Distributed Message Service for Kafka

User Guide

Issue 01

Date 2024-04-01

HUAWEI

HUAWEI CLOUD COMPUTING TECHNOLOGIES CO., LTD.

Trademarks and Permissions


and other Huawei trademarks are the property of Huawei Technologies Co., Ltd.
All other trademarks and trade names mentioned in this document are the property of their respective holders.

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 Permissions Management

1.1 Creating a 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.

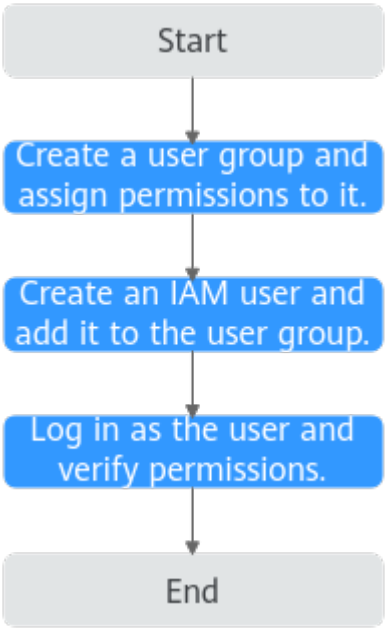
This section describes the procedure for granting permissions (see **Figure 1-1**).

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 1-1 Process for granting DMS for Kafka permissions



1. On the IAM console, **create a user group and grant it permissions**.
   DMS ReadOnlyAccess is used as an example.
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 have insufficient permissions to 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 to access the service, the **DMS ReadOnlyAccess** policy is in effect.

1.2 DMS for Kafka Custom Policies

Custom policies can be created to supplement the system-defined policies of DMS for Kafka. For the actions that can be added for custom policies, see **Permissions Policies and Supported Actions**.

You can create custom policies in either of the following ways:

- Visual editor: Select cloud services, actions, resources, and request conditions. This does not require knowledge of policy syntax.
- JSON: Edit JSON policies from scratch or based on an existing policy.

For details, see **Creating a Custom Policy**. The following section contains 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 Custom Policies

- Example 1: Allowing users to delete and restart instances

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

- Example 2: Denying instance deletion

  A policy with only "Deny" permissions must be used in conjunction with other policies to take effect. If the permissions assigned to a user include both "Allow" and "Deny", the "Deny" permissions take precedence over the "Allow" permissions.

  For example, if you want to assign all of the permissions of the **DMS FullAccess** policy to a user, except for deleting instances, you can create a custom policy to deny only instance deletion. When you apply both the **DMS FullAccess** policy and the custom policy denying instance deletion, since "Deny" always takes precedence over "Allow", the "Deny" will be applied for that one conflicting permission. The user will then be able to perform all operations on instances except deleting instances. The following is an example of a deny policy:

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

1.3 DMS for Kafka Resources

A resource is an object that exists within a service. DMS for Kafka resources are Kafka instances. You can select them by specifying their paths.

Table 1-1 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 instance, add the *instance ID* to the end. You can also use an asterisk * to indicate any instance. For example: **DMS:*:*:kafka:*** indicates any Kafka instance.

1.4 DMS for Kafka Request Conditions

Request conditions are useful for fine tuning when a custom policy takes effect. A request condition consists of a condition key and operator. 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-level condition keys (starting with a service name such as *dms:*) are available only for operations of a specific service. 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, you can use the condition key **dms:ssl** to check whether SASL is enabled for a Kafka instance. The following table lists the predefined condition keys of DMS for Kafka.

Table 1-2 Predefined condition keys of DMS for Kafka

Condition Key	Operator	Description
dms:connector	Bool Null	Whether Smart Connect is enabled
dms:publicIP	Bool Null	Whether public access is enabled
dms:ssl	Bool Null	Whether SASL is enabled

2 Single-node Kafka Instances

A single-node Kafka instance has only one broker. These instances do not guarantee performance or reliability and are for trial use or testing only. In the production environment, use cluster instances.

NOTE

Single-node Kafka instances are available only in CN East-Shanghai1 and CN South-Guangzhou regions.

Parameters for Purchase

Parameter settings that are unique to single-node instances are listed in **Table 2-1**.

Table 2-1 Parameters of a single-node instance

Parameter	Description
AZ	Only one AZ
Version	Only v2.7
Broker flavor	Only kafka.2u4g.single.small and kafka.2u4g.single
Broker quantity	Only one
Storage space per broker	100 GB to 10,000 GB
Ciphertext access	Not supported

Instance Function Differences

Table 2-2 compares the functions of single-node and cluster instances.

Table 2-2 Function differences

Function	Single-Node	Cluster
Modifying instance specifications	×	√
Changing the instance access mode	You can only enable or disable plaintext for public network access.	Options: • Enabling private network ciphertext access • Enabling/Disabling private network plaintext access • Enabling/Disabling public network plaintext access • Enabling/Disabling public network ciphertext access
Resetting Kafka password	×	√
Viewing disk usage	×	√
Reassigning partitions	×	√
Configuring topic permissions	×	√
Managing users	×	√
Viewing rebalancing logs	×	√
Smart Connect	×	√
Managing Kafka quotas	×	√
Modifying Kafka parameters	×	√

3 Preparing Required Resources

Overview

Before creating a Kafka instance, ensure the availability of resources, including a virtual private cloud (VPC), subnet, security group, and security group rules. Each Kafka instance is deployed in a VPC and bound to a specific subnet and security group. In this way, Kafka provides an isolated virtual network environment and security protection policies that you can easily configure and manage.

To access a Kafka instance over a public network, prepare an elastic IP address (EIP) in advance.

Required Resources

Table 3-1 lists the resources required by a Kafka instance.

Table 3-1 Kafka resources

Resource	Requirement	Operations
VPC and subnet	Different Kafka instances can use the same or different VPCs and subnets based on site requirements. Note the following when creating a VPC and a subnet: • The VPC must be created in the same region as the Kafka instance. • Use the default settings when creating a VPC and subnet.	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**.

Resource	Requirement	Operations
Security group	Different Kafka instances can use the same or different security groups. To use Kafka instances, add the security group rules described in **Table 3-2**. Other rules can be added based on site requirements. **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 3-2**.	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**.
EIP	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**.

Table 3-2 Security group rules

Direction	Protocol	Port	Source	Description
Inbound	TCP	9094	0.0.0.0/0	Access a Kafka instance through the public network (without SSL encryption).
Inbound	TCP	9092	0.0.0.0/0	Access a Kafka instance within a VPC (without SSL encryption).
Inbound	TCP	9095	0.0.0.0/0	Access a Kafka instance through the public network (with SSL encryption).
Inbound	TCP	9093	0.0.0.0/0	Access a Kafka instance within a VPC (with SSL encryption).

Direction	Protocol	Port	Source	Description
Inbound	TCP	9011	198.19.128.0/17	Access a Kafka instance across VPCs using a VPC endpoint (with or without SSL).
Inbound	TCP	9011	0.0.0.0/0	Access a Kafka instance using DNAT (with or without SSL).

4 Buying an Instance

Scenario

Kafka instances are physically isolated and exclusively occupied by each tenant. You can customize the computing capabilities and storage space of an instance based on service requirements.

Before You Start

- Before buying a Kafka instance, ensure that a VPC configured with security groups and subnets is available.
- (Optional) If you want to access a Kafka instance over a public network, prepare an elastic IP address (EIP) in advance.

Procedure

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

Step 2 Specify **Billing Mode**, **Region**, **Project**, and **AZ**.

Step 3 Enter an instance name and select an enterprise project.

Step 4 Configure the following instance parameters:

Specifications: Select **Cluster** or **Custom**. Alternatively, select **Single-node**.

- If you select **Cluster**, specify the version, broker flavor and quantity, and storage space to be supported by the Kafka instance based on site requirements. Cluster instances support Kafka versions 1.1.0, 2.7, and 3.x.
- If you select **Custom**, the system calculates the broker quantity and storage space for different flavors based on your specified parameters (creation traffic peak, retrieval traffic, number of replicas per topic, total number of partitions, and size of messages created during the retention period). You can select one of the recommended flavors as required.
- If you select **Single-node**, a v2.7 instance with one broker will be created. For details about single-node instances, see **Single-node Kafka Instances**.

If you select Cluster, specify the version, broker flavor and quantity, and storage space to be supported by the Kafka instance based on site requirements.

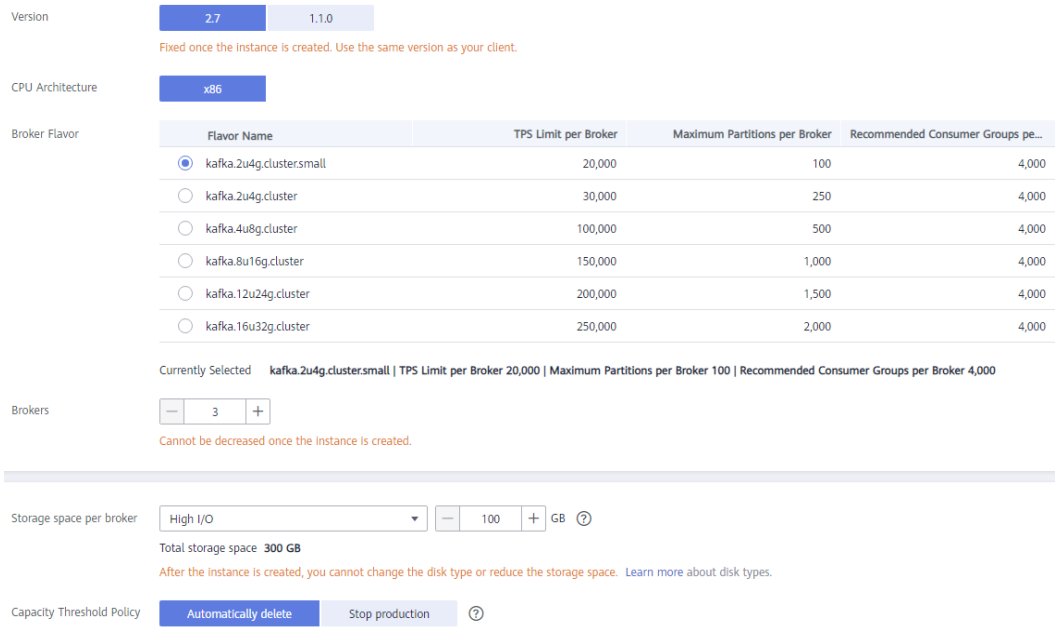
1. **Version**: Kafka v1.1.0 and v2.7 are supported. v2.7 is recommended. **The version cannot be changed once the instance is created.**
2. **CPU Architecture**: The x86 architecture is supported.
3. **Broker Flavor**: Select broker specifications that best fit your business needs.

   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.
4. For **Brokers**, specify the broker quantity.
5. **Storage Space per Broker**: Disk type and total disk space for storing the instance data. **The disk type cannot be changed once the instance is created.**

   The storage space is the total space to be consumed by all replicas. Specify the storage space based on the expected service message size and the number of replicas. For example, if the required disk size to store the data for the retention period is 100 GB, the disk capacity must be at least: 100 GB x Number of replicas + 100 GB (reserved).

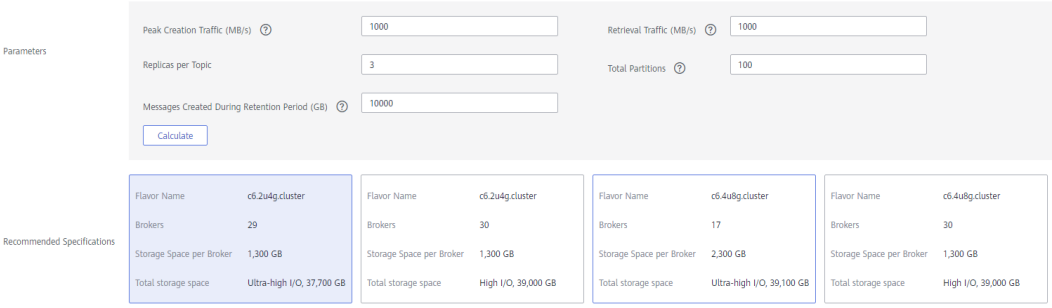
   Disks are formatted when an instance is created. As a result, the actual available disk space is 93% to 95% of the total disk space.
6. **Capacity Threshold Policy**: policy used when the disk usage reaches the threshold. The capacity threshold is 95%.
   - **Automatically delete**: Messages can be created 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.

Figure 4-1 Default specifications



If you select Custom, the system calculates the number of brokers and broker storage space for different flavors based on your specified peak creation traffic, retrieval traffic, number of replicas per topic, total number of partitions, and size of messages created during the retention period. You can select one of the recommended flavors as required.

Figure 4-2 Specification calculation



If you select Single-node, a v2.7 instance with one broker will be created.

1. **Version**: Kafka version, which can only be 2.7.
2. **CPU Architecture**: The x86 architecture is supported.
3. **Broker Flavor**: Select broker specifications that best fit your needs.
4. **Brokers**: The instance can have only one broker.
5. **Storage space per broker**: Select the desired disk type for storing Kafka data. The disk space is 100 GB and cannot be changed.

   The disk type cannot be changed once the instance is created.

   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.
6. **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.

Step 5 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.
- For **Private IP Addresses**, select **Auto** or **Manual**.
  - **Auto**: The system automatically assigns an IP address from the subnet.
  - **Manual**: Select IP addresses from the drop-down list.

NOTE

In the CN North-Beijing4, CN East-Shanghai1, and CN South-Guangzhou regions, **Private IP Addresses** has been moved to the **Private Network Access** area. For details, see **Step 6**.

- 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.

Step 6 Configure the instance access mode.

Table 4-1 Instance access modes

Public or Private Network	Plaintext or Ciphertext	Description
Private Network Access	Plaintext Access	Clients connect to the Kafka instance without SASL authentication. Once enabled, private network access cannot be disabled. Enable plaintext or ciphertext access, or both.
	Ciphertext Access	Clients connect to the Kafka instance with SASL authentication. Once enabled, private network access cannot be disabled. Enable plaintext or ciphertext access, or both. **To disable ciphertext access, contact customer service.** If you enable **Ciphertext Access**, specify a security protocol, SASL/PLAIN, username, and password.
	Private IP Addresses	Select **Auto** or **Manual**. • **Auto**: The system automatically assigns an IP address from the subnet. • **Manual**: Select IP addresses from the drop-down list. If the number of selected IP addresses is less than the number of brokers, the remaining IP addresses will be automatically assigned.
Public Network Access	Plaintext Access	Clients connect to the Kafka instance without SASL authentication. Enable or disable plaintext access, and configure addresses for public network access.

Public or Private Network	Plaintext or Ciphertext	Description
	Ciphertext Access	Clients connect to the Kafka instance with SASL authentication. Enable or disable ciphertext access, and configure addresses for public network access. If you enable **Ciphertext Access**, specify a security protocol, SASL/PLAIN, username, and password.
	Public IP Addresses	Select the number of public IP addresses as required. If EIPs are insufficient, click **Create Elastic IP** to create EIPs. Then, return to the Kafka console and click next to **Public IP Address** to refresh the public IP address list. **Kafka instances only support IPv4 EIPs.**

NOTE

This function is unavailable for single-node instances.

The security protocol, SASL/PLAIN mechanism, username, and password are described as follows.

Table 4-2 Ciphertext access parameters

Parameter	Value	Description
Security Protocol	SASL_SSL	SASL is used for authentication. Data is encrypted with SSL certificates for high-security transmission. This protocol supports the SCRAM-SHA-512 and PLAIN mechanisms. **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.

Parameter	Value	Description
	SASL_PLAINTEXT	SASL is used for authentication. Data is transmitted in plaintext for high performance. This protocol supports the SCRAM-SHA-512 and PLAIN mechanisms. SCRAM-SHA-512 authentication is recommended for plaintext transmission.
SASL/PLAIN	-	• If **SASL/PLAIN** is disabled, the SCRAM-SHA-512 mechanism is used for username and password authentication. • 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 ciphertext access is enabled.
Username and Password	-	Username and password used by the client to connect to the Kafka instance. The username cannot be changed once ciphertext access is enabled.

NOTE

Instance access modes are available only in the CN North-Beijing4, CN East-Shanghai1, and CN South-Guangzhou regions.

Step 7 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 a new 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.

NOTE

- In the CN North-Beijing4, CN East-Shanghai1, and CN South-Guangzhou regions, **Kafka SASL_SSL** has been moved to the **Private Network Access** and **Public Network Access** areas. For details, see **Step 6**.
- Single-node instances do not have this parameter.

Step 8 Specify the required duration.

This parameter is displayed only if the billing mode is yearly/monthly.

Step 9 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 a new instance.

   NOTE

   Public Network Access is no longer under **Advanced Settings** in the CN North-Beijing4, CN East-Shanghai1, and CN South-Guangzhou regions. For details, see **Step 6**.

2. Configure **Smart Connect**.

   Smart Connect is used for data synchronization between heterogeneous systems. You can configure Smart Connect tasks to synchronize data between Kafka and another cloud service or between two Kafka instances.

   NOTE

   This parameter is not available for single-node Kafka instances.

3. Configure **Automatic Topic Creation**.

   This setting is disabled by default. You can enable or disable it as required.

   If automatic topic creation is enabled, the system automatically creates a topic when a message is created in or retrieved from a topic that does not exist. This topic has the following default settings: 3 partitions, 3 replicas, aging time 72 hours, and synchronous replication and flushing disabled.

   After you change the value of the **log.retention.hours**, **default.replication.factor**, or **num.partitions** parameter, automatically created topics later use the new value. For example, if **num.partitions** is set to **5**, an automatically created topic will have the following settings: 5 partitions, 3 replicas, aging time 72 hours, and synchronous replication and flushing disabled.

4. 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 your organization has configured tag policies for DMS for Kafka, add tags to Kafka instances based on the policies. If a tag does not comply with the policies, Kafka instance creation may fail. Contact your organization administrator to learn more about tag policies.

- 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 **Managing Instance Tags**.

5. Enter a description of the instance.

Step 10 Click **Buy**.

Step 11 Confirm the instance information, and read and agree to the *HUAWEI CLOUD Customer Agreement*. If you have selected the yearly/monthly billing mode, click **Pay Now** and make the payment as prompted. If you have selected the pay-per-use mode, click **Submit**.

Step 12 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 **Creation failed** state, delete it by referring to **Deleting an Instance**. Then create a new one. If the instance creation fails again, contact customer service.

  NOTE

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

----End

5 Accessing a Kafka Instance

5.1 Accessing a Kafka Instance Without SASL

This section describes how to use an open-source Kafka client to access a Kafka instance if SASL access is not enabled for the instance. There are two scenarios. For cross-VPC access, see **Cross-VPC Access to a Kafka Instance**. For DNAT-based access, see **Using DNAT to Access a Kafka Instance**.

For details on how to use Kafka clients in different languages, visit **https://cwiki.apache.org/confluence/display/KAFKA/Clients**.

NOTE

- The following describes the procedure for accessing a Kafka instance using CLI. To access an instance in your service code, see the **Developer Guide**.
- For instances purchased in July 2020 and later, each Kafka broker allows a maximum of 1000 connections from each IP address by default. For instances purchased before July 2020, each Kafka broker allows a maximum of 200 connections from each IP address by default. Excess connections will be rejected. You can change the limit by referring to **Modifying Kafka Parameters**.

Prerequisites

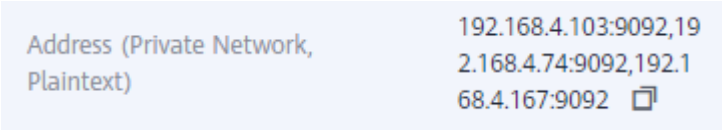
- Security group rules have been properly configured.

  To access a Kafka instance with SASL disabled, configure proper security group rules. For details about security group configuration requirements, see **Table 3-2**.
- The instance connection address has been obtained.
  - For intra-VPC access, use port 9092. Obtain the instance connection address in the **Connection** section of the **Basic Information** tab page.

    Figure 5-1 Kafka instance connection addresses for intra-VPC access without SASL (in regions except CN North-Beijing4, CN East-Shanghai1, and CN South-Guangzhou)

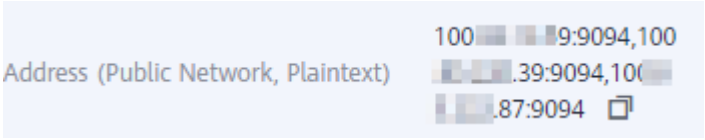
Figure 5-2 Kafka instance connection addresses for intra-VPC access without SASL (in regions CN North-Beijing4, CN East-Shanghai1, and CN South-Guangzhou)



– For public access, use port 9094. Obtain the instance connection address in the **Connection** section of the **Basic Information** tab page.

Figure 5-3 Kafka instance connection addresses for public access without SASL (in regions except CN North-Beijing4, CN East-Shanghai1, and CN South-Guangzhou)

Figure 5-4 Kafka instance connection addresses for public access without SASL (in regions CN North-Beijing4, CN East-Shanghai1, and CN South-Guangzhou)



- 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.
- An ECS has been created. For intra-VPC access, ensure that its VPC, subnet, and security group configurations are the same as those of the Kafka instance. **JDK v1.8.111 or later** has been installed on the ECS, 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**. For public access, use **Instance Address (Public Network)**/**Address (Public Network, Plaintext)**. For intra-VPC access, use **Instance Address (Private Network)**/**Address (Private Network, Plaintext)**.
- *{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**. For public access, use **Instance Address (Public Network)**/**Address (Public Network, Plaintext)**. For intra-VPC access, use **Instance Address (Private Network)**/**Address (Private Network, Plaintext)**.
- *{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

5.2 Accessing a Kafka Instance with SASL

If you enable SASL_SSL when creating an instance, data will be encrypted before transmission for enhanced security.

For security purposes, TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256, TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256, and TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 are supported for instances created on and before March 20, 2021.
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 is also supported for instances created after March 20, 2021.

This section describes how to use an open-source Kafka client to access a Kafka instance if SASL has been enabled for the instance. There are two scenarios. For cross-VPC access, see **Cross-VPC Access to a Kafka Instance**. For DNAT-based access, see **Using DNAT to Access a Kafka Instance**.

NOTE

- For instances purchased in July 2020 and later, each Kafka broker allows a maximum of 1000 connections from each IP address by default. For instances purchased before July 2020, each Kafka broker allows a maximum of 200 connections from each IP address by default. Excess connections will be rejected. You can change the limit by referring to **Modifying Kafka Parameters**.
- 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 **Accessing a Kafka Instance Without SASL**.
- The following describes the procedure for accessing a Kafka instance using CLI. To access an instance in your service code, see the **Developer Guide**.

Prerequisites

- Security group rules have been properly configured.

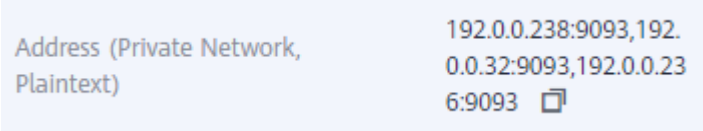
  To access a Kafka instance with SASL enabled, configure proper security group rules. For details about security group configuration requirements, see **Table 3-2**.

- The instance connection address has been obtained.
  - For intra-VPC access, use port 9093. Obtain the instance connection address in the **Connection** section of the **Basic Information** tab page.

    Figure 5-5 Kafka instance connection addresses for intra-VPC access with SASL (in regions except CN North-Beijing4, CN East-Shanghai1, and CN South-Guangzhou)

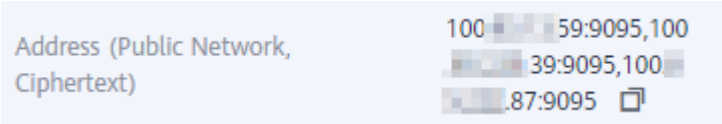
Figure 5-6 Kafka instance connection addresses for intra-VPC access with SASL (in regions CN North-Beijing4, CN East-Shanghai1, and CN South-Guangzhou)



– For public access, use port 9095. Obtain the instance connection address in the **Connection** section of the **Basic Information** tab page.

Figure 5-7 Kafka instance connection addresses for public access with SASL (in regions except CN North-Beijing4, CN East-Shanghai1, and CN South-Guangzhou)

Figure 5-8 Kafka instance connection addresses for public access with SASL (in regions CN North-Beijing4, CN East-Shanghai1, and CN South-Guangzhou)



- 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, configure either of them for connections. If **SASL Mechanism** is not displayed, PLAIN is used by default.

  Figure 5-9 SASL mechanism in use

- The security protocol in use is known.

  In the **Connection** area on the Kafka instance details page, view **Security Protocol**. If **Security Protocol** is not displayed, SASL_SSL is used by default.
- 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.
- An ECS has been created. For intra-VPC access, ensure that its VPC, subnet, and security group configurations are the same as those of the Kafka instance. **JDK v1.8.111 or later** has been installed on the ECS, 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
  ```

  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.

- **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
  ```


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.

Step 4 Modify the Kafka CLI configuration file based on the **security protocol**.

- **SASL_SSL**: Find the **consumer.properties** and **producer.properties** files in the **/config** directory of the Kafka CLI and add the following content to the files:

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

  Parameter description:

  - **ssl.truststore.location**: path for storing the **client.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**.

- **SASL_PLAINTEXT**: Find the **consumer.properties** and **producer.properties** files in the **/config** directory of the Kafka CLI and add the following content to the files:

```
security.protocol=SASL_PLAINTEXT
```

Step 5 Access the **/bin** directory of the Kafka CLI.

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

Step 6 Run the following command to create messages:

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

Parameter description:

- *{connection-address}*: the address obtained in **Prerequisites**. For public access, use **Instance Address (Public Network)**/**Address (Public Network, Ciphertext)**. For intra-VPC access, use **Instance Address (Private Network)**/**Address (Private Network, Ciphertext)**.
- *{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 7 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**. For public access, use **Instance Address (Public Network)**/**Address (Public Network, Ciphertext)**. For intra-VPC access, use **Instance Address (Private Network)**/**Address (Private Network, Ciphertext)**.
- *{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

5.3 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 17, 2023 do not have Kafka Manager. Kafka Manager's functions are provided on the Kafka console.

Table 5-1 Kafka Manager functions on the Kafka console

Kafka Manager	Kafka Console
Viewing topics about an instance	View the topic list on the **Topics** page.

Kafka Manager	Kafka Console
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 5-2**.

Table 5-2 Security group rule

Direction	Protocol	Port	Source	Description
Inbound	TCP	9999	0.0.0.0/0	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.

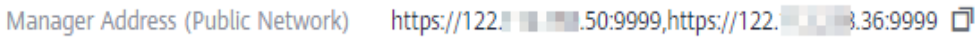
Step 2 Obtain the Kafka Manager address on the instance details page.

- If public network access has been disabled, the Kafka Manager address is **Manager Address (Private Network)**.

Figure 5-10 Kafka Manager address (private network)

- If public network access has been enabled, the Kafka Manager address is **Manager Address (Public Network)**.

  Figure 5-11 Kafka Manager address (public network)

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

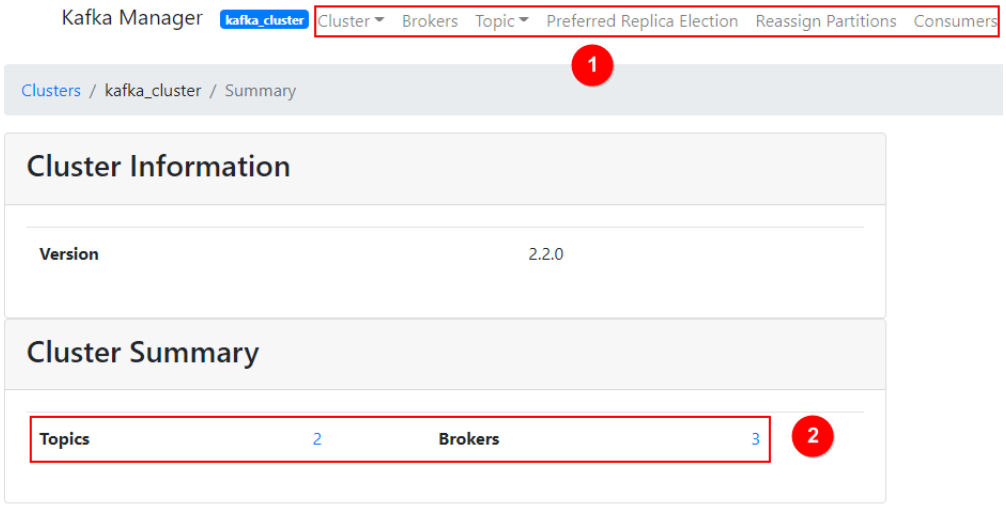
In Kafka Manager, you can view the monitoring statistics and broker information of your Kafka clusters.

- Information about clusters

  Click **Clusters** to view the information about clusters. **Figure 5-12** 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 5-12 Information about clusters



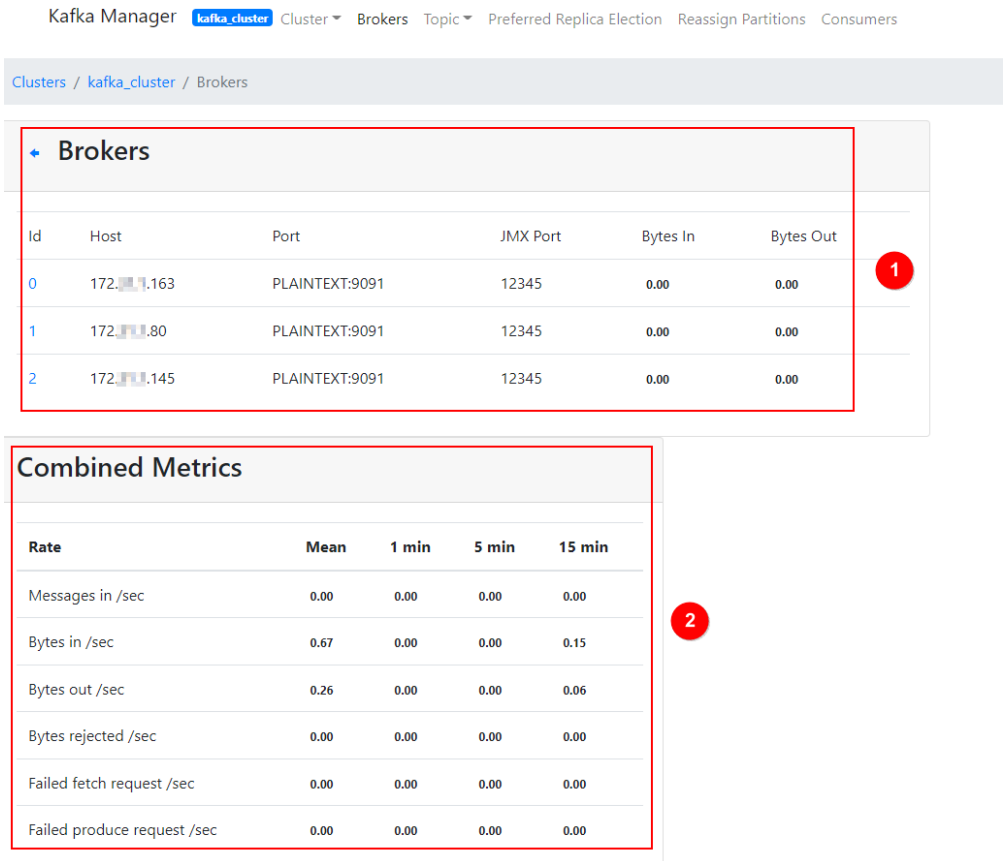
- Combined information about all brokers of a cluster

  This page shows statistics of brokers of a cluster. **Figure 5-13** 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 5-13 Viewing the combined information about all brokers in a cluster



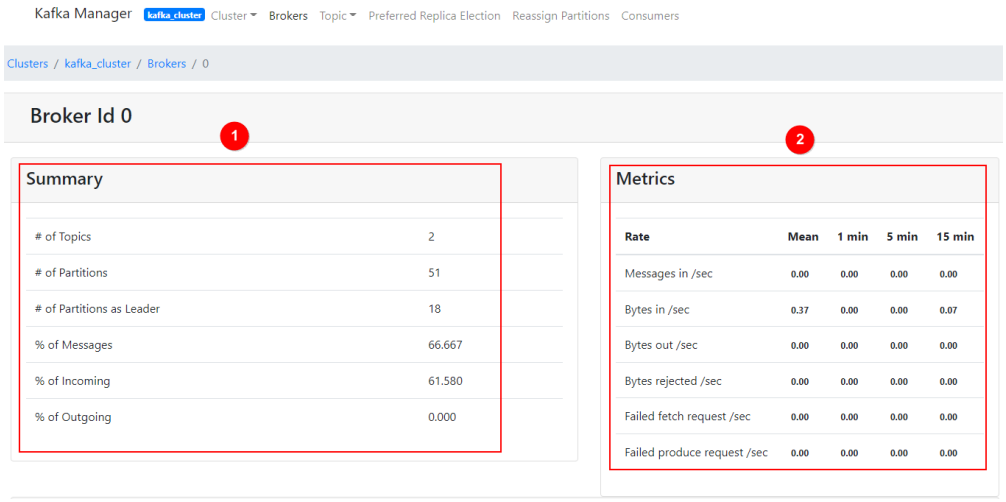
- Information about a specific broker

  Click the ID of a broker to view its statistics. **Figure 5-14** 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 5-14 Viewing information about a broker



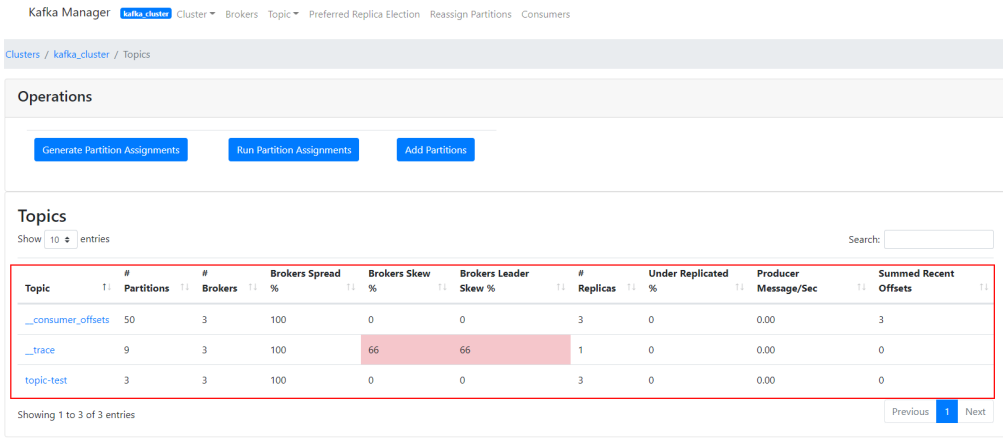
- Topics of an instance

  In the navigation bar, choose **Topic** > **List**. The displayed page shows the list of topics and information about the topics, as shown in **Figure 5-15**.

NOTICE

Topics starting with "__" are internal topics. To avoid service faults, do not perform any operation on these topics.

Figure 5-15 Topics of an instance

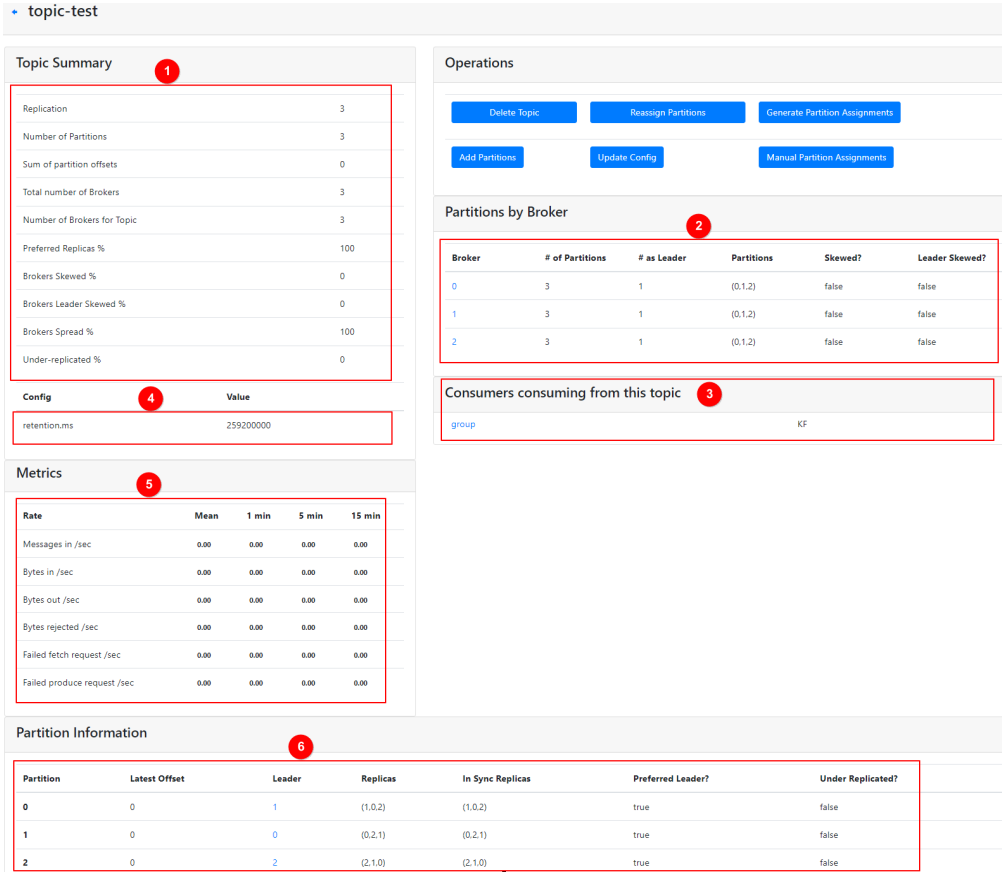


- Details of a topic

Click the name of a topic to view its details on the displayed page, as shown in **Figure 5-16**.

- 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 5-16 Details of a topic



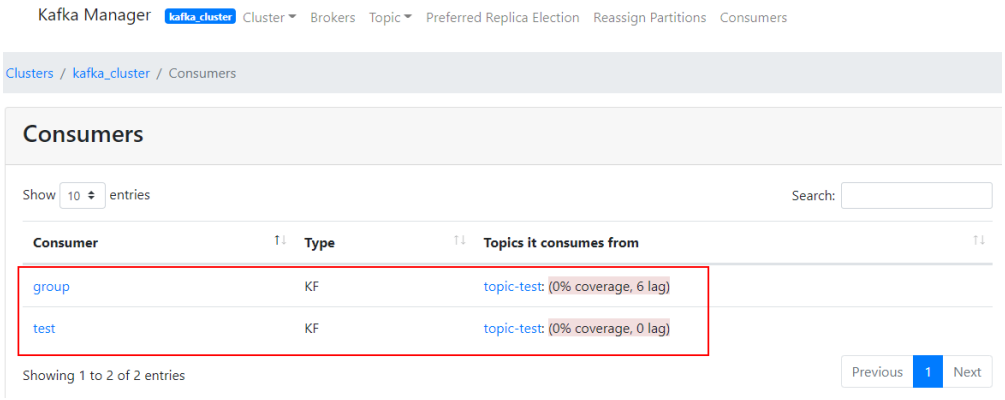
- List of consumers

  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.

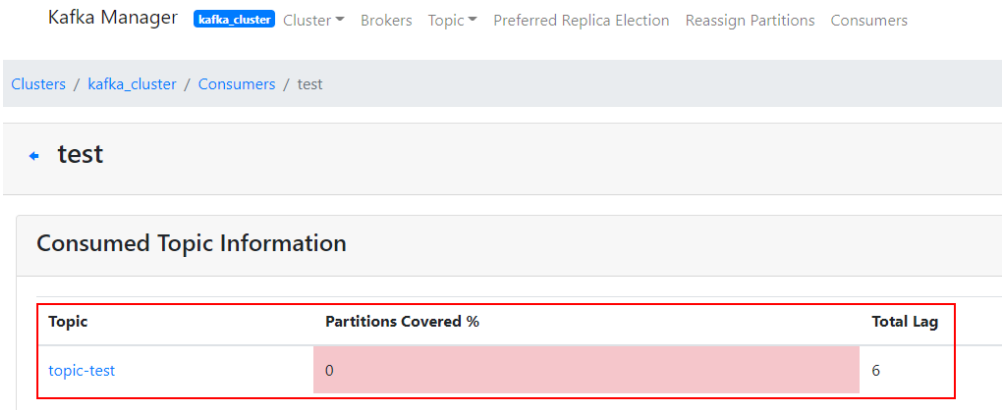
Figure 5-17 Viewing the list of consumers



- 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**).

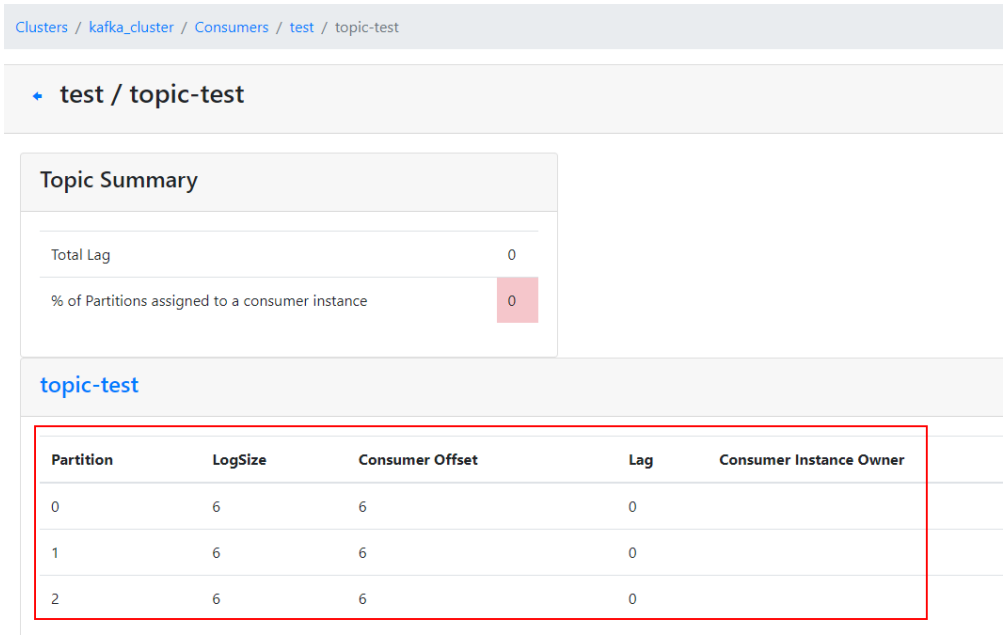
Figure 5-18 Viewing consumer details



- Details of topics in a consumer

  Click the name of a topic to view retrieval 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 5-19 Viewing details of a topic



5.4 Cross-VPC Access to a Kafka Instance

Context

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**.
- Create a cloud connection and load the VPCs that need to communicate with each other to the cloud connection. For details, see **Connecting VPCs in the Same Account**.
- Use VPC Endpoint (VPCEP) to establish a cross-VPC connection.

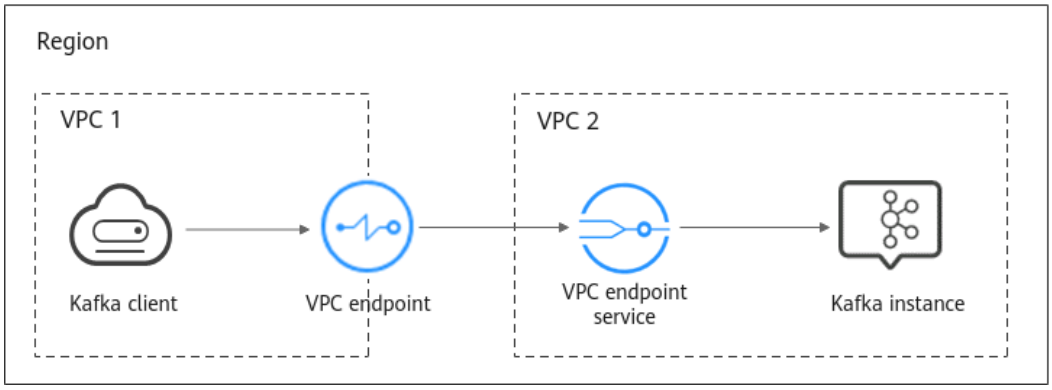
Scenario

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

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

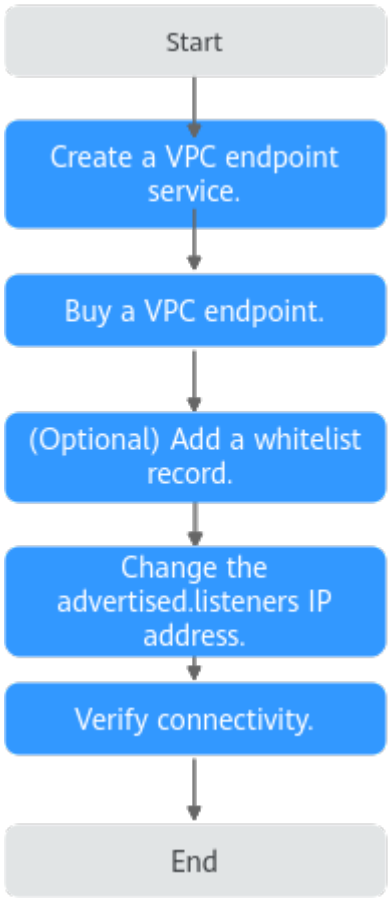
- A VPC endpoint service can be a Kafka instance which is accessed using VPC endpoints.
- A VPC endpoint is a secure and private channel for connecting a VPC to a VPC endpoint service.

Figure 5-20 Working principle of accessing a Kafka instance across VPCs



Procedure

Figure 5-21 Process for accessing a Kafka instance across VPCs



Creating a VPC Endpoint Service

Step 1 Log in to the management 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 **Middleware** > **Distributed Message Service (for Kafka)Application** > to open the console of DMS for Kafka.

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

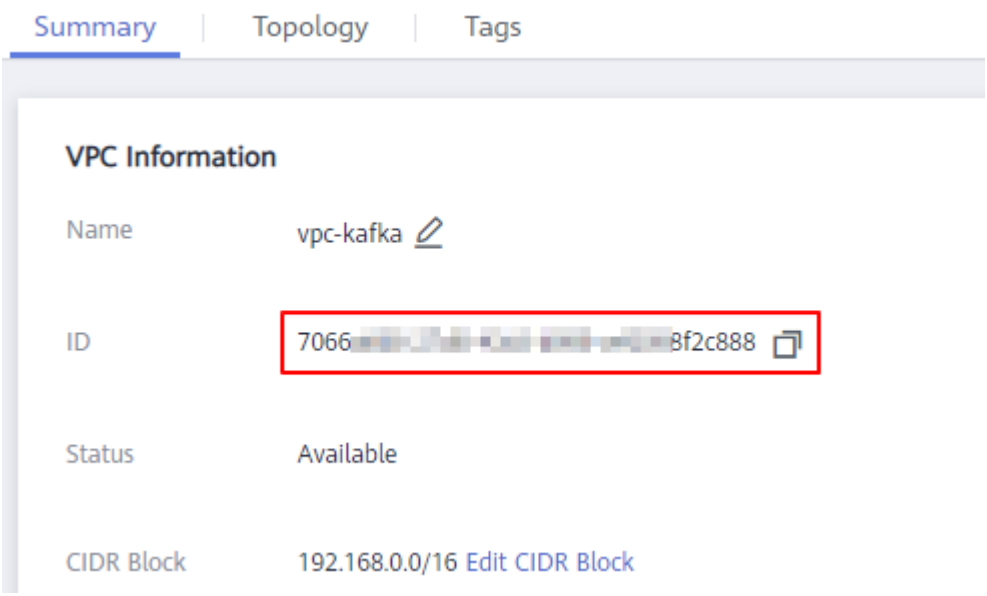
Figure 5-22 Cross-VPC access–related listeners IP addresses and corresponding port IDs of the Kafka instance

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

Figure 5-23 Viewing the VPC to which the Kafka instance belongs

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

Figure 5-24 Obtaining the VPC ID



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

```
curl -i -k -H 'Accept:application/json' -H 'Content-Type:application/json;charset=utf8' -X POST -H "X-Auth-Token:$token" -d '{"port_id":"38axxxeac","vpc_id":"706xxx888","ports":[{"protocol":"TCP","client_port":9011,"server_port":9011 }],"approval_enabled":false,"service_type":"interface","server_type":"VM"}' https://{endpoint}/v1/{project_id}/vpc-endpoint-services
```

Parameter description:

- **token**: an access credential issued to an IAM user to bear its identity and permissions. For details on how to obtain a token, see **Obtaining a User Token**.
- **port_id**: one of the port IDs obtained in **Step 5**.
- **vpc_id**: VPC ID obtained in **Step 7**.
- **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 Record

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**.

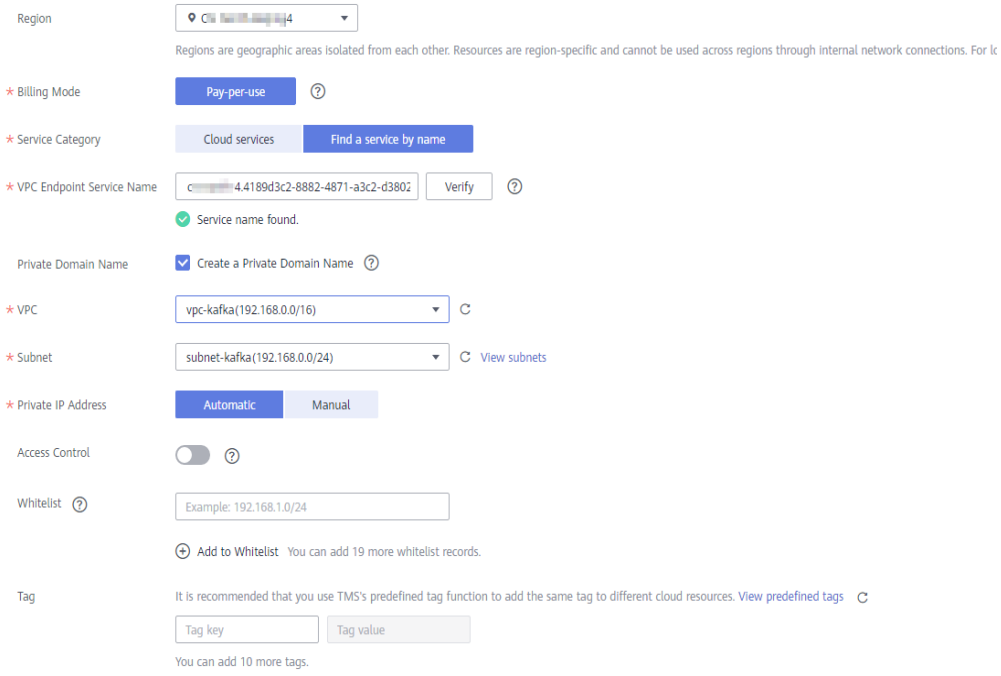
Step 2 Click **Buy VPC Endpoint**.

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 5-25 VPC endpoint parameters



Step 4 Click **Next**.

Step 5 Confirm the configurations and submit the request.

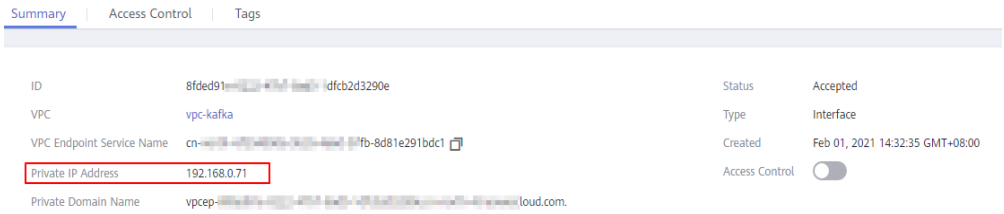
Step 6 Go back to the VPC endpoint list and check whether the status of the created VPC endpoint has changed to **Accepted**. The **Accepted** state means that the VPC endpoint has been connected to the VPC endpoint service.

Figure 5-26 Checking the VPC endpoint status

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 5-27 Viewing the private IP address



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

Changing the advertised.listeners IP Address

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

Step 2 Click the desired Kafka instance to view the instance 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 5-28 Changing the advertised.listeners IP addresses

----End

Verifying Connectivity

Check whether messages can be created and retrieved by referring to **Accessing a Kafka Instance Without SASL** or **Accessing a Kafka Instance with SASL**.

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 5-28** 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.

5.5 Using DNAT to Access a Kafka Instance

Scenario

You can use destination NAT (DNAT) to access a Kafka instance so that the instance can provide services on the public network through port mapping.

Prerequisites

You have purchased EIPs. The number of EIPs is the same as the number of brokers in the Kafka instance.

Step 1: Obtain Information About the Kafka Instance

Step 1 Log in to the management console.

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

NOTE

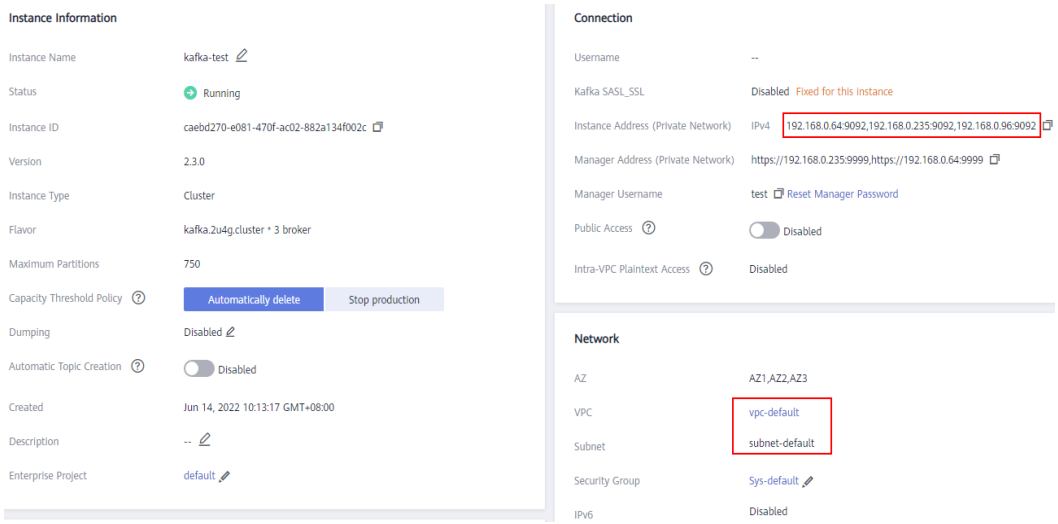
Select the region where your Kafka instance is located.

Step 3 Click in the upper left corner and choose **Middleware** > **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 **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 5-29 Kafka instance information



----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.

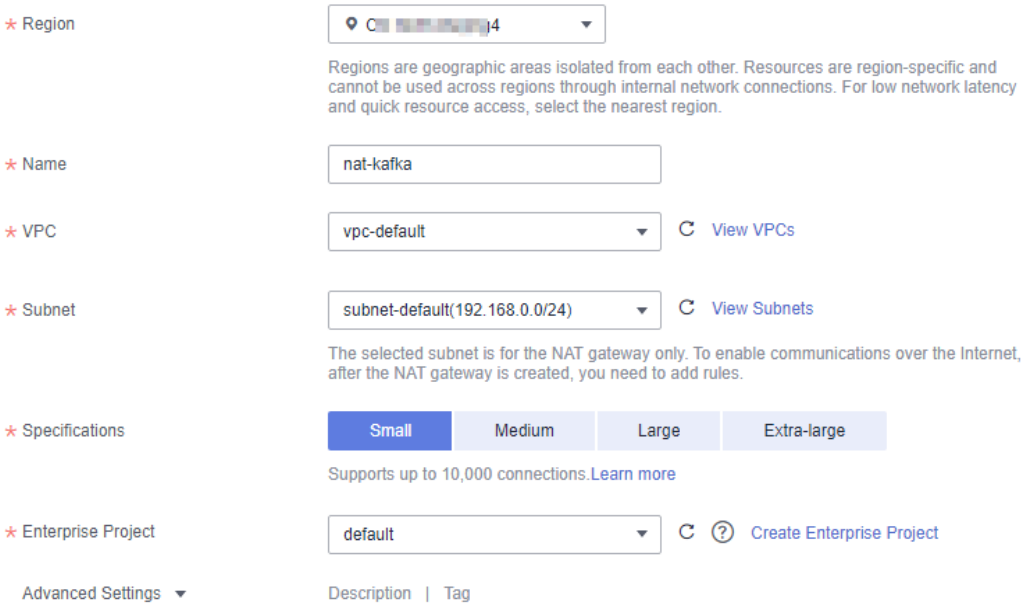
Step 2 Click **Buy Public NAT Gateway**.

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 **Buying a Public NAT Gateway**.

Figure 5-30 Buying a public NAT gateway



Step 4 Click **Next**.

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

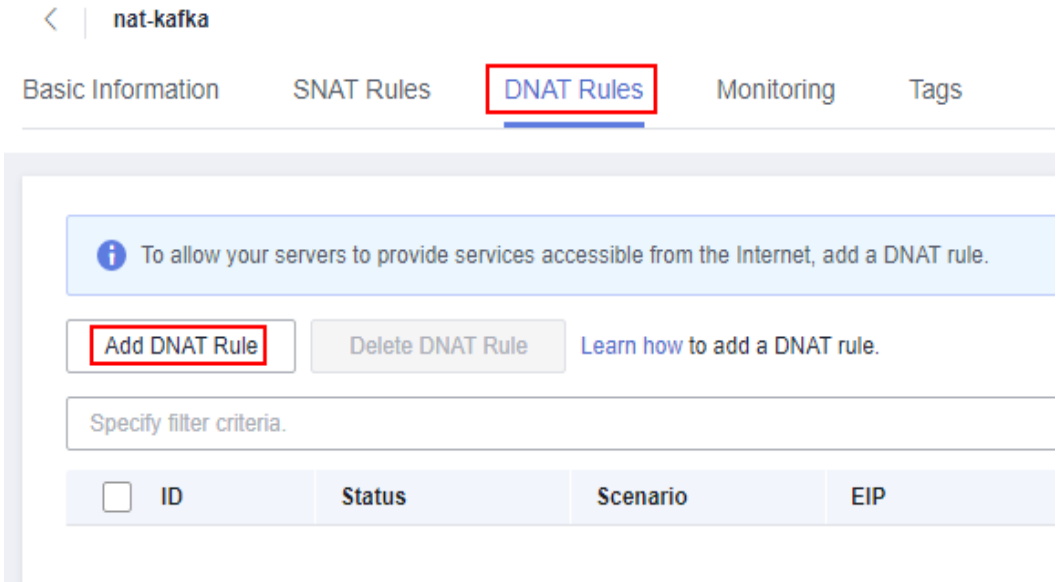
----End

Step 3: Add a DNAT Rule

Step 1 On **Public NAT Gateways** page, locate the row that contains 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 5-31 Public NAT gateway details

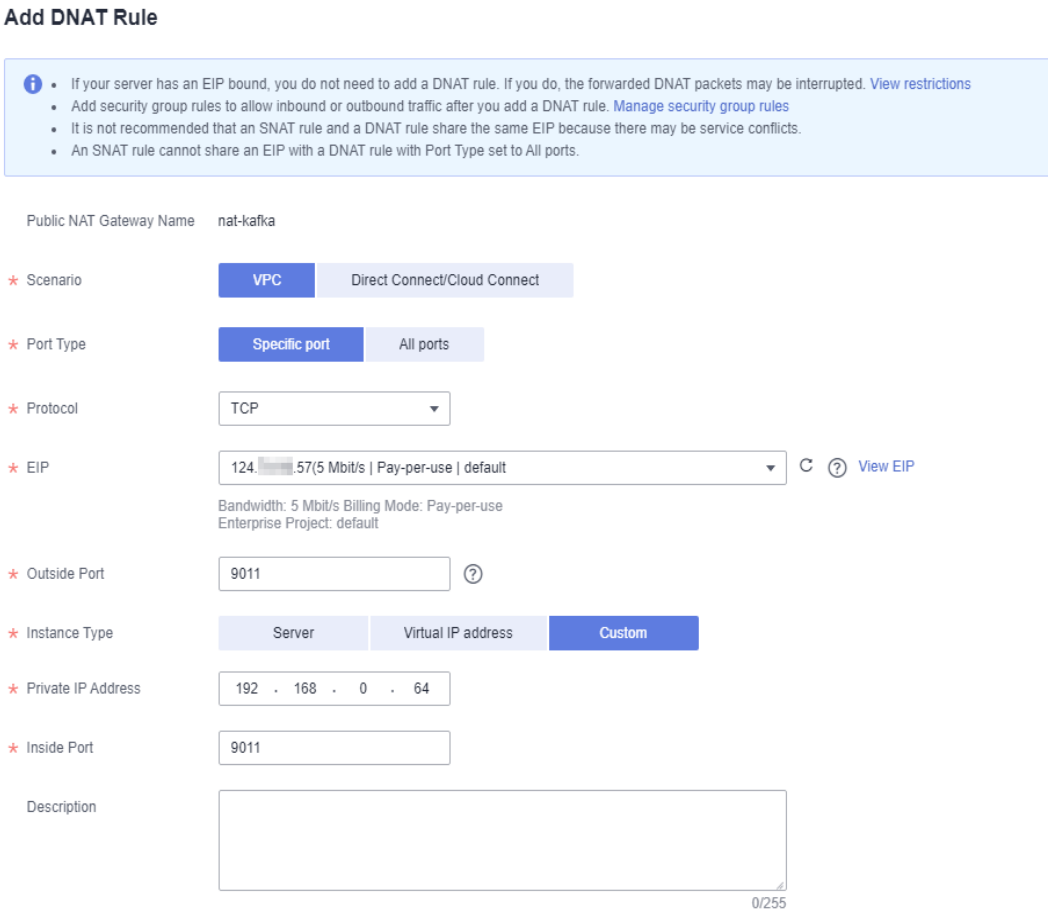


Step 3 Set the following parameters:

- **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 5-32 Adding a DNAT rule



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 5-33 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: Bind EIPs on the Kafka Console

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

Step 2 Click the desired Kafka instance to view the instance 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 5-34 Changing the advertised.listeners IP address (for DNAT access)

Cross-VPC Access

Modify

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 **Accessing a Kafka Instance Without SASL** or **Accessing a Kafka Instance with SASL**.

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 5-34** 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.

5.6 Generating and Replacing a Certificate

When connecting a Kafka client to a Kafka instance that has SASL enabled, use either the certificate provided by DMS for Kafka or your own certificate. This section describes how to generate your own certificate and use it to replace the one provided by DMS for Kafka.

To generate and replace certificates, contact background support personnel to enable the function for you. This function is available on a whitelist basis in all regions.

NOTE

Replacing the certificate will restart the instance. Exercise caution.

Prerequisites

- A Linux server is available.
- Kafka SASL_SSL has been enabled for the instance.

Step 1: Generating a Certificate

Step 1 Log in to the Linux server and run the following command to generate a keystore for the **server.keystore.jks** certificate:

```
keytool -genkey -keystore server.keystore.jks -alias localhost -validity 3650 -keyalg RSA
```

Enter a keystore password as prompted. The password must meet the following requirements:

- Contains 8 to 32 characters.
- Contains at least three of the following character types: letters, digits, spaces, and special characters `-!@#$ %^&*()-_=+\|[{}]:'",<.>/? and does not start with a hyphen (-).
- Cannot be a weak password. To check whether a password is weak, enter it in **Step 6**.

Enter the information about the certificate owner as prompted, such as the name, company, and city.

Step 2 Run the following command to generate a CA:

```
openssl req -new -x509 -keyout ca-key -out ca-cert -days 3650
```

Enter a PEM password as prompted.

Enter the information about the certificate owner as prompted.

Step 3 The certificate validity can be checked only after a truststore certificate is created. Run the following command to create a server truststore certificate with the generated CA:

```
keytool -keystore server.truststore.jks -alias CARoot -import -file ca-cert
```

Enter the server truststore password as prompted. The password must meet the following requirements:

- Contains 8 to 32 characters.
- Contains at least three of the following character types: letters, digits, spaces, and special characters `-!@#$ %^&*()-_=+\|[{}]:'",<.>/? and does not start with a hyphen (-).
- Cannot be a weak password. To check whether a password is weak, enter it in **Step 6**.

Enter **y** when the following information is displayed:

```
Trust this certificate?
```

Step 4 Run the following command to create a client truststore certificate with the CA:

```
keytool -keystore client.truststore.jks -alias CARoot -import -file ca-cert
```

Enter the client truststore password as prompted. This password is the value of **ssl.truststore.password** in the configuration file used by the client to connect to the Kafka instance.

Enter **y** when the following information is displayed:

```
Trust this certificate?
```

Step 5 Sign the server certificate.

1. Export the server certificate **server.cert-file**.

   ```
   keytool -keystore server.keystore.jks -alias localhost -certreq -file server.cert-file
   ```

   Enter the keystore password set in **Step 1** as prompted.

2. Sign the server certificate with the CA.

   ```
   openssl x509 -req -CA ca-cert -CAkey ca-key -in server.cert-file -out server.cert-signed -days 3650 -CAcreateserial
   ```

   Enter the PEM password set in **Step 2** as prompted.

3. Import the CA certificate to the server keystore.

   ```
   keytool -keystore server.keystore.jks -alias CARoot -import -file ca-cert
   ```

   Enter the keystore password set in **Step 1** as prompted.

   Enter **y** when the following information is displayed:

   ```
   Trust this certificate?
   ```

4. Import the signed server certificate to the server keystore.

   ```
   keytool -keystore server.keystore.jks -alias localhost -import -file server.cert-signed
   ```

   Enter the keystore password set in **Step 1** as prompted.

Step 6 Export the **server.keystore.jks** and **server.truststore.jks** certificates to the local PC.

Figure 5-35 Certificate directory

```
total 44
drwxr-xr-x  2 root root 4096 Aug 10 15:20 ./
drwxr-xr-x 10 root root 4096 Aug  8 17:04 ../
-rw-r--r--  1 root root 1322 Aug  8 17:07 ca-cert
-rw-r--r--  1 root root   41 Aug  8 17:09 ca-cert.srl
-rw-------  1 root root 1854 Aug  8 17:07 ca-key
-rw-r--r--  1 root root 1226 Aug  8 17:08 client.truststore.jks
-rw-r--r--  1 root root 1055 Aug  8 17:09 server.cert-file
-rw-r--r--  1 root root 1176 Aug  8 17:09 server.cert-signed
-rw-r--r--  1 root root 4693 Aug  8 17:10 server.keystore.jks
-rw-r--r--  1 root root 1226 Aug  8 17:08 server.truststore.jks
```

----End

Step 2: Replacing a Certificate

Step 1 Log in to the management console.

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

NOTE

Select the same region as your application service.

Step 3 Click and choose **Middleware** > **Distributed Message Service for Kafka** to open the console of DMS for Kafka.

Step 4 Click the desired instance to view its details.

Step 5 In the **Connection** area, click **Re-upload** next to **SSL Certificate**.

Step 6 Set the parameters for replacing the SSL certificate by referring to **Table 5-3**.

Table 5-3 Parameters for replacing the SSL certificate

Parameter	Description
Key Password	Enter the keystore password set in **Step 1**.
Keystore Password	Enter the keystore password set in **Step 1**.
Keystore File	Import the **server.keystore.jks** certificate.
Truststore Password	Enter the server truststore password set in **Step 3**.
Truststore File	Import the **server.truststore.jks** certificate.

Step 7 Click **OK**.

Step 8 Click **OK**.

On the **Background Tasks** page, if the certificate replacement task is **Successful**, the certificate is successfully replaced.

NOTE

After the original certificate is successfully replaced, you will download the certificate provided by DMS for Kafka rather than your own certificate by clicking **Download** on the **Basic Information** tab page.

----End

Step 3: Modifying Client Configuration Files

After a certificate is replaced, modify the **ssl.truststore.location** and **ssl.truststore.password** parameters in the **consumer.properties** and **producer.properties** files on the client, respectively.

```
security.protocol=SASL_SSL
ssl.truststore.location=/opt/kafka_2.12-2.7.2/config/client.truststore.jks
ssl.truststore.password=dms@kafka
ssl.endpoint.identification.algorithm=
```


- **ssl.truststore.location**: path for storing the **client.truststore.jks** certificate.
- **ssl.truststore.password**: **truststore password of the client certificate**
- **ssl.endpoint.identification.algorithm**: whether to verify the certificate domain name. **This parameter must be left blank, which indicates disabling domain name verification**.

5.7 Configuring Mutual SSL Authentication

Scenario

Mutual SSL authentication verifies the certificates of both the client and server during communication. This ensures that both parties involved in the communication are trusted.

Enable mutual SSL authentication to achieve high security.

To use mutual SSL authentication, contact background support personnel to enable it for you.

NOTE

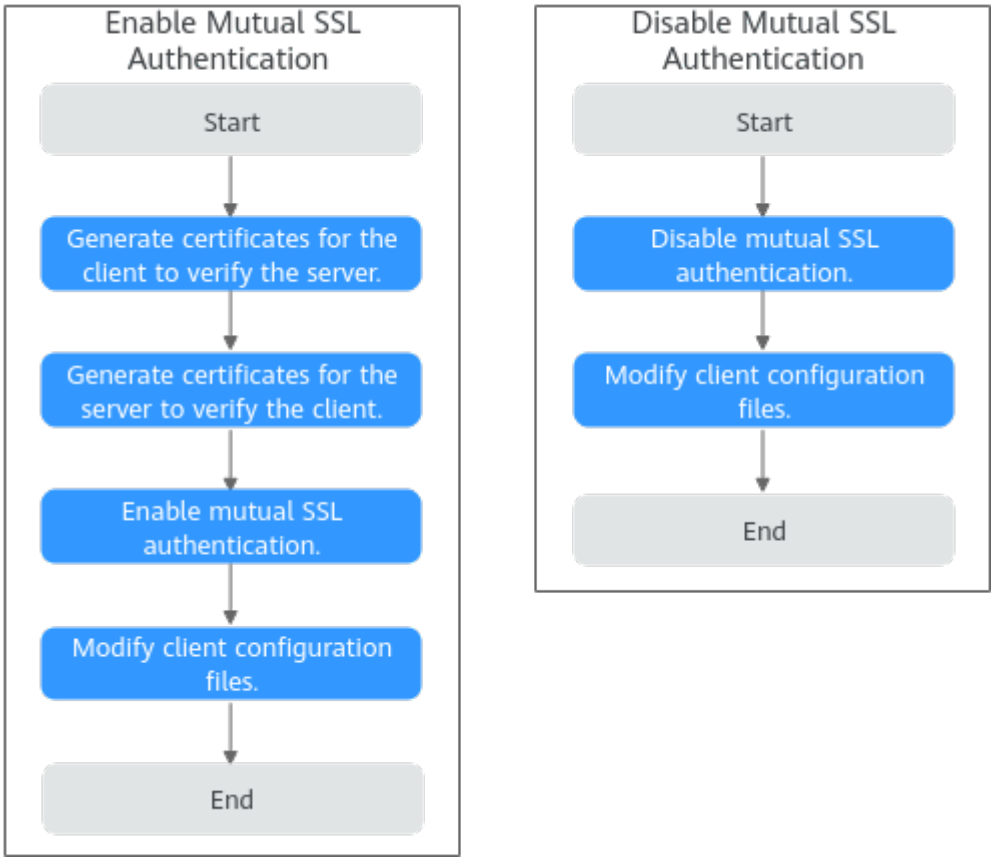
Enabling or disabling mutual SSL authentication will restart the instance. Exercise caution.

Prerequisites

- A Linux server is available.
- Kafka SASL_SSL has been enabled for the instance.

Overall Procedure

Figure 5-36 Overall procedure for configuring mutual SSL authentication



Step 1: Generate Certificates for the Client to Verify the Server

Step 1 Log in to the Linux server and run the following command to generate a keystore for the **server.keystore.jks** certificate:

```
keytool -genkey -keystore server.keystore.jks -alias localhost -validity 3650 -keyalg RSA
```

Enter a keystore password as prompted. The password must meet the following requirements:

- Contains 8 to 32 characters.
- Contains at least three of the following character types: letters, digits, spaces, and special characters `-!@#$ %^&*()-_=+\|[{}]:'",<.>/? and does not start with a hyphen (-).
- Cannot be a weak password. To check whether a password is weak, enter it in **Step 6**.

Enter the information about the certificate owner as prompted, such as the name, company, and city.

Step 2 Run the following command to generate a CA:

```
openssl req -new -x509 -keyout ca-key -out ca-cert -days 3650
```

Enter a PEM password as prompted.

Enter the information about the certificate owner as prompted.

Step 3 Run the following command to export the certificate from the **server.keystore.jks** file generated in **Step 1** and name the certificate **server.crt**:

```
keytool -keystore server.keystore.jks -alias localhost -certreq -file server.crt
```

Enter a keystore password as prompted.

Step 4 Run the following command to use the CA private key to sign **server.crt** and name the signed certificate **server-signed.crt**:

```
openssl x509 -req -CA ca-cert -CAkey ca-key -in server.crt -out server-signed.crt -days 3650 -CAcreateserial
```

Enter the PEM password set in **Step 2** as prompted.

Step 5 Run the following command to import the CA certificate and **server-signed.crt** to the keystore:

```
keytool -keystore server.keystore.jks -alias CARoot -import -file ca-cert
keytool -keystore server.keystore.jks -alias localhost -import -file server-signed.crt
```

Enter a keystore password as prompted.

Step 6 Run the following command to enable the client to trust the server certificate:

```
keytool -keystore client.truststore.jks -alias CARoot -import -file ca-cert
```

Enter a password for **client.truststore.jks** as prompted. The password must meet the following requirements:

- Contains 8 to 32 characters.
- Contains at least three of the following character types: letters, digits, spaces, and special characters `-!@#$ %^&*()-_=+\|[{}]:'",<.>/? and does not start with a hyphen (-).
- Cannot be a weak password. To check whether a password is weak, enter it in **Step 6**.

Step 7 Export the **client.truststore.jks** and **server.keystore.jks** certificates to the local PC.

----End

Step 2: Generate Certificates for the Server to Verify the Client

Step 1 Log in to the Linux server and run the following command to generate a keystore for the **client.keystore.jks** certificate:

```
keytool -genkey -keystore client.keystore.jks -alias localhost -validity 3650 -keyalg RSA
```

Enter a keystore password as prompted. The password must meet the following requirements:

- Contains 8 to 32 characters.
- Contains at least three of the following character types: letters, digits, spaces, and special characters `-!@#$ %^&*()-_=+\|[{}]:'",<.>/? and does not start with a hyphen (-).
- Cannot be a weak password. To check whether a password is weak, enter it in **Step 6**.

Enter the information about the certificate owner as prompted, such as the name, company, and city.

Step 2 Run the following command to generate a CA:

```
openssl req -new -x509 -keyout ca-key -out ca-cert -days 3650
```

Enter a PEM password as prompted.

Enter the information about the certificate owner as prompted.

Step 3 Run the following command to export the certificate from the **client.keystore.jks** file generated in **Step 1** and name the certificate **client.crt**:

```
keytool -keystore client.keystore.jks -alias localhost -certreq -file client.crt
```

Enter a keystore password as prompted.

Step 4 Run the following command to use the CA private key to sign **client.crt** and name the signed certificate **client-signed.crt**:

```
openssl x509 -req -CA ca-cert -CAkey ca-key -in client.crt -out client-signed.crt -days 3650 -CAcreateserial
```

Enter the PEM password set in **Step 2** as prompted.

Step 5 Run the following command to import the CA certificate and **client-signed.crt** to the keystore:

```
keytool -keystore client.keystore.jks -alias CARoot -import -file ca-cert
keytool -keystore client.keystore.jks -alias localhost -import -file client-signed.crt
```

Enter a keystore password as prompted.

Step 6 Run the following command to enable the server to trust the client certificate:

```
keytool -keystore server.truststore.jks -alias CARoot -import -file ca-cert
```

Enter a password for **server.truststore.jks** as prompted.

The password must meet the following requirements:

- Contains 8 to 32 characters.
- Contains at least three of the following character types: letters, digits, spaces, and special characters `-!@#$ %^&*()-_=+\|[{}]:'",<.>/? and does not start with a hyphen (-).
- Cannot be a weak password. To check whether a password is weak, enter it in **Step 6**.

Step 7 Export the **server.truststore.jks** and **client.keystore.jks** certificates to the local PC.

----End

Step 3: Enable Mutual SSL Authentication.

Step 1 Log in to the management console.

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

NOTE

Select the same region as your application service.

Step 3 Click and choose **Middleware** > **Distributed Message Service for Kafka** to open the console of DMS for Kafka.

Step 4 Click the desired Kafka instance.

Step 5 In the **Connection** area, click  next to **Mutual SSL Authentication**.

Step 6 In the displayed **Mutual SSL Authentication** dialog box, set the parameters by referring to **Table 5-4**.

Table 5-4 Parameters for enabling mutual SSL authentication

Parameter	Description
Key Password	Enter the **password of server.keystore.jks**.
Keystore Password	Enter the **password of server.keystore.jks**.
Keystore File	Import the **server.keystore.jks** certificate.
Truststore Password	Enter the **password of server.truststore.jks**.
Truststore File	Import the **server.truststore.jks** certificate.

NOTICE

Enabling mutual SSL authentication will restart the instance. Exercise caution.

Step 7 Click **OK**.

----End

Step 4: Modifying Client Configuration Files

After enabling mutual SSL authentication, modify the server certificate configuration and add the client certificate configurations in the **consumer.properties** and **producer.properties** files on the client.

```
security.protocol=SSL
ssl.truststore.location=/opt/kafka_2.12-2.7.2/config/client.truststore.jks
ssl.truststore.password=dms@kafka
ssl.endpoint.identification.algorithm=
# Add the following client certificate configurations:
ssl.keystore.location=/var/private/ssl/kafka/client.keystore.jks
ssl.keystore.password=txxx3
ssl.key.password=txxx3
```

- **security.protocol**: certificate protocol type. When enabling mutual SSL authentication, set this parameter to **SSL**.
- **ssl.truststore.location**: path for storing the **client.truststore.jks** certificate.
- **ssl.truststore.password**: **password of client.truststore.jks**.
- **ssl.endpoint.identification.algorithm**: whether to verify the certificate domain name. **This parameter must be left blank, which indicates disabling domain name verification**.
- **ssl.keystore.location**: path for storing the **client.keystore.jks** certificate.
- **ssl.keystore.password**: **password of client.keystore.jks**.

- **ssl.key.password**: **password of client.keystore.jks**.

Disabling Mutual SSL Authentication

Step 1 Log in to the management console.

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

NOTE

Select the same region as your application service.

Step 3 Click and choose **Middleware** > **Distributed Message Service for Kafka** to open the console of DMS for Kafka.

Step 4 Click the desired Kafka instance.

Step 5 In the **Connection** area, click next to **Mutual SSL Authentication**.

NOTICE

Disabling mutual SSL authentication will restart the instance. Exercise caution.

Step 6 After disabling mutual SSL authentication, modify the server certificate protocol and delete the client certificate configurations in the **consumer.properties** and **producer.properties** files on the client.

```
security.protocol=SASL_SSL
ssl.truststore.location=/opt/kafka_2.12-2.7.2/config/client.truststore.jks
ssl.truststore.password=dms@kafka
ssl.endpoint.identification.algorithm=
# Delete the following client certificate configurations:
ssl.keystore.location=/var/private/ssl/kafka.client.keystore.jks
ssl.keystore.password=txxx3
ssl.key.password=txxx3
```

security.protocol: certificate protocol type. When disabling mutual SSL authentication, set this parameter to **SASL_SSL**. You do not need to change the values of **ssl.truststore.location**, **ssl.truststore.password**, and **ssl.endpoint.identification.algorithm**.

----End

6 Managing Instances

6.1 Modifying Instance Specifications

Scenario

After creating a Kafka instance, you can increase or decrease its specifications. Table 6-1 lists available modification options.

Table 6-1 Specification modification options

Old/New Flavor	Modified Object	Increase	Decrease
New flavor	Broker quantity	√	×
	Storage space	√	×
	Broker flavor	√	√
Old flavor	Bandwidth	√	×
	Storage space	√	×
	Broker flavor	×	×

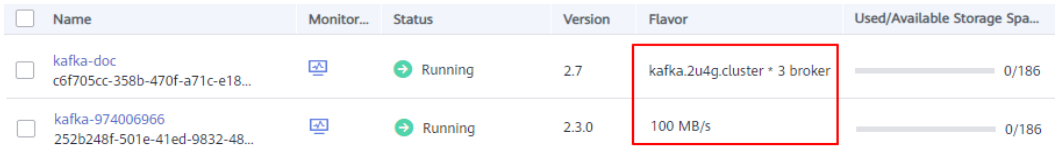
NOTE

Specifications cannot be changed for single-node Kafka instances.

Distinguishing between old and new specifications:

- Old specifications: In the instance list, the instance specification is displayed as bandwidth (for example, **100 MB/s**).
- New specifications: In the instance list, the instance specification is displayed as the ECS flavor multiplied by the number of brokers (for example, **kafka.2u4g.cluster*3 brokers**).

Figure 6-1 Instance list



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 6-2 Impact of specification modification

Modified Object	Impact
Broker quantity or bandwidth	• Adding brokers or increasing the bandwidth does not affect the original brokers or services. • When you increase the bandwidth or add brokers, 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

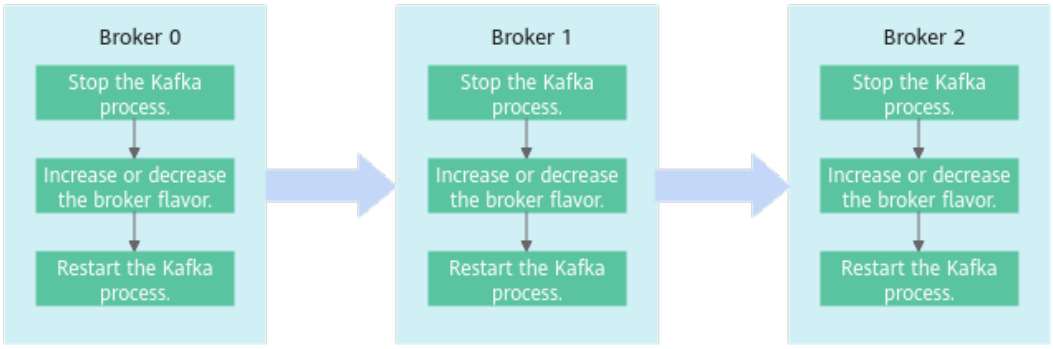
Broker flavor	• Single-replica topics do not support message creation and retrieval during this period. Services will be interrupted.
	• If a topic has multiple replicas, scaling up or down the broker flavor does not interrupt services, but may cause disorder of partition messages. Evaluate this impact 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)));`
	• If the total number of partitions created for an instance is greater than the upper limit allowed by a new flavor, scale-down cannot be performed. The maximum number of partitions varies with instance specifications. For details, see **Specifications**.
	For example, if 800 partitions have been created for a **kafka.4u8g.cluster*3** instance, you can no longer scale down the instance to **kafka.2u4g.cluster*3** because this flavor allows only 750 partitions.

Process of Increasing or Decreasing Broker Flavors

When you scale up or down 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 or down the flavor of Broker 0.
3. Restart the Kafka process on Broker 0.
4. Repeat **1** to **3** to scale up or down the flavor of Broker 1.
5. Repeat **1** to **3** to scale up or down the flavor of Broker 2.

Figure 6-2 Process of increasing or decreasing broker flavors



Procedure

Step 1 Log in to the management 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 **Middleware** > **Distributed Message Service (for Kafka)** to open the console of DMS for Kafka.

Step 4 In the row containing the instance for which you want to modify the specifications, choose **More** > **Modify Specifications** in the **Operation** column.

Step 5 Specify the required storage space, broker quantity, broker flavor, or bandwidth.

To modify old specifications, perform the following steps:

- Increase the bandwidth.

  Specify a new bandwidth and click **Next**. Confirm the configurations and click **Submit**.

  View the new bandwidth of the instance in the **Specifications** column in the instance list.

  NOTE

  After increasing the bandwidth, add the IP address of the new broker to the client connection configuration to improve reliability.

- Expand the storage space.

  Specify a new storage space and click **Next**. Confirm the configurations and click **Submit**.

  View the new storage space in the **Used/Available Storage Space (GB)** column in the instance list.

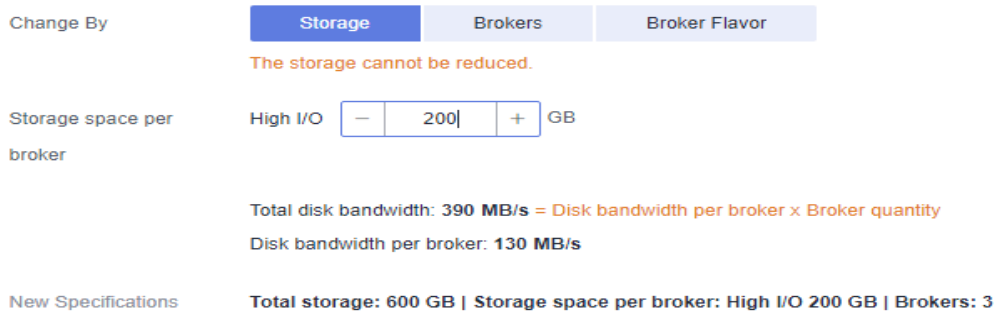
To modify new specifications, perform the following steps:

- Expand the storage space.

  For **Change By**, select **Storage**. For **Storage Space per Broker**, specify a new storage space, and click **Next**. Confirm the configurations and click **Submit**.

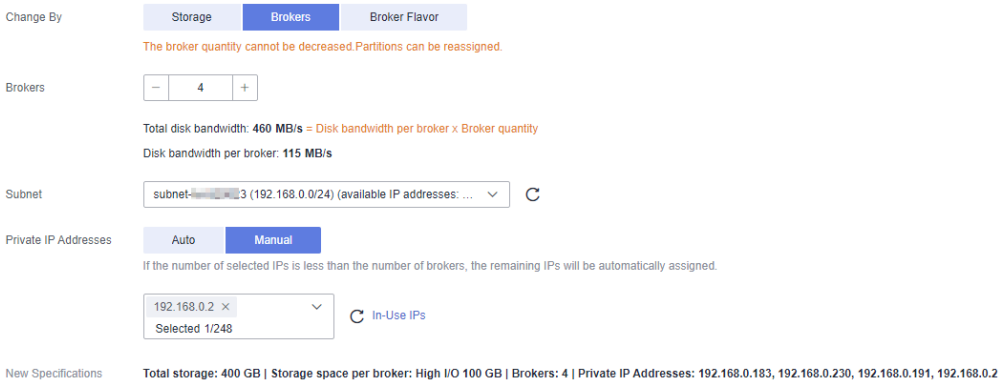
  View the new storage space (Storage space per broker x Number of brokers) in the **Used/Available Storage Space (GB)** column in the instance list.

Figure 6-3 Expanding the storage



- Add brokers.
  a. For **Change By**, select **Brokers**.
  b. For **Brokers**, specify the broker quantity.
  c. If public access is enabled, configure EIPs for the new brokers.
  d. For **Subnet**, retain the default settings.
  e. For **Private IP Addresses**, select **Auto** or **Manual**.
     - **Auto**: The system assigns an IP address from the subnet automatically.
     - **Manual**: Select the IP addresses for the new brokers from the drop-down list. If the number of selected IP addresses is less than the number of brokers, the remaining IP addresses will be automatically assigned.
  f. Click **Next**.
  g. Confirm the configurations and click **Submit**.
     View the number of brokers in the **Specifications** column in the instance list.

Figure 6-4 Adding brokers



NOTE

After adding brokers, add the IP addresses of the new brokers to the client connection configuration to improve reliability.

- Increase or decrease the broker flavor.

For **Change By**, select **Broker Flavor**. Then, select a new broker flavor and click **Next**. Confirm the configurations and click **Submit**.

View the broker flavor in the **Flavor** column in the instance list.

Figure 6-5 Increasing or decreasing the broker flavor

----End

6.2 Viewing an Instance

Scenario

View detailed information about a Kafka instance on the Kafka console, for example, the IP addresses and port numbers for accessing the instance.

Procedure

Step 1 Log in to the management 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 **Middleware** > **Distributed Message Service (for Kafka)** to open the console of DMS for Kafka.

Step 4 Search for a Kafka instance by specifying filters. You can filter instances by tag, status, name, version, flavor, used/available storage space, maximum partitions, billing mode, and enterprise project. For Kafka instance statuses, see **Table 6-3**.

Table 6-3 Kafka instance status description

Status	Description
Creating	The instance is being created.

Status	Description
Running	The instance is running properly. Only instances in the **Running** state can provide services.
Faulty	The instance is not running properly.
Starting	The status between **Frozen** and **Running**.
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.
Frozen	The instance is frozen.
Freezing	The status between **Running** and **Frozen**.
Upgrading	The instance is being upgraded.
Rolling back	The instance is being rolled back.

Step 5 Click the name of the desired Kafka instance and view detailed information about the instance on the **Basic Information** tab page.

Table 6-4 and **Table 6-5** describe 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 6-4 Connection parameters (in regions except CN North-Beijing4, CN East-Shanghai1, and CN South-Guangzhou)

Section	Parameter	Description
Connection	Username	Username for accessing the instance with SASL_SSL enabled.
	Kafka SASL_SSL	Whether SASL_SSL is enabled. This function is unavailable for single-node instances.
	Security Protocol	Security protocol used by the instance with SASL_SSL 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.

Section	Parameter	Description
	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 17, 2023 do not have this address.
	Manager Username	Username for connecting to Kafka Manager. Instances created since May 17, 2023 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 17, 2023 do not have this address.
	Intra-VPC Plaintext Access	Whether intra-VPC plaintext access is enabled.

Table 6-5 Connection parameters (in the CN North-Beijing4, CN East-Shanghai1, and CN South-Guangzhou)

Section	Parameter	Sub-Parameter	Description
Connection	Username	-	Username for accessing the instance with ciphertext access enabled.
	Private Network Access	Plaintext Access	Indicates whether plaintext access is enabled.
		Address (Private Network, Plaintext)	This parameter is displayed only after you enable **Plaintext Access**.

Section	Parameter	Sub-Parameter	Description
		Ciphertext Access	Indicates whether ciphertext access is enabled.
		Address (Private Network, Ciphertext)	This parameter is displayed only after you enable **Ciphertext Access**.
		Security Protocol	This parameter is displayed only after you enable **Ciphertext Access**.
	Public Network Access	Toggle switch	Indicates whether public access has been enabled.
		Plaintext Access	Indicates whether plaintext access is enabled.
		Address (Public Network, Plaintext)	This parameter is displayed only after you enable **Plaintext Access**.
		Ciphertext Access	Indicates whether ciphertext access is enabled.
		Address (Public Network, Ciphertext)	This parameter is displayed only after you enable **Ciphertext Access**.
		Security Protocol	This parameter is displayed only after you enable **Ciphertext Access**.
	SASL Mechanism	-	This parameter is displayed only after you enable **Ciphertext Access**.
	SSL Certificate	-	Click **Download** to download the SSL certificate for accessing the instance.

----End

6.3 Restarting an Instance

Scenario

Restart one or more Kafka instances at a time on the 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 management 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 **Middleware** > **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 view the instance details. 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

6.4 Deleting an Instance

Scenario

On the Kafka console, you can delete one or more Kafka instances that have been created.

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**, **Faulty**, or **Frozen**.
- Kafka instances in billed in the yearly/monthly mode cannot be deleted. To disable such a Kafka instance, choose **More** > **Unsubscribe** in the row containing the instance.

Deleting Kafka Instances

Step 1 Log in to the management 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 **Middleware** > **Distributed Message Service (for Kafka)** to open the console of DMS for Kafka.

Step 4 Delete Kafka instances using one of the following methods:

- 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 view its details. In the upper right corner, choose **More** > **Delete**.

NOTE

Kafka instances in the **Creating**, **Starting**, **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

6.5 Modifying the Information About an Instance

After creating a Kafka instance, you can modify some parameters of the instance, including the instance name, description, security group, and capacity threshold policy, based on service requirements.

Procedure

Step 1 Log in to the management 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 **Middleware** > **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 Modify the following parameters if needed:

- Instance Name
- Enterprise Project (Changing the enterprise project will not restart the instance.)
- Smart Connect
- Description
- Security Group
- Private Network Access (For details about how to modify it, see **Changing the Access Mode of an Instance**.)
- Public Network Access (For details about how to modify it, see **Configuring Public Access**.)
- Capacity Threshold Policy (Changing this setting will not restart the instance.)
- Automatic Topic Creation For details about how to enable or disable automatic topic creation, see **Enabling or Disabling Automatic Topic Creation**.
- Cross-VPC Access (See **Cross-VPC Access to a Kafka Instance** and **Using DNAT to Access a Kafka Instance**.)

After the parameters are modified, view the result in one of the following ways:

- If **Capacity Threshold Policy**, **Public Network Access**,, **Private Network Access**, or **Automatic Topic Creation** has been modified, you will be redirected to the **Background Tasks** page. The task progress and result are displayed.
- If **Instance Name**, **Description**, **Enterprise Project**, **Cross-VPC Access**, or **Security Group** has been modified, the result will be displayed in the upper right corner of the page.
- If **Smart Connect** has been modified, go to the **Background Tasks** page to view the task progress and result.

----End

6.6 Configuring 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.

> **NOTE**
>
> On the Kafka console, the procedures for enabling and disabling public access vary depending on the content displayed in the **Connection** area on the **Basic Information** page.
>
> In regions except CN North-Beijing4, CN East-Shanghai1, and CN South-Guangzhou, refer to **Disabling Public Network Access in Regions Except CN North-Beijing4, CN East-Shanghai1, and CN South-Guangzhou** and **Enabling Public Network Access in Regions Except CN North-Beijing4, CN East-Shanghai1, and CN South-Guangzhou**. In the CN North-Beijing4, CN East-Shanghai1, and CN South-Guangzhou regions, refer to **Enabling Public Network Access in the CN North-Beijing4, CN East-Shanghai1, and CN South-Guangzhou Regions** and **Disabling Public Network Access in the CN North-Beijing4, CN East-Shanghai1, and CN South-Guangzhou Regions**.

Prerequisites

- You can change the public access setting only when the Kafka instance is in the **Running** state.
- Kafka instances only support IPv4 EIPs. IPv6 EIPs are not supported.

Enabling Public Network Access in Regions Except CN North-Beijing4, CN East-Shanghai1, and CN South-Guangzhou

Step 1 Log in to the management console.

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

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

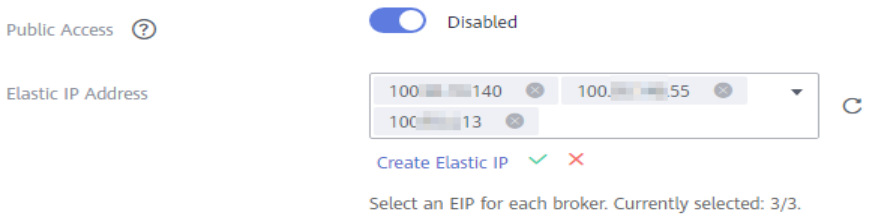
Step 3 Click and choose **Middleware** > **Distributed Message Service (for Kafka)** to open the console of DMS for Kafka.

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 and then click .

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

Figure 6-6 Enabling public access



After public access is enabled, configure security group rules listed in **Table 6-6** before attempting to access Kafka. For details about accessing Kafka, see **Accessing a Kafka Instance**.

Table 6-6 Security group rules (public network access)

Direction	Protocol	Port	Source	Description
Inbound	TCP	9094	0.0.0.0/0	Access Kafka through the public network (without SSL encryption).
Inbound	TCP	9095	0.0.0.0/0	Access Kafka through the public network (with SSL encryption).

----End

Disabling Public Network Access in Regions Except CN North-Beijing4, CN East-Shanghai1, and CN South-Guangzhou

Step 1 Log in to the management 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 **Middleware** > **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 6-7** before attempting to access Kafka in a VPC. For details about accessing Kafka, see **Accessing a Kafka Instance**.

Table 6-7 Security group rules (private network access)

Direction	Protocol	Port	Source	Description
Inbound	TCP	9092	0.0.0.0/0	Access a Kafka instance within a VPC (without SSL encryption).
Inbound	TCP	9093	0.0.0.0/0	Access a Kafka instance within a VPC (with SSL encryption).

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 6-7**.

----End

Enabling Public Network Access in the CN North-Beijing4, CN East-Shanghai1, and CN South-Guangzhou Regions

Step 1 Log in to the management console.

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

NOTE

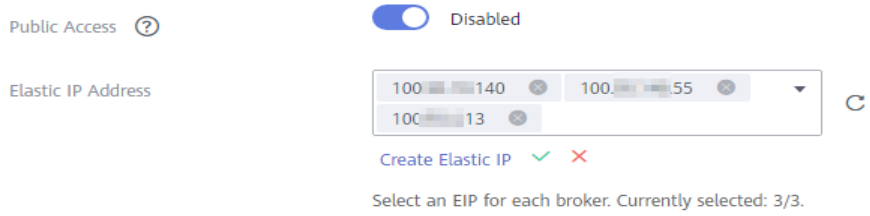
Select the region where your Kafka instance is located.

Step 3 Click and choose **Middleware** > **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 Network Access** to enable public access. For **Elastic IP Address**, select an EIP for each broker and then click to go to the **Background Tasks** page. If the status of the task turns to **Successful**, public access is successfully enabled.

Figure 6-7 Enabling public access



After public access is enabled, configure the **access mode (plaintext or ciphertext)** and security group rules listed in **Table 6-8** before attempting to access Kafka. For details about accessing Kafka, see **Accessing a Kafka Instance**.

Table 6-8 Security group rules

Direction	Protocol	Port	Source	Description
Inbound	TCP	9094	0.0.0.0/0	Access Kafka through the public network (without SSL encryption).

Direction	Protocol	Port	Source	Description
Inbound	TCP	9095	0.0.0.0/0	Access Kafka through the public network (with SSL encryption).

----End

Disabling Public Network Access in the CN North-Beijing4, CN East-Shanghai1, and CN South-Guangzhou Regions

Step 1 Log in to the management 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 **Middleware** > **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 Before disabling public access, disable **Plaintext Access** and **Ciphertext Access** next to **Public Network Access**. Then click next to **Public Network Access**. A confirmation dialog box is displayed.

Step 6 Click **OK**. The **Background Tasks** page is displayed. If the status of the task turns to **Successful**, public access is successfully disabled.

After public access is disabled, configure security group rules listed in **Table 6-9** before attempting to access Kafka in a VPC. For details about accessing Kafka, see **Accessing a Kafka Instance**.

Table 6-9 Security group rules (private network access)

Direction	Protocol	Port	Source	Description
Inbound	TCP	9092	0.0.0.0/0	Access a Kafka instance within a VPC (without SSL encryption).
Inbound	TCP	9093	0.0.0.0/0	Access a Kafka instance within a VPC (with SSL encryption).

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 6-9**.

----End

6.7 Changing the Access Mode of an Instance

You can access a Kafka instance in plaintext or ciphertext. This section describes how to change the access mode on the console.

NOTE

- When you change the access mode for the first time, some instances will restart. You can see the actual situation on the console. The restart takes about 75–80s. The instance will not be restarted when the access mode is changed again.
- You can change the instance access mode only in the CN North-Beijing4, CN East-Shanghai1, and CN South-Guangzhou regions.
- For a single-node instance, you can only enable or disable plaintext for public network access.

Prerequisites

You can change the access mode of a Kafka instance only when the instance is in the **Running** state.

Enabling Plaintext Access

Step 1 Log in to the management 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 **Middleware** > **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 An instance can be accessed in plaintext over the private network and public network. For details about how to enable plaintext access, see **Table 6-10**.

Table 6-10 Enabling plaintext access

Access Method	Enabling Plaintext Access
Private network plaintext access	1. Click next to **Plaintext Access** in the **Private Network Access** area. A confirmation dialog box is displayed. 2. Click **OK**. The **Background Tasks** page is displayed. If the status of the task turns to **Successful**, plaintext access is successfully enabled.
Public network plaintext access	1. Check that **Public Access** is enabled. If it is not enabled, enable it. For details, see **Configuring Public Access**. 2. Click next to **Plaintext Access** in the **Public Network Access** area. A confirmation dialog box is displayed. 3. Click **OK**. The **Background Tasks** page is displayed. If the status of the task turns to **Successful**, plaintext access is successfully enabled.

----End

Enabling Ciphertext Access

Step 1 Log in to the management 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 **Middleware** > **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 An instance can be accessed in ciphertext over the private network and public network. For details about how to enable ciphertext access, see **Table 6-11**.

Table 6-11 Enabling ciphertext access

Access Method	Enabling Ciphertext Access
Private network ciphertext access	1. Click  next to **Ciphertext Access** in the **Private Network Access** area. The **Private Network Ciphertext Access** dialog box is displayed. 2. Set the Kafka security protocol, SASL/PLAIN mechanism, username, and password, and click **OK**. The **Background Tasks** page is displayed. If the status of the task turns to **Successful**, ciphertext access is successfully enabled. **NOTE** – When enabling ciphertext access for the first time (including through private network and public network), you need to set the Kafka security protocol, SASL/PLAIN mechanism, username, and password. Next time when you enable ciphertext access, you only need to set the Kafka security protocol. – To disable private network ciphertext access, contact customer service.
Public network ciphertext access	1. Check that **Public Access** is enabled. If it is not enabled, enable it. For details, see **Configuring Public Access**. 2. Click  next to **Ciphertext Access** in the **Public Network Access** area. The **Public Network Ciphertext Access** dialog box is displayed. 3. Set the Kafka security protocol, SASL/PLAIN mechanism, username, and password, and click **OK**. The **Background Tasks** page is displayed. If the status of the task turns to **Successful**, ciphertext access is successfully enabled. **NOTE** When enabling ciphertext access for the first time (including through private network and public network), you need to set the Kafka security protocol, SASL/PLAIN mechanism, username, and password. Next time when you enable ciphertext access, you only need to set the Kafka security protocol.

The Kafka security protocol, SASL/PLAIN mechanism, username, and password are described as follows.

Table 6-12 Ciphertext access parameters

Parameter	Value	Description
Security Protocol	SASL_SSL	SASL is used for authentication. Data is encrypted with SSL certificates for high-security transmission. This protocol supports the SCRAM-SHA-512 and PLAIN mechanisms. **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.
	SASL_PLAINTEXT	SASL is used for authentication. Data is transmitted in plaintext for high performance. This protocol supports the SCRAM-SHA-512 and PLAIN mechanisms. SCRAM-SHA-512 authentication is recommended for plaintext transmission.
SASL/PLAIN	-	• If **SASL/PLAIN** is disabled, the SCRAM-SHA-512 mechanism is used for username and password authentication. • 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 ciphertext access is enabled.
Username and Password	-	Username and password used by the client to connect to the Kafka instance. The username cannot be changed once ciphertext access is enabled.

----End

Disabling Plaintext Access

Step 1 Log in to the management 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 **Middleware** > **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 An instance can be accessed in plaintext over the private network and public network. For details about how to disable plaintext access, see **Table 6-13**.

Table 6-13 Disabling plaintext access

Access Method	Disabling Plaintext Access
Private network plaintext access	Once enabled, private network access cannot be disabled. Enable plaintext or ciphertext access, or both. If ciphertext access is disabled, plaintext access cannot be disabled. 1. Click next to **Plaintext Access** in the **Private Network Access** area. A confirmation dialog box is displayed. 2. Click **OK**. The **Background Tasks** page is displayed. If the status of the task turns to **Successful**, plaintext access is successfully disabled.
Public network plaintext access	1. Click next to **Plaintext Access** in the **Public Network Access** area. A confirmation dialog box is displayed. 2. Click **OK**. The **Background Tasks** page is displayed. If the status of the task turns to **Successful**, plaintext access is successfully disabled.

----End

Disabling Ciphertext Access

Step 1 Log in to the management 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 **Middleware** > **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 An instance can be accessed in ciphertext over the private network and public network. For details about how to disable ciphertext access, see **Table 6-14**.

Table 6-14 Disabling ciphertext access

Access Method	Disabling Plaintext Access
Private network ciphertext access	To disable private network ciphertext access, contact customer service.
Public network ciphertext access	1. Click  next to **Ciphertext Access** in the **Public Network Access** area. A confirmation dialog box is displayed. 2. Click **OK**. The **Background Tasks** page is displayed. If the status of the task turns to **Successful**, ciphertext access is successfully disabled.

NOTE

After you disable ciphertext access, the created users will not be deleted. You do not need to create users again when you enable ciphertext access next time.

----End

6.8 Changing the Billing Mode from Pay-per-Use to Yearly/Monthly

Instances billed in the pay-per-use mode can be changed to the yearly/monthly billing mode.

NOTE

Changing the billing mode does not affect your applications.

This function is unavailable for v3.x instances during OBT.

Procedure

Step 1 Log in to the management 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 **Middleware** > **Distributed Message Service (for Kafka)** to open the console of DMS for Kafka.

Step 4 Use one of the following methods to change the billing mode from pay-per-use to yearly/monthly:

- Select one or more Kafka instances and click **Change to Yearly/Monthly Billing** in the upper left corner of the instance list. In the displayed **Change to Yearly/Monthly** dialog box, click **Yes**.

- In the row that contains the target Kafka instance, choose **More** > **Change to Yearly/Monthly Billing**.
- Click the desired Kafka instance to view the instance details. Click **More** > **Change to Yearly/Monthly Billing** in the upper right corner to go to the **Change Subscription** page.

Step 5 Select a renewal duration and click **Pay**. Make the payment as prompted.

----End

6.9 Resetting Kafka Password

Scenario

For a Kafka instance with SASL_SSL enabled, there are two ways to create an SASL_SSL user on the console. Accordingly, there are two ways to reset the SASL_SSL user's password:

- If an SASL_SSL user is created during instance creation, reset their password by referring to the following instructions.
- If an SASL_SSL user is created on the **Users** page, reset their password by referring to **Resetting the SASL_SSL Password**.

NOTE

This function is not available for single-node instances.

Prerequisites

- You can reset the Kafka password only if Kafka SASL_SSL has been enabled for the instance.
- You can reset the Kafka password only when the instance is in the **Running** state.

Procedure

Step 1 Log in to the management 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 **Middleware** > **Distributed Message Service (for Kafka)** to open the console of DMS for Kafka.

Step 4 Reset the Kafka instance password using either of the following methods:

- Choose **More** > **Reset Kafka Password** in the row containing the desired Kafka instance.
- Click the desired Kafka instance to view its details. Choose **More** > **Reset Kafka Password** in the upper left corner.

- Click the desired Kafka instance to view its details. On the **Basic Information** page, click **Reset Password** next to **Username** in the **Connection** section.
- Click the desired Kafka instance to view its details. On the **Users** page, click **Reset Password** in the row containing the desired user.

Step 5 In the **Reset Kafka Password** dialog box, 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

6.10 Resetting Kafka Manager Password

Scenario

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 17, 2023.

Prerequisites

A Kafka instance has been created and is in the **Running** state.

Procedure

Step 1 Log in to the management 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 **Middleware** > **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 view its details. In the upper right corner, choose **More** > **Reset Manager Password**.
- Click the desired Kafka instance to view its details. 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.

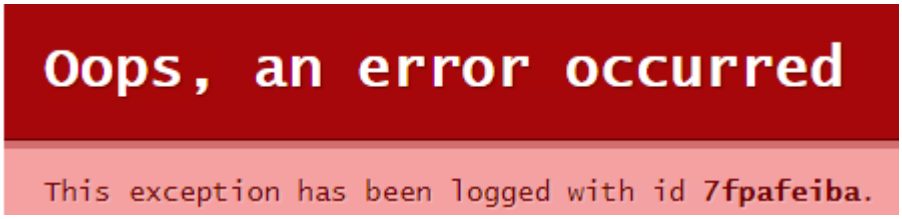
----End

6.11 Restarting Kafka Manager

Scenario

Restart Kafka Manager when you fail to log in to it or it cannot provide services as usual.

Figure 6-8 Error information



NOTE

Restarting Kafka Manager does not affect services.

This function is not available for instances created since May 17, 2023.

Procedure

Step 1 Log in to the management 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 **Middleware** > **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 view the instance details. 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

6.12 Disabling Kafka Manager

Scenario

If you no longer need Kafka Manager, disable it 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 management console.

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

NOTE

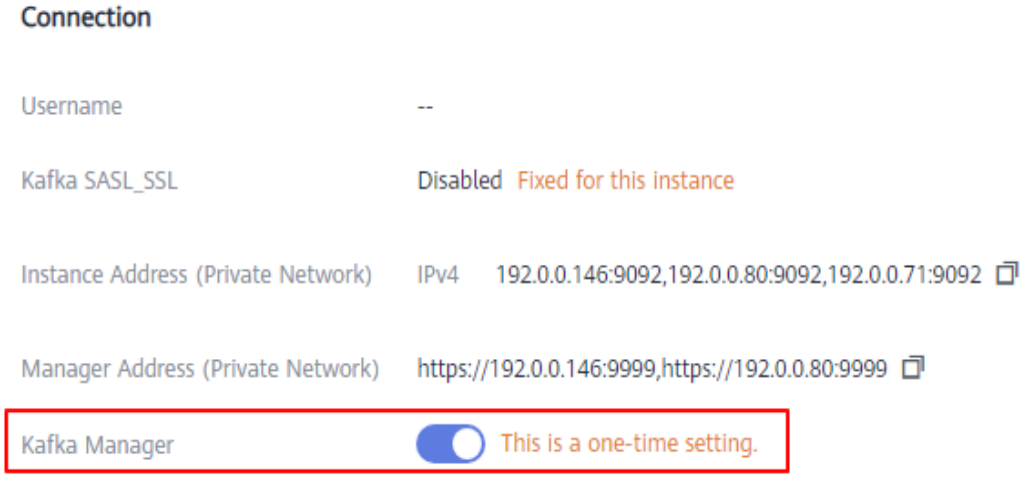
Select the region where your Kafka instance is.

Step 3 Click and choose **Middleware** > **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** section.

Figure 6-9 Disabling Kafka Manager



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

6.13 Managing 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.

If your organization has configured tag policies for DMS for Kafka, add tags to Kafka instances based on the tag policies. If a tag added on the **Tags** page does not comply with the tag policies, the tag fails to be added.

A tag consists of a tag key and a tag value. **Table 6-15** lists the tag key and value requirements.

Table 6-15 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.
Tag value	• Can contain 0 to 255 characters. • Can contain letters, digits, spaces, and special characters _.:=+-@ : = + - @ • Cannot start or end with a space.

Procedure

Step 1 Log in to the management 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 **Middleware** > **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

6.14 Viewing Background Tasks

After you initiate certain instance operations such as configuring public access and modifying the capacity threshold policy, 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.

Procedure

Step 1 Log in to the management 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 **Middleware** > **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 In the upper right corner, select **Background tasks**, click the time period next to the calendar icon, select the start time and end time, and click **OK**. Tasks started in the specified period are 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 only delete the records of tasks in the **Successful** or **Failed** state.

----End

6.15 Viewing Disk Usage

On the Kafka console, you can view the disk usage of each broker.

NOTE

This function is unavailable for single-node instances.

Procedure

Step 1 Log in to the management console.

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

NOTE

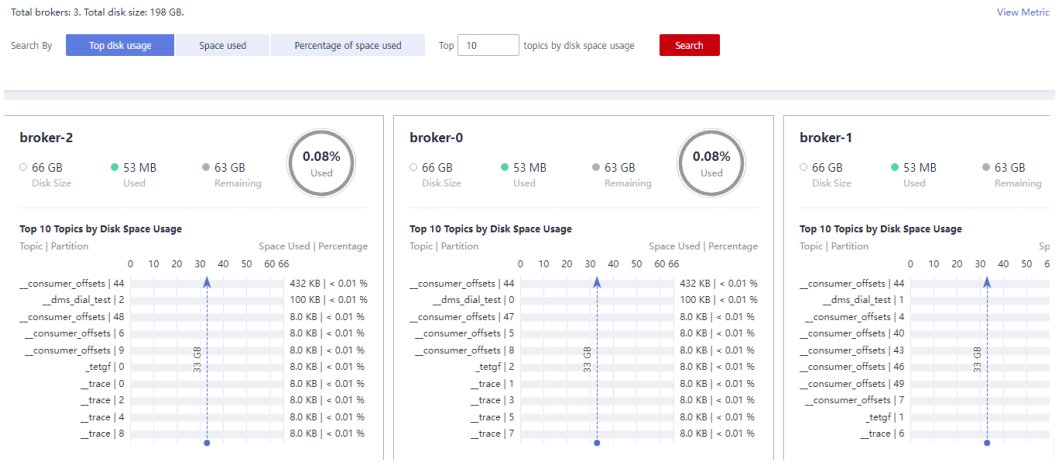
Select the region where your Kafka instance is located.

Step 3 Click  and choose **Middleware** > **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.

Figure 6-10 Viewing disk usage



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.

----End

6.16 Exporting the Instance List

Scenario

Export the Kafka instance list from the console.

Procedure

Step 1 Log in to the management 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 **Middleware** > **Distributed Message Service (for Kafka)** to open the console of DMS for Kafka.

Step 4 Click **Export** and choose to export all data or selected data to an XLSX file to export the instance list.

----End

7 Managing Topics

7.1 Creating a Topic

A topic is a stream of messages. If automatic topic creation is not enabled during Kafka instance creation, you need to manually create topics for creating and retrieving messages. If automatic topic creation has been enabled for the instance, this operation is optional.

If automatic topic creation is enabled, the system automatically creates a topic when a message is created in or retrieved from a topic that does not exist. This topic has the following default settings: 3 partitions, 3 replicas, aging time of 72 hours, and synchronous replication and flushing disabled. After you change the value of the **log.retention.hours**, **default.replication.factor**, or **num.partitions** parameter, automatically created topics later use the new value. For example, if **num.partitions** is set to **5**, an automatically created topic will have the following settings: 5 partitions, 3 replicas, aging time 72 hours, and synchronous replication and flushing disabled.

There is a limit on the total number of partitions in topics. **When the partition quantity limit is reached, you can no longer create topics.** The total number of partitions varies with specifications. For details, see **Specifications**.

Methods that can be used to manually create a topic:

- **Method 1: Creating a Topic on the Console**
- **Method 2: Creating a Topic on Kafka Manager**
- **Method 3: Creating a Topic by Using Kafka CLI**

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.

Instances created since May 17, 2023 do not have Kafka Manager. You cannot create topics for these instances using Kafka Manager.

Method 1: Creating a Topic on the Console

Step 1 Log in to the management console.

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

NOTE

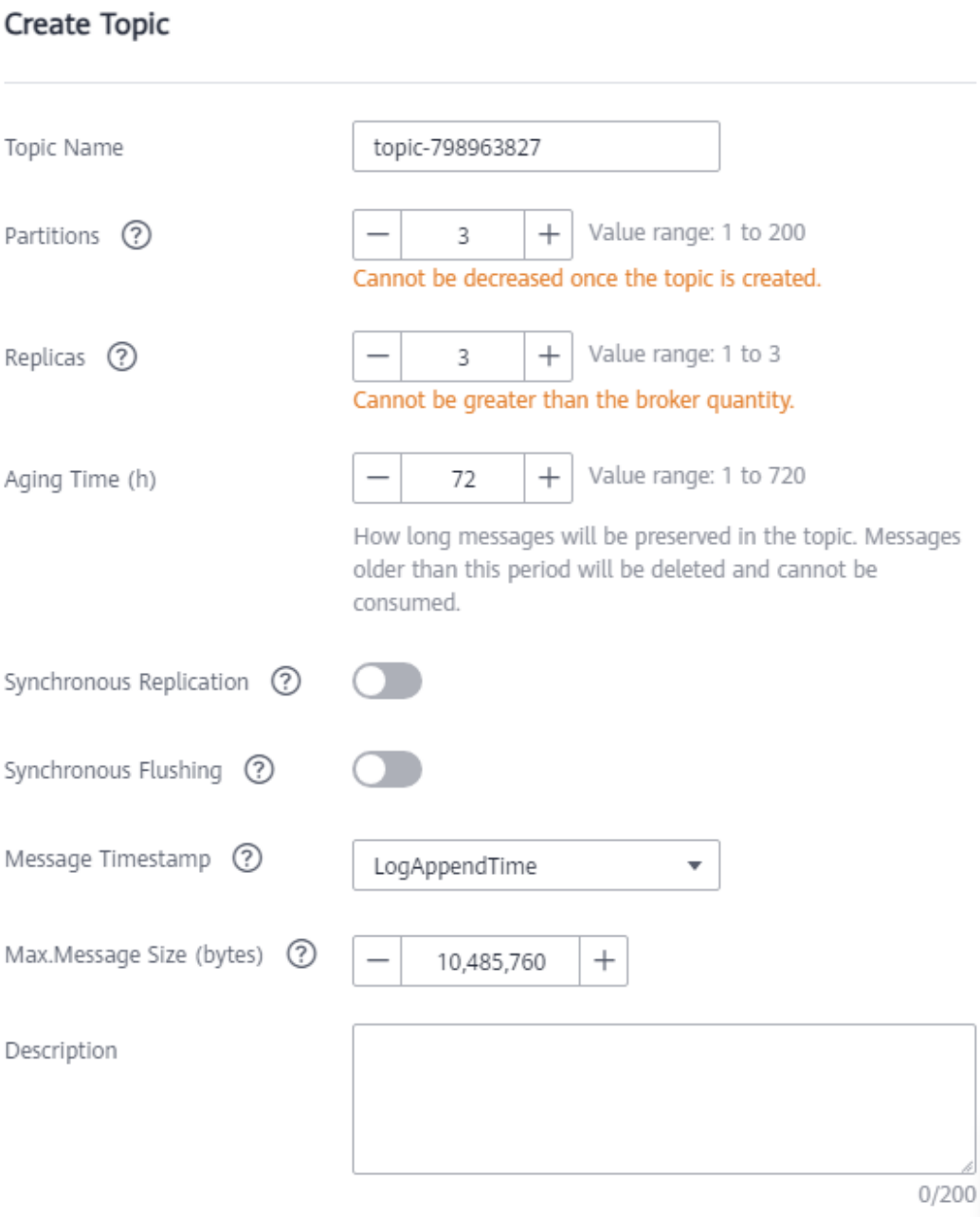
Select the region where your Kafka instance is located.

Step 3 Click  and choose **Middleware** > **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**.

Figure 7-1 Creating a topic



Step 6 Specify the topic parameters listed in the following table.

Table 7-1 Topic parameters

Parameter	Description
Topic Name	When creating a topic, you can modify the automatically generated topic name. Once the topic is created, you cannot modify its name.

Parameter	Description
Partitions	A larger number of partitions for a topic indicates more messages retrieved concurrently. If this parameter is set to **1**, messages will be retrieved in the FIFO order. Value range: 1 to 200 Default value: **3**
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. Default value: **3** **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 retrieved. Value range: 1 to 720 Default value: **72**
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 synchronous replication, set **acks** to **all** or **–1** on the client. Otherwise, this function will not take effect. If there is only one replica, synchronous replication cannot be enabled.
Synchronous Flushing	An indicator of whether a message is immediately flushed to disk once created. • Enabled: A message is immediately flushed to disk once it is created, resulting in 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 batch processing size allowed by Kafka. If message compression is enabled, 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.
Description	Topic description.

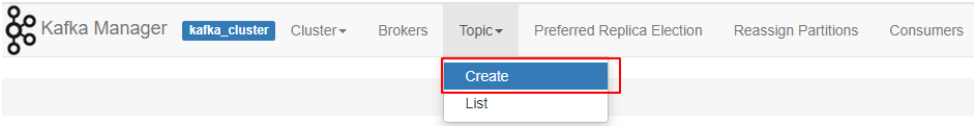
Step 7 Click **OK**.

----End

Method 2: Creating a Topic on Kafka Manager

Log in to Kafka Manager, choose **Topic** > **Create**, and set parameters as prompted.

Figure 7-2 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.

Method 3: Creating a Topic by Using Kafka CLI

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.

- If SASL is not enabled for the Kafka instance, run the following command in the */{directory where the CLI is located}*/**kafka_{version}/bin/** directory to create a topic:

  ```
  ./kafka-topics.sh --create --topic {topic_name} --bootstrap-server {broker_ip}:{port} --partitions {partition_num} --replication-factor {replication_num}
  ```

- If SASL has been enabled for the Kafka instance, perform the following steps to create a topic:

a. (Optional) If the SSL certificate configuration has been set, skip this step. Otherwise, perform the following operations:

   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 */{directory where the CLI is located}/**kafka_{version}/bin/*** directory to create a topic:

```
./kafka-topics.sh --create --topic {topic_name} --bootstrap-server {broker_ip}:{port} --partitions {partition_num} --replication-factor {replication_num} --command-config ./config/ssl-user-config.properties
```

7.2 Deleting a Topic

Delete a topic using either of the following methods:

- **By using the console**
- **By using Kafka CLI**

Prerequisites

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

Deleting a Topic on the Console

Step 1 Log in to the management 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 **Middleware** > **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 **Yes** to delete the topic.

----End

Deleting a Topic with the Kafka CLI

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.

- If SASL is not enabled for the Kafka instance, run the following command in the */{directory where the CLI is located}/***kafka_{version}/bin/** directory to delete a topic:
  ```
  ./kafka-topics.sh --bootstrap-server {broker_ip}:{port} --delete --topic {topic_name}
  ```
- If SASL has been enabled for the Kafka instance, perform the following steps to delete a topic:
  a. (Optional) If the SSL certificate configuration has been set, skip this step. Otherwise, perform the following operations:

     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 */{directory where the CLI is located}/***kafka_{version}/bin/** directory to delete a topic:
     ```
     ./kafka-topics.sh --bootstrap-server {broker_ip}:{port} --delete --topic {topic_name} --command-config ./config/ssl-user-config.properties
     ```

7.3 Modifying Topic Aging Time

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

After creating a topic, you can change its aging time based on service requirements. Changing the aging time does not affect services. The default aging time is 72 hours.

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 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 management 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 **Middleware** > **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

7.4 Changing Partition Quantity

After creating a topic, you can increase the number of partitions based on service requirements.

NOTE

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

Methods for changing the partition quantity:

- **Method 1: By Using the Console**
- **Method 2: By Using Kafka Manager**
- **Method 3: By using Kafka CLI**

Instances created since May 17, 2023 do not have Kafka Manager. You cannot modify topic partitions for these instances using Kafka Manager.

Method 1: By Using the Console

Step 1 Log in to the management 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 **Middleware** > **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 number of partitions of all topics cannot exceed the maximum number of partitions allowed by the instance.

----End

Method 2: By Using Kafka Manager

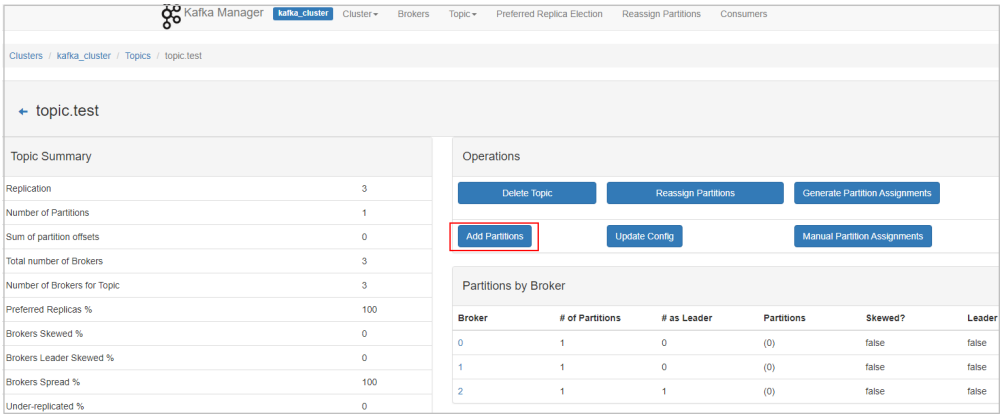
Step 1 Log in to Kafka Manager.

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

Step 3 Click a topic to view its details.

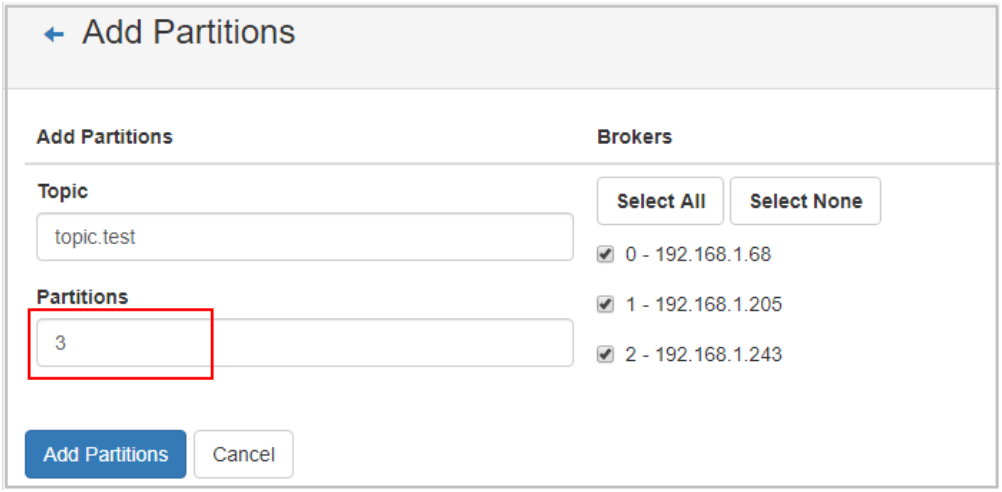
Step 4 Click **Add Partitions**.

Figure 7-3 Topic details page



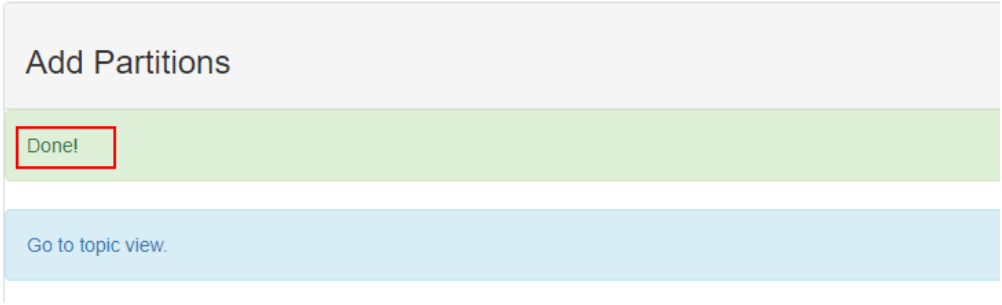
Step 5 Enter the number of partitions and click **Add Partitions**.

Figure 7-4 Adding partitions



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

Figure 7-5 Partitions added



NOTE

- The number of partitions can only be increased.
- The total number of partitions of all topics cannot exceed the maximum number of partitions allowed by the instance.

----End

Method 3: By Using Kafka CLI

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.

- If SASL is not enabled for the Kafka instance, run the following command in the **/*{directory where the CLI is located}*/kafka_{version}/bin/** directory to change the partition quantity:

  ```
  ./kafka-topics.sh --bootstrap-server {broker_ip}:{port} --topic {topic_name} --alter --partitions {partition_num}
  ```

- If SASL has been enabled for the Kafka instance, perform the following steps to change the partition quantity:

  a. (Optional) If the SSL certificate configuration has been set, skip this step. Otherwise, perform the following operations:

  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 **/*{directory where the CLI is located}*/kafka_{version}/bin/** directory to change the partition quantity:

  ```
  ./kafka-topics.sh --bootstrap-server {broker_ip}:{port} --topic {topic_name} --alter --partitions {partition_num} --command-config ./config/ssl-user-config.properties
  ```


7.5 Modifying Synchronous Replication and Flushing Settings

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: A message is immediately flushed to disk once created.

- Enabled: A message is immediately flushed to disk once it is created, resulting in higher reliability.
- Disabled: A message is stored in the memory instead of being immediately flushed to disk once created.

The following procedure describes how to modify synchronous replication and synchronous flushing settings on the console.

NOTE

Modifying synchronous replication and flushing settings will not restart the instance.

Procedure

Step 1 Log in to the management 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 **Middleware** > **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 Use either of the following methods to modify synchronous replication and synchronous flushing settings:

- Select one or more topics and click **Edit Topic** above the topic list.
- In the row that contains the topic whose synchronous replication and flushing settings are to be modified, click **Edit**.

Step 7 In the **Edit Topic** dialog box, enable or disable synchronous replication and synchronous flushing, and click **OK**.

- To enable them, click .
- To disable them, click .

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

7.6 Modifying Message Timestamp, Max. Message Size, and Description

Modify the message timestamp type, maximum message size, and description on the console.

- **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, 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 increased size.
- **Description**: description of the topic.

NOTE

You do not need to restart the instance after modifying the **Message Timestamp** parameter.

Procedure

Step 1 Log in to the management 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 **Middleware** > **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 Use one of the following methods to modify **Message Timestamp**, **Max. Message Size**, and **Description**:

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

----End

7.7 Reassigning Partitions

Scenario

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 the broker quantity is increased for an instance, the replicas of the original topic partitions are migrated to the new brokers.
- The leader partition is degraded to be a follower on a heavily loaded broker.
- The number of replicas is increased or decreased.

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

NOTE

This function is unavailable for single-node instances.

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%.
- 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.
- 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 a scheduled partition reassignment task has been configured, no other reassignment can be executed until this existing task is executed.
- 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 aging time based on the topic consumption

so that historical data of the topic can be deleted in a timely manner to accelerate the migration.

Preparing for Partition Reassignment

- To reduce the amount of data to be migrated, decrease the topic aging time without affecting services and wait for messages to age. After the reassignment is complete, you can restore the aging time.
- Ensure that the target broker has sufficient disk capacity. 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 management 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 **Middleware** > **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 **Topics** tab.

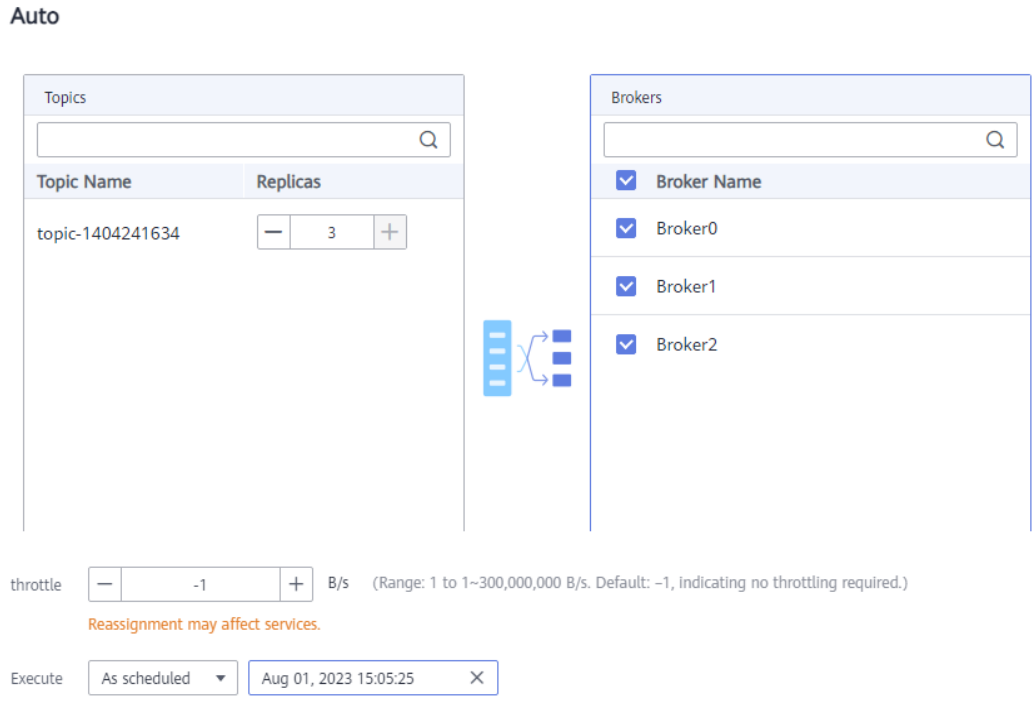
Step 6 Reassign partitions using either of the following methods:

- 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.

- In the **Brokers** area, select the brokers to assign the topic's partition replicas to.
- In the **Topics** area, enter the number of replicas to be automatically reassigned. The number of replicas must be less than or equal to the number of brokers.
- Specify **throttle**. The default value is **-1**, indicating that there is no throttle (recommended if the instance load is light). If a throttle is required, 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 7-6 Setting automatic reassignment parameters



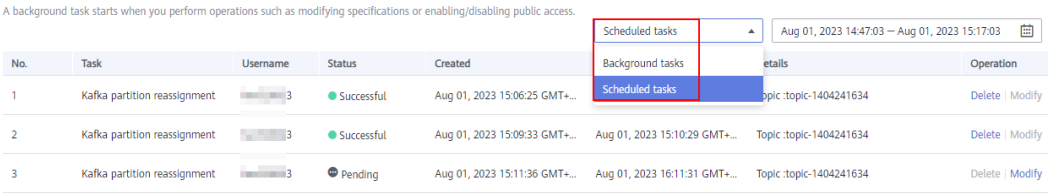
Step 8 (Optional) Click **Calculate**. **Time Required** indicates how long automatic balancing will take.

Step 9 Click **OK**. The topic list is displayed.

In the upper left corner of the topic list, click **View details** to view the reassignment task status on the **Background Tasks** page that is displayed.

- For a non-scheduled reassignment task, select **Background tasks** in the upper right corner of the **Background Tasks** page to view the task status. When the task status is **Successful**, reassignment has completed.
- For a scheduled reassignment task, select **Scheduled tasks** in the upper right corner of the **Background Tasks** page to view the task status. When the task status is **Pending**, reassignment has not been executed. When the task status is **Successful**, reassignment has completed. Select **Background tasks** in the upper right corner of the page to view the task status. When the task status is **Successful**, reassignment has completed.

Figure 7-7 Background Tasks page



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 a scheduled partition reassignment task has been configured, no other reassignment can be executed until this existing task is executed.

----End

Manual Reassignment

Step 1 Log in to the management 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 **Middleware** > **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 **Topics** tab.

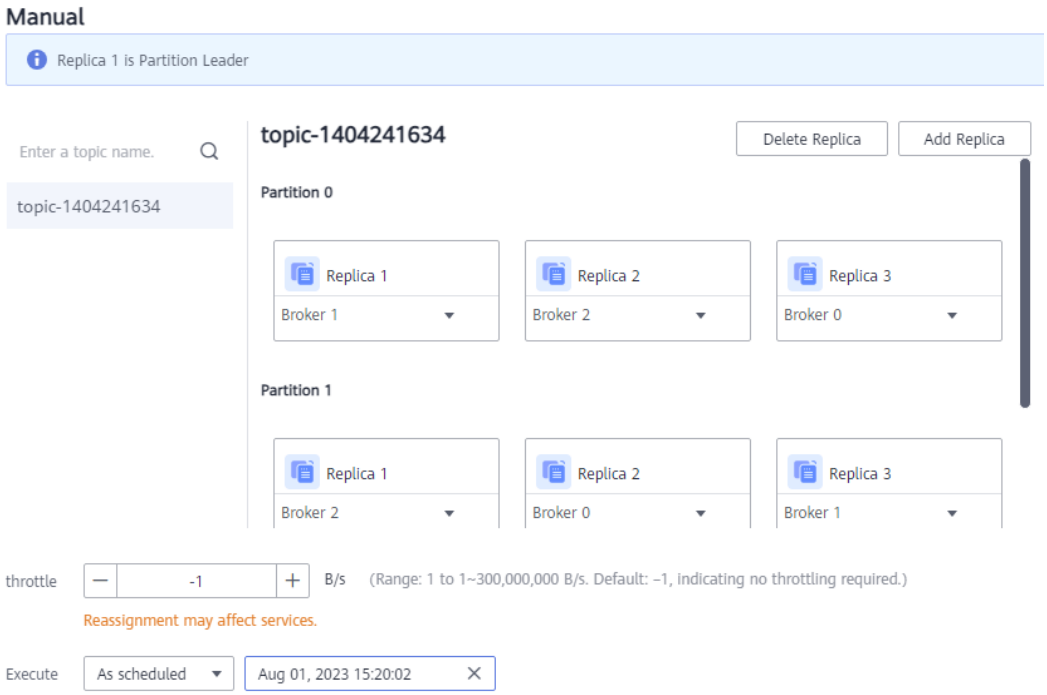
Step 6 Reassign partitions using either of the following methods:

- 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 (recommended if the instance load is light). If a throttle is required, 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 7-8 Setting manual reassignment parameters



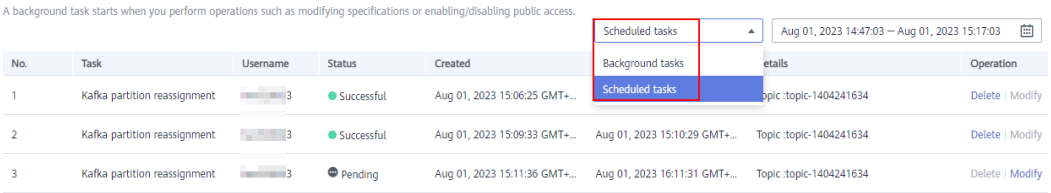
Step 8 (Optional) Click **Calculate**. **Time Required** indicates how long manual balancing will take.

Step 9 Click **OK**. The topic list is displayed.

In the upper left corner of the topic list, click **View details** to view the reassignment task status on the **Background Tasks** page that is displayed.

- For a non-scheduled reassignment task, select **Background tasks** in the upper right corner of the **Background Tasks** page to view the task status. When the task status is **Successful**, reassignment has completed.
- For a scheduled reassignment task, select **Scheduled tasks** in the upper right corner of the **Background Tasks** page to view the task status. When the task status is **Pending**, reassignment has not been executed. When the task status is **Successful**, reassignment has completed. Select **Background tasks** in the upper right corner of the page to view the task status. When the task status is **Successful**, reassignment has completed.

Figure 7-9 Background Tasks page



NOTE

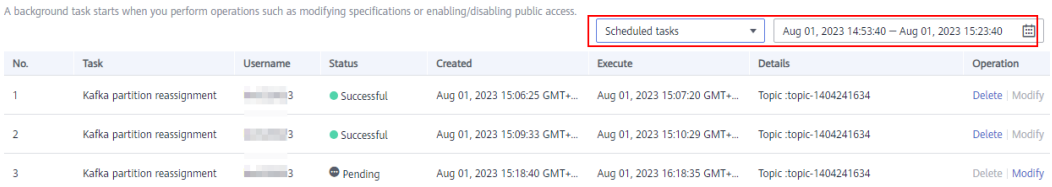
- You cannot delete topics whose reassignment tasks have started. Otherwise, the tasks will never complete.
- You cannot modify the partition quantity of topics whose reassignment tasks have started.
- Reassignment tasks cannot be manually stopped. Please wait until they complete.
- If a scheduled partition reassignment task has been configured, no other reassignment can be executed until this existing task is executed.

----End

Re-scheduling Partition Reassignment

Step 1 In the upper right corner of the **Background Tasks** page, select **Scheduled tasks** and a time segment to search for the desired task.

Figure 7-10 Selecting Scheduled tasks and a time segment

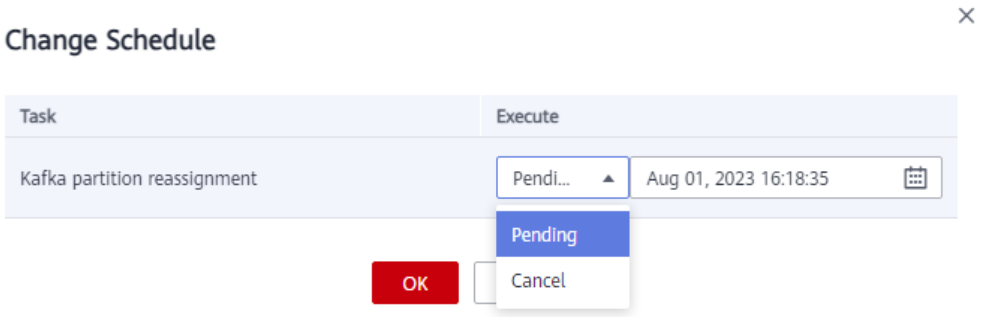


Step 2 In the row that contains the desired task, click **Modify**.

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 7-11**) and click **OK**.

Figure 7-11 Canceling a scheduled reassignment task



----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. One replica needs to be added for Topic 1 and Topic 2 respectively.

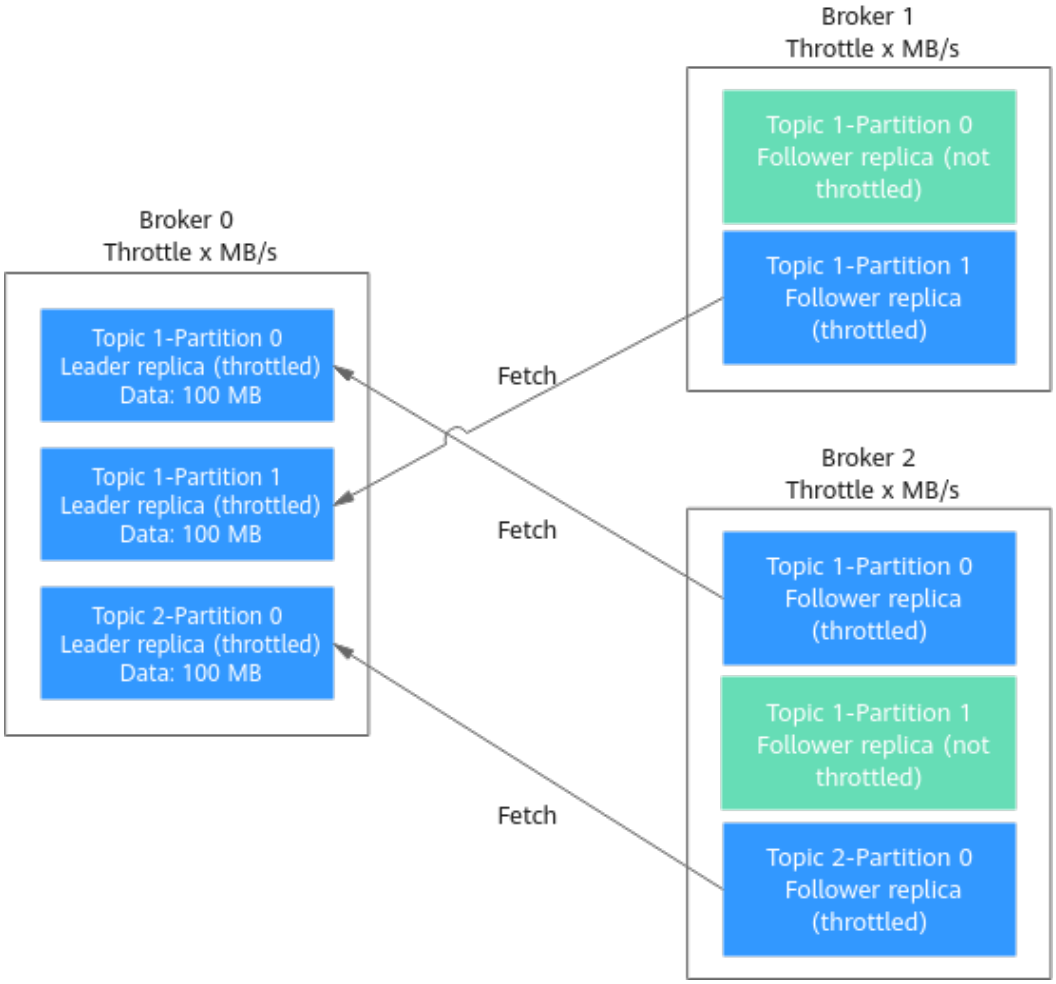
Table 7-2 Replica distribution before reassignment

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 7-3 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 7-12 Reassignment scenario 1



As shown in **Figure 7-12**, 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.

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. One replica needs to be added for Topic 1 and Topic 2 respectively.

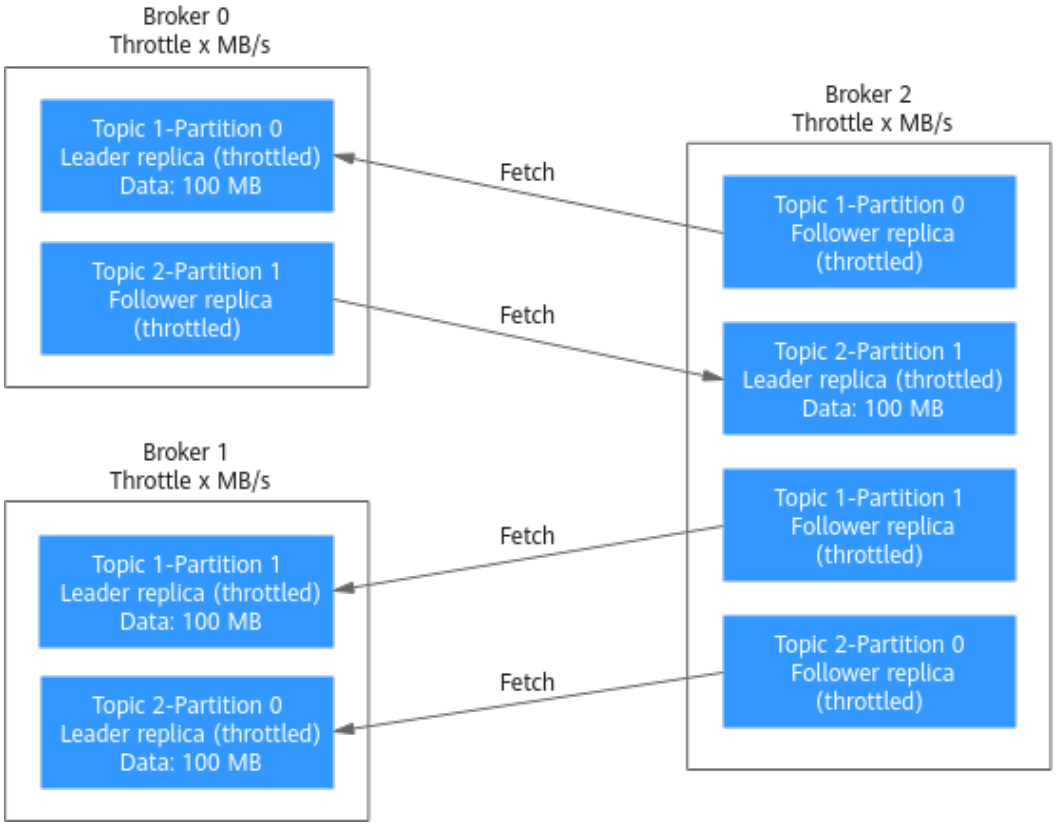
Table 7-4 Replica distribution before reassignment

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 7-5 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 7-13 Reassignment scenario 2



As shown in **Figure 7-13**, 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.

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. Messages are produced on Topic 1, causing replication.

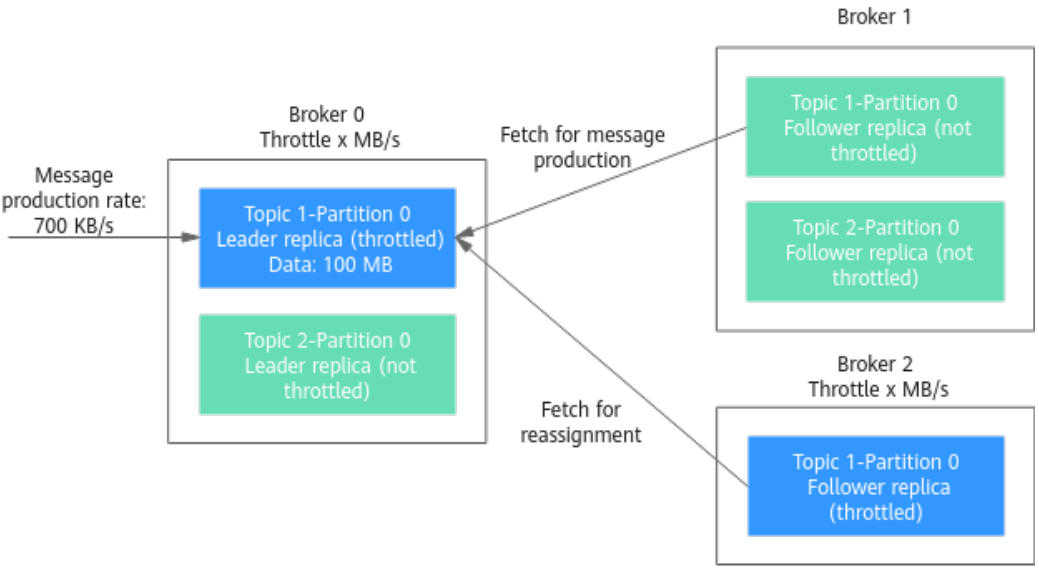
Table 7-6 Replica distribution before reassignment

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 7-7 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 7-14 Reassignment scenario 3



As shown in **Figure 7-14**, 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.

7.8 Viewing Sample Code

On the console, view sample code for creating and retrieving messages in Java, Go, and Python.

Procedure

Step 1 Log in to the management 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 **Middleware** > **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

7.9 Exporting the 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 management 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 **Middleware** > **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, description, and whether synchronous replication and flushing are enabled.

----End

7.10 Configuring Topic Permissions

DMS for Kafka supports ACL permission management for topics. You can differentiate the operations that different users are allowed to perform on a topic by granting the users different permissions.

This section describes how to grant topic permissions to a SASL_SSL user. For details about how to create a SASL_SSL user, see **Creating a SASL_SSL User**.

NOTE

This function is unavailable for single-node instances.

Constraints

- If no SASL_SSL user is granted any permission for a topic and **allow.everyone.if.no.acl.found** is set to **true**, all users can subscribe to or publish messages to the topic.
- If **allow.everyone.if.no.acl.found** is set to **true**, 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 one or more SASL_SSL users are granted permissions for a topic, only the authorized users can subscribe to or publish messages to the topic.
- 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.
- (Optional) A SASL_SSL user has been created. For details, see **Creating a SASL_SSL User**.

Configuring Topic Permissions

Step 1 Log in to the management 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 **Middleware** > **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**.

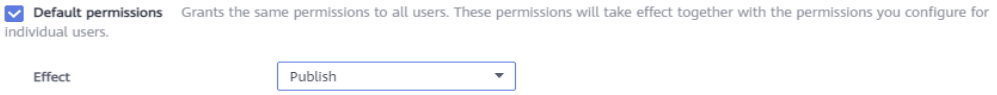
Step 6 In the row that contains the topic for which you want to configure user permissions, click **Grant User Permission**.

In the upper part of the **Grant User Permission** dialog box, the topic information is displayed, including the topic name, number of partitions, aging time, number of replicas, and whether synchronous flushing and replication are enabled. You can enable **Default permissions** to grant the same permissions for all users. You can use the search box to search for a user if there are many SASL_SSL users. In the **Users** area, the list of created SASL_SSL users is displayed. In the **Selected** area, you can grant permissions to the selected SASL_SSL users.

Step 7 Grant topic permissions to users.

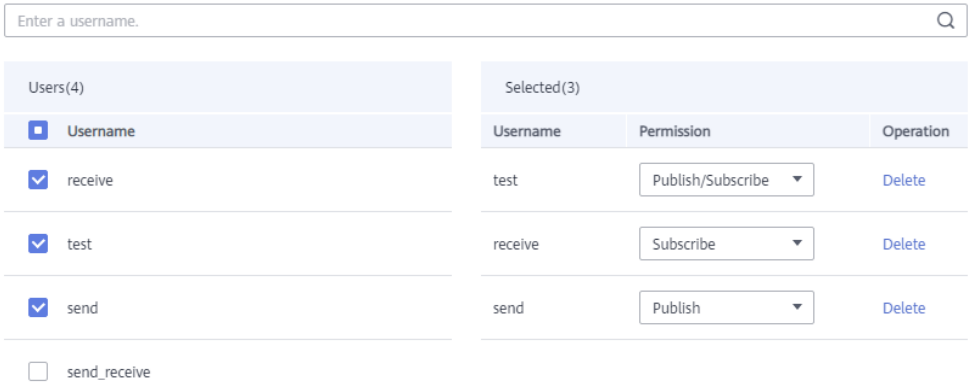
- To grant the same permissions to all users, select **Default permissions** and then select permissions. As shown in the following figure, all users have the permission to publish messages to this topic.

Figure 7-15 Granting the same rights to all users



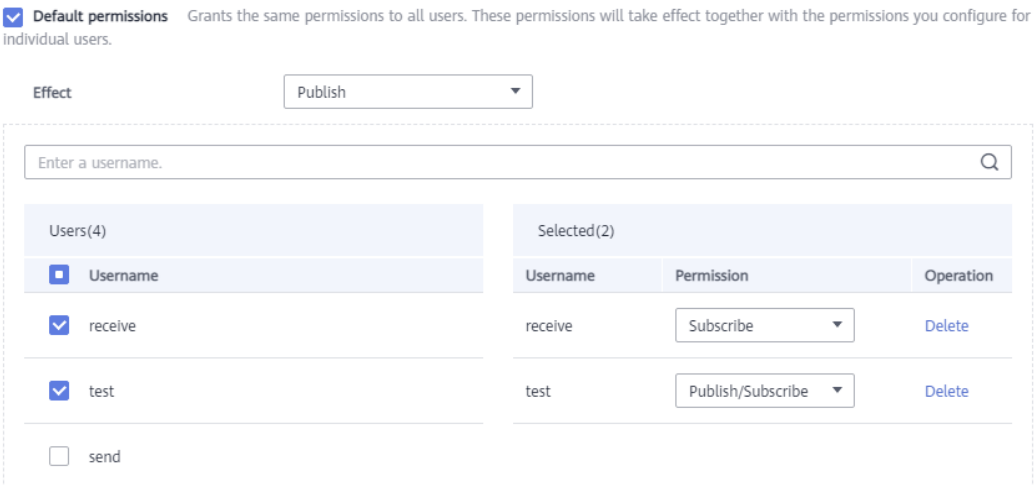
- To grant different permissions to different users, do not select **Default permissions**. In the **Users** area of the **Grant User Permission** dialog box, select target users. In the **Selected** area, configure permissions (**Subscribe**, **Publish**, or **Publish/Subscribe**) for the users. As 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 7-16 Granting permissions to individual users



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, while the **send** user can only publish messages to the topic.

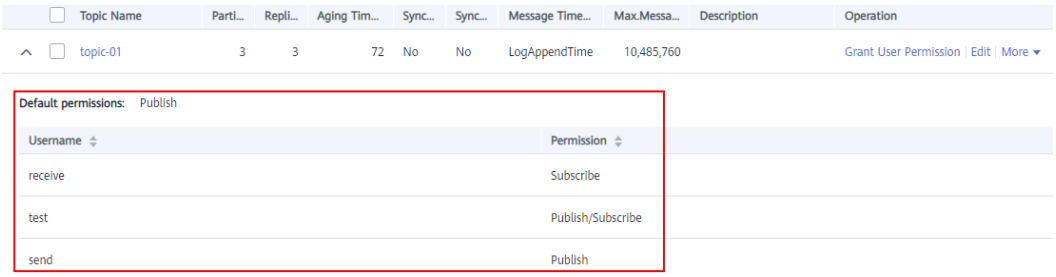
Figure 7-17 Granting topic permissions to users



Step 8 Click **OK**.

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

Figure 7-18 Viewing authorized users and their permissions



----End

(Optional) Deleting Topic Permissions

Step 1 Log in to the management 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 **Middleware** > **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 that contains the topic for which you want to remove user permissions, click **Grant User Permission**.

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

----End

7.11 Enabling or Disabling Automatic Topic Creation

If automatic topic creation is enabled, the system automatically creates a topic when a message is created in or retrieved from a topic that does not exist. This topic has the following default settings: 3 partitions, 3 replicas, aging time 72 hours, and synchronous replication and flushing disabled. After you change the value of the **log.retention.hours**, **default.replication.factor**, or **num.partitions** parameter, automatically created topics later use the new value. For example, if **num.partitions** is set to **5**, an automatically created topic will have the following settings: 5 partitions, 3 replicas, aging time 72 hours, and synchronous replication and flushing disabled.

Procedure

Step 1 Log in to the management 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 **Middleware** > **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 Click or next to **Automatic Topic Creation**. The **Confirm** dialog box is displayed.

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

7.12 Viewing Topic Details

On the console, you can view subscriptions to a topic, offsets and number of messages in each partition, and producer addresses of a Kafka instance.

Procedure

Step 1 Log in to the management 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 **Middleware** > **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 **Topics** tab.

Step 6 Click a topic to view its details.

The general information, subscriptions, partitions, and producers are displayed.

- General information: topic name, brokers, partitions, and creation time

  NOTE

  - The creation time is not displayed for topics created on and before Jul 10, 2023.
  - The creation time is not displayed for topics automatically created, created by commands or code in clients, or created with Kafka Manager.

- Subscriptions: consumer group name and status, Coordinator (ID), and accumulation

In the **Operation** column, click **Details**. The consumer group details are displayed.

- Partitions: partition No., minimum offset, maximum offset, number of messages, and message update time
- Producers: broker address, producer address, and producer connected time

  NOTE

  - The producer information is displayed only when a producer is producing a message into the topic.
  - For topics created on or before Jul 10, 2023, **Producers** is not displayed on the topic details page.

----End

8 Managing Messages

8.1 Querying Messages

Scenario

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 management 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 **Middleware** > **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 left navigation pane, choose **Message Query**.

Step 6 Set the query parameters by referring to **Table 8-1**.

Table 8-1 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.
Content	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 8-1 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

8.2 Deleting a Message

Scenario

Delete messages on the console.

NOTICE

Deleted messages cannot be recovered.

Prerequisites

Before deleting a message, set the **auto.offset.reset** parameter on the client. **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 management 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 **Middleware** > **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 **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 8-2**.

Figure 8-2 Deleting messages

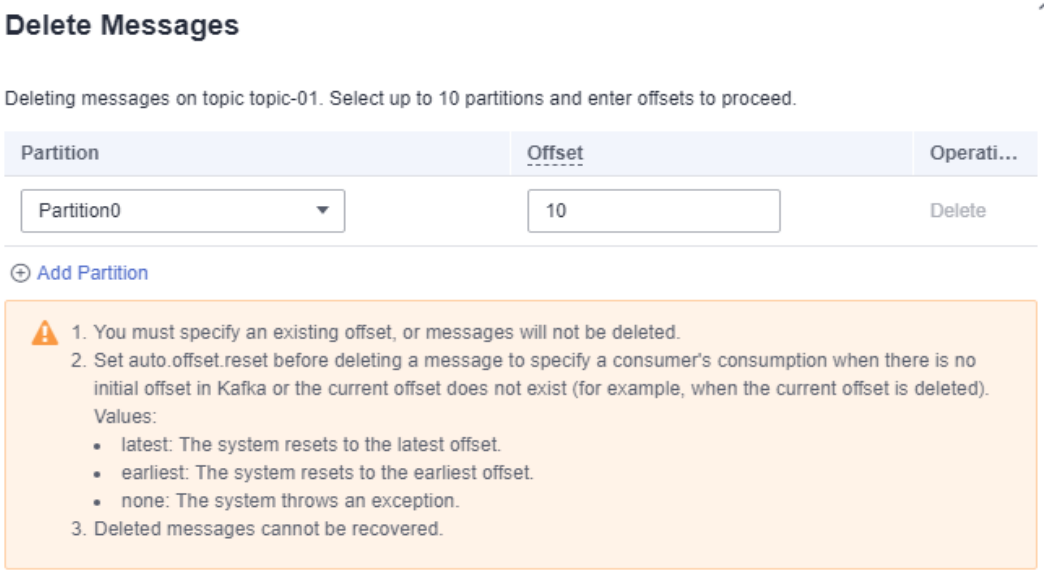


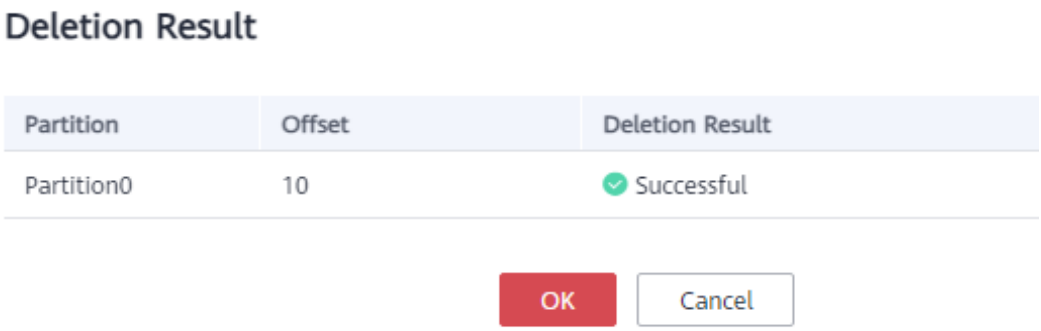
Table 8-2 Parameters for deleting a message

Parameter	Description
Partition	Select the ID of the partition where the message is located.
Offset	Enter the offset. Data before this offset 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 8-3 Deletion result



----End

8.3 Producing a Message

Scenario

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 only when the instance is in the **Running** state.

Procedure

Step 1 Log in to the management 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 **Middleware** > **Distributed Message Service (for Kafka)** to open the console of DMS for Kafka.

Step 4 Click the desired Kafka instance to view details.

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 8-3**.

Table 8-3 Message parameters

Parameter	Description
Message Body	Message content.

Parameter	Description
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

9 Managing Users

9.1 Creating a SASL_SSL User

DMS for Kafka supports ACL permission management for topics. You can differentiate the operations that different users are allowed to perform on a topic by granting the users different permissions.

This section describes how to create a SASL_SSL user after SASL_SSL is enabled for a Kafka instance. For details about how to grant user permissions, see **Configuring Topic Permissions**.

For Kafka instances created before July 15, 2023, a maximum of 20 users can be created for each instance. For Kafka instances created since July 15, 2023, a maximum of 500 users can be created for each instance.

NOTE

This function is unavailable for single-node instances.

Prerequisites

SASL_SSL has been enabled when you create the Kafka instance.

Procedure

Step 1 Log in to the management 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 **Middleware** > **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 **Users** page, click **Create User**.

Step 6 In the displayed **Create User** dialog box, set the username, password, and description, and click **OK**.

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

----End

9.2 Resetting the SASL_SSL Password

Scenario

For a Kafka instance with SASL_SSL enabled, there are two ways to create an SASL_SSL user on the console. Accordingly, there are two ways to reset the SASL_SSL user's password:

- If an SASL_SSL user is created on the **Users** page, reset their password by referring to the following instructions.
- If an SASL_SSL user is created during instance creation, reset their password by referring to **Resetting Kafka Password**.

NOTE

This function is unavailable for single-node instances.

Prerequisites

- You can reset the SASL_SSL password only if Kafka SASL_SSL has been enabled for the instance.
- You can reset the SASL_SSL password only when the instance is in the **Running** state.

Procedure

Step 1 Log in to the management 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 **Middleware** > **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 **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

9.3 Modifying SASL_SSL User Description

Scenario

After creating a SASL_SSL user, you can modify its description based on service requirements.

NOTE

This function is unavailable for single-node instances.

Procedure

Step 1 Log in to the management 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 **Middleware** > **Distributed Message Service (for Kafka)** to open the console of DMS for Kafka.

Step 4 Click the desired Kafka instance to view details.

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

Step 6 In the row containing the user to be edited, click **Edit**.

Step 7 Modify the description and click **OK**.

After the modification is successful, you can view the new description in the **Description** column.

----End

9.4 Deleting a SASL_SSL User

This section describes how to delete a SASL_SSL user.

NOTE

This function is unavailable for single-node instances.

Procedure

Step 1 Log in to the management 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 **Middleware** > **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 Delete a SASL_SSL user using either of the following methods:

- On the **Users** page, click **Delete** in the row that contains the SASL_SSL user to be deleted.
- On the **Users** page, select one or more SASL_SSL users and click **Delete** above the list.

NOTE

The SASL_SSL user configured during the creation of a Kafka instance cannot be deleted.

Step 6 In the displayed **Delete User** dialog box, click **OK** to delete the SASL_SSL user.

----End

10 Managing Consumer Groups

10.1 Creating a Consumer Group

Create a consumer group on the console.

If the **auto.create.groups.enable** parameter has been enabled for the instance, a consumer group is automatically created when a consumer attempts to enter a group that does not exist. Then creating a consumer group is optional.

NOTE

- This function is supported for instances created on or after April 25, 2023.
- 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.

Procedure

Step 1 Log in to the management 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 **Middleware** > **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 Enter the consumer group name and description, and click **OK**.

View the new consumer group in the consumer group list.

----End

10.2 Querying Consumer Group Details

View the consumer group list, consumer list, and consumer offsets.

Prerequisites

The consumer list 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 Group List (Console)

Step 1 Log in to the management 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 **Middleware** > **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, Coordinator (ID), and description 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 10-1 Consumer group list

Step 6 (Optional) To query a specific consumer group, enter the consumer group name in the search box and click .

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

----End

Viewing the Consumer Group List (Kafka CLI)

- If SASL is not enabled for the Kafka instance, run the following command in the */{directory where the CLI is located}*/**kafka_{version}/bin/** directory to query the consumer group list:
  ```
  ./kafka-consumer-groups.sh --bootstrap-server {broker_ip}:{port} --list
  ```
- If SASL has been enabled for the Kafka instance, perform the following steps to query the consumer group list:
  a. (Optional) If the SSL certificate configuration has been set, skip this step. Otherwise, perform the following operations:

     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 */{directory where the CLI is located}*/**kafka_{version}/bin/** directory to query the consumer group list:
     ```
     ./kafka-consumer-groups.sh --bootstrap-server {broker_ip}:{port} --list --command-config ./config/ssl-user-config.properties
     ```

Viewing the Consumer List (Console)

Step 1 Log in to the management 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 **Middleware** > **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.

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 click  .

----End

Viewing the Consumer List (Kafka CLI)

- If SASL is not enabled for the Kafka instance, run the following command in the */{directory where the CLI is located}/***kafka_{version}/bin/** directory to query the consumer list:
  ```
  ./kafka-consumer-groups.sh --bootstrap-server {broker_ip}:{port} --group {group_name} --members --describe
  ```
- If SASL has been enabled for the Kafka instance, perform the following steps to query the consumer list:
  a. (Optional) If the SSL certificate configuration has been set, skip this step. Otherwise, perform the following operations:

     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 */{directory where the CLI is located}/***kafka_{version}/bin/** directory to query the consumer list:
     ```
     ./kafka-consumer-groups.sh --bootstrap-server {broker_ip}:{port} --group {group_name} --members --describe --command-config ./config/ssl-user-config.properties
     ```

Viewing Consumer Offsets (Console)

Step 1 Log in to the management 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 **Middleware** > **Distributed Message Service (for Kafka)** to open the console of DMS for Kafka.

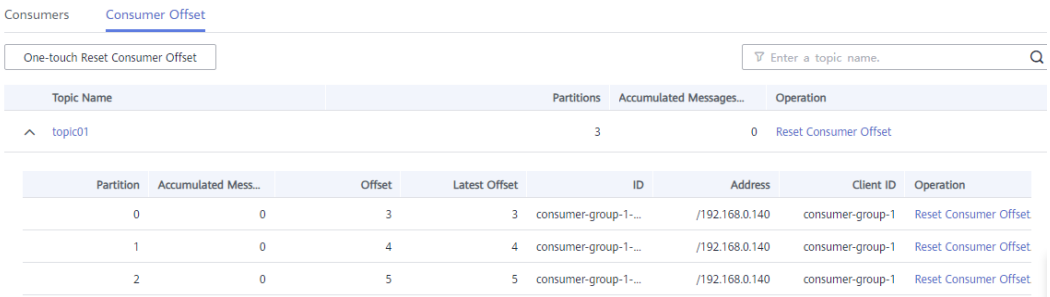
Step 4 Click the desired Kafka instance to view its details.

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

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, message consumption progress in each partition of the topic (accumulated messages, offset, latest offset, consumer ID, consumer address, and client ID).

Figure 10-2 Consumption progress



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

----End

Viewing Consumer Offsets (Kafka CLI)

- If SASL is not enabled for the Kafka instance, run the following command in the */{directory where the CLI is located}*/**kafka_{version}/bin/** directory to query consumer offsets:
  ```
  ./kafka-consumer-groups.sh --bootstrap-server {broker_ip}:{port} --offsets --describe --all-groups
  ```
- If SASL has been enabled for the Kafka instance, perform the following steps to query consumer offsets:
  a. (Optional) If the SSL certificate configuration has been set, skip this step. Otherwise, perform the following operations:

     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 */{directory where the CLI is located}*/**kafka_{version}/bin/** directory to query consumer offsets:
     ```
     ./kafka-consumer-groups.sh --bootstrap-server {broker_ip}:{port} --offsets --describe --all-groups --command-config ./config/ssl-user-config.properties
     ```

10.3 Deleting a Consumer Group

You can delete a consumer group using either of the following methods:

- Method 1: Delete a consumer group on the console.
- Method 2: Use **Kafka CLI** to delete a consumer group. (Ensure that the Kafka instance version is the same as the CLI version.)

Prerequisites

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

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.

Method 1: Deleting a Consumer Group on the Console

Step 1 Log in to the management 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 **Middleware** > **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

Method 2: Using the CLI to Delete a Consumer Group

The following uses Linux as an example.

Step 1 Download Kafka CLI **v1.1.0**, **v2.3.0**, or **v2.7.2**. Ensure that the Kafka instance and the CLI are of the same version.

Step 2 Use the CLI to connect to the Kafka instance. For details, see **Accessing a Kafka Instance Without SASL** or **Accessing a Kafka Instance with SASL**.

Step 3 In the **/***{directory where the CLI is located}***/kafka_***{version}***/bin/** directory, run the following command to delete a consumer group:

./kafka-consumer-groups.sh --bootstrap-server *{Kafka instance connection address}* **--delete --group** *{consumer group name}*

```
[root@zk-server-1 bin]# ./kafka-consumer-groups.sh --bootstrap-server
192.168.1.245:9091,192.168.1.86:9091,192.168.1.128:9091 --delete --group bbbb
Note: This will not show information about old Zookeeper-based consumers.
Deletion of requested consumer groups ('bbbb') was successful.
```


NOTE

If SASL authentication is enabled for the Kafka instance, the **--command-config *{consumer.properties file with SASL authentication}*** parameter must be added to the preceding commands. For details about the **consumer.properties** file, see **Accessing a Kafka Instance with SASL**.

----End

10.4 Resetting the Consumer Offset

Resetting the consumer offset is to change the retrieval position of a consumer.

NOTICE

Messages may be retrieved 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).

Procedure

Step 1 Log in to the management 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 **Middleware** > **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 in all topics, click **One-touch Reset Consumer Offset** above the list.

Step 8 In the displayed **Reset Consumer Offset** dialog box, set the parameters by referring to **Table 10-1**.

Table 10-1 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.
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

10.5 Viewing Consumer Connection Addresses

You can view connection addresses of consumers using either of the following methods:

- Method 1: View consumer connection addresses on the management console.
- Method 2: View consumer connection addresses on Kafka Manager.

NOTE

- The connection address of a consumer can be viewed only when the consumer is connected to a Kafka instance.
- Due to cache reasons, the consumer connection addresses displayed on Kafka Manager may not be used currently. To solve this problem, restart Kafka Manager.
- Instances created on or after May 17, 2023 do not have Kafka Manager. You cannot view consumer addresses of these instances using Kafka Manager.

Method 1: Viewing Consumer Addresses on Console

Step 1 Log in to the management 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 **Middleware** > **Distributed Message Service (for Kafka)** to open the console of DMS for Kafka.

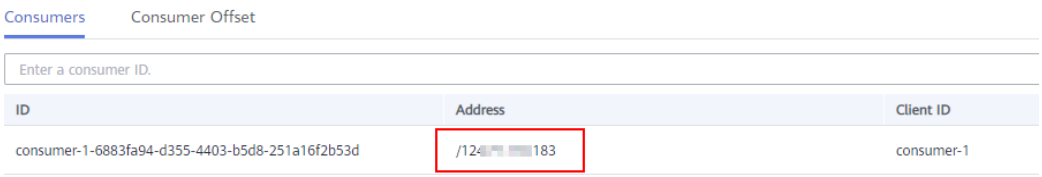
Step 4 Click the desired Kafka instance to view the instance details.

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.

Figure 10-3 Consumer list



----End

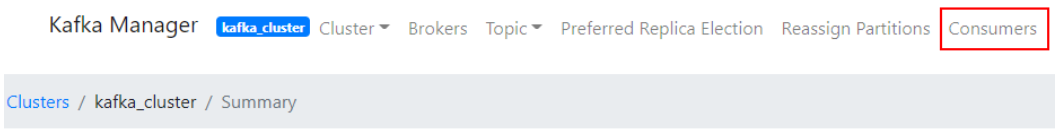
Method 2: Viewing Consumer Addresses on Kafka Manager

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

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

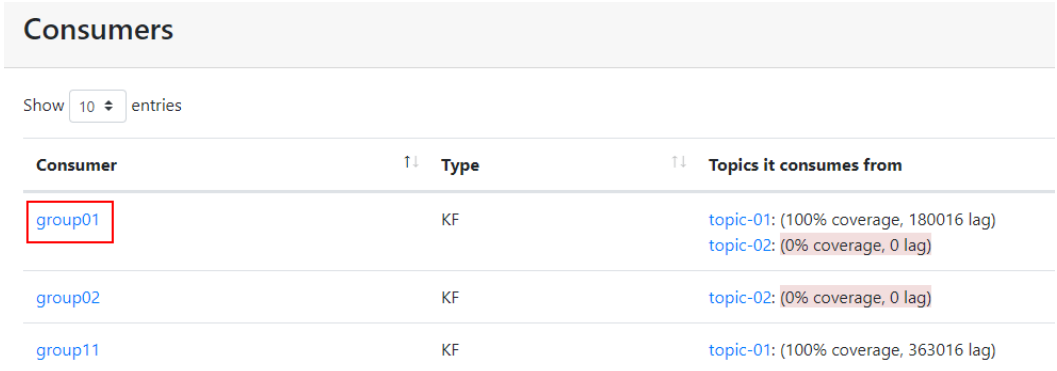
Step 3 On the top menu bar, choose **Consumers**.

Figure 10-4 Navigation bar



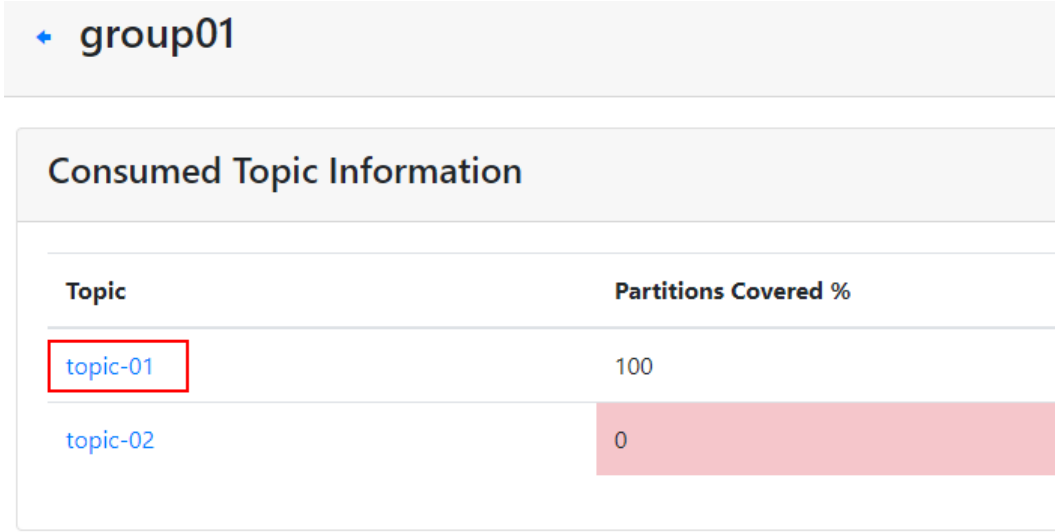
Step 4 Click the desired consumer group to view the topics that the group has subscribed to.

Figure 10-5 Consumer group list



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

Figure 10-6 Topics that the consumer group has subscribed to



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

Figure 10-7 Topic details page

----End

10.6 Viewing Rebalancing Logs

Scenario

Rebalancing logs record rebalancing details, including the time, reason, and triggering client of rebalancing. This section describes how to view rebalancing logs on the console.

Rebalancing logs are stored and can be queried in Log Tank Service (LTS).

NOTE

This function is unavailable for single-node instances.

What Is Rebalancing?

Rebalancing is to reallocate subscription relationships between consumers and topic partitions in a consumer group. During rebalancing, all consumers in the consumer group stop consuming messages until rebalancing completes.

Possible causes of rebalancing:

- The number of consumer group members changes. For example, a new consumer joins the group or a consumer quits the group.
- The number of topics subscribed to by a consumer group changes.
- The number of topic partitions subscribed to by a consumer group changes.

Constraints

- Rebalancing logging is automatically disabled when you scale the instance. If you want to continue using it, enable it again.
- Rebalancing logging is not available for instances created before April 6, 2023.
- By default, rebalancing logs are retained for 7 days. To store the logs for longer, **modify the log group retention period on the LTS console**.
- Enabling rebalancing logging will create a log group, log stream, and dashboard in LTS. Fees are generated based on the log volume. For details, see **LTS pricing details**.

Prerequisites

- Ensure that you have permissions to create log groups and log streams in LTS.
- Rebalancing logging can be enabled or disabled only when the Kafka instance is in the **Running** state.

Enabling Rebalancing Logging

Step 1 Log in to the management 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 **Middleware** > **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 **Rebalancing Logs**.

Step 6 Click **Enable Logging**. If the message "Rebalancing logging enabled" is displayed in the upper right corner of the page, the rebalancing log function is enabled successfully.

Enabling rebalancing logging will create a log group and log stream in LTS.

----End

Viewing Rebalancing Logs

Step 1 Log in to the management 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 **Middleware** > **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 **Rebalancing Logs**.

Step 6 On the **Dashboard** tab page, view the number of consumer group rebalancing times and reasons. On the **Logs** tab page, view rebalancing logs.

For details about how to search for logs, see **Log Search**.

An example rebalancing log:

```
{
   "level":"INFO",
   "timestamp":"2023-03-23 17:23:22,906",
   "message":{
      "leaderId":"consumer-1-177817b6-1f29-4717-8a83-dda8eaab1635",
      "generationId":"1",
      "reason":"Assignment received from leader for group KMOffsetCache-dms-vm-fa3cf9d6-manager-
shared-server-0 for generation 1",
      "groupId":"KMOffsetCache-dms-vm-fa3cf9d6-manager-shared-server-0",
      "coordinatorId":"0",
      "type":"END_REBALANCE",
      "group":"GroupMetadata(groupId=KMOffsetCache-dms-vm-fa3cf9d6-manager-shared-server-0,
generation=1, protocolType=Some(consumer), currentState=CompletingRebalance,
members=Map(consumer-1-177817b6-1f29-4717-8a83-dda8eaab1635 -&gt;
MemberMetadata(memberId=consumer-1-177817b6-1f29-4717-8a83-dda8eaab1635, clientId=consumer-1,
clientHost=/172.31.2.168, sessionTimeoutMs=10000, rebalanceTimeoutMs=300000,
supportedProtocols=List(range), )))"
   }
}
```


Table 10-2 describes the parameters.

Table 10-2 Rebalancing parameters

Parameter	Description
level	Level of the rebalancing log.
timestamp	Time of rebalancing.
leaderId	Leader consumer ID.
generationId	Generation ID of the consumer group. Generation is the number of times that a consumer group performs rebalancing. It is incremented by 1 each time a rebalancing is complete.
reason	Reason for triggering rebalancing.
groupId	Consumer group ID.
coordinatorId	Broker where the Coordinator component is.
type	Operation that triggered rebalancing.
group	Information about consumers in the consumer group.

----End

Disabling Rebalancing Logging

Step 1 Log in to the management 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 **Middleware** > **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 **Rebalancing Logs**.

Step 6 In the upper right corner of the page, click **Disable Logging**. In the dialog box that is displayed, click **OK**.

NOTICE

This only disables the rebalancing logging function. The log groups and log streams on LTS are retained and still generate fees. If you no longer need the logs, delete the **log groups** and **log streams** on LTS.

----End

10.7 Modifying Consumer Group Description

Scenario

After creating a consumer group, you can modify its description based on service requirements.

Procedure

Step 1 Log in to the management 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 **Middleware** > **Distributed Message Service (for Kafka)** to open the console of DMS for Kafka.

Step 4 Click the desired Kafka instance to view details.

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

Step 6 In the row containing the consumer group to be edited, click **Edit**.

Step 7 Modify the description and click **OK**.

After the modification is successful, you can view the new description in the **Description** column.

----End

10.8 Exporting Consumer Groups

Scenario

Export the consumer group list from the console.

Procedure

Step 1 Log in to the management 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 **Middleware** > **Distributed Message Service (for Kafka)** to open the console of DMS for Kafka.

Step 4 Click **Export** and choose to export all data or selected data to an XLSX file to export the consumer group list.

----End

11 Smart Connect

11.1 Enabling Smart Connect

Scenario

Smart Connect synchronizes data between Kafka and other cloud services (such as OBS) or between two Kafka instances for backup.

Procedure for using Smart Connect:

1. Enable Smart Connect.
2. Create a Smart Connect task.

This section describes how to enable Smart Connect.

NOTE

Smart Connect cannot be enabled for single-node Kafka instances.

Impact

Enabling Smart Connect incurs additional broker fees.

For example, if you enable Smart Connect for a kafka.4u8g.cluster instance, two more kafka.4u8g brokers will be created for Smart Connect and you need to pay for them.

Prerequisites

A Kafka instance has been created and is in the **Running** state.

Procedure

Step 1 Log in to the management 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 **Middleware** > **Distributed Message Service (for Kafka)** to open the console of DMS for Kafka.

Step 4 Enable Smart Connect using one of the following methods:

- In the row containing the desired Kafka instance, choose **More** > **Enable Smart Connect**.
- Click the desired Kafka instance to view its details. In the upper right corner, choose **More** > **Enable Smart Connect**.
- Click the desired Kafka instance to view its details. Click ✎ next to **Smart Connect**.
- Click the desired Kafka instance to view its details. In the navigation pane, choose **Smart Connect**. Click **Enable Smart Connect**.

Step 5 Click ⬭ to enable Smart Connect. Then click **Next**.

Step 6 On the displayed **Enabling Smart Connect for Kafka Instance** page, ensure that **Smart Connect** is enabled and click **Submit**.

----End

Follow-up Operations

Proceed to **Creating a Smart Connect Task (Kafka)**, **Creating a Smart Connect Task (Dumping)**, and **Creating a Smart Connect Task (Custom)** to synchronize data between DMS for Kafka and other cloud services.

11.2 Creating a Smart Connect Task (Kafka)

Scenario

Create a Smart Connect task to copy data unidirectionally or bidirectionally between two Kafka instances.

NOTE

- If you have enabled Smart Connect for an instance before July 1, 2022 and Kafka data replication is not available, **disable Smart Connect** and then enable it again.
- This function is unavailable for single-node instances.
- Data in the source Kafka instance is synchronized to the target Kafka instance in real time.

Restrictions

- A maximum of 18 Smart Connect tasks can be created for an instance.
- When you copy Kafka data, the two Kafka instances must be connected through the intranet. If they are in different VPCs, connect the network by referring to **Cross-VPC Access to a Kafka Instance**.

- After a Smart Connect task is created, task parameters cannot be modified.

Prerequisites

- You have **enabled Smart Connect**.
- A Kafka instance has been created and is in the **Running** state.
- A topic has been created.

Procedure

Step 1 Log in to the management 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 **Middleware** > **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 **Smart Connect**.

Step 6 On the displayed page, click **Create Task**.

Step 7 For **Task Name**, enter a unique Smart Connect task name.

Step 8 For **Task Type**, select **Copy Kafka data**.

Step 9 For **Start Immediately**, specify whether to execute the task immediately after the task is created. By default, the task is executed immediately. If you disable this option, you can enable it later in the task list.

Step 10 In the **Current kafka** area, set the instance alias.

The instance alias is used in the following scenarios:

- If you enable **Rename Topics**, the alias of the source instance will be added to the topic names of the target instance. For example, if the alias of the source instance is **A** and the target topic name is **test**, the renamed target topic will be **A.test**.
- After the Smart Connect task is created, a topic named **mm2-offset-syncs.***Target instance alias***.internal** is automatically created. If **Sync Consumer Offset** is enabled for the task, a topic named *Target instance alias***.checkpoints.internal** is automatically created. The two topics are used to store internal data. If they are deleted, data replication will fail.

Step 11 In the **Peer Kafka** area, configure the following parameters.

Table 11-1 Peer Kafka parameters

Parameter	Description
Instance Alias	Set the instance alias. The instance alias is used in the following scenarios: • If you enable **Rename Topics**, the alias of the source instance will be added to the topic names of the target instance. For example, if the alias of the source instance is **A** and the target topic name is **test**, the renamed target topic will be **A.test**. • After the Smart Connect task is created, a topic named **mm2-offset-syncs.***Target instance alias***.internal** is automatically created. If **Sync Consumer Offset** is enabled for the task, a topic named *Target instance alias***.checkpoints.internal** is automatically created. The two topics are used to store internal data. If they are deleted, data replication will fail.
Config Type	Options: • **Kafka address**: Enter Kafka instance addresses. • **Instance name**: Select an existing Kafka instance.
Instance Name	Set this parameter when **Config Type** is set to **Instance name**. Select an existing Kafka instance from the drop-down list. **The peer Kafka instance and the current Kafka instance must be in the same VPC. Otherwise, they cannot be identified.**
Kafka Address	Set this parameter when **Config Type** is set to **Kafka address**. Enter the IP addresses and port numbers for connecting to the Kafka instance. When you copy Kafka data, the two Kafka instances must be connected through the intranet. If they are in different VPCs, connect the network by referring to **Cross-VPC Access to a Kafka Instance**.
Authentication	There are two authentication modes: • **SASL_SSL**: indicates that SASL_SSL has been enabled for the instance. • **PLAINTEXT**: indicates that SASL_SSL is not enabled for the instance.
Authentication Mechanism	Set this parameter when **Authentication** is set to **SASL_SSL**. Select a mechanism for SASL authentication.

Parameter	Description
Username	Set this parameter when **Authentication** is set to **SASL_SSL**. Enter the username you set when enabling SASL_SSL during Kafka instance creation or when creating a SASL_SSL user.
Password	Set this parameter when **Authentication** is set to **SASL_SSL**. Enter the password you set when enabling SASL_SSL during Kafka instance creation or when creating a SASL_SSL user.

Step 12 In the **Rules** area, configure the following parameters.

Table 11-2 Parameters for configuring data replication rules

Parameter	Description
Sync Direction	There are three synchronization directions: • **Pull**: Replicates data from the peer Kafka instance to the current Kafka instance. • **Push**: Replicates data from the current Kafka instance to the peer Kafka instance. • **Both**: Bidirectional replication of Kafka instance data on both ends.
Topics	Specify the topics whose data is to be replicated. • **Regular expression**: Use a regular expression to match topics. • **Enter/Select**: Enter topic names. To enter multiple topic names, press **Enter** after entering each topic name. You can also select topics from the drop-down list. **A maximum of 20 topics can be entered or selected.** **NOTE** Data of topics whose names end with "internal" (for example, **topic.internal**) will not be synchronized.
Tasks	Number of data replication tasks. The default value is **2**. You are advised to use the default value. If **Sync Direction** is set to **Both**, the actual number of tasks will be twice the number of tasks you configure here.

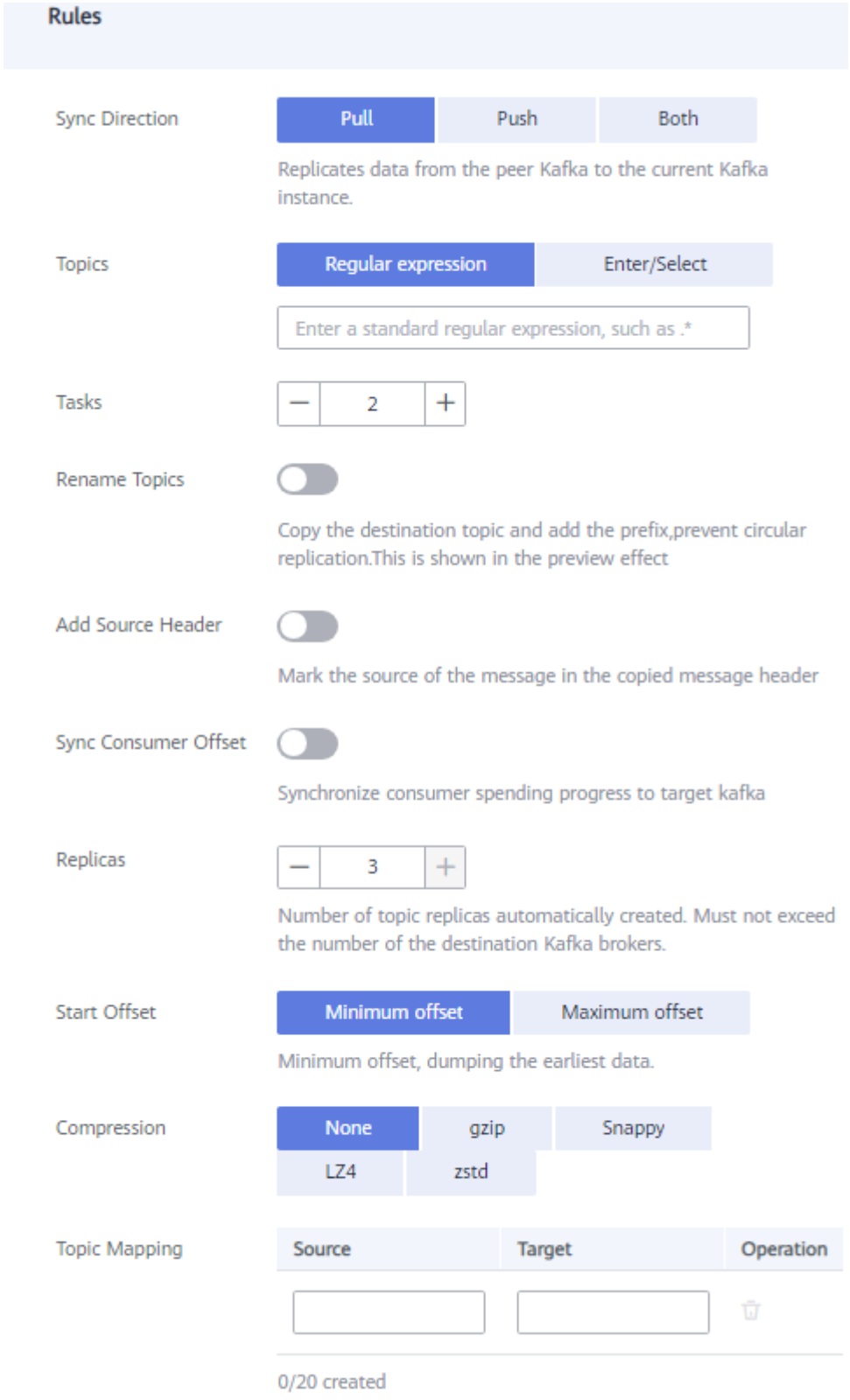
Parameter	Description
Rename Topics	Add the alias of the source Kafka instance before the target topic name to form a new name of the target topic. For example, if the alias of the source instance is **A** and the target topic name is **test**, the renamed target topic will be **A.test**. If you select **Both** for **Sync Direction**, enable **Rename Topics** to prevent infinite replication.
Add Source Header	The target topic receives the replicated messages. The message header contains the message source. If you select **Both** for **Sync Direction**, **Add Source Header** is enabled by default to prevent infinite replication.
Sync Consumer Offset	Enable this option to synchronize the consumer offset to the target Kafka instance. **NOTICE** After enabling **Sync Consumer Offset**, pay attention to the following: • The source and target Kafka instances cannot consume messages at the same time. Otherwise, the synchronized consumer offset will be abnormal. • The consumer offset is synchronized every minute. As a result, the consumer offset on the target end may be slightly smaller than that on the source end, and some messages are repeatedly consumed. The service logic of the consumer client must be able to handle repeated consumption. • The offset synchronized from the source end is not the same as the offset on the target end. Instead, there is a mapping relationship. If the consumer offset is maintained by the consumer client, the consumer client does not obtain the consumer offset from the target Kafka instance after switching consumption from the source Kafka instance to the target Kafka instance. As a result, the offset may be incorrect or the consumer offset may be reset.
Replicas	Number of topic replicas when a topic is automatically created in the peer instance. The value of this parameter cannot exceed the number of brokers in the peer instance. This parameter takes precedence over the **default.replication.factor** parameter set in the peer instance.
Start Offset	Options: • **Minimum offset**: dumping the earliest data • **Maximum offset**: dumping the latest data
Compression	Compression algorithm to use for copying messages.

Parameter	Description
Topic Mapping	Customize the target topic name. Maximum mappings: 20. **Rename Topic** and **Topic Mapping** cannot be configured at the same time.

NOTICE

- When creating a bidirectional replication task, you must enable **Rename Topics** or **Add Source Header** to prevent infinite replication. If you specify the same topic for a pull task and a push task between two instances (forming bidirectional replication), and **Rename Topics** and **Add Source Header** are not enabled for the two tasks, data will be replicated infinitely.
- If you create two or more tasks with the same configuration and enable **Sync Consumer Offset** for them, data will be repeatedly replicated and the consumer offset of the target topic will be abnormal.

Figure 11-1 Configuring data replication rules



Step 13 (Optional) In the lower right corner of the page, click **Check** to test the connectivity between the Kafka instances.

If "Connectivity check passed." is displayed, the Kafka instances are connected.

Step 14 Click **Create**. The Smart Connect task list page is displayed. The message "Task *xxx* was created successfully." is displayed in the upper right corner of the page.

After the Smart Connect task is created, a topic named **mm2-offset-syncs.***Target instance alias*.**internal** is automatically created. If **Sync Consumer Offset** is enabled for the task, a topic named *Target instance alias*.**checkpoints.internal** is automatically created. The two topics are used to store internal data. If they are deleted, data replication will fail.

----End

11.3 Creating a Smart Connect Task (Dumping)

Scenario

Create a Smart Connect task to dump Kafka instance data to OBS.

NOTE

- This function is unavailable for single-node instances.
- Data in the source Kafka instance is synchronized to the dumping file in real time.

Restrictions

- A maximum of 18 Smart Connect tasks can be created for an instance.
- After a Smart Connect task is created, task parameters cannot be modified.

Prerequisites

- You have **enabled Smart Connect**.
- A Kafka instance has been created and is in the **Running** state.
- A topic has been created.

Procedure

Step 1 Log in to the management 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 **Middleware** > **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 **Smart Connect**.

Step 6 On the displayed page, click **Create Task**.

Step 7 For **Task Name**, enter a unique Smart Connect task name.

Step 8 For **Task Type**, select **Dumping**.

Step 9 For **Start Immediately**, specify whether to execute the task immediately after the task is created. By default, the task is executed immediately. If you disable this option, you can enable it later in the task list.

Step 10 In the **Source** area, retain the default setting.

Step 11 In the **Topics** area, set parameters based on the following table.

Table 11-3 Topic parameters

Parameter	Description
Regular expression	A regular expression is used to subscribe to topics whose messages you want to dump.
Enter/Select	Enter or select the names of the topics to be dumped. Separate them with commas (,). **A maximum of 20 topics can be entered or selected.**

Step 12 In the **Target** area, set parameters based on the following table.

Table 11-4 Target parameters

Parameter	Description
Offset	Options: • **Minimum offset**: dumping the earliest data • **Maximum offset**: dumping the latest data
Dumping Period (s)	Interval for periodically dumping data. The time unit is second and the default interval is 300 seconds. No package files will be generated if there is no data within an interval.
AK	Access key ID. For details about how to obtain the AK, see **Access Keys**.
SK	Secret access key used together with the access key ID. For details about how to obtain the SK, see **Access Keys**.
Dumping Address	The OBS bucket used to store the topic data. You can select an existing OBS bucket from the drop-down list or click **Create Dumping Address** to create a new OBS bucket.
Dumping Directory	Directory for storing topic files dumped to OBS. Use slashes (/) to separate directory levels.

Parameter	Description
Time Directory Format	Data is saved to a hierarchical time directory in the dumping directory. For example, if the time directory is accurate to day, the directory will be in the format of *bucket name/file directory/year/month/day*.
Record Separator	Select a separator to separate OBS dumping records.
Use Storage Key	Specifies whether to dump keys.

NOTE

Do not use the key of a message as the dumping file name.

Step 13 Click **Create**. The Smart Connect task list page is displayed. The message "Task *xxx* was created successfully." is displayed in the upper right corner of the page.

----End

11.4 Creating a Smart Connect Task (Custom)

There are two types of custom Smart Connect tasks: **Kafka data replication** and **dumping**.

- Kafka data replication: Create a Smart Connect task for one-way or two-way data replication between two Kafka instances.
- Dumping: Create a Smart Connect task to dump Kafka data to an OBS bucket.

11.5 Managing Smart Connect Tasks

Scenario

View, delete, start, and pause Smart Connect tasks on the DMS for Kafka console.

NOTE

Smart Connect tasks cannot be managed for single-node Kafka instances.

Prerequisites

A Smart Connect task has been **created**.

Viewing Smart Connect Tasks

Step 1 Log in to the management 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 **Middleware** > **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 **Smart Connect**.

Step 6 Click a Smart Connect task name to go to the details page.

Step 7 View the basic information, source, and target of the Smart Connect task.

NOTE

The source and target are displayed on the task details page only when they have been configured for the Smart Connect task.

----End

Deleting a Smart Connect Task

Step 1 Log in to the management 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 **Middleware** > **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 **Smart Connect**.

Step 6 In the row containing the Smart Connect task to be deleted, click **Delete**.

Step 7 Click **OK**.

----End

Starting or Pausing a Smart Connect Task

After a task of a Kafka instance is paused, data of the instance will not be synchronized to another Kafka instance or other cloud services.

Step 1 Log in to the management 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 **Middleware** > **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 **Smart Connect**.

Step 6 Perform the required operation:

- To start a Smart Connect task, click **Start** in the row that contains the task.
- To pause a Smart Connect task, click **Pause** in the row that contains the task, then click **OK** in the dialog box that is displayed.

----End

11.6 Disabling Smart Connect

Scenario

This section describes how to disable Smart Connect.

Disabling Smart Connect does not affect services.

NOTE

Smart Connect cannot be disabled for single-node Kafka instances.

Impact

- Brokers related to Smart Connect are automatically deleted, and no longer generate fees.
- If you disable Smart Connect and then enable it again, deleted Smart Connect tasks cannot be retrieved and need to be created again.

Prerequisites

- A Kafka instance has been created and is in the **Running** state.
- **All Smart Connect tasks must be deleted**. This is to prevent running Smart Connect tasks from being lost after Smart Connect is disabled.

Procedure

Step 1 Log in to the management 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 **Middleware** > **Distributed Message Service (for Kafka)** to open the console of DMS for Kafka.

Step 4 Disable Smart Connect using either of the following methods:

- In the row containing the desired Kafka instance, choose **More** > **Disable Smart Connect**.
- Click the desired Kafka instance to view its details. In the upper right corner, choose **More** > **Disable Smart Connect**.

Step 5 Click  to disable Smart Connect. Then click **Next**.

Step 6 Ensure that **Smart Connect** is disabled and click **Submit**.

----End

12 Managing Kafka Quotas

12.1 Creating a Quota

Scenario

On the console, you can control the message production and consumption rate limits for users, clients, or topics.

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

NOTE

- This function is supported for instances created on or after November 10, 2022.
- This function is unavailable for single-node instances.

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 management 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 **Middleware** > **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 corner, click **Create Quota**. The **Create Quota** slide panel is displayed.

Step 8 Set quota parameters.

Table 12-1 Quota parameters

Parameter	Description
Username	Enter the name of the user to apply the quota to. 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 ID of the client to which the quota applies. 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.

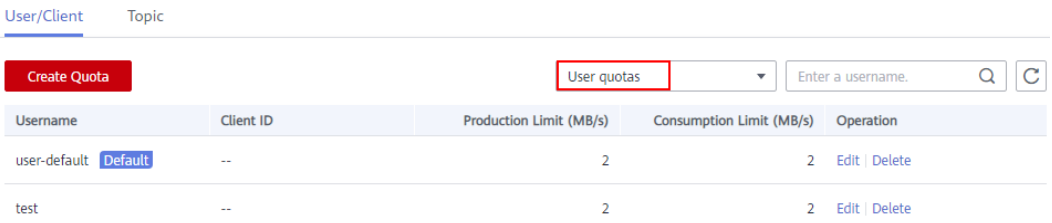
NOTE

- If SASL is not enabled for the instance, **Username** is not displayed in the **Create Quota** slide panel.
- **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 click to view the created quota.

Figure 12-1 Viewing the new quota



----End

Creating a Topic Quota

Step 1 Log in to the management 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 **Middleware** > **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 12-2 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 right corner, then click to view the created quota.

----End

12.2 Modifying a Quota

Scenario

After creating quotas, you can change the production or consumption rate limits.

NOTE

This function is unavailable for single-node instances.

Procedure

Step 1 Log in to the management 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 **Middleware** > **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 quota to be edited, 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.

Go to the **Kafka Quotas** > **Quotas** page and view the new production or consumption rate limit.

NOTE

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

----End

12.3 Deleting a Quota

Scenario

Delete a quota when it is no longer needed.

NOTE

This function is unavailable for single-node instances.

Procedure

Step 1 Log in to the management 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 **Middleware** > **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 quota to be deleted, click **Delete**.

Step 7 Click **Yes**. The **Background Tasks** page is displayed. If the status of the quota deletion task is **Successful**, the quota has been deleted.

----End

12.4 Viewing Quota Monitoring

View the usage of user quotas, client quotas, and topic quotas of each broker.

NOTE

This function is unavailable for single-node instances.

Procedure

Step 1 Log in to the management 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 **Middleware** > **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.

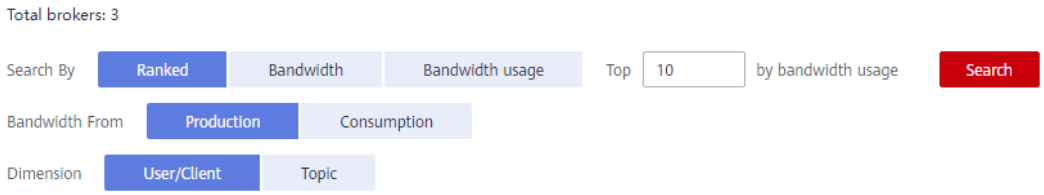
Step 5 In the navigation pane, choose **Kafka Quotas** > **Quota Monitoring**.

Step 6 Set quota monitoring parameters.

Table 12-3 Quota monitoring parameters

Parameter	Description
Search By	Specify a method of calculating rate limits. • **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 source of rate limit calculation. • **Production**: Count production rate limits. • **Consumption**: Count consumption rate limits.
Dimension	Specify the dimension of rate limit calculation. • **User/Client**: Count user/client rate limits. • **Topic**: Count topic rate limits.

Figure 12-2 Quota monitoring parameters



Step 7 Click **Search** to view the usage of user quotas, client quotas, and topic quotas of each broker.

----End

13 Modifying Kafka Parameters

Scenario

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.
- This function is not available for single-node instances.

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 management 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 **Middleware** > **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 13-2** and **Table 13-1**. Parameters of v2.3.0/v2.7 instances are described in **Table 13-3** and **Table 13-4**.

Table 13-1 Dynamic parameters (v1.1.0 instances)

Parameter	Description	Value Range	Default Value
auto.create.groups.enable	Whether to automatically create consumer groups. You can modify this parameter on the console only for instances created on or after April 25, 2023. For instances created before April 25, 2023, 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. The unit is minute. This is a static parameter for instances created before May 1, 2023.	1440–30240	20160

Table 13-2 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

Parameter	Description	Value Range	Default Value
message.max.bytes	Maximum length of a single message, in bytes.	0–10,485,760	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**	true
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	600,000
log.retention.hours	Duration (in hours) for retaining a log file. 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	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	The maximum session timeout (in ms) 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	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 or ciphertext access is used. This parameter cannot be modified for instances created before September 15, 2023.	true/false	true

Parameter	Description	Value Range	Default Value
num.partitions	The default number of partitions configured for each automatically created topic.	1 ~ 200	3
group.min.session.timeout.ms	The minimum session timeout (in ms) for consumers. A shorter timeout enables quicker failure detection but results in more frequent consumer heartbeating, which can overwhelm broker resources.	6000–300,000	6000

Table 13-3 Dynamic parameters (2.3.0/2.7 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, in bytes.	0–10,485,760	10,485,760
auto.create.groups.enable	Whether to automatically create consumer groups. You can modify this parameter on the console only for instances created on or after April 25, 2023. For instances created before April 25, 2023, 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

Parameter	Description	Value Range	Default Value
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**	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. The unit is minute. This is a static parameter for instances created before May 1, 2023.	1440–30240	20160

Table 13-4 Static parameters (2.3.0/2.7 instances)

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	600,000
log.retention.hours	Duration (in hours) for retaining a log file. 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	72
group.max.session.timeout.ms	The maximum session timeout (in ms) 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	1,800,000
default.replication.factor	The default number of replicas configured for an automatically created topic.	1–3	3

Parameter	Description	Value Range	Default Value
allow.everyone.if.no.acl.found	When this parameter is set to **true**, all users have access to resources without ACL. This parameter is displayed only when SASL is enabled for the instance or ciphertext access is used. This parameter cannot be modified for instances created before September 15, 2023.	true/false	true
num.partitions	The default number of partitions configured for each automatically created topic.	1 ~ 200	3
group.min.session.timeout.ms	The minimum session timeout (in ms) for consumers. A shorter timeout enables quicker failure detection but results in more frequent consumer heartbeating, which can overwhelm broker resources.	6000–300,000	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

14 Diagnosing Message Accumulation

Scenario

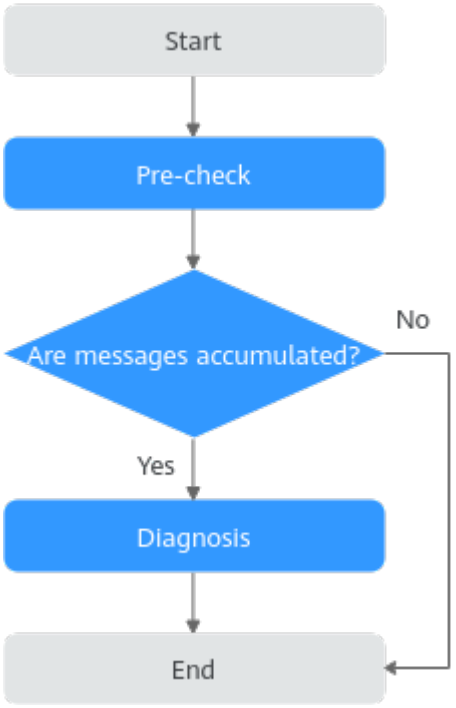
DMS for Kafka provides the message accumulation diagnosis function on the console. If there are accumulated messages, you can learn about the possible causes, affected partitions or brokers, and handling suggestions of the accumulation by viewing the diagnosis record.

Prerequisites

- A Kafka instance has been created, and a consumer group is consuming messages in non-assign mode.
- When a consumer group is being diagnosed, other consumer groups and other topics in the consumer group cannot be diagnosed.

Process Flow

Figure 14-1 Process of accumulation diagnosis



Step 1: Pre-check

Step 1 Log in to the management 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 **Middleware** > **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 left navigation pane, choose **Analysis & Diagnosis** > **Accumulation Diagnosis**.

Step 6 Select the consumer groups and topics to be diagnosed, and click **Pre-check**.

If the check is successful, the message "Pre-checked" is displayed in the upper part of the page, and the check results of the memory usage, CPU usage, partition subscription relationships, accumulated messages, and traffic burst are displayed.

If there is no risk shown in the **Accumulated Messages** area, message accumulation diagnosis cannot be performed. If there are any risks in the **Accumulated Messages** area and the consumer group is not consuming message in the assign mode, you can perform **message accumulation diagnosis**.

----End

Step 2: Diagnosis

Step 1 Click **Start Diagnosis**. In the **Diagnosis Records** area, a record in the **Diagnosing** state is displayed.

If the status changes to **Successful**, the diagnosis is complete.

Step 2 Locate the row that contains the target diagnosis record, and click **View Details**. The **Diagnosis Details page** is displayed.

Step 3 View the number of abnormal, failed, and normal items in the upper part of the page. In the **Diagnosed Item** area, click an abnormal item, such as **Rebalancing**, and view the possible causes, affected partitions or brokers, and handling suggestions.

----End

15 Quotas

What Is a Quota?

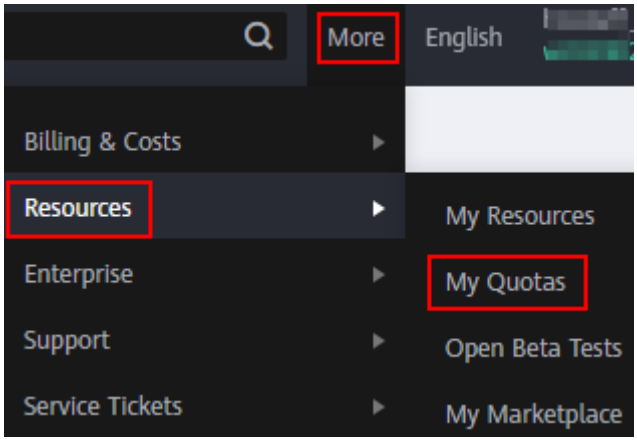
A quota is a limit on the quantity or capacity of a certain type of service resources that you can use, for example, the maximum number of Kafka instances that you can create.

If a quota cannot meet your needs, apply for a higher quota.

How Do I View My Quota?

1. Log in to the management console.
2. Click in the upper left corner to select a region and a project.
3. In the upper right corner of the page, choose **Resources** > **My Quotas**.
   The **Quotas** page is displayed.

   Figure 15-1 My Quotas



4. 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 management 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**.

16 Monitoring

16.1 Viewing Metrics

Scenario

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 management 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 **Middleware** > **Distributed Message Service (for Kafka)** to open the console of DMS for Kafka.

Step 4 View the instance metrics using either of the following methods:

- In the row containing the desired instance, click **View Metric**. On the Cloud Eye console, view the metrics of the instance, brokers, topics, and consumer groups. Metric data is reported to Cloud Eye every minute.
- Click the desired Kafka instance to view its details. In the navigation pane, choose **Monitoring** view. On the displayed page, view the metrics of the instance, brokers, topics, and consumer groups. Metric data is reported to Cloud Eye every minute.

----End

16.2 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 16-1 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	0–1800	Kafka instance	1 minute
current_topics	Topics	Number of created topics in the instance Unit: count	0–1800	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 16-2 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 on the EIP console 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 on the EIP console 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 **NOTE** This metric is supported by instances purchased in April 2020 or later.	• **1**: alive • **0**: not alive	Kafka instance broker	1 minute
broker_connections	Connections	Total number of TCP connections on the Kafka broker Unit: count **NOTE** This metric is supported by instances purchased in April 2020 or later.	0–65,535	Kafka instance broker	1 minute
broker_cpu_usage	CPU Usage	CPU usage of the Kafka VM Unit: % **NOTE** This metric is supported by instances purchased in April 2020 or later.	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 **NOTE** This metric is supported for instances purchased in June 2020 or later.	> 0	Kafka instance broker	1 minute

Metric ID	Metric Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
broker_disk_write_await	Average Disk Write Time	Average time for each disk I/O write in the monitoring period Unit: ms **NOTE** This metric is supported for instances purchased in June 2020 or later.	> 0	Kafka instance broker	1 minute
broker_total_bytes_in_rate	Inbound Traffic	Inbound traffic per second Unit: byte/s **NOTE** This metric is supported for instances purchased in June 2020 or later.	0–1,000,000,000	Kafka instance broker	1 minute
broker_total_bytes_out_rate	Outbound Traffic	Outbound traffic per second Unit: byte/s **NOTE** This metric is supported for instances purchased in June 2020 or later.	0–1,000,000,000	Kafka instance broker	1 minute
broker_disk_read_rate	Disk Read Speed	Read traffic on the disk Unit: byte/s, KB/s, MB/s, or GB/s **NOTE** This metric is supported for instances purchased on or after May 16, 2022.	≥ 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 **NOTE** This metric is supported for instances purchased on or after May 16, 2022.	≥ 0	Kafka instance broker	1 minute

Metric ID	Metric Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
network_bandwidth_usage	Network Bandwidth Usage	Network bandwidth usage Unit: % **NOTE** • This metric is supported for instances purchased since July 9 2023. • For instances purchased before July 9 2023, this metric is supported for brokers if they are added since July 9 2023.	0–100	Kafka instance broker	1 minute

Topic Metrics

Table 16-3 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 **Scope** 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 **Scope** 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 **Scope** 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 **Scope** 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 **Scope** 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 **Scope** 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 **Scope** is set to **Partition monitoring** on the **By Topic** tab page.	≥ 0	Topic in a Kafka instance	1 minute

Consumer Group Metrics

Table 16-4 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.	0~30000000	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.	0~30000000	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.	0~30000000	Consumer group of a Kafka instance	1 minute

Smart Connect Metrics

Table 16-5 Smart Connect metrics

Metric ID	Metric Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
kafka_wait_synchronize_data	Kafka Data to Sync	Data to synchronize in the Kafka migration task Unit: count	≥ 0	Smart Connect task of a Kafka instance	1 minute
kafka_synchronize_rate	Kafka Data Synced per Minute	Data synchronized per minute in the Kafka migration task Unit: count	≥ 0	Smart Connect task of a Kafka instance	1 minute

Metric ID	Metric Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
task_status	Task Status	Status of the current task	• **0**: abnormal • **1**: normal	Smart Connect task of a Kafka instance	1 minute
message_delay	Message Delay	Time elapsed between when a message is sent from the source and received by the target Unit: ms	≥ 0	Smart Connect task of a Kafka instance	1 minute

NOTE

- A Smart Connect task that bidirectionally copies Kafka data is split into two tasks for monitoring: *Smart Connect task name*_**source_0** and *Smart Connect task name*_**source_1**.
- If all messages in a topic have aged before the next synchronization, there is no Kafka data to be synchronized. However, since the Kafka data synchronization metric uses the offset value that contains aged data, **Kafka Data Synced per Minute** will display the number of aged messages.

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
connector_task	Smart Connect task of a Kafka instance

16.3 Configuring Alarm Rules

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 16-6 Kafka instance metrics to configure alarm rules for

Metric ID	Metric	Alarm Policy	Description	Handling Suggestion
broker_disk_usage	Disk Capacity Usage	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	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 instance **bandwidth or the number of brokers**. For details, see **Modifying Instance Specifications**.
broker_memory_usage	Memory Usage	Alarm threshold: original value > 90% Number of consecutive periods: 3 Alarm severity: critical	Memory usage of the Kafka VM.	Modify the instance **bandwidth or the number of brokers**. For details, see **Modifying Instance Specifications**.

Metric ID	Metric	Alarm Policy	Description	Handling Suggestion
current_partitions	Partitions	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 instance **bandwidth or the number of brokers**, or split the service to multiple instances. For details about how to modify the instance bandwidth or the number of brokers, see **Modifying Instance Specifications**.
broker_cpu_usage	CPU Usage	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 instance **bandwidth or the number of brokers**. For details, see **Modifying Instance Specifications**.

Metric ID	Metric	Alarm Policy	Description	Handling Suggestion
group_msgs	Accumulated Messages	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	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 management 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 **Middleware** > **Distributed Message Service (for Kafka)** to open the console of DMS for Kafka.

Step 4 In the row containing the desired instance, click **View Metric**.

You are redirected to the Cloud Eye console page displaying metrics of the selected instance.

Step 5 Hover the mouse pointer over a metric and click  to create an alarm rule for the metric.

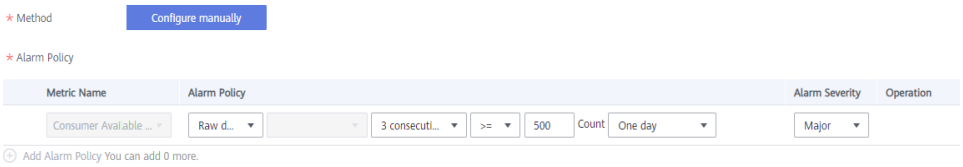
Step 6 Specify the alarm details.

For more information about creating alarm rules, see **Creating an Alarm Rule**.

1. Set the alarm name and description.
2. Specify the alarm policy and alarm severity.

   As shown in the following figure, if the original disk capacity usage exceeds 85% for three consecutive periods, an alarm is generated. If the alarm is not handled on time, an alarm notification is sent.

   Figure 16-1 Setting the alarm policy and alarm severity

3. Set the alarm notification configurations. If you enable **Alarm Notification**, set the validity period, notification object, and trigger condition.
4. Click **Create**.

----End

17 Auditing

17.1 Operations Logged by CTS

With Cloud Trace Service (CTS), you can record operations associated with DMS for Kafka for later query, audit, and backtrack operations.

Table 17-1 DMS for Kafka operations that can be recorded by CTS

Operation	Resource Type	Trace Name
Successfully creating an order for creating an instance	kafka	createDMSInstanceOrderSuccess
Successfully creating an instance	kafka	createDMSInstanceTaskSuccess
Failing to create an order for creating an instance	kafka	createDMSInstanceOrderFailure
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

Operation	Resource Type	Trace Name
Deleting multiple instance tasks at a time	kafka	batchDeleteDMSInstanceTask
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 modify an instance order	kafka	modifyDMSInstanceOrderSuc-cess
Failing to submit a request to modify an instance order	kafka	modifyDMSInstanceOrderFailure
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	resetDMSInstancePasswordSuc-cess
Failing to submit a request to reset instance password	kafka	resetDMSInstancePasswordFai-lure

Operation	Resource Type	Trace Name
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
Failing to restart an instance	kafka	restartDMSInstanceTaskFailure
Successfully submitting a request to restart multiple instances at a time	kafka	batchRestartDMSInstanceSuccess
Successfully restarting multiple instances at a time	kafka	batchRestartDMSInstanceTaskSuccess
Failing to submit a request to restart multiple instances at a time	kafka	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 enabling Smart Connect	kafka	createConnectorTaskSuccess

Operation	Resource Type	Trace Name
Successfully creating a Smart Connect task	kafka	createConnectorSinkTaskSuccess
Failing to enable Smart Connect	kafka	createConnectorTaskFailure
Failing to create a Smart Connect task	kafka	createConnectorSinkTaskFailure
Successfully freezing an instance	kafka	freezeDMSInstanceTaskSuccess
Failing to freeze an instance	kafka	freezeDMSInstanceTaskFailure
Successfully unfreezing an instance	kafka	unfreezeDMSInstanceTaskSuccess
Failing to unfreeze an instance	kafka	unfreezeDMSInstanceTaskFailure
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
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 resetting the consumer offset	kafka	Kafka_reset_consumer_offsetSuccess
Failing to reset the consumer offset	kafka	Kafka_reset_consumer_offsetFailure
Successfully creating a user	kafka	createUserSuccess

Operation	Resource Type	Trace Name
Failing to create a user	kafka	createUserFailure
Successfully deleting a user	kafka	deleteUserSuccess
Failing to delete a user	kafka	deleteUserFailure
Successfully updating user policies	kafka	updateUserPoliciesTaskSuccess
Failing to update user policies	kafka	updateUserPoliciesTaskFailure

17.2 Querying Real-Time Traces

Scenarios

After you enable CTS and the management tracker is created, CTS starts recording operations on cloud resources. After a data tracker is created, the system starts recording operations on data in OBS buckets. CTS stores operation records generated in the last seven days.

This section describes how to query and export operation records of the last seven days on the CTS console.

- **Viewing Real-Time Traces in the Trace List of the New Edition**
- **Viewing Real-Time Traces in the Trace List of the Old Edition**

Constraints

- Traces of a single account can be viewed on the CTS console. Multi-account traces can be viewed only on the **Trace List** page of each account, or in the OBS bucket or the **CTS/system** log stream configured for the management tracker with the organization function enabled.
- You can only query operation records of the last seven days on the CTS console. To store operation records for more than seven days, you must configure an OBS bucket to transfer records to it. Otherwise, you cannot query the operation records generated seven days ago.
- After performing operations on the cloud, you can query management traces on the CTS console 1 minute later and query data traces on the CTS console 5 minutes later.

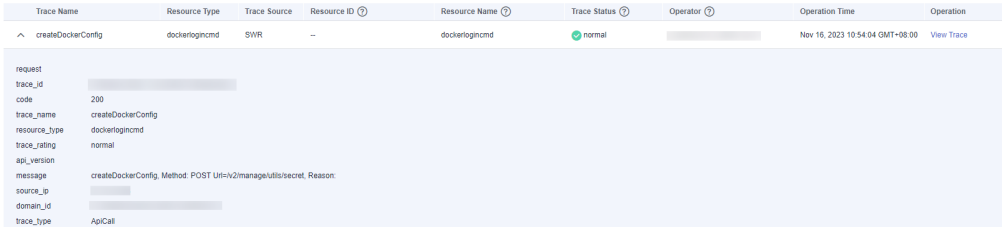
Viewing Real-Time Traces in the Trace List of the New Edition

1. Log in to the management console.

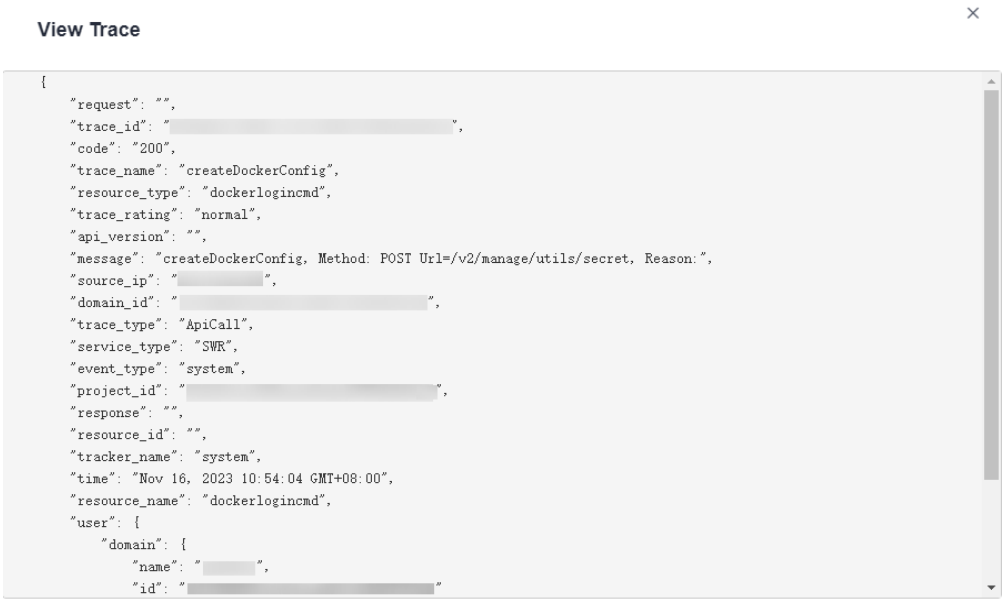
2. Click  in the upper left corner and choose **Management & GovernanceManagement & Deployment** > **Cloud Trace Service**. The CTS console is displayed.
3. Choose **Trace List** in the navigation pane on the left.
4. On the **Trace List** page, use advanced search to query traces. You can combine one or more filters.
   - **Trace Name**: Enter a trace name.
   - **Trace ID**: Enter a trace ID.
   - **Resource Name**: Enter a resource name. If the cloud resource involved in the trace does not have a resource name or the corresponding API operation does not involve the resource name parameter, leave this field empty.
   - **Resource ID**: Enter a resource ID. Leave this field empty if the resource has no resource ID or if resource creation failed.
   - **Trace Source**: Select a cloud service name from the drop-down list.
   - **Resource Type**: Select a resource type from the drop-down list.
   - **Operator**: Select one or more operators from the drop-down list.
   - **Trace Status**: Select **normal**, **warning**, or **incident**.
     - **normal**: The operation succeeded.
     - **warning**: The operation failed.
     - **incident**: The operation caused a fault that is more serious than the operation failure, for example, causing other faults.
   - Time range: Select **Last 1 hour**, **Last 1 day**, or **Last 1 week**, or specify a custom time range.
5. On the **Trace List** page, you can also export and refresh the trace list, and customize the list display settings.
   - Enter any keyword in the search box and click  to filter desired traces.
   - Click **Export** to export all traces in the query result as an .xlsx file. The file can contain up to 5000 records.
   - Click  to view the latest information about traces.
   - Click  to customize the information to be displayed in the trace list. If **Auto wrapping** is enabled (), excess text will move down to the next line; otherwise, the text will be truncated. By default, this function is disabled.
6. For details about key fields in the trace structure, see **Trace Structure**section "Trace References" > "Trace Structure" and **Example Traces**section "Trace References" > "Example Traces".
7. (Optional) On the **Trace List** page of the new edition, click **Go to Old Edition** in the upper right corner to switch to the **Trace List** page of the old edition.

Viewing Real-Time Traces in the Trace List of the Old Edition

1. Log in to the management console.
2. Click in the upper left corner and choose **Management & GovernanceManagement & Deployment** > **Cloud Trace Service**. The CTS console is displayed.
3. Choose **Trace List** in the navigation pane on the left.
4. Each time you log in to the CTS console, the new edition is displayed by default. Click **Go to Old Edition** in the upper right corner to switch to the trace list of the old edition.
5. Set filters to search for your desired traces. The following filters are available:
   - **Trace Type**, **Trace Source**, **Resource Type**, and **Search By**: Select a filter from the drop-down list.
     - If you select **Resource ID** for **Search By**, specify a resource ID.
     - If you select **Trace name** for **Search By**, specify a trace name.
     - If you select **Resource name** for **Search By**, specify a resource name.
   - **Operator**: Select a user.
   - **Trace Status**: Select **All trace statuses**, **Normal**, **Warning**, or **Incident**.
   - Time range: You can query traces generated during any time range in the last seven days.
   - Click **Export** to export all traces in the query result as a CSV file. The file can contain up to 5000 records.
6. Click **Query**.
7. On the **Trace List** page, you can also export and refresh the trace list.
   - Click **Export** to export all traces in the query result as a CSV file. The file can contain up to 5000 records.
   - Click to view the latest information about traces.
8. Click on the left of a trace to expand its details.



9. Click **View Trace** in the **Operation** column. The trace details are displayed.



10. For details about key fields in the trace structure, see **Trace Structure**section "Trace References" > "Trace Structure" and **Example Traces**section "Trace References" > "Example Traces".
11. (Optional) On the **Trace List** page of the old edition, click **New Edition** in the upper right corner to switch to the **Trace List** page of the new edition.

