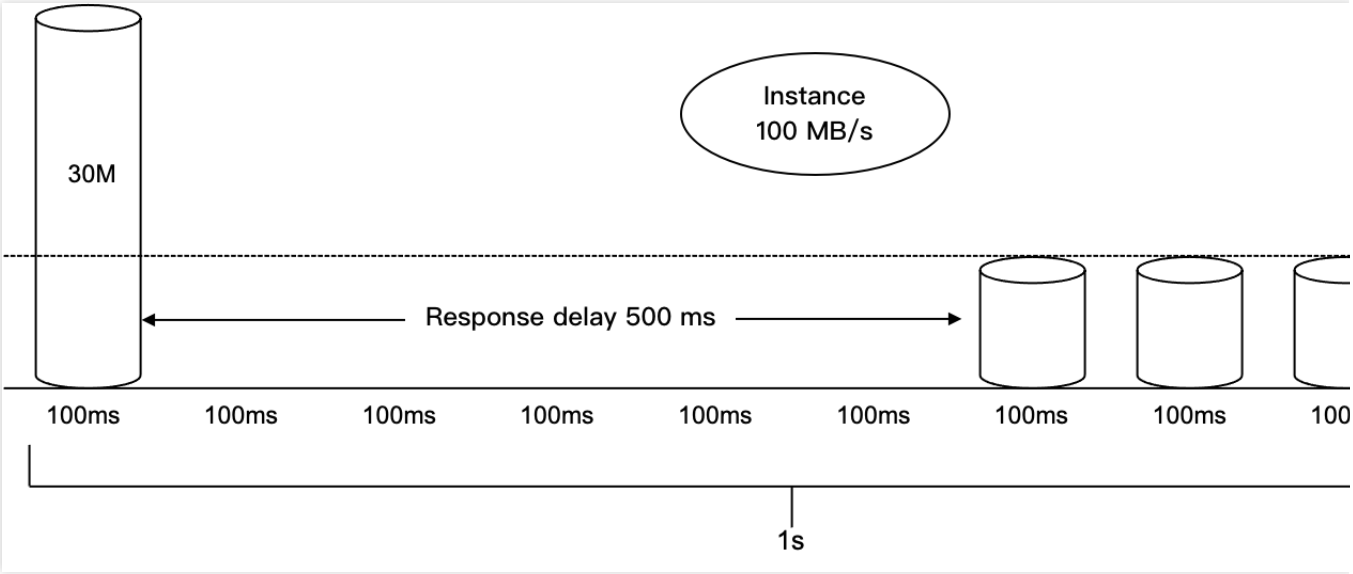
The traffic throttling policy divides every second (1,000 ms) into several time buckets. For example, if one second is divided into 10 time buckets, then the time of each bucket is 100 ms, and the traffic throttling threshold of each time bucket is 1/10 of the total instance specification speed. If the TCP request traffic in a certain time bucket exceeds such threshold, the delayed response time of the request will be increased according to the internal traffic throttling algorithm, so that the client cannot quickly receive the TCP response, thus achieving the traffic throttling effect in a period of time.

©2013-2022 Tencent Cloud. All rights reserved.



Why is traffic throttling triggered when the monitored production/consumption traffic is lower than the instance specification?

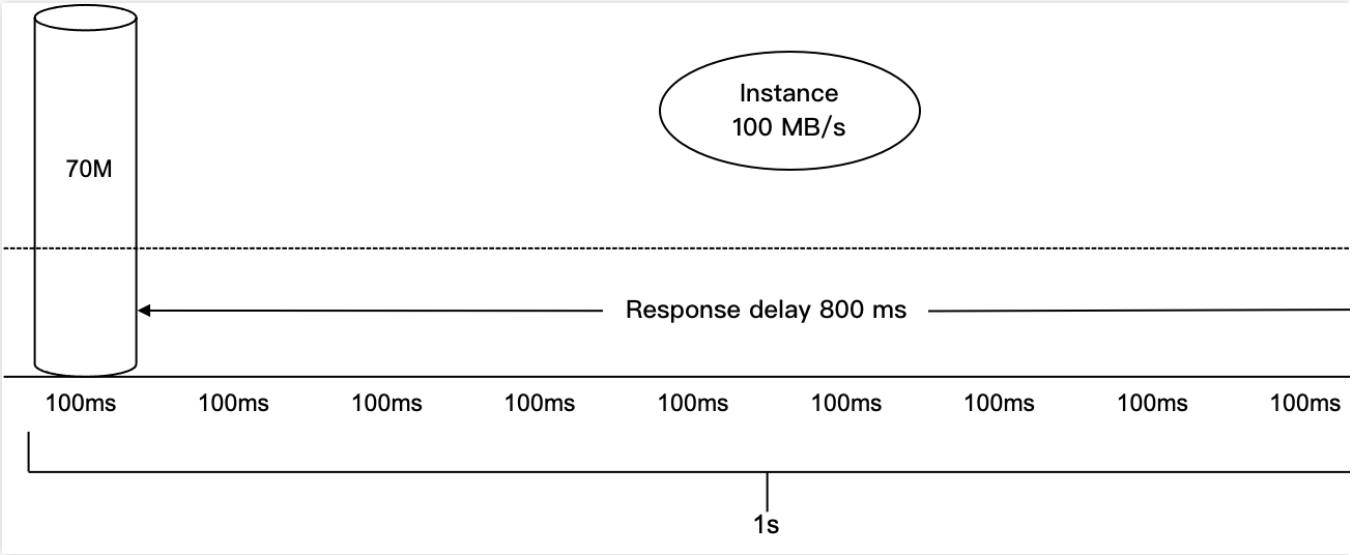
As mentioned above, traffic throttling is measured in milliseconds, but the monitoring platform in the console collects data at the second level and aggregates the data at the minute level (maximum or average value).
According to the principle of token bucket, a single bucket does not forcibly throttle the traffic. Suppose the bandwidth specification of instance A is 100 MB/sec, then the traffic throttling threshold of each 100-ms time bucket is 100 MB/10 = 10 MB/bucket. If the production traffic of instance A reaches 30 MB in the first 100-ms time bucket of a certain second (3 times the threshold), then the broker's traffic throttling policy will be triggered to increase the delayed response time. Suppose the original normal TCP response time is 100 ms, then the delay may be increased by 500 ms before response after the threshold is exceeded. The final traffic in this second is 30 MB * 1 + 0 MB * 5 + 10 MB * 4 = 70 MB, so the traffic speed in this second (70 MB/sec) is lower than the instance specification (100 MB/sec).



©2013-2022 Tencent Cloud. All rights reserved.

Why is the peak production/consumption traffic higher than the instance specification?

Suppose again the bandwidth specification of instance A is 100 MB/sec, then the traffic throttling threshold of each 100-ms time bucket is 10 MB. If the production traffic of instance A reaches 70 MB in the first 100-ms time bucket of a certain second (7 times the threshold), then the broker's traffic throttling policy will be triggered to increase the delayed response time. Suppose the original normal TCP response time is 100 ms, then the delay may be increased by 800 ms before response after the threshold is exceeded. After the response is returned at the 900th ms, the client immediately injects 70 MB of traffic into the 10th time bucket. The final traffic in this second is (70 MB * 1 + 0 MB * 8 + 70 MB * 1) = 140 MB, so the traffic speed in this second (140 MB/sec) is higher than the instance specification (100 MB/sec).



Why does the number of traffic throttling events surge?

The number of traffic throttling events is counted based on TCP requests. If instance A exceeds the traffic threshold in the first time bucket in a certain second, all TCP requests in the remaining time of this time bucket after the threshold is exceeded will be throttled and counted as traffic throttling events.

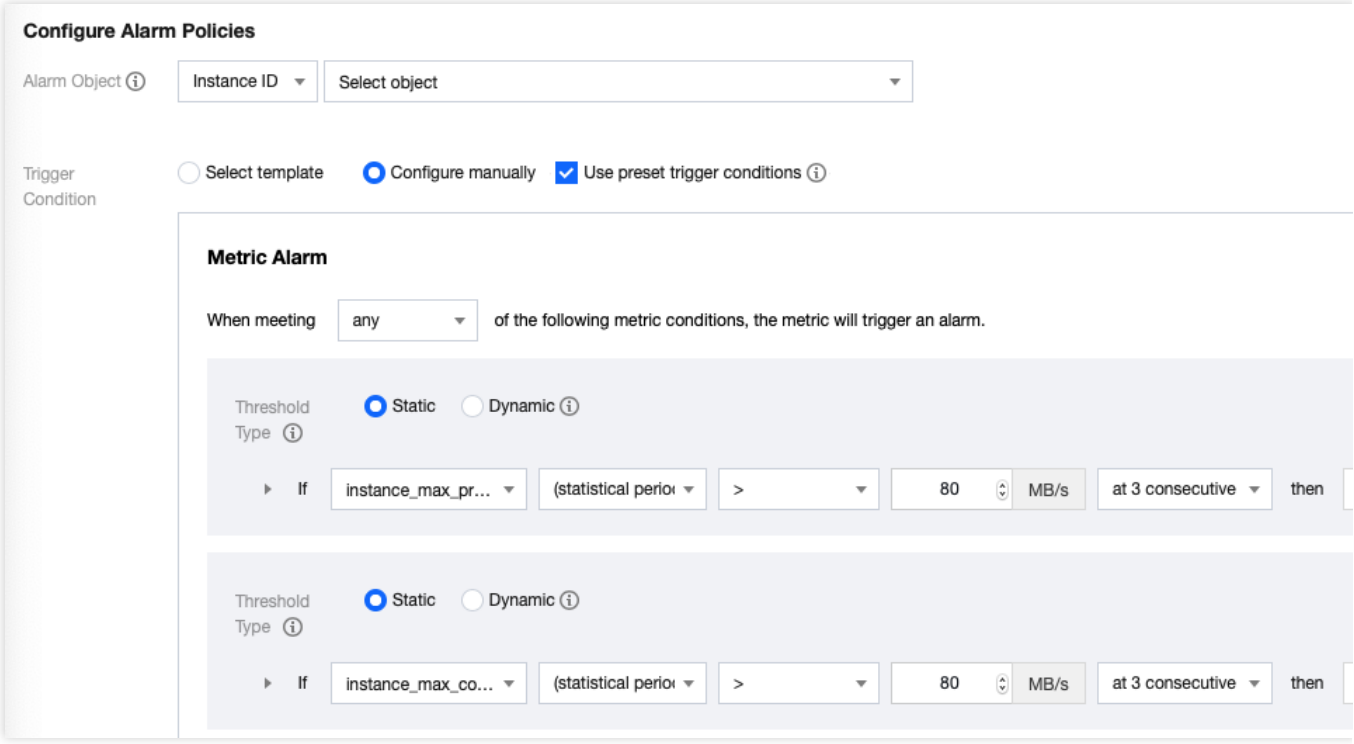
How does CKafka throttle traffic?

To ensure service stability, CKafka implement network traffic control strategies on both input and output messages. Throttling occurs when the total traffic of a user's replicas exceeds the purchased peak traffic.
When the producer traffic is throttled, CKafka will extend the response time of a TCP connection. The delay period depends on how much the instantaneous traffic exceeds the limit. It is similar to the principle of road traffic control. The more traffic flow, the higher the delay value from the delay algorithm, up to 5 minutes.
When the consumer traffic is throttled, CKafka will reduce the size of each `fetch.request.max.bytes` request to control the traffic.

©2013-2022 Tencent Cloud. All rights reserved.

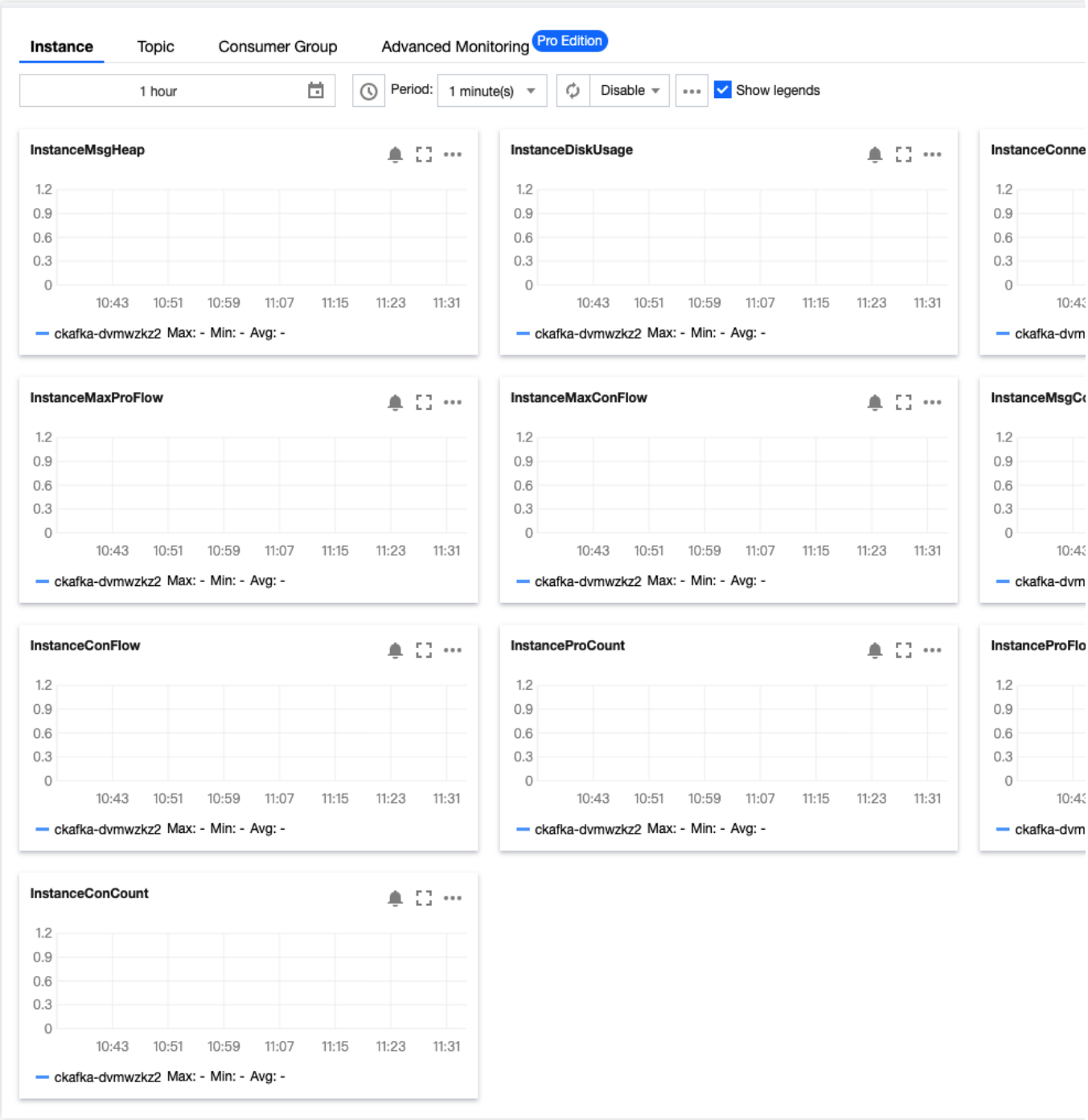
How do I determine whether CKafka has been throttled?

1. In the instance list, you can see the health status of each cluster. If it's "Warning", you can hover your mouse over it to view the detailed data. The data displays your peak traffic and the number of throttling occurrences, based on which you can determine whether this instance has been throttled.
2. You can click the **Monitoring** tab to view the max traffic value. If **the value of max traffic multiplied by replica count is greater than that of the purchased peak bandwidth**, you can determine that throttling has occurred at least once.



3. View the instance monitoring data on the monitoring page in the CKafka console. If the number of throttling occurrences is greater than 0, throttling has occurred.

©2013-2022 Tencent Cloud. All rights reserved.



©2013-2022 Tencent Cloud. All rights reserved.

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?

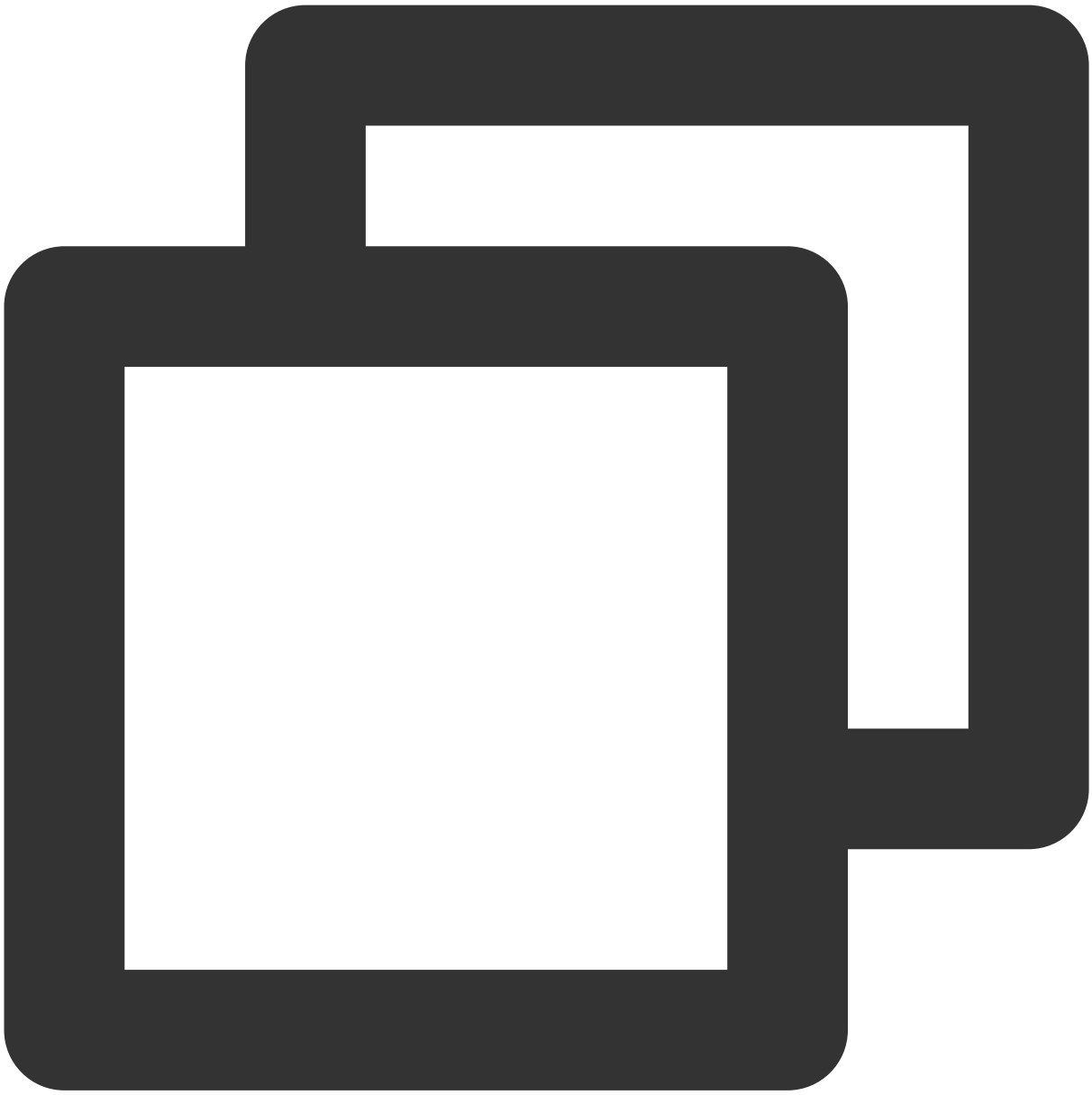
For a single client instance (a producer object created with the `new` command), the total number of connections established between it and all servers ranges from one to n (n refers to the number of brokers).
Each Java producer manages TCP connections as follows:
1. The `Sender` thread will be initiated 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.

©2013-2022 Tencent Cloud. All rights reserved.

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

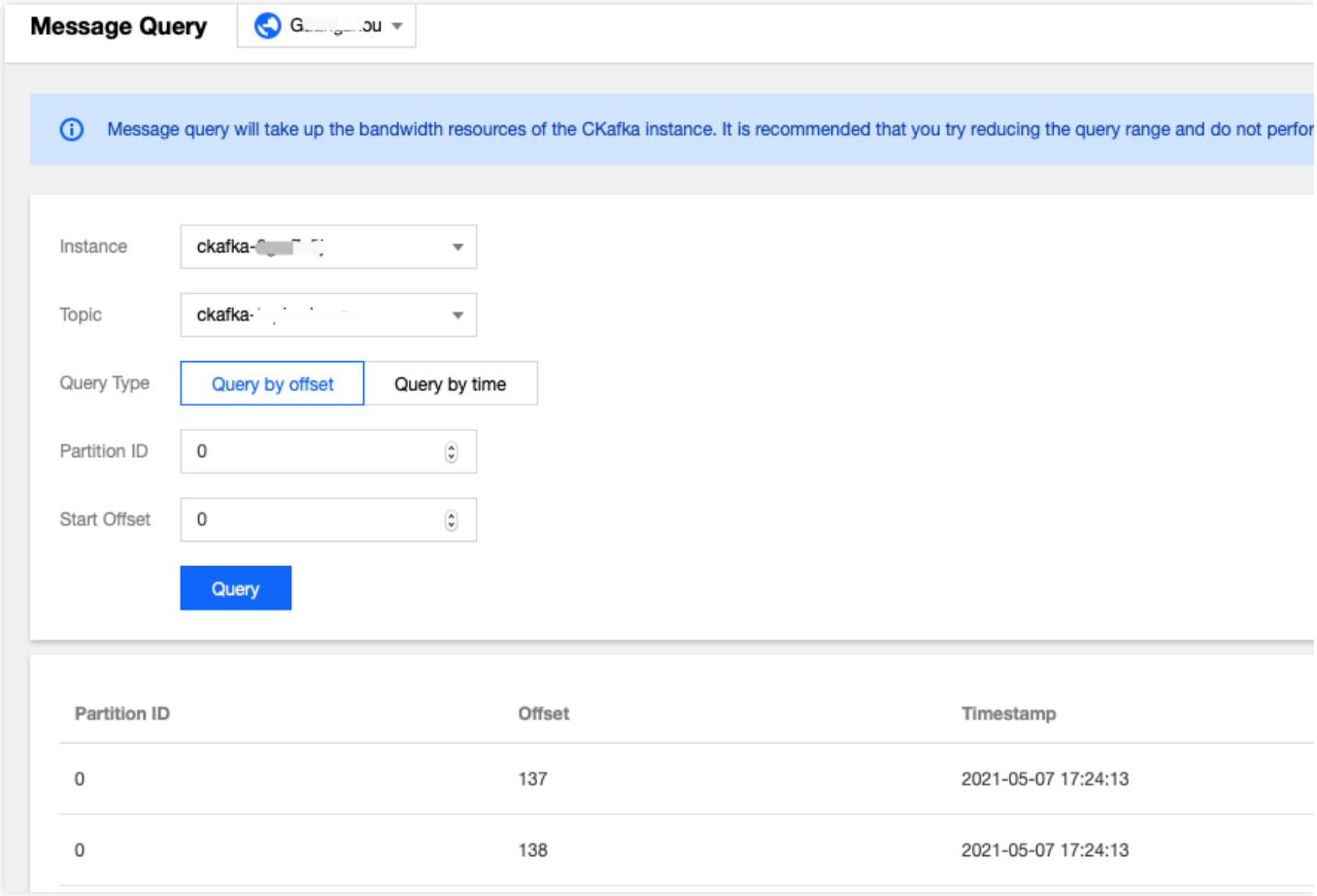


```
Future<RecordMetadata> future = producer.send(new ProducerRecord<>(topic, messageKe
RecordMetadata recordMetadata = future.get();
log.info("partition: {}", recordMetadata.partition());
log.info("offset: {}", recordMetadata.offset());
```

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.

©2013-2022 Tencent Cloud. All rights reserved.

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.



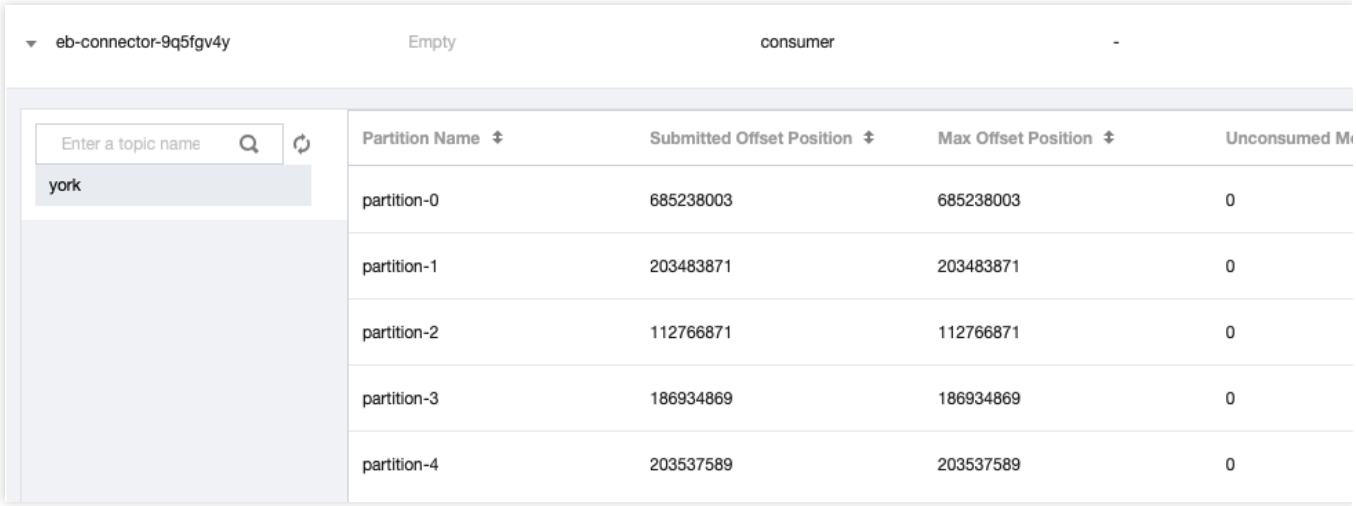
©2013-2022 Tencent Cloud. All rights reserved.

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:



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.

©2013-2022 Tencent Cloud. All rights reserved.

General

Last updated：2024-01-09 15:02:48

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

Currently, CKafka is fully 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?

©2013-2022 Tencent Cloud. All rights reserved.

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.

©2013-2022 Tencent Cloud. All rights reserved.

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.

©2013-2022 Tencent Cloud. All rights reserved.

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?

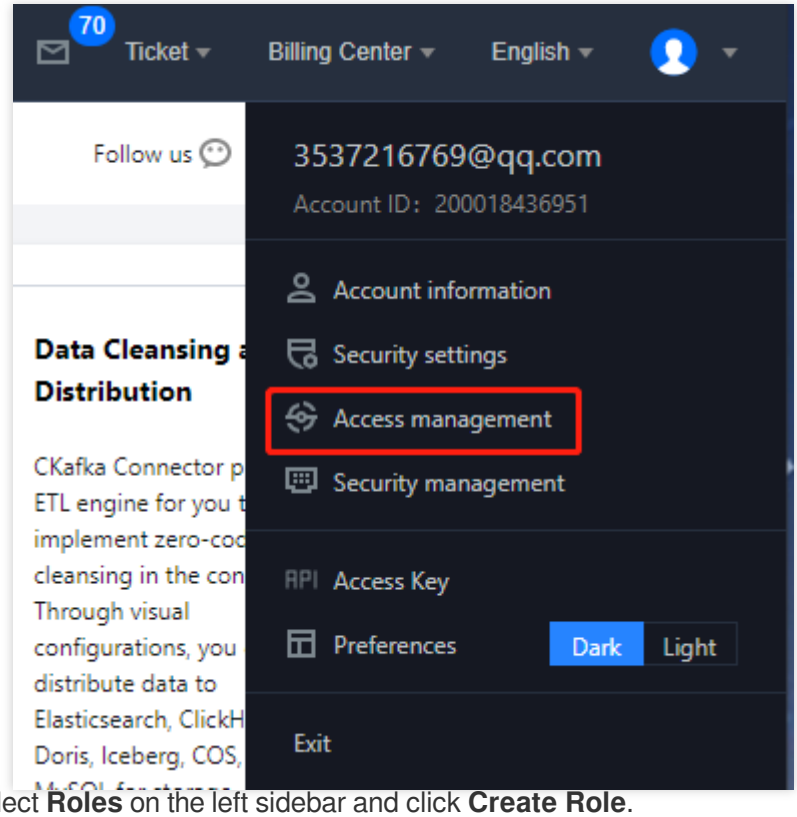
MongoDB uses change streams to record data changes. When any data in the database changes, the application can get the changed data through the change stream mechanism. You can consider it as a trigger executed in the application. As for what data the application wants to get in what form, the data can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

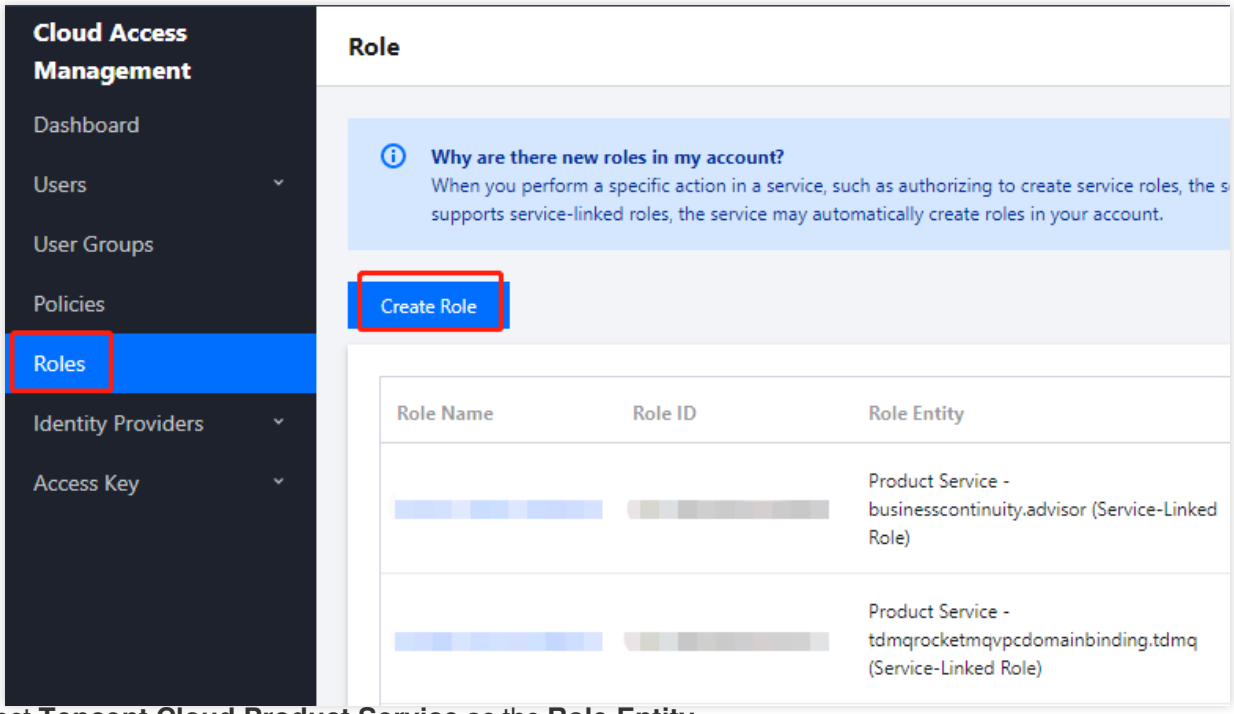
What should I do if a role is missing?

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

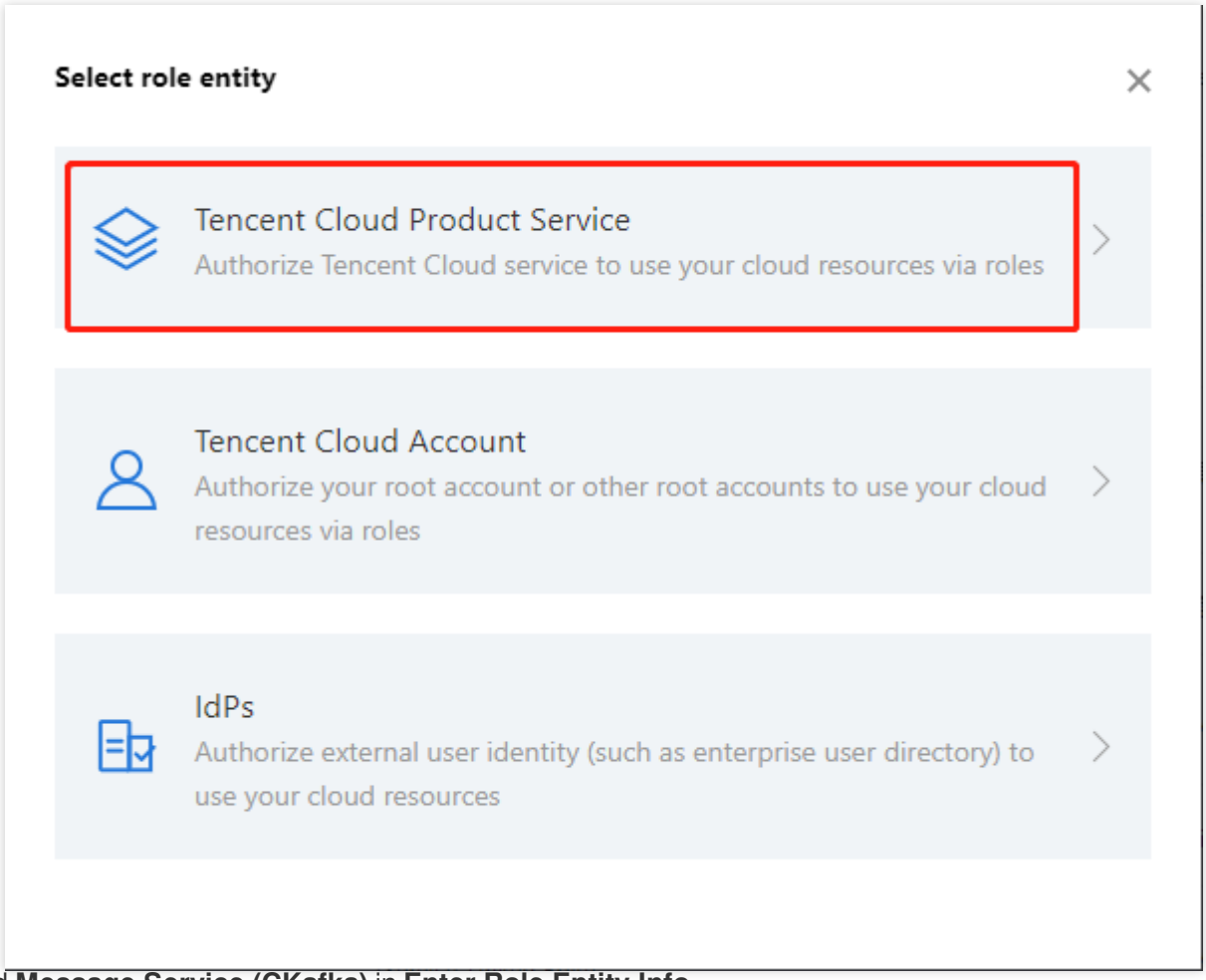


2. Select **Roles** on the left sidebar and click **Create Role**.

©2013-2022 Tencent Cloud. All rights reserved.

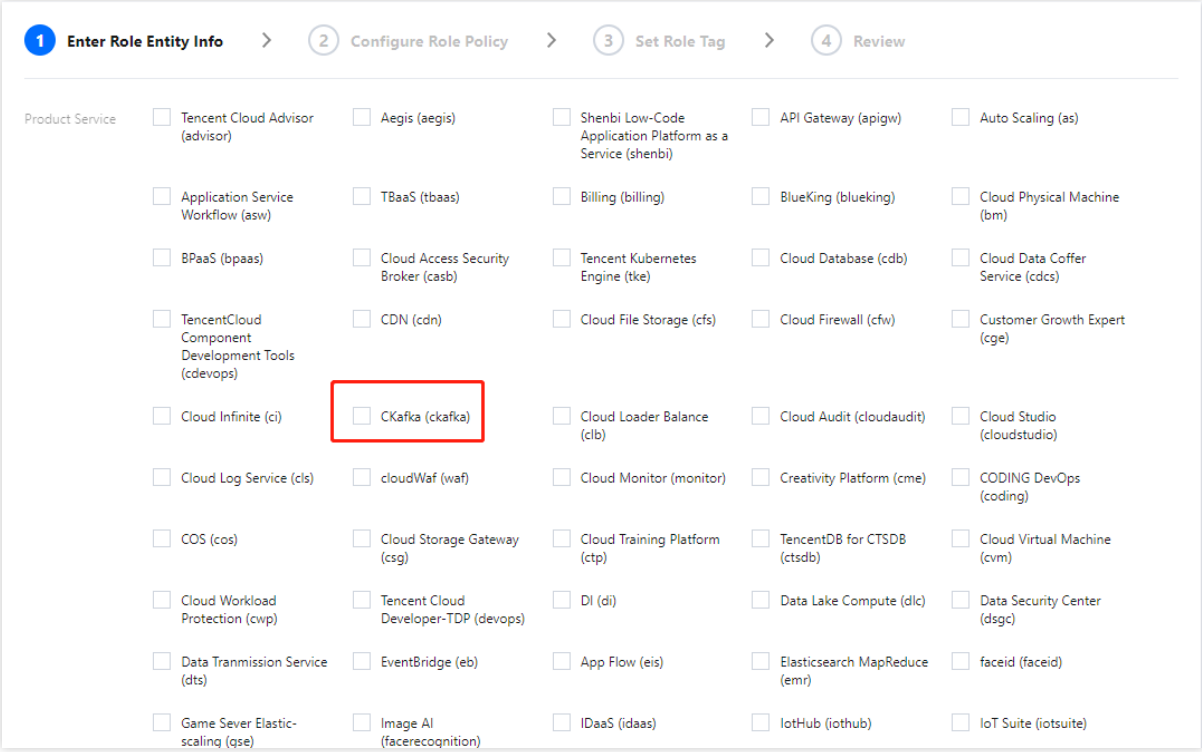


3. Select **Tencent Cloud Product Service** as the **Role Entity**.

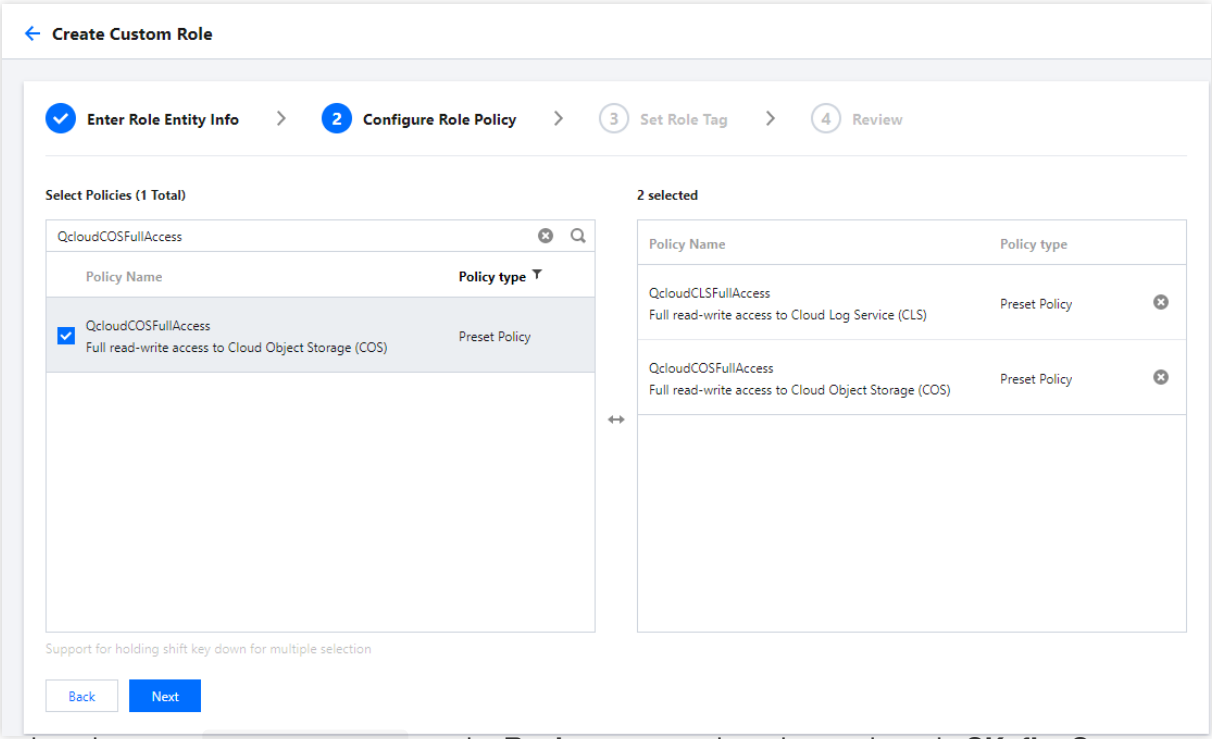


4. Find **Message Service (CKafka)** in **Enter Role Entity Info**.

©2013-2022 Tencent Cloud. All rights reserved.



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

©2013-2022 Tencent Cloud. All rights reserved.

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.

©2013-2022 Tencent Cloud. All rights reserved.