TDMQ for CKafka

Operation Guide

Product Documentation

©2013-2025 Tencent Cloud International Pte. Ltd.

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

Tencent Cloud

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

- Operation Guide
  - Instance Management
    - Creating Instance
    - Naming with Consecutive Numeric Suffixes or Designated Pattern String
    - Viewing Instance
    - Upgrading Instances
    - Downgrading Instances
    - Terminating/Returning Instances
    - Change from Pay-as-You-Go to Monthly Subscription
    - Upgrading Instance Version
    - Adding Routing Policy
    - Connecting to Prometheus
    - AZ Migration
    - Setting Maintenance Time
    - Setting Message Size
  - Topic Management
    - Instance Topic
      - Creating Topic
      - Viewing Topic
      - Deleting Topic
      - Sending Message
      - Setting Topic Traffic Throttling Rule
    - Elastic Topic
      - Topic Management
      - Subscription Management
  - Consumer Group
    - Consumer Group Overview
    - Creating Consumer Group
    - Querying Consumer Group
    - Deleting Consumer Group
    - Setting Offset
    - Automatically Resetting Offset
  - Monitoring and Alarms
    - Viewing Monitoring Data
    - Querying Advanced Monitoring (Pro Edition)

©2013-2025 Tencent Cloud International Pte. Ltd.

- Configuring Alarms
- Suggestions for Alarm Configuration
- Viewing Advanced Ops Features (Pro Edition)

Smart Ops

- Elastic Scaling
  - Disk Utilization Policy
  - Dynamic Partition Processing
  - Elastic Bandwidth

Permission Management

- Configuring ACL Policy
- CAM

Tag Management

Querying Message

Event Center

Migration to Cloud

- Migration Scheme Overview
- Step 1. Purchasing CKafka Instance
- Step 2. Migrating Topic to Cloud
- Step 3. Migrating Data to Cloud
  - Scheme 1: Single-Producer Dual-Consumer Migration
  - Scheme 2: Single-Producer Single-Consumer Migration
  - Scheme 3: Migration with MirrorMaker
  - Scheme 4: Migrating Unconsumed Data

Data Compression

©2013-2025 Tencent Cloud International Pte. Ltd.

Operation Guide
Instance Management
Creating Instance

Last updated：2025-04-14 18:05:03

Overview

This document describes how to create instances and topics in the CKafka console.

Directions

1. Log in to the CKafka console.
2. Click **Instance List** on the left sidebar and click **Create** to enter the purchase page.
3. On the instance purchase page, set the configuration information for purchase based on your actual needs.

Configuration Item	Parameter	Required or Optional	Parameter Description
Basic Configuration	Product form	Yes	**Serverful:** The classic form of CKafka. Users can purchase clusters of corresponding specifications based on requirements. With the change in business volume, certain attention needs to be paid to the CKafka cluster. **Serverless:** A brand-new form of CKafka. The goal is to completely release the user's effort and focus more on business logic. **Currently in public beta.**
	Billing Mode	Yes	Pro edition instances support two modes: **Monthly Subscription** and **Pay-As-You-Go**. Advanced Edition Instances support **Monthly Subscription** mode. **Monthly Subscription:** Payment is required in advance to use resources. It is mainly suitable for usage scenarios where the business is relatively stable and used for a long time. **Pay-As-You-Go:** Use resources first and then pay, mainly suitable for short-term situations with uncertain peak traffic or for testing.
	Cluster Type	Yes	The professional edition is primarily aimed at production environment customers on a large scale. The advanced

			edition is mainly targeted at test environment customers in small-scale scenarios. For specific differences, refer to Product Specifications.
	Region	Yes	Select a region with resources close to those of the client deployment. For the regions currently supported by CKafka, see Regions and Availability Zones.
Cluster Configuration	Name	No	If not specified, the default is unnamed. When purchasing multiple instances, you can create instance suffix numbers automatically in ascending order and specify pattern string function. For specific operations, refer to Batch sequential naming or naming with specified pattern strings.
	Series	Yes	Elastic storage: Supported only in the professional edition. In this form, billing by actual usage can be achieved, and theoretically unlimited space is available. Suitable for scenarios where consumption is followed by deletion, with small or fluctuating storage consumption. For details, refer to Elastic storage. Fixed storage: Storage is billed on a monthly subscription basis. Suitable for scenarios with message durability requirements and stable storage consumption.
	Kafka version	Yes	Choose an appropriate Kafka version according to your business needs. See CKafka version selection suggestion.
	Peak Bandwidth	Yes	Estimate the resource amount of peak bandwidth according to the rule of **peak bandwidth of business traffic** × **number of replicas**. CKafka will accumulate the bandwidth consumption of all replicas to calculate the actual peak bandwidth.
	Elastic bandwidth capability	No	Once enabled, when bandwidth usage exceeds original cluster specifications, elastic space is provided. Elastic bandwidth is billed based on actual usage. For details, please see Elastic Bandwidth. This feature's allowlist is in grayscale. If you need to enable it, please Contact Us. Once enabled, it cannot be disabled.
	Partition specification	Yes	The Partition limit for a CKafka instance is the cumulative total of **number of partitions * number of replicas**. The number of partitions included in the package (i.e., the minimum value) is free of charge. Additional partitions are billed in units of 100. Downgrading is not currently supported.
	Message	Yes	Ranges from 24 to 2160 hours. The default message

	retention		retention time is 72 hours. Messages will be deleted after exceeding the set retention duration to preserve sufficient disk space. CKafka supports automatic adjustment of disk utilization. After the disk utilization reaches the threshold, you can set the Dynamic Message Retention Policy to reduce message retention time or set the Automatic Disk Capacity Expansion to adjust disk space. For details, see Disk Water Level Processing.
	Cross-AZ Deployment	No	The professional edition supports deployment in a maximum of 4 different availability zones, and the advanced edition supports deployment in a maximum of 2 different availability zones. For how cross-availability zone deployment works, please refer to Cross-AZ Deployment.
Network Configuration	VPC Network	Yes	If users need to connect to other private networks, please refer to Add routing policy to modify the routing access rules.
	Custom SSL Certificates	No	When the access method is SSL, the SSL certificate is provided by the server by default. If you need to select a custom certificate, you can enable it. For the specific applicable method, refer to Custom SSL Certificates.
Other configuration	Tag	No	Tags are used to manage resources by category from different dimensions. For the method of use, please see Tag Management.
	Automatic Renewal	No	After checking, when the account balance is sufficient, instances and public network bandwidth will be auto-renewed monthly after expiration.

4. When the mouse is placed in the configuration costs in the bottom-right corner, you can view the billing item details.

5. Confirm the fee and then click **Buy Now**. Wait for 3 - 5 minutes to complete instance creation.

Health Status Description

CKafka has an inspection program for each instance. The inspection program checks the number of connections, disk usage percentage, production peak bandwidth, and consumption peak bandwidth of the instance. When these metrics exceed certain thresholds, different health statuses will be generated. The note is as follows:

Metric	Threshold (N)	Status Description

Number of connections (default maximum value 5,000)	N ≤ 80%	Healthy
	80% < N ≤ 95%	Alarm
	N > 95%	Abnormal
Disk usage percentage	N ≤ 80%	Healthy
	80% < N ≤ 95%	Alarm
	N > 95%	Abnormal
Production peak bandwidth (excluding replica bandwidth)	N ≤ 80%	Healthy
	80% < N ≤ 100%	Alarm
	N > 100%	Abnormal
Consumption peak bandwidth	N ≤ 80%	Healthy
	80% < N ≤ 100%	Alarm
	N > 100%	Abnormal

Note：

The default maximum number of connections is 5000. Threshold judgment is based on the percentage of the maximum value. Exceeding the maximum number of connections for an instance will cause the client to be unable to create new connections. If you consider this maximum value unreasonable in actual business, you can contact us to apply for an increase.

©2013-2025 Tencent Cloud International Pte. Ltd.

Naming with Consecutive Numeric Suffixes or Designated Pattern String

Last updated：2024-01-09 14:47:32

Overview

To allow you to name batch created CKafka instances according to a rule during creation, the features of automatically incrementing suffixed numbers and specifying pattern strings are provided.

Automatically incrementing suffixed numbers: If you purchase multiple CKafka instances, numbers will be suffixed to their names by default, such as ckafka1, ckafka2, ckafka3... For more information, see Automatically incrementing suffixed numbers.

Specifying the pattern string:

Specifying one pattern string: This is suitable if you need to create n instances and specify their names with incrementing serial numbers starting from x (such as ckafka3, ckafka4, ckafka5...). For more information, see Specifying one pattern string.

Specifying multiple pattern strings: This is suitable if you need to create n instances named with multiple prefixes and specified serial numbers for each prefix (such as ckafka3-big10-test, ckafka4-big11-test, ckafka5-big12-test). For more information, see Specifying multiple pattern strings.

Directions

Automatically incrementing suffixed numbers

This feature allows you to name batch purchased instances with the same prefix and automatically incrementing suffixed numbers.

Note:

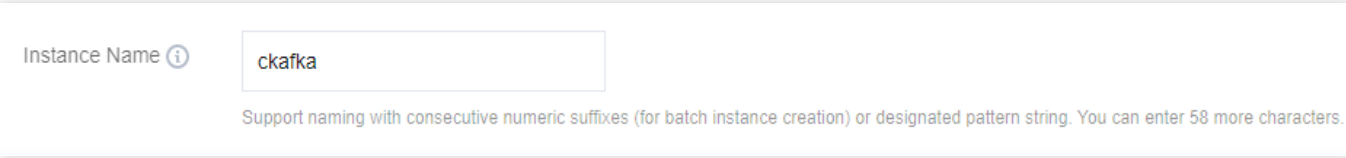
The created instances are suffixed with numbers starting from 1 by default. You cannot specify the starting number.

The following example assumes that you want to purchase 3 CKafka instances and name them in the format of "ckafka+serial number" (i.e., ckafka1, ckafka2, and ckafka3).

Operations on the purchase page

Operations through API

1. Purchase 3 instances as instructed in Creating Instance. Enter the instance name in the format of **prefix+serial number**. In this case, enter `ckafka` as the instance name:



2. Follow the prompts on the page and make the payment.

In the ModifyInstanceAttributes API, set the relevant fields:

Instance name: Set `InstanceName` to `ckafka`.

Specifying the pattern string

This feature allows you to name batch purchased instances in a complex format with specified serial numbers. You can use one or more pattern strings in instance names as needed. The instance name with a specified pattern string is in the format of **{R:x}**, where `x` indicates the starting number in the generated instance names and can be positive integers only.

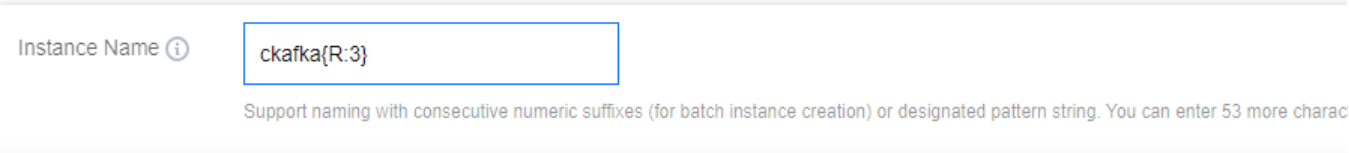
Specifying one pattern string

The following example assumes that you want to create 3 instances and name them with incrementing numbers starting from 3.

Operations on the purchase page

Operations through API

1. Purchase instances as instructed in Creating Instance. Enter the instance name in the format of **prefix+specified pattern string {R:x}**. In this case, enter `ckafka{R:3}` as the instance name:



2. Follow the prompts on the page and make the payment.

In the ModifyInstanceAttributes API, set the relevant fields:

Instance name: Set `InstanceName` to `ckafka{R:3}`.

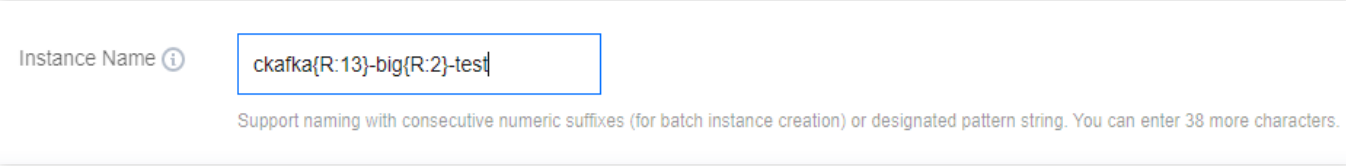
Specifying multiple pattern strings

The following example assumes that you want to create 3 instances and name them with the prefixes of `ckafka`, `big`, and `test` and serial numbers after `ckafka` and `big` incrementing from 13 and 2 respectively (i.e., ckafka13-big2-test, ckafka14-big3-test, and ckafka15-big4-test).

Operations on the purchase page

Operations through API

1. Purchase 3 instances as instructed in Creating Instance. Enter the instance name in the format of **"prefix+specified pattern string {R:x}-prefix+specified pattern string {R:x}-prefix"**. In this case, enter `ckafka{R:13}-big{R:2}-test` as the instance name:



2. Follow the prompts on the page and make the payment.

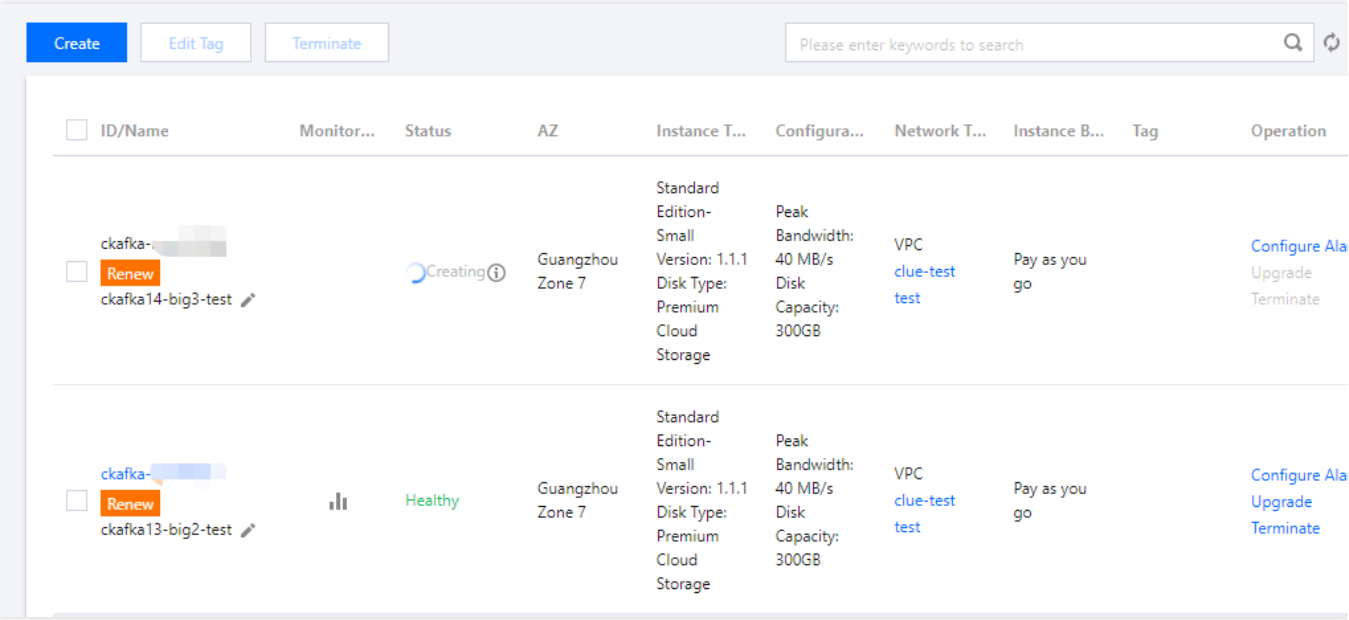
In the ModifyInstanceAttributes API, set the relevant fields:

Instance name: Set `InstanceName` to `ckafka{R:13}-big{R:2}-test`.

Feature Verification

After you batch create instances through automatically incrementing suffixed numbers or specifying the pattern string, you can verify the set instance names as follows:

Log in to the CKafka console and view the newly created instances. You can see that the batch purchased instances are named according to the rule you set:



Viewing Instance

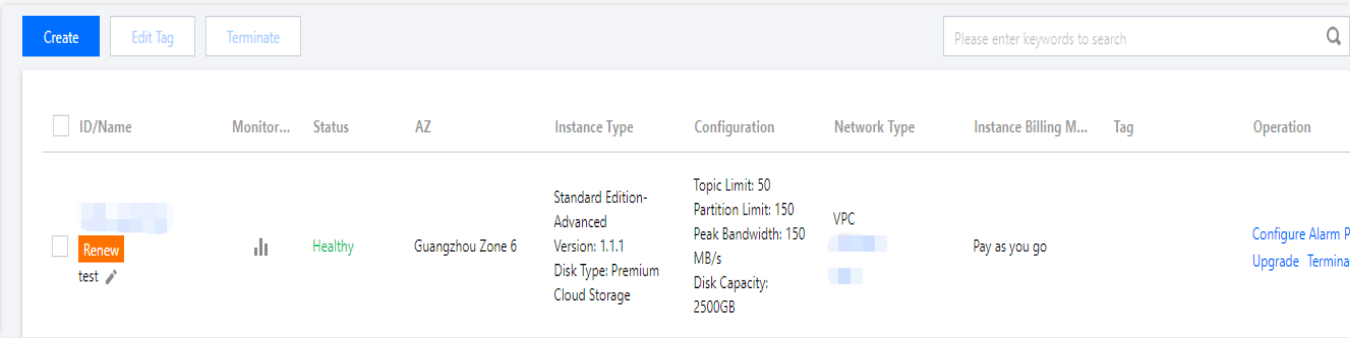
Last updated：2024-01-09 14:47:31

Overview

This document describes how to view the configuration information and status of an instance in the CKafka console.

Directions

1. Log in to the CKafka console.
2. Click **Instance List** on the left sidebar and click the **ID/Name** of the target instance to enter its **Basic Info** page, where you can view various information of the instance such as the status, configuration information, access mode, message retention period, and automatic topic creation.



Note:

The private IP and port (such as `10.6.206.110:9092`) in **Configuration Information** represent the communication address used to get the backend service. There may be multiple ports in a real access address. If access control is configured on your server, open ports 9092–9192 to the internet on it. As brokers may be automatically scaled out, the number of ports that need to be opened will increase after the scale-out, and you should reserve a sufficient number of ports.

Once you enable the automatic topic creation feature for the server, when you use or access metadata of a topic that does not exist, it will be automatically created with the configured number of replicas and partitions.

The total number of topics that can be automatically created varies by instance specification. For more information, see Billing Overview.

Health status description

CKafka runs an inspection program for each instance, which checks the instance's number of connections, disk utilization, production peak bandwidth, and consumption peak bandwidth. Different health status are displayed when these metrics exceed certain thresholds as detailed below:

Metric	Threshold (N)	Status Description
Number of connections (up to 5,000 by default)	N ≤ 80%	Healthy
	80% < N ≤ 95%	Alarmed
	N > 95%	Abnormal
Disk utilization	N ≤ 80%	Healthy
	80% < N ≤ 95%	Alarmed
	N > 95%	Abnormal
Production peak bandwidth (excluding the bandwidth of replicas)	N ≤ 80%	Healthy
	80% < N ≤ 100%	Alarmed
	N > 100%	Abnormal
Consumption peak bandwidth	N ≤ 80%	Healthy
	80% < N ≤ 100%	Alarmed
	N > 100%	Abnormal

Note:

The maximum number of connections is 5,000 by default, and you should determine the threshold based on the percentage of the maximum value. After the number of instance connections exceeds this maximum value, the client cannot create more connections. If this maximum value is unreasonable in your actual business, you can submit a ticket to increase it.

Upgrading Instances

Last updated：2025-04-08 17:48:55

Overview

If the current instance specifications cannot meet your business needs, you can upgrade them in the console.

Note:

The instance specification upgrade is a smooth upgrade, and your service will not be interrupted.

Upgrading may add new ports. You can later click View All IPs and Ports in the Access Method module on the Instance Details page to check. When configuring the security group, you need to allow access to all ports.

Prerequisites

Before upgrading, please check the following items:

1. Check whether the instance has unavailable public network routes, supporting networks, and VPC networks. For more information, see Adding Routing Policy.
2. Check whether the instance has unsynced replicas. For more information, see Viewing Topic.
3. Check whether the instance has unfinished tasks (e.g., data migration), abnormally created topics, or abnormally deleted topic data.

Note:

If there are unfinished tasks, it is recommended to wait until all are completed before proceeding with the upgrade. If any task encounters an exception, submit a ticket for assistance.

Directions

1. Log in to the CKafka Console.
2. Select **Instance List** in the left sidebar. After finding the target instance, click on the operation bar **More** > **Upgrade**.
3. On the specification configuration page, select the corresponding model based on peak bandwidth and disk capacity.

Notes:

Upgrade and supplement the price difference daily. For details, please refer to Monthly Subscription and Product Configuration Upgrade Instructions.

©2013-2025 Tencent Cloud International Pte. Ltd.

4. click **Next** to perform mode setting for modification.

Notes:

- When resource change of the underlying layer is identified during configuration adjustment and data migration is required. During the migration process, after each partition is migrated, the leader switch of the partition will be performed. For risks of leader switch, please refer to Typical Scenarios of Leader Switch.

The port during specification adjustment may change, and the actual communication IP address for the public network may vary (access point address unchanged). You can view all IPs and ports on the details page > access method. If your server has configured access restrictions (security group), open the port range on the server. For specific operations, see bind security group.

During the cluster configuration adjustment period, it is recommended that you do not perform operations on Topic management features, such as adding Topic or editing Topic attributes. For related features, please see related feature list.

Change time: When resource upgrade and data migration are identified, you can choose to immediately execute or use a custom time (it is recommended to choose night execution to reduce the impact on business).

Upgrade mode: If the backend detects that instance migration is required during the upgrade, you can select an upgrade mode as needed. Otherwise, you do not need to select an upgrade mode.

Stable mode: In Stable Mode, CKafka will limit the speed of data migration during the upgrade to preserve the instance's bandwidth attributes. This option is suitable for scenarios where you do not want to affect business operations.

High-Speed mode: In high-speed mode, CKafka will not limit the speed of data migration during the upgrade, which may affect the bandwidth for production and consumption of instances. This option is suitable for low business peak hours or scenarios where service suspension is acceptable.

5. Click **Submit** to complete the instance upgrade. You can view the real-time progress of the upgrade in the Status Column.If a scheduled upgrade was set, you can modify the scheduled time in the Status Bar.

Possible Causes of Upgrade Failure

1. One possible reason is that the disk resources in the current AZ cannot meet the requirements of this upgrade. It is recommended to contact the Tencent Cloud customer service to check whether there are sufficient resources.
2. If the high-speed mode is selected during the instance upgrade, and there are bandwidth-intensive production tasks in the cluster, the data migration delay will increase. You can view monitoring data to see whether there are excessive peaks of production and consumption traffic during the upgrade. For more information, see Viewing Monitoring Data.
3. The upgrade process takes too long because the maximum message size configured for the broker being migrated is 8 MB, while the target broker is set to 1 MB. This causes the target broker to be unable to receive oversized messages, leading to data migration failure and extending the upgrade duration. You can submit a ticket for assistance.

©2013-2025 Tencent Cloud International Pte. Ltd.

4. During the upgrade or migration of a cluster, an exception may occur while updating the broker IP, causing the new cluster to fail to pull the broker IP data. If you observe no monitoring data for an extended period, please submit a ticket for assistance. For more information, see Viewing Monitoring Data.

Non-operational Functions During Upgrade

During cluster reconfiguration, it is recommended that you do not operate Topic management-related functions, such as creating Topics or editing Topic attributes. The list of related functions is as follows.

Function Module	Specific Functions	Description
Topic management list	Create a Topic	Create a Topic under the cluster.
	Edit a Topic	Edit Topic attributes under the cluster, such as changing the number of partitions or replicas.
	Delete a Topic	Delete a Topic under the cluster.
	Throttling	Configure throttling policies for Topics.
	Manual partition balancing	Manually perform partition balancing to resolve uneven partition allocation issues.
Smart ops (Professional Edition exclusive feature)	Disk water level processing	Change dynamic message retention policy, or automatic disk expansion, switch status.
	Dynamic partition processing	Perform manual partition balancing diversion, or configure automated partition balancing policy.
	Bandwidth elastic configuration	Change the switch status of elastic bandwidth

©2013-2025 Tencent Cloud International Pte. Ltd.

Downgrading Instances

Last updated：2025-04-08 17:48:55

Operation Scenarios

ou can reduce the instance specification through the console, such as peak bandwidth, disk capacity, and the number of Partitions, to avoid wasting extra resources and costs. The downgrading operation of the instance specification is a smooth change, and your service will not be interrupted.

Note：

Standard Edition instance cannot be downgraded.

Prerequisites

You need to check the following before the downgrade:

1. Check whether the instance has unavailable public network routes, supporting networks, VPC networks and so on. For more information, see Adding Routing Policy.
2. Check whether the instance has unsynced replicas. For further details, see Viewing Topic Details.
3. Check whether the instance has unfinished tasks (data migration), abnormally created topics, or abnormally deleted topic data.
4. Check the resource usage of the instance in the last 7 days. If the peak bandwidth or peak storage of the instance in the last 7 days is larger than your target downgrade specification, the downgrade service may be damaged. Please determine whether to perform the downgrade operation according to your actual business need.

Note：

If there are unfinished tasks, we recommend that you wait for all the tasks to be completed before the downgrade. If task execution is abnormal, submit a ticket for assistance.

Directions

1. Log in to the CKafka console.
2. On the instance list page, select **More** > **Downgrade** in the **Operation** column to enter the upgrade page.
3. On the instance downgrade page, select the target specification for the downgrade.

4. Confirm the new configuration fee and then click **Next** to perform the change mode setting.

Notes:

When resource change of the underlying layer is identified during instance resizing and data migration is required. During the migration process, after each partition is migrated, the Leader of the partition will be switched. Please refer to Typical Scenario for the risks of Leader switch.
The port during specification adjustment may change, and the actual communication IP address for the public network may vary (access point address unchanged). You can check on the details page - access method - view all IPs and ports. If your server has configured access restrictions (security group), please open the port range on the server. For specific operations, see bind to security group.
During the cluster configuration adjustment period, it is recommended that you do not perform operations on Topic management features, such as add Topic or edit Topic attributes. For related features, please see Feature List.
Change time: When resource upgrade and data migration are identified, you can choose to execute immediately or at a custom time (it is recommended to choose night execution to reduce the impact on business).
Upgrade mode: When resource upgrade and data migration are identified, you can select the upgrade mode according to actual business needs; if no data migration is required, there is no need to select the upgrade mode.
Stable mode: CKafka will limit the data migration speed during the upgrade process and retain the bandwidth attribute of the instance to the maximum extent, which is suitable for scenarios where you do not want to interfere with the business.
High-speed mode: CKafka will not limit the data migration speed during the upgrade process, which will affect the production and consumption bandwidth of the instance. It is suitable for off-peak hours or scenarios where service interruption is allowed.
5. Click **Submit** and complete the downgrade as prompted.

Features Unavailable During Downgrade

During cluster reconfiguration, it is recommended not to operate on Topic management-related features, such as adding new Topics or editing Topic attributes. The following is a list of related features:

Feature Module	Specific Feature	Description
Topic management list	Create a Topic	Create a Topic within the cluster.
	Edit a Topic	Edit Topic attributes within the cluster, such as changing the number of partitions or replicas.
	Delete a Topic	Delete a Topic within the cluster.
	Traffic throttling	Configure traffic throttling policies for the Topic.
	Manual partition traffic balancing	Manually rebalance partition traffic to address issues of uneven partition distribution.
Intelligent Ops	Disk watermark	Modify the dynamic message retention policy or enable automatic

(exclusive feature of Professional Edition)	management	disk scale-out, with an on/off toggle.
	Dynamic partition management	Manually rebalance partition traffic, or configure automated partition balancing policies.
	Elastic bandwidth configuration	Change the status of elastic bandwidth.

©2013-2025 Tencent Cloud International Pte. Ltd.

Terminating/Returning Instances

Last updated：2025-04-11 16:16:19

Overview

If you no longer need a CKafka instance, you can terminate and release it.
The lifecycle of a CKafka instance refers to all statuses from the start of the instance to its release. Properly managing an instance in different statuses can ensure that applications running on it can provide services economically and efficiently. The statuses of an instance are as follows:

Status	Status Attribute	Description
Creating	Intermediate status	The instance has been created but is not running yet.
Running	Stable status	The instance is running normally, and the disk space, traffic, and the number of connections are all within the normal range.
Deleting	Intermediate status	The instance is being deleted through the console or API.
Isolated	Intermediate status	The instance is isolated through the console or API, or it is in a seven-day isolation status due to overdue payments.
Failed to create	Intermediate status	You have purchased an instance through the console or API but are not assigned the instance. In this case, submit a ticket.
Failed to delete	Stable status	CKafka fails to release resources after the instance is manually deleted or is not renewed 14 days after expiration (including the 14th day).

Directions

Terminate and Release Annual and Monthly Subscription Instances
Terminate and Release Pay- As-You-Go Instances

Manual Return

You can manually terminate a monthly-subscribed instance before it expires by following the steps below:
1. Log in to the CKafka console.
2. Select **Instance List** in the left sidebar, find the target instance, and click the action bar **More** > **Terminate/Return**.

3. Confirm in the pop-up window for returning the instance, and then the instance can be terminated.

Note:

After a monthly subscribed instance is terminated, it will be in “Isolated” status for seven days in the CKafka console. Please note to backup data in advance before termination.

Isolated instances cannot produce or consume data. Data and configurations saved in CKafka will not be terminated, but **expired messages will still be automatically deleted according to the open-source Kafka mechanism**.

For instances in the seven-day isolation status, you can go to the CKafka console to renew them by clicking **Renew** in the **Operation** column on the instance list page. Successfully renewed instances can go back to the running status and be normally used.

For more information, see Refunds.

Automatic termination upon expiration/overdue payment

Monthly-subscribed instances can be retained in the CKafka console for up to seven natural days after they expire or have overdue payments. You can continue to use them if you renew them within 7 days after expiration. For details, see Payment Overdue.

If the CKafka instance is not renewed within 7 (inclusive) days after expiration, its resources will be released within 24 hours after expiration, and data will be cleared and cannot be restored.

Destroying instances

A monthly-subscribed instance will be retained for seven days in the CKafka console after manual termination or automatic termination upon expiration/overdue payment, during which it will be in “Isolated” status. After 7 (inclusive) days, it will be completely released, and you can delete it completely.

Note:

After deletion, all data will be cleared and cannot be recovered. Please back up the data in advance.

1. Log in to the CKafka console.
2. On the instance list page, click **More** > **Terminate/Return** in the **Operation** column.
3. In the pop-up window, click **Submit** to delete the instance.lease.

For pay-as-you-go instances, you can directly completely terminate the instance from the console.

Notes:

Completely terminating an instance will immediately destroy the instance from the account instance list. Afterwards, the instance can no longer be restored through renewal or other methods.

After termination, all data will be erased and unrecoverable. Back up data in advance.

1. Log in to the CKafka console.
2. Select **Instance List** in the left sidebar, find the target instance, and click the action bar **More** > **Terminate/Return**.
3. Confirm the destruction details and then click **Next**. In the pop-up window for confirming termination, **confirm destruction** to terminate the instance.

Change from Pay-as-You-Go to Monthly Subscription

Last updated：2025-03-11 14:46:23

Overview

For your convenience, CKafka allows you to change the billing mode from pay-as-you-go to monthly subscription, so that you can convert temporarily used pay-as-you-go instances into monthly subscribed instances for long-term and stable use. You can perform the change operation in the console.

Change Rules

You can follow the rules below to use the billing mode change feature in the CKafka console:
When a pay-as-you-go instance is changed to monthly subscribed instance, a renewal order will be generated. You must pay the order for the change to take effect. If you don't make the payment or your payment is unsuccessful, you can check and handle the order on the Billing Center > Order Management page.
For instances with billing mode changed from **Pay-as-You-Go** to **Monthly Subscription**, the five-day free returns are not supported.
Upon the successful completion of the payment and billing mode change, the instance will be billed on a monthly subscription basis, with the start time being the moment the change takes effect.
You can't perform repeated billing mode change on the instance before making the payment.
You won't be able to pay an unpaid order if the order amount has changed because of your instance configuration adjustment (such as adjusting configurations/bandwidth/disk and reinstalling system). You need to cancel the unpaid order in Billing Center > Order Management before performing a new change operation.
The change from pay-as-you-go instances to monthly subscribed instances is irreversible.
The change from pay-as-you-go instances to monthly subscribed instances can't be performed in batches.

Directions

1. Log in to the CKafka console.
2. On the instance list page, select the target pay-as-you-go instance, and click **More** > **Change from Pay-as-You-Go to Monthly Subscription** in the **Operation** column.
3. In the **Change from Pay-as-You-Go to Monthly Subscription** pop-up window, set the renewal duration and whether to automatically renew as needed.

©2013-2025 Tencent Cloud International Pte. Ltd.

4. Click **OK** to complete the change operation.

Upgrading Instance Version

Last updated：2024-11-07 14:49:02

The **Professional Edition** of TDMQ for CKafka supports minor kernel version upgrades, enabling new feature usage, performance enhancement, and issue resolution.

Scenarios

Automatic upgrade
Scenario 1: When a severe bug or security vulnerability occurs in CKafka, the system will perform kernel minor version upgrade during the maintenance time and send upgrade notifications through the Message Center and SMS.
Scenario 2: When a CKafka cluster is migrated due to the cluster configuration upgrade, storage capacity expansion/reduction, or CKafka version upgrade, the system will automatically upgrade the cluster's kernel to the latest minor version.
Manual upgrade
In addition to automatic upgrades, users can manually upgrade the minor kernel version through the console.

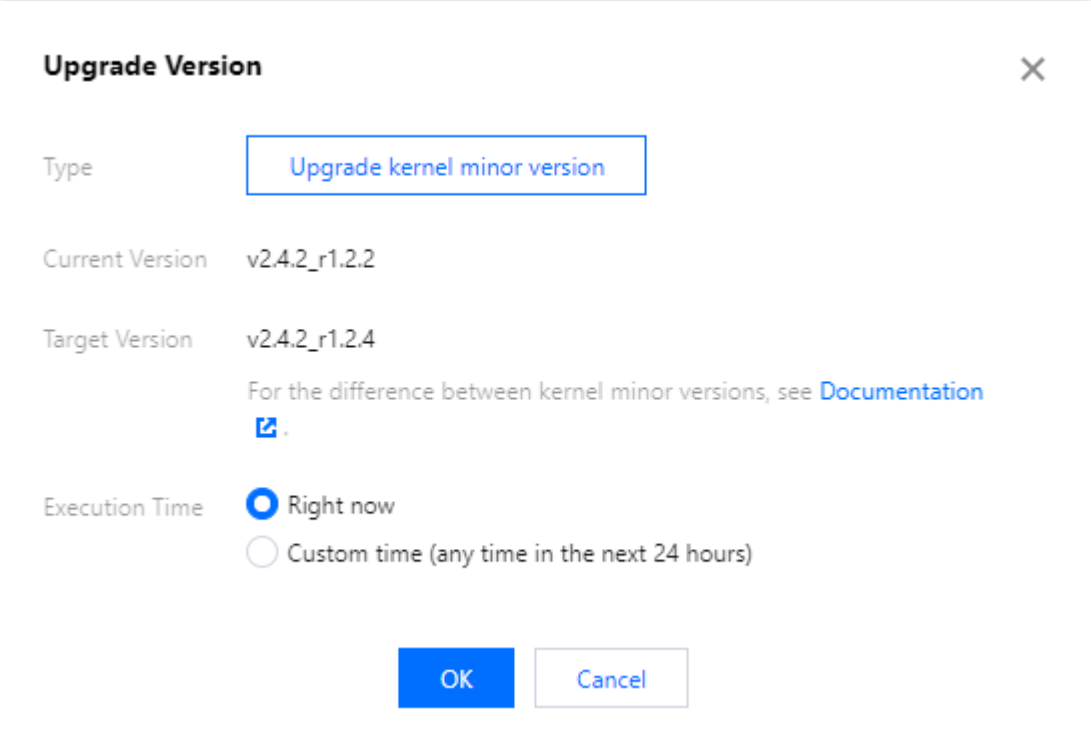
Note

This feature is only supported for Pro Edition.
Cluster switch will be required after the version upgrade is completed, which will cause your instance to disconnect for seconds. We recommend that you conduct the switch during the instance maintenance time. Make sure that your business has a reconnection mechanism.
Instances can be upgraded from an earlier version to a later one but cannot be downgraded.
The instance's minor version is automatically detected and upgraded by the system, and you cannot select the target version.
If an instance is upgraded to a compatible version, no billing changes will be caused.

Directions

1. Log in to the CKafka console.
2. On the left sidebar, select **Instance List** and click the ID/name of the target instance to enter the instance details page.
3. In the **Basic Info** module, click **Upgrade Version** on the right of the instance version, and set the target version and upgrade time.

©2013-2025 Tencent Cloud International Pte. Ltd.



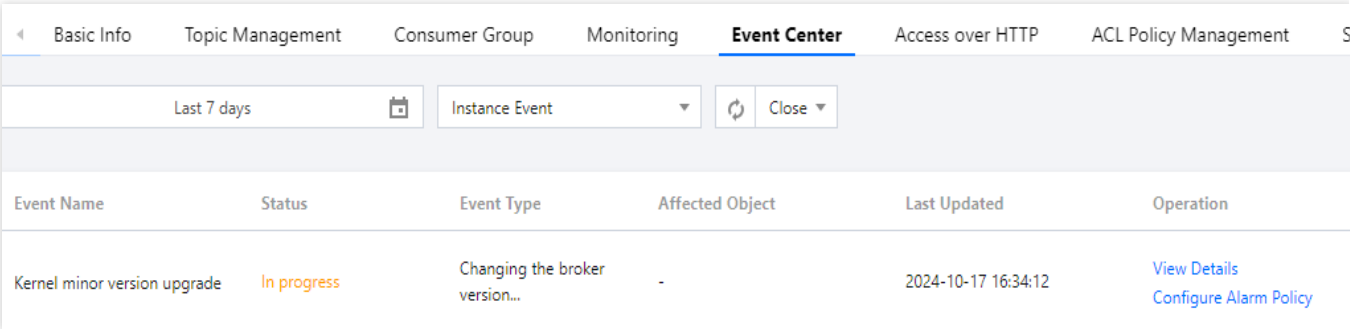
Current Version: The current kernel version.

Target Version: The target version is the latest broker version by default. The system will automatically detect the minor version. If the **Upgrade kernel minor version** option is grayed out, it indicates that the instance is already on the latest minor version. For differences between kernel minor versions, see Broker Release Notes.

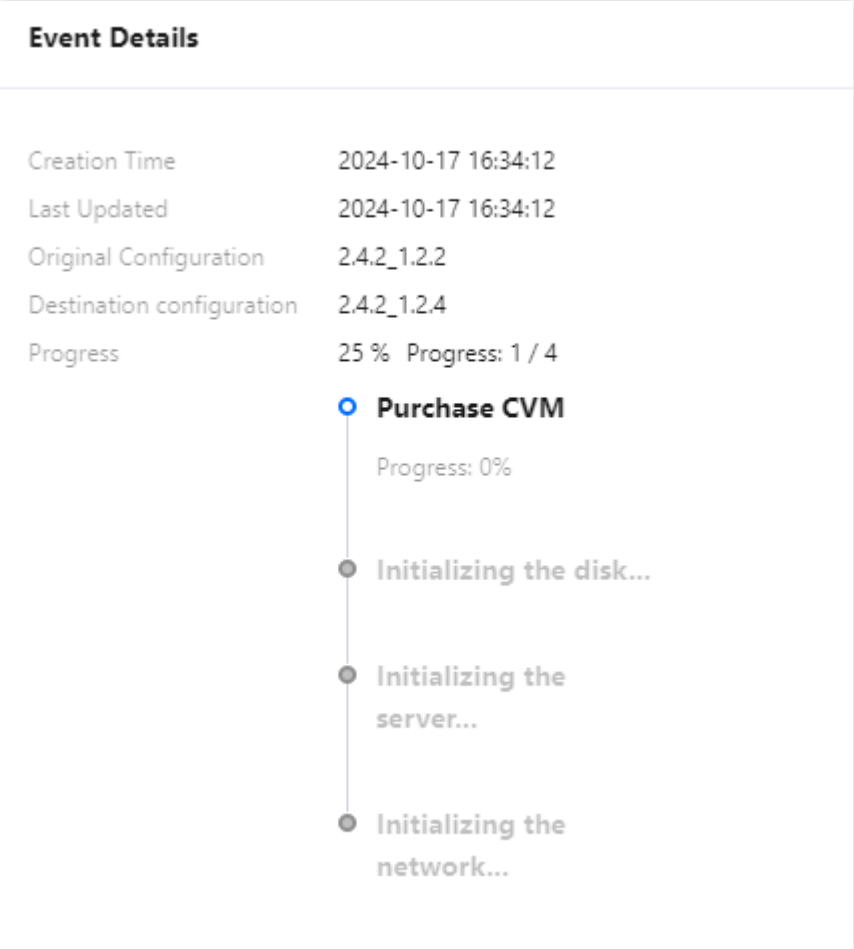
Execution Time: You can select **Immediate execution** or **Custom time** (any time within the next 24 hours). We recommend that you select a time during off-peak hours.

4. Click **OK** to submit the upgrade task.

5. Select the **Event Center** tab at the top of the page, and you can see a record of instance version upgrade.



6. Click **View Details** in the **Operation** column of the record to view the detailed upgrade task progress.



7. Return to the instance list page. You can see that the status of the instance has changed to **Changing configurations...**, and you can also view the upgrade progress.

8. During instance reconfiguration, it is recommended not to operate on related features under Topic Management and Intelligent Ops to avoid unexpected errors that may cause reconfiguration failure.

Adding Routing Policy

Last updated：2024-01-09 14:47:32

Overview

This document describes how to configure a routing rule in the CKafka console to enhance network access control in public/private network transfers. For more information on public network access, see Configuring ACL Policy.

Route Type	VPC	Public Domain Name Access
Access mode	PLAINTEXT SASL_PLAINTEXT SASL_SSL (only supported by Pro Edition instances) SASL_SCRAM (only supported by instances on v2.4.1; for existing instances, submit a ticket for application)	SASL_PLAINTEXT SASL_SSL (only supported by Pro Edition instances)

Directions

Note:

Up to five routes can be created for an instance and there can be only one public network route.

VPC

Public domain name access

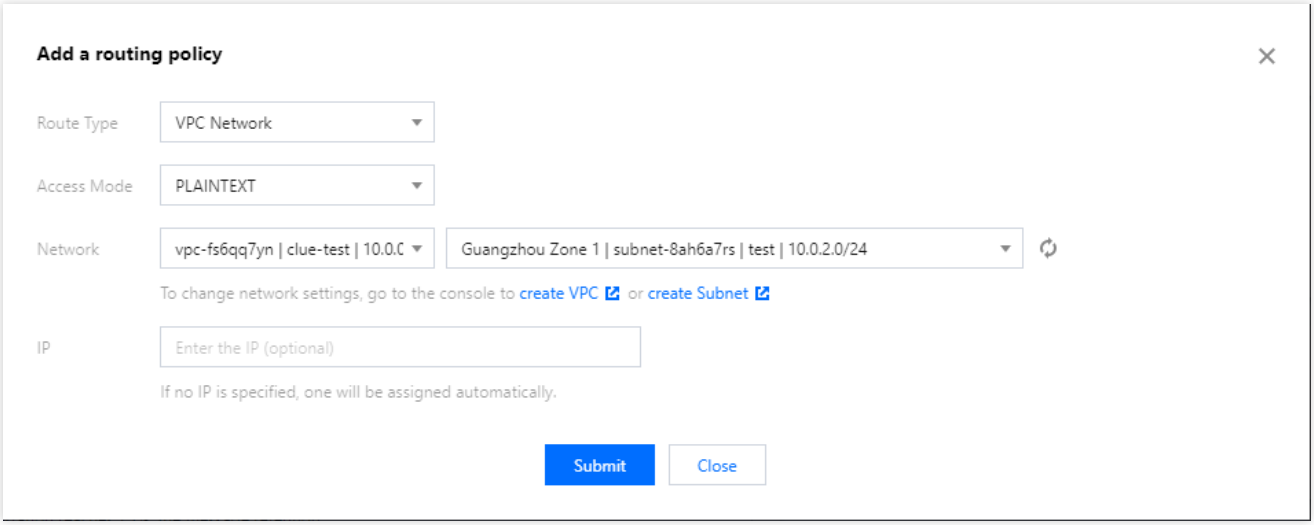
Operation scenario: When purchasing an instance, if you select VPC and choose a corresponding VPC environment (such as VPC A), then CKafka services (such as data production and consumption) can only be accessed from VPC A. If you subsequently find that you need to access the CKafka services in VPC A from other VPCs (such as VPC B), you can select an appropriate routing policy for VPC by configuring the access mode.

Suggestion: To ensure security, this access mode supports user management and ACL policy configuration to manage user access permission. Configure as appropriate.

Directions:

1. Log in to the CKafka console.
2. Click **Instance List** on the left sidebar and then the **ID/Name** of your instance to enter the instance details page.
3. On the page you enter, click **Add a routing policy** in the "Access Mode" module.
4. In the pop-up window, select **VPC Network** as the route type and select the access mode and network.

©2013-2025 Tencent Cloud International Pte. Ltd.



Note:

If you select VPC access, you can specify the IP to keep it unchanged when changing the access mode.

5. Click **Submit** to add the VPC network.

6. Click **View All IPs and Ports** in the **Operation** column to view the list of IPs and ports that should be opened.

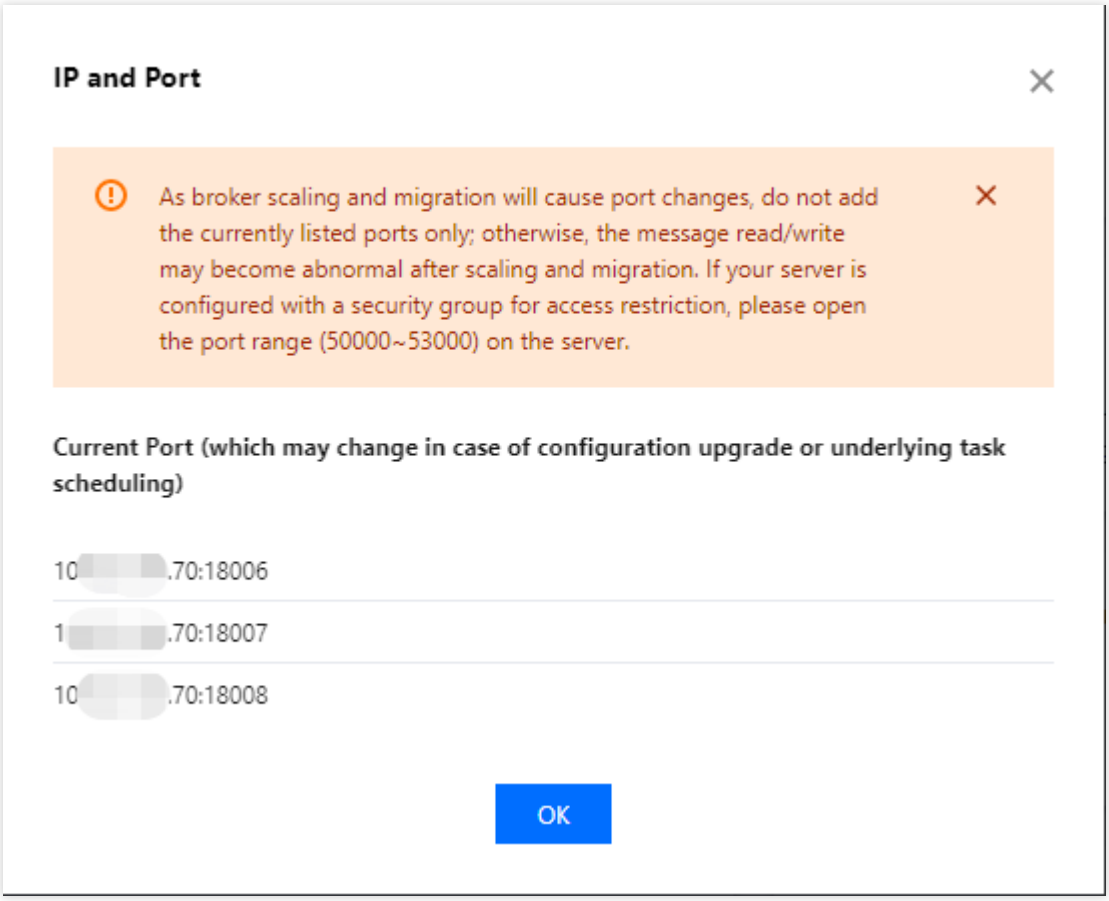
Note:

As broker scaling and migration will cause port changes, do not add only the current listed ports; otherwise, message read/write may become abnormal after scaling and migration. If your server is configured with security groups for access restrictions, you should open the following port ranges on the server:

Port range that should be opened for the VPC route: 9092–60000

Port range that should be opened for the public network route: 50000–53000

Port range that should be opened for the supportive route: 6000–12000

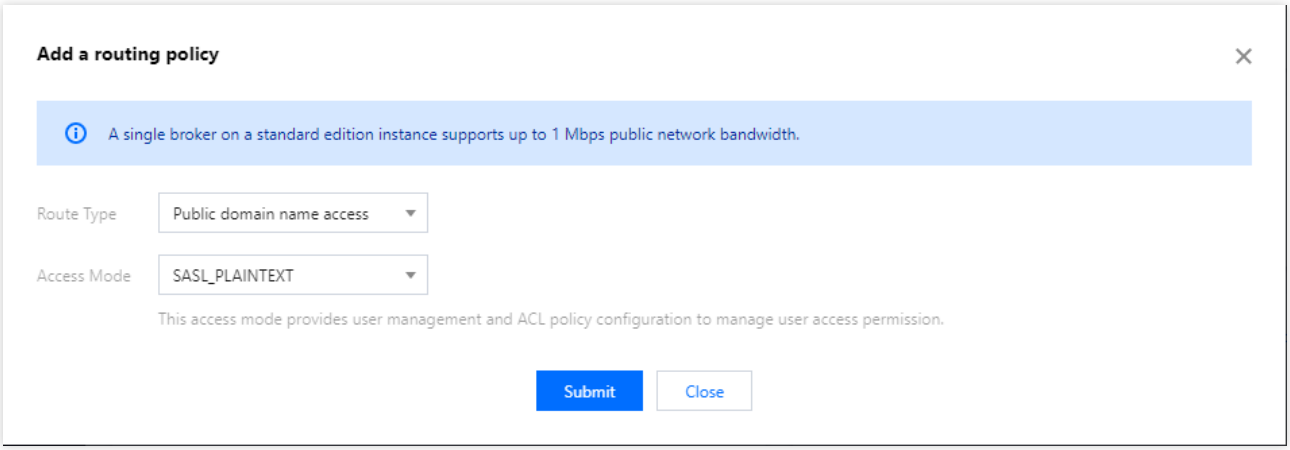


Operation scenario: If your consumer or producer is located in a self-built data center or another cloud, you can produce and consume data in CKafka through public network access.

Suggestion: To ensure security, Kafka offers various security authentication mechanisms, which mainly fall into two categories: SSL and SASL2. SASL/PLAIN is a more commonly used authentication method based on account and password. CKafka supports SASL_PLAINTEXT and SASL_SSL authentication. We recommend that you configure the authentication method as appropriate when selecting public domain name access.

Directions:

1. Log in to the CKafka console.
2. Click **Instance List** on the left sidebar and then the **ID/Name** of your instance to enter the instance details page.
3. On the page you enter, click **Add a routing policy** in the "Access Mode" module.
4. In the pop-up window, select **Public domain name access** as the route type and select the access mode and network.



5. Click **Submit** to add the public network routing policy.
6. Click **View All IPs and Ports** in the **Operation** column to view the list of IPs and ports that should be opened.

Note:

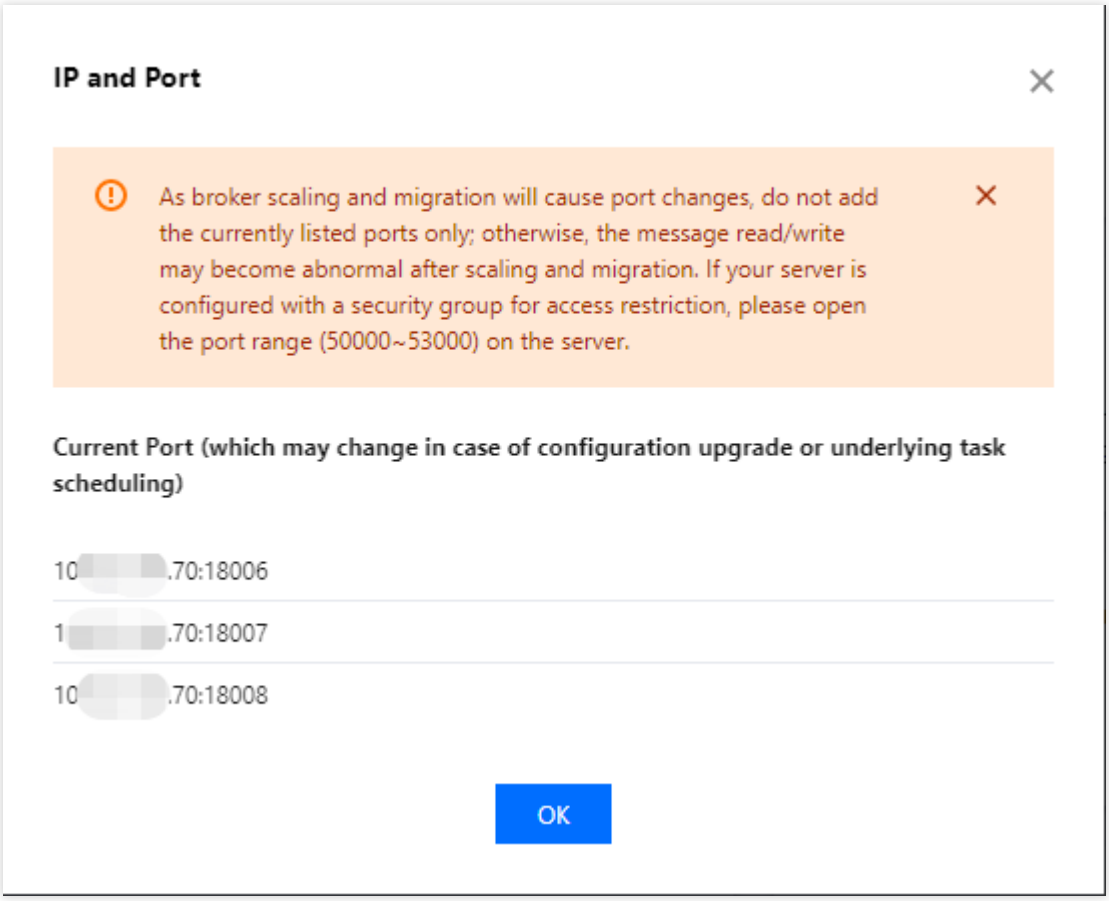
As broker scaling and migration will cause port changes, do not add only the current listed ports; otherwise, message read/write may become abnormal after scaling and migration. If your server is configured with security groups for access restrictions, you should open the following port ranges on the server:

Port range that should be opened for the VPC route: 9092–60000

Port range that should be opened for the public network route: 50000–53000

Port range that should be opened for the supportive route: 6000–12000

CKafka provides 3 Mbps public network bandwidth free of charge by default, which can be increased for Pro Edition instances. For detailed directions, see Public Network Bandwidth Management.


IP and Port
As broker scaling and migration will cause port changes, do not add the currently listed ports only; otherwise, the message read/write may become abnormal after scaling and migration. If your server is configured with a security group for access restriction, please open the port range (50000~53000) on the server.
Current Port (which may change in case of configuration upgrade or underlying task scheduling)
10 .70:18006
1 .70:18007
10 .70:18008
OK

Connecting to Prometheus

Last updated：2024-11-07 14:49:02

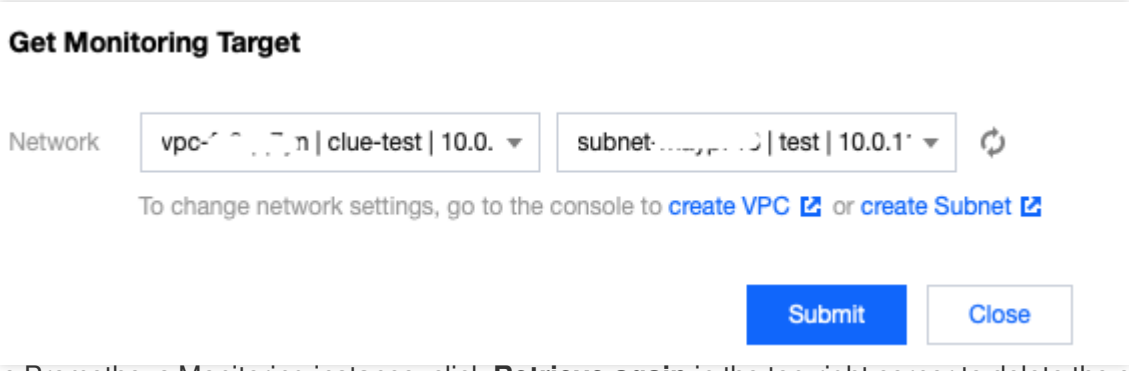
Overview

By default, a CKafka Pro Edition instance allows connection with external monitoring services for instance monitoring through the provided access point, including monitoring metrics of open-source Kafka such as **unsynced replicas and incoming message rate**.

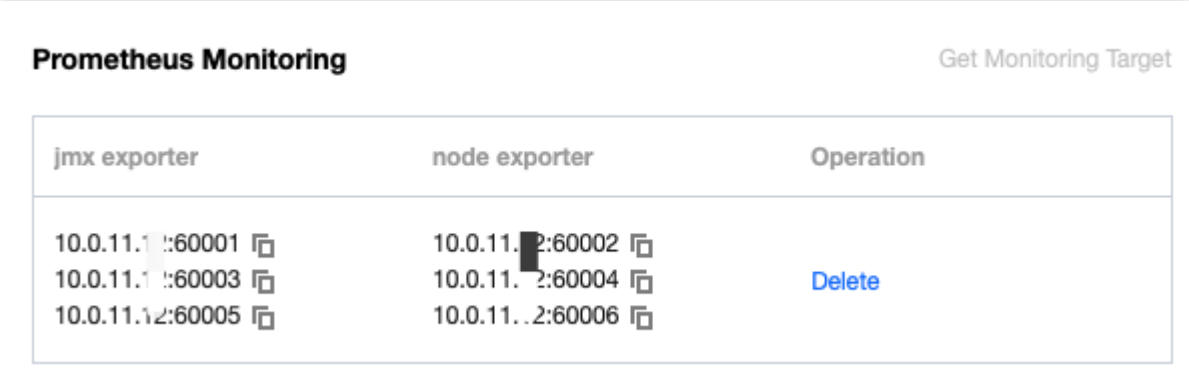
CKafka Pro Edition instances currently provide Prometheus to scrape the metric information of broker nodes, including basic monitoring metrics such as CPU, memory usage, and system load, as well as the metrics exposed by the broker's JMX.

Directions

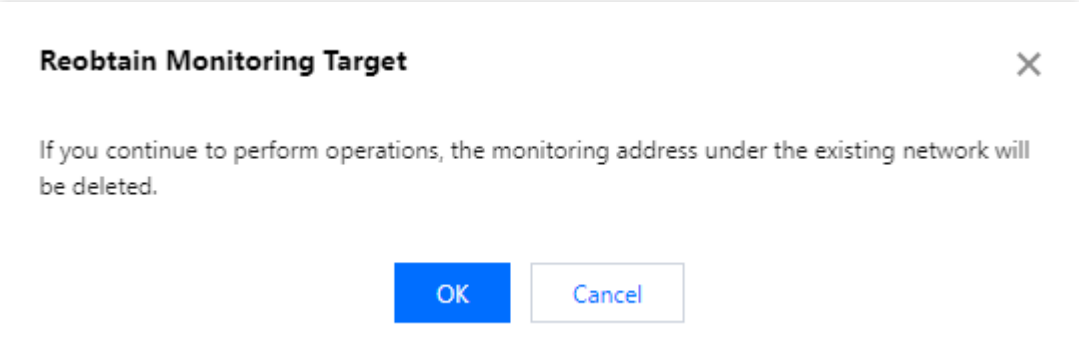
1. Log in to the CKafka console.
2. Select **Instance List** on the left sidebar and click the **ID** of the target instance to enter its basic information page.
3. Click **Get Monitoring Target** in the top-right corner of the **Prometheus Monitoring** module and select the VPC and subnet.
4. Click **Submit** to obtain a set of monitoring targets.



5. In the Prometheus Monitoring instance, click **Retrieve again** in the top-right corner to delete the current monitoring addresses under the existing network.



6. Click **Confirm** to retrieve the monitoring targets again.



7. Download Prometheus and configure the monitoring scrape address.

7.1 Enter the directory of the Prometheus package and run the following command to decompress it.

```
tar -vxf prometheus-2.30.3.linux-amd64.tar.gz
```

7.2 Modify the `prometheus.yml` configuration file by adding the `jmx_exporter` and `node_exporter` scrape tasks.

```
scrape_configs:
  # The job name is added as a label `job=<job_name>` to any timeseries scraped fro
  - job_name: "prometheus"
    # metrics_path defaults to '/metrics'
    # scheme defaults to 'http'.
    static_configs:
      - targets: ["localhost:9090"]

  - job_name: "broker-jmx-exporter"
    scrape_interval: 5s
    metrics_path: '/metrics'
    static_configs:
      - targets: ['10.x.x.0:60001','10.x.x.0:60003','10.x.x.0:60005']
        labels:
           application: 'broker-jmx'
  - job_name: "broker-node-exporter"
    scrape_interval: 10s
```

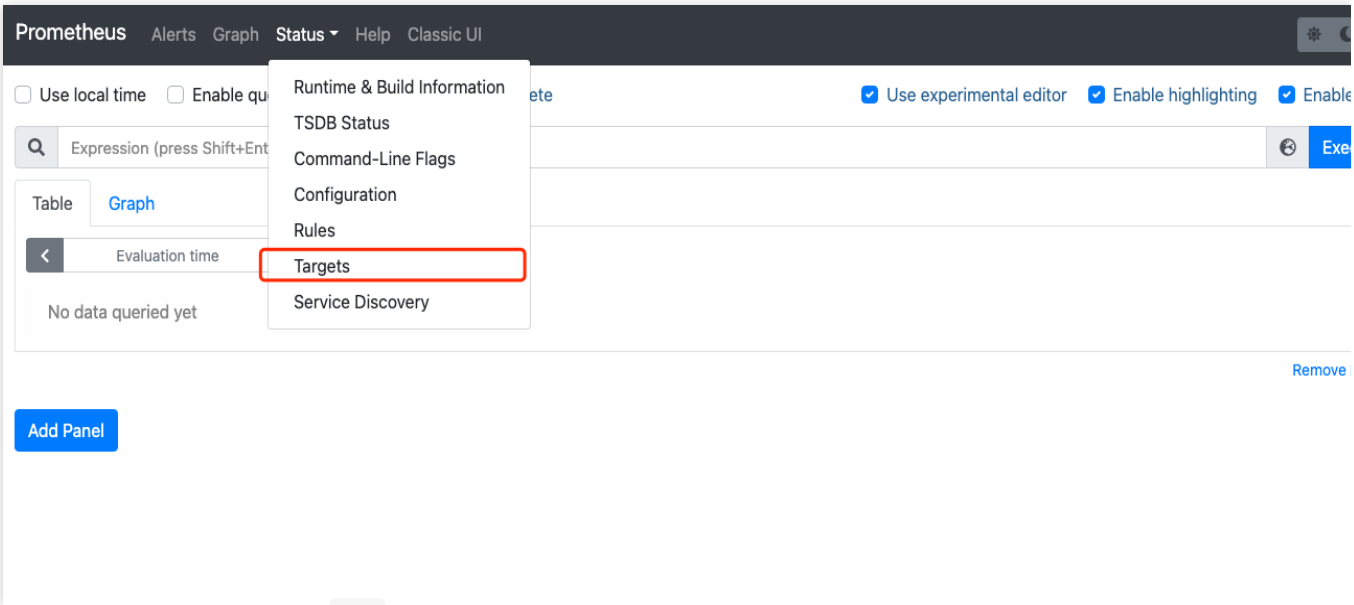
```
    metrics_path: '/metrics'
    static_configs:
      - targets: ['10.x.x.0:60002','10.x.x.0:60004','10.x.x.0:60006']
        labels:
          application: 'broker-node'
```

Here, `broker-jmx-exporter` is the tag configured for the `jmx` metric of the broker scraped by Prometheus, `Targets` contains the information of the mapped port, `broker-node-exporter` is the tag configured for the basic metrics of the node of the scraped broker, and `scrape_interval` is the frequency of scraping metric data.

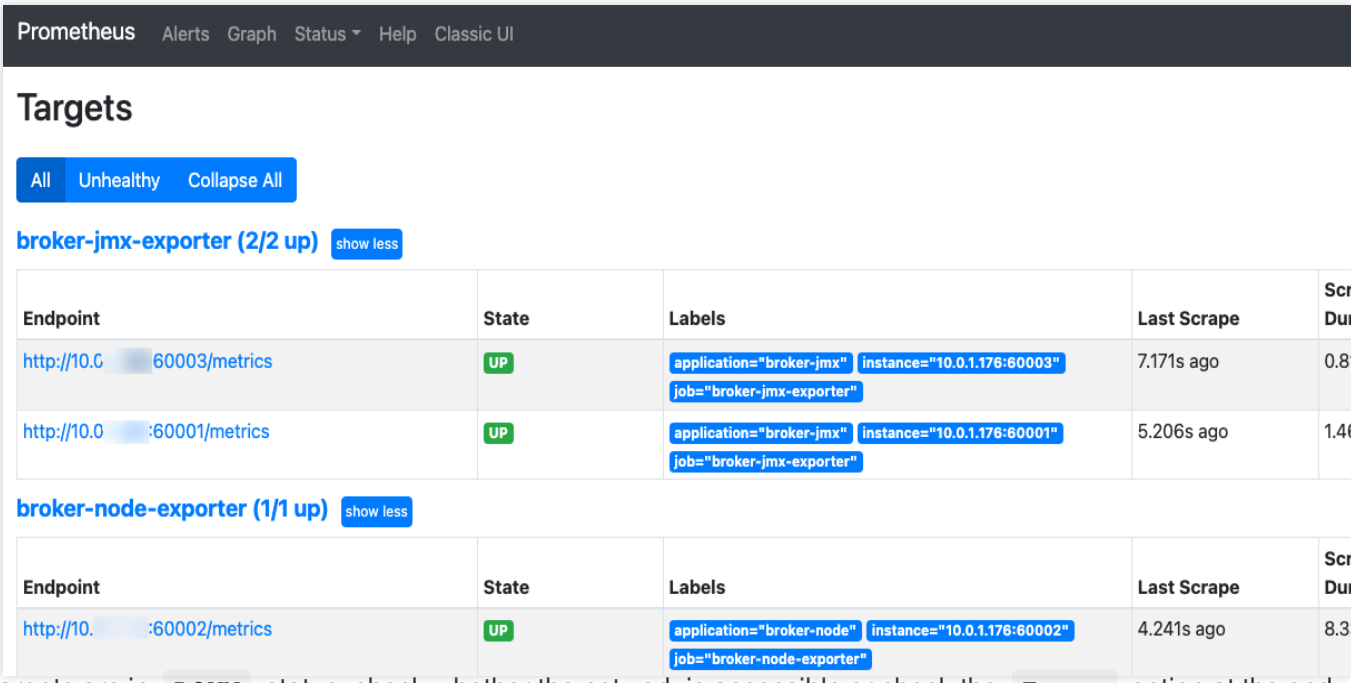
7.3 Start Prometheus.

```
./prometheus --config.file=prometheus.yml --web.enable-lifecycle
```

7.4 Open the UI provided by Prometheus to check whether the connected targets are normal by entering `http://localhost:9090` in the browser for example.



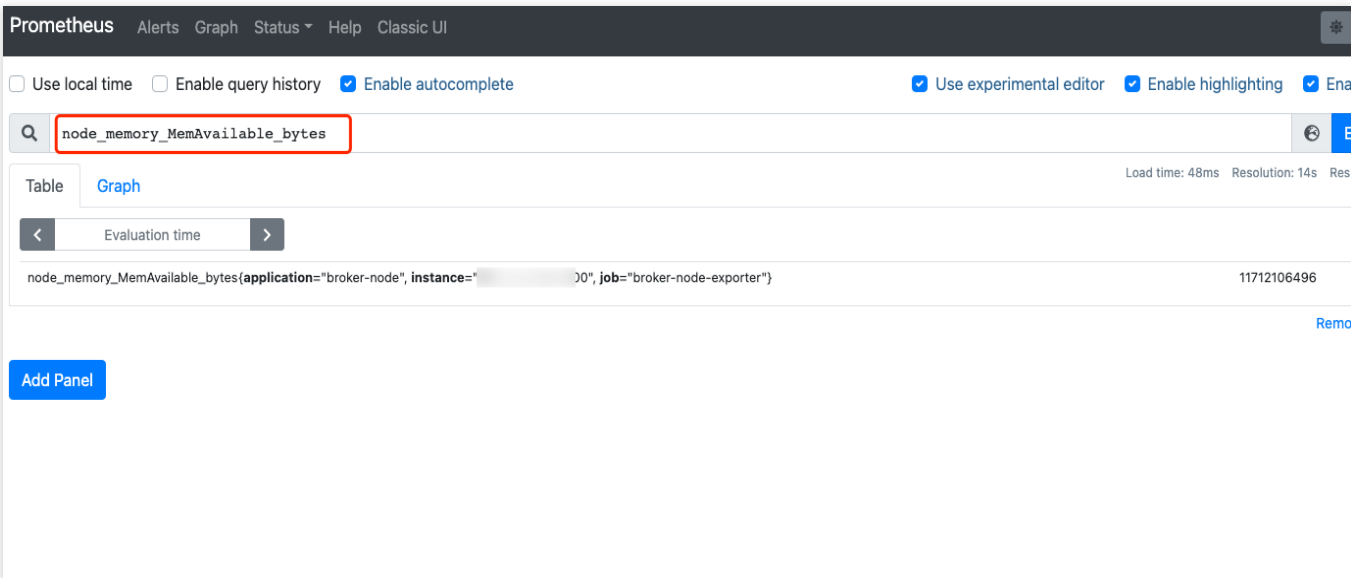
7.5 Confirm that all targets are in `UP` status.



If the targets are in `DOWN` status, check whether the network is accessible or check the `Error` option at the end of the status bar for the cause.

8. Query the monitoring metric data.

Click **Graph**, enter a metric name such as `node_memory_MemAvailable_bytes` , and click **Execute** to view the monitoring data.



AZ Migration

Last updated：2024-11-07 14:49:03

Overview

You can migrate a **CKafka Pro Edition** instance to another AZ in the same region. All its attributes, configurations, and connection address will stay unchanged after the migration. The time it takes to migrate the instance is subject to the instance's data volume.
For example, you can migrate to a new AZ in the following scenarios:
If you want to modify an instance's type, but the current AZ doesn't support the new instance type, you can migrate the instance to an AZ supporting the new type.
If the current AZ has no remaining resources for scaling, you can also migrate the instance to another AZ in the same region with sufficient resources to meet your business needs.

Prerequisites

The instance is running.
The region where the instance is located has multiple AZs to support cross-AZ migration.

Billing Description

This feature is free of charge. There is no charge even for migrating an instance from a single AZ to multiple AZs.

Description

If the original instance is deployed in a single AZ, you can switch the AZ or upgrade to multi-AZ deployment.
If the original instance is deployed in multiple AZs, you can switch AZs but cannot downgrade to single-AZ deployment.

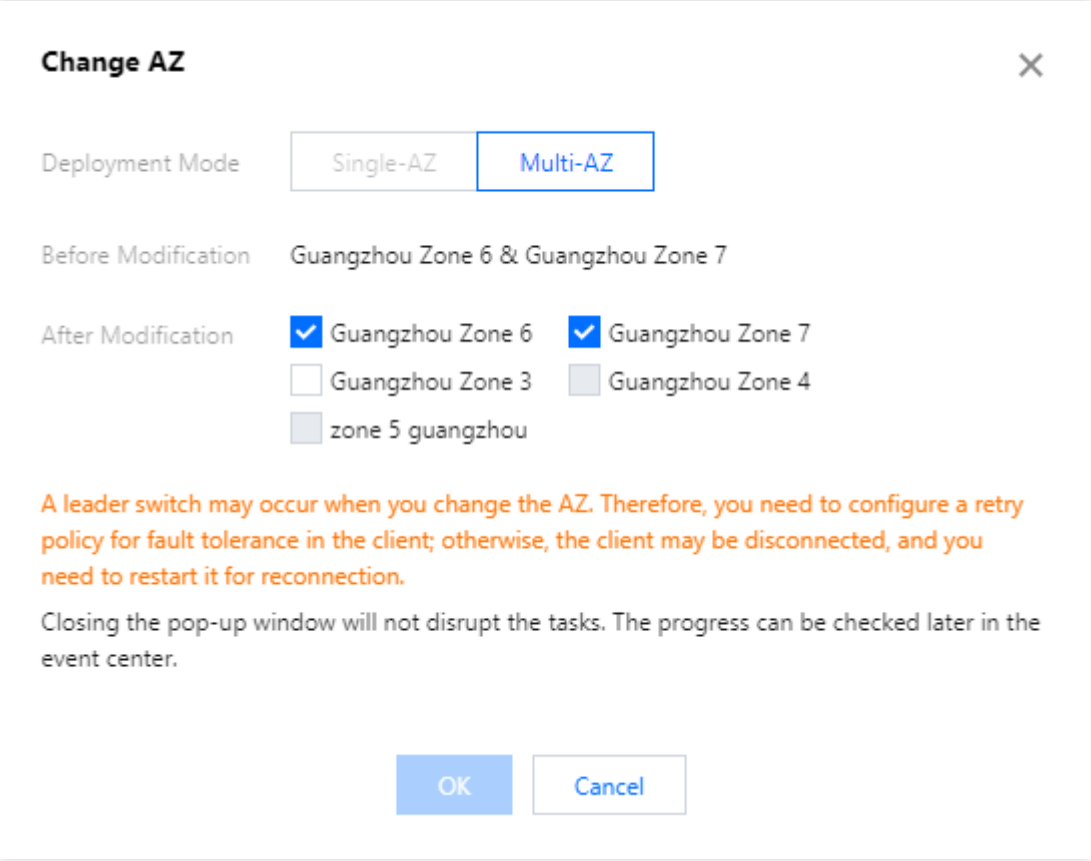
Migration Type

Migration Type	Applicable Scenario

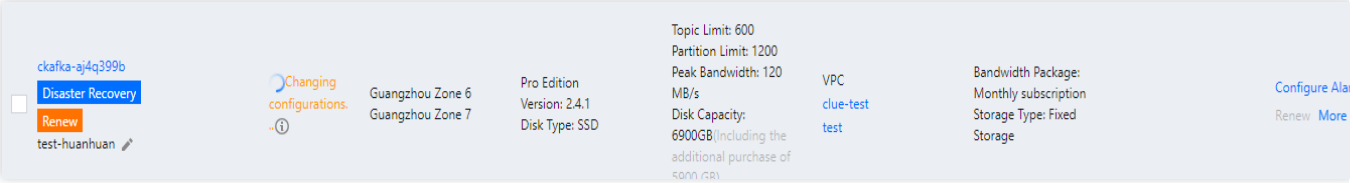
Migration from one AZ to another AZ	The AZ where the instance is located is under full load or other conditions that affect the instance performance.
Migration from one AZ to multiple AZs	You can improve the disaster recovery capability of the instance and implement cross-data center disaster recovery. Source and replica instances are located in different AZs. Multi-AZ instances can withstand higher levels of disasters than single-AZ instances. For example, the latter can tolerate server- and rack-level failures, while the former can tolerate data center-level failures.

Directions

1. Log in to the CKafka console.
2. Click **Instance List** on the left sidebar and click the **ID/Name** of the target instance to enter the basic information page.
3. In the **Basic Info** module, click **Edit** on the right of the current AZ and select the AZ you want to switch to.



4. Click **OK**. It will take 5–10 minutes to complete the configuration adjustment. You can view the progress in the **Status** column in the instance list.


ckafka-aj4q399b
Disaster Recovery
Renew
test-huanhuan
Changing configurations.
Guangzhou Zone 6
Guangzhou Zone 7
Pro Edition
Version: 2.4.1
Disk Type: SSD
Topic Limit: 600
Partition Limit: 1200
Peak Bandwidth: 120 MB/s
Disk Capacity: 6900GB(Including the additional purchase of
VPC
clue-test
test
Bandwidth Package: Monthly subscription
Storage Type: Fixed Storage
Configure Ala
Renew More

Setting Maintenance Time

Last updated：2024-01-09 14:47:32

Overview

The maintenance time is critical for CKafka. The backend system performs maintenance operations on your CKafka instance from time to time to ensure its stability. To minimize the potential impact on your business, we recommend that you set an acceptable maintenance period for your business instance, usually during off-peak hours.

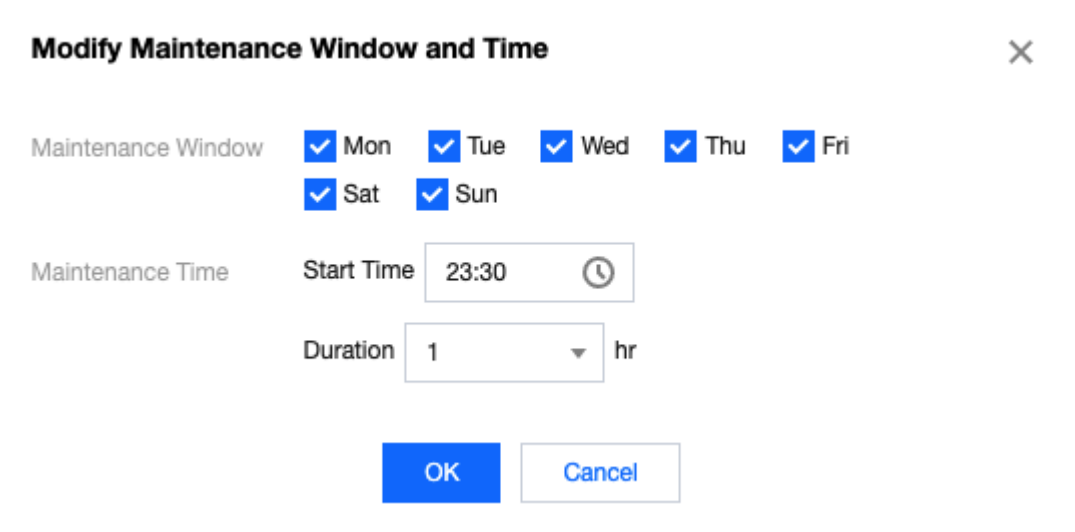
Note:

The default system maintenance time starts at 11:00 PM and lasts one hour. You can modify the time for CKafka Pro Edition instances but not for Standard Edition instances.

Before maintenance is carried out for CKafka, notifications will be sent to the recipients configured in your Tencent Cloud account by SMS and email.

Setting the maintenance time

1. Log in to the CKafka console.
2. Click **Instance List** on the left sidebar and then the **ID/Name** of your instance to enter the **Basic Info** page.
3. In the **Basic Info** module, click **Edit** next to the system maintenance time.
4. In the pop-up window, select **Maintenance Window** and **Maintenance Time**.



5. Click **OK**.

Setting Message Size

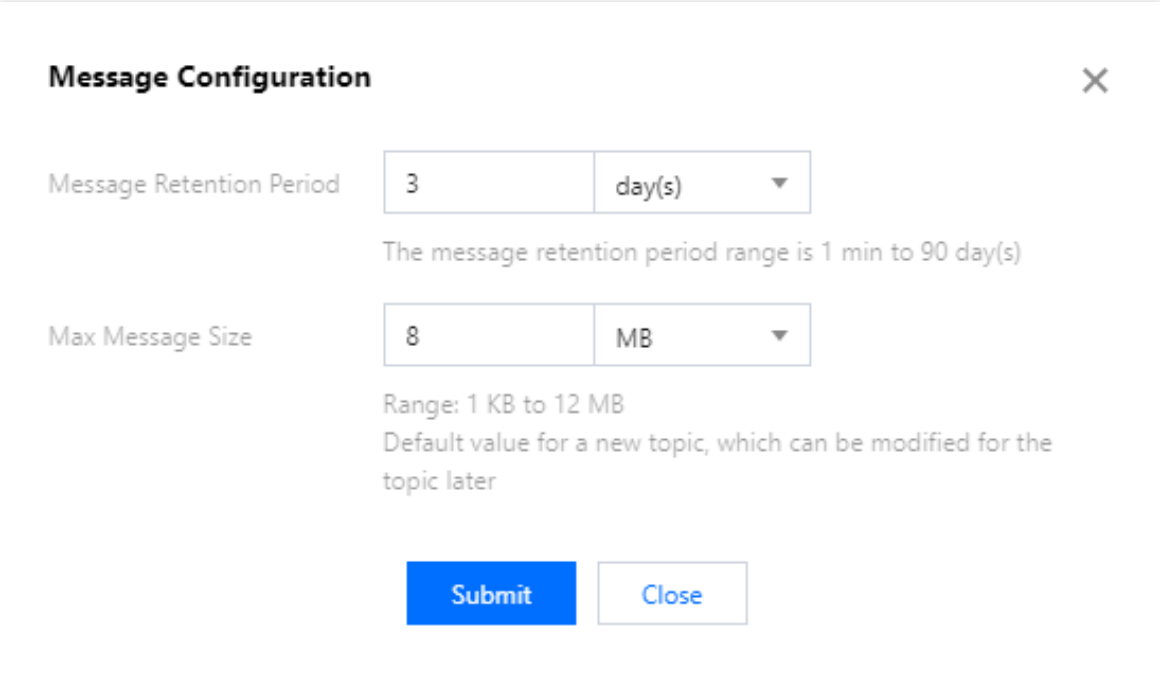
Last updated：2024-01-09 14:47:32

Operation Scenarios

CKafka supports instance-level configuration of the default message size, which serves as the default initial value when a new topic is created. It can later be modified for each topic.

Configuring Default Message Size

1. Log in to the CKafka console.
2. Click **Instance List** on the left navigation bar, and click **ID/Name** of the target instance to enter the instance details page.
3. In the instance details page, click **Modify Configuration** in the message configuration module.
4. After setting the message retention period and max message size, click Submit.



Setting Message Size for a Topic

1. Log in to the CKafka console.
2. Choose **Instance List** from the left sidebar, and click **ID/Name** of the target instance to enter the instance details page.
3. On the instance details page, click on the **Topic Management** tab and then click **Edit** in the operation bar.
4. On the Edit Topic page, click Show Advanced Configuration, and configure **max.message.bytes** to modify the maximum message size for the topic.

Topic Management
Instance Topic
Creating Topic

Last updated：2025-04-08 17:55:26

Overview

Topic is the name of a category where messages can be stored and published. CKafka uses the concept of topic externally, that is, a producer writes messages to a topic, while a consumer read messages from it. To implement horizontal scaling, a topic actually consists of multiple partitions. When a performance bottleneck occurs, you can scale out the topic by adding more partitions.
This document describes how to create a topic in the CKafka console.

Prerequisites

You have created an instance as instructed in Creating Instance.

Directions

Step 1. Create a topic

1. Log in to the CKafka console.
2. In the instance list, click the **ID/Name** of your instance to enter the instance details page.
3. On the instance details page, click **Topic Management** at the top and click **Create**.
4. In the **Create Topic** window, set the **Partition Count**, **Replica Count**, and other parameters.

Parameter	Description
Name	Topic name, cannot be changed after input. The name can only contain letters, numbers, underscores, hyphens, and periods. Does not support starting with double underscores.
Partition Count	A concept of physical partition. A Topic can contain one or more partitions. CKafka uses partitions as the allocation unit. The default deployment architecture has at least 3 nodes. It is recommended to start with at least 3 partitions for a more balanced data distribution. For partition count configuration, refer to the Parameter Configuration Instructions.
Number of	The number of replicas of a Partition is used to ensure high availability of the Partition. To

Replicas	ensure data reliability, 2 replicas are enabled by default. The number of replicas is also counted as the number of partitions. For example, if a customer creates 1 Topic, 6 partitions, and 2 replicas, then the total Partition quota used is 1 × 6 × 2 = 12. **Notes:** Setting it to a single replica cannot guarantee availability. Proceed with caution.
Tag	Tags are used to manage resources by category from different dimensions. For more details, see Tag Management.
retention.ms	Message retention time in the Topic dimension, ranging from 1 minute to 90 days.

5. (Optional) Click to expand advanced configuration and set the following parameters:

Parameter Name	Default Value	Parameter Range	Description
Message Timestamp Type	LogAppendTime (Recommended)	LogAppendTime/CreateTime	CreateTime indicates the local time of the client. Since the client time may deviate from the server time, you need to check whether the written time is the correct time. LogAppendTime indicates the time when the server-side Broker receives a message.
Predefined ACL Policy	Off	Enable / Disable	Check the preconfigured ACL policy. For details about the ACL policy, see Configure ACL Policy.
cleanup.policy	delete	delete/compact	Supports deleting logs by storage time or compressing logs by key (compact mode is required when using Kafka Connect).
min.insync.replicas	1	Cannot be larger than the secondary quantity.	When the producer sets request.required.acks

			to 1, min.insync.replicas specifies the minimum number of replicas.
unclean.leader.election.enable	true	true/false	Specifies whether it is possible to set replicas that are not in the ISR as leaders.
segment.ms	/	1 to 90 Days	The duration of fragment scrolling, in milliseconds.
retention.bytes	Defaults to the message retention size of the instance.	1 to 1,024 GB	number of partitions * rentention.bytes = message retention size of the current topic. For a topic, if both message retention time and message retention size are set, the actual message retention will be based on the threshold that is reached first.
max.message.bytes	-	1 KB to 12 MB	When sending data, the client aggregates data sent to the same partition and sends them in a unified manner. The server compares the message size of each batch.

6.Click **Submit** to complete Topic creation.

Viewing Topic

Last updated：2025-04-11 16:22:22

Overview

After creating a Topic, you can view relevant information about the Topic in the console, such as partition status, node distribution, production connection relationships, subscription relationships, etc., stay informed about the Topic status in real time, and handle anomalies promptly.

View Partition Status

During the business operation process, if anomalies such as message backlog or too high disk load occur in the CKafka cluster, you can view the partition status in the CKafka console to understand the message volume on the server and the consumption progress of each partition, and pinpoint and analyze issues.

1. Log in to the CKafka console.
2. Click **Instance List** on the left sidebar and then the **ID/Name** of your instance to enter the instance details page.
3. On the instance details page, click **Topic Management** to view the topic information and enter the topic list page.
4. On the topic list page, click the right triangle icon on the left of the topic name to view the topic details.

Item	Description
Partition Name	Partition name.
leader	The leader processes all read/write requests in the partition, and the follower passively and periodically replicates the data from the leader.
Replica	Replica list.
ISR	Replicas with synced messages.
Start Offset	The last position of message consumption.
End Offset	The last position of message write.
Messages	The number of stored messages.
Unsynced Replicas	The number of unsynced replicas. You can filter partitions with unsynced replicas.

©2013-2025 Tencent Cloud International Pte. Ltd.

Viewing Topic Node Distribution

On the Topic management page, click on the operation bar **More** > **Production Connection Relationship** to view the replication distribution of the Topic on the broker node.

Viewing Production Connection Relationship

Note:
Currently, you can view the producer connections only of Pro Edition instances on v2.4 or later.
On the Topic management page, click on the operation bar **More** > **Production Connection Relationship** to view the producer list information connected to the Topic.

View Subscription Relationship

On the Topic management page, click on the operation bar **More** > **Subscription Relationship** to view the consumer group information subscribing to the Topic, perform offset settings for the consumer group, and view consumer group details.

©2013-2025 Tencent Cloud International Pte. Ltd.

Deleting Topic

Last updated：2025-04-08 17:55:26

Overview

This document describes how to delete a topic in the CKafka console.

Directions

1. Log in to the CKafka console.
2. Select **Instance List** on the left sidebar and click the **ID/Name** of your instance to enter the instance details page.
3. On the instance details page, select **Topic Management** and click **Delete** in the **Operation** column.
4. In the window that pops up, click **Delete** to delete the topic.

Note:

Deleting a topic will also delete the messages stored in the topic. Do so with caution; when there are subscriptions under the current Topic, they will also be deleted synchronously.

Deleting a topic is an async operation. After the configuration is deleted successfully, the ZooKeeper configuration will take effect in 1 minute. If a topic with the same name is created during this period, the system will prompt an error code [4000]10011, and you can try again later.

©2013-2025 Tencent Cloud International Pte. Ltd.

Sending Message

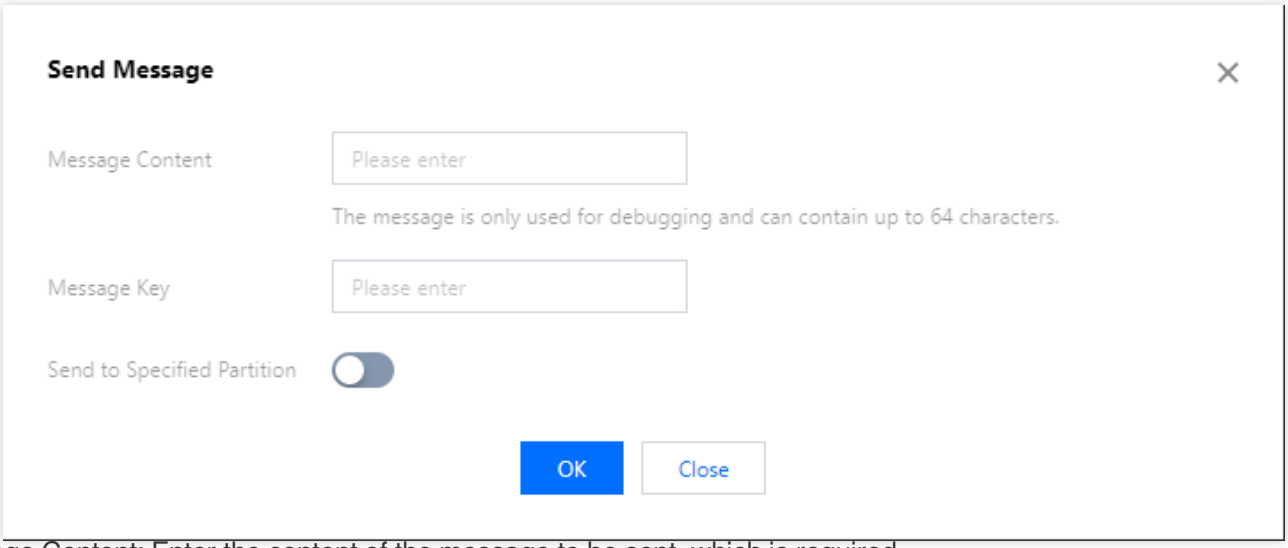
Last updated：2024-01-09 14:47:32

Overview

This document describes how to send a message and view its record in message query after creating a topic in the CKafka console.

Directions

1. Log in to the CKafka console.
2. Select **Instance List** on the left sidebar and click the **ID/Name** of your instance to enter the instance details page.
3. On the instance details page, select **Topic Management** and click **Send Message** in the **Operation** column.



Message Content: Enter the content of the message to be sent, which is required.
Message Key: Enter the sending key, which is optional.
Send to Specified Partition: This parameter supports sending messages to the specified partition, which is disabled by default.
4. Click **OK** to send the message. In the **Sent the message successfully** pop-up window, click **Message Query** to view the message just sent.

Setting Topic Traffic Throttling Rule

Last updated：2025-04-08 17:55:26

Overview

You can set topic traffic throttling rules to prevent the excessive traffic of one topic from affecting other topics.

Note:

Setting topic traffic throttling rules are only available to broker on versions 1.1.1, 2.4.1, and 2.8.1.

Directions

1. Log in to the CKafka console.
2. Select **Instance List** on the left sidebar and click the **ID/Name** of your instance to enter the instance details page.
3. On the instance details page, select **Topic Management**.
4. Click **More** > **Traffic Throttling** in the **Operation** column to set the traffic throttling threshold.

Maximum Topic Production Traffic: This value excludes replica traffic and ranges from 1 MB/s to the maximum bandwidth purchased for the instance / number of replicas of the topic.

Maximum Topic Consumption Traffic: This value ranges from 1 MB/s to the maximum bandwidth purchased for the instance.The underlying layer throttles the traffic for brokers, and the actual traffic throttling value (equal to an integer multiple of the number of brokers) may be slightly different from the set value.For soft traffic throttling mechanism, see Traffic Throttling.

©2013-2025 Tencent Cloud International Pte. Ltd.

Elastic Topic
Topic Management

Last updated：2024-01-09 14:47:31

Overview

Topic is a category name where messages are stored and published. Producers write messages to topics, and consumers read messages from topics. To enable horizontal scaling, a topic is divided into multiple partitions. This allows you to horizontally scale your resources by adding more partitions in case of performance bottlenecks. A topic is the smallest unit of message subscription and publishing in Connector. You can use a topic to denote a type of streaming data.
CKafka supports the **elastic topic** type. You can directly create an elastic topic in the console with no need to purchase a CKafka instance first. You can use the SDK to send and receive messages through the elastic topic. You can also use the elastic topic as the data source or target of a data task.
Note:
Currently, billing has not been started for elastic topics. We will inform you of the billing start date in advance.

Use Limits

A single elastic topic can have up to 500 partitions, and the elastic topic bandwidth has no default size. It is recommended that the bandwidth not exceed 500 Mbps.
The maximum message size supported by an elastic topic is 12 MB.

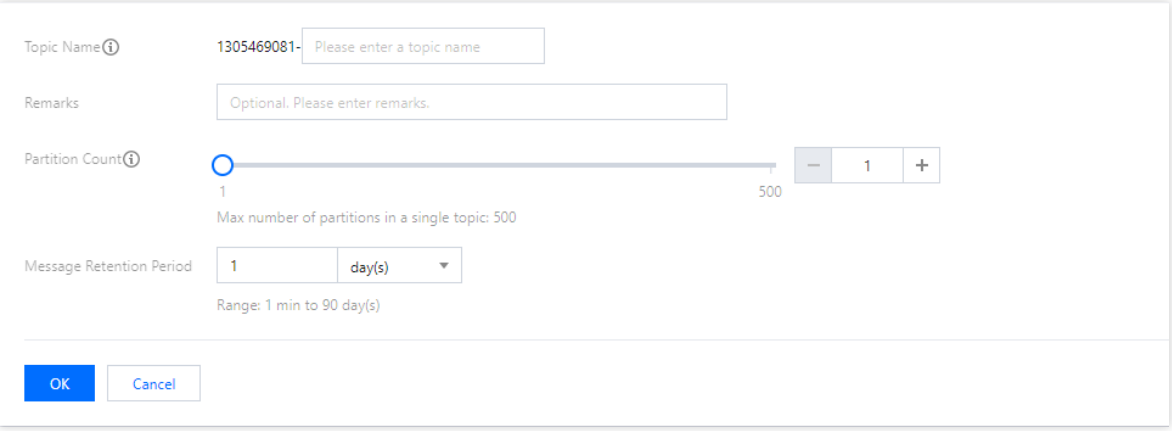
Directions

Creating a topic

1. Log in to the CKafka console.
2. Click **Elastic Topic** on the left sidebar, select the region, and click **Create Topic**.
3. Enter the topic information and click **OK**.

Topic Name: The topic name is in the format of "appid-[the name you enter]". It can only contain letters, digits, underscores, or symbols ("-" and ".").
Remarks: Enter the optional remarks of the topic.

Partition Count: It is a concept in physical partition, where one topic can contain one or more partitions. CKafka uses partition as an allocation unit. A single topic can have up to 500 partitions.
Message Retention Period: This value can range from 1 minute to 90 days. When the disk capacity is insufficient (that is, the disk utilization reaches 90%), old messages will be deleted in advance to ensure the service availability.



Querying a topic

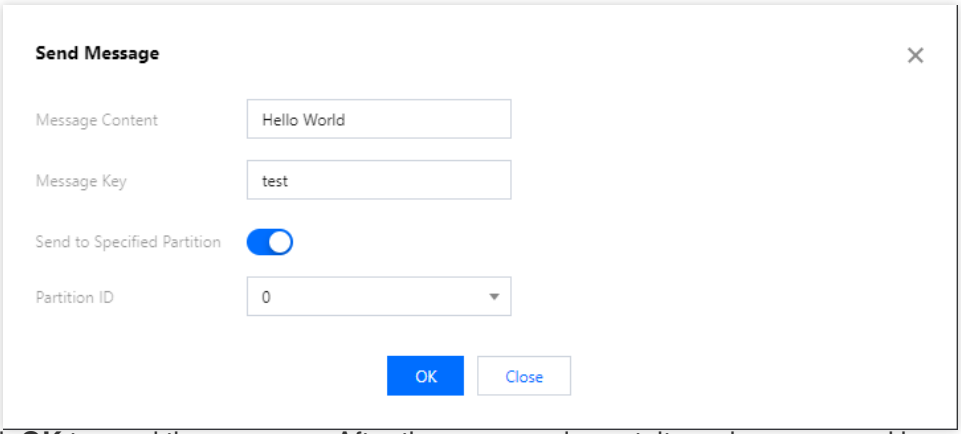
You can search for topics by topic name in the search box in the top-right corner of the Elastic Topic list. Connector will perform a fuzzy match and display the search results.

Editing a topic

1. In the elastic topic list, click **Edit** in the **Operation** column of the target topic.
2. In the pop-up window, you can edit the topic remarks, partition count, and message retention period.
3. Click **OK** to complete your edits.

Sending a message

You can manually send messages to the specified topic in the Ckafka console.
1. In the Elastic Topic list, click **Send Message** in the **Operation** column of the target topic.
2. Enter the message content and message key. You can also specify to send the message to the specified partition.

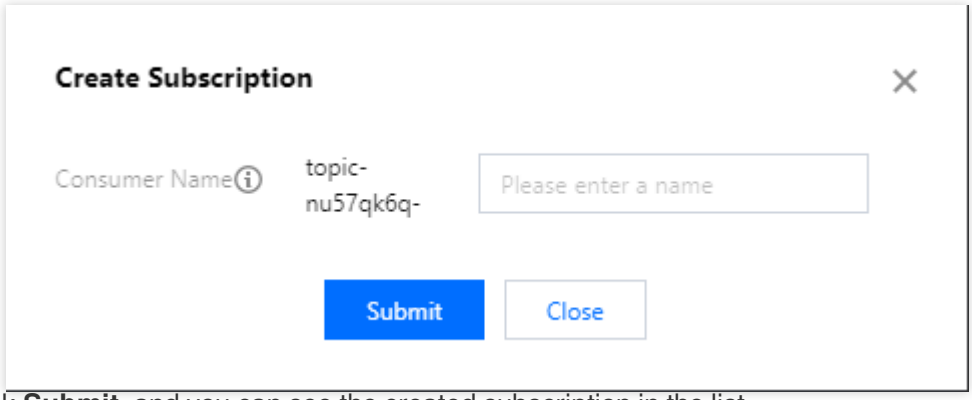


3. Click **OK** to send the message. After the message is sent, it can be consumed by any subscribers to the topic.

Adding a subscription

You can manually create a subscription in the console.

1. In the Elastic Topic, click the **ID** of the target topic to enter its basic information page.
2. Select the **Subscription** tab at the top, click **Create Subscription**, and enter the consumer name in the pop-up window.



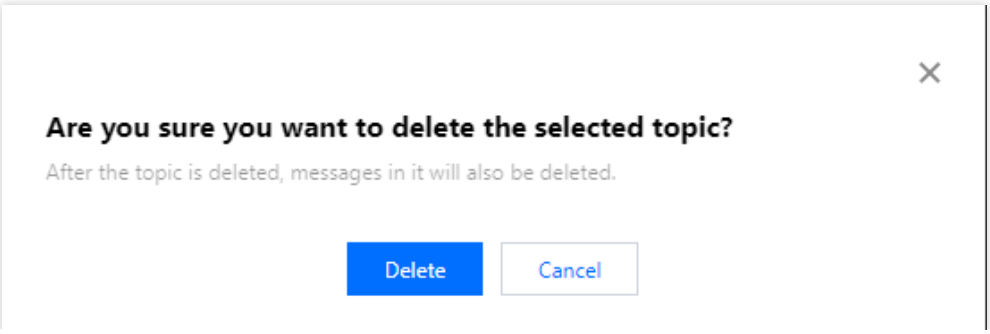
3. Click **Submit**, and you can see the created subscription in the list.

Deleting a topic

Note:

After a topic is deleted, all unconsumed messages retained in it will be cleared; therefore, proceed with caution.

1. In the Elastic Topic list, click **Delete** in the **Operation** column of the target topic.
2. In the pop-up window, click **Delete**.



Subscription Management

Last updated：2025-04-08 18:04:20

Overview

A subscription represents a specific consumer and its subscription to a topic. A consumer can consume all messages in a topic after subscribing to it. CKafka's subscription service provides the feature of saving the consumption offset on the server. This document describes how to add a subscription and reset the consumption offset in the CKafka console.

Prerequisites

You have created a topic as instructed in Creating Topic.

Directions

Adding a subscription

Add a subscription as instructed in Instance Topic > Creating Topic.

Setting an offset

In scenarios such as offline data processing, sometimes it is necessary to reset the offset to consume message history.

1. In the Topic List, click the target topic ID to enter the topic's **Basic Info** page.
2. Select the **Subscription** tab at the top and click **Offset Settings** in the **Operation** column of the target subscription.
3. In the **Offset Settings** window, select **Topic** or **Partition** as the dimension of settings and click **Next**.

Set based on Topic: It is suitable for scenarios such as batch resetting of consumption offsets and setting consumption offsets according to businesses.

Set based on Partition: It is suitable for setting consumption offset for each partition after knowing the consumption situation of each Partition.

4. Select the topic or partition for which to reset the offset (if no topic is selected, the offset will be reset for all topics by default) and click **Next**.
5. Specify the offset.

©2013-2025 Tencent Cloud International Pte. Ltd.

Note：
The offset value should be between the minimum offset and the maximum offset. If it is configured to be smaller than the minimum offset, the consumption will start from the minimum offset; if it is larger than the maximum offset, the consumption will start from the maximum offset.
Make sure that there are no consumers in a consumer group before resetting the group.

Viewing consumer details

On the subscription list page, click **View Consumer Details** in the **Operation** column to view the information of consumers in the consumer group and the relationship between a consumer and the topic subscribed to by the consumer.
On the subscription list page, click the small triangle on the left of a consumer group name to display the information of the topic subscribed to by the consumer group, including topic name, number of partitions, submitted offset position, maximum offset position, and number of unconsumed messages.
Click **View Details** in the **Operation** column to view offset consumption at the partition level.
Note：
As the offset information is maintained on the consumer side, the offset position is subject to the way the consumer submits the offset. It is displayed asynchronously and does not necessarily represent real-time consumption conditions.

Deleting a subscription

In the subscription list, click **Delete** in the **Operation** column of the target consumer and click **OK** to delete the consumer.
Note：
After you delete the consumer group, when the consumers in it re-establish connections for consumption, the offset will be reset, and consumption will start from the beginning.

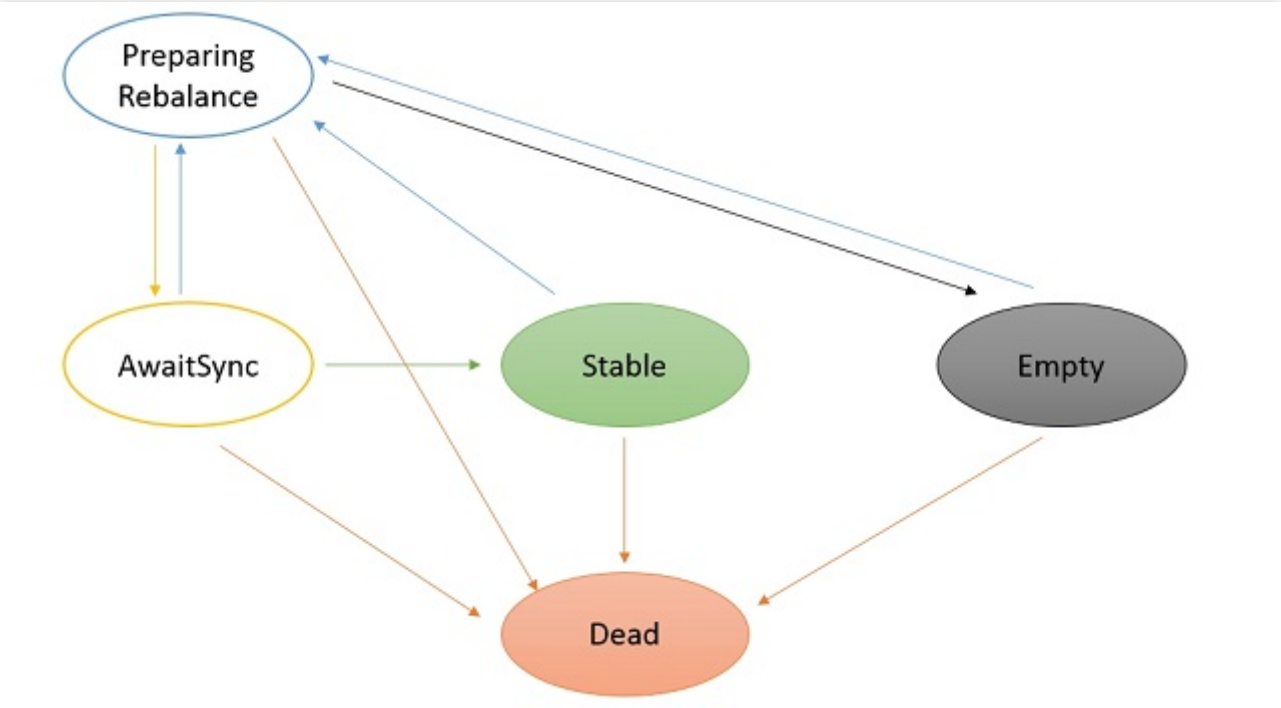
©2013-2025 Tencent Cloud International Pte. Ltd.

Consumer Group
Consumer Group Overview

Last updated：2024-01-09 14:47:32

Consumer Group Status Description

A consumer group can be in Dead, Empty, Preparing Rebalance, AwaitingSync, or Stable state. Most common states are Empty, Stable, and Dead. The transformation of the consumer group's state machine is as shown below:



Dead: consumer group does not have any members and metadata has been removed.
Empty: consumer group currently does not have any members. If all offsets in the group have expired, the group's state will be Dead. Generally, a newly created group is in Empty state by default.
As required by the open-source Kafka v0.10.x, **a consumer group will be automatically deleted if it does not have any members for more than 7 days.**
Stable: every consumer in the consumer group has joined and the state is stable.

Rebalance State Details

Cause of rebalance

According to the consumer group's state machine, the group may perform rebalance when the consumer group is in Empty, AwaitingSync, or Stable state. Rebalance may occur in the following scenarios:

A consumer subscribes to a topic.

A consumer is closed.

A consumer is considered to be in Dead state by the group coordinator.

If a consumer does not send a heartbeat to the group coordinator within the `session.timeout.ms` time period, the consumer will be considered to be in Dead state and rebalance will be initiated. For more information, please see CKafka Common Parameter Configuration Guide.

The number of partitions has increased.

A topic that does not exist is subscribed to.

If you subscribe to a topic that has not been created yet, rebalance will occur when the topic is created. Similarly, if a topic that has already been subscribed to is deleted, rebalance will also occur.

Application crashes.

Rebalance process analysis

Taking the mechanism of Kafka v0.10 as an example, the rebalance process is analyzed as follows:

1. A consumer that wants to join a consumer group will send a JoinGroup request to the group coordinator. The first consumer to join a group will become the group leader.
2. The leader will receive a list of all consumers in the group from the coordinator and will be responsible for assigning partitions to consumers in the group. Partition assignment can be implemented through the PartitionAssignor API.
3. When the assignment is done, the leader will send the assignment result to the coordinator. The coordinator will then send the result to all consumers.

Each consumer can only view the partition assigned to it. The leader is the only consumer that can get the information of consumers in the consumer group and their partitions.

The above process will be executed once whenever rebalance occurs.

©2013-2025 Tencent Cloud International Pte. Ltd.

Creating Consumer Group

Last updated：2025-04-08 18:06:10

Overview

Consumer Group is an expandable and fault-tolerant Consumer mechanism provided by Kafka. There are multiple consumers in a Consumer Group. All consumers in the Group coordinate together to consume messages of subscribed topics.
CKafka supports direct creation of Consumer Groups on the console. This article introduces the specific steps to create a Consumer Group on the console.
Note:
We recommend that you limit the number of consumer groups in each instance to 500 or less, since excess consumer groups will result in certain constraints.

Directions

Creating a consumer group

1. Log in to the CKafka console.
2. Click **Instance List** on the left sidebar and then the **ID/Name** of your instance to enter the instance details page.
3. On the instance details page, select the **Consumer Group** tab and click **Create Consumer Group**.

4. In the pop-up window, enter the consumer group name and select the topic to which you want to subscribe. Note that the consumer group name cannot contain quotation marks ("").
Note:
You can select multiple topics.
5. Click **Submit**, and you can see the created consumer group in the consumer group list.

Disabling automatic consumer group creation

CKafka allows you to disable automatic consumer group creation in the console. After it is disabled, only existing consumer groups in the console can be used for consumption, while new data sync tasks cannot be created normally.

Note:
Only Pro Edition 2.4.1 and later are supported.

©2013-2025 Tencent Cloud International Pte. Ltd.

Querying Consumer Group

Last updated：2025-04-08 18:06:10

Overview

This document describes how to view the consumer group information of an instance in the CKafka console.

Directions

1. Log in to the CKafka console.
2. Click **Instance List** on the left sidebar and then the **ID/Name** of your instance to enter the instance details page.
3. On the instance details page, click the **Consumer Group** tab to view the consumer group information of the current CKafka instance.

On the **Consumer Group** list page, click **View Consumer Details** in the **Operation** column to view the information of consumers in the consumer group and the relationship between a consumer and the topic subscribed to by the consumer.
On the **Consumer Group** list page, click the small triangle on the left of a consumer group name to display the information of the topic subscribed to by the consumer group, including topic name, number of partitions, submitted offset position, maximum offset position, and number of unconsumed messages. You can click **View Details** in the **Operation** column to view offset consumption at the partition level.
Note:
As the offset information is maintained on the consumer side, the offset position is subject to the way the consumer submits the offset. It is displayed asynchronously and does not necessarily represent real-time consumption conditions.

©2013-2025 Tencent Cloud International Pte. Ltd.

Deleting Consumer Group

Last updated：2024-11-07 14:49:02

Overview

In some cases, a consumer group may not consume for a long time before consuming again. You can delete it, so that when the consumers in it establish a connection again, the offset will be reset, and consumption will start from the beginning.

Deleting a consumer group

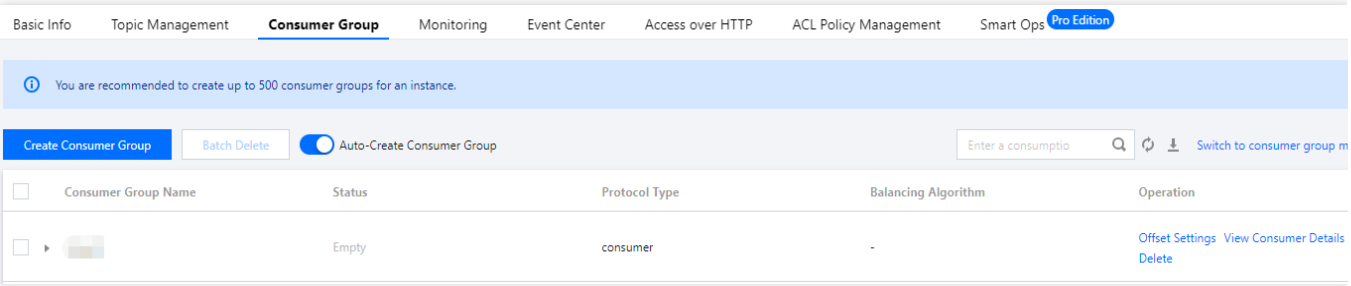
Note:

Only empty consumer groups with a broker version not below v1.1.1 can be deleted.

1. Log in to the CKafka console.
2. Click **Instance List** on the left sidebar and then the **ID/Name** of your instance to enter the instance details page.
3. On the instance details page, click **Delete** in the **Operation** column of the consumer group to be deleted.

To delete a single Consumer Group: Click **Delete** in the operation bar of the target Consumer Group to delete that Consumer Group directly.

To batch delete multiple Consumer Groups: Select all Consumer Groups you wish to delete, then click **Batch Deletion** at the top to delete the selected Consumer Groups in batch.

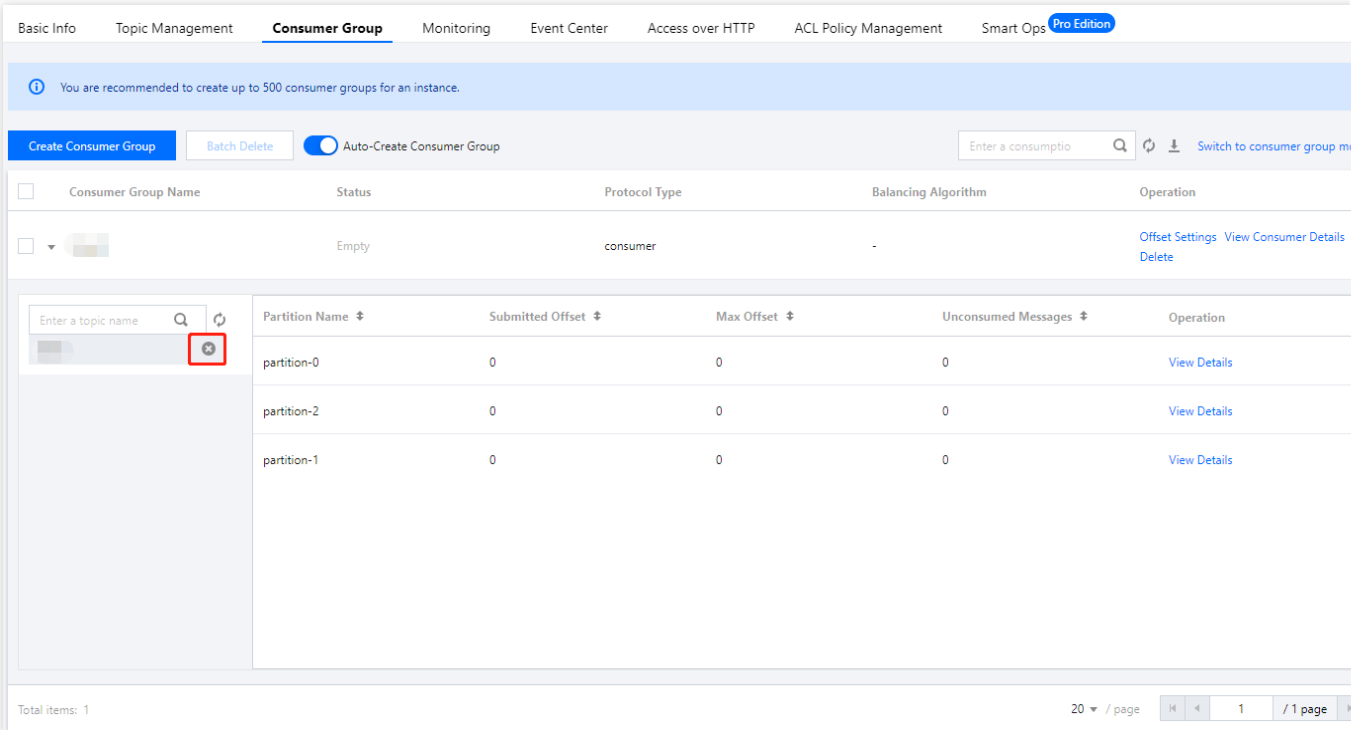


Deleting a subscription to an associated topic

If the status of a consumer group is **Empty**, the subscriptions of a topic associated with it can be deleted.

On the **Consumer Group** list page, click the

icon next to the subscription to be deleted and click **OK** to delete it.


Basic Info
Topic Management
Consumer Group
Monitoring
Event Center
Access over HTTP
ACL Policy Management
Smart Ops
Pro Edition
You are recommended to create up to 500 consumer groups for an instance.
Create Consumer Group
Batch Delete
Auto-Create Consumer Group
Enter a consumptio
Switch to consumer group m
Consumer Group Name
Status
Protocol Type
Balancing Algorithm
Operation
Empty
consumer
-
Offset Settings
View Consumer Details
Delete
Enter a topic name
Partition Name
Submitted Offset
Max Offset
Unconsumed Messages
Operation
partition-0
0
0
0
View Details
partition-2
0
0
0
View Details
partition-1
0
0
0
View Details
Total items: 1
20 / page
1
/ 1 page

Setting Offset

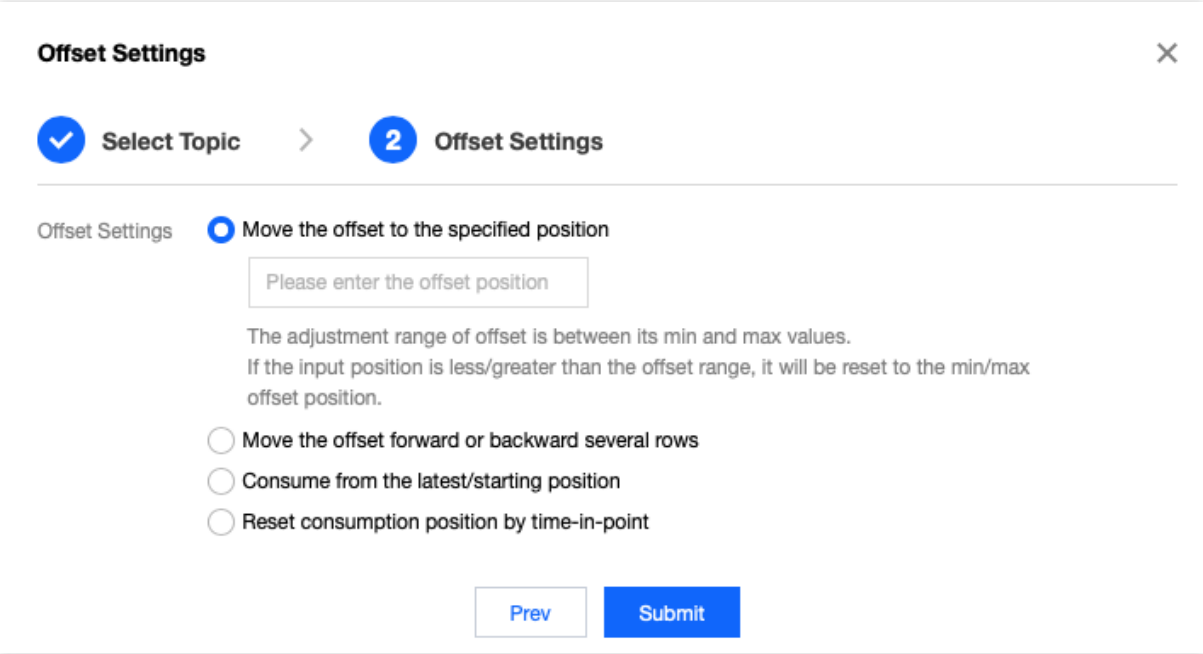
Last updated：2024-01-09 14:47:32

Overview

In scenarios such as offline data processing, sometimes it is necessary to reset the offset to consume earlier messages.

Directions

1. Log in to the CKafka console.
2. Click **Instance List** on the left sidebar and click the **ID/Name** of the target instance to enter the instance details page.
3. On the instance details page, select **Consumer Group** and click **Offset Settings** in the **Operation** column.
4. In the **Offset Settings** window, select **Topic** or **Partition** as the dimension of settings and click **Next**.
5. Select the topic or partition for which to reset the offset (if no topic is selected, the offset will be reset for all topics by default) and click **Next**.
6. Specify the offset.



Note:

The offset value should be between the minimum offset and the maximum offset. If it is configured to be smaller than the minimum offset, the consumption will start from the minimum offset; if it is larger than the maximum offset, the consumption will start from the maximum offset.

Make sure that there are no consumers in a consumer group before resetting the group.

©2013-2025 Tencent Cloud International Pte. Ltd.

Automatically Resetting Offset

Last updated：2024-01-09 14:47:32

This document describes the concept and usage of the `auto.offset.reset` parameter.

What is `auto.offset.reset`?

The `auto.offset.reset` parameter defines where to start consumption if the offset of the partition to be consumed cannot be obtained. For example, it specifies how the offset will be initialized if no offset is configured for the broker (such as upon initial consumption or when the offset expired for more than seven days) or how it will be reset if the error OFFSET_OUT_OF_RANGE occurs.
The `auto.offset.reset` parameter has the following valid values:
`earliest` : Reset to the minimum offset in the partition.
`latest` : Reset to the maximum offset in the partition. This is the default value.
`none` : Report an `OffsetOutOfRangeException` exception without resetting the offset.

When will `OFFSET_OUT_OF_RANGE` occur?

This error indicates that the offset committed by the client is out of the offset range allowed by the broker. For example, if `LogStartOffset` and `LogEndOffset` of partition 1 in `topicA` are 100 and 300 respectively, but the offset committed by the client is less than 100 or greater than 300, then the broker will return this error, and the offset will be reset.
This error may occur in the following cases:
If an offset is set on the client but not used for consumption during a period of time, it will be deleted on the broker due to log scrolling after the message retention period set for the topic has elapsed. In this case, if it is recommitted by the client after the deletion, this error will occur.
If the client commits an abnormal offset due to issues such as SDK bugs and network packet loss, this error will occur.
If there are unsynced replicas on the broker and the leader is switched, follower replicas will be truncated. In this case, if the offset committed by the client falls in the truncated range, this error will occur. For more information on leader switch, see Client > What is leader switch?.

`auto.offset.reset=none` Description

Background

You don't want the offset to be automatically reset, as your business doesn't allow such large-scale repeated consumption.

Note:

In this case, the consumer group will report an error for failing to find the offset in its first consumption. Therefore, you need to manually set the offset in `catch` .

Note

After `auto.offset.reset` is set to `none` , automatic offset reset can be avoided; however, as the automatic reset mechanism is disabled, when a new partition is added, the client doesn't know where to start consuming the new partition, and an exception will occur. In this case, you need to manually set a consumer group offset and start consuming.

Instructions

During consumption, if you set `auto.offset.reset` to `none` for the consumer, you need to capture the exception `NoOffsetForPartitionException` and set the offset in `catch` on your own. You can select one of the following methods based on your actual business needs:

Specify the offset. You need to maintain the offset, which is convenient for retries.

Specify to start consumption from the beginning.

Use the nearest available offset.

Obtain and set the offset based on the timestamp.

Below is the sample code:

```
package com.tencent.tcb.operation.ckafka.plain;

import com.google.common.collect.Lists;
import com.tencent.tcb.operation.ckafka.JavaKafkaConfigurer;
import java.time.Instant;
import java.time.temporal.ChronoUnit;
import java.util.ArrayList;
import java.util.Collection;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
import java.util.Map.Entry;
import java.util.Properties;
import org.apache.kafka.clients.CommonClientConfigs;
import org.apache.kafka.clients.consumer.ConsumerConfig;
import org.apache.kafka.clients.consumer.ConsumerRecord;
import org.apache.kafka.clients.consumer.ConsumerRecords;
import org.apache.kafka.clients.consumer.KafkaConsumer;
import org.apache.kafka.clients.consumer.NoOffsetForPartitionException;
import org.apache.kafka.clients.consumer.OffsetAndTimestamp;
```

```
import org.apache.kafka.clients.producer.ProducerConfig;
import org.apache.kafka.common.PartitionInfo;
import org.apache.kafka.common.TopicPartition;
import org.apache.kafka.common.config.SaslConfigs;

public class KafkaPlainConsumerDemo {

    public static void main(String args[]) {
        // Set the path of the JAAS configuration file.
        JavaKafkaConfigurer.configureSaslPlain();

        // Load `kafka.properties`.
        Properties kafkaProperties = JavaKafkaConfigurer.getKafkaProperties();

        Properties props = new Properties();
        // Set the access point. Obtain the access point of the corresponding topic
        props.put(ProducerConfig.BOOTSTRAP_SERVERS_CONFIG, kafkaProperties.getPrope

        // Set the access protocol.
        props.put(CommonClientConfigs.SECURITY_PROTOCOL_CONFIG, "SASL_PLAINTEXT");
        // Set the PLAIN mechanism.
        props.put(SaslConfigs.SASL_MECHANISM, "PLAIN");
        // Set the maximum interval between two polls.
        // If the consumer does not return a heartbeat message within the interval,
        props.put(ConsumerConfig.SESSION_TIMEOUT_MS_CONFIG, 30000);
        // Set the maximum number of messages that can be polled at a time.
        // Do not set this parameter to an excessively large value. If polled messa
        props.put(ConsumerConfig.MAX_POLL_RECORDS_CONFIG, 30);
        // Set the method for deserializing messages.
        props.put(ConsumerConfig.KEY_DESERIALIZER_CLASS_CONFIG,
                "org.apache.kafka.common.serialization.StringDeserializer");
        props.put(ConsumerConfig.VALUE_DESERIALIZER_CLASS_CONFIG,
                "org.apache.kafka.common.serialization.StringDeserializer");
        // Set the consumer group for the current consumer instance. You need to ap
        // The instances in the same consumer group consume messages in load balanc
        props.put(ConsumerConfig.GROUP_ID_CONFIG, kafkaProperties.getProperty("grou

        // Consumption offset Note If `auto.offset.reset` is set to `none`, the con
        props.put(ConsumerConfig.AUTO_OFFSET_RESET_CONFIG, "none");
        // Construct a consumer object. This generates a consumer instance.
        KafkaConsumer<String, String> consumer = new KafkaConsumer<String, String>(
        // Set one or more topics to which the consumer group subscribes.
        // We recommend that you configure consumer instances with the same `GROUP_
        List<String> subscribedTopics = new ArrayList<String>();
        // If you want to subscribe to multiple topics, add the topics here.
        // You must create the topics in the console in advance.
        String topicStr = kafkaProperties.getProperty("topic");
```

```
        String[] topics = topicStr.split(",");
        for (String topic : topics) {
            subscribedTopics.add(topic.trim());
        }
        consumer.subscribe(subscribedTopics);
        // Consume messages in loop.
        while (true){
            try {
                ConsumerRecords<String, String> records = consumer.poll(1000);
                // All messages must be consumed before the next poll, and the tota
                for (ConsumerRecord<String, String> record : records) {
                    System.out.println(
                            String.format("Consume partition:%d offset:%d", record.
                }
            } catch (NoOffsetForPartitionException e) {
                System.out.println(e.getMessage());

                // If you set `auto.offset.reset` to `none`, you need to capture th
                // Sample 1. Specify the offset. You need to maintain the offset, w
                Map<Integer, Long> partitionBeginOffsetMap = getPartitionOffset(con
                Map<Integer, Long> partitionEndOffsetMap = getPartitionOffset(consu
                consumer.seek(new TopicPartition(topicStr, 0), 0);

                // Sample 2. Specify to start consumption from the beginning
                consumer.seekToBeginning(Lists.newArrayList(new TopicPartition(topi

                // Sample 3. Use the nearest available offset
                consumer.seekToEnd(Lists.newArrayList(new TopicPartition(topicStr,

                // Sample 4. Obtain and set the offset based on the timestamp. For
                Map<TopicPartition, Long> timestampsToSearch = new HashMap<>();
                Long value = Instant.now().minus(300, ChronoUnit.SECONDS).toEpochMi
                timestampsToSearch.put(new TopicPartition(topicStr, 0), value);
                Map<TopicPartition, OffsetAndTimestamp> topicPartitionOffsetAndTime
                        .offsetsForTimes(timestampsToSearch);
                for (Entry<TopicPartition, OffsetAndTimestamp> entry : topicPartiti
                        .entrySet()) {
                    TopicPartition topicPartition = entry.getKey();
                    OffsetAndTimestamp entryValue = entry.getValue();
                    consumer.seek(topicPartition, entryValue.offset()); // Specify
                }


            }
        }
    }
```

```
    /**
     * Get the earliest and nearest offsets of the topic
     * @param consumer
     * @param topicStr
     * @param beginOrEnd true begin; false end
     * @return
     */
    private static Map<Integer, Long> getPartitionOffset(KafkaConsumer<String, Stri
            boolean beginOrEnd) {
        Collection<PartitionInfo> partitionInfos = consumer.partitionsFor(topicStr)
        List<TopicPartition> tp = new ArrayList<>();
        Map<Integer, Long> map = new HashMap<>();
        partitionInfos.forEach(str -> tp.add(new TopicPartition(topicStr, str.parti
        Map<TopicPartition, Long> topicPartitionLongMap;
        if (beginOrEnd) {
            topicPartitionLongMap = consumer.beginningOffsets(tp);
        } else  {
            topicPartitionLongMap = consumer.endOffsets(tp);
        }
        topicPartitionLongMap.forEach((key, beginOffset) -> {
            int partition = key.partition();
            map.put(partition, beginOffset);
        });
        return map;
    }

}
```

Monitoring and Alarms
Viewing Monitoring Data

Last updated：2025-04-09 11:17:09

Overview

CKafka allows you to monitor the resources created under your account, such as instances, topics, and consumer groups, so that you can keep track of the status of your resources in real time and troubleshoot possible issues to ensure stable business operations.
This document describes how to view monitoring metrics and their descriptions in the CKafka console.

Monitoring Metrics Meaning and Descriptions

The following are the descriptions of CKafka monitoring metrics. For the metrics marked with the word **Recommended** in the remarks, it is recommended to configure monitoring and alerting policies for them based on user feedback and statistics. Please configure monitoring and alerting policies reasonably according to the actual situation.
Instance
Topic
Consumer Group

Monitoring Metrics	Unit	Description
Maximum production traffic of an instance **(recommended)**	MB	The peak message bandwidth for the production of a single replica of an instance, excluding the bandwidth for replica production (reference for calculating the usage percentage of instance production bandwidth), take the maximum value in unit time.
Instance production traffic	MB	Production traffic of the instance (excluding traffic generated by replicas), summed according to the selected time granularity.
Instance Production Bandwidth Percentage	%	The percentage of instance production bandwidth in the quota, calculate the average in unit time.
Maximum consumption traffic of the instance **(recommended)**	MB	The peak bandwidth of instance consumption messages. There is no concept of replica when consuming. Take the maximum value in unit time. (Reference for calculating the usage percentage of instance consumption bandwidth).

©2013-2025 Tencent Cloud International Pte. Ltd.

Instance consumption traffic	MB	Instance consumption traffic (excluding traffic generated by replicas), summed according to the selected time granularity.
Instance Consumption Bandwidth Percentage	%	The percentage of instance consumption bandwidth in the quota, calculate the average in unit time.
Instance Production Message Count	Count	Number of messages produced by the instance, summed according to the selected time granularity.
Number of messages consumed by the instance	Count	Number of messages consumed by the instance, summed according to the selected time granularity.
Total number of messages written to disk by the instance	Count	Total number of messages written to disk by the instance (excluding replicas), taking the latest value in the selected time granularity.
Cluster message request rate	Count/s	The situation of cluster instance message request rate, including production, consumption, metadata, and Offset submission, calculating the average in unit time.
Cluster TPS details	Count/s	The change in cluster TPS, calculating the average in unit time.
Instance disk usage	MB	The disk usage of the instance (including replicas), taking the latest value in the selected time granularity.
Disk usage percentage **(recommended)**	%	The percentage of the current disk occupancy to the total disk capacity of the instance specification, calculate the average in unit time.
Instance production throttling times	Count	Sum of instance production throttling times according to the selected time granularity.
Instance consumption throttling times	Count	Sum of instance consumption throttling times according to the selected time granularity.
Number of instance connections **(recommended)**	Count	Sum of the number of client connections to the server, summed according to the selected time granularity.

Topic Monitoring

Monitoring Metrics	Description
Maximum Production Traffic (MB/s)	Topic maximum production traffic (excluding replica traffic).
Maximum Consumption Traffic (MB/s)	Topic maximum consumption traffic.

©2013-2025 Tencent Cloud International Pte. Ltd.

Number of consumed messages	The actual number of consumed messages of the topic is summed according to the selected time granularity.
Consumption Traffic (MB)	The actual consumption traffic of the topic (excluding the traffic generated by the replica) is summed according to the selected time granularity.
Total amount of messages occupying disk (MB) **(recommended)**	The total amount of messages occupied by the topic on disk (excluding replicas), with the latest value taken according to the selected time granularity.
Number of produced messages	The actual number of messages produced by the topic is summed according to the selected time granularity.
Production Traffic (MB)	The actual production traffic of the topic (excluding the traffic generated by replicas) is summed according to the selected time granularity.
Total number of messages written to disk (Count)	The total number of messages written to the topic (excluding replicas), with the latest value taken according to the selected time granularity.
topic consumption throttling Count (Count/s)	Sum of Topic consumption throttling counts according to the selected time granularity.
topic production throttling Count (Count/s)	Summed Topic production throttling counts according to the selected time granularity.

Topic-Partition Monitoring

Monitoring Metrics	Description
Consumed Message Count (Count)	Number of messages consumed in the Partition, which is the total number of messages in the selected time period.
Consumption Traffic (MB)	Actual consumption traffic of the Partition (excluding traffic generated by replicas), summed according to the selected time granularity.
Total number of messages written to disk (Count)	The total number of messages actually written to the disk in the Partition (excluding those generated by replicas), which is the latest value in the selected time granularity.
Total amount of messages occupying disk (MB)	The total amount of messages actually occupying the disk in the Partition (excluding replicas), which is the latest value in the selected time granularity.
Number of produced messages (Count)	The total number of messages actually produced in the Partition, summed according to the selected time granularity.
Production Traffic (MB)	The actual production traffic of the Partition (excluding traffic generated by replicas),

	summed according to the selected time granularity.

Consumer Group-Topic Monitoring

Monitoring Metrics	Description
Topic maximum offset	The maximum offset of all partitions in the current topic.
Topic consumption offset (distinguishing offset maximum value)	The maximum offset consumed by the consumer group in all partitions of the current topic.
The number of unconsumed messages in the topic (items)	The total number of unconsumed messages in all partition consumer groups of the current topic.
Topic consumption speed (items/minute)	The sum of consumption rates of all partition consumer groups of the current topic.

Consumer Group-Partition Monitoring

Monitoring Metrics	Description
Partition consumption speed (items/minute)	The consumption rate of the consumer group in the partition (items/minute).
Current consumption offset	Consumer group: The current consumption offset of the partition.
The maximum offset of the current partition	The maximum offset of the current partition.
The number of unconsumed messages (items) **(recommended)**	The number of unconsumed messages in the partition by the consumer group.

For more information on the metric names and descriptions, see Topic Monitoring Metrics.

CKafka monitoring API documentation

For more information, see the following documents:

Topic Monitoring Metrics

Instance Monitoring Metrics

Consumer Group Monitoring Metrics

Monitoring Entrance

1. Log in to the CKafka console.

2. In the instance list, click the **ID**/**Name** of the target instance to enter the instance details page.
3. At the top of the instance details page, Click **Monitor**, select resource tags, and set the time range to view the monitoring data about the instance.

Icon	Description
	Click to view monitoring metric year-on-year and quarter-on-quarter comparison.
	Click to refresh and obtain the latest monitoring data.
	Click to copy the chart to the dashboard. For more information about the dashboard, please see What is a Dashboard.
	Check this box to display legend information on the chart.

Note:
You can click the following tabs to view the monitoring data of instances, topic, and consumer groups.

Querying Advanced Monitoring (Pro Edition)

Last updated：2025-04-09 11:17:10

Overview

CKafka Pro Edition supports advanced monitoring. You can view metrics such as core services, production, consumption, and broker GC in the console, making it easier for you to troubleshoot CKafka issues.
This document describes how to view advanced monitoring metrics in the console and explains their meanings.

Monitoring metric description

Core Service Monitoring

Production

Consumption

Instance Resources

Broker GC

Monitoring Metric	Description	Normal Range
Network idle rate	This value is used to measure the current remaining I/O resources for concurrent request processing. The closer to 1 it is, the idler the instance is.	This value generally fluctuates between 0.5 and 1. If it falls below 0.3, the load is considered high.
Request Queue Depth	This value indicates the number of production requests that have not been processed. If this value is too large, it may be because that the number of concurrent requests is high, the CPU load is high, or the disk I/O hits a bottleneck.	If this value stays at 2000, the cluster load is considered high. If it is below 2000, it can be ignored.
Unsynced Replicas	This value indicates the number of unsynced replicas in the cluster. When there are unsynced replicas of an instance, there may be a health problem with the cluster.	If this value stays above 5 (this is because that some built-in topic partitions of Tencent Cloud may be offline and has nothing to do with the business), the cluster needs to be fixed. In case that the broker occasionally fluctuates, this value may surge and then become stable, which is normal.
ZK	This value indicates the number of	There is no normal range for this metric.

Reconnections	reconnections of the persistent connection between the broker and ZooKeeper. Network fluctuations and high cluster loads may cause disconnections and reconnections, thus leading to a leader switch. For more information, see FAQs > Client >What is leader switch.	The number of ZK reconnections is cumulative, so a large number does not necessarily mean that there is a problem with the cluster. This metric is for reference only.
ISR Expansions	This value is the number of Kafka ISR expansions. It will increase by 1 when an unsynced replica catches up with the data from the leader and rejoins the ISR.	There is no normal range for this metric. Expansions occur when the cluster fluctuates. No action is required unless this value stays above 0.
ISR Shrinks	This value is the number of Kafka ISR shrinks. It is counted when the broker is down and ZooKeeper reconnects.	There is no normal range for this metric. Shrinks occur when the cluster fluctuates. No action is required unless this value stays above 0.
Instance Broker storage utilization rate	Take the maximum disk usage of each Broker in the cluster per unit time.	This value generally ranges from 0 to 100. If this value reaches above 80, scaling out is required.

Monitoring Metric	Description	Normal Range
The number of Broker node production traffic throttling	The number of production traffic throttling is triggered on each Broker node.	-
Broker node production traffic (Bytes)	The production traffic for each Broker node.	-
Maximum production traffic (Bytes/s)	The maximum production traffic for each Broker node.	-
Production Traffic Usage Rate by Single Broker (%)	Percentage of production traffic usage by a single machine to purchased traffic.	This value generally ranges from 0 to 100. If this value reaches above 80 for a long-term period, scaling up needs to be considered.
Total Production Duration	This value indicates the total duration of a production request, which is based on metrics such as the request queue duration, local	This value generally ranges between 0 and 100 ms. It is normal to have a value falling in the range

	processing duration, and delayed response duration. At each point in time, the total duration is not equal to the sum of the following five metrics, because each metric is averaged.	of up to 1000 ms when the data volume is high. No action is required unless this value stays above 1000 ms.
Request Queue Duration	This value indicates the amount of time a production request waits in the queue of requests to be received. It means that the request packet waits for subsequent processing.	This value generally ranges between 0 and 50 ms. It is normal to have a value falling in the range of up to 200 ms when the data volume is high. No action is required unless this value stays above 200 ms.
Local Processing Duration	This value indicates the amount of time a production request is processed by the leader broker, i.e., the duration between the request packet is obtained from the request queue and it is written to the local page cache.	This value generally ranges between 0 and 50 ms. It is normal to have a value falling in the range of up to 200 ms when the data volume is high. No action is required unless this value stays above 200 ms.
ACK Wait Time	This value indicates the amount of time a production request waits for data to be synced. It is greater than 0 only when the client ack is -1; in other words, it is 0 as long as ack is 1 or 0.	This value generally ranges between 0 and 200 ms. It is normal to have a value falling in the range of up to 500 ms when the data volume is high. No action is required unless this value stays above 500 ms. This value for a multi-AZ instance is greater than that for a single-AZ instance when ack is -1. For more information, see Multi-AZ Deployment.
Response Delay	This value indicates the amount of time it takes the system to delay returning a packet to a production request. This value will always be 0 as long as the traffic of the instance does not exceed the purchased traffic, and it will be greater than 0 if the traffic is throttled.	This value will be 0 as long as the instance does not exceed the limit. If the limit is exceeded, there will be a delay of 0–5 minutes proportional to the excess; in other words, the maximum value is 5 minutes.
Response Queue Duration	This value indicates the amount of time a production request waits in the response queue. It means that the request packet waits to be sent to the client.	This value generally ranges between 0 and 50 ms. It is normal to have a value falling in the range

		of up to 200 ms when the data volume is high. No action is required unless this value stays above 200 ms.
95th Percentile of Delayed Response Duration in Production	The time taken for 95% of production requests to receive delayed responses from the system is less than this value.	This value is long-term 0 when the instance is not rate-limited. If the instance exceeds the limit, it will be delayed by 0 - 5 minutes according to the percentage of exceeding the limit. That is, the maximum value of this field is 5 minutes.

Monitoring Metric	Description	Normal Range
The number of Broker node consumption traffic throttling	The number of consumption traffic throttling is triggered on each Broker node.	-
Broker node consumption traffic (Bytes)	The consumption traffic for each Broker node.	-
Maximum consumption traffic (Bytes/s)	The maximum consumption traffic for each Broker node.	-
Consumption Traffic Usage Rate by Single Broker	Percentage of consumption traffic usage by a single machine to purchased traffic.	This value generally ranges from 0 to 100. If this value reaches above 80 for a long-term period, scaling up needs to be considered.
Total Consumption Duration	This value indicates the total duration of a consumption request, which is based on metrics such as the request queue duration and local processing duration. At each point in time, the total duration is not equal to the sum of the following five metrics, because each metric is averaged.	This value generally ranges between 500 and 1000 ms (the default `fetch.max.wait.ms` on the client is 500 ms). It is normal to have a value falling in the range of up to 5000 ms when the data volume is high.
Request Queue Duration	This value indicates the amount of time a consumption request waits in the request queue. It means that the request packet waits for subsequent processing.	This value generally ranges between 0 and 50 ms. It is normal to have a value falling in the range of up to 200 ms when the data volume is high.

		No action is required unless this value stays above 200 ms.
Local Processing Duration	This value indicates the amount of time it takes a consumption request to pull data from the leader broker, i.e., reading data from the local disk.	This value generally ranges between 0 and 500 ms. It is normal to have a value falling in the range of up to 1000 ms when the data volume is high. No action is required unless the value stays above 1000 ms, because the consumer sometimes may read cold data, which will consume a lot of time.
Consumption Wait Duration	The default `fetch.max.wait.ms` on the client is 500 ms. This value indicates the amount of time the client allows the server to wait before returning any packet to the client when the client cannot read any data.	This value is generally around 500 ms (the default `fetch.max.wait.ms` on the client is 500 ms), subject to the client's parameter settings.
Response Delay	This value indicates the amount of time it takes the system to delay returning a packet to a consumption request. This value will always be 0 as long as the traffic of the instance does not exceed the purchased traffic, and it will be greater than 0 if the traffic is throttled.	This value will be 0 as long as the instance does not exceed the limit. If the limit is exceeded, there will be a delay of 0–5 minutes proportional to the excess; in other words, the maximum value is 5 minutes.
Response Queue Duration	This value indicates the amount of time a consumption request waits in the response queue. It means that the request packet waits to be sent to the client.	This value generally ranges between 0 and 50 ms. It is normal to have a value falling in the range of up to 200 ms when the data volume is high. No action is required unless this value stays above 200 ms.
95th Percentile of Consumption Delayed Response Duration	The time taken for 95% of consumption requests to receive delayed responses from the system is less than this value.	This value is long-term 0 when the instance is not rate-limited. If the instance exceeds the limit, it will be delayed by 0 - 5 minutes according to the percentage of exceeding the limit. That is, the maximum value of this field is 5 minutes.

Monitoring Metric	Description	Normal Range
CPU Utilization (%)	This is the percentage of CPU time used by a process in a period of time to the total CPU time.	This value generally falls between 1 and 100. If it is above 90 in more than 5 consecutive statistical periods, the system load is very high and needs to be handled.
Private Network Inbound Bandwidth (MB)	This is the bandwidth that a CVM instance can reach for communication in the cluster. It limits the private network bandwidth and packet receiving capabilities according to different specifications.	This value is generally greater than 0 (CVM monitoring in the cluster will generate data). If there is no inbound bandwidth, the CVM service is abnormal or the network is unreachable.
Private Network Outbound Bandwidth (MB)	This is the bandwidth that a CVM instance can reach for communication in the cluster. It limits the private network bandwidth and packet sending capabilities according to different specifications.	This value is generally greater than 0 (CVM monitoring in the cluster will generate data). If there is no outbound bandwidth, the CVM service is abnormal or the network is unreachable.
Memory Utilization (%)	This is the percentage of the total memory space minus the used memory space to the total memory space.	This value generally falls between 1 and 100. If it is above 90, the program uses too much memory and some processes need to be handled.
Public Network Inbound Bandwidth (MB)	This is the bandwidth that CVM can reach in public network communication. The maximum public network bandwidth and packet receiving capabilities vary by specification.	It will be greater than 0 if there is public network inbound traffic; otherwise, it will be 0.
Public Network Outbound Bandwidth (MB)	This is the bandwidth that CVM can reach in public network communication. The maximum public network bandwidth and packet sending capabilities vary by specification.	It will be greater than 0 if there is public network outbound traffic; otherwise, it will be 0.

Monitoring Metric	Description	Normal Range
Heap Memory Utilization (%)	The percentage of the Broker's JVM heap memory usage in the total heap memory, take the maximum in a unit of time.	The maximum value of this metric generally fluctuates between 80 and 90. If it shows an upward trend for a long-term, attention required for the usage method of heap memory.
Young GC Count	Young GC count of the broker.	This value generally ranges between 0 and 300. If it stays above 300, the GC parameters need to be adjusted.
Full GC Count	Full GC count of the broker.	This value is generally 0. Actions are required if it is greater than 0.

Directions

1. Log in to the CKafka console.
2. In the instance list, click the **ID/Name** of the target instance to enter the instance details page.
3. At the top of the instance details page, click **Monitoring** > **Advanced Monitoring**, select the metric to be viewed, and set the time range to view the monitoring data.

Monitoring information display

Note:

You can click the following tabs to view the monitoring information of the core service, production, consumption, instance resource, and broker GC.

Core Services

Production

Consumption

Instance Resources

Broker GC

Causes of monitoring metric exceptions

The following describes causes of certain monitoring metric exceptions.

Metric	Exception Cause
Cluster load (%)	When the cluster load exceeds 70% for single AZ deployment, 35% for 2 AZ deployment, or 47% for 3 AZ deployment, it indicates a high cluster load despite low bandwidth utilization. This may signal the need for a cluster bandwidth scale-out based on the cluster load metric.
CPU Utilization (%)	When you find that it is above 90% in more than 5 consecutive statistical periods, you can first check whether there are message compression and message format conversion. If the client machine has sufficient CPU resources, we recommend that you enable Snappy compression. You can observe the request queue depth at the same time. If this value is too large, it may be because that the request volume is too high, which may also cause a high CPU load.
Unsynced Replicas (Count)	When this value is above 0, there are unsynced replicas in the cluster. This is usually due to broker node exceptions or network issues. You can troubleshoot through the broker logs.
Full GC Count (Count)	An occasional old GC occurrence may be caused by disk I/O or CVM issues. You can check to see whether brokers with the same IP have the same issue. If so, submit a ticket for assistance.
Request Queue Depth (Count)	If the client's production and consumption time out but the CVM load remains normal, it means the request queue length of the CVM instance has reached the upper limit, which is 500 by default. You can submit a ticket to adjust it appropriately according to the purchased resource configuration.

©2013-2025 Tencent Cloud International Pte. Ltd.

Configuring Alarms

Last updated：2025-04-09 11:17:10

Overview

TDMQ for CKafka (CKafka) not only provides multiple monitoring metrics for running CKafka clusters to monitor their operational status, but also provides the configuration alarm feature for key metrics. You can configure alarm rules for monitoring metrics. When a monitoring metric reaches the set alarm threshold, the observability platform can notify you by email, SMS, WeChat, phone calls and other methods, helping you promptly identify cluster issues and handle them.

Directions

Configuring an alarm policy

The created alarm can determine whether an alarm notification should be sent based on the comparison between the monitoring metric and the given threshold in the selected time period. You can take appropriate precautionary or remedial measures in a timely manner when the alarm is triggered because the status of CKafka changes. Properly creating alarm policies can help improve the robustness and reliability of your applications.

Note:

Be sure to configure alarms for your instance to prevent exceptions caused by traffic spikes or specification limits.

1. Log in to the CKafka console.
2. In the instance list, click **Configure Alarm Policy** in the **Operation** column to enter the alarm configuration page.
3. On the alarm configuration page, select a policy type and instance, and set the alarm rule and notification template.

Monitoring Type: Select **Tencent Cloud services**.

Policy Type: Select **CKafka**.

Alarm Object: Select the CKafka resource for which to configure the alarm policy.

Trigger Condition: You can select **Select template** or **Configure manually**. The latter is selected by default. For more information on manual configuration, see the description below. For more information on how to create a template, see Creating a trigger condition template.

Note:

Metric: For example, if you select 1 minute as the statistical period for the "Disk Utilization" metric, then if the disk utilization exceeds the threshold for N consecutive data points, an alarm will be triggered.

Alarm Frequency: For example, "Alarm once every 30 minutes" means that there will be only one alarm triggered every 30 minutes if a metric exceeds the threshold in several consecutive statistical periods. Another alarm will be triggered

only if the metric exceeds the threshold again in the next 30 minutes. For the metrics for which we recommend that you configure an alarm policy, see Monitoring and Alarm Policies Recommended for CKafka.

Notification Template: You can select an existing notification template or create one to set the alarm recipient objects and receiving channels.

4. Click **Complete**.

Note:

For more information on alarms, see Creating Alarm Policy.

Creating a trigger condition template

1. On the **Configure Alarm Policy** page, select **Select Template** for **Trigger Condition** and click **Create Trigger Condition Template**.
2. On the template creation page, configure the policy type.

Policy Type: Select **CKafka**.

Apply preset trigger conditions: Select this option and the system recommended alarm policy will be displayed.

3. After confirming that everything is correct, click **Save**.

4. Return to the alarm policy creation page and click **Refresh**. The alarm policy template just configured will be displayed.

CKafka Alert Configuration Suggestions

Metrics that affect the stability of business data, please see CKafka Data Reliability Statement.

The following are some metrics that require special attention and recommended alarm configurations during your use of CKafka:

Metric	Recommended Alarm Configuration	Description
Disk utilization (%)	The statistical period is 1 minute, and if this value is above 80% in 5 consecutive periods, an alarm will be triggered once every 30 minutes.	The average disk utilization refers to the average of the disk utilization values of all nodes in the cluster. If the disk utilization is too high, the node will not have enough disk space to sustain messages assigned to it, making it unable to store messages. We recommend you clear the data or scale the cluster in time when this value exceeds 75%.
Unconsumed messages (count)	The statistical period is 5 minutes, and if this value is above 8,000 in 10 consecutive periods, an alarm will be triggered once every 30 minutes.	Too many retained messages will cause the broker node disk utilization to soar up, so no more messages can be received, and the service will stop. In this case, you should perform scaling.

Production peak bandwidth (MB/s)	The statistical period is 1 minute, and if this value is above the purchased bandwidth specification of the instance in 5 consecutive periods, an alarm will be triggered once every 10 minutes.	This refers to the maximum value of traffic per second within one minute. You can determine whether the purchased traffic cap is exceeded and upgrade the specification accordingly.

©2013-2025 Tencent Cloud International Pte. Ltd.

Suggestions for Alarm Configuration

Last updated：2024-01-09 14:47:32

CKafka not only provides a number of monitoring metrics for running CKafka clusters to monitor their health, but also allows you to configure alarms for key metrics, so that you can identify cluster problems and address them in a timely manner. For more information, see Viewing Monitoring Information and Configuring Alarm.

This document describes some metrics that require special attention during your use of CKafka, as well as recommended alarm configurations:

Metric	Recommended Alarm Configuration	Description
Disk utilization (%)	The statistical period is 1 minute, and if this value is above 80% in 5 consecutive periods, an alarm will be triggered once every 30 minutes.	The average disk utilization refers to the average of the disk utilization values of all nodes in the cluster. If the disk utilization is too high, the node will not have enough disk space to sustain messages assigned to it, making it unable to store messages. We recommend you clear the data or scale the cluster in time when this value exceeds 75%.
Unconsumed messages (count)	The statistical period is 5 minutes, and if this value is above 8,000 in 10 consecutive periods, an alarm will be triggered once every 30 minutes.	Too many retained messages will cause the broker node disk utilization to soar up, so no more messages can be received, and the service will stop. In this case, you should perform scaling.
Production peak bandwidth (MB/s)	The statistical period is 1 minute, and if this value is above the purchased bandwidth specification of the instance in 5 consecutive periods, an alarm will be triggered once every 10 minutes.	This refers to the maximum value of traffic per second within one minute. You can determine whether the purchased traffic cap is exceeded and upgrade the specification accordingly.

©2013-2025 Tencent Cloud International Pte. Ltd.

Viewing Advanced Ops Features (Pro Edition)

Last updated：2024-11-07 14:49:02

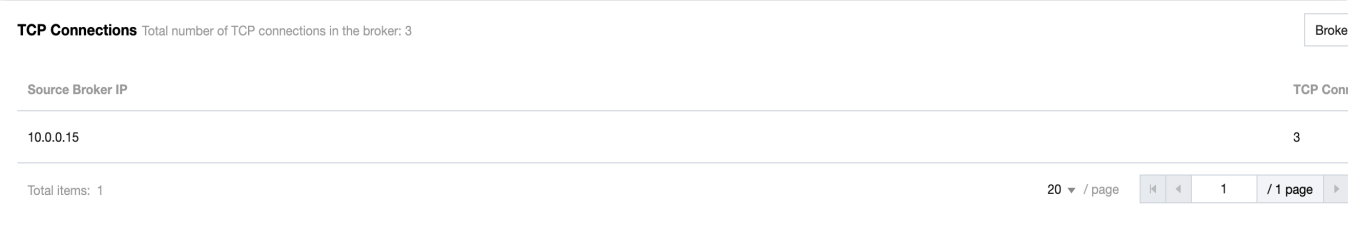
Overview

CKafka Pro Edition supports advanced Ops features. You can view the number of TCP connections, unsynced replica details, and topic/consumer group rankings in the console, making it easier for you to troubleshoot CKafka issues.

Directions

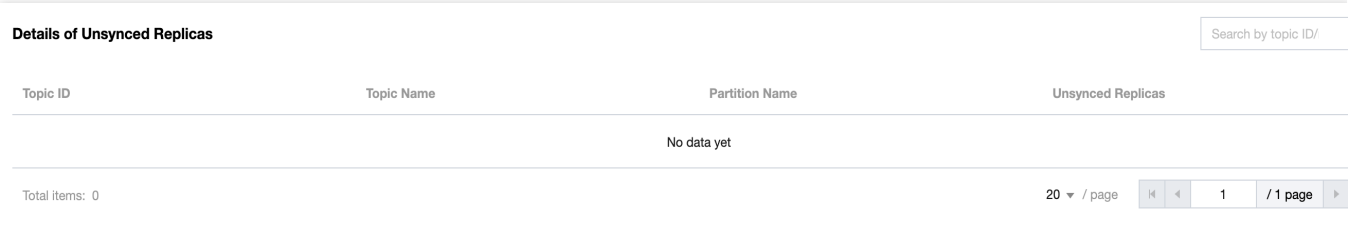
1. Log in to the CKafka console.
2. In the instance list, select a region and click the **ID/Name** of the target instance to enter the instance details page.
3. At the top of the instance details page, click **Monitoring** > **Dashboard** and set the time range to view related ranking information.

TCP Connections: Displays the total number of TCP connections to the current broker. It allows you to view the connections to each server more easily when the number of instance connections is about to be used up. When the total TCP connections remain below 500, full data download is supported.



Advanced Topic Metrics

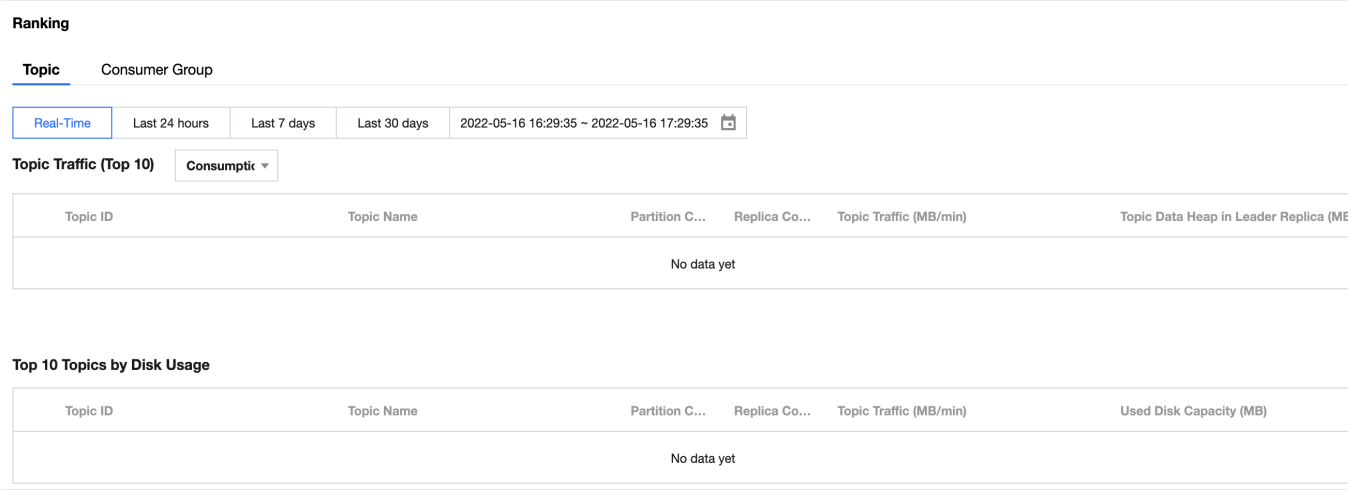
Details of Unsynced Replicas: It displays the details of unsynced replicas.



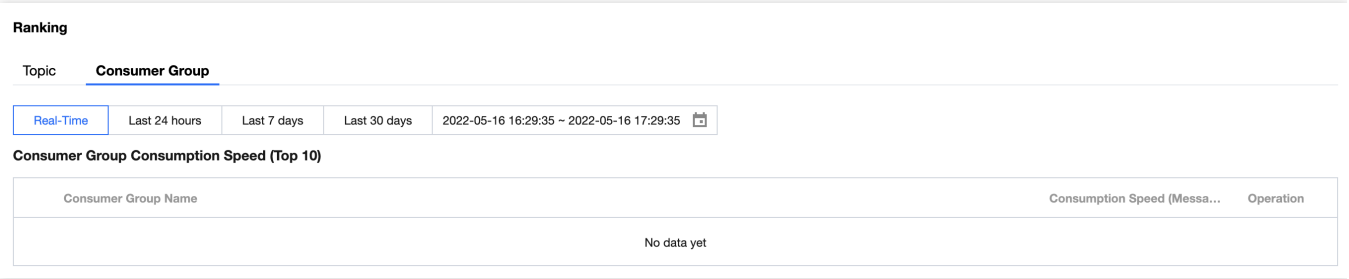
Node distribution: Displays the replica distribution of a specified topic across broker nodes.

Ranking:

Topic: Displays the top 10 topics in terms of production and consumption traffic as well as disk usage. Supports viewing topic ranking statistics for specific broker nodes.



Consumer Group: It displays the top 10 consumer groups in terms of consumption speed.



Smart Ops
Elastic Scaling
Disk Utilization Policy

Last updated：2025-04-09 11:18:55

Overview

CKafka supports automatic adjustment of the disk utilization. After the disk utilization reaches the threshold, you can set the **Dynamic Message Retention Policy** to reduce the message retention time or set the **Automatic Disk Capacity Expansion** to adjust the disk space.
Dynamic Message Retention Policy: After you set the message retention time, expired messages will be deleted. In case of message surges, normal production and consumption cannot be performed after the disk is full. If you set a dynamic retention policy, when the disk utilization reaches a certain percentage, a certain proportion of data will be automatically expired to avoid the above issue.
Automatic Disk Capacity Expansion: When the disk load gets heavy, messages cannot be produced and consumed normally. After the automatic disk capacity expansion policy is set, when the disk load reaches the trigger threshold, the disk capacity will be automatically adjusted according to the policy, thus avoiding this problem.

Policy Type	Supported by CKafka Standard Edition	Supported by CKafka Pro Edition
Dynamic message retention	Yes	Yes
Automatic disk capacity expansion	No	Yes

Note:
You can enable either dynamic message retention or automatic disk capacity expansion at any time.

Feature Entry

Standard Edition

1. Log in to the CKafka console.
2. Click **Instance List** on the left sidebar and then the **ID/Name** of your instance to enter the instance details page.

3. In the **Disk Utilization Policy** module on the instance's basic information page, you can configure a policy to adjust the disk space.

Pro Edition

1. Log in to the CKafka console.
2. Click **Instance List** on the left sidebar and then the **ID/Name** of your instance to enter the instance details page.
3. Select the **Smart Ops** tab at the top of the page, select the **Auto Scaling** tab, and configure the disk space adjustment policy in the **Disk Utilization Policy** module.

Configuring a Disk Utilization Policy

Dynamic Message Retention Policy

Automatic Disk Capacity Expansion Policy

1. In the **Disk Utilization Policy** module, enable **Dynamic Retention Policy**.

Note:

The default dynamic policy reduces the message retention time by 10% when the disk utilization reaches 75%.

2. Click **Configure** in the **Operation** column of the dynamic retention policy to configure **Dynamic Policy** and **Minimum Retention Period**.

Dynamic Policy: After message retention time adjustment is triggered, the broker will delete the oldest historical data based on the new retention time. This feature has a certain delay.

Minimum Retention Period: It can be 1 minute to 30 hours. If the dynamic retention period is lower than this parameter, no dynamic adjustment will be triggered.

3. Click **View** in the **Adjustment Record** column to enter the event center and view the adjustment records of automatic disk capacity expansion.

1. In the **Disk Utilization Processing** module, enable **Automatic Disk Capacity Expansion**.

Note:

When the disk utilization reaches 75%, the disk capacity will be automatically expanded by 10% by default. The maximum disk capacity is as displayed in the console, which is the upper limit of the disk corresponding to the current instance specification.

2. Click **Configure** in the **Operation** column of the automatic disk capacity expansion policy to configure **Dynamic Policy** and **Max Disk Capacity**.

Dynamic Policy: After the disk load reaches the trigger threshold, the disk capacity will be automatically adjusted according to the capacity expansion policy, with a certain delay.

Note:

The disk capacity will be expanded in increments of 100 GB.

Max Disk Capacity: Automatic disk capacity expansion will no longer be triggered after the disk capacity has been expanded to the specified value.

3. Click **View** in the **Adjustment Record** column to enter the event center and view the adjustment records of automatic disk capacity expansion.

Dynamic Partition Processing

Last updated：2023-04-27 17:28:38

Overview

CKafka supports dynamic partition processing. After you enable **Automatic Partition Balancing**, CKafka will automatically check the partition distribution of topics on your specified time and then select a time during off-peak hours to initiate partition balancing.

Note

This feature is only available for CKafka Pro Edition instances.

Directions

1. Log in to the CKafka console.
2. Click **Instance List** on the left sidebar and then the **ID/Name** of your instance to enter the instance details page.
3. Select the **Smart Ops** tab at the top and click **Elastic Scaling**.
4. In the **Dynamic Partition Processing** module, enable **Automatic Partition Balancing**.
5. Click **Configure** in the **Operation** column to set the automatic partition balancing policy.

Custom time: You can customize the time to initiate partition balancing. We recommend that you select a time during off-peak hours to avoid affecting your business.

Auto-selected time: CKafka will select the time to initiate partition balancing during off-peak hours calculated by automatic analysis.

6. Click **View** in the **Adjustment Record** column to enter the event center and view the adjustment records of automatic partition balancing.

©2013-2025 Tencent Cloud International Pte. Ltd.

Elastic Bandwidth

Last updated：2025-06-24 17:07:39

Background

TDMQ for CKafka offers fixed bandwidth specifications, and supports the upgrade and downgrade feature. This facilitates users to make adjustments when there are changes in business. However, in the actual business operation scenarios, we often find unpredictable changes in business traffic. Also, due to potential data migration, the expansion time cannot be controlled, thus causing unexpected impacts on the business.
To address such issues, the Pro Edition of TDMQ for CKafka introduces the capability of elastic bandwidth. That is, when the business exceeds the cluster's fixed specifications, a certain degree of elastic space will still be provided to ensure the normal operation of the business.

Feature Overview

The Pro Edition of TDMQ for CKafka supports elastic bandwidth configuration. This is suitable for scenarios where there is occasional sudden traffic on the business. The part exceeding the instance specifications is charged on a pay-as-you-go basis, and the incremental part that can be added has an upper limit. If the business traffic exceeds the instance specifications for most of the time, it is recommended to quickly scale up or down using the existing upgrade and downgrade capability.

Billing Description

The pay-as-you-go method is adopted for elastic bandwidth. Once the billing mode is enabled, no fees will be incurred if elastic bandwidth is not used. The Pro Edition's console offers relevant monitoring metrics to help with observation.
The billing items for elastic bandwidth consist of two parts: elastic production traffic and elastic consumption traffic.
The related billing methods are outlined in the following table:

Billing Items	**Unit Price**	Calculation Formula	Settlement Method
Elastic Production Fees	0.006 USD/GB/hour	Total Elastic Production Traffic (GB) (excluding traffic generated by replicas) x Elastic Pricing (USD/GB) x Number of Replicas	The time is settled hourly. If the time is less than one hour, it is counted as one hour. Billing is pushed daily
Elastic	0.006	Total Elastic Consumption Traffic	The time is settled

©2013-2025 Tencent Cloud International Pte. Ltd.

Consumption Fees	USD/GB/hour	(GB) (excluding traffic generated by replicas) x Elastic Pricing (USD/GB)	hourly. If the time is less than one hour, it is counted as one hour. Billing is pushed daily

Key Metrics

When you enable elastic bandwidth in CKafka pro edition, you will see two elastic bandwidth metrics on the monitoring interface. Based on these metrics, you can observe whether the elastic space has been triggered for use, and it can also serve as a reference for billing calculations.

Elastic bandwidth production traffic

Sum of elastic bandwidth production traffic usage of this cluster instance in the selected time granularity.

elastic bandwidth consumption traffic

Sum of elastic bandwidth consumption traffic usage of this cluster instance in the selected time granularity.

Elastic Bandwidth Space

Available space of elastic bandwidth. It is related to Pro Edition cluster specifications, and below is the corresponding elastic interval.

Bandwidth Range (MB/s)	Upward Adjustment Margin (MB/s)
20~100	30
120~300	60
320~600	100
620~900	200
920~1200	600
1,600 and above	700

Use Limits

1. Current elastic bandwidth configuration does not support deactivation once enabled.
2. This feature is in allowlist grayscale testing. To enable, you may Contact Us.

©2013-2025 Tencent Cloud International Pte. Ltd.

Permission Management
Configuring ACL Policy

Last updated：2024-01-09 14:49:39

Overview

This document describes how to configure SASL authentication and ACL (access control list) rules in the CKafka console to enhance access control in public/private network transfers and permission control in production and consumption of resources such as topics.

Note:

Kafka offers various security authentication mechanisms, which mainly include SSL and SASL. SASL/PLAIN is a more commonly used authentication method that is based on account and password. CKafka supports SASL_PLAINTEXT authentication. For more information, see Adding Routing Policy.

An ACL helps you define a set of permission rules to allow/deny users to read/write topic resources through IPs.

Directions

Creating a user

1. Log in to the CKafka console.
2. On the topbar, select a region and click the **ID**/**Name** of the target instance.
3. On the instance details page, click the **ACL Policy Management** tab at the top.
4. On the **User Management** tab, click **Create** and enter the username and password to create a user.

Configuring an ACL policy

1. On the **ACL Policy Management** page, select the **Policy List** tab.
2. Click **Batch Set** to grant permissions to the user.

Note:

If only allow rules are configured, any IPs not in those rules cannot connect to the instance.

If only deny rules are configured, any IPs not in those rules can connect to the instance.

If both allow and deny rules are configured, only IPs in allow rules can connect to the instance.

Instances on v2.4.1 or later

Instances on other versions

You can grant permissions to the user through the **Topics**, **Topic name prefix**, or **Preset Rule** option.

Note:

You can enter multiple IPs or IP ranges and separate them by `;` when configuring the ACL policy. If the IP is empty, the permission will be added for **all IPs** by default.

Topics: Select multiple topics that need to be configured with the same ACL policy.

Topic name prefix: Fuzzy match topics that need to be configured with the same ACL policy by topic name prefix. You need to specify the fuzzy matching rule name. After this is set, when a new topic with a name containing the specified prefix is added, the system will automatically configure the specified ACL policy for it.

Note:

Up to five fuzzy match rules can be set.

Preset Rule: A set of rules can be preset and automatically applied during subsequent topic creation.

Note:

Up to five preset rules can be set.

You can grant permissions to the user through the **Topics** or **Preset Rule** option.

Note:

You can enter multiple IPs or IP ranges and separate them by `;` . If the IP is empty, the permission will be added for **all IPs** by default.

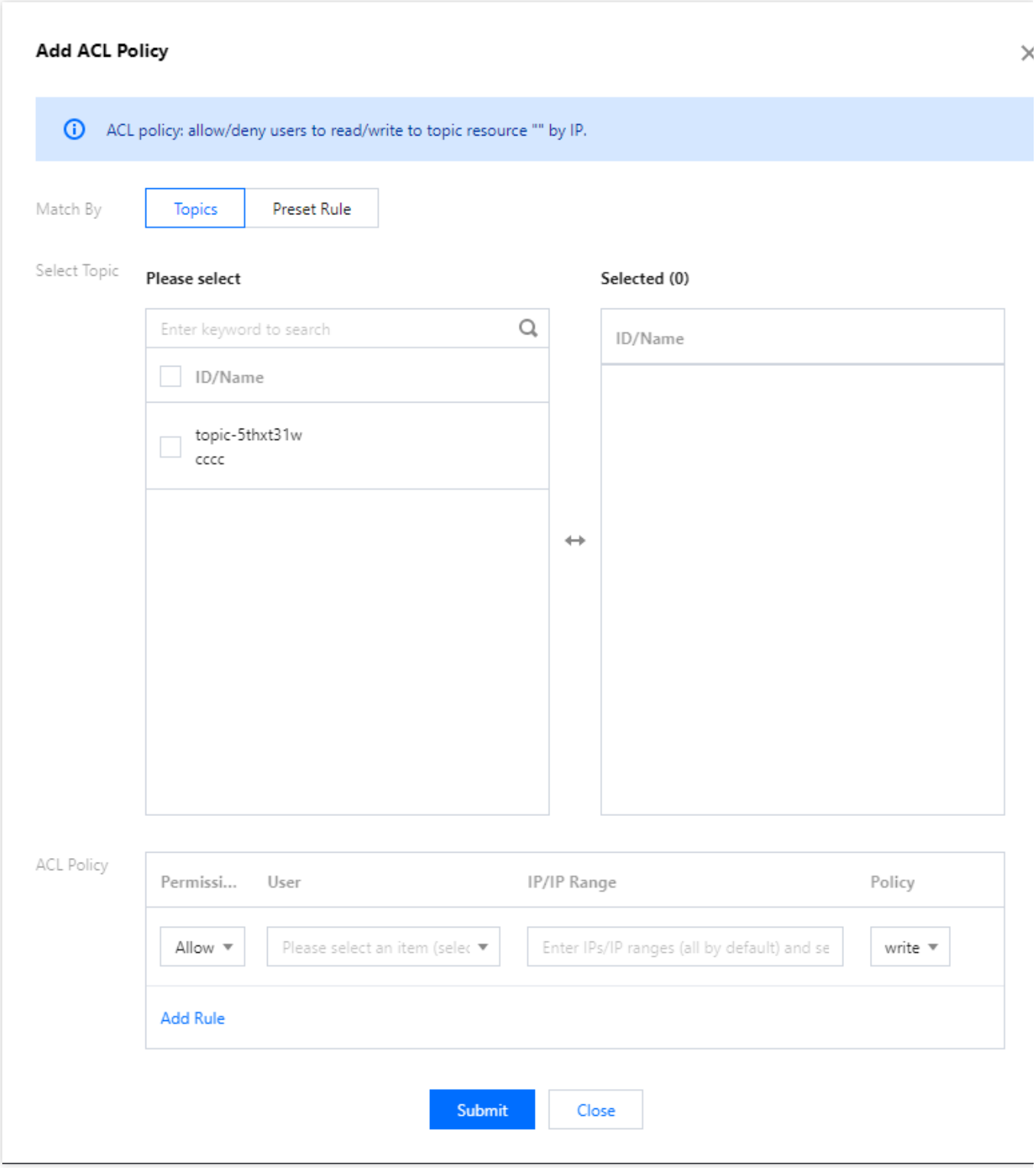
Topics: Select multiple topics that need to be configured with the same ACL policy.

Preset Rule: A set of rules can be preset and automatically applied during subsequent topic creation.

Note:

Up to five preset rules can be set.

©2013-2025 Tencent Cloud International Pte. Ltd.



Next steps: After the authorization is completed, the user can access CKafka through the SASL access point and consume messages by using the PLAIN mechanism. For more information, see the SDK documentation.

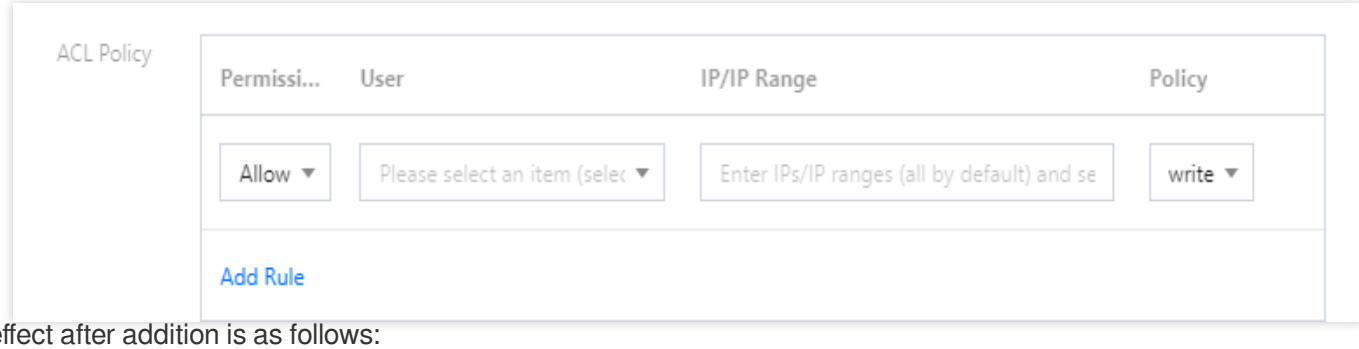
Use limits

1. Enabling routing only affects the authentication method during access, while the set ACL policy takes effect globally.

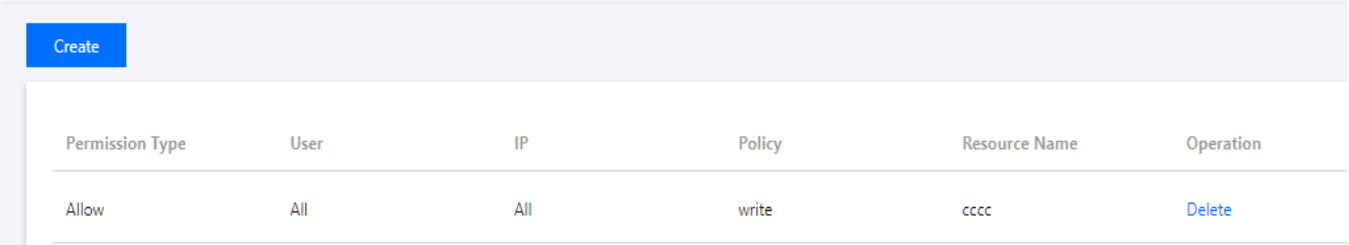
2. If you use the PLAINTEXT method to access CKafka while enabling public network access routing, the ACL previously set for the topics will still take effect. If you want PLAINTEXT access to be unaffected, add the read/write permissions of all users for the topics that PLAINTEXT needs to access.

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:



3. If a topic is already being used by another Tencent Cloud service (e.g., log shipping in CLS, message dump in SCF, and component consumption in EMR), enabling ACL policy is equivalent to imposing restrictions on the permissions of these linked capabilities, and they will directly become unavailable. Therefore, be sure to do so with caution. In such cases, we recommend that you produce the same data to another topic for separate processing instead of configuring a unified ACL policy on the same topic.

Viewing a preset rule

1. On the ACL policy management page, select **Preset Rule**.
2. In the preset rule list, click **Details** in the **Operation** column to view the details of a rule.

Deleting a preset rule

1. On the ACL policy management page, select **Preset Rule**.
2. In the preset rule list, click **Delete** in the **Operation** column to delete a rule.

The impact of deleting the preset rule varies by the type of rule match:

If the rule is a fuzzy match rule, it will no longer be automatically applied to new topics or take effect for topics to which it is already applied.

If the rule is not a fuzzy match rule, it will no longer be automatically applied to new topics but will still take effect for topics to which it is already applied.

CAM

Last updated：2024-01-09 14:49:39

Basic CAM Concepts

The root account authorizes sub-accounts by associating policies. The policy setting can be specific to the level of **[API, Resource, User/User Group, Allow/Deny, and Condition]**.

Account

Root account: It owns all Tencent Cloud resources and can access any of its resources.
Sub-account: It includes sub-users and collaborators.
Sub-user: It is created and fully owned by a root account.
Collaborator: It has the identity of a root account. After it is added as a collaborator of the current root account, it becomes one of the sub-accounts of the current root account and can switch back to its root account identity.
Identity credential: It includes login credentials and access certificates. **Login credential** refers to a user's login name and password. **Access certificate** refers to Tencent Cloud API keys (`SecretId` and `SecretKey`).

Resource and permission

Resource: An object that is operated in Tencent Cloud services, such as a CVM instance, a COS bucket, or a VPC instance.
Permission: It is an authorization that allows or forbids users to perform certain operations. By default, **a root account has full access to all resources under it**, while **a sub-account does not have access to any resources under its root account**.
Policy: It is a syntax rule that defines and describes one or more permissions. The **root account** performs authorization by **associating policies** with users/user groups.
View CAM documentation >>

Relevant Documents

Content	Document
Understand the relationship between policies and users	Concepts
Understand the basic structure of policies	Element Reference
Check CAM-enabled products	Overview

©2013-2025 Tencent Cloud International Pte. Ltd.

Sample CAM Policy

Full access policy for CKafka

Grant a sub-user full access (including resource creation and management) to the CKafka service.

```
{
  "version": "2.0",
  "statement": [
    {
      "action": [
            "name/ckafka:*",
            "name/monitor:GetMonitorData"
      ],
      "resource": "*",
      "effect": "allow"
    }
  ]
}
```

You can also configure the system's full read/write policy to support this permission.

1. Log in to the CAM console.
2. Click Policies on the left sidebar.
3. In the policy list, click **Create Custom Policy**.
4. In the **Select Policy Creation Method** pop-up window, select **Create by Policy Syntax**.
5. In **Template Type**, search for **CKafka**, select **QcloudCKafkaFullAccess** (full access to CKafka), and click **Next**.
6. Click **Complete**.

Read-only policy for a CKafka instance

1. Create a policy with the Policy Generator and grant permission for listing operations and product monitoring.

```
{
 "version": "2.0",
 "statement": [
     {
         "effect": "allow",
         "action": [
             "name/ckafka:ListInstance",
             "name/monitor:GetMonitorData"
         ],
         "resource": [
             "*"
         ]
     }
 ]
```

```
}
```

2. Grant read-only access to the specified instance.

Note:

List* APIs don't support authentication at the resource level.

```
{
    "version": "2.0",
    "statement": [
        {
            "effect": "allow",
            "action": [
                "name/monitor:GetMonitorData",
                "name/ckafka:Get*"
            ],
            "resource": [
                "qcs::ckafka:gz::ckafkaId/uin/$createUin/$instanceId"
            ]
        }
    ]
}
```

You can also configure the system's read-only policy to support this permission.

1. Log in to the CAM console.
2. Click Policies on the left sidebar.
3. In the policy list, click **Create Custom Policy**.
4. In the **Select Policy Creation Method** pop-up window, select **Create by Policy Syntax**.
5. In **Template Type**, search for **CKafka**, select **QcloudCkafkaReadOnlyAccess** (read-only access to CKafka), and click **Next**.
6. Click **Complete**.

Tag Management

Last updated：2025-04-09 10:57:19

Tag is a marker provided by Tencent Cloud for identifying cloud resources. It is a Key-Value pair. Tags can help you conveniently classify and manage TDMQ CKafka Edition resources from various dimensionalities (for example, business, purpose, owner).

Notes:

Tencent Cloud will not use the tags you set. Tags are only used for your management of TDMQ CKafka version resources.

Use Limits

Quantity Limit

One resource can bind up to 50 tags.

Naming Restrictions

Tag Key	Tag Value
Tag keys starting with `qcs:` , `project:` , `item` , etc., are reserved for the system, and creating system-reserved tag keys is prohibited.	-
In UTF-8, a tag key must be at least 1 and at most 127 Unicode characters.	In UTF-8, a tag value must be at least 1 and at most 255 Unicode characters.
Supports characters encoded in UTF-8 format, spaces, digits, and special characters. Does not support those starting or ending with a space. Supported in English: + - = . _ : / @ ( ) [ ] , ; > Supported in Chinese: + - = / @ ( ) [ ] :	
Case sensitivity.	

Usage Examples

Case Background

A company has 10 CKafka message queue instances on Tencent Cloud, which are owned by three departments: Ecommerce, Gaming, and Entertainment. The instances are used for services such as marketing campaigns, game A,

game B, and post-production. The OPS owners of the three departments are Zhang San, Li Si, and Wang Wu, respectively.

Set Tag

To facilitate management, the company categorizes its CKafka resources with tags and defines the following tag key-value pairs.

Tag Key	Tag Value
Department	Ecommerce, Game, and Entertainment
Business	Marketing, Game A, Game B, and Post-production
OPS Owner	Tom, Jane, and Harry

Bind the tag key-value pairs to CKafka. The relation between resources and the tag key-value pairs is shown in the table below.

ID	Department	Business	OPS Owner
ckafka-1jqwv1	E-commerce	Marketing campaigns	Zhang San
ckafka-1jqwv12	E-commerce	Marketing campaigns	Wang Wu
ckafka-1jqwv13	Games	Game A	Zhang San
ckafka-1jqwv13	Games	Game B	Zhang San
ckafka-1jqwv14	Games	Game B	Zhang San
ckafka-1jqwv15	Games	Game B	Li Si
ckafka-1jqwv16	Games	Game B	Li Si
ckafka-1jqwv17	Games	Game B	Li Si
ckafka-1jqwv18	Entertainment	Post-production	Wang Wu
ckafka-1jqwv19	Entertainment	Post-production	Wang Wu
ckafka-1jqwv110	Entertainment	Post-production	Wang Wu

Take ckafka-1jqwv1 as an example. We can add the following three groups of tags to this instance:

Tag Key	Tag Value
dept	ecommerce

business	mkt
owner	zhangsan

Similarly, you can also set appropriate tags for other resources based on their department, business scope, and owner information.

Set Tags in TDMQ CKafka Version Console Settings

As detailed above, after you complete the design of the tag key and tag value, you can log in to the TDMQ CKafka Version console to perform tag settings.

1. Log in to the CKafka console.
2. On the cluster management list page, after selecting a region, check the cluster whose tags need to be edited and click edit resource tag at the top of the page.

3. Set tag in the pop-up "Edit Tag" window.

4. click **Confirm**. A prompt indicating successful modification will appear, and the tags bound to it can be viewed in the resource tag column of the cluster.

Filtering Resources by Tag Key

You can filter out clusters bound to a specific tag in the following steps:

1. In the search box at the top right of the page, select a tag.
2. In the pop-up window after "Tag:", select the tag you want to search for and click ok to perform search. For example: Select `Tag: owner:zhangsan` to filter out the cluster bound with tag key `owner:zhangsan` .

Editing a Tag

1. On the cluster management list page, after selecting a region, check the cluster whose tags need to be edited and click edit resource tag at the top of the page.

Notes:

Supports batch editing tags for up to 20 resources.

2. In the pop-up "Edit Tag" window, add, modify or delete tags based on actual needs.

Querying Message

Last updated：2024-11-07 15:35:58

Overview

When encountering a consumption exception, you can troubleshoot the problem by querying the message in the CKafka console. This document describes how to query messages in the CKafka console.

The CKafka console supports query by offset and time, which are used in different scenarios:

Query by offset: You know the ID of a topic's partition to which the message is sent and the offset of the message.

Query by time: You don't know the offset of the message but know when the message is sent.

Note:

The console does not list all the messages after the specified time or offset. You can query at most 20 messages at a time.

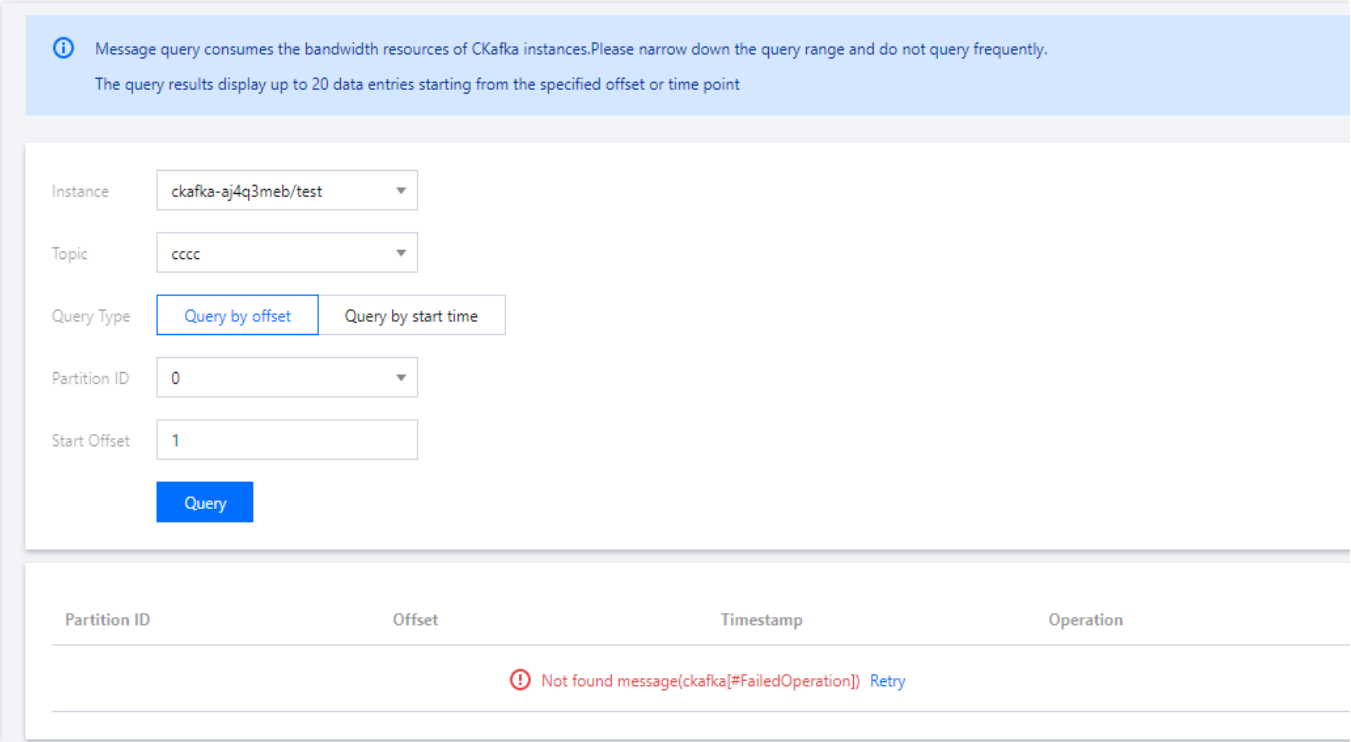
Refrain from frequently querying as message querying also consumes bandwidth.

Currently, message queries only support querying messages within 1 MB.

Directions

Query by offset

1. Log in to the CKafka console.
2. Select **Instance List** on the left sidebar, select a region, and click the **ID/Name** of the target instance to enter the instance details page.
3. On the instance details page, select **Topic Management** and click **More**> **Message Query** in the **Operation** column on the right of the target topic.
4. Select **Query by offset**, enter the partition ID and starting offset, and click **Query* to view the message details.



Partition ID: Topic partition to which the message is sent.

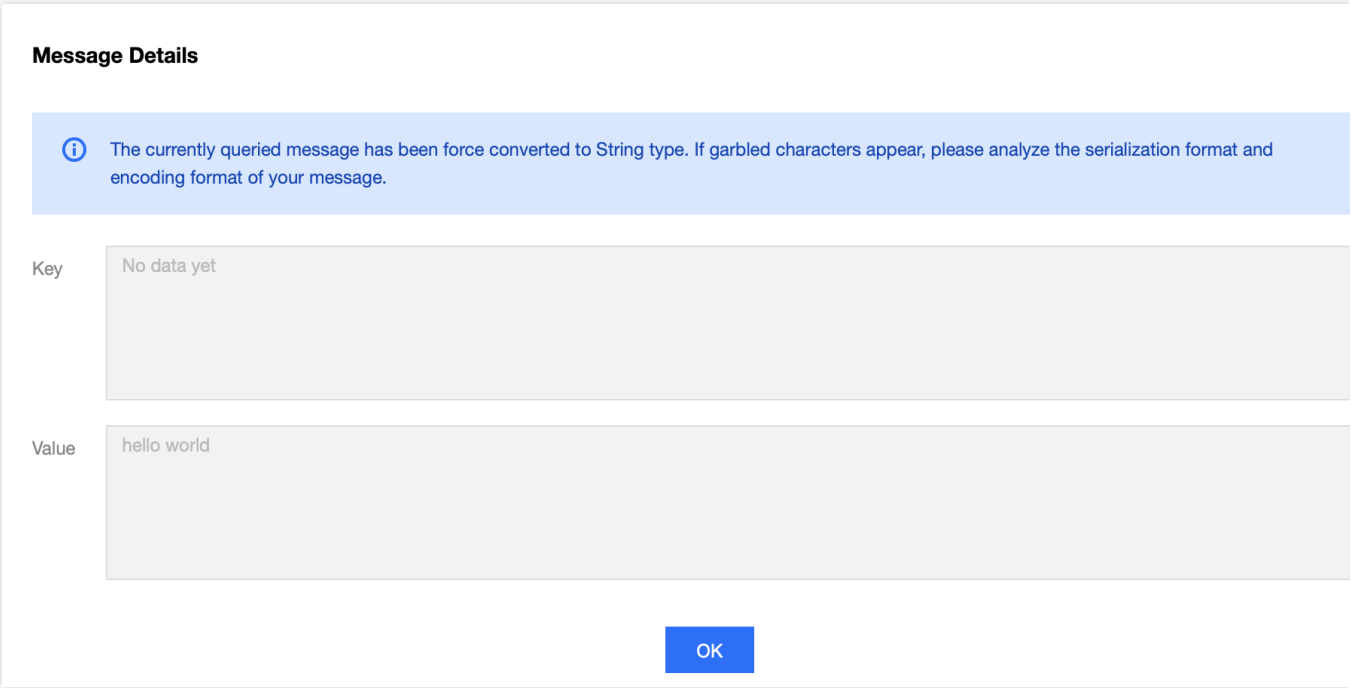
Offset: Consumer offset of the message.

Timestamp: `timestamp` in `ProducerRecord`.

Operation:

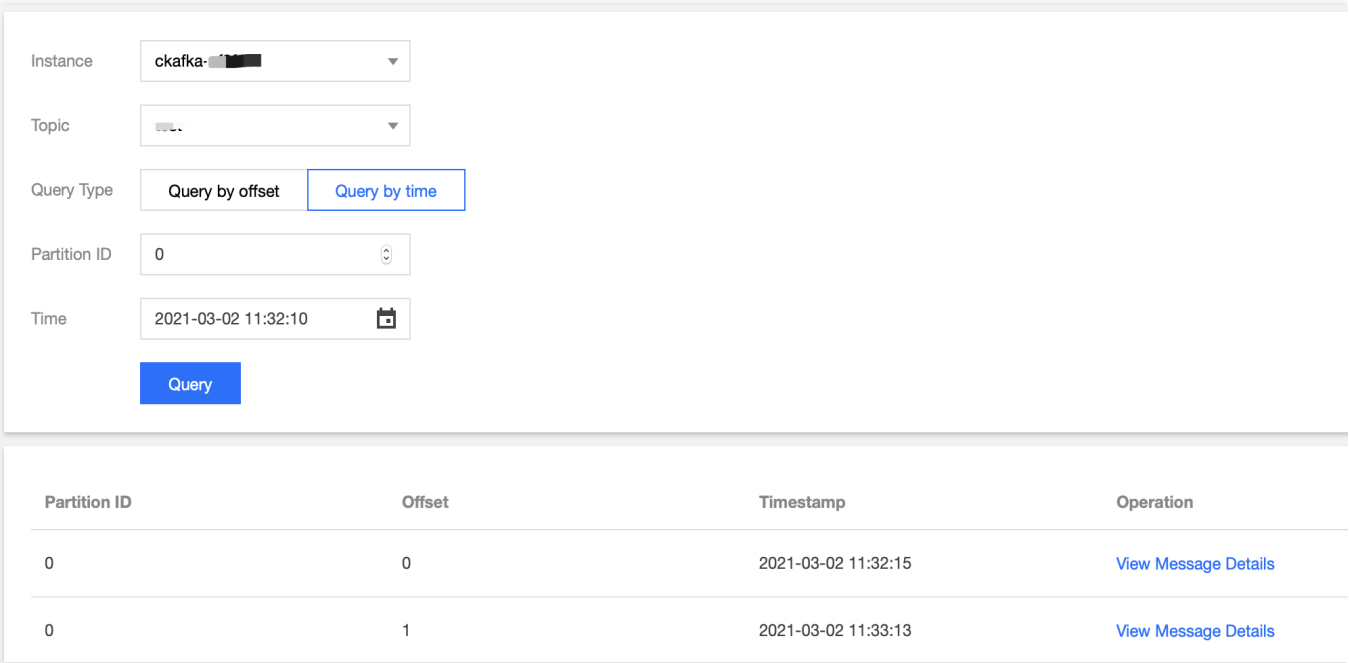
Click **Download Message** to download the message in the format of "header, key, value".

Click **Message Details** to view the key and value of the message.



Query by time

1. Log in to the CKafka console.
2. Select **Instance List** on the left sidebar, select a region, and click the **ID/Name** of the target instance to enter the instance details page.
3. On the instance details page, select **Topic Management** and click **More**> **Message Query** in the **Operation** column on the right of the target topic.
4. Select **Query by start time**, enter the partition ID and time, and click **Query** to view the message details.



Partition ID: Topic partition to which the message is sent.

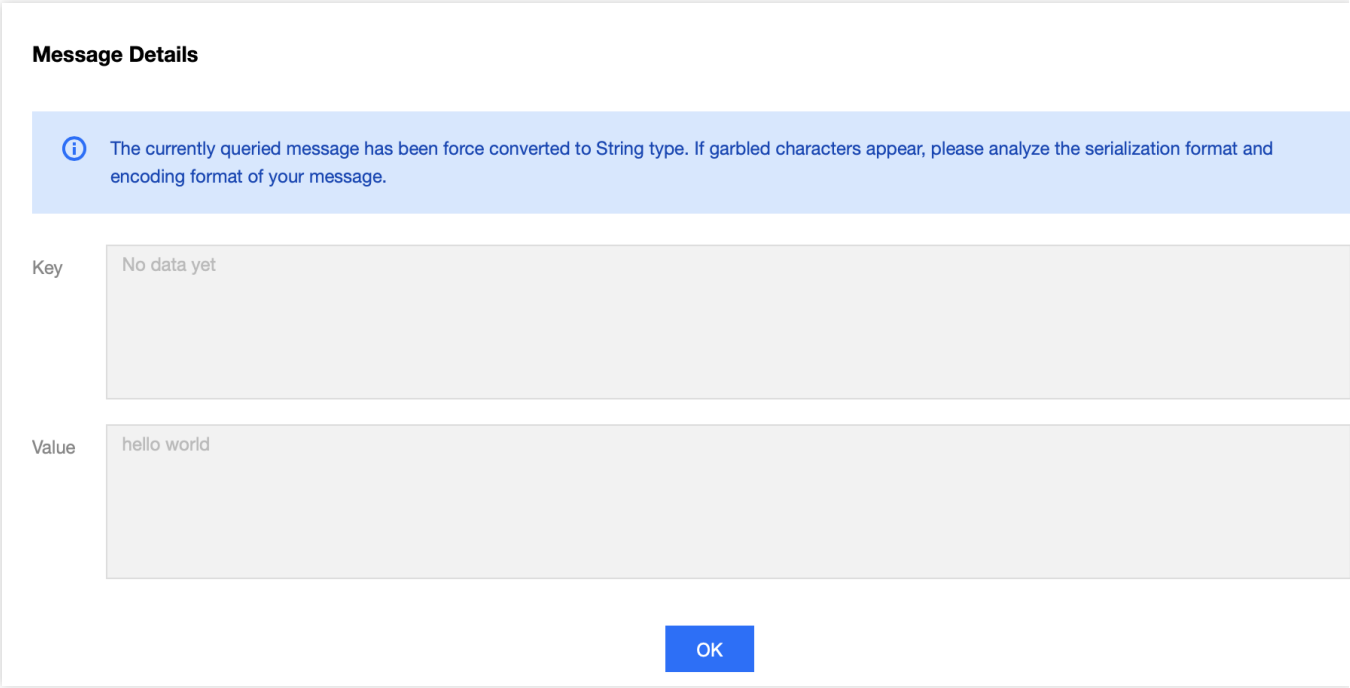
Offset: Consumer offset of the message.

Timestamp: `timestamp` in `ProducerRecord`.

Operation:

Click **Download Message** to download the message, including its headers, key, and value. Batch message downloading is supported.

Click **View Details** to view the details of the message.


Message Details
The currently queried message has been force converted to String type. If garbled characters appear, please analyze the serialization format and encoding format of your message.
Key
No data yet
Value
hello world
OK

Event Center

Last updated：2025-04-09 11:18:55

The event center manages, stores, analyzes, and displays the event data generated by CKafka in a unified manner. You can easily view the event data details there.

The events currently supported by CKafka include:

Category	Event Type
Instance events	Automatic disk capacity expansion Dynamic message retention policy for disks Automated partition balancing Manual partition balancing Kafka version upgrade Kernel minor version upgrade Upgrade configuration Downgrade configuration Public network bandwidth adjustment Routing policy change ACL policy change User addition and deletion Modify AZ Enable elastic bandwidth capability Disable elastic bandwidth capability
Diagnosis events	Scheduled instance diagnosis Instant instance diagnosis
Broker changes	Consumer group member heartbeat timeout Consumer group member update Consumer group rebalancing Leader switch Cluster node launch Cluster node removal

Instance events

Upgrade configuration

Downgrade configuration

Public network bandwidth adjustment

Upgrade from Standard Edition to Pro Edition

Routing policy change

ACL policy change

©2013-2025 Tencent Cloud International Pte. Ltd.

User addition and deletion
Automatic disk capacity expansion
Dynamic disk message retention policy
Partition balancing
Kafka version upgrade
Kernel minor version upgrade
AZ change
Diagnosis events
Scheduled instance diagnosis
Instant instance diagnosis
Broker changes
Consumer group member heartbeat timeout
Consumer group member update
Consumer group rebalancing
Leader switch
Cluster node launch
Cluster node removal
This document describes how to view event details and configure event notification rules in the CKafka console.

Entering the event center

1. Log in to the CKafka console.
2. Click **Instance Management** on the left sidebar and click the **ID** of the target instance to enter the instance details page.
3. Select the **Event Center** tab at the top of the page, set the time range (last 6 hours, last 7 days, last 30 days, or a custom time range), and select the target event type.

4. On the **Event Details** page, you can perform the following operations:
Click **View Details** in the **Operation** column to view the event details on the right.

Click **Configure Alarm Policy** in the **Operation** column to configure a notification template for the event and send event details to the target object.

Migration to Cloud
Migration Scheme Overview

Last updated：2024-01-09 14:49:39

Overview

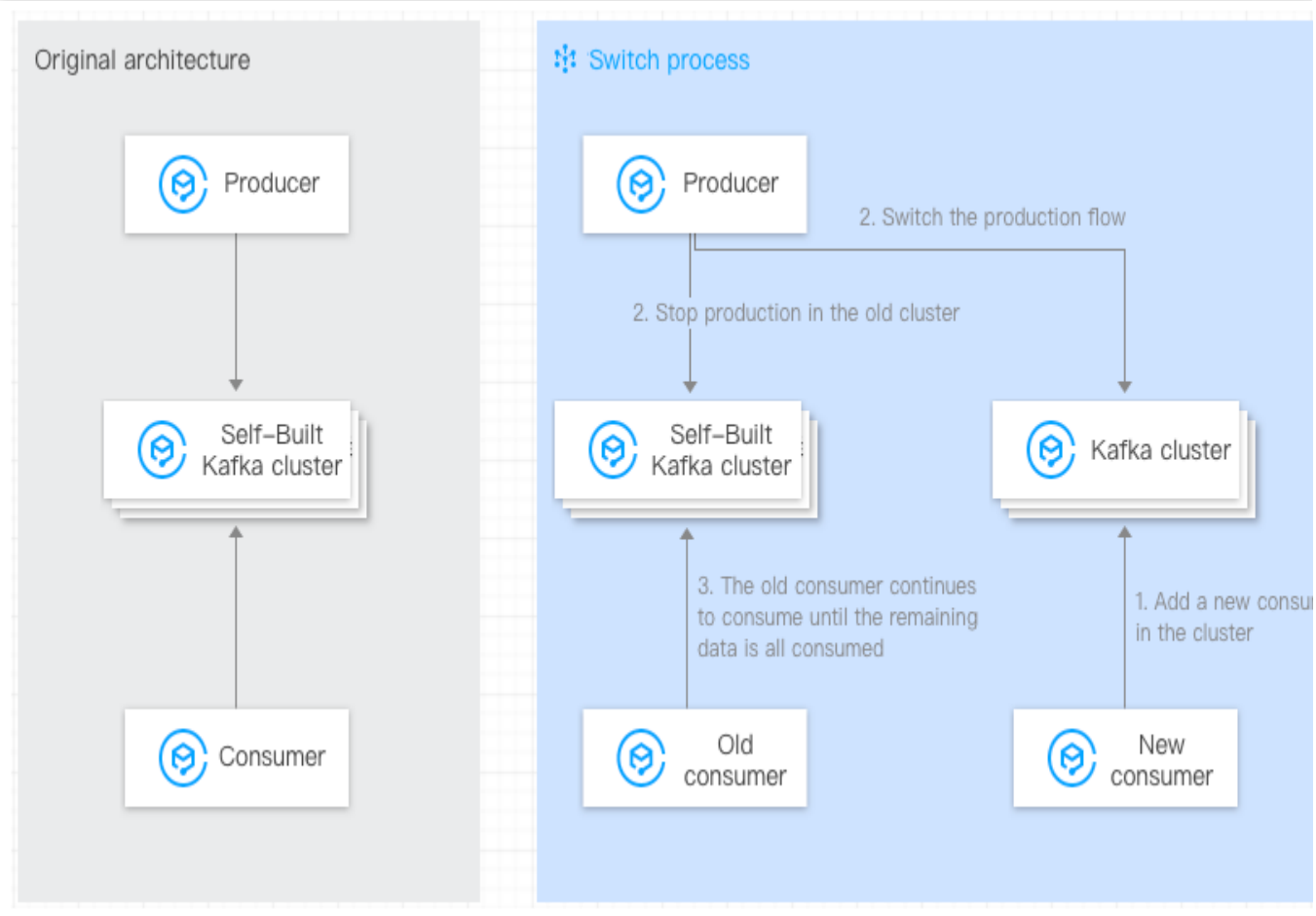
This document describes the schemes for migrating a self-built Kafka cluster to a CKafka cluster. You can choose an appropriate scheme according to your own business scenarios.

Scheme Description

Scheme 1. Single-Producer dual-consumer

This scheme is simple, clear, and easy to implement, with no data heap for smooth transition.

©2013-2025 Tencent Cloud International Pte. Ltd.



Scheme ideas:

1. Migrate the topic metadata.
2. Keep the original consumer in the self-built Kafka cluster intact.
3. Start a new consumer in CKafka and configure the bootstrap-server of the new CKafka cluster for consumption.
4. Wait for all consumers to listen on the new CKafka cluster.
5. Switch the production of the self-built cluster to the new CKafka cluster (by configuring the bootstrap-server of the new CKafka cluster).
6. Deactivate the original consumer in the self-built Kafka cluster after it finishes consuming the remaining data in the cluster.

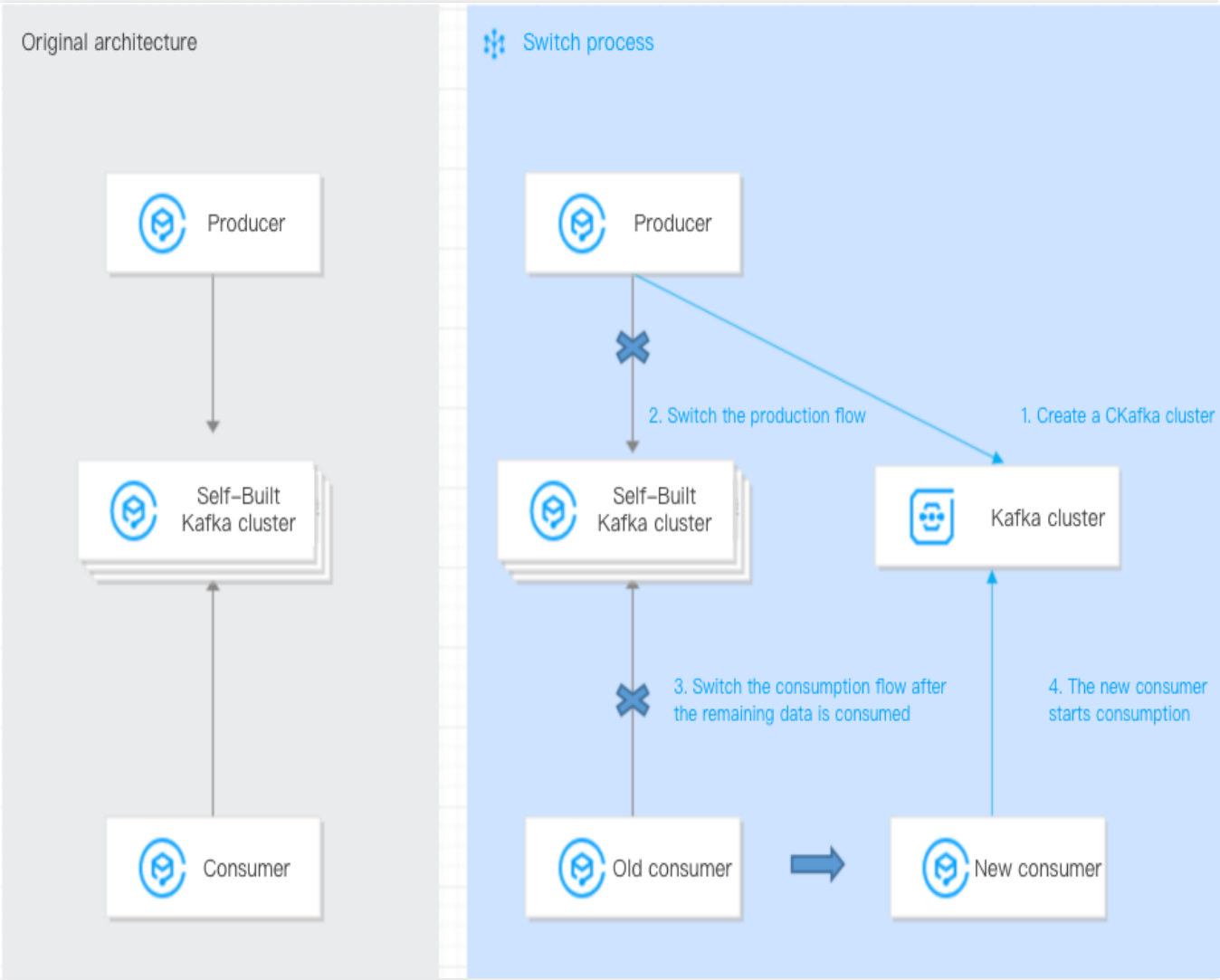
Pros and cons of the scheme:

Pros: the overall migration process is simple, clear, and easy to implement, with no data heap for smooth transition.

Cons: an additional consumer is required.

Scheme 2. Single-Producer single-consumer

This scheme is simple, clear, and easy to implement.



Scheme ideas:

1. Migrate the topic metadata.
2. Switch the production of the self-built Kafka cluster to the new CKafka cluster (by configuring the bootstrap-server of the new CKafka cluster).
3. Wait for the consumer in the self-built cluster to consume the remaining data.
4. Switch the old consumer to the new CKafka cluster for consumption (by configuring the bootstrap-server of the new CKafka cluster).

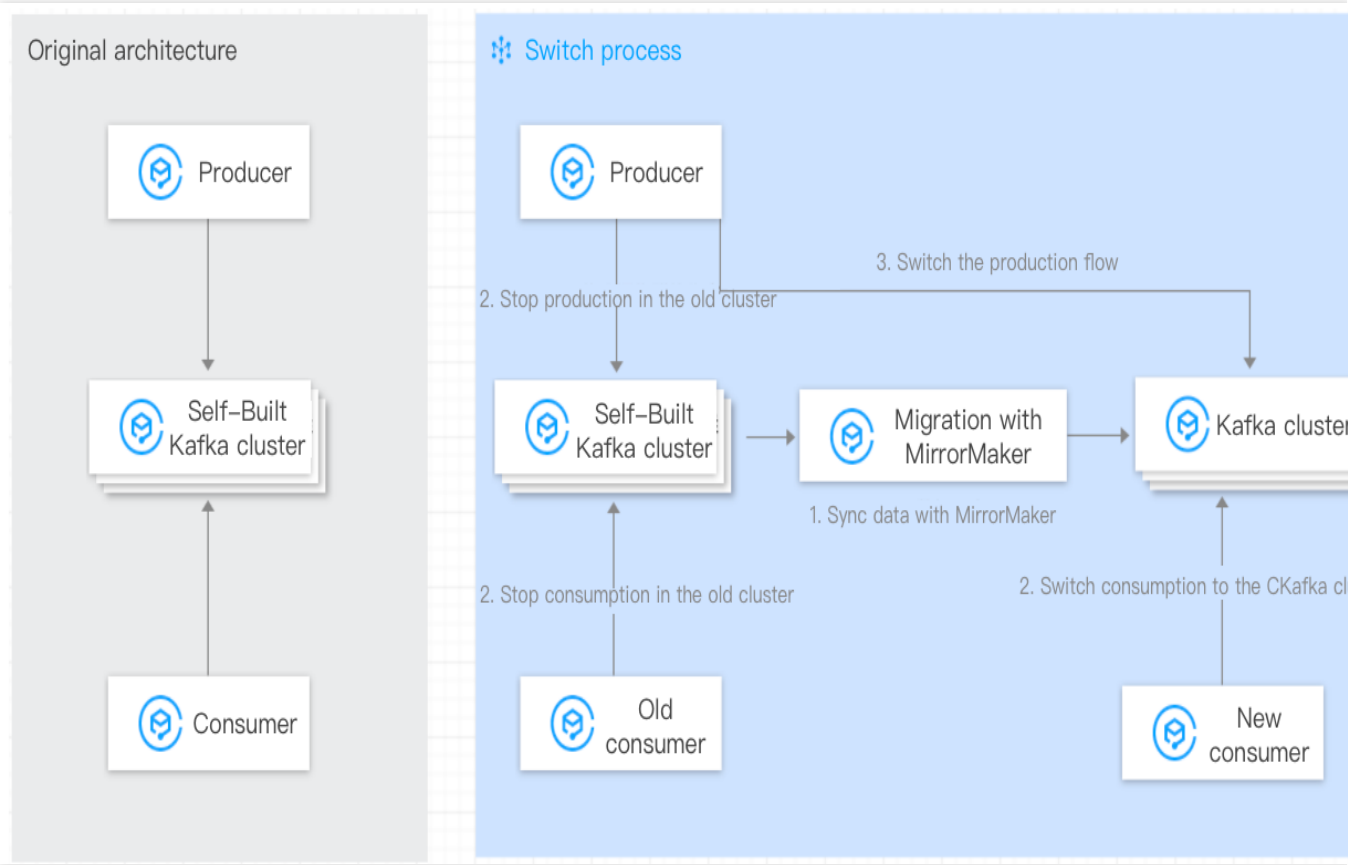
Pros and cons of the scheme:

Pros: the overall migration process is simple, clear, and easy to implement for smooth transition.

Cons: after the production is switched to the CKafka cluster, before the old consumer is switched to the CKafka cluster, there will be a certain amount of data heap in the CKafka cluster.

Scheme 3. Migration with MirrorMaker

This scheme migrates the existing data in the self-built Kafka cluster to CKafka.



Scheme ideas:

1. Migrate the topic metadata.
2. Keep the original consumer in the self-built Kafka cluster intact.
3. Start the data sync feature of MirrorMaker.
4. Wait for the data sync to complete, modify the consumer configuration, and switch the consumer.
5. Wait for the data sync to complete, modify the producer configuration, and switch the producer.
6. The migration is completed.

Pros and cons of the scheme:

Pros: the overall migration process is simple, clear, and easy to implement, where historical data can be synced to the CKafka cluster.

Cons: the consumer needs to consume data from the beginning after switch to the destination cluster, so consumption idempotency must be ensured.

Step 1. Purchasing CKafka Instance

Last updated：2024-01-09 14:49:39

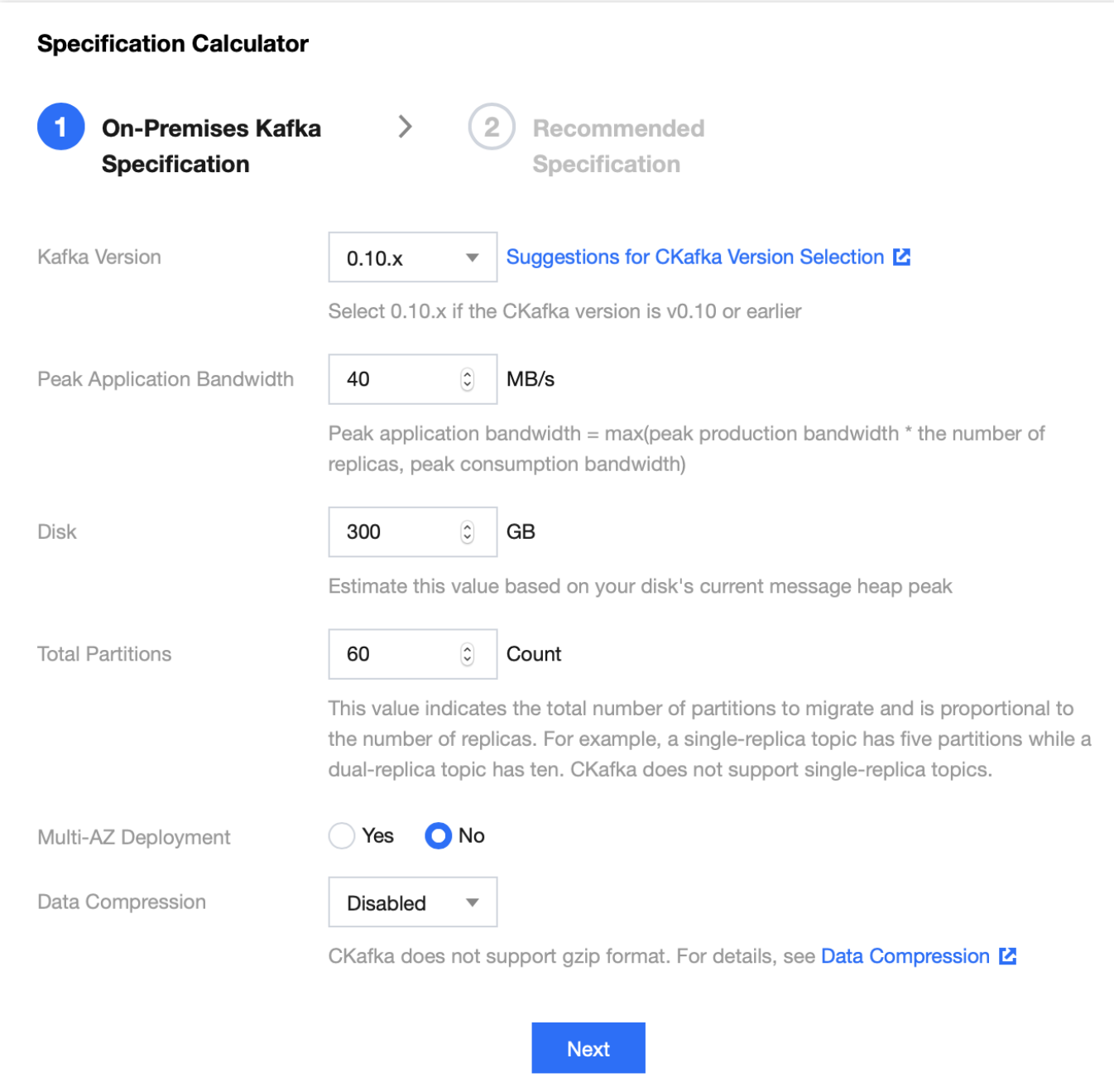
Overview

This document describes how to use the **Specification Calculator** in the CKafka console to find the appropriate CKafka instance specification for your self-built Kafka cluster that needs to be migrated to the cloud.

Directions

1. Log in to the CKafka console.
2. On the left sidebar, select **Migration to Cloud**, select the region you want to migrate to, and click **Specification Calculator**.
3. Enter the specification of your self-built Kafka cluster on the specification calculator page.

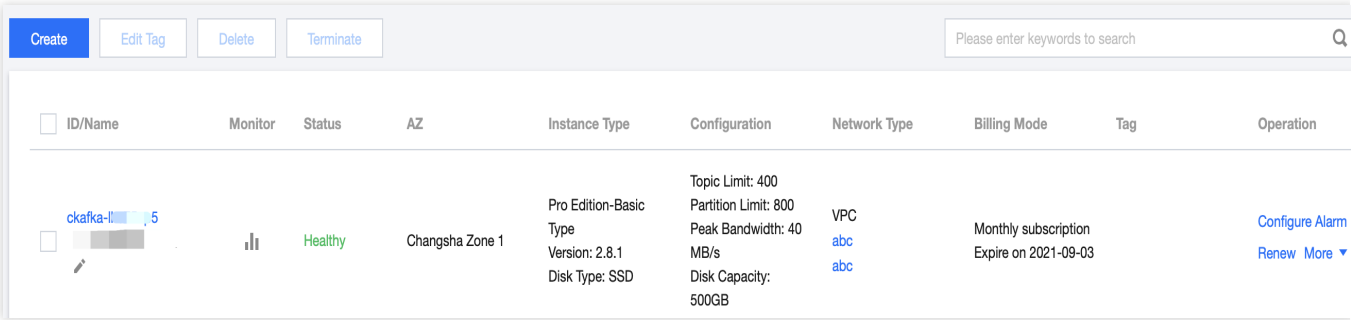
©2013-2025 Tencent Cloud International Pte. Ltd.



Parameter	Description
Kafka Version	Select the version of your self-built Kafka cluster. For more information on CKafka version selection, please see Suggestions for CKafka Version Selection.
Peak Business Bandwidth	Peak business bandwidth = max (production peak bandwidth * number of replicas, consumption peak bandwidth).
Disk	Estimate according to the current peak value of actual disk usage.
Total Partitions	Total number of partitions of the topic to be migrated. The number of replicas should be considered. For example, if one topic replica has 5 partitions, then two replicas have 10

	partitions in total. CKafka does not support single-replica topics.
Multi-AZ Deployment	Select whether to deploy in multiple AZs according to your business needs. For more information, see Multi-AZ Deployment.
Data Compression	CKafka does not support the Gzip compression format. For more information, see Data Compression.

4. Click **Next** to get the recommended CKafka instance specification.
5. Click **Buy Now** to redirect to the instance purchase page.
6. Confirm the purchase information, click **Buy Now**, wait 5–10 minutes, and you can see that the instance is created on the instance list page.



Step 2. Migrating Topic to Cloud

Last updated：2025-03-18 10:02:54

Overview

This document describes how to use the migration tool provided by CKafka to migrate topics from a self-built Kafka cluster to a CKafka instance.

Prerequisites

You have purchased a CKafka instance.
You have downloaded Python 2.x.

Directions

1. Download the migration tool and decompress it on a server that can connect to the broker and ZooKeeper of the self-built instance.
2. Enter the configuration parameters in the `ckafka-migrate.py` file.

Note:
Make sure that the networks of the server where the subsequent migration operation is performed, the CKafka instance, and the self-built Kafka cluster are interconnected.
The user corresponding to the key of TencentCloud API needs to have the write permission for CKafka. We recommend that you use the key pair of the root account.

```
   # your local broker ip:port list
# Broker list of the self-built instance ["broker1:port1","ip2:port2"]
bootstrapServers = ["$ip:$port"]

# your local zk ip:port list
# ZooKeeper list of the self-built instance ["zk1:port1","zk2:port2"]. This is
an optional configuration. If it is not configured, the source cluster
information will be obtained via `bootstrapServers`.
sourceZk = ["$ip:$port"]
# This parameter is required if ZooKeeper has enabled authentication. Format:
[("digest", "$user:$password")]
zkAuthData = []
# This parameter is required if the ZooKeeper path has a prefix. Sample value:
"/cluster1".
```

```
zkPathPrefix = ""

# your cloud instanceId
# CKafka instance ID "ckafka-xxx"
instanceId = "$yourinstanceId"
# topic regex,just migrate match topics
# Regular expression of the topic name. If it is not empty, only matched topics
will be migrated.
topicRegex = ""

# your secretId and secretKey
# Account key pair
secretId = "$yoursecretId"
secretKey = "$yoursecretKey"

# your cloud instance region
# CKafka instance region and code:
# Guangzhou: ap-guangzhou; Shanghai: ap-shanghai; Nanjing: ap-nanjing; Beijing:
ap-beijing; Chengdu: ap-chengdu; Chongqing: ap-chongqing;
# Hong Kong (China): ap-hongkong; Singapore: ap-singapore; Tokyo: ap-tokyo;
Silicon Valley: na-siliconvalley;
# Virginia: na-ashburn; Taipei (China): ap-taipei; Tianjin: ap-tianjin;
Shanghai Finance: ap-shanghai-fsi;
# Shenzhen Finance: ap-shenzhen-fsi; Shenzhen: ap-shenzhen; Frankfurt: eu-
frankfurt; Seoul: ap-seoul; Qingyuan: ap-qingyuan;
# Beijing Finance: ap-beijing-fsi; Bangkok: ap-bangkok; Changsha: ap-changsha-
ec; Jakarta: ap-jakarta;
# Hangzhou ec: ap-hangzhou-ec; São Paulo: sa-saopaulo; Shanghai Auto-Driving
Cloud: ap-shanghai-adc; Wuhan ec: ap-wuhan-ec, Xi'an ec: ap-xian-ec
region = "ap-tokyo"

# If you make sure the migrate topic List, please modify checkFlag = 1
# Check flag. If it is 0, only the list of topics to be migrated will be
displayed, with no migration performed. You can set it to 0 first to check the
list of topics. After confirming that everything is correct, you can change it
to 1 to start migration.
# 0: List the topics to be migrated and then stop the script
# 1: List the topics to be migrated and then start migration
checkFlag = 0

# force transfer your cloud-topic config to migrate
# If the value is 0, a local topic will not be migrated to CKafka if its
attributes do not match those of CKafka. If the value is 1, the topic attribute
values out of the value range of CKafka will be forcibly converted to the
closest values within the value range.
# For example, if the CKafka topic supports only 1, 2 or 3 replicas, but a
local topic has 5 replicas, it will not be migrated to CKafka. If you set the
```

```
parameter to 1, then a 3-replica topic, which is closest to the original number
of replicas (5), will be created for the CKafka topic.
# 0: Skip the unmatched topic or topic attributes if the numbers of replicas or
attributes of the local and CKafka topics do not match.
# 1: Forcibly migrate all topics to CKafka if the numbers of replicas or
attributes of the local and CKafka topics do not match. Then, modify CKafka
topic attributes to be as close as possible to those self-built ones, without
modifying any data in the local self-built Kafka cluster.
force = 0
```

Parameter	Description
bootstrapServers	Broker list of the self-built instance ["ip1:port1","ip2:port2"]
sourceZk	ZooKeeper list of the self-built instance ["zk1:port1","zk2:port2"]. This is an optional configuration. If it is not configured, the source cluster information will be obtained via `bootstrapServers` .
zkAuthData	This parameter is required if ZooKeeper has enabled authentication. Format: [("digest", "$user:$password")]
zkPathPrefix	This parameter is required if the ZooKeeper path has a prefix. Sample value: "/cluster1".
instanceId	ID of the CKafka instance you purchased in Step 1. Purchasing CKafka Instance. You can copy the ID on the **Instance List** page in the CKafka console.
secretId	Account key pair – ID
secretKey	Account key pair – password
region	The deployment region you selected in Step 1. Purchasing CKafka Instance. The region codes are available in the script annotation.
checkFlag	Check flag. If it is 0, only the list of topics to be migrated will be displayed and migration will not start; otherwise, the topics will start to be migrated.
topicRegex	Regular expression of the topic name. If it is empty, all topics will be migrated; otherwise, only matched topics will be migrated.
force	It specifies whether to forcibly migrate topics. If the value is 0, a local topic will not be migrated to CKafka if its attributes do not match those of CKafka. If the value is 1, the topic attribute values out of the value range of CKafka will be forcibly converted to the closest values within the value range.

3. Set the `checkFlag` parameter in `ckafka-migrate.py` to 0, run `python ckafka-migrate.py` to execute the script, and check the list of topics that need to be migrated based on the output result.

Note:

If some self-built topics are missing, this may be because their names are invalid, or the numbers of replicas or attribute values of the self-built and CKafka topics do not match.

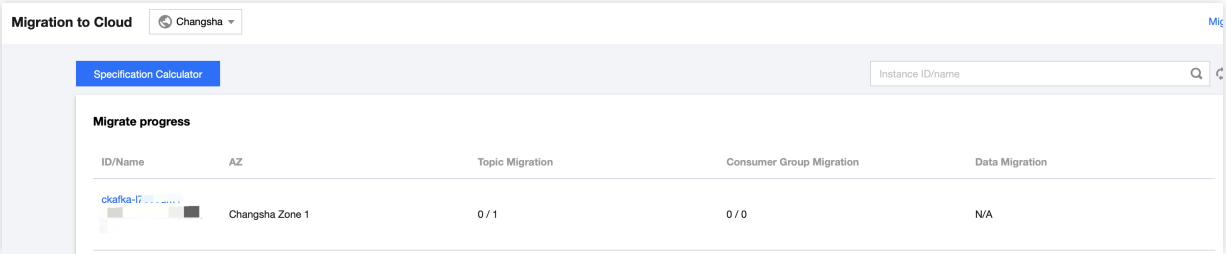
```
[root@VM-0-21-centos ~/migrateCloud/migrateToCkafkaTool]# python ckafka-migrate.py
check topic List to migrate: ["test1", "test0", "test108", "test109", "test5", "test4", "test7", "test6", "test102", "test103", "test100",
 "test101", "test106", "test107", "test104", "test105", "test182", "test183", "test180", "test181", "test186", "test187", "test184", "test
185", "test188", "test189", "test46", "test47", "test44", "test45", "test42", "test43", "test40", "test41", "test48", "test49", "test8", "
test179", "test178", "test177", "test176", "test175", "test174", "test173", "test172", "test171", "test170", "test55", "test54", "test57",
 "test56", "test51", "test50", "test53", "test52", "test59", "test58", "test168", "test169", "test164", "test165", "test166", "test167", "
test160", "test161", "test162", "test163", "test60", "test61", "test62", "test63", "test64", "test65", "test66", "test67", "test68", "test
69", "test2", "test159", "test158", "test151", "test150", "test153", "test152", "test155", "test154", "test157", "test156", "test79", "tes
t78", "test77", "test76", "test75", "test74", "test73", "test72", "test71", "test70", "test146", "test147", "test144", "test145", "test142
", "test143", "test140", "test141", "test148", "test149", "test82", "test83", "test80", "test81", "test86", "test87", "test84", "test85",
"test88", "test89", "test3", "test133", "test132", "test131", "test130", "test137", "test136", "test135", "test134", "test139", "test138",
 "test91", "test90", "test93", "test92", "test95", "test94", "test97", "test96", "test99", "test98", "test35", "test19", "test18", "test11
", "test10", "test13", "test12", "test15", "test14", "test17", "test16", "test120", "test121", "test122", "test123", "test124", "test125",
 "test126", "test127", "test128", "test129", "test24", "test25", "test26", "test27", "test20", "test21", "test22", "test23", "test28", "te
st29", "test", "test119", "test118", "test115", "test114", "test117", "test116", "test111", "test110", "test113", "test112", "test195", "t
est194", "test197", "test196", "test191", "test190", "test193", "test192", "test199", "test198", "test33", "test32", "test31", "test30", "
test37", "test36", "test9", "test34", "test39", "test38"]
```

4. Set the `checkFlag` parameter in `ckafka-migrate.py` to 1 and run `python ckafka-migrate.py` to execute the script to start migrating the topics.

```
[root@VM-0-21-centos ~/migrateCloud/migrateToCkafkaTool]# python ckafka-migrate.py

preper to migrate these topics ["test1", "test0", "test108", "test109", "test5", "test4", "test7", "test6", "test102", "test103", "test100
", "test101", "test106", "test107", "test104", "test105", "test182", "test183", "test180", "test181", "test186", "test187", "test184", "te
st185", "test188", "test189", "test46", "test47", "test44", "test45", "test42", "test43", "test40", "test41", "test48", "test49", "test8",
 "test179", "test178", "test177", "test176", "test175", "test174", "test173", "test172", "test171", "test170", "test55", "test54", "test57
", "test56", "test51", "test50", "test53", "test52", "test59", "test58", "test168", "test169", "test164", "test165", "test166", "test167",
 "test160", "test161", "test162", "test163", "test60", "test61", "test62", "test63", "test64", "test65", "test66", "test67", "test68", "te
st69", "test2", "test159", "test158", "test151", "test150", "test153", "test152", "test155", "test154", "test157", "test156", "test79", "t
est78", "test77", "test76", "test75", "test74", "test73", "test72", "test71", "test70", "test146", "test147", "test144", "test145", "test1
42", "test143", "test140", "test141", "test148", "test149", "test82", "test83", "test80", "test81", "test86", "test87", "test84", "test85"
, "test88", "test89", "test3", "test133", "test132", "test131", "test130", "test137", "test136", "test135", "test134", "test139", "test138
", "test91", "test90", "test93", "test92", "test95", "test94", "test97", "test96", "test99", "test98", "test35", "test19", "test18", "test
11", "test10", "test13", "test12", "test15", "test14", "test17", "test16", "test120", "test121", "test122", "test123", "test124", "test125
", "test126", "test127", "test128", "test129", "test24", "test25", "test26", "test27", "test20", "test21", "test22", "test23", "test28", "
test29", "test", "test119", "test118", "test115", "test114", "test117", "test116", "test111", "test110", "test113", "test112", "test195",
"test194", "test197", "test196", "test191", "test190", "test193", "test192", "test199", "test198", "test33", "test32", "test31", "test30",
 "test37", "test36", "test9", "test34", "test39", "test38"] to cloud instance ckafka-lxmap3v4 ......

create Migreate to Cloud Task Success!

1/201 migrate topic test1 to ckafka-lxmap3v4 success,topicId:topic-9c9q3i2i


2/201 migrate topic test0 to ckafka-lxmap3v4 success,topicId:topic-o57vfbw2
```

5. Log in to the CKafka console, view the task list on the **Migration to Cloud** page, and wait for the topics to be completely migrated.

The task list is as shown below:



The result of successful migration is as shown below:

```
199/201 migrate topic test34 to ckafka-lxmap3v4 success,topicId:topic-a4bo0ske

200/201 migrate topic test39 to ckafka-lxmap3v4 success,topicId:topic-bv7pezvi

201/201 migrate topic test38 to ckafka-lxmap3v4 success,topicId:topic-m96t1b0e

migrate topics to cloud finished!
```

Step 3. Migrating Data to Cloud
Scheme 1: Single-Producer Dual-Consumer Migration

Last updated：2024-01-09 14:49:39

Overview

This document describes how to use the single-producer dual-consumer scheme to migrate data from a self-built Kafka cluster to a CKafka cluster.

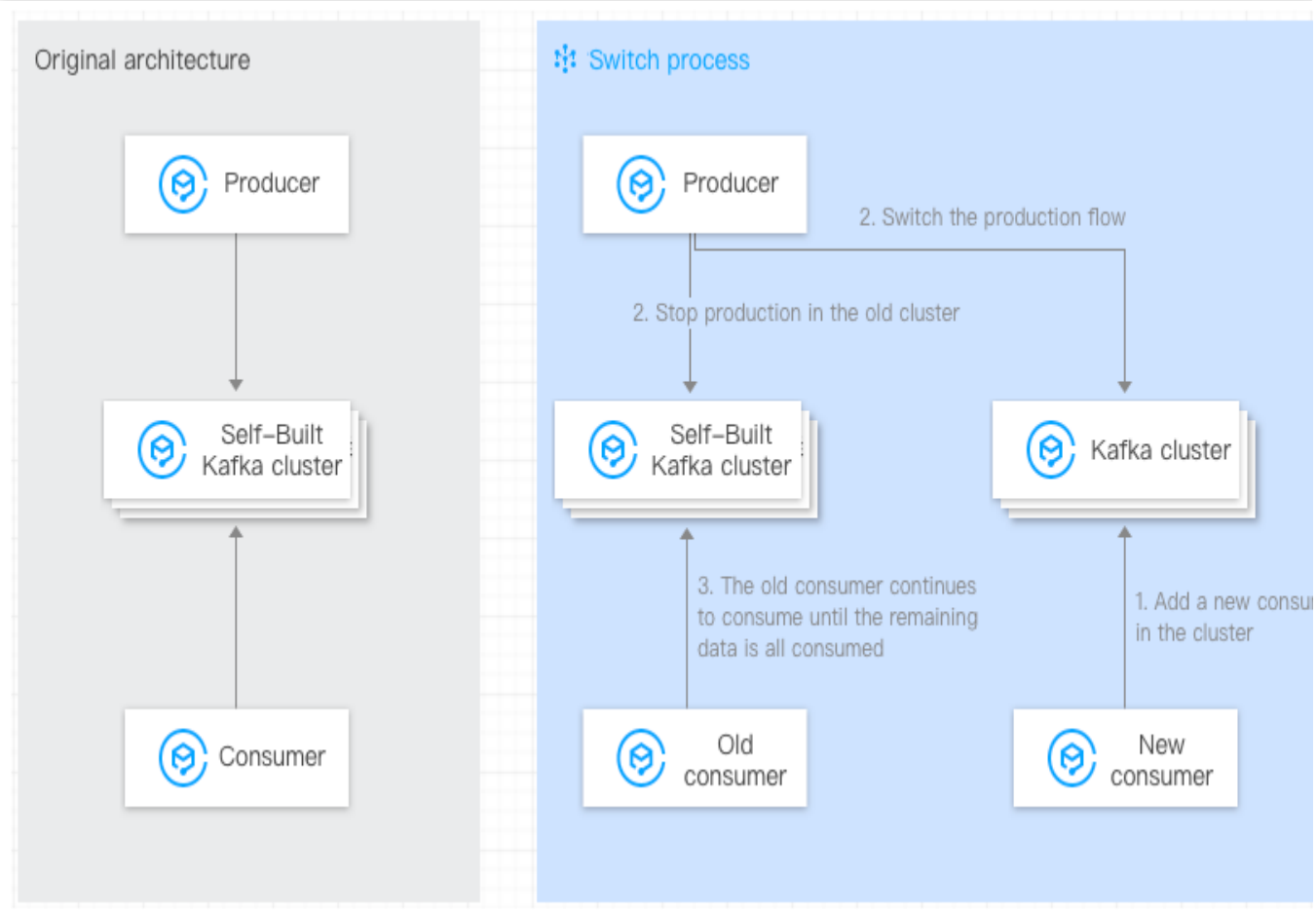
Prerequisites

You have already purchased a CKafka instance.
You have already migrated the topic to CKafka.

Directions

If your requirement for message ordering is not high, you can migrate the data while it is consumed by multiple consumers in parallel.
The single-producer dual-consumer scheme is simple, clear, and easy to implement, with no data heap for smooth transition; however, it requires adding a new consumer.
The migration steps are as follows:

©2013-2025 Tencent Cloud International Pte. Ltd.



1. Keep the old consumer intact, start a new consumer in CKafka, and configure the bootstrap-server of the new CKafka cluster for consumption.

You need to configure the accessed network of the CKafka instance as the IP in `--bootstrap-server` by copying the information in the **Network** column in the **Access Mode** section on the **Instance Details** page in the console.

```
./kafka-console-consumer.sh --bootstrap-server xxx.xxx.xxx.xxx:9092 --from-beginning --new-consumer --topic topicName --consumer.config ../config/consumer.properties
```

2. Switch the production flow so that the producer produces data to the CKafka instance.

Change the IP in the `broker-list` to the accessed network of the CKafka instance and `topicName` to the topic name in the CKafka instance:

```
./kafka-console-producer.sh --broker-list xxx.xxx.xxx.xxx:9092 --topic topicName
```

3. The original consumer does not need to be configured and can continue to consume the data in your self-built Kafka cluster. After such data is all consumed, the migration is completed.

Note:

The above commands are test commands. In actual business operations, just modify the broker address configured for the corresponding application and then restart the application.

©2013-2025 Tencent Cloud International Pte. Ltd.

Scheme 2: Single-Producer Single-Consumer Migration

Last updated：2024-01-09 14:49:39

Overview

This document describes how to use the single-producer single-consumer scheme to migrate data from a self-built Kafka cluster to a CKafka cluster.

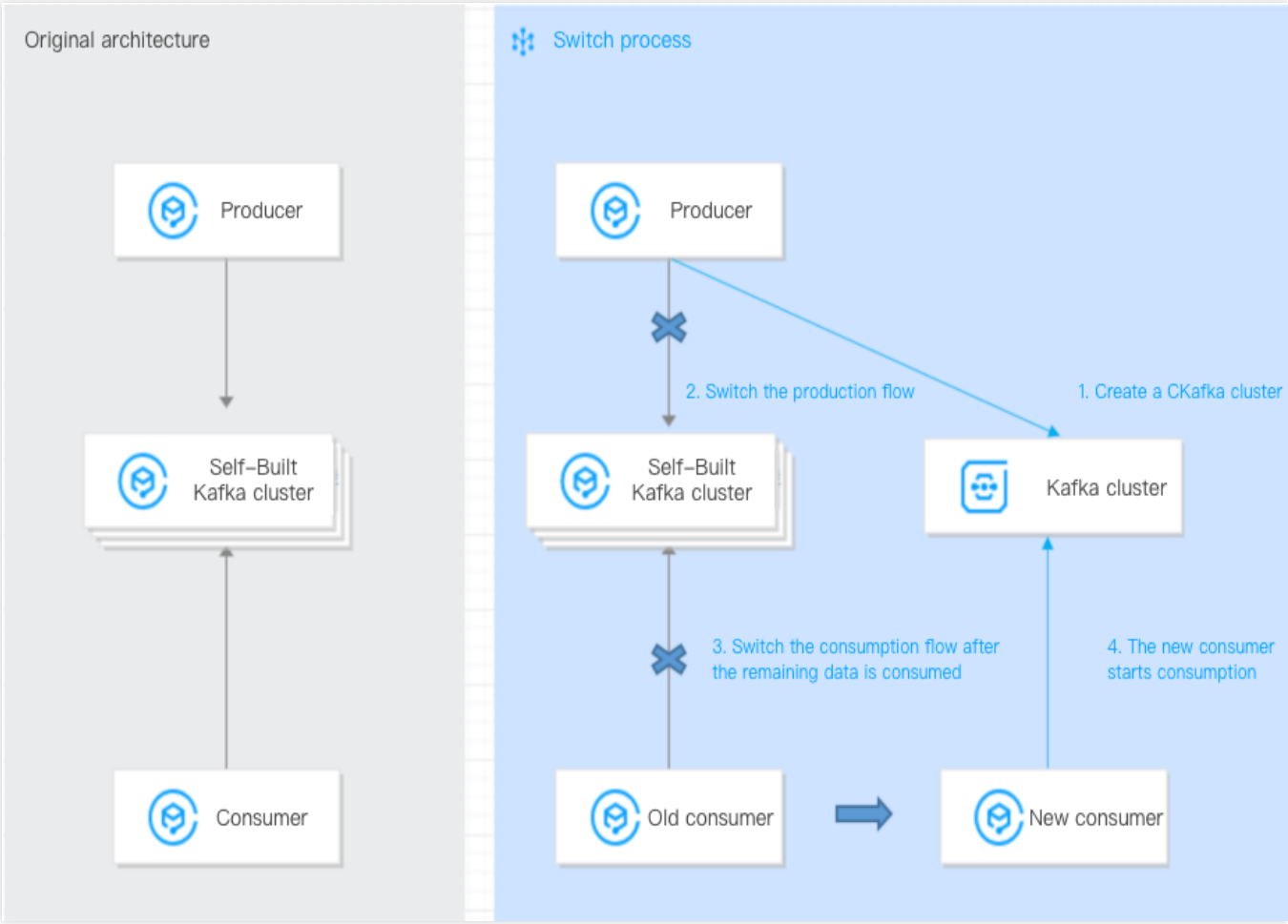
Prerequisites

You have already purchased a CKafka instance.
You have already migrated the topic to CKafka.

Directions

The prerequisite for guaranteeing message ordering is to strictly limit data consumption to only one consumer. Therefore, timing of the migration is vital.
The single-producer single-consumer scheme is simple, clear, and easy to implement; however, after the production is switched to the new cluster, before the old consumer is switched to the new cluster, there will be a certain amount of data heap in the new cluster.
The migration steps are as follows:

©2013-2025 Tencent Cloud International Pte. Ltd.



1. Switch the production flow so that the producer produces data to the CKafka instance.

Configure the accessed network of the CKafka instance as the IP in `broker-list` by copying the information in the **Network** column in the **Access Mode** section on the **Instance Details** page in the console, and change the `topicName` to the topic name in the CKafka instance.

```
./kafka-console-producer.sh --broker-list xxx.xxx.xxx.xxx:9092 --topic topicName
```

2. The original consumer does not need to be configured and can continue to consume the data in your self-built Kafka cluster until all data is consumed.

3. When the consumption by the original consumer is completed, switch to the new CKafka cluster for consumption through the following configuration (let only one consumer consume the data to guarantee message ordering). If you add a new consumer, you need to configure the accessed network of the CKafka instance as the IP in `--bootstrap-server`.

Note:

If the original consumer is a CVM instance, it can continue to consume the data.

```
./kafka-console-consumer.sh --bootstrap-server xxx.xxx.xxx.xxx:9092 --from-beginning --new-consumer --topic topicName --consumer.config
```

```
../config/consumer.properties
```

4. The consumer continues to consume data in the CKafka cluster after switch, and the migration is completed (if the original consumer is a CVM instance, it can continue to consume the data).

Note:

The above commands are test commands. In actual business operations, just modify the broker address configured for the corresponding application and then restart the application.

Scheme 3: Migration with MirrorMaker

Last updated：2024-01-09 14:49:40

Overview

This document describes how to use MirrorMaker to migrate data from a self-built Kafka cluster to a CKafka cluster. Kafka MirrorMaker can back up data in a self-built Kafka cluster to a CKafka cluster in the following way: MirrorMaker can use a consumer to consume messages in the self-built Kafka cluster and then use a producer to transfer the message data to the CKafka cluster. After this, you can transfer the production and consumption configurations of the client to the accessed network of the CKafka cluster so as to complete data migration.

Prerequisites

You have already purchased a CKafka instance.
You have already migrated the topic to CKafka.

Directions

1. Download MirrorMaker and decompress it on the local server.

Note:

This document takes kafka_2.11-1.1.1.tgz as an example:

2. Configure the `consumer.properties` file.

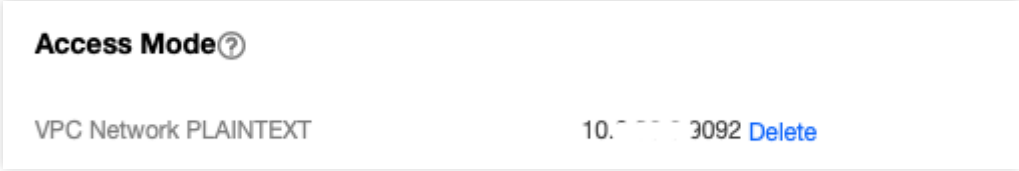
```
# list of brokers used for bootstrapping knowledge about the rest of the cluster
# format: host1:port1,host2:port2 ...
bootstrap.servers=localhost:9092

# consumer group id
group.id=test-consumer-group

partition.assignment.strategy=org.apache.kafka.clients.consumer.RoundRobinAssignor
# What to do when there is no initial offset in Kafka or if the current
# offset does not exist any more on the server: latest, earliest, none
#auto.offset.reset=
```

Parameter	Description
bootstrap.servers	List of broker access points of the self-built cluster.

group.id	ID of the consumer group used during data migration. It must be different from the r groups in the self-built cluster.
partition.assignment.strategy	Partition assignment policy, such as `partition.assignment.strategy=org.apache.kafka.clients.con`

3. Configure the `producer.properties` file.

```
# list of brokers used for bootstrapping knowledge about the rest of the cluster
# format: host1:port1,host2:port2 ...
bootstrap.servers=localhost:9092

# specify the compression codec for all data generated: none, gzip, snappy, lz4
compression.type=none
```

Parameter	Description
bootstrap.servers	Accessed network of the CKafka instance, which can be copied from the **Network** column in t **Access Mode** section on the **Instance Details** page in the console. 
compression.type	Data compression type. CKafka does not support the Gzip compression format.

4. Start MirrorMaker in the `.bin` directory to start migration.

```
sh bin/kafka-mirror-maker.sh --consumer.config config/consumer.properties --
producer.config config/producer.properties --whitelist topicName
```

Note:

`whitelist` is a regular expression, and topics whose names hit it will be migrated.

5. Run `kafka-consumer-groups.sh` in the `bin` directory to view the consumption progress of the self-built cluster.

```
bin/kafka-consumer-groups.sh --new-consumer --describe --bootstrap-server self-
built cluster access point --group test-consumer-group
```

Note:

`group` is the ID of the consumer group used during data migration.

```
[root@VM-0-36-centos ~]# /usr/local/services/kafka_2.11-1.1.1/bin/kafka-consumer-groups.sh --bootstrap-server 10.0.0.36:9092 --descri
group mm1-migrate-ckafka
Note: This will not show information about old Zookeeper-based consumers.

TOPIC           PARTITION  CURRENT-OFFSET  LOG-END-OFFSET  LAG            CONSUMER-ID                                          
          CLIENT-ID
test26          0          -               461             -              mm1-migrate-ckafka-0-598a37fa-d5f1-4ad4-ba7c-579b8c7a6d5e /
.0.36     mm1-migrate-ckafka-0
test193         0          -               460             -              mm1-migrate-ckafka-0-598a37fa-d5f1-4ad4-ba7c-579b8c7a6d5e /
.0.36     mm1-migrate-ckafka-0
test89          0          -               461             -              mm1-migrate-ckafka-0-598a37fa-d5f1-4ad4-ba7c-579b8c7a6d5e /
.0.36     mm1-migrate-ckafka-0
test114         0          -               461             -              mm1-migrate-ckafka-0-598a37fa-d5f1-4ad4-ba7c-579b8c7a6d5e /
.0.36     mm1-migrate-ckafka-0
test1           3          -               2251            -              mm1-migrate-ckafka-0-598a37fa-d5f1-4ad4-ba7c-579b8c7a6d5e /
.0.36     mm1-migrate-ckafka-0
```

Subsequent Operations

After data migration, transfer the production and consumption configurations of the client to the access point of the CKafka cluster.

Scheme 4: Migrating Unconsumed Data

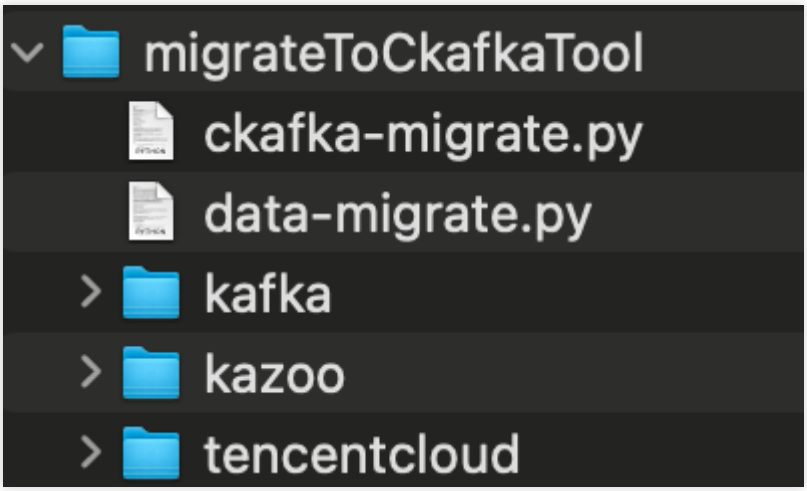
Last updated：2024-01-09 14:49:39

Overview

This document describes how to migrate unconsumed messages in a source cluster to the topics of a target cluster during the migration to the cloud.

Prerequisites

1. The source cluster has stopped message production and consumption.
2. The source cluster retains the unconsumed messages long enough to prevent them from being automatically deleted when they expire during the migration.
3. The migration script must be written in Python 2.x series on a version later than v2.7.1, preferably v2.7.5.
4. You have downloaded the migration tool migrateToCkafkaTool. Then go to the `migrateToCkafkaTool` directory, modify the configurations of the `data-migrate.py` file, and run `python data-migrate.py` .



How It Works

The script first scans the list of all consumer groups in the source cluster to get the list of topics (with unconsumed messages) to which the consumer has subscribed. Then the script will get the **submitted offset** and **end offset** of these topics (if a topic is subscribed to by multiple consumer groups, the script will get the smallest submitted offset). After consuming messages between the two offsets, the script will produce these messages to the corresponding topic partitions of the target cluster.

Directions

1. Create a topic in the target cluster

In the following example, the source cluster is `ckafka-47bd7goz` and the target cluster is `ckafka-kzamzogr`. We have created four topics with the same partition count as the source cluster: `test1`, `test2`, `test3`, and `test4` in the target cluster.

In the source cluster `ckafka-47bd7goz`, there are two consumer groups named `test123-group` and `test34-group`. The former subscribes to topics `test1`, `test2`, and `test3`, and the latter, `test3` and `test4`.

2. Download the migration toolkit

After downloading the migration tool, open the script and enter the address configurations of the source and target clusters. Set the value of `checkFlag` to `0` and run the script to check the topics to be migrated and their offsets.

```
# your local broker ip:port list
# Broker list of the self-built instance ["broker1:port1","ip2:port2"]
bootstrapServers = ["[redacted]"]   → Source instance

# your cloud ckafka instance address
# Cloud instance address "x.x.x.x:9092"
instanceAddress = "[redacted]"   → Target instance

# A dedicated group for migration data, default migrateLocalToCkafkaGroup, can be customized but should not have
# the same name as the existing consumer group in the cluster, do not delete it after the self-built cluster is
# generated, otherwise the migration progress will be lost
# A dedicated consumer group for data migration, which is `migrateLocalToCkafkaGroup` by default and can be customized. The name of the custom consumer group cannot be the same as that of any existing co
  group. Once you create a custom consumer group in the self-built cluster, don't delete it; otherwise, migration progress data will get lost. migrateGroupId = "migrateLocalToCkafkaGroup"

# topic regex,just migrate match topics
# Topic name regex, which is left empty by default, indicating that all topics will be migrated; otherwise, only topics that match the regex will be migrated.
topicRegex = ""

# The mapping between the local topic and the cloud topic, if not set, the default local and cloud have the same name
# The mapping between the local and cloud topics. If it is not set, the local and cloud topics will have the same name.
# example: localTopicAndCloudTopicMap = {"topic1-local": "topic1-cloud", "topic2-local": "topic2-cloud"}
localTopicAndCloudTopicMap = {}

# if you make sure the migrate topic List,please modify checkFlag = 1
# Check flag. If it is set to 0, the list of topics to be migrated will only be displayed but not migrated. You can set it to 0 first to check the topics to be migrated, confirm that everything is correct, and change it to 1 to start migration.
# 0: List the partition offset data of the topics to be migrated and stop the script.
# 1: List the partition offset data of the topics to be migrated and start migration.
checkFlag = 0   → Set the value to 0 first. If the script runs normally, set the value to 1 and run the script again.

logFileName = "data-migrate"
```

After the script is run, some information will be output, and a text log will be written to the current directory.

```
ckafka-migrate.py  data-migrate_220914_12:09:56.log  data-migrate.py  kafka  kazoo  tencentcloud
```

3. Check the output information

Check the **Prepare to migrate** information, which is the offset information of the topics to be migrated, based on the output information displayed on the screen or the text log file.

```
----------------------------Topic:[test3]  Partition:(0)  migrate------------------------

Prepare to migrate local-topic:[test3] partition:0 startOffset:187800 ~ endOffset:264322 data to cloud-topic:[test3]



----------------------------Topic:[test2]  Partition:(0)  migrate------------------------

Prepare to migrate local-topic:[test2] partition:0 startOffset:1873 ~ endOffset:2665 data to cloud-topic:[test2]



----------------------------Topic:[test2]  Partition:(1)  migrate------------------------

Prepare to migrate local-topic:[test2] partition:1 startOffset:1860 ~ endOffset:2621 data to cloud-topic:[test2]



----------------------------Topic:[test2]  Partition:(2)  migrate------------------------

Prepare to migrate local-topic:[test2] partition:2 startOffset:1830 ~ endOffset:2591 data to cloud-topic:[test2]
```

For example, you can check the subscription information of the topic `test3` in the source cluster. It is subscribed to by the consumer groups `test123-group` and `test34-group` at the same time.
Theoretically, if a topic is subscribed to by multiple consumer groups, it will be synced from the smallest submitted offset, which is 187800 in this example as expected.

```
----------------------------Topic:[test3]  Partition:(0)  migrate------------------------

Prepare to migrate local-topic:[test3] partition:0 startOffset:187800 ~ endOffset:264322 data to cloud-topic:[test3]
```

If any messages in topic `test1` of the source cluster have expired and the consumer group has a submitted offset falling between expired messages, the topic `test1` will be synced from the smallest offset of non-expired messages.

Let's say you want to migrate the data from partition 0 of `test1`. The script will prompt you that messages have expired when the smallest offset (5226) of non-expired messages exceeds the offset (3713) submitted by the consumer group. Therefore, the offset 5226 becomes the start offset of the migration task and the largest offset of partition 0, making it impossible to migrate any messages from this partition. In this case, you will see a prompt saying `skip migrate...`, indicating that the migration of data in this partition will be skipped.

```
-------------------------Topic:[test1]  Partition:(0)  migrate----------------------

the local-topic:[test1] partition:0 topic_begionOffset:5226 large than group_startOffset:3713,and reset startOffset to 5226

skip migrate local-topic:[test1] partition:0 startOffset:5226 ~ endOffset:5226 data to cloud-topic:[test1],beacuse startOffset is equal or more than endOffs



--------------------------Topic:[test1]  Partition:(1)  migrate----------------------

the local-topic:[test1] partition:1 topic_begionOffset:5340 large than group_startOffset:3784,and reset startOffset to 5340

skip migrate local-topic:[test1] partition:1 startOffset:5340 ~ endOffset:5340 data to cloud-topic:[test1],beacuse startOffset is equal or more than endOffs
```

4. Start the migration

After checking that the output information is correct in the previous step, set the value of `checkFlag` to 1 and start the migration.

```
---------------------------Topic:[test3]  Partition:(0)  migrate-----------------------


Prepare to migrate local-topic:[test3] partition:0 startOffset:187800 ~ endOffset:264322 data to cloud-topic:[t
begion migrate local-topic:[test3] partition:0 startOffset:187800 ~ endOffset:264322 data to cloud-topic:[test3
finish migrate local-topic:[test3] partition:0 startOffset:187800 ~ endOffset:264322 data to cloud-topic:[test3](
---------------------------Topic:[test2]  Partition:(0)  migrate-----------------------


Prepare to migrate local-topic:[test2] partition:0 startOffset:1873 ~ endOffset:2665 data to cloud-topic:[test2
begion migrate local-topic:[test2] partition:0 startOffset:1873 ~ endOffset:2665 data to cloud-topic:[test2]
finish migrate local-topic:[test2] partition:0 startOffset:1873 ~ endOffset:2665 data to cloud-topic:[test2]
---------------------------Topic:[test2]  Partition:(1)  migrate-----------------------


Prepare to migrate local-topic:[test2] partition:1 startOffset:1860 ~ endOffset:2621 data to cloud-topic:[test2
begion migrate local-topic:[test2] partition:1 startOffset:1860 ~ endOffset:2621 data to cloud-topic:[test2]
finish migrate local-topic:[test2] partition:1 startOffset:1860 ~ endOffset:2621 data to cloud-topic:[test2]
---------------------------Topic:[test2]  Partition:(2)  migrate-----------------------


Prepare to migrate local-topic:[test2] partition:2 startOffset:1830 ~ endOffset:2591 data to cloud-topic:[test2
begion migrate local-topic:[test2] partition:2 startOffset:1830 ~ endOffset:2591 data to cloud-topic:[test2]
finish migrate local-topic:[test2] partition:2 startOffset:1830 ~ endOffset:2591 data to cloud-topic:[test2]
```

5. Check whether the number of messages is the same before and after the migration

Taking `test3` for example, if you want to migrate 76,522 unconsumed messages from `test123-group`, and all these messages have actually been written to the `test3` topic of the target instance, data migration is considered successful.

Data Compression

Last updated：2024-01-09 14:49:40

Overview

Data compression can reduce network I/O transmission traffic and disk usage. This document describes the message formats supported for data compression and how to configure data compression based on your needs.

Message Format

Currently, CKafka supports two versions of message formats: v1 and v2 (imported in Kafka 0.11.0.0). CKafka is compatible with Kafka 0.9, 0.10, 1.1, 2.4, and 2.8.
Different configurations apply to different versions, which are described as below:
Message format conversion is mainly for compatibility with consumer programs on legacy versions. A CKafka cluster usually has message formats on multiple versions (v1 and v2).
The broker will decompress and recompress messages on a new version to convert them to the legacy format.
Message format conversion affects performance greatly as it requires extra compression and decompression operations. It also makes Ckafka's excellent zero-copy feature useless. Therefore, **you must use the same message format**.
Zero-copy: This feature can prevent costly data copy in kernel state when data is transferred in disks or over networks to implement fast data transfer.

Compression Algorithm Comparison

Snappy is the officially recommended compression algorithm. Its analysis process is as follows:
The performance of a compression algorithm is evaluated mainly based on two metrics: compression ratio and compression/decompression throughput.
Versions earlier than CKafka 2.1.0 support three compression algorithms: Gzip, Snappy, and LZ4.
The comparison of **performance metric** between the three algorithms in the actual use of CKafka is as shown below:
Compression ratio: LZ4 > Gzip > Snappy
Throughput: LZ4 > Snappy > Gzip
Comparison of **physical resource** usage is as shown below:
Bandwidth: As Snappy has the lowest compression ratio, its network bandwidth usage is the highest.

©2013-2025 Tencent Cloud International Pte. Ltd.

CPU: CPU usage is similar for each compression algorithm. Snappy uses more CPU resources during compression, while Gzip uses more CPU resources during decompression.
Therefore, the recommended order of the three compression algorithms under normal circumstances is LZ4 > Gzip > Snappy.
This recommended order has been well tested in most cases in the production environment. However, in extreme cases, LZ4 will increase the CPU load.
The analysis shows that LZ4 performs differently depending on the source data. Therefore, we recommend that you use the more stable Snappy compression algorithm if you are more concerned about the CPU usage.

Configuring Data Compression

A producer can use the following method to configure data compression:

```
Properties props = new Properties();
props.put("bootstrap.servers", "localhost:9092");
props.put("acks", "all");
props.put("key.serializer",
"org.apache.kafka.common.serialization.StringSerializer");
props.put("value.serializer",
"org.apache.kafka.common.serialization.StringSerializer");

// After the producer is started, all its produced message sets will be
compressed, which can greatly reduce the network transmission bandwidth and
disk usage of the Kafka broker.

// Note that different versions have different configurations. Currently,
versions 0.9 or earlier do not support compression. Versions 0.10 or later do
not support Gzip compression.

props.put("compression.type", " lz4 ");
Producer<String, String> producer = new KafkaProducer<>(props);

In most cases, after receiving a message from the producer, the broker will
retain it as-is without making any modifications.
```

Note

When data is sent to CKafka, `compression.codec` cannot be set.
Gzip compression is not supported by default. To use it, submit a ticket.
As Gzip compression causes high CPU consumption, if it is used, all messages will become `InValid` .

The program cannot run properly when the LZ4 compression is used. Possible causes include:

The message format is incorrect. The default message version of CKafka is v0.10.2. You need to use the message format v1.

The setting method for SDK varies by CKafka client. You can query the setting method in the open-source community (such as the description for C/C++ Client) to set the version of the message format.