Distributed Message Service for Kafka

User Guide

Issue 01

Date 2024-12-02

HUAWEI CLOUD COMPUTING TECHNOLOGIES CO., LTD.

Trademarks and Permissions

Notice

The purchased products, services and features are stipulated by the contract made between Huawei Cloud and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

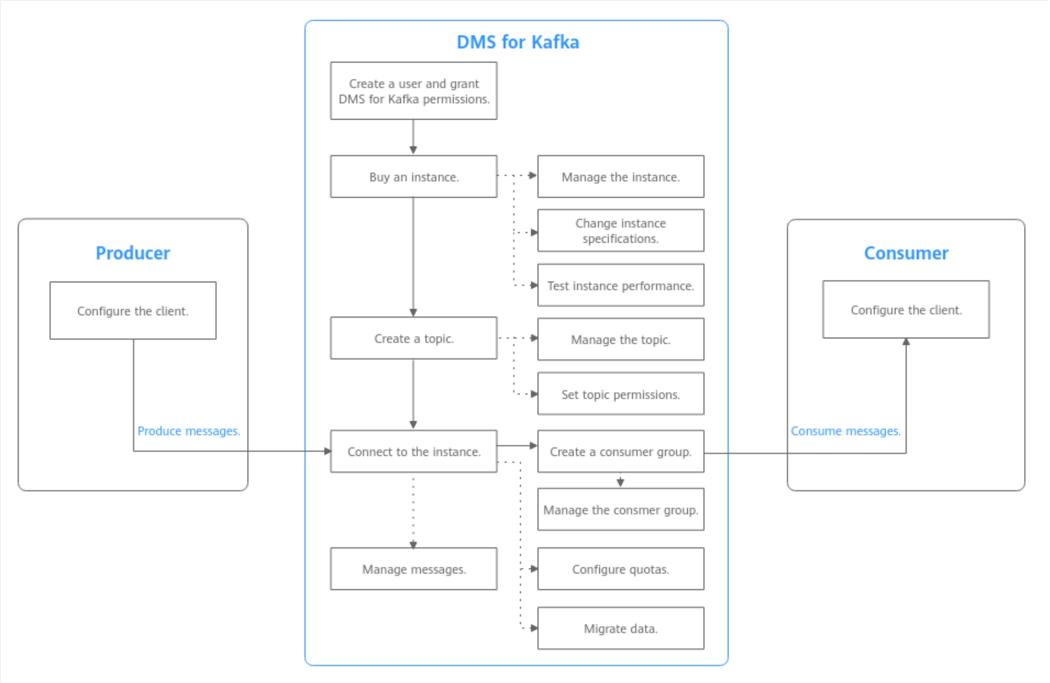
The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

Contents

1 Process of Using Kafka

Distributed Message Service for Kafka is a message queuing service that is based on the open-source Apache Kafka. It provides Kafka instances with isolated computing, storage, and bandwidth resources. The following figure shows the process of message production and consumption using a Kafka instance.

Figure 1-1 Process of using Kafka



1. **Creating an IAM User and Granting DMS for Kafka Permissions**

   Create IAM users and grant them only the DMS for Kafka permissions required to perform a given task based on their job responsibilities.

2. **Buying a Kafka Instance**

   Kafka instances are tenant-exclusive, and physically isolated in deployment.

3. **Creating a Kafka Topic**

   Create a topic for storing messages so that producers can produce messages and consumers can subscribe to messages.

4. **Connecting to an Instance**

   The client uses commands to connect to Kafka instances in a private or public network, and produces and consumes messages.

2 Permissions Management

2.1 Creating an IAM User and Granting DMS for Kafka Permissions

This section describes how to use **Identity and Access Management (IAM)** for fine-grained permissions control for your Distributed Message Service (DMS) for Kafka resources. With IAM, you can:

- Create IAM users for personnel based on your enterprise's organizational structure. Each IAM user has their own identity credentials for accessing DMS for Kafka resources.
- Grant users only the permissions required to perform a given task based on their job responsibilities.
- Entrust another HUAWEI ID or cloud service to perform efficient O&M on your DMS for Kafka resources.

If your HUAWEI ID meets your permissions requirements, you can skip this section.

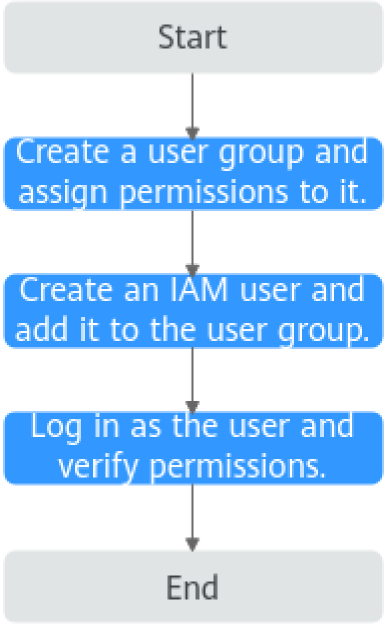
This section describes the procedure for granting permissions (see **Figure 2-1**).

Prerequisites

Learn about the permissions (see **System-defined roles and policies supported by DMS for Kafka**) supported by DMS for Kafka and choose policies according to your requirements. For the permissions of other services, see **System Permissions**.

Process Flow

Figure 2-1 Process for granting DMS for Kafka permissions


Start
Create a user group and assign permissions to it.
Create an IAM user and add it to the user group.
Log in as the user and verify permissions.
End

1. For the following example, **create a user group on the IAM console** and assign the **DMS ReadOnlyAccess** policy to the group.
2. **Create an IAM user and add it to the created user group**.
3. **Log in as the IAM user** and verify permissions.

   In the authorized region, perform the following operations:

   - Choose **Service List** > **Distributed Message Service (for Kafka)**. Then click **Buy Instance** on the console of DMS for Kafka. If a message appears indicating that you cannot perform the operation, the **DMS ReadOnlyAccess** policy is in effect.
   - Choose **Service List** > **Elastic Volume Service**. If a message appears indicating that you have insufficient permissions, the **DMS ReadOnlyAccess** policy is in effect.
   - Choose **Service List** > Distributed Message Service for Kafka. If the Kafka instance list can be displayed, the **DMS ReadOnlyAccess** policy is in effect.

Example Custom Policies

You can create custom policies to supplement the system-defined policies of DMS for Kafka. For details about actions supported in custom policies, see **Permissions Policies and Supported Actions**

To create a custom policy, choose either visual editor or JSON.

- Visual editor: Select cloud services, actions, resources, and request conditions. This does not require knowledge of policy syntax.
- JSON: Create a JSON policy or edit an existing one.

For details, see **Creating a Custom Policy**. The following lists examples of common DMS for Kafka custom policies.

NOTE

- DMS for Kafka permissions policies are based on DMS. Therefore, when assigning permissions, select DMS permissions policies.
- Due to data caching, a policy involving Object Storage Service (OBS) actions will take effect five minutes after it is attached to a user, user group, or project.

- Example 1: Grant permission to delete and restart instances.

```
{
    "Version": "1.1",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "dms:instance:modifyStatus",
                "dms:instance:delete"
            ]
        }
    ]
}
```

- Example 2: Grant permission to deny instance deletion.

  A policy with only "Deny" permissions must be used together with other policies. If the permissions granted to an IAM user contain both "Allow" and "Deny", the "Deny" permissions take precedence over the "Allow" permissions.

  Assume that you want to grant the permissions of the **DMS FullAccess** policy to a user but want to prevent them from deleting instances. You can create a custom policy for denying instance deletion, and attach this policy together with the **DMS FullAccess** policy to the user. As an explicit deny in any policy overrides any allows, the user can perform all operations on DMS for Kafka excepting deleting instances.

  Example policy denying instance deletion:

```
{
        "Version": "1.1",
        "Statement": [
                {
                        "Effect": "Deny",
                        "Action": [
                                "dms:instance:delete"
                        ]
                }
        ]
}
```

DMS for Kafka Resources

A resource is an object that exists within a service. DMS for Kafka resources include **kafka**. To select these resources, specify their paths.

Table 2-1 DMS for Kafka resources and their paths

Resource	Resource Name	Path
kafka	Instance	[Format] DMS:*:*: kafka:*instance ID* [Notes] For instance resources, IAM automatically generates the prefix (**DMS:*:*:kafka:**) of the resource path. For the path of a specific resource, add the *instance ID* to the end. You can also use an asterisk ***** to indicate any resource. For example: **DMS:*:*:kafka:*** indicates any Kafka instance.

DMS for Kafka Request Conditions

Request conditions are useful in determining when a custom policy is in effect. A request condition consists of condition keys and operators. Condition keys are either global or service-level and are used in the Condition element of a policy statement. **Global condition keys** (starting with **g:**) are available for operations of all services, while service-specific condition keys (starting with a service name such as **dms:**) are available only for operations of specific services. An operator must be used together with a condition key to form a complete condition statement.

DMS for Kafka has a group of predefined condition keys that can be used in IAM. For example, to define an "Allow" permission, use the condition dms:ssl to filter instances by SASL configurations. The following table lists the DMS for Kafka predefined condition keys.

Table 2-2 Predefined condition keys of DMS for Kafka

Condition Key	Operator	Description
dms:publicIP	Bool	Whether public access is enabled
dms:ssl	Bool	Whether SSL is enabled

3 Buying a Kafka Instance

Kafka instances are tenant-exclusive, and physically isolated in deployment. You can customize the computing capabilities and storage space of a Kafka instance as required.

Preparing Instance Dependencies

Before creating a Kafka instance, prepare the resources listed in **Table 3-1**.

Table 3-1 Kafka resources

Resource	Requirement	Operations
VPC and subnet	You need to configure a VPC and subnet for the Kafka instance as required. You can use the current account's existing VPC and subnet, or create new ones. Note: VPCs must be created in the same region as the Kafka instance.	For details on how to create a VPC and a subnet, see **Creating a VPC**. If you need to create and use a new subnet in an existing VPC, see **Creating a Subnet for the VPC**.
Security group	Different Kafka instances can use the same or different security groups. Before accessing a Kafka instance, configure security groups based on the access mode. For details, see **Table 5-2**.	For details on how to create a security group, see **Creating a Security Group**. For details on how to add rules to a security group, see **Adding a Security Group Rule**.

Resource	Requirement	Operations
EIP	To access a Kafka instance on a client over a public network, create EIPs in advance. Note the following when creating EIPs: ● The EIPs must be created in the same region as the Kafka instance. ● The number of EIPs must be the same as the number of Kafka instance brokers. ● **The Kafka console cannot identify IPv6 EIPs.**	For details about how to create an EIP, see **Assigning an EIP**.

Procedure

Step 1 Go to the **Buy Instance** page.

Step 2 Specify **Billing Mode**.

Step 3 Select a region.

DMS for Kafka instances in different regions cannot communicate with each other over an intranet. Select a nearest location for low latency and fast access.

Step 4 Select a **Project**.

Projects isolate compute, storage, and network resources across geographical regions. For each region, a preset project is available.

Step 5 Select an **AZ**.

An AZ is a physical region where resources use independent power supply and networks. AZs are physically isolated but interconnected through an internal network.

Select one, three, or more AZs as required. The AZs cannot be changed once the instance is created.

Step 6 Enter an **Instance Name**.

You can customize a name that complies with the rules: 4–64 characters; starts with a letter; can contain only letters, digits, hyphens (-), and underscores (_).

Step 7 Select an **Enterprise Project**.

This parameter is for enterprise users. An enterprise project manages cloud resources. The enterprise project management service unifies cloud resources in projects, and resources and members in a project. The default project is **default**.

Step 8 Configure the following instance specifications:

Specifications: Select **Default** or **Custom**.

- **Default**: Specify the version, broker flavor and quantity, disk type, and disk size to be supported by the Kafka instance as required.
  a. **Version**: 1.1.0, 2.3.0, 2.7, and 3.x are available. **The version cannot be changed once the instance is created.**
  b. **Broker Flavor**: Select a broker flavor that best fit your needs.

     Maximum number of partitions per broker x Number of brokers = Maximum number of partitions of an instance. If the total number of partitions of all topics exceeds the upper limit of partitions, topic creation fails.
  c. For **Brokers**, specify the broker quantity.
  d. **Storage space per broker**: Disk type and size for storing the instance data. **The disk type cannot be changed once the Kafka instance is created.**

     The storage space is consumed by message replicas, logs, and metadata. Specify the storage space based on the expected service message size, the number of replicas, and the reserved disk space. Each Kafka broker reserves 33 GB disk space for storing logs and metadata.

     Disks are formatted when an instance is created. As a result, the actual available disk space is 93% to 95% of the total disk space.
  e. **Capacity Threshold Policy**: Policy used when the disk usage reaches the threshold. The capacity threshold is 95%.
     - **Automatically delete**: Messages can be created and retrieved, but 10% of the earliest messages will be deleted to ensure sufficient disk space. This policy is suitable for scenarios where no service interruption can be tolerated. Data may be lost.
     - **Stop production**: New messages cannot be created, but existing messages can still be retrieved. This policy is suitable for scenarios where no data loss can be tolerated.
- **Custom**: The system calculates **Brokers** and **Storage Space per Broker**, and provides **Recommended Specifications** based on your specified parameters: **Peak Creation Traffic**, **Retrieval Traffic**, **Replicas per Topic**, **Total Partitions**, and **Messages Created During Retention Period**. **This function is unavailable for v3.x.**

Step 9 Configure the instance network parameters.

- Select a VPC and a subnet.

  A VPC provides an isolated virtual network for your Kafka instances. You can configure and manage the network as required.

  > NOTE
  >
  > After the Kafka instance is created, its VPC and subnet cannot be changed.
- Select a security group.

  A security group is a set of rules for accessing a Kafka instance. You can click **Manage Security Group** to view or create security groups on the network console.

  Before accessing a Kafka instance on the client, configure security group rules based on the access mode. For details about security group rules, see **Table 5-2**.

Step 10 Configure **Kafka SASL_SSL**.

This parameter indicates whether to enable SASL authentication when a client connects to the instance. If you enable **Kafka SASL_SSL**, data will be encrypted for transmission to enhance security.

This setting is enabled by default. **It cannot be changed after the instance is created.** If you want to use a different setting, you must create an instance.

After **Kafka SASL_SSL** is enabled, you can determine whether to enable **SASL/PLAIN**. If **SASL/PLAIN** is disabled, the SCRAM-SHA-512 mechanism is used to transmit data. If **SASL/PLAIN** is enabled, both the SCRAM-SHA-512 and PLAIN mechanisms are supported. You can select either of them as required. The **SASL/PLAIN** setting cannot be changed once the instance is created.

What are SCRAM-SHA-512 and PLAIN mechanisms?

- SCRAM-SHA-512: uses the hash algorithm to generate credentials for usernames and passwords to verify identities. SCRAM-SHA-512 is more secure than PLAIN.
- PLAIN: a simple username and password verification mechanism.

If you enable **Kafka SASL_SSL**, you must also set the username and password for accessing the instance.

Step 11 Click **Advanced Settings** to configure more parameters.

1. Configure public access.

   Public access is disabled by default. You can enable or disable it as required.

   After public access is enabled, configure an IPv4 EIP for each broker.

   After enabling **Public Access**, you can enable or disable **Intra-VPC Plaintext Access**. **If it is enabled, data will be transmitted in plaintext when you connect to the instance through a private network, regardless of whether SASL_SSL is enabled. This setting cannot be changed after the instance is created.** Exercise caution. If you want to use a different setting, you must create an instance.

2. Configure **Automatic Topic Creation**.

   This setting is disabled by default. You can configure it as required.

   If this option is enabled, a topic will be automatically created when a message is produced in or consumed from a topic that does not exist. The default topic parameters are listed in **Table 3-2**.

   For cluster instances, after you change the value of the **log.retention.hours** (retention period), **default.replication.factor** (replica quantity), or **num.partitions** (partition quantity) parameter, the value will be used in later topics that are automatically created. For example, assume that **num.partitions** is changed to 5, an automatically created topic has parameters listed in **Table 3-2**.

Table 3-2 Topic parameters

Parameter	Default Value	Modified Value
Partitions	3	5

Parameter	Default Value	Modified Value
Replicas	3	3
Aging Time (h)	72	72
Synchronous Replication	Disabled	Disabled
Synchronous Flushing	Disabled	Disabled
Message Timestamp	CreateTime	CreateTime
Max. Message Size (bytes)	10,485,760	10,485,760

3. Specify **Tags**.

   Tags are used to identify cloud resources. When you have multiple cloud resources of the same type, you can use tags to classify them based on usage, owner, or environment.

   – If you have predefined tags, select a predefined pair of tag key and value. You can click **View predefined tags** to go to the Tag Management Service (TMS) console and view or create tags.

   – You can also create new tags by specifying **Tag key** and **Tag value**.

   Up to 20 tags can be added to each Kafka instance. For details about the requirements on tags, see **Configuring Kafka Instance Tags**.

4. Enter a **Description** of the instance for 0–1024 characters.

Step 12 Click **Buy**.

Step 13 Confirm the instance information, read and agree to the *HUAWEI CLOUD Customer Agreement*, and click **Submit**.

Step 14 Return to the instance list and check whether the Kafka instance has been created.

It takes 3 to 15 minutes to create an instance. During this period, the instance status is **Creating**.

- If the instance is created successfully, its status changes to **Running**.
- If the instance is in the **Failed** state, delete it by referring to **Deleting Kafka Instances** and try creating another one. If the instance creation fails again, contact customer service.

  NOTE

  Instances that fail to be created do not occupy other resources.

----End

4 Configuring Topics

4.1 Creating a Kafka Topic

Topics store messages created by producers and subscribed by consumers. If **Automatic Topic Creation** is not enabled during Kafka instance creation, you need to manually create topics. If **Automatic Topic Creation** has been enabled for the instance, this operation is optional.

Automatic Topic Creation indicates that a topic will be automatically created when a message is produced in or consumed from a topic that does not exist. The default topic parameters are listed in **Table 4-1**.

After you change the value of the **log.retention.hours** (retention period), **default.replication.factor** (replica quantity), or **num.partitions** (partition quantity) parameter, the value will be used in later topics that are automatically created. For example, assume that **num.partitions** is changed to 5, an automatically created topic has parameters listed in **Table 4-1**.

Table 4-1 Topic parameters

Parameter	Default Value	Modified To
Partitions	3	5
Replicas	3	3
Aging Time (h)	72	72
Synchronous Replication	Disabled	Disabled
Synchronous Flushing	Disabled	Disabled
Message Timestamp	CreateTime	CreateTime
Max. Message Size (bytes)	10,485,760	10,485,760

Methods that can be used to manually create a topic:

- Creating a Topic on the Console
- Creating a Topic on Kafka Manager
- Creating a Topic on the Client

NOTE

Instances created since May 15, 2024 do not have Kafka Manager. You cannot create topics for these instances using Kafka Manager.

Constraints

- The total number of partitions in topics is limited. **When the partition quantity limit is reached, you can no longer create topics.** The total number of partitions varies by instance specifications. For details, see **Specifications**.
- If an instance node is faulty, an internal service error may be reported when you query messages in a topic with only one replica. Therefore, you are not advised using a topic with only one replica.

Creating a Topic on the Console

Step 1 Log in to the console.

Step 2 Click 📍 in the upper left corner to select a region.

NOTE

Select the region where your Kafka instance is located.

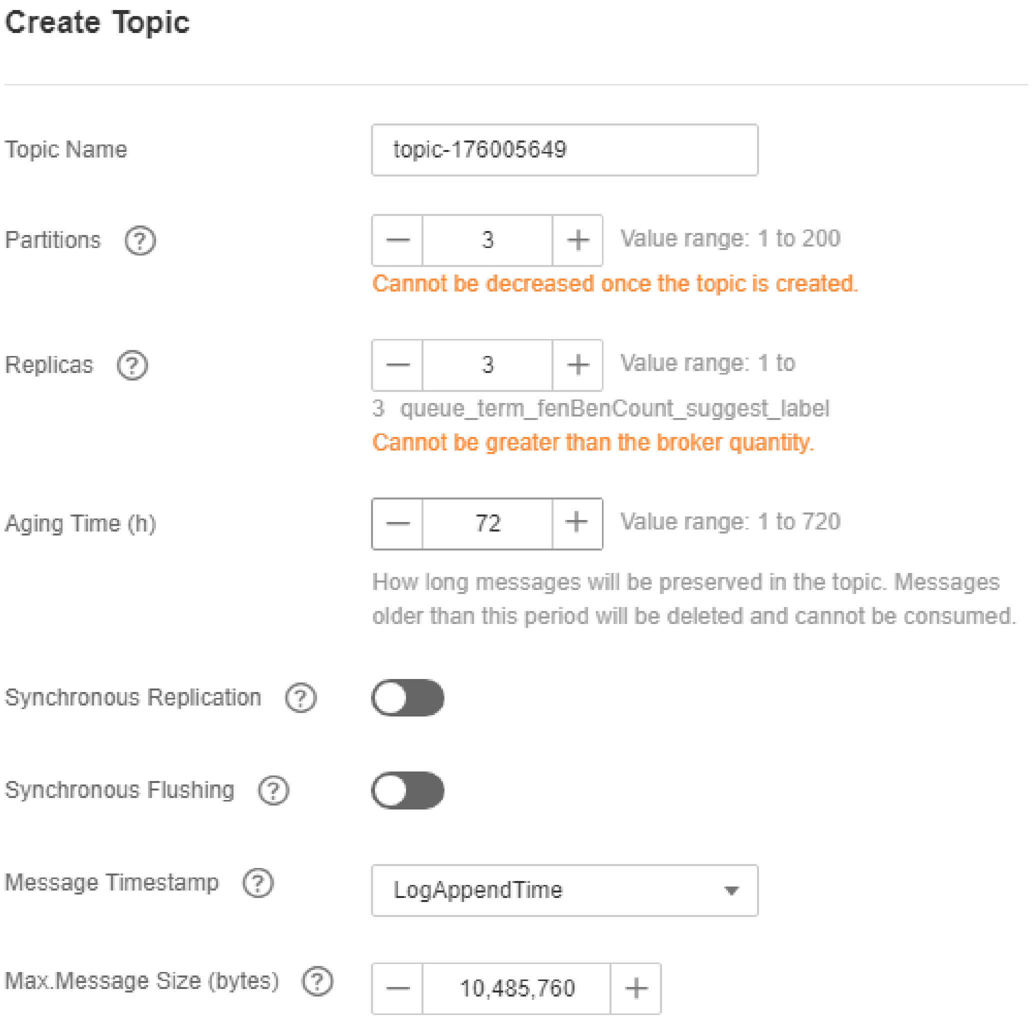
Step 3 Click ☰ and choose **Application** > **Distributed Message Service (for Kafka)** to open the console of DMS for Kafka.

Step 4 Click the desired Kafka instance to view the instance details.

Step 5 In the navigation pane, choose **Topics**. Then click **Create Topic**.

Step 6 Enter a topic name, specify other parameters, and click **OK**.

Figure 4-1 Creating a topic



Create Topic

Topic Name: topic-176005649

Partitions: 3 Value range: 1 to 200
Cannot be decreased once the topic is created.

Replicas: 3 Value range: 1 to 3 queue_term_fenBenCount_suggest_label
Cannot be greater than the broker quantity.

Aging Time (h): 72 Value range: 1 to 720
How long messages will be preserved in the topic. Messages older than this period will be deleted and cannot be consumed.

Synchronous Replication

Synchronous Flushing

Message Timestamp: LogAppendTime

Max.Message Size (bytes): 10,485,760

Table 4-2 Topic parameters

Parameter	Description
Topic Name	Customize a name that contains 3 to 200 characters, starts with a letter or underscore (_), and contains only letters, digits, periods (.), hyphens (-), and underscores (_). The name must be different from preset topics: ● __consumer_offsets ● __transaction_state ● __trace ● __connect-status ● __connect-configs ● __connect-offsets Once the topic is created, you cannot modify its name. **NOTE** Due to the limitation of the Kafka kernel, topics whose names contain only period or underscore difference cannot be created. For example, assume that the **Topic_1** topic is created, creating a topic named **Topic.1** will fail and throw the **Topic 'topic.1' collides with existing topics: topic_1** exception.
Partitions	Number of partitions in the topic. If the number of partitions is the same as that of consumers, the larger the partitions, the higher the consumption concurrency. If this parameter is set to **1**, messages will be retrieved in the FIFO order. Value range: 1 to 200
Replicas	A higher number of replicas delivers higher reliability. Data is automatically backed up on each replica. When one Kafka broker becomes faulty, data is still available on other brokers. If this parameter is set to **1**, only one set of data is available. Value range: 1 to number of brokers **NOTE** If an instance node is faulty, an internal service error may be reported when you query messages in a topic with only one replica. Therefore, you are not advised using a topic with only one replica.
Aging Time (h)	The period that messages are retained for. Consumers must retrieve messages before this period ends. Otherwise, the messages will be deleted and can no longer be consumed. Value range: 1–720

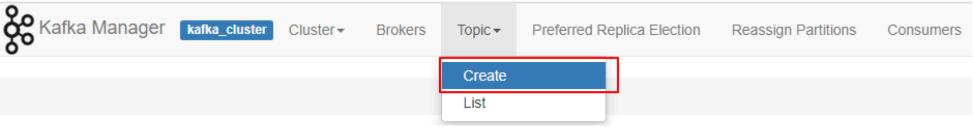
Parameter	Description
Synchronous Replication	A message is returned to the client only after the message creation request has been received and the message has been acknowledged by all replicas. After enabling this, set the parameter **acks** to **all** or **-1** in the configuration file or production code on the producer client. If there is only one replica, synchronous replication cannot be enabled.
Synchronous Flushing	A message is immediately flushed to disk once it is produced, bringing higher reliability. When this option is disabled, a message is stored in the memory instead of being immediately flushed to disk once produced.
Message Timestamp	Timestamp type of a message. Options: ● **CreateTime**: time when the producer created the message. ● **LogAppendTime**: time when the broker appended the message to the log.
Max. Message Size	Maximum batch processing size allowed by Kafka. If message compression is enabled in the client configuration file or code of producers, this parameter indicates the size after compression. If this is increased and there are consumers older than 0.10.2, the consumers' fetch size must also be increased so that they can fetch record batches this large. Value range: 0 to 10,485,760

----End

Creating a Topic on Kafka Manager

Log in to Kafka Manager, choose **Topic** > **Create**, and set parameters as prompted. To ensure performance, a partition number within 200 is recommended for each topic.

Figure 4-2 Creating a topic on Kafka Manager



NOTICE

If a topic name starts with a special character, for example, a number sign (#), monitoring data cannot be displayed.

Creating a Topic on the Client

If your client is v2.2 or later, you can use **kafka-topics.sh** to create topics and manage topic parameters.

NOTICE

- If a topic name starts with a special character, for example, a number sign (#), monitoring data cannot be displayed.
- For an instance with SASL enabled, if **allow.everyone.if.no.acl.found** is set to **false**, topics cannot be created through the client.

- For a Kafka instance with SASL disabled, run the following command in the **/bin** directory of the Kafka client:

  ```
  ./kafka-topics.sh --create --topic ${topic-name} --bootstrap-server ${connection-address} --partitions ${number-of-partitions} --replication-factor ${number-of-replicas}
  ```

  Parameter description:

  – **topic-name**: topic name, which can be customized.

  – **connection-address**: can be obtained from the **Connection** area on the **Basic Information** page on the Kafka console.

  – **number-of-partitions**: number of partitions in a topic. To ensure performance, a partition number within 200 is recommended for each topic.

  – **number-of-replicas**: number of replicas in a topic.

  Example:

  ```
  [root@ecs-kafka bin]# ./kafka-topics.sh --create --topic topic-01 --bootstrap-server 
  192.168.xx.xx:9092,192.168.xx.xx:9092,192.168.xx.xx:9092 --partitions 3 --replication-factor 3
  Created topic topic-01.
  [root@ecs-kafka bin]#
  ```

- For a Kafka instance with SASL enabled, do as follows:

  a. (Optional) If the username and password, and the SSL certificate has been configured, skip this step and go to **b**. Otherwise, do as follows:

     Create the **ssl-user-config.properties** file in the **/config** directory of the Kafka client. Add the username and password, and the SSL certificate configuration by referring to **Step 3**.

  b. Run the following command in the **/bin** directory of the Kafka client:

     ```
     ./kafka-topics.sh --create --topic ${topic-name} --bootstrap-server ${connection-address} --partitions ${number-of-partitions} --replication-factor ${number-of-replicas} --command-config ../config/ssl-user-config.properties
     ```

     Parameter description:

     ▪ **topic-name**: topic name, which can be customized.

     ▪ **connection-address**: can be obtained from the **Connection** area on the **Basic Information** page on the Kafka console.

     ▪ **number-of-partitions**: number of partitions in a topic. To ensure performance, a partition number within 200 is recommended for each topic.

- **number-of-replicas**: number of replicas in a topic.

Example:

```
[root@ecs-kafka bin]# ./kafka-topics.sh --create --topic topic-01 --bootstrap-server
192.168.xx.xx:9093,192.168.xx.xx:9093,192.168.xx.xx:9093 --partitions 3 --replication-factor 3 --
command-config ../config/ssl-user-config.properties
Created topic topic-01.
[root@ecs-kafka bin]#
```

4.2 Configuring Kafka Topic Permissions

Kafka instances with SASL_SSL enabled support access control list (ACL) for topics. You can differentiate user permissions by granting users different permissions in a topic.

This section describes how to grant topic permissions to users after SASL_SSL is enabled for Kafka instances.

Constraints

- If parameter **allow.everyone.if.no.acl.found** is set to **true** and no topic is granted for a user, all users can subscribe to or publish messages to the topic. If permissions for a topic have been granted to one or more users, only these users can subscribe to or publish messages to the topic. The value of **allow.everyone.if.no.acl.found** can be **modified**.
- If **allow.everyone.if.no.acl.found** is set to **false**, only the authorized users can subscribe to or publish messages to the topic. The value of **allow.everyone.if.no.acl.found** can be **modified**.
- If both the default and individual user permissions are configured for a topic, the union of the permissions is used.

Prerequisites

- SASL_SSL has been enabled when you create the Kafka instance.
- **A user is created**.

Procedure

Step 1 Log in to the console.

Step 2 Click 📍 in the upper left corner to select a region.

📖 NOTE

Select the region where your Kafka instance is located.

Step 3 Click ☰ and choose **Application** > **Distributed Message Service (for Kafka)** to open the console of DMS for Kafka.

Step 4 Click the desired Kafka instance to view the instance details.

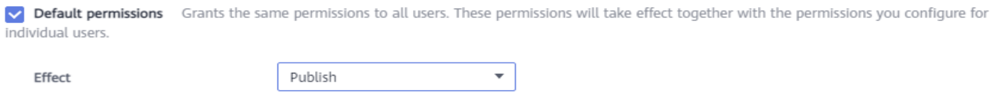
Step 5 In the navigation pane, choose **Topics**.

Step 6 In the row containing the desired topic, click **Grant User Permission**.

Step 7 Grant topic permissions to users.

- To grant the same permissions to all users, select **Default permissions** and then select permissions. As shown in the following figure, all users have the permission to publish messages to this topic.

Figure 4-3 Granting the same rights to all users

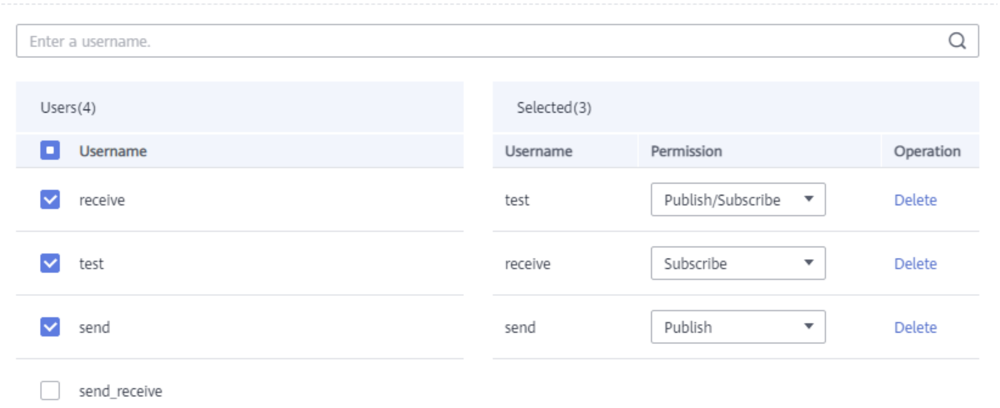


☑ **Default permissions** Grants the same permissions to all users. These permissions will take effect together with the permissions you configure for individual users.

Effect: Publish

- To grant different permissions to different users, do not select **Default permissions**. In the **Users** area of the **Grant User Permission** dialog box, select target users. If there are many users, enter the username in the search box for a quick search. In the **Selected** area, configure permissions (**Subscribe**, **Publish**, or **Publish/Subscribe**) for the users. As shown in the following figure, only the **test**, **send**, and **receive** users can subscribe to or publish messages to this topic. The **send_receive** user cannot subscribe to or publish messages to this topic.

Figure 4-4 Granting permissions to individual users



Enter a username.

Users(4)

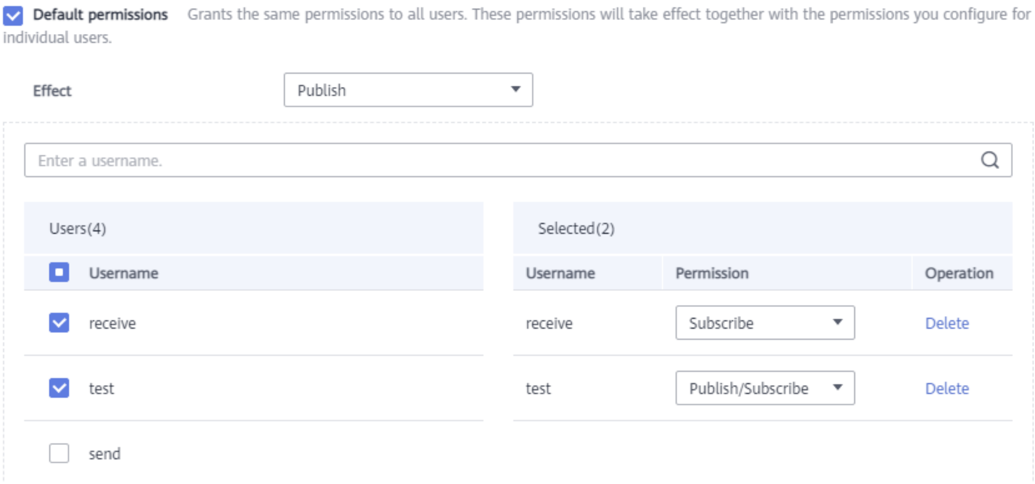
■	Username
☑	receive
☑	test
☑	send
□	send_receive

Selected(3)

Username	Permission	Operation
test	Publish/Subscribe	Delete
receive	Subscribe	Delete
send	Publish	Delete

If both the default and individual user permissions are configured for a topic, the union of the permissions is used. As shown in the following figure, the **test** and **receive** users can subscribe to and publish messages to this topic.

Figure 4-5 Granting topic permissions to users



☑ **Default permissions** Grants the same permissions to all users. These permissions will take effect together with the permissions you configure for individual users.

Effect: Publish ▾

Enter a username.

Users(4)

■	Username
☑	receive
☑	test
□	send

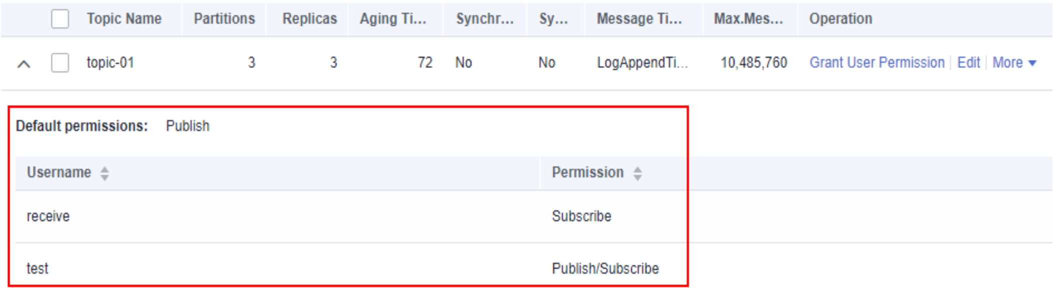
Selected(2)

Username	Permission	Operation
receive	Subscribe ▾	Delete
test	Publish/Subscribe ▾	Delete

Step 8 Click **OK**.

On the **Topics** tab page, click ⌄ next to the topic name to view the authorized users and their permissions.

Figure 4-6 Viewing authorized users and their permissions



	□	Topic Name	Partitions	Replicas	Aging Ti...	Synchr...	Sy...	Message Ti...	Max.Mes...	Operation
^	□	topic-01	3	3	72	No	No	LogAppendTi...	10,485,760	Grant User Permission | Edit | More ▾

Default permissions: Publish

Username	Permission
receive	Subscribe
test	Publish/Subscribe

----End

Deleting Topic Permissions

Step 1 Log in to the console.

Step 2 Click 📍 in the upper left corner to select a region.

NOTE

Select the region where your Kafka instance is located.

Step 3 Click ☰ and choose **Application** > **Distributed Message Service (for Kafka)** to open the console of DMS for Kafka.

Step 4 Click the desired Kafka instance to view the instance details.

Step 5 In the navigation pane, choose **Topics**.

Step 6 In the row containing the desired topic, click **Grant User Permission**.

Step 7 In the **Selected** area of the displayed **Grant User Permission** dialog box, locate the row that contains the user whose permissions are to be removed, click **Delete**, and click **OK**.

----End

4.3 Managing Topics

4.3.1 Modifying Kafka Topic Configurations

This section describes how to modify configurations in **Table 4-3** of a Kafka topic on the console.

NOTE

Modifying **Synchronous Replication**, **Synchronous Flushing**, **Message Timestamp**, or **Max. Message Size** does not require instance restart.

Table 4-3 Kafka topic configuration parameters

Parameter	Description
Partitions	Number of partitions in a topic. For details about how to change, see **Changing Kafka Partition Quantity**.
Aging Time (h)	Maximum message retention. For details about how to change, see **Changing Kafka Message Retention Period**.
Replicas	Number of replicas of each topic partition. To modify it, see **Modifying Kafka Topic Replicas**.
Synchronous Replication	A message is returned to the client only after the message creation request has been received and the message has been acknowledged by all replicas.
Synchronous Flushing	● Enabled: A message is immediately flushed to disk once it is created, bringing higher reliability. ● Disabled: A message is stored in the memory instead of being immediately flushed to disk once created.
Message Timestamp	Timestamp type of a message. Options: ● **CreateTime**: time when the producer created the message. ● **LogAppendTime**: time when the broker appended the message to the log.

Parameter	Description
Max. Message Size	Maximum size of messages to be processed in batches. If message compression is enabled, this parameter indicates the size after compression. If this value is increased and the consumer version is earlier than 0.10.2, the consumers' fetch size must also be increased so that they can obtain the latest value.

Procedure

Step 1 Log in to the console.

Step 2 Click 📍 in the upper left corner to select a region.

NOTE

Select the region where your Kafka instance is located.

Step 3 Click ☰ and choose **Application** > **Distributed Message Service (for Kafka)** to open the console of DMS for Kafka.

Step 4 Click the desired Kafka instance to view the instance details.

Step 5 In the navigation pane, choose **Topics**.

Step 6 Modify topic configurations in either of the following ways:

- Select one or more topics and click **Edit Topic** above the topic list.
- In the row containing the desired topic, click **Edit**.

Step 7 In the **Edit Topic** dialog box, change configurations and click **OK**.

NOTE

- If there is only one replica, **Synchronous Replication** cannot be enabled.
- After enabling synchronous replication, set **acks** to **all** or **–1** on the client. Otherwise, this function will not take effect.

----End

4.3.2 Changing Kafka Partition Quantity

After creating a topic, you can increase the number of partitions as required.

NOTE

Changing the number of partitions does not restart the instance or affect services.

Methods for changing the partition quantity:

- **Modifying Topic Partitions on the Console**
- **Modifying Topic Partitions on Kafka Manager**
- **Modifying Topic Partitions on the Client**

NOTE

Instances created since May 15, 2024 do not have Kafka Manager. You cannot modify topic partitions for these instances using Kafka Manager.

Modifying Topic Partitions on the Console

Step 1 Log in to the console.

Step 2 Click 📍 in the upper left corner to select a region.

NOTE

Select the region where your Kafka instance is located.

Step 3 Click ☰ and choose **Application** > **Distributed Message Service (for Kafka)** to open the console of DMS for Kafka.

Step 4 Click the desired Kafka instance to view the instance details.

Step 5 In the navigation pane, choose **Topics**.

Step 6 Modify the number of partitions using either of the following methods:

- Select one or more topics and click **Edit Topic** in the upper left corner.
- In the row containing the desired topic, click **Edit**.

Step 7 In the **Edit Topic** dialog box, enter the number of partitions and click **OK**.

NOTE

- The number of partitions can only be increased.
- To ensure performance, the Kafka console allows a maximum of 200 partitions for each topic.
- The total partitions of all topics cannot exceed the maximum partitions of an instance. The maximum partitions of an instance vary by instance specifications. For details, see **Kafka Instance Specifications**.

----End

Modifying Topic Partitions on Kafka Manager

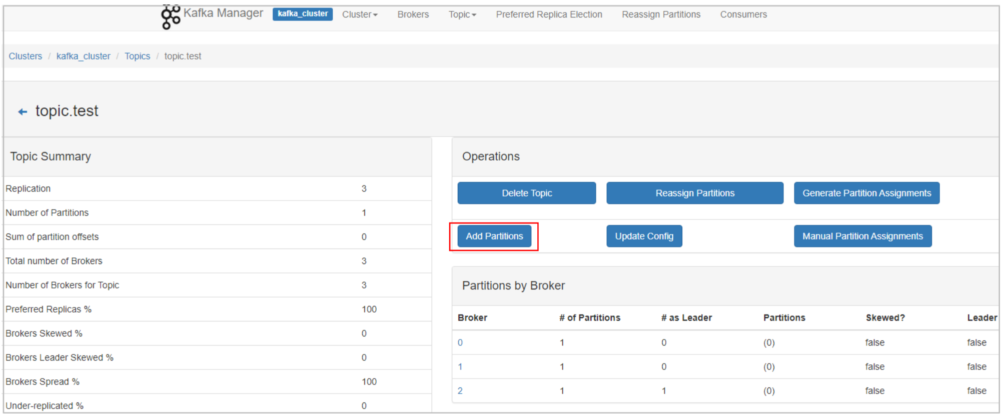
Step 1 **Log in to Kafka Manager**.

Step 2 Choose **Topic** > **List** to view the list of topics.

Step 3 Click a topic to view its details.

Step 4 Click **Add Partitions**.

Figure 4-7 Topic details page



Topic Summary

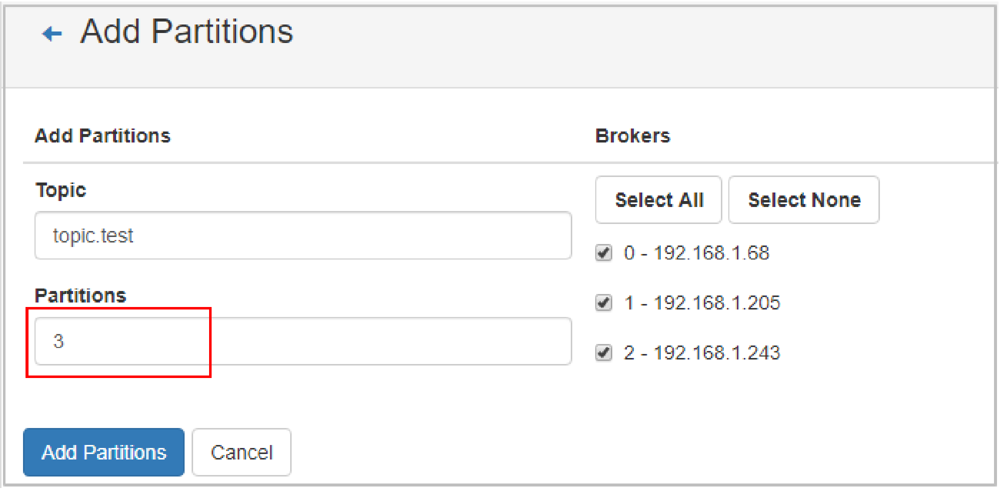
Replication	3
Number of Partitions	1
Sum of partition offsets	0
Total number of Brokers	3
Number of Brokers for Topic	3
Preferred Replicas %	100
Brokers Skewed %	0
Brokers Leader Skewed %	0
Brokers Spread %	100
Under-replicated %	0

Partitions by Broker

Broker	# of Partitions	# as Leader	Partitions	Skewed?	Leader
0	1	0	(0)	false	false
1	1	0	(0)	false	false
2	1	1	(0)	false	false

Step 5 Enter the number of partitions and click **Add Partitions**.

Figure 4-8 Adding partitions



Topic: topic.test

Partitions: 3

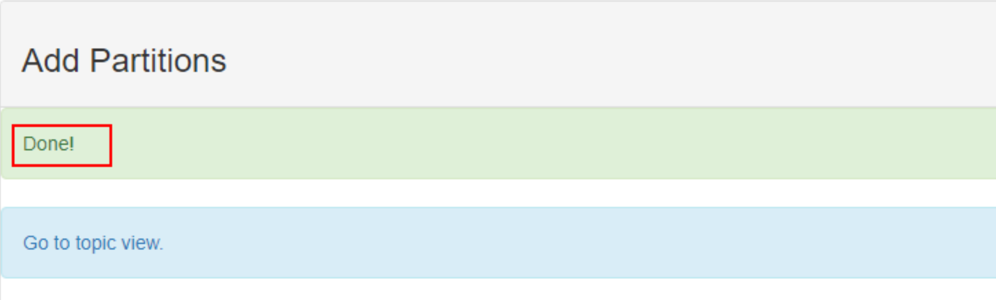
☑ 0 - 192.168.1.68

☑ 1 - 192.168.1.205

☑ 2 - 192.168.1.243

If "Done" is displayed, the partitions are added successfully.

Figure 4-9 Partitions added



Done!

Go to topic view.

NOTE

- The number of partitions can only be increased.
- To ensure performance, a partition number within 200 is recommended for each topic.
- The total partitions of all topics cannot exceed the maximum partitions of an instance. The maximum partitions of an instance vary by instance specifications. For details, see **Kafka Instance Specifications**.

----End

Modifying Topic Partitions on the Client

If your Kafka client version is later than 2.2, you can use **kafka-topics.sh** to change the partition quantity.

NOTICE

For an instance with SASL enabled, if **allow.everyone.if.no.acl.found** is set to **false**, topic partition quantity cannot be modified through the client.

- For a Kafka instance with SASL disabled, run the following command in the **/bin** directory of the Kafka client:

  ```
  ./kafka-topics.sh --bootstrap-server ${connection-address} --topic ${topic-name} --alter --partitions $
  {number-of-partitions}
  ```

  Parameter description:

  - **connection-address**: can be obtained from the **Connection** area on the **Basic Information** page on the Kafka console.
  - **topic-name**: topic name.
  - **number-of-partitions**: number of partitions in a topic. To ensure performance, a partition number within 200 is recommended for each topic.

  Example:

  ```
  [root@ecs-kafka bin]# ./kafka-topics.sh --bootstrap-server
  192.168.xx.xx:9092,192.168.xx.xx:9092,192.168.xx.xx:9092 --topic topic-01 --alter --partitions 6
  [root@ecs-kafka bin]#
  ```

- For a Kafka instance with SASL enabled, do as follows:

  a. (Optional) If the username and password, and the SSL certificate has been configured, skip this step and go to **b**. Otherwise, do as follows:

     Create the **ssl-user-config.properties** file in the **/config** directory of the Kafka client. Add the username and password, and the SSL certificate configuration by referring to **Step 3**.

  b. Run the following command in the **/bin** directory of the Kafka client:

     ```
     ./kafka-topics.sh --bootstrap-server ${connection-address} --topic ${topic-name} --alter --
     partitions ${number-of-partitions} --command-config ../config/ssl-user-config.properties
     ```

     Parameter description:

     - **connection-address**: can be obtained from the **Connection** area on the **Basic Information** page on the Kafka console.
     - **topic-name**: topic name.

- **number-of-partitions**: number of partitions in a topic. To ensure performance, a partition number within 200 is recommended for each topic.

Example:

```
[root@ecs-kafka bin]# ./kafka-topics.sh --bootstrap-server
192.168.xx.xx:9093,192.168.xx.xx:9093,192.168.xx.xx:9093 --topic topic-01 --alter --partitions 6 --
command-config ../config/ssl-user-config.properties
[root@ecs-kafka bin]#
```

4.3.3 Modifying Kafka Topic Replicas

The replicas of a Kafka topic can be modified as required.

Reassigning partitions can modify replicas automatically or manually on the Kafka console. For more information, see **Modifying Replicas by Automatic Reassignment** and **Modifying Replicas by Manual Reassignment**.

Operation Impact

- Partition reassignment on topics with a large amount of data consumes a large amount of network and storage bandwidth. As a result, service requests may time out or the latency may increase. Therefore, you are advised to perform reassignment during off-peak hours. Compare the current instance load based on the instance specifications to decide whether the remaining instance capacity can support partition reassignment. Do not reassign partitions when there is insufficient bandwidth or when the CPU usage is greater than 90%. To view data volume and CPU usage of a topic, see **Message Size** and **CPU Usage** on the monitoring page. For details, see **Viewing Kafka Metrics**.
- A throttle refers to the upper limit of the bandwidth for replication of a topic, to ensure that other topics on the instance are not affected. Note that throttles apply to replication triggered by both normal message production and partition reassignment. If the throttle is too small, normal message production may be affected, and partition reassignment may never complete. If partitions are continuously reassigned, contact customer service.
- You cannot delete topics whose reassignment tasks have started. Otherwise, the tasks will never complete.
- You cannot modify the partition quantity of topics whose reassignment tasks have started.
- Reassignment tasks cannot be manually stopped. Please wait until they complete.
- After partition reassignment, the metadata of the topic changes. If the producer does not support the retry mechanism, a few requests will fail, causing some messages to fail to be produced.
- Reassignment takes a long time if the topic has a large amount of data. You are advised to **decrease the topic retention period** based on the topic consumption so that historical data of the topic can be deleted in a timely manner to accelerate the migration. To view data volume of a topic, see **Message Size** on the monitoring page. For details, see **Viewing Kafka Metrics**.

Preparing for Partition Reassignment

- To reduce the amount of data to be migrated, decrease the topic retention period without affecting services and wait for messages to age. After the reassignment is complete, you can restore the retention period. To change the topic retention period, see **Changing Kafka Message Retention Period**.
- The target broker should have sufficient disk space. To check available disk space of each broker, see **Viewing Kafka Disk Usage**. If the remaining disk capacity of the target broker is close to the amount of data to be migrated to the broker, **expand the disk capacity** before the reassignment.

Modifying Replicas by Automatic Reassignment

Step 1 Log in to the console.

Step 2 Click 📍 in the upper left corner to select a region.

NOTE

Select the region where your Kafka instance is located.

Step 3 Click ☰ and choose **Application** > **Distributed Message Service (for Kafka)** to open the console of DMS for Kafka.

Step 4 Click the desired Kafka instance to view the instance details.

Step 5 In the navigation pane, choose **Topics**.

Step 6 Go to the **Auto** page in either of the following ways:

- Select one or more topics and choose **Reassign** > **Auto** above the topic list.
- In the row containing the desired topic, choose **More** > **Reassign** > **Auto**.

Step 7 Modify the replicas.

Table 4-4 Parameters of automatic reassignment

Parameter	Description
Brokers	Select the brokers to assign the topic's partition replicas to.
Topics	Enter the number of replicas. This number must be less than or equal to the number of brokers.
Max. Bandwidth	Specify **throttle**. The default value is **-1**, indicating that there is no throttle. If the instance has low workload (for example, only 30/300 MB/s is used), you are not advised to limit the bandwidth. Otherwise, you are advised to set it to a value greater than or equal to the total production bandwidth of the to-be-reassigned topic multiplied by the maximum number of replicas of the to-be-reassigned topic. For details, see **Calculating a Throttle**.

Parameter	Description
Execute	Specify when to execute the reassignment. ● **Now** means to execute it immediately. ● **As scheduled** means to execute it at the scheduled time.

Step 8 (Optional) Click **Calculate**. **Time Required** indicates how long automatic balancing will take.

The one-click calculation function does not affect the performance of Kafka instances.

Step 9 Click **OK**.

The following table lists how to check whether reassignment is complete (scheduled and non-scheduled tasks):

Table 4-5 Checking the reassignment result

Task Type	Reassignment Result
Background tasks	In the upper left corner of the topic list, click **View details** and the **Background Tasks** > **Background tasks** page is displayed. The reassignment task is complete when it is in the **Successful** state, which means that the replicas are modified.
Scheduled tasks	1. The **Background Tasks** > **Scheduled tasks** page is displayed. This page only shows whether scheduled tasks start to execute instead of whether they are successful. – When the task status is **Pending**, reassignment has not been executed. – When the task status is **Successful**, reassignment has started. – When the task status is **Cancel**, reassignment has been canceled. 2. Click **Background tasks** tab page. When the task status is **Successful**, reassignment has completed, which means that the replicas are modified.

----End

Modifying Replicas by Manual Reassignment

Step 1 Log in to the console.

Step 2 Click 📍 in the upper left corner to select a region.

NOTE

Select the region where your Kafka instance is located.

Step 3 Click ☰ and choose **Application** > **Distributed Message Service (for Kafka)** to open the console of DMS for Kafka.

Step 4 Click the desired Kafka instance to view the instance details.

Step 5 In the navigation pane, choose **Topics**.

Step 6 Go to the **Manual** page in either of the following ways:

- Select a topic and choose **Reassign** > **Manual** above the topic list. Manual reassignment does not support batch operations.
- In the row containing the desired topic, choose **More** > **Reassign** > **Manual**.

Step 7 Modify the replicas.

- In the upper right corner of the **Manual** dialog box, click **Delete Replica** or **Add Replica** to reduce or increase the number of replicas for each partition of the topic.
- Under the name of the replica to be reassigned, click the broker name or ▼ and select the target broker to migrate the replica to. Assign replicas of the same partition to different brokers.
- Specify **throttle**. The default value is **-1**, indicating that there is no throttle. If the instance has low workload (for example, only 30/300 MB/s is used), you are not advised to limit the bandwidth. Otherwise, you are advised to set it to a value greater than or equal to the total production bandwidth of the to-be-reassigned topic multiplied by the maximum number of replicas of the to-be-reassigned topic. For details, see **Calculating a Throttle**.
- For **Execute**, specify when to execute the reassignment. **Now** means to execute it immediately. **As scheduled** means to execute it at the scheduled time.

Step 8 (Optional) Click **Calculate**. **Time Required** indicates how long manual balancing will take.

The one-click calculation function does not affect the performance of Kafka instances.

Step 9 Click **OK**.

The following table lists how to check whether reassignment is complete (scheduled and non-scheduled tasks):

Table 4-6 Checking the reassignment result

Task Type	Reassignment Result
Background tasks	In the upper left corner of the topic list, click **View details** and the **Background Tasks** > **Background tasks** page is displayed. The reassignment task is complete when it is in the **Successful** state, which means that the replicas are modified.

Task Type	Reassignment Result
Scheduled tasks	1. The **Background Tasks** > **Scheduled tasks** page is displayed. This page only shows whether scheduled tasks start to execute instead of whether they are successful. – When the task status is **Pending**, reassignment has not been executed. – When the task status is **Successful**, reassignment has started. – When the task status is **Cancel**, reassignment has been canceled. 2. Click **Background tasks** tab page. When the task status is **Successful**, reassignment has completed, which means that the replicas are modified.

----End

4.3.4 Exporting the Kafka Topic List

Export the topic list on the console. Batch export is supported.

Prerequisites

A topic has been created.

Procedure

Step 1 Log in to the console.

Step 2 Click 📍 in the upper left corner to select a region.

📖 **NOTE**

Select the region where your Kafka instance is located.

Step 3 Click ☰ and choose **Application** > **Distributed Message Service (for Kafka)** to open the console of DMS for Kafka.

Step 4 Click the desired Kafka instance to view the instance details.

Step 5 In the navigation pane, choose **Topics**.

Step 6 Click ↗ in the upper right to export the topic list.

The topic list contains the following information: topic name, number of partitions, number of replicas, aging time, message timestamp, max. message size, and whether synchronous replication and flushing are enabled.

----End

4.3.5 Reassigning Kafka Partitions

Partition reassignment is to reassign replicas of a partition to different brokers to solve the problem of unbalanced broker load.

Partition reassignment is required in the following scenarios:

- After you add brokers to an instance, new topics are created on new brokers, and the original topics are still on the original brokers, resulting in unbalanced partitions. To migrate the replicas of the original topic partitions to the new brokers, reassign partitions.
- The leader partition is degraded to be a follower on a heavily loaded broker.
- The replica quantity of a topic can be changed during partition reassignment.

The DMS for Kafka console provides **automatic** and **manual** reassignment. Automatic reassignment is recommended because it ensures that leaders are evenly distributed.

Operation Impact

- Partition reassignment on topics with a large amount of data consumes a large amount of network and storage bandwidth. As a result, service requests may time out or the latency may increase. Therefore, you are advised to perform reassignment during off-peak hours. Compare the current instance load based on the instance specifications to decide whether the remaining instance capacity can support partition reassignment. Do not reassign partitions when there is insufficient bandwidth or when the CPU usage is greater than 90%. To view data volume and CPU usage of a topic, see **Message Size** and **CPU Usage** on the monitoring page. For details, see **Viewing Kafka Metrics**.
- A throttle refers to the upper limit of the bandwidth for replication of a topic, to ensure that other topics on the instance are not affected. Note that throttles apply to replication triggered by both normal message production and partition reassignment. If the throttle is too small, normal message production may be affected, and partition reassignment may never complete. If partitions are continuously reassigned, contact customer service.
- You cannot delete topics whose reassignment tasks have started. Otherwise, the tasks will never complete.
- You cannot modify the partition quantity of topics whose reassignment tasks have started.
- Reassignment tasks cannot be manually stopped. Please wait until they complete.
- After partition reassignment, the metadata of the topic changes. If the producer does not support the retry mechanism, a few requests will fail, causing some messages to fail to be produced.
- Reassignment takes a long time if the topic has a large amount of data. You are advised to **decrease the topic retention period** based on the topic consumption so that historical data of the topic can be deleted in a timely manner to accelerate the migration. To view data volume of a topic, see **Message Size** on the monitoring page. For details, see **Viewing Kafka Metrics**.

Preparing for Partition Reassignment

- To reduce the amount of data to be migrated, decrease the topic retention period without affecting services and wait for messages to age. After the reassignment is complete, you can restore the retention period. To change the topic retention period, see **Changing Kafka Message Retention Period**.
- The target broker should have sufficient disk space. To check available disk space of each broker, see **Viewing Kafka Disk Usage**. If the remaining disk capacity of the target broker is close to the amount of data to be migrated to the broker, **expand the disk capacity** before the reassignment.

Auto Reassignment

Step 1 Log in to the console.

Step 2 Click 📍 in the upper left corner to select a region.

NOTE

Select the region where your Kafka instance is located.

Step 3 Click ☰ and choose **Application** > **Distributed Message Service (for Kafka)** to open the console of DMS for Kafka.

Step 4 Click the desired Kafka instance to view the instance details.

Step 5 In the navigation pane, choose **Topics**.

Step 6 Reassign partitions using either of the following methods:

- Select one or more topics and choose **Reassign** > **Auto** above the topic list.
- In the row that contains the desired topic, choose **More** > **Reassign** > **Auto**.

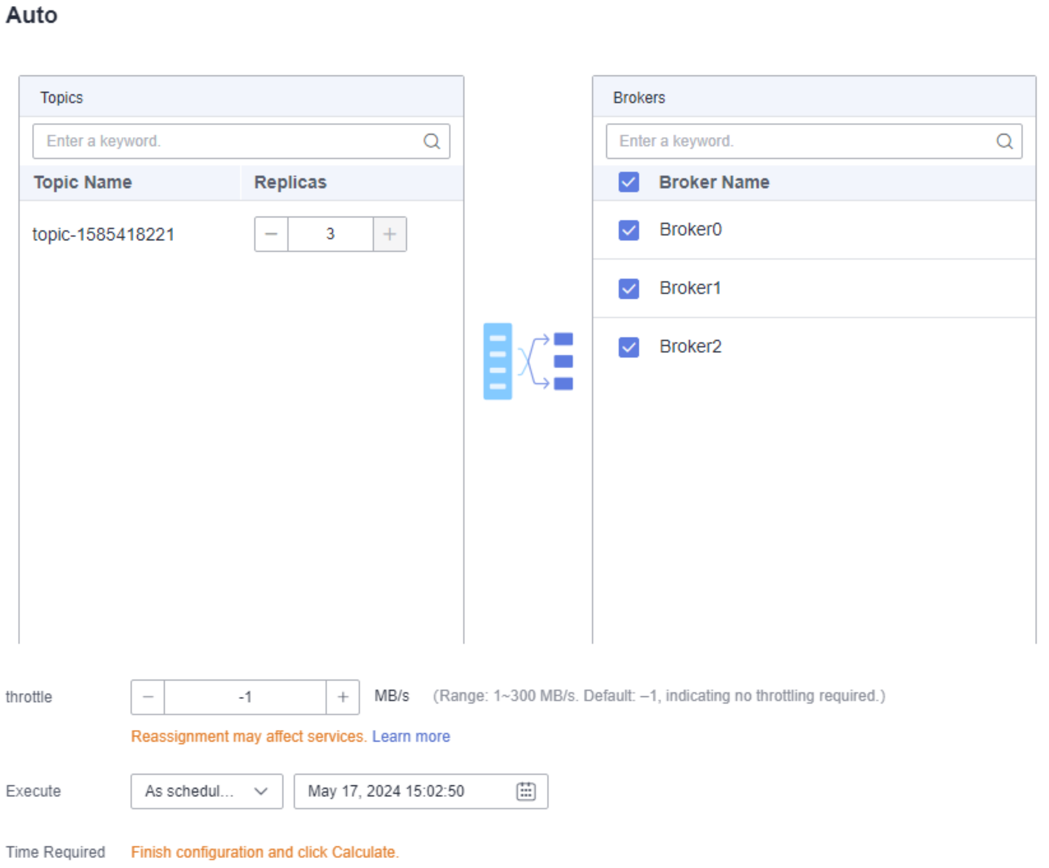
Step 7 Set automatic reassignment parameters.

Table 4-7 Parameters of automatic reassignment

Parameter	Description
Brokers	Select the brokers to assign the topic's partition replicas to.
Topics	Enter the number of replicas to be automatically reassigned. The number of replicas must be less than or equal to the number of brokers.

Parameter	Description
Max. Bandwidth	Specify **throttle**. The default value is **-1**, indicating that there is no throttle. If the instance has low workload (for example, only 30/300 MB/s is used), you are not advised to limit the bandwidth. Otherwise, you are advised to set it to a value greater than or equal to the total production bandwidth of the to-be-reassigned topic multiplied by the maximum number of replicas of the to-be-reassigned topic. For details, see **Calculating a Throttle**.
Execute	Specify when to execute the reassignment. ● **Now** means to execute it immediately. ● **As scheduled** means to execute it at the scheduled time.

Figure 4-10 Setting automatic reassignment parameters



Auto

Topics

Enter a keyword.

Topic Name	Replicas
topic-1585418221	3

Brokers

Enter a keyword.

☑ Broker Name
☑ Broker0
☑ Broker1
☑ Broker2

throttle -1 MB/s (Range: 1~300 MB/s. Default: –1, indicating no throttling required.)

Reassignment may affect services. Learn more

Execute As schedul... May 17, 2024 15:02:50

Time Required Finish configuration and click Calculate.

Step 8 (Optional) Click **Calculate**. **Time Required** indicates how long automatic balancing will take.

The one-click calculation function does not affect the performance of Kafka instances.

Step 9 Click **OK**.

The following table lists how to check whether reassignment is complete (scheduled and non-scheduled tasks):

Table 4-8 Checking the reassignment result

Task Type	Reassignment Result
Background tasks	In the upper left corner of the topic list, click **View details** and the **Background Tasks** > **Background tasks** page is displayed. The reassignment task is complete when it is in the **Successful** state.
Scheduled tasks	1. The **Background Tasks** > **Scheduled tasks** page is displayed. This page only shows whether scheduled tasks start to execute instead of whether they are successful. – When the task status is **Pending**, reassignment has not been executed. – When the task status is **Successful**, reassignment has started. – When the task status is **Cancel**, reassignment has been canceled. 2. Click **Background tasks** tab page. When the task status is **Successful**, reassignment has completed.

NOTE

- You cannot delete topics whose reassignment tasks have started. Otherwise, the tasks will never complete.
- You cannot modify the partition quantity of topics whose reassignment tasks have started.
- Reassignment tasks cannot be manually stopped. Please wait until they complete.
- If partition reassignment has been scheduled, reassignment cannot be scheduled again for any topic in this instance until this reassignment is executed.

----End

Manual Reassignment

Step 1 Log in to the console.

Step 2 Click 📍 in the upper left corner to select a region.

NOTE

Select the region where your Kafka instance is located.

Step 3 Click ☰ and choose **Application** > **Distributed Message Service (for Kafka)** to open the console of DMS for Kafka.

Step 4 Click the desired Kafka instance to view the instance details.

Step 5 In the navigation pane, choose **Topics**.

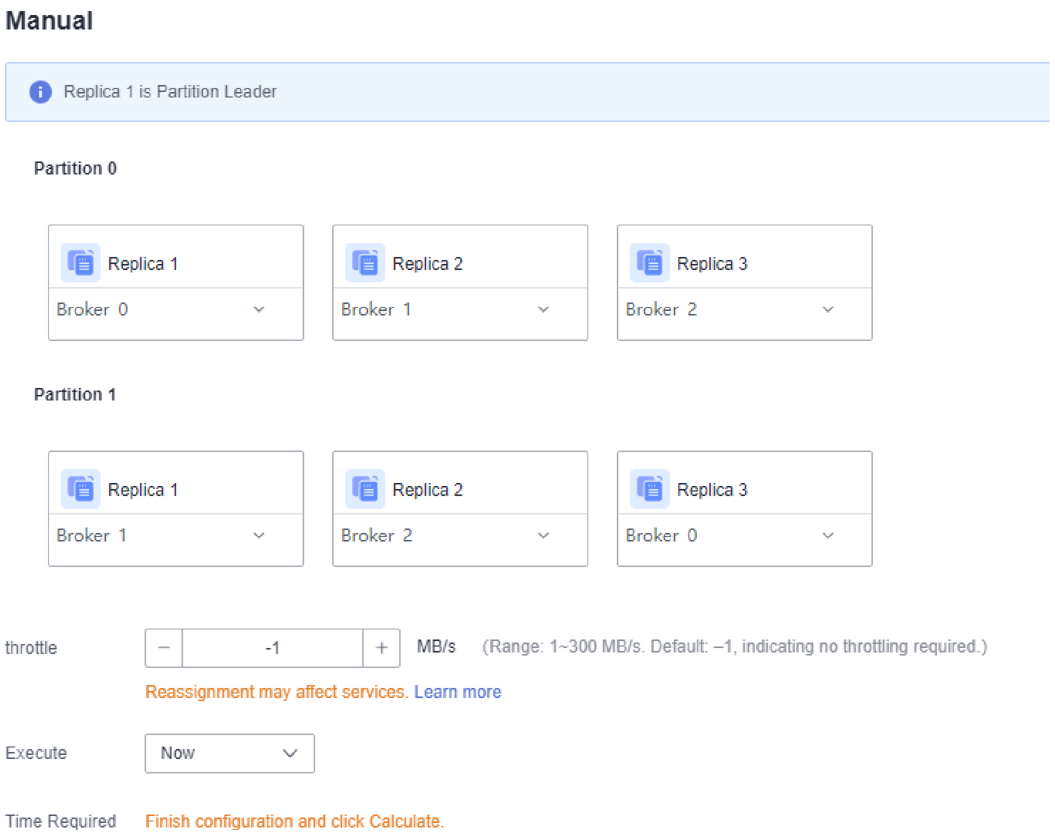
Step 6 Reassign partitions using either of the following methods:

- Select a topic and choose **Reassign** > **Manual** above the topic list. Manual reassignment does not support batch operations.
- In the row that contains the desired topic, choose **More** > **Reassign** > **Manual**.

Step 7 Set manual reassignment parameters.

- In the upper right corner of the **Manual** dialog box, click **Delete Replica** or **Add Replica** to reduce or increase the number of replicas for each partition of the topic.
- Under the name of the replica to be reassigned, click the broker name or ▼ and select the target broker to migrate the replica to. Assign replicas of the same partition to different brokers.
- Specify **throttle**. The default value is **-1**, indicating that there is no throttle If the instance has low workload (for example, only 30/300 MB/s is used), you are not advised to limit the bandwidth. Otherwise, you are advised to set it to a value greater than or equal to the total production bandwidth of the to-be-reassigned topic multiplied by the maximum number of replicas of the to-be-reassigned topic. For details, see **Calculating a Throttle**.
- For **Execute**, specify when to execute the reassignment. **Now** means to execute it immediately. **As scheduled** means to execute it at the scheduled time.

Figure 4-11 Setting manual reassignment parameters



Manual

Replica 1 is Partition Leader

Partition 0

Replica 1	Replica 2	Replica 3
Broker 0	Broker 1	Broker 2

Partition 1

Replica 1	Replica 2	Replica 3
Broker 1	Broker 2	Broker 0

throttle: -1 MB/s (Range: 1~300 MB/s. Default: –1, indicating no throttling required.)

Reassignment may affect services. Learn more

Execute: Now

Time Required: Finish configuration and click Calculate.

Step 8 (Optional) Click **Calculate**. **Time Required** indicates how long manual balancing will take.

The one-click calculation function does not affect the performance of Kafka instances.

Step 9 Click **OK**.

The following table lists how to check whether reassignment is complete (scheduled and non-scheduled tasks):

Table 4-9 Checking the reassignment result

Task Type	Reassignment Result
Background tasks	In the upper left corner of the topic list, click **View details** and the **Background Tasks** > **Background tasks** page is displayed. The reassignment task is complete when it is in the **Successful** state.
Scheduled tasks	1. The **Background Tasks** > **Scheduled tasks** page is displayed. This page only shows whether scheduled tasks start to execute instead of whether they are successful. – When the task status is **Pending**, reassignment has not been executed. – When the task status is **Successful**, reassignment has started. – When the task status is **Cancel**, reassignment has been canceled. 2. Click **Background tasks** tab page. When the task status is **Successful**, reassignment has completed.

NOTE

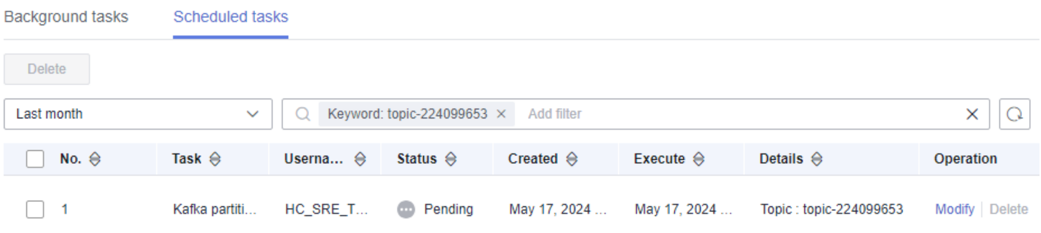
- You cannot delete topics whose reassignment tasks have started. Otherwise, the tasks will never complete.
- You cannot modify the partition quantity of topics whose reassignment tasks have started.
- Reassignment tasks cannot be manually stopped. Please wait until they complete.
- If partition reassignment has been scheduled, reassignment cannot be scheduled again for any topic in this instance until this reassignment is executed.

----End

Re-scheduling Partition Reassignment

Step 1 On the **Scheduled tasks** tab page on the **Background Tasks** page, click the drop-down box in the upper left corner, select a time period, enter the desired topic name in the search box, and press **Enter**.

Figure 4-12 Querying reassignment schedules



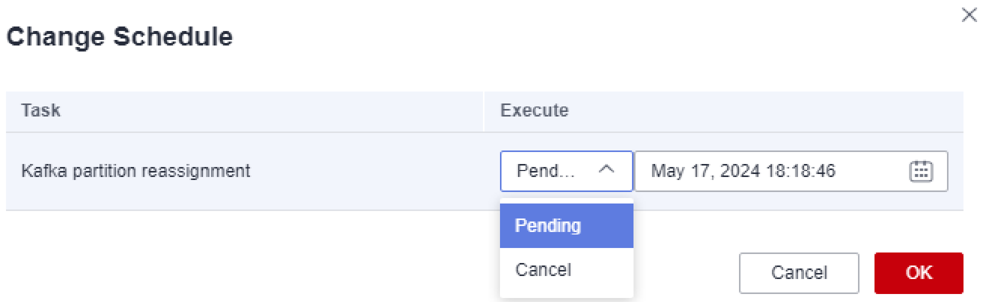
No.	Task	Userna...	Status	Created	Execute	Details	Operation
1	Kafka partiti...	HC_SRE_T...	Pending	May 17, 2024 ...	May 17, 2024 ...	Topic : topic-224099653	Modify | Delete

Step 2 In the row that contains the desired task, click **Modify**.

Step 3 In the **Change Schedule** dialog box, change the schedule or cancel the scheduled task.

- To change the schedule, select a time and click **OK**.
- To cancel the task, select **Cancel** (as shown in **Figure 4-13**) and click **OK**.

Figure 4-13 Canceling a reassignment schedule



Task	Execute
Kafka partition reassignment	Pend... May 17, 2024 18:18:46

----End

Calculating a Throttle

Throttles are affected by the execution duration of the reassignment, leader/ follower distribution of partition replicas, and message production rate.

- A throttle limits the replication traffic of all partitions in a broker.
- Replicas added after the assignment are regarded as followers, and existing replicas are regarded as leaders. Throttles on leaders and followers are separated.
- Throttles do not distinguish between replication caused by normal message production and that caused by partition reassignment. Therefore, the traffic generated in both cases is throttled.

Assume that the partition reassignment task needs to be completed within 200s and each replica has 100 MB data. Calculate the throttle in the following scenarios:

Scenario 1: Topic 1 has two partitions and two replicas, and Topic 2 has one partition and one replica. All leader replicas are on the same broker, as shown in Table 4-10. One replica needs to be added for Topic 1 and Topic 2 respectively, as shown in Table 4-11.

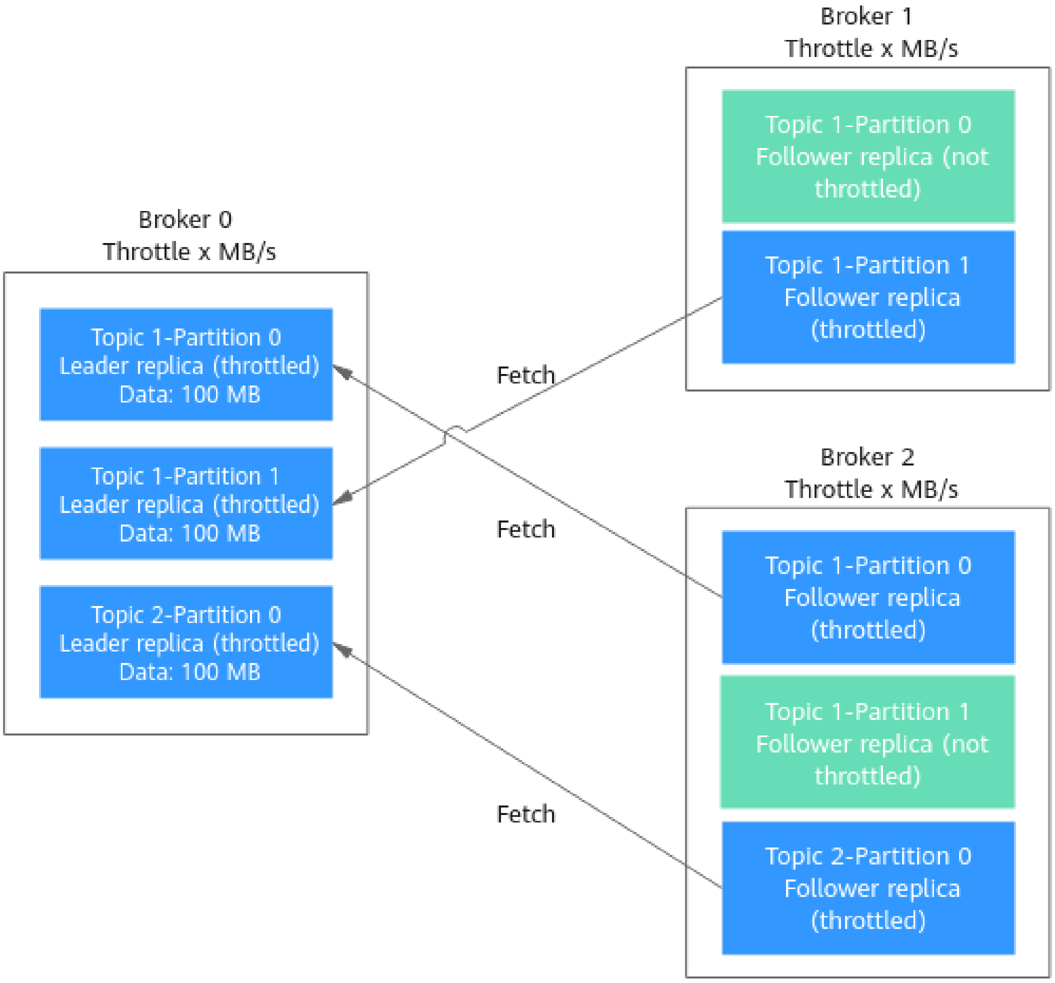
Table 4-10 Replica distribution before reassignment

Topic Name	Partition Name	Broker of Leader Replica	Broker of Follower Replica
Topic 1	0	0	0, 1
Topic 1	1	0	0, 2
Topic 2	0	0	0

Table 4-11 Replica distribution after reassignment

Topic Name	Partition Name	Broker of Leader Replica	Broker of Follower Replica
Topic 1	0	0	0, 1, 2
Topic 1	1	0	0, 1, 2
Topic 2	0	0	0, 2

Figure 4-14 Reassignment scenario 1



As shown in **Figure 4-14**, three replicas fetch data from Broker 0. Each replica on Broker 0 has 100 MB data. Broker 0 has only leader replicas, and Broker 1 and Broker 2 have only follower replicas.

- Bandwidth required by Broker 0 to complete partition reassignment within 200s = (100 MB + 100 MB + 100 MB)/200s = 1.5 MB/s
- Bandwidth required by Broker 1 to complete partition reassignment within 200s = 100 MB/200s = 0.5 MB/s
- Bandwidth required by Broker 2 to complete partition reassignment within 200s = (100 MB + 100 MB)/200s = 1 MB/s

In conclusion, to complete the partition reassignment task within 200s, set the throttle to a value greater than or equal to 1.5 MB/s. The bandwidth should be set to be greater than or equal to 2 MB/s because the limit on it on the console must be an integer.

Scenario 2: Topic 1 has two partitions and one replica, and Topic 2 has two partitions and one replica. Leader replicas are on different brokers, as shown in Table 4-12. One replica needs to be added for Topic 1 and Topic 2 respectively, as shown in Table 4-13.

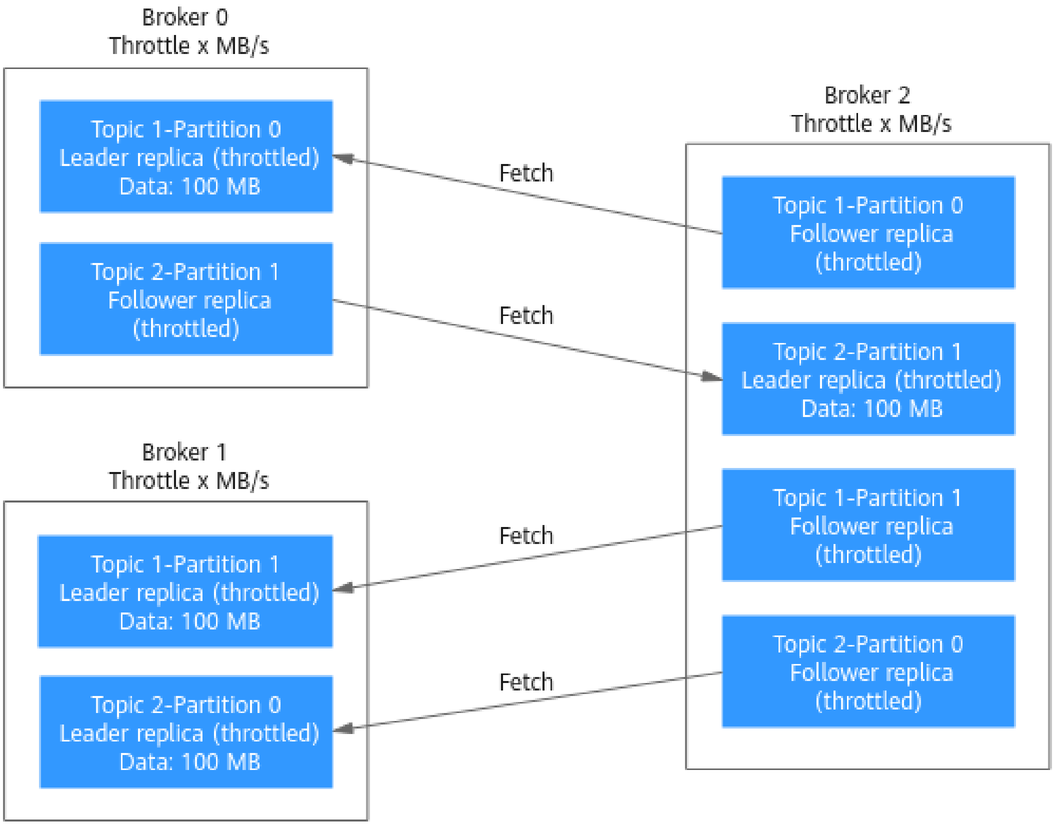
Table 4-12 Replica distribution before reassignment

Topic Name	Partition Name	Broker of Leader Replica	Broker of Follower Replica
Topic 1	0	0	0
Topic 1	1	1	1
Topic 2	0	1	1
Topic 2	1	2	2

Table 4-13 Replica distribution after reassignment

Topic Name	Partition Name	Broker of Leader Replica	Broker of Follower Replica
Topic 1	0	0	0, 2
Topic 1	1	1	1, 2
Topic 2	0	1	1, 2
Topic 2	1	2	2, 0

Figure 4-15 Reassignment scenario 2



As shown in **Figure 4-15**, Broker 1 has only leader replicas, and Broker 0 and Broker 2 have both leader and follower replicas. Leader and follower replicas on Broker 0 and Broker 2 are throttled separately.

- Bandwidth required by Broker 0 (leader) to complete partition reassignment within 200s = 100 MB/200s = 0.5 MB/s
- Bandwidth required by Broker 0 (follower) to complete partition reassignment within 200s = 100 MB/200s = 0.5 MB/s
- Bandwidth required by Broker 1 to complete partition reassignment within 200s = (100 MB + 100 MB)/200s = 1 MB/s
- Bandwidth required by Broker 2 (leader) to complete partition reassignment within 200s = 100 MB/200s = 0.5 MB/s
- Bandwidth required by Broker 2 (follower) to complete partition reassignment within 200s = (100 MB + 100 MB + 100 MB)/200s = 1.5 MB/s

In conclusion, to complete the partition reassignment task within 200s, set the throttle to a value greater than or equal to 1.5 MB/s. The bandwidth should be set to be greater than or equal to 2 MB/s because the limit on it on the console must be an integer.

Scenario 3: Both Topic 1 and Topic 2 have one partition and two replicas. All leader replicas are on the same broker. One replica needs to be added to Topic 1, as shown in Table 4-14. Messages are produced on Topic 1, causing replication, as shown in Table 4-15.

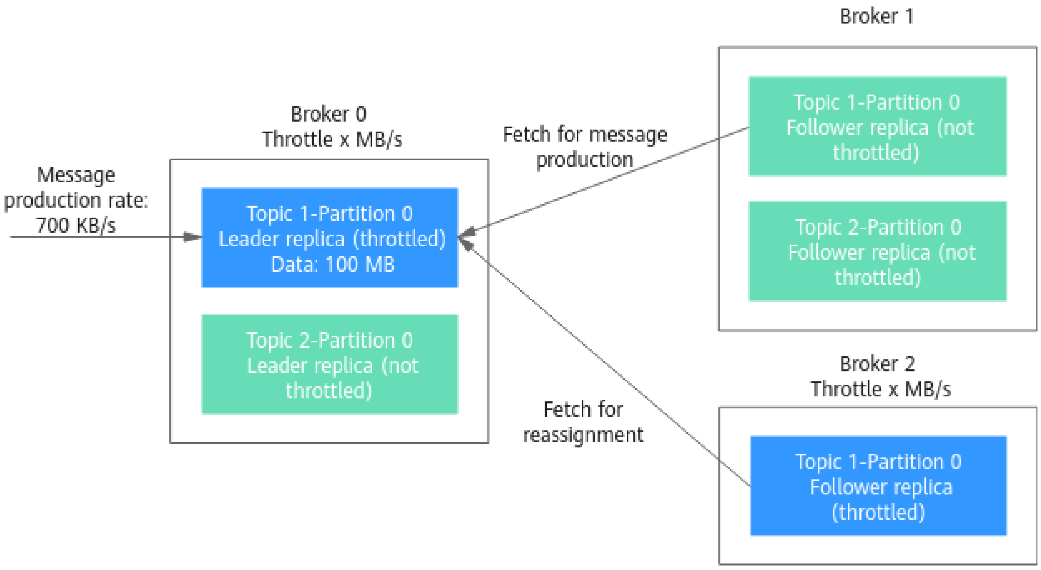
Table 4-14 Replica distribution before reassignment

Topic Name	Partition Name	Broker of Leader Replica	Broker of Follower Replica
Topic 1	0	0	0, 1
Topic 2	0	0	0, 1

Table 4-15 Replica distribution after reassignment

Topic Name	Partition Name	Broker of Leader Replica	Broker of Follower Replica
Topic 1	0	0	0, 1, 2
Topic 2	0	0	0, 1

Figure 4-16 Reassignment scenario 3



As shown in **Figure 4-16**, one replica needs to fetch data from Broker 0 for partition reassignment, and the other replica needs to fetch data from Broker 0 for message production. Since the throttle does not distinguish between message production and partition reassignment, the traffic caused by both is limited and counted.

- Bandwidth required by Broker 0 to complete partition reassignment within 200s = (100 MB + 700 KB/s x 200s)/200s + 700 KB/s= 1.9 MB/s
- Bandwidth required by Broker 2 to complete partition reassignment within 200s = 100 MB/200s = 0.5 MB/s

In conclusion, to complete the partition reassignment task within 200s, set the throttle to a value greater than or equal to 1.9 MB/s. The bandwidth should be set to be greater than or equal to 2 MB/s because the limit on it on the console must be an integer.

4.3.6 Configuring Automatic Topic Creation

Automatic Topic Creation indicates that a topic will be automatically created when a message is produced in or consumed from a topic that does not exist. By default, the topic has parameters listed in **Table 4-16**.

After you change the value of the **log.retention.hours** (retention period), **default.replication.factor** (replica quantity), or **num.partitions** (partition quantity) parameter, the value will be used in later topics that are automatically created. For example, assume that **num.partitions** is changed to 5, an automatically created topic has parameters listed in **Table 4-16**.

Table 4-16 Topic parameters

Parameter	Default Value	Modified To
Partitions	3	5
Replicas	3	3
Aging Time (h)	72	72
Synchronous Replication	Disabled	Disabled
Synchronous Flushing	Disabled	Disabled
Message Timestamp	CreateTime	CreateTime
Max. Message Size (bytes)	10,485,760	10,485,760

Procedure

Step 1 Log in to the console.

Step 2 Click 📍 in the upper left corner to select a region.

NOTE

Select the region where your Kafka instance is located.

Step 3 Click ☰ and choose **Application** > **Distributed Message Service (for Kafka)** to open the console of DMS for Kafka.

Step 4 Click the desired Kafka instance to view its details.

Step 5 In the **Instance Information** area, click ● or ○ next to **Automatic Topic Creation**. The **Confirm** dialog box is displayed.

NOTE

Enabling or disabling automatic topic creation may cause instance restarts.

Step 6 Click **OK**.

You can view the execution status of the task on the **Background Tasks** page.

----End

4.3.7 Deleting a Kafka Topic

Delete a topic using either of the following methods:

- **Deleting a Kafka Topic (Console)**
- **Deleting a Kafka Topic on the Client**

Prerequisites

- A Kafka instance has been created, and a topic has been created in this instance.
- The instance is in the **Running** state.

Constraint

If your Kafka instances are connected using Logstash, stop Logstash before deleting topics. Otherwise, services may crash.

Deleting a Kafka Topic (Console)

Step 1 Log in to the console.

Step 2 Click 📍 in the upper left corner to select a region.

NOTE

Select the region where your Kafka instance is located.

Step 3 Click ☰ and choose **Application** > **Distributed Message Service (for Kafka)** to open the console of DMS for Kafka.

Step 4 Click the desired Kafka instance to view the instance details.

Step 5 In the navigation pane, choose **Topics**.

Step 6 Delete topics using either of the following methods:

- Select one or more topics and click **Delete Topic** in the upper left corner.
- In the row containing the topic you want to delete, choose **More** > **Delete**.

Step 7 In the **Delete Topic** dialog box that is displayed, click **OK** to delete the topic.

----End

Deleting a Kafka Topic on the Client

If your Kafka client version is later than 2.2, you can use **kafka-topics.sh** to delete topics.

NOTICE

For an instance with SASL enabled, if **allow.everyone.if.no.acl.found** is set to **false**, topics cannot be deleted through the client.

- For a Kafka instance with SASL disabled, run the following command in the **/bin** directory of the Kafka client:

  ```
  ./kafka-topics.sh --bootstrap-server ${connection-address} --delete --topic ${topic-name}
  ```

  Parameter description:

  - **connection-address**: can be obtained from the **Connection** area on the **Basic Information** page on the Kafka console.
  - **topic-name**: topic name.

  Example:

  ```
  [root@ecs-kafka bin]# ./kafka-topics.sh --bootstrap-server
  192.168.xx.xx:9092,192.168.xx.xx:9092,192.168.xx.xx:9092 --delete --topic topic-01
  [root@ecs-kafka bin]#
  ```

- For a Kafka instance with SASL enabled, do as follows:

  a. (Optional) If the SSL certificate has been configured, skip this step and go to **b**. Otherwise, do as follows:

     Create the **ssl-user-config.properties** file in the **/config** directory of the Kafka client and add the SSL certificate configurations by referring to **Step 3**.

  b. Run the following command in the **/bin** directory of the Kafka client:

     ```
     ./kafka-topics.sh --bootstrap-server ${connection-address} --delete --topic ${topic-name} --
     command-config ../config/ssl-user-config.properties
     ```

     Parameter description:

     - **connection-address**: can be obtained from the **Connection** area on the **Basic Information** page on the Kafka console.
     - **topic-name**: topic name.

     Example:

     ```
     [root@ecs-kafka bin]# ./kafka-topics.sh --bootstrap-server
     192.168.xx.xx:9093,192.168.xx.xx:9093,192.168.xx.xx:9093 --delete --topic topic-01 --command-
     config ../config/ssl-user-config.properties
     [root@ecs-kafka bin]#
     ```

5 Connecting to an Instance

5.1 Configuring Kafka Network Connections

5.1.1 Kafka Network Connection Conditions

A client can connect to a Kafka instance over a public or private network. Notes before using a private network:

- By default, a client and a Kafka instance are interconnected when they are deployed in a VPC.
- If they are not, you need to interconnect them because of isolation among VPCs.

Table 5-1 lists how to access a Kafka instance on a client.

Table 5-1 Access modes

Mode	How To Do	Reference
Public access	Enable public access on the Kafka console and configure elastic IPs (EIPs). A client can connect to the Kafka instance through the EIPs.	**Configuring Kafka Public Access**
	Configure port mapping using DNAT. The client can connect to the Kafka instance in a public network.	**Accessing Kafka in a Public Network Using DNAT**
Private access	A client and a Kafka instance are interconnected when they are deployed in a VPC.	-
	When a client and a Kafka instance are deployed in different VPCs of the same region, connect the client and the Kafka instance across VPCs using a VPC endpoint.	**Accessing Kafka Using a VPC Endpoint Across VPCs**

Mode	How To Do	Reference
	When a client and a Kafka instance are deployed in different VPCs of the same region, interconnect two VPCs using a VPC peering connection.	**VPC Peering Connection**

Before accessing a Kafka instance on a client, configure the following rules in the security group of the instance.

NOTE

After a security group is created, its default inbound rule allows communication among ECSs within the security group and its default outbound rule allows all outbound traffic. In this case, you can access a Kafka instance within a VPC, and do not need to add rules according to **Table 5-2**.

Table 5-2 Security group rules

Direction	Protocol	Type	Port	Source	Description
Inbound	TCP	IPv4	9094	IP address or IP address group of the Kafka client	Accessing a Kafka instance over a public network (without SSL)
Inbound	TCP	IPv4	9092	IP address or IP address group of the Kafka client	• Accessing a Kafka instance over a private network within a VPC (without SSL) • Accessing a Kafka instance using a peering connection across VPCs (without SSL)
Inbound	TCP	IPv4	9095	IP address or IP address group of the Kafka client	Accessing a Kafka instance over a public network (with SSL)
Inbound	TCP	IPv4	9093	IP address or IP address group of the Kafka client	• Accessing a Kafka instance over a private network within a VPC (with SSL) • Accessing a Kafka instance using a peering connection across VPCs (with SSL)

Direction	Protocol	Type	Port	Source	Description
Inbound	TCP	IPv4	9011	198.19.128.0/17	Accessing a Kafka instance using a VPC endpoint across VPCs (with or without SSL)
Inbound	TCP	IPv4	9011	IP address or IP address group of the Kafka client	Accessing a Kafka instance using DNAT (with or without SSL)

5.1.2 Configuring Kafka Public Access

To access a Kafka instance over a public network, enable public access and configure EIPs for the instance. If you no longer need public access to the instance, you can disable it as required.

Prerequisites

- You can change the public access setting only when the Kafka instance is in the **Running** state.

Notes and Constraints

Kafka instances only support IPv4 EIPs. IPv6 EIPs are not supported.

Enabling Public IPv4 Access

Step 1 Log in to the console.

Step 2 Click 📍 in the upper left corner to select a region.

NOTE

Select the region where your Kafka instance is located.

Step 3 Click ☰ and choose **Application** > **Distributed Message Service (for Kafka)** to open the console of DMS for Kafka.

Step 4 Click a Kafka instance to go to the **Basic Information** page.

Step 5 Click ○ next to **Public Access** to enable public access. For **Elastic IP Address**, select an EIP for each broker.

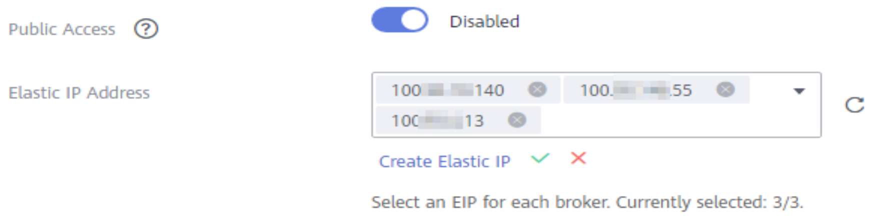
If the EIPs are insufficient, do as follows to set them.

1. Click **Create Elastic IP** to go to the **Buy EIP** page and purchase EIPs. For details, see **Assigning an EIP**.
2. After the purchase is complete, return to the public access enabling page.

3. Click ⟳ after **Elastic IP Address**, select an EIP for each broker and then click ✓.

4. You can view the operation progress on the **Background Tasks** page. If the task status is **Successful**, the modification has succeeded.

Figure 5-1 Enabling public access



Public Access ?

Disabled

Elastic IP Address

100[redacted]140 ⊗ 100.[redacted]55 ⊗ 100[redacted]13 ⊗

Create Elastic IP ✓ ✕

Select an EIP for each broker. Currently selected: 3/3.

After public access is enabled, configure security group rules listed in **Table 5-3** before attempting to access Kafka. For details about accessing Kafka, see **Connecting to an Instance**.

Table 5-3 Kafka instance security group rules (public IPv4 access)

Direction	Protocol	Type	Port	Source	Description
Inbound	TCP	IPv4	9094	IP address or IP address group of the Kafka client	Accessing Kafka over a public network (without SSL)
Inbound	TCP	IPv4	9095	IP address or IP address group of the Kafka client	Accessing Kafka over a public network (with SSL)

----End

Disabling Public IPv4 Access

Step 1 Log in to the console.

Step 2 Click ⌖ in the upper left corner to select a region.

NOTE

Select the region where your Kafka instance is located.

Step 3 Click ☰ and choose **Application** > **Distributed Message Service (for Kafka)** to open the console of DMS for Kafka.

Step 4 Click a Kafka instance to go to the **Basic Information** page.

Step 5 Click next to **Public Access**.

You can view the operation progress on the **Background Tasks** page. If the task status is **Successful**, the modification has succeeded.

After public access is disabled, configure security group rules listed in **Table 5-4** before attempting to access Kafka in a VPC. For details about accessing Kafka, see **Connecting to an Instance**.

Table 5-4 Kafka instance security group rules (private access)

Direction	Protocol	Type	Port	Source	Description
Inbound	TCP	IPv4	9092	IP address or IP address group of the Kafka client	Accessing a Kafka instance over a private network within a VPC (without SSL)
Inbound	TCP	IPv4	9093	IP address or IP address group of the Kafka client	Accessing a Kafka instance over a private network within a VPC (with SSL)

NOTE

After a security group is created, its default inbound rule allows communication among ECSs within the security group and its default outbound rule allows all outbound traffic. In this case, you can access a Kafka instance within a VPC, and do not need to add rules according to **Table 5-4**.

----End

5.1.3 Accessing Kafka Using a VPC Endpoint Across VPCs

VPCs are logically isolated from each other. If a Kafka instance and a Kafka client are in different VPCs within a region, they cannot communicate with each other. In this case, you can use one of the following methods to access a Kafka instance across VPCs:

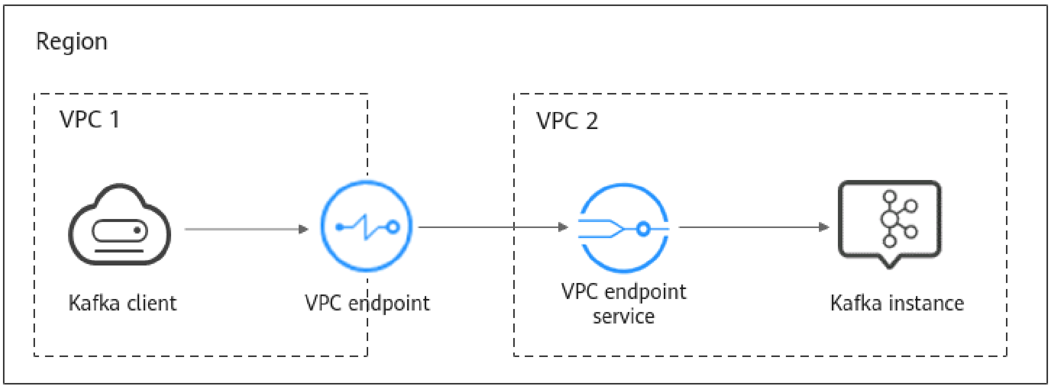
- Establish a VPC peering connection to allow two VPCs to communicate with each other. For details, see **VPC Peering Connection**.
- Use VPC Endpoint (VPCEP) to establish a cross-VPC connection.

The following describes how to use VPCEP to implement cross-VPC access.

VPCEP provides two types of resources: VPC endpoint services and VPC endpoints.

- A VPC endpoint service can be a Kafka instance which is accessed using VPC endpoints.
- A VPC endpoint is a secure and private channel for connecting a VPC to a VPC endpoint service.

Figure 5-2 Working principle of accessing a Kafka instance across VPCs



Creating a VPC Endpoint Service

Step 1 Log in to the console.

Step 2 Click in the upper left corner to select a region.

NOTE

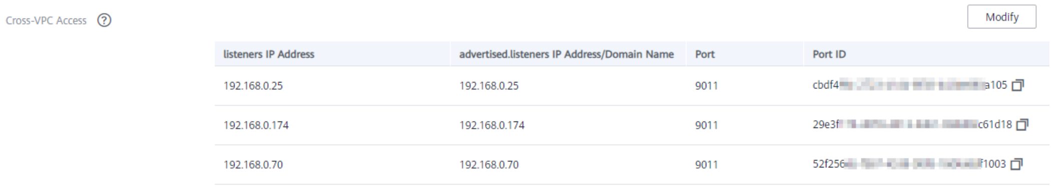
Select the region where your Kafka instance is located.

Step 3 Click ☰ in the upper left corner and choose **Application** > **Distributed Message Service (for Kafka)Application** > to open the console of DMS for Kafka.

Step 4 Click the desired Kafka instance to view its details.

Step 5 In the **Advanced Settings** section on the **Basic Information** tab page, obtain the listeners IP addresses and port IDs of the instance for **Cross-VPC Access**.

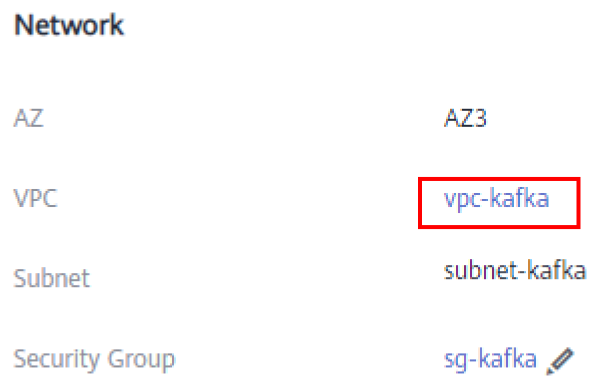
Figure 5-3 Cross-VPC access–related listeners IP addresses and corresponding port IDs of the Kafka instance



listeners IP Address	advertised.listeners IP Address/Domain Name	Port	Port ID
192.168.0.25	192.168.0.25	9011	cbdf4[redacted]a105
192.168.0.174	192.168.0.174	9011	29e3f[redacted]c61d18
192.168.0.70	192.168.0.70	9011	52f256[redacted]f1003

Step 6 In the **Network** section on the **Basic Information** tab page, view the VPC to which the Kafka instance belongs.

Figure 5-4 Viewing the VPC to which the Kafka instance belongs



Network

AZ: AZ3

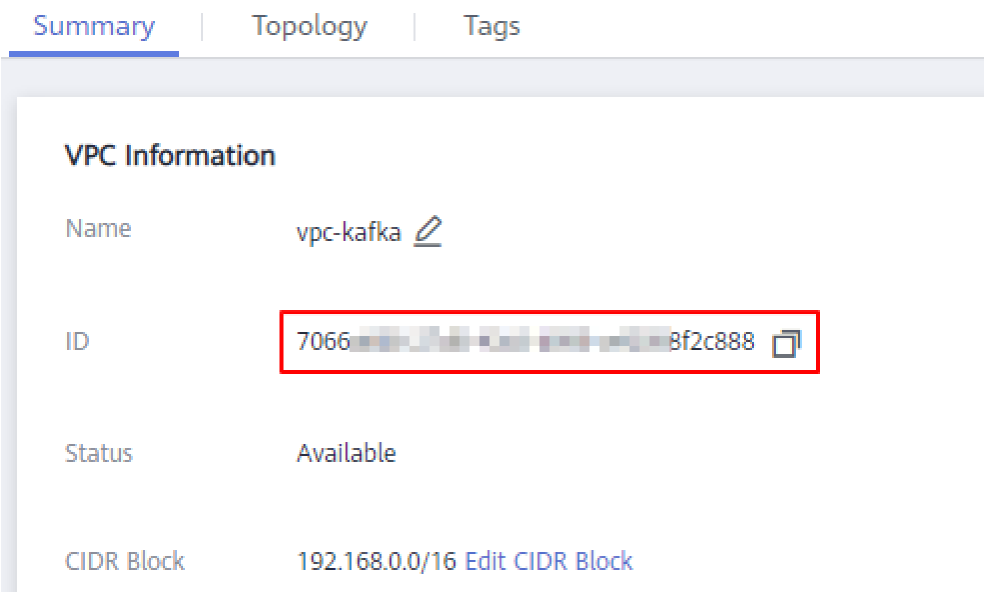
VPC: vpc-kafka

Subnet: subnet-kafka

Security Group: sg-kafka

Step 7 Click the VPC to obtain the VPC ID on the VPC console.

Figure 5-5 Obtaining the VPC ID



VPC Information

Name: vpc-kafka

ID: 7066[redacted]8f2c888

Status: Available

CIDR Block: 192.168.0.0/16 Edit CIDR Block

Step 8 Call the VPC Endpoint API to create a VPC endpoint service. For details, see **Creating a VPC Endpoint Service**.

```
POST https://{endpoint}/v1/{project_id}/vpc-endpoint-services
```

Set the following request parameter to the specified values, and other parameters as required.

- **port_id**: one of the port IDs obtained in **Step 5**.
- **vpc_id**: VPC ID obtained in **Step 7**.
- **server_type**: **VM**
- **client_port**: **9011**
- **server_port**: **9011**
- **protocol**: **TCP**
- **approval_enabled**: **false**
- **service_type**: **interface**

- **endpoint**: VPCEP endpoint obtained from Regions and Endpoints. The region must be the same as that of the Kafka instance.
- **project_id**: project ID obtained from **Obtaining a Project ID**. The region must be the same as that of the Kafka instance.

Record the value of **service_name** in the response. This parameter indicates the name of the VPC endpoint service.

Step 9 Repeat **Step 8** to create VPC endpoint services for other port IDs obtained in **Step 5** and record the VPC endpoint service names.

----End

(Optional) Adding a Whitelist

The VPC endpoint service can be used across accounts through a whitelist.

If the Kafka client and Kafka instance belong to different accounts, add the ID of the account to which the Kafka client belongs to the whitelist of the endpoint service. For details, see **Add a Whitelist Record**.

Buying a VPC Endpoint

Step 1 Click ☰ in the upper left corner of the management console. Then choose **Network** > **VPC Endpoint**.

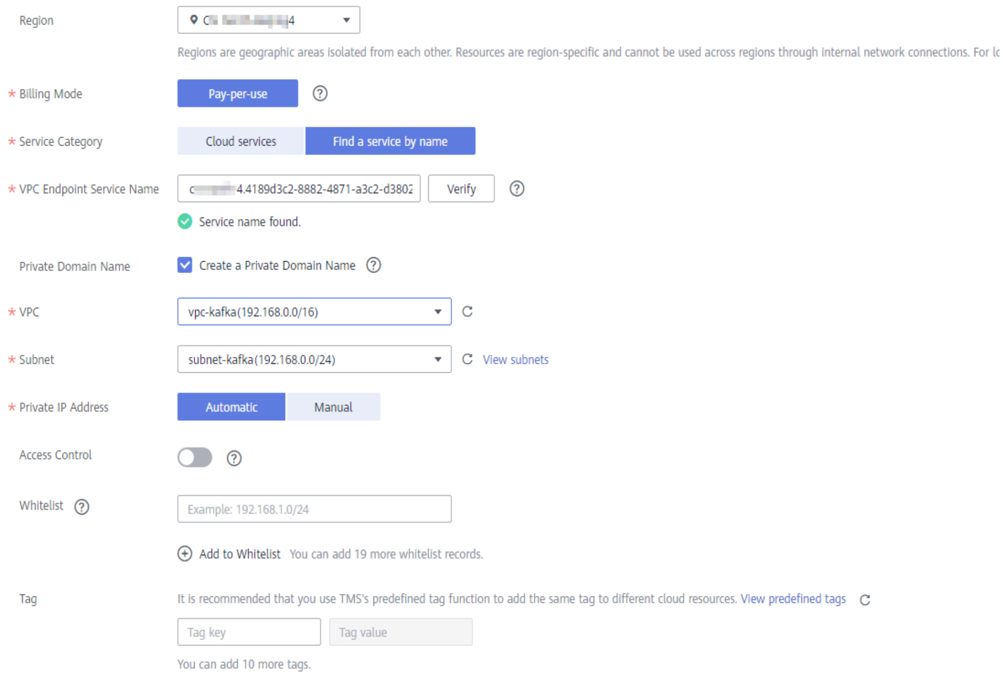
Step 2 Click **Buy VPC Endpoint**.

Step 3 Set the following parameters:

- **Region**: Select the region that the Kafka instance is in.
- **Service Category**: Select **Find a service by name**.
- **VPC Endpoint Service Name**: Enter the VPC endpoint service name recorded in **Step 8** and click **Verify**. If **Service name found** is displayed, proceed with subsequent operations.
- **VPC**: Select the VPC that the Kafka client is in.
- **Subnet**: Select the subnet that the Kafka client is in.
- **Private IP Address**: Select **Automatic**.

Retain the default values for other parameters. For details, see **Buying a VPC Endpoint**.

Figure 5-6 VPC endpoint parameters



Region: C[redacted]4
Regions are geographic areas isolated from each other. Resources are region-specific and cannot be used across regions through internal network connections. For lc

* Billing Mode: Pay-per-use

* Service Category: Cloud services | Find a service by name

* VPC Endpoint Service Name: c[redacted]4.4189d3c2-8882-4871-a3c2-d3802 Verify
Service name found.

Private Domain Name: ☑ Create a Private Domain Name

* VPC: vpc-kafka(192.168.0.0/16)

* Subnet: subnet-kafka(192.168.0.0/24) View subnets

* Private IP Address: Automatic | Manual

Access Control: ○

Whitelist: Example: 192.168.1.0/24
Add to Whitelist You can add 19 more whitelist records.

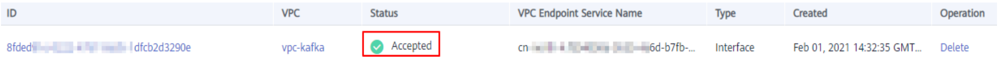
Tag: It is recommended that you use TMS's predefined tag function to add the same tag to different cloud resources. View predefined tags
Tag key | Tag value
You can add 10 more tags.

Step 4 Click **Next**.

Step 5 Confirm the configurations and submit the request.

Step 6 Go back to the VPC endpoint list and check whether the status of the created VPC endpoint has changed to **Accepted**. The **Accepted** state means that the VPC endpoint has been connected to the VPC endpoint service.

Figure 5-7 Checking the VPC endpoint status

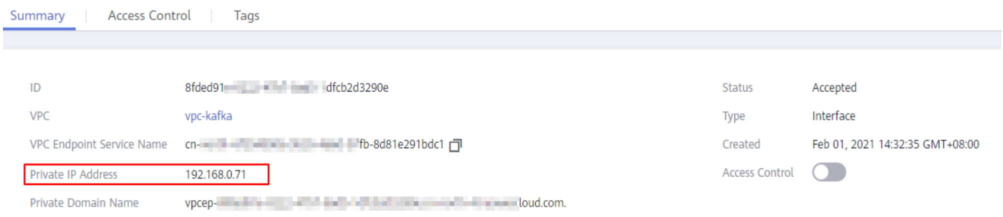


ID	VPC	Status	VPC Endpoint Service Name	Type	Created	Operation
8fded[redacted]dfcb2d3290e	vpc-kafka	Accepted	cn[redacted]6d-b7fb-...	Interface	Feb 01, 2021 14:32:35 GMT...	Delete

Step 7 Click the VPC endpoint ID. On the **Summary** tab page, obtain the private IP address.

You can use the private IP address to access the VPC endpoint service.

Figure 5-8 Viewing the private IP address



Summary | Access Control | Tags

ID	8fded91[redacted]dfcb2d3290e	Status	Accepted
VPC	vpc-kafka	Type	Interface
VPC Endpoint Service Name	cn-[redacted]fb-8d81e291bdc1	Created	Feb 01, 2021 14:32:35 GMT+08:00
Private IP Address	192.168.0.71	Access Control	○
Private Domain Name	vpcep-[redacted]loud.com.		

Step 8 Repeat **Step 1** to **Step 7** to buy a VPC endpoint for each VPC endpoint service created in **Step 9**, and view and record the private IP addresses of the VPC endpoint services.

----End

Modifying Parameter advertised.listeners IP

Step 1 Click ☰ in the upper left corner and choose **Application** > **Distributed Message Service (for Kafka)** to open the console of DMS for Kafka.

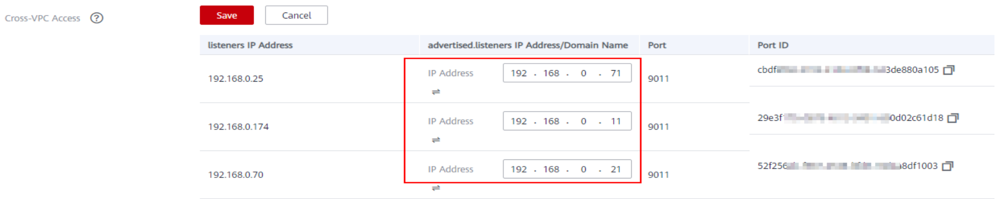
Step 2 Click the desired Kafka instance to view its details.

Step 3 On the **Advanced Settings** section of the **Basic Information** tab page, click **Modify** for **Cross-VPC Access** to change the value of **advertised.listeners IP address** to the private IP addresses recorded in **Step 7** and **Step 8**. Click **Save**.

NOTICE

Each IP address must match the corresponding port ID. Otherwise, the network will be disconnected.

Figure 5-9 Changing the advertised.listeners IP addresses



listeners IP Address	advertised.listeners IP Address/Domain Name	Port	Port ID
192.168.0.25	IP Address 192 . 168 . 0 . 71 ⇌	9011	cbdf[redacted]3de880a105
192.168.0.174	IP Address 192 . 168 . 0 . 11 ⇌	9011	29e3f[redacted]0d02c61d18
192.168.0.70	IP Address 192 . 168 . 0 . 21 ⇌	9011	52f256[redacted]8df1003

----End

Verifying Connectivity

Check whether messages can be created and retrieved by referring to **Connecting to Kafka Using the Client (SASL Disabled)** or **Connecting to Kafka Using the Client (SASL Enabled)**.

Notes:

- The address for connecting to a Kafka instance is in the format of "*advertised.listeners IP*:**9011**". For example, the addresses for connecting to the Kafka instance shown in **Figure 5-9** are **192.168.0.71:9011,192.168.0.11:9011,192.168.0.21:9011**.
- Configure inbound rules for the security group of the Kafka instance to allow access from **198.19.128.0/17** over port **9011**.
- If a network access control list (ACL) has been configured for the subnet of this instance, configure inbound rules for the network ACL to allow access from **198.19.128.0/17** and from the subnet used by the VPC endpoint.

NOTE

198.19.128.0/17 is the network segment allocated to the VPCEP service. To use VPCEP, allow access from this network segment.

5.1.4 Accessing Kafka in a Public Network Using DNAT

Enable public access in either of the following ways:

- On the Kafka console, access Kafka instances using EIPs. For details, see **Configuring Kafka Public Access**.
- Configure port mapping from EIPs to specified instance ports using destination NAT (DNAT).

This section describes how to access Kafka over a public network using DNAT.

Prerequisites

You have purchased EIPs of a quantity equal to the number of brokers in the Kafka instance. For details about how to purchase an EIP, see **Assigning an EIP**.

Step 1: Obtain Information About the Kafka Instance

Step 1 Log in to the console.

Step 2 Click ⊙ in the upper left corner to select a region.

NOTE

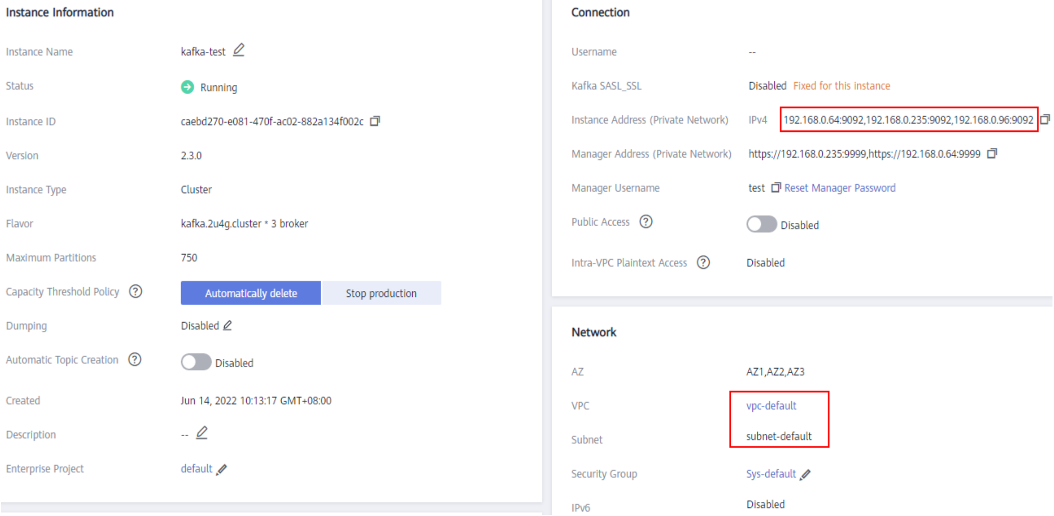
Select the region where your Kafka instance is located.

Step 3 Click ☰ in the upper left corner and choose **Application** > **Distributed Message Service (for Kafka)** to open the console of DMS for Kafka.

Step 4 Click the desired Kafka instance to view its details.

Step 5 In the **Connection** area on the **Basic Information** tab page, view and record the private network access addresses of the Kafka instance. In the **Network** area, view and record the VPC and subnet where the Kafka instance is located.

Figure 5-10 Kafka instance information



Instance Information

Instance Name	kafka-test
Status	Running
Instance ID	caebd270-e081-470f-ac02-882a134f002c
Version	2.3.0
Instance Type	Cluster
Flavor	kafka.2u4g.cluster * 3 broker
Maximum Partitions	750
Capacity Threshold Policy	Automatically delete / Stop production
Dumping	Disabled
Automatic Topic Creation	Disabled
Created	Jun 14, 2022 10:13:17 GMT+08:00
Description	--
Enterprise Project	default

Connection

Username	--
Kafka SASL_SSL	Disabled Fixed for this instance
Instance Address (Private Network)	IPv4 192.168.0.64:9092,192.168.0.235:9092,192.168.0.96:9092
Manager Address (Private Network)	https://192.168.0.235:9999,https://192.168.0.64:9999
Manager Username	test Reset Manager Password
Public Access	Disabled
Intra-VPC Plaintext Access	Disabled

Network

AZ	AZ1,AZ2,AZ3
VPC	vpc-default
Subnet	subnet-default
Security Group	Sys-default
IPv6	Disabled

----End

Step 2: Buy a Public NAT Gateway

Step 1 Click ☰ in the upper left corner of the management console and choose **Network** > **NAT Gateway**. The **Public NAT Gateways** page is displayed.

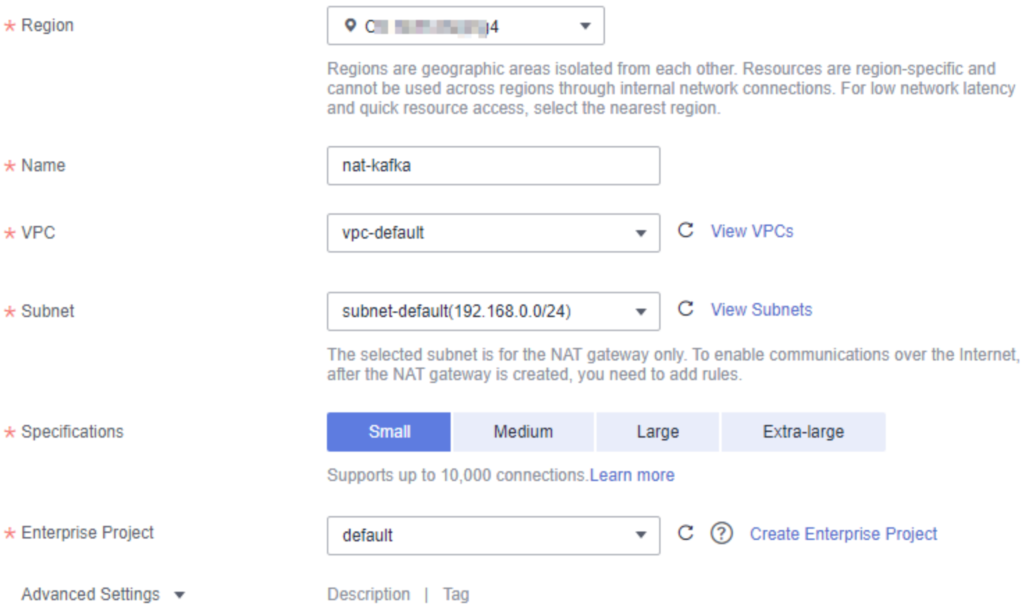
Step 2 Click **Buy Public NAT Gateway**.

Step 3 Set the following parameters:

- **Region**: Select the region that the Kafka instance is in.
- **Name**: Enter a name for the public NAT gateway.
- **VPC**: Select the VPC recorded in **Step 5**.
- **Subnet**: Select the subnet recorded in **Step 5**.
- **Enterprise Project**: Select an enterprise project as required.

Set other parameters as required. For details, see **Creating a Public NAT Gateway**.

Figure 5-11 Buying a public NAT gateway



* Region: C[redacted]4

Regions are geographic areas isolated from each other. Resources are region-specific and cannot be used across regions through internal network connections. For low network latency and quick resource access, select the nearest region.

* Name: nat-kafka

* VPC: vpc-default

* Subnet: subnet-default(192.168.0.0/24)

The selected subnet is for the NAT gateway only. To enable communications over the Internet, after the NAT gateway is created, you need to add rules.

* Specifications: Small

Supports up to 10,000 connections.Learn more

* Enterprise Project: default

Step 4 Click **Next**.

Step 5 Confirm the specifications and click **Submit**.

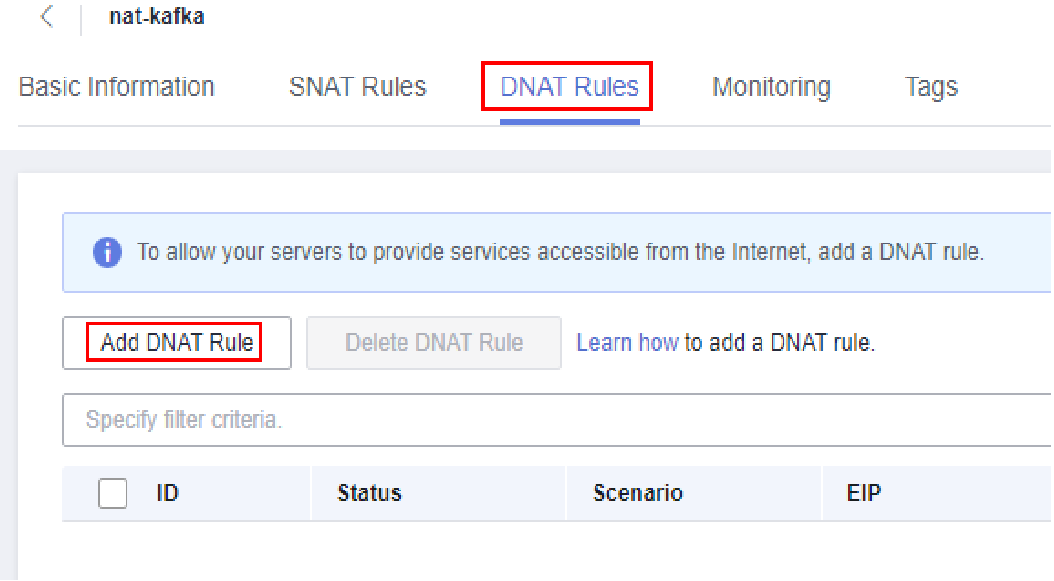
----End

Step 3: Add a DNAT Rule

Step 1 On **Public NAT Gateways** page, locate the row containing the newly purchased public NAT gateway and click **Configure Rules** in the **Operation** column.

Step 2 On the **DNAT Rules** tab page, click **Add DNAT Rule**.

Figure 5-12 Public NAT gateway details



To allow your servers to provide services accessible from the Internet, add a DNAT rule.

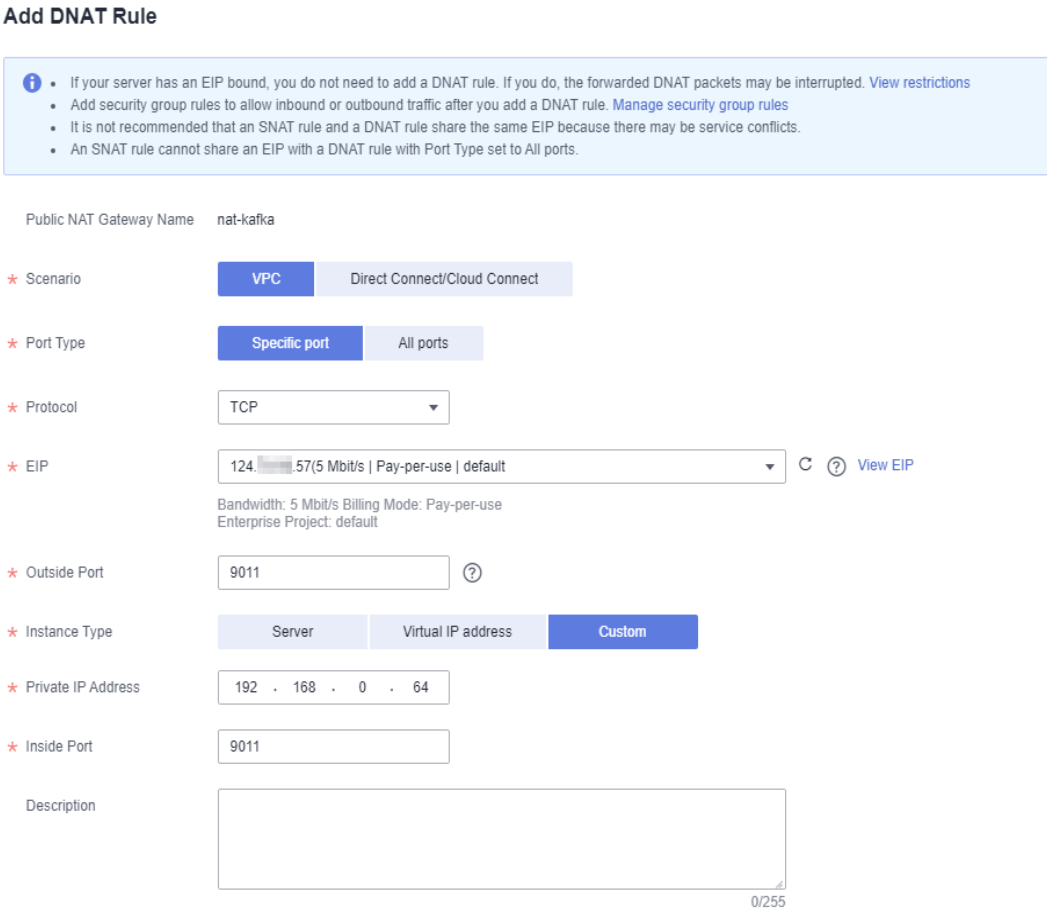
Learn how to add a DNAT rule.

Step 3 Set the following parameters:

- **Scenario**: Select **VPC**.
- **Port Type**: Select **Specific port**.
- **Protocol**: Select **TCP**.
- **EIP**: Select an EIP.
- **Outside Port**: Enter **9011**.
- **Instance Type**: Select **Custom**.
- **Private IP Address**: Enter one of the private network addresses of the Kafka instance recorded in **Step 5**.
- **Inside Port**: Enter **9011**.

For details about more parameters, see **Adding a DNAT Rule**.

Figure 5-13 Adding a DNAT rule



Add DNAT Rule

- If your server has an EIP bound, you do not need to add a DNAT rule. If you do, the forwarded DNAT packets may be interrupted. View restrictions
- Add security group rules to allow inbound or outbound traffic after you add a DNAT rule. Manage security group rules
- It is not recommended that an SNAT rule and a DNAT rule share the same EIP because there may be service conflicts.
- An SNAT rule cannot share an EIP with a DNAT rule with Port Type set to All ports.

Public NAT Gateway Name: nat-kafka

* Scenario: VPC | Direct Connect/Cloud Connect

* Port Type: Specific port | All ports

* Protocol: TCP

* EIP: 124.[redacted].57(5 Mbit/s | Pay-per-use | default

Bandwidth: 5 Mbit/s Billing Mode: Pay-per-use
Enterprise Project: default

* Outside Port: 9011

* Instance Type: Server | Virtual IP address | Custom

* Private IP Address: 192 . 168 . 0 . 64

* Inside Port: 9011

Description:

0/255

Step 4 Click **OK**.

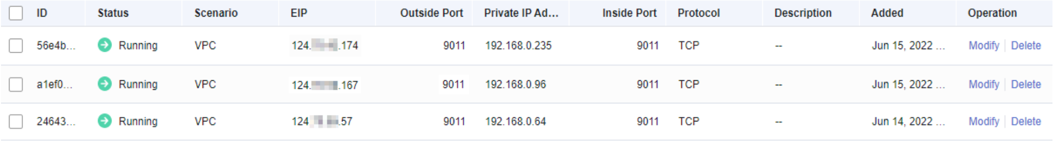
View the DNAT rule status in the DNAT rule list. If **Status** is **Running**, the rule has been added successfully.

Step 5 Create DNAT rules for other private network addresses of the Kafka instance recorded in **Step 5**. **Configure a unique EIP for each DNAT rule.**

For details about how to create a DNAT rule, see **Step 2** to **Step 4**.

Step 6 After all DNAT rules are created, click the **DNAT Rules** tab to view the created DNAT rules and record the EIPs corresponding to the private IP addresses.

Figure 5-14 DNAT rule list



□	ID	Status	Scenario	EIP	Outside Port	Private IP Ad...	Inside Port	Protocol	Description	Added	Operation
□	56e4b...	Running	VPC	124.[redacted].174	9011	192.168.0.235	9011	TCP	--	Jun 15, 2022 ...	Modify | Delete
□	a1ef0...	Running	VPC	124.[redacted].167	9011	192.168.0.96	9011	TCP	--	Jun 15, 2022 ...	Modify | Delete
□	24643...	Running	VPC	124[redacted].57	9011	192.168.0.64	9011	TCP	--	Jun 14, 2022 ...	Modify | Delete

----End

Step 4: Map EIPs to the Port 9011 of Private IP Addresses

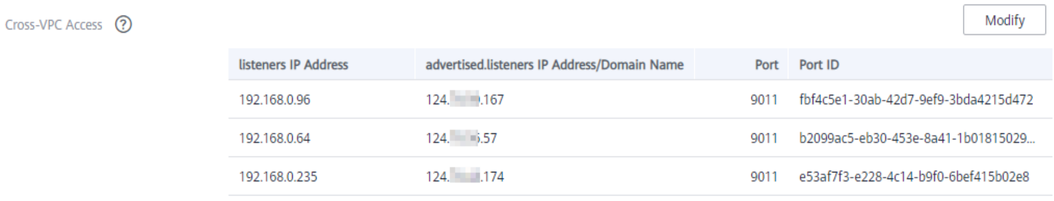
Step 1 Click ☰ in the upper left corner and choose **Application** > **Distributed Message Service (for Kafka)** to open the console of DMS for Kafka.

Step 2 Click the desired Kafka instance to view its details.

Step 3 In the **Advanced Settings** section on the **Basic Information** tab page, click **Modify** next to **Cross-VPC Access**.

Step 4 Change the values of **advertised.listeners IP Address/Domain Name** to the EIPs in the DNAT rules. Ensure that the mapping between the private network addresses and the EIPs is consistent with that recorded in **Step 6**. Then click **Save**.

Figure 5-15 Changing the advertised.listeners IP address (for DNAT access)



listeners IP Address	advertised.listeners IP Address/Domain Name	Port	Port ID
192.168.0.96	124.[redacted].167	9011	fbf4c5e1-30ab-42d7-9ef9-3bda4215d472
192.168.0.64	124.[redacted].57	9011	b2099ac5-eb30-453e-8a41-1b01815029...
192.168.0.235	124.[redacted].174	9011	e53af7f3-e228-4c14-b9f0-6bef415b02e8

----End

Step 5: Verify Connectivity

Check whether messages can be created and retrieved by referring to **Connecting to Kafka Using the Client (SASL Disabled)** or **Connecting to Kafka Using the Client (SASL Enabled)**.

Notes:

- The address for connecting to a Kafka instance is in the format of "*advertised.listeners IP*:**9011**". For example, the addresses for connecting to the Kafka instance shown in **Figure 5-15** are **124.*xxx.xxx*.167:9011,124.*xxx.xxx*.174:9011,124.*xxx.xxx*.57:9011**.
- Configure security group rules for the Kafka instance to allow inbound access over port **9011**.
- Public access must be enabled on the client connected to the Kafka instance.

5.2 Configuring Kafka Access Control

5.2.1 Converting a JKS SSL Certificate to a PEM One

This section describes how to convert an SSL certificate from the JKS format to the PEM format.

Prerequisites

- A Linux server is available. The server must install **Java Development Kit 1.8.111 or later** and **JAVA_HOME** and **PATH** environment variables are configured.

- Kafka SASL_SSL has been enabled for the instance.

Generating a PEM SSL Certificate

Step 1 Log in to the console.

Step 2 Click 📍 in the upper left corner to select a region.

NOTE

Select the region where your Kafka instance is in.

Step 3 Click ☰ and choose **Application** > **Distributed Message Service (for Kafka)** to open the console of DMS for Kafka.

Step 4 Click a Kafka instance name to go to the instance details page.

Step 5 Click **Download** next to **Connection** > **SSL Certificate**.

Step 6 Decompress the Zip package to obtain the JKS SSL certificate.

Step 7 Log in to the Linux server and upload the JKS SSL certificate to it.

Step 8 Run the following command to convert the certificate from JKS to the PKCS12 format:

```
keytool -importkeystore -srckeystore client.jks -destkeystore client.p12 -deststoretype PKCS12
```

1. Enter a PKCS12 certificate password when **Enter destination keystore password** is prompted. Remember the password for later use.
2. Enter a JKS certificate password when **Enter source keystore password** is prompted. Here, enter **dms@kafka**.

Example:

```
[root@ecs-test ~]# keytool -importkeystore -srckeystore client.jks -destkeystore client.p12 -deststoretype
PKCS12
Importing keystore client.jks to client.p12...
Enter destination keystore password:
Re-enter new password:
Enter source keystore password:
Entry for alias clientpublickey successfully imported.
Entry for alias huaweiequipmentrootca successfully imported.
Entry for alias huaweicloudca successfully imported.
Import command completed:  3 entries successfully imported, 0 entries failed or cancelled
```

Step 9 Run the following command to check the generated PKCS12 certificate **client.p12**:

```
ll
```

Example:

```
[root@ecs-test ~]# ll
-rw-r--r-- 1 root  root      3561 Jun 20 17:06 client.jks
-rw-r--r-- 1 root  root      4023 Oct 10 20:06 client.p12
```

Step 10 Run the following command to convert the certificate format from PKCS12 to PEM.

```
openssl pkcs12 -in client.p12 -out client.pem
```

Enter the PKCS12 certificate password set in **Step 8.1** as prompted.

Example:

```
[root@ecs-test ~]# openssl pkcs12 -in client.p12 -out client.pem
Enter Import Password:
```

Step 11 Run the following command to check the generated PEM certificate **client.pem**.

```
ll
```

Example:

```
[root@ecs-test ~]# ll
-rw-r--r-- 1 root  root      3561 Jun 20 17:06 client.jks
-rw-r--r-- 1 root  root      4023 Oct 10 20:06 client.p12
-rw------- 1 root  root       5384 Oct 10 20:24 client.pem
```

----End

Accessing a Kafka Instance Using a PEM Certificate

The following section demonstrates how to access a Kafka instance using a PEM certificate on a Java client.

Access a Kafka instance to produce and consume messages by referring to **Configuring Kafka Clients in Java**. Modify the SASL setting of the message production and consumption configuration files as follows:

```
# If the SASL mechanism is PLAIN, configure as follows:
sasl.mechanism=PLAIN
sasl.jaas.config=org.apache.kafka.common.security.plain.PlainLoginModule required \
username="username" \
password="password";

# If the SASL mechanism is SCRAM-SHA-512, configure as follows:
sasl.mechanism=SCRAM-SHA-512
sasl.jaas.config=org.apache.kafka.common.security.scram.ScramLoginModule required \
username="username" \
password="password";

#Set the Kafka security protocol.
security.protocol=SASL_SSL
# ssl truststore.location is the path for storing the SSL certificate. The following code uses the path format
in Windows as an example. Change the path format based on the actual running environment.
ssl.truststore.location=E:\\temp\\client.pem
# ssl.truststore.password is the server certificate password. To access a Kafka instance using a PEM
certificate, skip this parameter.
#ssl.truststore.password=dms@kafka
# ssl.endpoint.identification.algorithm indicates whether to verify the certificate domain name. This
parameter must be left blank, which indicates disabling domain name verification.
ssl.endpoint.identification.algorithm=
# Add the ssl.truststore.type parameter to specify the client certificate type to PEM.
ssl.truststore.type=PEM
```

5.2.2 Configuring Kafka ACL Users

Kafka instances with SASL_SSL enabled support access control list (ACL) for topics. You can differentiate user permissions by granting users different permissions in a topic.

This section describes how to create users, reset the password, and delete users with SASL_SSL enabled. For details about how to grant topic permissions for users, see **Configuring Kafka Topic Permissions**.

The maximum number of users that can be created for a Kafka instance is 20 or 500. Check the console for the actual limit.

There are two ways to create a user on the console. Accordingly, there are two ways to reset the user's password:

- Initial user: The user set in instance creation. If you forget your password, reset it by referring to **Resetting the Password (for the Initial User)**.
- Non-initial users: Users created on the **Users** page. If you forget your password, reset it by referring to **Resetting the Password (for Non-initial Users)**.

Before Resetting a Password

Resetting a user password will interrupt services. Change the user password in the client configuration file or code as soon as possible.

Constraints

- The initial user set in Kafka instance creation cannot be deleted.

Prerequisites

- SASL_SSL has been enabled when you create the Kafka instance.
- Kafka users can be configured only for Kafka instances in the **Running** state.

Creating a User

Step 1 Log in to the console.

Step 2 Click 📍 in the upper left corner to select a region.

NOTE

Select the region where your Kafka instance is located.

Step 3 Click ☰ and choose **Application** > **Distributed Message Service (for Kafka)** to open the console of DMS for Kafka.

Step 4 Click the desired Kafka instance to go to the instance details page.

Step 5 On the **Users** page, click **Create User**.

Step 6 Set user information by referring to **Configuring Kafka ACL Users**.

Table 5-5 User creation parameters

Parameter	Description
Username	The username used to access a Kafka instance, you can customize a name that complies with the rules: 4–64 characters; starts with a letter; can contain only letters, digits, hyphens (-), and underscores (_).

<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td>Password</td><td>The password used to access a Kafka instance. A password must meet the following requirements:
● Contains 8 to 32 characters.
● Contains at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters `~! @#$ %^&*()-_=+\|[{}];:'",<.>? and spaces, and cannot start with a hyphen (-).
● Cannot be the username spelled forward or backward.</td></tr>
</table>

Step 7 Click **OK**.

After the user is created, grant permissions to the user by referring to **Configuring Kafka Topic Permissions**.

----End

Resetting the Password (for the Initial User)

Step 1 Log in to the console.

Step 2 Click 📍 in the upper left corner to select a region.

NOTE

Select the region where your Kafka instance is located.

Step 3 Click ☰ and choose **Application** > **Distributed Message Service (for Kafka)** to open the console of DMS for Kafka.

Step 4 Reset the password for the initial user in either of the following ways.

- Choose **More** > **Reset Kafka Password** in the row containing the desired Kafka instance.
- Click the desired Kafka instance to go to the instance details page. Choose **More** > **Reset Kafka Password** in the upper left corner.
- Click the desired Kafka instance to go to the instance details page. On the **Basic Information** page, click **Reset Password** next to **Username** in the **Connection** section.
- Click the desired Kafka instance to go to the instance details page. On the **Users** page, click **Reset Password** in the row containing the desired user.

Step 5 Enter and confirm a new password, and click **OK**.

- If the password is successfully reset, a success message is displayed.
- If the password fails to be reset, a failure message is displayed. In this case, reset the password again. If you still fail to reset the password after multiple attempts, contact customer service.

NOTE

The system will display a success message only after the password is successfully reset on all brokers.

----End

Resetting the Password (for Non-initial Users)

Step 1 Log in to the console.

Step 2 Click 📍 in the upper left corner to select a region.

NOTE

Select the region where your Kafka instance is located.

Step 3 Click ☰ and choose **Application** > **Distributed Message Service (for Kafka)** to open the console of DMS for Kafka.

Step 4 Click the desired Kafka instance to go to the instance details page.

Step 5 On the **Users** page, click **Reset Password** in the row containing the desired user.

Step 6 Enter and confirm a new password, and click **OK**.

- If the password is successfully reset, a success message is displayed.
- If the password fails to be reset, a failure message is displayed. In this case, reset the password again. If you still fail to reset the password after multiple attempts, contact customer service.

NOTE

The system will display a success message only after the password is successfully reset on all brokers.

----End

Deleting a User

Step 1 Log in to the console.

Step 2 Click 📍 in the upper left corner to select a region.

NOTE

Select the region where your Kafka instance is located.

Step 3 Click ☰ and choose **Application** > **Distributed Message Service (for Kafka)** to open the console of DMS for Kafka.

Step 4 Click the desired Kafka instance to go to the instance details page.

Step 5 In the navigation pane, choose **Users**.

Step 6 Delete a user in either of the following ways:

- In the row containing the desired user, click **Delete**.

- Select one or more users and click **Delete** above the list.

Step 7 In the displayed **Delete User** dialog box, click **OK** to delete the user.

----End

5.3 Configuring the Kafka Client

5.3.1 Setting Parameters for Kafka Clients

This section provides recommendations on configuring common parameters for Kafka producers and consumers. Kafka clients in different versions may have different parameter names. The following parameters are supported in v1.1.0 and later. For details about other parameters and versions, see **Kafka Configuration**.

Table 5-6 Producer parameters

Parameter	Default Value	Recommended Value	Description
acks	1	**all** or **–1** (if high reliability mode is selected) **1** (if high throughput mode is selected)	Number of acknowledgments the producer requires the server to return before considering a request complete. This controls the durability of records that are sent. The value of this parameter can be any of the following: **0**: The producer will not wait for any acknowledgment from the server at all. The record will be immediately added to the socket buffer and considered sent. No guarantee can be made that the server has received the record, and the retries configuration will not take effect (as the client generally does not know of any failures). The offset given back for each record will always be set to –1. **1**: The leader will write the record to its local log but will respond without waiting until receiving full acknowledgement from all followers. If the leader fails immediately after acknowledging the record but before the followers have replicated it, the record will be lost. **all** or **-1**: The leader needs to wait until all backups in the ISR are written into logs. As long as any backup survives, data will not be lost. **min.insync.replicas** specifies the minimum number of replicas that must acknowledge a write for the write to be considered successful.

Parameter	Default Value	Recommended Value	Description
retries	0	/	Number of times that the client resends a message. Setting this parameter to a value greater than zero will cause the client to resend any record that failed to be sent. Note that this retry is no different than if the client re-sent the record upon receiving the error. Allowing retries will potentially change the ordering of records because if two batches are sent to the same partition, and the first fails and is retried but the second succeeds, then the records in the second batch may appear first. You are advised to configure producers so that they can be able to retry in case of network disconnections. Set **retries** to **3** and the retry interval **retry.backoff.ms** to **1000**.
request.timeout.ms	30000	/	Maximum amount of time (in ms) the client will wait for the response of a request. If the response is not received before the timeout elapses, the client will throw a timeout exception. Setting this parameter to a large value, for example, **127000** (127s), can prevent records from failing to be sent in high-concurrency scenarios.
block.on.buffer.full	TRUE	TRUE	Setting this parameter to **TRUE** indicates that when buffer memory is exhausted, the producer must stop receiving new message records or throw an exception. By default, this parameter is set to **TRUE**. However, in some cases, non-blocking usage is desired and it is better to throw an exception immediately. Setting this parameter to **FALSE** will cause the producer to instead throw "BufferExhaustedException" when buffer memory is exhausted.

Parameter	Default Value	Recommended Value	Description
batch.size	16384	262144	Default maximum number of bytes of messages that can be processed at a time. The producer will attempt to batch records together into fewer requests whenever multiple records are being sent to the same partition. This helps improve performance of both the client and the server. No attempt will be made to batch records larger than this size. Requests sent to brokers will contain multiple batches, one for each partition with data available to be sent. A smaller batch size will make batching less common and may reduce throughput (a batch size of zero will disable batching entirely). A larger batch size may use more memory as a buffer of the specified batch size will always be allocated in anticipation of additional records.
buffer.memory	33554432	67108864	Total bytes of memory the producer can use to buffer records waiting to be sent to the server. If records are sent faster than they can be delivered to the broker, the producer will stop sending records or throw a "block.on.buffer.full" exception. This setting should correspond roughly to the total memory the producer will use, but is not a rigid bound since not all memory the producer uses is used for buffering. Some additional memory will be used for compression (if compression is enabled) as well as for maintaining in-flight requests.

Parameter	Default Value	Recommended Value	Description
enable.idempotence	● Earlier than v3.0: false ● v3.0 and later: true	If idempotence is not required, you are advised to set this parameter to **false**.	If you have enabled idempotence on the producer client, and produced messages, message offsets are not continuous on the consumer client or on the **Message Query** page on the Kafka console. This is because enabling idempotence generates some metadata control messages during message production. These control messages are produced to topics, and are invisible to consumers.

Table 5-7 Consumer parameters

Parameter	Default Value	Recommended Value	Description
auto.commit.enable	TRUE	FALSE	If this parameter is set to **TRUE**, the offset of messages already fetched by the consumer will be periodically committed to ZooKeeper. This committed offset will be used when the process fails as the position from which the new consumer will begin. Constraints: If this parameter is set to **FALSE**, to avoid message loss, an offset must be committed to ZooKeeper after the messages are successfully consumed.

Parameter	Default Value	Recommended Value	Description
auto.offset.reset	latest	earliest	Indicates what to do when there is no initial offset in ZooKeeper or if the current offset has been deleted. Options: ● **earliest**: Automatically reset to the smallest offset. ● **latest**: The offset is automatically reset to the largest offset. ● **none**: The system throws an exception to the consumer if no offset is available. ● **anything else**: The system throws an exception to the consumer. **NOTE** If this parameter is set to **latest**, the producer may start to send messages to new partitions (if any) before the consumer resets to the initial offset. As a result, some messages will be lost.
connections.max.idle.ms	600000	30000	Timeout interval (in ms) for an idle connection. The server closes the idle connection after this period of time ends. Setting this parameter to **30000** can reduce the server response failures when the network condition is poor.
max.poll.records	500	Must be less than the value of **max.poll.interval.ms**.	The maximum number of messages that a consumer can pull from a broker at a time.
max.poll.interval.ms	300000	Increase this parameter if complex and time-consuming logic exists between two polls.	The maximum interval between consumer polls, in milliseconds. If this parameter is exceeded, the consumption fails and the consumer is removed from the consumer group, triggering rebalance.
heartbeat.interval.ms	3000	≥ 3000	Heartbeat interval between a consumer and Kafka, in milliseconds.

Parameter	Default Value	Recommended Value	Description
session.timeout.ms	10000	Set this parameter to at least 3 times the value of **heartbeat.interval.ms**.	The consumer-broker session timeout when the offset is managed by consumer group, in milliseconds.
fetch.max.bytes	1000000	max.request.size < message.max.bytes < fetch.max.bytes	The maximum bytes of a message that a consumer pulls from a broker at a time.

5.3.2 Suggestions on Using the Kafka Client

Consumers

1. Ensure that the owner thread does not exit abnormally. Otherwise, the client may fail to initiate consumption requests and the consumption will be blocked.
2. Commit messages only after they have been processed. Otherwise, the messages may fail to be processed and cannot be polled again.
3. Generally, do not commit every message. Otherwise, there will be many **OFFSET_COMMIT** requests, causing high CPU usage. For example, if a consumption request pulls 1000 messages and commits every one of them, TPS of the commit requests is 1000 times that of consumption. The smaller the message size, the larger the ratio. You can commit a specific number of messages in batches or enable **enable.auto.commit**. However, if the client is faulty, some cached consumption offset may be lost, resulting in repeated consumption. Therefore, you are advised to commit messages in batches based on service requirements.
4. A consumer cannot frequently join or leave a group. Otherwise, the consumer will frequently perform rebalancing, which blocks consumption.
5. The number of consumers in a consumer group must be within the total partitions subscribed by the consumer group. Otherwise, some consumers cannot pull messages.
6. Ensure that the consumer polls at regular intervals to keep sending heartbeats to the server. If the consumer stops sending heartbeats for long enough, the consumer session will time out and the consumer will be considered to have stopped. This will also block consumption.
7. Ensure that there is a limitation on the size of messages buffered locally to avoid an out-of-memory (OOM) situation.
8. Set the timeout for the consumer session to 30 seconds: session.timeout.ms=30000.
9. Kafka supports exactly-once delivery. Therefore, ensure the idempotency of processing messages for services.

10. Always close the consumer before exiting. Otherwise, consumers in the same group may be blocked within the timeout set by **session.timeout.ms**.
11. Do not start a consumer group name with a special character, such as a number sign (#). Otherwise, monitoring data of the consumer group cannot be displayed.

Producers

1. Synchronous replication: Set **acks** to **all**.
2. Retry message sending: Set **retries** to **3**.
3. Optimize message sending: For latency-sensitive messages, set **linger.ms** to **0**. For latency-insensitive messages, set **linger.ms** to a value ranging from **100** to **1000**.
4. Ensure that the producer has sufficient JVM memory to avoid blockages.
5. Set the timestamp to the local time. Messages will fail to age if the timestamp is a future time.
6. Try reusing producers. Do not create producers frequently. When idempotence is enabled (default for producer clients 3.0 and later), producing messages creates producer state objects on the server. Frequent creation results in too many objects to be reclaimed in time, causing server memory surges and performance deterioration. Set **enable.idempotence** to **false** if the idempotence is not required.

Topics

Recommended topic configurations: Use 3 replicas, enable synchronous replication, and set the minimum number of in-sync replicas to 2. The number of in-sync replicas cannot be the same as the number of replicas of the topic. Otherwise, if one replica is unavailable, messages cannot be produced.

You can enable or disable automatic topic creation. If automatic topic creation is enabled, the system automatically creates a topic when a message is produced in or consumed from a topic that does not exist. This topic has the following default settings: 3 partitions, 3 replicas, aging time 72 hours, synchronous replication and flushing disabled, **CreateTime** message timestamp, and maximum 10,485,760 bytes message size.

Others

Maximum number of connections: 3000

Maximum size of a message: 10 MB

Access Kafka using SASL_SSL. Ensure that your DNS service is capable of resolving an IP address to a domain name. Alternatively, map all Kafka broker IP addresses to host names in the **hosts** file. Prevent Kafka clients from performing reverse resolution. Otherwise, connections may fail to be established.

Apply for a disk space size that is more than twice the size of service data multiplied by the number of replicas. In other words, keep 50% of the disk space idle.

Avoid frequent full GC in JVM. Otherwise, message production and consumption will be blocked.

5.4 Connecting to Kafka Using the Client (SASL Disabled)

This section describes how to access a Kafka instance with SASL disabled on an open-source Kafka client. With SASL disabled, there is no authentication required in such a connection and data is transmitted in plaintext, which is friendly to performance.

Each Kafka broker allows a maximum of 1000 connections from each IP address. Excess connections will be rejected. You can change the limit by referring to **Modifying Kafka Instance Configuration Parameters**, that is, to modify parameter **max.connections.per.ip**.

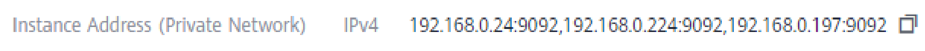
Prerequisites

- The network between the client and the Kafka instance has been established. For details about the network requirements, see **Kafka Network Connection Conditions**.
- Security group rules have been properly configured.

  Before accessing a Kafka instance with SASL disabled on a client, configure proper security group rules for the instance. For details, see **Table 5-2**.
- The Kafka instance addresses have been obtained.

  Obtain the instance connection addresses in the **Connection** area on the **Basic Information** page on the Kafka console.
  - For private access within a VPC, the Kafka connection addresses are shown as follows.

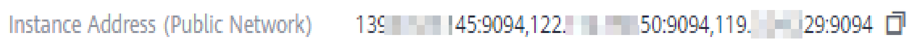
    Figure 5-16 Kafka instance addresses for private access within a VPC

    Instance Address (Private Network) IPv4 192.168.0.24:9092,192.168.0.224:9092,192.168.0.197:9092

  - For public access, the Kafka connection addresses are shown as follows.

    Figure 5-17 Kafka instance addresses for public access

    Instance Address (Public Network) 139[redacted]145:9094,122.[redacted]50:9094,119.[redacted]29:9094

- If automatic topic creation is not enabled for the Kafka instance, **create a topic** before connecting to the instance.
- Kafka CLI **v1.1.0**, **v2.3.0**, or **v2.7.2** is available. Ensure that the Kafka instance and the CLI use the same version.
- **JDK v1.8.111 or later** has been installed on the server, and the **JAVA_HOME** and **PATH** environment variables have been configured as follows:

  Add the following lines to the **.bash_profile** file in the home directory as an authorized user. In this command, **/opt/java/jdk1.8.0_151** is the JDK installation path. Change it to the path where you install JDK.

  ```
  export JAVA_HOME=/opt/java/jdk1.8.0_151
  export PATH=$JAVA_HOME/bin:$PATH
  ```

  Run the **source .bash_profile** command for the modification to take effect.

Accessing the Instance Using CLI

The following uses Linux as an example.

Step 1 Decompress the Kafka CLI package.

Access the directory where the CLI package is stored and run the following command to decompress the package:

```
tar -zxf [kafka_tar]
```

In the preceding command, *[kafka_tar]* indicates the name of the CLI package.

For example:

```
tar -zxf kafka_2.12-2.7.2.tgz
```

Step 2 Access the **/bin** directory of the Kafka CLI.

In Windows, you need to access the **/bin/windows** directory.

Step 3 Run the following command to create messages:

```
./kafka-console-producer.sh --broker-list ${connection-address} --topic ${topic-name}
```

Parameter description:

- *{connection-address}*: the address obtained in **Prerequisites**.
- *{topic-name}*: the name of the topic created for the Kafka instance. If automatic topic creation has enabled for the Kafka instance, set this parameter to the name of a created topic or a topic that has not been created.

The following example uses connection addresses **10.xx.xx.45:9094,10.xx.xx.127:9094,10.xx.xx.103:9094**. After running the preceding command, you can send a message to the Kafka instance by writing it and pressing **Enter**. Each line of content is sent as a message.

```
[root@ecs-kafka bin]# ./kafka-console-producer.sh --broker-list
10.xx.xx.45:9094,10.xx.xx.127:9094,10.xx.xx.103:9094  --topic topic-demo
>Hello
>DMS
>Kafka!
>^C[root@ecs-kafka bin]#
```

To stop creating messages, press **Ctrl**+**C** to exit.

Step 4 Run the following command to retrieve messages:

```
./kafka-console-consumer.sh --bootstrap-server ${connection-address} --topic ${topic-name} --group $
{consumer-group-name} --from-beginning
```

Parameter description:

- *{connection-address}*: the address obtained in **Prerequisites**.
- *{topic-name}*: the name of the topic created for the Kafka instance.
- *{consumer-group-name}*: the consumer group name set based on your service requirements. **If a consumer group name has been specified in the configuration file, ensure that you use the same name in the command line. Otherwise, consumption may fail.** If a consumer group name starts with a special character, such as a number sign (#), the monitoring data cannot be displayed.

Example:

```
[root@ecs-kafka bin]#  ./kafka-console-consumer.sh --bootstrap-server
10.xx.xx.45:9094,10.xx.xx.127:9094,10.xx.xx.103:9094 --topic topic-demo --group order-test --from-beginning
Kafka!
DMS
Hello
^CProcessed a total of 3 messages
[root@ecs-kafka bin]#
```

To stop retrieving messages, press **Ctrl**+**C** to exit.

----End

5.5 Connecting to Kafka Using the Client (SASL Enabled)

This section describes how to access a Kafka instance with SASL enabled on an open-source Kafka client. With SASL enabled, there is SASL authentication required in such a connection and data is transmitted securely in ciphertext.

For security purposes, **TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256** is supported.

- If intra-VPC plaintext access is enabled for an instance, data is transmitted in plaintext when you connect to the instance through a private network. For details about how to connect, see **Connecting to Kafka Using the Client (SASL Disabled)**.
- Each Kafka broker allows a maximum of 1000 connections from each IP address by default. Excess connections will be rejected. You can change the limit by referring to **Modifying Kafka Instance Configuration Parameters**, that is, to modify parameter **max.connections.per.ip**.

Prerequisites

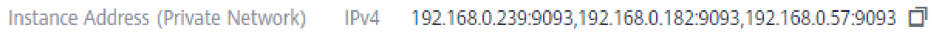
- The network between the client and the Kafka instance has been established. For details about the network requirements, see **Kafka Network Connection Conditions**.
- Security group rules have been properly configured.

  Before accessing a Kafka instance with SASL enabled on a client, configure proper security group rules for the instance. For details, see **Table 5-2**.
- The Kafka instance addresses have been obtained.

  Obtain the instance connection addresses in the **Connection** area on the **Basic Information** page on the Kafka console.

  – For private access within a VPC, the Kafka connection addresses are shown as follows.

  Figure 5-18 Kafka instance addresses for private access within a VPC

  Instance Address (Private Network) IPv4 192.168.0.239:9093,192.168.0.182:9093,192.168.0.57:9093

  – For public access, the Kafka connection addresses are shown as follows.

Figure 5-19 Kafka instance addresses for public access

Instance Address (Public Network) 139[redacted]145:9095,122.[redacted]50:9095,119[redacted]29:9095

- The SASL mechanism in use is known.

  In the **Connection** area on the Kafka instance details page, view **SASL Mechanism**. If both SCRAM-SHA-512 and PLAIN are enabled, use either of them in connection configurations. For instances that were created much earlier, if **SASL Mechanism** is not displayed on the instance details page, PLAIN is used by default.

  Figure 5-20 SASL mechanism in use

  SASL Mechanism SCRAM-SHA-512

- If automatic topic creation is not enabled for the Kafka instance, **create a topic** before connecting to the instance.
- The **client.truststore.jks** certificate has been downloaded. Click the Kafka instance to go to the **Basic Information** tab page. Click **Download** next to **SSL Certificate** in the **Connection** area. Download and decompress the package to obtain the client certificate file **client.truststore.jks**.
- Kafka CLI **v1.1.0**, **v2.3.0**, or **v2.7.2** is available. Ensure that the Kafka instance and the CLI use the same version.
- **JDK v1.8.111 or later** has been installed on the server, and the **JAVA_HOME** and **PATH** environment variables have been configured as follows:

  Add the following lines to the **.bash_profile** file in the home directory as an authorized user. In this command, **/opt/java/jdk1.8.0_151** is the JDK installation path. Change it to the path where you install JDK.

  ```
  export JAVA_HOME=/opt/java/jdk1.8.0_151
  export PATH=$JAVA_HOME/bin:$PATH
  ```

  Run the **source .bash_profile** command for the modification to take effect.

Accessing the Instance Using CLI

The following uses Linux as an example.

Step 1 Map hosts to IP addresses in the **/etc/hosts** file on the host where the client is located, so that the client can quickly parse the instance brokers.

Set IP addresses to the instance connection addresses obtained in **Prerequisites**. Set hosts to the names of instance hosts. Specify a unique name for each host.

For example:

```
10.154.48.120 server01
10.154.48.121 server02
10.154.48.122 server03
```

Step 2 Decompress the Kafka CLI package.

Access the directory where the CLI package is stored and run the following command to decompress the package:

```
tar -zxf [kafka_tar]
```

In the preceding command, *[kafka_tar]* indicates the name of the CLI package.

For example:

```
tar -zxf kafka_2.12-2.7.2.tgz
```

Step 3 Modify the Kafka CLI configuration file based on the **SASL mechanism**.

- **If PLAIN is used**, find the **consumer.properties** and **producer.properties** files in the **/config** directory of the Kafka CLI and add the following content to the files:

  ```
  sasl.jaas.config=org.apache.kafka.common.security.plain.PlainLoginModule required \
  username="**********" \
  password="**********";
  sasl.mechanism=PLAIN

  security.protocol=SASL_SSL
  ssl.truststore.location={ssl_truststore_path}
  ssl.truststore.password=dms@kafka
  ssl.endpoint.identification.algorithm=
  ```

  Parameter description:

  - **username** and **password**: username and password you set when enabling SASL_SSL during Kafka instance creation or when creating a SASL_SSL user.
  - **ssl.truststore.location**: path for storing the **client.truststore.jks** certificate. Even in Windows, you need to use slashes (/) for the certificate path. Do not use backslashes (\), which are used by default for paths in Windows. Otherwise, the client will fail to obtain the certificate.
  - **ssl.truststore.password**: server certificate password, which must be set to **dms@kafka** and cannot be changed.
  - **ssl.endpoint.identification.algorithm**: whether to verify the certificate domain name. **This parameter must be left blank, which indicates disabling domain name verification**.

- **If SCRAM-SHA-512 is used**, find the **consumer.properties** and **producer.properties** files in the **/config** directory of the Kafka CLI and add the following content to the files:

  ```
  sasl.jaas.config=org.apache.kafka.common.security.scram.ScramLoginModule required \
  username="**********" \
  password="**********";
  sasl.mechanism=SCRAM-SHA-512

  security.protocol=SASL_SSL
  ssl.truststore.location={ssl_truststore_path}
  ssl.truststore.password=dms@kafka
  ssl.endpoint.identification.algorithm=
  ```

  Parameter description:

  - **username** and **password**: username and password you set when enabling SASL_SSL during Kafka instance creation or when creating a SASL_SSL user.
  - **ssl.truststore.location**: path for storing the **client.truststore.jks** certificate. Even in Windows, you need to use slashes (/) for the certificate path. Do not use backslashes (\), which are used by default for paths in Windows. Otherwise, the client will fail to obtain the certificate.
  - **ssl.truststore.password**: server certificate password, which must be set to **dms@kafka** and cannot be changed.

– **ssl.endpoint.identification.algorithm**: whether to verify the certificate domain name. **This parameter must be left blank, which indicates disabling domain name verification**.

Step 4 Access the **/bin** directory of the Kafka CLI.

In Windows, you need to access the **/bin/windows** directory.

Step 5 Run the following command to create messages:

```
./kafka-console-producer.sh --broker-list ${connection addr} --topic ${topic name} --producer.config ../
config/producer.properties
```

Parameter description:

- *{connection-address}*: the address obtained in **Prerequisites**.
- *{topic-name}*: the name of the topic created for the Kafka instance. If automatic topic creation has enabled for the Kafka instance, set this parameter to the name of a created topic or a topic that has not been created.

The following example uses connection addresses **10.xx.xx.45:9095,10.xx.xx.127:9095,10.xx.xx.103:9095**.

After running the preceding command, you can send a message to the Kafka instance by writing it and pressing **Enter**. Each line of content is sent as a message.

```
[root@ecs-kafka bin]#./kafka-console-producer.sh --broker-list
10.xx.xx.45:9095,10.xx.xx.127:9095,10.xx.xx.103:9095  --topic topic-demo --producer.config ../config/
producer.properties
>Hello
>DMS
>Kafka!
>^C[root@ecs-kafka bin]#
```

To stop creating messages, press **Ctrl**+**C** to exit.

Step 6 Run the following command to retrieve messages:

```
./kafka-console-consumer.sh --bootstrap-server ${connection-address} --topic ${topic-name} --group $
{consumer-group-name} --from-beginning  --consumer.config ../config/consumer.properties
```

Parameter description:

- *{connection-address}*: the address obtained in **Prerequisites**.
- *{topic-name}*: the name of the topic created for the Kafka instance.
- *{consumer-group-name}*: the consumer group name set based on your service requirements. **If a consumer group name has been specified in the configuration file, ensure that you use the same name in the command line. Otherwise, consumption may fail.** If a consumer group name starts with a special character, such as a number sign (#), the monitoring data cannot be displayed.

Example:

```
[root@ecs-kafka bin]#  ./kafka-console-consumer.sh --bootstrap-server
10.xx.xx.45:9095,10.xx.xx.127:9095,10.xx.xx.103:9095 --topic topic-demo --group order-test --from-beginning
--consumer.config ../config/consumer.properties
Hello
DMS
Kafka!
^CProcessed a total of 3 messages
[root@ecs-kafka bin]#
```

To stop retrieving messages, press **Ctrl**+**C** to exit.

----End

5.6 Connecting to Kafka on the Console

This section describes how a Kafka instance produces messages on the console. Specified messages can be sent to a Kafka instance to verify service logic.

Prerequisites

- Messages can be produced in a topic only when the instance is in the **Running** state.
- **A topic** has been created.

Producing Messages on the Console

Step 1 Log in to the console.

Step 2 Click 📍 in the upper left corner to select a region.

NOTE

Select the region where your Kafka instance is located.

Step 3 Click ☰ in the upper left corner and choose **Application** > **Distributed Message Service (for Kafka)** to open the console of DMS for Kafka.

Step 4 Click the desired Kafka instance to go to the instance details page.

Step 5 In the navigation pane, choose **Topics**.

Step 6 Choose **More** > **Creating Messages** in the row that contains the desired topic. The **Creating Messages** dialog box is displayed.

Step 7 Set message parameters by referring to **Table 5-8**.

Table 5-8 Message parameters

Parameter	Description
Message Body	Message content. 0–2,000 characters.
Message Key	Message key.
Specify Partition	Indicates whether to enable the function of sending messages to a specified partition. ● Off: Messages are sent to partitions based on their key hash. ● On: Messages are sent to specified partitions. Requires the partition ID.

Step 8 Click **OK**.

You can view the sent messages on the **Message Query** page.

----End

6 Managing Messages

6.1 Viewing Kafka Messages

You can view the offset of different partitions, the message size, creation time, and body of messages in topics.

Procedure

Step 1 Log in to the console.

Step 2 Click 📍 in the upper left corner to select a region.

NOTE

Select the region where your Kafka instance is located.

Step 3 Click ☰ and choose **Application** > **Distributed Message Service (for Kafka)** to open the console of DMS for Kafka.

Step 4 Click the desired Kafka instance to go to the instance details page.

Step 5 In the left navigation pane, choose **Message Query**.

Step 6 Set the query parameters by referring to **Table 6-1**.

Table 6-1 Message query parameters

Parameter	Description
Topic Name	Name of the topic to be queried.
Partition	Partition where the messages are located. If no partition is specified, messages in all partitions of the topic are displayed in the query result.

Parameter	Description
Search By	The following methods are supported: ● **Creation time**: Search by the time that messages are created. ● **Offset**: Search by the message position.
Search For	This parameter is displayed only when **Search By** is set to **Creation time**. Enter a keyword in the message body. **NOTE** Due to resource and performance restrictions, query with content is limited to 10 results. Each search covers at most 10,000 records, or 200 MB. For large records (> 20 KB per message) or a long period, dump messages for offline query.

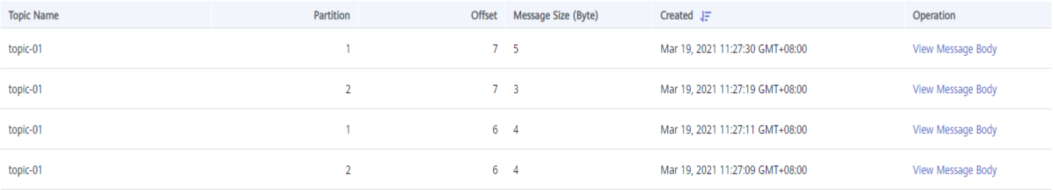
NOTE

If a topic contains a large amount of data, an internal service error may be reported when you query messages in a topic with only one replica. You can shorten the time range for query based on the data volume.

Step 7 Click **Search** to query messages.

The query result is as follows.

Figure 6-1 Querying topic messages



Topic Name	Partition	Offset	Message Size (Byte)	Created	Operation
topic-01	1	7	5	Mar 19, 2021 11:27:30 GMT+08:00	View Message Body
topic-01	2	7	3	Mar 19, 2021 11:27:19 GMT+08:00	View Message Body
topic-01	1	6	4	Mar 19, 2021 11:27:11 GMT+08:00	View Message Body
topic-01	2	6	4	Mar 19, 2021 11:27:09 GMT+08:00	View Message Body

Parameter description:

- **Topic Name**: name of the topic where the message is located
- **Partition**: partition where the message is located
- **Offset**: position of the message in the partition
- **Message Size (Byte)** size of the message
- **Created**: time when the message is created. The message creation time is specified by **CreateTime** when a producer creates messages. If this parameter is not set during message creation, the message creation time is year 1970 by default.

Step 8 Click **View Message Body**. In the displayed **View Message Body** dialog box, view the message content, including the topic name, partition, offset, creation time, and message body.

NOTE

> The console displays messages smaller than 4 KB. To view messages larger than 4 KB, click **Download Message**.

Step 9 (Optional) To restore the default settings, click **Reset**.

----End

6.2 Changing Kafka Message Retention Period

Aging time is a period that messages in a topic are retained for. Consumers must consume messages before this period ends. Otherwise, the messages will be deleted and can no longer be consumed.

The topic retention period is 72 hours by default, and can be changed later as required. Changing the aging time does not affect services.

You can change the aging time in either of the following ways:

- By editing the topic on the **Topics** tab page
- By changing the value of the **log.retention.hours** parameter on the **Parameters** tab page. For details, see **Modifying Kafka Instance Configuration Parameters**.

NOTE

> The **log.retention.hours** parameter takes effect only for topics that have no aging time configured. If there is aging time configured for a topic, it overrides the **log.retention.hours** parameter. For example, if the aging time of Topic01 is set to 60 hours and **log.retention.hours** is set to 72 hours, the actual aging time of Topic01 is 60 hours.

Procedure

Step 1 Log in to the console.

Step 2 Click 📍 in the upper left corner to select a region.

NOTE

> Select the region where your Kafka instance is located.

Step 3 Click ☰ and choose **Application** > **Distributed Message Service (for Kafka)** to open the console of DMS for Kafka.

Step 4 Click the desired Kafka instance to view the instance details.

Step 5 In the navigation pane, choose **Topics**.

Step 6 Modify the topic aging time using either of the following methods:

- Select one or more topics and click **Edit Topic** in the upper left corner.
- In the row containing the desired topic, click **Edit**.

Step 7 In the **Edit Topic** dialog box, enter the aging time and click **OK**.

----End

6.3 Deleting Kafka Messages

This section describes how to delete messages stored in a topic on the console.

Note

Deleting messages takes effect permanently.

Prerequisites

Before deleting a message, set the **auto.offset.reset** parameter in the code of consumption. **auto.offset.reset** specifies the consumption policy of a consumer when there is no initial offset in Kafka or the current offset does not exist (for example, the current offset has been deleted). Options:

- **latest**: The offset is automatically reset to the latest offset.
- **earliest**: The offset is automatically reset to the earliest offset.
- **none**: The system throws an exception to the consumer.

NOTICE

If this parameter is set to **latest**, the producer may start to send messages to new partitions (if any) before the consumer resets to the initial offset. As a result, some messages will be lost.

Procedure

Step 1 Log in to the console.

Step 2 Click 📍 in the upper left corner to select a region.

NOTE

Select the region where your Kafka instance is located.

Step 3 Click ☰ and choose **Application** > **Distributed Message Service (for Kafka)** to open the console of DMS for Kafka.

Step 4 Click the desired Kafka instance to go to the instance details page.

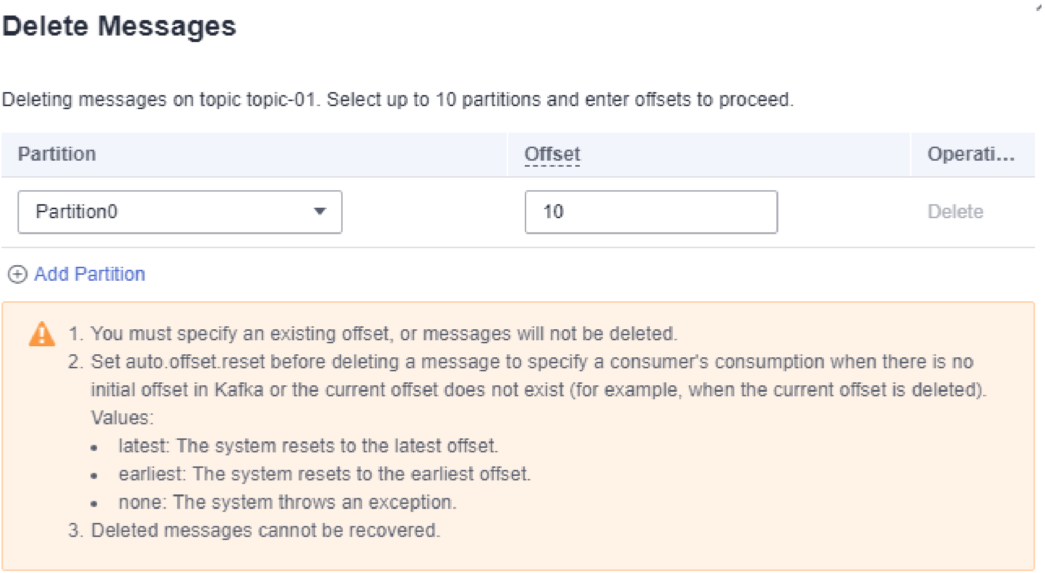
Step 5 In the navigation pane, choose **Topics**.

Step 6 Perform either of the following steps to display the **Delete Message** dialog box:

- If SASL has not been enabled for the instance, click **Delete Messages** in the row that contains the topic whose messages you want to delete.
- If SASL has been enabled for the instance, choose **More** > **Delete Messages** in the row that contains the topic whose messages you want to delete.

Step 7 Set the parameters for deleting messages, as shown in **Table 6-2**.

Figure 6-2 Deleting messages



Delete Messages

Deleting messages on topic topic-01. Select up to 10 partitions and enter offsets to proceed.

Partition	Offset	Operati...
Partition0	10	Delete

1. You must specify an existing offset, or messages will not be deleted.
2. Set auto.offset.reset before deleting a message to specify a consumer's consumption when there is no initial offset in Kafka or the current offset does not exist (for example, when the current offset is deleted). Values:
   - latest: The system resets to the latest offset.
   - earliest: The system resets to the earliest offset.
   - none: The system throws an exception.
3. Deleted messages cannot be recovered.

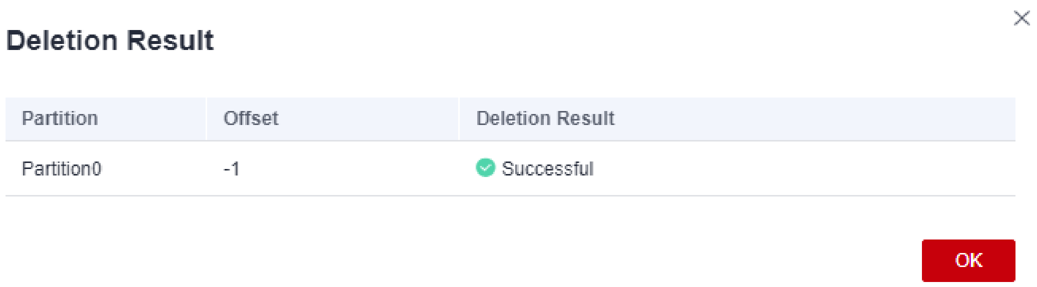
Table 6-2 Parameters for deleting a message

Parameter	Description
Partition	Select the ID of the partition where the message is located.
Offset	Enter an offset. The data after the earliest offset and before this offset will be deleted. For example, if the earliest offset is 2 and the entered offset is 5, the messages whose offset ranges from 2 to 4 will be deleted. **NOTE** ● If **Offset** is set to **-1**, all messages in the partition will be deleted. ● If the offset you entered is not between the earliest offset and the latest offset of the specified partition, no messages will be deleted.

To delete messages from multiple partitions, click **Add Partition** and specify the partition and offset for the messages to be deleted. 10 partitions can be deleted at most at a time.

Step 8 Click **OK**. The **Deletion Result** dialog box is displayed. Click **OK** to delete the messages.

Figure 6-3 Deletion result



Deletion Result

Partition	Offset	Deletion Result
Partition0	-1	✓ Successful

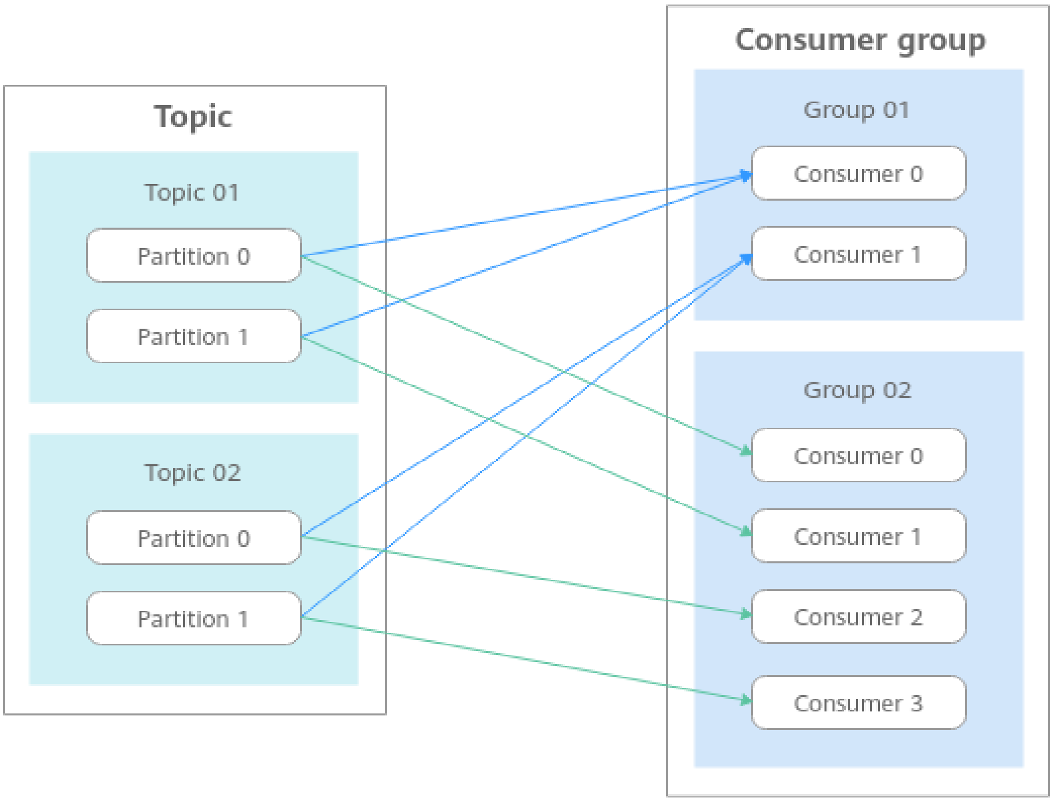
----End

7 Managing Consumer Groups

7.1 Creating a Kafka Consumer Group

A consumer subscribes to a topic. A consumer group consists of one or more consumers. Within a consumer group, each consumer can consume multiple partitions at the same time. Each partition can be consumed by one consumer at a time.

Figure 7-1 Example consumption



auto.create.groups.enable: a consumer group is automatically created when a consumer attempts to enter a group that does not exist.

- A consumer group is required before consuming messages when **auto.create.groups.enable** is **false** in **Configuring Parameters**. Otherwise, consumption will fail.
- A consumer group is created automatically before consuming messages when **auto.create.groups.enable** is **true** in **Configuring Parameters**.

Create a consumer group on the console.

NOTE

Creating a consumer group on the console does not require an instance restart.

Notes and Constraints

- If **auto.create.groups.enable** is set to **true**, the consumer group status is **EMPTY**, and no offset has been submitted, the system automatically deletes the consumer group 10 minutes later.
- If **auto.create.groups.enable** is set to **false**, the system does not automatically delete consumer groups. You can manually delete them.
- If a consumer group has never committed an offset, the group will be deleted after the Kafka instance restarts.

Procedure

Step 1 Log in to the console.

Step 2 Click 📍 in the upper left corner to select a region.

NOTE

Select the region where your Kafka instance is located.

Step 3 Click ☰ and choose **Application** > **Distributed Message Service (for Kafka)** to open the console of DMS for Kafka.

Step 4 Click the desired Kafka instance to view the instance details.

Step 5 In the navigation pane, choose **Consumer Groups**.

Step 6 Click **Create Consumer Group**.

Step 7 Set consumer group parameters by referring to **Table 7-1** and click **OK**.

Table 7-1 Consumer group parameters

Parameter	Description
Consumer Group Name	Enter 3 to 64 characters, starting with a letter or underscore (_). Use only letters, digits, periods (.), hyphens (-), and underscores (_). If a consumer group name starts with a special character, for example, a number sign (#), monitoring data cannot be displayed.

View the new consumer group in the consumer group list.

----End

7.2 Querying the Kafka Consumer Group List

After a consumer group is created, you can view its configuration and status.

Viewing the Consumer Group List (Console)

Step 1 Log in to the console.

Step 2 Click 📍 in the upper left corner to select a region.

NOTE

Select the region where your Kafka instance is located.

Step 3 Click ≡ and choose **Application** > **Distributed Message Service (for Kafka)** to open the console of DMS for Kafka.

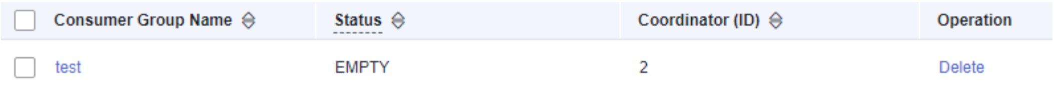
Step 4 Click the desired Kafka instance to view its details.

Step 5 In the navigation pane, choose the **Consumer Groups** tab.

The consumer group name, status, and Coordinator (ID) are displayed. Coordinator (ID) indicates the broker where the coordinator component is located. The consumer group status can be:

- **DEAD**: The consumer group has no member or metadata.
- **EMPTY**: The consumer group has metadata but has no member.
- **PREPARING_REBALANCE**: The consumer group is to be rebalanced.
- **COMPLETING_REBALANCE**: All members have joined the consumer group.
- **STABLE**: Members in the consumer group can consume messages normally.

Figure 7-2 Consumer group list



□	Consumer Group Name	Status	Coordinator (ID)	Operation
□	test	EMPTY	2	Delete

Step 6 (Optional) To query a consumer group, enter a consumer group name or status, Coordinator (ID), or keyword, then press **Enter**.

Step 7 (Optional) To refresh the consumer group list, click ⟳ in the upper right corner.

----End

Viewing the Consumer Group List (Kafka CLI)

- For a Kafka instance with SASL disabled, run the following command in the **/bin** directory of the Kafka client:

```
./kafka-consumer-groups.sh --bootstrap-server ${connection-address} --list
```

Parameter description: **connection-address** indicates the Kafka instance address, which can be obtained in the **Connection** area on the **Basic Information** page on the Kafka console.

Example:

```
[root@ecs-kafka bin]# ./kafka-consumer-groups.sh --bootstrap-server
192.168.xx.xx:9092,192.xx.xx.212:9092,192.xx.xx.147:9092 --list
test
__consumer-group-dial-test
[root@ecs-kafka bin]#
```

- For a Kafka instance with SASL enabled, do as follows:

  a. (Optional) If the username and password, and the SSL certificate has been configured, skip this step and go to **b**. Otherwise, do as follows:

     Create the **ssl-user-config.properties** file in the **/config** directory of the Kafka client. Add the username and password, and the SSL certificate configuration by referring to **Step 3**.

  b. Run the following command in the **/bin** directory of the Kafka client:

     ```
     ./kafka-consumer-groups.sh --bootstrap-server ${connection-address} --list --command-config ../
     config/ssl-user-config.properties
     ```

     Parameter description: **connection-address** indicates the Kafka instance address, which can be obtained in the **Connection** area on the **Basic Information** page on the Kafka console.

     Example:

     ```
     [root@ecs-kafka bin]# ./kafka-consumer-groups.sh --bootstrap-server
     192.168.xx.xx:9093,192.168.xx.xx:9093,192.168.xx.xx:9093 --list --command-config ../config/ssl-
     user-config.properties
     test
     __consumer-group-dial-test
     [root@ecs-kafka bin]#
     ```

7.3 Viewing Kafka Consumer Information

If a consumer group has consumers who are accessing a Kafka instance, you can view their connection information.

Note on Viewing the Consumer Address on Kafka Manager

Due to cache reasons, the consumer connection addresses displayed on Kafka Manager may have expired. In this case, restart Kafka Manager.

Prerequisites

The consumer list and connection address can be viewed only when consumers in a consumer group are connected to the Kafka instance (that is, the consumer group is in the **STABLE** state).

Viewing the Consumer List (Console)

Step 1 Log in to the console.

Step 2 Click 📍 in the upper left corner to select a region.

> **NOTE**
>
> Select the region where your Kafka instance is located.

Step 3 Click ☰ and choose **Application** > **Distributed Message Service (for Kafka)** to open the console of DMS for Kafka.

Step 4 Click a Kafka instance name to go to the instance details page.

Step 5 In the navigation pane, choose **Consumer Groups**.

Step 6 Click the name of the desired consumer group.

Step 7 On the **Consumers** tab page, view the consumer list.

In the consumer list, you can view the consumer ID, consumer address, and client ID.

Step 8 (Optional) To query a specific consumer, enter the consumer ID in the search box and press **Enter**.

----End

Viewing the Consumer List (Kafka CLI)

- For a Kafka instance with SASL disabled, run the following command in the **/bin** directory of the Kafka client:

  ```
  ./kafka-consumer-groups.sh --bootstrap-server ${connection-address} --group ${group-name} --
  members --describe
  ```

  Parameter description:

  – **connection-address**: can be obtained from the **Connection** area on the **Basic Information** page on the Kafka console.

  – **group-name**: consumer group name.

  Example:

  ```
  [root@ecs-kafka bin]# ./kafka-consumer-groups.sh --bootstrap-server
  192.168.xx.xx:9092,192.168.xx.xx:9092,192.168.xx.xx:9092 --group test --members --describe

  GROUP           CONSUMER-ID                                     HOST            CLIENT-ID
  #PARTITIONS
  test            console-consumer-571a64fe-b0c4-47ce-833d-9e0da5a88d14 /192.168.0.215  console-
  consumer 3
  [root@ecs-kafka bin]#
  ```

- For a Kafka instance with SASL enabled, do as follows:

  a. (Optional) If the username and password, and the SSL certificate has been configured, skip this step and go to **b**. Otherwise, do as follows:

     Create the **ssl-user-config.properties** file in the **/config** directory of the Kafka client. Add the username and password, and the SSL certificate configuration by referring to **Step 3**.

  b. Run the following command in the **/bin** directory of the Kafka client:

     ```
     ./kafka-consumer-groups.sh --bootstrap-server ${connection-address} --group ${group-name} --
     members --describe --command-config ../config/ssl-user-config.properties
     ```

     Parameter description:

     ▪ **connection-address**: can be obtained from the **Connection** area on the **Basic Information** page on the Kafka console.

- **group-name**: consumer group name.

Example:

```
[root@ecs-kafka bin]# ./kafka-consumer-groups.sh --bootstrap-server
192.168.xx.xx:9093,192.168.xx.xx:9093,192.168.xx.xx:9093 --group test --members --describe --
command-config ../config/ssl-user-config.properties

GROUP           CONSUMER-ID                                     HOST            CLIENT-ID
#PARTITIONS
test            console-consumer-566d0c82-07d3-4d87-9a6e-f57a9bc9fc69 /192.168.0.215  console-
consumer 3
[root@ecs-kafka bin]#
```

Viewing Consumer Connection Addresses (Console)

Step 1 Log in to the console.

Step 2 Click 📍 in the upper left corner to select a region.

NOTE

Select the region where your Kafka instance is located.

Step 3 Click ☰ and choose **Application** > **Distributed Message Service (for Kafka)** to open the console of DMS for Kafka.

Step 4 Click the desired Kafka instance to view the instance details.

Step 5 In the navigation pane, choose **Consumer Groups**.

Step 6 Click the desired consumer group.

Step 7 On the **Consumers** tab page, view the consumer addresses.

----End

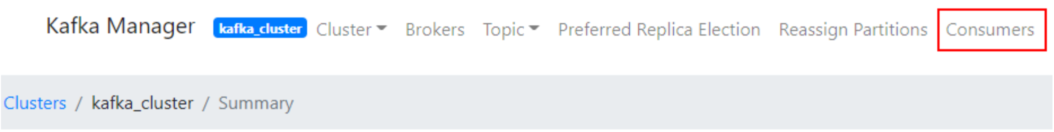
Viewing Consumer Connection Addresses (Kafka Manager)

Step 1 Log in to Kafka Manager.

Step 2 Click **kafka_cluster** to go to the cluster details page.

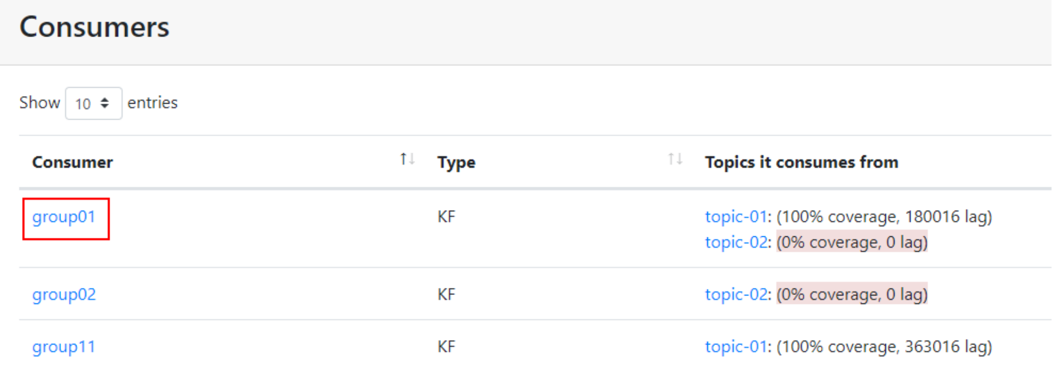
Step 3 On the top menu bar, choose **Consumers**.

Figure 7-3 Navigation bar



Step 4 Click the desired consumer group to view the topics that the group has subscribed to.

Figure 7-4 Consumer group list



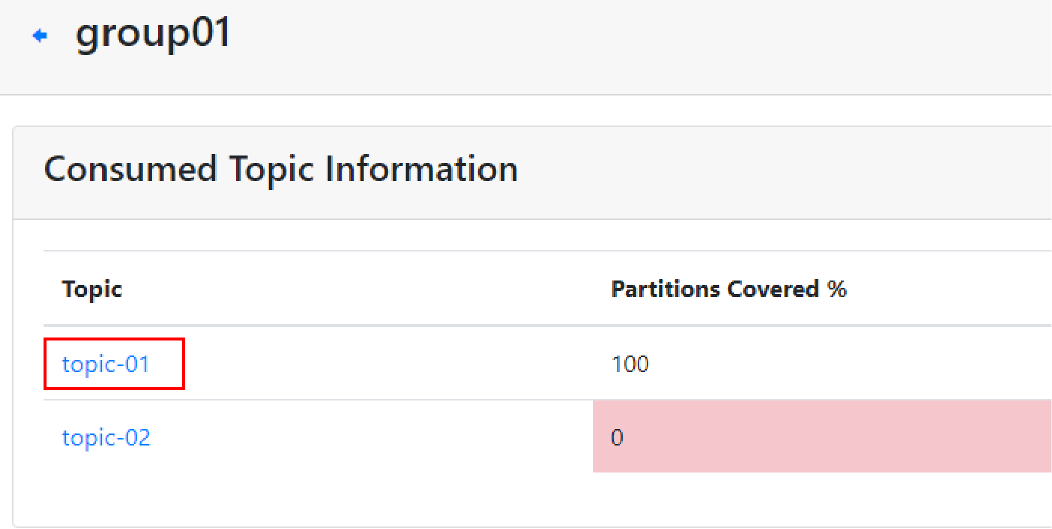
Consumers

Show 10 entries

Consumer	Type	Topics it consumes from
group01	KF	topic-01: (100% coverage, 180016 lag) topic-02: (0% coverage, 0 lag)
group02	KF	topic-02: (0% coverage, 0 lag)
group11	KF	topic-01: (100% coverage, 363016 lag)

Step 5 Click the desired topic to go to the topic details page.

Figure 7-5 Topics that the consumer group has subscribed to



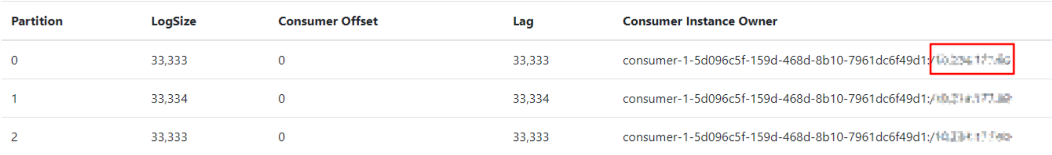
group01

Consumed Topic Information

Topic	Partitions Covered %
topic-01	100
topic-02	0

Step 6 In the **Consumer Instance Owner** column, view the consumer connection address.

Figure 7-6 Topic details page



Partition	LogSize	Consumer Offset	Lag	Consumer Instance Owner
0	33,333	0	33,333	consumer-1-5d096c5f-159d-468d-8b10-7961dc6f49d1:/[redacted]
1	33,334	0	33,334	consumer-1-5d096c5f-159d-468d-8b10-7961dc6f49d1:/[redacted]
2	33,333	0	33,333	consumer-1-5d096c5f-159d-468d-8b10-7961dc6f49d1:/[redacted]

----End

7.4 Viewing and Resetting Kafka Consumption Offsets

A consumption offset indicates the consumption progress of a consumer. This section describes how to view and reset consumption offsets.

Note

Messages may be consumed more than once after the offset is reset. Exercise caution when performing this operation.

Prerequisites

The consumer offset cannot be reset on the fly. You must first stop retrieval of the desired consumer group.

NOTICE

After a client is stopped, the server considers the client offline only after the time period specified in **ConsumerConfig.SESSION_TIMEOUT_MS_CONFIG** (1000 ms by default).

Viewing Consumer Offsets (Console)

Step 1 Log in to the console.

Step 2 Click 📍 in the upper left corner to select a region.

NOTE

Select the region where your Kafka instance is located.

Step 3 Click ☰ and choose **Application** > **Distributed Message Service (for Kafka)** to open the console of DMS for Kafka.

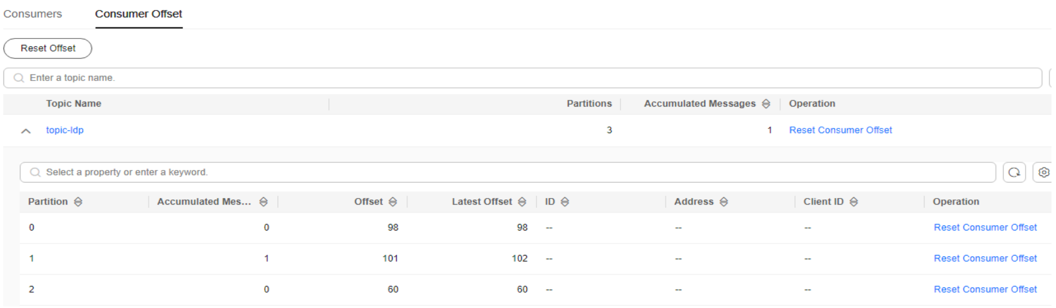
Step 4 Click the desired Kafka instance to view the instance details.

Step 5 In the navigation pane, choose **Consumer Groups**.

Step 6 Click the name of the desired consumer group.

Step 7 On the **Consumer Offset** tab page, view the list of topics that the consumer group has subscribed to, total number of messages accumulated in the topic, consumption progress in each partition of the topic (number of accumulated messages, offset, latest offset, consumer ID, consumer address, and client ID).

Figure 7-7 Consumer offsets



Topic Name	Partitions	Accumulated Messages	Operation
topic-ldp	3	1	Reset Consumer Offset

Partition	Accumulated Mes...	Offset	Latest Offset	ID	Address	Client ID	Operation
0	0	98	98	--	--	--	Reset Consumer Offset
1	1	101	102	--	--	--	Reset Consumer Offset
2	0	60	60	--	--	--	Reset Consumer Offset

Step 8 (Optional) To query the consumer offsets of a specific topic, enter the topic name in the search box and press **Enter**.

----End

Viewing Consumer Offsets (Kafka CLI)

- For a Kafka instance with SASL disabled, run the following command in the **/bin** directory of the Kafka client:

  ```
  ./kafka-consumer-groups.sh --bootstrap-server ${connection-address} --offsets --describe --all-groups
  ```

  Parameter description: **connection-address** indicates the Kafka instance address, which can be obtained in the **Connection** area on the **Basic Information** page on the Kafka console.

  Example:

  ```
  [root@ecs-kafka bin]# ./kafka-consumer-groups.sh --bootstrap-server
  192.168.xx.xx:9092,192.168.xx.xx:9092,192.168.xx.xx:9092 --offsets --describe --all-groups

  Consumer group '__consumer-group-dial-test' has no active members.

  GROUP                     TOPIC           PARTITION  CURRENT-OFFSET  LOG-END-OFFSET  LAG
  CONSUMER-ID     HOST            CLIENT-ID
  __consumer-group-dial-test __dms_dial_test 0          350             350             0               -
  -               -
  __consumer-group-dial-test __dms_dial_test 1          350             350             0               -
  -               -
  __consumer-group-dial-test __dms_dial_test 2          350             350             0               -
  -               -

  Consumer group 'test' has no active members.

  GROUP           TOPIC           PARTITION  CURRENT-OFFSET  LOG-END-OFFSET  LAG
  CONSUMER-ID     HOST            CLIENT-ID
  test            topic-01        0          5               5               0               -               -               -
  test            topic-01        1          3               3               0               -               -               -
  test            topic-01        2          10              10              0               -               -               -
  [root@ecs-kafka bin]#
  ```

- For a Kafka instance with SASL enabled, do as follows:

  a. (Optional) If the username and password, and the SSL certificate has been configured, skip this step and go to **b**. Otherwise, do as follows:

     Create the **ssl-user-config.properties** file in the **/config** directory of the Kafka client. Add the username and password, and the SSL certificate configuration by referring to **Step 3**.

  b. Run the following command in the **/bin** directory of the Kafka client:

     ```
     ./kafka-consumer-groups.sh --bootstrap-server ${connection-address} --offsets --describe --all-
     groups --command-config ../config/ssl-user-config.properties
     ```

     Parameter description: **connection-address** indicates the Kafka instance address, which can be obtained in the **Connection** area on the **Basic Information** page on the Kafka console.

     Example:

     ```
     [root@ecs-kafka bin]# ./kafka-consumer-groups.sh --bootstrap-server
     192.168.xx.xx:9093,192.168.xx.xx:9093,192.168.xx.xx:9093 --offsets --describe --all-groups --
     command-config ../config/ssl-user-config.properties

     Consumer group '__consumer-group-dial-test' has no active members.

     GROUP                     TOPIC           PARTITION  CURRENT-OFFSET  LOG-END-OFFSET
     LAG             CONSUMER-ID     HOST            CLIENT-ID
     __consumer-group-dial-test __dms_dial_test 0          347             347             0
     -               -               -
     ```

```
__consumer-group-dial-test __dms_dial_test 1        347          347          0
-              -              -
__consumer-group-dial-test __dms_dial_test 2        347          347          0
-              -              -

Consumer group 'test' has no active members.

GROUP          TOPIC          PARTITION  CURRENT-OFFSET  LOG-END-OFFSET  LAG
CONSUMER-ID     HOST            CLIENT-ID
test           topic-01       0         5            5            0            -            -            -
test           topic-01       1         3            3            0            -            -            -
test           topic-01       2         10           10           0            -            -            -
[root@ecs-kafka bin]#
```

Resetting Consumer Offsets

Step 1 Log in to the console.

Step 2 Click 📍 in the upper left corner to select a region.

NOTE

Select the region where your Kafka instance is located.

Step 3 Click ≡ and choose **Application** > **Distributed Message Service (for Kafka)** to open the console of DMS for Kafka.

Step 4 Click the desired Kafka instance to view the instance details.

Step 5 In the navigation pane, choose the **Consumer Groups** tab.

Step 6 Click the name of the desired consumer group.

Step 7 On the **Consumer Offset** tab page, you can perform the following operations:

- To reset the consumer offset of all partitions of a single topic, click **Reset Consumer Offset** in the row containing the desired topic.
- To reset the consumer offset of a single partition of a single topic, click **Reset Consumer Offset** in the row containing the desired partition.
- To reset the consumer offset of all partitions of all topics, click **Reset Consumer Offset**.

Step 8 In the displayed **Reset Consumer Offset** dialog box, set the parameters by referring to **Table 7-2**.

Table 7-2 Parameters for resetting the consumer offset

Parameter	Description
Reset By	You can reset an offset by: ● Time: Reset the offset to the specified time. ● Offset: Reset the offset to the specified position. If you reset offsets in batches, they can only be reset to the specified time.

Parameter	Description
Time	Set this parameter if **Reset By** is set to **Time**. Select a time point. After the reset is complete, retrieval starts from this time point. ● **Earliest**: earliest offset ● **Custom**: a custom time point ● **Latest**: latest offset
Offset	Set this parameter if **Reset By** is set to **Offset**. Enter an offset, which is greater than or equal to 0. After the reset is complete, retrieval starts from this offset.

Step 9 Click **OK**.

Step 10 Click **Yes** in the confirmation dialog box. The consumer offset is reset.

----End

7.5 Exporting Kafka Consumer Groups

You can export a list of consumer groups in a Kafka instance.

Procedure

Step 1 Log in to the console.

Step 2 Click 📍 in the upper left corner to select a region.

NOTE

Select the region where your Kafka instance is located.

Step 3 Click ☰ and choose **Application** > **Distributed Message Service (for Kafka)** to open the console of DMS for Kafka.

Step 4 Click the desired Kafka instance to view its details.

Step 5 In the navigation pane, choose **Consumer Groups**.

Step 6 Export consumer groups in either of the following ways:

- Select the desired consumer groups and choose **Export** > **Export selected data to an XLSX file** to export specified consumer groups.
- Choose **Export** > **Export all data to an XLSX file** to export all consumer groups.

----End

7.6 Deleting a Kafka Consumer Group

You can delete a consumer group in either of the following ways:

- On the console.
- Use **Kafka CLI**. (Ensure that the Kafka instance version is the same as the CLI version.)

Notes and Constraints

- If **auto.create.groups.enable** is set to **true**, the consumer group status is **EMPTY**, and no offset has been submitted, the system automatically deletes the consumer group 10 minutes later.
- If **auto.create.groups.enable** is set to **false**, the system does not automatically delete consumer groups. You can manually delete them.
- If a consumer group has never committed an offset, the group will be deleted after the Kafka instance restarts.

Prerequisites

The status of the consumer group to be deleted is **EMPTY**.

Deleting a Consumer Group on the Console

Step 1 Log in to the console.

Step 2 Click 📍 in the upper left corner to select a region.

NOTE

Select the region where your Kafka instance is located.

Step 3 Click ☰ and choose **Application** > **Distributed Message Service (for Kafka)** to open the console of DMS for Kafka.

Step 4 Click the desired Kafka instance to view the instance details.

Step 5 In the navigation pane, choose the **Consumer Groups** tab.

Step 6 Delete consumer groups using either of the following methods:

- Select one or more consumer groups and click **Delete Consumer Group** above the consumer group list.
- In the row containing the consumer group you want to delete, click **Delete**.

NOTICE

A consumer group can be deleted only when its status is **EMPTY**.

Consumer group statuses include:

- **DEAD**: The consumer group has no member or metadata.
- **EMPTY**: The consumer group has metadata but has no member.
- **PREPARING_REBALANCE**: The consumer group is to be rebalanced.
- **COMPLETING_REBALANCE**: All members have joined the consumer group.
- **STABLE**: Members in the consumer group can consume messages normally.

Step 7 In the displayed **Delete Consumer Group** dialog box, click **OK**.

----End

Using the CLI to Delete a Consumer Group

The following uses Linux as an example.

- For a Kafka instance with SASL disabled, run the following command in the **/bin** directory of the Kafka client:

  ```
  ./kafka-consumer-groups.sh --bootstrap-server ${connection-address} --delete --group ${consumer-group-name}
  ```

  Parameter description:

  – **connection-address**: can be obtained from the **Connection** area on the **Basic Information** page on the Kafka console.

  – **consumer-group-name**: consumer group name.

  Example:

  ```
  [root@ecs-kafka bin]# ./kafka-consumer-groups.sh --bootstrap-server
  192.168.xx.xx:9092,192.168.xx.xx:9092,192.168.xx.xx:9092 --delete --group group-01
  Deletion of requested consumer groups ('group-01') was successful.
  [root@ecs-kafka bin]#
  ```

- For a Kafka instance with SASL enabled, do as follows:

  a. (Optional) If the SSL certificate has been configured, skip this step and go to **b**. Otherwise, do as follows:

     Create the **ssl-user-config.properties** file in the **/config** directory of the Kafka client and add the SSL certificate configurations by referring to **Step 3**.

  b. In the **/bin** directory of the Kafka client, run the following command:

     ```
     ./kafka-consumer-groups.sh --bootstrap-server ${connection-address} --delete --group ${consumer-group-name} --command-config ../config/ssl-user-config.properties
     ```

     Parameter description:

     ■ **connection-address**: can be obtained from the **Connection** area on the **Basic Information** page on the Kafka console.

     ■ **consumer-group-name**: consumer group name.

     Example:

     ```
     [root@ecs-kafka bin]# ./kafka-consumer-groups.sh --bootstrap-server
     192.168.xx.xx:9093,192.168.xx.xx:9093,192.168.xx.xx:9093 --delete --group group-02 --command-config ../config/ssl-user-config.properties
     Deletion of requested consumer groups ('group-02') was successful.
     [root@ecs-kafka bin]#
     ```

8 Managing Quotas

8.1 Configuring Kafka Quotas

Kafka quotas can be configured for users, clients, or topics to limit the message production or consumption rate.

Rate limits for users and clients work on the entire broker, while topic rate limits work on a specific topic.

Operation Impact

- When the quota is reached, production/consumption latency increases.
- If the quota is small and the production rate is high, production may time out and messages may be lost. As a result, some messages fail to be produced.
- If the initial production/consumption traffic is heavy, and a small quota is set, the production/consumption latency increases and some messages fail to be produced. To ensure stable production and consumption, you are advised to first set the quota to half the traffic, and then half the quota each time you set it until the target quota is reached. For example, if the initial production traffic is 100 MB/s, you can set the production limit to 50 MB/s first. After production becomes stable, change the production limit to 25 MB/s until the target limit is reached.

Prerequisites

- To control user traffic, enable SASL_SSL when creating a Kafka instance and then obtain the username on the **Users** page on the console.
- To control client traffic, obtain the client ID from the client configuration.
- To control topic traffic, obtain the topic name from the **Topics** page.

Creating a User or Client Quota

Step 1 Log in to the console.

Step 2 Click 📍 in the upper left corner to select a region.

NOTE

Select the region where your Kafka instance is located.

Step 3 Click ☰ and choose **Application** > **Distributed Message Service (for Kafka)** to open the console of DMS for Kafka.

Step 4 Click the desired Kafka instance to view the instance details.

Step 5 In the navigation pane, choose **Kafka Quotas** > **Quotas**.

Step 6 Click the **User/Client** tab.

Step 7 In the upper left, click **Create Quota**. The **Create Quota** slide panel is displayed.

Step 8 Set quota parameters.

Table 8-1 Quota parameters

Parameter	Description
Username	Enter the name obtained in **Prerequisites**. To apply the quota to all users, click **Use Default** next to **Username**. After the quota is created, the username cannot be changed.
Client ID	Enter the client ID obtained in **Prerequisites**. To apply the quota to all clients, click **Use Default** next to **Client ID**. After the quota is created, the client ID cannot be changed.
Production Limit	Set an upper limit on the production rate. The unit is MB/s. If this parameter is left blank, no limit is set.
Consumption Limit	Set an upper limit on the consumption rate. The unit is MB/s. If this parameter is left blank, no limit is set.

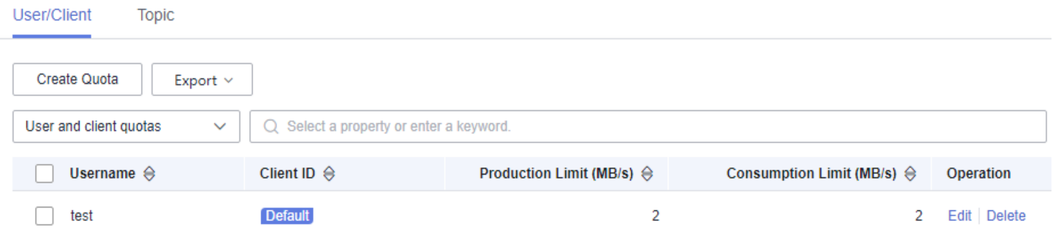
NOTE

- **Username** is not displayed in the **Create Quota** dialog box for instances with SASL disabled.
- **Username** and **Client ID** cannot be both empty.
- **Production Limit** and **Consumption Limit** cannot be both empty.

Step 9 Click **OK**. The **Background Tasks** page is displayed. If the status of the quota creation task is **Successful**, the quota has been created.

Go to the **Kafka Quotas** > **Quotas** page. On the **User/Client** tab page, select **User quotas**, **Client quotas**, or **User and client quotas**, then press **Enter** to view the created quota.

Figure 8-1 Viewing the new quota



☐ Username	Client ID	Production Limit (MB/s)	Consumption Limit (MB/s)	Operation
☐ test	Default	2	2	Edit | Delete

----End

Creating a Topic Quota

Step 1 Log in to the console.

Step 2 Click 📍 in the upper left corner to select a region.

NOTE

Select the region where your Kafka instance is located.

Step 3 Click ☰ and choose **Application** > **Distributed Message Service (for Kafka)** to open the console of DMS for Kafka.

Step 4 Click the desired Kafka instance to view the instance details.

Step 5 In the navigation pane, choose **Kafka Quotas** > **Quotas**.

Step 6 Click the **Topic** tab.

Step 7 In the upper left, click **Create Quota**. The **Create Quota** slide panel is displayed.

Step 8 Set quota parameters.

Table 8-2 Quota parameters

Parameter	Description
Topic Name	Enter the name of the topic to apply the quota to. After the quota is created, the topic cannot be changed.
Production Limit	Set an upper limit on the production rate. The unit is MB/s. If this parameter is left blank, no limit is set.
Consumption Limit	Set an upper limit on the consumption rate. The unit is MB/s. If this parameter is left blank, no limit is set.

NOTE

Production Limit and **Consumption Limit** cannot be both empty.

Step 9 Click **OK**. The **Background Tasks** page is displayed. If the status of the quota creation task is **Successful**, the quota has been created.

Go to the **Kafka Quotas** > **Quotas** page. On the **Topic** tab page, enter the name of the new quota in the upper left corner, then press **Enter** to view the created quota.

----End

Modifying a Quota

Step 1 Log in to the console.

Step 2 Click 📍 in the upper left corner to select a region.

> NOTE
>
> Select the region where your Kafka instance is located.

Step 3 Click ☰ and choose **Application** > **Distributed Message Service (for Kafka)** to open the console of DMS for Kafka.

Step 4 Click the desired Kafka instance to view the instance details.

Step 5 In the navigation pane, choose **Kafka Quotas** > **Quotas**.

Step 6 In the row containing the desired quota, click **Edit**.

Step 7 Change the production limit or consumption limit, and click **OK**. The **Background Tasks** page is displayed. If the status of the quota modification task is **Successful**, the quota has been modified.

Choose **Kafka Quotas** > **Quotas** and view the new production or consumption rate limit.

> NOTE
>
> **Production Limit** and **Consumption Limit** cannot be both empty.

----End

Exporting Quotas

Step 1 Log in to the console.

Step 2 Click 📍 in the upper left corner to select a region.

> NOTE
>
> Select the region where your Kafka instance is located.

Step 3 Click ☰ and choose **Application** > **Distributed Message Service (for Kafka)** to open the console of DMS for Kafka.

Step 4 Click the desired Kafka instance to view the instance details.

Step 5 In the navigation pane, choose **Kafka Quotas** > **Quotas**.

Step 6 Export quotas.

- For specified user/client quotas: On the **User/Client** tab page, select desired user/client quotas and choose **Export** > **Export selected data to an XLSX file**.

- For all user/client quotas: On the **User/Client** tab page, choose **Export** > **Export all data to an XLSX file**.
- For specified topic quotas: On the **Topic** tab page, select desired topic quotas and choose **Export** > **Export selected data to an XLSX file**.
- For all topic quotas: On the **Topic** tab page, choose **Export** > **Export all data to an XLSX file**.

----End

Deleting a Quota

Step 1 Log in to the console.

Step 2 Click 📍 in the upper left corner to select a region.

NOTE

Select the region where your Kafka instance is located.

Step 3 Click ☰ and choose **Application** > **Distributed Message Service (for Kafka)** to open the console of DMS for Kafka.

Step 4 Click the desired Kafka instance to view the instance details.

Step 5 In the navigation pane, choose **Kafka Quotas** > **Quotas**.

Step 6 In the row containing the desired quota, click **Delete**.

Step 7 Click **OK**. The **Background Tasks** page is displayed. If the status of the quota deletion task is **Successful**, the quota has been deleted.

----End

8.2 Monitoring Kafka Quotas

If quotas have been configured for a Kafka instance, the bandwidth usage by user/client/topic of each broker under certain quota policies can be viewed on the console.

Viewing Bandwidth Usage

Step 1 Log in to the console.

Step 2 Click 📍 in the upper left corner to select a region.

NOTE

Select the region where your Kafka instance is located.

Step 3 Click ☰ and choose **Application** > **Distributed Message Service (for Kafka)** to open the console of DMS for Kafka.

Step 4 Click the desired Kafka instance to view the instance details.

Step 5 In the navigation pane, choose **Kafka Quotas** > **Quota Monitoring**.

Step 6 Set the parameters to query bandwidth usage.

Table 8-3 Bandwidth usage query parameters

Parameter	Description
Search By	Specify the criteria by which the bandwidth usage is to be searched. ● **Ranked**: Show the specified number of users, clients, or topics that have used the most bandwidth. ● **Bandwidth**: Show users, clients, or topics whose bandwidth rate is higher than your specified value. ● **Bandwidth usage**: Show users, clients, or topics whose bandwidth usage is higher than your specified percentage.
Bandwidth From	Specify the bandwidth usage data source. ● **Production**: Count production bandwidth usage. ● **Consumption**: Count consumption bandwidth usage.
Dimension	Specify the bandwidth usage data dimension. ● **User/Client**: Count user/client bandwidth usage. ● **Topic**: Count topic bandwidth usage.

Step 7 Click **Search** to view the bandwidth usage of users, clients, and topics of each broker.

----End

9 Managing Instances

9.1 Viewing and Modifying Basic Information of a Kafka Instance

After creating a Kafka instance, you can view the details or modify some parameters of it on the console as required. These parameters include the instance name, description, security group, and capacity threshold policy.

Prerequisite

You can modify basic information of a Kafka instance when the instance is in the **Running** state.

Viewing Kafka Instance Details

Step 1 Log in to the console.

Step 2 Click 📍 in the upper left corner to select a region.

NOTE

Select the region where your Kafka instance is located.

Step 3 Click ☰ and choose **Application** > **Distributed Message Service (for Kafka)** to open the console of DMS for Kafka.

Step 4 You can filter Kafka instances by tag, status, name, version, flavor, used/available storage space, maximum partitions, billing mode, and enterprise project. Only enterprise users can filter instances by enterprise projects. For Kafka instance statuses, see **Table 9-1**.

Table 9-1 Kafka instance status description

Status	Description
Creating	The instance is being created.

Status	Description
Creation failed	The instance failed to be created.
Running	The instance is running properly. Only instances in the **Running** state can provide services.
Faulty	The instance is not running properly.
Restarting	The instance is being restarted.
Changing	The instance specifications or public access configurations are being modified.
Change failed	The instance specifications or public access configurations failed to be modified. You cannot restart, delete, or modify an instance in the **Change failed** state. Contact customer service.

Step 5 Click the name of the desired Kafka instance and view detailed information about the instance on the **Basic Information** tab page.

Table 9-2 describes the parameters for connecting to a Kafka instance. For details about other parameters, see the **Basic Information** tab page of the Kafka instance on the console.

Table 9-2 Connection parameters

Section	Parameter	Description
Connection	Username	Username for accessing the instance with SASL_SSL enabled.
	Kafka SASL_SSL	Whether SASL_SSL is enabled.
	SASL Mechanism	SASL mechanism used by the instance with SASL_SSL enabled.
	SSL Certificate	Click **Download** to download the SSL certificate for accessing the instance.
	Instance Address (Private Network)	Address for connecting to the instance when public access is disabled. The number of connection addresses is the same as that of brokers.
	Manager Address (Private Network)	Address for connecting to Kafka Manager when public access is disabled. Instances created since May 15, 2024 do not have this address.

Section	Parameter	Description
	Manager Username	Username for connecting to Kafka Manager. Instances created since May 15, 2024 do not have this username.
	Public Access	Indicates whether public access has been enabled for the instance.
	Instance Address (Public Network)	Address for connecting to the instance when public access is enabled. This parameter is displayed only when public access is enabled.
	Manager Address (Public Network)	Address for connecting to Kafka Manager when public access is enabled. This parameter is displayed only when **Public Access** is enabled. Instances created since May 15, 2024 do not have this address.
	Intra-VPC Plaintext Access	Whether intra-VPC plaintext access is enabled.

----End

Modifying Basic Information of a Kafka Instance

Step 1 Log in to the console.

Step 2 Click 📍 in the upper left corner to select a region.

NOTE

Select the region where your Kafka instance is located.

Step 3 Click ☰ and choose **Application** > **Distributed Message Service (for Kafka)** to open the console of DMS for Kafka.

Step 4 Click the desired Kafka instance to view its details.

Step 5 Modify the following parameters if needed:

Table 9-3 Modifiable Kafka parameters

Parameter	How to Modify	Result
Instance Name	Click ✎, enter a new name, and click ✓. Naming rules: 4–64 characters; starts with a letter; can contain only letters, digits, hyphens (-), and underscores (_).	The modification result is displayed in the upper right corner of the page.
Enterprise Project	Click ✎, select a new enterprise project from the drop-down list, and click ✓. Only for enterprise users. Modifying this parameter does not restart the instance.	The modification result is displayed in the upper right corner of the page.
Description	Click ✎, enter a new description, and click ✓. 0 to 1024 characters.	The modification result is displayed in the upper right corner of the page.
Security Group	Click ✎, select a new security group from the drop-down list, and click ✓. Modifying this parameter does not restart the instance.	The modification result is displayed in the upper right corner of the page.
Public Access	See **Configuring Kafka Public Access**.	You will be redirected to the **Background Tasks** page, which displays the modification progress and result.
Capacity Threshold Policy	Click the desired policy. In the displayed **Confirm** dialog box, click **OK**. Modifying this parameter does not restart the instance.	You will be redirected to the **Background Tasks** page, which displays the modification progress and result.
Automatic Topic Creation	Enable/Disable this **Automatic Topic Creation**. In the displayed **Confirm** dialog box, click **OK**. Changing this option may restart the instance.	You will be redirected to the **Background Tasks** page, which displays the modification progress and result.
Cross-VPC Access	See **Accessing Kafka Using a VPC Endpoint Across VPCs** and **Accessing Kafka in a Public Network Using DNAT**.	The modification result is displayed in the upper right corner of the page.

----End

9.2 Viewing Kafka Disk Usage

This section describes how to view the disk usage of each broker of a Kafka instance on the console.

Procedure

Step 1 Log in to the console.

Step 2 Click in the upper left corner to select a region.

NOTE

Select the region where your Kafka instance is located.

Step 3 Click ☰ and choose **Application** > **Distributed Message Service (for Kafka)** to open the console of DMS for Kafka.

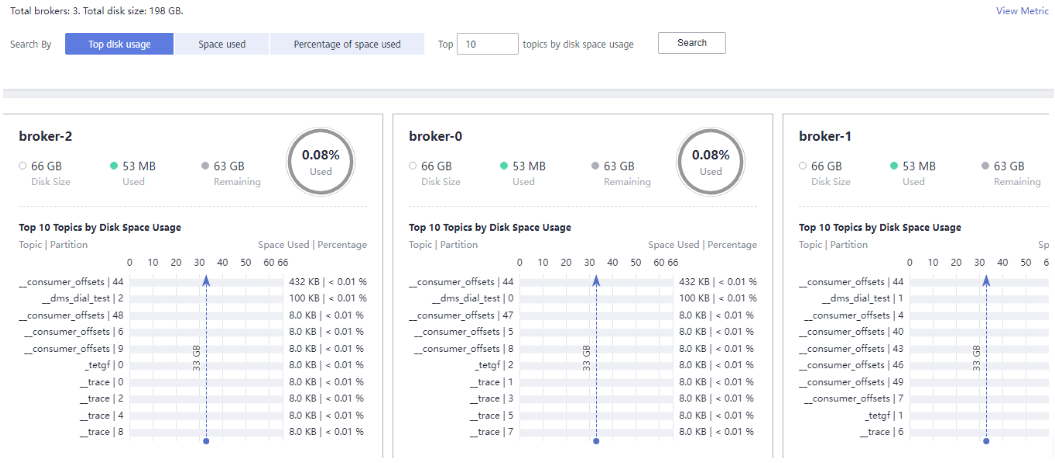
Step 4 Click a Kafka instance to go to the **Basic Information** page.

Step 5 Go to the **Disk Usage Statistics** page.

You can query topics that use the most disk space or topics that have used a specified amount or percentage of disk space.

In the upper right corner of the page, click **View Metric**. On the displayed Cloud Eye page, you can view metrics of Kafka instances.

Figure 9-1 Viewing disk usage



Total brokers: 3. Total disk size: 198 GB.

Search By: Top disk usage | Space used | Percentage of space used — Top 10 topics by disk space usage

broker-2

○ 66 GB Disk Size ● 53 MB Used ● 63 GB Remaining — 0.08% Used

Top 10 Topics by Disk Space Usage

Topic | Partition	Space Used | Percentage
__consumer_offsets | 44	432 KB | < 0.01 %
__dms_dial_test | 2	100 KB | < 0.01 %
__consumer_offsets | 48	8.0 KB | < 0.01 %
__consumer_offsets | 6	8.0 KB | < 0.01 %
__consumer_offsets | 9	8.0 KB | < 0.01 %
_tetgf | 0	8.0 KB | < 0.01 %
__trace | 0	8.0 KB | < 0.01 %
__trace | 2	8.0 KB | < 0.01 %
__trace | 4	8.0 KB | < 0.01 %
__trace | 8	8.0 KB | < 0.01 %

broker-0

○ 66 GB Disk Size ● 53 MB Used ● 63 GB Remaining — 0.08% Used

Top 10 Topics by Disk Space Usage

Topic | Partition	Space Used | Percentage
__consumer_offsets | 44	432 KB | < 0.01 %
__dms_dial_test | 0	100 KB | < 0.01 %
__consumer_offsets | 47	8.0 KB | < 0.01 %
__consumer_offsets | 5	8.0 KB | < 0.01 %
__consumer_offsets | 8	8.0 KB | < 0.01 %
_tetgf | 2	8.0 KB | < 0.01 %
__trace | 1	8.0 KB | < 0.01 %
__trace | 3	8.0 KB | < 0.01 %
__trace | 5	8.0 KB | < 0.01 %
__trace | 7	8.0 KB | < 0.01 %

broker-1

○ 66 GB Disk Size ● 53 MB Used ● 63 GB Remaining

Top 10 Topics by Disk Space Usage

Topic | Partition
__consumer_offsets | 44
__dms_dial_test | 1
__consumer_offsets | 4
__consumer_offsets | 40
__consumer_offsets | 43
__consumer_offsets | 46
__consumer_offsets | 49
__consumer_offsets | 7
_tetgf | 1
__trace | 6

----End

9.3 Viewing Kafka Background Tasks

After you initiate certain instance operations listed in **Table 9-4**, a background task will start for each operation. On the console, you can view the background task status and clear task information by deleting task records.

Table 9-4 Backend task list

Task Name	Description
Creating an instance	Creates a Kafka instance.
Restart Instance	Restarts a Kafka instance.
Modifying Kafka parameters	● Modifies configuration parameters of Kafka. ● Enables/Disables automatic topic creation.
Change capacity threshold policy	Changes capacity threshold policies for a Kafka instance.
Configure public network access	Enables/Disables public access.
Modify Specifications	● Expands the storage space. ● Adds brokers. ● Increases the broker flavor.
Create Quota	Creates user/client/topic quotas.
Modify Quota	Modifies quotas.
Delete Quota	Deletes user/client/topic quotas.
Kafka partition reassignment	Reassigns partitions of a topic.
Configure topic permission	Grants permissions to users in a topic.

Procedure

Step 1 Log in to the console.

Step 2 Click 📍 in the upper left corner to select a region.

NOTE

Select the region where your Kafka instance is located.

Step 3 Click ☰ and choose **Application** > **Distributed Message Service (for Kafka)** to open the console of DMS for Kafka.

Step 4 Click a Kafka instance to go to the **Basic Information** page.

Step 5 In the navigation pane, choose **Background Tasks**.

Step 6 On the **Background tasks** or **Scheduled tasks** tab page, click the time drop-down box, specify time, enter keywords in the search box, and press **Enter**. Tasks started in the specified time will be displayed.

On the **Background Tasks** page, you can also perform the following operations:

- Click ↻ to refresh the task status.
- Click **Delete**. In the displayed **Delete Task** dialog box, click **OK** to clear the task information.

  NOTE

  You can delete a task only when it is in either of the following situations:

  – The task is complete, which can be **Successful** or **Failed**.

  – The task is **Canceled**.

----End

9.4 Viewing Sample Code of Kafka Production and Consumption

Distributed Message Service for Kafka allows you to view sample Java, Go, and Python code of producing and consuming messages on the console. You can quickly complete Kafka client integration.

Procedure

Step 1 Log in to the console.

Step 2 Click 📍 in the upper left corner to select a region.

NOTE

Select the region where your Kafka instance is located.

Step 3 Click ☰ and choose **Application** > **Distributed Message Service (for Kafka)** to open the console of DMS for Kafka.

Step 4 Click the desired Kafka instance to view the instance details.

Step 5 In the navigation pane, choose **Topics**.

Step 6 Click **View Sample Code**. The **Sample Code** dialog box is displayed.

View sample code for creating and retrieving messages in Java, Go, and Python. Set **Access By** to **PlainText** to view the sample code where SASL_SSL authentication is disabled. Set **Access By** to **SASL_SSL** to view the sample code where SASL_SSL authentication is enabled.

----End

9.5 Modifying Kafka Instance Configuration Parameters

Your Kafka instances, topics, and consumers come with default configuration parameter settings. You can modify common parameters on the Kafka console. For details about parameters that are not listed on the console, see the **Kafka official website**.

Kafka instances have dynamic and static parameters:

- Dynamic parameters: Modifying dynamic parameters will not restart the instance.
- Static parameters: After static parameters are modified, you must manually restart the instance.

NOTE

Configuration parameters of some old instances cannot be modified. Check whether your instance parameters can be modified on the console. If they cannot be modified, contact customer service.

Prerequisites

You can modify configuration parameters of a Kafka instance when the instance is in the **Running** state.

Procedure

Step 1 Log in to the console.

Step 2 Click 📍 in the upper left corner to select a region.

NOTE

Select the region where your Kafka instance is located.

Step 3 Click ☰ and choose **Application** > **Distributed Message Service (for Kafka)** to open the console of DMS for Kafka.

Step 4 Click the desired Kafka instance to view the instance details.

Step 5 On the **Parameters** page, click **Edit** in the row containing the parameter to modify.

Parameters of v1.1.0 instances are described in **Table 9-5** and **Table 9-6**. Parameters of v2.3.0/v2.7 instances are described in **Table 9-7** and **Table 9-8**.

Table 9-5 Dynamic parameters (v1.1.0 instances)

Parameter	Description	Value Range	Default Value
auto.create.groups.enable	Whether to automatically create consumer groups. This parameter is not displayed on the console for some earlier instances. The function of automatically creating consumer groups is enabled by default and cannot be disabled on the console.	true/false	true
offsets.retention.minutes	The longest period a consumption position can be retained starts from the time of submission. Positions retained beyond this duration will be deleted. Each time a consumption position is submitted to a topic partition, its retention period resets to 0. This parameter is displayed as a static one for certain earlier instances.	1440–30240 Unit: minute	20160

Table 9-6 Static parameters (v1.1.0 instances)

Parameter	Description	Value Range	Default Value
min.insync.replicas	If a producer sets the acks parameter to **all** (or **-1**), the **min.insync.replicas** parameter specifies the minimum number of replicas that must acknowledge a write for the write to be considered successful.	1–3	1
message.max.bytes	Maximum length of a single message.	0–10,485,760 Unit: byte	10,485,760
unclean.leader.election.enable	Indicates whether to allow replicas not in the ISR set to be elected as the leader as a last resort, even though doing so may result in data loss.	**true** or **false**	false

Parameter	Description	Value Range	Default Value
connections.max.idle.ms	Idle connection timeout (in ms). Connections that are idle for the duration specified by this parameter will be closed.	5000–600,000 Unit: millisecond	600,000
log.retention.hours	Maximum duration for storing log files. This parameter takes effect only for topics that have no aging time configured. If there is aging time configured for topics, it overrides this parameter.	1–168 Unit: hour	72
max.connections.per.ip	The maximum number of connections allowed from each IP address. Request for new connections will be rejected once the limit is reached.	100–20,000	1000
group.max.session.timeout.ms	Maximum session timeout for consumers. A longer timeout gives consumers more time to process messages between heartbeats but results in a longer time to detect failures.	6000–1,800,000 Unit: millisecond	1,800,000
default.replication.factor	The default number of replicas configured for an automatically created topic.	1–3	3
allow.everyone.if.no.acl.found	When this parameter is set to **true**, all users can access resources without ACL rules. This parameter is displayed only when SASL is enabled for the instance. This parameter of some earlier instances cannot be modified.	true/false	true
num.partitions	The default number of partitions configured for each automatically created topic.	1 ~ 200	3
group.min.session.timeout.ms	Minimum session timeout for consumers. A shorter timeout enables quicker failure detection but results in more frequent consumer heartbeating, which can overwhelm broker resources.	6000–300,000 Unit: millisecond	6000

Table 9-7 Dynamic parameters (v2.3.0/v2.7)

Parameter	Description	Value Range	Default Value
min.insync.replicas	If a producer sets the acks parameter to **all** (or **-1**), the **min.insync.replicas** parameter specifies the minimum number of replicas that must acknowledge a write for the write to be considered successful.	1–3	1
message.max.bytes	Maximum length of a single message.	0–10,485,760 Unit: byte	10,485,760
auto.create.groups.enable	Whether to automatically create consumer groups. This parameter is not displayed on the console for some earlier instances. The function of automatically creating consumer groups is enabled by default and cannot be disabled on the console.	true/false	true
max.connections.per.ip	The maximum number of connections allowed from each IP address. Request for new connections will be rejected once the limit is reached.	100–20,000	1000
unclean.leader.election.enable	Indicates whether to allow replicas not in the ISR set to be elected as the leader as a last resort, even though doing so may result in data loss.	**true** or **false**	false
offsets.retention.minutes	The longest period a consumption position can be retained starts from the time of submission. Positions retained beyond this duration will be deleted. Each time a consumption position is submitted to a topic partition, its retention period resets to 0. This parameter is displayed as a static one for certain earlier instances.	1440–30240 Unit: minute	20160

Table 9-8 Static parameters (v2.3.0/v2.7)

Parameter	Description	Value Range	Default Value
connections.max.idle.ms	Idle connection timeout (in ms). Connections that are idle for the duration specified by this parameter will be closed.	5000–600,000 Unit: millisecond	600,000
log.retention.hours	Maximum duration for storing log files. This parameter takes effect only for topics that have no aging time configured. If there is aging time configured for topics, it overrides this parameter.	1–168 Unit: hour	72
group.max.session.timeout.ms	Maximum session timeout for consumers. A longer timeout gives consumers more time to process messages between heartbeats but results in a longer time to detect failures.	6000–1,800,000 Unit: millisecond	1,800,000
default.replication.factor	The default number of replicas configured for an automatically created topic.	1–3	3
allow.everyone.if.no.acl.found	When this parameter is set to **true**, all users can access resources without ACL rules. This parameter is displayed only when SASL is enabled for the instance. This parameter of some earlier instances cannot be modified.	true/false	true
num.partitions	The default number of partitions configured for each automatically created topic.	1 ~ 200	3
group.min.session.timeout.ms	Minimum session timeout for consumers. A shorter timeout enables quicker failure detection but results in more frequent consumer heartbeating, which can overwhelm broker resources.	6000–300,000 Unit: millisecond	6000

NOTE

- To modify multiple dynamic or static parameters at a time, click **Modify** above the parameter list.
- If you want to restore the default values, click **Restore Default** in the row containing the desired parameter.

Step 6 Click **Save**.

NOTE

Modifying dynamic parameters will not restart the instance. **Static parameter modification requires manual restart of the instance.**

----End

9.6 Configuring Kafka Instance Tags

Tags facilitate Kafka instance identification and management.

You can add tags to a Kafka instance when creating the instance or add tags on the **Tags** tab page of the created instance. Up to 20 tags can be added to an instance. Tags can be deleted.

A tag consists of a tag key and a tag value. **Table 9-9** lists the tag key and value requirements.

Table 9-9 Tag key and value requirements

Parameter	Requirements
Tag key	• Cannot be left blank. • Must be unique for the same instance. • Can contain 1 to 128 characters. • Can contain letters, digits, spaces, and special characters _ . : = + - @ • Cannot start or end with a space. • Cannot start with **_sys_**.
Tag value	• Can contain 0 to 255 characters. • Can contain letters, digits, spaces, and special characters _ . : = + - @ • Cannot start or end with a space in instance creation.

Procedure

Step 1 Log in to the console.

Step 2 Click 📍 in the upper left corner to select a region.

NOTE

Select the region where your Kafka instance is located.

Step 3 Click ≡ and choose **Application** > **Distributed Message Service (for Kafka)** to open the console of DMS for Kafka.

Step 4 Click the name of an instance.

Step 5 In the navigation pane on the left, choose **Tags**.

View the tags of the instance.

Step 6 Perform the following operations as required:

- Add a tag
  a. Click **Create/Delete Tag**.
  b. Enter a tag key and a tag value, and click **Add**.
     If you have predefined tags, select a predefined pair of tag key and value, and click **Add**.
  c. Click **OK**.
- Delete a tag
  Delete a tag using either of the following methods:
  – In the row containing the tag to be deleted, click **Delete**. In the **Delete Tag** dialog box, click **Yes**.
  – Click **Create/Delete Tag**. In the dialog box that is displayed, click ⊗ next to the tag to be deleted and click **OK**.

----End

9.7 Exporting the Kafka Instance List

You can export a list of instances on the DMS for Kafka console.

Procedure

Step 1 Log in to the console.

Step 2 Click 📍 in the upper left corner to select a region.

NOTE

Select the region where your Kafka instance is located.

Step 3 Click ≡ in the upper left corner and choose **Application** > **Distributed Message Service (for Kafka)** to open the console of DMS for Kafka.

Step 4 Export the instance list using either of the following methods:

- Select the desired instances and choose **Export** > **Export selected data to an XLSX file** to export specified instances.
- Choose **Export** > **Export all data to an XLSX file** to export all instances.

----End

9.8 Restarting a Kafka Instance

You can restart one or more Kafka instances in batches on the DMS for Kafka console.

NOTICE

When a Kafka instance is being restarted, message retrieval and creation requests of clients will be rejected.

Prerequisites

The status of the Kafka instance you want to restart is either **Running** or **Faulty**.

Procedure

Step 1 Log in to the console.

Step 2 Click 📍 in the upper left corner to select a region.

NOTE

Select the region where your Kafka instance is located.

Step 3 Click ☰ and choose **Application** > **Distributed Message Service (for Kafka)** to open the console of DMS for Kafka.

Step 4 Restart Kafka instances using one of the following methods:

- Select one or more Kafka instances and click **Restart** in the upper left corner.
- In the row containing the desired instance, click **Restart**.
- Click the desired Kafka instance to go to the instance details page. In the upper right corner, click **Restart**.

Step 5 In the **Restart Instance** dialog box, click **Yes** to restart the Kafka instance.

It takes 3 to 15 minutes to restart a Kafka instance. After the instance is successfully restarted, its status should be **Running**.

NOTE

Restarting a Kafka instance only restarts the instance process and does not restart the VM where the instance is located.

----End

9.9 Deleting Kafka Instances

Delete one or more Kafka instances at a time on the DMS for Kafka console.

NOTICE

Deleting a Kafka instance will delete the data in the instance without any backup. Exercise caution when performing this operation.

Prerequisites

The status of the Kafka instance you want to delete is **Running** or **Faulty**.

Procedure

Step 1 Log in to the console.

Step 2 Click 📍 in the upper left corner to select a region.

NOTE

Select the region where your Kafka instance is located.

Step 3 Click ☰ and choose **Application** > **Distributed Message Service (for Kafka)** to open the console of DMS for Kafka.

Step 4 Delete pay-per-use Kafka instances in either of the following ways:

- Select one or more Kafka instances and click **Delete** in the upper left corner.
- In the row containing the Kafka instance to be deleted, choose **More** > **Delete**.
- Click the desired Kafka instance to go to the instance details page. In the upper right corner, choose **More** > **Delete**.

NOTE

Kafka instances in the **Creating**, **Changing**, **Change failed**, or **Restarting** state cannot be deleted.

Step 5 In the **Delete Instance** dialog box, enter **DELETE** and click **OK** to delete the Kafka instance.

It takes 1 to 60 seconds to delete a Kafka instance.

----End

9.10 Using Kafka Manager

9.10.1 Accessing Kafka Manager

Kafka Manager is an open-source tool for managing Kafka. It can be used only through a web browser. In Kafka Manager, you can view the monitoring statistics and broker information of your Kafka clusters.

Instances created since May 15, 2024 do not have Kafka Manager. Kafka Manager's functions are provided on the Kafka console.

Table 9-10 Kafka Manager functions on the Kafka console

Kafka Manager	Kafka Console
Viewing topics about an instance	View the topic list on the **Topics** page.
Viewing basic information about a topic	View the basic information (including the number of replicas, number of partitions, and aging time) about each topic on the **Topics** page.
Reassigning topic partitions	Reassign partitions automatically or manually on the **Topics** page.
Updating topic configurations	Modify topic configuration parameters on the **Topics** page.
Viewing the consumer group list	View the consumer group list on the **Consumer Groups** page.
Viewing details about a specific consumer	On the **Consumer Groups** page, click a consumer group name to go to the consumer group details page and view consumers and their progress.
Viewing details of topics in a consumer group	On the **Consumer Groups** page, click a consumer group name to go to the consumer group details page. On the **Consumer Offset** tab page, view the topic list of the consumer group, the number of messages accumulated in each topic, and the consumption status of each partition.
Monitoring the cluster or topics	View monitoring information on the **Monitoring** page.

Prerequisites

Security group rules have been configured by referring to **Table 9-11**.

Table 9-11 Security group rule

Direction	Protocol	Port	Source	Description
Inbound	TCP	9999	IP address or IP address group of the Kafka client	Access Kafka Manager.

Logging In to Kafka Manager

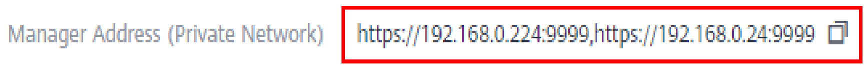
Step 1 Create a Windows ECS with the same VPC and security group configurations as the Kafka instance. For details, see **Purchasing an ECS**.

If public access has been enabled, this step is optional. You can access the instance using the local browser. You do not need to create a Windows ECS.

Step 2 Obtain the Kafka Manager address on the instance details page.

- If public network access has been disabled, the Kafka Manager address is **Manager Address (Private Network)**.

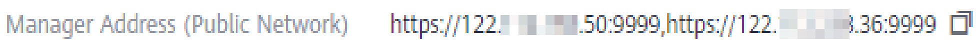
  Figure 9-2 Kafka Manager address (private network)

  Manager Address (Private Network) https://192.168.0.224:9999,https://192.168.0.24:9999

- If public network access has been enabled, the Kafka Manager address is **Manager Address (Public Network)**.

  Figure 9-3 Kafka Manager address (public network)

  Manager Address (Public Network) https://122.[redacted].50:9999,https://122.[redacted].36:9999

Step 3 Enter the Kafka Manager address in the web browser in the Windows ECS.

If public access is enabled, enter the Kafka Manager address in the address bar of the browser on the local PC. If public access is not enabled, log in to the ECS prepared in **Step 1** and enter the Kafka Manager address in the address bar of the browser on the ECS.

Step 4 Enter the username and password for logging in to Kafka Manager, which you set when creating the instance.

----End

Viewing Information in Kafka Manager

In Kafka Manager, you can view the monitoring statistics and broker information of your Kafka clusters.

- Information about clusters

  Click **Clusters** to view the information about clusters. **Figure 9-4** shows an example of the cluster information.

  – The top navigation bar provides the following functions, as shown in the red box 1 in the figure.

    ▪ **Cluster**: viewing the list of clusters and cluster information.

    ▪ **Brokers**: viewing information about brokers of a cluster.

    ▪ **Topic**: viewing information about topics in a cluster.

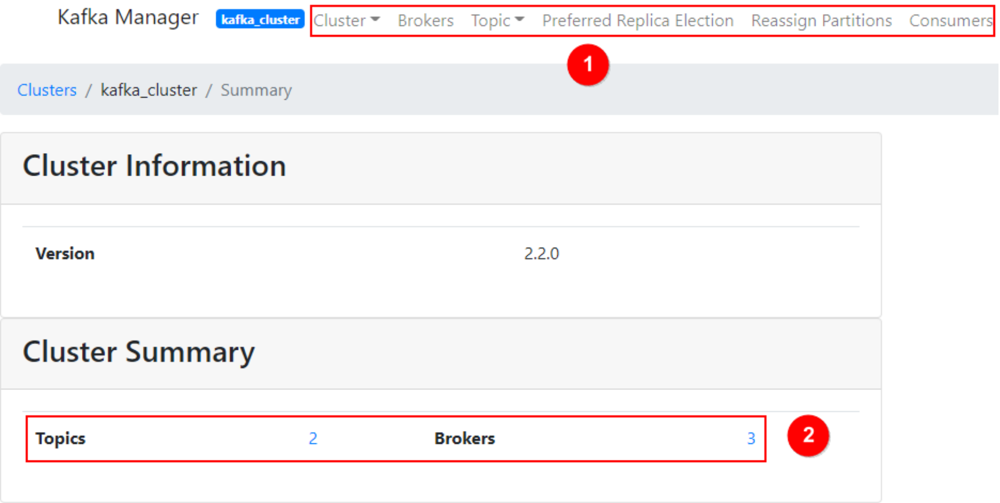
    ▪ **Preferred Replica Election**: electing the leader (preferred replica) of a topic. This operation is not recommended.

    ▪ **Reassign Partitions**: reassigning partitions. This operation is not recommended.

- **Consumers**: viewing the status of consumer groups in a cluster.

– Red box 2 shows an example of the cluster information summary, including the number of topics and brokers in the cluster.

Figure 9-4 Information about clusters



Cluster Information

Version	2.2.0

Cluster Summary

Topics	2	Brokers	3

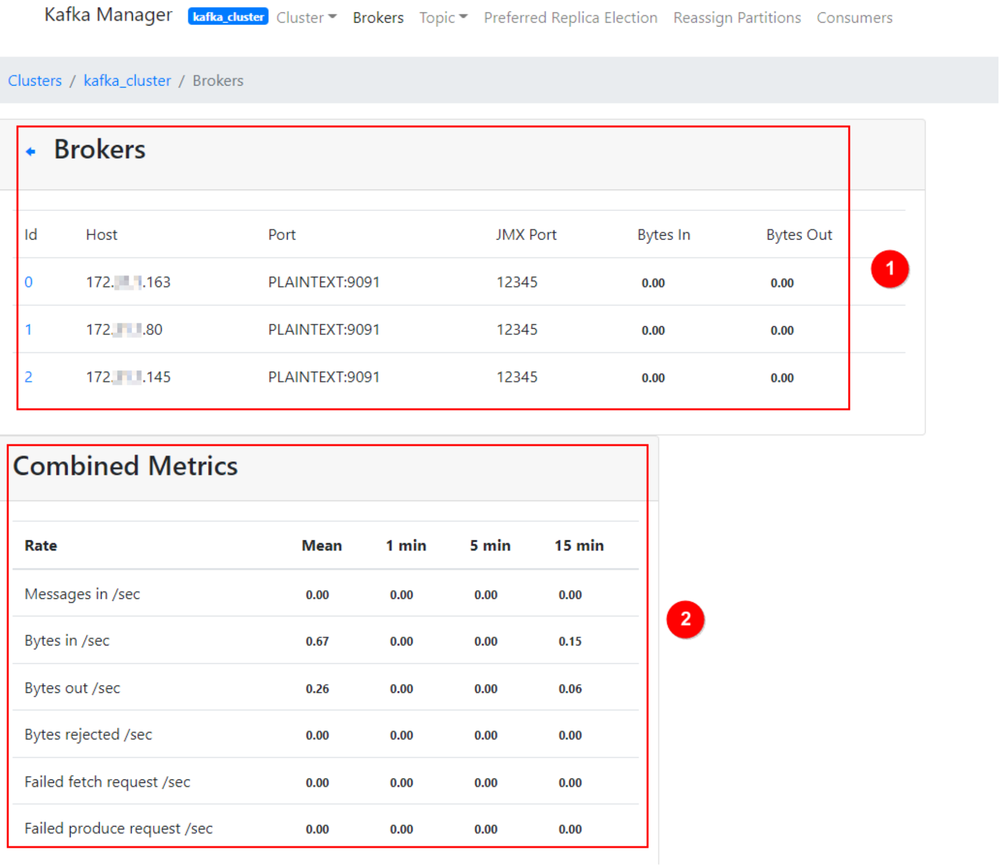
- Combined information about all brokers of a cluster

  This page shows statistics of brokers of a cluster. **Figure 9-5** shows an example of the storage configuration.

  – Red box 1 shows the list of brokers, including number of incoming and outgoing bytes of different brokers.

  – Red box 2 shows the monitoring metrics of the cluster.

Figure 9-5 Viewing the combined information about all brokers in a cluster



Brokers

Id	Host	Port	JMX Port	Bytes In	Bytes Out
0	172.[redacted].163	PLAINTEXT:9091	12345	0.00	0.00
1	172.[redacted].80	PLAINTEXT:9091	12345	0.00	0.00
2	172.[redacted].145	PLAINTEXT:9091	12345	0.00	0.00

Combined Metrics

Rate	Mean	1 min	5 min	15 min
Messages in /sec	0.00	0.00	0.00	0.00
Bytes in /sec	0.67	0.00	0.00	0.15
Bytes out /sec	0.26	0.00	0.00	0.06
Bytes rejected /sec	0.00	0.00	0.00	0.00
Failed fetch request /sec	0.00	0.00	0.00	0.00
Failed produce request /sec	0.00	0.00	0.00	0.00

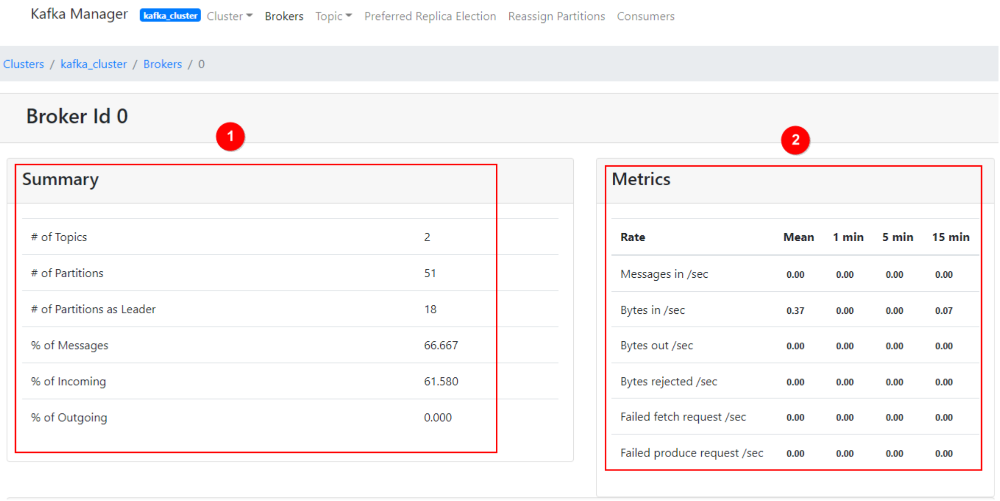
- Information about a specific broker

  Click the ID of a broker to view its statistics. **Figure 9-6** shows an example of the storage configuration.

  – Red box 1 shows the statistics of the broker, including the numbers of topics, partitions, and leaders, and percentages of messages, incoming traffic, and outgoing traffic.

  – Red box 2 shows the monitoring metrics of the broker.

Figure 9-6 Viewing information about a broker



Broker Id 0

Summary

# of Topics	2
# of Partitions	51
# of Partitions as Leader	18
% of Messages	66.667
% of Incoming	61.580
% of Outgoing	0.000

Metrics

Rate	Mean	1 min	5 min	15 min
Messages in /sec	0.00	0.00	0.00	0.00
Bytes in /sec	0.37	0.00	0.00	0.07
Bytes out /sec	0.00	0.00	0.00	0.00
Bytes rejected /sec	0.00	0.00	0.00	0.00
Failed fetch request /sec	0.00	0.00	0.00	0.00
Failed produce request /sec	0.00	0.00	0.00	0.00

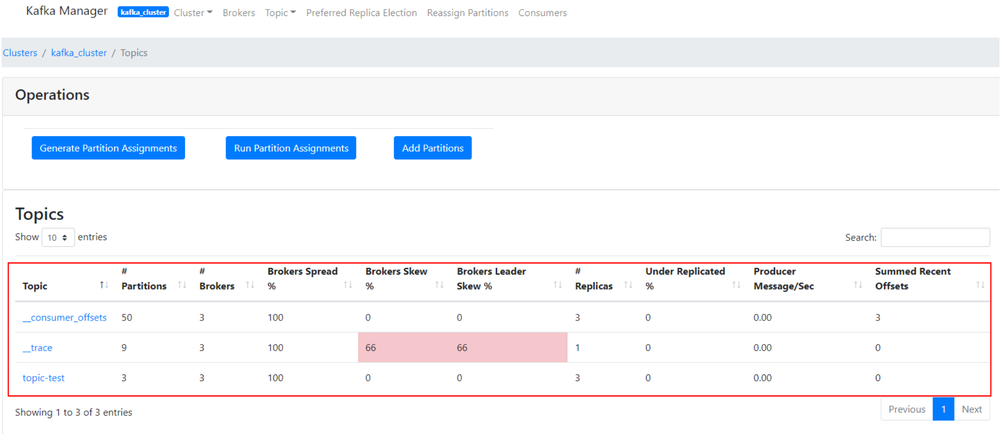
- Topics of an instance

  In the navigation bar, choose **Topic** > **List**. The displayed page shows the list of topics and information about the topics, as shown in **Figure 9-7**.

NOTICE

Topics starting with "__" are internal topics. To avoid service faults, do not perform any operation on these topics.

Figure 9-7 Topics of an instance



Topics

Show 10 entries

Topic	# Partitions	# Brokers	Brokers Spread %	Brokers Skew %	Brokers Leader Skew %	# Replicas	Under Replicated %	Producer Message/Sec	Summed Recent Offsets
__consumer_offsets	50	3	100	0	0	3	0	0.00	3
__trace	9	3	100	66	66	1	0	0.00	0
topic-test	3	3	100	0	0	3	0	0.00	0

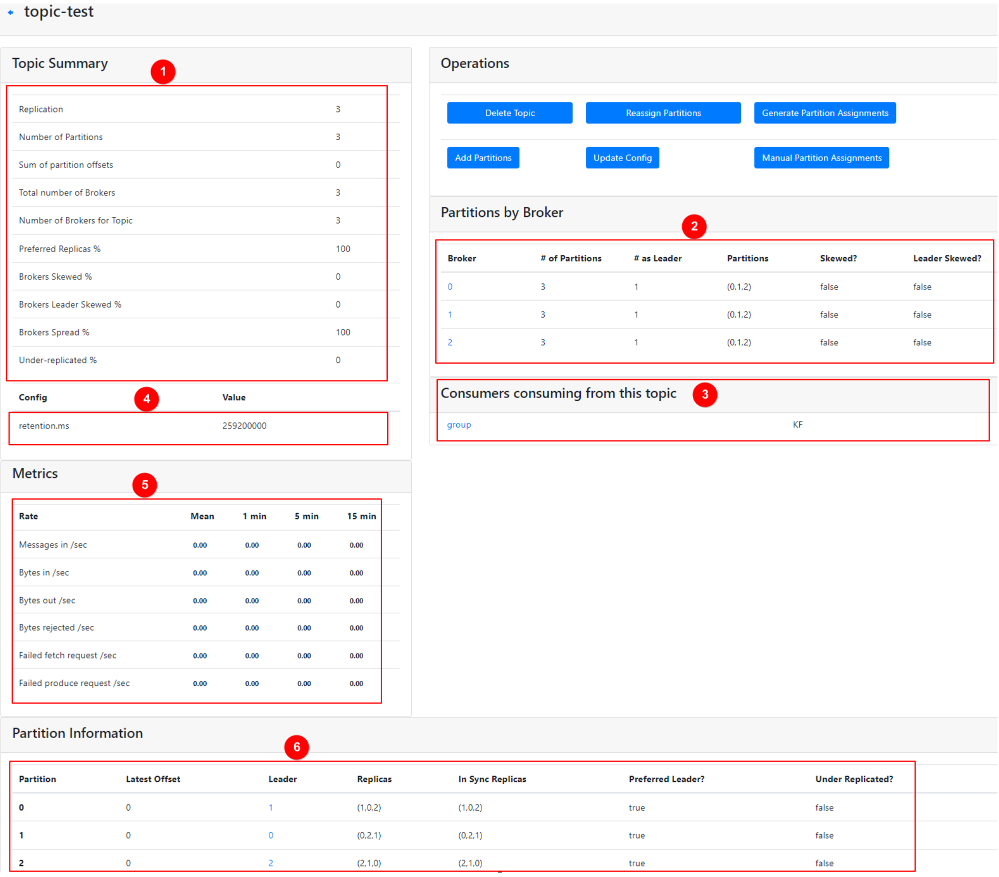
Showing 1 to 3 of 3 entries

- Details of a topic

  Click the name of a topic to view its details on the displayed page, as shown in **Figure 9-8**.

  - Red box 1: basic information about the topic, including **Replication**, **Number of Partitions**, and **Sum of Partition Offsets**.
  - Red box 2: information about partitions of different brokers.
  - Red box 3: consumer groups of the topic. Click the name of a consumer group name to view its details.
  - Red box 4: configurations of the topic. For details, see **https://kafka.apache.org/documentation/#topicconfigs**.
  - Red box 5: monitoring metrics of the topic.
  - Red box 6: information about partitions in the topic, including **Latest Offset**, **Leader** of a partition, **Replicas**, and **In Sync Replicas**.

Figure 9-8 Details of a topic



topic-test

Topic Summary

Replication	3
Number of Partitions	3
Sum of partition offsets	0
Total number of Brokers	3
Number of Brokers for Topic	3
Preferred Replicas %	100
Brokers Skewed %	0
Brokers Leader Skewed %	0
Brokers Spread %	100
Under-replicated %	0

Config	Value
retention.ms	259200000

Operations

Partitions by Broker

Broker	# of Partitions	# as Leader	Partitions	Skewed?	Leader Skewed?
0	3	1	(0,1,2)	false	false
1	3	1	(0,1,2)	false	false
2	3	1	(0,1,2)	false	false

Consumers consuming from this topic

group	KF

Metrics

Rate	Mean	1 min	5 min	15 min
Messages in /sec	0.00	0.00	0.00	0.00
Bytes in /sec	0.00	0.00	0.00	0.00
Bytes out /sec	0.00	0.00	0.00	0.00
Bytes rejected /sec	0.00	0.00	0.00	0.00
Failed fetch request /sec	0.00	0.00	0.00	0.00
Failed produce request /sec	0.00	0.00	0.00	0.00

Partition Information

Partition	Latest Offset	Leader	Replicas	In Sync Replicas	Preferred Leader?	Under Replicated?
0	0	1	(1,0,2)	(1,0,2)	true	false
1	0	0	(0,2,1)	(0,2,1)	true	false
2	0	2	(2,1,0)	(2,1,0)	true	false

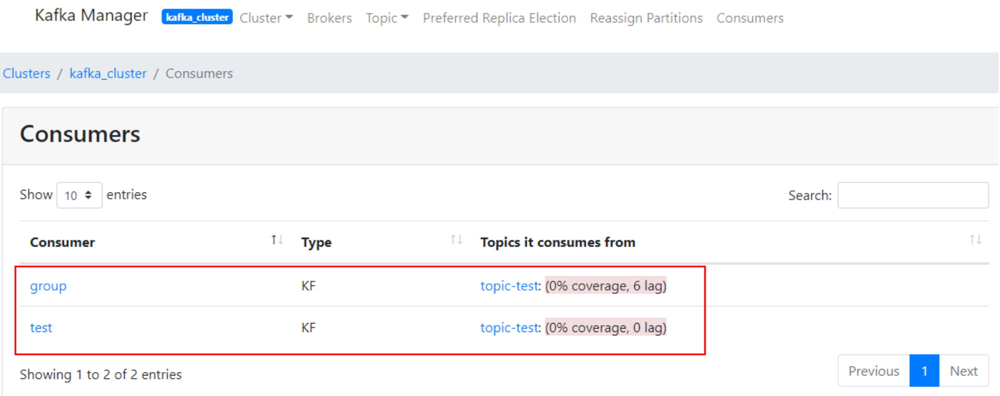
- List of consumers

  Click **Consumers** to view the list of consumers in a cluster.

  NOTE

  Only consumer groups that have retrieved messages in the last 14 days are displayed.

Figure 9-9 Viewing the list of consumers



Consumers

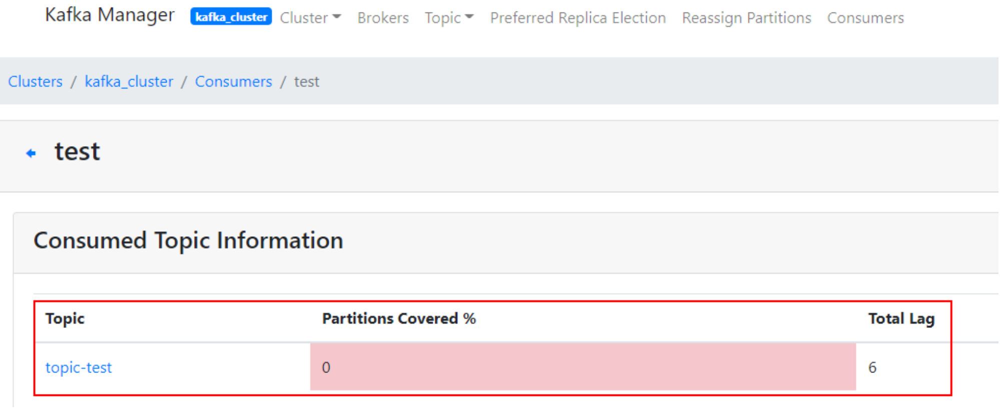
Consumer	Type	Topics it consumes from
group	KF	topic-test: (0% coverage, 6 lag)
test	KF	topic-test: (0% coverage, 0 lag)

Showing 1 to 2 of 2 entries

- Details of a specific consumer

  Click the name of a consumer to view its details, including the list of topics in the consumer and the number of messages that can be retrieved in each topic (**Total Lag**).

Figure 9-10 Viewing consumer details



Clusters / kafka_cluster / Consumers / test

test

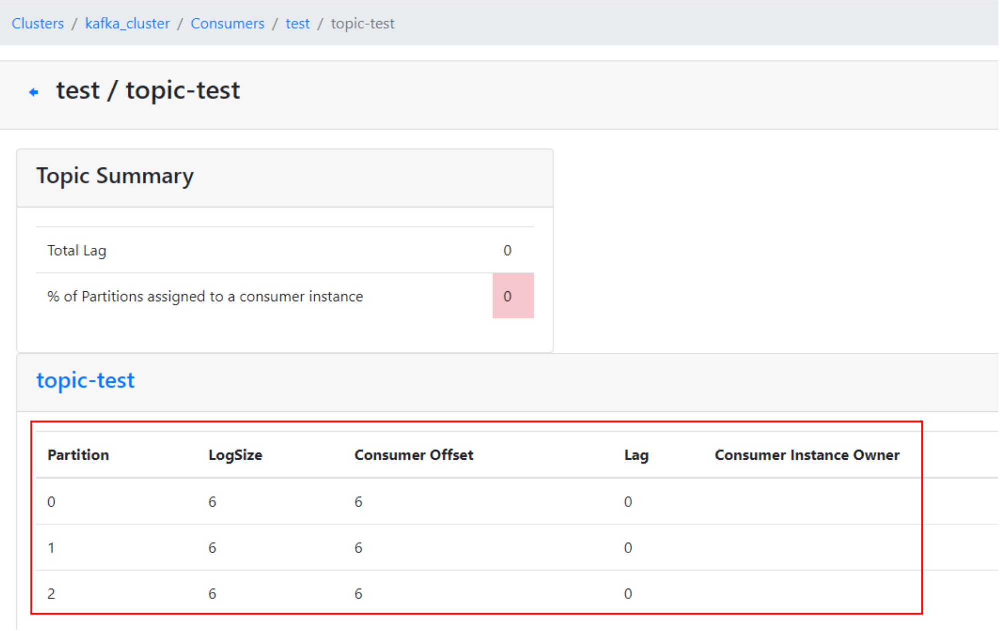
Consumed Topic Information

Topic	Partitions Covered %	Total Lag
topic-test	0	6

- Details of topics in a consumer

  Click the name of a topic to view consumption details of different partitions in the topic, including **Partition**, the number of messages in a partition (**LogSize**), progress of the retrieval (**Consumer Offset**), number of remaining messages in the partition that can be retrieved (**Lag**), and the latest consumer that retrieved from the partition (**Consumer Instance Owner**).

Figure 9-11 Viewing details of a topic



Clusters / kafka_cluster / Consumers / test / topic-test

test / topic-test

Topic Summary

Total Lag	0
% of Partitions assigned to a consumer instance	0

topic-test

Partition	LogSize	Consumer Offset	Lag	Consumer Instance Owner
0	6	6	0	
1	6	6	0	
2	6	6	0	

9.10.2 Resetting Kafka Manager Password

You can reset the password of Kafka Manager of a Kafka instance if you forget it.

This function is not available for instances created since May 15, 2024.

Prerequisites

A Kafka instance has been created and is in the **Running** state.

Procedure

Step 1 Log in to the console.

Step 2 Click 📍 in the upper left corner to select a region.

NOTE

Select the region where your Kafka instance is located.

Step 3 Click ☰ and choose **Application** > **Distributed Message Service (for Kafka)** to open the console of DMS for Kafka.

Step 4 Reset the Kafka Manager password using either of the following methods:

- In the row containing the desired Kafka instance, choose **More** > **Reset Manager Password**.
- Click the desired Kafka instance to go to the instance details page. In the upper right corner, choose **More** > **Reset Manager Password**.
- Click the desired Kafka instance to go to the instance details page. On the **Basic Information** page, click **Reset Manager Password** next to **Manager Username** in the **Connection** section.

Step 5 Enter and confirm a new password, and click **OK**.

- If the password is successfully reset, a success message is displayed.
- If the password fails to be reset, a failure message is displayed. Reset the password again. If you still fail to reset the password after multiple attempts, contact customer service.

NOTE

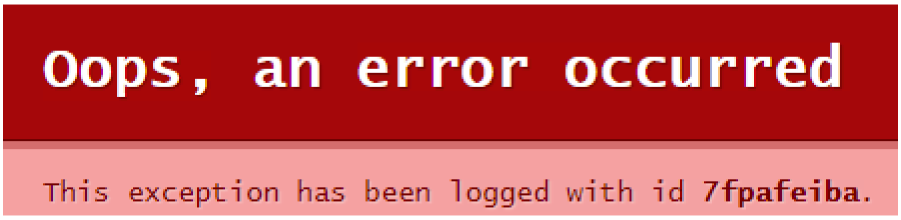
The system will display a success message only after the password is successfully reset on all brokers.

----End

9.10.3 Restarting Kafka Manager

Restart Kafka Manager when you fail to log in to it or it cannot provide services as usual.

Figure 9-12 Error information



Oops, an error occurred

This exception has been logged with id **7fpafeiba**.

NOTE

Restarting Kafka Manager does not affect services.

This function is not available for instances created since May 15, 2024.

Procedure

Step 1 Log in to the console.

Step 2 Click 📍 in the upper left corner to select a region.

NOTE

Select the region where your Kafka instance is located.

Step 3 Click ☰ and choose **Application** > **Distributed Message Service (for Kafka)** to open the console of DMS for Kafka.

Step 4 Restart Kafka Manager using either of the following methods:

- In the row containing the desired Kafka instance, choose **More** > **Restart Kafka Manager**.
- Click the desired Kafka instance to go to the instance details page. In the upper right corner, choose **More** > **Restart Kafka Manager**.

Step 5 Click **Yes**.

You can view the operation progress on the **Background Tasks** page. If the task status is **Successful**, the restart has succeeded.

----End

9.10.4 Disabling Kafka Manager

Kafka Manager consumes memory and CPU. To free some resources, disable this function. This section describes how to disable Kafka Manager on the console.

NOTE

- Once disabled, Kafka Manager cannot be enabled.
- Disabling Kafka Manager does not restart the instance.

Procedure

Step 1 Log in to the console.

Step 2 Click 📍 in the upper left corner to select a region.

NOTE

Select the region where your Kafka instance is.

Step 3 Click ☰ and choose **Application** > **Distributed Message Service (for Kafka)** to open the console of DMS for Kafka.

Step 4 Click the name of the desired Kafka instance.

Step 5 On the **Basic Information** tab page, click ⬤ next to **Kafka Manager** in the **Connection** area.

NOTE

After Kafka Manager is disabled, the Kafka Manager connection address will not be displayed on the console, the Kafka Manager password cannot be reset, and Kafka Manager cannot be restarted.

----End

10 Modifying Kafka Instance Specifications

After creating a Kafka instance, you can expand its storage space, increase the number of brokers, or increase the broker flavor. You can change instances to higher specifications, but not lower ones.

Only one object can be modified per operation: broker quantity, storage space, or broker flavor.

Impact of Specification Modification

It takes 5 to 10 minutes to modify specifications on one broker. The more brokers, the longer time the modification takes.

Table 10-1 Impact of specification modification

Modified Object	Impact
Broker quantity	• Adding brokers does not affect the original brokers or services. • When brokers are added, the storage space is proportionally expanded based on the current disk space. For example, assume that the original number of brokers of an instance is 3 and the disk size of each broker is 200 GB. If the broker quantity changes to 10 and the disk size of each broker is still 200 GB, the total disk size becomes 2000 GB. • New topics are created on new brokers, and the original topics are still on the original brokers, resulting in unbalanced partitions. You can **reassign partitions** to migrate the replicas of the original topic partitions to the new brokers.
Storage space	• You can expand the storage space 20 times. • Storage space expansion does not affect services.

Modified Object	Impact

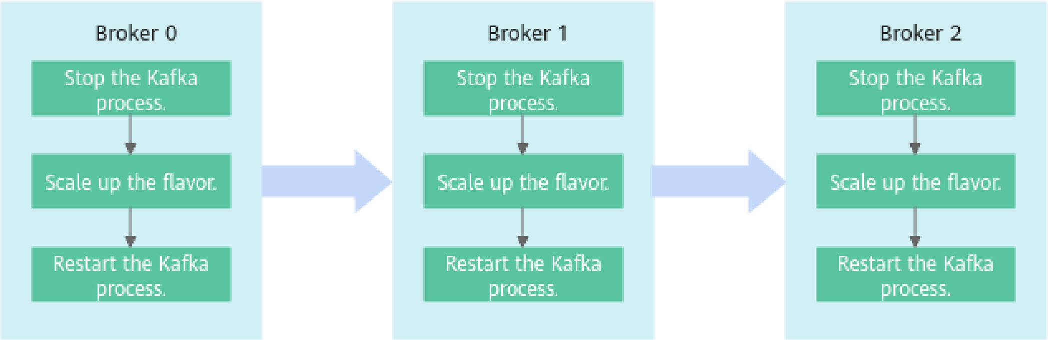
Broker flavor	● Single-replica topics do not support message production during this period. Services will be interrupted.
	● If a topic has multiple replicas, modifying the broker flavor does not interrupt services, but may cause disorder of partition messages. Evaluate this impact and avoid peak hours.
	● Broker rolling restarts will cause partition leader changes, interrupting connections for less than a minute when the network is stable. For multi-replica topics, configure the retry mechanism on the producer client. To do so:
	– If you use an open-source Kafka client, configure the **retries** parameter to a value in the range from 3 to 5.
	– If you use Flink, configure the retry policy by referring to the following code:
	`StreamExecutionEnvironment env = StreamExecutionEnvironment.getExecutionEnvironment();`
	`env.setRestartStrategy(RestartStrategies.fixedDelayRestart(3, Time.seconds(20)));`

Process of Increasing Broker Flavors

When you scale up the broker flavor, a rolling restart is performed on brokers. The following process takes three brokers as an example:

1. Stop the Kafka process on Broker 0.
2. Scale up the flavor of Broker 0.
3. Restart the Kafka process on Broker 0.
4. Repeat 1 to 3 to scale up the flavor of Broker 1.
5. Repeat 1 to 3 to scale up the flavor of Broker 2.

Figure 10-1 Process of increasing a broker flavor



Procedure

Step 1 Log in to the console.

Step 2 Click 📍 in the upper left corner to select a region.

> **NOTE**
>
> Select the region where your Kafka instance is located.

Step 3 Click ☰ and choose **Application** > **Distributed Message Service (for Kafka)** to open the console of DMS for Kafka.

Step 4 In the row containing the desired instance, choose **More** > **Modify Specifications** in the Operation column.

Step 5 Specify the storage space, number of brokers, or broker flavor as required.

- **Expand the storage space.**
  a. For **Change By**, select **Storage**. For **Storage space per broker**, specify a new storage space, and click **Next**. The storage space range varies by instance specifications. For details, see **Specifications**.
  b. Confirm the configurations and click **Submit**.
  c. Return to the instance list and check whether the change succeeded.
     - If the instance status has changed from **Changing** to **Running**, the change succeeded. View the new storage space (Storage space per broker x Number of brokers) in the **Used/Available Storage Space (GB)** column in the instance list.
     - If the instance status has changed from **Changing** to **Change failed**, the change failed. Move the cursor over **Change failed** to check the failure cause.

       Instances in the **Change failed** state cannot be restarted, modified, or deleted. After the instance status automatically changes from **Change failed** to **Running**, you can continue to perform operations on the instance. If the status does not change to **Running**, contact customer service.
- **Add brokers.**
  a. For **Change By**, select **Brokers**.
  b. For **Brokers**, specify the broker quantity. The broker quantity range varies by instance specifications. For more information, see **Kafka Instance Specifications**.
  c. If public access has been enabled, configure EIPs for the new brokers.
  d. Click **Next**.
  e. Confirm the configurations and click **Submit**.
  f. Return to the instance list and check whether the change succeeded.
     - If the instance status has changed from **Changing** to **Running**, the change succeeded. You can check the new broker quantity in the **Flavor** column.
     - If the instance status has changed from **Changing** to **Change failed**, the change failed. Move the cursor over **Change failed** to check the failure cause.

       Instances in the **Change failed** state cannot be restarted, modified, or deleted. After the instance status automatically changes from

Change failed to **Running**, you can continue to perform operations on the instance. If the status does not change to **Running**, contact customer service.

NOTE

After adding brokers, add the IP addresses of the new brokers to the client connection configuration to improve reliability.

- **Increase the broker flavor.**
  1. For **Change By**, select **Broker Flavor**.
  2. Specify a new broker flavor.
  3. In the **Risk Check** area, check for risks.

     If any risk is found, handle it as prompted and click **Recheck**. If the risk does not need to be handled, select **I understand the risks.**
  4. Click **Next**, confirm the information, and click **Submit**.
  5. Return to the instance list and check whether the change succeeded.
     - If the instance status has changed from **Changing** to **Running**, the change succeeded. You can check the new broker flavor in the **Flavor** column.
     - If the instance status has changed from **Changing** to **Change failed**, the change failed. Move the cursor over **Change failed** to check the failure cause.

       Instances in the **Change failed** state cannot be restarted, modified, or deleted. After the instance status automatically changes from **Change failed** to **Running**, you can continue to perform operations on the instance. If the status does not change to **Running**, contact customer service.

----End

11 Migrating Data

11.1 Kafka Data Migration Overview

You can migrate Kafka services to connect message producers and consumers to a new Kafka instance and can even migrate persisted message data to the new Kafka instance. Kafka services can be migrated in the following two scenarios:

- Migrating services to the cloud without downtime

  Services that have high requirements on continuity must be smoothly migrated to the cloud because they cannot afford a long downtime.
- Re-deploying services on the cloud

  A Kafka instance deployed within an AZ is not capable of cross-AZ disaster recovery. For higher reliability, you can re-deploy services to an instance that is deployed across AZs.

Preparation

1. Configure the network environment.

   A Kafka instance can be accessed within a VPC or over a public network. For public network access, the producer and consumer must have public access permissions, and the following security group rules must be configured.

Table 11-1 Security group rules

Direction	Protocol	Port	Source	Description
Inbound	TCP	9094	IP address or IP address group of the Kafka client	Accessing a Kafka instance in a public network (without SSL)

Direction	Protocol	Port	Source	Description
Inbound	TCP	9095	IP address or IP address group of the Kafka client	Accessing a Kafka instance in a public network (with SSL)

2. Create the target Kafka instance.

   The specifications of the target instance cannot be lower than the original specifications. For more information, see **Buying a Kafka Instance**.

3. Create a topic in the target Kafka instance.

   Create a topic with the same configurations as the original Kafka instance, including the topic name, number of replicas, number of partitions, message aging time, and whether to enable synchronous replication and flushing. For more information, see **Creating a Kafka Topic**.

Migration Scheme 1: Migrating the Production First

Migrate the message production service to the new Kafka instance. After migration, the original Kafka instance will no longer produce messages. After all messages of the original Kafka instance are consumed, migrate the message consumption service to the new Kafka instance to consume messages of this instance.

This is a common migration scheme. It is simple and easy to control on the service side. During the migration, the message sequence is ensured, so this scheme is **suitable for scenarios with strict requirements on the message sequence**. However, latency may occur because there is a period when you have to wait for all data to be consumed.

Step 1 Change the Kafka connection address of the producer to that of the new Kafka instance.

Step 2 Restart the production service so that the producer can send new messages to the new Kafka instance.

Step 3 Check the consumption progress of each consumer group in the original Kafka instance until all data in the original Kafka instance is consumed.

Step 4 Change the Kafka connection addresses of the consumers to those of the new Kafka instance.

Step 5 Restart the consumption service so that consumers can consume messages from the new Kafka instance.

Step 6 Check whether consumers consume messages properly from the new Kafka instance.

Step 7 The migration is complete.

----End

Migration Scheme 2: Migrating the Production Later

Use multiple consumers for the consumption service. Some consume messages from the original Kafka instance, and others consume messages from the new Kafka instances. Then, migrate the production service to the new Kafka instance so that all messages can be consumed in time.

For a certain period of time, the consumption service consumes messages from both the original and new Kafka instances. Before the migration, message consumption from the new Kafka instance has already started, so there is no latency. However, early on in the migration, data is consumed from both the original and new Kafka instances, so the messages may not be consumed in the order that they are produced. This scheme is **suitable for services that require low latency but do not require strict message sequence**.

Step 1 Start new consumer clients, set the Kafka connection addresses to that of the new Kafka instance, and consume data from the new Kafka instance.

NOTE

Original consumer clients must continue running. Messages are consumed from both the original and new Kafka instances.

Step 2 Change the Kafka connection address of the producer to that of the new Kafka instance.

Step 3 Restart the producer client to migrate the production service to the new Kafka instance.

Step 4 After the production service is migrated, check whether the consumption service connected to the new Kafka instance is normal.

Step 5 After all data in the original Kafka is consumed, close the original consumption clients.

Step 6 The migration is complete.

----End

How Do I Migrate Persisted Data Along with Services?

You can migrate consumed data from the original instance to a new instance by using the open-source tool **MirrorMaker**. This tool mirrors the original Kafka producer and consumer into new ones and migrates data to the new Kafka instance.

Note that each cloud Kafka instance stores data in three replicas. Therefore, the storage space of the new instance should be three times that of the original single-replica message storage.

12 Testing Instance Performance

12.1 Kafka Production Rate and CPU Usage

This section describes performance tests on Distributed Message Service (DMS) for Kafka. The performance is measured by the message production rate on the client side and CPU usage on the server side. The tests cover the following scenarios:

- Scenario 1 (batch size): same Kafka instance, same topics, different message size settings
- Scenario 2 (cross-AZ or intra-AZ production): same Kafka instance, same topics, different AZ settings for the client and server
- Scenario 3 (number of replicas): same Kafka instance, different numbers of replicas
- Scenario 4 (synchronous or asynchronous replication): same Kafka instance, topics with different replication settings

Table 12-1 Test parameters

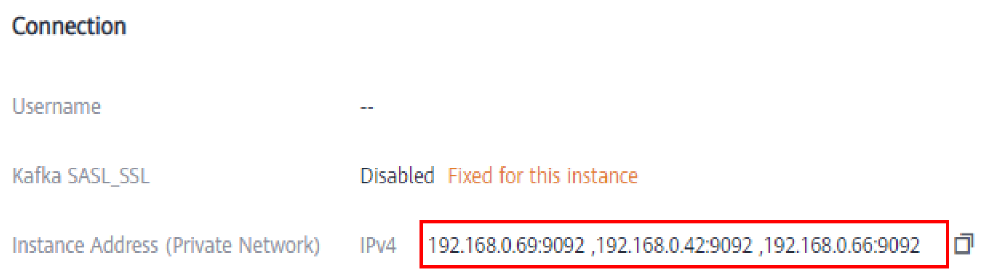
Partitions	Replicas	Synchronous Replication	batch.size	Cross-AZ Production
3	1	No	1 KB	No
3	1	No	16 KB	No
3	1	No	1 KB	Yes
3	3	Yes	1 KB	No
3	3	No	1 KB	No

Environment

Perform the following steps to set up the test environment.

1. Purchase a Kafka instance with parameters specified as follows. For details about how to purchase one, see **Buying a Kafka Instance**.
   - **Region**:
   - **Project**:
   - **AZ**: Select **1**.
   - **Instance Name**: Enter "kafka-test".
   - **Enterprise Project**: Select **default**.
   - **Version**: Select **2.7**.
   - **Broker Flavor**: Select **kafka.2u4g.cluster**.
   - **Brokers**: **3**
   - **Storage space per broker**: ultra-high I/O, 200 GB
   - **Capacity Threshold Policy**: Select **Automatically delete**.
   - **VPC**: Select a VPC.
   - **Security Group**: Select a security group.
   - **Advanced Settings**: Do not enable **Public Network Access**, **SASL_SSL**, and **Automatic Topic Creation**.

   After the purchase, obtain on the instance details page.



Connection

Username --

Kafka SASL_SSL Disabled Fixed for this instance

Instance Address (Private Network) IPv4 192.168.0.69:9092 ,192.168.0.42:9092 ,192.168.0.66:9092

2. Create three topics with parameters specified as follows for the purchased Kafka instance. For details, see **Creating a Kafka Topic**.
   - Topic-01: 3 partitions, 1 replica, asynchronous replication
   - Topic-02: 3 partitions, 3 replicas, asynchronous replication
   - Topic-03: 3 partitions, 3 replicas, synchronous replication
3. Obtain the test tool.

   Obtain **Kafka CLI v2.7.2**.
4. Purchase a server for the client.

   Buy two ECSs with the following configurations. For details about how to purchase an ECS, see **Purchasing an ECS**.
   - One ECS is 4 vCPUs | 8 GB, runs Linux, and is configured with the same region, AZ, VPC, subnet, and security group as the Kafka instance.
   - The other ECS is 4 vCPUs | 8 GB, runs Linux, and is configured with the same region, VPC, subnet, and security group but a different AZ from the Kafka instance.

   Perform the following operations on the ECSs:
   - Install **Java JDK** and configure the environment variables **JAVA_HOME** and **PATH**.

```
export JAVA_HOME=/root/jdk1.8.0_231
export PATH=$JAVA_HOME/bin:$PATH
```

– Download **Kafka CLI v2.7.2** and decompress it.

```
tar -zxf kafka_2.12-2.7.2.tgz
```

Script

```
./kafka-producer-perf-test.sh --producer-props bootstrap.servers=${connection address} acks=1 batch.size=${batch.size} linger.ms=0 --topic ${topic name} --num-records ${num-records} --record-size 1024 --throughput 102400
```

- **bootstrap.servers**: address of the Kafka instance obtained in **1**.
- **acks**: message synchronization policy. acks=1 indicates asynchronous replication, and acks=-1 indicates synchronous replication.
- **batch.size**: size of messages sent in each batch, in bytes.
- **linger.ms**: interval between two batches.
- **topic**: topic name set in **2**.
- **num-records**: total number of messages to be sent.
- **record-size**: size of each message.
- **throughput**: number of messages sent per second.

Procedure

Scenario 1: Varied Batch Sizes

Step 1 Log in to the client server, go to the **kafka_2.12-2.7.2/bin** directory, and run the following scripts.

Set **batch.size** to 1 KB, and run the following script:

```
./kafka-producer-perf-test.sh --producer-props bootstrap.servers=192.168.0.69:9092,192.168.0.42:9092,192.168.0.66:9092 acks=1 batch.size=1024 linger.ms=0 --topic Topic-01 --num-records 8000000 --record-size 1024 --throughput 102400
```

Result:

```
8000000 records sent, 27417.353814 records/sec (26.77 MB/sec), 1095.79 ms avg latency, 5989.00 ms max latency, 679 ms 50th, 2957 ms 95th, 3505 ms 99th, 5951 ms 99.9th.
```

Message production rate: 27,417 records/second

Set **batch.size** to 16 KB, and run the following script:

```
./kafka-producer-perf-test.sh --producer-props bootstrap.servers=192.168.0.69:9092,192.168.0.42:9092,192.168.0.66:9092 acks=1 batch.size=16384 linger.ms=0 --topic Topic-01 --num-records 100000000 --record-size 1024 --throughput 102400
```

Result:

```
100000000 records sent, 102399.213574 records/sec (100.00 MB/sec), 11.27 ms avg latency, 876 ms max latency, 1 ms 50th, 15 ms 95th, 384 ms 99th, 724 ms 99.9th.
```

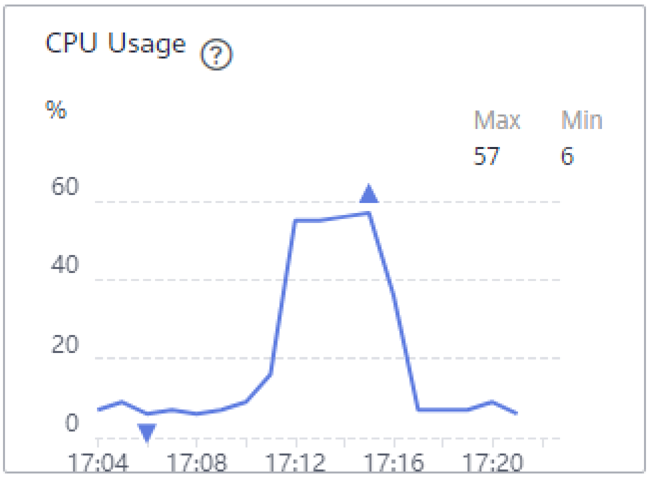
Message production rate: 102,399 records/second

Step 2 Log in to the Kafka console and click the name of the test instance.

Step 3 In the navigation pane, choose **Monitoring**.

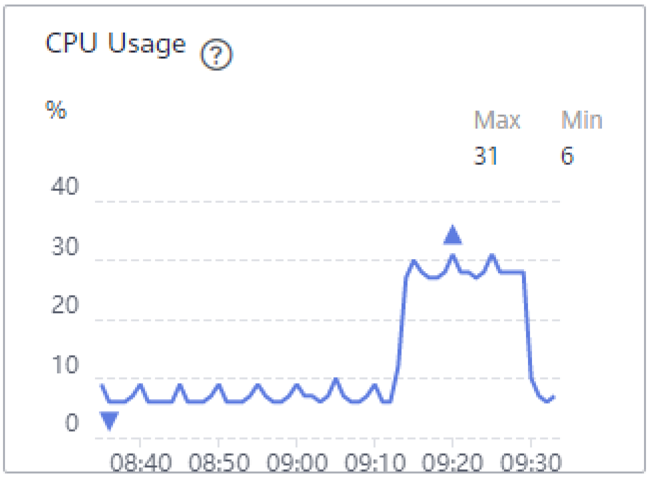
Step 4 On the **Brokers** tab page, view the CPU usage of the server nodes.

Figure 12-1 broker-0 CPU usage (batch.size = 1 KB)



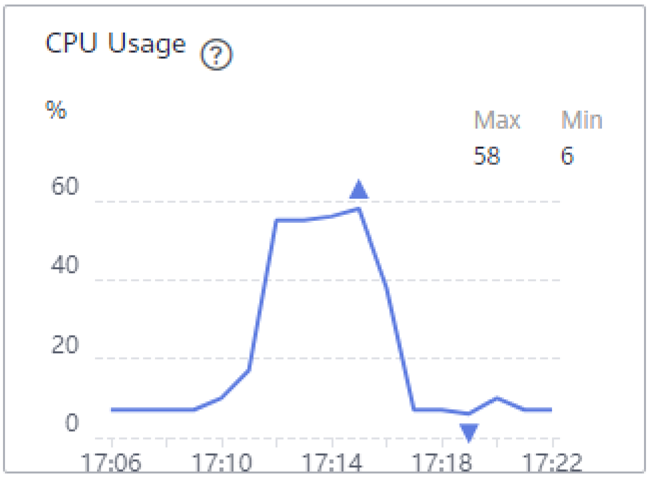
CPU usage: 57%

Figure 12-2 broker-0 CPU usage (batch.size = 16 KB)



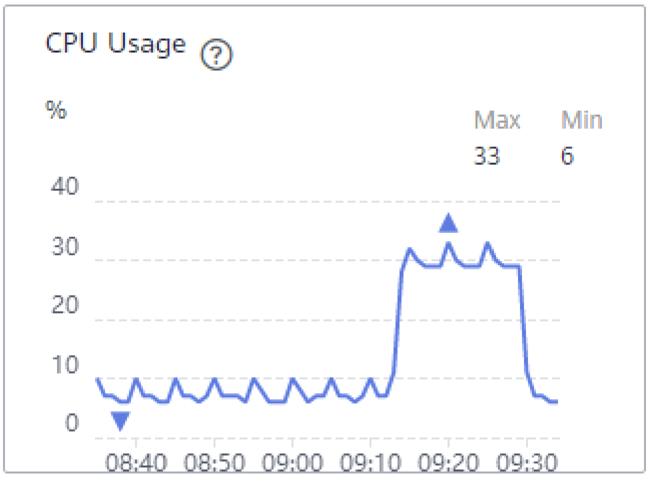
CPU usage: 31%

Figure 12-3 broker-1 CPU usage (batch.size = 1 KB)



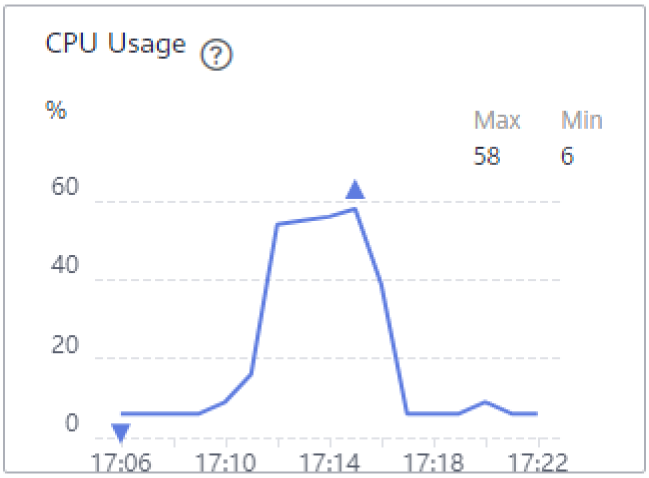
CPU usage: 58%

Figure 12-4 broker-1 CPU usage (batch.size = 16 KB)



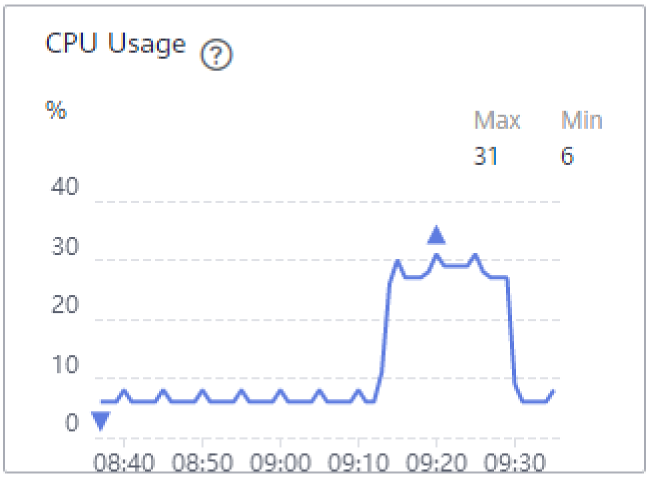
CPU usage: 33%

Figure 12-5 broker-2 CPU usage (batch.size = 1 KB)



CPU usage: 58%

Figure 12-6 broker-2 CPU usage (batch.size = 16 KB)



CPU usage: 31%

----End

Scenario 2: Cross-AZ or Intra-AZ Production

Step 1 Log in to the client server, go to the **kafka_2.12-2.7.2/bin** directory, and run the following scripts.

Configure the same AZ for the client and the instance, and run the following script:

```
./kafka-producer-perf-test.sh --producer-props
bootstrap.servers=192.168.0.69:9092,192.168.0.42:9092,192.168.0.66:9092 acks=1 batch.size=1024
linger.ms=0 --topic Topic-01 --num-records 8000000 --record-size 1024 --throughput 102400
```

Result:

```
8000000 records sent, 27417.353814 records/sec (26.77 MB/sec), 1095.79 ms avg latency, 5989.00 ms max
latency, 679 ms 50th, 2957 ms 95th, 3505 ms 99th, 5951 ms 99.9th.
```

Message production rate: 27,417 records/second

Configure different AZs for the client and the instance, and run the following script:

```
./kafka-producer-perf-test.sh --producer-props
bootstrap.servers=192.168.0.69:9092,192.168.0.42:9092,192.168.0.66:9092 acks=1 batch.size=1024
linger.ms=0 --topic Topic-01 --num-records 4000000 --record-size 1024 --throughput 102400
```

Result:

```
4000000 records sent, 15351.079181 records/sec (14.99 MB/sec), 1927.26 ms avg latency, 8974.00 ms max
latency, 15 ms 50th, 5925 ms 95th, 6838 ms 99th, 8925 ms 99.9th.
```

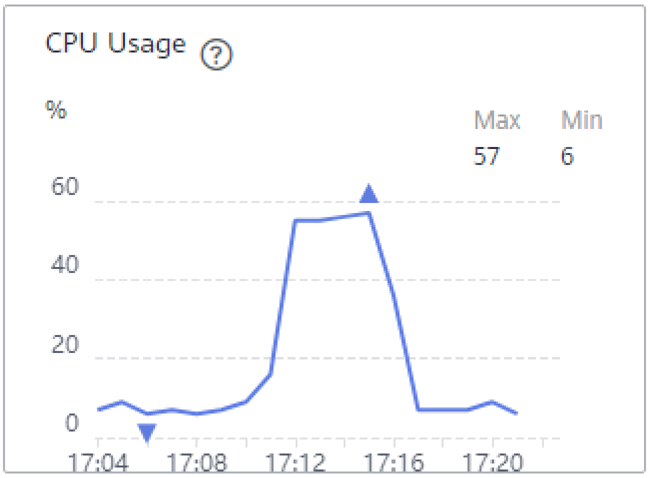
Message production rate: 15,351 records/second

Step 2 Log in to the Kafka console and click the name of the test instance.

Step 3 In the navigation pane, choose **Monitoring**.

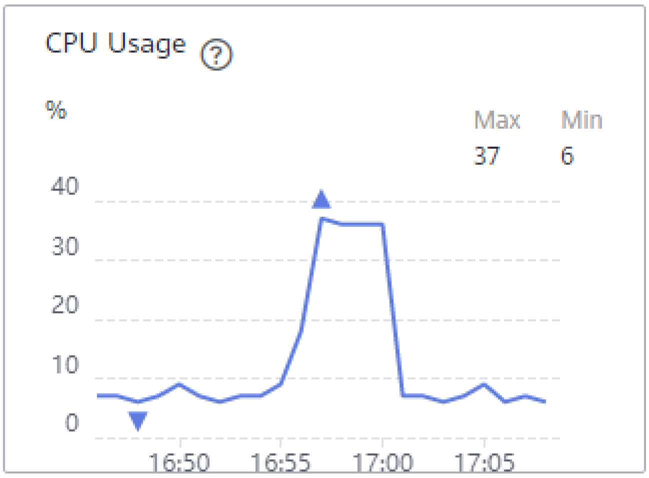
Step 4 On the **Brokers** tab page, view the CPU usage of the server nodes.

Figure 12-7 broker-0 CPU usage (same AZ)



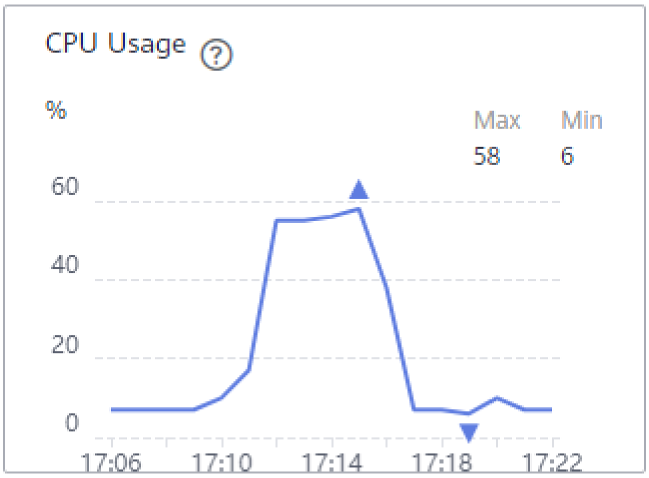
CPU usage: 57%

Figure 12-8 broker-0 CPU usage (different AZs)



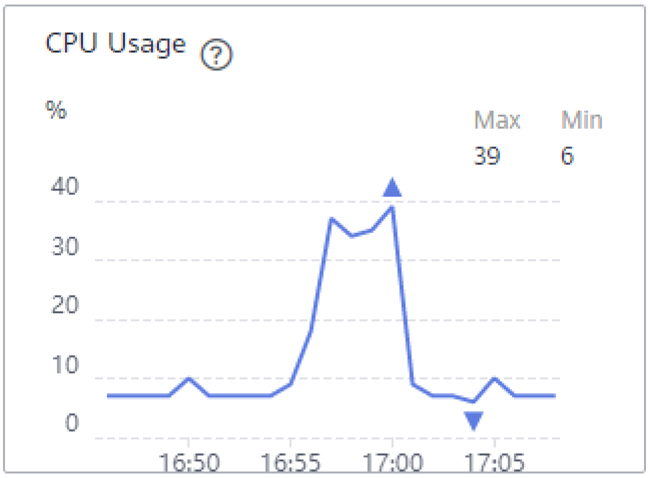
CPU usage: 37%

Figure 12-9 broker-1 CPU usage (same AZ)



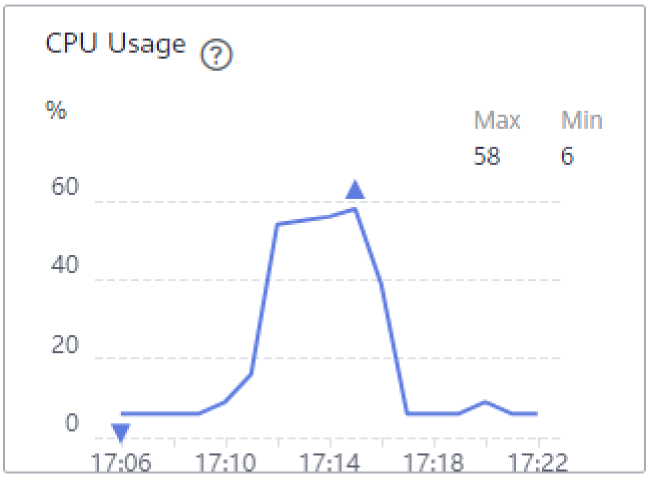
CPU usage: 58%

Figure 12-10 broker-1 CPU usage (different AZs)



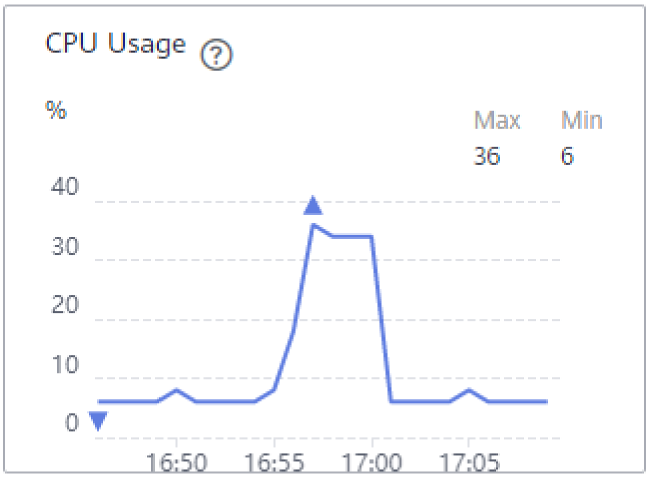
CPU usage: 39%

Figure 12-11 broker-2 CPU usage (same AZ)



CPU usage: 58%

Figure 12-12 broker-2 CPU usage (different AZs)



CPU usage: 36%

----End

Scenario 3: Varied Numbers of Replicas

Step 1 Log in to the client server, go to the **kafka_2.12-2.7.2/bin** directory, and run the following scripts.

For the **one-replica** topic, run the following script:

```
./kafka-producer-perf-test.sh --producer-props
bootstrap.servers=192.168.0.69:9092,192.168.0.42:9092,192.168.0.66:9092 acks=1 batch.size=1024
linger.ms=0 --topic Topic-01 --num-records 8000000 --record-size 1024 --throughput 102400
```

Result:

```
8000000 records sent, 27417.353814 records/sec (26.77 MB/sec), 1095.79 ms avg latency, 5989.00 ms max
latency, 679 ms 50th, 2957 ms 95th, 3505 ms 99th, 5951 ms 99.9th.
```

Message production rate: 27,417 records/second

For the **three-replica** topic, run the following script:

```
./kafka-producer-perf-test.sh --producer-props
bootstrap.servers=192.168.0.69:9092,192.168.0.42:9092,192.168.0.66:9092 acks=1 batch.size=1024
linger.ms=0 --topic Topic-02 --num-records 4000000 --record-size 1024 --throughput 102400
```

Result:

```
4000000 records sent, 11557.954473 records/sec (11.29 MB/sec), 2591.66 ms avg latency, 8071.00 ms max
latency, 2566 ms 50th, 5396 ms 95th, 6276 ms 99th, 8003 ms 99.9th.
```

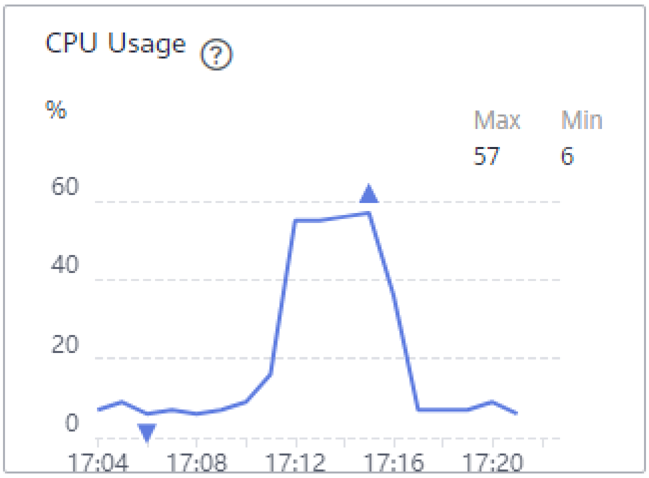
Message production rate: 11,558 records/second

Step 2 Log in to the Kafka console and click the name of the test instance.

Step 3 In the navigation pane, choose **Monitoring**.

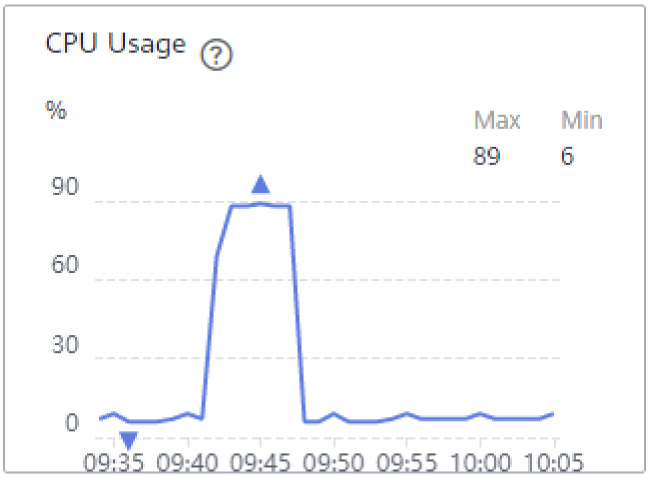
Step 4 On the **Brokers** tab page, view the CPU usage of the server nodes.

Figure 12-13 broker-0 CPU usage (one replica)



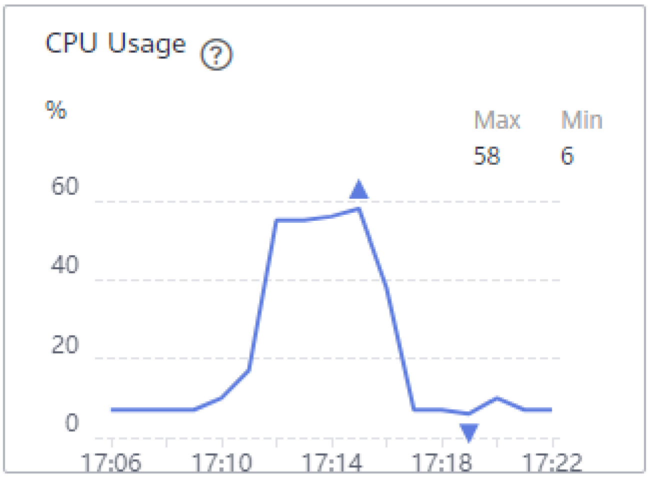
CPU usage: 57%

Figure 12-14 broker-0 CPU usage (three replicas)



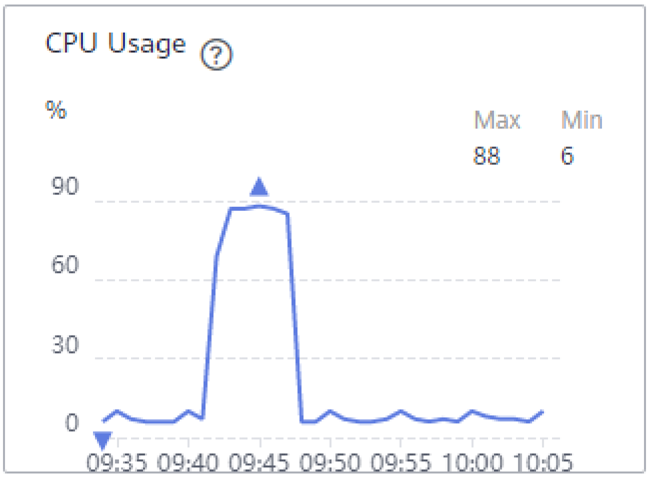
CPU usage: 89%

Figure 12-15 broker-1 CPU usage (one replica)



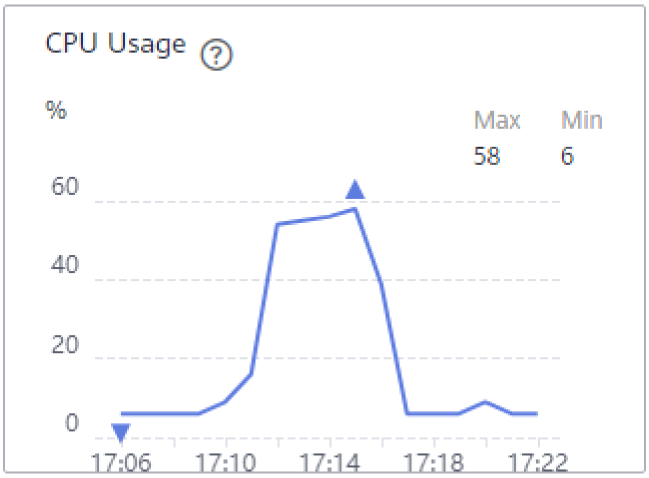
CPU usage: 58%

Figure 12-16 broker-1 CPU usage (three replicas)



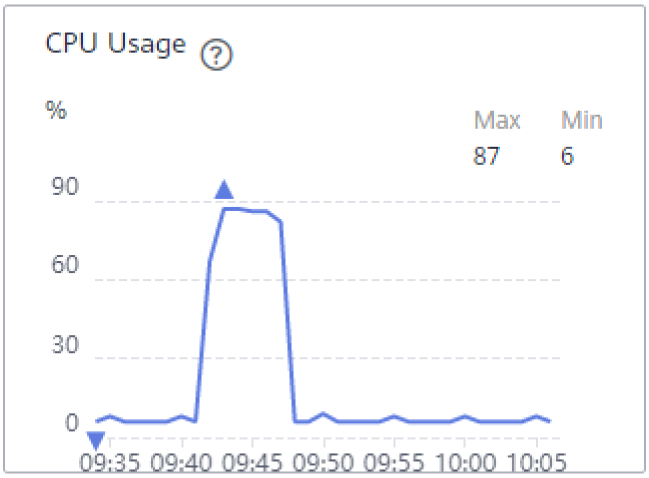
CPU usage: 88%

Figure 12-17 broker-2 CPU usage (one replica)



CPU usage: 58%

Figure 12-18 broker-2 CPU usage (three replicas)



CPU usage: 87%

----End

Scenario 4: Synchronous/Asynchronous Replication

Step 1 Log in to the client server, go to the **kafka_2.12-2.7.2/bin** directory, and run the following scripts.

For **asynchronous replication**, run the following script:

```
./kafka-producer-perf-test.sh --producer-props
bootstrap.servers=192.168.0.69:9092,192.168.0.42:9092,192.168.0.66:9092 acks=1 batch.size=1024
linger.ms=0 --topic Topic-02 --num-records 4000000 --record-size 1024 --throughput 102400
```

Result:

```
4000000 records sent, 11557.954473 records/sec (11.29 MB/sec), 2591.66 ms avg latency, 8071.00 ms max
latency, 2566 ms 50th, 5396 ms 95th, 6276 ms 99th, 8003 ms 99.9th.
```

Message production rate: 11,558 records/second

For **synchronous replication**, run the following script:

```
./kafka-producer-perf-test.sh --producer-props
bootstrap.servers=192.168.0.69:9092,192.168.0.42:9092,192.168.0.66:9092 acks=-1 batch.size=1024
linger.ms=0 --topic Topic-03 --num-records 1000000 --record-size 1024 --throughput 102400
```

Result:

```
1000000 records sent, 3859.960628 records/sec (3.77 MB/sec), 7675.00 ms avg latency, 13852.00 ms max
latency, 7695 ms 50th, 11056 ms 95th, 12907 ms 99th, 13809 ms 99.9th.
```

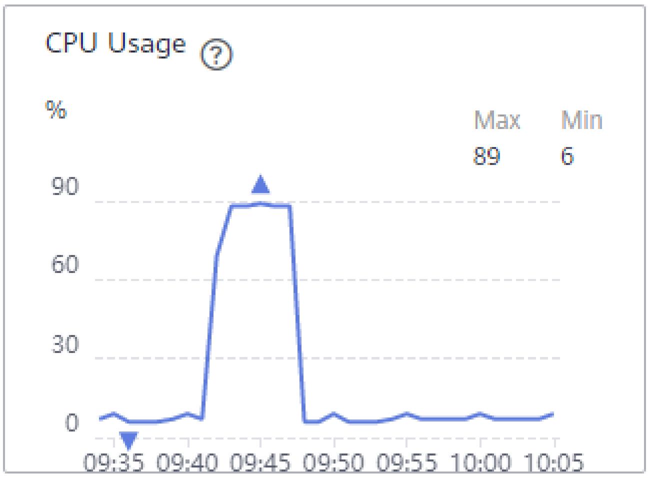
Message production rate: 3859 records/second

Step 2 Log in to the Kafka console and click the name of the test instance.

Step 3 In the navigation pane, choose **Monitoring**.

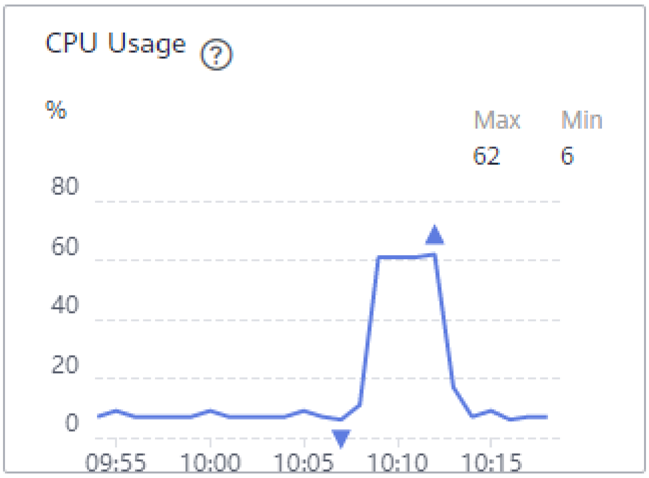
Step 4 On the **Brokers** tab page, view the CPU usage of the server nodes.

Figure 12-19 broker-0 CPU usage (asynchronous replication)



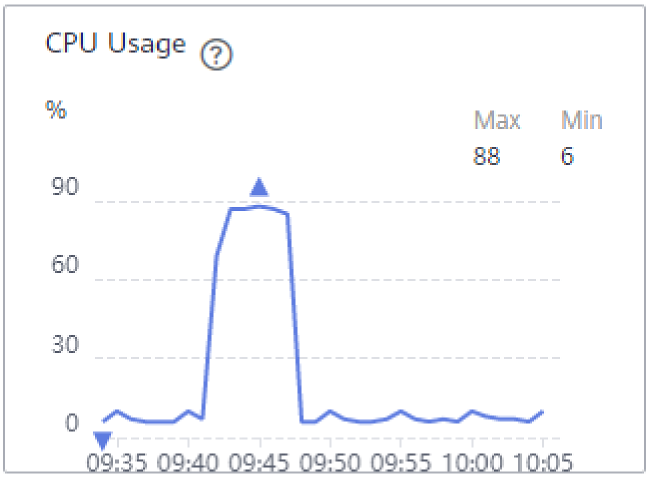
CPU usage: 89%

Figure 12-20 broker-0 CPU usage (synchronous replication)



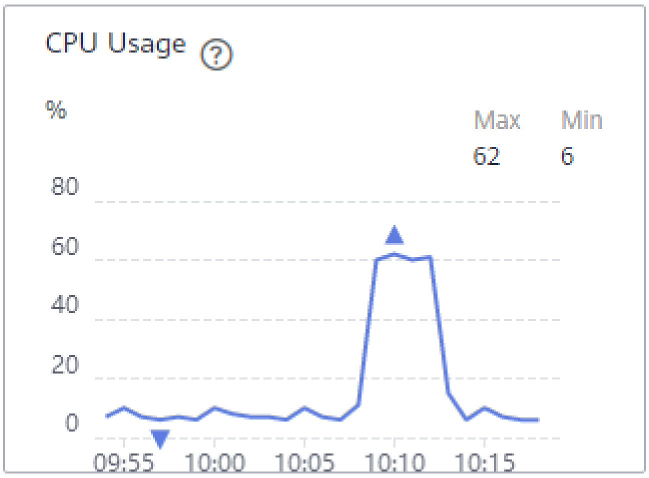
CPU usage: 62%

Figure 12-21 broker-1 CPU usage (asynchronous replication)



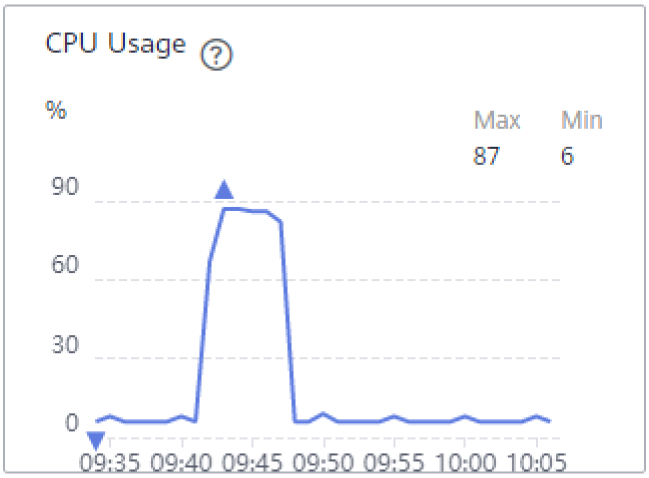
CPU usage: 88%

Figure 12-22 broker-1 CPU usage (synchronous replication)



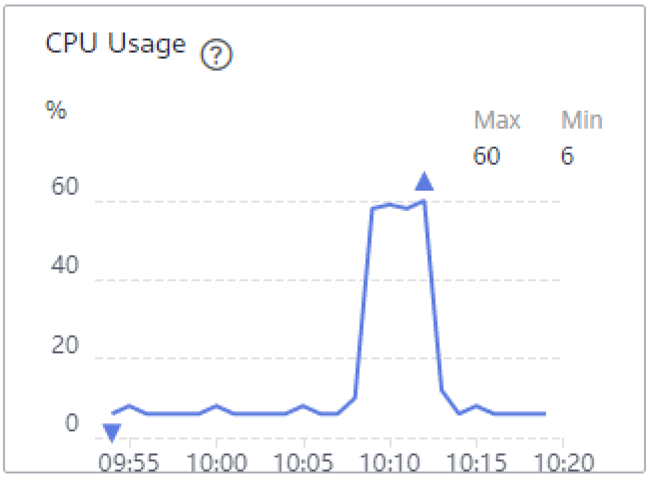
CPU usage: 62%

Figure 12-23 broker-2 CPU usage (asynchronous replication)



CPU usage: 87%

Figure 12-24 broker-2 CPU usage (synchronous replication)



CPU usage: 60%

----End

Result

Table 12-2 Testing results

Partitions	Replicas	Synchronous Replication	batch.size	Cross-AZ Production	Message Production Rate on the Client Side (Records/Second)	CPU Usage on the Server Side (broker-0)	CPU Usage on the Server Side (broker-1)	CPU Usage on the Server Side (broker-2)
3	1	No	1 KB	No	27417	57%	58%	58%
3	1	No	16 KB	No	102399	31%	33%	31%
3	1	No	1 KB	Yes	15351	37%	39%	36%
3	3	Yes	1 KB	No	3859	62%	62%	60%
3	3	No	1 KB	No	11558	89%	88%	87%

Based on the test results, the following conclusions are drawn (for reference only):

- When the **batch.size** of production requests is 16 times larger, the message production rate increases, and the CPU usage decreases.
- Compared with cross-AZ production, intra-AZ production significantly increases message production rate and CPU usage.

- When the number of replicas changes from 1 to 3, the message production rate decreases significantly, and the CPU usage increases.
- Compared with synchronous replication, asynchronous replication increases the message production rate and the CPU usage.

12.2 Kafka Instance TPS

TPS tests can be performed in the following scenarios:

- Scenario 1 (whether SASL is enabled): same topic, different SASL settings
- Scenario 2 (synchronous or asynchronous replication): same instance, topics with different replication settings
- Scenario 3 (synchronous or asynchronous flushing): same instance, topics with different flushing settings
- Scenario 4 (disk type): same topic, instances with different disk types
- Scenario 5 (number of partitions): same instance, topics with different number of partitions

Environment

Perform the following steps to set up the test environment.

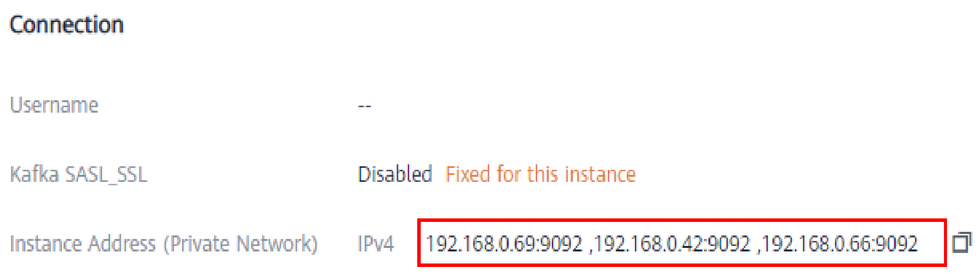
1. Purchase Kafka instances with parameters specified in **Table 12-3**. For more information, see **Buying a Kafka Instance**.

Table 12-3 Instance parameters

Instance Name	Brokers	Broker Flavor	SASL	Storage space per broker
kafka-01	3	kafka.2u4g.cluster	Yes	Ultra-high I/O
kafka-02	3	kafka.4u8g.cluster	Yes	Ultra-high I/O
kafka-03	3	kafka.8u16g.cluster	Yes	Ultra-high I/O
kafka-04	3	kafka.12u24g.cluster	Yes	Ultra-high I/O
kafka-05	3	kafka.16u32g.cluster	Yes	Ultra-high I/O
kafka-06	3	kafka.2u4g.cluster	No	Ultra-high I/O
kafka-07	3	kafka.4u8g.cluster	No	Ultra-high I/O

Instance Name	Brokers	Broker Flavor	SASL	Storage space per broker
kafka-08	3	kafka.8u16g.cluster	No	Ultra-high I/O
kafka-09	3	kafka.12u24g.cluster	No	Ultra-high I/O
kafka-10	3	kafka.16u32g.cluster	No	Ultra-high I/O
kafka-11	3	kafka.2u4g.cluster	No	High I/O
kafka-12	3	kafka.4u8g.cluster	No	High I/O
kafka-13	3	kafka.8u16g.cluster	No	High I/O
kafka-14	3	kafka.12u24g.cluster	No	High I/O
kafka-15	3	kafka.16u32g.cluster	No	High I/O

After the purchase, obtain **Instance Address (Private Network)** on the instance details page.



Connection

Username --

Kafka SASL_SSL Disabled Fixed for this instance

Instance Address (Private Network) IPv4 192.168.0.69:9092 ,192.168.0.42:9092 ,192.168.0.66:9092

2. Create topics with parameters specified in **Table 12-4** for each instance purchased above. For details about how to create topics, see **Creating a Kafka Topic**.

Table 12-4 Topic parameters

Topic Name	Synchronous Replication	Synchronous Flushing	Replicas	Partitions
topic-01	No	No	3	30
topic-02	Yes	No	3	30
topic-03	No	Yes	3	30

Topic Name	Synchronous Replication	Synchronous Flushing	Replicas	Partitions
topic-04	No	No	3	3
topic-05	No	No	3	12
topic-06	No	No	3	100

3. Obtain the test tool.

   Obtain **Kafka CLI v2.7.2**.

4. Purchase a server for the client.

   Buy a Linux ECS (with the same region, AZ, VPC, subnet, and security group as the Kafka instance). For details about how to purchase an ECS, see **Purchasing an ECS**.

   Perform the following operations on the ECSs:

   – Install **Java JDK** and configure the environment variables **JAVA_HOME** and **PATH**.

   ```
   export JAVA_HOME=/root/jdk1.8.0_231
   export PATH=$JAVA_HOME/bin:$PATH
   ```

   – Download **Kafka CLI v2.7.2** and decompress it.

   ```
   tar -zxf kafka_2.12-2.7.2.tgz
   ```

Script

```
./kafka-producer-perf-test.sh --producer-props bootstrap.servers=${connection address} acks=1
batch.size=16384 linger.ms=10 --topic ${topic name} --num-records 10000000 --record-size 1024 --
throughput -1 --producer.config ../config/producer.properties
```

- **bootstrap.servers**: address of the Kafka instance obtained in **1**.
- **acks**: message synchronization policy. acks=1 indicates asynchronous replication, and acks=-1 indicates synchronous replication.
- **batch.size**: size of messages sent in each batch, in bytes.
- **linger.ms**: interval between two batches.
- **topic**: topic name set in **2**.
- **num-records**: total number of messages to be sent.
- **record-size**: size of each message.
- **throughput**: number of messages sent per second.

Result

Scenario 1 (whether SASL is enabled): same topic (30 partitions, 3 replicas, asynchronous replication, and asynchronous flushing), instances with SASL enabled or disabled. The test result is as follows.

Table 12-5 Test results

Instance Flavor	Storage space per broker	Brokers	TPS (SASL Enabled)	TPS (SASL Disabled)
kafka.2u4g.cluster	Ultra-high I/O	3	100,000	280,000
kafka.4u8g.cluster	Ultra-high I/O	3	170,000	496,000
kafka.8u16g.cluster	Ultra-high I/O	3	200,000	730,000
kafka.12u24g.cluster	Ultra-high I/O	3	320,000	790,000
kafka.16u32g.cluster	Ultra-high I/O	3	360,000	1,000,000

Conclusion: When messages are produced to Kafka instances with the same flavor and topic but different access modes, instances without SASL show higher TPS than those with SASL.

Scenario 2 (synchronous/asynchronous replication): same instance (ultra-high I/O, three brokers, SASL disabled), topics with different replication settings, and number of producer processes is three. The test result is as follows.

Table 12-6 Test results

Instance Flavor	Synchronous Flushing	Replicas	Partitions	TPS (Synchronous Replication)	TPS (Asynchronous Replication)
kafka.2u4g.cluster	No	3	30	100,000	280,000
kafka.4u8g.cluster	No	3	30	230,000	496,000
kafka.8u16g.cluster	No	3	30	342,000	730,000
kafka.12u24g.cluster	No	3	30	383,000	790,000
kafka.16u32g.cluster	No	3	30	485,000	1,000,000

Conclusion: When messages are produced to different topics of a Kafka instance, topics with asynchronous replication show higher TPS than those with synchronous replication when other topic parameters are the same.

Scenario 3 (synchronous/asynchronous replication flushing): same instance (ultra-high I/O, three brokers, SASL disabled), topics with different flushing settings. The test result is as follows.

Table 12-7 Test results

Instance Flavor	Synchronous Replication	Replicas	Partitions	TPS (Synchronous Flushing)	TPS (Asynchronous Flushing)
kafka.2u4g.cluster	No	3	30	30,000	280,000
kafka.4u8g.cluster	No	3	30	32,500	496,000
kafka.8u16g.cluster	No	3	30	36,100	730,000
kafka.12u24g.cluster	No	3	30	37,400	790,000
kafka.16u32g.cluster	No	3	30	40,400	1,000,000

Conclusion: When messages are produced to different topics of a Kafka instance, topics with asynchronous flushing show significantly higher TPS than those with synchronous flushing when other topic parameters are the same.

Scenario 4 (different disk types): same topic (30 partitions, 3 replicas, asynchronous replication, and asynchronous flushing) with different disk types. The test result is as follows.

Table 12-8 Test results

Instance Flavor	Brokers	SASL	TPS (High I/O)	TPS (Ultra-High I/O)
kafka.2u4g.cluster	3	No	110,000	250,000
kafka.4u8g.cluster	3	No	135,000	380,000
kafka.8u16g.cluster	3	No	213,000	480,000
kafka.12u24g.cluster	3	No	240,000	577,000

Instance Flavor	Brokers	SASL	TPS (High I/O)	TPS (Ultra-High I/O)
kafka.16u32g.cluster	3	No	280,000	840,000

Conclusion: When messages are produced to the same topics of Kafka instances with the same flavor but different disk types, instances with ultra-high I/O disks show higher TPS than those with high I/O disks.

Scenario 5 (different numbers of partitions): same instance (ultra-high I/O, three brokers, SASL disabled), topics with different number of partitions. The test result is as follows.

Table 12-9 Test results

Instance Flavor	Synchronous Flushing	Synchronous Replication	Replicas	TPS (3 Partitions)	TPS (12 Partitions)	TPS (100 Partitions)
kafka.2u4g.cluster	No	No	3	250,000	260,000	250,000
kafka.4u8g.cluster	No	No	3	330,000	280,000	260,000
kafka.8u16g.cluster	No	No	3	480,000	410,000	340,000
kafka.12u24g.cluster	No	No	3	570,000	750,000	520,000
kafka.16u32g.cluster	No	No	3	840,000	1,000,000	630,000

Conclusion: When messages are produced to topics with different partition quantities of a Kafka instance, instances with more partitions show higher performance when other parameters are the same. However, performance reaches a peak and then deteriorates when partitions continue to increase.

13 Applying for Increasing Kafka Quotas

What Is a Quota?

A quota is a limit on the quantity or capacity of a certain type of service resources that you can use, for example, the maximum number of Kafka instances that you can create.

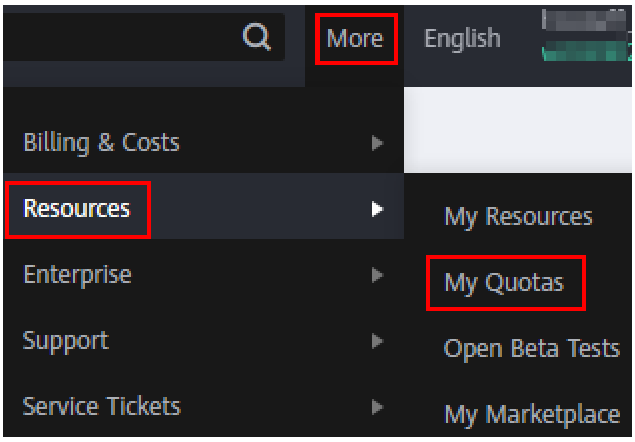
If a quota cannot meet your needs, apply for a higher quota.

How Do I View My Quota?

1. Log in to the console.
2. In the upper right corner of the page, choose **Resources** > **My Quotas**.

   The **Quotas** page is displayed.

   Figure 13-1 My Quotas

3. On the **Quotas** page, view the used and total quotas of resources.

   If a quota cannot meet your needs, apply for a higher quota by performing the following operations.

How Do I Increase My Quota?

1. Log in to the console.
2. In the upper right corner of the page, choose **Resources** > **My Quotas**.

   The **Service Quota** page is displayed.
3. Click **Increase Quota**.
4. On the **Create Service Ticket** page, set the parameters.

   In the **Problem Description** area, enter the required quota and the reason for the quota adjustment.
5. Read the agreements and confirm that you agree to them, and then click **Submit**.

14 Monitoring and Alarms

14.1 Viewing Kafka Metrics

Cloud Eye monitors Kafka instance metrics in real time. You can view these metrics on the Cloud Eye console.

Prerequisites

At least one Kafka instance has been created. The instance has at least one available message.

Procedure

Step 1 Log in to the console.

Step 2 Click 📍 in the upper left corner to select a region.

NOTE

Select the region where your Kafka instance is located.

Step 3 Click ☰ and choose **Application** > **Distributed Message Service (for Kafka)** to open the console of DMS for Kafka.

Step 4 View the instance metrics in either of the following ways:

- In the row containing the desired instance, click **View Metric**. The Cloud Eye console is displayed. Click **By Instance**, **By Broker**, **By Topic**, or **By Consumer Group** to view monitoring data. The data is updated every minute.
- Click the desired Kafka instance to view its details. Choose **Monitoring**. The monitoring data can be viewed on the **By Instance**, **By Broker**, **By Topic**, and **By Consumer Group** tab pages. The data is updated every minute.

----End

14.2 Kafka Metrics

Introduction

This section describes metrics reported by DMS for Kafka to Cloud Eye as well as their namespaces and dimensions. You can use the Cloud Eye console or **APIs** to query the Kafka metrics and alarms, or view Kafka instance metrics on the **Monitoring** page of the DMS for Kafka console.

For example, you can call the **API** to query the monitoring data of the **Disk Capacity Usage** metric.

Namespace

SYS.DMS

Instance Metrics

Table 14-1 Instance metrics

Metric ID	Metric Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
current_partitions	Partitions	Number of used partitions in the instance Unit: Count	0–100,000	Kafka instance	1 minute
current_topics	Topics	Number of created topics in the instance Unit: Count	0–100,000	Kafka instance	1 minute
group_msgs	Accumulated Messages	Total number of accumulated messages in all consumer groups of the instance Unit: Count	0–1,000,000,000	Kafka instance	1 minute

Broker Metrics

Table 14-2 Broker metrics

Metric ID	Metric Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
broker_data_size	Message Size	Total size of messages in the broker Unit: byte, KB, MB, GB, TB or PB	0–5,000,000,000,000	Kafka instance broker	1 minute
broker_messages_in_rate	Message Creation Rate	Number of messages created per second Unit: count/s	0–500,000	Kafka instance broker	1 minute
broker_bytes_out_rate	Message Retrieval	Number of bytes retrieved per second Unit: byte/s, KB/s, MB/s, or GB/s	0–500,000,000	Kafka instance broker	1 minute
broker_bytes_in_rate	Message Creation	Number of bytes created per second Unit: byte/s, KB/s, MB/s, or GB/s	0–500,000,000	Kafka instance broker	1 minute
broker_public_bytes_in_rate	Public Inbound Traffic	Inbound traffic over public networks per second Unit: byte/s, KB/s, MB/s, or GB/s **NOTE** You can view this metric if public access has been enabled and EIPs have been assigned to the instance.	0–500,000,000	Kafka instance broker	1 minute
broker_public_bytes_out_rate	Public Outbound Traffic	Outbound traffic over public networks per second Unit: byte/s, KB/s, MB/s, or GB/s **NOTE** You can view this metric if public access has been enabled and EIPs have been assigned to the instance.	0–500,000,000	Kafka instance broker	1 minute

Metric ID	Metric Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
broker_fetch_mean	Average Message Retrieval Processing Duration	Average time that the broker spends processing message retrieval requests Unit: ms	0–10,000	Kafka instance broker	1 minute
broker_produce_mean	Average Message Creation Processing Duration	Average time that the broker spends processing message creation requests Unit: ms	0–10,000	Kafka instance broker	1 minute
broker_cpu_core_load	Average Load per CPU Core	Average load of each CPU core of the Kafka VM Unit: %	0–20	Kafka instance broker	1 minute
broker_disk_usage	Disk Capacity Usage	Disk usage of the Kafka VM Unit: %	0–100	Kafka instance broker	1 minute
broker_memory_usage	Memory Usage	Memory usage of the Kafka VM Unit: %	0–100	Kafka instance broker	1 minute

Metric ID	Metric Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
broker_heap_usage	JVM Heap Memory Usage of Kafka	Heap memory usage of the Kafka JVM Unit: %	0–100	Kafka instance broker	1 minute
broker_alive	Broker Alive	Whether the Kafka broker is alive	- **1**: alive - **0**: not alive	Kafka instance broker	1 minute
broker_connections	Connections	Total number of TCP connections on the Kafka broker Unit: count	0–65,535	Kafka instance broker	1 minute
broker_cpu_usage	CPU Usage	CPU usage of the Kafka VM Unit: %	0–100	Kafka instance broker	1 minute
broker_disk_read_await	Average Disk Read Time	Average time for each disk I/O read in the monitoring period Unit: ms	> 0	Kafka instance broker	1 minute
broker_disk_write_await	Average Disk Write Time	Average time for each disk I/O write in the monitoring period Unit: ms	> 0	Kafka instance broker	1 minute
broker_total_bytes_in_rate	Inbound Traffic	Inbound traffic per second Unit: byte/s	0–1,000,000,000	Kafka instance broker	1 minute
broker_total_bytes_out_rate	Outbound Traffic	Outbound traffic per second Unit: byte/s	0–1,000,000,000	Kafka instance broker	1 minute

Metric ID	Metric Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
broker_disk_read_rate	Disk Read Speed	Read traffic on the disk Unit: byte/s, KB/s, MB/s, or GB/s	≥ 0	Kafka instance broker	1 minute
broker_disk_write_rate	Disk Write Speed	Write traffic on the disk Unit: byte/s, KB/s, MB/s, or GB/s	≥ 0	Kafka instance broker	1 minute

Topic Metrics

Table 14-3 Topic metrics

Metric ID	Metric Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
topic_bytes_in_rate	Message Creation	Number of bytes created per second Unit: byte/s, KB/s, MB/s, or GB/s **NOTE** This metric is available only when **Monitoring Type** is set to **Basic monitoring** on the **By Topic** tab page.	0–500,000,000	Topic in a Kafka instance	1 minute

Metric ID	Metric Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
topic_bytes_out_rate	Message Retrieval	Number of bytes retrieved per second Unit: byte/s, KB/s, MB/s, or GB/s **NOTE** This metric is available only when **Monitoring Type** is set to **Basic monitoring** on the **By Topic** tab page.	0–500,000,000	Topic in a Kafka instance	1 minute
topic_data_size	Message Size	Total size of messages in the queue Unit: byte, KB, MB, GB, TB or PB **NOTE** This metric is available only when **Monitoring Type** is set to **Basic monitoring** on the **By Topic** tab page.	0–5,000,000,000,000	Topic in a Kafka instance	1 minute
topic_messages	Total Messages	Total number of messages in the queue Unit: count **NOTE** This metric is available only when **Monitoring Type** is set to **Basic monitoring** on the **By Topic** tab page.	≥ 0	Topic in a Kafka instance	1 minute
topic_messages_in_rate	Message Creation Rate	Number of messages created per second Unit: count/s **NOTE** This metric is available only when **Monitoring Type** is set to **Basic monitoring** on the **By Topic** tab page.	0–500,000	Topic in a Kafka instance	1 minute

Metric ID	Metric Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
partition_messages	Partition Messages	Total number of messages in the partition Unit: count **NOTE** This metric is available only when **Monitoring Type** is set to **Partition monitoring** on the **By Topic** tab page.	≥ 0	Topic in a Kafka instance	1 minute
produced_messages	Created Messages	Number of messages that have been created Unit: count **NOTE** This metric is available only when **Monitoring Type** is set to **Partition monitoring** on the **By Topic** tab page.	≥ 0	Topic in a Kafka instance	1 minute

Consumer Group Metrics

Table 14-4 Consumer group metrics

Metric ID	Metric Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
messages_consumed	Retrieved Messages	Number of messages that have been retrieved in the consumer group Unit: count **NOTE** This metric is available only when **Topic** is set to a specific topic name and **Monitoring Type** is set to **Partition monitoring** on the **By Consumer Group** tab page.	≥ 0	Consumer group of a Kafka instance	1 minute
messages_remained	Available Messages	Number of messages that can be retrieved in the consumer group Unit: count **NOTE** This metric is available only when **Topic** is set to a specific topic name and **Monitoring Type** is set to **Partition monitoring** on the **By Consumer Group** tab page.	≥ 0	Consumer group of a Kafka instance	1 minute
topic_messages_remained	Topic Available Messages	Number of remaining messages that can be retrieved from the specified topic in the consumer group Unit: Count **NOTE** This metric is available only when **Topic** is set to a specific topic name and **Monitoring Type** is set to **Basic monitoring** on the **By Consumer Group** tab page.	0 to 2^{63}–1	Consumer group of a Kafka instance	1 minute

Metric ID	Metric Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
topic_messages_consumed	Topic Retrieved Messages	Number of messages that have been retrieved from the specified topic in the consumer group Unit: Count **NOTE** This metric is available only when **Topic** is set to a specific topic name and **Monitoring Type** is set to **Basic monitoring** on the **By Consumer Group** tab page.	0 to 2^{63}–1	Consumer group of a Kafka instance	1 minute
consumer_messages_remained	Accumulated Messages (Consumer Available Messages)	Number of remaining messages that can be retrieved in the consumer group Unit: Count **NOTE** This metric is available only when **Topic** is set to **All topics** on the **By Consumer Group** tab page.	0 to 2^{63}–1	Consumer group of a Kafka instance	1 minute
consumer_messages_consumed	Consumer Retrieved Messages	Number of messages that have been retrieved in the consumer group Unit: Count **NOTE** This metric is available only when **Topic** is set to **All topics** on the **By Consumer Group** tab page.	0 to 2^{63}–1	Consumer group of a Kafka instance	1 minute

Metric ID	Metric Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
messages_consumed_per_min	Partition Consumption Rate	Number of messages consumed from the specified queue partition in the consumer group every minute Unit: count/minute **NOTE** - This metric is available only when **Topic** is set to a specific topic name and **Monitoring Type** is set to **Partition monitoring** on the **By Consumer Group** tab page. - Some instances do not support this metric. Check whether your instance supports it on the console.	0–30,000,000	Consumer group of a Kafka instance	1 minute
topic_messages_consumed_per_min	Queue Consumption Rate	Number of messages consumed from the specified queue in the consumer group every minute Unit: count/minute **NOTE** - This metric is available only when **Topic** is set to a specific topic name and **Monitoring Type** is set to **Basic monitoring** on the **By Consumer Group** tab page. - Some instances do not support this metric. Check whether your instance supports it on the console.	0–30,000,000	Consumer group of a Kafka instance	1 minute

Metric ID	Metric Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
consumer_messages_consumed_per_min	Consumer Group Consumption Rate	Number of messages consumed from the consumer group every minute Unit: count/minute **NOTE** ● This metric is available only when **Topic** is set to **All topics** on the **By Consumer Group** tab page. ● Some instances do not support this metric. Check whether your instance supports it on the console.	0–30,000,000	Consumer group of a Kafka instance	1 minute

Dimension

Key	Value
kafka_instance_id	Kafka instance
kafka_broker	Kafka instance broker
kafka_topics	Kafka instance topic
kafka_partitions	Partition in a Kafka instance
kafka_groups-partitions	Partition consumer group in a Kafka instance
kafka_groups_topics	Topic consumer group in a Kafka instance
kafka_groups	Consumer group of a Kafka instance

14.3 Configuring a Kafka Alarm Rule

This section describes the alarm rules of some metrics and how to configure them. In actual services, you are advised to configure alarm rules for metrics based on the following alarm policies:

Table 14-5 Alarm policies and handling of Kafka instances

Metric ID	Metric	Monitored Object	Alarm Policy	Description	Handling Suggestion
broker_disk_usage	Disk Capacity Usage	Broker	Alarm threshold: original value > 80% Number of consecutive periods: 1 Alarm severity: critical	Disk usage of the Kafka VM	Modify the instance **storage space**. For details, see **Modifying Instance Specifications**.
broker_cpu_core_load	Average Load per CPU Core	Broker	Alarm threshold: original value > 2 Number of consecutive periods: 3 Alarm severity: major	Average load of each CPU core of the Kafka VM.	Check whether the metric has been approaching or exceeding the alarm threshold for a long time. If yes, modify the **number of brokers**. For details, see **Modifying Instance Specifications**.
broker_memory_usage	Memory Usage	Broker	Alarm threshold: original value > 90% Number of consecutive periods: 3 Alarm severity: critical	Memory usage of the Kafka VM.	Modify the **number of brokers**. For details, see **Modifying Instance Specifications**.

Metric ID	Metric	Monitored Object	Alarm Policy	Description	Handling Suggestion
current_partitions	Partitions	Instance	Alarm threshold: original value > 90% of the maximum allowed number of partitions. The partition limit varies depending on instance specifications. For details, see **Specifications**. Number of consecutive periods: 1 Alarm severity: major	Number of used partitions in the instance.	If new topics are required, modify the number of brokers, or split the service to multiple instances. For details about how to modify the number of brokers, see **Modifying Instance Specifications**.
broker_cpu_usage	CPU Usage	Broker	Alarm threshold: original value > 90% Number of consecutive periods: 3 Alarm severity: major	CPU usage of the Kafka VM.	Check whether the metric has been approaching or exceeding the alarm threshold for a long time. If yes, modify the **number of brokers**. For details, see **Modifying Instance Specifications**.

Metric ID	Metric	Monitored Object	Alarm Policy	Description	Handling Suggestion
group_msgs	Accumulated Messages	Instance	Alarm threshold: original value > 90% of the upper limit. The upper limit is customized. Number of consecutive periods: 1 Alarm severity: major	Total number of accumulated messages in all consumer groups of the instance	Delete idle consumer groups, if any. You can also accelerate message retrieval, for example, by increasing the number of consumers.
topic_messages_remained	Topic Available Messages	Consumer group	Alarm threshold: original value > 90% of the upper limit. The upper limit is customized. Number of consecutive periods: 1 Alarm severity: major	Number of remaining messages that can be retrieved from the specified topic in the consumer group.	Check whether the consumer code logic is correct, for example, by checking whether the consumer stops consuming messages due to an exception. You can also accelerate message retrieval, for example, by adding topic consumers. Ensure that the number of partitions is greater than or equal to the number of consumers.

Procedure

Step 1 Log in to the console.

Step 2 Click 📍 in the upper left corner to select a region.

NOTE

Select the region where your Kafka instance is located.

Step 3 Click ☰ and choose **Application** > **Distributed Message Service (for Kafka)** to open the console of DMS for Kafka.

Step 4 Go to the monitoring page in either of the following ways:

- Click **View Metric** in the row containing the desired Kafka instance.
- Click the desired Kafka instance to go to the instance details page. Choose **Monitoring** in the navigation pane.

Step 5 Hover the mouse pointer over a metric and click + to create an alarm rule for the metric. The **Create Alarm Rule** page is displayed.

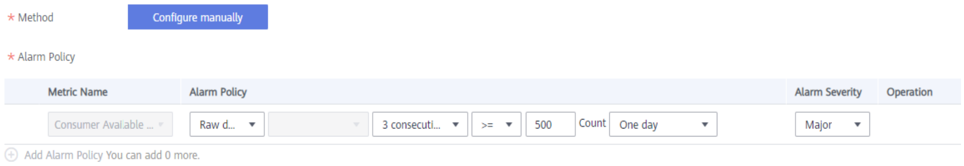
Step 6 Specify the alarm details.

For more information about creating alarm rules, see **Creating an Alarm Rule**.

1. Set the alarm name and description.
2. Set the alarm policy.

   As shown in the following figure, if the original disk capacity usage is equal to or higher than 85% for three consecutive periods, an alarm is generated. If the alarm is not handled on time, an alarm notification is sent.

   Figure 14-1 Setting the alarm policy and alarm severity

   * Method: Configure manually

   * Alarm Policy

Metric Name	Alarm Policy	Alarm Severity	Operation
Consumer Available ...	Raw d... ▾ ▾ 3 consecuti... ▾ >= ▾ 500 Count One day ▾	Major ▾	

3. Set the alarm notification configurations.

   If you enable **Alarm Notification**, specify **Notification Window**, **Notification Object**, and **Trigger Condition**.
4. Click **Create**.

----End

15 Viewing Kafka Audit Logs

With Cloud Trace Service (CTS), you can record operations associated with DMS for Kafka for later query, audit, and backtrack operations.

Prerequisite

CTS has been enabled.

DMS for Kafka Operations Supported by CTS

Table 15-1 DMS for Kafka operations that can be recorded by CTS

Operation	Resource Type	Trace Name
Successfully creating an instance	kafka	createDMSInstanceTaskSuccess
Failing to create an instance	kafka	createDMSInstanceTaskFailure
Successfully deleting an instance that failed to be created	kafka	deleteDMSCreateFailureInstancesSuccess
Failing to delete an instance that failed to be created	kafka	deleteDMSCreateFailureInstancesFailure
Successfully deleting an instance	kafka	deleteDMSInstanceTaskSuccess
Failing to delete an instance	kafka	deleteDMSInstanceTaskFailure
Deleting multiple instance tasks at a time	kafka	batchDeleteDMSInstanceTask

Operation	Resource Type	Trace Name
Successfully submitting a request to delete multiple instances at a time	kafka	batchDeleteDMSInstanceSuccess
Successfully deleting multiple instances at a time	kafka	batchDeleteDMSInstanceTask-Success
Failing to submit a request to delete multiple instances at a time	kafka	batchDeleteDMSInstanceFailure
Failing to delete multiple instances at a time	kafka	batchDeleteDMSInstanceTask-Failure
Successfully submitting a request to scale up an instance	kafka	extendDMSInstanceSuccess
Successfully scaling up an instance	kafka	extendDMSInstanceTaskSuccess
Failing to submit a request to scale up an instance	kafka	extendDMSInstanceFailure
Failing to scale up an instance	kafka	extendDMSInstanceTaskFailure
Successfully submitting a request to reset instance password	kafka	resetDMSInstancePasswordSuccess
Failing to submit a request to reset instance password	kafka	resetDMSInstancePasswordFailure
Successfully submitting a request to restart an instance	kafka	restartDMSInstanceSuccess
Successfully restarting an instance	kafka	restartDMSInstanceTaskSuccess
Failing to submit a request to restart an instance	kafka	restartDMSInstanceFailure

Operation	Resource Type	Trace Name
Failing to restart an instance	kafka	restartDMSInstanceTaskFailure
Successfully submitting a request to restart multiple instances at a time	instance	batchRestartDMSInstanceSuccess
Successfully restarting multiple instances at a time	kafka	batchRestartDMSInstanceTaskSuccess
Failing to submit a request to restart multiple instances at a time	instance	batchRestartDMSInstanceFailure
Failing to restart multiple instances at a time	kafka	batchRestartDMSInstanceTaskFailure
Successfully submitting a request to modify instance information	kafka	modifyDMSInstanceInfoSuccess
Successfully modifying instance information	kafka	modifyDMSInstanceInfoTaskSuccess
Failing to submit a request to modify instance information	kafka	modifyDMSInstanceInfoFailure
Failing to modify instance information	kafka	modifyDMSInstanceInfoTaskFailure
Successfully deleting a background task	kafka	deleteDMSBackendJobSuccess
Failing to delete a background task	kafka	deleteDMSBackendJobFailure
Successfully creating a topic for a Kafka instance	kafka	Kafka_create_topicSuccess
Failing to create a topic for a Kafka instance	kafka	Kafka_create_topicFailure
Successfully deleting a topic from a Kafka instance	kafka	Kafka_delete_topicsSuccess

Operation	Resource Type	Trace Name
Failing to delete a topic for a Kafka instance	kafka	Kafka_delete_topicsFailure
Successfully enabling automatic topic creation	kafka	enable_auto_topicSuccess
Failing to enable automatic topic creation	kafka	enable_auto_topicFailure
Successfully modifying a topic	kafka	Kafka_alter_topicsSuccess
Failing to modify a topic	kafka	Kafka_alter_topicsFailure
Successfully reassigning partitions	kafka	kafka_reassignmentTaskSuccess
Failing to reassign partitions	kafka	kafka_reassignmentTaskFailure
Successfully submitting a partition reassignment request	kafka	kafka_reassignmentSuccess
Failing to submit a partition reassignment request	kafka	kafka_reassignmentFailure
Successfully resetting the consumer offset	kafka	Kafka_reset_consumer_offsetSuccess
Failing to reset the consumer offset	kafka	Kafka_reset_consumer_offsetFailure
Successfully deleting consumer groups in batches	kafka	Kafka_batch_delete_groupSuccess
Failing to delete consumer groups in batches	kafka	Kafka_batch_delete_groupFailure
Successfully creating a user	kafka	createUserSuccess
Failing to create a user	kafka	createUserFailure
Successfully deleting a user	kafka	deleteUserSuccess

Operation	Resource Type	Trace Name
Failing to delete a user	kafka	deleteUserFailure
Successfully updating user policies	kafka	updateUserPoliciesTaskSuccess
Failing to update user policies	kafka	updateUserPoliciesTaskFailure

Viewing Audit Logs

See **Querying Real-Time Traces**.