

Distributed Message Service for Kafka

User Guide

Issue 04
Date 2024-05-28



Copyright © Huawei Cloud Computing Technologies Co., Ltd. 2024. All rights reserved.

No part of this document may be reproduced or transmitted in any form or by any means without prior written consent of Huawei Cloud Computing Technologies Co., Ltd.

Trademarks and Permissions



HUAWEI and other Huawei trademarks are the property of Huawei Technologies Co., Ltd.

All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

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

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

Contents

1 Service Overview	1
1.1 What Is DMS for Kafka?	1
1.2 Product Advantages	1
1.3 Application Scenarios	2
1.4 Specifications	4
1.5 Comparing Kafka, RabbitMQ, and RocketMQ	8
1.6 Comparing DMS for Kafka and Open-Source Kafka	10
1.7 Notes and Constraints	12
1.8 Related Services	16
1.9 Basic Concepts	17
1.10 Permissions Management	18
2 Getting Started	23
2.1 Getting Started with Kafka to Produce and Consume Messages	23
3 Process of Using Kafka	31
4 Permissions Management	33
4.1 Creating a User and Granting DMS for Kafka Permissions	33
5 Buying a Kafka Instance	37
6 Configuring Topics	42
6.1 Creating a Kafka Topic	42
6.2 Configuring Kafka Topic Permissions	47
6.3 Managing Topics	49
6.3.1 Viewing Kafka Topic Details	49
6.3.2 Modifying Kafka Topic Configurations	51
6.3.3 Changing Kafka Partition Quantity	52
6.3.4 Exporting the Kafka Topic List	55
6.3.5 Reassigning Kafka Partitions	55
6.3.6 Configuring Automatic Topic Creation	67
6.3.7 Deleting a Kafka Topic	68
7 Connecting to an Instance	70
7.1 Configuring Kafka Network Connections	70
7.1.1 Kafka Network Connection Conditions	70

7.2 Configuring Kafka Access Control.....	71
7.2.1 Configuring Kafka Users.....	71
7.3 Configuring the Kafka Client.....	74
7.3.1 Setting Parameters for Kafka Clients.....	74
7.3.2 Suggestions on Using the Kafka Client.....	77
7.4 Connecting to Kafka Using the Client (SASL Disabled).....	79
7.5 Connecting to Kafka Using the Client (SASL Enabled).....	81
7.6 Connecting to Kafka on the Console.....	85
8 Managing Messages.....	87
8.1 Viewing Kafka Messages.....	87
8.2 Changing Kafka Message Retention Period.....	89
8.3 Deleting Kafka Messages.....	90
9 Managing Consumer Groups.....	93
9.1 Creating a Kafka Consumer Group.....	93
9.2 Querying the Kafka Consumer Group List.....	94
9.3 Viewing Kafka Consumer Details.....	95
9.4 Viewing and Resetting Kafka Consumption Offsets.....	98
9.5 Exporting Kafka Consumer Groups.....	101
9.6 Deleting a Kafka Consumer Group.....	101
10 Managing Quotas.....	104
10.1 Configuring Kafka Quotas.....	104
10.2 Monitoring Kafka Quotas.....	108
11 Managing Instances.....	110
11.1 Viewing and Modifying Basic Information of a Kafka Instance.....	110
11.2 Viewing Kafka Disk Usage.....	112
11.3 Viewing Kafka Background Tasks.....	113
11.4 Viewing Sample Code of Kafka Production and Consumption.....	114
11.5 Modifying Kafka Instance Configuration Parameters.....	114
11.6 Configuring Kafka Instance Tags.....	120
11.7 Exporting the Kafka Instance List.....	121
11.8 Restarting a Kafka Instance.....	122
11.9 Deleting Kafka Instances.....	123
11.10 Using Kafka Manager.....	123
11.10.1 Kafka Manager Overview.....	124
11.10.2 Resetting Kafka Manager Password.....	130
11.10.3 Restarting Kafka Manager.....	131
12 Modifying Kafka Instance Specifications.....	133
13 Migrating Data.....	136
13.1 Kafka Data Migration Overview.....	136
14 Applying for Increasing Kafka Quotas.....	139

15 Monitoring and Alarms	141
15.1 Viewing Kafka Monitoring Metrics	141
15.2 Kafka Metrics	142
15.3 Configuring a Kafka Alarm Rule	153
16 Viewing Kafka Audit Logs	157
17 FAQs	161
17.1 Instances	161
17.1.1 Why Can't I Select Two AZs?	161
17.1.2 Why Can't I View the Subnet and Security Group Information When Creating a DMS Instance?..	161
17.1.3 How Do I Select Storage Space for a Kafka Instance?	161
17.1.4 How Do I Choose Between High I/O and Ultra-high I/O?	162
17.1.5 Which Capacity Threshold Policy Should I Use?	162
17.1.6 Which Kafka Versions Are Supported?	162
17.1.7 What Is the ZooKeeper Address of a Kafka Instance?	162
17.1.8 Are Kafka Instances in Cluster Mode?	162
17.1.9 Can I Modify the Port for Accessing a Kafka Instance?	163
17.1.10 How Long Are Kafka SSL Certificates Valid for?	163
17.1.11 How Do I Synchronize Data from One Kafka Instance to Another?	164
17.1.12 How Do I Change the SASL_SSL Setting of a Kafka Instance?	164
17.1.13 How Do I Modify the SASL Mechanism?	164
17.1.14 Will a Kafka Instance Be Restarted After Its Enterprise Project Is Modified?	164
17.1.15 Are Kafka Brokers and ZooKeeper Deployed on the Same VM or on Different VMs?	164
17.1.16 Which Cipher Suites Are Supported by Kafka?	164
17.1.17 Can I Change an Instance from Single-AZ Deployment to Multi-AZ Deployment?	164
17.1.18 Does DMS for Kafka Support Cross-AZ Disaster Recovery? Where Can I View the AZs Configured for an Existing Instance?	164
17.1.19 Do Kafka Instances Support Disk Encryption?	165
17.1.20 Can I Change the VPC and Subnet After a Kafka Instance Is Created?	165
17.1.21 Where Can I Find Kafka Streams Use Cases?	165
17.1.22 Can I Upgrade Kafka Instances?	165
17.1.23 Why Is the Version on the Console Different from That in Kafka Manager?	165
17.2 Specification Modification	165
17.2.1 Does Specification Modification Affect Services?	165
17.2.2 Will Data Migration Be Involved When I Increase Specifications?	166
17.2.3 What Can I Do When I Fail to Increase Specifications Due to Insufficient Resources?	166
17.3 Connections	166
17.3.1 How Do I Select and Configure a Security Group?	166
17.3.2 Can I Access a Kafka Instance Over a Public Network?	168
17.3.3 How Many Connection Addresses Does a Kafka Instance Have by Default?	168
17.3.4 Do Kafka Instances Support Cross-Region Access?	169
17.3.5 Do Kafka Instances Support Cross-VPC Access?	169
17.3.6 Do Kafka Instances Support Cross-Subnet Access?	169

17.3.7 Does DMS for Kafka Support Authentication with Kerberos?.....	169
17.3.8 Does DMS for Kafka Support Password-Free Access?.....	170
17.3.9 Does DMS for Kafka Support Authentication on Clients by the Server?.....	170
17.3.10 Can I Use PEM SSL Truststore When Connecting to a Kafka Instance with SASL_SSL Enabled?...	170
17.3.11 What Are the Differences Between JKS and CRT Certificates?.....	170
17.3.12 Which TLS Version Does DMS for Kafka Support?.....	170
17.3.13 Is There a Limit on the Number of Client Connections to a Kafka Instance?.....	170
17.3.14 How Many Connections Are Allowed from Each IP Address?.....	171
17.3.15 Can I Change the Private Network Addresses of a Kafka Instance?.....	171
17.3.16 Is the Same SSL Certificate Used for Different Instances?.....	171
17.3.17 Why Is It Not Recommended to Use a Sarama Client for Messaging?.....	171
17.4 Topics and Partitions.....	172
17.4.1 Is There a Limit on the Number of Topics in a Kafka Instance?.....	172
17.4.2 Why Is Partition Quantity Limited?.....	173
17.4.3 Can I Reduce the Partition Quantity?.....	174
17.4.4 Why Do I Fail to Create Topics?.....	174
17.4.5 Do Kafka Instances Support Batch Importing Topics or Automatic Topic Creation?.....	174
17.4.6 Why Do Deleted Topics Still Exist?.....	175
17.4.7 Can I View the Disk Space Used by a Topic?.....	175
17.4.8 Can I Add ACL Permissions for Topics?.....	175
17.4.9 What Should I Do If Kafka Storage Space Is Used Up Because Retrieved Messages Are Not Deleted?.....	175
17.4.10 How Do I Increase the Partition Quantity?.....	175
17.4.11 Will a Kafka Instance Be Restarted After Its Automatic Topic Creation Setting Is Modified?.....	176
17.4.12 Can I Delete Unnecessary Topics in a Consumer Group?.....	176
17.4.13 What Can I Do If a Consumer Fails to Retrieve Messages from a Topic Due to Insufficient Permissions?.....	176
17.4.14 Why Does an Instance Contain Default Topics <code>_trace</code> and <code>_consumer_offsets</code> ?.....	177
17.5 Consumer Groups.....	178
17.5.1 Do I Need to Create Consumer Groups, Producers, and Consumers for Kafka Instances?.....	178
17.5.2 Will a Consumer Group Without Active Consumers Be Automatically Deleted in 14 Days?.....	178
17.5.3 Why Do I See a Deleted Consumer Group on Kafka Manager?.....	178
17.5.4 Why Can't I View Consumers When Instance Consumption Is Normal?.....	179
17.6 Messages.....	179
17.6.1 What Is the Maximum Size of a Message that Can be Created?.....	179
17.6.2 Why Does Message Poll Often Fail During Rebalancing?.....	179
17.6.3 Why Can't I Query Messages on the Console?.....	180
17.6.4 What Can I Do If Kafka Messages Are Accumulated?.....	180
17.6.5 Why Do Messages Still Exist After the Retention Period Elapses?.....	181
17.6.6 Do Kafka Instances Support Delayed Message Delivery?.....	181
17.6.7 How Do I View the Number of Accumulated Messages?.....	181
17.6.8 Why Is the Message Creation Time Displayed as Year 1970?.....	182
17.7 Kafka Manager.....	182

17.7.1 Can I Configure a Kafka Manager Account to Be Read-Only?.....	182
17.7.2 Why Can't I See Broker Information After Logging In to Kafka Manager?.....	182
17.7.3 Yikes! Insufficient partition balance when creating topic : projectman_project_enterprise_project Try again.....	183
17.7.4 Can I Query the Body of a Message by Using Kafka Manager?.....	183
17.7.5 Can I Change the Port of the Kafka Manager Web UI?.....	183
17.7.6 Which Topic Configurations Can Be Modified on Kafka Manager?.....	183
17.7.7 Why Is Information Displayed on Kafka Manager Inconsistent with Cloud Eye Monitoring Data?.....	184
17.7.8 How Do I Change a Partition Leader for a Topic in Kafka Manager?.....	184
17.8 Monitoring & Alarm.....	187
17.8.1 Why Can't I View the Monitoring Data?.....	187
17.8.2 Why Is the Monitored Number of Accumulated Messages Inconsistent with the Message Quantity Displayed on the Kafka Console?.....	187
17.8.3 Why Is a Consumer Group Still on the Monitoring Page After Being Deleted?.....	188
18 Troubleshooting.....	189
18.1 Troubleshooting Kafka Connection Exceptions.....	189
18.2 Troubleshooting 6-Min Latency Between Message Creation and Retrieval.....	191
18.3 Troubleshooting Message Creation Failures.....	193
18.4 Troubleshooting Topic Deletion Failures.....	193
18.5 Troubleshooting Failure to Log In to Kafka Manager in Windows.....	194
18.6 Troubleshooting Error "Topic {{topic_name}} not present in metadata after 60000 ms" During Message Production or Consumption.....	195
A Change History.....	196

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. Data is synchronized between different AZs based on Kafka's in-sync replica (ISR) mechanism. A topic must have multiple data copies and distribute them across ISRs. When ISR replication is normal, the recovery point objective (RPO) is close to 0.
- 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. Users can configure up to 200 partitions for a topic, and expand the storage space and broker quantity. 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

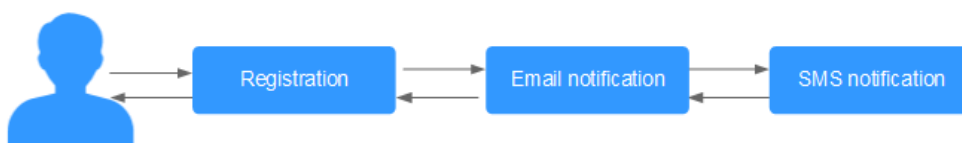


Figure 1-2 Asynchronous registration and notification using message queues

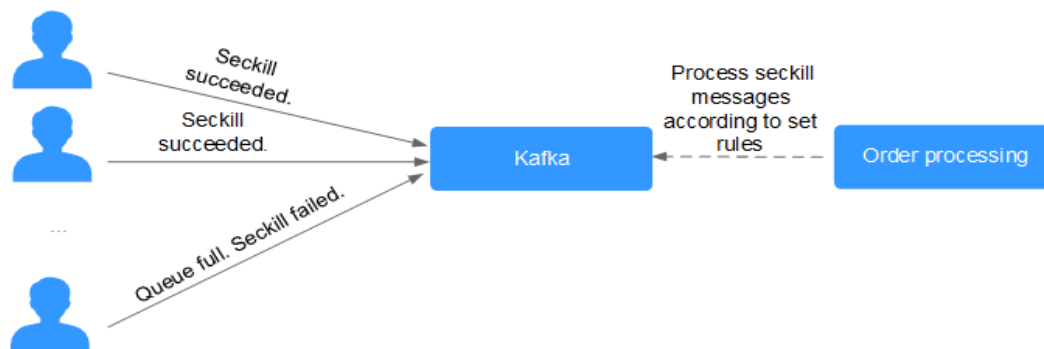


Traffic Control

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



Log Synchronization

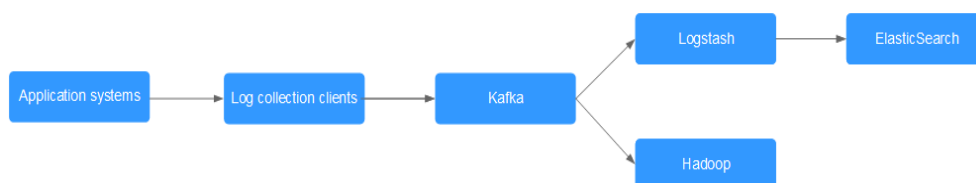
In large-scale service systems, logs of different applications are collected for quick troubleshooting, full-link tracing, and real-time monitoring.

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. Available options are kafka.2u4g.cluster, kafka.4u8g.cluster, kafka.8u16g.cluster, kafka.12u24g.cluster, and kafka.16u32g.cluster.

NOTE

For Kafka instances, the number of transactions per second (TPS) is the maximum number of messages that can be written per second. In the following table, transactions per second (TPS) are calculated assuming that the size of a message is 1 KB. The test instances are connected over the private network without SASL and use ultra-high I/O disks.

Table 1-1 Kafka instance specifications (cluster)

Flavor	Brokers	Maximum TPS per Broker	Maximum Partitions per Broker	Recommended Consumer Groups per Broker	Maximum Client Connections per Broker	Storage Space	Traffic per Broker (MB/s)
kafka.2u4g.cluster	3-30	30,000	250	20	2000	300 GB-300,000 GB	100
kafka.4u8g.cluster	3-30	100,000	500	100	4000	300 GB-600,000 GB	200
kafka.8u16g.cluster	3-50	150,000	1000	150	4000	300 GB-1,500,000 GB	375
kafka.12u24g.cluster	3-50	200,000	1500	200	4000	300 GB-1,500,000 GB	625
kafka.16u32g.cluster	3-50	250,000	2000	200	4000	300 GB-1,500,000 GB	750

Instance Specifications and Network Bandwidth

The network bandwidth of a Kafka instance consists of the following:

1. Network bandwidth used by the instance brokers
2. Bandwidth of the disk used by the instance brokers. For details, see [Disk Types and Performance](#).

Note:

- By default, Kafka tests are performed in the tail read scenario (that is, only the latest production data is consumed) instead of the cold read scenario (that is, historical data is consumed from the beginning).
- The bandwidth of an instance with an old flavor (such as 100 MB/s) is the total network bandwidth of the instance's all brokers.

Traffic calculation of instances with new flavors (such as kafka.2u4g.cluster) is described as follows:

- The read/write ratio is 1:1.

- The default number of topic replicas is 3.
- Total network traffic = Traffic per broker x Broker quantity
- Total instance traffic = Service traffic + Data replication traffic between brokers

Assume that the current flavor is kafka.2u4g.cluster, the traffic per broker is 100 MB/s, and the number of brokers is 3. What are the total network traffic, maximum read traffic, and maximum write traffic of the instance?

1. Total network traffic = Traffic per broker x Broker quantity = 100 MB/s x 3 = 300 MB/s
2. Maximum read traffic = Total instance network traffic/Default number of replicas/2 = 300 MB/s/3/2= 50 MB/s
3. Maximum write traffic = Total instance network traffic/Default number of replicas/2 = 300 MB/s/3/2 = 50 MB/s

Mapping Between Old and New Flavors

Table 1-2 compares the old and new Kafka instance flavors.

Table 1-2 Mapping between old and new Kafka instance flavors

Old Flavor		New Flavor	
Flavor	Total Instance Network Traffic	Flavor	Total Instance Network Traffic
300 MB/s	300 MB/s	kafka.2u4g.cluster * 3	300 MB/s
600 MB/s	600 MB/s	kafka.4u8g.cluster * 3	600 MB/s
1200 MB/s	1200 MB/s	kafka.4u8g.cluster * 6	1250 MB/s

Instances with new flavors have the following features:

- Better performance and cost effectiveness: They use exclusive resources. By contrast, old flavors use non-exclusive resources. If the load is heavy, resources conflicts will occur.
- Latest functions, for example, reassigning partitions.
- Flexible disk capacity: Only related to the broker quantity, and not to the flavor.
- More specification options: A wider range of combinations of broker flavor (over 10,000 MB/s) and quantity are available.

Flavor Selection

- kafka.2u4g.cluster with 3 brokers
Recommended for up to 6000 client connections, 60 consumer groups, and 90,000 TPS
- kafka.4u8g.cluster with 3 brokers

- Recommended for up to 12,000 client connections, 300 consumer groups, and 300,000 TPS
- kafka.8u16g.cluster with 3 brokers
Recommended for up to 12,000 client connections, 450 consumer groups, and 450,000 TPS
- kafka.12u24g.cluster with 3 brokers
Recommended for up to 12,000 client connections, 600 consumer groups, and 600,000 TPS
- kafka.16u32g.cluster with 3 brokers
Recommended for up to 12,000 client connections, 600 consumer groups, and 750,000 TPS

Storage Space Selection

Kafka instances can store messages in multiple replicas. The storage space is consumed by message replicas, logs, and metadata. When creating an instance, specify its storage space based on the expected service message size, the number of replicas, and reserved disk space. Each Kafka broker reserves 33 GB disk space for storing logs and metadata.

For example, if the expected service message size is 100 GB, the number of replicas is 2, and the number of brokers is 3, the disk size should be at least 299 GB (100 GB × 2 + 33 GB × 3).

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

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

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

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

Synchronous Replication ?

Synchronous Flushing ?

Message Timestamp ?

Max.Message Size (bytes) ?

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

Table 1-3 Functions

Feature	RocketMQ	Kafka	RabbitMQ
Priority queue	Not supported	Not supported	Supported. It is recommended that the priority be set to 0–10.

Feature	RocketMQ	Kafka	RabbitMQ
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.	Supports SSL.
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 1-4 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
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.

Category	Item	DMS for Kafka	Open-source Kafka
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.
	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 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 1-5 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	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> The storage space can be expanded but cannot be reduced. You can expand the storage space up to 20 times.

Item	Notes and Constraints
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.
Kerberos authentication	Not supported
Client connections from each IP address	Each Kafka broker allows a maximum of 1000 connections from each IP address by default. Excess connections will be rejected.

Topic

Table 1-6 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 .

Item	Notes and Constraints
Automatic topic creation	<p>Supported. If this option is enabled, a topic will be automatically created when a message is produced in or consumed from a topic that does not exist. By default, the topic has the following parameters:</p> <ul style="list-style-type: none"> ● Partitions: 3 ● Replicas: 3 ● Aging Time: 72 ● Synchronous Replication and Synchronous Flushing disabled ● Message Timestamp: CreateTime ● Max.Message Size (bytes): 10,485,760 <p>After you change the value of the log.retention.hours, default.replication.factor, or num.partitions parameter, the value will be used in later topics that are automatically created.</p> <p>For example, assume that num.partitions is changed to 5, an automatically created topic has the following parameters:</p> <ul style="list-style-type: none"> ● Partitions: 5 ● Replicas: 3 ● Aging Time: 72 ● Synchronous Replication and Synchronous Flushing disabled ● Message Timestamp: CreateTime ● Max.Message Size (bytes): 10,485,760
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.
Aging time	<p>The value of the log.retention.hours parameter takes effect only if the aging time has not been set for the topic.</p> <p>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.</p>
Batch importing and exporting topics	Batch export is supported, but batch import is not supported.

Item	Notes and Constraints
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
Broker faults	When some brokers of an instance are faulty, topics cannot be created, modified, or deleted, but can be queried.

Consumer Group

Table 1-7 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.
Broker faults	When some instance brokers are faulty, consumer groups cannot be created or deleted, or consumption progress cannot be reset, but consumer groups can be queried.

Message

Table 1-8 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 1-9 User notes and constraints

Item	Notes and Constraints
Number of users	The maximum users that can be created for a Kafka instance is 20 or 500. Check the console for the actual limit.
Broker faults	When some instance brokers are faulty, users cannot be created or deleted, or password cannot be reset, but users can be queried.

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.
- **Identity and Access Management (IAM)**
IAM enables you to easily manage users and control their access to cloud services and resources. Grant different users different Kafka permissions required to perform a given task based on their job responsibilities.
- **Cloud Eye (CES)**
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.
- **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.

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

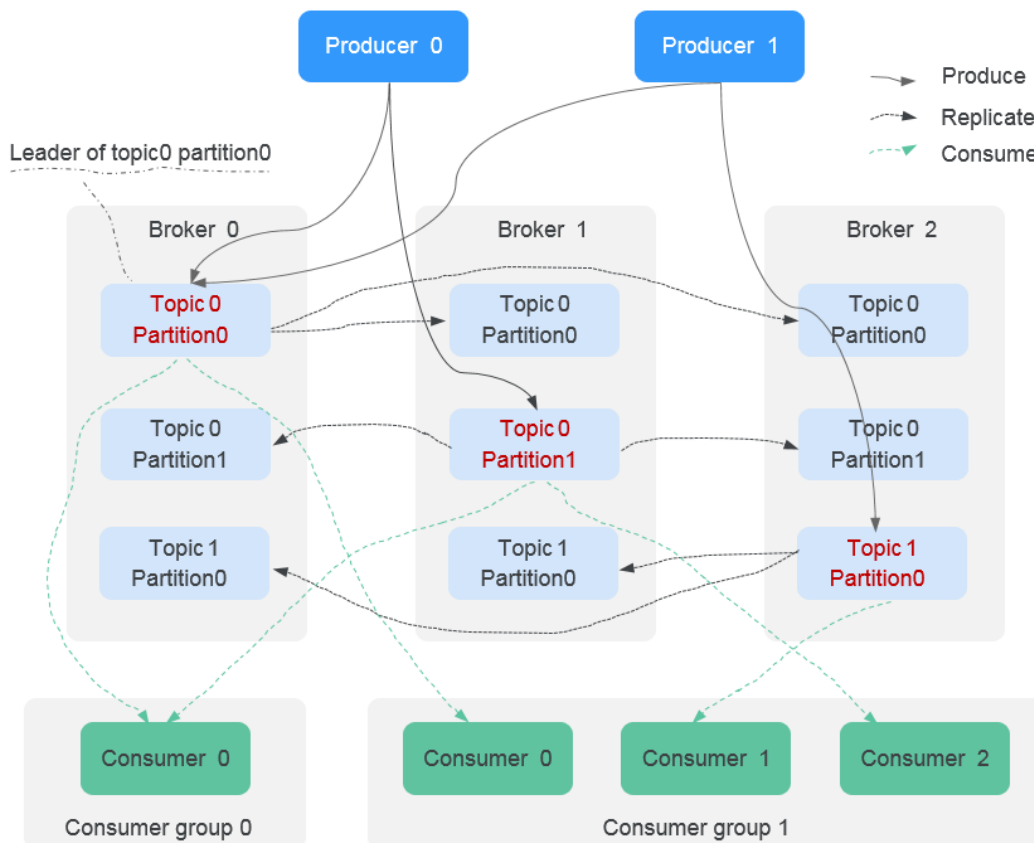
A replica is a redundant copy of a partition in a topic. Each partition can have one or more replicas, enabling message reliability.

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 is a free service. You only pay 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 fine-grained authorization strategy that defines permissions required to perform operations on specific cloud resources under certain conditions. This mechanism allows for more flexible policy-based authorization for more secure 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 1-10 lists all the system-defined policies supported by DMS for Kafka.

Table 1-10 System-defined 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
DMS ReadOnlyAccess	Read-only permissions for DMS. Users granted these permissions can only view DMS data.	System-defined policy	None

Role/Policy Name	Description	Type	Dependency
DMS VPCAccess	VPC operation permissions to assign to DMS agencies.	System-defined policy	None
DMS KMSAccess	KMS operation permissions to assign to DMS agencies.	System-defined policy	None

 **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 1-11 lists the common operations supported by each DMS for Kafka system policy. Select the policies as required.

Table 1-11 Common operations supported by each system-defined policy of DMS for Kafka

Operation	DMS FullAccess	DMS UserAccess	DMS ReadOnly Access	DMS VPCAccess	DMS KMSAccess
Creating instances	√	×	×	×	×
Modifying instances	√	×	×	×	×
Deleting instances	√	×	×	×	×
Modifying instance specifications	√	×	×	×	×
Restarting instances	√	√	×	×	×
Querying instance information	√	√	√	×	×

Fine-grained Authorization

To use a custom fine-grained policy, log in to the IAM console as an administrator and select the desired fine-grained permissions for DMS. [Table 1-12](#) describes fine-grained permission dependencies of DMS for Kafka.

Table 1-12 Fine-grained permission dependencies of DMS for Kafka

Permission	Description	Dependency
dms:instance:list	Viewing the instance list	None
dms:instance:get	Viewing instance details	None
dms:instance:create	Creating an instance	<ul style="list-style-type: none">vpc:vpcs:getvpc:ports:createvpc:securityGroups:getvpc:ports:getvpc:subnets:getvpc:vpcs:listvpc:publicIps:getvpc:publicIps:listvpc:ports:updatevpc:publicIps:updatevpc:ports:delete
dms:instance:getBackgroundTask	Viewing background task details	None
dms:instance:deleteBackgroundTask	Deleting a background task	None
dms:instance:modifyStatus	Restarting an instance	None
dms:instance:resetAuthInfo	Resetting an instance password	None
dms:instance:modifyAuthInfo	Changing an instance password	None

Permission	Description	Dependency
dms:instance:modify	Modifying an instance	<ul style="list-style-type: none"> • vpc:vpcs:get • vpc:ports:create • vpc:securityGroups:get • vpc:ports:get • vpc:subnets:get • vpc:vpcs:list • vpc:publicIps:get • vpc:publicIps:list • vpc:ports:update • vpc:publicIps:update • vpc:ports:delete
dms:instance:scale	Scaling up an instance	<ul style="list-style-type: none"> • vpc:vpcs:get • vpc:ports:create • vpc:securityGroups:get • vpc:ports:get • vpc:subnets:get • vpc:vpcs:list • vpc:publicIps:get • vpc:publicIps:list • vpc:ports:update • vpc:publicIps:update
dms:instance:delete	Deleting an instance	None

Helpful Links

- [What Is IAM?](#)
- [Creating a User and Granting DMS for Kafka Permissions](#)
- [Permissions Policies and Supported Actions](#)

2 Getting Started

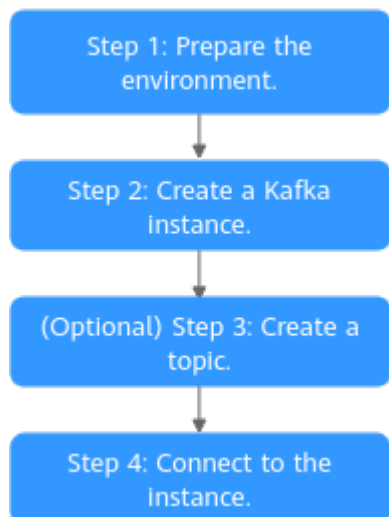
2.1 Getting Started with Kafka to Produce and Consume Messages

This section takes the example of creating and connecting to a Kafka 2.7 instance with SASL enabled to get you quickly started with Distributed Message Service (DMS) for Kafka.

You can also [create a Kafka instance by calling an API](#) and connect to the instance in your service code.

Process of Using DMS for Kafka

Figure 2-1 Procedure for using DMS for Kafka



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.**

You can select the specification and quantity and enable SASL when creating a Kafka instance. Enabling SASL secures data transmission with encryption.

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.**

Before connecting to a Kafka instance with SASL enabled, download the certificate and configure the connection in the client configuration file.

 **NOTE**

For details about Kafka concepts, see [Basic Concepts](#).

Step 1: Prepare the Environment

Step 1 To achieve fine-grained management of your cloud resources, create IAM user groups and users and grant specified permissions to the users. For details, see [Creating a User and Granting DMS for Kafka Permissions](#).

Step 2 Before creating a Kafka instance, ensure that a VPC and a subnet are available.

Configure the VPC and subnet for Kafka instances as required. You can use the current account's existing VPC and subnet, or create new ones. For details about how to create a VPC and a subnet, see [Creating a VPC](#). Note that the VPC must be in the same region as the Kafka instance.

Step 3 Before creating a Kafka instance, ensure that a security group is available.

Configure the security group for Kafka instances as required. You can use the current account's existing security groups, or create new ones. For details about how to create a security group, see [Creating a Security Group](#).

To connect to 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 address	Description
Inbound	TCP	9093	0.0.0.0/0	Access Kafka instance in the same VPC over private network (SSL enabled).

 **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 using the private network within a VPC, you do not need to add the rules described in [Table 2-1](#).

Step 4 Before connecting to a Kafka instance, ensure that you have created 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.

1. Log in to the console, click  in the upper left corner, click **Elastic Cloud Server** under **Computing**, and then create an ECS.
For details about how to create an ECS, see [Creating an ECS](#). If you already have an available ECS, skip this step.
2. Log in to the ECS.
3. Install Java JDK or JRE and configure the environment variables **JAVA_HOME** and **PATH**.

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 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](#).

4. Download an open-source Kafka client.

```
wget https://archive.apache.org/dist/kafka/2.7.2/kafka_2.12-2.7.2.tgz
```
5. Run the following command to decompress the package:

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

----End

Step 2: Create a Kafka Instance

- Step 1** Log in to the Kafka console, then click **Buy Instance** in the upper right corner of the page.
- Step 2** Set **Billing Mode** to **Pay-per-use**.
- Step 3** Select a region closest to your application to reduce latency and accelerate access.
- Step 4** Select a project from the **Project** 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** Specify the instance settings. For details, see [Table 2-2](#).

Table 2-2 Setting an instance

Parameter	Description
Version	Select 2.7 . Fixed once the instance is created. Use the same version as your client.
CPU Architecture	Select x86 .
Broker Flavor	Select kafka.2u4g.cluster .
Brokers	Enter 3
Storage Space per Broker	Select Ultra-high I/O and enter 100 GB. Total storage space = Storage space per broker × Broker quantity. After the instance is created, you cannot change the disk type.
Capacity Threshold Policy	Select Automatically delete .

Step 8 Configure the instance network. For details, see [Table 2-3](#).

Table 2-3 Configuring instance network

Parameter	Description
VPC	Select the created VPC and subnet. You cannot change the VPC and subnet after the instance is created.
Security Group	Select the created security group.

Step 9 Set the instance access mode. For details, see [Table 2-4](#).

Table 2-4 Setting instance access mode

Parameter	Description
Kafka SASL_SSL	SASL_SSL supported This setting is fixed once the instance is created.
SASL PLAIN	SASL PLAIN supported If SASL/PLAIN is enabled, both the SCRAM-SHA-512 and PLAIN mechanisms are supported. You can select either of them to connect Kafka instances on the client.
Username	Username for the client to connect to Kafka instances
Password	Password for the client to connect to Kafka instances

Step 10 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 and broker information of your Kafka clusters.

Step 11 Click **Advanced Settings**. For more information, see [Table 2-5](#).

Table 2-5 Advanced settings

Parameter	Description
Automatic Topic Creation	Do not enable it.
Tags	Skip it.
Description	Skip it.

Step 12 Click **Buy**.

Step 13 Confirm the instance settings.

Step 14 Return to the **DMS for Kafka** 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 is in the **Creation failed** state, delete it. Then create a new one. If the instance creation fails again, contact customer service.

 **NOTE**

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

----End

(Optional) Step 3: Create a Topic

Step 1 On the **DMS for Kafka** page, click a Kafka instance.

Step 2 In the navigation pane, choose **Topics**.

Step 3 Click **Create Topic**.

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

Table 2-6 Topic parameters

Parameter	Description
Topic Name	Enter "topic-01". Cannot be changed once the topic is created.
Partitions	Set it to 3. The more partitions, the higher the consumption concurrency.
Replicas	Set it to 3. Kafka automatically backs up data on each replica. If one broker is faulty, data will still be available. Reliability increases with the number of replicas of a topic. NOTE If an instance node is faulty, an internal service error may be reported when you query messages in a topic with only one replica. Therefore, you are not advised using a topic with only one replica.
Aging Time (h)	Set it to 72. How long messages will be preserved in the topic. Messages older than this period cannot be consumed. They will be deleted, and can no longer be consumed.
Synchronous Replication	Do not enable it.
Synchronous Flushing	Do not enable it.
Message Timestamp	Select CreateTime .
Max. Message Size (bytes)	Retain the default value.

----End

Step 4: Connect to a Kafka Instance to Produce and Consume Messages

Step 1 Obtain the instance connection address.

1. In the navigation pane, click **Basic Information**.
2. In the **Connection** area, view the connection address.

Figure 2-2 Kafka instance addresses (private network) for intra-VPC access

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

Step 2 Prepare the file for production and consumption configuration.

1. Log in to a Linux ECS.

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 [Step 1](#). 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
3. Download `client.truststore.jks` certificate.
On the Kafka console, click the Kafka instance to go to the **Basic Information** page. Click **Download** next to **SSL Certificate** in the **Connection** area. Decompress the package to obtain the client certificate file **client.truststore.jks**.
4. Add the following commands in both the **consumer.properties** and **producer.properties** files (PLAIN is used as an example).

```
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=
```

Description:

 - **username** and **password** are specified when enabling SASL_SSL during instance creation.
 - **ssl.truststore.location** is the path for storing the certificate obtained in [Step 2.3](#).
 - **ssl.truststore.password** is certified by the server, which must be set to **dms@kafka** and cannot be changed.
 - **ssl.endpoint.identification.algorithm** decides whether to verify the certificate domain name. Here, **this parameter must be left blank, which indicates disabling domain name verification**.

Step 3 Produce messages.

Go to the `/bin` directory of the Kafka client file and run the following command:

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

Description:

- `{connection addr}`: the address obtained in [Step 1](#).
- `{topic name}`: the topic name obtained in [Step 4](#).

For example, **192.xxx.xxx.xxx:9093**, **192.xxx.xxx.xxx:9093**, **192.xxx.xxx.xxx:9093** 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  
192.xxx.xxx.xxx:9093,192.xxx.xxx.xxx:9093,192.xxx.xxx.xxx:9093 --topic topic-demo --producer.config ../
```

```
config/producer.properties
>Hello
>DMS
>Kafka!
>^C[root@ecs-kafka bin]#
```

Press **Ctrl+C** to cancel.

Step 4 Consume messages.

Run the following command:

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

Description:

- *{connection addr}*: the address obtained in [Step 1](#).
- *{topic name}*: the topic name obtained in [Step 4](#).
- *{consumer group name}*: the consumer group name set as required. **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.

Sample:

```
[root@ecs-kafka bin]# ./kafka-console-consumer.sh --bootstrap-server
192.xxx.xxx.xxx:9093,192.xxx.xxx.xxx:9093,192.xxx.xxx.xxx:9093 --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]#
```

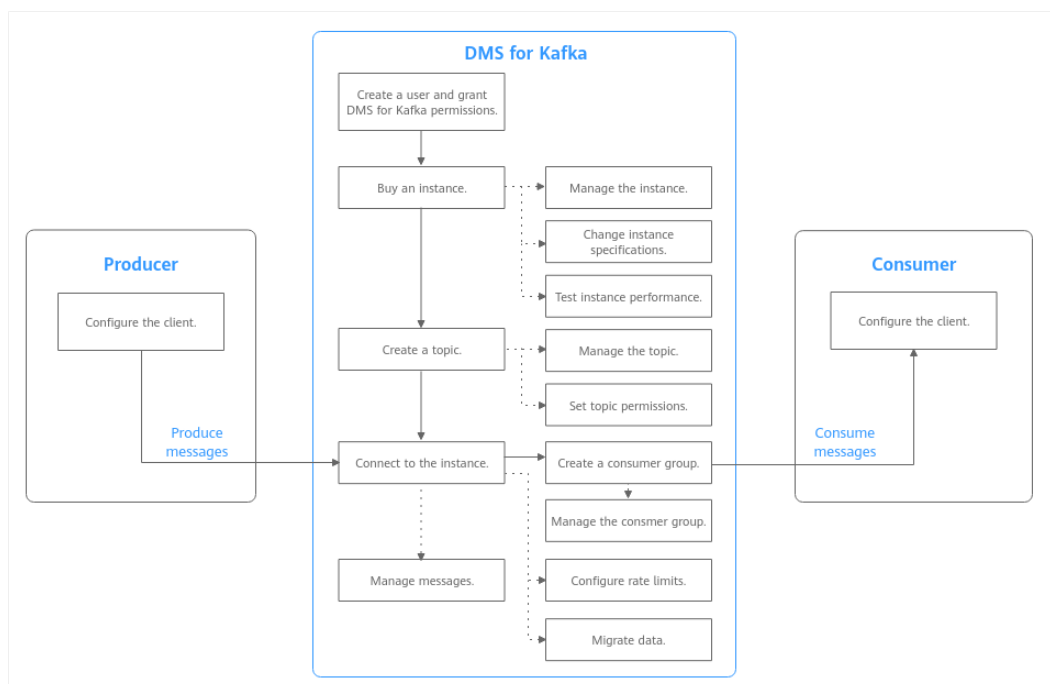
Press **Ctrl+C** to cancel.

----End

3 Process of Using Kafka

The following figure shows the process of using a Kafka instance to produce and consume messages.

Figure 3-1 Process of using Kafka



1. **Creating a User and Granting DMS for Kafka Permissions**
Create IAM users and grant them only the DMS for Kafka permissions required to perform a given task based on their job responsibilities.
2. **Buying a Kafka Instance**
Your Kafka instance will be exclusive and deployed in physical isolation.
3. **Creating a Kafka Topic**
Create a topic for storing messages so that producers can produce messages and consumers can subscribe to messages.
4. **Connecting to an Instance**

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

5. **Managing Messages**

View messages on the console.

4 Permissions Management

4.1 Creating a User and Granting DMS for Kafka Permissions

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

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

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

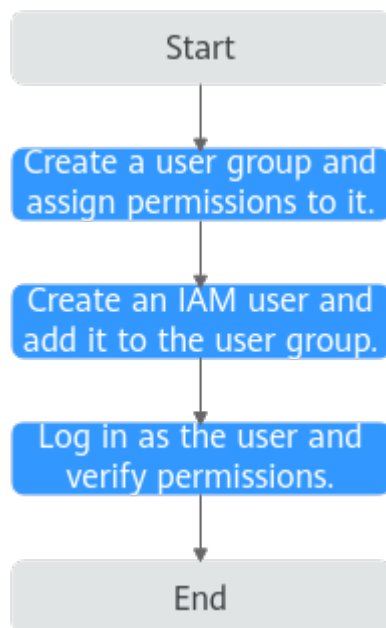
This section describes the procedure for granting permissions (see [Figure 4-1](#)).

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 4-1 Process for granting DMS for Kafka permissions



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

In the authorized region, perform the following operations:

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

Example Custom Policies

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

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

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

For details, see [Creating a Custom Policy](#). The following lists examples of common DMS for Kafka custom policies.

 NOTE

- DMS for Kafka permissions policies are based on DMS. Therefore, when assigning permissions, select DMS permissions policies.
- Due to data caching, a policy involving Object Storage Service (OBS) actions will take effect five minutes after it is attached to a user, user group, or project.
- Example 1: Grant permission to delete and restart instances.

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

- Example 2: Grant permission to deny instance deletion.

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

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

Example policy denying instance deletion:

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

DMS for Kafka Resources

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

Table 4-1 DMS for Kafka resources and their paths

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

DMS for Kafka Request Conditions

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

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

Table 4-2 Predefined condition keys of DMS for Kafka

Condition Key	Operator	Description
<code>dms:publicIP</code>	Bool	Whether public access is enabled
<code>dms:ssl</code>	Bool	Whether SSL is enabled

5 Buying a Kafka Instance

Scenario

Your Kafka instance will be exclusive and deployed in physical isolation. You can customize the computing capabilities and storage space of an instance based on service requirements.

Prerequisites


Before creating a Kafka instance, prepare the resources listed in [Table 5-1](#).

Table 5-1 Kafka resources

Resource	Requirement	Operations
VPC and subnet	Configure the VPC and subnet for Kafka instances as required. You can use the current account's existing VPC and subnet, or create new ones. Note: VPCs must be created in the same region as the Kafka instance.	For details about how to create a VPC and subnet, see the <i>Virtual Private Cloud User Guide</i> .
Security group	Different Kafka instances can use the same or different security groups.	For details about how to create a security group and configure security group rules, see the <i>Virtual Private Cloud User Guide</i> .


Buying a Kafka Instance

Step 1 Log in to the console.

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

 NOTE

Select the region your application is in.

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

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

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:** Retain the default value.

3. **Broker Flavor:** Select broker specifications that best fit your 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 upper limit of partitions, topic creation fails.

4. **Storage Space per Broker:** Disk type and total disk space for storing the instance data. **The disk type cannot be changed once the instance is created.**

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

Disks are formatted when an instance is created. As a result, the actual available disk space is 93% to 95% of the total disk space.

5. **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 Creating a Kafka instance

Version: 2.7 2.3.0 1.1.0
Software version of the instance.
Fixed once the instance is created. Use the same version as your client.

CPU Architecture: x86

Flavor Name	TPS Limit per Broker	Maximum Partitions per Br...	Recommended Consumer ...	Traffic per Broker (MB/s)
<input checked="" type="radio"/> kafka.2u4g.cluster	30,000	250	20	100
<input type="radio"/> kafka.4u8g.cluster	100,000	500	100	200
<input type="radio"/> kafka.8u16g.cluster	150,000	1,000	150	375
<input type="radio"/> kafka.12u24g.cluster	200,000	1,500	200	625
<input type="radio"/> kafka.16u32g.cluster	250,000	2,000	200	750

Currently Selected: kafka.2u4g.cluster | TPS Limit per Broker 30,000 | Maximum Partitions per Broker 250 | Recommended Consumer Groups per Broker 20 | Traffic per Broker (MB/s) 100

Brokers:
Cannot be decreased once the instance is created.

Total traffic: 300 MB/s (Total traffic = Traffic per broker x Broker quantity)
 Service traffic:
 * Recommended read traffic: 50 MB/s
 * Recommended write traffic: 50 MB/s
 Tip: Reserve 30% of traffic to accommodate traffic fluctuations.

Storage space per broker: GB ⓘ
 Total storage space: 300 GB
After the instance is created, you cannot change the disk type or reduce the storage space. Select a disk type for your required service I/O. [Learn more](#)

Total disk bandwidth: 510 MB/s = Disk bandwidth per broker x Broker quantity
 Disk bandwidth per broker: 170 MB/s

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 **Kafka SASL_SSL**.

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

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

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

What are SCRAM-SHA-512 and PLAIN mechanisms?

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

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

Step 10 Configure the username and password for logging in to Kafka Manager. **The Kafka Manager username cannot be changed once the 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 11 Click **Advanced Settings** to configure more parameters.

1. Configure **Automatic Topic Creation**.

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

If this option is enabled, a topic will be automatically created when a message is produced in or consumed from a topic that does not exist. By default, the topic has the following parameters:

- **Partitions:** 3
- **Replicas:** 3
- **Aging Time:** 72
- **Synchronous Replication** and **Synchronous Flushing** disabled
- **Message Timestamp:** CreateTime
- **Max.Message Size (bytes):** 10,485,760

After you change the value of the **log.retention.hours**, **default.replication.factor**, or **num.partitions** parameter, the value will be used in later topics that are automatically created.

For example, assume that **num.partitions** is changed to 5, an automatically created topic has the following parameters:

- **Partitions:** 5
- **Replicas:** 3
- **Aging Time:** 72
- **Synchronous Replication** and **Synchronous Flushing** disabled
- **Message Timestamp:** CreateTime
- **Max.Message Size (bytes):** 10,485,760

2. Specify **Tags**.

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

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

Up to 20 tags can be added to each Kafka instance. For details about the requirements on tags, see [Configuring Kafka Instance Tags](#).

3. Enter a description of the instance.

Step 12 Click **Buy**.

Step 13 Confirm the instance information, and click **Submit**.

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

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

- If the instance is created successfully, its status changes to **Running**.
- If the instance is in the **Creation failed** state, delete it by referring to [Deleting Kafka Instances](#). Then create a new one. If the instance creation fails again, contact customer service.

 **NOTE**

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

----End

6 Configuring Topics

6.1 Creating a Kafka 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.

Automatic topic creation: A topic will be automatically created when a message is produced in or consumed from a topic that does not exist. By default, the topic has the following parameters:

- **Partitions:** 3
- **Replicas:** 3
- **Aging Time:** 72
- **Synchronous Replication** and **Synchronous Flushing** disabled
- **Message Timestamp:** CreateTime
- **Max.Message Size (bytes):** 10,485,760

After you change the value of the **log.retention.hours**, **default.replication.factor**, or **num.partitions** parameter, the value will be used in later topics that are automatically created.

For example, assume that **num.partitions** is changed to 5, an automatically created topic has the following parameters:

- **Partitions:** 5
- **Replicas:** 3
- **Aging Time:** 72
- **Synchronous Replication** and **Synchronous Flushing** disabled
- **Message Timestamp:** CreateTime
- **Max.Message Size (bytes):** 10,485,760

Methods that can be used to manually create a topic:

- [Method 1: Creating a Topic on the Console](#)


- [Method 2: Creating a Topic on Kafka Manager](#)
- [Method 3: Creating a Topic by Using Kafka CLI](#)

Constraints

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

Method 1: Creating a Topic on the Console

Step 1 Log in to the console.

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

 **NOTE**

Select the region where your Kafka instance is located.

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

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

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

Figure 6-1 Creating a topic

Create Topic

Topic Name

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

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

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

Synchronous Replication ?

Synchronous Flushing ?

Message Timestamp ?

Max.Message Size (bytes) ?

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

Table 6-1 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.
Partitions	A larger number of partitions for a topic indicates more messages retrieved concurrently. If this parameter is set to 1 , messages will be retrieved in the FIFO order. Value range: 1 to 200 Default value: 3

Parameter	Description
Replicas	<p>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.</p> <p>Value range: 1 to number of brokers</p> <p>NOTE If an instance node is faulty, an internal service error may be reported when you query messages in a topic with only one replica. Therefore, you are not advised using a topic with only one replica.</p>
Aging Time (h)	<p>The period that messages are retained for. Consumers must retrieve messages before this period ends. Otherwise, the messages will be deleted and can no longer be consumed.</p> <p>Value range: 1 to 720</p> <p>Default value: 72</p>
Synchronous Replication	<p>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.</p> <p>After enabling synchronous replication, set acks to all or -1 on the client. Otherwise, this function will not take effect.</p> <p>If there is only one replica, synchronous replication cannot be enabled.</p>
Synchronous Flushing	<ul style="list-style-type: none"> • Enabled: A message is immediately flushed to disk once it is created, resulting in higher reliability. • Disabled: A message is stored in the memory instead of being immediately flushed to disk once created.
Message Timestamp	<p>Timestamp type of a message. Options:</p> <ul style="list-style-type: none"> • CreateTime: time when the producer created the message. • LogAppendTime: time when the broker appended the message to the log.
Max. Message Size	<p>Maximum batch processing size allowed by Kafka. If message compression is enabled, this parameter indicates the size after compression.</p> <p>If this is increased and there are consumers older than 0.10.2, the consumers' fetch size must also be increased so that they can fetch record batches this large.</p> <p>Value range: 0 to 10,485,760</p>

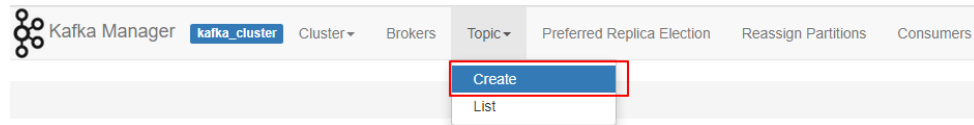
Step 7 Click **OK**.

----End

Method 2: Creating a Topic on Kafka Manager

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

Figure 6-2 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: Creating a Topic by Using Kafka CLI

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

NOTICE

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

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

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

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

- a. (Optional) If the username and password, and 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. Add the username and password, and the SSL certificate configuration 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
```

6.2 Configuring Kafka Topic Permissions

DMS for Kafka supports access control list (ACL) for topics. You can differentiate user permissions by granting users different permissions in a topic.

This section describes how to grant topic permissions to a user with SASL_SSL enabled. For details about how to create a user, see [Configuring Kafka Users](#).

Constraints


- If no user is granted any permission for a topic and **allow.everyone.if.no.acl.found** is set to **true**, all users can subscribe to or publish messages to the topic.
- If **allow.everyone.if.no.acl.found** is set to **false**, only the authorized users can subscribe to or publish messages to the topic. The value of **allow.everyone.if.no.acl.found** can be [modified](#).
- If one or more 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 user has been created. For details, see [Configuring Kafka Users](#).


Configuring Topic Permissions

Step 1 Log in to the console.

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

 **NOTE**

Select the region where your Kafka instance is located.

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

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

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

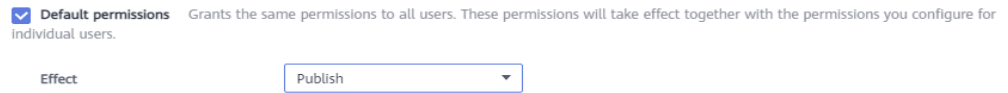
Step 6 In the row that contains the topic for which you want to configure user permissions, click **Grant User Permission**.

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

Step 7 Grant topic permissions to users.

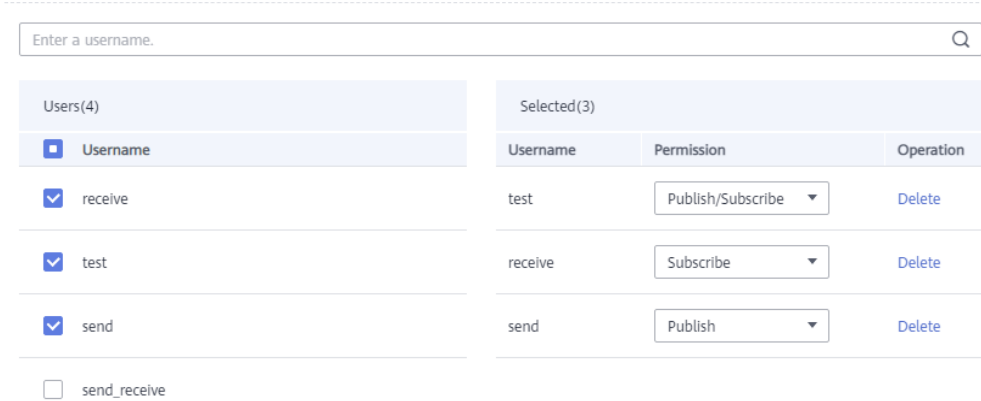
- To grant the same permissions to all users, select **Default permissions** and then select permissions. As shown in the following figure, all users have the permission to publish messages to this topic.

Figure 6-3 Granting the same rights to all users



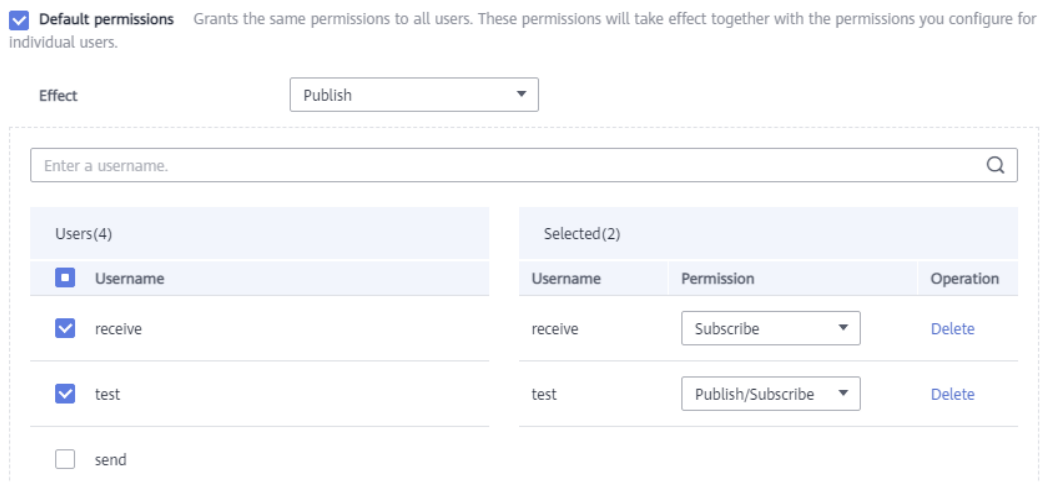
- To grant different permissions to different users, do not select **Default permissions**. In the **Users** area of the **Grant User Permission** dialog box, select target users. In the **Selected** area, configure permissions (**Subscribe**, **Publish**, or **Publish/Subscribe**) for the users. As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

Figure 6-4 Granting permissions to individual users



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

Figure 6-5 Granting topic permissions to users



Step 8 Click **OK**.


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

Figure 6-6 Viewing authorized users and their permissions


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

Default permissions: Publish	
Username	Permission
receive	Subscribe
test	Publish/Subscribe

----End

(Optional) Deleting Topic Permissions

Step 1 Log in to the console.

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

 **NOTE**

Select the region where your Kafka instance is located.

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

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

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

Step 6 In the row 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 user whose permissions are to be removed, click **Delete**, and click **OK**.

----End


6.3 Managing Topics

6.3.1 Viewing Kafka Topic Details

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


Viewing Topic Details

Step 1 Log in to the console.

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

 **NOTE**

Select the region where your Kafka instance is located.

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

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

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

Step 6 Click a topic to view its details.

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

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

 **NOTE**

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

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

Click the name of a consumer group name.

 **NOTE**

If an instance contains more than 10,000 consumer groups, the subscription relationships of topics cannot be queried.

Figure 6-7 Subscriptions

Consumer Group Name	Status	Coordinator(ID)	Accumulated Messages
group	STABLE	1	0

- Partitions: partition No., minimum offset, maximum offset, number of messages, and message update time

Figure 6-8 Partitions

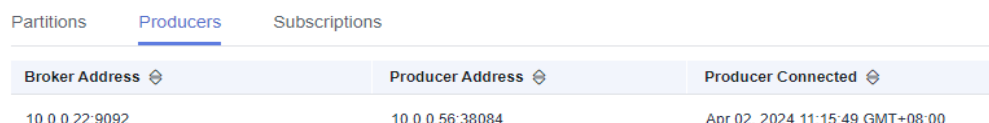
Partition	Minimum Offset	Maximum Offset	Messages	Updated
0	0	5	5	Apr 02, 2024 11:11:30 GMT+08:00
1	0	3	3	Apr 02, 2024 11:11:37 GMT+08:00
2	0	8	8	Apr 02, 2024 11:10:27 GMT+08:00

- Producers: broker address, producer address, and producer connected time

 NOTE

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

Figure 6-9 Producers



Broker Address	Producer Address	Producer Connected
10.0.0.22:9092	10.0.0.56:38084	Apr 02, 2024 11:15:49 GMT+08:00

----End

6.3.2 Modifying Kafka Topic Configurations

This section describes how to modify following configurations of a Kafka topic on the console.

Table 6-2 Kafka topic configuration parameters


Parameter	Description
Synchronous Replication	A message is returned to the client only after the message creation request has been received and the message has been acknowledged by all replicas.
Synchronous Flushing	<ul style="list-style-type: none"> • Enabled: A message is immediately flushed to disk once it is created, bringing higher reliability. • Disabled: A message is stored in the memory instead of being immediately flushed to disk once created.
Message Timestamp	Timestamp type of a message. Options: <ul style="list-style-type: none"> • CreateTime: time when the producer created the message. • LogAppendTime: time when the broker appended the message to the log.
Max. Message Size	Maximum size of messages to be processed in batches. If message compression is enabled, this parameter indicates the size after compression. If this value is increased and the consumer version is earlier than 0.10.2, the consumers' fetch size must also be increased so that they can obtain the latest value.

 NOTE

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


Modifying Topic Configurations

Step 1 Log in to the console.

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

 **NOTE**

Select the region where your Kafka instance is located.

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

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

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

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

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

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

 **NOTE**

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

----End

6.3.3 Changing Kafka Partition Quantity

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

 **NOTE**


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

Methods for changing the partition quantity:

- [Method 1: By using Kafka console](#)
- [Method 2: By using Kafka Manager](#)
- [Method 3: By using Kafka CLI](#)

Method 1: By using Kafka console

Step 1 Log in to the console.

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

 **NOTE**

Select the region where your Kafka instance is located.

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

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

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

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

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

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

 **NOTE**

- The number of partitions can only be increased.
- To ensure performance, the Kafka console allows a maximum of 200 partitions for each topic.
- The total number of partitions of all topics cannot exceed the maximum number of partitions allowed by the instance.

----End

Method 2: By using Kafka Manager

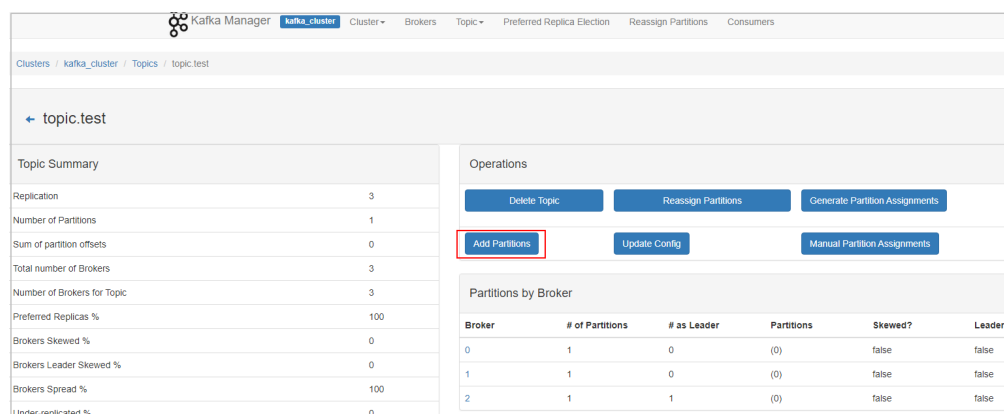
Step 1 [Log in to Kafka Manager](#).

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 6-10 Topic details page

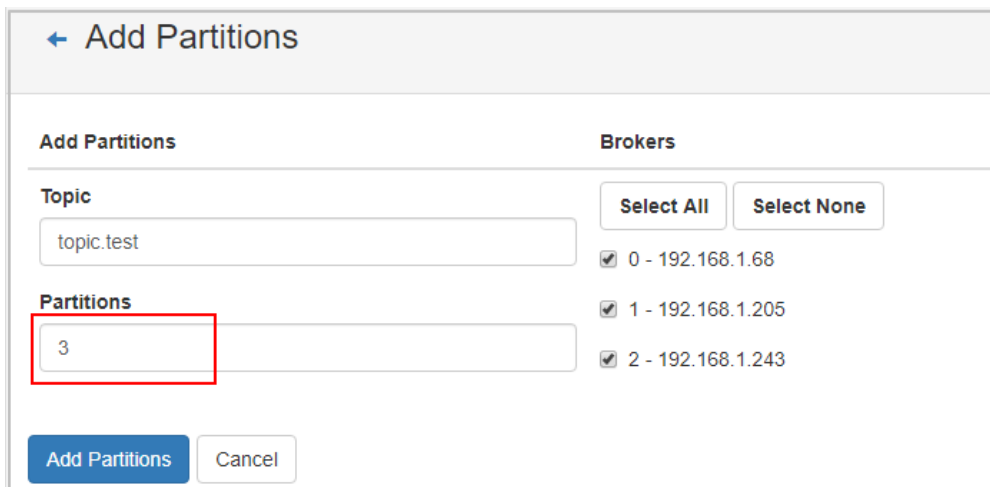


Topic Summary		Operations	
Replication	3	Delete Topic Reassign Partitions Generate Partition Assignments	
Number of Partitions	1	Add Partitions Update Config Manual Partition Assignments	
Sum of partition offsets	0		
Total number of Brokers	3		
Number of Brokers for Topic	3		
Preferred Replicas %	100		
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

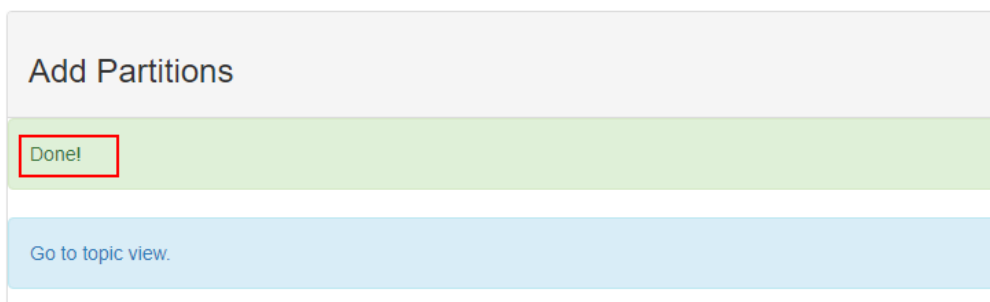
Step 5 Enter the number of partitions and click **Add Partitions**.

Figure 6-11 Adding partitions



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

Figure 6-12 Partitions added



NOTE

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

----End

Method 3: By Using Kafka CLI

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

NOTICE

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

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

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

- If SASL has been enabled for the Kafka instance, perform the following steps to change the partition quantity:
 - a. (Optional) If the username and password, and 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. Add the username and password, and the SSL certificate configuration 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
```

6.3.4 Exporting the Kafka Topic List


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

Prerequisites

[A topic](#) has been created.


Exporting the Topic List

Step 1 Log in to the console.

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


 **NOTE**

Select the region where your Kafka instance is located.

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

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

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

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

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

----End

6.3.5 Reassigning Kafka Partitions

Scenario

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

Partition reassignment is required in the following scenarios:

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

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

Operation Impact


- Partition reassignment on topics with a large amount of data consumes a large amount of network and storage bandwidth. As a result, service requests may time out or the latency may increase. Therefore, you are advised to perform reassignment during off-peak hours. Compare the current instance load based on the instance specifications to decide whether the remaining instance capacity can support partition reassignment. Do not reassign partitions when there is insufficient bandwidth or when the CPU usage is greater than 90%.
- A throttle refers to the upper limit of the bandwidth for replication of a topic, to ensure that other topics on the instance are not affected. Note that throttles apply to replication triggered by both normal message production and partition reassignment. If the throttle is too small, normal message production may be affected, and partition reassignment may never complete.
- You cannot delete topics whose reassignment tasks have started. Otherwise, the tasks will never complete.
- You cannot modify the partition quantity of topics whose reassignment tasks have started.
- Reassignment tasks cannot be manually stopped. Please wait until they complete.
- 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 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:

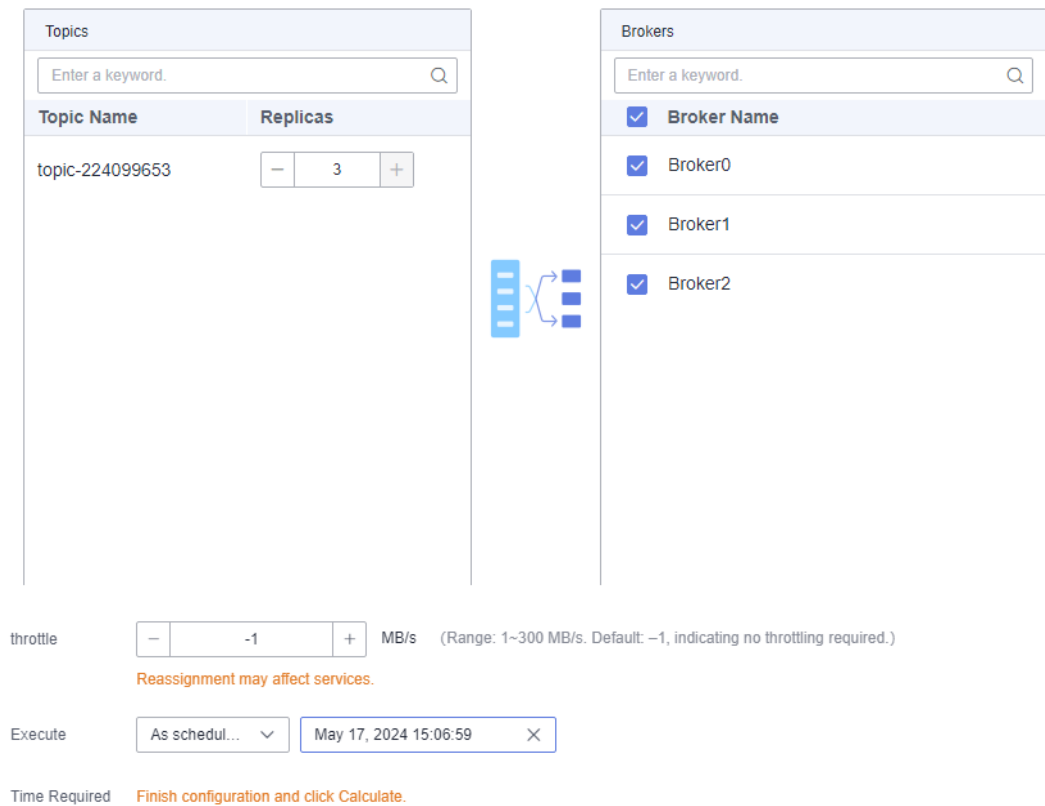
- Select one or more topics and choose **Reassign > Auto** above the topic list.
- In the row that contains the desired topic, choose **More > Reassign > Auto**.

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](#).
- For **Execute**, specify when to execute the reassignment. **Now** means to execute it immediately. **As scheduled** means to execute it at the scheduled time.

Figure 6-13 Setting automatic reassignment parameters

Auto



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

Step 9 Click **OK**.

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

Table 6-3 Checking the reassignment result

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

Task Type	Reassignment Result
Scheduled tasks	<ol style="list-style-type: none"> The Background Tasks > Scheduled tasks page is displayed. This page only shows whether scheduled tasks start to execute instead of whether they are successful. <ul style="list-style-type: none"> When the task status is Pending, reassignment has not been executed. When the task status is Successful, reassignment has started. Click Background tasks tab page. When the task status is Successful, reassignment has completed.


 **NOTE**

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

----End

Manual Reassignment

Step 1 Log in to the 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 a topic and choose **Reassign > Manual** above the topic list. Manual reassignment does not support batch operations.
- In the row that contains the desired topic, choose **More > Reassign > Manual**.

Step 7 Set manual reassignment parameters.

- In the upper right corner of the **Manual** dialog box, click **Delete Replica** or **Add Replica** to reduce or increase the number of replicas for each partition of the topic.

- Under the name of the replica to be reassigned, click the broker name or ▼ and select the target broker to migrate the replica to. Assign replicas of the same partition to different brokers.
- Specify **throttle**. The default value is **-1**, indicating that there is no throttle (recommended if the instance load is light). If a throttle is required, you are advised to set it to a value greater than or equal to the total production bandwidth of the to-be-reassigned topic multiplied by the maximum number of replicas of the to-be-reassigned topic. For details, see [Calculating a Throttle](#).
- For **Execute**, specify when to execute the reassignment. **Now** means to execute it immediately. **As scheduled** means to execute it at the scheduled time.

Figure 6-14 Setting manual reassignment parameters

Manual

ℹ Replica 1 is Partition Leader

Partition 0

Replica 1 Broker 2 ▼	Replica 2 Broker 1 ▼	Replica 3 Broker 0 ▼
-------------------------	-------------------------	-------------------------

Partition 1

Replica 1 Broker 0 ▼	Replica 2 Broker 2 ▼	Replica 3 Broker 1 ▼
-------------------------	-------------------------	-------------------------

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

Reassignment may affect services.

Execute

Time Required Finish configuration and click Calculate.

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

Step 9 Click **OK**.

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

Table 6-4 Checking the reassignment result

Task Type	Reassignment Result
Background tasks	In the upper left corner of the topic list, click View details and the Background Tasks > Background tasks page is displayed. The reassignment task is complete when it is in the Successful state.
Scheduled tasks	<ol style="list-style-type: none"> The Background Tasks > Scheduled tasks page is displayed. This page only shows whether scheduled tasks start to execute instead of whether they are successful. <ul style="list-style-type: none"> When the task status is Pending, reassignment has not been executed. When the task status is Successful, reassignment has started. Click Background tasks tab page. When the task status is Successful, reassignment has completed.

NOTE

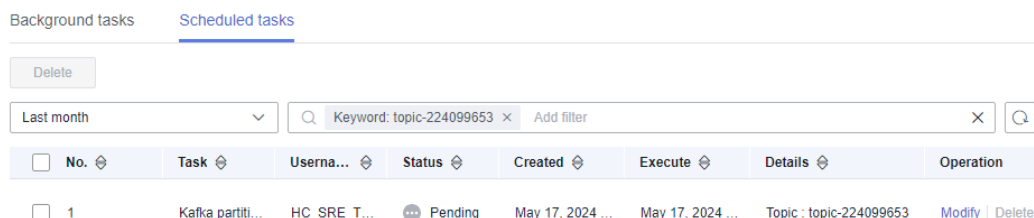
- You cannot delete topics whose reassignment tasks have started. Otherwise, the tasks will never complete.
- You cannot modify the partition quantity of topics whose reassignment tasks have started.
- Reassignment tasks cannot be manually stopped. Please wait until they complete.
- If a scheduled partition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

----End

Re-scheduling Partition Reassignment

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

Figure 6-15 Querying a scheduled reassignment task

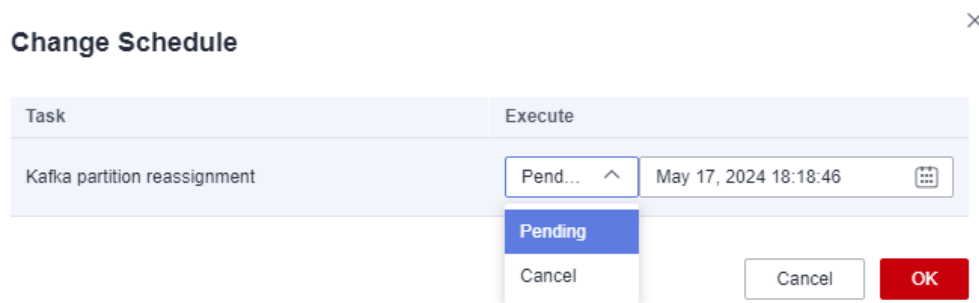


Step 2 In the row that contains the desired task, click **Modify**.

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

- To change the schedule, select a time and click **OK**.
- To cancel the task, select **Cancel** (as shown in [Figure 6-16](#)) and click **OK**.

Figure 6-16 Canceling a scheduled reassignment task



----End

Calculating a Throttle

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

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

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

Scenario 1: Topic 1 has two partitions and two replicas, and Topic 2 has one partition and one replica. All leader replicas are on the same broker. One replica needs to be added for Topic 1 and Topic 2 respectively.

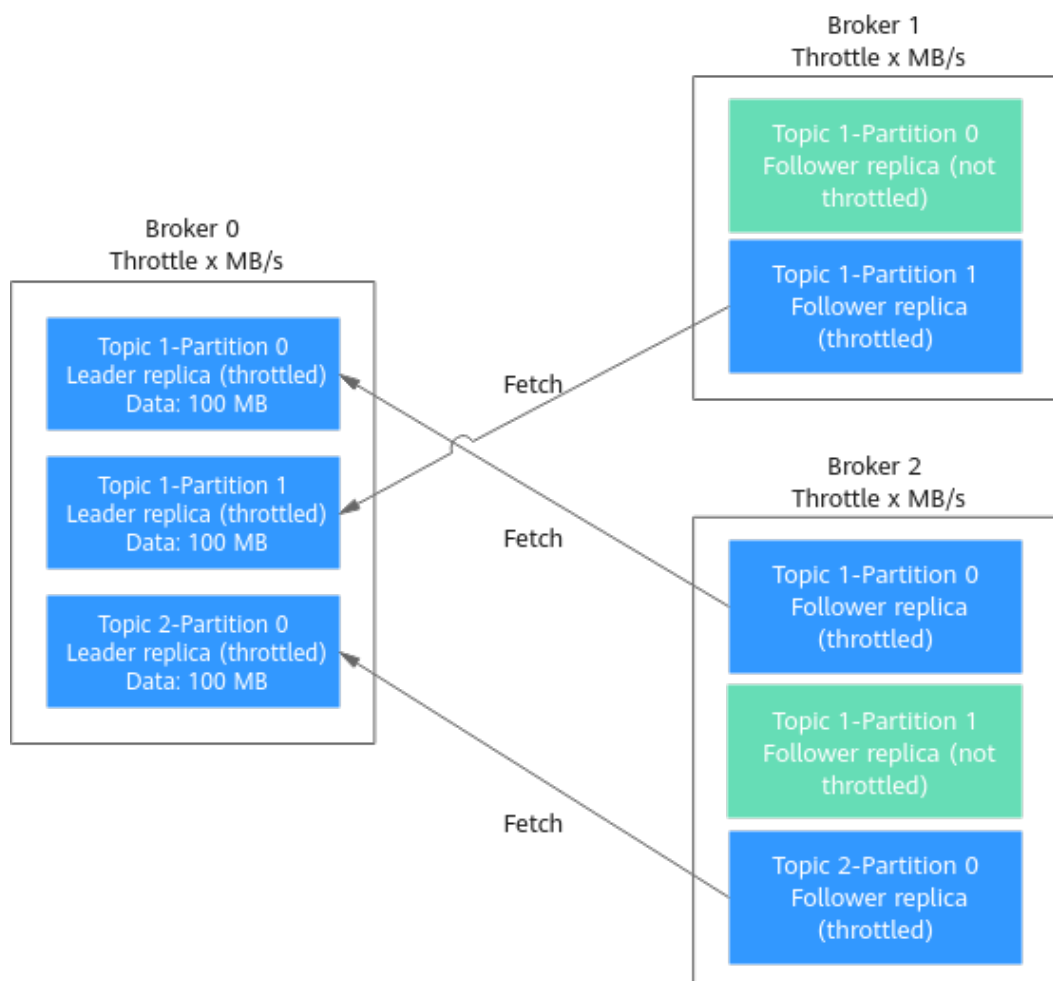
Table 6-5 Replica distribution before reassignment

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 6-6 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 6-17 Reassignment scenario 1



As shown in [Figure 6-17](#), 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 = $(100\text{ MB} + 100\text{ MB} + 100\text{ MB})/200\text{s} = 1.5\text{ MB/s}$
- Bandwidth required by Broker 1 to complete partition reassignment within 200s = $100\text{ MB}/200\text{s} = 0.5\text{ MB/s}$

- Bandwidth required by Broker 2 to complete partition reassignment within 200s = $(100 \text{ MB} + 100 \text{ MB})/200\text{s} = 1 \text{ MB/s}$

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.

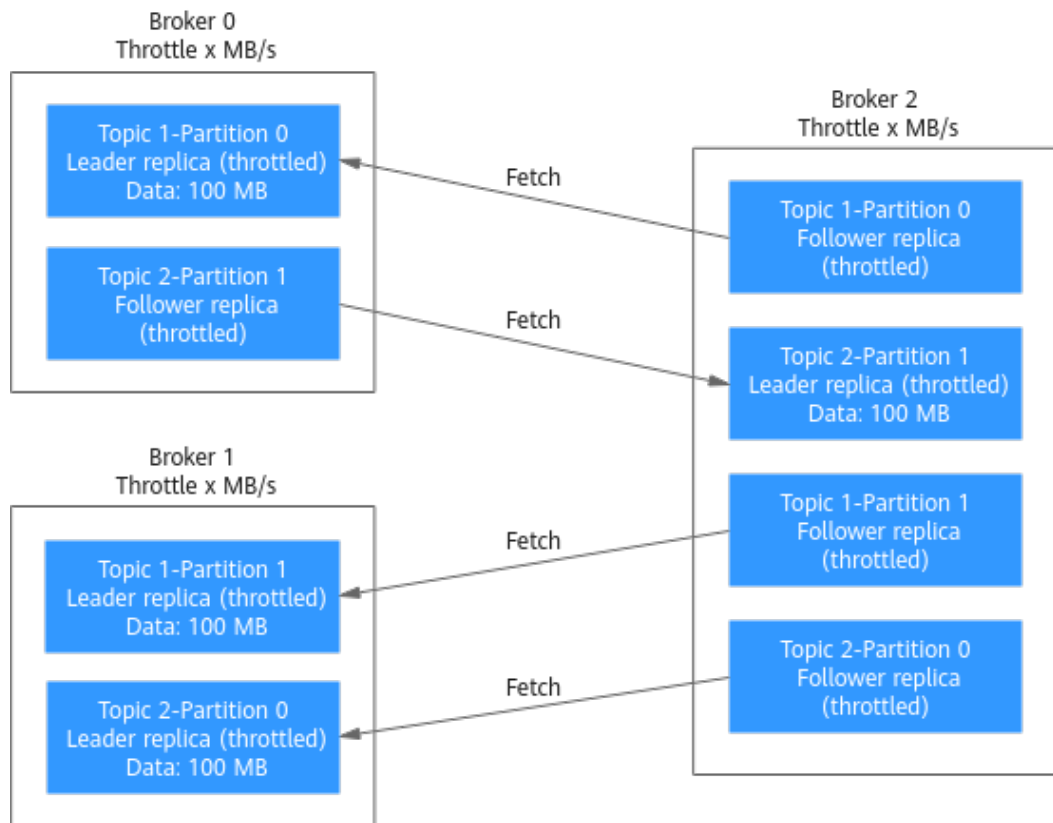
Table 6-7 Replica distribution before reassignment

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 6-8 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 6-18 Reassignment scenario 2



As shown in **Figure 6-18**, 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 = $100 \text{ MB}/200\text{s} = 0.5 \text{ MB/s}$
- Bandwidth required by Broker 0 (follower) to complete partition reassignment within 200s = $100 \text{ MB}/200\text{s} = 0.5 \text{ MB/s}$
- Bandwidth required by Broker 1 to complete partition reassignment within 200s = $(100 \text{ MB} + 100 \text{ MB})/200\text{s} = 1 \text{ MB/s}$
- Bandwidth required by Broker 2 (leader) to complete partition reassignment within 200s = $100 \text{ MB}/200\text{s} = 0.5 \text{ MB/s}$
- Bandwidth required by Broker 2 (follower) to complete partition reassignment within 200s = $(100 \text{ MB} + 100 \text{ MB} + 100 \text{ MB})/200\text{s} = 1.5 \text{ MB/s}$

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.

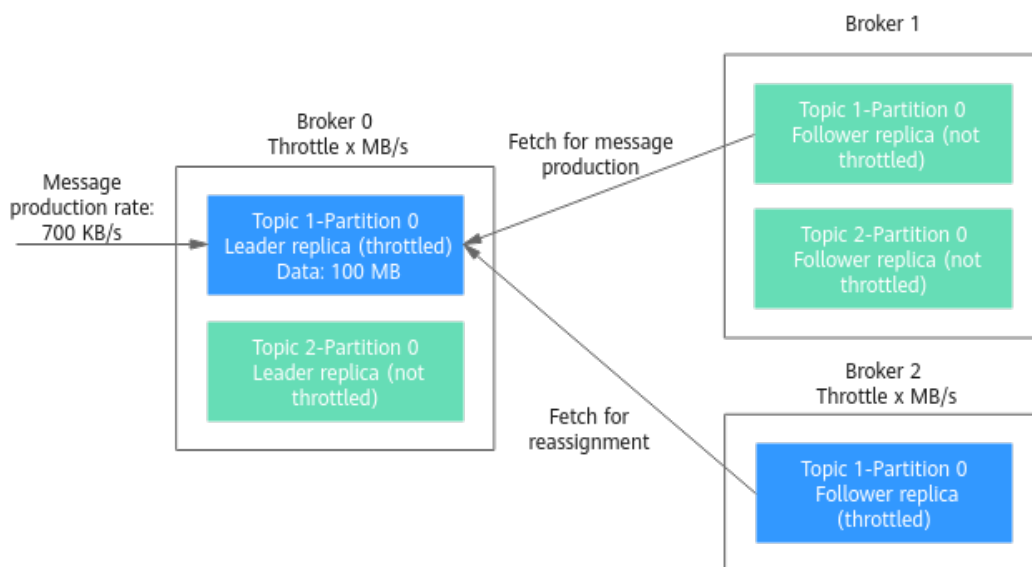
Table 6-9 Replica distribution before reassignment

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 6-10 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 6-19 Reassignment scenario 3



As shown in [Figure 6-19](#), 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 = $(100 \text{ MB} + 700 \text{ KB/s} \times 200\text{s}) / 200\text{s} + 700 \text{ KB/s} = 1.9 \text{ MB/s}$
- Bandwidth required by Broker 2 to complete partition reassignment within 200s = $100 \text{ MB} / 200\text{s} = 0.5 \text{ MB/s}$

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.

6.3.6 Configuring Automatic Topic Creation

Automatic topic creation: A topic will be automatically created when a message is produced in or consumed from a topic that does not exist. By default, the topic has the following parameters:

- **Partitions:** 3
- **Replicas:** 3
- **Aging Time:** 72
- **Synchronous Replication** and **Synchronous Flushing** disabled
- **Message Timestamp:** `CreateTime`
- **Max.Message Size (bytes):** 10,485,760


After you change the value of the `log.retention.hours`, `default.replication.factor`, or `num.partitions` parameter, the value will be used in later topics that are automatically created.

For example, assume that `num.partitions` is changed to 5, an automatically created topic has the following parameters:

- **Partitions:** 5
- **Replicas:** 3
- **Aging Time:** 72
- **Synchronous Replication** and **Synchronous Flushing** disabled
- **Message Timestamp:** `CreateTime`
- **Max.Message Size (bytes):** 10,485,760


Configuring Automatic Topic Creation

Step 1 Log in to the console.



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

 **NOTE**

Select the region where your Kafka instance is located.

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

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

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

Enabling or disabling automatic topic creation may cause instance restarts.

Step 6 Click **OK**.

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

----End

6.3.7 Deleting a Kafka Topic

Delete a topic using either of the following methods:

- [Deleting a Kafka Topic \(Console\)](#)
- [Deleting a Kafka Topic on the Client](#)

Prerequisites


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

Constraints

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


Deleting a Kafka Topic (Console)

Step 1 Log in to the console.

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

 **NOTE**

Select the region where your Kafka instance is located.

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

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

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

Step 6 Delete topics using either of the following methods:

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

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

----End

Deleting a Kafka Topic on the Client

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

NOTICE

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

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

```
./kafka-topics.sh --bootstrap-server {broker_ip}:{port} --delete --topic {topic_name}
```
- If SASL has been enabled for the Kafka instance, perform the following steps to delete a topic:
 - a. (Optional) If the SSL certificate configuration has been set, skip this step. Otherwise, perform the following operations:

Create the **ssl-user-config.properties** file in the `/config` directory of the Kafka client and add the SSL certificate configurations by referring to [Step 3](#).
 - b. Run the following command in the `/{{directory where the CLI is located}}/kafka_{version}/bin/` directory to delete a topic:

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

7 Connecting to an Instance

7.1 Configuring Kafka Network Connections

7.1.1 Kafka Network Connection Conditions

Notes before using a private network:

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

[Table 7-1](#) lists how a client can connect to a Kafka instance.

Table 7-1 Connection modes

Mode	How To Do	Reference
Private access	A client and a Kafka instance are interconnected when they are deployed in the same VPC.	-
	When a client and a Kafka instance are deployed in different VPCs of the same region, interconnect two VPCs using a VPC peering connection.	"VPC Peering Connection" in <i>Virtual Private Cloud User Guide</i>

Before connecting a client to a Kafka instance, allow accesses for the following security groups.

 **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 7-2](#).

Table 7-2 Security group rules

Direction	Protocol	Port	Source	Description
Inbound	TCP	9092	0.0.0.0/0	<ul style="list-style-type: none">• Access a Kafka instance within a VPC (without SSL encryption).• Access a Kafka instance across VPCs using a peering connection (without SSL encryption).
Inbound	TCP	9093	0.0.0.0/0	<ul style="list-style-type: none">• Access a Kafka instance within a VPC (with SSL encryption).• Access a Kafka instance across VPCs using a peering connection (with SSL encryption).

7.2 Configuring Kafka Access Control

7.2.1 Configuring Kafka Users

DMS for Kafka supports access control list (ACL) for topics. You can differentiate user permissions by granting users different permissions in a topic.

This section describes how to create users, reset the password, and delete users with SASL_SSL enabled. For details about how to grant user permissions, see [Configuring Kafka Topic Permissions](#).

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

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


- New user: A user set in instance creation. If you forget your password, reset it by referring to [Resetting the User Password \(for New Users\)](#).
- Non-new users: Users created on the **Users** page. If you forget your password, reset it by referring to [Resetting the User Password \(for Non-new Users\)](#).

Prerequisites

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


Creating a User

Step 1 Log in to the console.

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

 **NOTE**

Select the region where your Kafka instance is located.

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

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

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


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

After the user is created, grant permissions to the user by referring to [Configuring Kafka Topic Permissions](#).

----End

Resetting the User Password (for New Users)

Step 1 Log in to the console.

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

 **NOTE**

Select the region where your Kafka instance is located.

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

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

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

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

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


 **NOTE**

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

----End


Resetting the User Password (for Non-new Users)

Step 1 Log in to the console.

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

 **NOTE**

Select the region where your Kafka instance is located.

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

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

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

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

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


 **NOTE**

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

----End


Deleting a User

Step 1 Log in to the console.

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

 **NOTE**

Select the region where your Kafka instance is located.

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

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

Step 5 Delete a user in either of the following ways:

- On the **Users** page, click **Delete** in the row containing the desired user.
- On the **Users** page, select one or more users and click **Delete** above the list.

 **NOTE**

The user set in Kafka instance creation cannot be deleted.

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

----End

7.3 Configuring the Kafka Client

7.3.1 Setting Parameters for Kafka Clients

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

Table 7-3 Producer parameters

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

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

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

Table 7-4 Consumer parameters

Parameter	Default Value	Recommended Value	Description
auto.commit.enable	TRUE	FALSE	<p>If this parameter is set to TRUE, the offset of messages already fetched by the consumer will be periodically committed to ZooKeeper. This committed offset will be used when the process fails as the position from which the new consumer will begin.</p> <p>Constraints: If this parameter is set to FALSE, to avoid message loss, an offset must be committed to ZooKeeper after the messages are successfully consumed.</p>
auto.offset.reset	latest	earliest	<p>Indicates what to do when there is no initial offset in ZooKeeper or if the current offset has been deleted. Options:</p> <ul style="list-style-type: none"> • earliest: Automatically reset to the smallest offset. • latest: The offset is automatically reset to the largest offset. • none: The system throws an exception to the consumer if no offset is available. • anything else: The system throws an exception to the consumer. <p>NOTE If this parameter is set to latest, the producer may start to send messages to new partitions (if any) before the consumer resets to the initial offset. As a result, some messages will be lost.</p>
connections.max.idle.ms	600000	30000	<p>Timeout interval (in ms) for an idle connection. The server closes the idle connection after this period of time ends. Setting this parameter to 30000 can reduce the server response failures when the network condition is poor.</p>

7.3.2 Suggestions on Using the Kafka Client

Consumers

1. Ensure that the owner thread does not exit abnormally. Otherwise, the client may fail to initiate consumption requests and the consumption will be blocked.
2. Commit messages only after they have been processed. Otherwise, the messages may fail to be processed and cannot be polled again.

3. Generally, do not commit every message. Otherwise, there will be many **OFFSET_COMMIT** requests, causing high CPU usage. For example, if a consumption request pulls 1000 messages and commits every one of them, TPS of the commit requests is 1000 times that of consumption. The smaller the message size, the larger the ratio. You can commit a specific number of messages in batches or enable **enable.auto.commit**. However, if the client is faulty, some cached consumption offset may be lost, resulting in repeated consumption. Therefore, you are advised to commit messages in batches based on service requirements.
4. A consumer cannot frequently join or leave a group. Otherwise, the consumer will frequently perform rebalancing, which blocks consumption.
5. The number of consumers cannot be greater than the number of partitions in the topic. Otherwise, some consumers may fail to poll for messages.
6. Ensure that the consumer polls at regular intervals to keep sending heartbeats to the server. If the consumer stops sending heartbeats for long enough, the consumer session will time out and the consumer will be considered to have stopped. This will also block consumption.
7. Ensure that there is a limitation on the size of messages buffered locally to avoid an out-of-memory (OOM) situation.
8. Set the timeout for the consumer session to 30 seconds:
`session.timeout.ms=30000`.
9. Kafka supports exactly-once delivery. Therefore, ensure the idempotency of processing messages for services.
10. Always close the consumer before exiting. Otherwise, consumers in the same group may be blocked within the timeout set by **session.timeout.ms**.
11. Do not start a consumer group name with a special character, such as a number sign (#). Otherwise, monitoring data of the consumer group cannot be displayed.

Producers

1. Synchronous replication: Set **acks** to **all**.
2. Retry message sending: Set **retries** to **3**.
3. Optimize message sending: For latency-sensitive messages, set **linger.ms** to **0**. For latency-insensitive messages, set **linger.ms** to a value ranging from **100** to **1000**.
4. Ensure that the producer has sufficient JVM memory to avoid blockages.
5. Set the timestamp to the local time. Messages will fail to age if the timestamp is a future time.

Topics

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

You can enable or disable automatic topic creation. If automatic topic creation is enabled, the system automatically creates a topic when a message is 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, synchronous replication and flushing disabled, **CreateTime** message timestamp, and maximum 10,485,760 bytes message size.

Others

Maximum number of connections: 3000

Maximum size of a message: 10 MB

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

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

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

7.4 Connecting to Kafka Using the Client (SASL Disabled)

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

Each Kafka broker allows a maximum of 1000 connections from each IP address. Excess connections will be rejected. You can change the limit by referring to [Modifying Kafka Instance Configuration Parameters](#).

Prerequisites

- Security group rules have been properly configured.
To access a Kafka instance with SASL disabled, configure proper security group rules. For details about security group configuration requirements, see [Table 7-2](#).
- 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 7-1 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 

- If automatic topic creation is not enabled for the Kafka instance, [create a topic](#) before connecting to the instance.

- Kafka CLI [v1.1.0](#), [v2.3.0](#), or [v2.7.2](#) is available. Ensure that the Kafka instance and the CLI use the same version.
- An ECS has been created. For intra-VPC access, ensure that its VPC, subnet, and security group configurations are the same as those of the Kafka instance. [JDK v1.8.111 or later](#) has been installed on the ECS, and the **JAVA_HOME** and **PATH** environment variables have been configured as follows:

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

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

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

Accessing the Instance Using CLI

The following uses Linux as an example.

Step 1 Decompress the Kafka CLI package.

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

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

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

For example:

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

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

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

Step 3 Run the following command to create messages:

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

Parameter description:

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

The following example uses connection addresses **10.xx.xx.45:9092,10.xx.xx.127:9092,10.xx.xx.103:9092**. After running the preceding command, you can send a message to the Kafka instance by writing it and pressing **Enter**. Each line of content is sent as a message.

```
[root@ecs-kafka bin]# ./kafka-console-producer.sh --broker-list
10.xx.xx.45:9092,10.xx.xx.127:9092,10.xx.xx.103:9092 --topic topic-demo
>Hello
>DMS
>Kafka!
>^C[root@ecs-kafka bin]#
```

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

Step 4 Run the following command to retrieve messages:

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

Parameter description:

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

Example:

```
[root@ecs-kafka bin]# ./kafka-console-consumer.sh --bootstrap-server  
10.xx.xx.45:9092,10.xx.xx.127:9092,10.xx.xx.103:9092 --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

7.5 Connecting to Kafka Using the Client (SASL Enabled)

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 Instance Configuration Parameters](#).

Prerequisites

- Security group rules have been properly configured.
To access a Kafka instance with SASL enabled, configure proper security group rules. For details about security group configuration requirements, see [Table 7-2](#).
- 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 7-2 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 

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

Figure 7-3 SASL mechanism in use

SASL Mechanism SCRAM-SHA-512

- If automatic topic creation is not enabled for the Kafka instance, [create a topic](#) before connecting to the instance.
- The **client.truststore.jks** certificate has been downloaded. Click the Kafka instance to go to the **Basic Information** tab page. Click **Download** next to **SSL Certificate** in the **Connection** area. Download and decompress the package to obtain the client certificate file **client.truststore.jks**.
- Kafka CLI [v1.1.0](#), [v2.3.0](#), or [v2.7.2](#) is available. Ensure that the Kafka instance and the CLI use the same version.
- 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 addr} --topic ${topic name} --producer.config ../config/producer.properties
```

Parameter description:

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

The following example uses connection addresses **10.xx.xx.45:9093,10.xx.xx.127:9093,10.xx.xx.103:9093**.

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

```
[root@ecs-kafka bin]#./kafka-console-producer.sh --broker-list
10.xx.xx.45:9093,10.xx.xx.127:9093,10.xx.xx.103:9093 --topic topic-demo --producer.config ../config/
producer.properties
>Hello
>DMS
>Kafka!
>^C[root@ecs-kafka bin]#
```

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

Step 6 Run the following command to retrieve messages:

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

Parameter description:

- *{connection-address}*: the address obtained in [Prerequisites](#).
- *{topic-name}*: the name of the topic created for the Kafka instance.
- *{consumer-group-name}*: the consumer group name set based on your service requirements. **If a consumer group name has been specified in the configuration file, ensure that you use the same name in the command line. Otherwise, consumption may fail.** If a consumer group name starts

with a special character, such as an underscore (`_`) or a number sign (`#`), the monitoring data cannot be displayed.

Example:

```
[root@ecs-kafka bin]# ./kafka-console-consumer.sh --bootstrap-server
10.xx.xx.45:9093,10.xx.xx.127:9093,10.xx.xx.103:9093 --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

7.6 Connecting to Kafka on the Console

Scenario


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

Prerequisites

Messages can be produced only when the instance is in the **Running** state.


Producing Messages on the Console

Step 1 Log in to the console.

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

 **NOTE**

Select the region where your Kafka instance is located.

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

Step 4 Click the desired Kafka instance to view details.

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

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

Step 7 Set message parameters by referring to [Table 7-5](#).

Table 7-5 Message parameters

Parameter	Description
Message Body	Message content.

Parameter	Description
Message Key	Message key.
Specify Partition	Indicates whether to enable the function of sending messages to a specified partition. <ul style="list-style-type: none">• Off: Messages are sent to partitions based on their key hash.• On: Messages are sent to specified partitions. Requires the partition ID.

Step 8 Click **OK**.

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

----End

8 Managing Messages


8.1 Viewing Kafka Messages

Scenario

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

Viewing Kafka Messages

Step 1 Log in to the console.

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

 **NOTE**

Select the region where your Kafka instance is located.

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

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

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

Step 6 Set the query parameters by referring to [Table 8-1](#).

Table 8-1 Message query parameters

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

Parameter	Description
Search By	The following methods are supported: <ul style="list-style-type: none"> • Creation time: Search by the time that messages are created. • Offset: Search by the message position.
Search For	This parameter is displayed only when Search By is set to Creation time . Enter a keyword in the message body. NOTE Due to resource and performance restrictions, query with content is limited to 10 results. Each search covers at most 10,000 records, or 200 MB. For large records (> 20 KB per message) or a long period, dump messages for offline query.


 **NOTE**

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

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

The query result is as follows.

Figure 8-1 Querying topic messages

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

Parameter description:

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

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

 NOTE

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

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

----End

8.2 Changing Kafka Message Retention Period

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 Instance Configuration Parameters](#).

 NOTE

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


Changing Kafka Message Retention Period

Step 1 Log in to the console.

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

 NOTE

Select the region where your Kafka instance is located.

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

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

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

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

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

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

----End

8.3 Deleting Kafka Messages

Scenario

This section describes how to delete messages on the console.

NOTICE

Deleted messages cannot be recovered.

Prerequisites

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


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

NOTICE

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

Deleting Kafka Messages

Step 1 Log in to the console.

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

NOTE

Select the region where your Kafka instance is located.

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

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

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

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

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

- If SASL has been enabled for the instance, choose **More > Delete Messages** in the row that contains the topic whose messages you want to delete.

Step 7 Set the parameters for deleting messages, as shown in [Table 8-2](#).

Figure 8-2 Deleting messages

Delete Messages

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

Partition	Offset	Operati...
Partition0	10	Delete

[+ Add Partition](#)

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

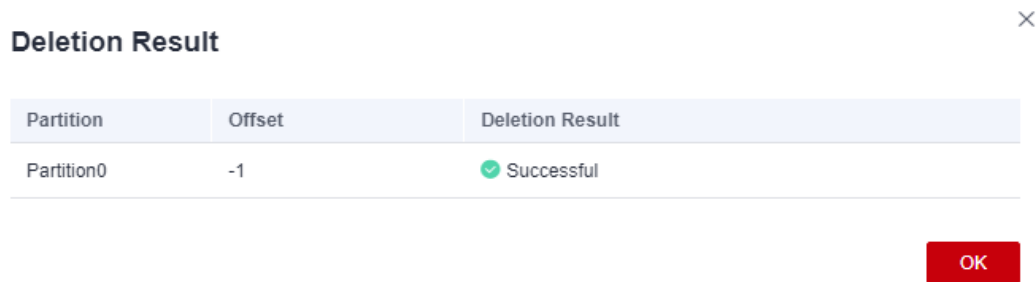
Table 8-2 Parameters for deleting a message

Parameter	Description
Partition	Select the ID of the partition where the message is located.
Offset	Enter the offset. Data before this offset will be deleted. NOTE <ul style="list-style-type: none"> • If Offset is set to -1, all messages in the partition will be deleted. • If the offset you entered is not between the earliest offset and the latest offset of the specified partition, no messages will be deleted.

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

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

Figure 8-3 Deletion result

A dialog box titled "Deletion Result" with a close button (X) in the top right corner. It contains a table with three columns: "Partition", "Offset", and "Deletion Result". The table has one row with the values "Partition0", "-1", and "Successful" (with a green checkmark icon). Below the table is a red "OK" button.

Partition	Offset	Deletion Result
Partition0	-1	✓ Successful

----End

9 Managing Consumer Groups

9.1 Creating a Kafka Consumer Group

Create a consumer group on the console.


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

 **NOTE**

- If **auto.create.groups.enable** is set to **true**, the consumer group status is **EMPTY**, and no offset has been submitted, the system automatically deletes the consumer group 10 minutes later.
- If **auto.create.groups.enable** is set to **false**, the system does not automatically delete consumer groups. You can manually delete them.
- Creating a consumer group on the console does not require instance restart.

Creating a Kafka Consumer Group

Step 1 Log in to the console.

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

 **NOTE**

Select the region where your Kafka instance is located.

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

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

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

Step 6 Click **Create Consumer Group**.

Step 7 Enter the consumer group name, and click **OK**.

View the new consumer group in the consumer group list.


----End

9.2 Querying the Kafka Consumer Group List

This section describes how to query the consumer group list.

Viewing the Consumer Group List (Console)

Step 1 Log in to the console.

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

 **NOTE**

Select the region where your Kafka instance is located.

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

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


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


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

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

Figure 9-1 Consumer group list

<input type="checkbox"/> Consumer Group Name	Status	Coordinator (ID)	Operation
<input type="checkbox"/> test	EMPTY	2	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 username and password, and 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. Add the username and password, and the SSL certificate configuration 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
```

9.3 Viewing Kafka Consumer Details

This section describes how to view the consumer list and consumer connection addresses.

NOTE


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

Prerequisites

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

Viewing the Consumer List (Console)

Step 1 Log in to the console.

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

NOTE

Select the region where your Kafka instance is located.

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


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

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

Step 6 Click the name of the desired consumer group.

Step 7 On the **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:
 - (Optional) If the username and password, and 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. Add the username and password, and the SSL certificate configuration by referring to [Step 3](#).
 - 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 Connection Addresses (Console)

Step 1 Log in to the console.

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

 **NOTE**

Select the region where your Kafka instance is located.

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

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

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

Step 6 Click the desired consumer group.

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

Figure 9-2 Consumer list

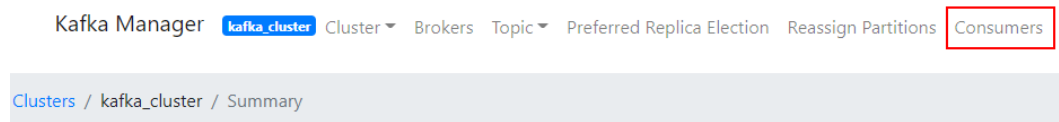
ID	Address	Client ID
consumer-1-6883fa94-d355-4403-b5d8-251a16f2b53d	/124...183	consumer-1

----End

Viewing Consumer Connection Addresses (Kafka Manager)

- Step 1** [Log in to Kafka Manager](#).
- Step 2** Click `kafka_cluster` to go to the cluster details page.
- Step 3** On the top menu bar, choose **Consumers**.

Figure 9-3 Navigation bar



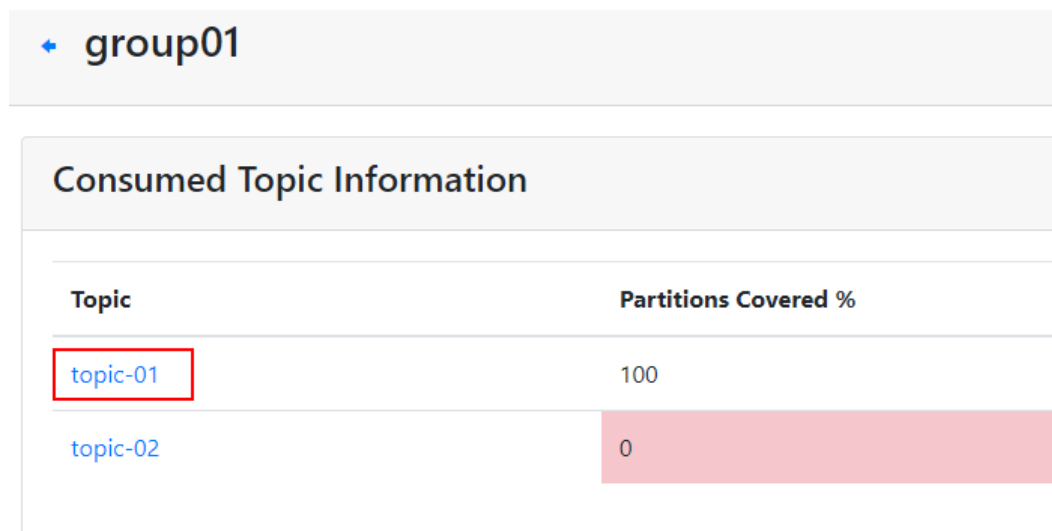
- Step 4** Click the desired consumer group to view the topics that the group has subscribed to.

Figure 9-4 Consumer group list

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

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

Figure 9-5 Topics that the consumer group has subscribed to



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

Figure 9-6 Topic details page

Partition	LogSize	Consumer Offset	Lag	Consumer Instance Owner
0	33,333	0	33,333	consumer-1-5d096c5f-159d-468d-8b10-7961dc6f49d1/10.234.177.46
1	33,334	0	33,334	consumer-1-5d096c5f-159d-468d-8b10-7961dc6f49d1/10.234.177.46
2	33,333	0	33,333	consumer-1-5d096c5f-159d-468d-8b10-7961dc6f49d1/10.234.177.46

----End

9.4 Viewing and Resetting Kafka Consumption Offsets

This section describes how to view and reset consumption offsets. Resetting consumption offsets is to change the consumption position for consumers.

NOTICE

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

Prerequisites


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

NOTICE

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


Viewing Consumer Offsets (Console)

Step 1 Log in to the console.

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

 **NOTE**

Select the region where your Kafka instance is located.

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

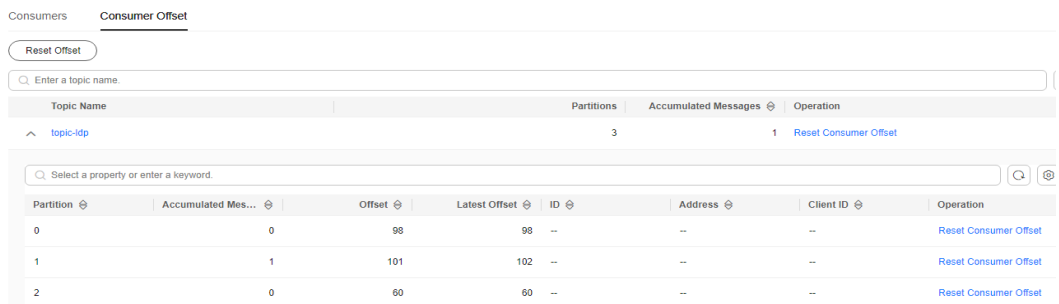
Step 4 Click the desired Kafka instance to view the instance details.

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

Step 6 Click the name of the desired consumer group.

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


Figure 9-7 Consumer offsets



The screenshot shows the 'Consumer Offset' tab in a console. At the top, there is a 'Reset Offset' button and a search box for 'Enter a topic name'. Below this is a table with columns: Topic Name, Partitions, Accumulated Messages, and Operation. A single row is visible with 'topic-ldp', '3', '1', and a 'Reset Consumer Offset' link. Below this is another search box 'Select a property or enter a keyword.' and a detailed table with columns: Partition, Accumulated Mes..., Offset, Latest Offset, ID, Address, Client ID, and Operation. The detailed table has three rows of data for partitions 0, 1, and 2.

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

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

Step 8 (Optional) To query the consumer offsets of a specific topic, enter the topic name in the search box and 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:
 - (Optional) If the username and password, and 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. Add the username and password, and the SSL certificate configuration 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
```

## Resetting Consumer Offsets

**Step 1** Log in to the console.

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

 **NOTE**

Select the region where your Kafka instance is located.

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

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

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

**Step 6** Click the name of the desired consumer group.

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

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

**Step 8** In the displayed **Reset Consumer Offset** dialog box, set the parameters by referring to [Table 9-1](#).

**Table 9-1** Parameters for resetting the consumer offset

| Parameter | Description                                                                                                                                                                                                                                                                                                                          |
|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Reset By  | You can reset an offset by: <ul style="list-style-type: none"> <li>• Time: Reset the offset to the specified time.</li> <li>• Offset: Reset the offset to the specified position.</li> </ul> If you reset offsets in batches, they can only be reset to the specified time.                                                          |
| Time      | Set this parameter if <b>Reset By</b> is set to <b>Time</b> .<br>Select a time point. After the reset is complete, retrieval starts from this time point. <ul style="list-style-type: none"> <li>• <b>Earliest</b>: earliest offset</li> <li>• <b>Custom</b>: a custom time point</li> <li>• <b>Latest</b>: latest offset</li> </ul> |

| Parameter | Description                                                                                                                                                                              |
|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Offset    | Set this parameter if <b>Reset By</b> is set to <b>Offset</b> .<br>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


## 9.5 Exporting Kafka Consumer Groups

### Scenario

Export the consumer group list from the console.

### Exporting Kafka Consumer Groups

**Step 1** Log in to the console.

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

 **NOTE**

Select the region where your Kafka instance is located.

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

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

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

**Step 6** Click **Export** and choose to export all data or selected data to an XLSX file to export the consumer group list.

----End

## 9.6 Deleting a Kafka Consumer Group

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

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

### Prerequisites


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

## Constraints

- If **auto.create.groups.enable** is set to **true**, the consumer group status is **EMPTY**, and no offset has been submitted, the system automatically deletes the consumer group 10 minutes later.
- If **auto.create.groups.enable** is set to **false**, the system does not automatically delete consumer groups. You can manually delete them.

## Method 1: Deleting a Consumer Group on the Console

**Step 1** Log in to the console.

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

 **NOTE**

Select the region where your Kafka instance is located.

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

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

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

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

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

---

**NOTICE**

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

---

Consumer group statuses include:

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

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

----End

## 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 [Connecting to Kafka Using the Client \(SASL Disabled\)](#) or [Connecting to Kafka Using the Client \(SASL Enabled\)](#).

**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 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 [Connecting to Kafka Using the Client \(SASL Enabled\)](#).

----End

# 10 Managing Quotas

---

## 10.1 Configuring Kafka Quotas

### Scenario

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

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

### Operation Impact


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

### Prerequisites

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


### Creating a User or Client Quota

**Step 1** Log in to the console.

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

 **NOTE**

Select the region where your Kafka instance is located.

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

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

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

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

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

**Step 8** Set quota parameters.

**Table 10-1** Quota parameters

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

 **NOTE**

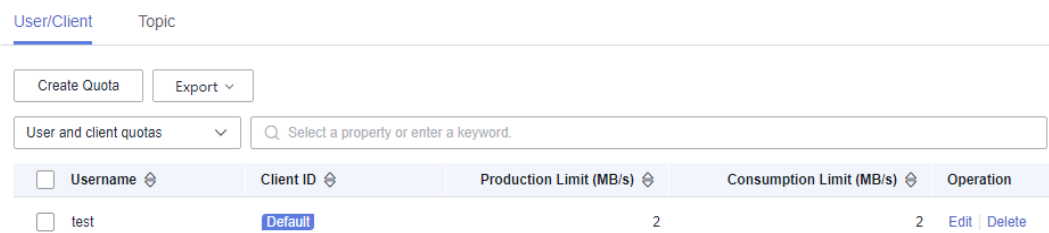
- **Username** is not displayed in the **Create Quota** dialog box for instances with SASL disabled.
- **Username** and **Client ID** cannot be both empty.
- **Production Limit** and **Consumption Limit** cannot be both empty.

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

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




**Figure 10-1** Viewing the new quota



----End

## Creating a Topic Quota

**Step 1** Log in to the console.

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

 **NOTE**

Select the region where your Kafka instance is located.

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

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

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

**Step 6** Click the **Topic** tab.

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

**Step 8** Set quota parameters.

**Table 10-2** Quota parameters

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

 **NOTE**

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


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

Go to the **Kafka Quotas > Quotas** page. On the **Topic** tab page, enter the name of the new quota in the upper left corner, then press **Enter** to view the created quota.

----End


## Modifying a Quota

**Step 1** Log in to the console.

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

 **NOTE**

Select the region where your Kafka instance is located.

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

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

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

**Step 6** In the row containing the desired quota, click **Edit**.

**Step 7** Change the production limit or consumption limit, and click **OK**. The **Background Tasks** page is displayed. If the status of the quota modification task is **Successful**, the quota has been modified.

Choose **Kafka Quotas > Quotas** and view the new production or consumption rate limit.


 **NOTE**

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

----End


## Exporting Quotas

**Step 1** Log in to the console.

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

 **NOTE**

Select the region where your Kafka instance is located.

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

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

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

**Step 6** Export quotas.


- On the **User/Client** tab page: Choose **Export > Export all data to an XLSX file** or **Export selected data to an XLSX file**.

- On the **Topic** tab page: Choose **Export > Export all data to an XLSX file** or **Export selected data to an XLSX file**.

----End


## Deleting a Quota

**Step 1** Log in to the console.

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

 **NOTE**

Select the region where your Kafka instance is located.

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

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

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

**Step 6** In the row containing the desired quota, click **Delete**.

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


----End

## 10.2 Monitoring Kafka Quotas

This section describes how to view the quota usage of users, clients, and topics of each broker on the console.


### Viewing Quota Usage

**Step 1** Log in to the console.

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

 **NOTE**

Select the region where your Kafka instance is located.

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

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

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

**Step 6** Set quota monitoring parameters.

**Table 10-3** Quota monitoring parameters

| Parameter      | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Search By      | Specify a method of calculating rate limits. <ul style="list-style-type: none"> <li>● <b>Ranked:</b> Show the specified number of users, clients, or topics that have used the most bandwidth.</li> <li>● <b>Bandwidth:</b> Show users, clients, or topics whose bandwidth rate is higher than your specified value.</li> <li>● <b>Bandwidth usage:</b> Show users, clients, or topics whose bandwidth usage is higher than your specified percentage.</li> </ul> |
| Bandwidth From | Specify the source of rate limit calculation. <ul style="list-style-type: none"> <li>● <b>Production:</b> Count production rate limits.</li> <li>● <b>Consumption:</b> Count consumption rate limits.</li> </ul>                                                                                                                                                                                                                                                  |
| Dimension      | Specify the dimension of rate limit calculation. <ul style="list-style-type: none"> <li>● <b>User/Client:</b> Count user/client rate limits.</li> <li>● <b>Topic:</b> Count topic rate limits.</li> </ul>                                                                                                                                                                                                                                                         |

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

----End

# 11 Managing Instances

## 11.1 Viewing and Modifying Basic Information of a Kafka Instance


### Scenario

This section describes how to view the details, and modify the basic information of a Kafka instance on the console.

After creating a Kafka instance, you can modify some parameters of it as required, including the instance name, description, security group, and capacity threshold policy.


### Viewing Kafka Instance Details

**Step 1** Log in to the console.

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

 **NOTE**

Select the region where your Kafka instance is located.

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

**Step 4** Search for a Kafka instance by specifying filters. You can filter instances by tag, status, name, version, flavor, used/available storage space, maximum partitions, billing mode, and enterprise project. For Kafka instance statuses, see [Table 11-1](#).

**Table 11-1** Kafka instance status description

| Status   | Description                    |
|----------|--------------------------------|
| Creating | The instance is being created. |

| Status          | Description                                                                                                                                                         |
|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Creation failed | The instance failed to be created.                                                                                                                                  |
| Running         | The instance is running properly.<br>Only instances in the <b>Running</b> state can provide services.                                                               |
| Faulty          | The instance is not running properly.                                                                                                                               |
| Restarting      | The instance is being restarted.                                                                                                                                    |
| Changing        | The instance specifications are being changed.                                                                                                                      |
| Change failed   | The instance specifications failed to be changed.<br>You cannot restart, delete, or modify an instance in the <b>Change failed</b> state. Contact customer service. |

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

**Table 11-2** 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 11-2** Connection parameters

| Section    | Parameter                          | Description                                                                                               |
|------------|------------------------------------|-----------------------------------------------------------------------------------------------------------|
| Connection | Username                           | Username for accessing the instance with SASL_SSL enabled.                                                |
|            | Kafka SASL_SSL                     | Whether SASL_SSL is enabled.                                                                              |
|            | SASL Mechanism                     | SASL mechanism used by the instance with SASL_SSL enabled.                                                |
|            | SSL Certificate                    | Click <b>Download</b> to download the SSL certificate for accessing the instance.                         |
|            | Instance Address (Private Network) | Address for accessing the instance.<br>The number of connection addresses is the same as that of brokers. |
|            | Manager Address (Private Network)  | Address for connecting to Kafka Manager.                                                                  |
|            | Manager Username                   | Username for connecting to Kafka Manager.                                                                 |

----End


## Modifying Basic Information of a Kafka Instance

**Step 1** Log in to the console.

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

 **NOTE**

Select the region where your Kafka instance is located.

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

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

**Step 5** Modify the following parameters if needed:

- Instance Name
- Enterprise Project (Changing does not require instance restart.)
- Description
- Security Group
- Capacity Threshold Policy (Changing does not require instance restart.)
- Automatic Topic Creation (Changing requires instance restart.)

After modifying them, view the result in one of the following ways:

- If **Capacity Threshold Policy** or **Automatic Topic Creation** has been modified, the **Background Tasks** page will be displayed and you can view the task progress and result there.
- If **Instance Name**, **Description**, **Enterprise Project**, or **Security Group** has been modified, the result will be displayed in the upper right corner of the page.


----End

## 11.2 Viewing Kafka Disk Usage

This section describes how to view the disk usage of each broker on the Kafka console.

### Viewing Kafka Disk Usage

**Step 1** Log in to the console.

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

 **NOTE**

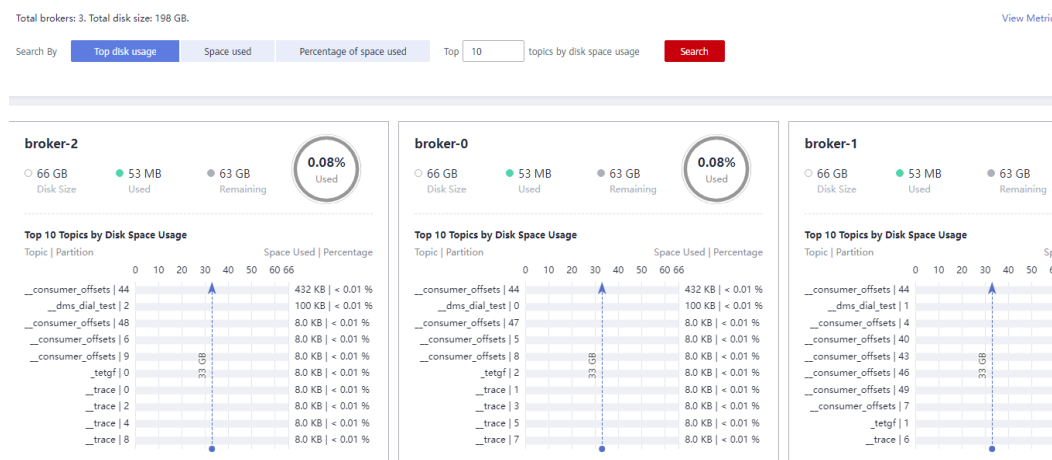
Select the region where your Kafka instance is located.

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

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

**Step 5** Go to the **Disk Usage Statistics** page.

**Figure 11-1** Viewing disk usage



You can query topics that use the most disk space or topics that have used a specified amount or percentage of disk space.

In the upper right corner of the page, click **View Metric**. On the displayed Cloud Eye page, you can view metrics of Kafka instances.

----End

## 11.3 Viewing Kafka Background Tasks

After you initiate certain instance operations such as 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.


### Viewing Kafka Background Tasks

**Step 1** Log in to the console.

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

 **NOTE**

Select the region where your Kafka instance is located.

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


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

**Step 5** In the navigation pane, choose **Background Tasks**.

**Step 6** On the **Background Tasks** page, click the time drop-down box, specify time, enter keywords in the search box, and press **Enter**. Tasks started in the specified time will be displayed.



On the **Background Tasks** page, you can also perform the following operations:

- Click  to refresh the task status.
- Click **Delete**. In the displayed **Delete Task** dialog box, click **OK** to clear the task information.

 **NOTE**

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


----End

## 11.4 Viewing Sample Code of Kafka Production and Consumption

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

### Viewing Sample Code of Kafka Production and Consumption

**Step 1** Log in to the console.

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

 **NOTE**

Select the region where your Kafka instance is located.

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

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

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

**Step 6** Click **View Sample Code**. The **Sample Code** dialog box is displayed.

View sample code for creating and retrieving messages in Java, Go, and Python. Set **Access By** to **PlainText** to view the sample code where SASL\_SSL authentication is disabled. Set **Access By** to **SASL\_SSL** to view the sample code where SASL\_SSL authentication is enabled.

----End

## 11.5 Modifying Kafka Instance Configuration Parameters

### Scenario

Your Kafka instances, topics, and consumers come with default configuration parameter settings. You can modify common parameters on the Kafka console.

For details about parameters that are not listed on the console, see the [Kafka official website](#).

Kafka instances have dynamic and static parameters:

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

 **NOTE**


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

## Prerequisites

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


## Modifying Kafka Instance Configuration Parameters

**Step 1** Log in to the console.

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

 **NOTE**

Select the region where your Kafka instance is located.

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

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

**Step 5** On the **Parameters** page, click **Edit** in the row containing the parameter to modify.

Parameters of v1.1.0 instances are described in [Table 11-4](#) and [Table 11-3](#).  
Parameters of v2.3.0/v2.7 instances are described in [Table 11-5](#) and [Table 11-6](#).

**Table 11-3** Dynamic parameters (v1.1.0 instances)

| Parameter                 | Description                                      | Value Range | Default Value |
|---------------------------|--------------------------------------------------|-------------|---------------|
| auto.create.groups.enable | Whether to automatically create consumer groups. | true/false  | true          |

| Parameter                 | Description                                                                                                                                                                                                                                                                                                                                                         | Value Range | Default Value |
|---------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|---------------|
| offsets.retention.minutes | <p>The longest period a consumption position can be retained starts from the time of submission. Positions retained beyond this duration will be deleted. Each time a consumption position is submitted to a topic partition, its retention period resets to 0. The unit is minute.</p> <p>This is a static parameter for instances created before May 1, 2023.</p> | 1440–30240  | 20160         |

**Table 11-4** Static parameters (v1.1.0 instances)

| Parameter                      | Description                                                                                                                                                                                                                     | Value Range                 | Default Value |
|--------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------|---------------|
| min.insync.replicas            | If a producer sets the acks parameter to <b>all</b> (or <b>-1</b> ), the <b>min.insync.replicas</b> 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.                                                                              | <b>true</b> or <b>false</b> | 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            | <p>Duration (in hours) for retaining a log file.</p> <p>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.</p>          | 1–168                       | 72            |

| Parameter                      | Description                                                                                                                                                                                                                                                    | Value Range    | Default Value |
|--------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------|---------------|
| max.connections.per.ip         | The maximum number of connections allowed from each IP address. Request for new connections will be rejected once the limit is reached.                                                                                                                        | 100–20,000     | 1000          |
| group.max.session.timeout.ms   | The maximum session timeout (in ms) for consumers. A longer timeout gives consumers more time to process messages between heartbeats but results in a longer time to detect failures.                                                                          | 6000–1,800,000 | 1,800,000     |
| default.replication.factor     | The default number of replicas configured for an automatically created topic.                                                                                                                                                                                  | 1–3            | 3             |
| allow.everyone.if.no.acl.found | When this parameter is set to <b>true</b> , all users can access resources without ACL rules.<br>This parameter is displayed only when SASL is enabled for the instance.<br>This parameter cannot be modified for instances created before September 15, 2023. | true/false     | true          |
| num.partitions                 | The default number of partitions configured for each automatically created topic.                                                                                                                                                                              | 1–200          | 3             |
| group.min.session.timeout.ms   | The minimum session timeout (in ms) for consumers. A shorter timeout enables quicker failure detection but results in more frequent consumer heartbeating, which can overwhelm broker resources.                                                               | 6000–300,000   | 6000          |

**Table 11-5** Dynamic parameters (2.3.0/2.7 instances)

| Parameter                      | Description                                                                                                                                                                                                                                                                                                                                                  | Value Range                 | Default Value |
|--------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------|---------------|
| min.insync.replicas            | If a producer sets the acks parameter to <b>all</b> (or <b>-1</b> ), the <b>min.insync.replicas</b> parameter specifies the minimum number of replicas that must acknowledge a write for the write to be considered successful.                                                                                                                              | 1–3                         | 1             |
| message.max.bytes              | Maximum length of a single message, in bytes.                                                                                                                                                                                                                                                                                                                | 0–10,485,760                | 10,485,760    |
| auto.create.groups.enable      | Whether to automatically create consumer groups.                                                                                                                                                                                                                                                                                                             | true/false                  | true          |
| max.connections.per.ip         | The maximum number of connections allowed from each IP address. Request for new connections will be rejected once the limit is reached.                                                                                                                                                                                                                      | 100–20,000                  | 1000          |
| unclean.leader.election.enable | Indicates whether to allow replicas not in the ISR set to be elected as the leader as a last resort, even though doing so may result in data loss.                                                                                                                                                                                                           | <b>true</b> or <b>false</b> | true          |
| offsets.retention.minutes      | The longest period a consumption position can be retained starts from the time of submission. Positions retained beyond this duration will be deleted. Each time a consumption position is submitted to a topic partition, its retention period resets to 0. The unit is minute.<br><br>This is a static parameter for instances created before May 1, 2023. | 1440–30240                  | 20160         |

**Table 11-6** Static parameters (2.3.0/2.7 instances)

| Parameter                      | Description                                                                                                                                                                                                                                                            | Value Range    | Default Value |
|--------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------|---------------|
| connections.max.idle.ms        | Idle connection timeout (in ms). Connections that are idle for the duration specified by this parameter will be closed.                                                                                                                                                | 5000–600,000   | 600,000       |
| log.retention.hours            | Duration (in hours) for retaining a log file.<br><br>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             |
| allow.everyone.if.no.acl.found | When this parameter is set to <b>true</b> , all users can access resources without ACL rules.<br><br>This parameter is displayed only when SASL is enabled for the instance.<br><br>This parameter cannot be modified for instances created before September 15, 2023. | true/false     | true          |
| num.partitions                 | The default number of partitions configured for each automatically created topic.                                                                                                                                                                                      | 1–200          | 3             |
| group.min.session.timeout.ms   | The minimum session timeout (in ms) for consumers. A shorter timeout enables quicker failure detection but results in more frequent consumer heartbeating, which can overwhelm broker resources.                                                                       | 6000–300,000   | 6000          |

 NOTE

- To modify multiple dynamic or static parameters at a time, click **Modify** above the parameter list.
- If you want to restore the default values, click **Restore Default** in the row containing the desired parameter.

**Step 6** Click **Save**.

 NOTE

Modifying dynamic parameters will not restart the instance. **Static parameter modification requires manual restart of the instance.**

----End

## 11.6 Configuring Kafka Instance Tags

Tags facilitate Kafka instance identification and management.

You can add tags to a Kafka instance when creating the instance or add tags on the **Tags** tab page of the created instance. Up to 20 tags can be added to an instance. Tags can be deleted.

A tag consists of a tag key and a tag value. [Table 11-7](#) lists the tag key and value requirements.

**Table 11-7** Tag key and value requirements

| Parameter | Requirements                                                                                                                                                                                                                                                                                                            |
|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Tag key   | <ul style="list-style-type: none"> <li>• Cannot be left blank.</li> <li>• Must be unique for the same instance.</li> <li>• Can contain 1 to 128 characters.</li> <li>• Can contain letters, digits, spaces, and special characters <code>_:=+@ : = + - @</code></li> <li>• Cannot start or end with a space.</li> </ul> |
| Tag value | <ul style="list-style-type: none"> <li>• Can contain 0 to 255 characters.</li> <li>• Can contain letters, digits, spaces, and special characters <code>_:=+@ : = + - @</code></li> <li>• Cannot start or end with a space.</li> </ul>                                                                                   |

### Configuring Kafka Instance Tags

**Step 1** Log in to the console.

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

 **NOTE**

Select the region where your Kafka instance is located.


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

**Step 4** Click the name of an instance.

**Step 5** In the navigation pane on the left, choose **Tags**.

View the tags of the instance.

**Step 6** Perform the following operations as required:

- Add a tag
  - a. Click **Create/Delete Tag**.
  - b. Enter a tag key and a tag value, and click **Add**.  
If you have predefined tags, select a predefined pair of tag key and value, and click **Add**.
  - c. Click **OK**.
- Delete a tag  
Delete a tag using either of the following methods:
  - In the row containing the tag to be deleted, click **Delete**. In the **Delete Tag** dialog box, click **Yes**.
  - Click **Create/Delete Tag**. In the dialog box that is displayed, click  next to the tag to be deleted and click **OK**.

----End


## 11.7 Exporting the Kafka Instance List

### Scenario

This section describes how to export the Kafka instance list from the console.


### Exporting the Kafka Instance List

**Step 1** Log in to the console.

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

 **NOTE**

Select the region where your Kafka instance is located.

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

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

----End



## 11.8 Restarting a Kafka 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**.

### Restarting a Kafka Instance

**Step 1** Log in to the console.

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

#### NOTE

Select the region where your Kafka instance is located.

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

**Step 4** 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

## 11.9 Deleting Kafka Instances

### Scenario

Delete one or more Kafka instances at a time on the Kafka console.

#### NOTICE


Deleting a Kafka instance will delete the data in the instance without any backup. Exercise caution when performing this operation.

### Prerequisites

The status of the Kafka instance you want to delete is **Running** or **Faulty**.

### Deleting Kafka Instances

**Step 1** Log in to the console.

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

#### NOTE

Select the region where your Kafka instance is located.

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

**Step 4** Delete pay-per-use Kafka instances in either of the following ways:

- Select one or more Kafka instances and click **Delete** in the upper left corner.
- In the row containing the Kafka instance to be deleted, choose **More > Delete**.
- Click the desired Kafka instance to 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, enter **DELETE** and click **OK** to delete the Kafka instance.

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

----End

## 11.10 Using Kafka Manager

## 11.10.1 Kafka Manager Overview

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.

**Table 11-8** Kafka Manager functions on the Kafka console

| Kafka Manager                                 | Kafka Console                                                                                                                                                                                                                                                                                |
|-----------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Viewing topics about an instance              | View the topic list on the <b>Topics</b> page.                                                                                                                                                                                                                                               |
| Viewing basic information about a topic       | View the basic information (including the number of replicas, number of partitions, and aging time) about each topic on the <b>Topics</b> page.                                                                                                                                              |
| Reassigning topic partitions                  | Reassign partitions automatically or manually on the <b>Topics</b> page.                                                                                                                                                                                                                     |
| Updating topic configurations                 | Modify topic configuration parameters on the <b>Topics</b> page.                                                                                                                                                                                                                             |
| Viewing the consumer group list               | View the consumer group list on the <b>Consumer Groups</b> page.                                                                                                                                                                                                                             |
| Viewing details about a specific consumer     | On the <b>Consumer Groups</b> page, click a consumer group name to go to the consumer group details page and view consumers and their progress.                                                                                                                                              |
| Viewing details of topics in a consumer group | On the <b>Consumer Groups</b> page, click a consumer group name to go to the consumer group details page. On the <b>Consumer Offset</b> tab page, view the topic list of the consumer group, the number of messages accumulated in each topic, and the consumption status of each partition. |
| Monitoring the cluster or topics              | View monitoring information on the <b>Monitoring</b> page.                                                                                                                                                                                                                                   |

### Prerequisites

Security group rules have been configured by referring to [Table 11-9](#).

**Table 11-9** Security group rule

| Direction | Protocol | Port | Source    | Description           |
|-----------|----------|------|-----------|-----------------------|
| Inbound   | TCP      | 9999 | 0.0.0.0/0 | Access Kafka Manager. |

## Logging In to Kafka Manager

- Step 1** Create a Windows ECS with the same VPC and security group configurations as the Kafka instance. For details, see [Purchasing an ECS](#).
- Step 2** Obtain the Kafka Manager address on the instance details page.
- Step 3** Enter the Kafka Manager address in the web browser in the Windows ECS.
- Step 4** Enter the username and password for logging in to Kafka Manager, which you set when creating the instance.

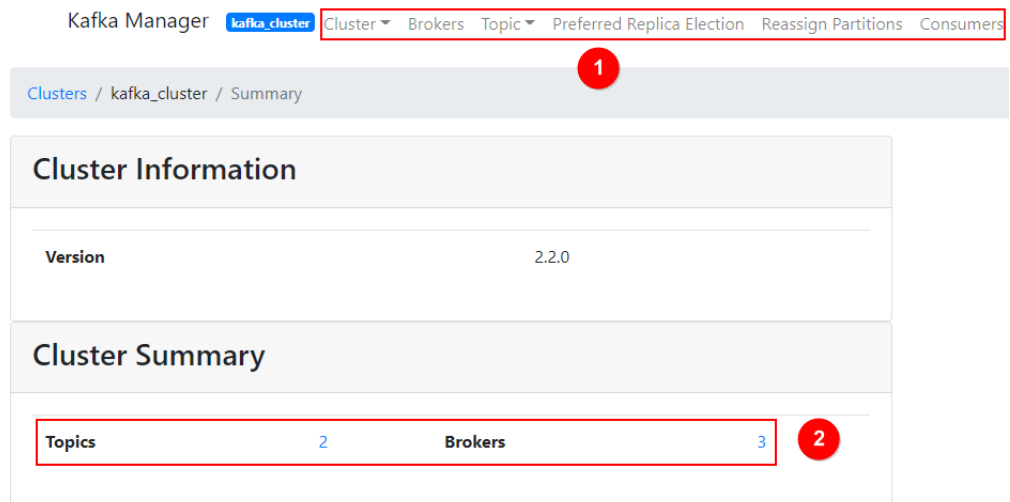
----End

## Viewing Information in Kafka Manager

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

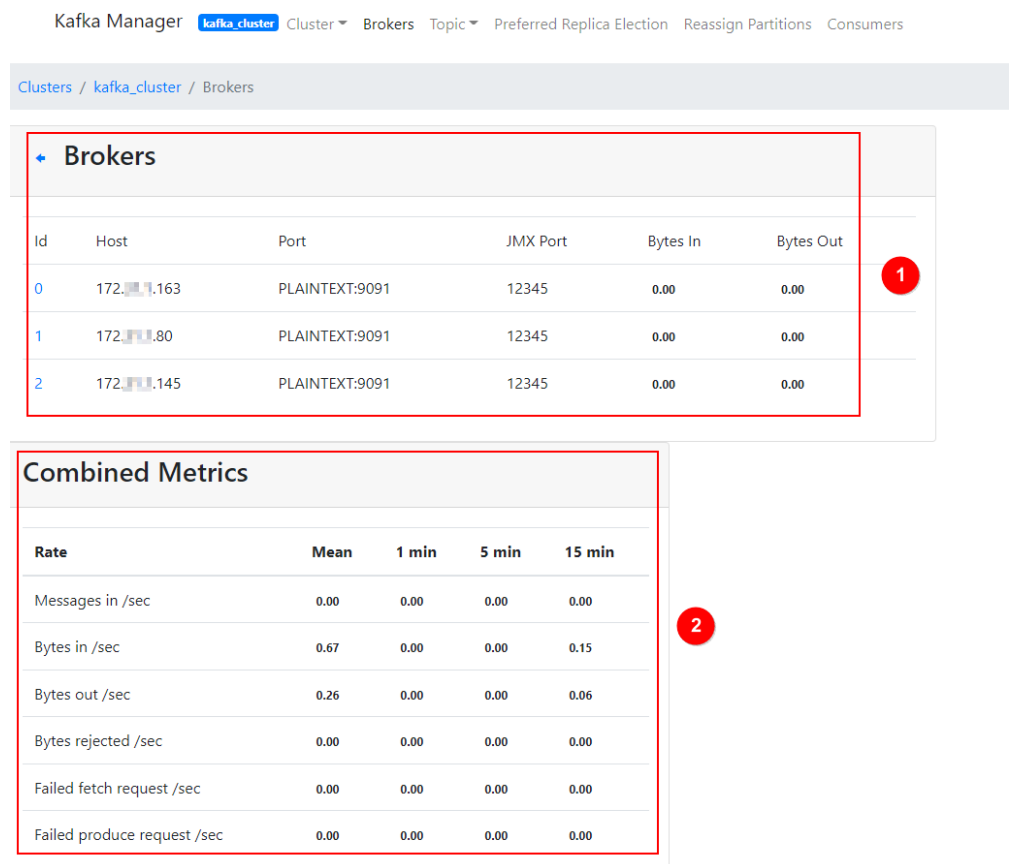
- Information about clusters
  - Click **Clusters** to view the information about clusters. [Figure 11-2](#) shows an example of the cluster information.
    - The top navigation bar provides the following functions, as shown in the red box 1 in the figure.
      - **Cluster**: viewing the list of clusters and cluster information.
      - **Brokers**: viewing information about brokers of a cluster.
      - **Topic**: viewing information about topics in a cluster.
      - **Preferred Replica Election**: electing the leader (preferred replica) of a topic. This operation is not recommended.
      - **Reassign Partitions**: reassigning partitions. This operation is not recommended.
      - **Consumers**: viewing the status of consumer groups in a cluster.
    - Red box 2 shows an example of the cluster information summary, including the number of topics and brokers in the cluster.

**Figure 11-2** Information about clusters



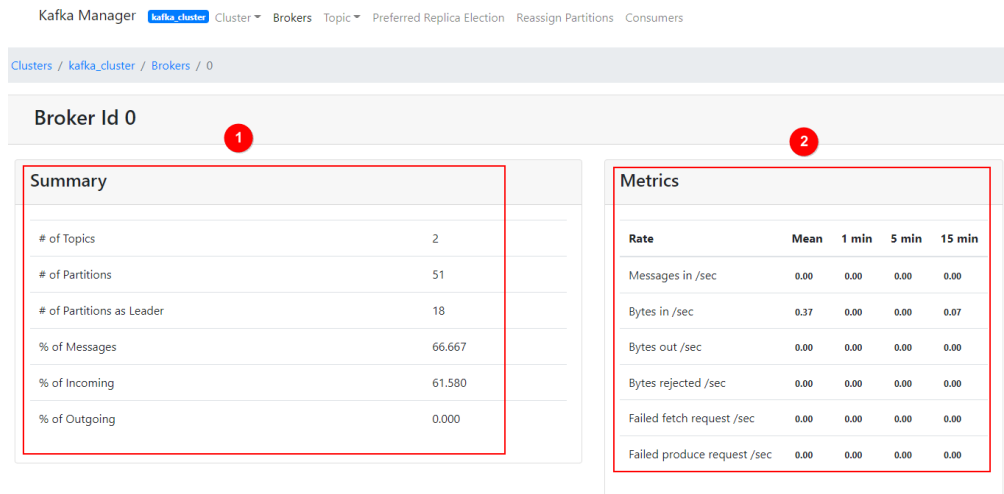
- Combined information about all brokers of a cluster  
This page shows statistics of brokers of a cluster. **Figure 11-3** shows an example of the storage configuration.
  - Red box 1 shows the list of brokers, including number of incoming and outgoing bytes of different brokers.
  - Red box 2 shows the monitoring metrics of the cluster.

**Figure 11-3** Viewing the combined information about all brokers in a cluster



- Information about a specific broker
  - Click the ID of a broker to view its statistics. **Figure 11-4** 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 11-4** Viewing information about a broker

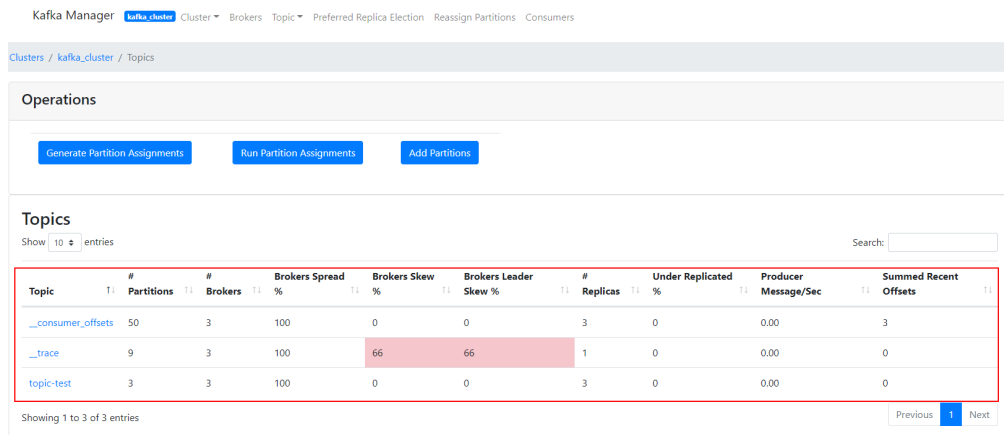


- Topics of an instance
  - In the navigation bar, choose **Topic > List**. The displayed page shows the list of topics and information about the topics, as shown in **Figure 11-5**.

**NOTICE**

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

**Figure 11-5** Topics of an instance

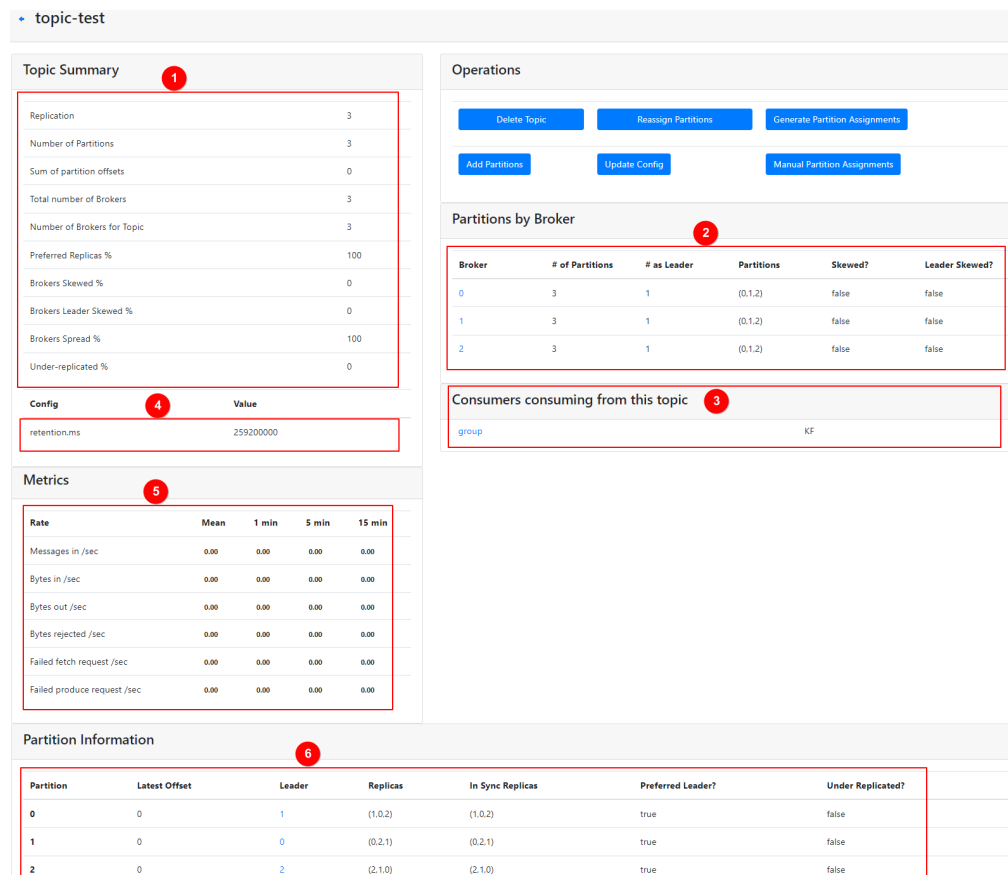


- Details of a topic

Click the name of a topic to view its details on the displayed page, as shown in **Figure 11-6**.

- 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 11-6** Details of a topic



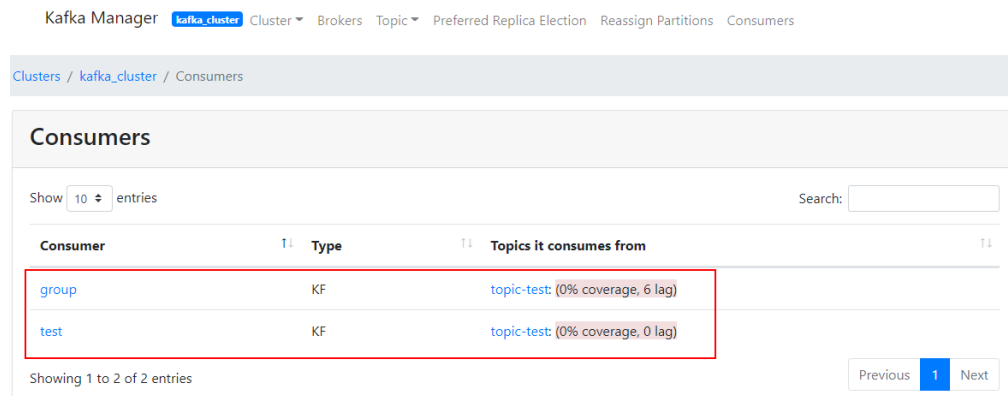
- List of consumers

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

**NOTE**

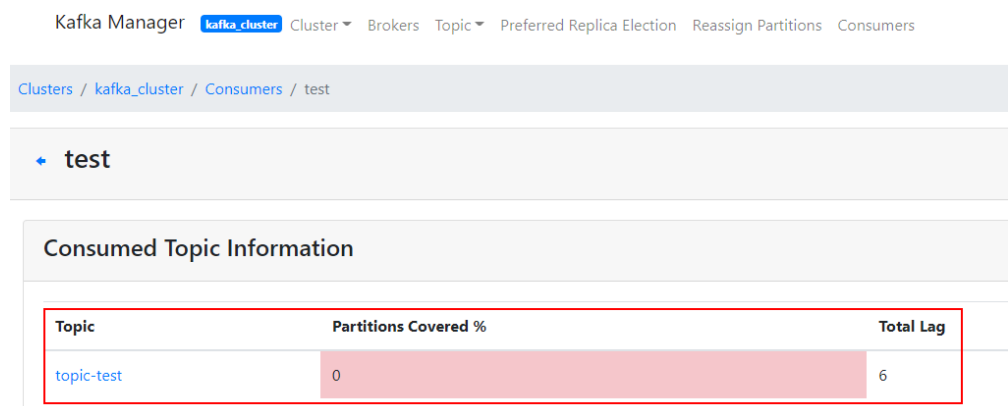
Only consumer groups that have retrieved messages in the last 14 days are displayed.

**Figure 11-7** Viewing the list of consumers



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

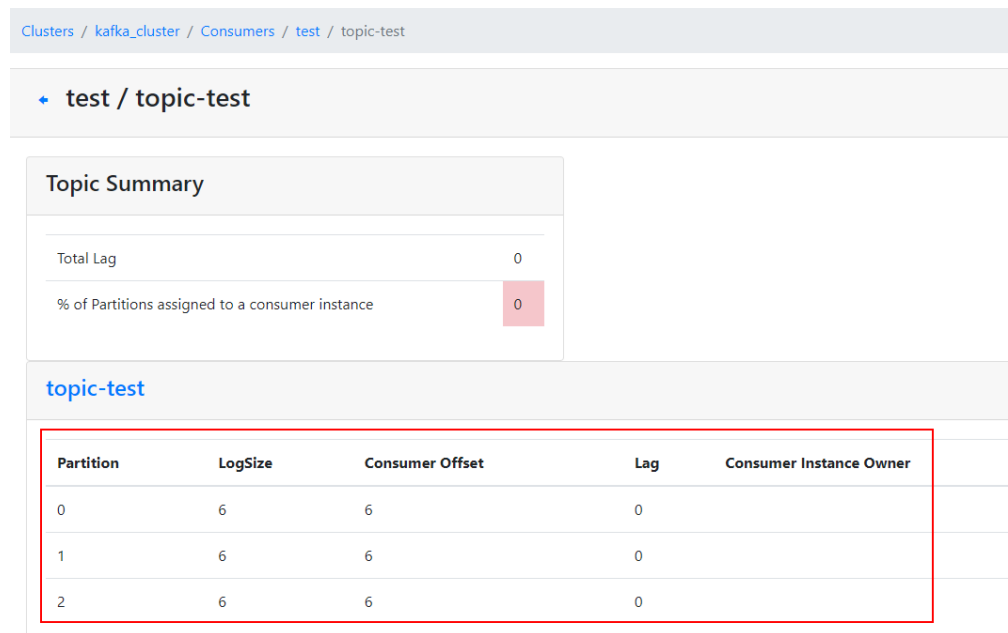
**Figure 11-8** Viewing consumer details



- Details of topics in a consumer  
Click the name of a topic to view retrieval details of different partitions in the topic, including **Partition**, the number of messages in a partition (**LogSize**), progress of the retrieval (**Consumer Offset**), number of remaining messages in the partition that can be retrieved (**Lag**), and the latest consumer that retrieved from the partition (**Consumer Instance Owner**).



**Figure 11-9** Viewing details of a topic



## 11.10.2 Resetting Kafka Manager Password

### Scenario


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

### Prerequisites

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

### Resetting Kafka Manager Password

**Step 1** Log in to the console.

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

 **NOTE**

Select the region where your Kafka instance is located.

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

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

- In the row containing the desired Kafka instance, choose **More > Reset Manager Password**.
- Click the desired Kafka instance to view its details. In the upper right corner, choose **More > Reset Manager Password**.
- Click the desired Kafka instance to view its details. On the **Basic Information** page, click **Reset Manager Password** next to **Manager Username** in the **Connection** section.

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

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

 **NOTE**

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

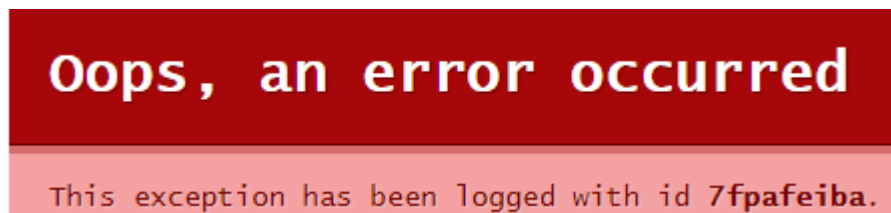
----End

## 11.10.3 Restarting Kafka Manager

### Scenario

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

**Figure 11-10** Error information




 **NOTE**

Restarting Kafka Manager does not affect services.

### Restarting Kafka Manager

**Step 1** Log in to the console.

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

 **NOTE**

Select the region where your Kafka instance is located.

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

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

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

**Step 5** Click **Yes**.

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

----End

# 12 Modifying Kafka 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, **kafka.2u4g.cluster\*3 brokers**).

Figure 12-1 Instance list

| <input type="checkbox"/> | Name                                             | Status                                                                                      | Version | Flavor                        | Used/Available Storage Spa...                                                               |
|--------------------------|--------------------------------------------------|---------------------------------------------------------------------------------------------|---------|-------------------------------|---------------------------------------------------------------------------------------------|
| <input type="checkbox"/> | kafka-doc<br>c6f705cc-358b-470f-a71c-e18...      |  Running | 2.7     | kafka.2u4g.cluster * 3 broker |  0/186 |
| <input type="checkbox"/> | kafka-974006966<br>252b248f-501e-41ed-9832-48... |  Running | 2.3.0   | 100 MB/s                      |  0/186 |

## Impact of Specification Modification


It takes 5 to 10 minutes to modify specifications on one broker. The more brokers, the longer time the modification takes.

**Table 12-1** Impact of specification modification

| Modified Object              | Impact                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Bandwidth or broker quantity | <ul style="list-style-type: none"> <li>Increasing the bandwidth or adding brokers does not affect the original brokers or services.</li> <li>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.</li> <li>New topics are created on new brokers, and the original topics are still on the original brokers, resulting in unbalanced partitions. You can <b>reassign partitions</b> to migrate the replicas of the original topic partitions to the new brokers.</li> </ul> |
| Storage space                | <ul style="list-style-type: none"> <li>You can expand the storage space 20 times.</li> <li>Storage space expansion does not affect services.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |

## Modifying Kafka Instance Specifications

**Step 1** Log in to the console.

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

 **NOTE**

Select the region where your Kafka instance is located.

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

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

**Step 5** Specify the storage space, number of brokers, or bandwidth as required.

**To modify old specifications, perform the following steps:**

- Increase the bandwidth.

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

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

 **NOTE**

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

- Expand the storage space.  
Specify a new storage space and click **Next**. Confirm the configurations and click **Submit**.  
View the new storage space in the **Used/Available Storage Space (GB)** column in the instance list.

**To modify new specifications, perform the following steps:**

- Expand the storage space.  
For **Change By**, select **Storage**. For **Storage Space per Broker**, specify a new storage space, and click **Next**. Confirm the configurations and click **Submit**.  
View the new storage space (Storage space per broker x Number of brokers) in the **Used/Available Storage Space (GB)** column in the instance list.
- Add brokers.  
For **Change 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**

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

----End

# 13 Migrating Data

## 13.1 Kafka Data Migration Overview

### Scenario

You can migrate Kafka services to connect message producers and consumers to a new Kafka instance and can even migrate persisted message data to the new Kafka instance. Kafka services can be migrated in the following two scenarios:

- Migrating services to the cloud without downtime  
Services that have high requirements on continuity must be smoothly migrated to the cloud because they cannot afford a long downtime.
- Re-deploying services on the cloud  
A Kafka instance deployed within an AZ is not capable of cross-AZ disaster recovery. For higher reliability, you can re-deploy services to an instance that is deployed across AZs.

### Preparation

1. Configure the network environment.

To connect to a Kafka instance in a private network, configure a security group with parameters specified as follows.

**Table 13-1** Security group rules

| Direction | Protocol    | Port | Source    | Description                                                                                                                                                                                   |
|-----------|-------------|------|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Inbound   | Select TCP. | 9092 | 0.0.0.0/0 | <ul style="list-style-type: none"><li>• Access a Kafka instance within a VPC (without SSL).</li><li>• Access a Kafka instance across VPCs using a peering connection (without SSL).</li></ul> |

| Direction | Protocol    | Port | Source    | Description                                                                                                                                                                            |
|-----------|-------------|------|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Inbound   | Select TCP. | 9093 | 0.0.0.0/0 | <ul style="list-style-type: none"> <li>Access a Kafka instance within a VPC (with SSL).</li> <li>Access a Kafka instance across VPCs using a peering connection (with SSL).</li> </ul> |

2. Create a Kafka instance.

The specifications of the new instance cannot be lower than the original specifications. For more information, see [Buying a Kafka Instance](#).

3. Create a topic.

Create a topic with the same configurations as the original Kafka instance, including the topic name, number of replicas, number of partitions, message aging time, and whether to enable synchronous replication and flushing. For more information, see [Creating a Kafka Topic](#).

## Migration Scheme 1: Migrating the Production First

Migrate the message production service to the new Kafka instance. After migration, the original Kafka instance will no longer produce messages. After all messages of the original Kafka instance are consumed, migrate the message consumption service to the new Kafka instance to consume messages of this instance.

- Step 1** Change the Kafka connection address of the producer to that of the new Kafka instance.
- Step 2** Restart the production service so that the producer can send new messages to the new Kafka instance.
- Step 3** Check the consumption progress of each consumer group in the original Kafka instance until all data in the original Kafka instance is consumed.
- Step 4** Change the Kafka connection addresses of the consumers to those of the new Kafka instance.
- Step 5** Restart the consumption service so that consumers can consume messages from the new Kafka instance.
- Step 6** Check whether consumers consume messages properly from the new Kafka instance.
- Step 7** The migration is complete.

----End

This is a common migration scheme. It is simple and easy to control on the service side. During the migration, the message sequence is ensured, so this scheme is **suitable for scenarios with strict requirements on the message sequence**. However, latency may occur because there is a period when you have to wait for all data to be consumed.



## Migration Scheme 2: Migrating the Production Later

Use multiple consumers for the consumption service. Some consume messages from the original Kafka instance, and others consume messages from the new Kafka instances. Then, migrate the production service to the new Kafka instance so that all messages can be consumed in time.

- Step 1** Start new consumer clients, set the Kafka connection addresses to that of the new Kafka instance, and consume data from the new Kafka instance.

 **NOTE**

Original consumer clients must continue running. Messages are consumed from both the original and new Kafka instances.

- Step 2** Change the Kafka connection address of the producer to that of the new Kafka instance.
- Step 3** Restart the producer client to migrate the production service to the new Kafka instance.
- Step 4** After the production service is migrated, check whether the consumption service connected to the new Kafka instance is normal.
- Step 5** After all data in the original Kafka is consumed, close the original consumption clients.
- Step 6** The migration is complete.

----End

In this scheme, the migration process is controlled by services. For a certain period of time, the consumption service consumes messages from both the original and new Kafka instances. Before the migration, message consumption from the new Kafka instance has already started, so there is no latency. However, early on in the migration, data is consumed from both the original and new Kafka instances, so the messages may not be consumed in the order that they are produced. This scheme is **suitable for services that require low latency but do not require strict message sequence**.

## How Do I Migrate Persisted Data Along with Services?

You can migrate consumed data from the original instance to a new instance by using the open-source tool [MirrorMaker](#). This tool mirrors the original Kafka producer and consumer into new ones and migrates data to the new Kafka instance.

Note that each cloud Kafka instance stores data in three replicas. Therefore, the storage space of the new instance should be three times that of the original single-replica message storage.

# 14 Applying for Increasing Kafka Quotas



---

## What Is a Quota?

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

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

## How Do I View My Quota?

1. Log in to the console.
2. 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 **Quotas** page is displayed.
4. On the **Quotas** page, view the used and total quotas of resources.  
If a quota cannot meet your needs, apply for a higher quota by performing the following operations.

## How Do I Increase My Quota?

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 and Alarms

---

## 15.1 Viewing Kafka Monitoring 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.

### Viewing Kafka Monitoring Metrics

**Step 1** Log in to the console.

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

 **NOTE**

Select the region where your Kafka instance is located.

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

**Step 4** View the instance metrics in either of the following ways:

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

----End

## 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, or view Kafka instance metrics on the **Monitoring** page of the DMS for Kafka console.

For example, you can call the [API](#) to query the monitoring data of the **Disk Capacity Usage** metric.

### Namespace

SYS.DMS

### Instance Metrics

**Table 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<br>Unit: count                                   | 0-1800          | Kafka instance   | 1 minute                     |
| current_topics     | Topics               | Number of created topics in the instance<br>Unit: count                                    | 0-1800          | Kafka instance   | 1 minute                     |
| group_msgs         | Accumulated Messages | Total number of accumulated messages in all consumer groups of the instance<br>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<br>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<br>Unit: count/s                                | 0–500,000           | Kafka instance broker | 1 minute                     |
| broker_bytes_out_rate        | Message Retrieval       | Number of bytes retrieved per second<br>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<br>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<br>Unit: byte/s, KB/s, MB/s, or GB/s  | 0–500,000,000       | Kafka instance broker | 1 minute                     |
| broker_public_bytes_out_rate | Public Outbound Traffic | Outbound traffic over public networks per second<br>Unit: byte/s, KB/s, MB/s, or GB/s | 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<br>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<br>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<br>Unit: %                              | 0-20        | Kafka instance broker | 1 minute                     |
| broker_disk_usage    | Disk Capacity Usage                           | Disk usage of the Kafka VM<br>Unit: %                                                 | 0-100       | Kafka instance broker | 1 minute                     |
| broker_memory_usage  | Memory Usage                                  | Memory usage of the Kafka VM<br>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<br>Unit: %                             | 0-100                                                                                | Kafka instance broker | 1 minute                     |
| broker_alive                | Broker Alive                   | Whether the Kafka broker is alive                                         | <ul style="list-style-type: none"> <li>• 1: alive</li> <li>• 0: not alive</li> </ul> | Kafka instance broker | 1 minute                     |
| broker_connections          | Connections                    | Total number of TCP connections on the Kafka broker<br>Unit: count        | 0-65,535                                                                             | Kafka instance broker | 1 minute                     |
| broker_cpu_usage            | CPU Usage                      | CPU usage of the Kafka VM<br>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<br>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<br>Unit: ms | > 0                                                                                  | Kafka instance broker | 1 minute                     |
| broker_total_bytes_in_rate  | Inbound Traffic                | Inbound traffic per second<br>Unit: byte/s                                | 0-1,000,000,000                                                                      | Kafka instance broker | 1 minute                     |
| broker_total_bytes_out_rate | Outbound Traffic               | Outbound traffic per second<br>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<br>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<br>Unit: byte/s, KB/s, MB/s, or GB/s | ≥ 0         | Kafka instance broker | 1 minute                     |
| network_bandwidth_usage | Network Bandwidth Usage | Network bandwidth usage<br>Unit: %                             | 0-100       | 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<br>Unit: byte/s, KB/s, MB/s, or GB/s<br><b>NOTE</b><br>This metric is available only when <b>Scope</b> is set to <b>Basic monitoring</b> on the <b>By Topic</b> 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_output_rate | Message Retrieval     | Number of bytes retrieved per second<br>Unit: byte/s, KB/s, MB/s, or GB/s<br><b>NOTE</b><br>This metric is available only when <b>Scope</b> is set to <b>Basic monitoring</b> on the <b>By Topic</b> 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<br>Unit: byte, KB, MB, GB, TB or PB<br><b>NOTE</b><br>This metric is available only when <b>Scope</b> is set to <b>Basic monitoring</b> on the <b>By Topic</b> 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<br>Unit: count<br><b>NOTE</b><br>This metric is available only when <b>Scope</b> is set to <b>Basic monitoring</b> on the <b>By Topic</b> tab page.                      | ≥ 0                 | Topic in a Kafka instance | 1 minute                     |
| topic_messages_in_rate  | Message Creation Rate | Number of messages created per second<br>Unit: count/s<br><b>NOTE</b><br>This metric is available only when <b>Scope</b> is set to <b>Basic monitoring</b> on the <b>By Topic</b> 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<br>Unit: count<br><b>NOTE</b><br>This metric is available only when <b>Scope</b> is set to <b>Partition monitoring</b> on the <b>By Topic</b> tab page. | $\geq 0$    | Topic in a Kafka instance | 1 minute                     |
| produced_messages  | Created Messages   | Number of messages that have been created<br>Unit: count<br><b>NOTE</b><br>This metric is available only when <b>Scope</b> is set to <b>Partition monitoring</b> on the <b>By Topic</b> tab page. | $\geq 0$    | Topic in a Kafka instance | 1 minute                     |

## Consumer Group Metrics

Table 15-4 Consumer group metrics

| Metric ID                  | Metric Name                | Description                                                                                                                                                                                                                                                                                                               | Value Range     | Monitor ed Object                   | Mo nitor ing Peri od (Ra w Dat a) |
|----------------------------|----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|-------------------------------------|-----------------------------------|
| messag es_cons umed        | Retriev ed Messag es       | Number of messages that have been retrieved in the consumer group<br>Unit: count<br><b>NOTE</b><br>This metric is available only when <b>Topic</b> is set to a specific topic name and <b>Monitoring Type</b> is set to <b>Partition monitoring</b> on the <b>By Consumer Group</b> tab page.                             | $\geq 0$        | Consum er group of a Kafka instance | 1 min ute                         |
| messag es_rem ained        | Availab le Messag es       | Number of messages that can be retrieved in the consumer group<br>Unit: count<br><b>NOTE</b><br>This metric is available only when <b>Topic</b> is set to a specific topic name and <b>Monitoring Type</b> is set to <b>Partition monitoring</b> on the <b>By Consumer Group</b> tab page.                                | $\geq 0$        | Consum er group of a Kafka instance | 1 min ute                         |
| topic_ messag es_rem ained | Topic Availab le Messag es | Number of remaining messages that can be retrieved from the specified topic in the consumer group<br>Unit: Count<br><b>NOTE</b><br>This metric is available only when <b>Topic</b> is set to a specific topic name and <b>Monitoring Type</b> is set to <b>Basic monitoring</b> on the <b>By Consumer Group</b> tab page. | 0 to $2^{63}-1$ | Consum er group of a Kafka instance | 1 min ute                         |

| Metric ID                   | Metric Name                                        | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | Value Range             | Monitored Object                   | Monitoring Period (Raw Data) |
|-----------------------------|----------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------|------------------------------------|------------------------------|
| topic_messages_consumed     | Topic Retrieved Messages                           | <p>Number of messages that have been retrieved from the specified topic in the consumer group</p> <p>Unit: Count</p> <p><b>NOTE</b><br/>This metric is available only when <b>Topic</b> is set to a specific topic name and <b>Monitoring Type</b> is set to <b>Basic monitoring</b> on the <b>By Consumer Group</b> tab page.</p>                                                                                                                                                                      | 0 to 2 <sup>63</sup> -1 | Consumer group of a Kafka instance | 1 minute                     |
| consumer_messages_remaining | Accumulated Messages (Consumer Available Messages) | <p>Number of remaining messages that can be retrieved in the consumer group</p> <p>Unit: Count</p> <p><b>NOTE</b></p> <ul style="list-style-type: none"> <li>On the Cloud Eye console, this metric is available only when <b>Queues</b> is set to <b>All queues</b> on the <b>Consumer Groups</b> tab page.</li> <li>On the <b>Monitoring</b> page of the DMS console, this metric is available only when <b>Topic</b> is set to <b>All topics</b> on the <b>By Consumer Group</b> tab page.</li> </ul> | 0 to 2 <sup>63</sup> -1 | Consumer group of a Kafka instance | 1 minute                     |

| Metric ID                  | Metric Name                 | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | Value Range             | Monitored Object                   | Monitoring Period (Raw Data) |
|----------------------------|-----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------|------------------------------------|------------------------------|
| consumer_messages_consumed | Consumer Retrieved Messages | <p>Number of messages that have been retrieved in the consumer group</p> <p>Unit: Count</p> <p><b>NOTE</b></p> <ul style="list-style-type: none"> <li>On the Cloud Eye console, this metric is available only when <b>Queues</b> is set to <b>All queues</b> on the <b>Consumer Groups</b> tab page.</li> <li>On the <b>Monitoring</b> page of the DMS console, this metric is available only when <b>Topic</b> is set to <b>All topics</b> on the <b>By Consumer Group</b> tab page.</li> </ul>               | 0 to 2 <sup>63</sup> -1 | Consumer group of a Kafka instance | 1 minute                     |
| messages_consumed_per_min  | Partition Consumption Rate  | <p>Number of messages consumed from the specified queue partition in the consumer group every minute</p> <p>Unit: count/minute</p> <p><b>NOTE</b></p> <ul style="list-style-type: none"> <li>This metric is available only when <b>Topic</b> is set to a specific topic name and <b>Monitoring Type</b> is set to <b>Partition monitoring</b> on the <b>By Consumer Group</b> tab page.</li> <li>Some instances do not support this metric. Check whether your instance supports it on the console.</li> </ul> | 0-30,000,000            | Consumer group of a Kafka instance | 1 minute                     |

| Metric ID                          | Metric Name                     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | Value Range  | Monitored Object                   | Monitoring Period (Raw Data) |
|------------------------------------|---------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------|------------------------------------|------------------------------|
| topic_messages_consumed_per_min    | Queue Consumption Rate          | <p>Number of messages consumed from the specified queue in the consumer group every minute</p> <p>Unit: count/minute</p> <p><b>NOTE</b></p> <ul style="list-style-type: none"> <li>This metric is available only when <b>Topic</b> is set to a specific topic name and <b>Monitoring Type</b> is set to <b>Basic monitoring</b> on the <b>By Consumer Group</b> tab page.</li> <li>Some instances do not support this metric. Check whether your instance supports it on the console.</li> </ul> | 0–30,000,000 | Consumer group of a Kafka instance | 1 minute                     |
| consumer_messages_consumed_per_min | Consumer Group Consumption Rate | <p>Number of messages consumed from the consumer group every minute</p> <p>Unit: count/minute</p> <p><b>NOTE</b></p> <ul style="list-style-type: none"> <li>This metric is available only when <b>Topic</b> is set to <b>All topics</b> on the <b>By Consumer Group</b> tab page.</li> <li>Some instances do not support this metric. Check whether your instance supports it on the console.</li> </ul>                                                                                         | 0–30,000,000 | Consumer group of a Kafka instance | 1 minute                     |

## Dimension

| Key               | Value                 |
|-------------------|-----------------------|
| kafka_instance_id | Kafka instance        |
| kafka_broker      | Kafka instance broker |
| kafka_topics      | Kafka instance topic  |

| Key                     | Value                                        |
|-------------------------|----------------------------------------------|
| 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 a Kafka Alarm Rule

This section describes the alarm rules of some metrics and how to configure them. In actual services, you are advised to configure alarm rules for metrics based on the following alarm policies:

**Table 15-5** Kafka instance metrics to configure alarm rules for

| Metric ID            | Metric                    | Alarm Policy                                                                                          | Description                                    | Handling Suggestion                                                                                                                                                                                                                          |
|----------------------|---------------------------|-------------------------------------------------------------------------------------------------------|------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| broker_disk_usage    | Disk Capacity Usage       | Alarm threshold: original value > 80%<br>Number of consecutive periods: 1<br>Alarm severity: critical | Disk usage of the Kafka VM                     | Modify the instance <b>storage space</b> . For details, see <a href="#">Modifying Instance Specifications</a> .                                                                                                                              |
| broker_cpu_core_load | Average Load per CPU Core | Alarm threshold: original value > 2<br>Number of consecutive periods: 3<br>Alarm severity: major      | Average load of each CPU core of the Kafka VM. | Check whether the metric has been approaching or exceeding the alarm threshold for a long time. If yes, modify the instance <b>bandwidth or the number of brokers</b> . For details, see <a href="#">Modifying Instance Specifications</a> . |




| Metric ID           | Metric       | Alarm Policy                                                                                                                                                                                                                                                                        | Description                                | Handling Suggestion                                                                                                                                                                                                                                                              |
|---------------------|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| broker_memory_usage | Memory Usage | <p>Alarm threshold: original value &gt; 90%</p> <p>Number of consecutive periods: 3</p> <p>Alarm severity: critical</p>                                                                                                                                                             | Memory usage of the Kafka VM.              | Modify the instance <b>bandwidth or the number of brokers</b> . For details, see <a href="#">Modifying Instance Specifications</a> .                                                                                                                                             |
| current_partitions  | Partitions   | <p>Alarm threshold: original value &gt; 90% of the maximum allowed number of partitions. The partition limit varies depending on instance specifications. For details, see <a href="#">Specifications</a>.</p> <p>Number of consecutive periods: 1</p> <p>Alarm severity: major</p> | Number of used partitions in the instance. | If new topics are required, modify the instance <b>bandwidth or the number of brokers</b> , or split the service to multiple instances. For details about how to modify the instance bandwidth or the number of brokers, see <a href="#">Modifying Instance Specifications</a> . |
| broker_cpu_usage    | CPU Usage    | <p>Alarm threshold: original value &gt; 90%</p> <p>Number of consecutive periods: 3</p> <p>Alarm severity: major</p>                                                                                                                                                                | CPU usage of the Kafka VM.                 | Check whether the metric has been approaching or exceeding the alarm threshold for a long time. If yes, modify the instance <b>bandwidth or the number of brokers</b> . For details, see <a href="#">Modifying Instance Specifications</a> .                                     |

| Metric ID             | Metric                   | Alarm Policy                                                                                                                                          | Description                                                                                        | Handling Suggestion                                                                                                                                                                                                                                                                                                        |
|-----------------------|--------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| group_msgs            | Accumulated Messages     | Alarm threshold: original value > 90% of the upper limit. The upper limit is customized.<br>Number of consecutive periods: 1<br>Alarm severity: major | Total number of accumulated messages in all consumer groups of the instance                        | Delete idle consumer groups, if any. You can also accelerate message retrieval, for example, by increasing the number of consumers.                                                                                                                                                                                        |
| topic_messages_remain | Topic Available Messages | Alarm threshold: original value > 90% of the upper limit. The upper limit is customized.<br>Number of consecutive periods: 1<br>Alarm severity: major | Number of remaining messages that can be retrieved from the specified topic in the consumer group. | Check whether the consumer code logic is correct, for example, by checking whether the consumer stops consuming messages due to an exception. You can also accelerate message retrieval, for example, by adding topic consumers. Ensure that the number of partitions is greater than or equal to the number of consumers. |

## Configuring a Kafka Alarm Rule

**Step 1** Log in to the console.

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


 **NOTE**

Select the region where your Kafka instance is located.

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

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

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

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

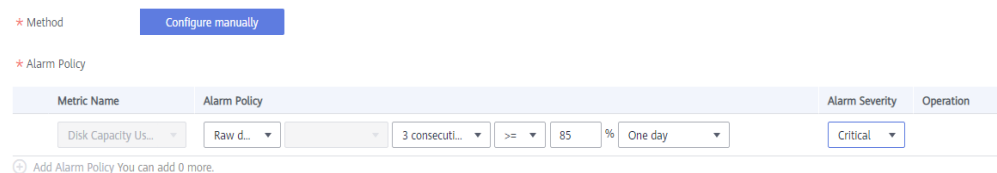
**Step 6** Specify the alarm details.

For more information about creating alarm rules, see [Creating an Alarm Rule](#).

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

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

**Figure 15-1** Setting the alarm policy and alarm severity



\* Method Configure manually

\* Alarm Policy

| Metric Name         | Alarm Policy                             | Alarm Severity | Operation |
|---------------------|------------------------------------------|----------------|-----------|
| Disk Capacity Us... | Raw d...<br>3 consecuti...<br>>=<br>85 % | One day        | Critical  |

⊖ 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 Viewing Kafka Audit Logs

## Scenario

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

## Prerequisite

CTS has been enabled.

## DMS for Kafka Operations Supported by CTS

**Table 16-1** DMS for Kafka operations that can be recorded by CTS

| Operation                                                   | Resource Type | Trace Name                             |
|-------------------------------------------------------------|---------------|----------------------------------------|
| Successfully creating an instance                           | kafka         | createDMSInstanceTaskSuccess           |
| Failing to create an instance                               | kafka         | createDMSInstanceTaskFailure           |
| Successfully deleting an instance that failed to be created | kafka         | deleteDMSCreateFailureInstancesSuccess |
| Failing to delete an instance that failed to be created     | kafka         | deleteDMSCreateFailureInstancesFailure |
| Successfully deleting an instance                           | kafka         | deleteDMSInstanceTaskSuccess           |
| Failing to delete an instance                               | kafka         | deleteDMSInstanceTaskFailure           |
| Deleting multiple instance tasks at a time                  | kafka         | batchDeleteDMSInstanceTask             |

| Operation                                                                | Resource Type | Trace Name                         |
|--------------------------------------------------------------------------|---------------|------------------------------------|
| Successfully submitting a request to delete multiple instances at a time | kafka         | batchDeleteDMSInstanceSuccess      |
| Successfully deleting multiple instances at a time                       | kafka         | batchDeleteDMSInstanceTask-Success |
| Failing to submit a request to delete multiple instances at a time       | kafka         | batchDeleteDMSInstanceFailure      |
| Failing to delete multiple instances at a time                           | kafka         | batchDeleteDMSInstanceTask-Failure |
| Successfully submitting a request to scale up an instance                | kafka         | extendDMSInstanceSuccess           |
| Successfully scaling up an instance                                      | kafka         | extendDMSInstanceTaskSuccess       |
| Failing to submit a request to scale up an instance                      | kafka         | extendDMSInstanceFailure           |
| Failing to scale up an instance                                          | kafka         | extendDMSInstanceTaskFailure       |
| Successfully submitting a request to reset instance password             | kafka         | resetDMSInstancePasswordSuccess    |
| Failing to submit a request to reset instance password                   | kafka         | resetDMSInstancePasswordFailure    |
| Successfully submitting a request to restart an instance                 | kafka         | restartDMSInstanceSuccess          |
| Successfully restarting an instance                                      | kafka         | restartDMSInstanceTaskSuccess      |
| Failing to submit a request to restart an instance                       | kafka         | restartDMSInstanceFailure          |

| Operation                                                                 | Resource Type | Trace Name                         |
|---------------------------------------------------------------------------|---------------|------------------------------------|
| Failing to restart an instance                                            | kafka         | restartDMSInstanceTaskFailure      |
| Successfully submitting a request to restart multiple instances at a time | kafka         | batchRestartDMSInstanceSuccess     |
| Successfully restarting multiple instances at a time                      | kafka         | batchRestartDMSInstanceTaskSuccess |
| Failing to submit a request to restart multiple instances at a time       | kafka         | batchRestartDMSInstanceFailure     |
| Failing to restart multiple instances at a time                           | kafka         | batchRestartDMSInstanceTaskFailure |
| Successfully submitting a request to modify instance information          | kafka         | modifyDMSInstanceInfoSuccess       |
| Successfully modifying instance information                               | kafka         | modifyDMSInstanceInfoTaskSuccess   |
| Failing to submit a request to modify instance information                | kafka         | modifyDMSInstanceInfoFailure       |
| Failing to modify instance information                                    | kafka         | modifyDMSInstanceInfoTaskFailure   |
| Successfully deleting a background task                                   | kafka         | deleteDMSBackendJobSuccess         |
| Failing to delete a background task                                       | kafka         | deleteDMSBackendJobFailure         |
| Successfully creating a topic for a Kafka instance                        | kafka         | Kafka_create_topicSuccess          |
| Failing to create a topic for a Kafka instance                            | kafka         | Kafka_create_topicFailure          |
| Successfully deleting a topic from a Kafka instance                       | kafka         | Kafka_delete_topicsSuccess         |

| Operation                                      | Resource Type | Trace Name                         |
|------------------------------------------------|---------------|------------------------------------|
| Failing to delete a topic for a Kafka instance | kafka         | Kafka_delete_topicsFailure         |
| Successfully enabling automatic topic creation | kafka         | enable_auto_topicSuccess           |
| Failing to enable automatic topic creation     | kafka         | enable_auto_topicFailure           |
| Successfully 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      |

## Viewing Audit Logs

See [Querying Real-Time Traces](#).

# 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 [Viewing or Modifying User Group Information](#).

### 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 the disk, 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.

For details about how to create a Kafka instance, see [Buying a Kafka Instance](#).

### 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:

- Use port 9092 for accessing a Kafka instance without SASL.
- Use port 9093 for accessing a Kafka instance with SASL.

Ensure that proper rules have been configured for the security group of the instance. For details, see [How Do I Select and Configure a Security Group?](#)

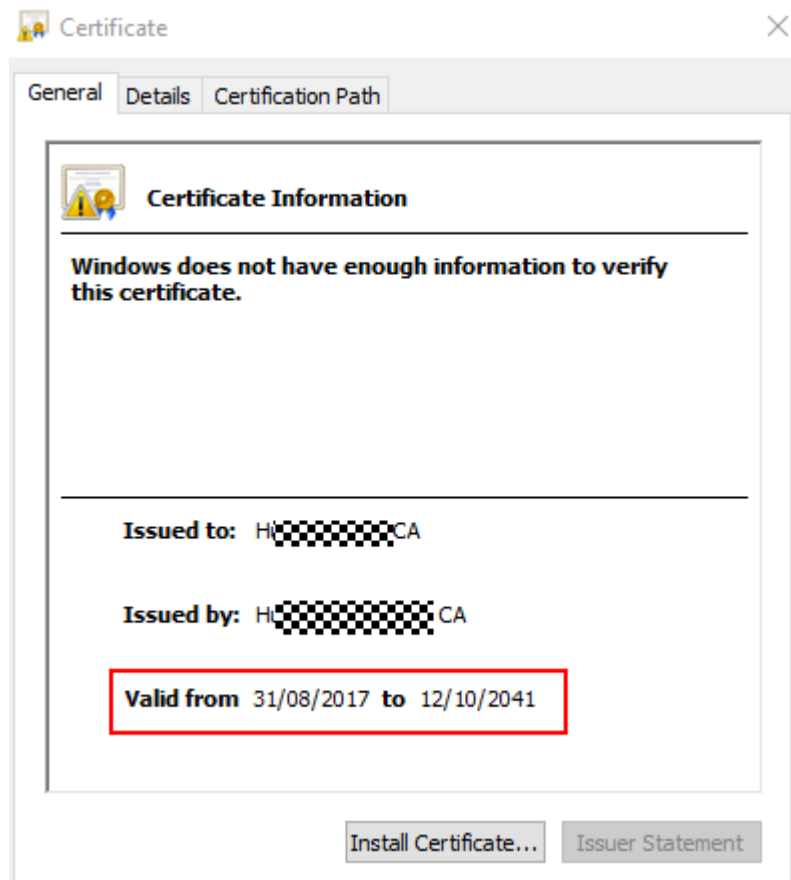
## 17.1.10 How Long Are Kafka SSL Certificates Valid for?

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.

To check the validity of the SSL certificate, perform the following steps:

- Step 1** Decompress the package downloaded from the Kafka instance console to obtain `phy_ca.crt`.
- Step 2** Double-click `phy_ca.crt`. The **Certificate** dialog box is displayed.
- Step 3** On the **General** tab page, view the certificate validity period.

**Figure 17-1** Certificate validity period



----End

### **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 17-2** 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?

No. Kafka instances do not support disk encryption.

### 17.1.20 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.

### 17.1.21 Where Can I Find Kafka Streams Use Cases?

You can find Kafka Streams use cases on the [official Kafka website](#).

### 17.1.22 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.

### 17.1.23 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.

## 17.2 Specification Modification

### 17.2.1 Does Specification Modification Affect Services?

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

## 17.2.2 Will Data Migration Be Involved When I Increase Specifications?

No. Data will not be migrated when you increase specifications.

## 17.2.3 What Can I Do When I Fail to Increase Specifications Due to Insufficient Resources?

**Symptom:** Specifications fail to be increased, and a message is displayed indicating that the underlying ECS/EVS resources are insufficient. However, the required ECSs can be purchased on the ECS console.

**Possible cause:** The underlying resource quota is different from the available flavor quota displayed on the console.

**Solution:** Contact customer service to increase the quota.

## 17.3 Connections

### 17.3.1 How Do I Select and Configure a Security Group?

Kafka instances can be accessed within a VPC. 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.

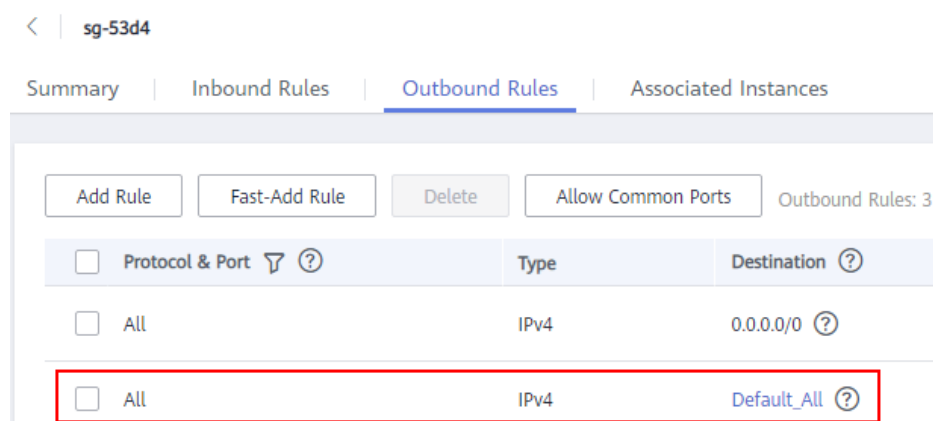
Assume that the security groups of the client and Kafka instance are **sg-53d4** and **Default\_All**, respectively. You can specify a security group or IP address as the destination in the following rule. A security group is used as an example.

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 17-3** Configuring a security group for the client

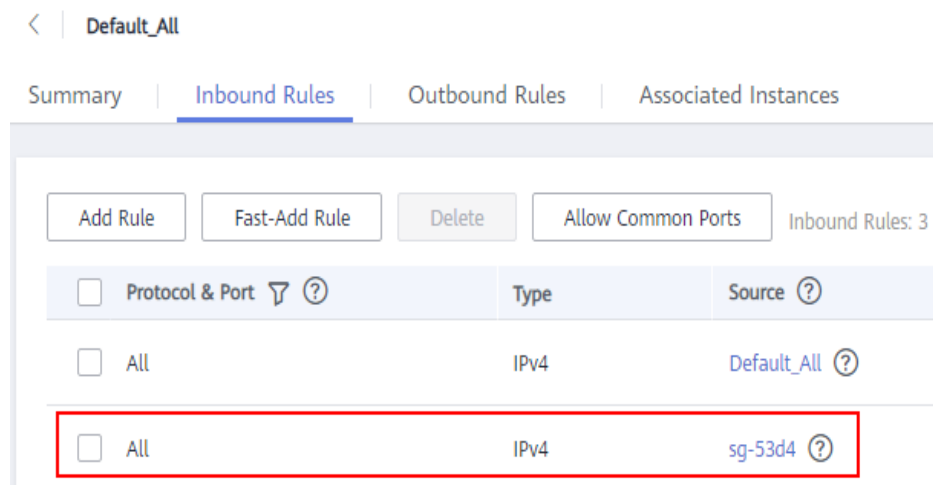


To ensure that your client can access the Kafka instance, add the following rule to the security group configured for the instance.

**Table 17-3** Security group rule

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

**Figure 17-4** Configuring the security group for the Kafka instance



----End

### 17.3.2 Can I Access a Kafka Instance Over a Public Network?

No. To ensure security, the ECS that serves as a client and the Kafka instance that the client will access must belong to the same VPC.

### 17.3.3 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 17-4** Kafka instance specifications (cluster)

| Flavor                     | Bro<br>kers | Maxi<br>mum<br>TPS<br>per<br>Broke<br>r | Maxi<br>mum<br>Parti<br>tions<br>per<br>Broke<br>r | Reco<br>mme<br>nded<br>Cons<br>umer<br>Grou<br>ps<br>per<br>Broke<br>r | Maximu<br>m<br>Client<br>Connect<br>ions per<br>Broker | Storage<br>Space      | Traffic per<br>Broker<br>(MB/s) |
|----------------------------|-------------|-----------------------------------------|----------------------------------------------------|------------------------------------------------------------------------|--------------------------------------------------------|-----------------------|---------------------------------|
| kafka.2u<br>4g.cluste<br>r | 3–<br>30    | 30,00<br>0                              | 250                                                | 20                                                                     | 2000                                                   | 300 GB–<br>300,000 GB | 100                             |
| kafka.4u<br>8g.cluste<br>r | 3–<br>30    | 100,0<br>00                             | 500                                                | 100                                                                    | 4000                                                   | 300 GB–<br>600,000 GB | 200                             |

| Flavor               | Brokers | Maximum TPS per Broker | Maximum Partitions per Broker | Recommended Consumer Groups per Broker | Maximum Client Connections per Broker | Storage Space       | Traffic per Broker (MB/s) |
|----------------------|---------|------------------------|-------------------------------|----------------------------------------|---------------------------------------|---------------------|---------------------------|
| kafka.8u16g.cluster  | 3-50    | 150,000                | 1000                          | 150                                    | 4000                                  | 300 GB-1,500,000 GB | 375                       |
| kafka.12u24g.cluster | 3-50    | 200,000                | 1500                          | 200                                    | 4000                                  | 300 GB-1,500,000 GB | 625                       |
| kafka.16u32g.cluster | 3-50    | 250,000                | 2000                          | 200                                    | 4000                                  | 300 GB-1,500,000 GB | 750                       |

### 17.3.4 Do Kafka Instances Support Cross-Region Access?

Yes. You can access a Kafka instance across regions by using direct connections.

### 17.3.5 Do Kafka Instances Support Cross-VPC Access?

Yes. You can 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](#).

### 17.3.6 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.

### 17.3.7 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 [Connecting to Kafka Using the Client \(SASL Enabled\)](#).



### 17.3.8 Does DMS for Kafka Support Password-Free Access?

Yes. No password is required for accessing a Kafka instance with SASL disabled. For details, see [Connecting to Kafka Using the Client \(SASL Disabled\)](#).

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

No.

### 17.3.10 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.

### 17.3.11 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.

### 17.3.12 Which TLS Version Does DMS for Kafka Support?

TLS 1.2.

### 17.3.13 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.

**Table 17-5** Number of connections of earlier Kafka instances

| Assured Bandwidth | Max. Connections |
|-------------------|------------------|
| 100 MB/s          | 3000             |
| 300 MB/s          | 10,000           |
| 600 MB/s          | 20,000           |
| 1200 MB/s         | 20,000           |

**Table 17-6** Number of connections of later Kafka instances

| Flavor             | Max. Connections |
|--------------------|------------------|
| kafka.2u4g.cluster | 2000             |
| kafka.4u8g.cluster | 4000             |

| Flavor               | Max. Connections |
|----------------------|------------------|
| kafka.8u16g.cluster  | 4000             |
| kafka.12u24g.cluster | 4000             |
| kafka.16u32g.cluster | 4000             |

### 17.3.14 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 Instance Configuration Parameters](#).

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

No, and you cannot specify the IP addresses.

### 17.3.16 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

### 17.3.17 Why Is It Not Recommended to Use a Sarama Client for Messaging?

#### Symptom

If a Sarama client is used to send and receive messages, the following issues may occur:

- Sarama cannot detect partition changes. Adding topic partitions requires client restart to enable consumption.
- Sarama's default **MaxProcessingTime** is 100 ms. When this limit is reached, consumers can no longer consume messages.
- If consumer offsets reset from the oldest (earliest) position, all messages starting from the earliest position may be repeatedly consumed after the client restarts.

- A consumer that subscribes to multiple topics may not be able to consume any message from specific partitions.

## Solution

Use `confluent-kafka-go` as the Kafka client library.

For details, see [Table 17-7](#).

**Table 17-7** Comparing common Go clients

| Client                          | Pros                                                                                                                                                                                                                                             | Cons                                                                                                                                                                                                                         |
|---------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>confluent-kafka-go</code> | <ul style="list-style-type: none"><li>• An official Kafka client by Confluent that supports full Kafka compatibility and all Kafka features</li><li>• High stability and performance, and low latency based on <code>librdkafka</code></li></ul> | High compiling complexity because Go compilers need extra resources to configure the imported C++ libraries                                                                                                                  |
| <code>kafka-go</code>           | <ul style="list-style-type: none"><li>• A simple and lite Kafka client easy for learning and usage</li><li>• Reduced application size and complexity with limited library and fewer dependencies</li></ul>                                       | <ul style="list-style-type: none"><li>• Fewer advanced functions and configurations than <code>confluent-kafka-go</code></li><li>• Applicable only to simple scenarios that require low performance and throughput</li></ul> |
| Sarama                          | Better asynchronization and higher concurrency (written in the original Go language)                                                                                                                                                             | <ul style="list-style-type: none"><li>• Many bugs, limited documentation</li><li>• Deteriorates application performance when processing a large number of messages due to large memory consumption</li></ul>                 |

## 17.4 Topics and Partitions

### 17.4.1 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 17-8** Kafka instance specifications (cluster)

| Flavor               | Brokers | Maximum TPS per Broker | Maximum Partitions per Broker | Recommended Consumer Groups per Broker | Maximum Client Connections per Broker | Storage Space       | Traffic per Broker (MB/s) |
|----------------------|---------|------------------------|-------------------------------|----------------------------------------|---------------------------------------|---------------------|---------------------------|
| kafka.2u4g.cluster   | 3-30    | 30,000                 | 250                           | 20                                     | 2000                                  | 300 GB-300,000 GB   | 100                       |
| kafka.4u8g.cluster   | 3-30    | 100,000                | 500                           | 100                                    | 4000                                  | 300 GB-600,000 GB   | 200                       |
| kafka.8u16g.cluster  | 3-50    | 150,000                | 1000                          | 150                                    | 4000                                  | 300 GB-1,500,000 GB | 375                       |
| kafka.12u24g.cluster | 3-50    | 200,000                | 1500                          | 200                                    | 4000                                  | 300 GB-1,500,000 GB | 625                       |
| kafka.16u32g.cluster | 3-50    | 250,000                | 2000                          | 200                                    | 4000                                  | 300 GB-1,500,000 GB | 750                       |

## 17.4.2 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 17-9** Kafka instance specifications (cluster)

| Flavor                       | Bro<br>kers | Maxi<br>mum<br>TPS<br>per<br>Broke<br>r | Maxi<br>mum<br>Parti<br>tions<br>per<br>Brok<br>er | Reco<br>mme<br>nded<br>Cons<br>umer<br>Grou<br>ps<br>per<br>Broke<br>r | Maximu<br>m<br>Client<br>Connect<br>ions per<br>Broker | Storage<br>Space           | Traffic per<br>Broker<br>(MB/s) |
|------------------------------|-------------|-----------------------------------------|----------------------------------------------------|------------------------------------------------------------------------|--------------------------------------------------------|----------------------------|---------------------------------|
| kafka.2u<br>4g.cluste<br>r   | 3–<br>30    | 30,00<br>0                              | 250                                                | 20                                                                     | 2000                                                   | 300 GB–<br>300,000 GB      | 100                             |
| kafka.4u<br>8g.cluste<br>r   | 3–<br>30    | 100,0<br>00                             | 500                                                | 100                                                                    | 4000                                                   | 300 GB–<br>600,000 GB      | 200                             |
| kafka.8u<br>16g.clust<br>er  | 3–<br>50    | 150,0<br>00                             | 1000                                               | 150                                                                    | 4000                                                   | 300 GB–<br>1,500,000<br>GB | 375                             |
| kafka.12<br>u24g.clu<br>ster | 3–<br>50    | 200,0<br>00                             | 1500                                               | 200                                                                    | 4000                                                   | 300 GB–<br>1,500,000<br>GB | 625                             |
| kafka.16<br>u32g.clu<br>ster | 3–<br>50    | 250,0<br>00                             | 2000                                               | 200                                                                    | 4000                                                   | 300 GB–<br>1,500,000<br>GB | 750                             |

### 17.4.3 Can I Reduce 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.

### 17.4.4 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 by instance specifications. For details, see [Specifications](#).

Solution: Scale up the instance or delete unnecessary topics.

### 17.4.5 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.

### 17.4.6 Why Do Deleted Topics Still Exist?

**Possible cause:** 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.

**Solution:** Disable automatic topic creation.

### 17.4.7 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 **View Metric** next to the Kafka instance name to go to the Cloud Eye console. On the **Topics** tab page, set **Topic** 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.

### 17.4.8 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 Kafka Topic Permissions](#).

### 17.4.9 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.

### 17.4.10 How Do I Increase the Partition Quantity?

You can increase the partition quantity by increasing the bandwidth or 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 or add brokers as required. For details, see [Modifying Instance Specifications](#).

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

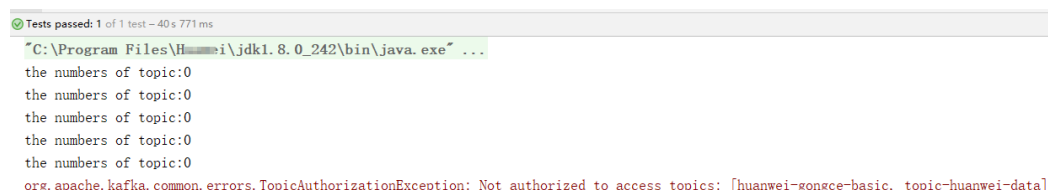
Yes. A Kafka instance will be restarted if you enable or disable automatic topic creation for it.

## 17.4.12 Can I Delete Unnecessary Topics in a Consumer Group?

Just simply unsubscribe from them on the Kafka client.

## 17.4.13 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.

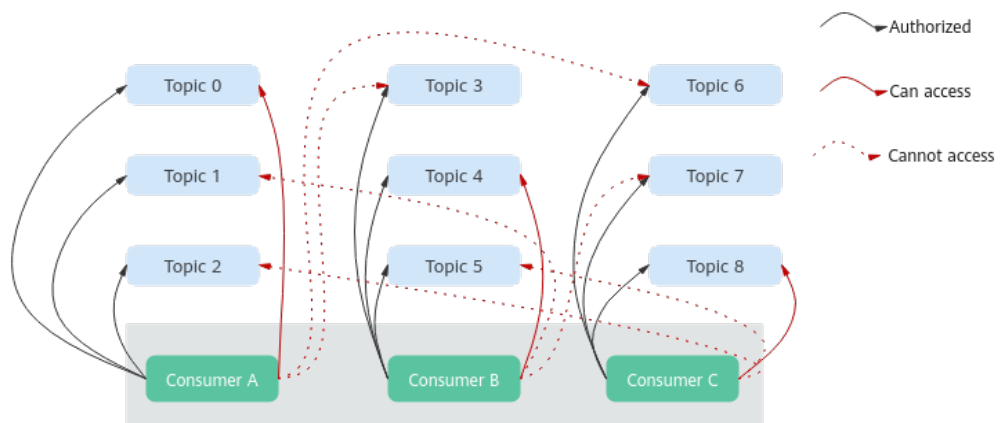


```
Tests passed: 1 of 1 test - 40s 771ms
"C:\Program Files\H...i\jdk1.8.0_242\bin\java.exe" ...
the numbers of topic:0
the numbers of topic:0
the numbers of topic:0
the numbers of topic:0
the numbers of topic:0
the numbers of topic:0
org.apache.kafka.common.errors.TopicAuthorizationException: Not authorized 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 17-5 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.

### 17.4.14 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  entries

| Topic                           | # Partitions | # Brokers | Brokers Spread % |
|---------------------------------|--------------|-----------|------------------|
| <code>__consumer_offsets</code> | 50           | 3         | 100              |
| <code>__trace</code>            | 9            | 3         | 100              |
| <code>topic-01</code>           | 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.



## 17.5 Consumer Groups

### 17.5.1 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 [Connecting to an Instance](#).

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

This depends on the `offsets.retention.minutes` and `auto.create.groups.enable` parameters.

- For instances created before Apr 25, 2023, `auto.create.groups.enable` is set to `true` by default. `offsets.retention.minutes` determines how long before a consumer group is deleted automatically, which can be changed on the console. For details, see [Modifying Kafka Instance Configuration Parameters](#).
- For instances created on or after Apr 25, 2023:
  - If `auto.create.groups.enable` is `false`, you need to manually delete consumer groups.

#### NOTE

If `auto.create.groups.enable` is set to `false` for a Kafka instance created before Feb 5, 2024, and an empty consumer group subscribed to only one topic, deleting the topic also removes the consumer group.

- If `auto.create.groups.enable` is `true`, a consumer group that has never committed an offset will be automatically deleted after 10 minutes.
- If `auto.create.groups.enable` is `true`, and a consumer group that has committed an offset, `offsets.retention.minutes` determines how long before the group is deleted automatically, which can be changed on the console. For details, see [Modifying Kafka Instance Configuration Parameters](#).

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.

### 17.5.3 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.

## 17.5.4 Why Can't I View Consumers When Instance Consumption Is Normal?

Check whether Flink is used for consumption. Flink uses the assign mode and the client assigns specific partitions to be consumed, so you cannot see any consumer on the Kafka console.

## 17.6 Messages

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

10 MB.

### 17.6.2 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**.

### 17.6.3 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.

### 17.6.4 What Can I Do If Kafka Messages Are Accumulated?

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

**Solution:**

1. Log in to the Kafka console and click the instance for which the alarm is generated. The instance details page is displayed.
2. In the navigation pane, choose **Monitoring**.
3. On the **By Consumer Group** tab page, view **Consumer Retrieved Messages** to find the consumer group with accumulated messages.

4. In the navigation pane, choose **Consumer Groups**.
5. Check whether there are consumers in the consumer group where messages are accumulated. If yes, contact the service party to accelerate their consumption. If no, contact the customer to delete unused consumer groups.

## 17.6.5 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.

## 17.6.6 Do Kafka Instances Support Delayed Message Delivery?

No.

## 17.6.7 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 the Kafka Consumer Group List](#).
- 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 Kafka Monitoring 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 17-6 Viewing the total number of messages accumulated in each topic

```

[root@dm-vn-2a65fbf1-kafka-share-zk-server-2 bin]# ./kafka-consumer-groups.sh --bootstrap-server 172.11.1.111:9092 --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-54209-1-888da1b5-4664-4f52-836a-731059011e0d | 172.31.7.137 | consumer-console-consumer-54209-1 |
| console-consumer-54209 | test2 | 1         | -              | 3              | -   | consumer-console-consumer-54209-1-888da1b5-4664-4f52-836a-731059011e0d | 172.31.7.137 | consumer-console-consumer-54209-1 |
| console-consumer-54209 | test2 | 2         | -              | 3              | -   | consumer-console-consumer-54209-1-888da1b5-4664-4f52-836a-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](#).

## 17.6.8 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.

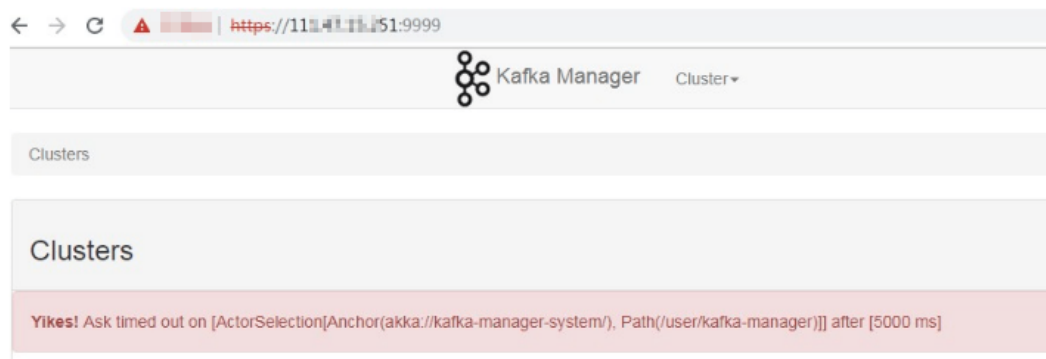
## 17.7 Kafka Manager

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

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

### 17.7.2 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.



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

### 17.7.3 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.

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

No. Kafka Manager does not support message body querying.

### 17.7.5 Can I Change the Port of the Kafka Manager Web UI?

No.

### 17.7.6 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**.



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

← topic-619913173

| 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   |

Operations

Delete Topic
Reassign Partitions
Generate Partition Assignments

Add Partitions
Update Config
Manual Partition Assignments

Partitions by Broker

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

### 17.7.7 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**.

### 17.7.8 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  entries

| Topic                             | # Partitions | # Brokers | Brokers Spread % | Brokers Skew % | Brokers Leader Skew % |
|-----------------------------------|--------------|-----------|------------------|----------------|-----------------------|
| <a href="#">_consumer_offsets</a> | 50           | 3         | 100              | 0              | 0                     |
| <a href="#">_trace</a>            | 9            | 3         | 100              | 0              | 0                     |
| <a href="#">topic-test</a>        | 3            | 3         | 100              | 0              | 0                     |

Showing 1 to 3 of 3 entries

- Click **Manual Partition Assignments**.

Figure 17-7 Topic details

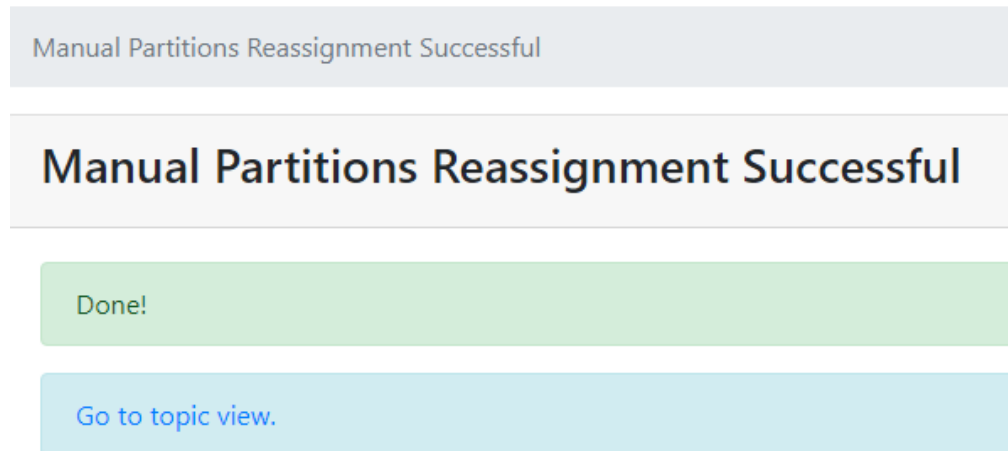
Figure 17-8 Page for modifying partition leaders

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

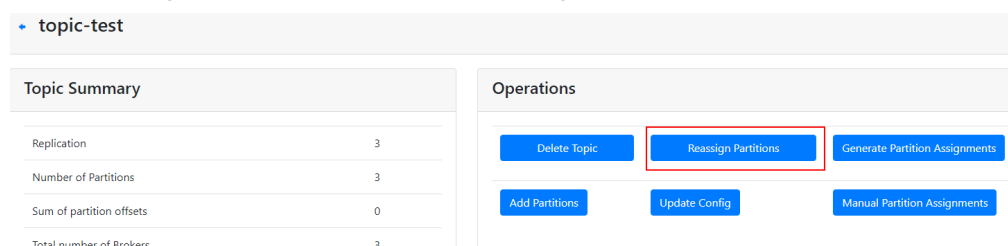
- Change the leader and click **Save Partition Assignment**.



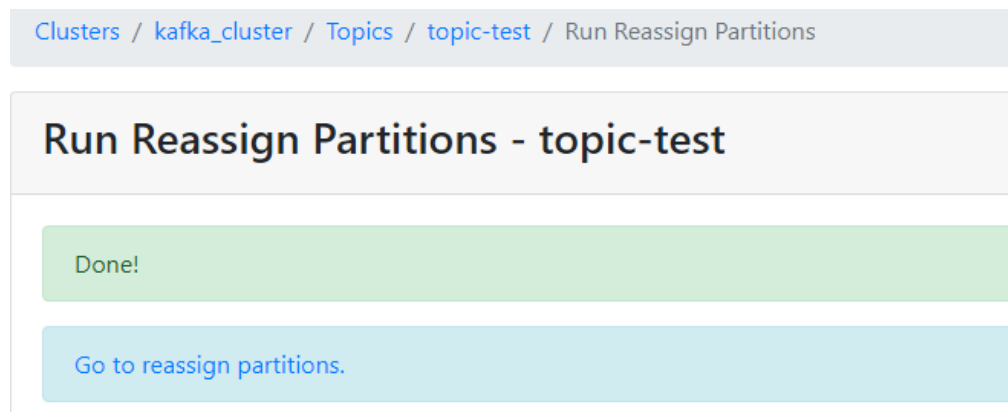
If the modification is successful, the information shown in the following figure is displayed.



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



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



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.

## 17.8 Monitoring & Alarm

### 17.8.1 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.

### 17.8.2 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.

### **17.8.3 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.Alert.getSSLException(Alert.java:192)
at sun.security.ssl.SSLEngineImpl.fatal(SSLEngineImpl.java:1789)
at sun.security.ssl.Handshaker.fatalISE(Handshaker.java:318)
at sun.security.ssl.Handshaker.fatalISE(Handshaker.java:318)
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:1837)
at sun.security.ssl.Handshaker$1.run(Handshaker.java:978)
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:482)
at org.apache.kafka.common.network.SslTransportLayer.handshakeInwrap(SslTransportLayer.java:484)
at org.apache.kafka.common.network.SslTransportLayer.doHandshake(SslTransportLayer.java:348)
... 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:

```
[2020-05-28T00:35:30.054] [ERROR] [[logstash.outputs.kafka] Unable to create Kafka producer from given configuration [-kafka_error_message=org.apache.kafka.common.KafkaException: Failed to construct kafka producer, :cause=org.apache.kafka.common.KafkaException: org.apache.kafka.common.KafkaException: Failed to load SSL keystore /opt/rf/cloud/logstash/pipeline/wm-logstash-cn-north-4/client.truststore.jks of type JKS]
```

**Solution:**

- Check whether the **client.truststore.jks** file exists in the corresponding address.
- Check the permissions on the processes and files.
- 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_thread0] [impl.KafkaClientImpl:207]ready poll_topic is CSBPromotionManagementService_PromotionTopic
020-05-11 01:11:23.704 INFO [eventpull_thread0] [kafka.PullJobDetail:171] pull event from kafka cost time 200, topic CSBPromotionManagementService_PromotionTopic,eventlist []
020-05-11 01:11:24.029 ERROR [PublishEventToKafka_Thread] [impl.KafkaClientImpl:200] send event to kafka failed, topic=[CSBPromotionCouponService_CouponTopic], eventId = [01700-99999-0000000100000-0] ex=
org.apache.kafka.common.errors.TimeoutException: Topic =CSBPromotionCouponService_CouponTopic not present in metadata after 60000 ms.
020-05-11 01:11:24.217 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 0
020-05-11 01:11:24.724 INFO [pool-20-thread-1] [impl.KafkaClientImpl:112] end.getTopicList
020-05-11 01:11:24.900 INFO [eventpull_thread0a] [impl.KafkaClientImpl:207]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

- 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.
- The EIP inbound traffic decreases.  
If the EIP technical support personnel cannot find any problem, this is not the cause of the symptom.
- 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.

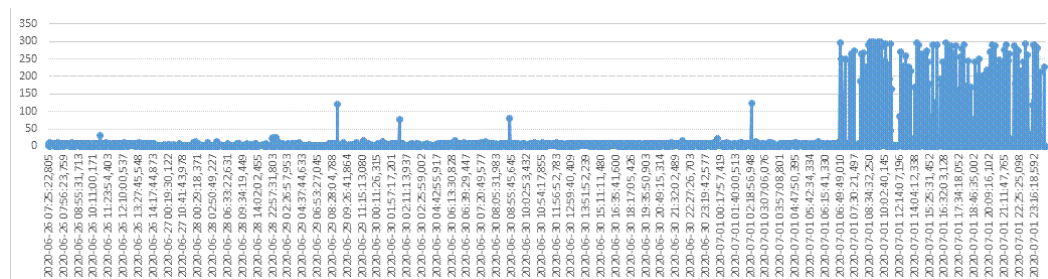
## Detailed Analysis

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 **REBLANCING**.
- Stabilized group**  
 The consumer group completes rebalance, and its status changes to **STABILIZED**.
- Member consumer-xxx 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

- Use different consumer groups for different services to reduce the impact of a single consumer blocking access.
- 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://192.168.1.9:9999> 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: f21aa116b661c6a (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: t0R4RgFHTN2pjUhiJqkFPQ (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 15:46:52,500] 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                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 2024-05-28 | <p>This release incorporates the following changes:</p> <ul style="list-style-type: none"><li>• Messages can be queried by keywords. See <a href="#">Viewing Kafka Messages</a>.</li><li>• Partition reassignment can be scheduled. See <a href="#">Reassigning Kafka Partitions</a>.</li><li>• Increased the maximum number of users that can be created to 500. See <a href="#">Configuring Kafka Users</a>.</li><li>• Added consumer group creation, message production, and message deletion on the console. See <a href="#">Creating a Kafka Consumer Group</a>, <a href="#">Connecting to Kafka on the Console</a>, and <a href="#">Deleting Kafka Messages</a>.</li><li>• Changed the upper limit of the topic retention period to 720 hours. See <a href="#">Creating a Kafka Topic</a>.</li></ul>                                                                                                                                                                                                 |
| 2023-05-09 | <p>This release incorporates the following changes:</p> <ul style="list-style-type: none"><li>• Added <a href="#">Getting Started</a>.</li><li>• Added sections <a href="#">Viewing Kafka Disk Usage</a>, <a href="#">Reassigning Kafka Partitions</a>, <a href="#">Viewing and Resetting Kafka Consumption Offsets</a>, and <a href="#">Managing Quotas</a>.</li><li>• Modified the operation description on the instance details page in sections <a href="#">Restarting a Kafka Instance</a>, <a href="#">Deleting Kafka Instances</a>, and <a href="#">Connecting to Kafka Using the Client (SASL Disabled)</a>.</li><li>• Added support for the SASL mechanism in sections <a href="#">Buying a Kafka Instance</a> and <a href="#">Connecting to Kafka Using the Client (SASL Enabled)</a>.</li><li>• Added description about new instance flavors in <a href="#">Specifications</a>, <a href="#">Buying a Kafka Instance</a>, and <a href="#">Modifying Kafka Instance Specifications</a>.</li></ul> |
| 2022-10-27 | <p>This release incorporates the following changes:</p> <ul style="list-style-type: none"><li>• Added <a href="#">Troubleshooting</a></li></ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |

| Date       | Description                               |
|------------|-------------------------------------------|
| 2022-08-12 | This issue is the first official release. |