Distributed Message Service for Kafka

User Guide

Issue	06
Date	2023-03-09

HUAWEI CLOUD COMPUTING TECHNOLOGIES CO., LTD.

Trademarks and Permissions

Notice

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

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

Contents

1 Service Overview

1.1 What Is DMS for Kafka?

Apache Kafka is distributed message middleware that features high throughput, data persistence, horizontal scalability, and stream data processing. It adopts the publish-subscribe pattern and is widely used for log collection, data streaming, online/offline system analytics, and real-time monitoring.

Distributed Message Service (DMS) for Kafka is a message queuing service that uses the open-source Apache Kafka. It provides Kafka premium instances with isolated computing, storage, and bandwidth resources. DMS for Kafka allows you to apply and configure resources based on service requirements. It can be used out of the box and frees you from deployment and O&M so that you can focus on the agile development of your applications.

Readers' Guide

This documentation introduces DMS for Kafka and its differences from Apache Kafka. You will learn about the detailed information about the specifications, console operations, and client access to instances of DMS for Kafka.

For more information about the basic knowledge of Kafka or technical details about creating and retrieving messages, please go to the **official Apache Kafka website**.

1.2 Product Advantages

DMS for Kafka provides easy-to-use message queuing based on Apache Kafka. Services can be quickly migrated to the cloud without any change, reducing maintenance and usage costs.

- Rapid deployment

  Simply set instance information on the DMS for Kafka console, submit your order, and a complete Kafka instance will be automatically created and deployed.

- Service migration without modifications

  DMS for Kafka is compatible with open-source Kafka APIs and supports all message processing functions of open-source Kafka.

  If your application services are developed based on open-source Kafka, you can easily migrate them to DMS for Kafka after specifying a few authentication configurations.

  NOTE

  Kafka instances are compatible with Apache Kafka v1.1.0, v2.3.0, and v2.7. Keep the client and server versions the same.

- Security

  Operations on Kafka instances are recorded and can be audited. Messages can be encrypted before storage.

  In addition to Simple Authentication and Security Layer (SASL) authentication, Virtual Private Clouds (VPCs) and security groups also provide security controls on network access.

- Data reliability

  Kafka instances support data persistence and replication. Messages can be synchronously or asynchronously replicated between replicas and flushed to disk.

- High availability

  Kafka runs in clusters, enabling failover and fault tolerance so that services can run smoothly.

  Kafka instance brokers can be deployed across AZs to enhance service availability.

- Simple O&M

  The cloud service platform provides a whole set of monitoring and alarm services, eliminating the need for 24/7 attendance. Kafka instance metrics are monitored and reported, including the number of partitions, topics, and accumulated messages. You can configure alarm rules and receive SMS or email notifications on how your services are running in real time.

- Massive accumulation and scaling

  Kafka features high scalability because it runs in a distributed system, or cluster. You can configure up to 100 partitions for a topic. The storage space can be also expanded. This means that billions of messages can be accumulated, suitable for scenarios requiring high concurrency, high performance, and large-scale access.

- Flexible specifications

  You can customize the bandwidth and storage space for the instance and the number of partitions and replicas for topics in the instance.

1.3 Application Scenarios

Kafka is popular message-oriented middleware that features highly reliable, asynchronous message delivery. It is widely used for transmitting data between different systems in many industries, including enterprise application, payment, telecommunications, e-commerce, social networking, instant messaging, video, Internet of Things, and Internet of Vehicle.

Asynchronous Communication

Non-core or less important messages are sent asynchronously to receiving systems, so that the main service process is not kept waiting for the results of other systems, allowing for faster responses.

For example, Kafka can be used to send a notification email and SMS message after a user has registered with a website, providing fast responses throughout the registration process.

Figure 1-1 Serial registration and notification

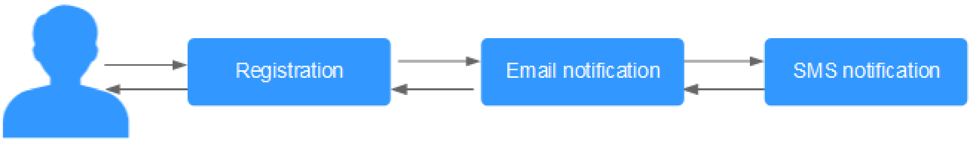
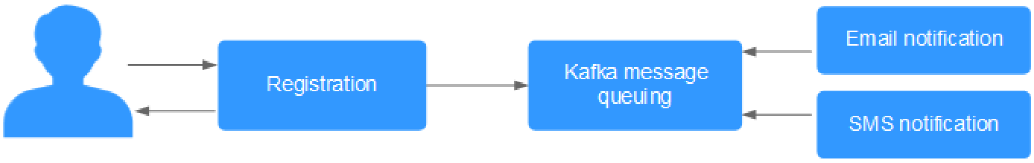


Figure 1-2 Asynchronous registration and notification using message queues

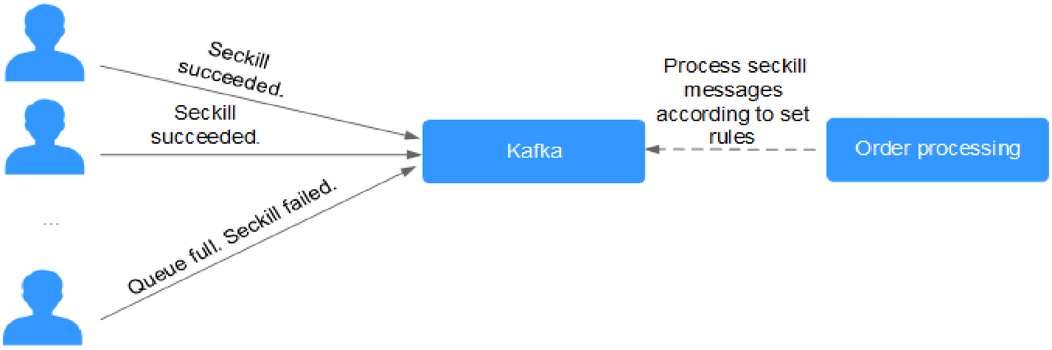


Traffic Control

In e-commerce systems or large-scale websites, there is a processing capability gap between upstream and downstream systems. Traffic bursts from upstream systems with high processing capabilities may have a large impact on downstream systems with lower processing capabilities. For example, online sales promotions involve a huge amount of traffic flooding into e-commerce systems. Kafka provides a three-day buffer by default for hundreds of millions of messages, such as orders and other information. In this way, message consumption systems can process the messages during off-peak periods.

In addition, flash sale traffic bursts originating from frontend systems can be handled with Kafka, keeping the backend systems from crashing.

Figure 1-3 Traffic burst handling using Kafka


Seckill succeeded.
Seckill succeeded.
...
Queue full. Seckill failed.
Kafka
Process seckill messages according to set rules
Order processing

Log Synchronization

In large-scale service systems, logs of different applications are collected for quick troubleshooting, full-link tracing, and real-time monitoring.

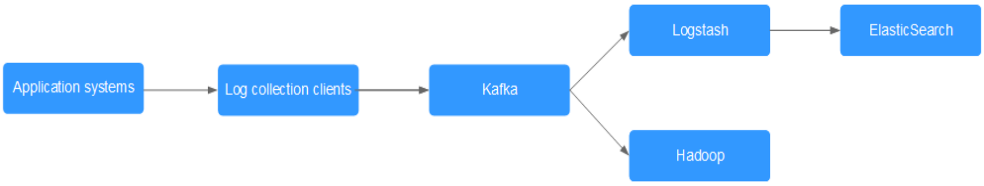
Kafka is originally designed for this scenario. Applications asynchronously send log messages to message queues over reliable transmission channels. Other components can read the log messages from message queues for further analysis, either in real time or offline. In addition, Kafka can collect key log information to monitor applications.

Log synchronization involves three major components: log collection clients, Kafka, and backend log processing applications.

1. The log collection clients collect log data from a user application service and asynchronously send the log data in batches to Kafka clients.

   Kafka clients receive and compress messages in batches. This only has a minor impact on the service performance.
2. Kafka persists logs.
3. Log processing applications, such as Logstash, subscribe to messages in Kafka and retrieve log messages from Kafka. Then, the messages are searched for by file search services or delivered to big data applications such as Hadoop for storage and analysis.

Figure 1-4 Log synchronization process



NOTE

Logstash is for log analytics, Elasticsearch is for log search, and Hadoop is for big data analytics. They are all open-source tools.

1.4 Specifications

Kafka Instance Specifications

Kafka instances are compatible with open-source Kafka v1.1.0, v2.3.0, and v2.7. The instance specifications are represented by the ECS flavor and the number of brokers. Available options are kafka.2u4g.cluster, kafka.4u8g.cluster, kafka.8u16g.cluster, kafka.12u24g.cluster, and kafka.16u32g.cluster.

NOTE

In the following table, transactions per second (TPS) are calculated assuming that the size of a message is 1 KB.

Table 1-1 Kafka instance specifications

Flavor	Brokers	Maximum TPS per Broker	Maximum Partitions per Broker	Maximum Consumer Groups per Broker	Maximum Client Connections per Broker	Storage Space
kafka.2u4g.cluster	3–30	30,000	250	20	2000	300 GB–300,000 GB
kafka.4u8g.cluster	3–30	100,000	500	100	4000	300 GB–600,000 GB
kafka.8u16g.cluster	3–30	150,000	1000	150	4000	300 GB–900,000 GB
kafka.12u24g.cluster	3–30	200,000	1500	200	4000	300 GB–900,000 GB
kafka.16u32g.cluster	3–30	250,000	2000	200	4000	300 GB–900,000 GB

Flavor Selection

- kafka.2u4g.cluster with 3 brokers

  Recommended for up to 3000 client connections, 60 consumer groups, and 100,000 TPS
- kafka.4u8g.cluster with 3 brokers

  Recommended for up to 10,000 client connections, 300 consumer groups, and 300,000 TPS
- kafka.8u16g.cluster with 3 brokers

  Recommended for up to 20,000 client connections, 600 consumer groups, and 600,000 TPS
- kafka.12u24g.cluster with 3 brokers

  Recommended for up to 20,000 client connections, 600 consumer groups, and 900,000 TPS
- kafka.16u32g.cluster with 3 brokers

  Recommended for up to 20,000 client connections, 600 consumer groups, and 1,200,000 TPS

Storage Space Selection

Kafka instances support multi-replica storage. The storage space is consumed by all replicas. When creating an instance, specify its storage space based on the expected service message size and the number of replicas.

For example, if the estimated message size is 100 GB, the disk capacity must be at least: 100 GB x Number of replicas + 100 GB (reserved space).

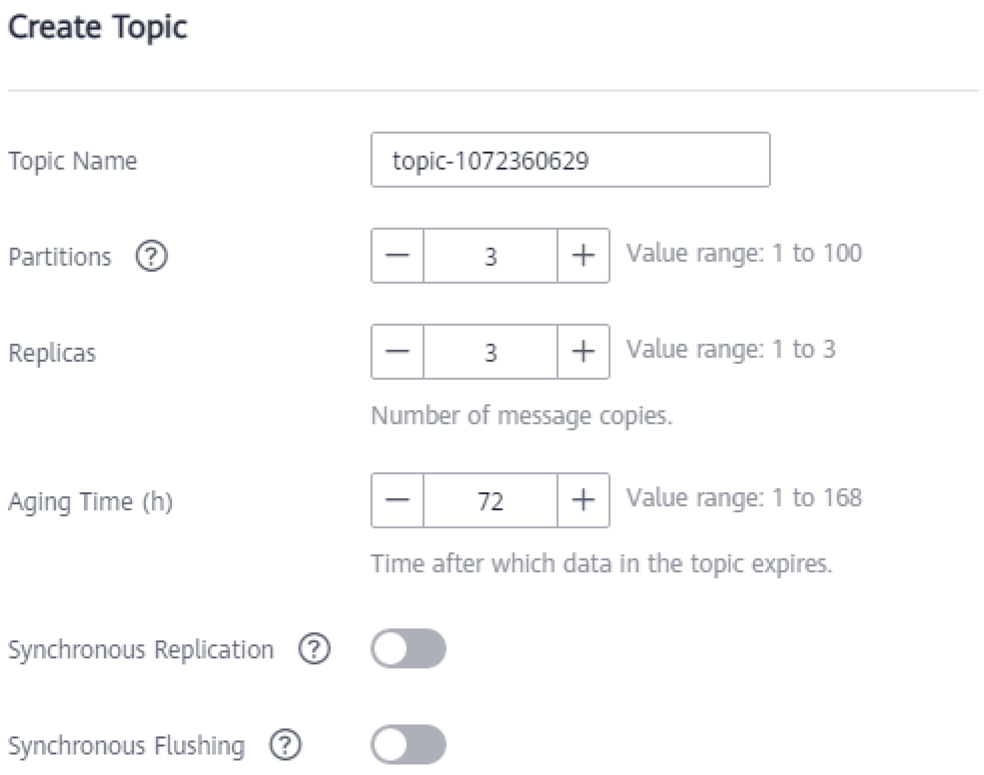
The storage space can be expanded as your service grows.

Topic Quantity

There are limits on the topic quantity and the aggregate number of partitions in the topics. When the partition quantity limit is reached, you can no longer create topics.

The number of topics is related to the maximum number of partitions allowed (see **Figure 1-5**) and the specified number of partitions in each topic (see **Table 1-1**).

Figure 1-5 Setting the number of partitions



Create Topic

Topic Name: topic-1072360629

Partitions: 3 Value range: 1 to 100

Replicas: 3 Value range: 1 to 3
Number of message copies.

Aging Time (h): 72 Value range: 1 to 168
Time after which data in the topic expires.

Synchronous Replication

Synchronous Flushing

The maximum number of partitions allowed for an instance with kafka.2u4g.cluster and 3 brokers is 750.

- If the number of partitions of each topic in the instance is 3, the maximum number of topics is 750/3 = 250.
- If the number of partitions of each topic in the instance is 1, the maximum number of topics is 750/1 = 750.

1.5 Comparing Kafka, RabbitMQ, and RocketMQ

Feature	RocketMQ	Kafka	RabbitMQ
Priority queue	Not supported	Not supported	Supported. It is recommended that the priority be set to 0–10.
Delayed queue	Supported	Not supported	Supported
Dead letter queue	Supported	Not supported	Supported
Message retry	Supported	Not supported	Not supported
Retrieval mode	Pull-based and push-based	Pull-based	Pull-based and push-based
Message broadcasting	Supported	Supported	Supported
Message tracking	Supported	Supports offset and timestamp tracking.	Not supported. Once a message retrieval has been acknowledged, RabbitMQ will be notified that the message can be deleted.
Message accumulation	Supported	Supports higher accumulation performance than RabbitMQ thanks to high throughput.	Supported
Persistence	Supported	Supported	Supported
Message tracing	Supported	Not supported	Supported by the firehose feature or the rabbitmq_tracing plugin. However, rabbitmq_tracing reduces performance and should be used only for troubleshooting.
Message filtering	Supported	Supported	Not supported, but can be encapsulated.
Multi-tenancy	Supported	Not supported	Supported
Multi-protocol	Compatible with RocketMQ.	Only supports Apache Kafka.	RabbitMQ is based on AMQP and supports MQTT and STOMP.

Feature	RocketMQ	Kafka	RabbitMQ
Multi-language	Supports clients in multiple programming languages.	Kafka is written in Scala and Java and supports clients in multiple programming languages.	RabbitMQ is written in Erlang and supports clients in multiple programming languages.
Throttling	Planned	Supports throttling on producer or consumer clients.	Supports credit-based throttling on producers, a mechanism that triggers protection from within.
Ordered message delivery	Message order is maintained within a queue.	Supports partition-level FIFO.	Not supported. Supports FIFO only for single-threaded message queuing without advanced features such as delayed queues or priority queues.
Security	Supports SSL authentication.	Supports SSL and SASL authentication and read/write permissions control.	Similar to Kafka.
Transactional messages	Supported	Supported	Supported

1.6 Comparing DMS for Kafka and Open-Source Kafka

DMS for Kafka is compatible with open-source Kafka and has customized and enhanced Kafka features. In addition to the advantages of open-source Kafka, DMS for Kafka provides more reliable and useful features.

Table 1-2 Differences between DMS for Kafka and open-source Kafka

Category	Item	DMS for Kafka	Open-source Kafka
Ease of use	Readily available	Instances can be created intuitively within minutes and used right out of the box with visualized operations and real-time monitoring.	Preparing server resources and installing and configuring the software is time-consuming and prone to mistakes.
	APIs	Instances can be managed easily by calling RESTful APIs.	N/A

Category	Item	DMS for Kafka	Open-source Kafka
Costs	On-demand use	Multiple specifications are available to suit different needs. The instance broker quantity and disk space can be expanded without downtime.	Expenses are incurred for setting up a message service and occupying underlying resources.
	Fully managed	Services are readily available without requiring additional hardware resources or expenses.	Users must prepare hardware resources and set up the service by themselves, and bear high usage and maintenance costs.
Proven success	Mature	DMS has been deployed in many cloud products and proven successful in large e-commerce events. It is also used in the clouds of carrier-grade customers across the world, and meets strict carrier-grade reliability standards. DMS closely follows up with community updates to continuously fix known open-source vulnerabilities and add support for new features.	Using open-source software requires lengthy self-development and verification and has had few successful cases.
	Feature-rich	While maintaining 100% open-source compatibility, DMS further optimizes open-source code to improve performance and reliability, and provides message querying, and many other features.	Functionality is limited and requires self-development.
Reliability	Highly available	DMS supports cross-AZ deployment to improve reliability. In addition, automatic fault detection and alarms ensure reliable operations of key services.	High availability requires self-development or open-source code implementation, which are costly and cannot guarantee reliability.

Category	Item	DMS for Kafka	Open-source Kafka
	Simple O&M	O&M is entirely transparent to tenants with a full set of monitoring and alarm functions. O&M personnel will be informed of any exceptions, eliminating the need for 24/7 attending.	Users need to develop and optimize O&M functions, especially alarm notification functions. Otherwise, manual attendance is required.
	Secure	DMS uses VPC isolation, disk encryption, and SSL channel encryption.	Security must be hardened by users themselves.

1.7 Notes and Constraints

This section describes the notes and constraints on DMS for Kafka.

Instance

Table 1-3 Instance notes and constraints

Item	Notes and Constraints
Kafka ZooKeeper	Kafka clusters are managed using ZooKeeper. Opening ZooKeeper may cause misoperations and service losses. Currently, ZooKeeper is used only within Kafka clusters and does not provide services externally.
Version	● The service version can be 1.1.0, 2.3.0, or 2.7. Kafka instances cannot be upgraded once they are created. ● Clients later than version 0.10 are supported. Use a version that is consistent with the service version.
Logging in to the VM where the Kafka brokers reside	Not supported
Storage	● The storage space can be expanded but cannot be reduced. ● You can expand the storage space up to 20 times.
Bandwidth or broker quantity	The bandwidth and broker quantity can be increased but cannot be decreased.
VPC, subnet, and AZ	After an instance is created, its VPC, subnet, and AZ cannot be modified.

Item	Notes and Constraints
Kerberos authentication	Not supported

Topic

Table 1-4 Topic notes and constraints

Item	Notes and Constraints
Total number of topic partitions	The total number of topic partitions is related to the instance specifications. For details, see **Specifications**. Kafka manages messages by partition. If there are too many partitions, message creation, storage, and retrieval will be fragmented, affecting the performance and stability. If the total number of partitions of topics reaches the upper limit, you cannot create more topics.
Number of partitions in a topic	Based on the open-source Kafka constraints, the number of partitions in a topic can be increased but cannot be decreased.
Topic quantity	The topic quantity is related to the total number of topic partitions and number of partitions in each topic. For details, see **Specifications**.
Automatic topic creation	Supported. 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.
Synchronous replication	If a topic has only one replica, synchronous replication cannot be enabled.
Replica quantity	Single-replica topics are not recommended. 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 to use a topic with only one replica.

Item	Notes and Constraints
Aging time	The value of the **log.retention.hours** parameter takes effect only if the aging time has not been set for the topic. 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.
Batch importing and exporting topics	Batch export is supported, but batch import is not supported.
Topic name	If a topic name starts with a special character, for example, an underscore (_) or a number sign (#), monitoring data cannot be displayed.
Delay queues	Not supported

Consumer Group

Table 1-5 Consumer group notes and constraints

Item	Notes and Constraints
Creating consumer groups, consumers, and producers	Consumer groups, consumers, and producers are generated automatically when you use the instance.
Resetting the consumer offset	Messages may be retrieved more than once after the offset is reset.
Consumer group name	If a consumer group name starts with a special character, for example, an underscore (_) or a number sign (#), monitoring data cannot be displayed.

Message

Table 1-6 Message notes and constraints

Item	Notes and Constraints
Message size	The maximum length of a message is 10 MB. If the length exceeds 10 MB, the production fails.

User

Table 1-7 User notes and constraints

Item	Notes and Constraints
Number of users	A maximum of 20 SASL_SSL users can be created for a Kafka instance.

1.8 Related Services

- Cloud Trace Service (CTS)

  CTS generates traces to provide you with a history of operations performed on cloud service resources. The traces include operation requests sent using the management console or open APIs, as well as the operation results. You can view all generated traces to query, audit, and backtrack performed operations.

  For details about the operations recorded by CTS, see **Operations Logged by CTS**.

- Virtual Private Cloud (VPC)

  Kafka instances run in VPCs and use the IP addresses and bandwidth of VPC. Security groups of VPCs enhance the security of network access to the Kafka instances.

- Elastic Cloud Server (ECS)

  An ECS is a basic computing unit that consists of vCPUs, memory, OS, and EVS disks. Kafka instances run on ECSs. A broker corresponds to an ECS.

- Elastic Volume Service (EVS)

  EVS provides block storage services for ECSs. All Kafka data, such as messages, metadata, and logs, is stored in EVS disks.

- Cloud Eye

  Cloud Eye is an open platform that provides monitoring, alarm reporting, and alarm notification for your resources in real time.

  NOTE

  The values of all Kafka instance metrics are reported to Cloud Eye every minute.

- Elastic IP (EIP)

  The EIP service provides independent public IP addresses and bandwidth for Internet access. Kafka instances bound with EIPs can be accessed over public networks.

- Tag Management Service (TMS)

  TMS is a visualized service for fast and unified cross-region tagging and categorization of cloud services.

  Tags facilitate Kafka instance identification and management.

- Data Encryption Workshop (DEW)

  When creating a Kafka instance, you can specify whether to enable disk encryption. Enabling disk encryption improves data security. Disk encryption depends on DEW.

1.9 Basic Concepts

DMS for Kafka of the cloud service platform uses Kafka as the message engine. This chapter presents explanations of basic concepts of Kafka.

Topic

A topic is a category for messages. Messages are created, retrieved, and managed in the form of topics.

Topics adopt the publish-subscribe pattern. Producers publish messages into topics. One or more consumers subscribe to the messages in the topics. The producers and consumers are not directly linked to each other.

Producer

A producer publishes messages into topics. The messages are then delivered to other systems or modules for processing as agreed.

Consumer

A consumer subscribes to messages in topics and processes the messages. For example, a monitoring and alarm platform (a consumer) subscribing to log messages in certain topics can identify alarm logs and then send SMS or email alarm notifications.

Broker

A broker is a Kafka process in a Kafka cluster. Each process runs on a server, so a broker includes the storage, bandwidth, and other server resources.

Partition

A topic is divided into partitions. Messages are distributed to multiple partitions to achieve scalability and fault tolerance.

Replica

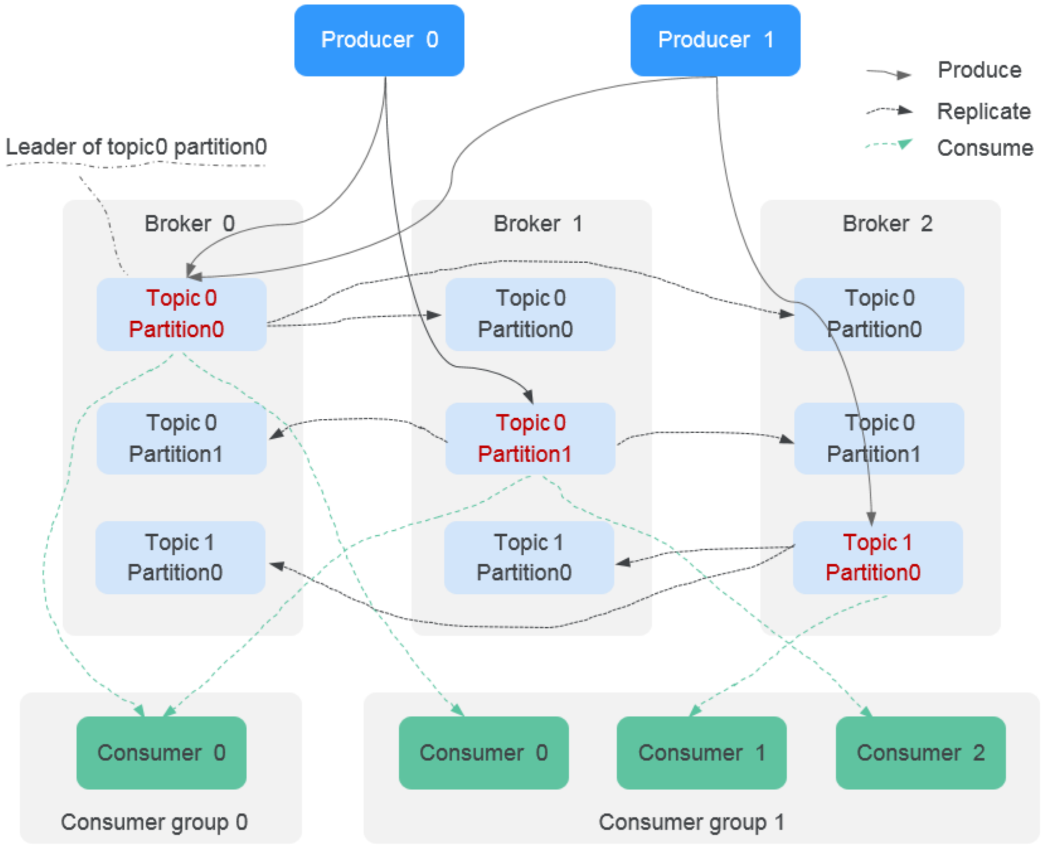
A replica is a redundant copy of a partition in a topic. Each partition can have one or more replicas, enabling message reliability.

Messages in each partition are fully replicated and synchronized, preventing data loss if one replica fails.

Each partition has one replica as the leader which handles the creation and retrievals of all messages. The rest replicas are followers which replicate the leader.

Topics and partitions are logical concepts, while replicas and brokers are physical concepts. The following diagram shows the relationships between partitions, brokers, and topics in messages streaming.

Figure 1-6 Kafka message streaming



Aging Time

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.

1.10 Permissions Management

You can use Identity and Access Management (IAM) to manage DMS for Kafka permissions and control access to your resources. IAM provides identity authentication, permissions management, and access control.

You can create IAM users for your employees, and assign permissions to these users on a principle of least privilege (PoLP) basis to control their access to specific resource types. For example, you can create IAM users for software developers and assign specific permissions to allow them to use Kafka instance resources but prevent them from being able to delete resources or perform any high-risk operations.

If your account does not require individual IAM users for permissions management, skip this section.

IAM can be used free of charge. You pay only for the resources in your account.

For more information, see **IAM Service Overview**.

NOTE

Permissions policies of DMS for Kafka are based on DMS. Therefore, when assigning permissions, select DMS permissions policies.

DMS for Kafka Permissions

By default, new IAM users do not have any permissions assigned. To assign permissions to these new users, add them to one or more groups, and attach permissions policies or roles to these groups.

DMS for Kafka is a project-level service deployed and accessed in specific physical regions. When assigning DMS for Kafka permissions to a user group, specify region-specific projects where the permissions will take effect. If you select **All projects**, the permissions will be granted for all region-specific projects. When accessing DMS for Kafka, the users need to switch to a region where they have been authorized to use this service.

You can grant permissions by using roles and policies.

- Roles: A type of coarse-grained authorization mechanism that provides only a limited number of service-level roles. When using roles to grant permissions, you also need to assign dependency roles. However, roles are not an ideal choice for fine-grained authorization and secure access control.
- Policies: A type of fine-grained authorization mechanism that defines permissions required to perform operations on specific cloud resources under certain conditions. This mechanism allows for more flexible policy-based authorization for securer access control. For example, you can grant DMS for Kafka users only the permissions for managing instances. Most policies define permissions based on APIs. For the API actions supported by DMS for Kafka, see **Permissions Policies and Supported Actions**.

Table 1-8 lists all the system-defined roles and policies supported by DMS for Kafka.

Table 1-8 System-defined roles and policies supported by DMS for Kafka

Role/Policy Name	Description	Type	Dependency
DMS FullAccess	Administrator permissions for DMS. Users granted these permissions can perform all operations on DMS.	System-defined policy	None
DMS UserAccess	Common user permissions for DMS, excluding permissions for creating, modifying, deleting, and scaling up instances.	System-defined policy	None

Role/Policy Name	Description	Type	Dependency
DMS ReadOnlyAccess	Read-only permissions for DMS. Users granted these permissions can only view DMS data.	System-defined policy	None
DMS Administrator	Administrator permissions for DMS.	System-defined role	This role depends on the **Tenant Guest** and **VPC Administrator** roles.

NOTE

System-defined policies contain OBS actions. Due to data caching, the policies take effect five minutes after they are attached to a user, user group, or enterprise project.

Table 2 lists the common operations supported by each DMS for Kafka system policy or role. Select the policies or roles as required.

Table 1-9 Common operations supported by each system-defined policy or role of DMS for Kafka

Operation	DMS FullAccess	DMS UserAccess	DMS ReadOnlyAccess
Creating instances	√	×	×
Modifying instances	√	×	×
Deleting instances	√	×	×
Modifying instance specifications	√	×	×
Restarting instances	√	√	×
Querying instance information	√	√	√

Helpful Links

- **What Is IAM?**

- **Creating a User and Granting DMS for Kafka Permissions**
- **Permissions Policies and Supported Actions**

1.11 Billing

DMS for Kafka supports pay-per-use.

Billing Items

DMS for Kafka is billed based on Kafka instance specifications and storage space.

Table 1-10 DMS for Kafka billing

Billing Item	Description
Instance	● Kafka instances are billed based on their ECS flavor and broker quantity. When purchasing an instance, select appropriate ECS flavors and the number of brokers based on service evaluation. **Table 1-11** lists the performance per broker. ● Kafka instances can be billed on a pay-per-use (hourly) basis.
Storage space	● Queues are billed based on the storage space. For each type of instance specification, you can choose the common I/O, high I/O, or ultra-high I/O disk type to meet your service requirements. You can specify the number of replicas. For example, if the disk size required to store message data is 500 GB and there are three replicas, the disk capacity should be at least: 500 GB x 3 = 1500 GB. ● Storage space options in 100 GB increments are described in **Table 1-11**. ● The storage space can be billed on a pay-per-use (hourly) basis.

Table 1-11 Kafka instance specifications

Flavor	Brokers	Maximum TPS per Broker	Maximum Partitions per Broker	Maximum Consumer Groups per Broker	Maximum Client Connections per Broker	Storage Space
kafka.2u4g.cluster	3–30	30,000	250	20	2000	300 GB–300,000 GB
kafka.4u8g.cluster	3–30	100,000	500	100	4000	300 GB–600,000 GB
kafka.8u16g.cluster	3–30	150,000	1000	150	4000	300 GB–900,000 GB
kafka.12u24g.cluster	3–30	200,000	1500	200	4000	300 GB–900,000 GB
kafka.16u32g.cluster	3–30	250,000	2000	200	4000	300 GB–900,000 GB

Billing Modes

Pay-per-use (hourly) mode: More flexible, enabling you to start and stop services anytime. You pay only for what you use. The minimum time unit is one hour. Less than an hour is recorded as an hour.

Changing Configurations

- You can change the number of brokers for a Kafka instance. You will then be billed based on the new specifications immediately after the change.
- You can also change the storage space of Kafka queues. You will be billed based on the new storage space immediately after the storage space increase. Storage space can only be increased, and cannot be decreased. The minimum increment is 100 GB.

2 Getting Started

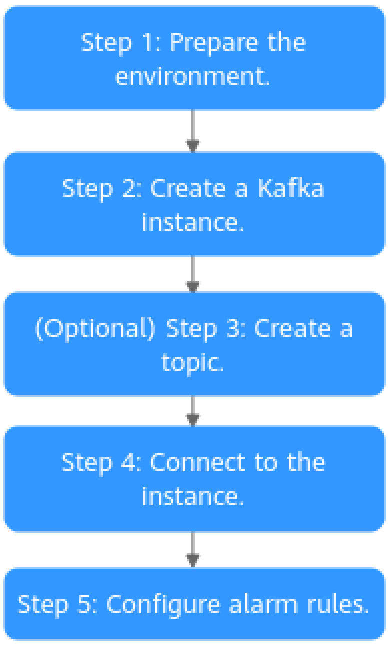
2.1 Introduction

This document provides instructions for getting started with Distributed Message Service (DMS) for Kafka, including creating a Kafka instance on the console and connecting to a Kafka instance through an Elastic Cloud Server (ECS).

You can also **create a Kafka instance by calling an API** and connect to the instance in your service code.

Procedure

Figure 2-1 Procedure for using DMS for Kafka


Step 1: Prepare the environment.
Step 2: Create a Kafka instance.
(Optional) Step 3: Create a topic.
Step 4: Connect to the instance.
Step 5: Configure alarm rules.

1. **Prepare the environment.**

   A Kafka instance runs in a Virtual Private Cloud (VPC). Before creating a Kafka instance, ensure that a VPC is available.

After a Kafka instance is created, download and install the Kafka open-source client on your ECS before creating and retrieving messages.

2. **Create a Kafka instance.**

   When creating an instance, you can choose whether to enable SASL. If SASL is enabled, data is encrypted for transmission, improving data security. The SASL setting can be configured only when you create an instance. After an instance is created, the SASL setting cannot be changed.

3. (Optional) **Create a topic.**

   If automatic topic creation is not enabled during Kafka instance creation, you need to manually create topics for creating and retrieving messages.

4. Connect to the instance.

   You can connect to a Kafka instance with or without SASL.

   - **Without SASL**: Supports private network access and public network access.
   - **With SASL**: Supports private network access and public network access.

5. **Configure alarm rules.**

   Configure alarm rules for a Kafka instance to monitor the service running status.

NOTE

For details about Kafka concepts, see **Basic Concepts**.

2.2 Step 1: Prepare the Environment

VPC

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

Step 1 Before creating a Kafka instance, ensure that a VPC and a subnet are available.

For details, see **Creating a VPC**. If you already have an available VPC and subnet, you do not need to create new ones.

Note the following when creating a VPC and subnet:

- The VPC and the Kafka instance must be in the same region.
- Use the default settings when creating a VPC and subnet.

Step 2 Before creating a Kafka instance, ensure that a security group is available.

For details, see **Creating a Security Group**. If you already have an available security group, you do not need to create a new one.

Note the following when creating a security group:

- Set **Template** to **Custom**.
- To use Kafka instances, add the security group rules described in **Table 2-1**. Other rules can be added based on site requirements.

Table 2-1 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).
Inbound	TCP	9999	0.0.0.0/0	Access Kafka Manager.

NOTE

After a security group is created, it has a default inbound rule that allows communication among ECSs within the security group and a default outbound rule that allows all outbound traffic. If you access your Kafka instance within a VPC, you do not need to add the rules described in **Table 2-1**.

----End

(Optional) EIP

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

For details, see **Assigning an EIP**.

Note the following when creating EIPs:

- The EIPs must be created in the region the Kafka instance is in.
- The number of EIPs must be the same as the number of Kafka instance brokers.
- **The Kafka console cannot identify IPv6 EIPs.**

ECS

Before connecting to a Kafka instance, ensure that you have purchased an ECS, installed the JDK, configured environment variables, and downloaded an open-source Kafka client. The following steps describe how to complete these preparations. A Linux ECS is taken as an example. For more information on how to install JDK and configure the environment variables for a Windows ECS, please search the Internet.

Step 1 Log in to the management console. In the upper left corner, hover the mouse pointer over ☰. Under **Compute**, click **Elastic Cloud Server**, and then create an ECS.

For details, see **Creating an ECS**. If you already have an available ECS, skip this step.

Step 2 Log in to the ECS.

Step 3 Install JDK or JRE, and add the following contents to **.bash_profile** in the home directory to configure the environment variables **JAVA_HOME** and **PATH**. In this command, **/opt/java/jdk1.8.0_151** is the JDK installation path. Change it to the path where you install JDK or JRE.

```
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.

NOTE

Use Oracle JDK instead of ECS's default JDK (for example, OpenJDK), because ECS's default JDK may not be suitable. Obtain Oracle JDK 1.8.111 or later from **Oracle's official website**.

Step 4 Download an open-source Kafka client.

If the version of the Kafka instance is 1.1.0, download the client at **https://archive.apache.org/dist/kafka/1.1.0/kafka_2.11-1.1.0.tgz**.

```
wget https://archive.apache.org/dist/kafka/1.1.0/kafka_2.11-1.1.0.tgz
```

If the version of the Kafka instance is 2.3.0, download the client at **https://archive.apache.org/dist/kafka/2.3.0/kafka_2.11-2.3.0.tgz**.

```
wget https://archive.apache.org/dist/kafka/2.3.0/kafka_2.11-2.3.0.tgz
```

If the version of the Kafka instance is 2.7, download the client at **https://archive.apache.org/dist/kafka/2.7.2/kafka_2.12-2.7.2.tgz**.

```
wget https://archive.apache.org/dist/kafka/2.7.2/kafka_2.12-2.7.2.tgz
```

Step 5 Run the following command to decompress the package:

```
tar -zxf ${kafka_tar}
```

In the preceding command, *kafka_tar* indicates the name of the client package. For example:

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

----End

Follow-Up Procedure

Step 2: Create a Kafka Instance

2.3 Step 2: Create a Kafka Instance

Prerequisites

Ensure that a VPC is available. For details about how to create a VPC, see the *Virtual Private Cloud User Guide*.

If you already have an available VPC, you do not need to create a new one.

Procedure

Step 1 Log in to the Kafka console, and click **Buy Instance** in the upper right corner.

Step 2 Select a billing mode.

Step 3 Select a region closest to your application to reduce latency and accelerate access.

Step 4 Select a project from the drop-down list.

Step 5 Select one AZ or at least three AZs.

Step 6 Specify the instance name and the enterprise project.

Step 7 Configure the following instance parameters:

1. **Version**: Kafka v1.1.0, v2.3.0, 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. For **Brokers**, specify the broker quantity.

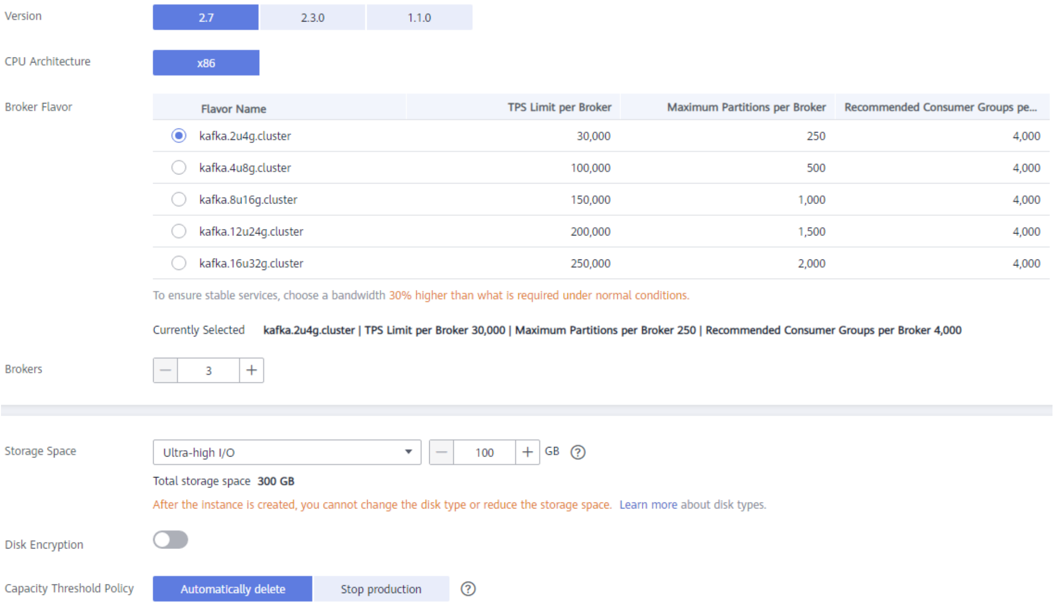
   Maximum number of partitions per broker x Number of brokers = Maximum number of partitions of an instance. If the total number of partitions of all topics exceeds the maximum number of partitions allowed for an instance, topic creation will fail.
4. **Storage Space**: Disk type and total disk space for 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 space).

   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.
5. **Disk Encryption**: Specify whether to enable disk encryption. Enabling disk encryption improves data security. Disk encryption depends on Data Encryption Workshop (DEW). If you enable disk encryption, select a KMS key. If no key is available, click **View KMS Keys** to go to the DEW console and create one. **This parameter cannot be modified once the instance is created.**
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 2-2 Buying a Kafka instance



Version: 2.7 | 2.3.0 | 1.1.0

CPU Architecture: x86

Broker Flavor

	Flavor Name	TPS Limit per Broker	Maximum Partitions per Broker	Recommended Consumer Groups pe...
●	kafka.2u4g.cluster	30,000	250	4,000
○	kafka.4u8g.cluster	100,000	500	4,000
○	kafka.8u16g.cluster	150,000	1,000	4,000
○	kafka.12u24g.cluster	200,000	1,500	4,000
○	kafka.16u32g.cluster	250,000	2,000	4,000

To ensure stable services, choose a bandwidth 30% higher than what is required under normal conditions.

Currently Selected **kafka.2u4g.cluster | TPS Limit per Broker 30,000 | Maximum Partitions per Broker 250 | Recommended Consumer Groups per Broker 4,000**

Brokers: 3

Storage Space: Ultra-high I/O 100 GB

Total storage space **300 GB**

After the instance is created, you cannot change the disk type or reduce the storage space. Learn more about disk types.

Disk Encryption

Capacity Threshold Policy: Automatically delete | Stop production

Step 8 Configure the instance network parameters.

1. Select a VPC and a subnet.

   NOTE

   After the Kafka instance is created, its VPC and subnet cannot be changed.

2. 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 9 Configure the username and password for logging in to Kafka Manager. **The Kafka Manager username cannot be changed once an instance is created.**

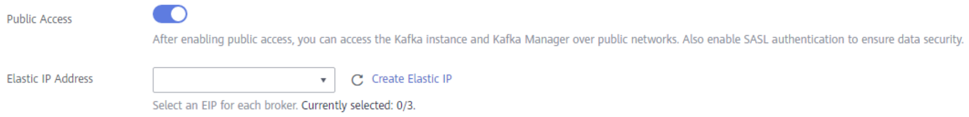
Kafka Manager is an open-source tool for managing Kafka clusters. After a Kafka instance is created, you can go to the instance details page to obtain the address for logging in to Kafka Manager. In Kafka Manager, you can view the monitoring statistics and broker information of your Kafka clusters.

Step 10 Click **More Settings** to configure more parameters.

1. Configure **Public Access**.

   Public access is disabled by default. You can enable it or keep it disabled as required. After public access is enabled, configure an IPv4 EIP for each broker.

Figure 2-3 Configuring public access



Public Access ●

After enabling public access, you can access the Kafka instance and Kafka Manager over public networks. Also enable SASL authentication to ensure data security.

Elastic IP Address

Select an EIP for each broker. Currently selected: 0/3.

2. Configure **Kafka SASL_SSL**.

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

   Kafka SASL_SSL is disabled by default. You can enable or disable it as required. **This setting cannot be changed after the instance is created.** If you want to use a different setting, you must create a new instance.

   If you enable **Kafka SASL_SSL**, 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.

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 many cloud resources of the same type, you can use tags to classify cloud resources by dimension (for example, by usage, owner, or environment).

   – If you have created 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 entering **Tag key** and **Tag value**.

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

5. Enter a description of the instance.

Step 11 Click **Buy**.

Step 12 Confirm the instance information.

Step 13 Return to the **Kafka Premium** page and check whether the 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 fails to be created, view **Instance Creation Failures**. Delete the instance and create another instance. If the instance creation fails again, contact customer service.

NOTE

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

----End

Follow-Up Procedure

(Optional) Step 3: Create a Topic

2.4 (Optional) Step 3: Create 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 is enabled, this step is optional. The system automatically creates a topic when a message is created. This topic has the following default settings: 3 partitions, 3 replicas, aging time 72 hours, and synchronous replication and flushing disabled.

The following describes three methods to manually create a topic.

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

Method 1: Creating a Topic on the Console

Step 1 Log in to the Kafka console, and select the region where the Kafka instance is located.

Step 2 Click a Kafka instance.

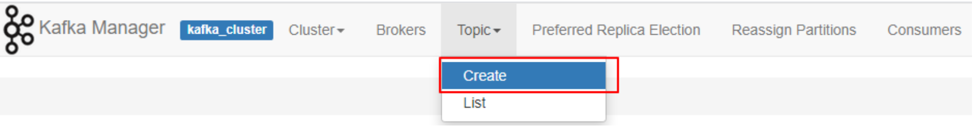
Step 3 On the **Topics** tab page, click **Create Topic**.

Step 4 Enter the topic name, specify other parameters, and click **OK**.

----End

Method 2: Creating a Topic on Kafka Manager

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



NOTICE

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

Method 3: Create 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, an underscore (_) or a number sign (#), monitoring data cannot be displayed.

- 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 **Connecting to an Instance with SASL**.

  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
     ```

Follow-Up Procedure

Step 4: Connect to a Kafka Instance to Create and Retrieve Messages

2.5 Step 4: Connect to a Kafka Instance to Create and Retrieve Messages

2.5.1 Connecting to an Instance Without SASL

This section describes how to connect to a Kafka instance in a private or public network using a CLI, without using SASL certificates.

Private network access and public network access differ only in the connection IP addresses and ports. For private network access, use port 9092. For public network access, use port 9094.

The following describes only the procedure for public network access. For private network access, replace the IP addresses with the actual ones.

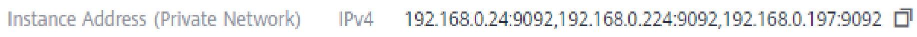
NOTE

Each Kafka broker allows a maximum of 1000 connections from each IP address by default. Excess connections will be rejected. You can change the limit by **modifying the Kafka parameters**.

Prerequisites

- You have correctly configured security group rules. For details, see **Table 2-1**.
- 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.

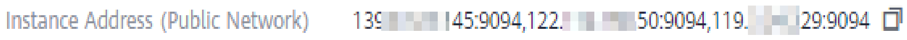
    Figure 2-4 Kafka instance connection addresses for intra-VPC access without SASL

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

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

    Figure 2-5 Kafka instance connection addresses for public access without SASL

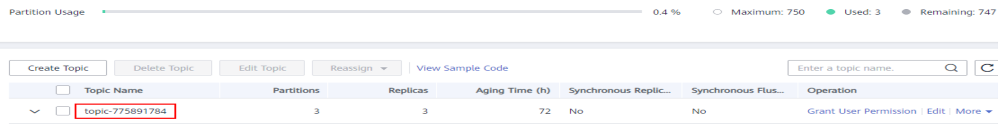
    

    Instance Address (Public Network) 139[redacted]145:9094,122.[redacted]50:9094,119.[redacted]29:9094

- If automatic topic creation is not enabled for the Kafka instance, obtain the topic name.

  You can obtain the name of the topic created in **(Optional) Step 3: Create a Topic** on the **Topics** tab page of the instance.

  Figure 2-6 Viewing the topic name

  Partition Usage 0.4 % ○ Maximum: 750 ● Used: 3 ● Remaining: 747

	Topic Name	Partitions	Replicas	Aging Time (h)	Synchronous Replic...	Synchronous Flus...	Operation
□	topic-775891784	3	3	72	No	No	Grant User Permission | Edit | More

- You have purchased an ECS, installed the JDK, configured the environment variables, and downloaded a Kafka client. For details, see **Step 1: Prepare the Environment**.

Creating Messages

Go to the **/bin** directory of the Kafka client file and run the following command to create messages:

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

Parameter description:

- *{connection-address}*: the address obtained in **Prerequisites**
- *{topic-name}*: the name of the topic created for the Kafka instance

For example, 10.3.196.45:9094, 10.78.42.127:9094, and 10.4.49.103:9094 are the public access addresses of the Kafka instance..

After running the preceding command, you can send a message to the Kafka instance by entering the information as prompted and pressing **Enter**. Contents in each line are sent as a message.

```
[root@ecs-kafka bin]# ./kafka-console-producer.sh --broker-list
10.3.196.45:9094,10.78.42.127:9094,10.4.49.103:9094  --topic topic-demo
>Hello
>DMS
>Kafka!
>^C[root@ecs-kafka bin]#
```

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

Retrieving Messages

Run the following command to retrieve messages:

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

Parameter description:

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

The following is an example:

```
[root@ecs-kafka bin]#  ./kafka-console-consumer.sh --bootstrap-server
10.3.196.45:9094,10.78.42.127:9094,10.4.49.103:9094 --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.

Follow-Up Procedure

You can configure alarm rules for monitoring metrics to receive notifications in a timely manner when instances, brokers, or topics are abnormal.

Step 5: Configure Alarm Rules

2.5.2 Connecting to an Instance with SASL

This section describes how to connect to a Kafka instance in a private or public network using a CLI and SASL certificates.

Private network access and public network access differ only in the connection IP addresses and ports. For intra-VPC access, use port 9093. For public access, use port 9095.

The following describes only the procedure for public network access. For private network access, replace the IP addresses with the actual ones.

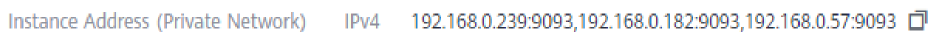
NOTE

Each Kafka broker allows a maximum of 1000 connections from each IP address by default. Excess connections will be rejected. You can change the limit by **modifying the Kafka parameters**.

Prerequisites

- You have correctly configured security group rules. For details, see **Table 2-1**.
- 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.

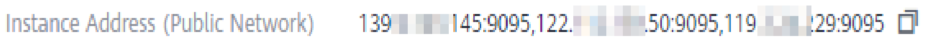
    Figure 2-7 Kafka instance connection addresses for intra-VPC access with SASL

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

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

    Figure 2-8 Kafka instance connection addresses for public access with SASL

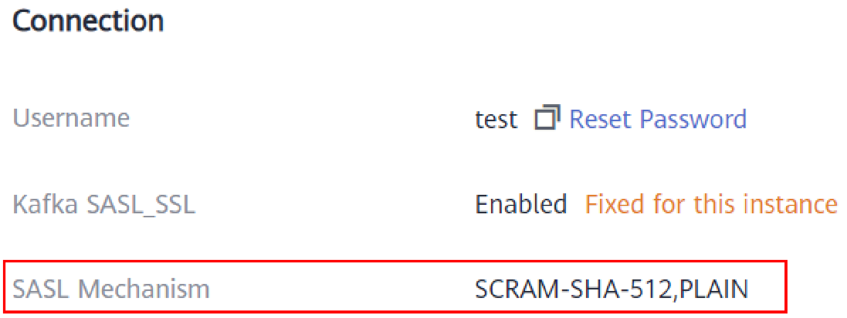
    

    Instance Address (Public Network)   139[redacted]145:9095,122.[redacted].50:9095,119[redacted]29:9095

- The SASL mechanism in use is known.

  In the **Connection** area on the Kafka instance details page, view **SASL Mechanism**. If both SCRAM-SHA-512 and PLAIN are enabled, configure either of them for connections. If **SASL Mechanism** is not displayed, PLAIN is used by default.

Figure 2-9 SASL mechanism in use



Connection

Username test Reset Password

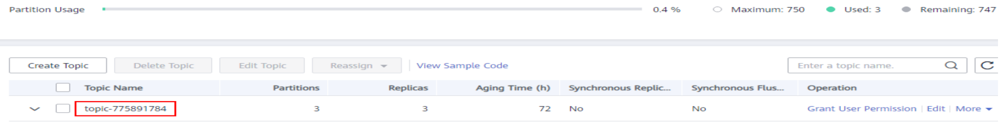
Kafka SASL_SSL Enabled Fixed for this instance

SASL Mechanism SCRAM-SHA-512,PLAIN

- If automatic topic creation is not enabled for the Kafka instance, obtain the topic name.

  You can obtain the name of the topic created in **(Optional) Step 3: Create a Topic** on the **Topics** tab page of the instance.

  Figure 2-10 Viewing the topic name

  Partition Usage 0.4 % ○ Maximum: 750 ● Used: 3 ● Remaining: 747

	Topic Name	Partitions	Replicas	Aging Time (h)	Synchronous Replic...	Synchronous Flus...	Operation
□	topic-775891784	3	3	72	No	No	Grant User Permission | Edit | More

- You have purchased an ECS, installed the JDK, configured the environment variables, and downloaded a Kafka client. For details, see **Step 1: Prepare the Environment**.

Configuring the Configuration File for Message Creation and Retrieval

Step 1 Log in to a Linux ECS.

Step 2 Map hosts to IP addresses in the **/etc/hosts** file on the ECS, 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 3 Download **client.truststore.jks**. On the Kafka console, click the instance. On the instance details page, click **Download** next to **SSL Certificate** in the **Connection** area.

Decompress the package to obtain the client certificate file **client.truststore.jks**.

Step 4 Modify the Kafka CLI configuration file based on the **SASL mechanism**.

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

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

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

Parameter description:

- **username** and **password**: username and password you set when enabling SASL_SSL during Kafka instance creation or when creating a SASL_SSL user.
- **ssl.truststore.location**: path for storing the certificate obtained in **Step 3**.
- **ssl.truststore.password**: server certificate password, which must be set to **dms@kafka** and cannot be changed.
- **ssl.endpoint.identification.algorithm**: whether to verify the certificate domain name. **This parameter must be left blank, which indicates disabling domain name verification**.

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

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

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

Parameter description:

- **username** and **password**: username and password you set when enabling SASL_SSL during Kafka instance creation or when creating a SASL_SSL user.
- **ssl.truststore.location**: path for storing the certificate obtained in **Step 3**.
- **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**.

----End

Creating Messages

Go to the **/bin** directory of the Kafka client file and 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**
- *{topic-name}*: the name of the topic created for the Kafka instance

For example, **10.3.196.45:9095,10.78.42.127:9095,10.4.49.103:9095** are the connection addresses of the Kafka instance.

After running the preceding command, you can send a message to the Kafka instance by entering the information as prompted and pressing **Enter**. Contents in each line are sent as a message.

```
[root@ecs-kafka bin]#./kafka-console-producer.sh --broker-list
10.3.196.45:9095,10.78.42.127:9095,10.4.49.103:9095  --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.

Retrieving Messages

Run the following command to retrieve messages:

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

Parameter description:

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

The following is an example:

```
[root@ecs-kafka bin]#  ./kafka-console-consumer.sh --bootstrap-server 10.xxx.xxx.202:9095,10.xxx.xxx.
197:9095,10.xxx.xxx.68:9095 --topic topic-demo --group order-test --from-beginning --consumer.config ../
config/consumer.properties
Hello
Kafka!
DMS
^CProcessed a total of 3 messages
[root@ecs-kafka bin]#
```

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

Follow-Up Procedure

You can configure alarm rules for monitoring metrics to receive notifications in a timely manner when instances, brokers, or topics are abnormal.

Step 5: Configure Alarm Rules

2.6 Step 5: Configure 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.

NOTE

Approach Upper Limit in the following table indicates whether the performance of the current resource is close to the upper limit. If the performance is close to the upper limit, the performance supported by the current resource is the alarm threshold set in the alarm policy. If the performance continues to increase, services may become abnormal.

Table 2-2 Kafka instance metrics to configure alarm rules for

Metric ID	Metric	Alarm Policy	Metric Description and Alarm Handling
broker_disk_usage	Disk Capacity Usage	Alarm threshold: original value > 80% Number of consecutive periods: 1 Alarm severity: critical	Metric description: disk usage of the Kafka VM. Alarm handling: 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	Metric description: average load of each CPU core of the Kafka VM. Alarm handling: 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	Metric description: memory usage of the Kafka VM. Alarm handling: Modify the instance **bandwidth or the number of brokers**. For details, see **Modifying Instance Specifications**.

Metric ID	Metric	Alarm Policy	Metric Description and Alarm Handling
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	Metric description: number of used partitions in the instance. Alarm handling: 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	Metric description: CPU usage of the Kafka VM. Alarm handling: 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**.
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	Metric description: total number of accumulated messages in all consumer groups of the instance. Alarm handling: Delete idle consumer groups, if any. You can also accelerate message retrieval, for example, by increasing the number of consumers.

Metric ID	Metric	Alarm Policy	Metric Description and Alarm Handling
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	Metric description: number of remaining messages that can be retrieved from the specified topic in the consumer group. Alarm handling: 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 Kafka console, and select the region where the Kafka instance is located.

Step 2 Click next to the Kafka instance name to go to the instance monitoring page of the Cloud Eye console.

Step 3 Hover the mouse pointer over a metric and click + to create an alarm rule for the metric.

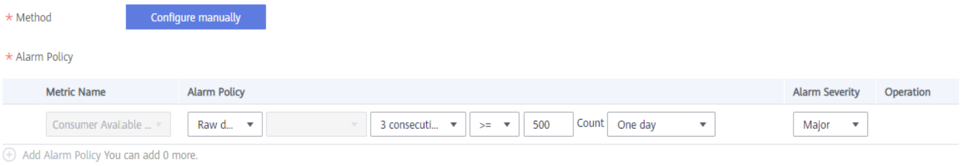
Step 4 Specify the alarm details.

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 2-11 Setting the alarm policy and alarm severity

   * Method: Configure manually

   * Alarm Policy

Metric Name	Alarm Policy							Alarm Severity	Operation
Consumer Available ...	Raw d...		3 consecuti...	>=	500	Count	One day	Major	

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

4. Click **Create**.

----End

3 Permissions Management

3.1 Creating a User and Granting DMS for Kafka Permissions

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

- Create IAM users for employees based on your enterprise's organizational structure. Each IAM user will have their own security credentials for accessing DMS for Kafka resources.
- Manage permissions on a principle of least permissions (PoLP) basis.
- Entrust another account or cloud service to perform efficient O&M on your DMS for Kafka resources.

If your account does not need IAM, skip this section.

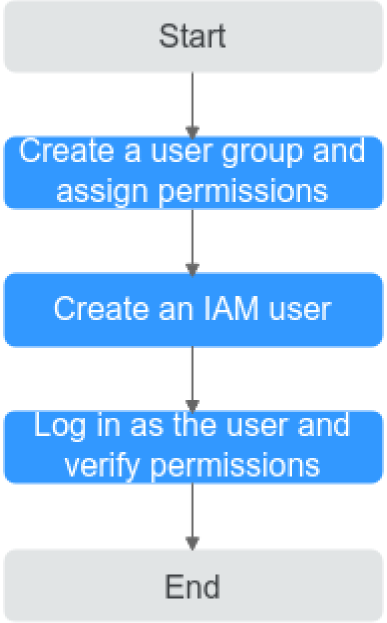
This section describes the procedure for granting permissions (see **Figure 3-1**).

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 3-1 Process for granting DMS for Kafka permissions


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

1. **Create a user group and assign permissions**.

   Create a user group on the IAM console, and assign the **DMS ReadOnlyAccess** policy to the group.

2. **Create an IAM user**.

   Create a user on the IAM console and add the user to the group created in **1**.

3. **Log in** and verify permissions.

   Log in to the management console as the user you created, and verify that the user has the assigned permissions.

   – 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.

3.2 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:delete
               dms:instance:modifyStatus
             "
         ]
      }
   ]
}
```

- 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"
                  ]
            }
      ]
}
```

3.3 DMS for Kafka Resources

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

Table 3-1 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.

3.4 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 3-2 Predefined condition keys of DMS for Kafka

Condition Key	Operator	Description
dms:publicIP	Bool IsNullOrEmpty BoolIfExists	Whether public access is enabled
dms:ssl	Bool IsNullOrEmpty BoolIfExists	Whether SASL is enabled

4 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.

To encrypt the disk, prepare a key in Data Encryption Workshop (DEW) in advance.

Required Resources

Table 4-1 lists the resources required by a Kafka instance.

Table 4-1 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 about how to create a VPC and subnet, see the *Virtual Private Cloud User Guide*.

Resource	Requirement	Operations
Security group	Different Kafka instances can use the same or different security groups. Note the following when creating a security group: - Set **Template** to **Custom**. - To use Kafka instances, add the security group rules described in **Table 4-2**. 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 4-2**.	For details about how to create a security group and configure security group rules, see the *Virtual Private Cloud User Guide*.
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" in *Elastic IP User Guide*.
Key	To encrypt the disk for a Kafka instance, prepare a key in advance. The key must be created in the region your Kafka instance is in.	For details about how to create a key, see "Creating a CMK" in the *Data Encryption Workshop User Guide*.

Table 4-2 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).

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	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).
Inbound	TCP	9999	0.0.0.0/0	Access Kafka Manager.

5 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.
- (Optional) To encrypt the disk, prepare a key in DEW in advance.

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 your application is in.

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

Step 4 Click **Buy Instance** in the upper right corner of the page.

By default, you can create a maximum of 100 Kafka instances for each project. To create more instances, contact customer service to increase your quota.

Step 5 Specify **Billing Mode**, **Region**, **Project**, and **AZ**.

Step 6 Enter an instance name and select an enterprise project.

Step 7 Configure the following instance parameters:

1. **Version**: Kafka v1.1.0, v2.3.0, and v2.7 are supported. v2.7 is recommended. **The version cannot be changed once the instance is created.**

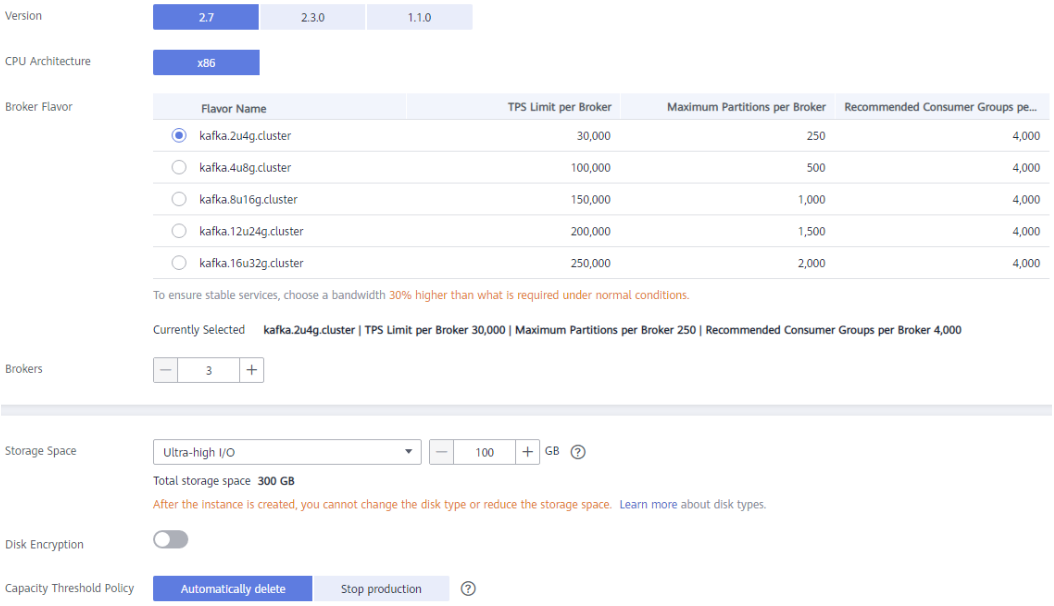
2. **CPU Architecture**: The x86 architecture is supported.
3. **Broker Flavor**: Select broker specifications that best fit your business needs. For **Brokers**, specify the broker quantity.

   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 maximum number of partitions allowed for an instance, topic creation will fail.
4. **Storage Space**: 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 space).

   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.
5. **Disk Encryption**: Specify whether to enable disk encryption. Enabling disk encryption improves data security. Disk encryption depends on Data Encryption Workshop (DEW). If you enable disk encryption, select a KMS key. If no key is available, click **View KMS Keys** to go to the DEW console and create one. **This parameter cannot be modified once the instance is created.**
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 5-1 Buying a Kafka instance



Version: 2.7 | 2.3.0 | 1.1.0

CPU Architecture: x86

Broker Flavor

	Flavor Name	TPS Limit per Broker	Maximum Partitions per Broker	Recommended Consumer Groups pe...
●	kafka.2u4g.cluster	30,000	250	4,000
○	kafka.4u8g.cluster	100,000	500	4,000
○	kafka.8u16g.cluster	150,000	1,000	4,000
○	kafka.12u24g.cluster	200,000	1,500	4,000
○	kafka.16u32g.cluster	250,000	2,000	4,000

To ensure stable services, choose a bandwidth 30% higher than what is required under normal conditions.

Currently Selected kafka.2u4g.cluster | TPS Limit per Broker 30,000 | Maximum Partitions per Broker 250 | Recommended Consumer Groups per Broker 4,000

Brokers: 3

Storage Space: Ultra-high I/O 100 GB

Total storage space 300 GB

After the instance is created, you cannot change the disk type or reduce the storage space. Learn more about disk types.

Disk Encryption

Capacity Threshold Policy: Automatically delete | Stop production

Step 8 Configure the instance network parameters.

- Select a VPC and a subnet.

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

  NOTE

  After the Kafka instance is created, its VPC and subnet cannot be changed.

- Select a security group.

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

Step 9 Configure the username and password for logging in to Kafka Manager. **The Kafka Manager username cannot be changed once an instance is created.**

Kafka Manager is an open-source tool for managing Kafka clusters. After a Kafka instance is created, you can go to the instance details page to obtain the address for logging in to Kafka Manager. In Kafka Manager, you can view the monitoring statistics and broker information of your Kafka clusters.

Step 10 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.

2. Configure **Kafka SASL_SSL**.

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

   Kafka SASL_SSL is disabled by default. You can enable or disable it as required. **This setting cannot be changed after the instance is created.** If you want to use a different setting, you must create a new instance.

If you enable **Kafka SASL_SSL**, 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.

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 many cloud resources of the same type, you can use tags to classify them by dimension (for example, use, owner, or environment).

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

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

5. Enter a description of the instance.

Step 11 Click **Buy**.

Step 12 Confirm the instance information, and click **Submit**.

Step 13 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 fails to be created, view **Instance Creation Failures**. Delete the instance by referring to **Deleting an Instance** and create another instance. If the instance creation fails again, contact customer service.

NOTE

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

----End

6 Accessing a Kafka Instance

6.1 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.

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

NOTE

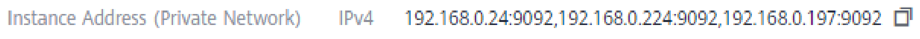
Prerequisites

- Security group rules have been correctly configured.

  To access a Kafka instance with SASL disabled, configure correct security group rules. For details about security group configuration requirements, see **Table 4-2**.

- 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.

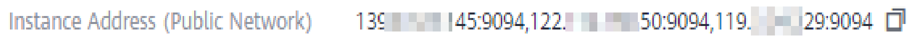
    Figure 6-1 Kafka instance connection addresses for intra-VPC access without SASL

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

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

    Figure 6-2 Kafka instance connection addresses for public access without SASL

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

- If automatic topic creation is not enabled for the Kafka instance, **create a topic** before connecting to the instance.

- Kafka CLI **v1.1.0**, **v2.3.0**, or **v2.7.2** is available. Ensure that the Kafka instance and the CLI are of 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)**. For intra-VPC access, use **Instance Address (Private Network)**.
- *{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 **10.3.196.45:9094,10.78.42.127:9094,10.4.49.103:9094**. 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
10.3.196.45:9094,10.78.42.127:9094,10.4.49.103:9094  --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)**. For intra-VPC access, use **Instance Address (Private Network)**.
- *{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 an underscore (_) or a number sign (#), the monitoring data cannot be displayed.

Example:

```
[root@ecs-kafka bin]#  ./kafka-console-consumer.sh --bootstrap-server
10.3.196.45:9094,10.78.42.127:9094,10.4.49.103:9094 --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

6.2 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.

This section describes how to use an open-source Kafka client to access a Kafka instance if SASL has been enabled for the instance.

NOTE

Each Kafka broker allows a maximum of 1000 connections from each IP address by default. Excess connections will be rejected. You can change the limit by referring to **Modifying Kafka Parameters**.

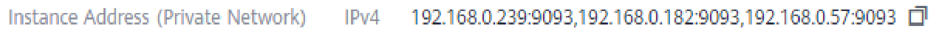
Prerequisites

- Security group rules have been correctly configured.

  To access a Kafka instance with SASL enabled, configure correct security group rules. For details about security group configuration requirements, see **Table 4-2**.

- 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 6-3 Kafka instance connection addresses for intra-VPC access with SASL

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

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

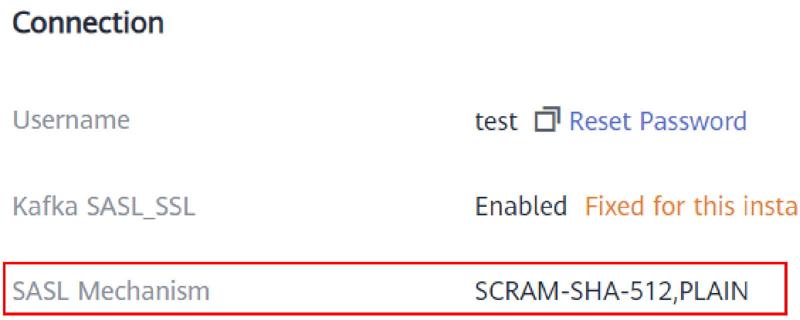
    Figure 6-4 Kafka instance connection addresses for public access with SASL

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

- The SASL mechanism in use is known.

  In the **Connection** area on the Kafka instance details page, view **SASL Mechanism**. If both SCRAM-SHA-512 and PLAIN are enabled, configure either of them for connections. If **SASL Mechanism** is not displayed, PLAIN is used by default.

  Figure 6-5 SASL mechanism in use

  Connection

Username	test Reset Password
Kafka SASL_SSL	Enabled Fixed for this instance
SASL Mechanism	SCRAM-SHA-512,PLAIN

- 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 are of 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

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

Parameter description:

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

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

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

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

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

  Parameter description:

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

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

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

Step 5 Run the following command to create messages:

```
./kafka-console-producer.sh --broker-list ${connection-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)**. For intra-VPC access, use **Instance Address (Private Network)**.
- *{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 **10.3.196.45:9095,10.78.42.127:9095,10.4.49.103:9095**.

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
10.3.196.45:9095,10.78.42.127:9095,10.4.49.103:9095  --topic topic-demo --producer.config ../config/
producer.properties
```

```
>Hello
>DMS
>Kafka!
>^C[root@ecs-kafka bin]#
```

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

Step 6 Run the following command to retrieve messages:

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

Parameter description:

- *{connection-address}*: the address obtained in **Prerequisites**. For public access, use **Instance Address (Public Network)**. For intra-VPC access, use **Instance Address (Private Network)**.
- *{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 an underscore (_) or a number sign (#), the monitoring data cannot be displayed.

Example:

```
[root@ecs-kafka bin]#  ./kafka-console-consumer.sh --bootstrap-server
10.3.196.45:9095,10.78.42.127:9095,10.4.49.103:9095 --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

6.3 Connecting to Kafka Manager and Viewing Kafka Information

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.

Prerequisites

You have correctly configured the security group rules. For details, see **Preparing Required Resources**.

Logging In to Kafka Manager

Step 1 (Optional) 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.

Kafka Manager is deployed in active/standby mode, so **Manager Address (Private Network)** or **Manager Address (Public Network)** may show two addresses. The port number is 9999.

Step 3 Enter the Kafka Manager address in the web browser in the Windows ECS.

If public access is enabled, enter the Kafka Manager address in the address bar of the local browser. 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.

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

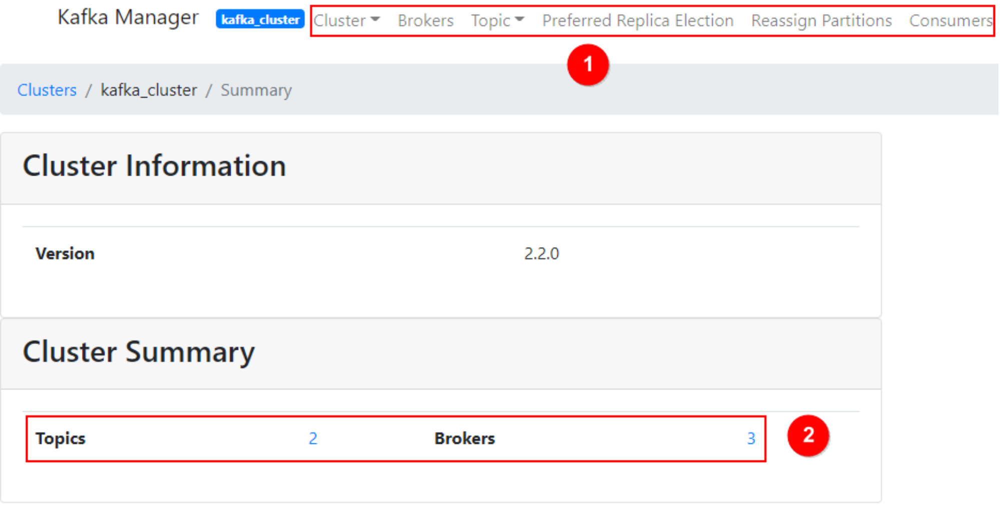
In Kafka Manager, you can view the monitoring statistics and broker information of your Kafka clusters.

- Information about clusters

  Click **Clusters** to view the information about clusters. **Figure 6-6** shows an example of the storage configuration.

  - 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 6-6 Information about clusters



Cluster Information

Version	2.2.0

Cluster Summary

Topics	2	**Brokers**	3

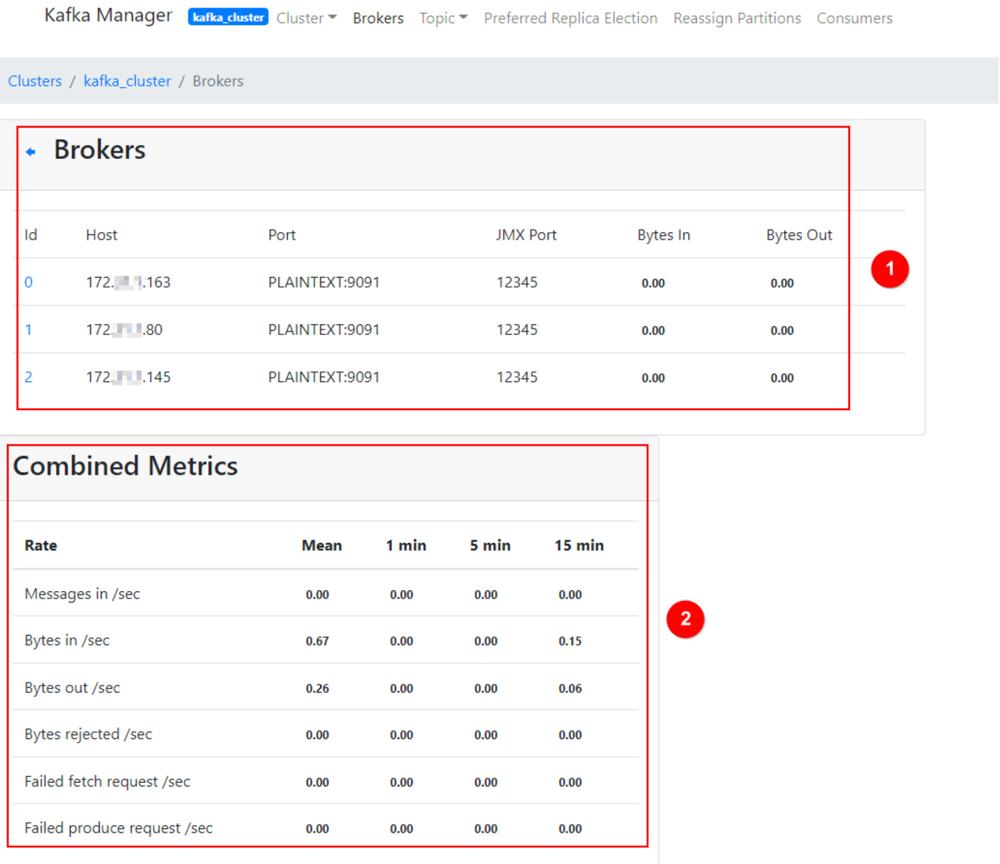
- Combined information about all brokers of a cluster

  This page shows statistics of brokers of a cluster. **Figure 6-7** 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 6-7 Viewing the combined information about all brokers in a cluster



Brokers

Id	Host	Port	JMX Port	Bytes In	Bytes Out
0	172.[redacted].163	PLAINTEXT:9091	12345	0.00	0.00
1	172.[redacted].80	PLAINTEXT:9091	12345	0.00	0.00
2	172.[redacted].145	PLAINTEXT:9091	12345	0.00	0.00

Combined Metrics

Rate	Mean	1 min	5 min	15 min
Messages in /sec	0.00	0.00	0.00	0.00
Bytes in /sec	0.67	0.00	0.00	0.15
Bytes out /sec	0.26	0.00	0.00	0.06
Bytes rejected /sec	0.00	0.00	0.00	0.00
Failed fetch request /sec	0.00	0.00	0.00	0.00
Failed produce request /sec	0.00	0.00	0.00	0.00

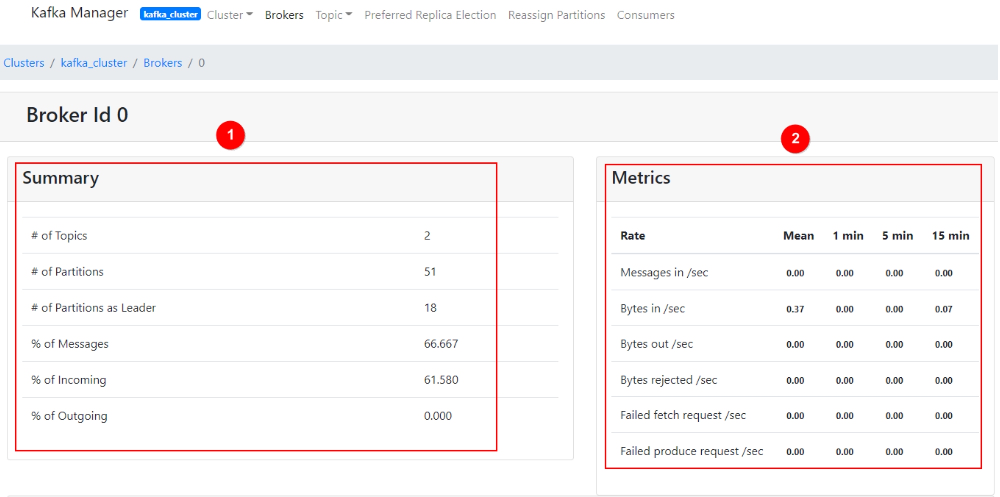
- Information about a specific broker

  Click the ID of a broker to view its statistics. **Figure 6-8** 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 6-8 Viewing information about a broker



Broker Id 0

Summary

# of Topics	2
# of Partitions	51
# of Partitions as Leader	18
% of Messages	66.667
% of Incoming	61.580
% of Outgoing	0.000

Metrics

Rate	Mean	1 min	5 min	15 min
Messages in /sec	0.00	0.00	0.00	0.00
Bytes in /sec	0.37	0.00	0.00	0.07
Bytes out /sec	0.00	0.00	0.00	0.00
Bytes rejected /sec	0.00	0.00	0.00	0.00
Failed fetch request /sec	0.00	0.00	0.00	0.00
Failed produce request /sec	0.00	0.00	0.00	0.00

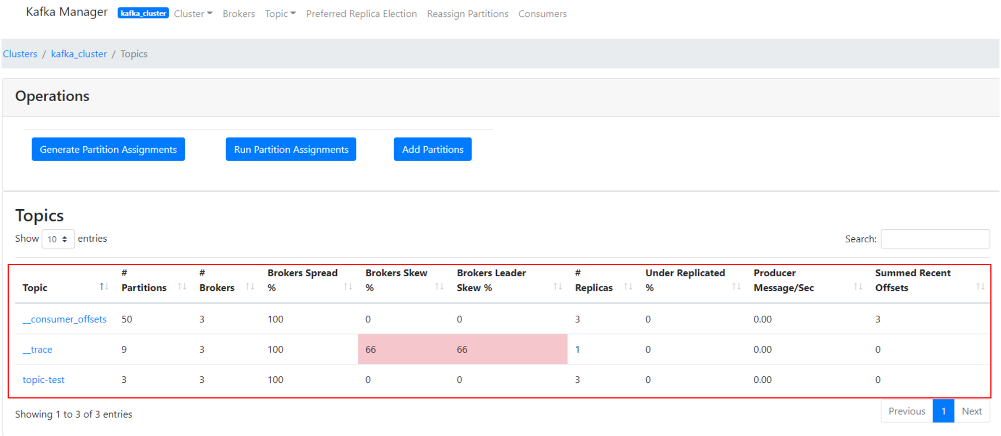
- Topics of an instance

  In the navigation bar, choose **Topic** > **List**. The displayed page shows the list of topics and information about the topics, as shown in **Figure 6-9**.

NOTICE

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

Figure 6-9 Topics of an instance



Operations

Topics

Topic	# Partitions	# Brokers	Brokers Spread %	Brokers Skew %	Brokers Leader Skew %	# Replicas	Under Replicated %	Producer Message/Sec	Summed Recent Offsets
__consumer_offsets	50	3	100	0	0	3	0	0.00	3
__trace	9	3	100	66	66	1	0	0.00	0
topic-test	3	3	100	0	0	3	0	0.00	0

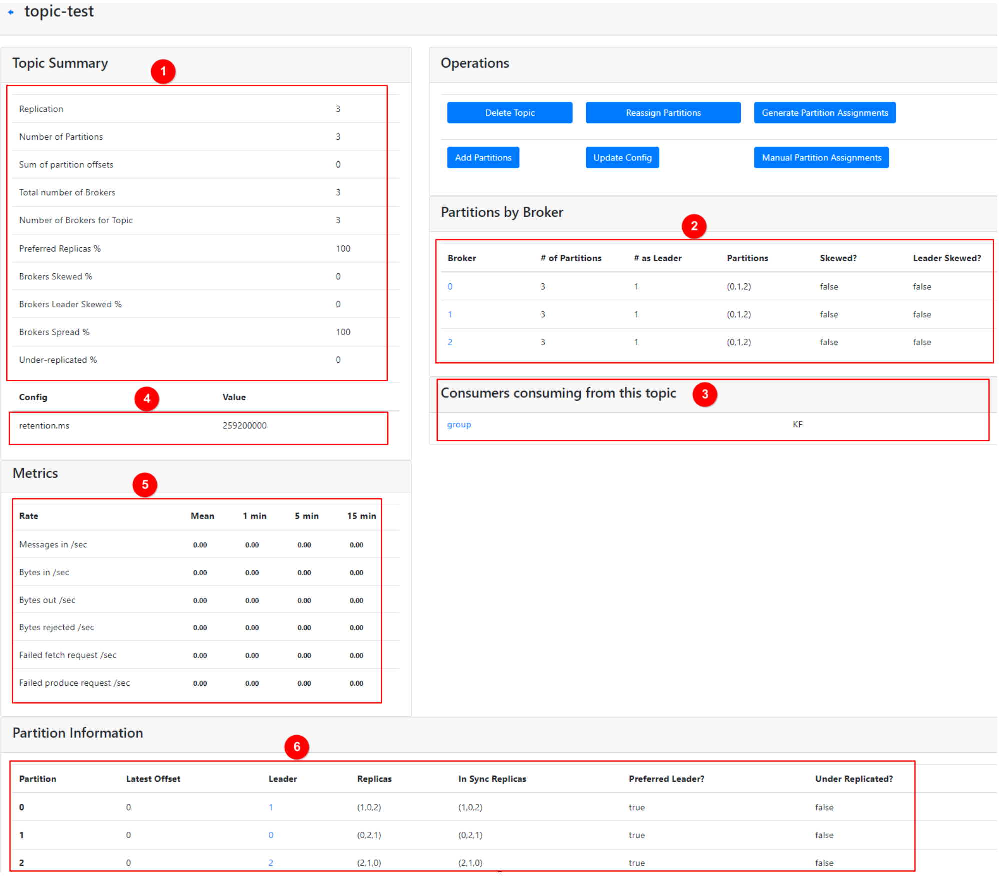
Showing 1 to 3 of 3 entries

- Details of a topic

Click the name of a topic to view its details on the displayed page, as shown in **Figure 6-10**.

- 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 6-10 Details of a topic



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

Config	Value
retention.ms	259200000

Broker	# of Partitions	# as Leader	Partitions	Skewed?	Leader Skewed?
0	3	1	(0,1,2)	false	false
1	3	1	(0,1,2)	false	false
2	3	1	(0,1,2)	false	false

group	KF

Rate	Mean	1 min	5 min	15 min
Messages in /sec	0.00	0.00	0.00	0.00
Bytes in /sec	0.00	0.00	0.00	0.00
Bytes out /sec	0.00	0.00	0.00	0.00
Bytes rejected /sec	0.00	0.00	0.00	0.00
Failed fetch request /sec	0.00	0.00	0.00	0.00
Failed produce request /sec	0.00	0.00	0.00	0.00

Partition	Latest Offset	Leader	Replicas	In Sync Replicas	Preferred Leader?	Under Replicated?
0	0	1	(1,0,2)	(1,0,2)	true	false
1	0	0	(0,2,1)	(0,2,1)	true	false
2	0	2	(2,1,0)	(2,1,0)	true	false

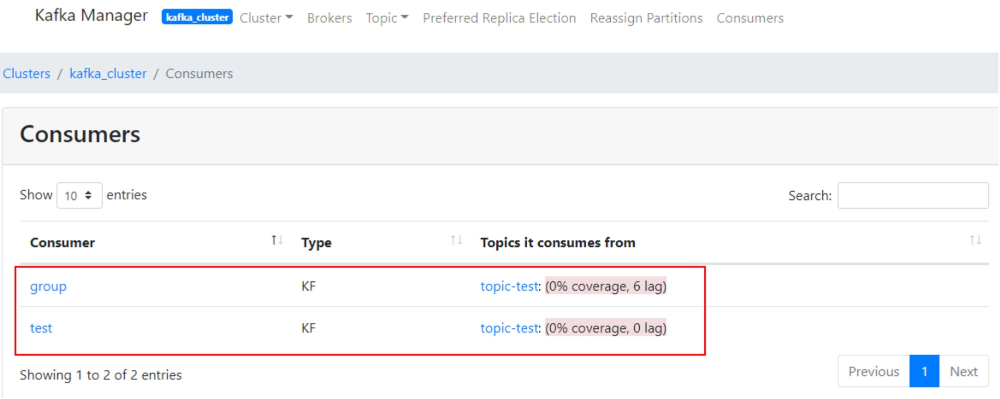
- List of consumers

  Click **Consumers** to view the list of consumers in a cluster.

  NOTE

  Only consumer groups that have retrieved messages in the last 14 days are displayed.

Figure 6-11 Viewing the list of consumers



Consumers

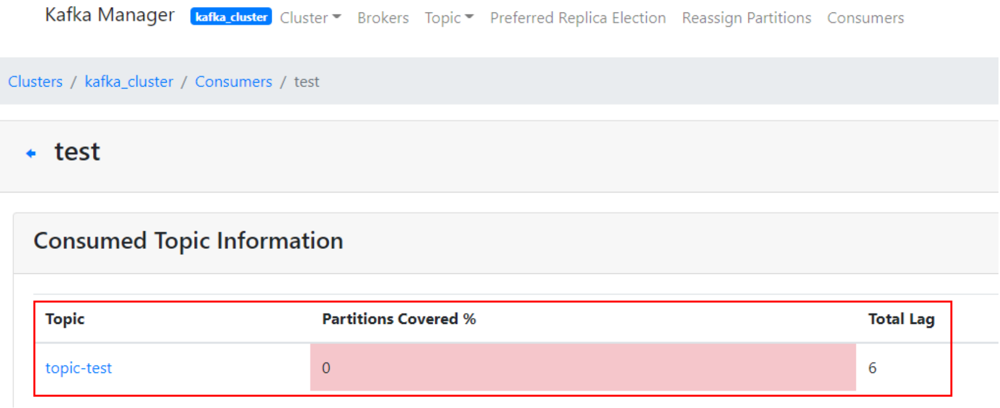
Consumer	Type	Topics it consumes from
group	KF	topic-test: (0% coverage, 6 lag)
test	KF	topic-test: (0% coverage, 0 lag)

Showing 1 to 2 of 2 entries

- Details of a specific consumer

  Click the name of a consumer to view its details, including the list of topics in the consumer and the number of messages that can be retrieved in each topic (**Total Lag**).

Figure 6-12 Viewing consumer details



test

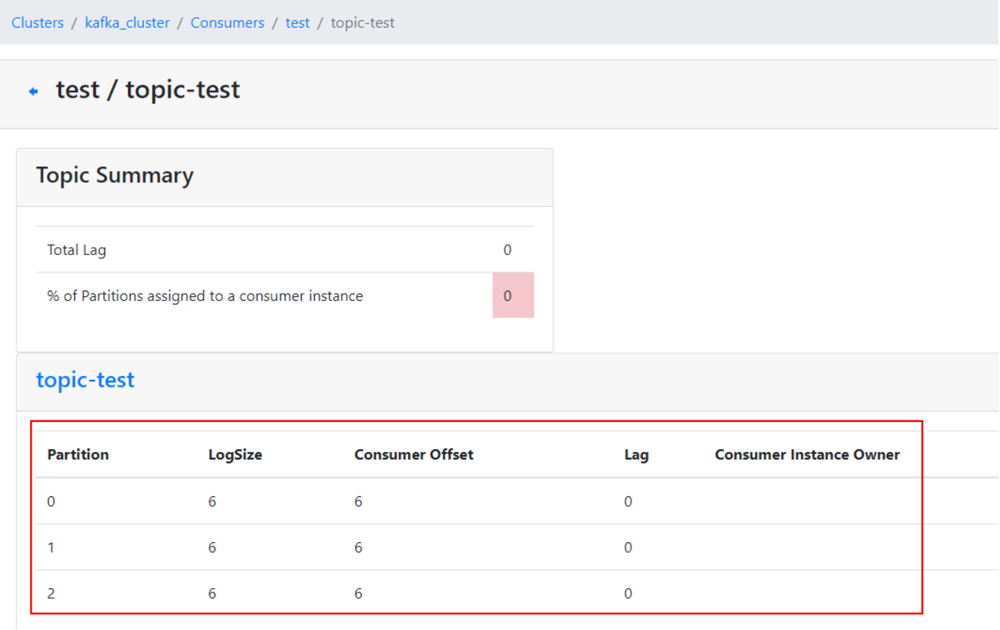
Consumed Topic Information

Topic	Partitions Covered %	Total Lag
topic-test	0	6

- Details of topics in a consumer

  Click the name of a topic to view 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 6-13 Viewing details of a topic



Clusters / kafka_cluster / Consumers / test / topic-test

test / topic-test

Topic Summary

Total Lag	0
% of Partitions assigned to a consumer instance	0

topic-test

Partition	LogSize	Consumer Offset	Lag	Consumer Instance Owner
0	6	6	0	
1	6	6	0	
2	6	6	0	

7 Managing Instances

7.1 Modifying Instance Specifications

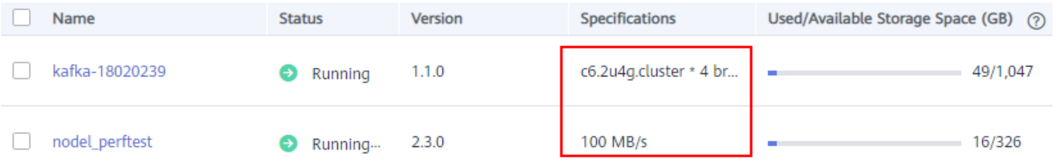
Scenario

After creating a Kafka instance, you can increase its storage space, bandwidth (only for old specifications), and broker quantity (only for new specifications).

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, **c6.2u4g.cluster*4 brokers**).

Figure 7-1 Instance list



□	Name	Status	Version	Specifications	Used/Available Storage Space (GB) ⓘ
□	kafka-18020239	Running	1.1.0	c6.2u4g.cluster * 4 br...	49/1,047
□	nodel_perftest	Running...	2.3.0	100 MB/s	16/326

Notes and Constraints

- You can expand the storage space 20 times.
- If you increase the bandwidth or add brokers, the maximum number of partitions will also be increased. When you increase the bandwidth or change the broker quantity, 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.

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 **Application** > **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, number of brokers, 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
  >
  > - Bandwidth is increased by scaling out brokers. The original brokers and services are not affected.
  > - New topics are created on new brokers. After the bandwidth is increased, the original topics are still on the original brokers. To use new brokers, migrate the topics by using Kafka Manager or create new topics.
  > - If public access is enabled and EIPs are configured for the instance, configure EIPs for the new brokers when expanding the bandwidth.

- 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.

  > NOTE
  >
  > - Storage space expansion does not affect services.
  > - Available storage space = Actual storage space – Storage space for storing logs and ZooKeeper data – Disk formatting loss
  >
  >   For example, if the storage space is expanded to 700 GB, the storage space for storing logs and ZooKeeper data is 100 GB, and the disk formatting loss is 7 GB, then the available storage space after capacity expansion will be 593 GB.

To modify new specifications, perform the following steps:

- Expand the storage space.

  For **Modify 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.

NOTE

- Storage space expansion does not affect services.
- Available storage space = Actual storage space – Storage space for storing logs and ZooKeeper data – Disk formatting loss

  For example, if the storage space is expanded to 700 GB, the storage space for storing logs and ZooKeeper data is 100 GB, and the disk formatting loss is 7 GB, then the available storage space after capacity expansion will be 593 GB.

- Add brokers.

  For **Modify By**, select **Brokers**. Then, enter the number of brokers and click **Next**. Confirm the configurations and click **Submit**.

  View the number of brokers in the **Specifications** column in the instance list.

  NOTE

  - Adding brokers does not affect the original brokers or services.
  - New topics are created on new brokers. After the bandwidth is increased, the original topics are still on the original brokers. To use new brokers, migrate the topics by using Kafka Manager or create new topics.
  - If public access is enabled and EIPs are configured for the instance, configure EIPs for the new brokers.

----End

7.2 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 **Application** > **Distributed Message Service for Kafka** to open the console of DMS for Kafka.

Step 4 Search for a Kafka instance by tag, status, name, ID, or connection address. **Table 7-1** describes the various possible statuses of a Kafka instance.

Table 7-1 Kafka instance status description

Status	Description
Creating	The instance is being created.
Running	The instance is running properly. Only instances in the **Running** state can provide services.
Faulty	The instance is not running properly.
Restarting	The instance is being restarted.
Changing	The instance specifications or public access configurations are being modified.
Change failed	The instance specifications or public access configurations failed to be modified.

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

Table 7-2 describes the parameters for connecting to a Kafka instance. For details about other parameters, see the **Basic Information** tab page of the Kafka instance on the console.

Table 7-2 Connection parameters

Section	Parameter	Description
Connection	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.
	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.

----End

7.3 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 **Application** > **Distributed Message Service for Kafka** to open the console of DMS for Kafka.

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

- Select one or more Kafka instances and click **Restart** in the upper left corner.
- In the row containing the desired instance, click **Restart**.
- Click the desired Kafka instance to 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

7.4 Deleting an Instance

Scenario

On the Kafka console, you can delete one or more Kafka instances that have been created or failed to be 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** or **Faulty**.

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 **Application** > **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**, **Changing**, **Change failed**, or **Restarting** state cannot be deleted.

Step 5 In the **Delete Instance** dialog box, click **Yes** to delete the Kafka instance.

It takes 1 to 60 seconds to delete a Kafka instance.

----End

Deleting Kafka Instances That Failed to Be Created

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 **Application** > **Distributed Message Service for Kafka** to open the console of DMS for Kafka.

Step 4 If there are Kafka instances that failed to be created, **Instance Creation Failures** and quantity information will be displayed.

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

Step 5 Click **Instance Creation Failures** or the icon or quantity next to it.

Step 6 Delete Kafka instances that failed to be created in either of the following ways:

- To delete all Kafka instances that failed to be created at once, click **Clear Failed Instance**.
- To delete a single Kafka instance that failed to be created, click **Delete** in the row containing the chosen Kafka instance.

----End

7.5 Modifying the Information About an Instance

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

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 **Application** > **Distributed Message Service for Kafka** to open the console of DMS for Kafka.

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

Step 5 Modify the following parameters if needed:

- Instance Name
- Enterprise Project (Changing the enterprise project will not restart the instance.)
- Description
- Security Group
- Public Access (For details about how to change the public access configuration, see **Configuring Public Access**.)
- Capacity Threshold Policy (Modifying this setting will not restart the instance.)
- Automatic Topic Creation (Modifying this setting will restart the instance.)

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

- If **Capacity Threshold Policy**, **Public Access**, or **Automatic Topic Creation** has been modified, you will be redirected to the **Background Tasks** page, which displays the modification progress and result.
- If **Instance Name**, **Description**, **Enterprise Project**, or **Security Group** has been modified, the modification result will be displayed on the upper right corner of the page.

----End

7.6 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.

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 **Application** > **Distributed Message Service for Kafka** to open the console of DMS for Kafka.

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

Step 5 Configure public access.

NOTE

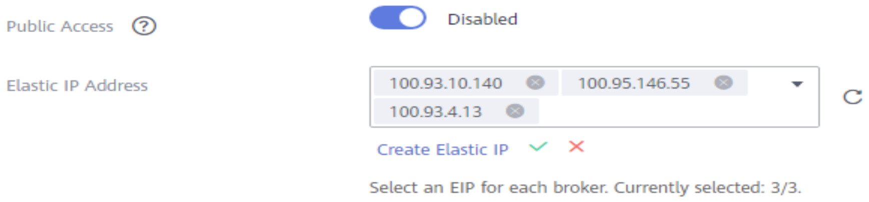
- You can change the public access setting only when the Kafka instance is in the **Running** state.
- Only IPv4 EIPs can be bound to Kafka instances.

Enabling public access

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 7-2 Configuring public access



Public Access ? Disabled

Elastic IP Address: 100.93.10.140, 100.95.146.55, 100.93.4.13

Create Elastic IP ✓ ✕

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

After public access is enabled, configure security group rules listed in **Table 7-3** before attempting to access Kafka. For details about accessing Kafka, see **Accessing a Kafka Instance**.

Table 7-3 Security group rules

Directio n	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).

Disabling public access

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.

----End

7.7 Resetting Kafka Manager Password

Scenario

You can reset the password of Kafka Manager of a Kafka instance if you forget it.

NOTE

You can reset the password of Kafka Manager only for a Kafka instance in the **Running** state.

Prerequisites

A Kafka instance 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 **Application** > **Distributed Message Service for Kafka** to open the console of DMS for Kafka.

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

- In the row containing the desired Kafka instance, choose **More** > **Reset Manager Password**.
- Click the desired Kafka instance to 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** tab 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.

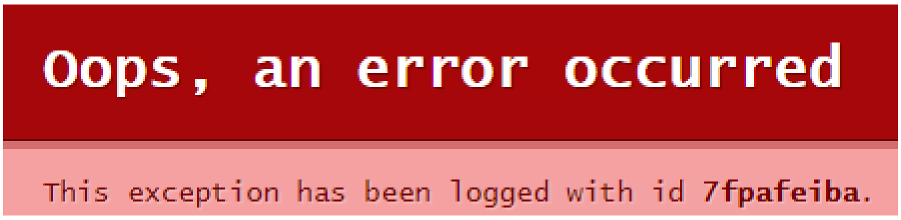
----End

7.8 Restarting Kafka Manager

Scenario

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

Figure 7-3 Error information



Oops, an error occurred

This exception has been logged with id **7fpafeiba**.

NOTE

Restarting Kafka Manager does not affect services.

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 **Application** > **Distributed Message Service for Kafka** to open the console of DMS for Kafka.

Step 4 Restart Kafka Manager using either of the following methods:

- In the row containing the desired Kafka instance, choose **More** > **Restart Kafka Manager**.
- Click the desired Kafka instance to view the instance details. In the upper right corner, choose **More** > **Restart Kafka Manager**.

Step 5 Click **OK**.

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

----End

7.9 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 modified and deleted.

A tag consists of a tag key and a tag value. **Table 7-4** lists the tag key and value requirements.

Table 7-4 Tag key and value requirements

Parameter	Requirements
Tag key	● Cannot be left blank. ● Must be unique for the same instance. ● Can contain a maximum of 36 characters. ● Cannot contain the following characters: =*<>\,|/ ● Cannot start or end with a space.
Tag value	● Cannot be left blank. ● Can contain a maximum of 43 characters. ● Cannot contain the following 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 **Application** > **Distributed Message Service for Kafka** to open the console of DMS for Kafka.

Step 4 Click the name of an instance.

Step 5 Click the **Tags** tab.

View the tags of the instance.

Step 6 Perform the following operations as required:

- Add a tag
  1. Click **Create/Delete Tag**.
  2. 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**.
  3. 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

7.10 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 **Application** > **Distributed Message Service for Kafka** to open the console of DMS for Kafka.

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

Step 5 Click the **Background Tasks** tab.

A list of background tasks is displayed.

Step 6 In the upper right corner, 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 **Yes** to clear the task information.

  NOTE

  You can only delete the records of tasks in the **Successful** or **Failed** state.

----End

7.11 Viewing Disk Usage

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

Procedure

Step 1 Log in to the management console.

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

NOTE

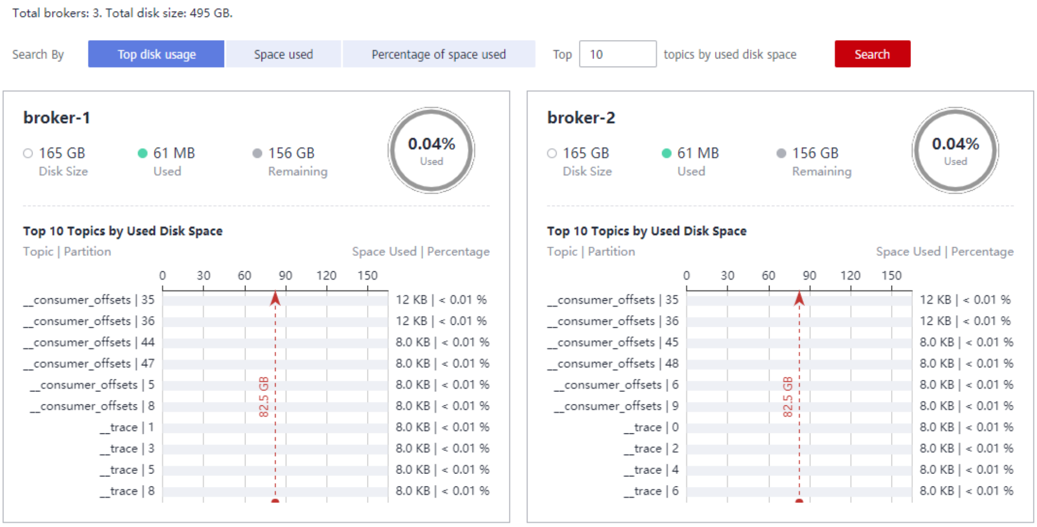
Select the region where your Kafka instance is located.

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

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

Step 5 Click the **Disk Usage Statistics** tab.

Figure 7-4 Viewing disk usage



Total brokers: 3. Total disk size: 495 GB.

broker-1

○ 165 GB Disk Size · ● 61 MB Used · ● 156 GB Remaining · 0.04% Used

Top 10 Topics by Used Disk Space

Topic | Partition	Space Used | Percentage
__consumer_offsets | 35	12 KB | < 0.01 %
__consumer_offsets | 36	12 KB | < 0.01 %
__consumer_offsets | 44	8.0 KB | < 0.01 %
__consumer_offsets | 47	8.0 KB | < 0.01 %
__consumer_offsets | 5	8.0 KB | < 0.01 %
__consumer_offsets | 8	8.0 KB | < 0.01 %
__trace | 1	8.0 KB | < 0.01 %
__trace | 3	8.0 KB | < 0.01 %
__trace | 5	8.0 KB | < 0.01 %
__trace | 8	8.0 KB | < 0.01 %

broker-2

○ 165 GB Disk Size · ● 61 MB Used · ● 156 GB Remaining · 0.04% Used

Top 10 Topics by Used Disk Space

Topic | Partition	Space Used | Percentage
__consumer_offsets | 35	12 KB | < 0.01 %
__consumer_offsets | 36	12 KB | < 0.01 %
__consumer_offsets | 45	8.0 KB | < 0.01 %
__consumer_offsets | 48	8.0 KB | < 0.01 %
__consumer_offsets | 6	8.0 KB | < 0.01 %
__consumer_offsets | 9	8.0 KB | < 0.01 %
__trace | 0	8.0 KB | < 0.01 %
__trace | 2	8.0 KB | < 0.01 %
__trace | 4	8.0 KB | < 0.01 %
__trace | 6	8.0 KB | < 0.01 %

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

8 Managing Topics

8.1 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 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 instance 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: Create 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 to use a topic with only one replica.

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.

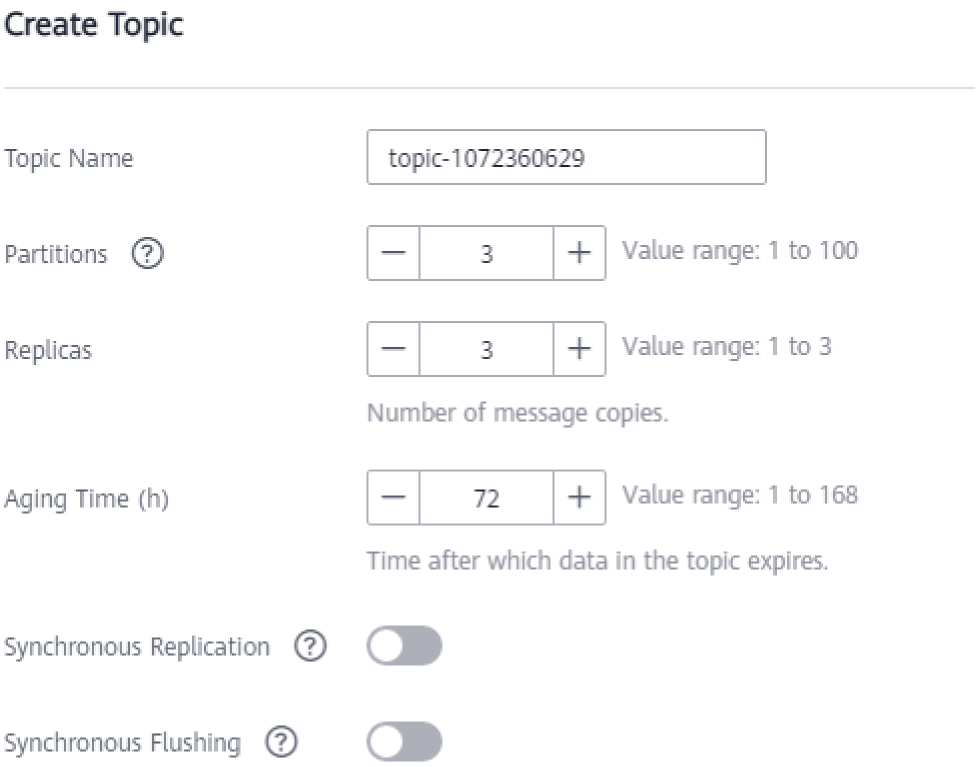
Step 3 Click ☰ and choose **Application** > **Distributed Message Service for Kafka** to open the console of DMS for Kafka.

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

Step 5 Click the **Topics** tab, and click **Create Topic**.

The **Create Topic** dialog box is displayed.

Figure 8-1 Creating a topic



Create Topic

Topic Name: topic-1072360629

Partitions ⓘ: 3 — Value range: 1 to 100

Replicas: 3 — Value range: 1 to 3

Number of message copies.

Aging Time (h): 72 — Value range: 1 to 168

Time after which data in the topic expires.

Synchronous Replication ⓘ: off

Synchronous Flushing ⓘ: off

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

Table 8-1 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 100 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 to use 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 168 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.

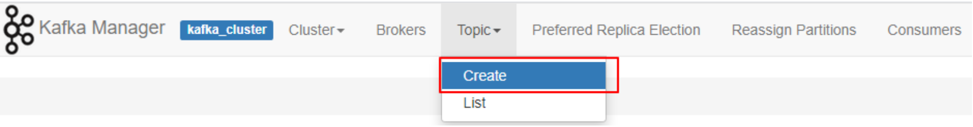
Step 7 Click **OK**.

----End

Method 2: Creating a Topic on Kafka Manager

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

Figure 8-2 Creating a topic on Kafka Manager



NOTICE

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

Method 3: Create 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, an underscore (_) or a number sign (#), monitoring data cannot be displayed.

- 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
     ```

8.2 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 **Application** > **Distributed Message Service for Kafka** to open the console of DMS for Kafka.

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

Step 5 Click the **Topics** tab.

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.

- 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
```

8.3 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 **Application** > **Distributed Message Service for Kafka** to open the console of DMS for Kafka.

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

Step 5 Click the **Topics** tab.

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

8.4 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 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**

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 **Application** > **Distributed Message Service for Kafka** to open the console of DMS for Kafka.

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

Step 5 Click the **Topics** tab.

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 100 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

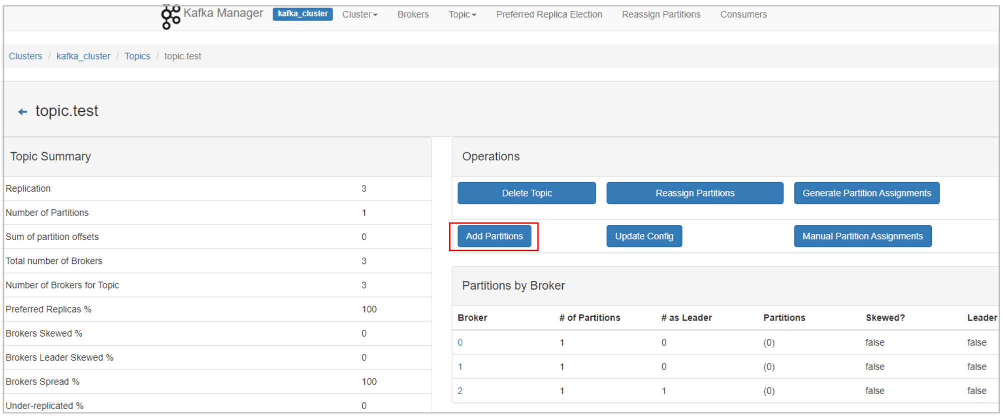
Step 1 **Log in to Kafka Manager**.

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 8-3 Topic details page



Topic Summary

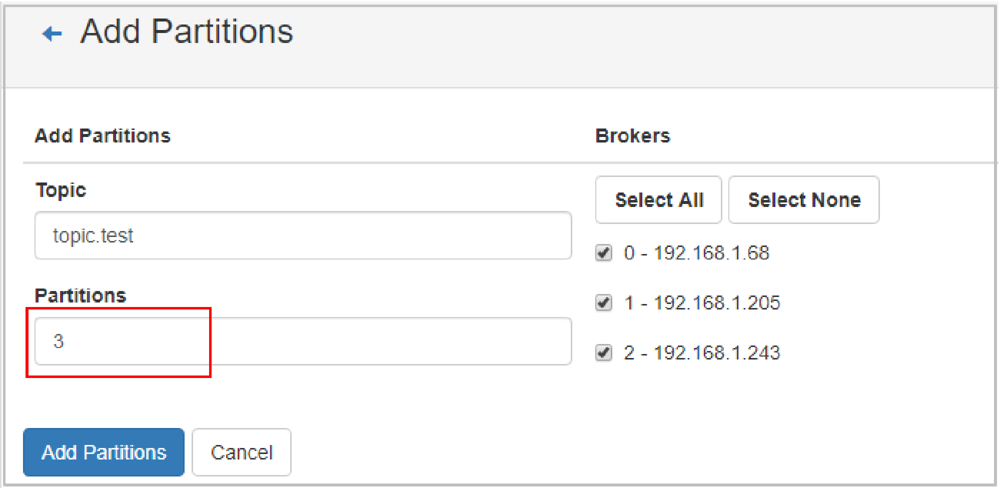
Replication	3
Number of Partitions	1
Sum of partition offsets	0
Total number of Brokers	3
Number of Brokers for Topic	3
Preferred Replicas %	100
Brokers Skewed %	0
Brokers Leader Skewed %	0
Brokers Spread %	100
Under-replicated %	0

Partitions by Broker

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

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

Figure 8-4 Adding partitions



Topic: topic.test

Partitions: 3

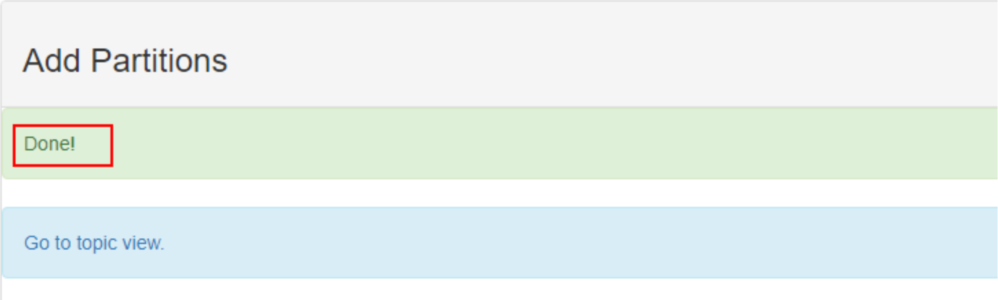
☑ 0 - 192.168.1.68

☑ 1 - 192.168.1.205

☑ 2 - 192.168.1.243

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

Figure 8-5 Partitions added



Done!

Go to topic view.

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.

- 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
     ```

8.5 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.

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 **Application** > **Distributed Message Service for Kafka** to open the console of DMS for Kafka.

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

Step 5 Click the **Topics** tab.

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

8.6 Partition Reassignment

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.

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.

- 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.
- 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 **Application** > **Distributed Message Service for Kafka** to open the console of DMS for Kafka.

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

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

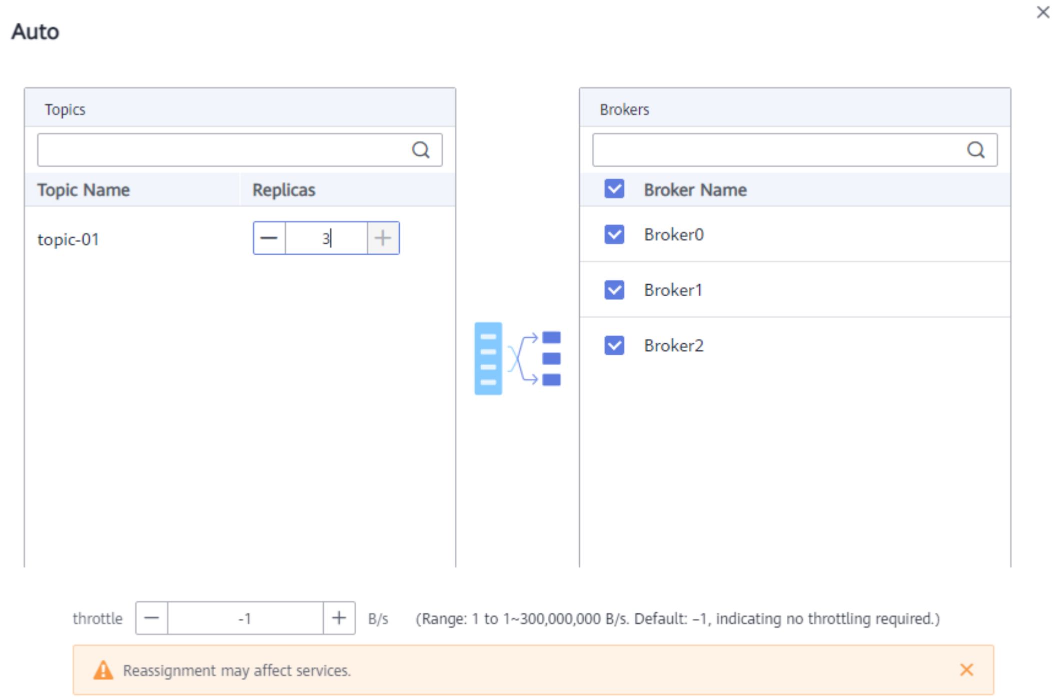
Step 6 Reassign partitions using either of the following methods:

- 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**.

Figure 8-6 Setting automatic reassignment parameters



Auto

Topic Name	Replicas
topic-01	3

	Broker Name
☑	Broker0
☑	Broker1
☑	Broker2

throttle -1 B/s (Range: 1 to 1~300,000,000 B/s. Default: –1, indicating no throttling required.)

Reassignment may affect services.

Step 8 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. When the task status is **Successful**, reassignment has completed.

NOTE

- You cannot delete topics whose reassignment tasks have started. Otherwise, the tasks will never complete.
- You cannot modify the partition quantity of topics whose reassignment tasks have started.
- Reassignment tasks cannot be manually stopped. Please wait until they complete.

----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 **Application** > **Distributed Message Service for Kafka** to open the console of DMS for Kafka.

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

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

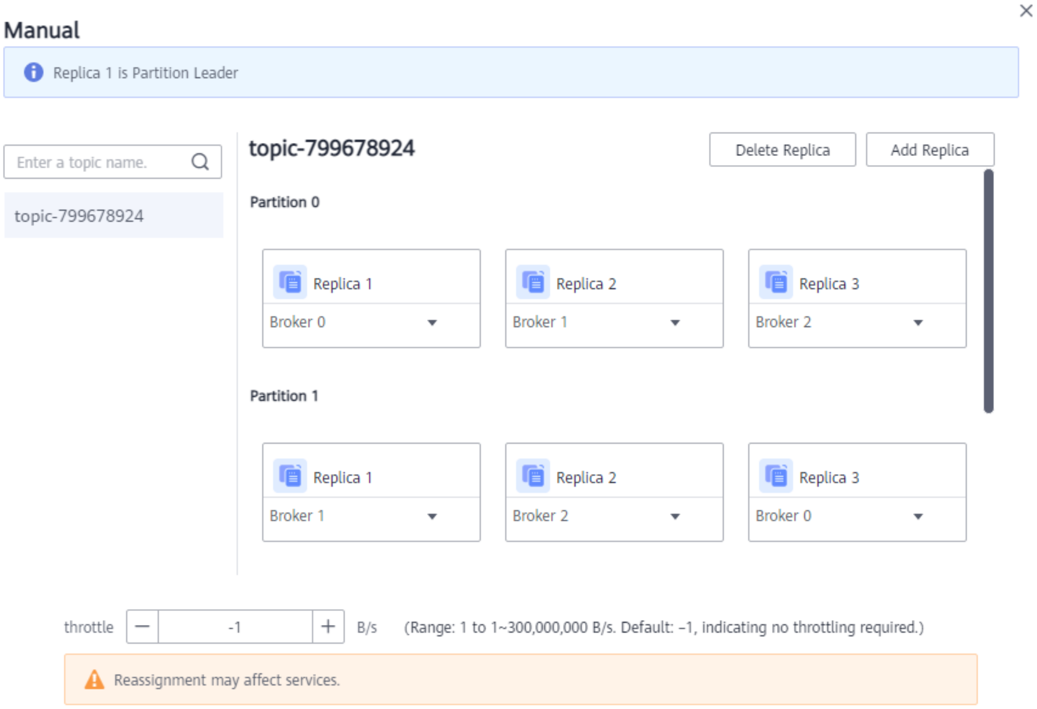
Step 6 Reassign partitions using either of the following methods:

- Select the desired topic and choose **Reassign** > **Manual** above the topic list.
- 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**.

Figure 8-7 Setting manual reassignment parameters



Manual

Replica 1 is Partition Leader

topic-799678924

Partition 0

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

Partition 1

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

throttle -1 B/s (Range: 1 to 1~300,000,000 B/s. Default: –1, indicating no throttling required.)

Reassignment may affect services.

Step 8 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. When the task status is **Successful**, reassignment has completed.

NOTE

- You cannot delete topics whose reassignment tasks have started. Otherwise, the tasks will never complete.
- You cannot modify the partition quantity of topics whose reassignment tasks have started.
- Reassignment tasks cannot be manually stopped. Please wait until they complete.

----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.

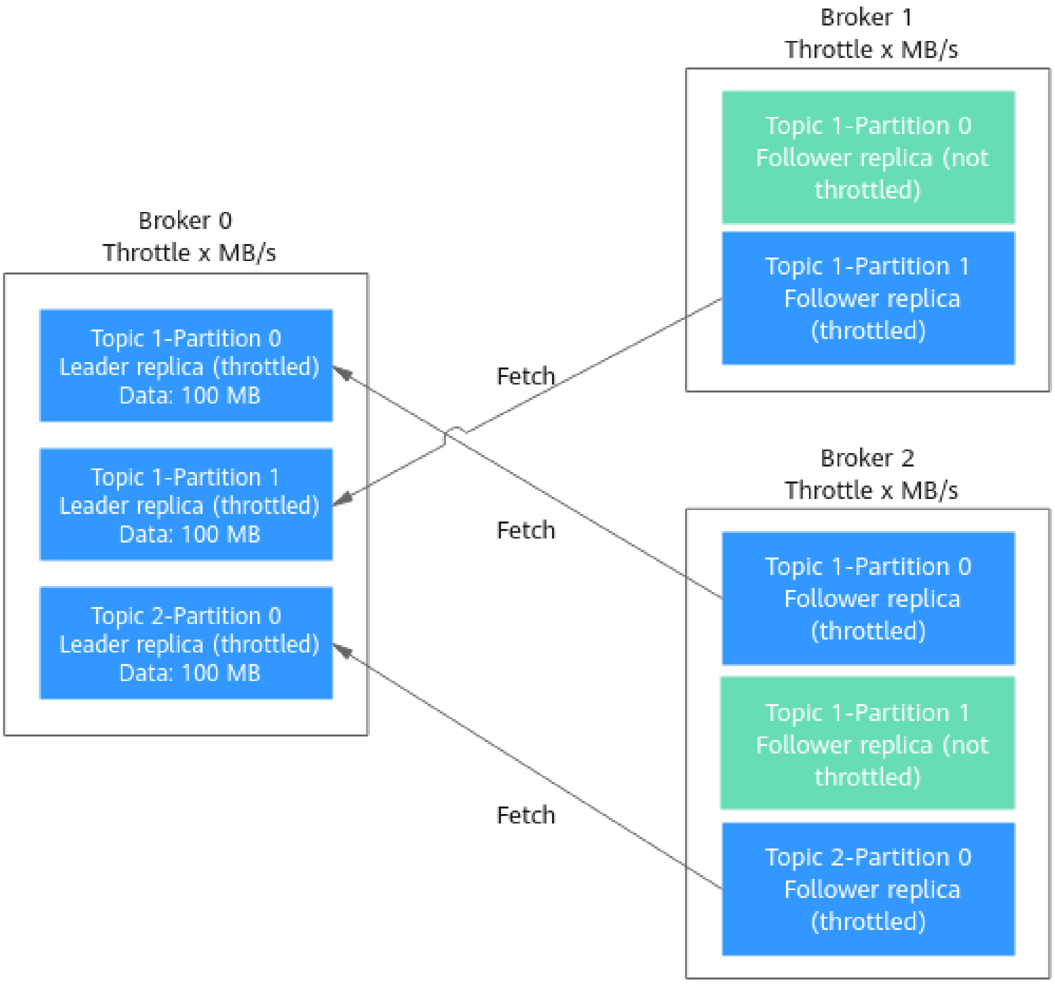
Table 8-2 Replica distribution before reassignment

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 8-3 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 8-8 Reassignment scenario 1



As shown in **Figure 8-8**, 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.

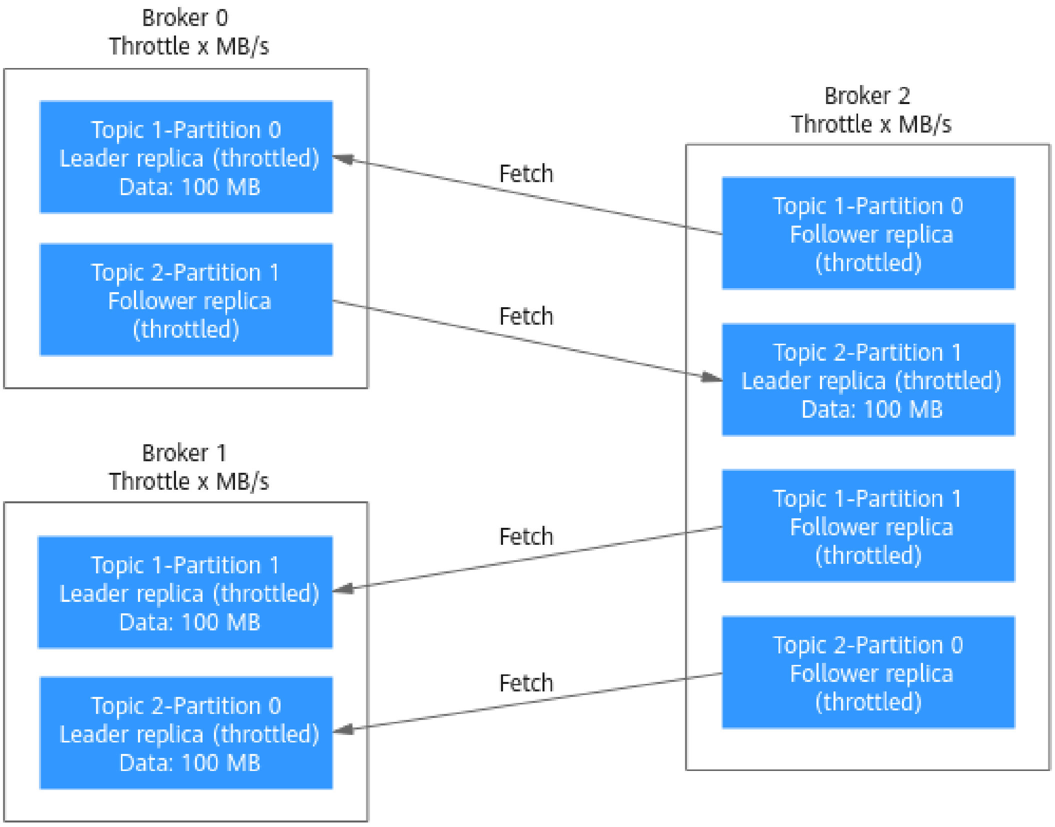
Table 8-4 Replica distribution before reassignment

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 8-5 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 8-9 Reassignment scenario 2



As shown in **Figure 8-9**, 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.

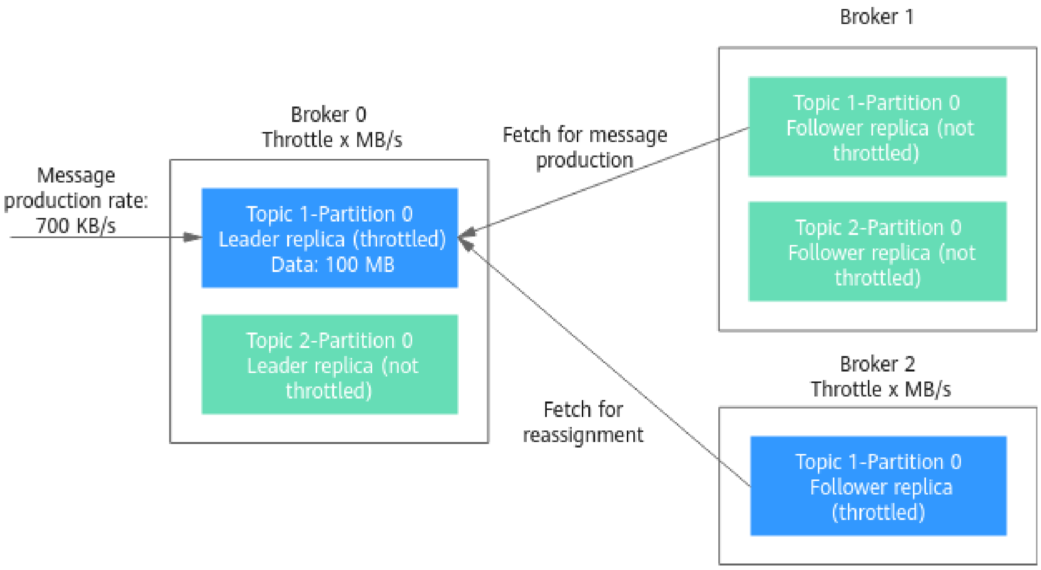
Table 8-6 Replica distribution before reassignment

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 8-7 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 8-10 Reassignment scenario 3


Broker 1
Topic 1-Partition 0 Follower replica (not throttled)
Topic 2-Partition 0 Follower replica (not throttled)
Broker 0 Throttle x MB/s
Fetch for message production
Message production rate: 700 KB/s
Topic 1-Partition 0 Leader replica (throttled) Data: 100 MB
Topic 2-Partition 0 Leader replica (not throttled)
Broker 2 Throttle x MB/s
Fetch for reassignment
Topic 1-Partition 0 Follower replica (throttled)

As shown in **Figure 8-10**, 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.

8.7 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 **Application** > **Distributed Message Service for Kafka** to open the console of DMS for Kafka.

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

Step 5 Click the **Topics** tab.

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. In the sample code, you can see whether SASL_SSL authentication is enabled. If **Access By** is **PLAINTEXT**, SASL_SSL authentication is disabled. If **Access By** is **SASL_SSL**, SASL_SSL authentication is enabled.

----End

8.8 Exporting Topics

Export topics 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 **Application** > **Distributed Message Service for Kafka** to open the console of DMS for Kafka.

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

Step 5 Click the **Topics** tab.

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, and whether synchronous replication and flushing are enabled.

----End

8.9 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**.

Constraints

- If no SASL_SSL user is granted any permission for a topic, all users can subscribe to or publish messages to the topic.
- 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 **Application** > **Distributed Message Service for Kafka** to open the console of DMS for Kafka.

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

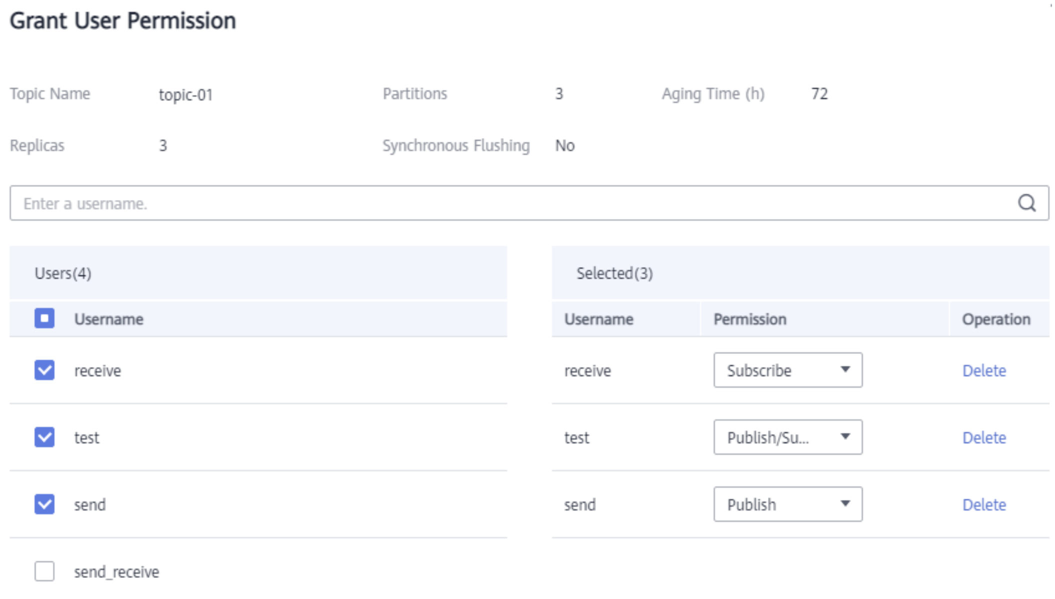
Step 5 In the navigation pane, choose the **Topics** tab.

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 is enabled. In the middle part, 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 SASL_SSL users.

Step 7 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.

Figure 8-11 Granting user permissions



Grant User Permission

Topic Name topic-01 Partitions 3 Aging Time (h) 72

Replicas 3 Synchronous Flushing No

Enter a username.

Users(4)

	Username
☑	receive
☑	test
☑	send
□	send_receive

Selected(3)

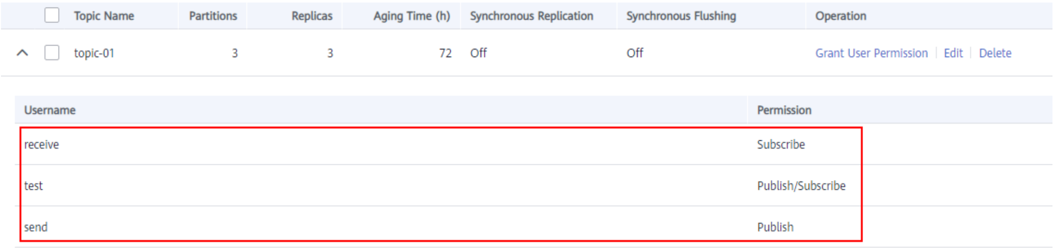
Username	Permission	Operation
receive	Subscribe	Delete
test	Publish/Su...	Delete
send	Publish	Delete

As shown in **Figure 8-11**, only the **test**, **send**, and **receive** users can subscribe to or publish messages to topic-01. The **send_receive** user cannot subscribe to or publish messages to topic-01.

Step 8 Click **OK**.

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

Figure 8-12 Viewing authorized users and their permissions



	Topic Name	Partitions	Replicas	Aging Time (h)	Synchronous Replication	Synchronous Flushing	Operation
□	topic-01	3	3	72	Off	Off	Grant User Permission | Edit | Delete

Username	Permission
receive	Subscribe
test	Publish/Subscribe
send	Publish

----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 **Application** > **Distributed Message Service for Kafka** to open the console of DMS for Kafka.

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

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

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

9 Managing Messages

9.1 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 **Application** > **Distributed Message Service for Kafka** to open the console of DMS for Kafka.

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

Step 5 Click the **Message Query** tab. Then specify the topic name, partition, and the search method.

If no partition is specified, messages in all partitions of the topic are displayed.

You can search by the following methods:

- **Creation time**: Search by the time that messages are created.
- **Offset**: Search by the message position.

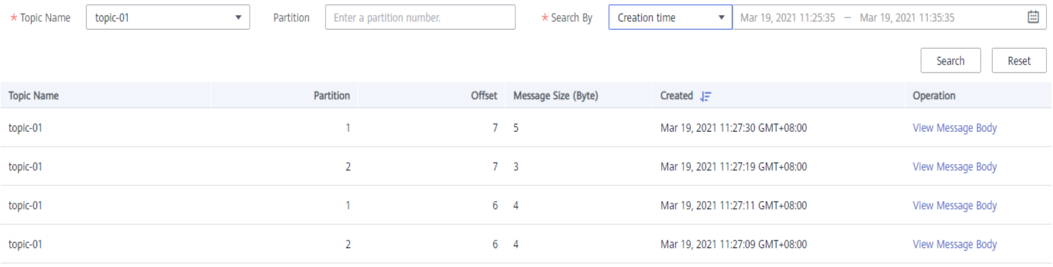
NOTE

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 6 Click **Search** to query messages.

The query result is as follows.

Figure 9-1 Querying topic messages



* Topic Name: topic-01

Partition: Enter a partition number.

* Search By: Creation time — Mar 19, 2021 11:25:35 – Mar 19, 2021 11:35:35

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 7 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.

Step 8 (Optional) To restore the default settings, click **Reset**.

----End

10 Managing Users

10.1 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**.

A maximum of 20 users can be created for a Kafka instance.

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 **Application** > **Distributed Message Service for Kafka** to open the console of DMS for Kafka.

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

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

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

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

----End

10.2 Resetting the SASL_SSL Password

Scenario

If you forget the password of a SASL_SSL user created on the **Users** tab page, you can reset the password and use the new password to connect to the Kafka instance.

NOTE

- 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 **Application** > **Distributed Message Service for Kafka** to open the console of DMS for Kafka.

Step 4 Click the name of the desired Kafka instance.

Step 5 On the **Users** tab 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

10.3 Deleting a SASL_SSL User

This section describes how to delete a SASL_SSL user.

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 **Application** > **Distributed Message Service for Kafka** to open the console of DMS for Kafka.

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

Step 5 Delete a SASL_SSL user using either of the following methods:

- On the **Users** tab page, click **Delete** in the row that contains the SASL_SSL user to be deleted.
- On the **Users** tab 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 **Yes** to delete the SASL_SSL user.

----End

11 Managing Consumer Groups

11.1 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 **Application** > **Distributed Message Service for Kafka** to open the console of DMS for Kafka.

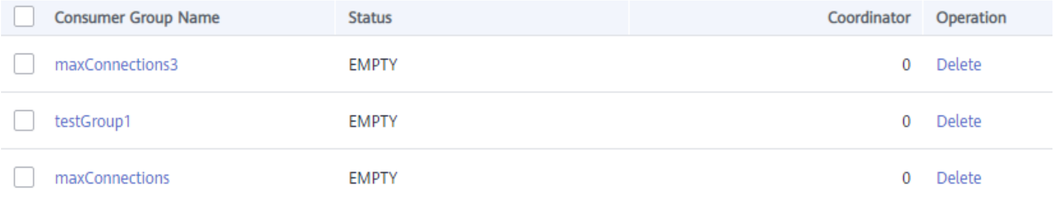
Step 4 Click the desired Kafka instance to view its details.

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

The consumer group name, status, and Coordinator are displayed. **Coordinator** 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 11-1 Consumer group list



□	Consumer Group Name	Status	Coordinator	Operation
□	maxConnections3	EMPTY	0	Delete
□	testGroup1	EMPTY	0	Delete
□	maxConnections	EMPTY	0	Delete

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 **Application** > **Distributed Message Service for Kafka** to open the console of DMS for Kafka.

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

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

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 **Application** > **Distributed Message Service for Kafka** to open the console of DMS for Kafka.

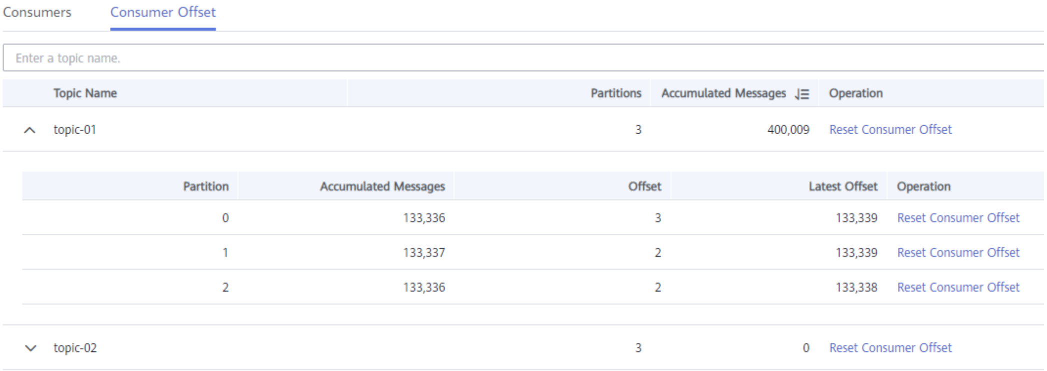
Step 4 Click the desired Kafka instance to view its details.

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

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, number of messages accumulated in each partition of the topic, offset of each partition, and latest offset.

Figure 11-2 Consumer offsets



	Topic Name	Partitions	Accumulated Messages	Operation
^	topic-01	3	400,009	Reset Consumer Offset
v	topic-02	3	0	Reset Consumer Offset

Partition	Accumulated Messages	Offset	Latest Offset	Operation
0	133,336	3	133,339	Reset Consumer Offset
1	133,337	2	133,339	Reset Consumer Offset
2	133,336	2	133,338	Reset Consumer Offset

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
```

11.2 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**.

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 **Application** > **Distributed Message Service for Kafka** to open the console of DMS for Kafka.

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

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

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

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

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

Consumer group statuses include:

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

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

----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

11.3 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.

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 **Application** > **Distributed Message Service for Kafka** to open the console of DMS for Kafka.

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

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

Step 6 Click the name of the desired consumer group.

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

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

Step 8 In the displayed **Reset Consumer Offset** dialog box, set the parameters by referring to **Table 11-1**.

Table 11-1 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.
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 Time Range**: 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

11.4 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.

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.

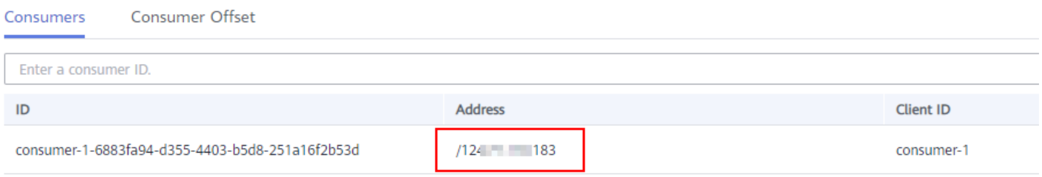
Step 3 Click ☰ and choose **Application** > **Distributed Message Service for Kafka** to open the console of DMS for Kafka.

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

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

Step 6 Click the desired consumer group.

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



ID	Address	Client ID
consumer-1-6883fa94-d355-4403-b5d8-251a16f2b53d	/124[redacted]183	consumer-1

Figure 11-3 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**.

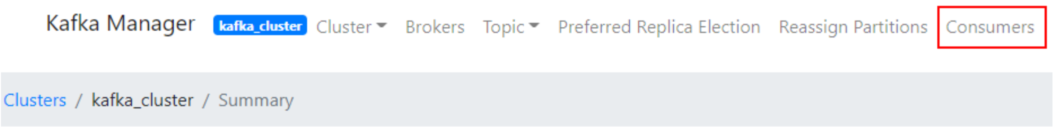
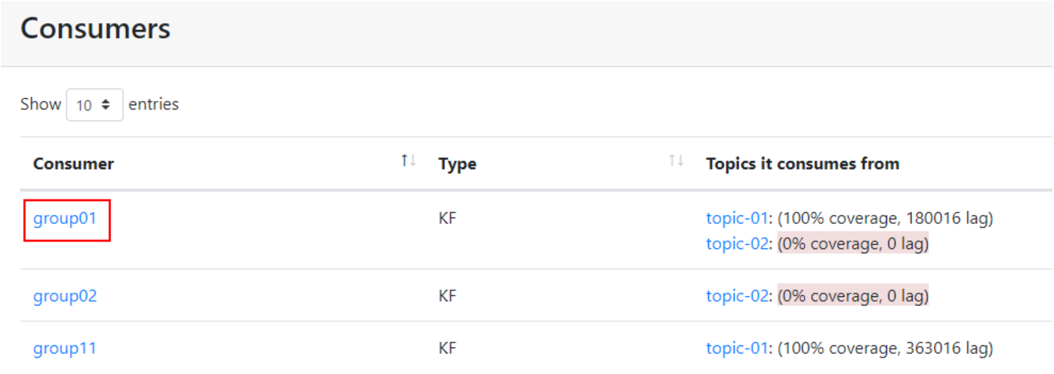


Figure 11-4 Navigation bar

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

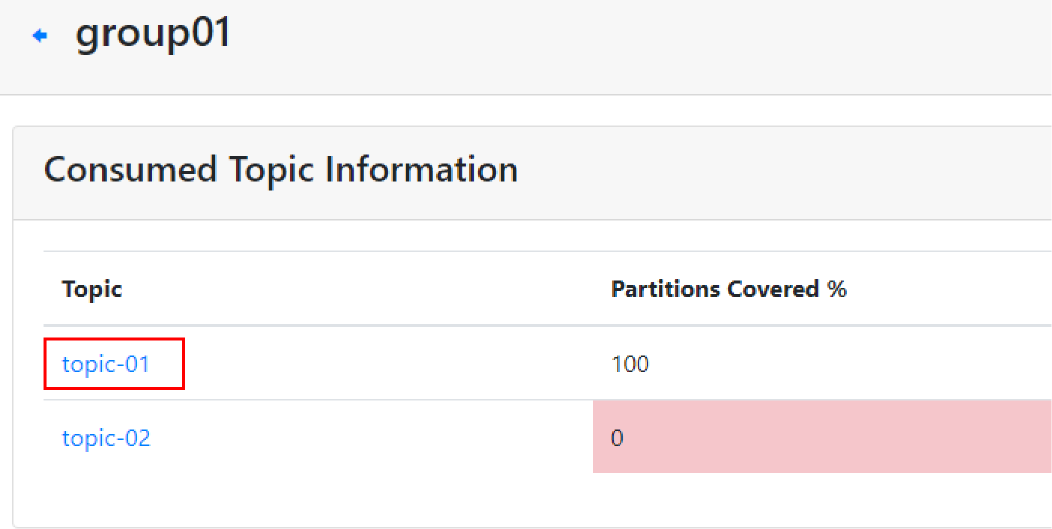


Consumer	Type	Topics it consumes from
group01	KF	topic-01: (100% coverage, 180016 lag) topic-02: (0% coverage, 0 lag)
group02	KF	topic-02: (0% coverage, 0 lag)
group11	KF	topic-01: (100% coverage, 363016 lag)

Figure 11-5 Consumer group list

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

Figure 11-6 Topics that the consumer group has subscribed to



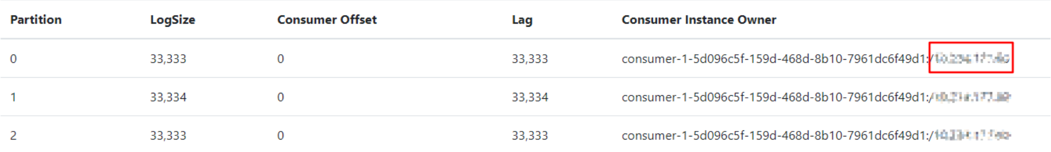
group01

Consumed Topic Information

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

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

Figure 11-7 Topic details page



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

----End

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 or clients.

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.

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 **Application** > **Distributed Message Service for Kafka** to open the console of DMS for Kafka.

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

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

Step 6 In the upper left corner, click **Create Quota**. The **Create Quota** slide panel is displayed.

Step 7 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 byte/s. If this parameter is left blank, no limit is set.
Consumption Limit	Set an upper limit on the consumption rate. The unit is byte/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 8 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** page, and click **user**, **client**, **user-client**, or ▾ to view the created quota.

- **user**: View quotas that apply only to users.
- **client**: View quotas that apply only to clients.
- **user-client**: View quotas that apply to both users and clients.

----End

12.2 Modifying a Quota

Scenario

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

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 **Application** > **Distributed Message Service for Kafka** to open the console of DMS for Kafka.

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

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

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** 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.

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 **Application** > **Distributed Message Service for Kafka** to open the console of DMS for Kafka.

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

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

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

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**.

Parameters of v1.1.0 instances are all static parameters. v2.3.0/v2.7 instances have both dynamic and static parameters.

- Dynamic parameters: Modifying dynamic parameters will not restart the instance.
- Static parameters: After static parameters are modified, you must manually restart the instance.

NOTE

Configuration parameters of some old instances cannot be modified. Check whether your instance parameters can be modified on the console. If they cannot be modified, contact customer service.

Prerequisites

You can modify configuration parameters of a Kafka instance when the instance is in the **Running** state.

Procedure

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

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

Step 5 On the **Parameters** tab page, click **Edit** in the row containing the parameter to modify. The parameters of v1.1.0 instances are described in **Table 13-1**. The parameters of v2.3.0/v2.7 instances are described in **Table 13-2** and **Table 13-3**.

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

Parameter	Description	Value Range	Default Value
min.insync.replicas	If a producer sets the acks parameter to **all** (or **-1**), the **min.insync.replicas** parameter specifies the minimum number of replicas that must acknowledge a write for the write to be considered successful.	1–3	1
message.max.bytes	Maximum length of a single message, 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

Parameter	Description	Value Range	Default Value
default.replication.factor	The default number of replicas configured for an automatically created topic.	1–3	3
num.partitions	The default number of partitions configured for each automatically created topic.	1–100	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 13-2 Dynamic parameters (v2.3.0/v2.7 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
max.connections.per.ip	The maximum number of connections allowed from each IP address. Request for new connections will be rejected once the limit is reached.	100–20,000	1000
unclean.leader.election.enable	Indicates whether to allow replicas not in the ISR set to be elected as the leader as a last resort, even though doing so may result in data loss.	**true** or **false**	true

Table 13-3 Static parameters (v2.3.0/v2.7 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
num.partitions	The default number of partitions configured for each automatically created topic.	1–100	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 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. Click ◔ (the **My Quota** icon) in the upper right corner.

   The **Service Quota** page is displayed.
4. On the **Service Quota** 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?

The system does not support online quota adjustment. To increase a quota, contact customer service by calling the hotline or sending an email. We will process your request as soon as possible and will inform you of the processing progress by phone or email.

Before you contact customer service, prepare the following information:

- Account name, project name, and project ID

  To obtain the preceding information, log in to the management console, click the username in the upper-right corner, and choose **My Credentials** from the drop-down list.
- Quota information, including:
  - Service name
  - Quota type

  - Required quota

To increase a quota, contact the administrator.

15 Monitoring

15.1 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 **Application** > **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:

- Click 🖥 next to a Kafka instance name. On the Cloud Eye console, view the metrics of the instance, nodes, 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, nodes, topics, and consumer groups. Metric data is reported to Cloud Eye every minute.

----End

15.2 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.

For example, you can call the **API**to query the monitoring data of the **Disk Capacity Usage** metric.

Namespace

SYS.DMS

Instance Metrics

Table 15-1 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 15-2 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	● **1**: alive ● **0**: not alive	Kafka instance broker	1 minute
broker_connections	Connections	Total number of TCP connections on the Kafka broker Unit: count	0–65,535	Kafka instance broker	1 minute
broker_cpu_usage	CPU Usage	CPU usage of the Kafka VM Unit: %	0–100	Kafka instance broker	1 minute
broker_disk_read_await	Average Disk Read Time	Average time for each disk I/O read in the monitoring period Unit: ms	> 0	Kafka instance broker	1 minute
broker_disk_write_await	Average Disk Write Time	Average time for each disk I/O write in the monitoring period Unit: ms	> 0	Kafka instance broker	1 minute
broker_total_bytes_in_rate	Inbound Traffic	Inbound traffic per second Unit: byte/s	0–1,000,000,000	Kafka instance broker	1 minute
broker_total_bytes_out_rate	Outbound Traffic	Outbound traffic per second Unit: byte/s	0–1,000,000,000	Kafka instance broker	1 minute

Metric ID	Metric Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
broker_disk_read_rate	Disk Read Speed	Read traffic on the disk Unit: byte/s, KB/s, MB/s, or GB/s	≥ 0	Kafka instance broker	1 minute
broker_disk_write_rate	Disk Write Speed	Write traffic on the disk Unit: byte/s, KB/s, MB/s, or GB/s	≥ 0	Kafka instance broker	1 minute

Topic Metrics

Table 15-3 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 **Queues** 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 **Queues** 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 **Queues** 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 **Queues** 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 **Queues** 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 **Queues** 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 **Queues** tab page.	≥ 0	Topic in a Kafka instance	1 minute

Consumer Group Metrics

Table 15-4 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 **Queue** is set to a specified 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 **Queue** is set to a specified 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 **Queue** is set to a specified 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 **Queue** is set to a specified 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	Consumer Available Messages	Number of remaining messages that can be retrieved in the consumer group Unit: Count **NOTE** This metric is available only when **Queue** is set to **All queues** 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 **Queue** is set to **All queues** on the **By Consumer Group** tab page.	0 to 2^{63}–1	Consumer group of a Kafka instance	1 minute

Dimension

Key	Value
kafka_instance_id	Kafka instance
kafka_broker	Kafka instance broker

Key	Value
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

15.3 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 15-5 Kafka instance metrics to configure alarm rules for

Metric ID	Metric	Alarm Policy	Metric Description and Alarm Handling
broker_disk_usage	Disk Capacity Usage	Alarm threshold: original value > 80% Number of consecutive periods: 1 Alarm severity: critical	Metric description: disk usage of the Kafka VM. Alarm handling: 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	Metric description: average load of each CPU core of the Kafka VM. Alarm handling: 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	Metric Description and Alarm Handling
broker_memory_usage	Memory Usage	Alarm threshold: original value > 90% Number of consecutive periods: 3 Alarm severity: critical	Metric description: memory usage of the Kafka VM. Alarm handling: Modify the instance **bandwidth or the number of brokers**. For details, see **Modifying Instance Specifications**.
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	Metric description: number of used partitions in the instance. Alarm handling: 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	Metric description: CPU usage of the Kafka VM. Alarm handling: 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	Metric Description and Alarm Handling
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	Metric description: total number of accumulated messages in all consumer groups of the instance. Alarm handling: 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	Metric description: number of remaining messages that can be retrieved from the specified topic in the consumer group. Alarm handling: 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 **Application** > **Distributed Message Service for Kafka** to open the console of DMS for Kafka.

Step 4 Click 📈 next to a Kafka instance name.

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.

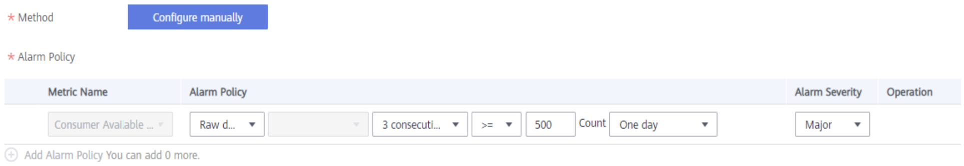
Step 6 Specify the alarm details.

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 15-1 Setting the alarm policy and alarm severity

   * Method: Configure manually

   * Alarm Policy

Metric Name	Alarm Policy							Alarm Severity	Operation
Consumer Available ...	Raw d...		3 consecuti...	>=	500	Count	One day	Major	

   Add Alarm Policy You can add 0 more.

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

----End

16 Auditing

16.1 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 16-1 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	batchDeleteDMSInstanceTaskSuccess
Failing to submit a request to delete multiple instances at a time	kafka	batchDeleteDMSInstanceFailure
Failing to delete multiple instances at a time	kafka	batchDeleteDMSInstanceTaskFailure
Successfully submitting a request to modify an instance order	kafka	modifyDMSInstanceOrderSuccess
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	resetDMSInstancePasswordSuccess
Failing to submit a request to reset instance password	kafka	resetDMSInstancePasswordFailure

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

Operation	Resource Type	Trace Name
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
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

16.2 Viewing Audit Logs

See **Querying Traces on the CTS Console**.

17 FAQs

17.1 Instances

17.1.1 Why Can't I Select Two AZs?

To improve the reliability of a Kafka instance, you are advised to select three AZs or more when creating the instance. You cannot select two AZs.

Each Kafka instance contains three ZooKeeper nodes. The ZooKeeper cluster manages Kafka instance configurations. If the ZooKeeper cluster is faulty, the Kafka instance cannot run properly. At least two ZooKeepers are required for the cluster to run properly.

Assume that you select only two AZs. AZ 1 has one ZooKeeper node, and AZ 2 has two. If AZ 1 is faulty, the instance can be used properly. If AZ 2 is faulty, the cluster cannot be used. In this case, the availability rate of the Kafka instance is just 50%. Therefore, do not select 2 AZs.

17.1.2 Why Can't I View the Subnet and Security Group Information When Creating a DMS Instance?

This may be because you do not have the **Server Administrator** and **VPC Administrator** permissions. For details about how to add permissions to a user group, see **Modifying User Group Permissions**.

17.1.3 How Do I Select Storage Space for a Kafka Instance?

The storage space is the space for storing messages (including messages in replicas), logs and metadata. When specifying storage space, specify the disk type and disk size. For more information about disks, see **Disk Types and Performance**.

For example, if the required disk size to store data for the retention period is 100 GB, the disk capacity must be at least: **100 GB x Number of replicas + 100 GB (reserved space)**. In a Kafka cluster, each node uses a 33 GB disk to store logs and ZooKeeper data. Therefore, the actual available storage space is less than the purchased storage space.

The number of replicas (3 by default) can be configured when you create a topic. If automatic topic creation has been enabled, each automatically created topic has three replicas by default. You can change this quantity by setting **default.replication.factor** on the **Parameters** tab page.

17.1.4 How Do I Choose Between High I/O and Ultra-high I/O?

- High I/O: The average latency is 1 to 3 ms, and the maximum bandwidth is 150 MB/s (read + write).
- Ultra-high I/O: The average latency is 1 ms, and the maximum bandwidth is 350 MB/s (read + write).

You are advised to select ultra-high I/O, because ultra-high I/O disks deliver much higher bandwidth than high I/O.

17.1.5 Which Capacity Threshold Policy Should I Use?

The following policies are supported:

- Stop production

  When the memory usage reaches the disk capacity threshold (95%), new messages will no longer be created, but existing messages can still be retrieved until they are discarded. The default retention time is three days. This policy is suitable for scenarios where no data losses can be tolerated.
- Automatically delete

  When the memory usage reaches the disk capacity threshold (95%), 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.

Select a proper policy based on requirements for data and service reliability. Both policies are only used for handling extreme scenarios. **To avoid extreme scenarios, buy sufficient disk space in the first place.**

17.1.6 Which Kafka Versions Are Supported?

Kafka v1.1.0, v2.3.0, and v2.7.

17.1.7 What Is the ZooKeeper Address of a Kafka Instance?

Kafka instances are managed using ZooKeeper. Opening ZooKeeper may cause misoperations and service losses. ZooKeeper is used only within Kafka clusters and does not provide services externally.

You can use open-source Kafka clients to connect to Kafka instances and call the native APIs to create and retrieve messages.

17.1.8 Are Kafka Instances in Cluster Mode?

Yes. A Kafka instance is a cluster that consists of three or more brokers.

17.1.9 Can I Modify the Port for Accessing a Kafka Instance?

No. You must access a Kafka instance through one of the following ports:

- Accessing a Kafka instance **without** SASL:

  Use port 9092 for intra-VPC access and port 9094 for public access.
- Accessing a Kafka instance **with** SASL:

  Use port 9093 for intra-VPC access and port 9095 for public access.

Ensure that correct rules have been configured for the security group of the instance. For details, see **How Do I Select and Configure a Security Group?**

17.1.10 How Long Are Kafka SSL Certificates Valid for?

The certificates are valid for more than 15 years. You do not need to worry about certificate expiration. The certificates are used for one-way authentication when enabling SASL for Kafka instances.

17.1.11 How Do I Synchronize Data from One Kafka Instance to Another?

Unfortunately, you cannot synchronize two Kafka instances in real time. To migrate services from one instance to another, create messages to both instances. After all messages in the original instance have been retrieved or aged, you can migrate services to the new instance.

17.1.12 How Do I Change the SASL_SSL Setting of a Kafka Instance?

The SASL_SSL setting cannot be changed once the instance has been created. Be careful when configuring this setting during instance creation. If you need to change the setting, you must create another instance.

17.1.13 How Do I Modify the SASL Mechanism?

After an instance is created, its SASL mechanism cannot be modified. If you want to change it, create an instance again.

17.1.14 Will a Kafka Instance Be Restarted After Its Enterprise Project Is Modified?

No. A Kafka instance will not be restarted if you modify its enterprise project.

17.1.15 Are Kafka Brokers and ZooKeeper Deployed on the Same VM or on Different VMs?

Kafka brokers and ZooKeeper are deployed on the same VM.

17.1.16 Which Cipher Suites Are Supported by Kafka?

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.

17.1.17 Can I Change an Instance from Single-AZ Deployment to Multi-AZ Deployment?

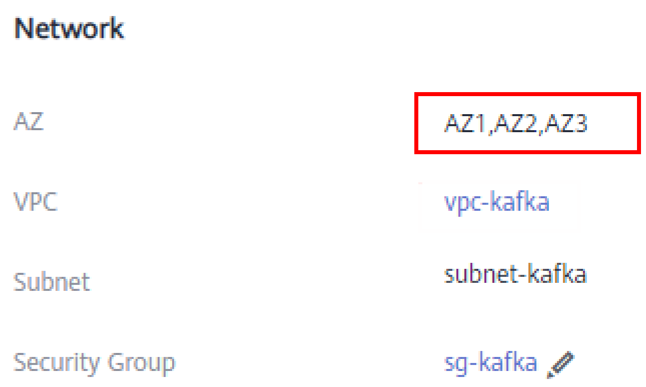
No. The AZ configuration cannot be changed once the instance is purchased. To use multiple AZs, buy another instance.

17.1.18 Does DMS for Kafka Support Cross-AZ Disaster Recovery? Where Can I View the AZs Configured for an Existing Instance?

DMS for Kafka supports cross-AZ disaster recovery. If you select multiple AZs when buying an instance, cross-AZ disaster recovery will be available.

You can view the AZs configured for an instance in the **Network** section on the **Basic Information** tab page of the instance. If there are multiple AZs, cross-AZ disaster recovery is available.

Figure 17-1 Instance basic information



Network

AZ	AZ1,AZ2,AZ3
VPC	vpc-kafka
Subnet	subnet-kafka
Security Group	sg-kafka

17.1.19 Do Kafka Instances Support Disk Encryption?

Yes.

17.1.20 Does Specification Modification Affect Services?

No. Bandwidth, broker quantity, or storage space expansion does not affect services.

17.1.21 Can I Change the VPC and Subnet After a Kafka Instance Is Created?

No. Once an instance is created, its VPC and subnet cannot be changed.

17.1.22 Where Can I Find Kafka Streams Use Cases?

You can find Kafka Streams use cases on the **official Kafka website**.

17.1.23 Can I Upgrade Kafka Instances?

No. Kafka instances cannot be upgraded once they are created. To use a higher Kafka version, create another Kafka instance.

17.1.24 Why Is the Version on the Console Different from That in Kafka Manager?

The version displayed on the console is used for your instance. Kafka Manager uses the common configuration of open-source Kafka 2.2.0. Therefore, the version displayed in Kafka Manager is 2.2.0, which is irrelevant to the version of your Kafka instance.

17.1.25 How Do I Bind an EIP Again?

On the DMS for Kafka console, click the name of the target Kafka instance. Disable **Public Access** in the **Connection** section on the **Basic Information** tab page, and then enable it again. Select the EIP to be bound.

17.2 Connections

17.2.1 How Do I Select and Configure a Security Group?

Kafka instances can be accessed within a VPC or over public networks. Before accessing a Kafka instance, configure a security group.

Intra-VPC Access

Step 1 Check whether the client and instance use the same security group.

- If they use the same security group, check whether the security group has the default inbound rule that allows communication among ECSs within the security group and the default outbound rule that allows all outbound traffic. If these rules are available, you do not need to add more rules. If these rules are not available, add rules according to **Table 17-1**.

Table 17-1 Security group rules

Direction	Protocol	Port	Source	Description
Inbound	TCP	9092	0.0.0.0/0	Accessing an instance within a VPC (with SSL encryption disabled)
Inbound	TCP	9093	0.0.0.0/0	Accessing an instance within a VPC (with SSL encryption enabled)

- If they use different security groups, go to **Step 2**.

Step 2 Configure security group rules as follows.

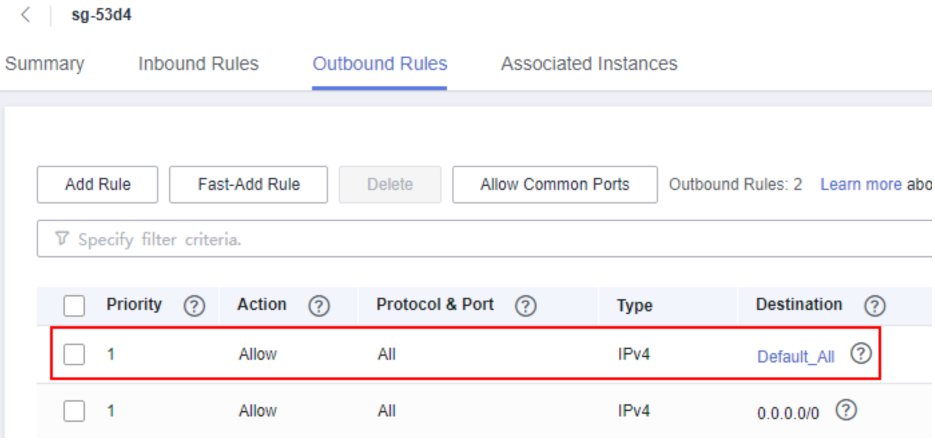
Assume that the security groups of the client and Kafka instance are **sg-53d4** and **Default_All**, respectively. You can specify a security group or IP address as the destination in the following rule. A security group is used as an example.

To ensure that your client can access the Kafka instance, add the following rule to the security group configured for the client:

Table 17-2 Security group rule

Direction	Action	Protocol & Port	Destination
Outbound	Allow	All	Default_All

Figure 17-2 Configuring a security group for the client



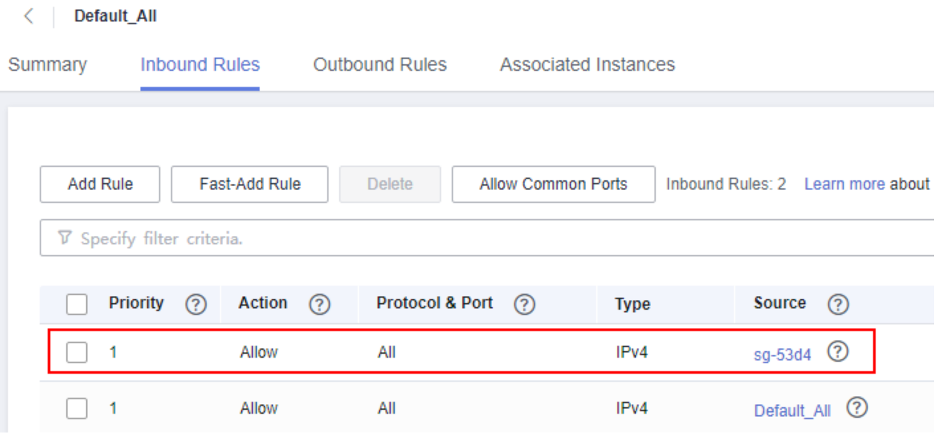
	Priority	Action	Protocol & Port	Type	Destination
□	1	Allow	All	IPv4	Default_All
□	1	Allow	All	IPv4	0.0.0.0/0

To ensure that your client can access the Kafka instance, add the following rule to the security group configured for the instance.

Table 17-3 Security group rule

Direction	Action	Protocol & Port	Source
Inbound	Allow	All	sg-53d4

Figure 17-3 Configuring the security group for the Kafka instance



□	Priority	Action	Protocol & Port	Type	Source
□	1	Allow	All	IPv4	sg-53d4
□	1	Allow	All	IPv4	Default_All

----End

Public Access

Configure security group rules according to **Table 17-4**.

Table 17-4 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).
Inbound	TCP	9095	0.0.0.0/0	Access Kafka through the public network (with SSL encryption).

17.2.2 Can I Access a Kafka Instance Over a Public Network?

Yes. For details, see the **instance access instructions**.

17.2.3 How Many Connection Addresses Does a Kafka Instance Have by Default?

The number of connection addresses of a Kafka instance is the same as the number of brokers of the instance. The following table lists the number of brokers corresponding to each flavor.

Table 17-5 Kafka instance specifications

Flavor	Brokers	Maximum TPS per Broker	Maximum Partitions per Broker	Maximum Consumer Groups per Broker	Maximum Client Connections per Broker	Storage Space
kafka. 2u4g.cluster	3–30	30,000	250	20	2000	300 GB–300,000 GB
kafka. 4u8g.cluster	3–30	100,000	500	100	4000	300 GB–600,000 GB
kafka. 8u16g.cluster	3–30	150,000	1000	150	4000	300 GB–900,000 GB
kafka. 12u24g.cluster	3–30	200,000	1500	200	4000	300 GB–900,000 GB
kafka. 16u32g.cluster	3–30	250,000	2000	200	4000	300 GB–900,000 GB

17.2.4 Do Kafka Instances Support Cross-Region Access?

Yes. You can access a Kafka instance across regions over a public network or by using direct connections.

17.2.5 Do Kafka Instances Support Cross-VPC Access?

Yes. You can use 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**.

17.2.6 Do Kafka Instances Support Cross-Subnet Access?

Yes.

If the client and the instance are in the same VPC, cross-subnet access is supported. By default, subnets in the same VPC can communicate with each other.

17.2.7 Does DMS for Kafka Support Authentication with Kerberos?

No, Kerberos authentication is not supported. Kafka supports client authentication with SASL and API calling authentication using tokens and AK/SK.

To access an instance in SASL mode, you need the certificates provided by DMS for Kafka. For details, see **Accessing a Kafka Instance with SASL**.

17.2.8 Does DMS for Kafka Support Password-Free Access?

Yes. No password is required for accessing a Kafka instance with SASL disabled. For details, see **Accessing a Kafka Instance Without SASL**.

17.2.9 How Do I Obtain the Public Access Address After Public Access Is Enabled?

Click the name of your Kafka instance. In the **Connection** section on the **Basic Information** tab page, view **Instance Address (Public Network)**.

For details about how to connect to a Kafka instance, see **Accessing a Kafka Instance**.

17.2.10 Does DMS for Kafka Support Authentication on Clients by the Server?

No.

17.2.11 Can I Use PEM SSL Truststore When Connecting to a Kafka Instance with SASL_SSL Enabled?

No. You can only use JKS certificates for connecting to instances in Java.

17.2.12 What Are the Differences Between JKS and CRT Certificates?

JKS certificates are used for connecting to instances in Java and CRT certificates are used for connecting to instances in Python.

17.2.13 Which TLS Version Does DMS for Kafka Support?

TLS 1.2.

17.2.14 Is There a Limit on the Number of Client Connections to a Kafka Instance?

Yes. The maximum allowed number of client connections varies by instance specifications.

- If the flavor is **kafka.2u4g.cluster**, a maximum of 2000 client connections are allowed.
- If the flavor is **kafka.4u8g.cluster**, a maximum of 4000 client connections are allowed.
- If the flavor is **kafka.8u16g.cluster**, a maximum of 4000 client connections are allowed.

- If the flavor is **kafka.12u24g.cluster**, a maximum of 4000 client connections are allowed.
- If the flavor is **kafka.16u32g.cluster**, a maximum of 4000 client connections are allowed.

17.2.15 How Many Connections Are Allowed from Each IP Address?

Each Kafka broker allows a maximum of 1000 connections from each IP address by default. Excess connections will be rejected. You can change the limit by referring to **Modifying Kafka Parameters**.

17.2.16 Can I Change the Private Network Addresses of a Kafka Instance?

No, and you cannot specify the IP addresses.

17.2.17 Is the Same SSL Certificate Used for Different Instances?

Yes. All Kafka instances and users use the same SSL certificate.

To obtain the SSL certificate, perform the following steps:

Step 1 On the Kafka console, click the name of your instance.

Step 2 In the **Connection** area on the **Basic Information** tab page, click **Download** next to **SSL Certificate**.

----End

17.3 Topics and Partitions

17.3.1 Is There a Limit on the Number of Topics in a Kafka Instance?

The number of topics is related to the total number of topic partitions and the number of partitions in each topic. There is an upper limit on the aggregate number of partitions of topics. When this limit is reached, no more topics can be created.

The partition limit varies depending on the flavor, as shown in the following table.

Table 17-6 Kafka instance specifications

Flavor	Brokers	Maximum TPS per Broker	Maximum Partitions per Broker	Maximum Consumer Groups per Broker	Maximum Client Connections per Broker	Storage Space
kafka. 2u4g.cluster	3–30	30,000	250	20	2000	300 GB–300,000 GB
kafka. 4u8g.cluster	3–30	100,000	500	100	4000	300 GB–600,000 GB
kafka. 8u16g.cluster	3–30	150,000	1000	150	4000	300 GB–900,000 GB
kafka. 12u24g.cluster	3–30	200,000	1500	200	4000	300 GB–900,000 GB
kafka. 16u32g.cluster	3–30	250,000	2000	200	4000	300 GB–900,000 GB

17.3.2 Why Is Partition Quantity Limited?

Kafka manages messages by partition. If there are too many partitions, message creation, storage, and retrieval will be fragmented, affecting the performance and stability. If the total number of partitions of topics reaches the upper limit, you cannot create more topics.

The partition limit varies depending on the flavor, as shown in the following table.

Table 17-7 Kafka instance specifications

Flavor	Brokers	Maximum TPS per Broker	Maximum Partitions per Broker	Maximum Consumer Groups per Broker	Maximum Client Connections per Broker	Storage Space
kafka. 2u4g.cluster	3–30	30,000	250	20	2000	300 GB–300,000 GB

Flavor	Brokers	Maximum TPS per Broker	Maximum Partitions per Broker	Maximum Consumer Groups per Broker	Maximum Client Connections per Broker	Storage Space
kafka.4u8g.cluster	3–30	100,000	500	100	4000	300 GB–600,000 GB
kafka.8u16g.cluster	3–30	150,000	1000	150	4000	300 GB–900,000 GB
kafka.12u24g.cluster	3–30	200,000	1500	200	4000	300 GB–900,000 GB
kafka.16u32g.cluster	3–30	250,000	2000	200	4000	300 GB–900,000 GB

17.3.3 Can I Change the Partition Quantity?

No. If you want to use fewer partitions, delete the corresponding topic, create another one, and specify the desired number of partitions.

17.3.4 Why Do I Fail to Create Topics?

Possible cause: The aggregate number of partitions of created topics has reached the upper limit. The maximum number of partitions varies with instance specifications. For details, see **Specifications**.

Solution: Scale up the instance or delete unnecessary topics.

17.3.5 Do Kafka Instances Support Batch Importing Topics or Automatic Topic Creation?

Automatic topic creation is supported, but batch topic import is not supported. You can only export topics in batches.

Enable automatic topic creation using one of the following methods:

- When creating an instance, enable automatic topic creation.
- After an instance is created, enable automatic topic creation on the **Basic Information** tab page.

17.3.6 Why Do Deleted Topics Still Exist?

This may be because automatic topic creation has been enabled and a consumer is connecting to the topic. If no existing topics are available for message creation, new topics will be automatically created.

To solve this problem, disable automatic topic creation.

17.3.7 Can I View the Disk Space Used by a Topic?

Yes. Use either of the following methods to check the disk space used by a topic:

- Click next to the Kafka instance name to go to the Cloud Eye console. On the **Queues** tab page, set **Queue** to the name of the topic whose disk space you want to view and **Scope** to **Basic monitoring**. The **Message Size** metric reflects the message size of the selected topic.
- Click the desired Kafka instance to view its details. In the navigation pane, choose **Monitoring**. On the **By Topic** tab page, set **Topic** to the name of the topic whose disk space you want to view and **Monitoring Type** to **Basic monitoring**. The **Message Size** metric reflects the message size of the selected topic.

17.3.8 Can I Add ACL Permissions for Topics?

If you have enabled SASL_SSL for your Kafka instance, you can configure ACL permissions for your topics. On the **Topics** tab page of the Kafka console, click **Grant User Permission** in the row that contains the topic for which you want to configure user permissions.

For details, see **Configuring Topic Permissions**.

17.3.9 What Should I Do If Kafka Storage Space Is Used Up Because Retrieved Messages Are Not Deleted?

Messages are not deleted immediately after being retrieved. They are deleted only when the aging time expires.

You can shorten the aging time or expand the storage space.

17.3.10 How Do I Increase the Partition Quantity?

You can increase the partition quantity by adding brokers.

To do so, go to the Kafka console, locate the row that contains the desired instance, and choose **More** > **Modify Specifications**. On the page that is displayed, increase the bandwidth as required. For details, see **Modifying Instance Specifications**.

17.3.11 Will a Kafka Instance Be Restarted After Its Automatic Topic Creation Setting Is Modified?

Yes. A Kafka instance will be restarted if you enable or disable automatic topic creation for it.

17.3.12 How Do I Disable Automatic Topic Creation?

1. On the Kafka console, click the name of your instance.
2. In the **Instance Information** section of the **Basic Information** tab page, click next to **Automatic Topic Creation** to disable automatic topic creation.

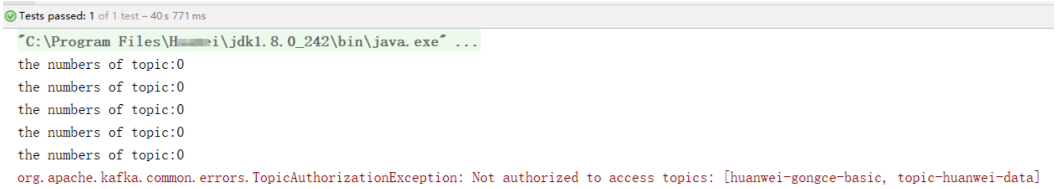
You can view the execution status of the task on the **Background Tasks** tab page.

17.3.13 Can I Delete Unnecessary Topics in a Consumer Group?

Yes, just simply unsubscribe from it on the Kafka client.

17.3.14 What Can I Do If a Consumer Fails to Retrieve Messages from a Topic Due to Insufficient Permissions?

Symptom: Different consumers in a consumer group have different topic permissions. When a consumer attempts to retrieve messages from a topic, the error message "Not authorized to access topics." is displayed, and the message retrieval fails.



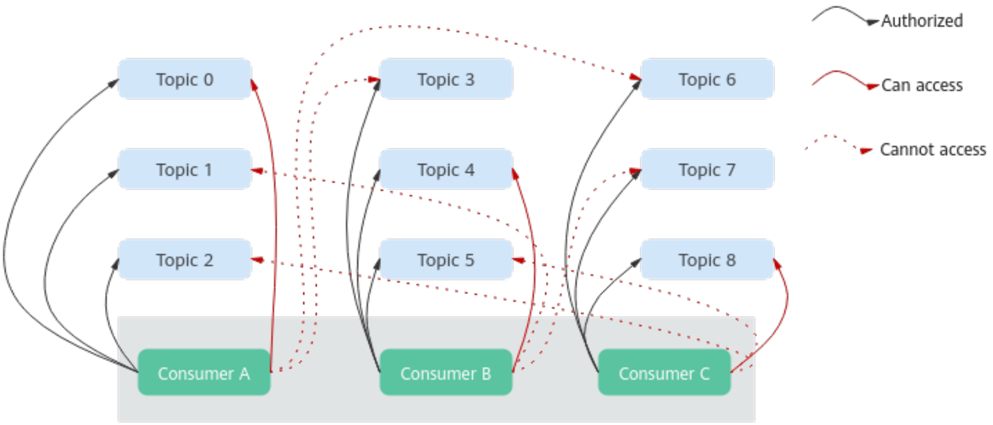
✓ Tests passed: 1 of 1 test – 40 s 771 ms

```
"C:\Program Files\H[redacted]i\jdk1.8.0_242\bin\java.exe" ...
the numbers of topic:0
the numbers of topic:0
the numbers of topic:0
the numbers of topic:0
the numbers of topic:0
org.apache.kafka.common.errors.TopicAuthorizationException: Not authorized to access topics: [huanwei-gongce-basic, topic-huanwei-data]
```

Analysis: When assigning partitions, the leader of the consumer group does not consider the permissions of individual consumers. Instead, the leader assigns partitions based on the overall subscription of the consumer group. In this case, consumers may be assigned topics that they do not have access to.

For example, consumers A, B, and C are in the same consumer group. Consumer A has subscribed to and has permissions to access Topics 0, 1, and 2; consumer B has subscribed to and has permissions to access Topics 3, 4, and 5; consumer C has subscribed to and has permissions to access Topics 6, 7, and 8. Assume that each topic has only one partition. Based on the partition assignment determined by the leader, consumer A may be assigned Topics 0, 3, and 6; consumer B is assigned Topics 1, 4, and 7; and consumer C is assigned Topics 2, 5, and 8. In this case, consumer A does not have permissions to access Topics 3 and 6, resulting in the error.

Figure 17-4 Consumer access permissions

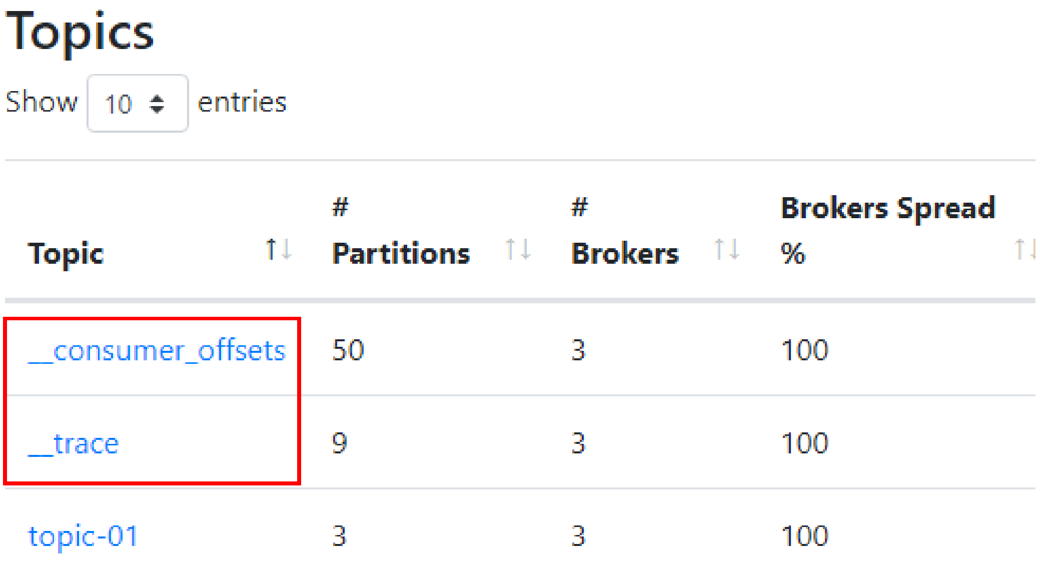


Solution:

- If all consumers must be in the same consumer group (**group.id** is the same), grant the same topic access permissions to all the consumers.
- If the consumers do not need to be in the same consumer group, change the value of **group.id** to ensure that each consumer is in a separate consumer group.

17.3.15 Why Does an Instance Contain Default Topics __trace and __consumer_offsets?

Symptom: Topics named **__trace** and **__consumer_offsets** are found on Kafka Manager.



Topics

Show 10 entries

Topic	# Partitions	# Brokers	Brokers Spread %
__consumer_offsets	50	3	100
__trace	9	3	100
topic-01	3	3	100

Handling method: **__trace** and **__consumer_offsets** are preset topics in a Kafka instance. You are not advised to delete them. If they are deleted, the instance may become unavailable.

17.4 Consumer Groups

17.4.1 Do I Need to Create Consumer Groups, Producers, and Consumers for Kafka Instances?

No. They are generated automatically when you use the instance.

For details about creating and retrieving messages after connecting to a Kafka instance, see **Accessing a Kafka Instance**.

17.4.2 Will a Consumer Group Without Active Consumers Be Automatically Deleted in 14 Days?

This depends on the **offsets.retention.minutes** parameter.

- For instances created before Jun 16, 2020, the default value of **offsets.retention.minutes** is 2,147,483,646 minutes, which is about 1,491,308 days. In this case, consumer groups will not be deleted 14 days later.
- For instances created on or after Jun 16, 2020, the default value of **offsets.retention.minutes** is 20,160 minutes, which is 14 days. In this case, consumer groups will be deleted 14 days later.

Kafka uses the **offsets.retention.minutes** parameter to control how long to keep offsets for a consumer group. If offsets are not committed within this period, they will be deleted. If Kafka determines that there are no active consumers in a consumer group (for example, when the consumer group is empty) and there are no offsets, Kafka will delete the consumer group.

17.4.3 Why Do I See a Deleted Consumer Group on Kafka Manager?

After a consumer group is deleted on a client, it no longer exists, but may still be displayed on Kafka Manager because of Kafka Manager's cache.

Use either of the following methods to solve the problem:

- Restart Kafka Manager.
- Kafka Manager displays only the consumer groups that have retrieval records in the last 14 days. If you do not want to restart Kafka Manager, wait for 14 days until the consumer group is automatically cleared.

17.5 Messages

17.5.1 What Is the Maximum Size of a Message that Can be Created?

10 MB.

17.5.2 Why Does Message Poll Often Fail During Rebalancing?

Rebalancing is a process where partitions of topics are re-allocated for a consumer group.

In normal cases, rebalancing occurs inevitably when a consumer is added to or removed from a consumer group. However, if a consumer is regarded as abnormal and removed from the consumer group, message retrieval may fail.

This may happen in the following scenarios:

1. Heartbeat requests are not sent in time.

   A consumer sends heartbeat requests to the broker at the interval specified by **heartbeat.interval.ms**. If the broker does not receive any heartbeat request from the consumer within the period specified by **session.timeout.ms**, the broker considers that the consumer is abnormal and removes the consumer from the consumer group, triggering rebalancing.
2. The interval between retrievals is too long.

The maximum number of messages that a consumer can retrieve at a time is specified by **max.poll.records**. In most cases, a client processes the retrieved data before starting the next retrieval. The processing may be prolonged when a large number of messages are retrieved at a time and cannot be processed within the time specified by **max.poll.interval.ms**, or when an exception occurs during the process (for example, data needs to be written to the backend database, but the backend database pressure is too high, resulting in high latency). If the consumer does not send the next retrieval request within the time specified by **max.poll.interval.ms**, the broker considers that the consumer is inactive and removes it from the consumer group, triggering rebalancing.

Solutions and Troubleshooting Methods

Scenario 1: Heartbeat requests are not sent in time.

Solution: On the consumer client, set the value of **session.timeout.ms** to three times the value of **heartbeat.interval.ms**.

Scenario 2: The interval between retrievals is too long.

Troubleshooting methods:

1. Check the time required for processing a single message and whether the time required for processing a specified number (**max.poll.records**) of messages exceeds the time specified by **max.poll.interval.ms**.
2. Check whether message processing requires network connections, such as writing data to the database and calling backend APIs, and whether the backend is normal in rebalancing scenarios.

Solution: On the consumer client, decrease the value of **max.poll.records**.

17.5.3 Why Can't I Query Messages on the Console?

- **Possible cause 1**: The message has been aged.

  Solution: Change the aging time.
- **Possible cause 2**: The createTime timestamp of the message is incorrect.

  On the console, messages are queried based on the timestamp, which is generated by the client. Different clients have different processing policies. The default value may be **0** or **-1**. As a result, message may fail to be queried.

  Solution: Check whether the value of createTime is correctly configured.
- **Possible cause 3**: The disk usage exceeds 95%, and **Capacity Threshold Policy** is set to **Automatically delete**.

  If **Capacity Threshold Policy** is set to **Automatically delete**, the earliest 10% of messages will be deleted when 95% of the disk capacity is used, to ensure sufficient disk space. In this case, the messages that do not reach the aging time are also deleted and cannot be queried.

  Solution: Modify the capacity threshold policy or expand the disk capacity. If **Capacity Threshold Policy** is set to **Stop production**, new messages will no longer be created when the disk usage reaches the capacity threshold (95%), but existing messages can still be retrieved until the aging time arrives. This policy is suitable for scenarios where no data losses can be tolerated.

17.5.4 What Can I Do If Kafka Messages Are Accumulated?

Symptom: An alarm is generated for the **Accumulated Messages** metric.

Solution: Log in to Kafka Manager, find the consumer group where messages are accumulated, and check whether any consumer is retrieving messages in the consumer group. If yes, accelerate the retrieval on the service end. If no, delete unused consumer groups on the customer end.

17.5.5 Why Do Messages Still Exist After the Retention Period Elapses?

If the aging time has been set for a topic, the value of the **log.retention.hours** parameter does not take effect for the topic. The value of the **log.retention.hours** parameter takes effect only if the aging time has not been set for the topic.

Possible cause 1: Each partition of a topic consists of multiple segment files of the same size (500 MB). When the size of messages stored in a segment file reaches 500 MB, another segment file is created. Kafka deletes segment files instead of messages. Kafka requires that at least one segment file be reserved for storing messages. If the segment file in use contains aged messages, the segment file will not be deleted. Therefore, the aged messages will remain.

Solution: Wait until the segment is no longer in use or delete the topic where messages have reached their retention period.

Possible cause 2: In a topic, there is a message whose **CreateTime** is a future time. For example, assume that it is January 1, and the **CreateTime** is February 1. The message will not be aged after 72 hours from now. As a result, messages created subsequently will also not be aged.

Solution: Delete the topic where the **CreateTime** of a message is a future time.

17.5.6 Do Kafka Instances Support Delayed Message Delivery?

No.

17.5.7 How Do I View the Number of Accumulated Messages?

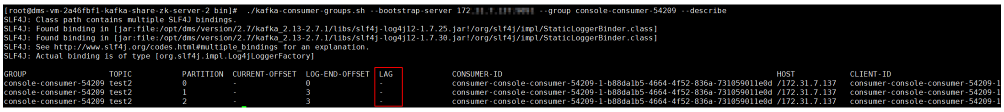
View the number of accumulated messages using any of the following methods:

- On the **Consumer Groups** page of an instance, click the name of the consumer group whose accumulated messages are to be viewed. The consumer group details page is displayed. On the **Consumer Offset** tab page, view the number of messages accumulated in each topic of your target consumer group. For details, see **Querying Consumer Group Details**.
- On the **Monitoring** tab page of an instance, click the **By Consumer Group** tab. Select the desired consumer group for **Consumer Group** and **All queues** for **Queue**. The **Consumer Available Messages** metric reflects the number of messages accumulated in all topics of this consumer group. For details about viewing the monitoring data, see **Viewing Metrics**.
- On the **Consumer Groups** tab page of the Cloud Eye console, click the **By Consumer Group** tab. Select the desired consumer group for **Consumer Group** and **All queues** for **Queue**. The **Consumer Available Messages** metric

reflects the number of messages accumulated in all topics of this consumer group. For details about viewing the monitoring data, see **Viewing Metrics**.

- On the **Kafka client**, run the **kafka-consumer-groups.sh --bootstrap-server** *{Kafka connection address}* **--describe --group** *{Consumer group}* command in the **/***{directory where the CLI is located}***/kafka_{version}/bin/** directory to view the number of messages accumulated in each topic of the consumer group. **LAG** indicates the total number of messages accumulated in each topic.

Figure 17-5 Viewing the total number of messages accumulated in each topic



```
[root@dms-vm-2a46fbf1-kafka-share-zk-server-2 bin]#  ./kafka-consumer-groups.sh --bootstrap-server 172[redacted] --group console-consumer-54209 --describe
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/opt/dms/version/2.7/kafka_2.13-2.7.1/libs/slf4j-log4j12-1.7.25.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/opt/dms/version/2.7/kafka_2.13-2.7.1/libs/slf4j-log4j12-1.7.30.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]

GROUP                  TOPIC           PARTITION  CURRENT-OFFSET  LOG-END-OFFSET  LAG             CONSUMER-ID                                                      HOST            CLIENT-ID
console-consumer-54209 test2           0          -               0               -               consumer-console-consumer-54209-1-b88da1b5-4664-4f52-836a-731059011e0d /172.31.7.137   consumer-console-consumer-54209-1
console-consumer-54209 test2           1          -               3               -               consumer-console-consumer-54209-1-b88da1b5-4664-4f52-836a-731059011e0d /172.31.7.137   consumer-console-consumer-54209-1
console-consumer-54209 test2           2          -               3               -               consumer-console-consumer-54209-1-b88da1b5-4664-4f52-836a-731059011e0d /172.31.7.137   consumer-console-consumer-54209-1
```

NOTE

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

17.5.8 Why Is the Message Creation Time Displayed as Year 1970?

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.

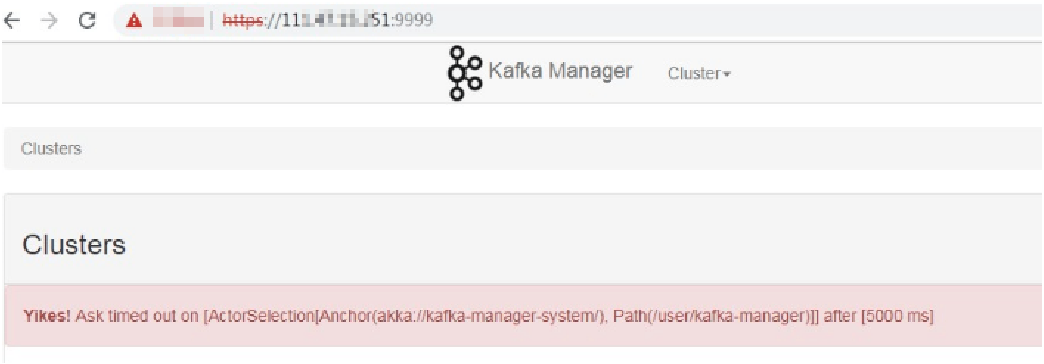
17.6 Kafka Manager

17.6.1 Can I Configure a Kafka Manager Account to Be Read-Only?

No. You cannot configure a Kafka Manager account to be read-only.

17.6.2 Why Can't I See Broker Information After Logging In to Kafka Manager?

Symptom: The Kafka Manager page is displayed, but the broker information cannot be displayed.



Clusters

Yikes! Ask timed out on [ActorSelection[Anchor(akka://kafka-manager-system/), Path(/user/kafka-manager)]] after [5000 ms]

Cause: This is an issue with the open-source Kafka. To solve the problem, contact customer service and restart Kafka Manager.

17.6.3 Yikes! Insufficient partition balance when creating topic : projectman_project_enterprise_project Try again.

Symptom:

The topic cannot be created in Kafka Manager, and the error message "Yikes! Insufficient partition balance when creating topic : projectman_project_enterprise_project Try again." is displayed.

Cause: The number of partitions exceeds the upper limit and no more topics can be created.

Solution: Increase the instance specifications, which will automatically increase the allowed number of partitions.

17.6.4 Can I Query the Body of a Message by Using Kafka Manager?

No. Kafka Manager does not support message body querying.

17.6.5 Can I Change the Port of the Kafka Manager Web UI?

No.

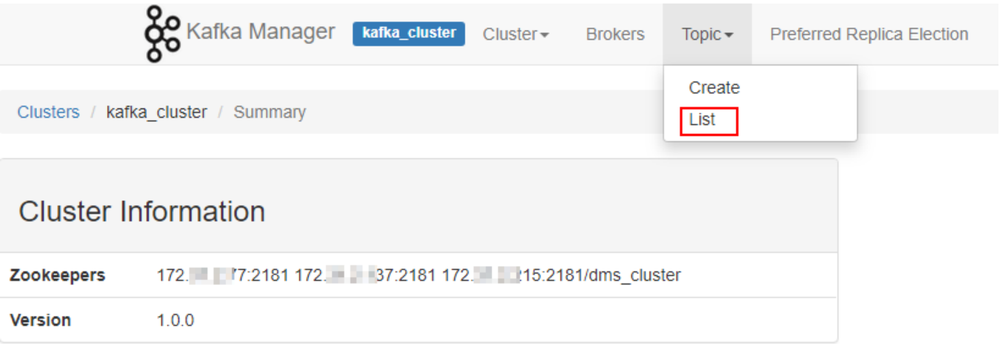
17.6.6 Which Topic Configurations Can Be Modified on Kafka Manager?

On Kafka Manager, the following topic configurations can be modified: **max.message.bytes**, **segment.index.bytes**, **segment.jitter.ms**, **min.cleanable.dirty.ratio**, **retention.bytes**, **file.delete.delay.ms**, **compression.type**, **flush.ms**, **cleanup.policy**, **unclean.leader.election.enable**, **flush.messages**, **retention.ms**, **min.insync.replicas**, **delete.retention.ms**, **preallocate**, **index.interval.bytes**, **segment.bytes**, and **segment.ms**.

Perform the following steps to modify the topic configurations:

1. **Log in to Kafka Manager**.
2. Click **kafka_cluster**.

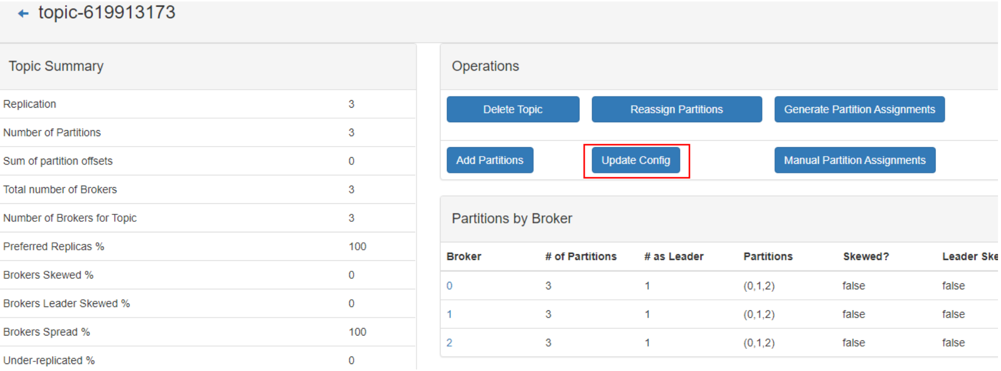
3. Choose **Topic** > **List**.


Kafka Manager
kafka_cluster
Cluster
Brokers
Topic
Preferred Replica Election
Create
List
Clusters / kafka_cluster / Summary

Cluster Information

Zookeepers	172.[redacted]7:2181 172.[redacted]37:2181 172.[redacted]15:2181/dms_cluster
Version	1.0.0

4. Click a topic whose configurations you want to modify.
5. Click **Update Config**.



Topic Summary

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

Partitions by Broker

Broker	# of Partitions	# as Leader	Partitions	Skewed?	Leader Ske
0	3	1	(0,1,2)	false	false
1	3	1	(0,1,2)	false	false
2	3	1	(0,1,2)	false	false

17.6.7 Why Is Information Displayed on Kafka Manager Inconsistent with Cloud Eye Monitoring Data?

Symptom: After a consumer group is deleted from the backend, the consumer group is not displayed on Cloud Eye but still exists on Kafka Manager.

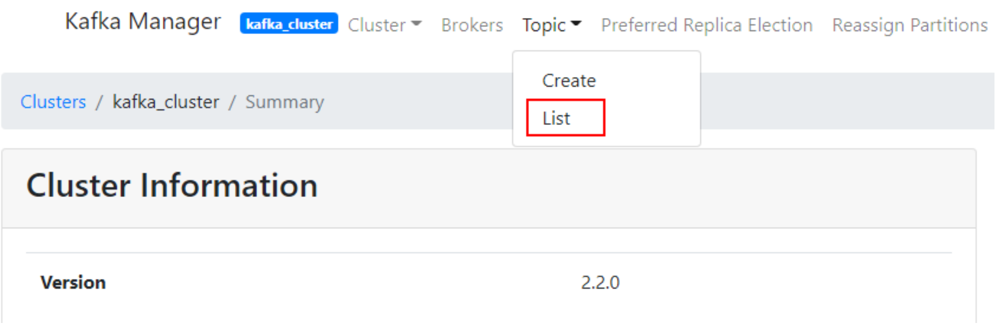
Cause: Kafka Manager has cache data.

Solution: Log in to the Kafka console, locate the row that contains the target instance, and choose **More** > **Restart Kafka Manager**.

17.6.8 How Do I Change a Partition Leader for a Topic in Kafka Manager?

Perform the following steps:

1. **Log in to Kafka Manager**.
2. Choose **Topic** > **List**.



Kafka Manager kafka_cluster Cluster Brokers Topic Preferred Replica Election Reassign Partitions

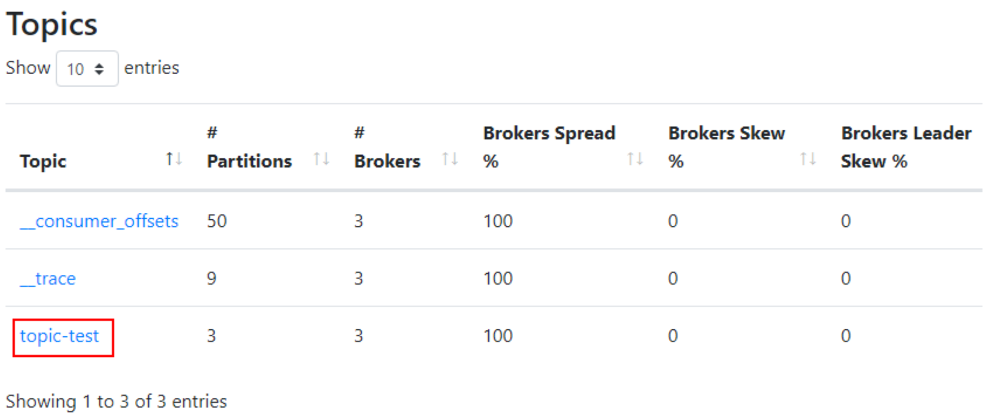
Clusters / kafka_cluster / Summary

Create
List

Cluster Information

Version	2.2.0

3. Click the topic name (for example, **topic-test**) for which a partition leader is to be modified.



Topics

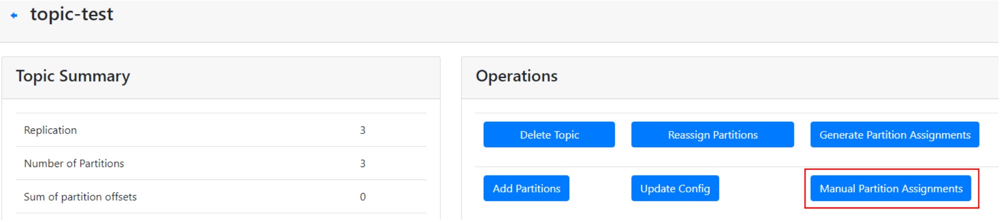
Show 10 entries

Topic	# Partitions	# Brokers	Brokers Spread %	Brokers Skew %	Brokers Leader Skew %
__consumer_offsets	50	3	100	0	0
__trace	9	3	100	0	0
topic-test	3	3	100	0	0

Showing 1 to 3 of 3 entries

4. Click **Manual Partition Assignments**.

Figure 17-6 Topic details



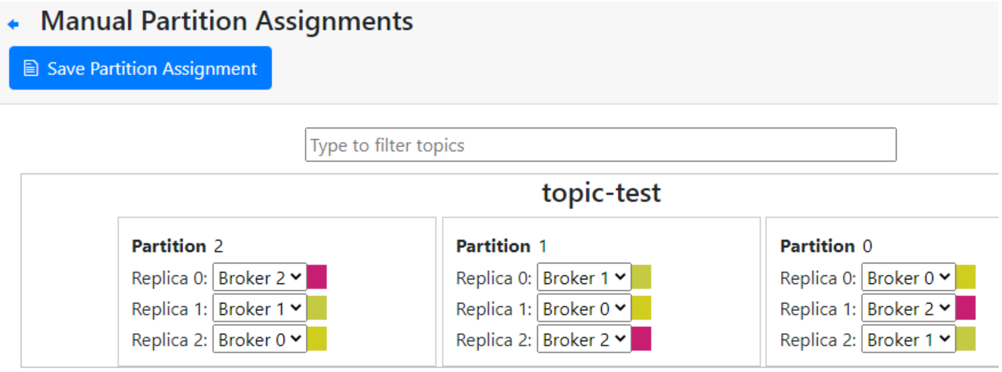
topic-test

Topic Summary

Replication	3
Number of Partitions	3
Sum of partition offsets	0

Operations

Figure 17-7 Page for modifying partition leaders



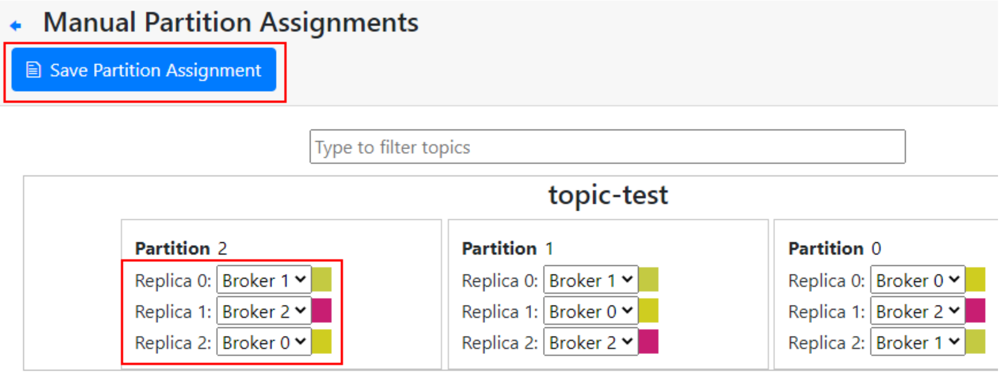
Manual Partition Assignments

topic-test

Partition 2	Partition 1	Partition 0
Replica 0: Broker 2 🟥	Replica 0: Broker 1 🟨	Replica 0: Broker 0 🟨
Replica 1: Broker 1 🟨	Replica 1: Broker 0 🟨	Replica 1: Broker 2 🟥
Replica 2: Broker 0 🟨	Replica 2: Broker 2 🟥	Replica 2: Broker 1 🟨

For example, in the preceding figure, the leader (Replica 0) of Partition 2 is on Broker 2.

5. Change the leader and click **Save Partition Assignment**.



Manual Partition Assignments

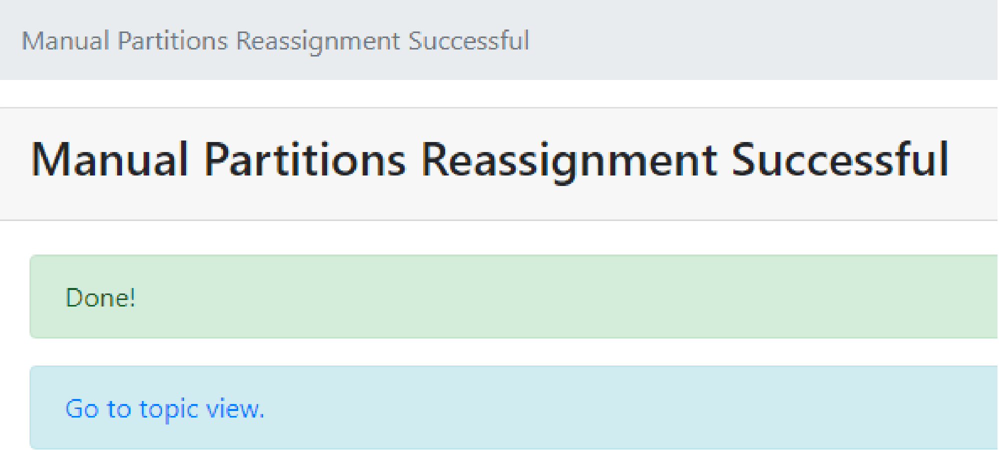
Save Partition Assignment

Type to filter topics

topic-test

Partition 2	Partition 1	Partition 0
Replica 0: Broker 1	Replica 0: Broker 1	Replica 0: Broker 0
Replica 1: Broker 2	Replica 1: Broker 0	Replica 1: Broker 2
Replica 2: Broker 0	Replica 2: Broker 2	Replica 2: Broker 1

If the modification is successful, the information shown in the following figure is displayed.



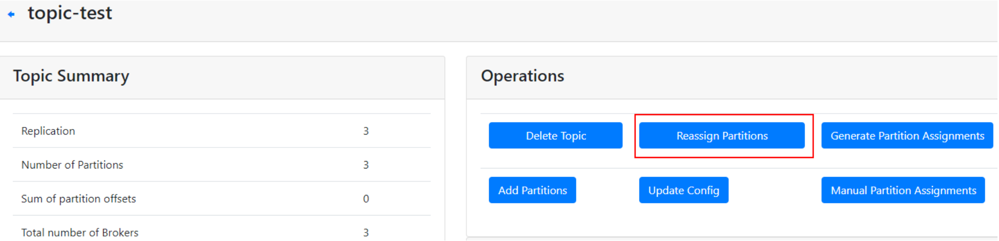
Manual Partitions Reassignment Successful

Manual Partitions Reassignment Successful

Done!

Go to topic view.

6. Click **Go to topic view**.
7. Click **Reassign Partitions** to save the change.

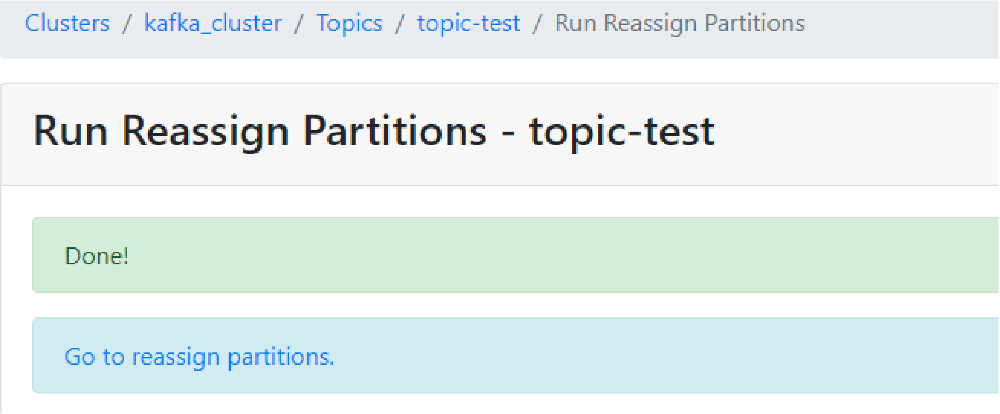


topic-test

Topic Summary

Replication	3
Number of Partitions	3
Sum of partition offsets	0
Total number of Brokers	3

After the change is saved, the information shown in the following figure is displayed.



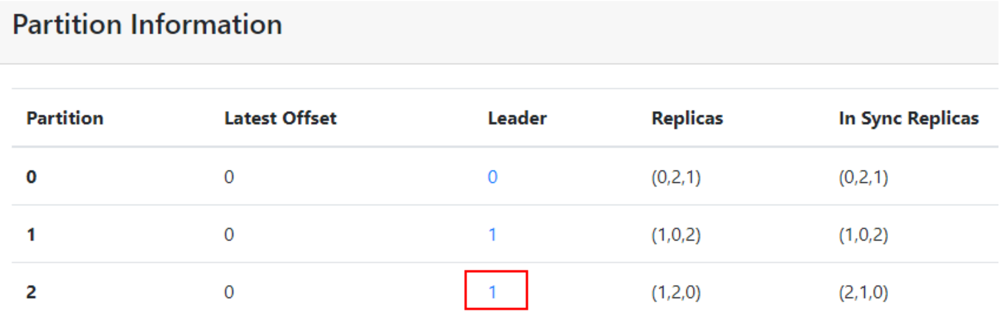
Clusters / kafka_cluster / Topics / topic-test / Run Reassign Partitions

Run Reassign Partitions - topic-test

Done!

Go to reassign partitions.

8. In the breadcrumb navigation, click the topic name. On the topic details page that is displayed, view the partition details.



Partition Information

Partition	Latest Offset	Leader	Replicas	In Sync Replicas
0	0	0	(0,2,1)	(0,2,1)
1	0	1	(1,0,2)	(1,0,2)
2	0	1	(1,2,0)	(2,1,0)

As shown in the preceding figure, the leader of partition 2 has been changed from 2 to 1.

17.7 Monitoring & Alarm

17.7.1 Why Can't I View the Monitoring Data?

If topic monitoring data is not displayed, the possible causes are as follows:

- The topic name starts with a special character, such as an underscore (_) or a number sign (#).
- No topic is created in the Kafka instance.

Solution:

- Delete topics whose names contain special characters.
- Create a topic.

If consumer group monitoring data is not displayed, the possible causes are as follows:

- The consumer group name starts with a special character, such as an underscore (_) or a number sign (#).
- No consumers in the group have connected to the instance.

17.7.2 Why Is the Monitored Number of Accumulated Messages Inconsistent with the Message Quantity Displayed on the Kafka Console?

Symptom: The monitoring data shows that there are 810 million accumulated messages. However, the Kafka console shows that there are 100 million messages in all six topics of the instance.

Analysis: The two statistics methods are different. The Kafka console shows the number of messages that have not been retrieved. The monitoring data shows the number of accumulated messages in the topics multiplied by the number of consumer groups.

17.7.3 Why Is a Consumer Group Still on the Monitoring Page After Being Deleted?

The monitoring data is reported every minute. The reported data will be displayed on the monitoring page after being sorted. This process takes less than 20 minutes. After deleting a consumer group, you can wait for a while before checking the latest monitoring data.

18 Troubleshooting

18.1 Troubleshooting Kafka Connection Exceptions

Overview

This section describes how to troubleshoot Kafka connection problems.

Problem Classification

If the connection to a Kafka instance is abnormal, perform the following operations to troubleshoot the problem:

- **Checking the Network**
- **Checking Consumer and Producer Configurations**
- **Checking for Common Errors on Java Clients**
- **Checking for Common Errors on the Go Client**

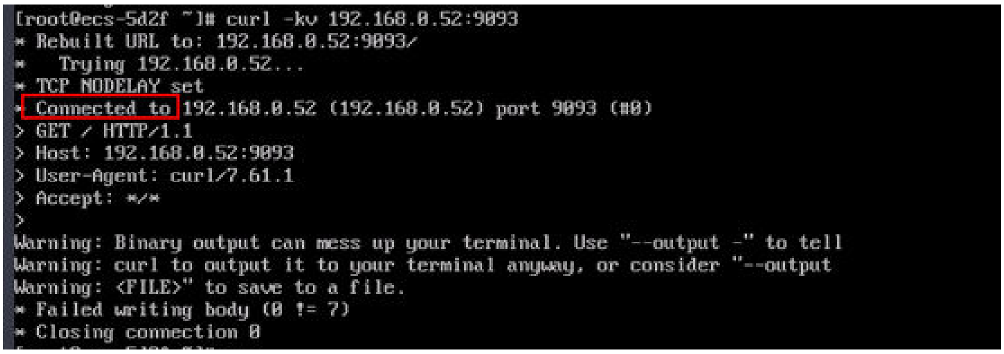
Checking the Network

Ensure that the client and the Kafka instance can be connected. If they cannot be connected, check the network.

For example, if you have enabled SASL for the Kafka instance, run the following command:

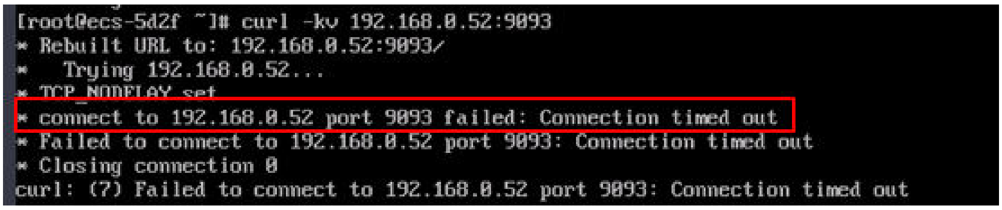
curl -kv {ip}:{port}

- If the network is normal, information similar to the following is displayed:



```
[root@ecs-5d2f ~]# curl -kv 192.168.0.52:9093
* Rebuilt URL to: 192.168.0.52:9093/
*   Trying 192.168.0.52...
* TCP_NODELAY set
* Connected to 192.168.0.52 (192.168.0.52) port 9093 (#0)
> GET / HTTP/1.1
> Host: 192.168.0.52:9093
> User-Agent: curl/7.61.1
> Accept: */*
>
Warning: Binary output can mess up your terminal. Use "--output -" to tell
Warning: curl to output it to your terminal anyway, or consider "--output
Warning: <FILE>" to save to a file.
* Failed writing body (0 != 7)
* Closing connection 0
```

- If the network is abnormal or disconnected, information similar to the following is displayed:



```
[root@ecs-5d2f ~]# curl -kv 192.168.0.52:9093
* Rebuilt URL to: 192.168.0.52:9093/
*   Trying 192.168.0.52...
* TCP_NODELAY set
* connect to 192.168.0.52 port 9093 failed: Connection timed out
* Failed to connect to 192.168.0.52 port 9093: Connection timed out
* Closing connection 0
curl: (7) Failed to connect to 192.168.0.52 port 9093: Connection timed out
```

Solution:

1. Check whether the client and the Kafka instance are in the same VPC. If they are not in the same VPC, **establish a VPC peering connection**
2. Check whether the security group rules are correctly configured. For details, see **How Do I Select and Configure a Security Group?**

Checking Consumer and Producer Configurations

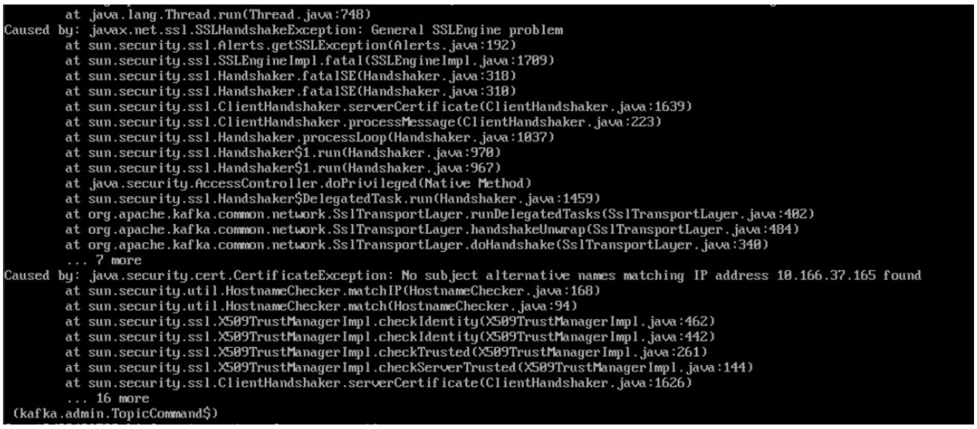
View logs to check whether the parameters printed during initialization of the consumer and producer are the same as those set in the configuration files.

If they are different, check the parameters in the configuration files.

Checking for Common Errors on Java Clients

- Error 1: Domain name verification is not disabled.

  The following error information is displayed:



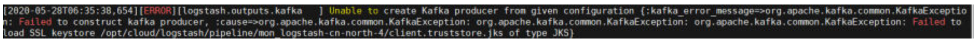
```
        at java.lang.Thread.run(Thread.java:748)
Caused by: javax.net.ssl.SSLHandshakeException: General SSLEngine problem
        at sun.security.ssl.Alerts.getSSLException(Alerts.java:192)
        at sun.security.ssl.SSLEngineImpl.fatal(SSLEngineImpl.java:1709)
        at sun.security.ssl.Handshaker.fatalSE(Handshaker.java:318)
        at sun.security.ssl.Handshaker.fatalSE(Handshaker.java:310)
        at sun.security.ssl.ClientHandshaker.serverCertificate(ClientHandshaker.java:1639)
        at sun.security.ssl.ClientHandshaker.processMessage(ClientHandshaker.java:223)
        at sun.security.ssl.Handshaker.processLoop(Handshaker.java:1037)
        at sun.security.ssl.Handshaker$1.run(Handshaker.java:970)
        at sun.security.ssl.Handshaker$1.run(Handshaker.java:967)
        at java.security.AccessController.doPrivileged(Native Method)
        at sun.security.ssl.Handshaker$DelegatedTask.run(Handshaker.java:1459)
        at org.apache.kafka.common.network.SslTransportLayer.runDelegatedTasks(SslTransportLayer.java:402)
        at org.apache.kafka.common.network.SslTransportLayer.handshakeUnwrap(SslTransportLayer.java:484)
        at org.apache.kafka.common.network.SslTransportLayer.doHandshake(SslTransportLayer.java:340)
        ... 7 more
Caused by: java.security.cert.CertificateException: No subject alternative names matching IP address 10.166.37.165 found
        at sun.security.util.HostnameChecker.matchIP(HostnameChecker.java:168)
        at sun.security.util.HostnameChecker.match(HostnameChecker.java:94)
        at sun.security.ssl.X509TrustManagerImpl.checkIdentity(X509TrustManagerImpl.java:462)
        at sun.security.ssl.X509TrustManagerImpl.checkIdentity(X509TrustManagerImpl.java:442)
        at sun.security.ssl.X509TrustManagerImpl.checkTrusted(X509TrustManagerImpl.java:261)
        at sun.security.ssl.X509TrustManagerImpl.checkServerTrusted(X509TrustManagerImpl.java:144)
        at sun.security.ssl.ClientHandshaker.serverCertificate(ClientHandshaker.java:1626)
        ... 16 more
 (kafka.admin.TopicCommand$)
```

  Solution: Leave the **ssl.endpoint.identification.algorithm** parameter in the **consumer.properties** and **producer.properties** files empty to disable domain name verification.

```
ssl.endpoint.identification.algorithm=
```

- Error 2: SSL certificates fail to be loaded.

  The following error information is displayed:

  ```
  [2020-05-28T06:35:38,654][ERROR][logstash.outputs.kafka   ] Unable to create Kafka producer from given configuration {:kafka_error_message=>org.apache.kafka.common.KafkaExceptio
  n: Failed to construct kafka producer, :cause=>org.apache.kafka.common.KafkaException: org.apache.kafka.common.KafkaException: org.apache.kafka.common.KafkaException: Failed to
  load SSL keystore /opt/cloud/logstash/pipeline/mon_logstash-cn-north-4/client.truststore.jks of type JKS}
  ```

  Solution:

  a. Check whether the **client.truststore.jks** file exists in the corresponding address.

  b. Check the permissions on the processes and files.

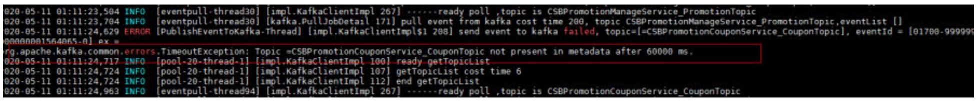
  c. Check whether the **ssl.truststore.password** parameter in the **consumer.properties** and **producer.properties** files is correctly set.

     ssl.truststore.password is the server certificate password, which must be set to **dms@kafka** and cannot be changed.

     ```
     ssl.truststore.password=dms@kafka
     ```

- Error 3: The topic name is incorrect.

  The following error information is displayed:

  ```
  020-05-11 01:11:23,504 INFO  [eventpull-thread30] [impl.KafkaClientImpl 267] ------ready poll ,topic is CSBPromotionManageService_PromotionTopic
  020-05-11 01:11:23,704 INFO  [eventpull-thread30] [kafka.PullJobDetail 171] pull event from kafka cost time 200, topic CSBPromotionManageService_PromotionTopic,eventList []
  020-05-11 01:11:24,629 ERROR [PublishEventToKafka-Thread] [impl.KafkaClientImpl$1 208] send event to kafka failed, topic=[=CSBPromotionCouponService_CouponTopic], eventId = [01700-99999
  00000001564065-0] ex =
  rg.apache.kafka.common.errors.TimeoutException: Topic =CSBPromotionCouponService_CouponTopic not present in metadata after 60000 ms.
  020-05-11 01:11:24,717 INFO  [pool-20-thread-1] [impl.KafkaClientImpl 100] ready getTopicList
  020-05-11 01:11:24,724 INFO  [pool-20-thread-1] [impl.KafkaClientImpl 107] getTopicList cost time 6
  020-05-11 01:11:24,724 INFO  [pool-20-thread-1] [impl.KafkaClientImpl 112] end getTopicList
  020-05-11 01:11:24,963 INFO  [eventpull-thread94] [impl.KafkaClientImpl 267] ------ready poll ,topic is CSBPromotionCouponService_CouponTopic
  ```

  Solution: Create a new topic or enable the automatic topic creation function.

Checking for Common Errors on the Go Client

The Go client fails to connect to Kafka over SSL and the error "first record does not look like a TLS handshake" is returned.

Solution: Enable the TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 cipher suite (which is disabled by default).

18.2 Troubleshooting 6-Min Latency Between Message Creation and Retrieval

Symptom

The duration from message creation to retrieval occasionally reaches 6 minutes, which is not tolerable to services.

Possible Causes

1. Service requests are stacked and cannot be processed in time.

   According to the monitoring data, only up to 50 messages are stacked and up to 10 messages are created per second, which is within the processing capability limit, so this is not the cause of the symptom.

2. The EIP inbound traffic decreases.

   If the EIP technical support personnel cannot find any problem, this is not the cause of the symptom.

3. The consumer group is behaving abnormally.

According to the server logs, the consumer group is going through frequent rebalance operations. While most rebalance operations are completed within seconds, some can take several minutes. Messages cannot be retrieved until the rebalance is complete.

This is the cause of the symptom.

Detailed Analysis

A consumer group may exhibit the following three types of behavior in the log:

- **Preparing to rebalance group 1**

  The consumer group starts rebalance, and its status changes to **REABLANCING**.

- **Stabilized group**

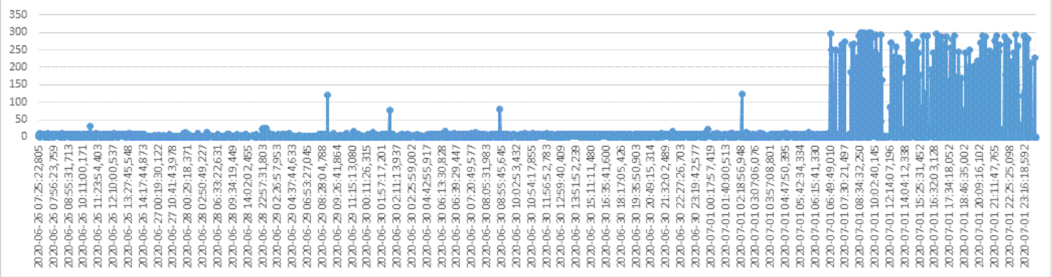
  The consumer group completes rebalance, and its status changes to **STABILIZED**.

- **Member consumer-1-0e5db2c6-a9ff-4ad4-a332-1e5b288c8aea in group 1 has failed**

  A consumer in a consumer group leaves the group if **the consumer has not communicated with the server for a long time**. This is usually triggered if the message processing is prolonged and the process is blocked.

The following figure shows the duration between **Preparing** and **Stabilized**. **The time shown in the figure is UTC+0.**

Figure 18-1 Consumer group rebalance



This set of data shows that rebalance performance of the consumer group deteriorates after 06:49 on July 1. As a result, the client becomes abnormal.

Root Cause

Sometimes, **a consumer cannot respond to rebalancing in a timely manner. As a result, the entire consumer group is blocked** until the consumer responds.

Workaround

1. Use different consumer groups for different services to reduce the impact of a single consumer blocking access.
2. **max.poll.interval.ms** sets the maximum interval for a consumer group to request message consumption. If a consumer does not initiate another consumption request before timeout, the server triggers rebalancing. You can increase the default value of **max.poll.interval.ms**.

Solution

1. Use different consumer groups for different services.
2. Optimize the service processing logic to improve the processing efficiency and reduce the blocking time.

Background Knowledge

A consumer group can be either **REBALANCING** or **STABILIZED**.

- **REBALANCING**: If a consumer joins or leaves a consumer group, the metadata of the consumer group changes and **no consumers in the consumer group can retrieve messages**.
- **STABILIZED**: The metadata has been synchronized by all consumers in the consumer group, including existing ones. Rebalancing has completed and the consumer group is stabilized. Consumers in the consumer group **can retrieve messages normally**.

A consumer group works as follows:

1. A consumer leaves or joins the group, changing the consumer group metadata recorded at the server. The server updates the consumer group status to **REBALANCING**.
2. The server **waits for all consumers** (including existing ones) to synchronize the latest metadata.
3. After **all consumers** have synchronized the latest metadata, the server updates the consumer group status to **STABILIZED**.
4. Consumers retrieve messages.

18.3 Troubleshooting Message Creation Failures

Symptom

The system displays the error message "Disk error when trying to access log file on the disk".

Root Cause

The disk usage of the broker is too high.

Solution

Expand the disk space by referring to **Modifying Instance Specifications**.

18.4 Troubleshooting Topic Deletion Failures

Symptom

A deleted topic still exists.

Root Cause

Automatic topic creation has been enabled for the instance, and a consumer is connecting to the topic. If services are not stopped, message creation will continue, and new topics will be automatically created.

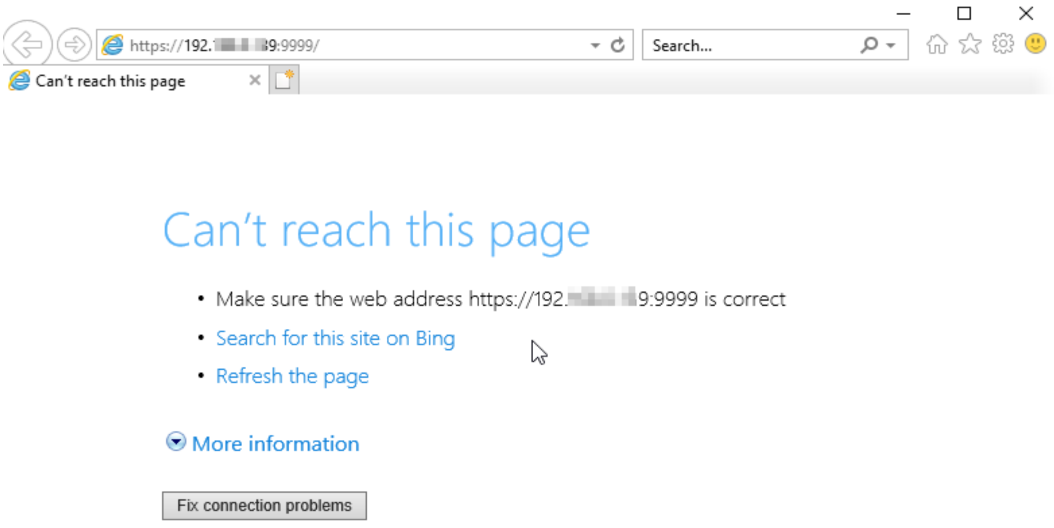
Solution

Disable automatic topic creation for the instance and then try again to delete the topic.

18.5 Troubleshooting Failure to Log In to Kafka Manager in Windows

Symptom

After the Kafka Manager address is entered in the address box of the browser in Windows, the login fails and an error is displayed.



Can't reach this page

- Make sure the web address https://192.[redacted]9:9999 is correct
- Search for this site on Bing
- Refresh the page

More information

Fix connection problems

Root Cause

1. The Windows server and the Kafka instance are not in the same VPC and subnet, or the security group configurations are incorrect.
2. Kafka Manager is abnormal.

Solution

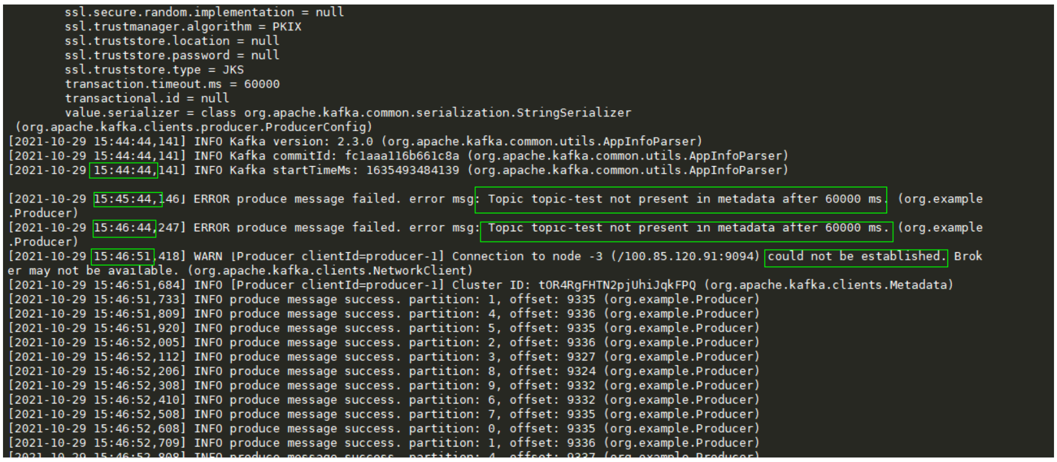
1. Check whether the Windows server and the Kafka instance are in the same VPC and subnet.
   - If they are in the same VPC and subnet, go to **2**.
   - If they are not in the same VPC and subnet, change the VPC and subnet of the Windows server to the same as those of the Kafka instance.

2. Check whether the security group is correctly configured. For details on how to configure a security group, see **How Do I Select and Configure a Security Group?**
   - If the security group is correctly configured, go to **3**.
   - If the security group is not correctly configured, modify the configuration.
3. On the Kafka console, restart Kafka Manager. For details, see **Restarting Kafka Manager**.

18.6 Troubleshooting Error "Topic {{topic_name}} not present in metadata after 60000 ms" During Message Production or Consumption

Symptom

For a Kafka instance deployed in multiple AZs, if one of the AZs is faulty, error message "Topic {{topic_name}} not present in metadata after 60000 ms" may be reported on the Kafka client during message production or consumption, as shown in the following figure.



```
        ssl.secure.random.implementation = null
        ssl.trustmanager.algorithm = PKIX
        ssl.truststore.location = null
        ssl.truststore.password = null
        ssl.truststore.type = JKS
        transaction.timeout.ms = 60000
        transactional.id = null
        value.serializer = class org.apache.kafka.common.serialization.StringSerializer
 (org.apache.kafka.clients.producer.ProducerConfig)
[2021-10-29 15:44:44,141] INFO Kafka version: 2.3.0 (org.apache.kafka.common.utils.AppInfoParser)
[2021-10-29 15:44:44,141] INFO Kafka commitId: fc1aaa116b661c8a (org.apache.kafka.common.utils.AppInfoParser)
[2021-10-29 15:44:44,141] INFO Kafka startTimeMs: 1635493484139 (org.apache.kafka.common.utils.AppInfoParser)

[2021-10-29 15:45:44,146] ERROR produce message failed. error msg: Topic topic-test not present in metadata after 60000 ms. (org.example
.Producer)
[2021-10-29 15:46:44,247] ERROR produce message failed. error msg: Topic topic-test not present in metadata after 60000 ms. (org.example
.Producer)
[2021-10-29 15:46:51,418] WARN [Producer clientId=producer-1] Connection to node -3 (/100.85.120.91:9094) could not be established. Brok
er may not be available. (org.apache.kafka.clients.NetworkClient)
[2021-10-29 15:46:51,684] INFO [Producer clientId=producer-1] Cluster ID: tOR4RgFHTN2pjUhiJqkFPQ (org.apache.kafka.clients.Metadata)
[2021-10-29 15:46:51,733] INFO produce message success. partition: 1, offset: 9335 (org.example.Producer)
[2021-10-29 15:46:51,809] INFO produce message success. partition: 4, offset: 9336 (org.example.Producer)
[2021-10-29 15:46:51,920] INFO produce message success. partition: 5, offset: 9335 (org.example.Producer)
[2021-10-29 15:46:52,005] INFO produce message success. partition: 2, offset: 9336 (org.example.Producer)
[2021-10-29 15:46:52,112] INFO produce message success. partition: 3, offset: 9327 (org.example.Producer)
[2021-10-29 15:46:52,206] INFO produce message success. partition: 8, offset: 9324 (org.example.Producer)
[2021-10-29 15:46:52,308] INFO produce message success. partition: 9, offset: 9332 (org.example.Producer)
[2021-10-29 15:46:52,410] INFO produce message success. partition: 6, offset: 9332 (org.example.Producer)
[2021-10-29 15:46:52,508] INFO produce message success. partition: 7, offset: 9335 (org.example.Producer)
[2021-10-29 15:46:52,608] INFO produce message success. partition: 0, offset: 9335 (org.example.Producer)
[2021-10-29 15:46:52,709] INFO produce message success. partition: 1, offset: 9336 (org.example.Producer)
[2021-10-29 15:46:52,808] INFO produce message success. partition: 4, offset: 9337 (org.example.Producer)
```

Solution

You can use any of the following methods to solve this problem:

- Upgrade the Kafka client to v2.7 or later, and set **socket.connection.setup.timeout.ms** to a value greater than 1s and less than the value of **request.timeout.ms** divided by the number of Kafka server nodes.
- Change the value of **request.timeout.ms** of the Kafka client to a value greater than 127s.
- Change the Linux network parameter **net.ipv4.tcp_syn_retries** of the Kafka client to **3**.

A Change History

Date	Description
2023-03-09	This release incorporates the following change: - Added the disk encryption function, as described in section **Buying an Instance**. - Added **Getting Started**.
2023-02-08	This release incorporates the following changes: - Added description about new specifications in **Specifications**, **Buying an Instance**, and **Modifying Instance Specifications**. - Added support for SASL/PLAIN in sections **Buying an Instance** and **Accessing a Kafka Instance with SASL**. - Added **Viewing Disk Usage**, **Partition Reassignment**, and **Managing Kafka Quotas**.
2022-10-27	This release incorporates the following changes: - Added **Troubleshooting** - Added sections **Viewing Sample Code**, **Managing Consumer Groups**, and **Modifying Kafka Parameters**. - Added support for Kafka v2.7 in sections **Specifications**, **Buying an Instance**, and **Which Kafka Versions Are Supported?**.
2022-01-24	This release incorporates the following changes: - Supported public access to instances, as described in **Configuring Public Access**. - Supported users' topic permission settings, as described in **Managing Users**. - Allowed you to change the number of topic partitions on the console, as described in **Changing Partition Quantity**.

Date	Description
2021-11-26	This release incorporates the following changes: - Added section **Billing**.
2020-11-06	This issue is the first official release.