Alibaba Cloud

Message Queue for Apache Kafka User guide

Document Version: 20210622

C-J Alibaba Cloud

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.

- You shall download and obtain this document from the Alibaba Cloud website or other Alibaba Cloudauthorized 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 because of product version upgrade, adjustment, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and an updated version of this document will be released through Alibaba Cloud-authorized channels from time to time. You should 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 this document based on the "status quo", "being defective", and "existing functions" of its products and services. 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 take legal responsibility for any errors or lost profits incurred by any organization, company, or individual arising from download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, take responsibility for any indirect, consequential, punitive, contingent, 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 and/or its affiliates 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 directly contact Alibaba Cloud for any errors of this document.

Document conventions

| Style | Description | Example |
|--------------|--|--|
| A Danger | 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. |
| O Warning | 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. |
| C) Notice | 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. |
| ? Note | 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 listinstanceid Instance_ID |
| | | |
| [] or [a b] | This format is used for an optional value, where only one item can be selected. | ipconfig [-all -t] |

Table of Contents

| 1.Instances | 80 |
|--|----|
| 1.1. View endpoints | 08 |
| 1.2. Upgrade instance specifications | 08 |
| 1.3. Modify the message configuration | 11 |
| 1.4. Upgrade the instance version | 13 |
| 1.5. Configure the whitelist | 15 |
| 1.6. Rebalance the topic traffic | 18 |
| 1.7. View task execution records | 21 |
| 1.8. FAQ | 23 |
| 1.8.1. Which open-source Apache Kafka versions are supporte | 23 |
| 1.8.2. How can I select the instance edition? | 23 |
| 1.8.3. How can I obtain the VPC information required for de | 24 |
| 1.8.4. What is impact of upgrading brokers? | 24 |
| 1.8.5. Why am I unable to change the region or network typ | 24 |
| 1.8.6. When are old messages deleted in Message Queue for | 25 |
| 1.8.7. How can I fast test whether the Message Queue for Ap | 25 |
| 1.8.8. Are delayed messages supported? | 29 |
| 1.8.9. Can I access a Message Queue for Apache Kafka insta | 29 |
| 1.8.10. Can we connect Message Queue for Apache Kafka ins | 29 |
| 1.8.11. Do I have to select the 172.16.0.0/12 CIDR block for a | 30 |
| 1.8.12. Is data compression supported? | 31 |
| 2.Topic | 32 |
| 2.1. View partition status | 32 |
| 2.2. Topics | 33 |
| 2.2.1. Why is the total number of topics (partitions) limited? | 33 |
| 2.2.2. Why am I unable to reduce the number of partitions f | 33 |

| 2.2.3. How many partitions are added each time I purchase | 33 |
|--|----|
| 2.2.4. What are the types of topic storage? | 33 |
| 2.2.5. Can I get one topic free for each purchased topic for | 33 |
| 2.2.6. Why does "Last Updated At" display 1970 for the part | 34 |
| 2.2.7. Why am I unable to reduce partitions after deleting a | 34 |
| 2.2.8. Is the compact log cleanup policy supported? | 34 |
| 3.Consumer groups | 35 |
| 3.1. Reset consumer offsets | 35 |
| 3.2. View the consumption status | 36 |
| 3.3. Consumer groups | 40 |
| 3.3.1. What is the consumer group count? | 40 |
| 3.3.2. Why is 1970 displayed for "Last Consumed At" for the | 40 |
| 3.3.3. Why does a deleted consumer group still receive messa | 40 |
| 3.3.4. Why do the consumption time points of messages in d | 41 |
| 3.3.5. What does "Last Consumed At" mean in the console? | 42 |
| 3.3.6. How do I determine which IP addresses are consuming | 42 |
| 3.3.7. Where can I find the best practices for consumption? | 42 |
| 3.3.8. Do the NFS cloud disks attached to consumers affect t | 42 |
| 3.3.9. How can I manage the consumer offset in the console? | 43 |
| 4.Tags | 44 |
| 4.1. Overview | 44 |
| 4.2. Bind a tag | 44 |
| 4.3. Edit a tag | 45 |
| 4.4. Unbind a tag | 46 |
| 4.5. Use tags to retrieve resources | 47 |
| 5.Monitoring and alerting | 48 |
| 5.1. Monitor resources and set alerts | 48 |
| 5.2. FAQ | 51 |

| 5.2.1. Why do I need to upgrade my instance to support mo 52 |
|---|
| 5.2.2. What can I do if Insufficient Data is displayed for Stat |
| 5.2.3. How do I monitor open-source Apache Kafka? 53 |
| 5.2.4. Why can I view monitoring information with my Aliba 53 |
| 5.2.5. Why am I unable to log on to the instance with Mess 53 |
| 5.2.6. Why did I receive a message accumulation alert after 53 |
| 5.2.7. What are dead partitions and false accumulation? 54 |
| 5.2.8. Why is the number of accumulated messages indicated 54 |
| 6.Connectors 55 |
| 6.1. Overview 55 |
| 6.2. Enable the connector feature 57 |
| 6.3. Create a connector 58 |
| 6.3.1. Create a Function Compute sink connector |
| 6.3.2. Create a MaxCompute sink connector 73 |
| 6.3.3. Create an OSS sink connector84 |
| 6.3.4. Create an Elasticsearch sink connector |
| 6.4. View task configurations of a connector 105 |
| 6.5. View connector logs 106 |
| 6.6. Suspend a connector 106 |
| 6.7. Resume a connector 107 |
| 6.8. Delete a connector 108 |
| 6.9. Modify connector configurations 108 |
| 6.10. Test a connector 110 |
| 6.11. Enable Internet access for a connector |
| 7.Migration 113 |
| 7.1. Overview 113 |
| 7.2. Evaluate specifications 116 |
| 7.3. Migrate topics 117 |

| 7.3.1. Migrate topic metadata from a user-created Kafka clust | 117 |
|---|-----|
| 7.3.2. Migrate topic metadata between Message Queue for A | 119 |
| 7.4. Migrate consumer groups | 121 |
| 7.4.1. Migrate consumer group metadata from a user-created | 121 |
| 7.4.2. Migrate consumer group metadata between Message Q | 123 |
| 7.5. Migrate data | 125 |
| 7.5.1. Migrate data to the cloud | 125 |
| 7.6. View the migration progress | 128 |
| 8.Query messages | 129 |
| | |

1.Instances 1.1. View endpoints

To send and subscribe to Message Queue for Apache Kafka messages by using the software development kit (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 Default Endpoint which 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.
- 2. In the top navigation bar, select a region.
- 3. In the left-side navigation pane, click Instances.
- 4. On the Instance Details page, click the target instance.
- 5. Check the endpoint information in the Basic Information section.

| Basic Information | Upgrade Instance |
|---------------------------------|-----------------------------|
| Instance ID: | Instance Name: test2 🖉 |
| Instance Type: Standard Edition | Cluster Type: VPC Instance |
| Peak Traffic: 20 MB/s | Disk Size: 500 GB |
| Disk Type: Ultra Disk | Instance Type: VPC Instance |
| VPC ID: | VSwitch ID: |
| Zone: zonea | Topics: 50 (Partitions 400) |
| Public Network Traffic: Mbps | Open-Source Edition 0.10.2 |
| Internal Version Latest Version | |
| Default Endpoint: | |

? Note

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

More information

Use the default endpoint to send and subscribe to messages

1.2. Upgrade instance specifications

This topic describes how to upgrade the specifications of an instance in the Message Queue for Apache Kafka console. You can upgrade the instance edition, traffic specification, disk capacity, and topic specification.

Scenarios

- The disk usage of your Message Queue for Apache Kafka instance is high and will affect service running.
- The traffic of your Message Queue for Apache Kafka instance continuously exceeds the traffic specification that you purchase. As a result, the instance fails to meet your business needs.

Procedure

- 1. Log on to the Message Queue for Apache Kafka console.
- 2. In the top navigation bar, select the region where your instance is located.
- 3. In the left-side navigation pane, click **Instances**.
- 4. On the **Instances** page, click the name of the instance that you want to manage.
- 5. On the Instance Details page, click Upgrade Instance in the upper-right corner of the Basic Information section.
- 6. In the **Instance Upgrade Risks** dialog box, read the risks of upgrading an instance, 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.**.
- 7. On the **Upgrade/Downgrade** page, change the specifications.

| Parameter | Description |
|------------------|---|
| Edition | You can upgrade an instance from the Standard Edition to the Professional Edition. You can upgrade the traffic specification of a Professional Edition instance to a higher specification. You cannot downgrade an instance from the Professional Edition to the Standard Edition. For more information about editions and pricing of Message Queue for Apache Kafka instances, see Billing. |
| Internet Traffic | Public traffic is divided into read traffic and write traffic. The maximum read traffic and maximum write traffic provided by Message Queue for Apache Kafka are the same. Select a bandwidth based on your peak read or write traffic, whichever is higher. This billable item applies only to instances of the Internet and VPC type. |

| Parameter | Description |
|-----------------------|---|
| Traffic Specification | When you upgrade the traffic specification of your instance, the following limits apply: Edition Standard Edition: You can select a traffic specification that supports the maximum traffic of 120 MB/s. If you need a traffic specification that supports the maximum traffic of more than 120 MB/s, upgrade your instance to the Professional Edition. Then, upgrade the traffic specification. Professional Edition (High Write): You can select a traffic specification that supports the maximum traffic of 2,000 MB/s. Professional Edition (High Read): You can select a traffic specification that supports the maximum read traffic of 150 MB/s and the maximum write traffic of 30 MB/s. Disk type Ultra disk: If the maximum traffic supported by the traffic specification that you want to select exceeds 120 MB/s, the corresponding cluster will be scaled out. After the upgrade is complete, you must rebalance topic traffic. For more information, see Rebalance the topic traffic. SSD: If the maximum traffic supported by the traffic specification that you want to select exceeds 300 MB/s, the corresponding cluster will be scaled out. After the upgrade is complete, you must rebalance topic traffic. For more information, see Rebalance the topic traffic. For more information, see Rebalance the topic traffic. |
| Disk Capacity | The default recommended disk capacity varies with the traffic specification. If you adjust the traffic specification, the disk capacity is adjusted accordingly. You can adjust the disk capacity based on your business needs. |
| Supported Topics | Each time you purchase a topic, a quota of 16 partitions is added. The number of topics that you can create on a Professional Edition instance is twice the number of topics that you purchase. |

♥ Notice

- You can only upgrade but cannot downgrade instance specifications.
- When you upgrade instance specifications, brokers in the cluster restart one by one. This may bring the following risks:
 - The client will temporarily disconnect and reconnect, which may cause a few errors.
 - The messages that have been sent will not be lost after the upgrade. If a message fails to be sent during the upgrade, we recommend that you resend it. You can configure a retry mechanism on the client.
 - The upgrade lasts for about 30 minutes. More time is consumed for a larger increase in disk capacity. The service is not interrupted but messages may be distributed to a different partition for consumption. Therefore, evaluate the impact on your business before you proceed. We recommend that you upgrade instance specifications during off-peak hours.
- 8. Read and select the terms of service. Then, click Buy Now.

(?) **Note** After you upgrade the specifications of your instance, the time when your order will take effect appears on the upgrade order page.

Result

On the Instance Details page, the specifications after the upgrade appear.

1.3. Modify the message configuration

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

Prerequisites

A Message Queue for Apache Kafka instance is purchased and deployed, and it is in the Running state.

Procedure

- 1. Log on to the Message Queue for Apache Kafka console.
- 2. In the top navigation bar, select the region where your instance is located.
- 3. In the left-side navigation pane, click Instances.
- 4. On the **Instances** page, click the name of the instance that you want to manage.
- 5. In the Configurations section of the Instance Details page, click Configuration Change.
- 6. In the **Configuration Change** dialog box, set the parameters and click **Change**.

| Configuration Change | × |
|------------------------------------|--|
| Configurations | |
| Message Retention Period: 72 Hours | Maximum Message Size: 1 MB |
| | Change Cancel |
| Parameter | Description |
| Message Retention Period | The maximum message retention period when the disk capacity is sufficient. When the disk usage reaches 85%, the disk capacity is insufficient. In this case, the system deletes messages from the earliest stored ones to ensure service availability. Valid values: 24 hours to 480 hours. Default value: 72 hours. |
| Maximum Message Size | The maximum size of a message that you can send and receive in Message Queue for Apache Kafka . The maximum message size is 10 MB for instances of both the Standard Edition and the Professional Edition. Before you modify the configuration, make sure that the new value matches the configuration on the producer and consumer. |

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

| Alerts | × |
|---|---|
| This operation will cause servers in the cluster to restart one by one, which may impose the following risks: | |
| 1. This operation does not interrupt the service but may cause temporary service unavailability. | |
| If the client is not configured with automatic reconnection policies, it may fail to reconnect after bein disconnected. This is the most likely problem this operation may cause. | g |
| After the upgrade, check whether the application is running normally. | |
| I am aware of the service unavailability risk caused by restart of servers in the cluster. | |
| I am aware of the risks. Cancel | |

What's next

View task execution records

1.4. Upgrade the instance version

This topic describes how to upgrade the version of a Message Queue for Apache Kafka instance.

Prerequisites

The Message Queue for Apache Kafka instance is in the Running state.

Context

• Upgrade the major version of an instance

A major version upgrade is an upgrade from one major version to another major version. For example, you can upgrade a Message Queue for Apache Kafka instance from version 0.10.x to version 2.x.

? Note

- By default, the major version deployed for a Message Queue for Apache Kafka instance is 0.10.x. The default version is 0.10.2 for a new instance and 0.10 for an old instance. Version 0.10 may trigger bugs such as deadlocks and frequent rebalancing. We recommend that you upgrade the instance from version 0.10 to version 0.10.2. For more information about the upgrade, see Minor version upgrade.
- Message Queue for Apache Kafka instances support major versions 0.10.x and 2.x. Major version 0.10.x provides 0.10 and 0.10.2, while major version 2.x provides only 2.2.0.
- Upgrade the minor version of an instance

A minor version upgrade is an upgrade from one minor version to another minor version. For example, you can upgrade a Message Queue for Apache Kafka instance from version 0.10 to version 0.10.2, or from version 0.10.2 to the 0.10.2 kernel-optimized version.

Major version upgrade

The major version of a Message Queue for Apache Kafka Standard Edition instance cannot be upgraded from 0.10.x to 2.x. To upgrade the major version, you must first upgrade the instance from the Standard Edition to the Professional Edition. For more information about how to upgrade the instance edition, see Upgrade instance specifications.

♥ Notice

- The upgrade is free of charge and compatible with the existing SDK and API.
- To avoid impact on your business during the upgrade, make sure that your client supports automatic reconnection and can handle disconnections. By default, the client supports automatic reconnection.
- The upgrade takes about 25 minutes. During the upgrade, the service will not be interrupted, and your business will not be affected in normal cases.
- Instances of version 2.x use a different message storage format from that of 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 upgrade your production instance.
- We recommend that you perform the upgrade during off-peak hours.
- We also recommend that you update the client version after the upgrade to keep the versions of the client and broker consistent.
- 1. Log on to the Message Queue for Apache Kafka console.
- 2. In the top navigation bar, select the region where your instance is located.
- 3. In the left-side navigation pane, click Instances.
- 4. On the **Instances** page, click the name of the instance that you want to manage.
- 5. In the Basic Information section of the Instance Details tab on the Instance Details page, click Upgrade Major Version on the right of Major Version:.
- 6. In the Upgrade Major Version dialog box, perform the following operations:
 - i. Enter your name in the Name: field.
 - ii. Enter your emergency phone number in the Emergency phone number: field.
 - iii. Select Yes for Cross-Version Upgrade to 2.0.
 - iv. Click Upgrade.

Minor version upgrade

♥ Notice

- The upgrade is free of charge and compatible with the existing SDK and API.
- To avoid impact on your business during the upgrade, make sure that your client supports automatic reconnection and can handle disconnections. By default, the client supports automatic reconnection.
- The upgrade takes about 15 minutes. During the upgrade, the service will not be interrupted, and your business will not be affected in normal cases.
- We recommend that you perform the upgrade during off-peak hours.
- We also recommend that you update the client version after the upgrade to keep the versions of the client and broker consistent.

1. Log on to the Message Queue for Apache Kafka console.

- 2. In the top navigation bar, select the region where your instance is located.
- 3. In the left-side navigation pane, click Instances.
- 4. On the **Instances** page, click the name of the instance that you want to manage.
- 5. In the Basic Information section of the Instance Details tab on the Instance Details page, click Upgrade Minor Version on the right of Minor Version:.
- 6. In the **Upgrade Minor Version** dialog box, perform the following operations:
 - i. Enter your name in the Name: field.
 - ii. Enter your emergency phone number in the Emergency phone number: field.
 - iii. Click Upgrade.

1.5. Configure the whitelist

You can modify the whitelist to allow specified IP addresses or ports to access your Message Queue for Apache Kafka instance.

Prerequisites

A Message Queue for Apache Kafka instance is purchased and deployed, and it is in the Running state.

Procedure

Perform the following steps to add IP addresses or CIDR blocks to the whitelist of your instance:

- 1. Log on to the Message Queue for Apache Kafka console.
- 2. In the top navigation bar, select the region where your instance is located.
- 3. In the left-side navigation pane, click **Instances**.
- 4. On the **Instances** page, click the name of the instance that you want to manage.
- 5. In the Security Configuration section of the Instance Details page, click Security Change.
- 6. In the Security Change dialog box, click + Add IP to Whitelist, enter IP addresses or CIDR blocks, and then click Add.

🗘 Notice

- The whitelist can contain a maximum of 100 entries.
- You can add a maximum of 10 IP addresses or CIDR blocks in each entry to the whitelist. Separate them with commas (,).
- You can remove or add a single entry from or to the whitelist.
- You can remove the last entry from the whitelist. Proceed with caution because you can no longer access the Message Queue for Apache Kafka instance by using ports of the port range in the last entry after you remove this entry.

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

Instances of the VPC type

The port range is 9092/9092. By default, 0.0.0.0/0 is added to the whitelist. Clients can connect to the instance from a virtual private cloud (VPC).

| Security Change | × |
|---|-----|
| VPC Whitelist Port Range: 09094/9094 9092/9092 | |
| IP | Add |
| IP: | 面 |

• The port range is 9094/9094. By default, the CIDR block that you specify for the vSwitch when you deploy the instance is added to the whitelist. Clients can connect to the instance from the vSwitch of the VPC.

Note The port range 9094/9094 appears only after the access control list (ACL) feature is enabled. For more information about how to enable the ACL feature, see Authorize SASL users.

| Security Change | × |
|--|-----|
| VPC Whitelist Port Range: () 9094/9094 O 9092/9092 | |
| IP | Add |
| IP: | ā |

• Instances of the Internet and VPC type

- Access from a VPC
 - The port range is 9092/9092. By default, the CIDR block that you specify for the vSwitch when you deploy the instance is added to the whitelist. Clients can connect to the instance from the vSwitch of the VPC.

| - C Willen | St. Polt Range. 0 9094/9094 | 4 🥘 9092/9092 Publi | ic IP Whitelist Po | ort Range: O 9093/9093 | |
|------------|---|---|---|---|--|
| | | | | | Add |
| IP: | | | | | Ē |
| a. VF | C: The port range is 9092/9092. ss to the instances mapping the If be accessed. | The default whitelist is the CID P addresses in the range of the | R block of the VSwitc e CIDR block. If you s | ch specified during instance dep set the whitelist to 0.0.0.0/0, all i | oloyment, allowing Instances in the VPC |

• The port range is 9094/9094. By default, the CIDR block that you specify for the vSwitch when you deploy the instance is added to the whitelist. Clients can connect to the instance from the vSwitch of the VPC.

Note The port range 9094/9094 appears only after the ACL feature is enabled.
 For more information about how to enable the ACL feature, see Authorize SASL users.

| C Whitelist | Port Range: 💿 9094/9094 | O 9092/9092 Public IP Whiteli | ist Port Range: O 9093/9093 | |
|-------------------|---|---|---|--------------|
| | | | | Add |
| P: | - | | | Î |
| a. VPC: access | The port range is 9092/9092. The instances mapping the IP | he default whitelist is the CIDR block of the addresses in the range of the CIDR block | he VSwitch specified during instance deployment k. If you set the whitelist to 0.0.0.0/0, all instance | nt, allowing |

 Access from the Internet: The port range is 9093/9093. By default, 0.0.0.0/0 is added to the whitelist. Clients can connect to the instance over the Internet. Data security is guaranteed by using the permission authentication and data encryption mechanisms.

| PC Whitelist | Port Range: O 9094/9094 | 0 9092/9092 | Public IP Whitelist | Port Range: 💿 9093/9093 | 3 |
|--------------------|---|-------------------------|-------------------------|---------------------------------|-------------------------|
| | | | | | Add |
| IP: | | | | | Ē |
| 1. Publ a. VPC | ic Network and VPC Instances: The port range is 9092/9092. T | he default whitelist is | the CIDR block of the V | Switch specified during instanc | ce deployment, allowing |

7. Optional. To delete an IP address or CIDR block, find the IP address or CIDR block and click the Delete icon in the **Security Change** dialog box.

1.6. Rebalance the topic traffic

When you upgrade the traffic specification of a Message Queue for Apache Kafka instance, the corresponding cluster may be scaled out. After the cluster is scaled out, you must rebalance the topic traffic to distribute the traffic evenly across brokers in the scaled-out cluster. Otherwise, the original topic traffic is still distributed across the brokers that are in the cluster before the scale-out. The original topics are subject to the maximum traffic purchased before the scale-out. The new topics are not subject to this maximum 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 traffic specification of an instance and when cluster scale-out is triggered, see Upgrade instance specifications.

Usage notes

When your Message Queue for Apache Kafka instance is in the **Running (Pending Rebalancing)** state, you can use this instance to send and receive messages but cannot create resources such as topics and consumer groups in this instance. You must complete topic traffic rebalancing or choose not to rebalance topic traffic before you can create a resource.

Traffic rebalancing methods

The following table describes the traffic rebalancing methods supported by Message Queue for Apache Kafka .

Message Queue for Apache Kafka

User guide · Inst ances

| Traffic rebalancing method | Principle | Impact | Scenario | Duration |
|---------------------------------|--|---|--|----------|
| Add Partitions to All Topics | After the cluster is scaled out, the system adds partitions to the new brokers for all topics on the original brokers. | New messages in partitions are out of order. The number of partitions changes. If your client cannot automatically detect new partitions, you may need to restart the client or modify the client code. This may occur in scenarios where stream computing is performed or you send messages to or consume messages from specified partitions. | The partition order is not required. The partition to which messages are sent is not specified. The consumption method is Subscribe. | Seconds. |

Message Queue for Apache Kafka

User guide · Inst ances

| Traffic rebalancing method | Principle | Impact | Scenario | Duration |
|--|--|--|---|---|
| Migrate Partitions of All Topics (recommended) | 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. Cloud storage: No temporary internal traffic is generated. | All cluster scale- out scenarios are supported. | Local storage: minutes or hours. This duration depends on the amount of the data that you want to migrate from local storage. If the data volume is large, the migration may take several hours or longer. We recommend that you migrate the data during off-peak hours. Cloud storage: seconds. It takes about 30 seconds to migrate a topic. |
| Do Not Rebalance (not recommended) | You do not need to perform operations. The original topics are still distributed on the brokers of the cluster before the scale-out, and the new topics are evenly distributed across all brokers after the scale- out. | The original topics are subject to the maximum traffic purchased before the scale-out. If the original topic traffic is large, the traffic between brokers may be unbalanced. | The original topic traffic is small, and 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. | Immediately takes effect. |

Procedure

- 1. Log on to the Message Queue for Apache Kafka console.
- 2. In the top navigation bar, select the region where your instance is located.
- 3. In the left-side navigation pane, click **Instances**.
- 4. On the **Instances** page, click the name of the instance that you want to manage.

- 5. In the Status section of the Instance Details page, click Rebalance Now.
- 6. In the **Rebalancing Method** dialog box, select a rebalancing method. The following rebalancing methods are supported:
 - Add Partitions to All Topics
 - Select Add Partitions to All Topics and click OK.
 - Migrate Partitions of All Topics
 - a. Submit a ticket and ask Message Queue for Apache Kafka customer service to upgrade your broker to the latest version.
 - b. Select Migrate Partitions of All Topics and click OK.
 - Do Not Rebalance

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

Result

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

1.7. View task execution records

After restart tasks 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 the task type, start time, end time, and status. Restart tasks are tasks that you change configurations, upgrade the version, enable the access control list (ACL) feature, or upgrade specifications of an instance.

Prerequisites

A Message Queue for Apache Kafka instance is purchased and deployed, and it is in the Running state.

Procedure

- 1. Log on to the Message Queue for Apache Kafka console.
- 2. In the top navigation bar, select the region where your instance is located.
- 3. In the left-side navigation pane, click Instances.
- 4. On the **Instances** page, click the name of the instance that you want to manage.
- 5. On the **Instance Details** page, click the **Task Records** tab. On the **Task Records** tab, the execution records of restart tasks appear in the lower part.

| Parameter | Description | Example |
|-----------|-------------|---------|
|-----------|-------------|---------|

| Parameter | Description | Example |
|------------|--|-----------------------|
| Task Type | The type of the executed task. Valid values: Configuration Change: modifies 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 Specification: upgrades the configurations of the instance, including the instance edition, network type, traffic specification, disk capacity, and supported topics. For more information, see Upgrade instance specifications. Upgrade Version: upgrades the major and minor versions of the instance. For more information, see Upgrade the instance version. Enable ACL: enables the ACL feature provided by Message Queue for Apache Kafka . This feature is used to manage Simple Authentication and Security Layer (SASL) users and resource access. For more information, see Authorize SASL users. | Upgrade Specification |
| 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 status of the task. Valid values: Not Executed Executing Executed Canceled | Executed |

1.8. FAQ

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 by Message 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 instance specifications 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 Instance specifications. 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 Storage engine comparison.

• 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

| Instance ID/Name | VPC | Tags | Status | IPv4 CIDR Block | Available IP Addresses | IPv6 CIDR Block | Default vSwitch | Zone 💟 | Route Table | Route Table Type |
|------------------|------------|------|-------------|-----------------|---------------------------|------------------------|--------------------|--------|-------------|---------------------|
| | defaultvpc | ٠ | ✓ Available | 101100-001 | 4091 | Enable IPv6 CIDR Block | Yes | | | System |

(?) 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 instance specifications.

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

After you purchase and deploy a Message Queue for Apache Kafka instance, you can directly send messages in the Message Queue for Apache Kafka console to fast test whether the broker is normal.

Prerequisites

A Message Queue for Apache Kafka instance is purchased and deployed, and it is in the **Running** state.

Procedure

To test the Message Queue for Apache Kafka broker, perform the following steps:

- 1. Create a topic
- 2. Send messages
- 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 instance on which you want to create the topic, and then click **Create Topic**.

| ← Topics | (aibaba_proi.en.c | | | | | | | | |
|-----------------|--------------------------------|--|--|---|---------------------------|--|--------------------------|---------|-----|
| Instances | Supported Topics:50, Topics in | Use:2, Available Topics:48; Supported Pa | artitions:400, Partitions in Use:12, Ava | ilable Partitions:388. For more information | about the mappings betwee | an the instance and the quantities of topics and | partitions, see Billing. | | |
| Topics | Create Topic Enter a value. | Q | | | | | | | C © |
| Consumer Groups | Topic II | Instance 🕸 | Tags 🕀 | Service Status 4 | Description | Storage Engine | Created At 4 | Actions | |

4. In the **Create Topic** dialog box, set topic properties and then click **Create**.

| Create Topic | | × | | | | | | | |
|----------------|---|------|--|--|--|--|--|--|--|
| * Name | test | | | | | | | | |
| | The name must be 3 to 64 characters in length, and can contain letters, digits, hyphens (-), and underscores (_). It must contain at least one letter or one digit. | | | | | | | | |
| * Description | test | 4/64 | | | | | | | |
| * Partitions | 12 | | | | | | | | |
| | To prevent data skew, we recommend that you set the number of partitions to a multiple of 6. The number of partitions ranges from 1 to 360. If you have special requirements, submit a ticket. | | | | | | | | |
| Storage Engine | Cloud Storage Local Storage | | | | | | | | |
| | Accesses Alibaba Cloud disks at the underlying layer and stores data in three replicas in distributed mode. This storage engine features low latency, high performance, data persistence, and high reliability. | | | | | | | | |
| Message Type | Normal Message | | | | | | | | |
| | By default, messages with the same key are stored in the same partition in the order they are sent. If a broker in the cluster fails, the messages may be out of order. | | | | | | | | |
| Tag | Enter a tag. Separate multiple tags with commas (,), for example, a:b,c:d | | | | | | | | |

Descriptions of the parameters are provided:

- **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.
- **Part it ions**: the number of partitions for the topic, for example, 12.

Send messages

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

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

| ← Instances | 5: | 0 | | | | | | | | |
|---------------------------|---------------------------------------|---|-------------------------------|-----------------------------------|---|---------------------|---------------------------|-----------------------------|--------------|---|
| Instances | Supported Topics:360, Topics in Use:3 | 7, Available Topics:323; Supported Partitions | 2,000, Partitions in Use:540, | Available Partitions:1,460. For m | ore information about the mappings between th | e instance and th | e quantities of topics ar | nd partitions, see Billing. | | |
| Topics Consumer Groups | Create Topic Enter a value. | Q | | | | | | | с | ۲ |
| Message Query | Topic 🕀 | Instance 4* | Tags ≹ | Service Status | Description | Order Type √t | Storage Engine | Created At It | Actions | |
| Monitoring | | and contract of second | • | | create_by_kafka_connector_do_not | Normally Ordered | Cloud Storage | Feb 25, 2021, 11:45:21 | Send Message | |

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

| Send Messag | je | × |
|-----------------|------------------------|---|
| Торіс | connect-error-test0225 | |
| Partitions | 0 | |
| * Message Key | demo | |
| * Message Value | {"key": "test"} | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| Send Ca | ancel | |

Descriptions of the parameters are provided:

- **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 you send 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 and click **Partition Status** in the **Actions** column.

| nces: | 0 | | | | | | |
|-----------------------|--|---|------------------------------------|---|------------------------|-------------------------|----------------------------|
| Supported Topic | s:360, Topics in Use:37, Available Topics:323; Supported | Partitions:2,000, Partitions in Use:540 | Available Partitions:1,460. For mo | re information about the mappings between | n the instance and the | quantities of topics ar | d partitions, see Billing. |
| Create Topic | Enter a value. Q | | | | | | |
| Topic 🕸 | Instance 4 | Tags 🕀 | Service Status - 라 | Description | Order Type ∛t | Storage Engine | Created At 🗄 |
| and the second second | and the second second | - ` | | create_by_kafka_connector_do_not | Normally | Cloud Storage | Feb 25, 2021, 11:45:21 |

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

| Partition Status | | | | C | × |
|--------------------------|---------------------------|-------------------------|-------------------|---|---|
| Status | | Result | | | |
| Total Messages on Server | | 0 | | | |
| Last Updated At | | - | | | |
| Partition ID Jh | Minimum Offset √ t | Maximum Offset ∛ | Last Updated At 🕸 | | |
| 0 | 0 | 0 | - | | |
| 1 | 0 | 0 | | | |
| 2 | 0 | 0 | | | |
| 3 | 0 | 0 | | | |
| 4 | 0 | 0 | - | | |
| 5 | 0 | 0 | - | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| Close | | | | | |

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 instance and then click the Query by Offset tab.
 - i. In the Enter a topic. field, enter a topic.
 - ii. From the **Select a partition.** drop-down list, select the ID of the partition to which the message is sent.
 - iii. In the Enter an offset. field, enter an offset.
 - iv. Click the Search tab.

| ← Instances | S: | ٥ | | | |
|-------------------------------------|-------------------------------|-----------|------------------------------|--|---|
| Instances | Query by Offset Query by Time | | | | |
| Topics | constituting constant of the | 0 | ✓ 0 Search | Key (Size: 28B) | Download Message |
| Consumer Groups Message Query | Partition:0 | Offset: 0 | Time:Dec 8, 2020, 11:33:49 | connector-connector-test-128 | |
| Monitoring Connector(Public Prev | Partition:0 | Offset: 1 | @ Time:Dec 8, 2020, 11:33:50 | | |
| | Partition:0 | Offset: 2 | @ Time:Dec 8, 2020, 11:33:50 | | |
| | Partition:0 | Offset: 3 | Time:Dec 8, 2020, 11:33:50 | | |
| | Partition:0 | Offset: 4 | Time:Dec 8, 2020, 11:33:51 | For each queried message, up to 1 KB of its When a queried message exceeds 1 KB in part that exceeds this limit cannot be display content of a queried message, download th | content can be displayed in the console. size, it is automatically truncated and the ved in the console. To view the complete a message. Learn more. |
| | | | | Value (Size: 900B) | Download Message |
| | | | | Control of the second s | |

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?

Only Message Queue for Apache Kafka instances of the Internet and virtual private cloud (VPC) type can be accessed from the Internet.

Message Queue for Apache Kafka only supports instances of the VPC type. Therefore, you cannot access Message Queue for Apache Kafka instances from the Internet.

Notice You can select the network type only when you purchase 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 virtual private clouds (VPCs).

CEN

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

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

| Scenario | Configuration method |
|---------------------------|--|
| Same-account VPC peering | Connect two VPCs in the same region under the same account |
| Sume account of e peening | Connect two VPCs in different regions under the same account |
| | 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 connect Alibaba Cloud network resources around the world. CEN can also connect 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 the Internet.

• Nearest access and shortest path

CEN deploys multiple access points and forwarding points in more than 60 regions around the world to support access to the nearest nodes of Alibaba Cloud. It enables traffic to transmit over a responsive and latency-free network.

• Link redundancy and disaster recovery

CEN provides at least four redundant links between any two access points. Therefore, it features high availability and network redundancy. If a link fails, CEN ensures your services to run normally without network jitters or interruption.

• Systematic management

CEN has systematic network monitoring capabilities that automatically detect route conflicts caused by system changes. Therefore, it can 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 Establish IPsec-VPN connections between two VPCs.

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

VPN Gateway uses a hot-standby architecture and 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 Overview.

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.Topic2.1. View partition status

You can view the partition status to understand the total number of messages on the Message Queue for Apache Kafka broker or the consumption progress of each partition.

Prerequisites

A topic is created. For more information, see Create topics

Procedure

- 1. Log on to the Message Queue for Apache Kafka console.
- 2. In the top navigation bar, select the region where your instance is located.
- 3. In the left-side navigation pane, click **Instances**.
- 4. On the **Instances** page, click the name of the instance that you want to manage.
- 5. In the left-side navigation pane, click **Topics**.
- 6. On the **Topics** page, find the topic whose partition status you want to view, click the **i** icon in the **Actions** column, and then select **Partition Status**.

| Partition Status | | | c X |
|--------------------------|------------------|------------------|-------------------|
| Status | | Result | |
| Total Messages on Server | | 0 | |
| Last Updated At | | | |
| | | | |
| Partition ID 1 | Minimum Offset 🌓 | Maximum Offset ↓ | Last Updated At 小 |
| 0 | 0 | 0 | - |
| 1 | 0 | 0 | |
| 2 | 0 | 0 | |
| 3 | 0 | 0 | |
| 4 | 0 | 0 | - |
| 5 | 0 | 0 | - |

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 stored. |
| Partition ID | The ID of the partition that corresponds to the topic. |
| Minimum Offset | The minimum consumer offset of the topic in the current partition. |

| Parameter | Description |
|-----------------|--|
| Maximum Offset | The maximum consumer offset of the topic in the current partition. |
| Last Updated At | The time when the last message in the partition is stored. |

2.2. Topics

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

A large number of topics (partitions) will greatly reduce the cluster performance and stability.

In Message Queue for Apache Kafka , messages are stored and scheduled by partition. If messages are stored in a large number of 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. How many partitions are added each time I purchase a topic?

Each time you purchase a topic, 16 partitions are added.

In addition to the default number of partitions, 16 partitions are added each time you purchase a topic. Assume that you have purchased an instance that has 50 topics, a maximum traffic volume of 20 MB/s, and 400 default partitions. After you purchase another 10 topics for this instance, 160 partitions are added to the instance. The total number of partitions increases to 560.

2.2.4. 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 Storage engine comparison.

2.2.5. 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.

2.2.6. 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.

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

Problem description

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

Procedure

- 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.

2.2.8. Is the compact log cleanup policy supported?

Yes, Message Queue for Apache Kafka instances of open-source version 2.2.0 or later support the compact log cleanup policy.

Yes, Message Queue for Apache Kafka instances of open-source version 2.2.0 or later support the compact log cleanup policy. To support this policy, upgrade your instance to 2.2.0 or later. For more information about how to upgrade the open-source version of an instance, see Major version upgrade.

3.Consumer groups

3.1. Reset consumer offsets

You can reset a consumer offset to change the position from which a consumer starts to consume messages. You can reset the consumer offset to skip the accumulated or undesired messages instead of consuming them, or to consume messages stored after a point in time regardless of whether the messages stored before this point in time are consumed.

Prerequisites

All consumers are stopped. Message Queue for Apache Kafka does not allow you to reset offsets of online consumers.

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

Context

Message Queue for Apache Kafka allows you to reset consumer offsets in one of the following ways:

• Clear messages: If a consumer no longer wants to consume accumulated messages on the broker, you can choose to clear messages for the consumer. This way, the consumer offset for the consumer is set to the latest position.

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

• Start consumption at a specified point in time: You can reset the offset of a consumer group to a point in time "t" that is in the past or future. The point in time "t" is a point in time when a message is stored. Then, 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 your instance is located.
- 3. In the left-side navigation pane, click Instances.
- 4. On the **Instances** page, click the name of the instance that you want to manage.
- 5. In the left-side navigation pane, click **Consumer Groups**.
- 6. On the **Consumer Groups** page, find the consumer group whose consumer offset you want to reset, and click **Reset Consumer Offset** in the **Actions** column.
- 7. In the **Reset Consumer Offset** dialog box, select a 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 feature described at the beginning of this topic.

| Reset Consumer Off | set | × |
|--------------------|---|--------|
| Consumer Group | connect-cluster-vianau-kalku24u-test | |
| Topics | All Topics | ~ 7 |
| | Clear all accumulated messages and consume messages from the latest offset. | |
| | Reset Consumer Offset by Time | |
| | | |
| | | |
| | ок | Cancel |

• **Reset Consumer Offset by Time**: corresponds to the feature of starting consumption at a specified point in time. This feature is described at the beginning of this topic. If you select this policy, you must specify a point in time.

| Reset Consumer Offs | et | | \times |
|---------------------|--|--------|----------|
| Consumer Group | comrect-cluster-niarou-kalka2to-teat | | |
| Topics | All Topics | \sim | ? |
| | Clear all accumulated messages and consume messages from the latest offset. | | |
| | Reset Consumer Offset by Time | ė | 0 |
| | The allowed time ranges from Apr 20, 2021, 18:37:58 to Apr 23, 2021, 18:37:58. | | • |
| | ОК | Can | cel |

3.2. View the consumption status

When messages are accumulated or skewing occurs, you can view the subscriptions between consumer groups and topics and determine the status based on 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 the region where your instance is located.
- 3. In the left-side navigation pane, click **Instances**.
- 4. On the **Instances** page, click the name of the instance that you want to manage.
- 5. In the left-side navigation pane, click **Topics**.
- 6. On the **Topics** page, find the topic whose subscriptions you want to view, click the **:** icon in the

Actions column, and then select Subscription Relationship. In the Subscription Relationship dialog box, all the consumer groups that subscribe to the topic
appear.

| Subscription Relationship | | | | |
|----------------------------|-------------------|-----------------------|---------|--|
| Status | | Result | | |
| Торіс | | | | |
| Last Sent At 🕐 | | 19 Mar 2020, 17:53:59 | | |
| Subscribed Consumer Groups | | 1 | | |
| ID | Consumer Group ID | | Actions | |
| 1 | | | Details | |

7. In the **Consumer Group** column, find the consumer group whose consumption status you want to view, and click **Details** in the **Actions** column.

The message consumption details in each partition of the topic appear.

| Su | Subscription Relationship 🕜 | | | | | | | | × | |
|----|-----------------------------|-----------------------|------------|----------------------------|-------------------|------------|----------------------|----------------------|-----------------------|--|
| | Status Result | | | | | | | | | |
| | Topic | | | | | | | | | |
| | Last S | ent At 🔞 | | | | 19 Mar 202 | 0, 17:12:19 | | | |
| | Subso | ribed Consumer Group | 5 | | | 1 | | | | |
| | ID | | | Consumer Group I | D | | | Actions | | |
| | 1 | | | | 0 | | Details | | A | |
| | | Partition ID \ | owner 📀 | Maximum Offset I | Consu 1 | mer Offset | Messages Accumula | i ited J r | Last Consumed At 👩 🖡 | |
| | | 0 | | 9 | 9 | | 0 | | 19 Mar 2020, 17:12:19 | |
| | | 1 | (Distance) | 8 | 8 | | 0 | | 19 Mar 2020, 17:12:19 | |
| | | 2 | | 9 | 9 | | 0 | | 19 Mar 2020, 17:12:19 | |
| | | 3 | | 8 | 8 | | 0 | | 19 Mar 2020, 17:12:19 | |
| Pa | Parameter Description | | | | | | | | | |

topic.

The ID of the partition that corresponds to the

Partition ID

| Parameter | Description | | | |
|----------------------|---|--|--|--|
| | The ID and IP address of the online consumer that has subscribed to the topic are displayed in real time. | | | |
| owner | Note The value is in the format of ID>_/<ip address=""></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. | | | |
| | The total number of accumulated messages in the current partition. The value is equal to the maximum offset minus the consumer offset. | | | |
| Messages Accumulated | 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 consumer running status and improve the consumption speed. For more information, see Reset consumer offsets. | | | |
| 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 Kafka console.
- 2. In the top navigation bar, select the region where your instance is located.
- 3. In the left-side navigation pane, click **Instances**.
- 4. On the **Instances** page, click the name of the instance that you want to manage.
- 5. In the left-side navigation pane, click **Consumer Groups**.
- On the Consumer Groups page, find the consumer group whose subscriptions you want to view, and click Consumption Status in the Actions column.
 In the Consumption Status dialog box, all the topics to which the consumer group has subscribed and the Messages Accumulated and Last Consumed At of each topic appear.

| Consumption Status | | | c X |
|----------------------------|------------------------|----------------------------------|---------|
| Consumer Group | | connect reactory-sufficient feed | |
| Total Messages Accumulated | | 0 | |
| Last Consumed At 😨 | | - | |
| | | | |
| Topic 사 | Messages Accumulated 🕸 | Last Consumed At 😰 🌗 | Actions |
| drundy. | 0 | | Details |

7. In the **Topic** column, find the topic whose consumption status you want to view, and click **Details** in the **Actions** column.

The message consumption details in each partition of the topic appear.

| Consumption Status | 5 | | | | C | × |
|---------------------------|------------------------------|------------------------|-------------------|------------------------------|--------------------|---|
| Consumer Group | | | connecti niaros i | altabesteet | | |
| Total Messages Accumulate | ed | | 0 | | | |
| Last Consumed At 😨 | | | | | | |
| Торіс 🜗 | | Messages Accumulated 1 | Las | t Consumed At 👩 🌗 | Actions | |
| distante | | 0 | | | Details | |
| Partition ID ↓ | owner 👩 | Maximum Offset ႃ | Consumer Offset 1 | Messages Accumu- lated վr | Last Consumed At J | |
| 0 | 4,/102.108-0.25 | 0 | 0 | 0 | | |
| 1 | 4,1492,458,3,23 | 0 | 0 | 0 | | |
| 2 | soneumer- s_risc.net.s.cs | 0 | 0 | 0 | | |
| 3 | 5,792,901.029 | 0 | 0 | 0 | | Ŧ |

| Parameter | Description | | | | |
|----------------|--|--|--|--|--|
| Partition ID | The ID of the partition that corresponds to the topic. | | | | |
| | The ID and IP address of the online consumer that has subscribed to the topic are displayed in real time. | | | | |
| owner | Note The value is in the format of <client id="">_/<ip address=""></ip></client> You cannot view the owner information of offline consumers. | | | | |
| Maximum Offset | The maximum message consumer offset of the topic in the current partition. | | | | |

| Parameter | Description | | |
|----------------------|---|--|--|
| Consumer Offset | The message consumer offset of the topic in the current partition. | | |
| | The total number of accumulated messages in the current partition. The value is equal to the maximum offset minus the consumer offset. | | |
| Messages Accumulated | 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 consumer running status and improve the consumption speed. For more information, see Reset consumer offsets. | | |
| 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. What is the consumer group count?

By default, the consumer group count is twice the topic 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 instance specifications.

3.3.2. Why is 1970 displayed for "Last Consumed At" for the partitions of the topic that

corresponds to a consumer group?

The consumer group has not consumed any messages in the partitions of the topic.

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

3.3.3. Why does a deleted consumer group still receive messages?

Problem description

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

Procedure

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.

3.3.4. 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.

3.3.5. What does "Last Consumed At" mean in the console?

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

3.3.6. 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.

3.3.7. Where can I find the best practices for consumption?

For more information, see Best practices for subscribers.

3.3.8. 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.

3.3.9. 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 Reset consumer offsets 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 Best practices for tag design.

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.

i. \bigcirc 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.
- ii. 🗘 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.

| Create/Bind Tags | | × |
|---|----------|-------------------|
| Select Tag (| Select 3 | Binding Result |
| Bound successfully. Tag (Click the key or value to go to the corresponding tag interface): Keyteste Keyteste Keya'dad Resource ID: Disable Bind Other Resources | | |

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 a region.
- 3. In the left-side navigation pane, you can:
 - Edit tags of a Message Queue for Apache Kafka instance

In the left-side navigation pane, click **Overview**. On the **Instances** page, find the instance whose tags you want to edit, move the pointer over \mathcal{O} , and then click **Edit Tag**.

• Edit tags of a topic

In the left-side navigation pane, click **Topics**. On the **Topics** page, click the instance where the topic is located, find the topic, move the pointer over \mathcal{O} , and then click **Edit Tag**.

• Edit tags of a consumer group

In the left-side navigation pane, click **Consumer Groups**. On the **Consumer Groups** page, click the instance where the consumer group is located, find the consumer group, move the pointer over \emptyset , and then click **Edit Tag**.

- 4. In the Edit Tag dialog box:
 - Delete a tag

In the dotted line section, find the tag that you want to delete and click \mathbf{X} .

- Bind a tag
 - To bind an existing tag, click **Select Tag**, select the tag key that you want to bind from the **Tag Key** list, and then select the tag value that you want to bind from the **Tag Value** list.
 - To bind a new tag, click Create Tag, enter the tag key and tag value, and then click OK.

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

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.

Onte 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.

| K Back Delete Tag | |
|----------------------------------|-----------------|
| @Instance2 Topic 0 Group 1 | Bind Resource |
| Batch Operation \vee C Refresh | |
| Instance Topic Group | |
| Resource Name | Actions |
| data para di ignila | Unbind Resource |
| | Unbind Resource |

5.Monitoring and alerting

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.

- Prerequisit es
 - You have created an instance, a topic, and a consumer group. For more information, see Step 3: Create resources.
 - The consumer group you created has subscribed to the topic. For more information, see Message Queue for Apache Kafka demos.
- Procedure
 - i. 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).
 - ii. 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**.

iii. 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.

Prerequisit es

You have created an instance, a topic, and a consumer group. For more information, see Step 3: Create resources.

- Procedure
 - i. 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)**.
 - ii. 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**.
 - iii. 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.

iv. On the **Create Alarm Rule** page, set the alert rule and notification method. For more information, see Alert service.

| Create Ala | arm Rule 🔁 Back | to | | | | | | | | |
|------------|--------------------------|---|---------------|-------------|-------------------|-------|-----|------|-----------|----------|
| 1 | Related Resource | | | | | | | | | |
| | Product: | kafka | Ŧ |] | | | | | | |
| | Resource Range: | Instances | • | 0 | | | | | | |
| | Region: | (inclusi Inspira) | * |] | | | | | | |
| | Instances: | 188 | * |] | | | | | | |
| 2 | Set Alarm Rules | | | | | | | | | |
| | Alarm Rule: | | | | | | | | | |
| | Rule Description: | InstanceMessageInput | r 1Minut | e cycle 🔻 | Continue for 1 | Value | • | >= • | Threshold | Mbytes/s |
| | +Add Alarm Rul | e | | | | | | | | |
| | Mute for: | 24 h 👻 🖉 | | | | | | | | |
| | Effective Period: | 00:00 • To: 23:59 • | | | | | | | | |
| 3 | Notification Meth | od | | | | | | | | |
| | Notification | Contact Group | All | Selecte | ed Groups 0 count | | All | | | |
| | Contact: | Search Q | L. | | | | | | | |
| | | and states a | - | | | | | | | |
| | | | + | | | | | | | |
| | | | | | | | | | | |
| | | Quickly create a contact group | | | | | | | | |
| | | Phone + Text Message + Email + DingTa | lk (Critical) | 0 | | | | | | |
| | Notification Methods: | Text Message + Email + DingTalk (Warni Email + DingTalk (Info) | ng) | | | | | | | |
| | Auto Scoling (t | he corresponding scaling rule will be triggered | when the | alarm occu | re) | | | | | |
| | C Auto Scaling (0 | në corrësponding scaling rule wili be triggered | when the | | 15) | | | | | |
| | Email Subject: | The default format of email theme is Product | t Name + N | letric Name | e + Instance ID. | | | | | |
| | Email Remark: | Optional | | | | | | | | |
| | | | | | | | | | | |
| | | | | | | | 11 | | | |
| | HTTP CallBack: | for example: http://alart.aliyun.com:8080/ca | llback | | | | • | 0 | | |
| | | | | | | | | | | |
| | Confirm Ca | ancel | | | | | | | | |
| | | | | | | | | | | |
| | | | | | | | | | | |

➡ 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
 - i. 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).
 - ii. 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**.
 - iii. 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. FAQ

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?

Problem description

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

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 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 Step 2: Grant permissions to the RAM user.

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.

6.Connectors

6.1. Overview

Message Queue for Apache Kafka provides fully managed and maintenance-free connectors to synchronize data between Message Queue for Apache Kafka and other Alibaba Cloud services. This topic describes the types, procedure, and limits of connectors. It also describes how to synchronize data between services in different regions by using connectors.

Notice The connector feature of Message Queue for Apache Kafka is in public preview. This feature is independent of Message Queue for Apache Kafka instances. Therefore, you are not charged on the Message Queue for Apache Kafka side when you use a connector to synchronize data between Message Queue for Apache Kafka and another Alibaba Cloud service. Alibaba Cloud does not provide service level agreement (SLA) commitments for the connector feature in public preview. For information about the SLA commitments and billing of the services that are related to the connector feature, see the documentation of the services.

Types of connectors

Message Queue for Apache Kafka provides two categories of connectors:

| Connector | Description | References |
|---------------------------------|---|--|
| Function Compute sink connector | Synchronizes data from Message Queue for Apache Kafka to Function Compute. | Create a Function Compute sink connector |
| MaxCompute sink connector | Synchronizes data from Message Queue for Apache Kafka to MaxCompute. | Create a MaxCompute sink connector |
| OSS sink connector | Synchronizes data from Message Queue for Apache Kafka to Object Storage Service (OSS). | Create an OSS sink connector |
| Elasticsearch sink onnector | Synchronizes data from Message Queue for Apache Kafka to Elasticsearch. | Create an Elasticsearch sink connector |

• Sink connector: Sink connectors are used to synchronize data from Message Queue for Apache Kafka to other Alibaba Cloud services.

• Source connector: Source connectors are used to synchronize data from other Alibaba Cloud services to Message Queue for Apache Kafka .

| Connector | Description | References |
|-----------|-------------|------------|
|-----------|-------------|------------|

| Connector | Description | References |
|------------------------|--|--------------------------|
| MySQL source connector | Synchronizes data from ApsaraDB RDS for MySQL to Message Queue for Apache Kafka . | 创建MySQL Source Connector |

Procedure

Perform the following steps to use connectors:

- 1. Enable the connector feature
- 2. Create a connector.
 - Create a Function Compute sink connector
 - Create a MaxCompute sink connector
 - Create an OSS sink connector
 - Create an Elasticsearch sink connector
 - 。 创建MySQL Source Connector
- 3. Perform the following operations as required:
 - View task configurations of a connector
 - View connector logs
 - Suspend a connector
 - Resume a connector
 - Delete a connector
 - Modify connector configurations
 - Test a connector
 - Enable Internet access for a connector

Limits

The following table lists the limits of Message Queue for Apache Kafka on connectors.

| ltem | Limit |
|------------------------------|--|
| Maximum number of connectors | You can create up to three connectors for each instance. |

| ltem | Limit |
|---------|--|
| Regions | China (Hangzhou) China (Shanghai) China (Beijing) China (Zhangjiakou) China (Hohhot) China (Hohhot) China (Shenzhen) China (Chengdu) China (Hong Kong) Singapore (Singapore) Japan (Tokyo) |

Note To increase the number connectors for your instance or use connectors in other regions, submit aticket to Message Queue for Apache Kafka Customer Services.

Cross-region data synchronization

You must enable Internet access for a connector to synchronize data from an Alibaba Cloud service in one region to an Alibaba Cloud service in another region over the Internet. For more information, see Enable Internet access for a connector.

② Note If you need to use MySQL source connectors to synchronize data across regions, you must first activate Cloud Enterprise Network (CEN). For more information, see 创建MySQL Source Connector.

6.2. Enable the connector feature

This topic describes how to enable the connector feature for your Message Queue for Apache Kafka instance.

Prerequisites

A Message Queue for Apache Kafka instance is purchased and deployed, and the following conditions are met by the instance.

| ltem | Description |
|---------|--|
| Status | The status of the instance must be Running. |
| Version | The version of the Message Queue for Apache Kafka instance must be one of the following: The major version is 0.10.2, with the latest minor version. The major version is 2.2.0. |

Note In the Message Queue for Apache Kafka console, you can view the running status and version of the instance in the **Basic Information** section on the **Instance Details** page.

Procedure

- 1. Log on to the Message Queue for Apache Kafka console.
- 2. In the top navigation bar, select the region where your instance is located.
- 3. In the left-side navigation pane, click **Connector**.
- 4. On the **Connector** page, click an instance and click **Enable Connector**.
- 5. In the **Note** dialog box, click **OK**.

What's next

After the connector feature is enabled for your Message Queue for Apache Kafka instance, you can create a Function Compute or MaxCompute sink connector to synchronize data from your Message Queue for Apache Kafka instance to Function Compute or MaxCompute.

- Create a Function Compute sink connector
- Create a MaxCompute sink connector

6.3. Create a connector

6.3.1. Create a Function Compute sink connector

This topic describes how to create a Function Compute sink connector to synchronize data from a topic in your Message Queue for Apache Kafka instance to a function in Function Compute.

Prerequisites

Before you create a Function Compute sink connector, make sure that the following operations are completed:

- The connector feature is enabled for the Message Queue for Apache Kafka instance. For more information, see Enable the connector feature.
- A data source topic is created in the Message Queue for Apache Kafka instance . For more information, see Step 1: Create a topic.

A topic named fc-test-input is used in this example.

• A function is created in Function Compute. For more information, see Create a function in the Function Compute console.

Notice The function you create must be an event function.

An event function named hello_world is used in this example. This is an event function under the guide-hello_world service that runs in the Python runtime environment. The following sample code provides an example of the function:

```
# -*- coding: utf-8 -*-
import logging
# To enable the initializer feature
# please implement the initializer function as below:
# def initializer(context):
# logger = logging.getLogger()
# logger.info('initializing')
def handler(event, context):
logger = logging.getLogger()
logger.info('hello world:' + bytes.decode(event))
return 'hello world:' + bytes.decode(event)
```

Procedure

To synchronize data from a topic in your Message Queue for Apache Kafka instance to a function in Function Compute by using a Function Compute sink connector, perform the following steps:

1. (Optional)Enable Function Compute sink connectors to access Function Compute across regions.

Notice If you do not want Function Compute sink connectors to access Function Compute across regions, skip this step.

(Optional)Enable Internet access for Function Compute sink connectors

2. (Optional)Enable Function Compute sink connectors to access Function Compute across Alibaba Cloud accounts.

Notice If you do not want Function Compute sink connectors to access Function Compute across Alibaba Cloud accounts, skip this step.

- (Optional)Create a custom policy
- (Optional)Create a RAM role
- (Optional)Grant permissions
- 3. (Optional)Create the topics and consumer groups that are required by a Function Compute sink connector.

✓ Notice

- If you do not want to customize the names of the topics and consumer groups, skip this step.
- Some topics that are required by a Function Compute sink connector must use a local storage engine. If the major version of your Message Queue for Apache Kafka instance is 0.10.2, you cannot manually create topics that use a local storage engine. In major version 0.10.2, these topics must be automatically created.
- i. (Optional)Create topics for a Function Compute sink connector
- ii. (Optional)Create consumer groups for a Function Compute sink connector
- 4. Create and deploy a Function Compute sink connector
- 5. Verify the results.

- i. Send a test message
- ii. View function logs

(Optional)

Enable Internet access for Function Compute sink connectors

If you want Function Compute sink connectors to access other Alibaba Cloud services across regions, enable Internet access for Function Compute sink connectors. For more information, see Enable Internet access for a connector.

(Optional)

Create a custom policy

You can create a custom policy to access Function Compute by using the Alibaba Cloud account within which you want to use Function Compute.

- 1. Log on to the Resource Access Management (RAM) console.
- 2. In the left-side navigation pane, choose **Permissions > Policies**.
- 3. On the **Policies** page, click **Create Policy**.
- 4. On the Create Custom Policy page, create a custom policy.
 - i. In the Policy Name field, enter KafkaConnectorFcAccess.
 - ii. Set Configuration Mode to Script.
 - iii. In the **Policy Document** field, enter the custom policy script. The following sample code provides an example of the custom policy script for access to Function Compute:

```
{
    "Version": "1",
    "Statement": [
        {
            "Action": [
            "fc:InvokeFunction",
            "fc:GetFunction"
        ],
            "Resource": "*",
            "Effect": "Allow"
        }
    ]
}
```

iv. Click OK.

(Optional)

Create a RAM role

You can create a RAM role by using the Alibaba Cloud account within which you want to use Function Compute. When you create the RAM role, select a supported Alibaba Cloud service as the trusted service. You cannot select Message Queue for Apache Kafka as the trusted service of a RAM role. After you create the RAM role, you can modify the trust policy of the created RAM role.

- 1. In the left-side navigation pane, click **RAM Roles**.
- 2. On the RAM Roles page, click Create RAM Role.

- 3. In the Create RAM Role panel, create a RAM role.
 - i. Set Trusted entity type to Alibaba Cloud Service and click Next.
 - ii. Set Role Type to Normal Service Role. In the RAM Role Name field, enter AliyunKafkaConn ectorRole. From the Select Trusted Service drop-down list, select Function Compute. Then, click OK.
- 4. On the **RAM Roles** page, find and click **AliyunKafkaConnectorRole**.
- 5. On the AliyunKafkaConnectorRole page, click the Trust Policy Management tab, and then click Edit Trust Policy.
- 6. In the Edit Trust Policy panel, replace fc in the script with *alikaf ka* and click OK.

| RAM / RAM Roles / AliyunKafkaConnectorRole | | | |
|--|---------|--|---------------------|
| ← AliyunKafkaConnectorRole | | | |
| Basic Information | | | |
| Role Name AliyunKafkaConnectorRole | Created | Jun 17, 2020, 10:14:44 | |
| Note | ARN | acsiram: aliyunkafkaconnectorrole 🚨 Copy | |
| Maximum Session Duration 3600 Seconds Edit | | | |
| | | | |
| Permissions Trust Policy Management | | | |
| | | | |
| Edit Trust Policy | | | |
| 1 (| | | View in Full Screen |
| 2 "Statement": [| | | |
| 4 "Action": "sts:AssumeRole", | | | |
| 5 "Effect": "Allow", | | | |
| 6 "Principal": { | | | |
| 7 "Service": [| | | |
| 8 "alikafka.aliyuncs.com" | | | |
| 19 | | | |
| | | | |
| 12], | | | |
| 13 "Version": "1" | | | |
| 14 | | | - I |
| | | | |

(Optional)

Grant permissions

You can grant the created RAM role the permissions to access Function Compute by using the Alibaba Cloud account within which you want to use Function Compute.

- 1. In the left-side navigation pane, click **RAM Roles**.
- 2. On the RAM Roles page, find AliyunKafkaConnectorRole and click Add Permissions in the Actions column.
- 3. In the Add Permissions panel, attach the KafkaConnectorFcAccess policy.
 - i. In the Select Policy section, click Custom Policy.
 - ii. In the Authorization Policy Name column, find and click KafkaConnectorFcAccess.
 - iii. Click OK.
 - iv. Click Complete.

(Optional)

Create topics for a Function Compute sink connector

In the Message Queue for Apache Kafka console, you can manually create the five topics that are required by a Function Compute sink connector.

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

- 4. On the **Instances** page, click the name of the instance that you want to manage.
- 5. In the left-side navigation pane, click **Topics**.
- 6. On the **Topics** page, click **Create Topic**.
- 7. In the **Create Topic** panel, set the properties of a topic and click **Create**.

| Торіс | Description |
|-----------------------------|---|
| Offset storage topic | The topic that is used to store consumer offsets. Topic: the name of the topic. We recommend that you start the name with connect-offset. Partitions: the number of partitions in the topic. Set it to a value greater than 1. Storage Engine: the storage engine of the topic. Set the parameter to Local Storage. cleanup.policy: the log cleanup policy for the topic. Set the parameter to compact. |
| Task configuration topic | The topic that is used to store task configurations. Topic: the name of the topic. We recommend that you start the name with connect-config. Partitions: the number of partitions in the topic. Set the parameter to 1. Storage Engine: the storage engine of the topic. Set the parameter to Local Storage. cleanup.policy: the log cleanup policy for the topic. Set the parameter to compact. |
| Task status topic | The topic that is used to store task status. Topic: the name of the topic. We recommend that you start the name with connect-status. Partitions: the number of partitions in the topic. We recommend that you set the parameter to 6. Storage Engine: the storage engine of the topic. Set the parameter to Local Storage. cleanup.policy: the log cleanup policy for the topic. Set the parameter to compact. |
| Dead-letter queue topic | The topic that is used to store the abnormal data of the connector framework. To save topic resources, you can create a topic as both the dead-letter queue topic and the abnormal data topic. Topic: the name of the topic. We recommend that you start the name with connect-error. Partitions: the number of partitions in the topic. We recommend that you set the parameter to 6. Storage Engine: the storage engine of the topic. Valid values: Local Storage and Cloud Storage. |

| Торіс | Description | | |
|---------------------|--|--|--|
| Abnormal data topic | The topic that is used to store the abnormal data of the connector. To save topic resources, you can create a topic as both the dead-letter queue topic and the abnormal data topic. | | |
| | • Topic: the name of the topic. We recommend that you start the name with connect-error. | | |
| | • Partitions: the number of partitions in the topic. We recommend that you set the parameter to 6. | | |
| | Storage Engine: the storage engine of the topic. Valid values: Local Storage and Cloud Storage. | | |
| | | | |

(Optional)

Create consumer groups for a Function Compute sink connector

In the Message Queue for Apache Kafka console, you can create the two consumer groups that are required by a Function Compute sink connector.

- 1. Log on to the Message Queue for Apache Kafka console.
- 2. In the top navigation bar, select the region where your instance is located.
- 3. In the left-side navigation pane, click Instances.
- 4. On the **Instances** page, click the name of the instance that you want to manage.
- 5. In the left-side navigation pane, click **Consumer Groups**.
- 6. On the **Consumer Groups** page, click **Create Consumer Group**.
- 7. In the **Create Consumer Group** panel, set the properties of a consumer group and click **Create**.

| Consumer Group | Description |
|-------------------------------|--|
| Connector task consumer group | The consumer group that is used by the data synchronization task of the connector. The name of this consumer group must be in the connect- <i>T</i> ask name format. |
| Connector consumer group | The consumer group that is used by the connector. We recommend that you start the name of this consumer group with connect-cluster. |

Create and deploy a Function Compute sink connector

You can create and deploy a Function Compute sink connector that synchronizes data from Message Queue for Apache Kafka to Function Compute.

- 1. Log on to the Message Queue for Apache Kafka console.
- 2. In the top navigation bar, select the region where your instance is located.
- 3. In the left-side navigation pane, click **Instances**.
- 4. On the **Instances** page, click the name of the instance that you want to manage.

- 5. In the left-side navigation pane, click **Connector (Public Preview)**.
- 6. On the Connector (Public Preview) page, click Create Connector.
- 7. In the Create Connector panel, perform the following steps:
 - i. In the Enter Basic Information step, set the parameters that are described in the following table, and click Next.

○ Notice By default, the Authorize to Create Service Linked Role check box is selected. This means that Message Queue for Apache Kafka will create a service-lined role based on the following rules:

- If no service-linked role is created, Message Queue for Apache Kafka automatically creates a service-linked role for you to use the Function Compute sink connector.
- If you have created a service-linked role, Message Queue for Apache Kafka does not create it again.

For more information about service-linked roles, see Service-linked roles.

| Parameter | Description | Example |
|-------------------|--|--|
| Connector Name | The name of the connector. Take note of the following rules when you specify a connector name: The connector name must be 1 to 48 characters in length. It can contain digits, lowercase letters, and hyphens (-), but cannot start with a hyphen (-). The connector name must be unique within a Message Queue for Apache Kafka instance. The data synchronization task of the connector must use a consumer group that is named in the connect-<i>Task nam e</i> format. If you have not manually created such a consumer group, the system automatically creates a consumer group for you. | kafka-fc-sink |
| Dump Path | The source and destination of data transfer. Select a data source from the first drop-down list and a destination from the second drop-down list. | Message Queue for Apache Kafka Dump To Function Compute |

ii. In the **Configure Source Instance** step, set the parameters that are described in the following table, and click **Next**.

(?) Note If you have created the required topics and consumer groups, set Create Resource to Manually and enter the information about the created resources. Otherwise, set Create Resource to Automatically.

Parameter

Description

Example

| Parameter | Description | Example |
|-----------------------------------|---|-----------------------------------|
| VPC ID | The ID of the virtual private cloud (VPC) where the data synchronization task runs. The default value is the ID of the VPC where the Message Queue for Apache Kafka instance is deployed. You do not need to enter a value. | vpc- bp1xpdnd3l*** |
| VSwitch | The ID of the vSwitch based on which the data synchronization task runs. The vSwitch must be deployed in the same VPC as the Message Queue for Apache Kafka instance. The default value is the ID of the vSwitch that you specify for the Message Queue for Apache Kafka instance. | vsw- bp1d2jgg81*** |
| Data Source Topic | The name of the topic from which data is to be synchronized. | fc-test-input |
| Consumer Offset | The offset where consumption starts. Valid values: latest: Consumption starts from the latest offset. earliest: Consumption starts from the initial offset. | latest |
| Consumer Thread Concurrency | The number of concurrent consumer threads to synchronize data from the data source topic. Default value: 3. Valid values: 3 6 9 12 | 3 |
| Connector consumer group | The consumer group that is used by the connector. We recommend that you start the name of this consumer group with connect-cluster. | connect-cluster- kafka-fc-sink |
| Offset Storage Topic | The topic that is used to store consumer offsets. Topic: the name of the topic. We recommend that you start the name with connect-offset. Partitions: the number of partitions in the topic. Set it to a value greater than 1. Storage Engine: the storage engine of the topic. Set the parameter to Local Storage. cleanup.policy: the log cleanup policy for the topic. Set the parameter to compact. | connect-offset- kafka-fc-sink |

| Parameter | Description | Example |
|--------------------------------|---|----------------------------------|
| Task configuration Topic | The topic that is used to store task configurations. Topic: the name of the topic. We recommend that you start the name with connect-config. Partitions: the number of partitions in the topic. Set the parameter to 1. Storage Engine: the storage engine of the topic. Set the parameter to Local Storage. cleanup.policy: the log cleanup policy for the topic. Set the parameter to compact. | connect-config- kafka-fc-sink |
| Task status Topic | The topic that is used to store task status. Topic: the name of the topic. We recommend that you start the name with connect-status. Partitions: the number of partitions in the topic. We recommend that you set the parameter to 6. Storage Engine: the storage engine of the topic. Set the parameter to Local Storage. cleanup.policy: the log cleanup policy for the topic. Set the parameter to compact. | connect-status- kafka-fc-sink |
| Dead letter queue Topic | The topic that is used to store the abnormal data of the connector framework. To save topic resources, you can create a topic as both the dead-letter queue topic and the abnormal data topic. Topic: the name of the topic. We recommend that you start the name with connect-error. Partitions: the number of partitions in the topic. We recommend that you set the parameter to 6. Storage Engine: the storage engine of the topic. Valid values: Local Storage and Cloud Storage. | connect-error- kafka-fc-sink |
| Abnormal Data Topic | The topic that is used to store the abnormal data of the connector. To save topic resources, you can create a topic as both the dead-letter queue topic and the abnormal data topic. Topic: the name of the topic. We recommend that you start the name with connect-error. Partitions: the number of partitions in the topic. We recommend that you set the parameter to 6. Storage Engine: the storage engine of the topic. Valid values: Local Storage and Cloud Storage. | connect-error- kafka-fc-sink |

iii. In the **Configure Destination Instance** step, set the parameters that are described in the following table, and click **Next**.

| Parameter | Description | Example |
|------------------------------------|--|---|
| Cross- account/Cross -region | Specifies whether the Function Compute sink connector synchronizes data to Function Compute across Alibaba Cloud accounts or regions. By default, this parameter is set to No. Valid values: No: The Function Compute sink connector synchronizes data to Function Compute by using the same Alibaba Cloud account and in the same region. Yes: The Function Compute sink connector synchronizes data to Function Compute across regions by using the same Alibaba Cloud account, in the same region by using different Alibaba Cloud accounts, or across regions by using different Alibaba Cloud accounts. | No |
| Region | The region where Function Compute is activated. By default, the region where the Function Compute sink connector resides is selected. To synchronize data across regions, enable Internet access for the connector and select the destination region. For more information, see Enable Internet access for Function Compute sink connectors. | cn-hangzhou |
| Service Endpoint | The endpoint of Function Compute. In the Function Compute console, you can view the endpoint of Function Compute in the Common Info section on the Overview page. Internal endpoint: We recommend that you use the internal endpoint because it has lower latency. The internal endpoint can be used when the Message Queue for Apache Kafka instance and Function Compute are in the same region. Public endpoint: We recommend that you do not use the public endpoint because it has high latency. The public endpoint because it has high latency. The public endpoint can be used when the Message Queue for Apache Kafka instance and Function Compute are in different regions. To use the public endpoint, you must enable Internet access for the connector. For more information, see Enable Internet access for Function Compute sink connectors. Notice When Cross-account/Cross-region is set to Yes, Service Endpoint is displayed. | http://188***.cn - hangzhou.fc.aliy uncs.com |

| Parameter | Description | Example |
|--------------------------------|---|------------------------------|
| Alibaba Cloud Account | The ID of the Alibaba Cloud account that you can use to log on to Function Compute. In the Function Compute console, you can view the ID of the Alibaba Cloud account in the Common Info section on the Overview page. | 188*** |
| RAM Role | The name of the RAM role that Message Queue for Apache Kafka is authorized to assume to access Function Compute. If you do not need to use a different Alibaba Cloud account, you must create a RAM role and grant the RAM role the required permissions. Then, enter the name of the RAM role. For more information, see Create a custom policy, Create a RAM role, and Grant permissions. If you want to use a different Alibaba Cloud account, you must create a RAM role by using the Alibaba Cloud account, you must create a RAM role by using the Alibaba Cloud account within which you want to use Function Compute. Then, grant the RAM role the required permissions and enter the name of the RAM role. For more information, see Create a custom policy, Create a RAM role, and Grant permissions. Notice When Cross-account/Cross-region is set to Yes, RAM Role is displayed. | AliyunKafkaConn ectorRole |
| Service Name | The name of the service in Function Compute. | guide- hello_world |
| Function Name | The name of the function under the service in Function Compute. | hello_world |
| Service Version or Alias | The version or alias of the service in Function Compute. | LATEST |

| Parameter | Description | Example |
|----------------------|---|--------------|
| Transmission Mode | The mode in which messages are sent. Valid values: Async: This value is recommended. Sync: This value is not recommended. In synchronous mode, if Function Compute takes a long time to process messages, Message Queue for Apache Kafka also takes a long time to process messages. If Function Compute takes more than 5 minutes to process a batch of messages, the Message Queue for Apache Kafka client rebalances the traffic. | Asynchronous |
| | The number of messages that can be sent at a time. Default value: 20. The connector aggregates the messages to be sent at a time based on the maximum number of messages and the maximum allowed message size in a request. The maximum allowed message size cannot exceed 6 MB in synchronous mode or 128 KB in asynchronous mode. Assume that messages are sent in asynchronous mode and up to 20 messages can be sent at a time. If you want to send 18 messages, 17 messages have a total size of 127 KB, and one message is 200 KB in size. In this case, the connector sends the 17 messages as a single batch, and then separately sends the message whose size is more than 128 KB. | |
| | Note If you set key to null when you send a message, the request does not contain key. If you set value to null, the request does not contain value. | |
| | If the size of messages in a batch do not exceed the maximum allowed message size in a request, the request contains the content of the messages. The following code provides a sample request: | |
| | | |
| | | |

| Parameter | Desdription | Example |
|------------|--|---------|
| Batch Size | <pre>{ "key":"this is the message's key2", "offset":8, "overflowFlag":false, "partition":4, "timestamp":1603785325438, "topic":"Test", "valueSize":28 }, { { "key":"this is the message's key9", "overflowFlag":false, "partition":4, "timestamp":1603785325440, "topic":"Test", "valueSize":28 }, { "key":"this is the message's value9", "valueSize":28 }, { "key":"this is the message's value12", "value8::e":29 }, { "key":"this is the message's value38", "value8::e":29 } } </pre> | 50 |
| | | |

| Parameter | If a single message exceeds the maximum allowed Description message size in a request, the request does not | Example |
|-----------|---|---------|
| | contain the content of the message. The following code provides a sample request: | |
| | <pre>[</pre> | |
| | Note To obtain the content of the message, you must pull the message by using its offset. | |
| | | |
| Retries | The number of retries allowed after a message fails to be sent. Default value: 2. Valid values: 1 to 3. In some cases where a message fails to be sent, retries are not supported. The following information describes the types of Error codes and whether they support retries: | |
| | 4XX: does not support a retry except for 429.5XX: supports a retry. | 2 |
| | Note The connector calls the InvokeFunction operation to send a message to Function Compute. | |

| Parameter | Description | Example |
|-------------------------------|---|---------|
| Failure Handling Policy | Specifies whether to continue subscribing to the partition where an error occurs after the message fails to be sent. Valid values: log: retains the subscription to the partition where an error occurs and prints the logs. | log |
| | fail: stops the subscription to the partition where an error occurs and prints the logs. | |
| | Note For more information about how to view the connector logs, see View connector logs. For information about how to troubleshoot errors based on error codes, see Error codes. To resume the subscription to the partition where an error occurs, submit a ticket to Message Queue for Apache Kafka Customer Services. | |

- iv. In the Preview/Submit step, confirm the configurations of the connector and click Submit.
- 8. In the **Create Connector** panel, click **Deploy**. To configure Function Compute resources, click **Configure Function** to go to the Function Compute console and complete the operation.

Send a test message

After you deploy the Function Compute sink connector, you can send a message to the data source topic in Message Queue for Apache Kafka to test whether the message can be synchronized to Function Compute.

- 1. On the **Connector (Public Preview)** page, find the connector that you created, and click **Test** in the **Actions** column.
- 2. On the **Topics** page, select your instance, find the **fc-test-input** topic, click the More icon in the **Actions** column, and then select **Send Message**.
- 3. In the Send Message panel, set the parameters used to send a test message.
 - i. In the Partitions field, enter 0.
 - ii. In the Message Key field, enter 1.
 - iii. In the Message Value field, enter 1.
 - iv. Click Send.

View function logs

After you send a message to the data source topic in Message Queue for Apache Kafka , you can view the function logs to check whether the message is received. For more information, see Configure Log Service resources and view function execution logs.

The test message that you sent appears in the logs.
| ← hello | _worl | d | | | | | | Service Version:LA | ATEST 🗸 🔟 |
|------------------------|---|-----------------|------|--|--|--|------------------------------|--------------------|-------------|
| Overview | Code | Triggers | Log | Function Metrics | Asynchronous Invocation Configuration | ARN - | | | - 0 |
| Basic Query | Basic Query Advanced Query | | | | | | | | |
| @ functio | © 1 HourfRelative) ▼ Auto Refresh Index Attributes | | | | | | | | |
| ∨ i he] | lo_world | AND LATES | т | | | | | 🖗 🕜 🛛 Search | n & Analyze |
| 8 0 14:48:57 | 8 0 14:4857 14:55:30 15:02:10 15:09:30 15:16:10 15:23:30 15:30:30 15:37:30 15:44:30 | | | | | | | | |
| | | | | | Log Er | ntries:10 Search Status:The results are accurate. | | | |
| Raw Logs | | Graph | | | | | Display Content Column | Column Settings | Ţ. |
| Quick Analy | sis | < | Tir | ne ≜▼ Con | tent | | | | |
| You have n to query | ot specified yet. Add it r (Help) | a field Iow. | Q No | v 17, 15:29:36 | | | | | |
| | | 2 | Q No | v 17, 15:29:36so to fun mes qua sen vers | ource_: gpic_: guide-hello_world cionName: hello_world sasge: 2020-11-17107/29:36.429Z 5b662c89- ti/value?11/valueSize*3]] liffer: LATEST wiceName: guide-hello_world sionId: | 45f0-44a9-a152-17a027ab687c [INFO] hello world:[['key'^11','offset'-1,'overflowFlag'false,"partiti | ion":0,"timestamp":160559817 | 74308,"topic":"fc- | -test- |

6.3.2. Create a MaxCompute sink connector

This topic describes how to create a MaxCompute sink connector to synchronize data from a data source topic of a Message Queue for Apache Kafka instance to a MaxCompute table.

Prerequisites

Before you create a MaxCompute sink connector, make sure that the following operations are completed:

- The connector feature is enabled for the Message Queue for Apache Kafka instance. For more information, see Enable the connector feature.
- A data source topic is created in the Message Queue for Apache Kafka instance. For more information, see Step 1: Create a topic.

A topic named maxcompute-test-input is used in this example.

• A MaxCompute table is created by using the MaxCompute client. For more information, see 创建表.

In this example, a MaxCompute table named test_kafka is created in a project named connector_test. You can execute the following statement to create a MaxCompute table named test_kafka:

CREATE TABLE IF NOT EXISTS test_kafka(topic STRING,partition BIGINT,offset BIGINT,key STRING,value ST RING) PARTITIONED by (pt STRING);

Procedure

To synchronize data from a data source topic in a Message Queue for Apache Kafka instance to a MaxCompute table by using a MaxCompute sink connector, perform the following steps:

- 1. Grant Message Queue for Apache Kafka the permissions to access MaxCompute.
 - Create a RAM role
 - Grant permissions
- 2. (Optional)Create the topics and consumer groups that are required by the MaxCompute sink connector.

If you do not want to customize the names of the topics and consumer groups, skip this step and select Automatically in the next step.

Notice Some topics that are required by a MaxCompute sink connector must use a local storage engine. If the major version of your Message Queue for Apache Kafka instance is 0.10.2, you cannot manually create topics that use a local storage engine. In major version 0.10.2, these topics must be automatically created.

- i. (Optional)Create topics for a MaxCompute sink connector
- ii. (Optional)Create consumer groups for a MaxCompute sink connector
- 3. Create and deploy a MaxCompute sink connector
- 4. Verify the results.
 - i. Send a test message
 - ii. View data in the MaxCompute table

Create a RAM role

When you create the Resource Access Management (RAM) role, select a supported Alibaba Cloud service as the trusted service. You cannot select Message Queue for Apache Kafka as the trusted service of a RAM role. After you create the RAM role, you can modify the trust policy of the created RAM role.

- 1. Log on to the RAM console.
- 2. In the left-side navigation pane, click RAM Roles.
- 3. On the RAM Roles page, click Create RAM Role.
- 4. In the Create RAM Role panel, perform the following operations:
 - i. Set Trusted entity type to Alibaba Cloud Service and click Next.
 - ii. Set **Role Type** to **Normal Service Role**. In the **RAM Role Name** field, enter *AliyunKafkaMaxC omputeUser1*. From the **Select Trusted Service** drop-down list, select **MaxCompute**. Then, click **OK**.
- 5. On the RAM Roles page, find and click AliyunKafkaMaxComputeUser1.
- 6. On the AliyunKafkaMaxComputeUser1 page, click the Trust Policy Management tab, and then click Edit Trust Policy.
- 7. In the Edit Trust Policy panel, replace fc in the script with *alikaf ka* and click OK.

| Edit Tru | ust Policy | × |
|---|--|---|
| RAM Role AliyunKafk | Name caMaxComputeUser1 | |
| Edit in | Full Screen | |
| 1 2 3 4 5 6 7 8 9 10 11 12 13 | <pre>{ "Statement": [{ "Action": "sts:AssumeRole", "Effect": "Allow", "Principal": { "Service": ["alikafka.aliyuncs.com"</pre> | |
| ОК | Cancel | |

Grant permissions

To enable the MaxCompute sink connector to synchronize messages to a MaxCompute table, you must grant at least the following permissions to the RAM role.

| Object | Action | Description |
|---------|----------------|---|
| Project | CreateInstance | The permissions to create instances in projects. |
| Table | Describe | The permissions to read the metadata of tables. |
| Table | Alter | The permissions to modify the metadata of tables or create and delete partitions. |

| Object | Action | Description |
|--------|--------|---|
| Table | Update | The permissions to overwrite data in tables or insert data to tables. |

For more information about the preceding permissions and how to grant these permissions, see Authorize users.

To grant the required permissions to AliyunKafkaMaxComputeUser1, perform the following steps:

- 1. Log on to the MaxCompute client.
- 2. Run the following command to add AliyunKafkaMaxComputeUser1 as a RAM user:

add user `RAM\$<accountid>:role/aliyunkafkamaxcomputeuser1`;

(?) Note Replace <accountid> with the ID of your Alibaba Cloud account.

- 3. Grant the RAM user the minimum permissions that are required to access MaxCompute.
 - i. Run the following command to grant the RAM user the permissions on projects:

grant CreateInstance on project connector_test to user `RAM\$<accountid>:role/aliyunkafkamaxco mputeuser1`;

? Note Replace <accountid> with the ID of your Alibaba Cloud account.

ii. Run the following command to grant the RAM user the permissions on tables:

grant Describe, Alter, Update on table test_kafka to user `RAM\$<accountid>:role/aliyunkafkamaxc omputeuser1`;

(?) Note Replace <accountid> with the ID of your Alibaba Cloud account.

Create topics for a MaxCompute sink connector

In the Message Queue for Apache Kafka console, you can manually create the topics that are required by a MaxCompute sink connector.

- 1. Log on to the Message Queue for Apache Kafka console.
- 2. In the top navigation bar, select the region where your instance is located.
- 3. In the left-side navigation pane, click Instances.
- 4. On the **Instances** page, click the name of the instance that you want to manage.
- 5. In the left-side navigation pane, click **Topics**.
- 6. On the **Topics** page, click **Create Topic**.
- 7. In the Create Topic panel, set the properties of a topic and click Create.

Topic

Description

| Торіс | Description |
|-----------------------------|---|
| Offset storage topic | The topic that is used to store consumer offsets. Topic: the name of the topic. We recommend that you start the name with connect-offset. Partitions: the number of partitions in the topic. Set it to a value greater than 1. Storage Engine: the storage engine of the topic. Set the parameter to Local Storage. cleanup.policy: the log cleanup policy for the topic. Set the parameter to compact. |
| Task configuration topic | The topic that is used to store task configurations. Topic: the name of the topic. We recommend that you start the name with connect-config. Partitions: the number of partitions in the topic. Set the parameter to 1. Storage Engine: the storage engine of the topic. Set the parameter to Local Storage. cleanup.policy: the log cleanup policy for the topic. Set the parameter to compact. |
| Task status topic | The topic that is used to store task status. Topic: the name of the topic. We recommend that you start the name with connect-status. Partitions: the number of partitions in the topic. We recommend that you set the parameter to 6. Storage Engine: the storage engine of the topic. Set the parameter to Local Storage. cleanup.policy: the log cleanup policy for the topic. Set the parameter to compact. |
| Dead-letter queue topic | The topic that is used to store the abnormal data of the connector framework. To save topic resources, you can create a topic as both the dead-letter queue topic and the abnormal data topic. Topic: the name of the topic. We recommend that you start the name with connect-error. Partitions: the number of partitions in the topic. We recommend that you set the parameter to 6. Storage Engine: the storage engine of the topic. Valid values: Local Storage and Cloud Storage. |

| Description |
|--|
| The topic that is used to store the abnormal data of the connector. To save topic resources, you can create a topic as both the dead-letter queue topic and the abnormal data topic. |
| • Topic: the name of the topic. We recommend that you start the name with connect-error. |
| Partitions: the number of partitions in the topic. We recommend that you set the parameter to 6. |
| Storage Engine: the storage engine of the topic. Valid values: Local Storage and Cloud Storage. |
| |

Create consumer groups for a MaxCompute sink connector

In the Message Queue for Apache Kafka console, you can manually create the two consumer groups that are required by a MaxCompute sink connector.

- 1. Log on to the Message Queue for Apache Kafka console.
- 2. In the top navigation bar, select the region where your instance is located.
- 3. In the left-side navigation pane, click **Instances**.
- 4. On the **Instances** page, click the name of the instance that you want to manage.
- 5. In the left-side navigation pane, click **Consumer Groups**.
- 6. On the **Consumer Groups** page, click **Create Consumer Group**.
- 7. In the **Create Consumer Group** panel, set the properties of a consumer group and click **Create**.

| Consumer Group | Description |
|-------------------------------|--|
| Connector task consumer group | The consumer group that is used by the data synchronization task of the connector. The name of this consumer group must be in the connect- <i>T</i> ask name format. |
| Connector consumer group | The consumer group that is used by the connector. We recommend that you start the name of this consumer group with connect-cluster. |

Create and deploy a MaxCompute sink connector

To create and deploy a MaxCompute sink connector that is used to synchronize data from Message Queue for Apache Kafka to MaxCompute, perform the following steps:

- 1. Log on to the Message Queue for Apache Kafka console.
- 2. In the top navigation bar, select the region where your instance is located.
- 3. In the left-side navigation pane, click Instances.
- 4. On the **Instances** page, click the name of the instance that you want to manage.
- 5. In the left-side navigation pane, click Connector (Public Preview).
- 6. On the Connector (Public Preview) page, click Create Connector.

- 7. In the Create Connector panel, perform the following steps:
 - i. In the Enter Basic Information step, set the parameters that are described in the following table, and click Next.

| Parameter | Description | Example |
|-------------------|--|---|
| Connector Name | The name of the connector. Take note of the following rules when you specify a connector name: The connector name must be 1 to 48 characters in length. It can contain digits, lowercase letters, and hyphens (-), but cannot start with a hyphen (-). The connector name must be unique within a Message Queue for Apache Kafka instance. The data synchronization task of the connector must use a consumer group that is named in the connect-<i>Task nam e</i> format. If you have not manually created such a consumer group, the system automatically creates a consumer group for you. | kafka- maxcompute- sink |
| Dump Path | The source and destination of data transfer. Select a data source from the first drop-down list and a destination from the second drop-down list. | Message Queue for Apache Kafka Dump To MaxCompute |

ii. In the **Configure Source Instance** step, set the parameters that are described in the following table, and click **Next**.

? Note If you have created the required topics and consumer groups, set Create Resource to Manually and enter the information about the created resources. Otherwise, set Create Resource to Automatically.

| Parameter | Description | Example |
|----------------------|---|---------------------------|
| VPC ID | The ID of the virtual private cloud (VPC) where the data synchronization task runs. The default value is the ID of the VPC where the Message Queue for Apache Kafka instance is deployed. You do not need to enter a value. | vpc- bp1xpdnd3l*** |
| VSwitch | The ID of the vSwitch based on which the data synchronization task runs. The vSwitch must be deployed in the same VPC as the Message Queue for Apache Kafka instance. The default value is the ID of the vSwitch that you specify for the Message Queue for Apache Kafka instance. | vsw- bp1d2jgg81*** |
| Data Source Topic | The name of the topic from which data is to be synchronized. | maxcompute- test-input |

| Parameter | Description | Example |
|--------------------------------|--|---|
| Consumer Offset | The offset where consumption starts. Valid values: latest: Consumption starts from the latest offset. earliest: Consumption starts from the initial offset. | latest |
| Connector consumer group | The consumer group that is used by the connector. We recommend that you start the name of this consumer group with connect-cluster. | connect-cluster- kafka- maxcompute- sink |
| Offset Storage Topic | The topic that is used to store consumer offsets. Topic: the name of the topic. We recommend that you start the name with connect-offset. Partitions: the number of partitions in the topic. Set it to a value greater than 1. Storage Engine: the storage engine of the topic. Set the parameter to Local Storage. cleanup.policy: the log cleanup policy for the topic. Set the parameter to compact. | connect-offset- kafka- maxcompute- sink |
| Task configuration Topic | The topic that is used to store task configurations. Topic: the name of the topic. We recommend that you start the name with connect-config. Partitions: the number of partitions in the topic. Set the parameter to 1. Storage Engine: the storage engine of the topic. Set the parameter to Local Storage. cleanup.policy: the log cleanup policy for the topic. Set the parameter to compact. | connect-config- kafka- maxcompute- sink |
| Task status Topic | The topic that is used to store task status. Topic: the name of the topic. We recommend that you start the name with connect-status. Partitions: the number of partitions in the topic. We recommend that you set the parameter to 6. Storage Engine: the storage engine of the topic. Set the parameter to Local Storage. cleanup.policy: the log cleanup policy for the topic. Set the parameter to compact. | connect-status- kafka- maxcompute- sink |

| Parameter | Description | Example |
|----------------------------|---|---|
| Dead letter queue Topic | The topic that is used to store the abnormal data of the connector framework. To save topic resources, you can create a topic as both the dead-letter queue topic and the abnormal data topic. Topic: the name of the topic. We recommend that you start the name with connect-error. Partitions: the number of partitions in the topic. We recommend that you set the parameter to 6. Storage Engine: the storage engine of the topic. Valid values: Local Storage and Cloud Storage. | connect-error- kafka- maxcompute- sink |
| Abnormal Data Topic | The topic that is used to store the abnormal data of the connector. To save topic resources, you can create a topic as both the dead-letter queue topic and the abnormal data topic. Topic: the name of the topic. We recommend that you start the name with connect-error. Partitions: the number of partitions in the topic. We recommend that you set the parameter to 6. Storage Engine: the storage engine of the topic. Valid values: Local Storage and Cloud Storage. | connect-error- kafka- maxcompute- sink |

iii. In the **Configure Destination Instance** step, set the parameters that are described in the following table, and click **Next**.

| Parameter | Description | Example |
|------------------------|--|--|
| MaxCompute Endpoint | The endpoint of MaxCompute. For more information, see Configure endpoints. VPC endpoint: We recommend that you use the VPC endpoint because it has lower latency. The VPC endpoint can be used when the Message Queue for Apache Kafka instance and the MaxCompute workspace are in the same region. Public endpoint: We recommend that you do not use the public endpoint because it has high latency. The public endpoint can be used when the Message Queue for Apache Kafka instance and the MaxCompute workspace are in different regions. To use the public endpoint, you must enable Internet access for the connector. For more information, see Enable Internet access for a connector. | http://service.c n- hangzhou.maxc ompute.aliyun- inc.com/api |

| Parameter | Description | Example |
|----------------------------------|---|--------------------------------|
| MaxCompute Workspace | The name of the MaxCompute workspace to which you want to synchronize data. | connector_test |
| MaxCompute Table | The name of the MaxCompute table to which you want to synchronize data. | test_kafka |
| Region of MaxCompute Table | The region where the MaxCompute table is created. | China (Hangzhou) |
| Alibaba Cloud Account ID | The ID of the Alibaba Cloud account that is used to access MaxCompute. | 188*** |
| RAM Role | The name of the RAM role that is assumed by Message Queue for Apache Kafka . For more information, see Create a RAM role. | AliyunKafkaMaxC omputeUser1 |
| Mode | The mode in which messages are synchronized to the MaxCompute sink connector. Default value: DEFAULT. Valid values: KEY: Only the keys of messages are retained and written into the Key column of the MaxCompute table. VALUE: Only the values of messages are retained and written into the Value column of the MaxCompute table. DEFAULT: Both keys and values of messages are retained and written into the Key and Value columns of the MaxCompute table. DEFAULT: Both keys and values of messages are retained and written into the Key and Value columns of the MaxCompute table. DEFAULT: Both keys and values of messages are retained and written into the Key and Value columns of the MaxCompute table. DEFAULT: Notice In DEFAULT mode, the CSV format is not supported. You can select only the TEXT and BINARY formats. | DEFAULT |

| Parameter | Description | Example |
|-----------|--|-----------|
| Format | The format in which messages are synchronized to the MaxCompute sink connector. Default value: TEXT. Valid values: TEXT: strings BINARY: byte arrays CSV: strings separated with commas (,) Notice If you set the parameter to CSV, the DEFAULT mode is not supported. Only the KEY and VALUE modes are supported. KEY mode: Only the keys of messages are retained. Keys are separated with commas (,) and then written into the MaxCompute table in the order of indexes. VALUE mode: Only the values of messages are retained. Values are separated with commas (.) and then written into the MaxCompute table in the order of indexes. | TEXT |
| Partition | The granularity level at which partitions are created. Default value: HOUR. Valid values: DAY: writes data into a new partition every day. HOUR: writes data into a new partition every hour. MINUTE: writes data into a new partition every minute. | HOUR |
| Time Zone | The time zone of the Message Queue for Apache Kafka producer client that sends messages to the data source topic of the MaxCompute sink connector. Default value: GMT 08:00. | GMT 08:00 |

iv. In the Preview/Submit step, confirm the configurations of the connector and click Submit.

8. In the Create Connector panel, click Deploy.

Send a test message

After you deploy the MaxCompute sink connector, you can send a message to the data source topic in Message Queue for Apache Kafka to test whether the message can be synchronized to MaxCompute.

- 1. On the **Connector (Public Preview)** page, find the connector that you created, and click **Test** in the **Actions** column.
- 2. On the **Topics** page, select your instance, find the **maxcompute-test-input** topic, click the More icon in the **Actions** column, and then select **Send Message**.
- 3. In the Send Message panel, set the parameters used to send a test message.
 - i. In the **Partitions** field, enter *O*.
 - ii. In the Message Key field, enter 1.
 - iii. In the Message Value field, enter 1.
 - iv. Click Send.

View data in the MaxCompute table

After you send a message to the data source topic in Message Queue for Apache Kafka , you can log on to the MaxCompute client and check whether the message is received.

To view the test_kafka table, perform the following steps:

- 1. Log on to the MaxCompute client.
- 2. Run the following command to view the partitions of the table:

show partitions test_kafka;

The following result is returned in this example:

pt=11-17-2020 15 OK

3. Run the following command to view the data stored in the partitions:

select * from test_kafka where pt ="11-17-2020 14";

The following result is returned in this example:

+-----+ | topic | partition | offset | key | value | pt | +-----+ | maxcompute-test-input | 0 | 0 | 1 | 1 | 11-17-2020 14 | +-----+

6.3.3. Create an OSS sink connector

This topic describes how to create an OSS sink connector to synchronize data from a topic in your Message Queue for Apache Kafka instance to Object Storage Service (OSS).

Prerequisites

Before you export data, make sure that the following operations are completed:

• The connector feature is enabled for the Message Queue for Apache Kafka instance. For more information, see Enable the connector feature.

- A data source topic is created in the Message Queue for Apache Kafka instance. For more information, see Step 1: Create a topic.
- Buckets are created in the OSS console. For more information, see Create buckets.
- Function Compute is activated. For more information, see Create a function in the Function Compute console.

Considerations

- To synchronize data from Message Queue for Apache Kafka to OSS, the Message Queue for Apache Kafka instance that contains the data source topic and the destination OSS bucket must be in the same region. Message Queue for Apache Kafka first synchronizes the data to Function Compute. Then, Function Compute synchronizes the data to OSS. For information about the limits on connectors, see Limits.
- OSS sink connectors are provided based on Function Compute. Function Compute provides you with a free quota. If your usage exceeds the free quota, you are charged for the excess based on the billing rules of Function Compute. For information about the billing details, see Billing.
- Function Compute allows you to query the logs of function calls. For more information, see Configure Log Service resources and view function execution logs.
- For message transfer, Message Queue for Apache Kafka serializes data into UTF-8-encoded strings and does not support the BINARY data type.

Create and deploy an OSS sink connector

- 1. Log on to the Message Queue for Apache Kafka console.
- 2. In the top navigation bar, select the region where your instance is located.
- 3. In the left-side navigation pane, click Instances.
- 4. On the **Instances** page, click the name of the instance that you want to manage.
- 5. In the left-side navigation pane, click **Connector (Public Preview)**.
- 6. On the Connector (Public Preview) page, click Create Connector.
- 7. In the Create Connector wizard, perform the following steps:

i. In the Enter Basic Information step, set the parameters that are described in the following table, and click Next.

Notice By default, the Authorize to Create Service Linked Role check box is selected. This means that Message Queue for Apache Kafka will create a service-lined role based on the following rules:

- If no service-linked role is created, Message Queue for Apache Kafka automatically creates a service-linked role for you to use the OSS sink connector to synchronize data from Message Queue for Apache Kafka to OSS.
- If you have created a service-linked role, Message Queue for Apache Kafka does not create it again.

| Parameter | Description | Example |
|-------------------|--|---|
| Connector Name | The name of the connector. Take note of the following rules when you specify a connector name: The connector name must be 1 to 48 characters in length. It can contain digits, lowercase letters, and hyphens (-), but cannot start with a hyphen (-). The connector name must be unique within a Message Queue for Apache Kafka instance. The data synchronization task of the connector must use a consumer group that is named in the connect-<i>Task nam e</i> format. If you have not manually created such a consumer group, the system automatically creates a consumer group for you. | kafka-oss-sink |
| Dump Path | The source and destination of data transfer. Select a data source from the first drop-down list and a destination from the second drop-down list. | Message Queue for Apache Kafka Dump To Object Storage Service |

For more information about service-linked roles, see Service-linked roles.

ii. In the **Configure Source Instance** step, set the parameters that are described in the following table, and click **Next**.

| Parameter | Description | Example |
|--------------------------------|---|----------------|
| Data Source Topic | The name of the topic from which data is to be synchronized. | oss-test-input |
| Consumer Offset | The offset where consumption starts. Valid values: latest: Consumption starts from the latest offset. earliest: Consumption starts from the initial offset. | latest |
| Consumer Thread Concurrency | The number of concurrent consumer threads to synchronize data from the data source topic. Default value: 6. Valid values: 6 12 | 6 |

iii. In the **Configure Destination Instance Configure Runtime Environment** step, set the parameters related to the destination OSS bucket.

| Parameter | Description | Example |
|------------------|---|-----------------------------------|
| Bucket Name | The name of the destination OSS bucket. | bucket_test |
| AccessKey ID | The AccessKey ID of your Alibaba Cloud account. | LT AI4GG2RGAjppjK******* |
| AccessKey Secret | The AccessKey secret of your Alibaba Cloud account . | WbGPVb5rrecVw3SQvEPw6R** ***** |

Make sure that your Alibaba Cloud account is granted at least the following permissions:

```
{
   "Version": "1",
   "Statement": [
    {
        "Action": [
           "oss:GetObject",
           "oss:PutObject"
    ],
        "Resource": "*",
        "Effect": "Allow"
    }
 ]
}
```

? Note

When Message Queue for Apache Kafka creates the data synchronization task, the AccessKey ID and AccessKey Secret parameters are passed to Function Compute as environment variables. Message Queue for Apache Kafka does not store the AccessKey ID or AccessKey secret of your Alibaba Cloud account.

iv. In the **Configure Destination Instance Configure Runtime Environment** step, set the parameters that are described in the following table, and click **Next**.

| Parameter | Description | Example |
|-----------|--|-------------------|
| VPC ID | The ID of the virtual private cloud (VPC) where the data synchronization task runs. The default value is the ID of the VPC where the Message Queue for Apache Kafka instance is deployed. You do not need to enter a value. | vpc-bp1xpdnd3l*** |

| Parameter | fail: stops the subscription Description to the partition where an | Example |
|-------------------------|--|---------------|
| Failure Handling Policy | error occurs and prints the logs. After an error occurs, you can view the error in the connector logs. Then, you can troubleshoot the error based on the returned error code. | log |
| | For information about how to view the connector logs, see View connector logs. For information about how to troubleshoot errors based on error codes, see Error codes. To resume the subscription to the partition where an error occurs, submit a ticket to Message Queue for Apache Kafka Customer Services. | |
| | | |
| Create Resource | The mode in which to create the consumer group and topics used for data synchronization. Valid values: Automatically and Manually . If you select Manually, enter resource names. | Automatically |

| Parameter | Description | Example |
|-----------------------------|---|------------------------------------|
| Connector consumer group | The consumer group that is used by the connector. We recommend that you start the name of this consumer group with connect-cluster. | connect-cluster-kafka-oss- sink |
| Offset Storage Topic | The topic that is used to store consumer offsets. Topic: the name of the topic. We recommend that you start the name with connect-offset. Partitions: the number of partitions in the topic. Set it to a value greater than 1. Storage Engine: the storage engine of the topic. Set the parameter to Local Storage. cleanup.policy: the log cleanup policy for the topic. Set the parameter to compact. | connect-offset-kafka-oss-sink |
| Task configuration Topic | The topic that is used to store task configurations. Topic: the name of the topic. We recommend that you start the name with connect-config. Partitions: the number of partitions in the topic. Set the parameter to 1. Storage Engine: the storage engine of the topic. Set the parameter to Local Storage. cleanup.policy: the log cleanup policy for the topic. Set the parameter to compact. | connect-config-kafka-oss-sink |

| Parameter | Description | Example |
|---------------------|---|-------------------------------|
| Task status Topic | The topic that is used to store task status. Topic: the name of the topic. We recommend that you start the name with connect-status. Partitions: the number of partitions in the topic. We recommend that you set the parameter to 6. Storage Engine: the storage engine of the topic. Set the parameter to Local Storage. cleanup.policy: the log cleanup policy for the topic. Set the parameter to compact. | connect-status-kafka-oss-sink |
| Abnormal Data Topic | The topic that is used to store the abnormal data of the connector. To save topic resources, you can create a topic as both the Dead letter queue Topic and the abnormal data topic. Topic: the name of the topic. We recommend that you start the name with connect-error. Partitions: the number of partitions in the topic. We recommend that you set the parameter to 6. Storage Engine: the storage engine of the topic. Valid values: Local Storage and Cloud Storage. | connect-error-kafka-oss-sink |

| Parameter | Description | Example |
|-------------------------|--|------------------------------|
| | The topic that is used to store the abnormal data of the connector framework. To save topic resources, you can create a topic as both the dead-letter queue topic and the Abnormal Data Topic topic. | |
| Dead letter queue Topic | Topic: the name of the topic. We recommend that you start the name with connect-error. | connect-error-kafka-oss-sink |
| | Partitions: the number of partitions in the topic. We recommend that you set the parameter to 6. | |
| | Storage Engine: the storage engine of the topic. Valid values: Local Storage and Cloud Storage. | |
| | | |

v. In the **Preview/Submit** step, confirm the configurations of the connector and click **Submit**.

8. In the Create Connector panel, click Deploy.

Send a test message

After you deploy the OSS sink connector, you can send a message to the data source topic in Message Queue for Apache Kafka to test whether the message can be synchronized to OSS.

- 1. On the **Connector (Public Preview)** page, find the connector that you created, and click **Test** in the **Actions** column.
- 2. On the **Topics** page, select your instance, find the data source topic, click the More icon in the **Actions** column, and then select **Send Message**.
- 3. In the **Send Message** panel, set the parameters used to send a test message.
 - i. In the Partitions field, enter 0.
 - ii. In the Message Key field, enter 1.
 - iii. In the Message Value field, enter 1.
 - iv. Click Send.

Verify the results

After you send a test message to the data source topic in Message Queue for Apache Kafka, you can check whether the message is synchronized to OSS on the Files page of the destination OSS bucket in the OSS console. For more information, see <u>Overview</u>.

If new objects are generated in the OSS bucket, the data is synchronized to OSS, as shown in the following figure.

| | Man Charles Carlins 1 | | ett | | | | | | T. 1 20 |
|----|-------------------------|------|----------|---|----------------------|----------------|--------------------|-------------------------------------|------------------------|
| ľ | oject storage service / | | riles | | | | | | 185 |
| | / China (Hangzho | u) ~ | | Ve | rsioning Unversioned | Access Control | List (ACL) Private | Storage Class Standard(Locally Redu | indant Storage) |
| | Overview | | Upload | Create Folder Parts Authorize Batch Operation V Refresh | | | | Enter a file name prefix | Q 🕸 |
| l, | Data Usage | > | | File Name | | Size | Storage Class | Updated At | Actions |
| L | Files | > | 5 | / <u>1 1/ 2021-02-25/</u> 0/ | | | | | |
| | Access Control | > | | 0000009_1_87D2bV | | 0.143KB | Standard | Feb 25, 2021, 12:41:40 | View Details More 🗸 |
| | Basic Settings | > | | 0000010_1_/UNVL | | 0.144KB | Standard | Feb 25, 2021, 12:43:00 | View Details |
| | Transmission | > | | 0000011_1_0/W-Qr | | 0.144KB | Standard | Feb 25, 2021, 12:46:56 | View Details More 🗸 |
| | Logging | > | | 0000012_1_EOv0Pw | | 0.162KB | Standard | Feb 25, 2021, 12:56:12 | View Details More 🗸 |
| | Data Processing | > | | | | | | | |

The data that is synchronized from Message Queue for Apache Kafka to OSS is in the following format:

```
[
{
    "key":"123",
    "offset":4,
    "overflowFlag":true,
    "partition":0,
    "timestamp":1603779578478,
    "topic":"Test",
    "value":"1",
    "valueSize":272687
}
]
```

Perform other operations

You can configure the Function Compute resources that are required by the OSS sink connector based on actual needs.

1. On the **Connector (Public Preview)** page, find the connector that you created, and click **Configure Function** in the **Actions** column.

You are redirected to the Function Compute console, where you can configure the resources as required.

6.3.4. Create an Elasticsearch sink connector

This topic describes how to create an Elasticsearch sink connector to export data from a topic in your Message Queue for Apache Kafka instance to an index of your Elasticsearch instance.

Prerequisites

Before you start, make sure that the following operations are complete:

- Enable the connector feature for your Message Queue for Apache Kafka instance. For more information, see Enable the connector feature.
- Create topics in your Message Queue for Apache Kafka instance. For more information, see Step 1: Create a topic.
- Create an instance and index in the Elasticsearch console. For more information, see Quick start.

(?) Note The version of the Elasticsearch client used by Function Compute is 7.7.0. To maintain compatibility, create an Elasticsearch instance of version 7.0 or later.

• Activate Function Compute.

Usage notes

- To export data from Message Queue for Apache Kafka to Elasticsearch, the Message Queue for Apache Kafka instance that contains the data source topic and the Elasticsearch instance must be in the same region. Message Queue for Apache Kafka first exports the data to Function Compute. Then, Function Compute exports the data to Elasticsearch. For information about limits on connectors, see Limits.
- Elasticsearch sink connectors are provided based on Function Compute. Function Compute provides you with a free quota. If your usage exceeds the free quota, you are charged for the excess based on the billing rules of Function Compute. For more information, see Billing.
- Function Compute allows you to query the logs of function calls. For more information, see Configure Log Service resources and view function execution logs.
- Message Queue for Apache Kafka serializes messages into UTF-8-encoded strings for message transfer. Message Queue for Apache Kafka does not support the BINARY data type.

Create and deploy an Elasticsearch sink connector

- 1. Log on to the Message Queue for Apache Kafka console.
- 2. In the top navigation bar, select the region where your instance is located.
- 3. In the left-side navigation pane, click Instances.
- 4. On the **Instances** page, click the name of the instance that you want to manage.
- 5. In the left-side navigation pane, click **Connector (Public Preview)**.
- 6. On the Connector (Public Preview) page, click Create Connector.
- 7. In the Create Connector wizard, perform the following steps:

i. In the Enter Basic Information step, set Connector Name, configure a dump path from Message Queue for Apache Kafka to Elasticsearch for Dump Path, and then click Next.

Notice By default, **Authorize to Create Service Linked Role** is selected. This means that Message Queue for Apache Kafka will create a service-linked role based on the following rules:

- If you have not created a service-linked role, Message Queue for Apache Kafka automatically creates a service-linked role for you to export data from Message Queue for Apache Kafka to Elasticsearch.
- If you have created a service-linked role, Message Queue for Apache Kafka does not create it again.

| Parameter | Description | Example |
|-------------------|---|--|
| Connector Name | The name of the connector. Take note of the following rules when you specify a connector name: The connector name must be 1 to 48 characters in length. It can contain digits, lowercase letters, and hyphens (-), but cannot start with a hyphen (-). The connector name must be unique within the Message Queue for Apache Kafka instance. The data synchronization task of the connector must use a consumer group that is named in the connect-<i>Task nam e</i> format. If you have not created such a consumer group, the system automatically creates a consumer group for you. | kafka- elasticsearch- sink |
| Dump Path | The source and destination of data transfer. Select a data source from the first drop-down list and a destination from the second drop-down list. | Dump from Message Queue for Apache Kafka to Elasticsearch |

For more information about service-linked roles, see Service-linked roles.

ii. In the **Configure Source Instance** step, set the parameters that are described in the following table, and click **Next**.

| Parameter | Description | Example |
|--------------------------------|---|--------------------------|
| Data Source Topic | The name of the topic from which data is to be synchronized. | elasticsearch-test-input |
| Consumer Offset | The offset where consumption starts. Valid values: latest: Consumption starts from the latest offset. earliest: Consumption starts from the initial offset. | latest |
| Consumer Thread Concurrency | The number of concurrent consumer threads to synchronize data from the data source topic. Default value: 6. Valid values: 6 12 | 6 |

iii. In the **Configure Destination Instance Configure Runtime Environment** step, set the parameters related to the destination Elasticsearch instance.

| Parameter | Description | Example |
|---------------------------|---|---|
| Elasticsearch Instance ID | The ID of the Elasticsearch instance. | es-cn-oew1o67x0000**** |
| Endpoint | The public or private endpoint of the Elasticsearch instance. For more information, see View the basic information of a cluster. | es-cn- oew1o67x0000****.elasticsear ch.aliyuncs.com |
| Port | The public or private port used to access the Elasticsearch instance. Valid values: 9200: for HTTP and HTTPS 9300: for TCP For more information, see View the basic information of a cluster. | 9300 |

| Parameter | Description | Example |
|-----------|--|--------------|
| Username | The username that is used to log on to the Kibana console. Default value: elastic. You can also customize the username. For more information, see Create a user. | elastic |
| Password | The password used to log on to the Kibana console. The password of the elastic user is specified when you create the Elasticsearch instance. If you forget the password, you can reset it. For more information about how to reset the password, see Reset the access password for an Elasticsearch cluster. | ***** |
| Index | The name of the Elasticsearch index. | elastic_test |

? Note

- The username and password will be used to initialize Elasticsearch objects. To ship messages by using bulk, make sure that the account has the write permission on the index.
- The username and password are passed to functions of Function Compute as environment variables when Message Queue for Apache Kafka creates a synchronization task. After the synchronization task is created, Message Queue for Apache Kafka does not save the username or password.
- iv. In the **Configure Destination Instance Configure Runtime Environment** section, set the parameters as required and click **Next**.

| Parameter | Description | Example |
|-----------|---|-------------------|
| VPC ID | The ID of the virtual private cloud (VPC) where the data synchronization task runs. The default value is the ID of the VPC where the Message Queue for Apache Kafka instance is deployed. You do not need to change the value. | vpc-bp1xpdnd3l*** |

| Parameter | Description | Example |
|-----------|--|-------------------|
| VSwitch | The ID of the vSwitch based on which the data synchronization task runs. The vSwitch must be deployed in the same VPC as the Message Queue for Apache Kafka instance. The default value is the ID of the vSwitch that you specify for the Message Queue for Apache Kafka instance. | vsw-bp1d2jgg81*** |
| | The error handling policy for a message that fails to be sent. Default value: log. Valid values: log and fail. I log: retains the subscription to the partition where an error occurs and prints the logs. After an error occurs, you can view the error in the connector logs. Then, you can troubleshoot the error based on the returned error code. Note For more information about how to view the connector logs. For more information about how to troubleshoot errors based on error codes, see Error codes. | |
| | | |

| | fail: stops the subscription | |
|-------------------------|--|---------|
| Parameter | Description to the partition where an | Example |
| | error occurs and prints the logs. After an error occurs, you can view the error in the connector logs. Then, you can troubleshoot the error based on the returned error code. | |
| Failure Handling Policy | Note For more information about how to view the connector logs, see View connector logs. For more information about how to troubleshoot errors based on error codes, see Error codes. To resume the subscription to the partition | log |
| | where an error occurs,submit a ticket to Message Queue for Apache Kafka Customer Services. | |
| | | |

| Parameter | Description | Example |
|-----------------------------|---|--|
| | | |
| Create Resource | The mode in which to create the consumer group and topics used for data synchronization. Valid values: Automatically and Manually . If you select Manually, enter resource names. | Automatically |
| Connector consumer group | The consumer group that is used by the connector. We recommend that you start the name of this consumer group with connect-cluster. | connect-cluster-kafka- elasticsearch-sink |
| Offset Storage Topic | The topic that is used to store consumer offsets. Topic: the name of the topic. We recommend that you start the name with connect-offset. Partitions: the number of partitions in the topic. Set it to a value greater than 1. Storage Engine: the storage engine of the topic. Set the value to Local Storage. cleanup.policy: the log cleanup policy for the topic. Set the value to compact. | connect-offset-kafka- elasticsearch-sink |

| Parameter | Description | Example |
|--------------------------|--|---|
| Task configuration Topic | The topic that is used to store task configurations. Topic: the name of the topic. We recommend that you start the name with connect-config. Partitions: the number of partitions in the topic. Set the value to 1. Storage Engine: the storage engine of the topic. Set the value to Local Storage. cleanup.policy: the log cleanup policy for the topic. Set the value to compact. | connect-config-kafka- elasticsearch-sink |
| Task status Topic | The topic that is used to store task status. Topic: the name of the topic. We recommend that you start the name with connect-status. Partitions: the number of partitions in the topic. We recommend that you set the value to 6. Storage Engine: the storage engine of the topic. Set the value to Local Storage. cleanup.policy: the log cleanup policy for the topic. Set the value to compact. | connect-status-kafka- elasticsearch-sink |

| Parameter | Description | Example |
|-------------------------|--|--|
| Abnormal Data Topic | The topic that is used to store the error data of the sink connector. To save topic resources, you can create a topic as both the Dead letter queue Topic and the error data topic. Topic: the name of the topic. We recommend that you start the name with connect-error. Partitions: the number of partitions in the topic. We recommend that you set the value to 6. Storage Engine: the storage engine of the topic. Set the value to Local Storage or Cloud Storage. | connect-error-kafka- elasticsearch-sink |
| Dead letter queue Topic | The topic that is used to store the error data of the connector framework. To save topic resources, you can create a topic as both the dead-letter queue topic and the Abnormal Data Topic. Topic: the name of the topic. We recommend that you start the name with connect-error. Partitions: the number of partitions in the topic. We recommend that you set the value to 6. Storage Engine: the storage engine of the topic. Set the value to Local Storage or Cloud Storage. | connect-error-kafka- elasticsearch-sink |

v. In the **Preview/Submit** step, confirm the configurations of the connector and click **Submit**.

8. In the connector list, find the connector that you created, and click **Actions** in the **Deploy** column. If the connector status changes to **Running**, the connector is deployed.

Create Function Compute resources

After you create and deploy the Elasticsearch sink connector in the Message Queue for Apache Kafka console. Function Compute automatically creates a service for the connector and names it in the service-<connector_name>-<random string> format.

- On the Connector (Public Preview) page, find the connector that you created, click the More icon in the Actions column, and then select Configure Function. The page is redirected to the Function Compute console.
- 2. In the Function Compute console, find the automatically created service and configure a VPC and vSwitch for the service. Make sure that the VPC and vSwitch are the same as those of your Elasticsearch instance. For more information, see Modify a service.

Send a test message

You can send a message to the data source topic in Message Queue for Apache Kafka to test whether the data can be exported to Elasticsearch.

- 1. On the **Connector (Public Preview)** page, find the connector that you created, and click **Test** in the **Actions** column.
- 2. On the **Topics** page, select your instance, find the data source topic, click the More icon in the **Actions** column, and then select **Send Message**.
- 3. In the **Send Message** panel, set the parameters used to send a test message.
 - i. In the Partitions field, enter 0.
 - ii. In the Message Key field, enter 1.
 - iii. In the Message Value field, enter 1.
 - iv. Click Send.

Verify the result

After you send a message to the data source topic in your Message Queue for Apache Kafka instance, log on to the Kibana console and run the GET /<index_name>/_search command to view the Elasticsearch index and verify whether data is exported.

The following code shows an example of the data exported from Message Queue for Apache Kafka to Elasticsearch.

```
{
 "took":8,
 "timed_out" : false,
 "_shards":{
 "total":5,
 "successful": 5,
 "skipped":0,
 "failed":0
},
 "hits":{
 "total":{
  "value":1,
  "relation": "eq"
 },
 "max_score": 1.0,
 "hits":[
  {
   "_index" : "product_****",
   "_type":"_doc",
   "_id" : "TX3TZHgBfHNEDGoZ****",
   "_score": 1.0,
   "_source":{
    "msg_body":{
     "key":"test",
     "offset":2,
     "overflowFlag": false,
     "partition":2,
     "timestamp" : 1616599282417,
     "topic": "dv****",
     "value" : "test1",
     "valueSize":8
    },
    "doc_as_upsert": true
   }
  }
 ]
}
}
}
```

6.4. View task configurations of a connector

This topic describes how to view the task configurations of a connector in the Message Queue for Apache Kafka console. The configurations include basic information, source instance configurations, and destination instance configurations of the connector.

Prerequisites

One of the connectors that are described in the following topics is created:

- Create a Function Compute sink connector
- Create a MaxCompute sink connector
- Create an OSS sink connector
- Create an Elasticsearch sink connector
- 创建MySQL Source Connector

Procedure

- 1. Log on to the Message Queue for Apache Kafka console.
- 2. In the top navigation bar, select the region where your instance is located.
- 3. In the left-side navigation pane, click Instances.
- 4. On the **Instances** page, click the name of the instance that you want to manage.
- 5. In the left-side navigation pane, click **Connector**.
- 6. On the **Connector (Public Preview)** page, find the connector whose configurations you want to view, click the **:** icon in the **Actions** column, and then select **View task configuration**.

The View task configuration panel displays the task configurations of the connector.

6.5. View connector logs

This topic describes how to view connector logs in the Message Queue for Apache Kafka console. The logs provide information for troubleshooting.

Prerequisites

One of the connectors that are described in the following topics is created:

- Create a Function Compute sink connector
- Create a MaxCompute sink connector
- Create an OSS sink connector
- Create an Elasticsearch sink connector
- 创建MySQL Source Connector

Procedure

- 1. Log on to the Message Queue for Apache Kafka console.
- 2. In the top navigation bar, select the region where your instance is located.
- 3. In the left-side navigation pane, click Instances.
- 4. On the **Instances** page, click the name of the instance that you want to manage.
- 5. In the left-side navigation pane, click **Connector**.
- On the Connector (Public Preview) page, find the connector whose logs you want to view, and click View Log in the Actions column. The View Log dialog box displays the logs of the connector.

6.6. Suspend a connector

This topic describes how to suspend a running connector in the Message Queue for Apache Kafka console.

Prerequisites

One of the connectors that are described in the following topics is created, and the connector is in the Running state:

- Create a Function Compute sink connector
- Create a MaxCompute sink connector
- Create an OSS sink connector
- Create an Elasticsearch sink connector
- 创建MySQL Source Connector

Procedure

- 1. Log on to the Message Queue for Apache Kafka console.
- 2. In the top navigation bar, select the region where your instance is located.
- 3. In the left-side navigation pane, click Instances.
- 4. On the **Instances** page, click the name of the instance that you want to manage.
- 5. In the left-side navigation pane, click **Connector**.
- 6. On the **Connector (Public Preview)** page, find the connector that you want to suspend, click the **i** icon in the **Actions** column, and then select **Suspended**.
- In the Note message, click OK.
 The Status column displays Suspended for the connector.

6.7. Resume a connector

This topic describes how to resume a suspended connector in the Message Queue for Apache Kafka console.

Prerequisites

One of the connectors that are described in the following topics is created, and the connector is in the Suspended state:

- Create a Function Compute sink connector
- Create a MaxCompute sink connector
- Create an OSS sink connector
- Create an Elasticsearch sink connector
- 创建MySQL Source Connector

Procedure

- 1. Log on to the Message Queue for Apache Kafka console.
- 2. In the top navigation bar, select the region where your instance is located.
- 3. In the left-side navigation pane, click **Instances**.
- 4. On the **Instances** page, click the name of the instance that you want to manage.
- 5. In the left-side navigation pane, click **Connector**.
- 6. On the Connector (Public Preview) page, find the connector that you want to resume, click the

- icon in the **Actions** column, and then select **Resume**.
- In the Note message, click OK. The Status column displays Running for the connector.

6.8. Delete a connector

Message Queue for Apache Kafka limits the number of connectors for each instance. If you no longer need a connector, you can delete it in the Message Queue for Apache Kafka console.

Prerequisites

One of the connectors that are described in the following topics is created:

- Create a Function Compute sink connector
- Create a MaxCompute sink connector
- Create an OSS sink connector
- Create an Elasticsearch sink connector
- 创建MySQL Source Connector

Procedure

- 1. Log on to the Message Queue for Apache Kafka console.
- 2. In the top navigation bar, select the region where your instance is located.
- 3. In the left-side navigation pane, click Instances.
- 4. On the **Instances** page, click the name of the instance that you want to manage.
- 5. In the left-side navigation pane, click Connector.
- 6. On the **Connector (Public Preview)** page, find the connector that you want to delete, click the icon in the **Actions** column, and then select **Delete**.
- 7. In the Delete message, click OK.

Notice When you delete a connector, the system deletes the five topics and two consumer groups that the connector requires, regardless of whether they were automatically or manually created.

6.9. Modify connector configurations

After you create a Function Compute sink connector, you can modify its configurations in the Message Queue for Apache Kafka console.

Prerequisites

Create a Function Compute sink connector

Procedure

- 1. Log on to the Message Queue for Apache Kafka console.
- 2. In the top navigation bar, select the region where your instance is located.
- 3. In the left-side navigation pane, click **Instances**.
- 4. On the **Instances** page, click the name of the instance that you want to manage.
- 5. In the left-side navigation pane, click Connector (Public Preview).
- 6. On the **Connector (Public Preview)** page, find the Function Compute sink connector whose configurations you want to modify, click the 👔 icon in the **Actions** column, and then select **Modify Configuration**.
- 7. In the **Modify Connector** panel, modify the following configurations as needed, and then click **OK**.

| Parameter | Description | | |
|-----------------------------|---|--|--|
| Consumer Thread Concurrency | The number of concurrent consumer threads to synchronize data from the data source topic. Default value: 3. Valid values: • 3 • 6 • 9 • 12 | | |
| | Specifies whether to continue subscribing to the partition where an error occurs after the message fails to be sent. Valid values: log: retains the subscription to the partition where an error occurs and prints the logs. fail: stops the subscription to the partition where an error occurs and prints the logs. | | |
| Retries | Note For more information about how to view the connector logs, see View connector logs. For information about how to troubleshoot errors based on error codes, see Error codes. To resume the subscription to the partition where an error occurs, submit a ticket to Message Queue for Apache Kafka Customer Services. | | |

| Parameter | Description | | |
|-----------|--|--|--|
| Retries | The number of retries allowed after a message fails to be sent. Default value: 2. Valid values: 1 to 3. In some cases where a message fails to be sent, retries are not supported. The following information describes the types of Error codes and whether they support retries: 4XX: does not support a retry except for 429. 5XX: supports a retry. | | |
| | Note The connector calls the InvokeFunction operation to send a message to Function Compute. | | |
| | | | |

Result

After the modification is complete, you can view the updated connector configurations in the **View** task configuration panel.

6.10. Test a connector

This topic describes how to test a connector by sending a test message to the connector in the Message Queue for Apache Kafka console.

Prerequisites

One of the connectors that are described in the following topics is created:

- Create a Function Compute sink connector
- Create a MaxCompute sink connector
- Create an OSS sink connector
- Create an Elasticsearch sink connector
- 创建MySQL Source Connector

Procedure

- 1. Log on to the Message Queue for Apache Kafka console.
- 2. In the top navigation bar, select the region where your instance is located.
- 3. In the left-side navigation pane, click **Instances**.
- 4. On the **Instances** page, click the name of the instance that you want to manage.
- 5. In the left-side navigation pane, click **Connector**.
- 6. On the **Connector (Public Preview)** page, find the connector that you want to test, and click **Test** in the **Actions** column.
- 7. On the **Topics** page, find the data source topic, and click **Send Message** in the **Actions** column.

- 8. In the **Send Message** panel, set the parameters used to send a test message.
 - i. In the Partitions field, enter 0.
 - ii. In the Message Key field, enter 1.
 - iii. In the Message Value field, enter 1.
 - iv. Click Send.

6.11. Enable Internet access for a connector

If you need to access other Alibaba Cloud services in other regions by using a connector, you must enable Internet access for the connector. This topic describes how to enable Internet access for a connector.

Prerequisites

Before you enable Internet access for a connector, make sure that the connector feature is enabled in your Message Queue for Apache Kafka instance. For more information, see Enable the connector feature.

Enable Internet access

Internet Alibaba Cloud Services Elastic IP addresses NAT Gateway VPC 1 (Region A) Message Queue for Apache Kafka VPC 2 (Region A) Connector

The following figure shows the solution to enabling Internet access for a connector.

To enable Internet access for a connector, perform the following steps:

- 1. Create a Network Address Translation (NAT) gateway for virtual private cloud (VPC) 1 where the Message Queue for Apache Kafka instance is deployed. For more information, see Create NAT gateways.
- 2. Bind an elastic IP address (EIP) to the created NAT gateway. For more information, see Associate an EIP with a NAT gateway.
- 3. Create Source Network Address Translation (SNAT) entries for the vSwitch that is used by the Message Queue for Apache Kafka instance on VPC 1. For more information, see Configure SNAT to

access the Internet.

7.Migration 7.1. Overview

This topic describes the advantages, principle, and process of migrating a user-created Kafka cluster to a Message Queue for Apache Kafka instance.

Advantages

For more information about the advantages of migrating a user-created Kafka cluster to a Message Queue for Apache Kafka instance, see Benefits.

How it works

To migrate a cluster for message queues, you only need to consume all messages in the old cluster. The producers and consumers are deployed in clusters and can be operated one by one, without being perceived by upper-layer services.

Procedure

Perform the following operations to migrate a user-created Kafka cluster to a Message Queue for Apache Kafka instance:

- 1. Evaluate the specifications of the user-created Kafka cluster to determine the edition of the Message Queue for Apache Kafka instance you want to purchase.For more information, see Evaluate specifications.
- 2. Purchase a Message Queue for Apache Kafka instance and deploy it based on the recommendation.





Message Queue for Apache Kafka

3. Migrate the topics of the user-created Kafka cluster to the Message Queue for Apache Kafka instance.For more information, see Migrate topic metadata from a user-created Kafka cluster to Message Queue for Apache Kafka.



4. Migrate the consumer groups of the user-created Kafka cluster to the Message Queue for Apache Kafka instance. For more information, see Migrate consumer group metadata from a user-created Kafka cluster to Message Queue for Apache Kafka.



5. (Optional)Migrate the data of the user-created Kafka cluster to the Message Queue for Apache Kafka instance.

Notice For message queues, after the data of a cluster is consumed, the data will not be used except for backup. Therefore, generally, we recommend that you do not migrate data except that you must back up data of the user-created Kafka cluster to the Message Queue for Apache Kafka instance.

For more information, see Migrate data to the cloud.



6. Enable a new consumer group for the Message Queue for Apache Kafka instance to consume messages of the Message Queue for Apache Kafka instance.



7. Enable a new producer for the Message Queue for Apache Kafka instance, deprecate the old producer, and enable the old consumer group to consume messages of the user-created Kafka cluster.



8. After all messages of the user-created Kafka cluster are consumed by the old consumer group, deprecate the old consumer group and the user-created Kafka cluster.



7.2. Evaluate specifications

This topic describes how to select an appropriate instance edition for a user-created Kafka cluster that is to be migrated to Alibaba Cloud by using the specification evaluation feature of Message Queue for Apache Kafka.

Prerequisites

An Alibaba Cloud account is created and real-name verification is completed. For more information, seeCreate your Alibaba Cloud account.

Evaluate specifications

? Note The specification evaluation feature of Message Queue for Apache Kafka follows the principle of lowest costs. We recommend Message Queue for Apache Kafka instances with the lowest price, which can meet your normal business needs.

To evaluate the specifications, perform the following steps:

- 1. Log on to the Message Queue for Apache Kafka console.
- 2. In the top navigation bar, select the region where your instance is located.
- 3. In the left-side navigation pane, click Migration.
- 4. Click the **Evaluation** tab.
- 5. In the **Instance Specifications** section, enter the specification information of the user-created Kafka cluster and click **Evaluate**.

The recommended Message Queue for Apache Kafka instance edition appears in the **Recommended** section.

| Evaluation Migration | | | | | | |
|----------------------------------|--------------|--------------------|--------------------------|----------------------------------|------------------|---------------|
| Instance Specifications | | | | | | |
| Major Version 🕘 0.10.x 🗸 🗸 | | | Migration across IDCs (| 🕽 🛞 Yes 🔘 No | | |
| Cluster Peak Traffic @ 1000 MB/s | | | Public Network Traffic @ | 0 MB/s | | |
| Disk Type 🕘 🖲 Ultra Disk 🕓 SSD | | | Disk Space 🔕 500 | G | | |
| Topics @ 50 | | | Partitions 😰 1000 | | | |
| Replicas 🔕 3 🗸 | | | | | | |
| Instance Application | | | | | | |
| Scenario 🔕 Big Data Computing 🗸 | | | Response Latency @ | No Special Requirements $\ \lor$ | | |
| Evaluate Reset | | | | | | |
| Recommended | | | | | | |
| Instance Type | Cluster Type | Traffic Peak Value | Disk Type | Disk Space | Supported Topics | Major Version |
| Professional Edition | VPC Instance | 1000 | Ultra Disk | 20000 | 600 | 0.10.2 |

For more information about Message Queue for Apache Kafka instance editions, see Instance specifications.

What's next

Purchase and deploy a Message Queue for Apache Kafka instance. For more information, see Access from a VPC.

7.3. Migrate topics 7.3.1. Migrate topic metadata from a usercreated Kafka cluster to Message Queue for Apache Kafka

This topic describes how to use a metadata migration tool to migrate topic metadata from a usercreated Kafka cluster to a Message Queue for Apache Kafka instance. The metadata of a topic is the basic information of the topic instead of the information stored in the topic.

Prerequisites

Before you migrate topic metadata from a user-created Kafka cluster to Message Queue for Apache

Kafka, make sure that you have completed the following steps:

- Download Java Development Kit (JDK) 8
- Download the migration tool Java Archive (JAR) file

✓ Notice

- After the migration, the corresponding topic metadata in the source user-created Kafka cluster is not deleted. Instead, a new topic with the same configuration is created in the destination Message Queue for Apache Kafka instance.
- Only the topic configuration is migrated, whereas the messages in the topic are not migrated.

Procedure

- 1. Start the command-line tool.
- 2. Run the **cd** command to switch to the directory where the migration tool is located.
- 3. Run the following command to confirm the topic metadata to be migrated: iava -iar kafka-migration n.iar TopicMigrationFromZk --sourceZkConnect 192.168.XX.XX --destAk <vourdestAccessKevId> --destSk <y ourdestAccessKeySecret> --destRegionId <yourdestRegionId> --destInstanceId <yourdestInstanceId>

| Parameter | Description |
|-----------------|---|
| sourceZkConnect | The IP address of the source user-created ZooKeeper cluster. |
| destAk | The AccessKey ID of the Alibaba Cloud account to which the destination Message Queue for Apache Kafka instance belongs. |
| destSk | The AccessKey secret of the Alibaba Cloud account to which the destination Message Queue for Apache Kafka instance belongs. |
| destRegionId | The ID of the region where the destination Message Queue for Apache Kafka instance is located. |
| destInstanceId | The ID of the destination Message Queue for Apache Kafka instance. |

The following code provides an example of the output to be confirmed:

13:40:08 INFO - Begin to migrate topics:[test]
13:40:08 INFO - Total topic number:1
13:40:08 INFO - Will create topic:test, isCompactTopic:false, partition number:1

4. Run the following command to commit the topic metadata to be migrated: iava -iar kafka-migratio n.iar TopicMigrationFromZk --sourceZkConnect 192.168.XX.XX --destAk <vourAccessKevId> --destSk <yourA ccessKeySecret> --destRegionId <yourRegionID> --destInstanceId <yourInstanceId> --commit

| Parameter Description |
|-----------------------|
|-----------------------|

| Parameter | Description |
|-----------|--|
| commit | Commits the topic metadata to be migrated. |

The following code provides an example of the output after the preceding commit:

```
13:51:12 INFO - Begin to migrate topics:[test]
13:51:12 INFO - Total topic number:1
13:51:13 INFO - cmd=TopicMigrationFromZk, request=null, response={"code":200,"requestId":"7F76C7
D7-AAB5-4E29-B49B-CD6F1E0F508B","success":true,"message":"operation success"}
13:51:13 INFO - TopicCreate success, topic=test, partition number=1, isCompactTopic=false
```

- 5. Check whether the topic metadata is migrated.
 - i. Log on to the Message Queue for Apache Kafka console.
 - ii. In the top navigation bar, select the region where the destination instance is located.
 - iii. In the left-side navigation pane, click **Topics**.
 - iv. On the **Topics** page, click the destination instance. The migrated topic appears on the **Topics** page.

7.3.2. Migrate topic metadata between Message Queue for Apache Kafka instances

This topic describes how to use a metadata migration tool to migrate topic metadata from a Message Queue for Apache Kafka instance to another Message Queue for Apache Kafka instance. The metadata of a topic is the basic information of the topic instead of the information stored in the topic.

Prerequisites

Before you migrate topic metadata between Message Queue for Apache Kafka instances, make sure that you have completed the following steps:

- Download Java Development Kit (JDK) 8
- Download the migration tool Java Archive (JAR) file

♥ Notice

- After the migration, the corresponding topic in the source Message Queue for Apache Kafka is not deleted. Instead, a new topic with the same configuration is created in the destination Message Queue for Apache Kafka instance.
- Only the topic configuration is migrated, whereas the messages in the topic are not migrated.

Procedure

- 1. Start the command-line tool.
- 2. Run the **cd** command to switch to the directory where the migration tool is located.
- 3. Run the following command to confirm the topic metadata to be migrated: iava -iar kafka-migratio n.jar TopicMigrationFromAliyun --sourceAk <yoursourceAccessKeyId> --sourceSk <yoursourceAccessKeySe

cret> --sourceRegionId <voursourceRegionId> --sourceInstanceId <voursourceInstanceId> --destAk <vourd estAccessKevId> --destSk <yourdestAccessKeySecret> --destRegionId <yourdestRegionId> --destInstanceId <yourdestInstanceId>

| Parameter | Description |
|------------------|---|
| sourceAk | The AccessKey ID of the Alibaba Cloud account to which the source Message Queue for Apache Kafka instance belongs. |
| sourceSk | The AccessKey secret of the Alibaba Cloud account to which the source Message Queue for Apache Kafka instance belongs. |
| sourceRegionId | The ID of the region where the source Message Queue for Apache Kafka instance is located. |
| sourceInstanceId | The ID of the source Message Queue for Apache Kafka instance. |
| destAk | The AccessKey ID of the Alibaba Cloud account to which the destination Message Queue for Apache Kafka instance belongs. |
| destSk | The AccessKey secret of the Alibaba Cloud account to which the destination Message Queue for Apache Kafka instance belongs. |
| destRegionId | The ID of the region where the destination Message Queue for Apache Kafka instance is located. |
| destInstanceId | The ID of the destination Message Queue for Apache Kafka instance. |

The following code provides an example of the output to be confirmed:

15:13:12 INFO - cmd=TopicMigrationFromAliyun, request=null, response={"total":4,"code":200,"reques tId":"1CBAB340-2146-43A3-8470-84D77DB8B43E","success":true,"pageSize":10000,"currentPage":1,"m essage":"operation success.","topicList":[{"instanceld":"alikafka_pre-cn-0pp1cng20***","localTopic":f alse,"createTime":1578558314000,"regionId":"cn-hangzhou","statusName":"Running","topic":"agdaga sdg","remark":"agdadgdasg","partitionNum":12,"compactTopic":false,"status":0,"tags":[]},{"instancel d":"alikafka_pre-cn-0pp1cng20***","localTopic":false,"createTime":1578558294000,"regionId":"cn-han gzhou","statusName":"Running","topic":"135215","remark":"1315215","partitionNum":12,"compactT opic":false,"status":0,"tags":[]},{"instanceld":"alikafka_pre-cn-0pp1cng20***","localTopic":false,"creat eTime":1578558214000,"regionId":"cn-hangzhou","statusName":"Running","topic"::false,"creat eTime":1578558214000,"regionId":"cn-hangzhou","statusName":"Running","topic"::false,"creat eTime":1578558214000,"regionId":"cn-hangzhou","statusName":"Running","topic"::false,"creat eTime":1578558214000,"regionId":"cn-hangzhou","statusName":"Running","topic"::false,"creat eTime":1578558214000,"regionId":"cn-hangzhou","statusName":"Running","topic"::false,"createTime":1578558141000,"regionId":"cn-hangzhou","statusName":" 13414","partitionNum":12,"compactTopic":false,"status":0,"tags":[]},{"instanceId":"alikafka_pre-cn-0p p1cng20***","localTopic":false,"createTime":1578558141000,"regionId":"cn-hangzhou","statusName": "Running","topic":"aete","remark":"est","partitionNum":12,"compactTopic":false,"status":0,"tags":[]}

15:13:12 INFO - Will create topic:agdagasdg, isCompactTopic:false, partition number:12 15:13:12 INFO - Will create topic:135215, isCompactTopic:false, partition number:12 15:13:12 INFO - Will create topic:1332, isCompactTopic:false, partition number:12 15:13:12 INFO - Will create topic:aete, isCompactTopic:false, partition number:12 4. Run the following command to commit the topic metadata to be migrated: iava -iar kafka-migratio n.iar TopicMigrationFromAlivun --sourceAk <voursourceAccessKevId> --sourceSk <voursourceAccessKevSe cret> --sourceRegionId <voursourceRegionId> --sourceInstanceId <voursourceInstanceId> --destAk <vourd estAccessKevId> --destSk <vourdestAccessKeySecret> --destRegionId <yourdestRegionId> --destInstanceId <yourdestInstanceId> --commit

| Parameter | Description |
|-----------|--|
| commit | Commits the topic metadata to be migrated. |

The following code provides an example of the output after the preceding commit:

16:38:30 INFO - cmd=TopicMigrationFromAliyun, request=null, response={"code":200,"requestId":"A0C A4D70-46D4-45CF-B9E0-B117610A26DB","success":true,"message":"operation success"} 16:38:30 INFO - TopicCreate success, topic=agdagasdg, partition number=12, isCompactTopic=false 16:38:36 INFO - cmd=TopicMigrationFromAliyun, request=null, response={"code":200,"requestId":"05E 88C75-64B6-4C87-B962-A63D906FD993","success":true,"message":"operation success"} 16:38:36 INFO - TopicCreate success, topic=135215, partition number=12, isCompactTopic=false 16:38:42 INFO - TopicCreate success, topic=135215, partition number=12, isCompactTopic=false 16:38:42 INFO - cmd=TopicMigrationFromAliyun, request=null, response={"code":200,"requestId":"9D5 4F6DB-6FA0-4F6D-B19A-09109F70BDDA","success":true,"message":"operation success"} 16:38:42 INFO - TopicCreate success, topic=1332, partition number=12, isCompactTopic=false 16:38:49 INFO - cmd=TopicMigrationFromAliyun, request=null, response={"code":200,"requestId":"6C2 65013-D15E-49AE-BE55-BF7657ADA1B7","success":true,"message":"operation success"} 16:38:49 INFO - TopicCreate success, topic=aete, partition number=12, isCompactTopic=false

- 5. Check whether the topic metadata is migrated.
 - i. Log on to the Message Queue for Apache Kafka console.
 - ii. In the top navigation bar, select the region where the destination instance is located.
 - iii. In the left-side navigation pane, click **Topics**.
 - iv. On the **Topics** page, click the destination instance. The migrated topic appears on the **Topics** page.

7.4. Migrate consumer groups

7.4.1. Migrate consumer group metadata from a user-created Kafka cluster to Message Queue for Apache Kafka

This topic describes how to use a metadata migration tool to migrate consumer group metadata from a user-created Kafka cluster to a Message Queue for Apache Kafka instance.

Prerequisites

Before you migrate consumer group metadata from a user-created Kafka cluster to Message Queue for Apache Kafka, make sure that you have completed the following steps:

• Download Java Development Kit (JDK) 8

• Download the migration tool Java Archive (JAR) file

✓ Notice

- After the migration, the corresponding consumer group in the source Kafka cluster is not deleted. Instead, a new consumer group with the same configuration is created in the destination Message Queue for Apache Kafka instance.
- Only the consumer group configuration is migrated. Topics and consumer offsets are not migrated.

Procedure

- 1. Start the command-line tool.
- 2. Run the cd command to switch to the directory where the migration tool is located.
- 3. Create a configuration file named *kafka.properties*. The *kafka.properties* file is used to create a Kafka consumer to retrieve consumer offset information from the user-created Kafka cluster. The configuration file contains the following content:

| <pre>## The endpoint. bootstrap.servers=localhost:9092 ## The consumer group, which contains no consumer offset information so that consumption starts fro m the first message. group.id=XXX ## You can skip the following configuration if security configuration is unavailable. ## The simple authentication and security layer (SASL)-based authentication.</pre> |
|---|
| <pre>#sasl.mechanism=PLAIN ## The access protocol. #security.protocol=SASL_SSL ## The path of the Secure Sockets Layer (SSL) root certificate. #ssl.truststore.location=/Users/***/Documents/code/aliware-kafka-demos/main/resources/kafka.client .truststore.jks ## The SSL password.</pre> |
| #ssl.truststore.password=*** ## The SASL path. #java.security.auth.login.config=/Users/***/kafka-java-demo/vpc-ssl/src/main/resources/kafka_client_j aas.conf |

4. Run the following command to confirm the consumer group metadata to be migrated: iava -iar kaf ka-migration.iar ConsumerGroupMigrationFromTopic --propertiesPath /usr/local/kafka_2.12-2.4.0/config/k afka.properties --destAk <vourAccessKeyId> --destSk <yourAccessKeySecret> --destRegionId <yourRegionI d> --destInstanceId <yourInstanceId>

| Parameter | Description |
|----------------|---|
| propertiesPath | The path of the <i>kafka.properties</i> configuration file. |
| destAk | The AccessKey ID of the Alibaba Cloud account to which the destination Message Queue for Apache Kafka instance belongs. |

| Parameter | Description |
|----------------|---|
| destSk | The AccessKey secret of the Alibaba Cloud account to which the destination Message Queue for Apache Kafka instance belongs. |
| destRegionId | The ID of the region where the destination Message Queue for Apache Kafka instance is located. |
| destInstanceId | The ID of the destination Message Queue for Apache Kafka instance. |

The following code provides an example of the output to be confirmed:

15:29:45 INFO - Will create consumer groups:[XXX, test-consumer-group]

5. Run the following command to commit the consumer group metadata to be migrated: iava -iar kaf ka-migration.iar ConsumerGroupMigrationFromTopic --propertiesPath /usr/local/kafka_2.12-2.4.0/config/k afka.properties --destAk LTAI4FwO5aK1mFYCspJ1h*** --destSk wvDxiiRO1tHPiL0oj7Y2Z7WDNkS*** --dest RegionId cn-hangzhou --destInstanceId alikafka_pre-cn-v0h1cng00*** --commit

| Parameter | Description |
|-----------|---|
| commit | Commits the consumer group metadata to be migrated. |

The following code provides an example of the output after the preceding commit:

15:35:51 INFO - cmd=ConsumerGroupMigrationFromTopic, request=null, response={"code":200,"reque stld":"C9797848-FD4C-411F-966D-0D4AB5D12F55","success":true,"message":"operation success"} 15:35:51 INFO - ConsumerCreate success, consumer group=XXX 15:35:57 INFO - cmd=ConsumerGroupMigrationFromTopic, request=null, response={"code":200,"reque stld":"3BCFDBF2-3CD9-4D48-92C3-385C8DBB9709","success":true,"message":"operation success"} 15:35:57 INFO - ConsumerCreate success, consumer group=test-consumer-group

- 6. Check whether the consumer group metadata is migrated.
 - i. Log on to the Message Queue for Apache Kafka console.
 - ii. In the top navigation bar, select the region where the destination instance is located.
 - iii. In the left-side navigation pane, click **Consumer Groups**.
 - iv. On the **Consumer Groups** page, click the destination instance. The migrated consumer group appears on the **Consumer Group** page.

7.4.2. Migrate consumer group metadata between Message Queue for Apache Kafka instances

This topic describes how to use a metadata migration tool to migrate consumer group metadata from one Message Queue for Apache Kafka instance to another Message Queue for Apache Kafka instance.

Prerequisites

Before you migrate consumer group metadata between Message Queue for Apache Kafka instances, make sure that you have completed the following steps:

- Download Java Development Kit (JDK) 8
- Download the migration tool Java Archive (JAR) file

♥ Notice

- After the migration, the corresponding consumer group in the source Message Queue for Apache Kafka instance is not deleted. Instead, a new consumer group with the same configuration is created in the destination Message Queue for Apache Kafka instance.
- Only the consumer group configuration is migrated. Topics and consumer offsets in the consumer group are not migrated.

Procedure

- 1. Start the command-line tool.
- 2. Run the **cd** command to switch to the directory where the migration tool is located.
- 3. Run the following command to confirm the consumer group metadata to be migrated: iava -iar kaf ka-migration.iar ConsumerGroupMigrationFromAlivun --sourceAk <voursourceAccessKevId> --sourceSk <v oursourceAccessKevSecret> --sourceRegionId <voursourceRegionId> --sourceInstanceId <voursourceInstanceId <voursourceInstanceId <voursourceSk <v oursourceAccessKevId> --destSk <vourdestAccessKevSecret> --destRegionId <voursourceRegionId> --destRegionId <voursourceInstanceId <voursourceSk <v ourdestAccessKevId> --destSk <vourdestAccessKevSecret> --destRegionId <voursourceSk <v ourdestRegionId> --destInstanceId <voursourceSk <v ourdestInstanceId>

| Parameter | Description |
|------------------|---|
| sourceAk | The AccessKey ID of the Alibaba Cloud account to which the source Message Queue for Apache Kafka instance belongs. |
| sourceSk | The AccessKey secret of the Alibaba Cloud account to which the source Message Queue for Apache Kafka instance belongs. |
| sourceRegionId | The ID of the region where the source Message Queue for Apache Kafka instance is located. |
| sourceInstanceId | The ID of the source Message Queue for Apache Kafka instance. |
| destAk | The AccessKey ID of the Alibaba Cloud account to which the destination Message Queue for Apache Kafka instance belongs. |
| destSk | The AccessKey secret of the Alibaba Cloud account to which the destination Message Queue for Apache Kafka instance belongs. |
| destRegionId | The ID of the region where the destination Message Queue for Apache Kafka instance is located. |

| Parameter | Description |
|----------------|--|
| destInstanceId | The ID of the destination Message Queue for Apache Kafka instance. |

The following code provides an example of the output to be confirmed:

10:54:26 INFO - cmd=ConsumerGroupMigrationFromAliyun, request=null, response={"code":200,"reque stId":"9793DADB-55A5-4D4E-9E9C-D4DA8B35370C","success":true,"consumerList":[{"instanceId":"alika fka_post-cn-0pp1h0uv6***","regionId":"cn-hangzhou","consumerId":"Demo","tags":[{"value":"","key" :"migration"}]],"message":"operation success."} 10:54:26 INFO - Will create consumer groups:[Demo]

4. Run the following command to commit the consumer group metadata to be migrated: iava -iar kaf ka-migration.iar ConsumerGroupMigrationFromAlivun --sourceAkLTAI4FwO5aK1mFYCspJ1h*** --sourceS k wvDxiiRO1tHPiL0oi7Y2Z7WDNkS*** --sourceRegionId cn-hangzhou --sourceInstanceId alikafka_post-cn-0 pp1h0uv6*** --destAkLTAI4FwO5aK1mFYCspJ1h*** --destSk wvDxiiRQ1tHPiL0oj7Y2Z7*** --destRegionId c n-hangzhou --destInstanceId alikafka_pre-cn-v0h1cng00*** --commit

| Parameter | Description |
|-----------|---|
| commit | Commits the consumer group metadata to be migrated. |

The following code provides an example of the output after the preceding commit:

10:54:40 INFO - cmd=ConsumerGroupMigrationFromAliyun, request=null, response={"code":200,"reque stld":"49E53B79-3C2C-4BCF-8BC8-07B0BB14B52A","success":true,"consumerList":[{"instanceId":"alikaf ka_post-cn-0pp1h0uv6***","regionId":"cn-hangzhou","consumerId":"Demo","tags":[{"value":"","key": "migration"}]]],"message":"operation success."} 10:54:41 INFO - cmd=ConsumerGroupMigrationFromAliyun, request=null, response={"code":200,"reque stld":"5AEEFB13-2A6B-4265-97CB-902CFA483339","success":true,"message":"operation success"} 10:54:41 INFO - ConsumerCreate success, consumer group=Demo

- 5. Check whether the consumer group metadata is migrated.
 - i. Log on to the Message Queue for Apache Kafka console.
 - ii. In the top navigation bar, select the region where the destination instance is located.
 - iii. In the left-side navigation pane, click **Consumer Groups**.
 - iv. On the **Consumer Groups** page, click the destination instance. The created consumer group appears on the **Consumer Group** page.

7.5. Migrate data

7.5.1. Migrate data to the cloud

This topic describes how to use MirrorMaker to migrate data in a user-created Kafka cluster to a Message Queue for Apache Kafka instance.

Prerequisites

The following operations are completed:

- Download MirrorMaker.
- Migrate topic metadata from a user-created Kafka cluster to Message Queue for Apache Kafka.

Context

Kafka mirroring can be used to back up data in Kafka clusters. MirrorMaker is the tool to implement this feature. You can use MirrorMaker to mirror the source user-created Kafka cluster to the destination cluster. The destination cluster is a Message Queue for Apache Kafka instance, as shown in the following figure. MirrorMaker uses a built-in consumer to consume messages from the user-created Kafka cluster and then uses a built-in producer to send these messages to the Message Queue for Apache Kafka instance.



Fore more information, see Apache Kafka MirrorMaker.

Precautions

- The topic names of the source and destination clusters must be consistent.
- The numbers of partitions of the source and destination clusters can be different.
- Data in the same partition may not be distributed to the same partition.
- By default, messages with the same key are distributed to the same partition.
- Normal messages may be out of order when the instance fails, whereas partitionally ordered messages remain in order when the instance fails.

Access from a VPC

1. Configure the *consumer.properties* file.

The endpoint of the user-created Kafka cluster. bootstrap.servers=XXX.XXX.XXX.9092 ## The consumer policy for distributing messages to partitions. partition.assignment.strategy=org.apache.kafka.clients.consumer.RoundRobinAssignor ## The name of the consumer group. group.id=test-consumer-group

2. Configure the *producer.properties* file.

The default endpoint of the Message Queue for Apache Kafka instance, which can be obtained in the Message Queue for Apache Kafka console. bootstrap.servers=XXX.XXX.XXX.3092 ## The data compression method. compression.type=none 3. Run the following command to start the migration process:

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

Access from the Internet

- 1. Download kafka.client.truststore.jks.
- 2. Configure the *kafka_client_jaas.conf* file.

```
KafkaClient {
    org.apache.kafka.common.security.plain.PlainLoginModule required
    username="your username"
    password="your password";
};
```

3. Configure the *consumer.properties* file.

```
## The endpoint of the user-created Kafka cluster.
bootstrap.servers=XXX.XXX.XXX.XXX:9092
## The consumer policy for distributing messages to partitions.
partition.assignment.strategy=org.apache.kafka.clients.consumer.RoundRobinAssignor
## The name of the consumer group.
group.id=test-consumer-group
```

4. Configure the *producer.properties* file.

The SSL endpoint of the Message Queue for Apache Kafka instance, which can be obtained in the Me
ssage Queue for Apache Kafka console.
bootstrap.servers=XXX.XXX.XXX.XXX.9093
The data compression method.
compression.type=none
The truststore (Use the file downloaded in Step 1).
ssl.truststore.location=kafka.client.truststore.jks
ssl.truststore.password=KafkaOnsClient
security.protocol=SASL_SSL
sasl.mechanism=PLAIN
If you use Message Queue for Apache Kafka 2.x with Simple Authentication and Security Layer (SASL)
authentication, you must configure the following parameter. This parameter is not required for Messag
e Queue for Apache Kafka earlier than v2.x.
ssl.endpoint.identification.algorithm=

5. Configure the *java.security.auth.login.config* file.

export KAFKA_OPTS="-Djava.security.auth.login.config=kafka_client_jaas.conf"

6. Run the following command to start the migration process:

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

Verify the result

You can check whet her MirrorMaker runs by using one of the following methods:

• Run kafka-consumer-groups.sh to view the consumption progress of the user-created cluster.

bin/kafka-consumer-groups.sh --new-consumer --describe --bootstrap-server endpoint of the user-created cluster --group test-consumer-group

• Send messages to the user-created cluster. In the Message Queue for Apache Kafka console, check the partition status of the topic, and check whether the total number of messages in the current broker is correct. You can view the specific content of a message in the Message Queue for Apache Kafka console. For more information, see Query messages.

7.6. View the migration progress

This topic describes how to view the progress of migrating data from a user-created Kafka cluster to Message Queue for Apache Kafka .

Prerequisites

The following operations are completed:

- 1. Purchase and deploy a Message Queue for Apache Kafka instance.
- 2. Start to migrate data from a user-created Kafka cluster to a Message Queue for Apache Kafka instance. The following migration types are supported:
 - Metadata migration
 - Migrate topic metadata from a user-created Kafka cluster to Message Queue for Apache Kafka
 - Migrate consumer group metadata from a user-created Kafka cluster to Message Queue for Apache Kafka
 - Data migration

Migrate data to the cloud

• Consumer offset migration

Notice You cannot migrate the consumer offset of a user-created Kafka cluster to Message Queue for Apache Kafka .

View the migration progress

Notice Message Queue for Apache Kafka does not support reporting information about data migration and consumer offset migration. If you have started data migration or consumer offset migration, you cannot view the migration progress in the Message Queue for Apache Kafka console.

To view the migration progress, perform the following steps:

- 1. Log on to the Message Queue for Apache Kafka console.
- 2. In the top navigation bar, select the region where your instance is located.
- 3. In the left-side navigation pane, click Migration.
- 4. Click the Migration tab.

All cloud migration tasks of the region appear on the Migration tab.

5. (Optional)Find the migration task and click **Details**.

8.Query messages

If an error occurs in message consumption, you can troubleshoot the error by querying the messages. In the Message Queue for Apache Kafka console, you can query messages by offset and by time.

Background information

You can use one of the following methods to query messages:

- If you can query logs to obtain the ID of the partition in the topic to which a message is sent and the message offset, we recommend that you query the message by offset. For more information, see Query messages by offset.
- If you do not know the message offset but know when the message is sent, we recommend that you query the message by time. For more information, see Query messages by time.

Precautions

• The Message Queue for Apache Kafka console displays a maximum of 1 KB of content for each queried message. The excess content of the message is omitted. If you need to view the complete message, download the message.

You can download messages only from Message Queue for Apache Kafka instances of Professional Edition. You can download up to 10 MB of messages at a time.

- If you use an instance of Standard Edition, you can query a maximum of 10 messages of up to 256 KB in total size at a time.
 - If the total size of the 10 queried messages exceeds 256 KB, only the first 256 KB of message content is displayed in the console.
 - If the total size of the 10 queried messages is less than 256 KB, the message content is completely displayed. However, you can view only up to 10 messages. In this case, check the actual consumption data of the consumer.
- If you use an instance of Professional Edition, you can query a maximum of 30 messages of up to 10 MB in total size at a time.
 - If the total size of the 30 queried messages exceeds 10 MB, only the first 10 MB of message content is displayed in the console.
 - If the total size of the 30 queried messages is less than 10 MB, the message content is completely displayed. However, you can view only up to 30 messages. In this case, check the actual consumption data of the consumer.

For more information about instance editions, see Billing.

- The query results are also related to the following message deletion policies of Message Queue for Apache Kafka :
 - If the disk usage is lower than 85%, messages whose retention period expires are deleted at 04:00:00 every day.
 - If the disk usage reaches 85%, messages whose retention period expires are immediately deleted.
 - If the disk usage reaches 90%, messages are deleted from the earliest one stored in the Message Queue for Apache Kafka broker, no matter whether their retention period expires.

Notice At least one storage file is retained by Message Queue for Apache Kafka when the messages are deleted. Therefore, when you query messages, the query results may contain messages whose retention period expires.

Query messages by offset

In Message Queue for Apache Kafka , each offset maps a message. If you know the location of a message, you can specify an offset to query the message.

- 1. Log on to the Message Queue for Apache Kafka console.
- 2. In the top navigation bar, select the region where your instance is located.
- 3. In the left-side navigation pane, click Instances.
- 4. On the **Instances** page, click the name of the instance that you want to manage.
- 5. In the left-side navigation pane, click Message Query.
- 6. On the Message Query page, click the Query by Offset tab.
- 7. Enter the name of the topic, select the partition, and enter the offset from which you want to query messages. Then, click **Search**.

On the **Query by Offset** tab, messages that start from the specified offset are displayed. For example, if the specified partition and offset are both 5, the returned results are messages that start from Offset 5 in Partition 5.

| Parameter | Description |
|-----------|--|
| Partition | The ID of the partition in the topic to which the message is sent. |
| Offset | The consumer offset of the message. |

| Parameter | Description |
|-----------|---|
| | The timestamp of the producer when the message is sent or the value of the timestamp field that you specify in the ProducerRecord parameter. |
| Time | Note If a value is specified for the timestamp field, the value is displayed. If no value is specified for the timestamp field, the system time when the message is sent is displayed. A value in the format of 1970/x/x x:x:x indicates that the timestamp field is set to 0 or an invalid value. You cannot specify a value for the timestamp field on clients of Message Queue for Apache Kafka version 0.9 and earlier. |
| | |

8. (Optional)Click a message to view its details.

| Parameter | Description |
|-----------|---|
| Кеу | The key of the queried message. The key is converted to a string. |
| Value | The content of the queried message. The content is converted to a string. |

9. (Optional)Click **Download Message** next to the Key or Value parameter to download the message.

♥ Notice

- This operation is supported only for instances of Professional Edition.
- You can download up to 10 MB of messages at a time. If the total size of the queried messages exceeds 10 MB, only the first 10 MB of message content can be downloaded.

Query messages by time

You can query messages in all partitions by time. If you do not know the message offset but know the time range in which the messages are sent, you can query the messages by time. You can specify a point in time in the time range to query messages that are sent near this point.

1. Log on to the Message Queue for Apache Kafka console.

- 2. In the top navigation bar, select the region where your instance is located.
- 3. In the left-side navigation pane, click Instances.
- 4. On the **Instances** page, click the name of the instance that you want to manage.
- 5. In the left-side navigation pane, click Message Query.
- 6. On the **Message Query** page, click the **Query by Time** tab.
- 7. Enter the name of the topic, select the partition, and specify a point in time from which you want to query messages. Then, click **Search**.

On the **Query by Time** tab, the query results are displayed. The query results may vary based on the partition that you specify.

- If you select All, the messages in all partitions are displayed.
- If you specify a partition, the messages in the specified partition are displayed.

| Parameter | Description |
|-----------|---|
| Partition | The ID of the partition in the topic to which the message is sent. |
| Offset | The consumer offset of the message. |
| | The timestamp of the producer when the message is sent or the value of the timestamp field that you specify in the ProducerRecord parameter. |
| Time | Note If a value is specified for the timestamp field, the value is displayed. If no value is specified for the timestamp field, the system time when the message is sent is displayed. A value in the format of 1970/x/x x:x:x indicates that the timestamp field is set to 0 or an invalid value. You cannot specify a value for the timestamp field on clients of Message Queue for Apache Kafka version 0.9 and earlier. |
| | |

8. (Optional)Click a message to view its details.

| Parameter | Description |
|-----------|---|
| Кеу | The key of the queried message. The key is converted to a string. |

| Parameter | Description |
|-----------|---|
| Value | The content of the queried message. The content is converted to a string. |

9. (Optional)Click **Download Message** next to the Key or Value parameter to download the message.

♦ Notice

- This operation is supported only for instances of Professional Edition.
- You can download up to 10 MB of messages at a time. If the total size of the queried messages exceeds 10 MB, only the first 10 MB of message content can be downloaded.