Alibaba Cloud Message Queue for Apache Kafka

Console guide

Issue: 20200708

Legal disclaimer

Alibaba Cloud reminds you to carefully read and fully understand the terms and conditions of this legal disclaimer before you read or use this document. If you have read or used this document, it shall be deemed as your total acceptance of this legal disclaimer.

1. You shall download and obtain this document from the Alibaba Cloud website or other Alibaba Cloud-authorized channels, and use this document for your own legal business activities only. The content of this document is considered confidential information of Alibaba Cloud. You shall strictly abide by the confidentiality obligations. No part of this document shall be disclosed or provided to any third party for use without the prior written consent of Alibaba Cloud.
2. No part of this document shall be excerpted, translated, reproduced, transmitted, or disseminated by any organization, company, or individual in any form or by any means without the prior written consent of Alibaba Cloud.
3. The content of this document may be changed due to product version upgrades, adjustments, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and the updated versions of this document will be occasionally released through Alibaba Cloud-authorized channels. You shall pay attention to the version changes of this document as they occur and download and obtain the most up-to-date version of this document from Alibaba Cloud-authorized channels.
4. This document serves only as a reference guide for your use of Alibaba Cloud products and services. Alibaba Cloud provides the document in the context that Alibaba Cloud products and services are provided on an "as is", "with all faults" and "as available" basis. Alibaba Cloud makes every effort to provide relevant operational guidance based on existing technologies. However, Alibaba Cloud hereby makes a clear statement that it in no way guarantees the accuracy, integrity, applicability, and reliability of the content of this document, either explicitly or implicitly. Alibaba Cloud shall not bear any liability for any errors or financial losses incurred by any organizations, companies, or individuals arising from their download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, bear responsibility for any indirect, consequential, exemplary, incidental, special, or punitive damages, including lost profits arising from the use or trust in this document, even if Alibaba Cloud has been notified of the possibility of such a loss.

5. By law, all the contents in Alibaba Cloud documents, including but not limited to pictures, architecture design, page layout, and text description, are intellectual property of Alibaba Cloud and/or its affiliates. This intellectual property includes, but is not limited to, trademark rights, patent rights, copyrights, and trade secrets. No part of this document shall be used, modified, reproduced, publicly transmitted, changed, disseminated, distributed, or published without the prior written consent of Alibaba Cloud and/or its affiliates. The names owned by Alibaba Cloud shall not be used, published, or reproduced for marketing, advertising, promotion, or other purposes without the prior written consent of Alibaba Cloud. The names owned by Alibaba Cloud include, but are not limited to, "Alibaba Cloud", "Aliyun", "HiChina", and other brands of Alibaba Cloud and/or its affiliates, which appear separately or in combination, as well as the auxiliary signs and patterns of the preceding brands, or anything similar to the company names, trade names, trademarks, product or service names, domain names, patterns, logos, marks, signs, or special descriptions that third parties identify as Alibaba Cloud and/or its affiliates.
6. Please contact Alibaba Cloud directly if you discover any errors in this document.

Document conventions

Style	Description	Example
	A danger notice indicates a situation that will cause major system changes, faults, physical injuries, and other adverse results.	 **Danger:** Resetting will result in the loss of user configuration data.
	A warning notice indicates a situation that may cause major system changes, faults, physical injuries, and other adverse results.	 **Warning:** Restarting will cause business interruption. About 10 minutes are required to restart an instance.
	A caution notice indicates warning information, supplementary instructions, and other content that the user must understand.	 **Notice:** If the weight is set to 0, the server no longer receives new requests.
	A note indicates supplemental instructions, best practices, tips, and other content.	 **Note:** You can use Ctrl + A to select all files.
>	Closing angle brackets are used to indicate a multi-level menu cascade.	Click **Settings** > **Network** > **Set network type**.
Bold	Bold formatting is used for buttons, menus, page names, and other UI elements.	Click **OK**.
Courier font	Courier font is used for commands.	Run the `cd /d C:/window` command to enter the Windows system folder.
Italic	Italic formatting is used for parameters and variables.	`bae log list --instanceid` *Instance_ID*
[] or [a|b]	This format is used for an optional value, where only one item can be selected.	`ipconfig [-all|-t]`

Style	Description	Example
{} or {a|b}	This format is used for a required value, where only one item can be selected.	switch {active|stand}

Contents

1 Instances

1.1 View endpoints

To send and subscribe to Message Queue for Apache Kafka messages by using the SDK, you need to configure the endpoint according to the network type of the instance. You can view the endpoint of your instance in the Message Queue for Apache Kafka console.

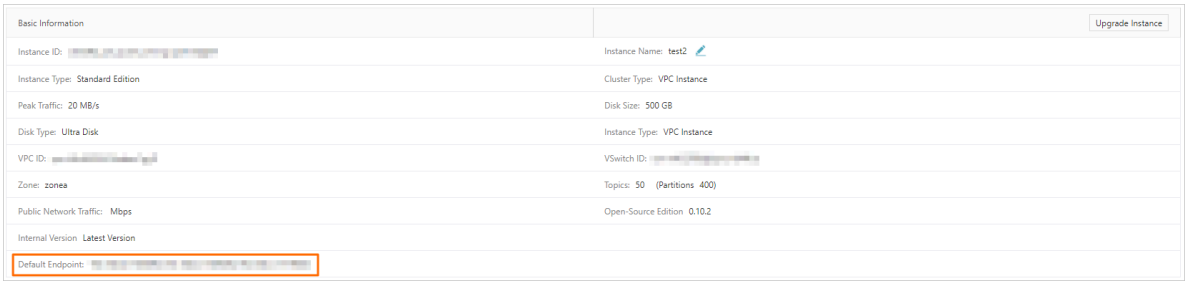
Background

Message Queue for Apache Kafka provides the following types of endpoints:

- Default Endpoint: is applicable to message sending and subscription in a Virtual Private Cloud (VPC) environment.

Procedure

1. Log on to the Message Queue for Apache Kafka console and select a region in the top navigation bar.
2. In the left-side navigation pane, click **Instance Details**.
3. On the **Instance Details** page, click the target instance.
4. Check the endpoint information in the **Basic Information** section.



Note:

- If the value of Instance Type is **VPC**, only **Default Endpoint** is displayed.

References

- #unique_5

1.2 Upgrade the instance configuration

If your Message Queue for Apache Kafka instance fails to meet your business requirements because the disk usage is continuously higher than 85% or the peak traffic continuously exceeds that you purchased, you can upgrade the instance configuration as needed.

Background

- Instance edition

  You can upgrade an instance from the Standard Edition to the Professional Edition.

- Peak traffic

  The peak traffic is restricted as follows during the upgrade:

  - If the purchased peak traffic is less than 120 MB/s, you can upgrade it to 120 MB/s at most.
  - If the purchased peak traffic is greater than 120 MB/s and no more than 300 MB/s, you can upgrade it to 300 MB/s at most.
  - If the purchased peak traffic is greater than 300 MB/s, you cannot upgrade it.

- Disk type

  The disk types are restricted as follows during the upgrade:

  - The disk type cannot be changed after the order is placed. Select the disk type with caution.
  - Ultra disks support peak traffic upgrade to 120 MB/s at most.
  - Solid State Drives (SSDs) support peak traffic upgrade to 300 MB/s at most.

Notes

Upgrading the instance specifications will cause instances in the cluster to restart one by one.

- If the client does not support the reconnection mechanism, the client may be unavailabl e after being disconnected.
- It will take about 30 minutes to upgrade the instance specifications. Services will not be interrupted but the messages may be out of order during the upgrade, that is, messages may be distributed to a different partition for consumption. Therefore, evaluate the impact on businesses before you proceed.

Prerequisites

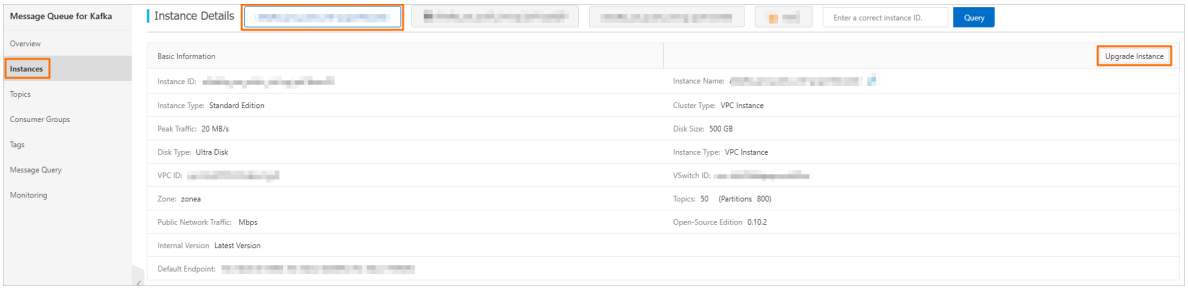
This topic describes how to upgrade a subscription Message Queue for Apache Kafka instance from the Standard Edition to the Professional Edition.

- The billing method is subscription.
- The instance is of the Standard Edition and it is in the **Running** state.

Procedure

To upgrade a subscription Message Queue for Apache Kafka instance from the Standard Edition to the Professional Edition, follow these steps:

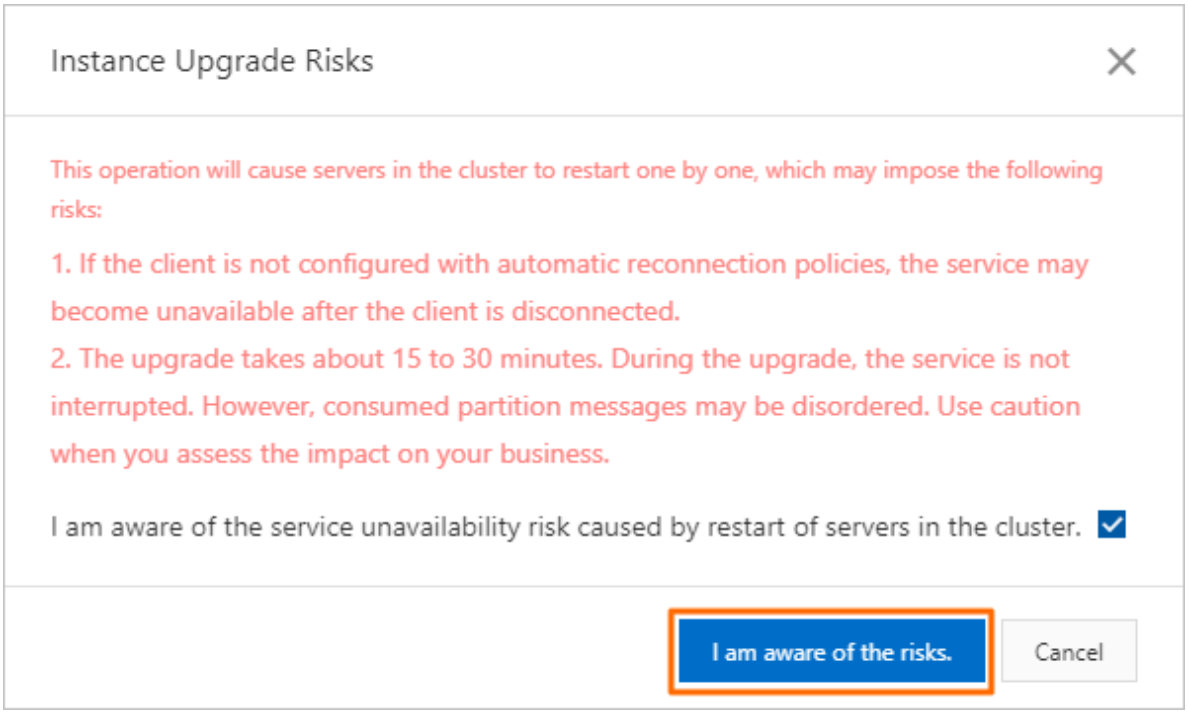
1. Log on to the Message Queue for Apache Kafka console and select a region in the top navigation bar.
2. In the left-side navigation pane, click **Instance Details**.
3. On the top of the **Instance Details** page, click the ID of the target instance. Click **Upgrade Instance** in the upper-right corner of the **Basic Information** section.



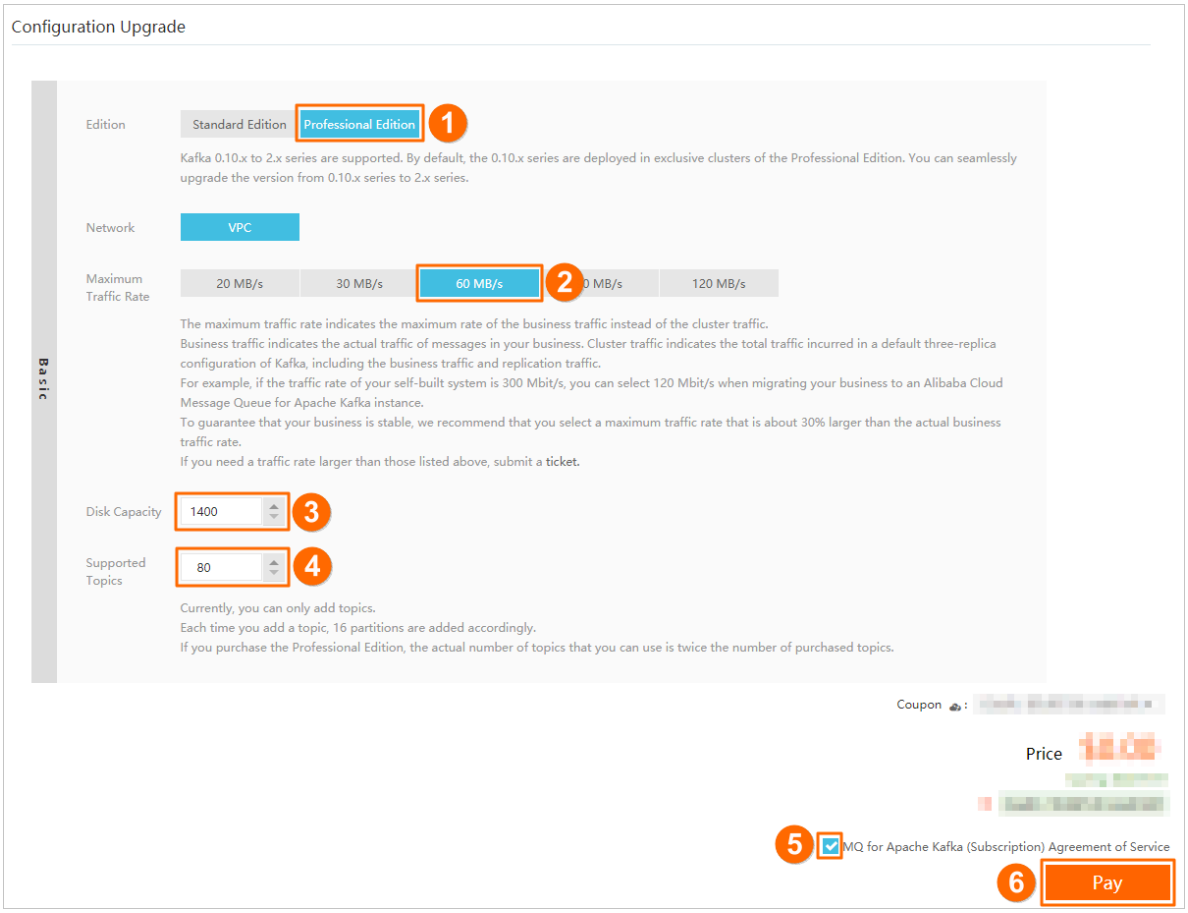
Note:

You can click the edit icon next to **Instance Name** and modify the instance as needed.

4. In the **Instance Upgrade Risks** dialog box, read the instance upgrade risks carefully. Confirm the risks, select **I am aware of the service unavailability risk caused by the restart of servers in the cluster.** and then click **I am aware of the risks.**



5. On the **Configuration Change** page, change configuraiton, select **Message Queue for Apache Kafka (Subscription) Terms of Service**, and then click **Pay**.



Notice:

- The recommended disk capacity is available for peak traffic. The disk capacity changes with the adjusted peak traffic.
- A longer time is consumed for a larger disk capacity span.
- After the specifications are upgraded, the effective time of the order is subject to that displayed on the upgrade order page.

Verification

Check the **Status** section on the **Instance Details** page.

- If the status is **Running**, the modification is successful.
- If the status is **Upgrading**, wait for a while.
- If the status is **Upgrade Failed**, submit a ticket.

What to do next

After the instance specifications are upgraded, you may need to modify the message configuration to adapt to the target instance edition. For more information, see Modify the message configuration.

References

- For more information about the disk usage or peak traffic, see Monitor resources and set alerts.
- For more information about billing methods and instance editions, see #unique_9.

1.3 Modify the message configuration

You can adjust the message retention period and the maximum message size based on the business requirements.

Background

Modifying the message configuration will cause instances in the cluster to restart one by one.

- If the client does not support the reconnection mechanism, the client may be unavailable after being disconnected.
- It will take 15 minutes to 30 minutes to modify the message configuration. Services will not be interrupted but the messages may be out of order during the upgrade, that is, messages may be distributed to a different partition for consumption. Therefore, evaluate the impact on businesses before you proceed.

Prerequisites

You have purchased a Message Queue for Apache Kafka instance, and the instance is in the Running state.

Procedure

1. Log on to the Message Queue for Apache Kafka console. In the top navigation bar, select a region where the instance is located.
2. In the left-side navigation pane, click **Instance Details**.
3. On the **Instance Details** page, click the target instance.
4. In the **Configurations** section, click **Configuration Change**.

5. In the **Configuration Change** dialog box, set parameters and click **Change**.

Parameter	Description
Message Retention Period	The maximum message retention period when the disk capacity is sufficient. • When the disk capacity is insufficient (that is, when the disk usage reaches 85%), the old messages are deleted in advance to ensure service availability. • The default value is 72 hours. The value ranges from 24 hours to 168 hours.
Maximum Message Length	The maximum size of a message that can be sent and received by Message Queue for Apache Kafka. • The upper limit for maximum message length is 10 MB, with no difference between the standard edition instance and the professional edition instance. • Before modifying the configuration, ensure that the target value matches the configuration on the producer and consumer.

6. In the **Alerts** dialog box, select **I am aware of the service unavailability risk caused by restart of servers in the cluster.** and then click **I am aware of the risks.**

Verification

Check the **Status** section on the **Instance Details** page.

- If the status is **Running**, the modification is successful.
- If the status is **Upgrading**, wait for a while.

1.4 Upgrade the instance version

This topic describes how to upgrade the Message Queue for Apache Kafka instance version so that you can use the related features.

Notes

- The Message Queue for Apache Kafka broker of version 0.10 may trigger bugs such as deadlocks and frequent rebalancing. We recommend that you upgrade the instance from version 0.10 to the stable version 0.10.2.

- If the **Open-Source Version** of the instance is 0.10 on the **Instance Details** page, and the upgrade button is available, you need to upgrade your instance to version 0.10.2.
- All newly purchased Message Queue for Apache Kafka instances are of version 0.10.2. The Message Queue for Apache Kafka team will gradually schedule a mandatory upgrade for instances of version 0.10. We recommend that you upgrade them beforehand.

Background

You can upgrade the open-source version or internal version of an instance to the required version in the Message Queue for Apache Kafka console. Upgrade of the two versions is compared as follows:

- Open-source version upgrade (major version upgrade)

  Upgrade the open-source version of the running Message Queue for Apache Kafka instance. For example, upgrade the open-source version of the instance from version 0.10.2 to version 2.2.0.

  Note:

  - The default deployment version of the Message Queue for Apache Kafka instance is version 0.10.x.
  - Message Queue for Apache Kafka Standard Edition does not support upgrade of the instance version from version 0.10.x to version 2.x. You need to upgrade the Standard Edition to the Professional Edition. For more information, see Upgrade the instance configuration.
  - Only the open-source versions 0.10.x and 2.x are supported for Message Queue for Apache Kafka instances.
  - Version 2.x is compatible with version 0.11.x and version 1.x.

- Internal version upgrade (minor version upgrade)

  Optimize the internal version of the running Message Queue for Apache Kafka instance. The open-source version of the instance does not change during the upgrade of the internal version. For example, the open-source version of an instance is version 0.10.2. After you upgrade the internal version of the instance, the open-source version of the instance is still version 0.10.2.

Upgrade the open-source version of an instance

Prerequisites

- You have purchased a Message Queue for Apache Kafka instance of the Professional Edition, and the instance is in the Running state.
- The open-source version of your Message Queue for Apache Kafka instance is version 0.10.x.

Procedure

1. Log on to the Message Queue for Apache Kafka console. In the top navigation bar, select the region where the instance is located.
2. In the left-side navigation pane, click **Instance Details**.
3. On the **Instance Details** page, click the ID of the target instance.
4. In the **Basic Information** section, when **Internal Version** is **Service Upgrade**, click **Service Upgrade** to upgrade the instance. When Internal Version is **Latest Version**, do not upgrade the instance.

   The instance version will be upgraded to the most suitable open-source version based on your specific instance conditions.
5. In the **Upgrade** dialog box, click the **Cross-Version Upgrade** tab.
   a. Enter your name in the **Name** field.
   b. Enter your emergency phone number in the **Emergency Phone Number** field.
   c. Select **Yes** for **Cross-Version Upgrade to 2.0**.
6. Click **Upgrade**.

Notice:

- If the client does not support the reconnection mechanism (enabled by default), the client may be unavailable after being disconnected. Ensure that the consumer supports the reconnection mechanism.
- It will take about 15 minutes for the upgrade. The service will not be interrupted during the upgrade and the business will not be affected.
- The message storage format of instances of version 2.x is different from that of the instances of version 0.10.x. Therefore, you cannot roll back to version 0.10.x after the upgrade. Proceed with caution.

- We recommend that you purchase a test instance for upgrade verification before you operate on the production instance.

Verification

The value of **Open-Source Edition** is **2.2.0** in the **Basic Information** section on the **Instance Details** page.

Upgrade the internal version of an instance

Prerequisites

- You have purchased a Message Queue for Apache Kafka instance, and the instance is in the Running state.
- The internal version of your Message Queue for Apache Kafka instance is not the latest version.

Procedure

1. Log on to the Message Queue for Apache Kafka console. In the top navigation bar, select the region where the instance is located.
2. In the left-side navigation pane, click **Instance Details**.
3. On the **Instance Details** page, click the ID of the target instance.
4. In the **Basic Information** section, when **Internal Version** is **Service Upgrade**, click **Service Upgrade** to upgrade the instance. When Internal Version is **Latest Version**, do not upgrade the instance.

   The instance version will be upgraded to the most suitable internal version based on your specific instance conditions.
5. In the **Upgrade** dialog box, set the following parameters so that we can contact you when an error occurs during the upgrade:
   - **Name**
   - **Emergency phone number**
6. Click **Upgrade**.

Notice:

- If the client does not support the reconnection mechanism (enabled by default), the client may be unavailable after being disconnected. Ensure that the consumer supports the reconnection mechanism.

- The upgrade will take about 15 minutes. The service will not be interrupted during the upgrade and the business will not be affected.

Verification

The value of **Internal Version** is **Latest Version** in the **Basic Information** section on the **Instance Details** page.

1.5 Configure the whitelist

You can modify the whitelist of IP addresses in the Virtual Private Cloud (VPC) port range to allow access from some IP addresses or IP ports to the Message Queue for Apache Kafka instance.

Prerequisites

You have purchased a Message Queue for Apache Kafka instance, and the instance is in the Running state.

Procedure

To add IP addresses or CIDR blocks to the whitelist, follow these steps:

1. Log on to the Message Queue for Apache Kafka console and select a region in the top navigation bar.
2. In the left-side navigation pane, click **Instance Details**.
3. On the **Instance Details** page, click the ID of the instance to be modified and click **Security Change** in the **Security Configuration** section.
4. In the **Security Change** dialog box, click **+ Add IP to Whitelist**, enter the required IP addresses or CIDR blocks, and click **Add**.

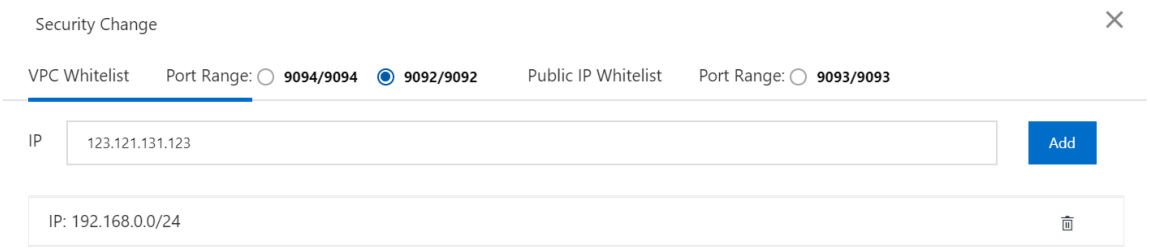
Notice:

- The whitelist can contain a maximum of 100 IP addresses or CIDR blocks.
- You can add IP addresses or CIDR blocks to the whitelist and use commas (,) to separate them. You can add a maximum of 10 IP addresses or CIDR blocks each time.
- You can delete or add only one IP address or CIDR block from or to the whitelist.

- You can delete the last IP address or CIDR block from the whitelist. Proceed with caution because the port range of the Message Queue for Apache Kafka cluster will be inaccessible after deletion.

The operations differ slightly for instances of different network types, with differences mainly in the port ranges.

- The VPC port range is 9092/9092. The default IP address and port number in the whitelist is 0.0.0.0/0, allowing access to the Message Queue for Apache Kafka instance through VPC networks.



- The VPC port range is 9094/9094. The default IP address and port number in the whitelist is 0.0.0.0/0, allowing access to the Message Queue for Apache Kafka instance through VPC networks.

Note:
The port range of 9094/9094 will only display after ACL is enabled. To enable ACL, see #unique_12/unique_12_Connect_42_section_39b_ody_elv.

5. (Optional) To delete the whitelist configuration, click the delete icon in the row where the IP address or CIDR block to be deleted is located in the **Security Change** dialog box.

1.6 Rebalance the topic traffic

When you upgrade the peak traffic of a Message Queue for Apache Kafka instance, the corresponding cluster may be scaled out. After the cluster is scaled out, you need to rebalance the topic traffic to distribute the traffic evenly across nodes in the scaled-out cluster. Otherwise, the original topic traffic is still distributed across the original nodes in

the cluster before the scale-out. The original topics are subject to the purchased peak traffic before the scale-out, and the new topics are not subject to this peak traffic.

Prerequisites

Your Message Queue for Apache Kafka instance is in the **Running (Pending Rebalancing)** state.

> **Note:**
> For more information about how to upgrade the instance peak traffic and when cluster scale-out is triggered, see Upgrade the instance configuration.

Notes

When your Message Queue for Apache Kafka instance is in the **Running (Pending Rebalancing)** state, you can receive and send messages by using this instance but cannot create resources such as topics and consumer groups in this instance.

Traffic rebalancing methods

The following table lists the principles, impacts, scenarios, and durations of the traffic rebalancing methods supported by Message Queue for Apache Kafka.

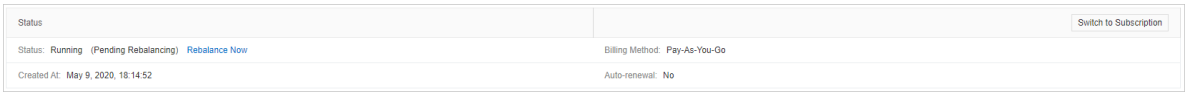
Traffic rebalancing method	Principle	Impact	Scenario	Duration
Add Partitions to All Topics	Add partitions to the new nodes after the cluster scale-out for all topics on the original nodes.	• New messages in partitions are out of order. • The number of partitions changes. **Note:** If your client cannot automatically detect new partitions in scenarios such as stream computing or delivery of messages to and consumption of messages from specified partitions, you may need to restart the client or modify the client code.	• The partition order is not required. • The partition to which messages are sent is not specified. • The consumption mode is Subscribe.	Seconds.

Traffic rebalancing method	Principle	Impact	Scenario	Duration
Migrate Partitions of All Topics (recommended) **Note:** Currently, only the latest version allows you to migrate partitions. You need to submit a ticket and ask Message Queue for Apache Kafka Customer Services to upgrade your broker to the latest version.	• Local storage: The kafka-reassign-partitions tool is used to migrate topic data in partitions. • Cloud storage: The mapping is modified and the topic data in partitions is not migrated.	• Local storage: Temporary internal traffic is generated. **Notice:** If you want to migrate a great amount of data from local storage, large temporary internal traffic is generated, which may last for several hours or longer. Therefore, evaluate the impact before you proceed. We recommend that you migrate the data during off-peak hours of service traffic. • Cloud storage: No temporary internal traffic is generated.	All cluster scale-out scenarios are supported.	• Local storage: minutes or hours. **Notice:** This duration depends on the amount of the data you want to migrate from local storage. If the data volume is large, the migration may take several hours or longer. • Cloud storage: seconds. **Note:** It takes about 30 seconds to migrate a topic.

Traffic rebalancing method	Principle	Impact	Scenario	Duration
Do Not Rebalance (not recommended)	You do not need to perform any operations. The original topics are still distributed on the nodes of the cluster before the scale-out, and the new topics are evenly distributed across all cluster nodes after the scale-out.	• The original topics are subject to the peak traffic before the scale-out. • If the original topic traffic is large, the traffic between cluster nodes may be unbalanced.	• The original topic traffic is very small, and the original topic traffic is not greatly improved after the cluster is scaled out. • New topics are created after the cluster is scaled out. Most of the traffic is directed to the new topics.	Takes effect immediately.

Procedure

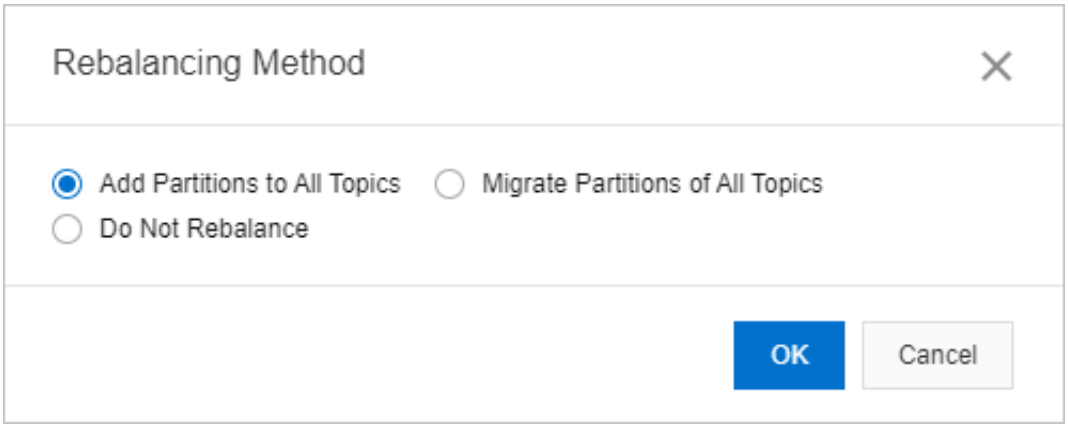
1. Log on to the Message Queue for Apache Kafka console.
2. In the top navigation bar, select the region of the instance for which you want to rebalance the topic traffic.
3. In the left-side navigation pane, click **Instances**.
4. On the **Instance Details** page, select the instance for which you want to rebalance the topic traffic. In the **Status** section, click **Redirect Now**.



5. In the **Rebalancing Method** dialog box, select a rebalancing method. Supported methods are as follows:

   - Add Partitions to All Topics

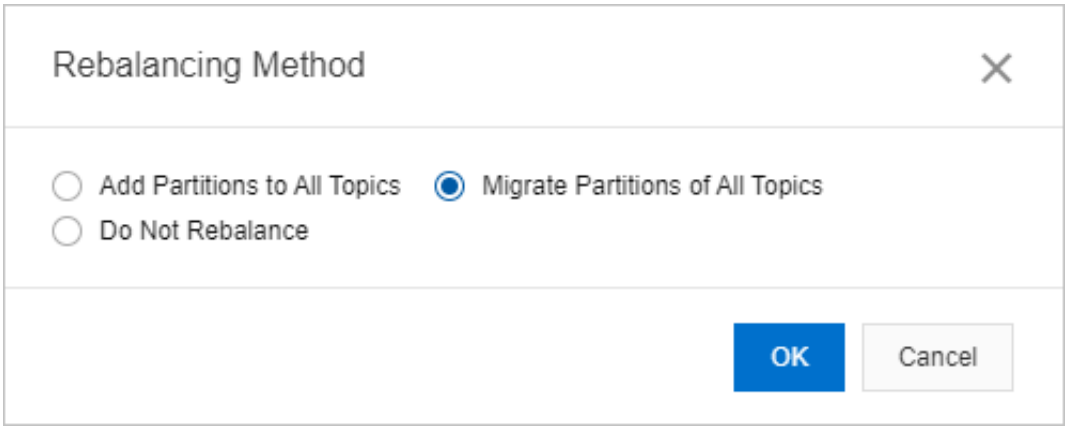
     Select **Add Partitions to All Topics** and then click **OK**.

   - Migrate Partitions of All Topics

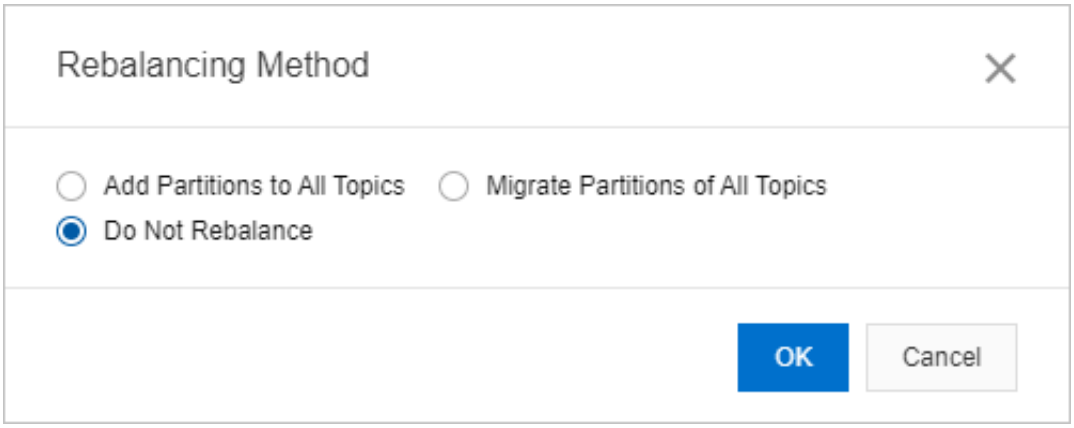
     a. Submit a ticket and ask Message Queue for Apache Kafka Customer Services to upgrade your broker to the latest version.

     b. Select **Migrate Partitions of All Topics** and then click **OK**.

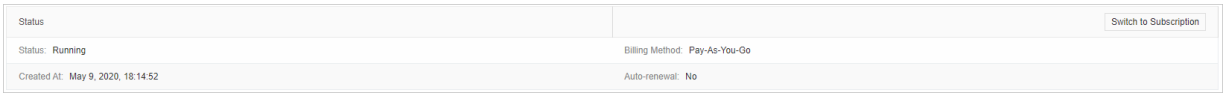
   - Do Not Rebalance

     Select **Do Not Rebalance** and then click **OK**.



Result

After the topic traffic is rebalanced, the instance status is **Running**.



1.7 View the execution records of tasks

After restart tasks, such as configuration change, version upgrade, ACL enabling and instance upgrade, are initiated for a Message Queue for Apache Kafka instance, you can view the execution records of these restart tasks in the Message Queue for Apache Kafka console to obtain the information such as task type, start time, end time and status.

Prerequisites

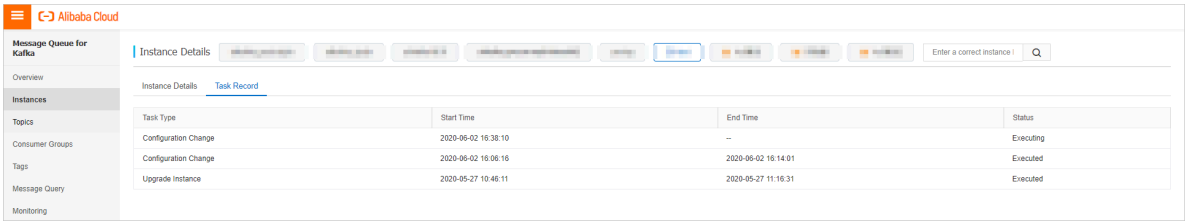
You have created and deployed a Message Queue for Apache Kafkainstance, and the instance is in running status. For more information, see #unique_15.

Procedure

1. Log on to the Message Queue for Apache Kafkaconsole.
2. Select a region in the top navigation bar.
3. Click **Instances** in the left-side navigation pane.

4. On the **Instances** page, select an instance and click **Task Records**.

   The execution records of restart tasks are displayed on the **Task Records** tab.



Parameter	Description	Example
Task Type	The type of the executed task. Valid values: • Configuration Change: You can modify the message configurations of the instance, including the message retention period and the maximum message size. For more information, see Modify the message configuration. • Upgrade Instance: You can upgrade the configurations of the instance, including the specification type, instance type, peak traffic, disk capacity, and topic specification. For more information, see Upgrade the instance configuration. • Service Upgrade: You can upgrade the major and minor versions of the instance. For more information, see Upgrade the instance version. • Enable ACL: The ACL feature is provided by Alibaba Cloud Message Queue for Apache Kafka to manage simple authentication and security layer (SASL) users and resource access permissions. For more information, see #unique_12.	Instance upgrade

Parameter	Description	Example
Started At	The time when the task starts to be executed.	2020-05-27 10:46:11
Ended At	The time when the task execution ends.	2020-05-27 11:16:31
Status	The current status of the task. Valid values: • Not Executed • Task Running • Executed • Canceled	Executed

1.8 Instances

1.8.1 Which open-source Apache Kafka versions are supported by Message Queue for Apache Kafka instances?

This topic describes the open-source Apache Kafka versions supported byMessage Queue for Apache Kafka instances.

- Message Queue for Apache Kafka Standard Edition: Only the open-source Apache Kafka version 0.10.x is supported and deployed by default. To use later versions, upgrade your instance to the Professional Edition first. For more information, see Upgrade the instance configuration and Upgrade the instance version.
- Message Queue for Apache Kafka Professional Edition: Open-source Apache Kafka versions 0.10.x to 2.x are supported. By default, version 0.10.x are deployed. To upgrade your Apache Kafka version, see Upgrade the instance version.

Note:

- The version 2.x is compatible with versions 0.10.x and 0.9.0.
- The version 0.10.x is compatible with the version 0.9.0.

1.8.2 How can I select the instance edition?

Message Queue for Apache Kafka provides the Standard Edition and Professional Edition. For more information, see #unique_9/unique_9_Connect_42_section_7cb_lev_cdq. You can select the instance edition based on the migration status.

- Standard Edition
  - Peak traffic = Total traffic in the cluster/3 (For optimization)
  - Disk size = Average traffic × Storage duration × 3 (replicas)
  - Number of topics: depends on the actual business demand.

    Note:
    We recommend that you optimize topics to reduce costs when migrating your data to the cloud.
- Professional Edition
  - Peak traffic = Total traffic in the cluster/3 (For optimization)
  - Disk size = Average traffic × Storage duration × n (replicas)

    Note:
    When you create a topic, n is 1 for cloud storage and 3 for local storage. For more information about the comparison between cloud storage and local storage, see #unique_19.
  - Number of topics: depends on the actual business demand.

    Note:
    We recommend that you optimize topics to reduce costs when migrating your data to the cloud.

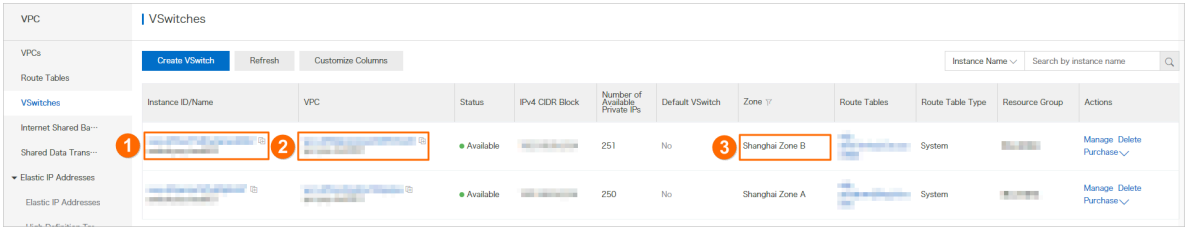
1.8.3 How can I obtain the VPC information required for deploying an instance?

To obtain the Virtual Private Cloud (VPC) information, follow these steps:

1. Log on to the VPC console.
2. In the left-side navigation pane, click **VSwitches**.

3. On the **VSwitches** page, obtain the following information:

   - VSwitch ID
   - VPC ID
   - Zone



Note:

In the Message Queue for Apache Kafka console, select the zone (A to G) displayed on this page. For example, if a VSwitch of the VPC is in **Zone B**, select **Zone B** in the Message Queue for Apache Kafka console.

1.8.4 What is impact of upgrading brokers?

Upgrading brokers has the following impact:

- During the upgrade process, all brokers in the Message Queue for Apache Kafka cluster are restarted sequentially. The service is not interrupted when the brokers are restarted. However, the messages consumed within 5 minutes after each broker is restarted may be out of order in the specific partition. In particular, ordered messages will not be out of order, but will be temporarily unavailable.
- Existing client connections may be interrupted in the restart process. Your clients must be able to automatically reconnect to other brokers that automatically take over the service.
- During the upgrade and restart of the brokers, the volumes of messages processed by each partition are also uneven. You need to evaluate the impact of the upgrade on your business.

It takes about 5 to 15 minutes to upgrade all the brokers. If you have multiple instances, you can upgrade a test cluster first, and upgrade the production cluster after the test cluster is upgraded.

1.8.5 Why am I unable to change the region or network type of the instance?

After an instance is purchased and deployed, its region and network properties are closely integrated with its physical resources and therefore cannot be changed. To change the region or network properties of an instance, release the instance and purchase a new instance.

1.8.6 When are old messages deleted in Message Queue for Apache Kafka?

- When disk usage is less than 85%, expired messages are deleted at 04:00 every day.
- When disk usage reaches 85%, expired messages are deleted immediately.
- When disk usage reaches 90%, old messages (expired or not) are deleted according to time.

Message Queue for Apache Kafka dynamically controls disk usage to prevent instance downtime due to insufficient disk space, which may affect service availability. We recommend that you keep disk usage at no more than 70% to ensure business health so that messages can be traced back. To resize disks, see Upgrade the instance configuration.

1.8.7 How can I quickly test whether the Message Queue for Apache Kafka broker is normal?

After creating and deploying a Message Queue for Apache Kafka instance, you can directly send messages through the Message Queue for Apache Kafka console to quickly test whether the broker is normal.

Prerequisites

You have created and deployed a Message Queue for Apache Kafkainstance, and the instance is in the **Running** state.

Procedure

Test the Message Queue for Apache Kafka broker as follows:

1. Create a topic
2. Send a message
3. View the partition status
4. Query a message by offset

You can repeat Steps 2 through 4 multiple times. If all repeated steps are successful, the broker works properly.

> **Note:**
> If the broker works properly but messages cannot be sent, we recommend that you check the caller, such as the native client or an ecosystem component.

Create a topic

Create a topic for receiving messages.

1. Log on to the Message Queue for Apache Kafka console.
2. In the left-side navigation pane, click **Topics**.
3. On the top of the **Topics** page, click the target instance, and then click **Create Topic**.
4. In the **Create Topic**dialog box, set topic properties and then click **Create**.

   The fields are described as follows:

   - **Topic**: the name of a topic, for example, demo.
   - **Tags**: the tag of the topic, for example, demo.
   - **Instance**: the ID of the instance, for example, alikafka_pre-cn-***.
   - **Description**: the description about the topic, for example, demo.
   - **Partitions**: the number of partitions for the topic, for example, 12.

Send a message

Send a message to the specified partition of the created topic.

1. On the **Topics** page, find the created topic. In the **Actions** column, click **Send Message**.

2. In the **Send Message**dialog box, set a partition and message properties, and then click **Send**.

   The fields are described as follows:

   - **Partition**: the ID of a partition, for example, 0.
   - **Message Key**: the key of the message to be sent, for example, demo.
   - **Message Value**: the value of the message, for example, {"key": "test"}.

View the partition status

After sending a message to the specified partition, you can view the partition status.

1. On the **Topics** page, find the topic to which the message is sent. In the **Actions** column, click **Partition Status**.

2. In the **Partition Status** dialog box, click **Refresh**.

Query a message by offset

Query a message based on the partition ID and offset.

1. In the left-side navigation pane, click **Message Query**.
2. On the **Message Query** page, click the target instance and click the **Query by Offset** tab.
   a) In the **Enter a topic.** field, enter a topic.
   b) From the **Select a partition.** drop-down list, select the ID of the partition to which the message is sent.
   c) In the **Enter an offset.** field, enter an offset.
   d) Click **Search**.

1.8.8 Are delayed messages supported?

Like Apache Kafka, Message Queue for Apache Kafka does not support delayed messages.

1.8.9 Can I access a Message Queue for Apache Kafka instance from the Internet?

The commercial edition of Message Queue for Apache Kafka service only supports access from Virtual Private Cloud (VPC).

> **Notice:**
> You can select the network type only when purchasing the instance and cannot change it after purchase.

1.8.10 Can we connect Message Queue for Apache Kafka instances in two different VPCs?

Yes. You can use Cloud Enterprise Network (CEN) or Virtual Private Network (VPN) Gateway to connect Message Queue for Apache Kafka instances in two different VPCs.

CEN

CEN allows you to establish private channels between VPCs. With automatic route distribution and learning, CEN can speed up network convergence, improve quality and security in cross-network communication, and implement interconnection among network-wide resources. For more information, see What is Cloud Enterprise Network?

You can use a CEN instance to connect two VPCs under the same account or different accounts. The following table describes the scenarios.

Scenarios	Configuration methods
Same-account VPC-to-VPC connection	Connect two VPCs in the same region under the same account
	Connect two VPCs in different regions under the same account
Cross-account VPC-to-VPC connection	Connect two VPCs in the same region under different accounts
	Connect two VPCs in different regions under different accounts

CEN has the following benefits:

- Worldwide connection

  CEN is an enterprise-class network that can interconnect Alibaba Cloud network resources around the world. CEN can also interconnect local data centers that are

already connected to the Alibaba Cloud network. CEN validates the IP address ranges of the connected networks and ensures that the IP address ranges are not in conflict. In addition, CEN automatically forwards and learns multi-node routes through controllers to rapidly converge global routes.

- Low latency and high speed

  CEN provides low-latency and high-speed network transmission. The maximum access rate between local networks can reach the port forwarding rate of the gateway device. In global network communication, the latency of CEN is much shorter than that of a public network.

- Nearest access and shortest path

  CEN deploys multiple access points and forwarding points in more than 60 regions around the world to support nearest access to Alibaba Cloud, enabling traffic to travel through a responsive and latency-free network.

- Link redundancy and disaster recovery

  CEN features high availability and network redundancy by providing at least four redundant links between any two access points. If a link fails, CEN ensures your services to operate normally without network jitter or interruption.

- Systematic management

  CEN has systematic network monitoring capabilities that automatically detect route conflicts caused by system changes to ensure the network stability.

VPN Gateway

VPN Gateway is an Internet-based networking service that supports route-based IPsec-VPN connections. You can use IPsec-VPN connections to connect VPCs securely and reliably. For more information, see #unique_33.

VPN Gateway offers the following benefits:

- High security

  You can use the IKE and IPsec protocols to encrypt data to ensure secure and reliable data transmission.

- High availability

  With a the hot-standby architecture, VPN Gateway supports failover within seconds to ensure session persistence and zero service downtime.

- Low cost

  The encrypted Internet-based channels established by VPN Gateway are more cost-effective than leased lines.

- Easy configuration

  VPN Gateway is a ready-to-use service that requires no additional configuration.

1.8.11 Do I have to select the 172.16.0.0/12 CIDR block for a classic network?

No. However, the configuration is the simplest when the 172.16.0.0/12 CIDR block is used. For more information about other CIDR blocks, see #unique_35.

1.8.12 Is data compression supported?

Yes. The Message Queue for Apache Kafka broker can send and receive compressed data.

To use this feature, you need to set compression-related parameters on a Message Queue for Apache Kafka client. When you set compression-related parameters on a Message Queue for Apache Kafka client, note the following points:

- Compression format: Formats such as Snappy, LZ4, and GZIP are supported. The GZIP format consumes a large quantity of CPU resources. Therefore, we recommend that you use Snappy or LZ4.
- Scenarios: Generally, CPU resources are more expensive than traffic and storage resources. Therefore, we recommend that you use compression only in scenarios that require a high compression ratio, such as logs.
- CPU consumption: Compression occupies extra CPU resources, more than 20% on average. You can test the extra CPU consumption based on the actual scenario.

2 Topics

2.1 View partition status

To view the total number of messages on the Message Queue for Apache Kafka broker or the consumption progress of each partition, you can query the partition status.

Prerequisites

You have created a topic. For more information, see #unique_39/unique_39_Connect_42_section_zm0_ysj_343.

Procedure

1. Log on to the Message Queue for Apache Kafka console and select a region in the top navigation bar.
2. In the left-side navigation pane, click **Topics**.

3. On the **Topics** page, select the target instance, find the target topic, and then click **Partition Status** in the **Actions** column.

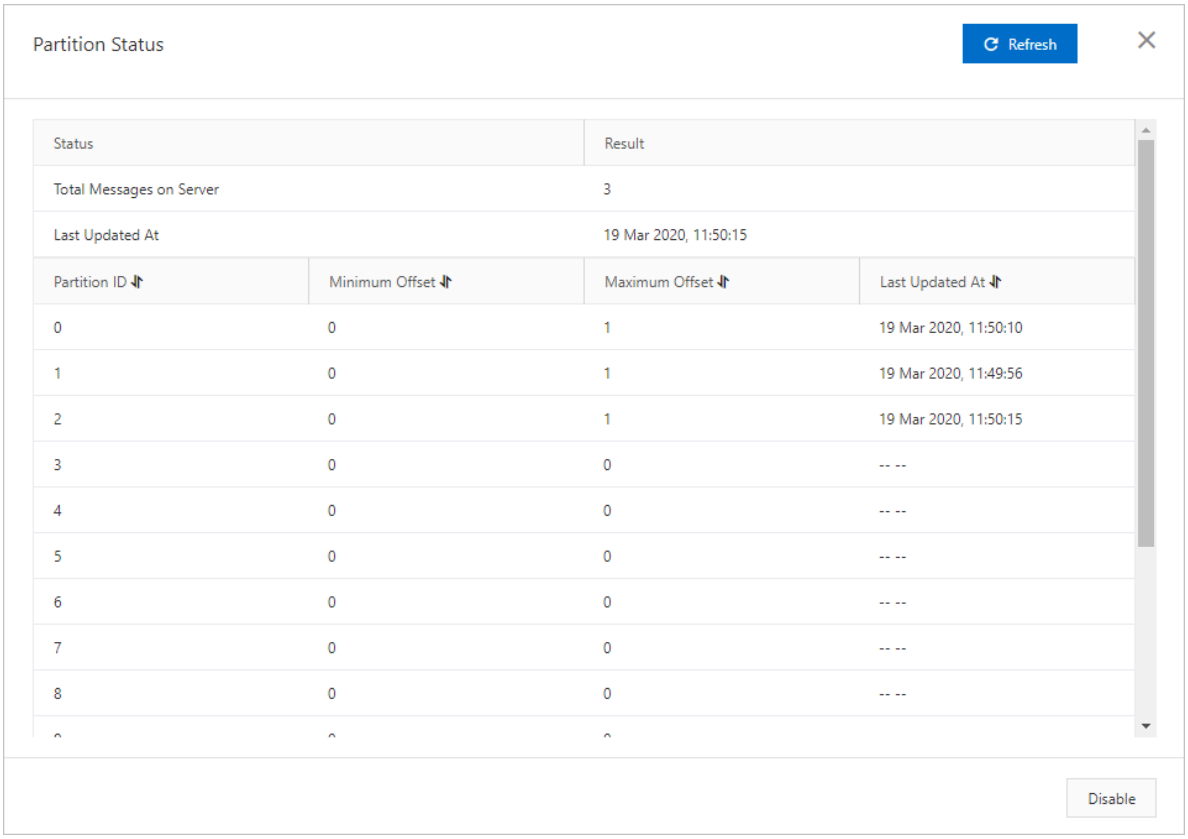


Table 2-1: Partition status information

Parameter	Description
Total Messages on Server	The total number of messages in all partitions.
Last Updated At	The time when the last message in all partitions is saved.
Partition ID	The ID of the partition corresponding to the topic.
Minimum Offset	The minimum message consumer offset of the topic in the current partition.
Maximum Offset	The maximum message consumer offset of the topic in the current partition.
Last Updated At	The time when the last message in the partition is saved.

2.2 Topics

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

In Message Queue for Apache Kafka, messages are stored and scheduled by partition. If messages are stored in too many partitions, the cluster performance and stability are greatly reduced.

2.2.2 Why am I unable to reduce the number of partitions for a topic?

This is caused by design constraints of Message Queue for Apache Kafka.

2.2.3 Why is automatic creation of topics not supported?

Automatic creation of topics facilitates usage but increases the O&M difficulty and easily causes system instability. In Message Queue for Apache Kafka, topics must be authenticated. Therefore, Message Queue for Apache Kafka does not automatically create topics. However, you can create topics and consumer groups by using the console, API operations, or automated orchestration tools.

- In the console: Create a topic
- By using API operations: #unique_44
- By using Terraform: alicloud_alikafka_topic

2.2.4 What is the relationship between the topic count and partition count?

In addition to the default number of partitions, 16 partitions are added for each additional topic. For example, if you purchased an instance with 50 topics, 20 MB/s peak traffic, and 400 partitions by default, after you add 10 topics, the number of partitions of this instance is increased by 160, and the total number of partitions becomes 560.

2.2.5 What are the types of topic storage?

Currently, Message Queue for Apache Kafka instances of the Professional Edition support two types of topic storage engines: cloud storage and local storage. For more information, see #unique_19.

2.2.6 Can I get one topic free for each purchased topic for instances of the Professional Edition?

The number of topics available for an instance of the Professional Edition is twice the number of topics you purchased. For example, if you purchase an instance with 50 topics, the actual number of topics you can create in the instance is 100.

2.2.7 Why does "Last Updated At" display 1970 for the partitions of a topic?

If no messages are sent to the partitions of a topic, "Last Updated At" for the partitions displays 1970.

2.2.8 Why am I unable to reduce partitions after deleting a topic?

Symptoms

After you delete a topic with XX partitions and create this topic again with the partition count set to a value less than XX, the system prompts that **the topic is created but the partition count cannot be less than the previously configured number, so the partition count is reset to XX.**

Causes

In earlier versions, after a topic is deleted, its routing information is not completely cleared . As a result, you cannot create a topic with fewer partitions than those of the deleted topic . After you upgrade your instance to the new version, the routing information about any topic you deleted in the original version is still retained. To completely clear the routing information, you need to create the previously deleted topic and then delete it. Then, you can create a topic with any partition count as needed.

Solutions

1. Upgrade the instance version to the latest version.

   In the Message Queue for Apache Kafka console, go to the **Instance Details** page. In the **Basic Information** section, view the current **Internal Version**.

   - If **Latest Version** is displayed, you do not need to upgrade the version.
   - If **Service Upgrade** is displayed, click **Service Upgrade** to upgrade the version.

2. Create -> Delete -> Create again

   Go to the **Topics** page, create the previously deleted topic, delete it, and then create a topic and set the partition count.

2.2.9 Can the average number of partitions for each topic be reduced?

Warning:

Reducing the average number of partitions for each topic may reduce the concurrency, affect the topic throughput, and cause message congestion.

If you want to reduce the average number of partitions for each topic, submit a ticket.

2.2.10 Is data compression supported?

The latest version of Message Queue for Apache Kafka supports data compression. For more information about how to upgrade your version to the latest version, see Upgrade the instance version.

3 Consumer groups

3.1 Reset consumer offsets

You can reset a consumer offset to change the current consumption position of a consumer. You can reset the consumer offset to skip the accumulated or undesired messages instead of consuming them, or to consume messages after a certain time point regardless of whether the messages before this time point have been consumed.

Prerequisites

You have stopped all consumers. Message Queue for Apache Kafka does not support resetting offsets of online consumers.

> **Notice:**
> After the consumer is stopped, the broker waits for a period of time, which is specified in ConsumerConfig.SESSION_TIMEOUT_MS_CONFIG (10,000 ms by default), and then determines that the consumer has been stopped.

Context

Message Queue for Apache Kafka supports the following modes of consumer offset resetting:

- Clear messages: If the subscriber does not want to consume accumulated messages on the broker anymore, the subscriber can clear messages and specify the consumption offset to the latest position.

  > **Notice:**
  > Accumulated messages are not deleted. Only the consumer offset is changed.

- Start consumption at the specified time point: You can reset the offset of a consumer group to a time point "t" in the past or future, that is, a time point when a message is stored. After the reset, the consumer group subscribes to messages stored after "t".

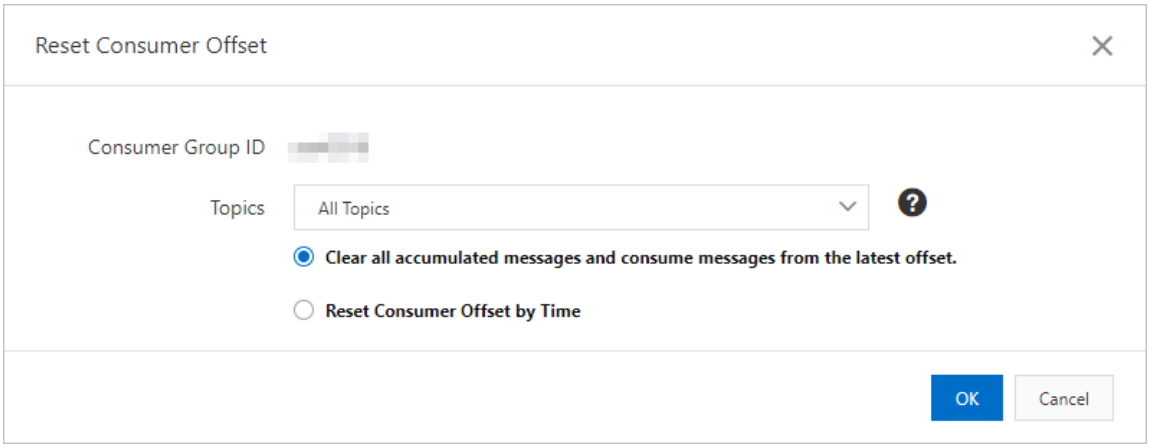
Procedure

1. Log on to the Message Queue for Apache Kafka console.
2. In the top navigation bar, select the region where the target instance is located.
3. In the left-side navigation pane, click **Consumer Groups**.

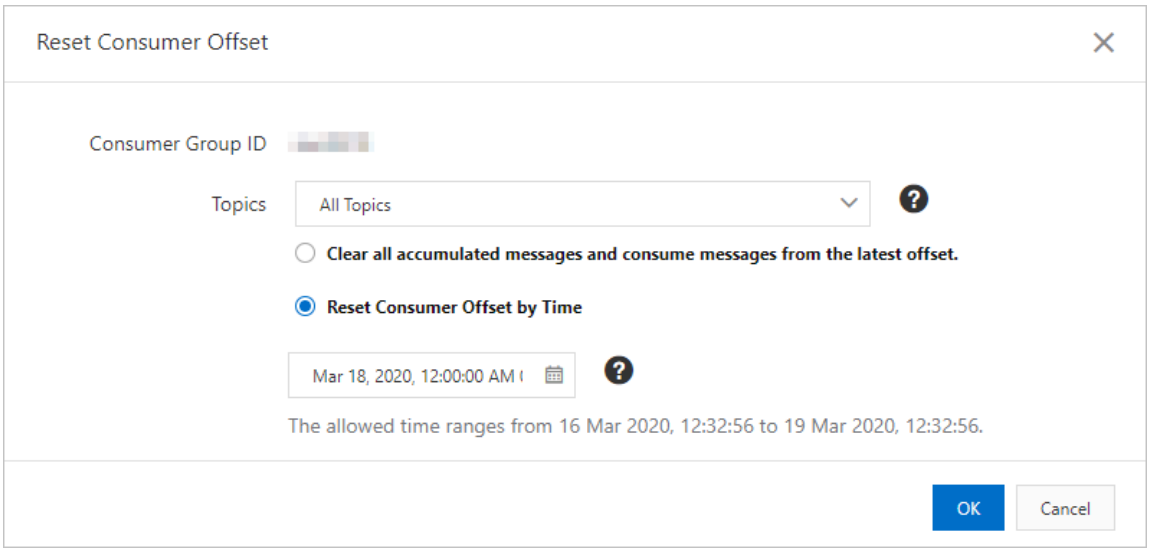
4. On the top of the **Consumer Groups** page, click the target instance, find the target consumer group, and then click **Reset Consumer Offset** in the **Actions**column.
5. In the **Reset Consumer Offset** dialog box, select the topic from the **Topics** drop-down list, select a resetting policy, and then click **OK**.

   Message Queue for Apache Kafka supports the following resetting policies:

   - **Clear all accumulated messages and consume messages from the latest offset.**: corresponds to the message clearing function described at the beginning of this topic.



   - **Reset Consumer Offset by Time**: corresponds to the function of starting consumption at the specified time point described at the beginning of this topic. If you select this policy, you need to specify the time point.



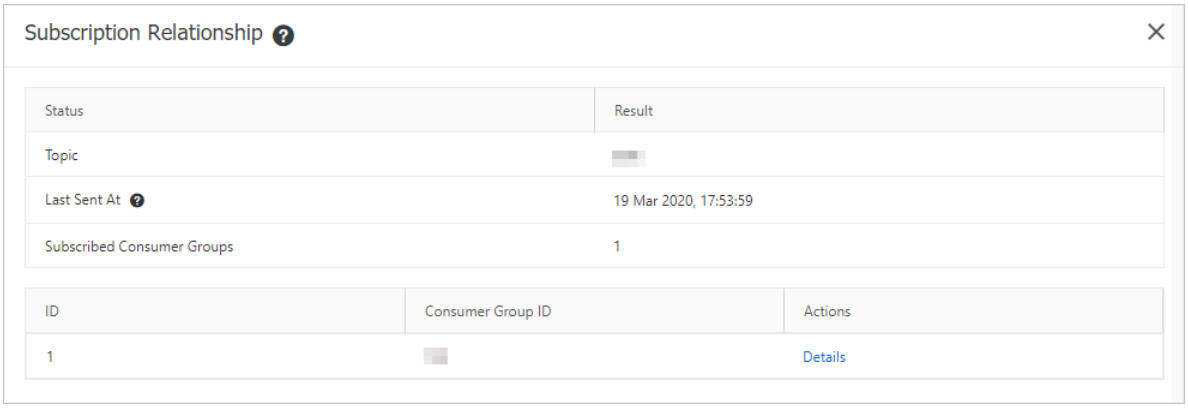
3.2 View the consumption status

When message accumulation or skewing occurs, you can view the subscription relationship between the consumer group and the topic and determine the consumption progress of each partition.

View consumer groups that subscribe to a topic

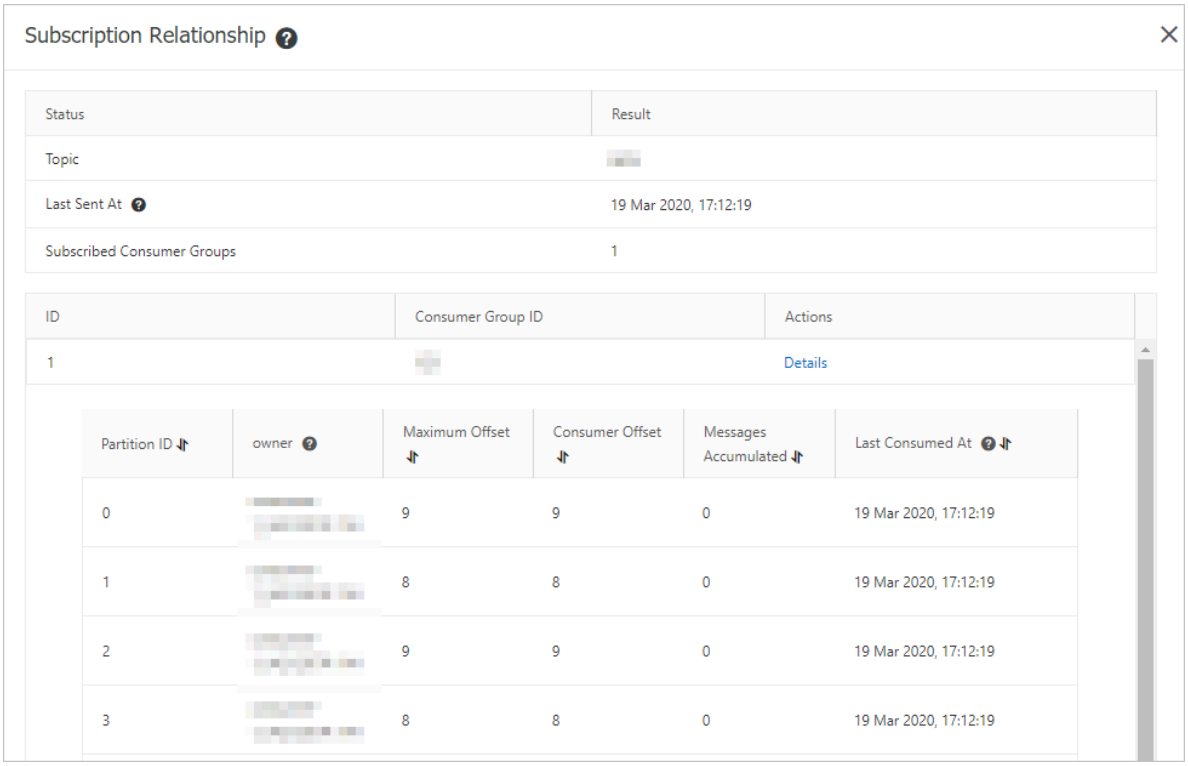
1. Log on to the Message Queue for Apache Kafka console.
2. In the top navigation bar, select a region.
3. In the left-side navigation pane, click **Topics**.
4. On the **Topics** page, select the target instance, find the target topic, and in the **Actions** column, choose **More** > **Subscription Relationship**.

   In the **Subscription Relationship** dialog box, all the consumer groups that subscribe to the topic are displayed.



5. In the **Consumer Group ID** column, find the target Consumer Group, and in the **Actions** column, click **Details**.

   The message consumption details in each partition of this topic are displayed.



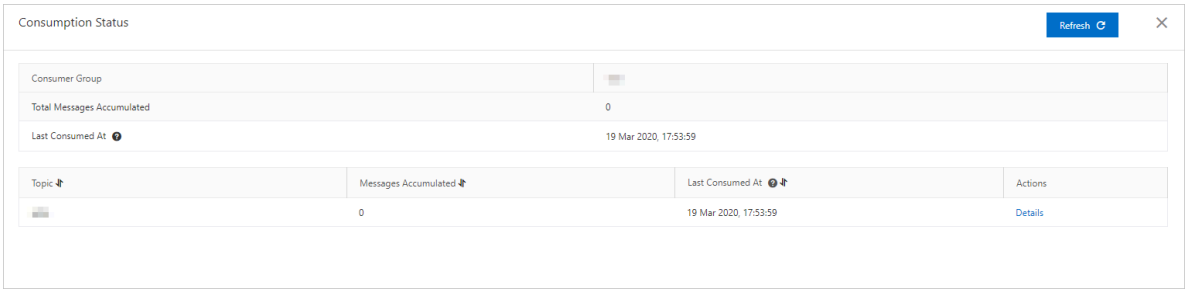
Parameter	Description
Partition ID	The ID of the partition corresponding to the topic.
owner	The ID and IP address of the online consumer that has subscribed to a specified topic are displayed in real time. **Note:** • The value format is <Client ID>_/<IP>. • You cannot view the owner information of offline consumers.
Maximum Offset	The maximum message consumer offset of the topic in the current partition.
Consumer Offset	The message consumer offset of the topic in the current partition.

Parameter	Description
Messages Accumulated	The total number of accumulated messages in the current partition, that is, the maximum offset minus the consumer offset. **Notice:** Messages Accumulated is a key metric. If a large number of messages are accumulated, the consumer may be blocked or the consumption speed cannot keep up with the production speed. In this case, you need to analyze the running status of the consumer and improve the consumption speed. For more information, see #unique_55.
Last Consumed At	The time when the last message consumed by the consumer group was sent to the broker for storage.

View topics to which a consumer group subscribes

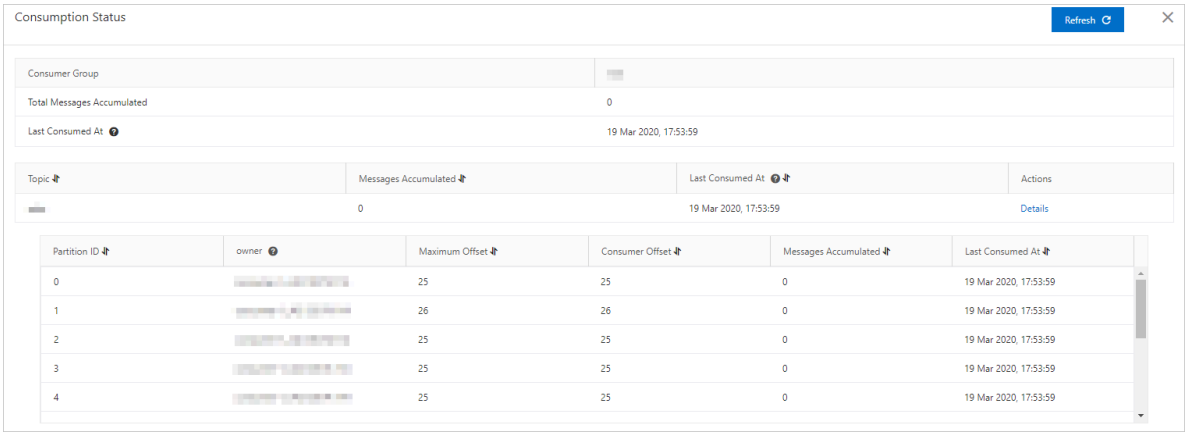
1. Log on to the Message Queue for Apache Kafkaconsole.
2. In the top navigation bar, select a region.
3. In the left-side navigation pane, click **Consumer Groups**.
4. On the **Consumer Groups** page, click the target instance, find the target consumer group, and then click **Consumption Status** in the **Actions** column.

   In the **Consumption Status** dialog box, all the topics to which this consumer group subscribe and its **Total Messages Accumulated** and **Messages Accumulated** are displayed.



5. In the **Topic** column, find the target Consumer Group, and in the **Actions** column, click **Details**.

   The message consumption details in each partition of this topic are displayed.



Parameter	Description
Partition ID	The ID of the partition corresponding to the topic.
owner	The ID and IP address of the online consumer that has subscribed to a specified topic are displayed in real time. **Note:** • The value format is <Client ID>_/<IP>. • You cannot view the owner information of offline consumers.
Maximum Offset	The maximum message consumer offset of the topic in the current partition.
Consumer Offset	The message consumer offset of the topic in the current partition.

Parameter	Description
Messages Accumulated	The total number of accumulated messages in the current partition, that is, the maximum offset minus the consumer offset. **Notice:** Messages Accumulated is a key metric. If a large number of messages are accumulated, the consumer may be blocked or the consumption speed cannot keep up with the production speed. In this case, you need to analyze the running status of the consumer and improve the consumption speed. For more information, see #unique_55.
Last Consumed At	The time when the last message consumed by the consumer group was sent to the broker for storage.

3.3 Consumer groups

3.3.1 Why is automatic creation of consumer groups not supported?

Automatic creation of consumer groups facilitates usage but increases the O&M difficulty and easily causes system instability. In Message Queue for Apache Kafka, consumer groups must be authenticated. Therefore, Message Queue for Apache Kafka does not automatically create consumer groups. However, you can create consumer groups by using the console, API operations, or automated orchestration tools.

- In the console: #unique_39/unique_39_Connect_42_section_sf8_uej_28w
- By using API operations: #unique_58
- By using Terraform: alicloud_alikafka_consumer_group

3.3.2 What is the consumer group count?

By default, the consumer group count is twice the topic count. For example, if an instance has 50 topics, you can create up to 100 consumer groups in this instance. To increase the number of consumer groups, you can increase the number of topics. The number of

consumer groups is increased by two each time a topic is added. For more information, see Upgrade the instance configuration.

3.3.3 Why does "Last Consumed At" display 1970 for the partitions of the topic that corresponds to a consumer group?

If the consumer group has not consumed any messages in the partitions of a topic, "Last Consumed At" for these partitions displays 1970.

3.3.4 Why does a deleted consumer group still receive messages?

Symptoms

A deleted consumer group still receives messages.

Causes

In earlier versions, when a consumer group is deleted, its routing information is not completely cleared. Therefore, a deleted consumer group can still receive messages. After you upgrade your instance to the new version, the routing information about any consumer group you deleted in the original version is still retained. To completely clear the routing information, you need to create the previously deleted consumer group and then delete it. Then, you can create a consumer group as needed.

Solutions

1. Upgrade the instance version to the latest version.

   Upgrade the instance version to the latest version

   In the Message Queue for Apache Kafka console, go to the **Instance Details** page. In the **Basic Information** section, view the current **Internal Version**.

   - If **Latest Version** is displayed, you do not need to upgrade the instance version.
   - If **Service Upgrade** is displayed, click **Service Upgrade** to upgrade the version.

2. Create the deleted consumer group, delete it, and then create a new one.

   Go to the **Consumer Groups** page, create the previously deleted consumer group, delete it, and then create a consumer group.

3.3.5 Why do the consumption time points of messages in different partitions vary greatly with an obvious lag or why are they in disorder?

In the Message Queue for Apache Kafka console, when you click **Consumption Status** on the **Consumer Group** tab, **Last Consumed At** appears, which indicates when the last consumed message was stored in the current partition, rather than the time when the message was consumed. For more information about the preceding operation, see View the consumption status.

If the time indicated by **Last Consumed At** for a partition is earlier than that for other partitions in the console, this partition may receive messages from the producer earlier than other partitions.

Each consumer instance in the same consumer group consumes messages in an evenly divided number of partitions. If the number (N) of consumers can be divided by 24 (the default number of partitions) with no remainder, each consumer consumes messages in N /24 partitions. Under this condition, whether messages are evenly consumed depends on whether the producer evenly sends messages to each partition.

If the number (N) of consumers cannot be divided by 24 (the default number of partitions) with no remainder, some consumers may process messages of one more partition than other consumers do. Assume that there are 5 consumers and 24 partitions. Four consumers each consume messages of 5 partitions, and the remaining 1 consumer consumes messages of 4 partitions. The speed of consumption mainly depends on the processing performance of the consumers (clients). If all consumers have the same processing performance, the four consumers that each consume messages of five partitions may consume messages more slowly than the consumer that consumes messages of four partitions.

3.3.6 What does "Last Consumed At" mean in the console?

It means the time when the last consumed message was stored. If no message accumulati on occurs, the time is close to the message sending time.

3.3.7 How do I determine which IP addresses are consuming messages?

Log on to the console and view the consumption status based on consumer groups. Click **Details** to view the **owner** of each partition and then the corresponding IP address.

If the **owner** is empty, the client is offline or in the **Rebalance** state.

3.3.8 Where can I find the best practices for consumption?

For more information, see #unique_66.

3.3.9 Do the NFS cloud disks attached to consumers affect the message processing speed of consumers?

If the consumption speed is affected by the data storage at the client, the data storage is processed synchronously in the main thread where messages are processed. This blocks message pulling and processing.

We recommend that you use independent and separate threads to process messages and store processing results. Messages are pulled and then consumed, even after they are cached. This ensures fast consumption.

A network file system (NFS) affects performance for the following two reasons:

- The NFS does not run fast enough.
- The NFS is a type of shared network storage, which is accessed by multiple nodes and processes through contention. This reduces efficiency and this is why performance is degraded when the number of consumers increases. To solve this problem, you can attach a cloud disk to each node at the consumer end to store processing results independently. This prevents performance degradation even when more consumers contend for NFS resources.

Each attached cloud disk stores data independently. To store processing results centrally in the same NFS, you can use an asynchronous tool or thread to forward the processing results stored in cloud disks to the NFS. This prevents message processing from being blocked by synchronous storage in the NFS.

Asynchronous processing is an effective way to avoid drops in efficiency due to resource access issues.

3.3.10 How can I manage the consumer offset in the console?

This topic describes how to manage the consumer offset when the consumer stops reading messages due to an exception.

The consumer offset is not necessarily committed after a message is consumed. The broker records the consumer offset committed by the consumer.

The committing mechanism depends on the client SDK you use. Generally, the following two mechanisms are supported:

- Automatic committing: The SDK commits the consumer offset of the latest consumed message plus 1 at an interval.
- Manual committing: The consumer offset of the latest consumed message plus 1 is committed through the application.

On the **Consumer Groups** page in the console, you can click **Consumption Status** to view the latest consumer offset committed. The consumer continues the consumption from this consumer offset. For more information, see View the consumption status.

You can manually change the consumer offset recorded by the broker in the console. You can move it backward for repeat consumption or move it forward to skip consumption.

 **Notice:**

To #unique_55 in the console, you need to stop the consumption client first. Otherwise, the reset result may be overwritten by that of the consumer.

4 Tags

4.1 Overview

Tags can identify resources. You can use tags to classify Message Queue for Apache Kafka resources for easy resource search and aggregation. Message Queue for Apache Kafka allows you to bind or unbind tags to or from instances, topics, and consumer groups.

Scenarios

You can use tags to group Message Queue for Apache Kafka resources you created for easy retrieval and batch operations.

Instructions

- Each tag consists of a key-value pair.
- A tag must have a unique tag key.

  For example, an ECS instance is bound to the `city:shanghai` tag. If the instance is subsequently bound to the `city:newyork` tag, the `city:shanghai` tag is automatically unbound from the instance.
- Tag information is not shared across regions. For example, tags created in the China (Hangzhou) region are not visible to the China (Shanghai) region.
- Tags are deleted when they are not bound to any resources.
- For more information about the best practices for tag design, see #unique_71.

Limits

- Up to 20 tags can be bound to a resource.
- A tag can be bound to a maximum of 50 resources.
- A maximum of 20 tags can be bound or unbound at a time.

References

- Bind a tag
- Edit a tag
- Unbind a tag
- Use tags to retrieve resources

4.2 Bind a tag

This topic describes how to bind a tag for Message Queue for Apache Kafka resources such as instances, topics, and consumer groups to facilitate resource classification.

Context

A tag consists of a tag key and a tag value and is used to mark Message Queue for Apache Kafka resources. If your account has multiple Message Queue for Apache Kafka resources that are associated with each other, you can bind tags to these resources for easy classification and management. For more information, see Overview.

Procedure

1. Log on to the Message Queue for Apache Kafka console.
2. In the top navigation bar, select your region.
3. In the left-side navigation pane, click **Tags**.
4. On the **Tags** page, click **Create/Bind Tags**.
5. Perform the following steps according to the **Create/Bind Tags** wizard.

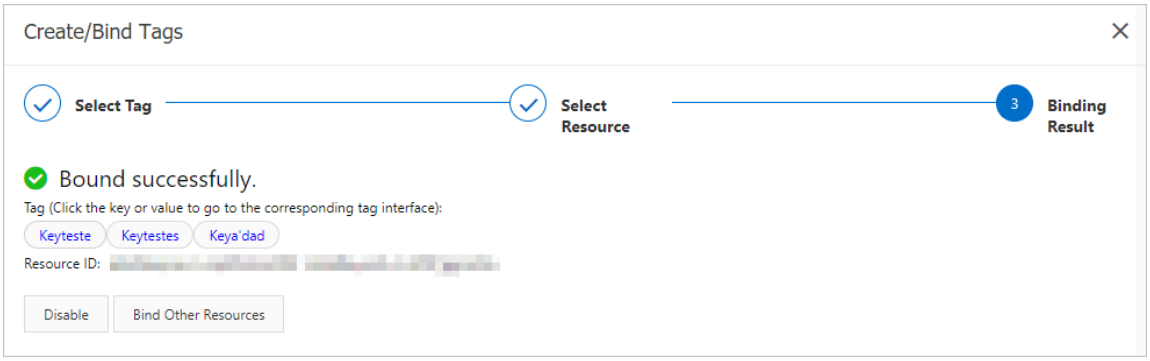
a) **Notice:**
Up to 20 tags are supported and each tag must be unique.

On the **Select Tag** page, select one or more tags, and then click **Next**.

- **Tag Key**: Select an existing key or enter a new key. This parameter is required.
- **Tag Value**: Select an existing value or enter a new value. This parameter is optional.

b) **Notice:**
You can select up to 50 resources of the same type.

On the **Select Resource** page, select one or more resources, and then click **OK**.



What's next

- Unbind a tag
- Edit a tag
- Use tags to retrieve resources

4.3 Edit a tag

This topic describes how to edit a resource tag.

Edit a tag

1. Log on to the Message Queue for Apache Kafka console.
2. In the top navigation bar, select your region.
3. In the left-side navigation pane, you can:
   - Edit tags of a Message Queue for Apache Kafka instance

     Click **Overview**. On the **Overview** page, find the target instance, and choose **Tags** > **Edit Tag**.
   - Edit tags of a topic

     Click **Topics**. On the **Topics** page, select the instance of the target topic, find the target topic, and then choose **Tags** > **Edit Tag**.
   - Edit tags of a consumer group

     Click **Consumer Groups**. On the **Consumer Groups** page, select the instance of the target consumer group, find the target consumer group, and then choose **Tags** > **Edit Tag**.
4.  **Notice:**

Up to 20 tags can be bound to each resource. You can bind or unbind no more than 20 tags at a time.

In the **Edit Tag** dialog box, you can:

- Delete a tag

  In the dotted line section, find the target tag and then click the **Delete** icon.
- Bind a tag
  - To bind an existing tag, click **Select Tag**. Select the target tag key from the **Tag Key** list and select the target tag value from the **Tag Value** list.
  - To bind a new tag, click **Create Tag**, enter the tag key and tag value, and then click **OK**.

5. Click **OK**.

4.4 Unbind a tag

When tags are bound to an unsuitable resource or are no longer applicable to resource management and retrieval, you can unbind tags from resources in batches.

Prerequisites

Bind a tag

Context

- You can unbind up to 20 tags at a time.
- After a tag is unbound, if the number of resources bound to the tag is zero, the tag is automatically deleted.

Procedure

1. Log on to the Message Queue for Apache Kafka console.
2. In the left-side navigation pane, click **Tags**.
3. On the **Tags** page, select a tag key from the tag key list on the left.
4. Select a tag value in the tag value section on the right.

   Note:
   If no tag value is available, **Null** is displayed.
5. Select one or more resources, and choose **Batch Operation** > **Unbind Tag**.
6. On the **Tags** page, click **Refresh**.

4.5 Use tags to retrieve resources

This topic describes how to retrieve a resource by using its bound tag.

Prerequisites

Bind a tag

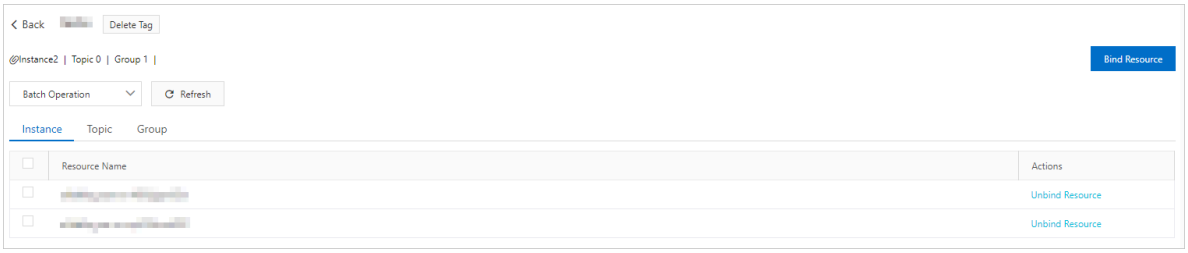
Procedure

1. Log on to the Message Queue for Apache Kafka console.
2. In the top navigation bar, select your region.
3. In the left-side navigation pane, click **Tags**.
4. On the **Tags** page, select a tag key from the tag key list on the left, and then select a tag value in the tag value section on the right.

   Note:

   If no tag value is available, **Null** is displayed.

5. In the tag section on the right, select the tag of the resource you want to view.



5 Alerts

5.1 Monitor resources and set alerts

Message Queue for Apache Kafka allows you to monitor the resources created under your account, including instances, topics, and consumer groups. This helps you know the status of these resources in minutes.

Currently, the following metrics are provided for various resource types:

- Instance metrics:
  - Production Traffic of Instance Messages (bytes/s)
  - Consumption Traffic of Instance Messages (bytes/s)
  - Instance Disk Usage (%)

  > **Note:**
  > Instance Disk Usage (%) indicates the maximum disk usage of each node of the instance.
- Topic metrics:
  - Production Traffic of Topic Messages (bytes/s)
  - Consumption Traffic of Topic Messages (bytes/s)
- Consumer group metrics:
  - Total Messages Not Consumed by Consumer Group

You can also set alert rules for these metrics. Message Queue for Apache Kafka connects to CloudMonitor. Therefore, you can directly create alert rules in the CloudMonitor console. When the metric value exceeds the alert threshold you set, CloudMonitor notifies you through SMS, email, TradeManager, or DingTalk chatbot to help you deal with exceptions in a timely manner.

View monitoring data

No matter whether you set alerts, you can view the statistics of resource metrics in the Message Queue for Apache Kafka console.

- Prerequisites
  - You have created an instance, a topic, and a consumer group. For more information, see #unique_39.
  - The consumer group you created has subscribed to the topic. For more information, see Message Queue for Apache Kafka demos.
- Procedure
  1. Log on to the Message Queue for Apache Kafka console. In the top navigation bar, select the region where the target resource is located, for example, **China (Hangzhou)**.
  2. In the left-side navigation pane, click **Monitoring and Alerts**. On the **Monitoring and Alerts** page, select the target resource.
     - To view the monitoring data of an instance, click the **Instance** tab.
     - To view the monitoring data of a topic or consumer group, select the instance of the topic or consumer group on the top of the page, and then click **Topic** or **Consumer Group**.
  3. Find the target resource, and click **View Monitoring** in the **Actions** column.

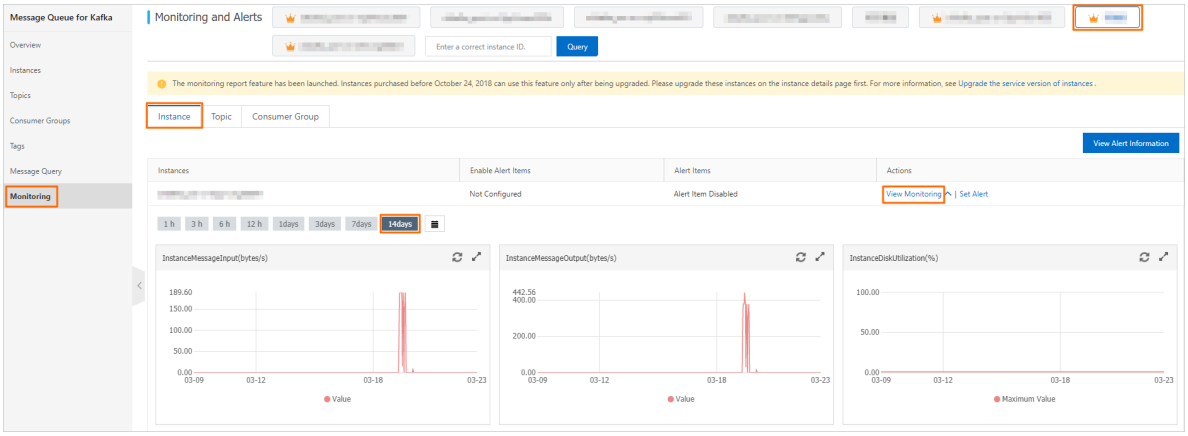
     You can view the data of the last 1 hour, 3 hours, 6 hours, 12 hours, 1 day, 3 days, 7 days, or 14 days, or click the time range picker to select a time range.

     If you want to specify a time range, you can view data for the last 31 days at most (data generated 31 days earlier is not retained), that is, in the time range picker, **End Time** is the current system time, and **Start Time** is up to 31 days ago. If **End Time** is not the current system time, you can view data for any 7 days in the last 31 days.

     Note:
     The data aggregation cycle of a metric is 1 minute, that is, the metric is calculated once every minute. The calculated value in bytes/s can be considered as the average value of the metric within 1 minute.

- Verification

  Corresponding metrics and monitoring data appear under the resource.



Set an alert

You can create alert rules to receive notifications about exceptions in a timely manner.

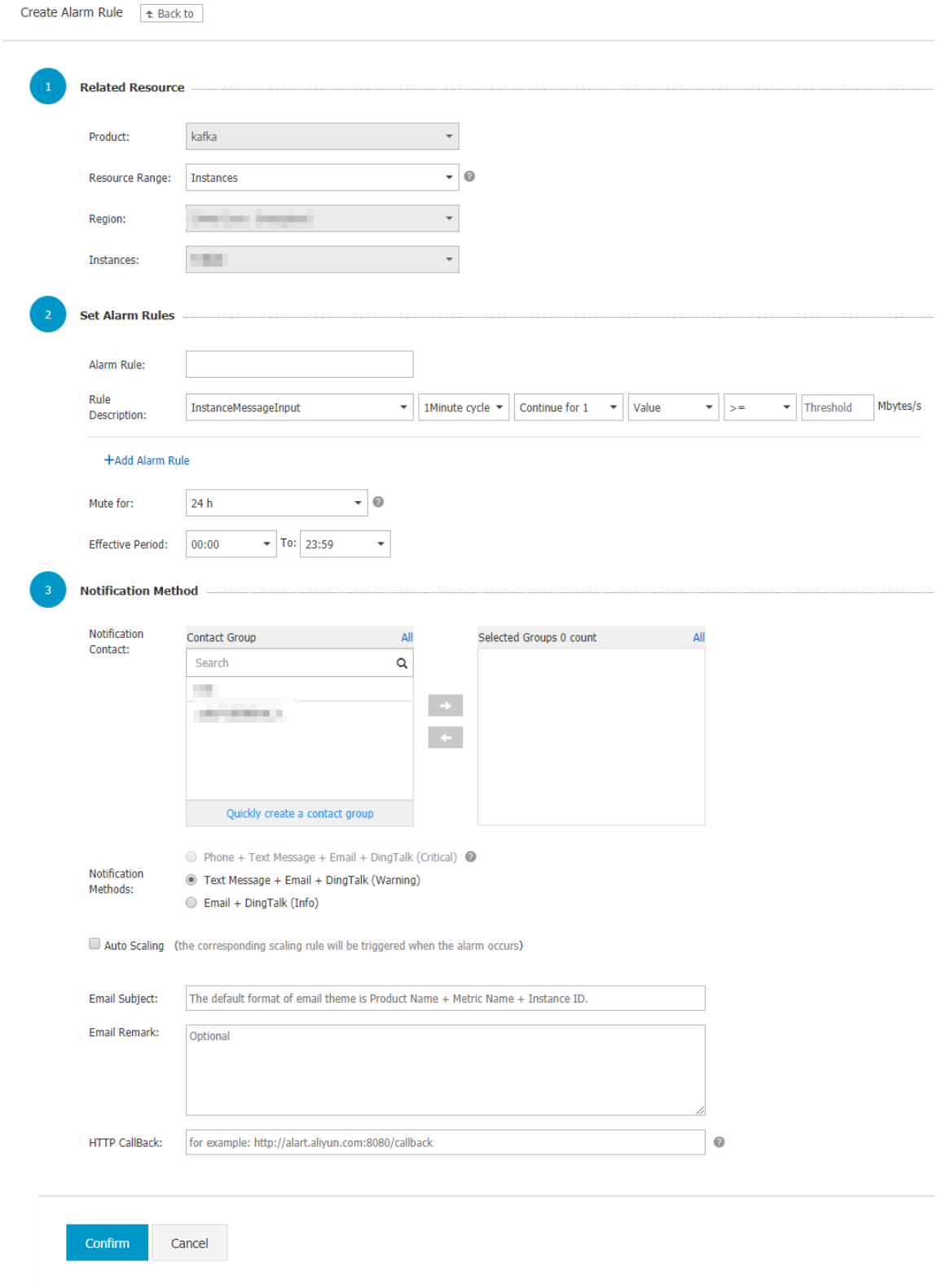
- Prerequisites

  You have created an instance, a topic, and a consumer group. For more information, see #unique_39.

- Procedure

  1. Log on to the Message Queue for Apache Kafka console. In the top navigation bar, select the region where the target resource is located, for example, **China (Hangzhou)**.
  2. In the left-side navigation pane, click **Monitoring and Alerts**. On the **Monitoring and Alerts** page, select the target resource.
     - To set an alert for an instance, click the **Instance** tab.
     - To set an alert for a topic or consumer group, select the instance of the topic or consumer group on the top of the page, and then click **Topic** or **Consumer Group**.
  3. Find the target resource, and click **Set Alerts** in the **Actions** column.

     The page is redirected to the **Create Alarm Rule** page of the CloudMonitor console.
  4. On the **Create Alarm Rule** page, set the alert rule and notification method. For more information, see #unique_77.



Notice:

- The system does not support batch alert setting across instances.

- When you set an alert rule for metrics **Production Traffic of Topic Messages** and **Consumption Traffic of Topic Messages**, we recommend that you do not select **Any** in the **Topic** field. When **Any** is selected, all topics are selected.
- Avoid using "between" and multiple expressions when setting specific rules.
- In CloudMonitor, you can set up to 50 alert rules for free. If you want to set more rules, you need to upgrade your CloudMonitor instance.

- Verification

  For more information, see View alert information.

View alert information

You can view the alert rules you created and the corresponding alert information.

- Prerequisites

  You have created an alert rule. For more information, see Set an alert.

- Procedure

  1. Log on to the Message Queue for Apache Kafka console. In the top navigation bar, select the region where the target resource is located, for example, **China (Hangzhou)**.
  2. In the left-side navigation pane, click **Monitoring and Alerts**. On the **Monitoring and Alerts** page, select the target resource.

     - To view the alert information of an instance, click the **Instance** tab.
     - To view the alert information of a topic or consumer group, select the instance of the topic or consumer group on the top of the page, and then click **Topic** or **Consumer Group**.
  3. On the **Monitoring and Alerts** page, you can view the alert details in either of the following ways:

     - Click **View Alert Information**.

       The page is redirected to the **Alarm Rules** page in the CloudMonitor console. By default, all alert rules of Message Queue for Apache Kafka and their statuses are displayed. You can view, modify, disable, enable, and delete alert rules.
     - Click **Alert Items: X** in **Alert Items** of a resource. X indicates the number of alert rules you have set for this resource.

       In the **Alert Items** window, view all the alert rules of the resource and corresponding alert information. Find the target alert rule and click **View** in the **Actions** column. The Basic Information page of this alert rule appears in the CloudMonitor console. On this page, you can view all the information of this alert rule, and can modify, disable, enable, and delete the alert rule.

5.2 Monitoring and alerts

5.2.1 Why do I need to upgrade my instance to support monitoring and alerts?

You can view and configure the monitoring and alerts feature in the console. However, Message Queue for Apache Kafka instances deployed before November 2018 do not support reporting the monitoring and alerts data. To use the reporting function, you need to upgrade your Message Queue for Apache Kafka instance.

For more information about how to upgrade your Message Queue for Apache Kafka instance, see Upgrade the instance version.

5.2.2 What can I do if Insufficient Data is displayed for Status in the Alert Items dialog box?

Symptoms

Log on to the Message Queue for Apache Kafka console and go to the **Monitoring and Alerts** page. In the **Alert Items** column, click **Alert Items: x**. In the **Alert Items** dialog box, the **Status** column displays **Insufficient Data**.

Causes

The Message Queue for Apache Kafka instance is not upgraded to the version that supports alert data reporting after alert rules are configured.

Solutions

1. Log on to the Message Queue for Apache Kafka console. In the top navigation bar, select the region where the instance is located.
2. In the left-side navigation pane, click **Instances**.
3. On the **Instance Details** page, click the ID of the target instance.
4. In the **Basic Information** section, when **Internal Version** is **Service Upgrade**, click **Service Upgrade** to upgrade the instance. When Internal Version is **Latest Version**, do not upgrade the instance.

   The instance version will be upgraded to the most suitable internal version based on your specific instance conditions.
5. In the **Upgrade** dialog box, set the following parameters so that we can contact you when an error occurs during the upgrade:
   - **Name**
   - **Emergency Phone Number**
6. Click **Upgrade**.

Notice:

- If the client does not support the reconnection mechanism (enabled by default), the client may be unavailable after being disconnected. Ensure that the consumer supports the reconnection mechanism.

- The upgrade will take about 15 minutes. The service will not be interrupted during the upgrade and the business will not be affected.

5.2.3 How do I monitor open-source Apache Kafka?

For more information, see the following links:

- How to Monitor Kafka
- Monitoring Kafka performance metrics

5.2.4 Why can I view monitoring information with my Alibaba Cloud account, but cannot do so as a RAM user?

RAM users can view monitoring information after being granted the AliyunCloudMonitorReadOnlyAccess permission by the Alibaba Cloud account. For more information, see #unique_83/unique_83_Connect_42_section_hxj_k93_hbr.

5.2.5 Why am I unable to log on to the instance with Message Queue for Apache Kafka deployed?

Message Queue for Apache Kafka is fully hosted and maintained by the Message Queue for Apache Kafka professional team. You do not need to log on to the instance. Some basic cluster information is transparently transmitted through monitoring and alerts.

5.2.6 Why did I receive a message accumulation alert after I deleted the consumer group?

The consumer group is logically deleted from the console. However, the information such as the consumer offset on the broker is not deleted. The accumulation alert is handled based on the consumer offset. Therefore, you still receive the accumulation alert.

If you do not want to receive the accumulation alert after deleting the consumer group, you can perform the following operations:

- Disable the accumulation alert.
- Wait until the consumer offset expires.

  The Message Queue for Apache Kafka consumer offset is stored in an internal topic and cannot be directly deleted. If the consumer offset is not updated after the message retention period ends, it will be cleared due to expiration.

  Notice:

Earlier users who have not enabled the clearing mechanism need to upgrade the broker to the latest version on the **Instance Details** page.

5.2.7 What are dead partitions and false accumulation?

A dead partition is a partition to which no data has been sent for a long time. Dead partitions do not affect usage but interfere with monitoring and alerts. When dead partitions exist, the system always shows that messages are accumulated. The reason is that no data has been sent to dead partitions for a long time. The consumer no longer commits the offset, which expires after the retention period ends. When a partition has no offset recorded on the broker, by default, the maximum number of accumulated messages is calculated by subtracting the maximum offset by the minimum offset. False accumulation is reported when a dead partition still contains messages.

When accumulated to a large size, these messages are deleted upon expiration. When the number of accumulated messages is small, it may take a long time to delete these messages because the broker retains at least a 1 GB segment.

The solution is to send messages as evenly as possible and make sure that each partition has data. If the amount of service data is small, regularly send heartbeat data to each partition.

5.2.8 Why is the number of accumulated messages indicated by an alert different from that displayed in the console?

Generally, this is due to dead partitions. When a partition has no offset recorded on the broker, by default, the console calculates the number of accumulated messages by subtracting the maximum offset by the minimum offset. However, the accumulation alert takes the maximum offset as the number of accumulated messages for a dead partition by default. Currently, this problem is solved for the broker in a new version to be released , where the number of accumulated messages is calculated by subtracting the maximum offset by the minimum offset.

6 Query messages

If message consumption is abnormal at a certain time point, you can query which messages were sent at that time point and the content of these messages. The Message Queue for Apache Kafka console provides two message query methods: query by offset and query by time, to improve the troubleshooting efficiency.

Background

You can query messages according to the following recommended methods:

- If you can query logs to get the ID of the partition to which the message is sent and the message offset, we recommend that you use Query by offset.
- If you do not know the message offset but know when it is sent, we recommend that you use Query by time.

Notice:

- A maximum of 1 KB content can be displayed in the console for each message. Content that exceeds 1 KB is automatically truncated. If you want to view the complete message , download the message.

  Currently, Message Queue for Apache Kafka instances of only the Professional Edition allow you to download messages of up to 10 MB. Buy Now>>
- If you use an instance of the Standard Edition, you can query up to 256 KB messages and display up to 10 messages.
  - If the size of 10 messages exceeds 256 KB, only the content within 256 KB is displayed in the console.
  - If the size of 10 messages is less than 256 KB, up to 10 messages are displayed in the console. See the actual consumption data of the consumer.
- If you use an instance of Professional Edition, you can query up to 10 MB messages and display up to 30 messages.
  - If the size of 30 messages exceeds 10 MB, only the content within 10 MB is displayed in the console.
  - If the size of 30 messages is less than 10 MB, up to 30 messages are displayed in the console. See the actual consumption data of the consumer.

  For more information about the instance version, see #unique_9.

- Results that you can query is related to the information cleanup policy of Message Queue for Apache Kafka. The cleanup policy is described as follows:
  - Messages that exceed the storage duration are deleted, but at least one storage file is retained. For example, if the message storage duration is 72 hours, all messages that exceed 72 hours are cleared, but the last storage file is retained. Even if the messages in this file are all stored for more than 72 hours, the messages in this file can still be queried.
  - If the total message size exceeds 85% of the capacity of the message storage disk, the earliest message is cleared until the remaining capacity drops below 85%.

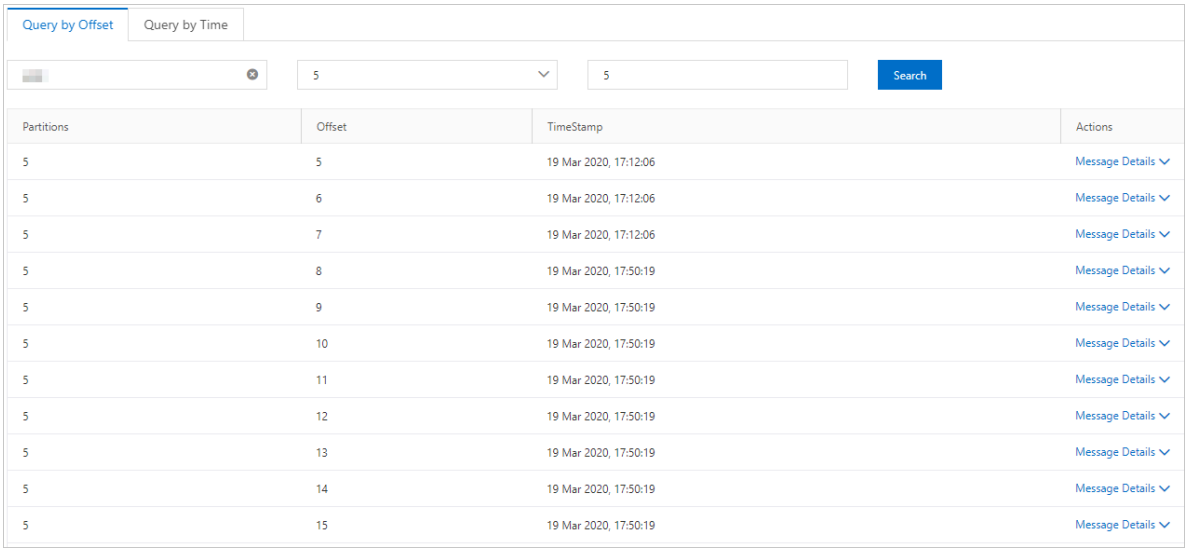
Query by offset

In Message Queue for Apache Kafka, an offset maps a message. You can specify an offset to query the corresponding message after determining the location of the message to be queried.

1. Log on to the Message Queue for Apache Kafka console. In the top navigation bar, select the region where the instance you want to query is located.
2. In the left-side navigation pane, click **Message Query**. On the **Message Query** page, find the target instance, and click the **Query by Offset** tab.

3. In the three fields, enter the topic name, select the target partition, enter the offset according to the instructions on the page, and then click **Search**.

   On the **Query by Offset** tab, up to 10 messages starting from the specified offset are displayed. For example, if both the specified partition and offset are 5, the returned results are messages starting from the offset 5, as shown in the following figure:



   Eight messages are returned, indicating that the maximum offset of the partition is 12 or that the size of the returned messages exceeds 256 KB.

   The fields in the search result are described as follows:

   - **Partition**: The value is same as the partition ID you selected in Step 3.
   - **Offset**: Up to 10 offsets starting from the offset you specified in Step 3 are displayed.
   - **TimeStamp**: The value is the timestamp of the producer when the message is sent or that you specified in ProducerRecord. If this field is not configured, the system time when the message is sent is used by default. If this field is configured, it is displayed based on the configured value. If the value is 1970/x/x x:x:x, the sending time is set to 0 or an invalid value. This time cannot be set in Apache Kafka clients of 0.9 and earlier versions.

4. Optional: Click **Message Details** in the **Actions** column. The fields are described as follows:

   - **Key(size:XXB)**: the size of the serialized message key or value. The value is serialized KeySize/serializedValueSize in ConsumerRecord.
   - **Value(size:XXB)**: the specific content of the queried message, which has been forcibly converted to a string.

5. **For Professional Edition only**: Click **Download Message** next to the key or value to download the message. You can download messages of up to 10 MB. Content exceeding 10 MB is not displayed.

Query by time

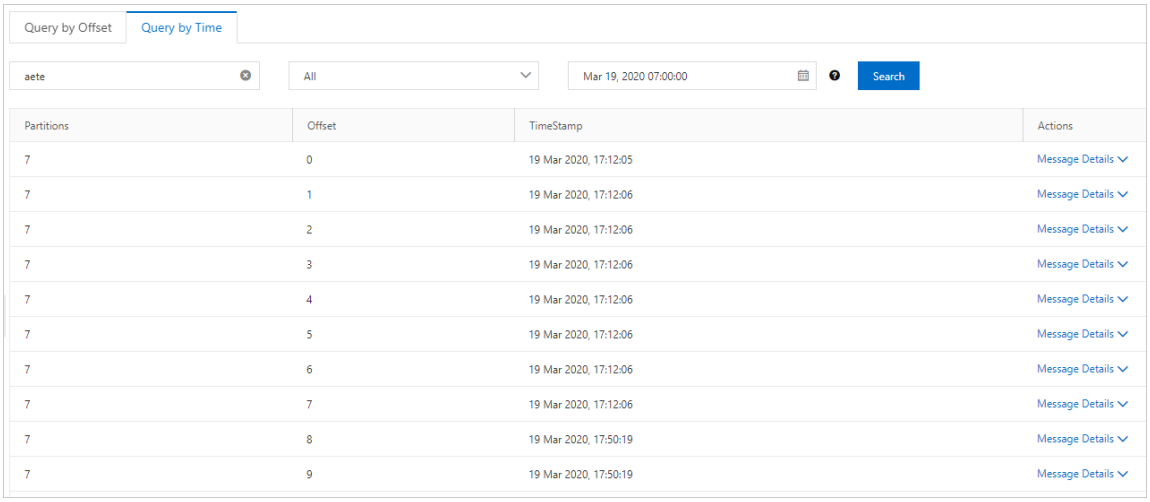
You can query messages in all partitions by time. If you are not sure about the message location, but know the time range in which messages are sent, you can specify a time point in the time range to query messages near the message sending time point.

1. Log on to the Message Queue for Apache Kafka console. In the top navigation bar, select the region where the instance you want to query is located.
2. In the left-side navigation pane, click **Message Query**. On the **Message Query** page, find the target instance, and click the **Query by Time** tab.
3. In the three fields, enter the topic name, select the target partition, enter the time point, and then click **Search**.

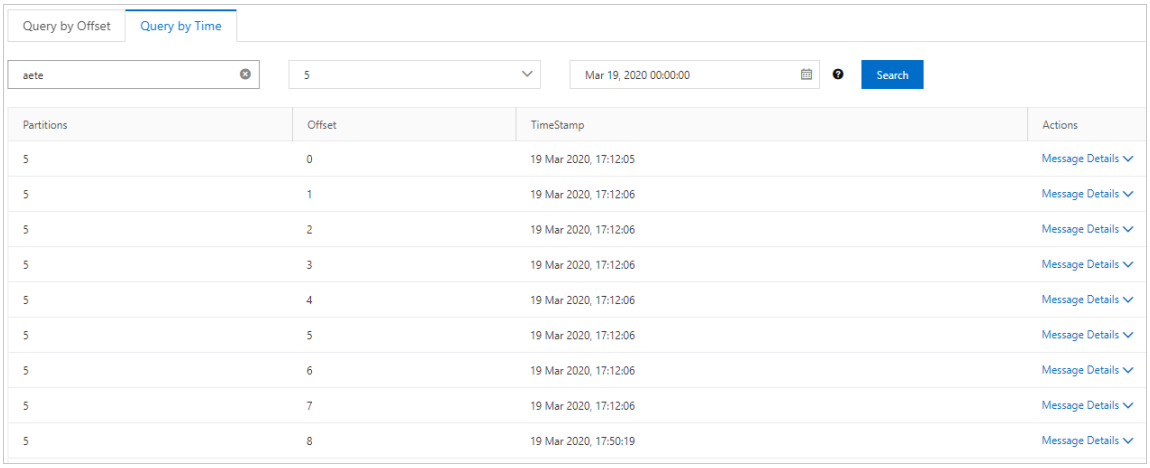
   On the **Query by Time** tab, the search result is displayed. The partition value affects the search result.

   - If you select **All** for **Partition**, messages near the specified time point in the randomly displayed partition of the instance are displayed. A total of 10 messages are

displayed at most. For example, if you select **All** for Partition and set the time to 2019-05-07 00:00:00, the search result is shown in the following figure.



- If a partition is specified, up to 10 messages near the specified time point in the partition are displayed. For example, if you select 5 for Partition and set the time to 2019-05-07 00:00:00, the search result is shown in the following figure.



The fields in the search result are described as follows:

- **Partition**: The value displayed depends on whether a partition is specified in Step 3. If **All** is selected, partitions of the instance are randomly displayed.
- **Offset**: If you specify a partition in Step 3, the offsets near the specified time point in this partition are displayed. If **All** is selected for Partition in Step 3, offsets near the specified time point in any partition of the instance are randomly displayed. A maximum of 10 offsets are displayed regardless of whether a partition is specified.
- **TimeStamp**: The value is the timestamp of the producer when the message is sent or that you specified in ProducerRecord. If this field is not configured, the system time when the message is sent is used by default. If this field is configured, it is displayed

based on the configured value. If the value is 1970/x/x x:x:x, the sending time is set to 0 or an invalid value. This time cannot be set in Apache Kafka clients of 0.9 and earlier versions.

4. Optional: Click **Message Details** in the **Actions** column. The fields are described as follows:
   - **Key(size:XXB)**: the size of the serialized message key or value. The value is `serializedKeySize`/`serializedValueSize` in `ConsumerRecord`.
   - **Value(size:XXB)**: the specific content of the queried message, which has been forcibly converted to a string.
5. **For Professional Edition only**: Click **Download Message** next to the key or value to download the message. You can download messages of up to 10 MB. Content exceeding 10 MB is not displayed.