# Distributed Message Service for Kafka

# **User Guide**

 Issue
 01

 Date
 2022-08-12





HUAWEI TECHNOLOGIES CO., LTD.

#### Copyright © Huawei Technologies Co., Ltd. 2022. 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 Technologies Co., Ltd.

#### **Trademarks and Permissions**

NUAWEI and other Huawei trademarks are trademarks 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 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 and RabbitMQ	6
1.6 Comparing DMS for Kafka and Open-Source Kafka	8
1.7 Notes and Constraints	10
1.8 Related Services	11
1.9 Basic Concepts	12
1.10 Permissions Management	13
2 Permissions Management	. 17
2.1 Creating a User and Granting DMS for Kafka Permissions	17
2.2 DMS for Kafka Custom Policies	18
3 Preparing Required Resources	20
4 Buying an Instance	22
5 Accessing a Kafka Instance	. 26
5.1 Accessing a Kafka Instance Without SASL	26
5.2 Accessing a Kafka Instance with SASL	28
5.3 Connecting to Kafka Manager and Viewing Kafka Information	31
6 Managing Instances	37
6.1 Modifying Instance Specifications	37
6.2 Viewing an Instance	38
6.3 Restarting an Instance	39
6.4 Deleting an Instance	40
6.5 Modifying the Information About an Instance	41
6.6 Resetting SASL_SSL User Password	42
6.7 Resetting Kafka Manager Password	43
6.8 Restarting Kafka Manager	44
6.9 Managing Instance Tags	45
6.10 Viewing Background Tasks	46

7 Managing Topics	48
7.1 Creating a Topic	
7.2 Deleting a Topic	51
7.3 Modifying Topic Aging Time	52
7.4 Changing Partition Quantity	
8 Managing Messages	56
8.1 Querying Messages	56
9 Managing Users	58
9.1 Creating a SASL_SSL User	
9.2 Granting Permissions to a SASL_SSL User	59
10 Managing Consumer Groups	61
10.1 Deleting a Consumer Group	61
10.2 Viewing Consumer Connection Addresses	62
11 Modifying Kafka Parameters	64
12 Quotas	69
13 Monitoring	71
12.1 Vofka Matrice	
13.1 Kalka Metrics	
13.3 Viewing Metrics	80
14 Auditing	
14 Additing	
14.1 Operations Logged by CTS	
15 FAQs	86
15.1 Instances	
15.1.1 Why Can't I Select Two AZs?	
15.1.2 Why Can't I View the Subnet and Security Group Information When Creating a DMS Instan	ce?86
15.1.3 How Do I Select Storage Space for a Kafka Instance?	
15.1.4 How Do I Choose Between High I/O and Ultra-high I/O?	
15.1.5 Which Capacity Threshold Policy Should I Use?	8/
15.1.6 Which Katka Versions Are Supported?	
15.1.7 What is the Zookeeper Address of a Karka Instance?	87
15.1.6 Are Karka Instances In Cluster Mode:	0/
15.1.9 Call F Modify the Connection Address for Accessing a Karka Instance?	
15.1.10 How Long Are Kalka SSE Certificates Valid for :	
15.1.12 How Do I Change the SASL SSL Setting of a Kafka Instance?	00 
15.1.13 Will a Kafka Instance Be Restarted After Its Enterprise Project Is Modified?	
15.1.14 Are Kafka Brokers and Zookeener Denloved on the Same VM or on Different VMs?	
15.1.15 Which Cipher Suites Are Supported by Kafka?	

15.1.16 Can I Change an Instance from Single-AZ Deployment to Multi-AZ Deployment?	88
15.1.17 Does DMS for Kafka Support Cross-AZ Disaster Recovery? Where Can I View the AZs Configure	ed
for an Existing Instance?	89
15.1.18 Do Kafka Instances Support Disk Encryption?	89
15.1.19 Does Specification Modification Affect Services?	89
15.1.20 Can I Change the VPC and Subnet After a Kafka Instance Is Created?	89
15.1.21 Where Can I Find Kafka Streams Use Cases?	89
15.1.22 Can I Upgrade Kafka Instances?	89
15.1.23 Why Is the Version on the Console Different from That in Kafka Manager?	89
15.2 Connections	89
15.2.1 Troubleshooting Kafka Connection Exceptions	89
15.2.2 How Do I Select and Configure a Security Group?	92
15.2.3 Can I Access a Kafka Instance Over a Public Network?	92
15.2.4 How Many Connection Addresses Does a Kafka Instance Have by Default?	93
15.2.5 Do Kafka Instances Support Cross-Region Access?	93
15.2.6 Do Kafka Instances Support Cross-Subnet Access?	93
15.2.7 Why Can't I Access Kafka Using SSL from a Go Client?	93
15.2.8 What If Certificates Fail to Be Loaded for SASL Connection?	94
15.2.9 Does DMS for Kafka Support Password-Free Access?	94
15.2.10 Obtaining Kafka Clients	94
15.2.11 Does DMS for Kafka Support Authentication on Clients by the Server?	94
15.2.12 Can I Use PEM SSL Truststore When Connecting to a Kafka Instance with SASL_SSL Enabled?	94
15.2.13 What Are the Differences Between JKS and CRT Certificates?	94
15.2.14 Which TLS Version Does DMS for Kafka Support, 1.0, 1.1, or 1.2?	94
15.2.15 Is There a Limit on the Number of Connections to a Kafka Instance?	94
15.2.16 How Many Connections Are Allowed from Each IP Address?	95
15.3 Topics and Partitions	95
15.3.1 Is There a Limit on the Number of Topics in a Kafka Instance?	95
15.3.2 Why Is Partition Quantity Limited?	96
15.3.3 Can I Reduce the Partition Quantity?	96
15.3.4 Why Do I Fail to Create Topics?	96
15.3.5 Do Kafka Instances Support Batch Importing Topics or Automatic Topic Creation?	96
15.3.6 Why Do Deleted Topics Still Exist?	97
15.3.7 Can I View Disk Space Used by a Topic?	97
15.3.8 Can I Add ACL Permissions for Topics?	97
15.3.9 How Do I Create a Topic?	97
15.3.10 What Should I Do If Kafka Storage Space Is Used Up Because Retrieved Messages Are Not Deleted?	98
15.3.11 What Can I Do If Disk Usage Is as High as 96%?	98
15.3.12 How Do I Increase the Partition Quantity?	98
15.3.13 Will a Kafka Instance Be Restarted After Its Automatic Topic Creation Setting Is Modified?	98
15.3.14 How Do I Disable Automatic Topic Creation?	98
15.3.15 Can I Delete Unnecessary Topics in a Consumer Group?	99

15.3.16 What Can I Do If a Consumer Fails to Retrieve Messages from a Topic Due to Insufficient Permissions?	99
15.3.17 Why Does an Instance Contain Default Topicstrace andconsumer_offsets?	. 100
15.4 Consumer Groups	. 100
15.4.1 Do I Need to Create Consumer Groups, Producers, and Consumers for Kafka Instances?	. 100
15.4.2 Do I Need to Unsubscribe from a Topic Before Deleting a Consumer Group?	. 100
15.4.3 Will a Consumer Group Without Active Consumers Be Deleted in 14 Days?	. 100
15.4.4 Why Do I See a Deleted Consumer Group on Kafka Manager?	. 101
15.5 Messages	. 101
15.5.1 What Is the Maximum Size of a Message that Can be Created?	. 101
15.5.2 Why Does Message Poll Often Fail During Rebalancing?	. 101
15.5.3 Why Can't I Query Messages on the Console?	. 102
15.5.4 What Can I Do If Kafka Messages Are Accumulated?	102
15.5.5 Why Do Messages Still Exist After the Retention Period Elapses?	.103
15.5.6 Do Kafka Instances Support Delayed Message Delivery?	. 103
15.5.7 Does the Message Query Function on the Console Show the Replica Message Quantity?	103
15.5.8 How Do I View Accumulated Messages?	. 103
15.5.9 Why Is the Message Creation Time Displayed as Year 1970?	103
15.6 Kafka Manager	. 103
15.6.1 Can I Configure a Kafka Manager Account to Be Read-Only?	104
15.6.2 Why Can't I See Broker Information After Logging In to Kafka Manager?	. 104
15.6.3 Yikes! Insufficient partition balance when creating topic : projectman_project_enterprise_project again	t Try . 104
15.6.4 Can I Query the Body of a Message by Using Kafka Manager?	.104
15.6.5 Can I Change the Port of the Kafka Manager Web UI?	104
15.6.6 Which Topic Configurations Can Be Modified on Kafka Manager?	105
15.6.7 Why Is Information Displayed on Kafka Manager Inconsistent with Cloud Eye Monitoring Data	? . 105
15.6.8 How Do I Change a Partition Leader for a Topic in Kafka Manager?	106
15.7 Monitoring & Alarm	. 108
15.7.1 Why Can't I View the Monitoring Data?	. 108
15.7.2 How Do I Configure Alarms for a Kafka Instance?	109
15.7.3 Why Is the Monitored Number of Accumulated Messages Inconsistent with the Message Quan Displayed on the Kafka Console?	tity 110
15.7.4 Why Is a Consumer Group Still on the Monitoring Page After Being Deleted?	
A Change History	111

# 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 resources, configure topics, partitions, and replicas 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, API calling, 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 replicated synchronously or asynchronously between replicas.

• High availability

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

Brokers of a Kafka instance can be deployed across AZs to further enhance service availability.

• Simple O&M

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

• Massive accumulation and auto scaling

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

• Flexible specifications

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

## **1.3 Application Scenarios**

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

#### **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.



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 1.1.0, 2.3.0, and 2.7. The instance specifications are classified based on bandwidth, including 100 MB/s, 300 MB/s, 600 MB/s, and 1200 MB/s.

#### 

- The number of brokers varies according to the underlying resources, and the underlying resources vary from region to region. The following table lists the specifications.
- In the following table, transactions per second (TPS) are calculated assuming that the size of a message is 1 KB.

Band width	Brok ers	Underlying Resource Type	I/О Туре	TPS (High- Throughput )	TPS (Synchro nous Replicati on)	Maxim um Partitio ns
100 MB/s	3	c6_2 vCPUs   4 GB	High I/O	100,000	60,000	300
		c6_2 vCPUs   4 GB	Ultra- high I/O	100,000	80,000	300
300 MB/s	3	c6_4 vCPUs   8 GB	High I/O	300,000	150,000	900
		c6_4 vCPUs   8 GB	Ultra- high I/O	300,000	200,000	900
600 MB/s	4	c6_8 vCPUs   16 GB	Ultra- high I/O	600,000	300,000	1800
1200 MB/s	8	c6_8 vCPUs   16 GB	Ultra- high I/O	1.2 million	400,000	1800

 Table 1-1
 Kafka instance specifications

#### Bandwidth Selection

The bandwidth of a Kafka instance refers to the maximum read or write bandwidth. You are advised to select a bandwidth 30% higher than what is required.

• 100 MB/s

Recommended for up to 3000 client connections, 60 consumer groups, and 70 MB/s service traffic.

• 300 MB/s

Recommended for up to 10,000 client connections, 300 consumer groups, and 210 MB/s service traffic.

• 600 MB/s

Recommended for up to 20,000 client connections, 600 consumer groups, and 420 MB/s service traffic.

• 1200 MB/s

Recommended for up to 20,000 client connections, 600 consumer groups, and 840 MB/s service traffic.

#### **Storage Space Selection**

Kafka instances support storage with 1 to 3 replicas. The storage space is space consumed by all replicas. When creating an instance, specify its storage space based on the expected service message size and the number of replicas.

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

The storage space can be expanded as your service grows.

#### **Topic Quantity**

There is no limit on the topic quantity, but there is an upper limit on 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 and the specified number of partitions in each topic (see **Table 1-1**).

#### The maximum number of partitions for a 100 MB/s instance is 300.

- If the number of partitions of each topic in the instance is 3, the number of topics is 300/3 = 100.
- If the number of partitions of each topic in the instance is 1, the number of topics is 300/1 = 300.

### 1.5 Comparing Kafka and RabbitMQ

Kafka is pull-based and provides higher throughput. It is suitable for collecting and delivering large volumes of data, such as collecting and analyzing logs. RabbitMQ does not provide as high throughput as Kafka, but it offers more message queuing functions.

The following is a comparison analysis on the performance, data reliability, service availability, and functions of Kafka and RabbitMQ.

#### Performance

The performance of message-oriented middleware is measured by throughput. While RabbitMQ provides tens of thousands of QPS, Kafka provides millions.

However, if idempotency and transactions are enabled for Kafka, its performance will be compromised.

#### **Data Reliability**

Both Kafka and RabbitMQ provide the replication mechanism to ensure high data reliability.

#### Service Availability

Kafka runs in clusters and has partitions and replicas. Therefore, single-node failure does not affect services and the capacity of Kafka can be linearly scaled up.

RabbitMQ also supports clustered deployment with multiple node quantity options.

#### Functions

Both Kafka and RabbitMQ are popular open-source message-oriented middleware. They differ mainly in the functions, which are listed in the following table.

Tahla	1_2	Function	differences	hetween	Kafka	and	
Table	1-2	FUNCTION	unterences	Detween	Ndika	anu	RadditiviQ

Function	Kafka v1.1.0/2.3.0/2.7	RabbitMQ v3.7.17
Priority queue	Not supported	Supported. It is recommended that the priority be set to 0–10.
Delayed queue	Not supported	Supported
Dead letter queue	Not supported	Supported
Retry	Not supported	Not supported
Retrieval mode	Pull-based	Pull-based and push-based
Message broadcasti ng	Supported	Supported
Message tracking	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 accumulat ion	Supports higher accumulation performance than RabbitMQ thanks to high throughput.	Supported
Persistenc e	Supported	Supported
Message tracing	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	Not supported, but can be encapsulated.
Multi- tenancy	Not supported	Supported
Multi- protocol	Only Apache Kafka is supported.	RabbitMQ is based on AMQP and supports MQTT and STOMP.

Function	Kafka v1.1.0/2.3.0/2.7	RabbitMQ v3.7.17
Multi- language	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	Supports throttling on producer or consumer clients.	Supports credit-based throttling on producers, a mechanism that triggers protection from within.
Ordered message delivery	Supports partition-level FIFO.	Supports FIFO only for single- threaded message queuing without advanced features such as delayed queues or priority queues.
Security	Supports SSL and SASL authentication and read/ write permissions control.	Similar to Kafka.
ldempote ncy	Supports idempotency for a single producer session.	Not supported
Transactio n messages	Supported	Supported

#### 

The comparison is made between open-source Kafka and RabbitMQ. DMS for Kafka and DMS for RabbitMQ maintain open-source compatibility while supporting or enhancing features in the open-source versions.

# 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 1-3 Differences between DMS for Kafka and open-source Kafl	ka
--	----

Catego ry	ltem	DMS for Kafka	Open-source Kafka
Ease of use	Readily availab le	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.

Catego ry	ltem	DMS for Kafka	Open-source Kafka
	APIs	Instances can be managed easily by calling RESTful APIs.	N/A
Costs On- deman d use Multiple specifications are available to suit different needs. The instance bandwidth and disk space can be expanded without downtime.		Multiple specifications are available to suit different needs. The instance bandwidth and disk space can be expanded without downtime.	Expenses are incurred for setting up a message service and occupying underlying resources.
	Fully manag ed	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 successMature successful commerce the Vmall Festival. It the clouds customers world, and carrier-gra standards. follows up communit continuous open-sour and add su		DMS has been proven successful in large e- commerce events such as the Vmall 11.11 Shopping Festival. 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, dumping, tracing (available soon), and many other features.	Functionality is limited and requires self-development.
Reliabil ity	Highly availab le	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.

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

# **1.7 Notes and Constraints**

DMS for Kafka has the following constraints, as listed in Table 1-4.

ltem	Constraint	Description
Kafka ZooKeeper	Not exposed externally	DMS ZooKeeper does not provide services externally. It is only used to serve Kafka instances.
Version	The service version can be 1.1.0, 2.3.0, or 2.7.	Clients later than version 0.10 are supported. Use a version that is consistent with the service version.
Message size	10 MB	The message size cannot exceed 10 MB. Otherwise, the message creation will fail.
Logging in to the VM where the Kafka brokers reside	No supported	N/A

Table 1-4 Kafka usage restrictions

ltem	Constraint	Description
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.
Automatic topic creation	Supported	Configurable during instance creation. If it is enabled, a topic will be automatically created with 3 partitions and 3 replicas when a message is created in or retrieved from a topic that does not exist.
Creating consumer groups, consumers, and producers	Not required	Consumer groups, consumers, and producers are generated automatically when you use the instance.
Decreasing partition quantity	Not supported	The partition quantity cannot be decreased due to the limitations of Apache Kafka.

# **1.8 Related Services**

• Cloud Trace Service (CTS)

Cloud Trace Service (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 section "Operations That Can Be Recorded by CTS".

• VPC

Kafka premium 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 premium instances.

• Cloud Eye

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

#### **NOTE**

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

### **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

Messages in a topic 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 1-5 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 DMS for Kafka resources but prevent them from being able to delete resources or perform any high-risk operations.

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

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

For more information, see .

D NOTE

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

#### **DMS for Kafka Permissions**

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

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

You can grant permissions by using roles and policies.

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

**Table 1-5** lists all the system-defined roles and policies supported by DMS for Kafka.

Role/Policy Name	Description	Туре	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 ReadOnlyAcces s	Read-only permissions for DMS. Users granted these permissions can only view DMS data.	System- defined policy	None
DMS Administrator	Administrator permissions for DMS.	System- defined role	This role depends on the <b>Tenant Guest</b> and <b>VPC</b> <b>Administrator</b> roles.

 Table 1-5 System-defined roles and policies supported by DMS for Kafka

#### **NOTE**

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

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

**Table 1-6** Common operations supported by each system-defined policy or role of DMS for Kafka

Operation	DMS FullAccess	DMS UserAccess	DMS ReadOnlyAccess
Creating instances	$\checkmark$	×	×
Modifying instances	$\checkmark$	×	×
Deleting instances	$\checkmark$	×	×

Operation	DMS FullAccess	DMS UserAccess	DMS ReadOnlyAccess
Modifying instance specifications	$\checkmark$	×	×
Restarting instances	$\checkmark$	$\checkmark$	×
Querying instance information	$\checkmark$	$\checkmark$	$\checkmark$

# **2** Permissions Management

# 2.1 Creating a User and Granting DMS for Kafka Permissions

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

If your account does not require individual IAM users, skip this chapter.

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

#### Prerequisites

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

#### **Process Flow**



Figure 2-1 Process for granting DMS for Kafka permissions

1. Create a user group and assign permissions.

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

2. Create an IAM user.

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

3. Log in and verify permissions.

Log in to the console using the user created in **2**, and verify that the user only has read permissions for DMS for Kafka.

# 2.2 DMS for Kafka Custom Policies

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

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

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

For operation details, see "Creating a Custom Policy" in the *Identity and Access Management User Guide*. The following section contains examples of common DMS for Kafka custom policies.

#### D NOTE

{

}

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

#### **Example Custom Policies**

• Example 1: Allowing users to delete and restart instances

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

• Example 2: Denying instance deletion

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

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

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

# **3** Preparing Required Resources

#### Overview

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

#### **Required Resources**

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

Resource	Requirement	Operations
VPC and subnet	Different Kafka instances can use the same or different VPCs and subnets based on site requirements. Note the following when creating a VPC and subnet:	For details about how to create a VPC and subnet, see the <i>Virtual Private Cloud User Guide</i> .
	• The created VPC and the Kafka instance must be in the same region.	
	• Retain the default settings unless otherwise specified.	

Tahle	3-1	Kafka	resources
Table	3- I	Ndikd	resources

Resource	Requirement	Operations
Security group	<ul> <li>Different Kafka instances can use the same or different security groups. Note the following when creating a security group:</li> <li>Set Template to Custom.</li> </ul>	For details about how to create a security group and configure security group rules, see the <i>Virtual Private Cloud</i> <i>User Guide</i> .
• After a security group is created, retain the default inbound and outbound traffic rules.		
	<ul> <li>To use Kafka, add the security group rules described in Table</li> <li>3-2. Other rules can be added based on site requirements.</li> </ul>	

 Table 3-2
 Security group rules

Directio n	Protocol	Port	Source	Description
Inbound	ТСР	9092	0.0.0.0/0	Access Kafka within a VPC (without SSL encryption).
Inbound	ТСР	9093	0.0.0.0/0	Access Kafka within a VPC (with SSL encryption).
Inbound	ТСР	9999	0.0.0.0/0	Access Kafka Manager.

# **4** Buying an Instance

#### Scenario

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

#### **Before You Start**

- Before buying a Kafka instance, ensure that a VPC configured with security groups and subnets is available.
- The broker quantity varies according to the underlying resources, and the underlying resources vary from region to region. Therefore, different instances have different broker quantities.

#### Procedure

- **Step 1** Log in to the management console.
- **Step 2** Click O in the upper left corner to select a region.

**NOTE** 

Select the same region as your application service.

- **Step 3** Click and choose **Application** > **Distributed Message Service for Kafka** to open the console of DMS for Kafka.
- **Step 4** Click **Buy Instance** in the upper right corner of the page.

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

- Step 5 Specify Billing Mode, Region, Project, and AZ.
- **Step 6** Enter an instance name and select an enterprise project.
- **Step 7** Configure the following instance parameters:
  - 1. **Version**: Currently, Kafka v1.1.0, v2.3.0, and v2.7 are supported. v2.7 is recommended.

2. CPU Architecture: Currently, only x86 architecture is supported.

#### 3. Specifications:

You can view the ECS quantity and flavor, the maximum number of partitions allowed, and number of consumer groups recommended for each bandwidth option.

The **Maximum Partitions** parameter indicates the maximum number of partitions that can be created for a Kafka instance. If the total number of partitions of all topics exceeds this threshold, topic creation will fail.

4. **Storage Space**: Total disk space for storing the instance data.

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

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

- 100 MB/s bandwidth: The value range of Storage Space is 600–90,000 GB.
- 300 MB/s bandwidth: The value range of Storage Space is 1200–90,000 GB.
- 600 MB/s bandwidth: The value range of Storage Space is 2400–90,000 GB.
- 1200 MB/s bandwidth: The value range of Storage Space is 4800–90,000 GB.

**NOTE** 

- High I/O + 100 MB/s bandwidth: If the average message size is 1 KB, the transactions per second (TPS) can reach 100,000 in high throughput scenarios and 60,000 in synchronous replication scenarios.
- High I/O + 300 MB/s bandwidth: If the average message size is 1 KB, the TPS can reach 300,000 in high throughput scenarios and 150,000 in synchronous replication scenarios.
- Ultra-high I/O + 100 MB/s bandwidth: If the average message size is 1 KB, the TPS can reach 100,000 in high throughput scenarios and 80,000 in synchronous replication scenarios.
- Ultra-high I/O + 300 MB/s bandwidth: If the average message size is 1 KB, the TPS can reach 300,000 in high throughput scenarios and 200,000 in synchronous replication scenarios.
- Ultra-high I/O + 600 MB/s bandwidth: If the average message size is 1 KB, the TPS can reach 600,000 in high throughput scenarios and 300,000 in synchronous replication scenarios.
- Ultra-high I/O + 1200 MB/s bandwidth: If the average message size is 1 KB, the TPS can reach 1,200,000 in high throughput scenarios and 400,000 in synchronous replication scenarios.
- 5. **Capacity Threshold Policy**: policy used when the disk usage reaches the threshold. The default 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.
- 6. 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.

#### D NOTE

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

7. Select a security group.

A security group is a set of rules that control access to Elastic Cloud Servers (ECSs). It provides access policies for mutually trusted ECSs with the same security protection requirements in the same VPC.

You can click **Manage Security Group** to view or create security groups on the network console.

**Step 8** Configure the username and password for logging in to Kafka Manager.

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 9** Click **Advanced Settings** to configure more parameters.

1. Configure Kafka SASL\_SSL.

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

**This setting cannot be changed after the instance is created.** If you want to use a different SASL\_SSL setting after the instance is created, you must create a new instance.

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

2. Configure Automatic Topic Creation.

If it is enabled, a topic will be automatically created with 3 partitions and 3 replicas when a message is created in or retrieved from a topic that does not exist.

3. Specify Tags.

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

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

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

4. Enter a description of the instance.

Step 10 Click Buy.

- **Step 11** Confirm the instance information and click **Submit**.
- **Step 12** Return to the instance list and check whether the Kafka instance has been created.

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

- If the instance is created successfully, its status changes to **Running**.
- If the instance fails to be created, view **Instance Creation Failures**. Delete the instance by referring to **Deleting an Instance** and create another instance. If the instance creation fails again, contact customer service.

**NOTE** 

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

----End

# **5** Accessing a Kafka Instance

# 5.1 Accessing a Kafka Instance Without SASL

DMS for Kafka of the cloud service platform provides Kafka instances, which are physically isolated and exclusively occupied by each tenant. After creating a Kafka instance, you can use an open-source Kafka client to create and retrieve messages in the instance.

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** 

The Kafka server allows a maximum of 1000 connections from each IP address. Excess connections will be rejected.

#### Prerequisites

• Security group rules have been correctly configured.

A Kafka instance with SASL disabled can be accessed **within a VPC**. Correct security group rules must be configured for the instance. For details about security group configuration requirements, see **Table 3-2**.

• The instance connection address has been obtained.

For intra-VPC access, use port 9092. Obtain the instance connection address on the instance details page.

**Figure 5-1** Obtaining Kafka instance connection addresses for intra-VPC access without SASL

**Connection Address** 

IPV4

192.14.9092 ,192.14.9092 ,192.14.9092 ,192.14.9092 🗇

• A topic has been created for the Kafka instance. Otherwise, create a topic as instructed by **Creating a Topic**.

- Kafka CLI v1.1.0, v2.7, or v2.3.0 is available. Ensure that the Kafka instance and the CLI are of the same version.
- An ECS has been created, and 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 related environment variables have been configured.

#### 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\_2.11-2.3.0.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

The following example uses connection addresses **10.3.196.45:9092,10.78.42.127:9092,10.4.49.103:9092**. 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
10.3.196.45:9092,10.78.42.127:9092,10.4.49.103:9092 --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. **Ensure that the consumer group name in the command line**

# is the same as that specified in the configuration file, if any. Otherwise, message retrieval may fail.

Example:

```
[root@ecs-kafka bin]# ./kafka-console-consumer.sh --bootstrap-server
10.3.196.45:9092,10.78.42.127:9092,10.4.49.103:9092 --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

## 5.2 Accessing a Kafka Instance with SASL

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

For security purposes, TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_CBC\_SHA256, TLS\_ECDHE\_RSA\_WITH\_AES\_128\_CBC\_SHA256, and TLS\_ECDHE\_RSA\_WITH\_AES\_128\_GCM\_SHA256 are supported.

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** 

The Kafka server allows a maximum of 1000 connections from each IP address. Excess connections will be rejected.

#### Prerequisites

• Security group rules have been correctly configured.

A Kafka instance with SASL enabled can be accessed within a VPC. Correct security group rules must be configured for the instance. For details about security group configuration requirements, see **Table 3-2**.

• The instance connection address has been obtained.

For intra-VPC access, use port 9093. Obtain the instance connection address on the instance details page.

**Figure 5-2** Obtaining Kafka instance connection addresses for intra-VPC access with SASL enabled

**Connection Address** 

IPV4

192 164 1113:9093 ,192 164 0 33:9093 ,192 164 0 393:9093 🗂

- A topic has been created for the Kafka instance. Otherwise, create a topic as instructed by **Creating a Topic**.
- The client.truststore.jks certificate has been downloaded. Click the Kafka instance to go to the Basic Information page. Click 🕹 next to Kafka

**SASL\_SSL** in the **Advanced Settings** area. Download and decompress the package to obtain the client certificate file **client.truststore.jks**.

- Kafka CLI v1.1.0, v2.7, or v2.3.0 is available. Ensure that the Kafka instance and the CLI are of the same version.
- An ECS has been created, and 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 related environment variables have been configured.

#### 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\_2.11-2.3.0.tgz

**Step 3** Modify the configuration file of the Kafka CLI.

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=/opt/kafka_2.11-2.3.0/config/client.truststore.jks
ssl.truststore.password=dms@kafka
ssl.endpoint.identification.algorithm=
```

Parameter description:

- username and password are set when you enable Kafka SASL\_SSL during instance creation or when you create a SASL\_SSL user.
- **ssl.truststore.location**: path for storing the **client.truststore.jks** certificate. Even in Windows, you need to use slashes (/) for the certificate path. Do not

use backslashes (\), which are used by default for paths in Windows. Otherwise, the client will fail to obtain the certificate.

- ssl.truststore.password: server certificate password, which must be set to dms@kafka and cannot be changed.
- ssl.endpoint.identification.algorithm: whether to verify the certificate domain name. This parameter must be left blank, which indicates disabling domain name verification.
- Step 4 Access the /bin directory of the Kafka CLI.

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

**Step 5** Run the following command to create messages:

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

Parameter description:

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

The following example uses connection addresses **10.xxx.xxx.202:9093,10.xxx.xxx. 197:9093,10.xxx.xxx.68:9093**.

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.xxx.xxx.202:9093,10.xxx.xxx.
197:9093,10.xxx.xxx.68:9093 --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. Ensure that the consumer group name in the command line is the same as that specified in the configuration file, if any. Otherwise, message retrieval may fail.

Example:

[root@ecs-kafka bin]# ./kafka-console-consumer.sh --bootstrap-server 10.xxx.xxx.202:9093,10.xxx.xxx. 197:9093,10.xxx.xxx.68:9093 --topic topic-demo --group order-test --from-beginning --consumer.config ../ config/consumer.properties hello Kafka! DMS hello ^CProcessed a total of 4 messages [root@ecs-kafka bin]#
To stop retrieving messages, press **Ctrl+C** to exit.

----End

## 5.3 Connecting to Kafka Manager and Viewing Kafka Information

Kafka Manager is an open-source tool for managing Kafka. It can be used only through a web browser. In Kafka Manager, you can view the monitoring statistics and broker information of your Kafka clusters.

#### Prerequisites

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

#### Logging In to Kafka Manager

- **Step 1** Create a Windows ECS with the same VPC and security group configurations as the Kafka instance.
- **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 5-3** shows an example of the storage configuration.

- The top navigation bar provides the following functions, as shown in the red box 1 in the figure.
  - **Cluster**: viewing the list of clusters and cluster information.
  - Brokers: viewing information about brokers of a cluster.
  - **Topic**: viewing information about topics in a cluster.
  - Preferred Replica Election: electing the leader (preferred replica) of a topic. This operation is not recommended.
  - Reassign Partitions: reassigning partitions. This operation is not recommended.

- **Consumers**: viewing the status of consumer groups in a cluster.
- Red box 2 shows an example of the cluster information summary, including the number of topics and brokers in the cluster.

#### Figure 5-3 Information about clusters

Kafka N	lanager	kafka_cluster	Cluster 🔻	Brokers	Topic 🔻	Preferred Replica Election	Reassign Partitions	Consumers
						1		
Clusters / kafl	ka_cluster ,	/ Summary				-		
Cluster	Inforn	nation						
Version					2	.2.0		
Cluster	Summ	ary						
Topics			2	Bro	kers		3 2	

• Combined information about all brokers of a cluster

This page shows statistics of brokers of a cluster. **Figure 5-4** shows an example of the storage configuration.

- Red box 1 shows the list of brokers, including number of incoming and outgoing bytes of different brokers.
- Red box 2 shows the monitoring metrics of the cluster.

#### Figure 5-4 Viewing the combined information about all brokers in a cluster

Clusters / kafka_cluster / Brokers	Consumers
Brokers           Id         Host         Port         JMX Port         Bytes In         Bytes           0         172	
Id         Host         Port         JMX Port         Bytes In         Bytes           0         172.         1.163         PLAINTEXT:9091         12345         0.00         0.00	
0 172	Out
1 172	
2 172	

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

• Information about a specific broker

Click the ID of a broker to view its statistics. **Figure 5-5** shows an example of the storage configuration.

- Red box 1 shows the statistics of the broker, including the numbers of topics, partitions, and leaders, and percentages of messages, incoming traffic and outgoing traffic.
- Red box 2 shows the monitoring metrics of the broker.

#### Figure 5-5 Viewing information about a broker

Kafka Manager 🕼 kafka_duster Cluster 👻 Brokers	Topic  Preferred Replica Election Reassig	gn Partitions Consumers				
usters / kafka_cluster / Brokers / 0						
Broker Id 0			2			
Summary		Metrics				
# of Topics	2	Rate	Mean	1 min	5 min	15 min
# of Partitions	51	Messages in /sec	0.00	0.00	0.00	0.00
# of Partitions as Leader	18	Bytes in /sec	0.37	0.00	0.00	0.07
% of Messages	66.667	Bytes out /sec	0.00	0.00	0.00	0.00
% of Incoming	61.580	Bytes rejected /sec	0.00	0.00	0.00	0.00
% of Outgoing	0.000	Failed fetch request /se	0.00	0.00	0.00	0.00
		Failed produce request	/sec 0.00	0.00	0.00	0.00

#### • Topics of an instance

In the navigation bar, choose **Topic** > **List**. The displayed page shows the list of topics and information about the topics, as shown in **Figure 5-6**.

#### NOTICE

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

#### Figure 5-6 Topics of an instance

Kafka Manager	kalka_cluster Clus	ster 👻 Brokers	Topic - Preferred Re	eplica Election Reas	sign Partitions Consumers				
Clusters / kafka_cluster /	/ Topics								
Operations									
Generate Partitio	n Assignments	Run P	artition Assignments	Add Partition	ons				
Topics Show 10 + entries								Sear	ch:
Topic 🕕	# Partitions	# Brokers 1↓	Brokers Spread % ↑↓	Brokers Skew % Ti	Brokers Leader Skew %	# Replicas 1↓	Under Replicated %	Producer Message/Sec 🛛	Summed Recent Offsets
consumer_offsets	50	3	100	0	0	3	0	0.00	3
trace	9	3	100	66	66	1	0	0.00	0
topic-test	3	3	100	0	0	3	0	0.00	0
Chauden 1 to 2 of 2 on									

• Details of a topic

Click the name of a topic to view its details on the displayed page, as shown in **Figure 5-7**.

- 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.

topic-test											
pic Summary	1					Operations					
eplication					3	Delete	opic	Reassign Partitions	Genera	te Partition Assignments	
umber of Partitions					3						
n of partition offset	ts				0	Add Partitions	Upda	te Config	Manual	Partition Assignments	
al number of Broke	rs				3						
mber of Brokers for	r Topic				3	Partitions by	Broker	2			
ferred Replicas %					100	Broker	# of Partitions	# as Leader	Partitions	Skewed?	Leader Skew
kers Skewed %					0	0	3	1	(0.1.2)	false	false
okers Leader Skewed	d %				0	1	3	1	(0,1,2)	false	false
okers Spread %					100	2	3	1	(0,1,2)	false	false
nder-replicated %					0						
onfig	4	Va	lue			Consumers	consuming from	this topic	3		
tention.ms		25	9200000			group				KF	
letrics	5	Mean	1 min	5 min	15 min						
vtes in /sec		0.00	0.00	0.00	0.00						
tes out /sec		0.00	0.00	0.00	0.00						
es rejected /sec		0.00	0.00	0.00	0.00						
led fetch request /se	ec	0.00	0.00	0.00	0.00						
led produce request	t /sec	0.00	0.00	0.00	0.00						
rtition Inform	nation			6	]						_
artition	Latest Offset		Lea	der	Replicas	In Sync Replic	as	Preferred Leader	?	Under Replicated?	
	0		1		(1.0,2)	(1,0,2)		true		false	
	0		0		(0,2,1)	(0,2,1)		true		false	

#### Figure 5-7 Details of a topic

• List of consumers

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

#### **NOTE**

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

#### Figure 5-8 Viewing the list of consumers

Kafka Manager	kafka_cluster ⊂ Cluster ▼	Brokers Topic 🔻	Preferred Replica Election	Reassign Partitions	Consumers		
Clusters / kafka_cluster /	Consumers						
Consumers							
Show 10 🗢 entries					Search:		
Consumer	t↓	Туре	↑↓ Topics it consur	nes from			†↓
group		KF	topic-test: (0% c	overage, 6 lag)			
test		KF	topic-test: (0% c	overage, 0 lag)			
Showing 1 to 2 of 2 en	tries					Previous	Next

• Details of a specific consumer

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

#### Figure 5-9 Viewing consumer details

Торіс		Part	titions Co	vered %			Total Lag
Consumed T	opic Inforr	nation					
test							
sters / kafka_cluste	r / Consumers	/ test					
			DIOKCIS	Topic -	Preferred Replica Election	Reassign Partitions	Consumers

• 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**).

iguie o reming actuals of a copie
-----------------------------------

<pre>usters / kafka_cluster / Consumers / test / topic-test     test / topic-test</pre>							
Topic Sumn	nary						
Total Lag			0				
% of Partitions a	issigned to a consumer	Instance	0				
topic-test							
0	6	6		<b>Lag</b> 0			
1	6	6		0			
2	6	6		0			

# 6 Managing Instances

## 6.1 Modifying Instance Specifications

#### Scenario

You can increase the bandwidth and storage space of Kafka instances. The storage space can be expanded a maximum of 20 times. If you increase the bandwidth, the maximum number of partitions will also be increased.

You can change instances to higher specifications, but not lower ones.

#### Note

- The storage space and bandwidth cannot be expanded at the same time.
- During bandwidth increase, the storage space is proportionally expanded based on the current disk space.

#### Procedure

- **Step 1** Log in to the management console.
- **Step 2** Click <sup>1</sup> in the upper left corner to select a region.

**NOTE** 

Select the same region as your application service.

- Step 3 Click = and choose Application > Distributed Message Service for Kafka to open the console of DMS for Kafka.
- **Step 4** In the row containing the instance for which you want to modify the specifications, choose **More** > **Modify Specifications** in the **Operation** column.
- **Step 5** Specify the required storage space or bandwidth.

The storage space and bandwidth cannot be expanded at the same time.

• Specify a new bandwidth, click **Next**, confirm the details, and click **Submit**.

View the new **Bandwidth** of the instance on the instance details page.

#### **NOTE**

- Bandwidth is increased by scaling out brokers. The original brokers and services are not affected.
- New topics are created on new brokers. After the bandwidth is increased, the original topics are still on the original brokers. To use new brokers, migrate the topics by using Kafka Manager or create new topics.
- Specify a new storage space, click **Next**, confirm the details, and click **Submit**.

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

#### **NOTE**

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

```
----End
```

## 6.2 Viewing an Instance

#### Scenario

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

#### Procedure

- **Step 1** Log in to the management console.
- **Step 2** Click <sup>1</sup> in the upper left corner to select a region.

**NOTE** 

Select the same region as your application service.

- **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 project, tag, status, name, ID, or connection address. **Table 6-1** describes the various possible statuses of a Kafka instance.

Table 6-1 Kafka instance status description

Status	Description
Creating	The instance is being created.

Status	Description
Running	The instance is running properly.
	Only instances in the <b>Running</b> state can provide services.
Faulty	The instance is not running properly.
Starting	The status between <b>Frozen</b> and <b>Running</b> .
Restarting	The instance is being restarted.
Changing	The instance specifications are being changed.
Change failed	The instance specifications failed to be changed.
Frozen	The instance is frozen.
Freezing	The status between <b>Running</b> and <b>Frozen</b> .
Creation failed	The instance failed to be created.
Upgrading	The instance is being upgraded.
Rolling back	The instance is being rolled back.

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

----End

## 6.3 Restarting an Instance

#### Scenario

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

#### NOTICE

When a Kafka instance is being restarted, message retrieval and creation requests of clients will be rejected.

#### Prerequisites

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

#### Procedure

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

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

**NOTE** 

Select the same region as your application service.

- **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**.
- Step 5 Click Yes to restart the Kafka instance.

It takes 3 to 15 minutes to restart a Kafka instance. After the instance is successfully restarted, its status should be **Running**.

#### **NOTE**

Restarting a Kafka instance only restarts the instance process and does not restart the VM where the instance is located.

----End

### 6.4 Deleting an Instance

#### Scenario

With a few clicks on the Kafka console, you can delete one or more Kafka instances that have been created or failed to be created.

#### NOTICE

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

#### Prerequisites

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

#### **Deleting Kafka Instances**

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

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

**NOTE** 

Select the same region as your application service.

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

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

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

**NOTE** 

Kafka instances in the **Creating**, **Starting**, **Changing**, **Change failed**, or **Restarting** state cannot be deleted.

Step 5 Click Yes.

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

----End

#### Deleting Kafka Instances That Failed to Be Created

- **Step 1** Log in to the management console.
- **Step 2** Click  $\bigcirc$  in the upper left corner to select a region.

Select the same region as your application service.

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

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

D NOTE

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

**Step 4** Click the icon or quantity next to **Instance Creation Failures**.

The Instance Creation Failures dialog box is displayed

- **Step 5** Delete Kafka instances that failed to be created in either of the following ways:
  - To delete all Kafka instances that failed to be created at once, click **Clear Failed Instance**.
  - To delete a single Kafka instance that failed to be created, click **Delete** in the row containing the chosen Kafka instance.
  - ----End

## 6.5 Modifying the Information About an Instance

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

#### Procedure

- **Step 1** Log in to the management console.
- **Step 2** Click <sup>1</sup> in the upper left corner to select a region.

Select the same region as your application service.

- **Step 3** Click and choose **Application** > **Distributed Message Service for Kafka** to open the console of DMS for Kafka.
- **Step 4** Click the desired Kafka instance to view the instance details.
- **Step 5** Modify the following parameters if needed:
  - Instance Name
  - Time Window
  - Enterprise Project (Changing the enterprise project will not restart the instance.)
  - Description
  - Security Group
  - Capacity Threshold Policy (Modifying this setting will not restart the instance.)
  - Automatic Topic Creation (Modifying this setting will restart the instance.)

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

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

----End

### 6.6 Resetting SASL\_SSL User Password

#### Scenario

You can reset the SASL\_SSL user password if you forget it.

#### **NOTE**

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

#### Procedure

- **Step 1** Log in to the management console.
- **Step 2** Click <sup>1</sup> in the upper left corner to select a region.

**NOTE** 

Select the same region as your application service.

- **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.
- Step 5 On the Users tab, 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. Reset the password again. If you still fail to reset the password after multiple attempts, contact customer service.

**NOTE** 

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

----End

## 6.7 Resetting Kafka Manager Password

#### Scenario

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

#### **NOTE**

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

#### Prerequisites

A Kafka instance has been created.

#### Procedure

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

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

D NOTE

Select the same region as your application service.

- **Step 3** Click and choose **Application** > **Distributed Message Service for Kafka** to open the console of DMS for Kafka.
- **Step 4** Choose **More** > **Reset Manager Password** in the row containing the instance for which you want to reset the Kafka Manager password.
- Step 5 Enter and confirm a new password, and click OK.
  - If the password is successfully reset, a success message is displayed.
  - If the password fails to be reset, a failure message is displayed. Reset the password again. If you still fail to reset the password after multiple attempts, contact customer service.

**NOTE** 

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

----End

## 6.8 Restarting Kafka Manager

#### Scenario

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

Figure 6-1 Error information



#### **NOTE**

Restarting Kafka Manager does not affect services.

#### Procedure

- **Step 1** Log in to the management console.
- **Step 2** Click **I** in the upper left corner to select a region.

Select the same region as your application service.

- **Step 3** Click and choose **Application** > **Distributed Message Service for Kafka** to open the console of DMS for Kafka.
- **Step 4** Choose **More** > **Reset Manager Password** in the row containing the instance for which you want to reset the Kafka Manager password.

#### Step 5 Click OK.

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

----End

## 6.9 Managing Instance Tags

Tags facilitate Kafka instance identification and management.

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

A tag consists of a tag key and a tag value. **Table 6-2** lists the tag key and value requirements.

Parameter	Requirements
Tag key	Cannot be left blank.
	<ul> <li>Must be unique for the same instance.</li> </ul>
	<ul> <li>Can contain a maximum of 36 characters.</li> </ul>
	<ul> <li>Cannot contain the following characters: =*&lt;&gt; /</li> </ul>
	• Cannot start or end with a space.
Tag value	• Can contain a maximum of 43 characters.
	<ul> <li>Cannot contain the following characters: =*&lt;&gt; /</li> </ul>
	• Cannot start or end with a space.

Table 6-2 Tag key and value requirements

#### Procedure

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

**Step 2** Click <sup>I</sup> in the upper left corner to select a region.

**NOTE** 

Select the same region as your application service.

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

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

#### **Step 5** Click the **Tags** tab.

View the tags of the instance.

- **Step 6** Perform the following operations as required:
  - Add a tag
    - a. Click Add/Edit Tag.
    - b. Enter a tag key and a tag value, and click **Add**.
      - If you have created predefined tags, select a pair of tag key and value, and click **Add**.
    - c. Click **OK**.
  - Delete a tag
    - a. In the row containing the tag to be deleted, click **Delete**.
    - b. Click Yes.

----End

## 6.10 Viewing Background Tasks

After you initiate certain instance operations such as modifying the capacity threshold capacity, a background task will start for each operation. On the console, you can view the background task status and clear task information by deleting task records.

#### Procedure

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

**Step 2** Click <sup>1</sup> in the upper left corner to select a region.

**NOTE** 

Select the same region as your application service.

- **Step 3** Click and choose **Application** > **Distributed Message Service for Kafka** to open the console of DMS for Kafka.
- **Step 4** Click a Kafka instance to go to the **Basic Information** page.
- Step 5 Click the Background Tasks tab.

A list of background tasks is displayed.

**Step 6** Click the default period, select the start time and end time, and click **OK**. Tasks started in the specified period are displayed.

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

- Click Click to refresh the task status.
- To clear the record of a background task, click **Delete**.

#### 

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

----End

## 7 Managing Topics

## 7.1 Creating a Topic

A topic is a stream of messages. If automatic topic creation is not enabled during Kafka instance creation, you need to manually create topics for creating and retrieving messages. If you have enabled automatic topic creation during instance creation, you do not need to create topics manually. When a message is created, a topic will be automatically created with 3 partitions and 3 replicas.

A Kafka instance has no limit on the number of topics. However, if the total number of partitions of all topics reaches the upper limit, no more topics can be created. The total number of partitions varies with instance specifications. For details, see **Specifications**.

Methods that can be used to manually create a topic:

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

#### **NOTE**

If an instance node is faulty, an internal service error may be reported when you query messages in a topic with only one replica. Therefore, you are not advised to use a topic with only one replica.

#### Method 1: Creating a Topic on the Console

- **Step 1** Log in to the management console.
- **Step 2** Click **I** in the upper left corner to select a region.

**NOTE** 

Select the same region as your application service.

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

 $\times$ 

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

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

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

#### Figure 7-1 Creating a topic

#### Create Topic

Topic Name	topic-77229911					
Partitions	- 3 + Value range: 1 to 100					
	Number of partitions in the topic. Messages in the topic will distributed to these partitions to achieve scalability and fault tolerance.					
Replicas	- 3 + Value range: 1 to 3					
	Number of message copies. This number is fixed once the topic is created.					
Aging Time (h)	- 72 + Value range: 1 to 168					
	Time after which data in the topic expires.					
Synchronous Replication						
Synchronous Flushing						

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

Table 7-1 Topic parameters

Parameter	Description
Topic Name	When creating a topic, you can modify the automatically generated topic name.
	Once the topic is created, you cannot modify its name.
Partitions	A larger number of partitions for a topic indicates more messages retrieved concurrently.
	If this parameter is set to <b>1</b> , messages will be retrieved in the FIFO order.
	Value range: 1 to 100
	Default value: <b>3</b>

Parameter	Description
Replicas	A higher number of replicas delivers higher reliability. Data is automatically backed up on each replica. When one Kafka broker becomes faulty, data is still available on other brokers.
	If this parameter is set to <b>1</b> , only one set of data is available.
	Value range: 1 to 3
	Default value: <b>3</b>
	<b>NOTE</b> If an instance node is faulty, an internal service error may be reported when you query messages in a topic with only one replica. Therefore, you are not advised to use a topic with only one replica.
Aging Time (h)	The period that messages are retained for. Consumers must retrieve messages before this period ends. Otherwise, the messages will be deleted and can no longer be retrieved. Value range: 1 to 168
	Default value: <b>72</b>
Synchronous Replication	A message is returned to the client only after the message creation request has been received and the message has been acknowledged by all replicas.
	After enabling synchronous replication, configure acks = -1 on the client. Otherwise, this function will not take effect.
	If there is only one replica, synchronous replication cannot be enabled.
Synchronous Flushing	An indicator of whether a message is immediately flushed to disk once created.
	• Enabled: A message is immediately flushed to disk once it is created, resulting in higher reliability.
	• Disabled: A message is stored in the memory instead of being immediately flushed to disk once created.

#### Step 7 Click OK.

----End

#### Method 2: Creating a Topic on Kafka Manager

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

Figure 7-2	Creating a	a topic (	on Kafka	Manager
	J			

& Kafka Manager	kafka_cluster	Cluster -	Brokers	Topic 🗸	Preferred Replica Election	Reassign Partitions	Consumers
				Create			
				List			

#### Method 3: Create a Topic by Using Kafka CLI

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

- If SASL is not enabled for the Kafka instance, run the following command 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. 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 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-userconfig.properties

## 7.2 Deleting a Topic

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

#### Prerequisites

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

#### Procedure

- **Step 1** Log in to the management console.
- **Step 2** Click <sup>1</sup> in the upper left corner to select a region.

**NOTE** 

Select the same region as your application service.

- **Step 3** Click = and choose **Application** > **Distributed Message Service for Kafka** to open the console of DMS for Kafka.
- **Step 4** Click the desired Kafka instance to view the instance details.
- **Step 5** Click the **Topics** tab.
- **Step 6** Delete topics using either of the following methods:
  - Select one or more topics and click **Delete Topic** in the upper left corner.
  - In the row containing the topic you want to delete, click **Delete**.
- Step 7 Click Yes.

----End

## 7.3 Modifying Topic Aging Time

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

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

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

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

#### **NOTE**

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

#### Procedure

- **Step 1** Log in to the management console.
- **Step 2** Click <sup>I</sup> in the upper left corner to select a region.

#### **NOTE**

Select the same region as your application service.

- **Step 3** Click and choose **Application** > **Distributed Message Service for Kafka** to open the console of DMS for Kafka.
- **Step 4** Click the desired Kafka instance to view the instance details.
- **Step 5** Click the **Topics** tab.
- Step 6 Modify the topic aging time using either of the following methods:
  - Select one or more topics and click **Edit Topic** in the upper left corner.
  - In the row containing the desired topic, click **Edit**.
- **Step 7** Enter the aging time and click **OK**.

----End

## 7.4 Changing Partition Quantity

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

#### 

Changing the number of partitions does not affect services.

There are two methods:

- Method 1: By Using the Console
- Method 2: By Using Kafka Manager

#### Method 1: By Using the Console

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

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

Select the same region as your application service.

- **Step 3** Click = and choose **Application** > **Distributed Message Service for Kafka** to open the console of DMS for Kafka.
- **Step 4** Click the desired Kafka instance to view the instance details.
- **Step 5** Click the **Topics** tab.
- **Step 6** Modify the number of partitions using either of the following methods:
  - Select one or more topics and click **Edit Topic** in the upper left corner.
  - In the row containing the desired topic, click Edit.
- **Step 7** Enter the number of partitions and click **OK**.

**NOTE** 

- The number of partitions can only be increased.
- To ensure performance, the Kafka console allows a maximum of 100 partitions for each topic.
- The total number of partitions of all topics cannot exceed the maximum number of partitions allowed by the instance.

----End

#### Method 2: By Using Kafka Manager

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

- **Step 2** Choose **Topic** > **List** to view the list of topics.
- Step 3 Click a topic to view its details.
- Step 4 Click Add Partitions.

Figure 7-3 Topic details page

	Kafka Manager	kafka_cluster	Cluster -	Brokers	Торіс 🕶	Preferred Replica Ele	ection Reassi	gn Partitions	Consumers			
Clusters / kafka_cluster / Topics	/ topic.test											
← topic.test												
Topic Summary					Oper	rations						
Replication			3			Delete Topic		Reassign Parl	titions	Generate Par	tition Assignments	
Number of Partitions			1									
Sum of partition offsets			0		Add F	Partitions	Update C	config		Manual Partiti	on Assignments	
Total number of Brokers			3									
Number of Brokers for Topic			3		Parti	tions by Broker						
Preferred Replicas %			100		Broker	/ # of F	Partitions	# as Leader	Parti	tions	Skewed?	Leader
Brokers Skewed %			0		0	1		0	(0)		false	false
Brokers Leader Skewed %			0		1	1		-	(0)		false	false
Brokers Spread %			100		2	1		- 1	(0)		false	false
Under-replicated %			0		-	'			(0)		10100	10100

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

Figure 7-4 Adding partitions

dd Partitions	Brokers				
opic	Select All Select None				
topic.test	☑ 0 - 192.168.1.68				
artitions	✓ 1 - 192.168.1.205				
3	€ 2 - 192.168.1.243				

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

Figure 7-5 Partitions added

Add Partitions	
Done!	
Go to topic view.	

#### 

- 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

## **8** Managing Messages

## 8.1 Querying Messages

#### Scenario

You can view the size, creation time, and body of messages in topics.

#### Procedure

- **Step 1** Log in to the management console.
- **Step 2** Click O in the upper left corner to select a region.

**NOTE** 

Select the same region as your application service.

- **Step 3** Click and choose **Application** > **Distributed Message Service for Kafka** to open the console of DMS for Kafka.
- **Step 4** Click the desired Kafka instance to view the instance details.
- **Step 5** Click the **Message Query** tab. Then specify the topic name and the search method.

You can search by the following methods:

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

#### **NOTE**

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

The following shows messages that have been queried.

#### Figure 8-1 Querying topic messages

* Topic Name	topic-01	▼ Partition	Enter a partition number.		* Search By	Creation time •	Mar 19, 2021 11:25:35 — Mar 19, 2	2021 11:35:35	
								Search	iet
Topic Name		Pa	artition	Offset	Message Size (Byte)	Created 1		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	

**Message Size** indicates the size of messages stored on the disk. Click **View Message Body** to view details about a message.

----End

## **9** Managing Users

## 9.1 Creating a SASL\_SSL User

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

This section describes how to create a SASL\_SSL user after SASL\_SSL is enabled for a Kafka instance. For details about how to grant user permissions, see **Granting Permissions to a SASL\_SSL User**.

#### Prerequisites

SASL\_SSL has been enabled when you create the Kafka instance.

#### Procedure

- **Step 1** Log in to the management console.
- **Step 2** Click <sup>(Q)</sup> in the upper left corner to select a region.

**NOTE** 

Select the same region as your application service.

- **Step 3** Click and choose **Application** > **Distributed Message Service for Kafka** to open the console of DMS for Kafka.
- **Step 4** Click the desired Kafka instance to view the instance details.
- Step 5 On the Users tab page, click Create User.
- Step 6 Enter a username and password, and click OK.

After the SASL\_SSL user is created, grant permissions to the user by referring to **Granting Permissions to a SASL\_SSL User**.

#### D NOTE

If a SASL\_SSL user is no longer needed, locate the row that contains the user, and click **Delete**.

The SASL\_SSL user configured during the creation of a Kafka instance cannot be deleted.

----End

## 9.2 Granting Permissions to a SASL\_SSL User

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

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

If no SASL\_SSL user is granted any permission for a topic, all users can subscribe to or publish messages to the topic.

If one or more SASL\_SSL users are granted permissions for a topic, only the authorized users can subscribe to or publish messages to the topic.

#### Prerequisites

- SASL\_SSL has been enabled when you create the Kafka instance.
- (Optional) A SASL\_SSL user has been created. For details, see Creating a SASL\_SSL User.

#### Procedure

- **Step 1** Log in to the management console.
- **Step 2** Click <sup>1</sup> in the upper left corner to select a region.

**NOTE** 

Select the same region as your application service.

- **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**.
- **Step 7** In the **Users** area, select users. In the **Selected** area, configure permissions for users.

#### Figure 9-1 Granting user permissions

#### Grant User Permission

Topic Name	topic-01	Partitions	3	Aging Time (h)	72	
Replicas	3	Synchronous Flushing	No			
Enter a username						Q
Users(4)			Selected(3)			
Username			Username	Permission		Operation
receive			receive	Subscribe	•	Delete
🔽 test			test	Publish/Su.		Delete
send send			send	Publish	•	Delete
send_recei	ve					

As shown in **Figure 9-1**, only the **test**, **send**, and **receive** users can subscribe to or publish messages to topic-01. The **send\_receive** user cannot subscribe to or publish messages to topic-01.

#### Step 8 Click OK.

Click  $\checkmark$  next to the topic name to view the authorized users and their permissions.

#### Figure 9-2 Viewing authorized users and their permissions

	Topic Name	Partitions	Replicas	Aging Time (h)	Synchronous Replication	Synchronous Flushing		Operation
^ [	topic-01	3	3	72	Off	Off		Grant User Permission   Edit   Delete
Userna	ame						Permission	1
receive	e						Subscribe	
test							Publish/Su	ıbscribe
send							Publish	

----End

# **10** Managing Consumer Groups

## 10.1 Deleting a Consumer Group

You can use Kafka CLI to delete a consumer group.

#### Using the CLI to Delete a Consumer Group

The following uses Linux as an example.

- **Step 1** Download Kafka CLI **v1.1.0**, **v2.7**, or **v2.3.0**. Ensure that the Kafka instance and the CLI are of the same version.
- Step 2 Use the CLI to connect to the Kafka instance. For details, see Accessing a Kafka Instance Without SASL or Accessing a Kafka Instance with SASL.
- **Step 3** 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.

To query consumer group information, run the following commands:

• Querying the list of consumer groups:

**kafka-consumer-groups.sh** --**bootstrap-server** {*Kafka instance connection address*} --**list** 

[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 --list Note: This will not show information about old Zookeeper-based consumers. KMOffsetCache-zk-server-1 bbbb KMOffsetCache-zk-server-0

Querying the details of a consumer group:

**kafka-consumer-groups.sh** --**bootstrap-server** {*Kafka instance connection address*} --**describe** --**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 --describe --group bbbb Note: This will not show information about old Zookeeper-based consumers. Consumer group 'bbbb' has no active members.

TOPIC	PA		CURRENT-OFFSET	LOC	G-END-OFFSET	LAG	CONSUMER-ID	
test	1	336	336	0	-	-	-	
test	0	334	334	0	-	-	-	
test	2	334	334	0	-	-	-	

#### **NOTE**

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

----End

### **10.2 Viewing Consumer Connection Addresses**

You can view consumer connection addresses on Kafka Manager.

#### **NOTE**

- The connection address of a consumer can be viewed only when the consumer is connected to a Kafka instance.
- Due to cache reasons, the consumer connection addresses displayed on Kafka Manager may not be used currently. To solve this problem, restart Kafka Manager.

#### Viewing Consumer Addresses on Kafka Manager

- Step 1 Log in to Kafka Manager.
- **Step 2** Click **kafka\_cluster** to go to the cluster details page.
- Step 3 On the top menu bar, choose Consumers.

#### Figure 10-1 Navigation bar



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

#### Figure 10-2 Consumer group list

Consumers		
Show 10 + entries		
Consumer	†↓ Туре	<sup>↑↓</sup> 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 10-3 Topics that the consumer group has subscribed to

<ul> <li>◆ group01</li> </ul>				
Consumed Topic Information				
overed %				

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

Figure 10-4 Topic details page

Partition	LogSize	Consumer Offset	Lag	Consumer Instance Owner		
0	33,333	0	33,333	consumer-1-5d096c5f-159d-468d-8b10-7961dc6f49d1;		
1	33,334	0	33,334	consumer-1-5d096c5f-159d-468d-8b10-7961dc6f49d1;/:0.2)#.177.80		
2	33,333	0	33,333	consumer-1-5d096c5f-159d-468d-8b10-7961dc6f49d1:/10_3-t.1?fea		

----End

## **11** Modifying Kafka Parameters

#### Scenario

Your Kafka instances, topics, and consumers come with default configuration parameter settings. You can modify common parameters on the console. For details about parameters that are not listed on the console, see the **Kafka official website**.

Parameters of v1.1.0 instances are all static parameters. v2.3.0/2.7 instances have both dynamic and static parameters.

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

#### **NOTE**

Configuration parameters of some old instances cannot be modified. Check whether your instance parameters can be modified on the console.

#### Prerequisites

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

#### Procedure

- **Step 1** Log in to the management console.
- **Step 2** Click  $\bigcirc$  in the upper left corner to select a region.

Select the same region as your application service.

- **Step 3** Click and choose **Application** > **Distributed Message Service for Kafka** to open the console of DMS for Kafka.
- Step 4 Click the desired Kafka instance to view the instance details.

**Step 5** On the **Parameters** tab page, click **Edit** in the row containing the parameter to modify. **Table 11-1** describes the parameters of v1.1.0 instances. **Table 11-2** and **Table 11-3** describe the parameters of v2.3.0/v2.7 instances.

Parameter	Description	Value Range	Default Value
min.insync.replicas	If a producer sets the <b>acks</b> parameter to <b>all</b> (or <b>-1</b> ), the <b>min.insync.replicas</b> parameter specifies the minimum number of replicas in the in-sync replica (ISR) set. If this minimum is not met, the producer will raise an exception, and messages cannot be created.	1-3	1
message.max.byte s	Maximum length of a single message, in bytes.	0– 10,485,76 0	10,485,76 0
unclean.leader.ele ction.enable	Indicates whether to allow replicas not in the ISR set to be elected as the leader. If this parameter is set to <b>true</b> , data may be lost.	<b>true</b> or <b>false</b>	true
connections.max.i dle.ms	Idle connection timeout. Connections that are idle for the duration specified by this parameter will be closed.	5000– 600,000	600,000
log.retention.hour s	Duration (in hours) for retaining a log file. This parameter takes effect only for topics that have no aging time configured. If there is aging time configured for topics, it overrides this parameter.	1-168	72
max.connections.p er.ip	The maximum number of connections allowed from each IP address. Requests for new connections will be rejected once the limit is reached.	100– 20,000	1000
group.max.session .timeout.ms The maximum session timeout consumers. A longer timeout consumers more time to pro messages between heartbea results in a longer time to de failures.		6000– 1,800,000	1,800,000

**Table 11-1** Static parameters (v1.1.0 instances)

Parameter	Description	Value Range	Default Value
default.replication .factor	The default number of replicas configured for an automatically created topic.	1–3	3
num.partitions	The default number of partitions configured for an automatically created topic.	1–100	3
group.min.session. timeout.ms	The minimum session timeout for consumers. A shorter timeout enables quicker failure detection but results in more frequent consumer heartbeating, which can overwhelm broker resources.	6000– 300,000	6000

Table 11-2 Dynamic parameters (v2.3.0/2.7 instances)

Parameter	Description	Value Range	Default Value
min.insync.replicas	If a producer sets the <b>acks</b> parameter to <b>all</b> (or <b>-1</b> ), the <b>min.insync.replicas</b> parameter specifies the minimum number of replicas in the in-sync replica (ISR) set. If this minimum is not met, the producer will raise an exception, and messages cannot be created.	1-3	1
message.max.byte s	Maximum length of a single message, in bytes.	0– 10,485,76 0	10,485,76 0
unclean.leader.ele ction.enable	Indicates whether to allow replicas not in the ISR set to be elected as the leader. If this parameter is set to <b>true</b> , data may be lost.	<b>true</b> or <b>false</b>	true
Parameter	Description	Value Range	Default Value
----------------------------------	--	--------------------	------------------
connections.max.i dle.ms	Idle connection timeout. Connections that are idle for the duration specified by this parameter will be closed.	5000– 600,000	600,000
log.retention.hour s	Duration (in hours) for retaining a log file. This parameter takes effect only for topics that have no aging time configured. If there is aging time configured for topics, it overrides this parameter.	1-168	72
max.connections.p er.ip	The maximum number of connections allowed from each IP address. Requests for new connections will be rejected once the limit is reached.	100– 20,000	1000
group.max.session .timeout.ms	The maximum session timeout for consumers. A longer timeout gives consumers more time to process messages between heartbeats but results in a longer time to detect failures.	6000– 1,800,000	1,800,000
default.replication .factor	The default number of replicas configured for an automatically created topic.	1–3	3
num.partitions	The default number of partitions configured for an automatically created topic.	1–100	3
group.min.session. timeout.ms	The minimum session timeout for consumers. A shorter timeout enables quicker failure detection but results in more frequent consumer heartbeating, which can overwhelm broker resources.	6000- 300,000	6000

Table 11-3 Static parameters (v2.3.0/2.7 instances)

#### 

To modify multiple parameters at a time, select the parameters, and click **Modify** above the parameter list.

Step 6 Click Save.

#### **NOTE**

Modifying dynamic parameters will not restart the instance. Manually restart the instance after the static parameters are modified.

----End

# 12<sub>Quotas</sub>

#### What Is a Quota?

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

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

#### How Do I View My Quota?

- 1. Log in to the management console.
- 2. Click  $\bigcirc$  in the upper left corner to select a region and a project.
- 3. Click **(the My Quota** icon) in the upper right corner. The **Service Quota** page is displayed.
- On the Service Quota page, view the used and total quotas of resources.
   If a quota cannot meet your needs, apply for a higher quota by performing the following operations.

#### How Do I Increase My Quota?

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

Before you contact customer service, prepare the following information:

• Account name, project name, and project ID

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

- Quota information, including:
  - Service name
  - Quota type

#### - Required quota

To increase a quota, contact the administrator.

# **13** Monitoring

# 13.1 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 to query the Kafka metrics and alarms.

#### Namespace

SYS.DMS

#### **Instance Metrics**

Metric ID	Metric Name	Description	Value Range	Monitor ed Object	Mo nito ring Peri od (Ra W Dat a)
current _partiti ons	Partitio ns	Number of used partitions in the instance Unit: count	0–18,000	Kafka instance	1 min ute
current _topics	Topics	Number of created topics in the instance Unit: count	0–600	Kafka instance	1 min ute

#### Table 13-1 Instance metrics

Metric ID	Metric Name	Description	Value Range	Monitor ed Object	Mo nito ring Peri od (Ra W Dat a)
group_ msgs	Accum ulated Messag es	Total number of accumulated messages in all consumer groups of the instance Unit: count	0– 1,000,000, 000	Kafka instance	1 min ute

### **Broker Metrics**

Table <sup>•</sup>	13-2	Broker	metrics
--------------------	------	--------	---------

Metric ID	Metric Name	Description	Value Range	Monitor ed Object	Mo nito ring Peri od (Ra W Dat a)
broker_ data_si ze	Messag e Size	Total size of messages in the broker Unit: byte	0– 5,000,000, 000,000	Kafka instance broker	1 min ute
broker_ messag es_in_r ate	Messag e Creatio n Rate	Number of messages created per second Unit: count/s	0–500,000	Kafka instance broker	1 min ute
broker_ bytes_o ut_rate	Messag e Retriev al	Number of bytes retrieved per second Unit: byte/s	0– 500,000,00 0	Kafka instance broker	1 min ute
broker_ bytes_i n_rate	Messag e Creatio n	Number of bytes created per second Unit: byte/s	0- 500,000,00 0	Kafka instance broker	1 min ute

Metric ID	Metric Name	Description	Value Range	Monitor ed Object	Mo nito ring Peri od (Ra w Dat a)
broker_ public_ bytes_i n_rate	Public Inboun d Traffic	Inbound traffic over public networks per second Unit: byte/s	0– 500,000,00 0	Kafka instance broker	1 min ute
broker_ public_ bytes_o ut_rate	Public Outbou nd Traffic	Outbound traffic over public networks per second Unit: byte/s	0– 500,000,00 0	Kafka instance broker	1 min ute
broker_ fetch_ mean	Averag e Messag e Retriev al Process ing Duratio n	Average time that the broker spends processing message retrieval requests Unit: ms	0–10,000	Kafka instance broker	1 min ute
broker_ produc e_mea n	Averag e Messag e Creatio n Process ing Duratio n	Average time that the broker spends processing message creation requests Unit: ms	0–10,000	Kafka instance broker	1 min ute
broker_ cpu_cor e_load	Averag e Load per CPU Core	Average load of each CPU core of the Kafka VM	0–20	Kafka instance broker	1 min ute
broker_ disk_us age	Disk Capacit y Usage	Disk usage of the Kafka VM Unit: %	0–100	Kafka instance broker	1 min ute

Metric ID	Metric Name	Description	Value Range	Monitor ed Object	Mo nito ring Peri od (Ra W Dat a)
broker_ memor y_usag e	Memor y Usage	Memory usage of the Kafka VM Unit: %	0–100	Kafka instance broker	1 min ute
broker_ heap_u sage	JVM Heap Memor y Usage of Kafka	Heap memory usage of the Kafka JVM Unit: %	0–100	Kafka instance broker	1 min ute
broker_ alive	Broker Alive	Whether the Kafka broker is alive	1: alive 0: not alive	Kafka instance broker	1 min ute
broker_ connec tions	Connec tions	Total number of TCP connections on the Kafka broker Unit: count	> 0	Kafka instance broker	1 min ute
broker_ cpu_us age	CPU Usage	CPU usage of the Kafka VM Unit: %	0–100	Kafka instance broker	1 min ute
broker_ disk_re ad_awa it	Averag e Disk Read Time	Average time for each disk I/O read in the monitoring period Unit: ms	> 0	Kafka instance broker	1 min ute
broker_ disk_wr ite_awa it	Averag e Disk Write Time	Average time for each disk I/O write in the monitoring period Unit: ms	> 0	Kafka instance broker	1 min ute
broker_ total_b ytes_in _rate	Inboun d Traffic	Inbound traffic per second Unit: byte/s	> 0	Kafka instance broker	1 min ute

Metric ID	Metric Name	Description	Value Range	Monitor ed Object	Mo nito ring Peri od (Ra W Dat a)
broker_ total_b ytes_ou t_rate	Outbou nd Traffic	Outbound traffic per second Unit: byte/s	> 0	Kafka instance broker	1 min ute

## **Topic Metrics**

#### Table 13-3 Topic metrics

Metric ID	Metric Name	Description	Value Range	Monitor ed Object	Mo nito ring Peri od (Ra W Dat a)
topic_b ytes_in _rate	Messag e Creatio n	Number of bytes created per second Unit: byte/s NOTE This metric is available only when Scope is set to Basic monitoring on the Queues tab page.	0– 500,000,00 0	Topic in a Kafka instance	1 min ute
topic_b ytes_ou t_rate	Messag e Retriev al	Number of bytes retrieved per second Unit: byte/s NOTE This metric is available only when Scope is set to Basic monitoring on the Queues tab page.	0– 500,000,00 0	Topic in a Kafka instance	1 min ute

Metric ID	Metric Name	Description	Value Range	Monitor ed Object	Mo nito ring Peri od (Ra W Dat a)
topic_d ata_siz e	Messag e Size	Total size of messages in the queue Unit: byte <b>NOTE</b> This metric is available only when <b>Scope</b> is set to <b>Basic</b> <b>monitoring</b> on the <b>Queues</b> tab page.	0– 5,000,000, 000,000	Topic in a Kafka instance	1 min ute
topic_ messag es	Total Messag es	Total number of messages in the queue Unit: count NOTE This metric is available only when Scope is set to Basic monitoring on the Queues tab page.	≥ 0	Topic in a Kafka instance	1 min ute
topic_ messag es_in_r ate	Messag e Creatio n Rate	Number of messages created per second Unit: count/s NOTE This metric is available only when Scope is set to Basic monitoring on the Queues tab page.	0–500,000	Topic in a Kafka instance	1 min ute
partitio n_mess ages	Partitio n Messag es	Total number of messages in the partition Unit: count <b>NOTE</b> This metric is available only when <b>Scope</b> is set to <b>Partition</b> <b>monitoring</b> on the <b>Queues</b> tab page.	≥ 0	Topic in a Kafka instance	1 min ute
produc ed_mes sages	Create d Messag es	Number of messages that have been created Unit: count <b>NOTE</b> This metric is available only when <b>Scope</b> is set to <b>Partition</b> <b>monitoring</b> on the <b>Queues</b> tab page.	≥ 0	Topic in a Kafka instance	1 min ute

### **Consumer Group Metrics**

Table	13-4	Consumer	aroup	metrics
Tuble	13 -	consumer	group	methes

Metric ID	Metric Name	Description	Value Range	Monitor ed Object	Mo nito ring 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 Unit: count	≥ 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 Unit: count	≥ 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 Unit: Count	0 to 2 <sup>63</sup> –1	Consum er group of a Kafka instance	1 min ute
topic_ messag es_cons umed	Topic Retriev ed Messag es	Number of messages that have been retrieved from the specified topic in the consumer group Unit: Count	0 to 2 <sup>63</sup> –1	Consum er group of a Kafka instance	1 min ute
consum er_mes sages_r emaine d	Consu mer Availab le Messag es	Number of remaining messages that can be retrieved in the consumer group Unit: Count	0 to 2 <sup>63</sup> –1	Consum er group of a Kafka instance	1 min ute
consum er_mes sages_c onsum ed	Consu mer Retriev ed Messag es	Number of messages that have been retrieved in the consumer group Unit: Count	0 to 2 <sup>63</sup> –1	Consum er group of a Kafka instance	1 min ute

#### Dimension

Кеу	Value
kafka_instance_id	Kafka instance
kafka_broker	Kafka instance broker
kafka_topics	Topic in a Kafka instance
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

# **13.2 Configuring Alarm Rules**

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

#### **NOTE**

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

**Table 13-5** Kafka instance metrics to configure alarm rules for

Metric	Normal Range	Alarm Policy	Approac h Upper Limit	Metric Description and Alarm Handling
Disk Capacity Usage	0-100	Alarm threshold: original value > 85 Number of consecutiv e periods: 3 Alarm severity: critical	Yes	Metric description: disk usage of the Kafka VM. Unit: % Alarm handling: When an alarm is generated for this metric, modify the instance storage space. For details, see <b>Modifying Instance</b> <b>Specifications</b> .

Metric	Normal Range	Alarm Policy	Approac h Upper Limit	Metric Description and Alarm Handling
Average Load per CPU Core	0-2	Alarm threshold: original value > 1.5 Number of consecutiv e periods: 3 Alarm severity: major	Yes	Metric description: average load of each CPU core of the Kafka VM. Alarm handling: When an alarm is generated for this metric, modify the instance <b>bandwidth</b> . For details, see <b>Modifying Instance</b> <b>Specifications</b> .
Memory Usage	0-100	Alarm threshold: original value > 85 Number of consecutiv e periods: 3 Alarm severity: critical	Yes	Metric description: memory usage of the Kafka VM. Unit: % Alarm handling: When an alarm is generated for this metric, modify the instance <b>bandwidth</b> . For details, see <b>Modifying Instance</b> <b>Specifications</b> .

#### Procedure

- **Step 1** Log in to the management console.
- **Step 2** Click O in the upper left corner to select a region.

#### D NOTE

Select the same region as your application service.

- **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.

The **Create Alarm Rule** page is displayed.

#### **Step 6** Specify the alarm details.

For more information about creating alarm rules, see "Creating an Alarm Rule" in *Cloud Eye User Guide*.

- 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 13-1 Setting the alarm policy and alarm severity

* Alarm Policy           Metric Name         Alarm Policy         Alarm Severity         Operation	* Method	Configure manually		
Metric Name Alarm Policy Alarm Severity Operation	* Alarm Policy			
	Metric Name	Alarm Policy	Alarm Severity	Operation
Disk capacity Us  Kaw d  Kaw d  Kaw d  Uisk capacity Us  So One day  Citical  Citical	Disk Capac	ty Us 🔻 Raw d 🔻	▼     3 consecuti ▼     >=     ▼     0ne day     ▼	

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

----End

# **13.3 Viewing Metrics**

#### Scenario

Cloud Eye monitors Kafka instance metrics in real time. You can view these metrics on the Cloud Eye console.

#### Prerequisites

At least one Kafka instance has been created. The instance has at least one available message.

#### Procedure

- **Step 1** Log in to the management console.
- **Step 2** Click <sup>(Q)</sup> in the upper left corner to select a region.

**NOTE** 

Select the same region as your application service.

- 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**.

On the Cloud Eye console, view the metrics of the instance, nodes, queues, and consumer groups. Metric data is reported to Cloud Eye every minute.

----End

# **14** Auditing

# 14.1 Operations Logged by CTS

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

Operation	Resource Type	Trace Name
Successfully deleting a background task	kafka	deleteDMSBackendJobSuccess
Failing to delete a background task	kafka	deleteDMSBackendJobFailure
Successfully creating an order for creating an instance	kafka	createDMSInstanceOrderSuccess
Failing to create an order for creating an instance	kafka	createDMSInstanceOrderFailure
Successfully submitting a request to modify an instance order	kafka	modifyDMSInstanceOrderSuc- cess
Failing to submit a request to modify an instance order	kafka	modifyDMSInstanceOrderFailure
Successfully submitting a request to scale up an instance	kafka	extendDMSInstanceSuccess

Table 14-1 DMS for Kafka operations that can be recorded by CTS

Operation	Resource Type	Trace Name		
Failing to submit a request to scale up an instance	kafka	extendDMSInstanceFailure		
Successfully submitting a request to reset instance password	kafka	resetDMSInstancePasswordSuc- cess		
Failing to submit a request to reset instance password	kafka	resetDMSInstancePasswordFai- lure		
Successfully creating a topic for a Kafka instance	kafka	Kafka_platinum_create_topicSuc cess		
Failing to create a topic for a Kafka instance	kafka	Kafka_platinum_create_topicFail ure		
Successfully deleting a topic from a Kafka instance	kafka	Kafka_platinum_delete_topicsSu ccess		
Failing to delete a topic for a Kafka instance	kafka	Kafka_platinum_delete_topicsFai lure		
Successfully deleting an instance that failed to be created	kafka	deleteDMSCreateFailureInstan- cesSuccess		
Failing to delete an instance that failed to be created	kafka	deleteDMSCreateFailureInstan- cesFailure		
Successfully submitting a request to restart an instance	kafka	restartDMSInstanceSuccess		
Failing to submit a request to restart an instance	kafka	restartDMSInstanceFailure		
Successfully submitting a request to delete multiple instances at a time	kafka	batchDeleteDMSInstanceSuccess		
Failing to submit a request to delete multiple instances at a time	kafka	batchDeleteDMSInstanceFailure		

Operation	Resource Type	Trace Name		
Successfully submitting a request to restart multiple instances at a time	kafka	batchRestartDMSInstanceSuc- cess		
Failing to submit a request to restart multiple instances at a time	kafka	batchRestartDMSInstanceFailure		
Successfully submitting a request to modify instance information	kafka	modifyDMSInstanceInfoSuccess		
Failing to submit a request to modify instance information	kafka	modifyDMSInstanceInfoFailure		
Deleting multiple instance tasks at a time	kafka	batchDeleteDMSInstanceTask		
Successfully unfreezing an instance	kafka	unfreezeDMSInstanceTaskSuc- cess		
Failing to unfreeze an instance	kafka	unfreezeDMSInstanceTaskFai- lure		
Successfully freezing an instance	kafka	freezeDMSInstanceTaskSuccess		
Failing to freeze an instance	kafka	freezeDMSInstanceTaskFailure		
Successfully deleting an instance	kafka	deleteDMSInstanceTaskSuccess		
Failing to delete an instance	kafka	deleteDMSInstanceTaskFailure		
Successfully creating an instance	kafka	createDMSInstanceTaskSuccess		
Failing to create an kafka instance		createDMSInstanceTaskFailure		
Successfully scaling up kafka an instance		extendDMSInstanceTaskSuccess		
Failing to scale up an instance	kafka	extendDMSInstanceTaskFailure		

Operation	Resource Type	Trace Name
Successfully restarting an instance	kafka	restartDMSInstanceTaskSuccess
Failing to restart an instance	kafka	restartDMSInstanceTaskFailure
Successfully restarting multiple instances at a time	kafka	batchRestartDMSInstanceTask- Success
Failing to restart multiple instances at a time	kafka	batchRestartDMSInstanceTask- Failure
Successfully modifying instance information	kafka	modifyDMSInstanceInfoTaskSuc- cess
Failing to modify instance information	kafka	modifyDMSInstanceInfoTaskFai- lure

# 14.2 Viewing Audit Logs

See Querying Traces on the CTS Console.

# **15** FAQs

# 15.1 Instances

# 15.1.1 Why Can't I Select Two AZs?

A Kafka cluster uses 3 ZooKeeper nodes regardless of instance specifications. Kafka uses the ZooKeeper cluster to manage configurations. If the ZooKeeper cluster is faulty, Kafka will not work properly.

At least two ZooKeepers are required for the cluster to run properly.

Assume that you select only two AZs. AZ1 has one ZooKeeper node, and AZ2 has two. If AZ1 is faulty, the instance can be used properly. If AZ2 is faulty, the cluster cannot be used. In this case, the availability rate is just 50%. Therefore, do not select 2 AZs.

# 15.1.2 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.

## 15.1.3 How Do I Select Storage Space for a Kafka Instance?

The storage space is the space for storing messages. Storage space configurations for a Kafka instance include the disk type and disk size. Currently supported disk types are ultra-high I/O and high I/O.

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**). 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.

# 15.1.4 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.

# 15.1.5 Which Capacity Threshold Policy Should I Use?

Currently, the following policies are supported:

• Stop production

When the memory usage reaches 95% of the disk capacity threshold, 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, the earliest segment (1 GB) will be deleted to ensure uninterrupted services. However, 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.** 

## 15.1.6 Which Kafka Versions Are Supported?

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

### 15.1.7 What Is the ZooKeeper Address of a Kafka Instance?

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.

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

## 15.1.8 Are Kafka Instances in Cluster Mode?

Yes. A Kafka instance is a cluster.

# 15.1.9 Can I Modify the Connection Address 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 correct rules have been configured for the security group of the instance. For details, see **How Do I Select and Configure a Security Group?** 

### 15.1.10 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.

# 15.1.11 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.

# 15.1.12 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.

# 15.1.13 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.

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

Kafka brokers and ZooKeeper are deployed on the same VM.

### 15.1.15 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.

# 15.1.16 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.

## 15.1.17 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 on the **Basic Information** tab page of the instance. If there are multiple AZs, cross-AZ disaster recovery is available.

## 15.1.18 Do Kafka Instances Support Disk Encryption?

No. Kafka instances do not support disk encryption.

## **15.1.19 Does Specification Modification Affect Services?**

No. Bandwidth or storage space expansion does not affect services.

# 15.1.20 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.

## 15.1.21 Where Can I Find Kafka Streams Use Cases?

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

### 15.1.22 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.

# 15.1.23 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.

# **15.2 Connections**

# 15.2.1 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 fault:

- Checking the Network
- Checking Consumer and Producer Configurations
- Common Errors on Java Clients
- Common Errors on the Go Client

#### **Checking the Network**

Before connecting to a Kafka instance, ensure that the client and the instance are interconnected. If they cannot be connected, check the network connection.

For example, if you have enabled SASL\_SSL to access the Kafka instance, run the following command:

#### curl -kv {ip}:{port}

• If the network is normal, information similar to the following is shown:



• If the network is abnormal or disconnected, information similar to the following is shown:



#### Solution:

- 1. Check whether the client and the Kafka instance are in the same VPC.
- 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 the 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 file.

#### **Common Errors on Java Clients**

• Domain name verification enabled

The following error is displayed:

	at	java, lang. Thread. run(Thread. java: 748)
Caused	bu:	javax.net.ssl.SSLHandshakeException: General SSLEngine problem
	at	sun.security.ssl.Alerts.getSSLException(Alerts.java:192)
	at	sun.security.ssl.SSLEngineImpl.fatal(SSLEngineImpl.java:1709)
	at	sun.security.ssl.Handshaker.fatalSE(Handshaker.java:318)
	at	sun.security.ssl.Handshaker.fatalSE(Handshaker.java:310)
	at	sun.security.ssl.ClientHandshaker.serverCertificate(ClientHandshaker.java:1639)
	at	sun.security.ssl.ClientHandshaker.processMessage(ClientHandshaker.java:223)
	at	sun.security.ssl.Handshaker.processLoop(Handshaker.java:1037)
	at	sun.security.ssl.Handshaker\$1.run(Handshaker.java:970)
	at	sun .secur i ty .ssl .Handshaker\$1.run(Handshaker . java :967)
	at	java.security.AccessController.doPrivileged(Native Method)
	at	sun . secur i ty . ss l .Handshaker\$De lega ted Task .run (Handshaker . java : 1459)
	at	org.apache.kafka.common.network.SslTransportLayer.runDelegatedTasks(SslTransportLayer.java:402)
	at	org .apache .kafka .common .network .SslTransportLayer .handshakeUnwrap (SslTransportLayer .java :484)
	at	org .apache .kafka .common .network .SslTransportLayer .doHandshake(SslTransportLayer . java :340)
		. 7 more
Caused	by:	java.security.cert.CertificateException: No subject alternative names matching IP address 10.166.37.165 found
	at	sun.security.util.HostnameChecker.matchIP(HostnameChecker.java:168)
	at	sun.security.util.HostnameChecker.match(HostnameChecker.java:94)
	at	sun.security.ssl.X509TrustManagerImpl.checkIdentity(X509TrustManagerImpl.java:462)
	at	sun.security.ssl.X509TrustManagerImpl.checkIdentity(X509TrustManagerImpl.java:442)
	at	sun.security.ssl.X509TrustManagerImpl.checkTrusted(X509TrustManagerImpl.java:261)
	at	sun.security.ssl.X509TrustManagerImpl.checkServerTrusted(X509TrustManagerImpl.java:144)
	at	sun.security.ssl.ClientHandshaker.serverCertificate(ClientHandshaker.java:1626)
		. 16 more
(kaf ka	a.adı	min.TopicCommand\$)

**Solution**: Check the **consumer.properties** and **producer.properties** files, in which the **ssl.endpoint.identification.algorithm** parameter must be left empty, indicating that domain name verification is disabled.

ssl.endpoint.identification.algorithm=

• SSL certificate failing to be loaded

The following error is displayed:

20-05-28706:35:38,0541[ERX0R][logstash.outputs.kafka ] Unable to create Kafka producer from given configuration {:kafka\_error\_message=>org.apache.kafka.common.KafkaExceptic Failed to construct kafka producer, :cause=>org.apache.kafka.common.KafkaException: org.apache.kafka.common.KafkaException: Failed to

#### Solution:

- a. Check whether the **client.truststore.jks** file exists in the corresponding address.
- b. Check the permissions on the processes and files.
- c. Check whether the **ssl.truststore.password** parameter in the **consumer.properties** and **producer.properties** files is correctly set.

**ssl.truststore.password** is the server certificate password, which must be set to **dms@kafka** and cannot be changed. ssl.truststore.password=dms@kafka

• Incorrect topic name

The following error is displayed:



**Solution**: Create another topic or enable the automatic topic creation function.

#### **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\_CBC\_SHA256 or TLS\_RSA\_WITH\_AES\_128\_CBC\_SHA256 cipher suite (both are disabled by default).

## 15.2.2 How Do I Select and Configure a Security Group?

To access a Kafka instance within a VPC, configure the security group rules as follows:

Intra-VPC access

To access a Kafka instance, you must deploy your client on an ECS in the same VPC and subnet as the instance.

In addition, before you can access the instance through your client, you must configure correct rules for the security groups of both the ECS and the Kafka instance.

- a. You are advised to configure the same security group for the ECS and Kafka instance. After a security group is created, network access in the group is not restricted by default.
- b. If different security groups are configured, you may need to refer to the following configurations:

**NOTE** 

- Assume that security groups **sg-53d4** and **Default\_All** are configured respectively for your ECS and Kafka instance.
- You can specify a security group or IP address as the remote end in the following rules.

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

 Table 15-1
 Security group rule

Direction	Protocol & Port	Destination	
Outbound	All	Default_All	

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

Table 15-2 Security group rule

Direction	Protocol & Port	Source
Inbound	All	sg-53d4

## 15.2.3 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.

# 15.2.4 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.

Band width	Brok ers	Underlying Resource Type	I/O Type	TPS (High- Throughput )	TPS (Synchro nous Replicati on)	Maxim um Partitio ns
100 MB/s	3	c6_2 vCPUs   4 GB	High I/O	100,000	60,000	300
		c6_2 vCPUs   4 GB	Ultra- high I/O	100,000	80,000	300
300 MB/s	3	c6_4 vCPUs   8 GB	High I/O	300,000	150,000	900
		c6_4 vCPUs   8 GB	Ultra- high I/O	300,000	200,000	900
600 MB/s	4	c6_8 vCPUs   16 GB	Ultra- high I/O	600,000	300,000	1800
1200 MB/s	8	c6_8 vCPUs   16 GB	Ultra- high I/O	1.2 million	400,000	1800

Table 15-3 Kafka instance specifications

### 15.2.5 Do Kafka Instances Support Cross-Region Access?

Yes. You can access a Kafka instance across regions by using direct connections.

### 15.2.6 Do Kafka Instances Support Cross-Subnet Access?

Yes.

If the client and the instance are in the same VPC, cross-subnet access is supported.

If the client and the instance are in different VPCs, connect the two VPCs using VPC a peering connection. For details about VPC peering connection, see chapter "VPC Peering Connection" in the *Virtual Private Cloud User Guide*.

## 15.2.7 Why Can't I Access Kafka Using SSL from a Go Client?

Symptom: A Go client fails to connect to Kafka over SSL and the error message "first record does not look like a TLS handshake" is displayed.

**Solution:** Enable the TLS\_ECDHE\_RSA\_WITH\_AES\_128\_CBC\_SHA256 or TLS\_RSA\_WITH\_AES\_128\_CBC\_SHA256 cipher suite (both are disabled by default).

# 15.2.8 What If Certificates Fail to Be Loaded for SASL Connection?

If the certificate fails to be loaded during SASL connections, add the following parameters to the configuration file:

# Disable certificate domain name verification. ssl.endpoint.identification.algorithm=

## 15.2.9 Does DMS for Kafka Support Password-Free Access?

Yes. No password is required for accessing a Kafka instance with SASL disabled.

## 15.2.10 Obtaining Kafka Clients

Kafka premium instances are fully compatible with open-source clients. You can obtain **clients in other programming languages** and access your instance as instructed by the official Kafka website.

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

No.

# 15.2.12 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.

# 15.2.13 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.

# 15.2.14 Which TLS Version Does DMS for Kafka Support, 1.0, 1.1, or 1.2?

TLS 1.2.

# 15.2.15 Is There a Limit on the Number of Connections to a Kafka Instance?

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

- If the bandwidth is 100 MB/s, a maximum of 3000 connections are allowed.
- If the bandwidth is 300 MB/s, a maximum of 10,000 connections are allowed.

- If the bandwidth is 600 MB/s, a maximum of 20,000 connections are allowed.
- If the bandwidth is 1200 MB/s, a maximum of 20,000 connections are allowed.

# 15.2.16 How Many Connections Are Allowed from Each IP Address?

The Kafka server 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 Parameters**.

# **15.3 Topics and Partitions**

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

No, there is no limit on the topic quantity. However, there is an upper limit on the aggregate number of partitions of topics. After the partition limit is reached, you can no longer create topics.

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

Band width	Brok ers	Underlying Resource Type	I/O Туре	TPS (High- Throughput )	TPS (Synchro nous Replicati on)	Maxim um Partitio ns
100 MB/s	3	c6_2 vCPUs   4 GB	High I/O	100,000	60,000	300
		c6_2 vCPUs   4 GB	Ultra- high I/O	100,000	80,000	300
300 MB/s	3	c6_4 vCPUs   8 GB	High I/O	300,000	150,000	900
		c6_4 vCPUs   8 GB	Ultra- high I/O	300,000	200,000	900
600 MB/s	4	c6_8 vCPUs   16 GB	Ultra- high I/O	600,000	300,000	1800
1200 MB/s	8	c6_8 vCPUs   16 GB	Ultra- high I/O	1.2 million	400,000	1800

Table 15-4 Kafka instance specifications

# 15.3.2 Why Is Partition Quantity Limited?

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

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

Band width	Brok ers	Underlying Resource Type	I/О Туре	TPS (High- Throughput )	TPS (Synchro nous Replicati on)	Maxim um Partitio ns
100 MB/s	3	c6_2 vCPUs   4 GB	High I/O	100,000	60,000	300
		c6_2 vCPUs   4 GB	Ultra- high I/O	100,000	80,000	300
300 MB/s	3	c6_4 vCPUs   8 GB	High I/O	300,000	150,000	900
		c6_4 vCPUs   8 GB	Ultra- high I/O	300,000	200,000	900
600 MB/s	4	c6_8 vCPUs   16 GB	Ultra- high I/O	600,000	300,000	1800
1200 MB/s	8	c6_8 vCPUs   16 GB	Ultra- high I/O	1.2 million	400,000	1800

Table 15-5 Kafka instance specifications

## 15.3.3 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.

## 15.3.4 Why Do I Fail to Create Topics?

Possible cause: The aggregate number of partitions of created topics has reached the upper limit.

Solution: Scale up the instance or delete unnecessary topics.

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

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

# 15.3.6 Why Do Deleted Topics Still Exist?

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

To solve this problem, disable automatic topic creation.

## 15.3.7 Can I View Disk Space Used by a Topic?

Yes, you can view the disk space usage on the Cloud Eye console. On the **Queues** tab page, choose the target topic, and view the **Message Size** metric.

## 15.3.8 Can I Add ACL Permissions for Topics?

Yes. 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 Granting Permissions to a SASL\_SSL User.

### 15.3.9 How Do I Create a Topic?

You can create a topic by using one the following three methods:

By using the Kafka console

Create Topic				
Topic Name	topic-77229911			
Partitions	3       +       Value range: 1 to 100         Number of partitions in the topic. Messages in the topic will be distributed to these partitions to achieve scalability and fault tolerance.			
Replicas	3     +     Value range: 1 to 3       Number of message copies. This number is fixed once the topic is created.			
Aging Time (h)	72     +     Value range: 1 to 168       Time after which data in the topic expires.			
Synchronous Replication (?)				
Synchronous Flushing				

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

 $\times$ 

Kafka Manager	kafka_cluster	Cluster▼	Brokers	Topic 🗸	Preferred Replica Election	Reassign Partitions	Consumers
				Create			
				List			

#### • 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.

 If SASL is not enabled for the Kafka instance, run the following command 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:
  - i. Create the **ssl-user-config.properties** file in the **/config** directory of the Kafka client and add the SSL certificate configurations by referring to **Accessing a Kafka Instance with SASL**.
  - ii. Run the following command 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

# 15.3.10 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.

To solve this problem, you can shorten the aging time or expand the storage space.

### 15.3.11 What Can I Do If Disk Usage Is as High as 96%?

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

## 15.3.12 How Do I Increase the Partition Quantity?

You can increase the partition quantity by increasing the bandwidth.

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

# 15.3.13 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.

### 15.3.14 How Do I Disable Automatic Topic Creation?

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

Kat

Ca

In the Advanced Settings area on the Basic Information tab page, click next to Automatic Topic Creation, disable the option, and click .
 You can view the execution status of the task on the Background Tasks tab page.
 Advanced Settings

lvanced Settings				
ka SASL_SSL	Disabled	Automatic Topic Creation	?	Enabled
pacity Threshold Policy	Automatically delete 🖉			

# 15.3.15 Can I Delete Unnecessary Topics in a Consumer Group?

Yes. To delete a topic, simply unsubscribe from it on the Kafka client.

# 15.3.16 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.

**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.

#### 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.

# 15.3.17 Why Does an Instance Contain Default Topics \_\_trace and \_\_consumer\_offsets?

**Symptom:** Topics named **\_\_trace** and **\_\_consumer\_offsets** are found on Kafka Manager.

Topics Show 10 + entries							
Topic ↑↓	# Partitions ↑↓	# Brokers î↓	Brokers Spread				
consumer_offsets	50	3	100				
trace	9	3	100				
topic-01	3	3	100				

Handling method: \_\_trace and \_\_consumer\_offsets are preset topics in a Kafka instance. You are not advised to delete them. If they are deleted, the instance may become unavailable.

# **15.4 Consumer Groups**

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

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

# 15.4.2 Do I Need to Unsubscribe from a Topic Before Deleting a Consumer Group?

No. You can directly delete the consumer group.

# 15.4.3 Will a Consumer Group Without Active Consumers Be Deleted in 14 Days?

Yes.

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. The default value of **offsets.retention.minutes** is 20,160 minutes (14 days).

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.

# 15.4.4 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.

# 15.5 Messages

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

10 MB.

### 15.5.2 Why Does Message Poll Often Fail During Rebalancing?

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

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

This may happen in the following scenarios:

1. Heartbeat requests are not sent in time.

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

2. The interval between retrievals is too long.

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

request within the time specified by **max.poll.interval.ms**, the broker considers that the consumer is inactive and removes it from the consumer group, triggering rebalancing.

#### Solutions and Troubleshooting Methods

Scenario 1: Heartbeat requests are not sent in time.

**Solution**: 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.

### 15.5.3 Why Can't I Query Messages on the Console?

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

To solve the problem, 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.

To prevent this from happening, expand the disk capacity.

### 15.5.4 What Can I Do If Kafka Messages Are Accumulated?

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

**Solution**: Log in to Kafka Manager, find the consumer group where messages are accumulated, and check whether any consumer is retrieving messages in the consumer group. If yes, accelerate the retrieval on the service end. If no, delete unused consumer groups on the customer end.
## **15.5.5 Why Do Messages Still Exist After the Retention Period Elapses?**

**Symptom**: Messages still exist after reaching the retention period (for example, after 72 hours).

Possible cause 1: The segment files are in use.

**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.

#### 15.5.6 Do Kafka Instances Support Delayed Message Delivery?

No.

## 15.5.7 Does the Message Query Function on the Console Show the Replica Message Quantity?

No.

#### 15.5.8 How Do I View Accumulated Messages?

On the Kafka client, run the kafka-consumer-groups.sh --bootstrap-server {Kafka Connection Address} --describe --group {Consumer Group} command to view the accumulated messages in the consumer group.

#### **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 commands. For details about the configuration file consumer.properties, see the CLI access instructions provided in Accessing a Kafka Instance with SASL.
- For details on how to configure the Kafka client environment, see the CLI access instructions provided in Accessing a Kafka Instance Without SASL and Accessing a Kafka Instance with SASL.

## 15.5.9 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.

### 15.6 Kafka Manager

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

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

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



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

# 15.6.3 Yikes! Insufficient partition balance when creating topic : projectman\_project\_enterprise\_project Try again.

#### Symptom:

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

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

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

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

No. Kafka Manager does not support message body querying.

### 15.6.5 Can I Change the Port of the Kafka Manager Web UI?

No.

# 15.6.6 Which Topic Configurations Can Be Modified on Kafka Manager?

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

Perform the following steps to modify the topic configurations:

- 1. Log in to Kafka Manager.
- 2. Click kafka\_cluster.
- 3. Choose **Topic** > **List**.

	<b>&amp; Kafka Manager</b>	kafka_cluster	Cluster -	Brokers	Topic 🕶	Preferred Replica Election
Clusters / kafka	a_cluster / Summary				Create List	
Cluster In	formation					
Zookeepers	172. 7:2181 172.	37:2181 172	. 15:21	81/dms_clus	ter	
Version	1.0.0					

4. Click a topic whose configurations you want to modify.

#### 5. Click Update Config.

← topic-619913173							
Topic Summary		Operations	S				
Replication	3	Delete	Topic	Reassign Partitions	Gener	ate Partition Assignm	ients
Number of Partitions	3						
Sum of partition offsets	0	Add Partition	ns Upda	te Config	Manua	I Partition Assignme	nts
otal number of Brokers	3						
lumber of Brokers for Topic	3	Partitions	by Broker				
Preferred Replicas %	100	Broker	# of Partitions	# as Leader	Partitions	Skewed?	Lead
Irokers Skewed %	0	0	3	1	(0.1.2)	false	false
irokers Leader Skewed %	0	1	2	•	(0,1,2)	falco	falco
rokers Spread %	100	2	0	1	(0,1,2)	falco	falco
Jnder-replicated %	0	4	J	1	(0,1,2)	laise	laise

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

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

Cause: Kafka Manager has cache data.

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

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

Perform the following steps:

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

Kafka Manager 🛛 🔚 Kafka_duster Cluster 👻 Brokers	Topic ▼ Preferred Replica Election Reassign Partitions
Clusters / kafka_cluster / Summary	Create List
Cluster Information	
Version	2.2.0

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

#### Topics

Show 10 \$ entries

Topic ↑↓	# Partitions 1↓	# Brokers ↑↓	Brokers Spread % îl	Brokers Skew % ↑↓	Brokers Leader Skew %
consumer_offsets	50	3	100	0	0
trace	9	3	100	0	0
topic-test	3	3	100	0	0

Showing 1 to 3 of 3 entries

#### 4. Click Manual Partition Assignments.

#### Figure 15-1 Topic details

<ul> <li>topic-test</li> </ul>				
Topic Summary		Operations		
Replication	3	Delete Topic	Reassign Partitions	Generate Partition Assignments
Number of Partitions Sum of partition offsets	3	Add Partitions	Update Config	Manual Partition Assignments

#### Figure 15-2 Page for modifying partition leaders

► Manual   ■ Save Partitic	Partition Assignments n Assignment	5	
	Type to filte	r topics topic-test	
<b>P</b> a Ra Ra	artition 2 eplica 0: Broker 2 > eplica 1: Broker 1 > eplica 2: Broker 0 >	Partition 1 Replica 0: Broker 1 ✓ Replica 1: Broker 0 ✓ Replica 2: Broker 2 ✓	Partition 0 Replica 0: Broker 0 ✓ Replica 1: Broker 2 ✓ Replica 2: Broker 1 ✓

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

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

Manual Partition Assignments

Save Partition Assignment		
Type to fi	Iter topics	
	topic-test	
Partition 2	Partition 1	Partition 0
Replica 0: Broker 1 🗸	Replica 0: Broker 1 🗸	Replica 0: Broker 0 🗸
Replica 1: Broker 2 🗸	Replica 1: Broker 0 🗸	Replica 1: Broker 2 🗸
Replica 2: Broker 0 🗸	Replica 2: Broker 2 🗸	Replica 2: Broker 1 🗸

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

Manual Partitions Reassignment Successful

### Manual Partitions Reassignment Successful

Done!

Go to topic view.

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

<ul> <li>topic-test</li> </ul>				
Topic Summary		Operations		
Replication	3	Delete Topic	Reassign Partitions	Generate Partition Assignments
Number of Partitions	3	Add Partitions	Undate Config	Manual Partition Assignments
Sum of partition offsets				and the second s

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

Clusters / kafka\_cluster / Topics / topic-test / Run Reassign Partitions

### **Run Reassign Partitions - topic-test**

Done!		
Go to reassign partitions.		

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

Partition	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.

### 15.7 Monitoring & Alarm

### 15.7.1 Why Can't I View the Monitoring Data?

The possible causes are as follows:

**Partition Information** 

- The topic name starts with a special character, such as an underscore (\_) or a number sign (#).
- The consumer group name starts with a special character, such as an underscore (\_) or a number sign (#).

To solve the problem, delete topics and consumer groups whose names contain the special character.

15 FAQs

### 15.7.2 How Do I Configure Alarms for a Kafka Instance?

Configure alarm rules and handle alarms by referring to Table 15-6.

Metric	Normal Range	Alarm Policy	Approac h Upper Limit	Metric Description and Alarm Handling
Disk Capacity Usage	0-100	Alarm threshold: original value > 85 Number of consecutiv e periods: 3 Alarm severity: critical	Yes	Metric description: disk usage of the Kafka VM. Unit: % Alarm handling: When an alarm is generated for this metric, modify the instance storage space. For details, see <b>Modifying Instance</b> <b>Specifications</b> .
Average Load per CPU Core	0-2	Alarm threshold: original value > 1.5 Number of consecutiv e periods: 3 Alarm severity: major	Yes	Metric description: average load of each CPU core of the Kafka VM. Alarm handling: When an alarm is generated for this metric, modify the instance <b>bandwidth</b> . For details, see <b>Modifying Instance</b> <b>Specifications</b> .
Memory Usage	0-100	Alarm threshold: original value > 85 Number of consecutiv e periods: 3 Alarm severity: critical	Yes	Metric description: memory usage of the Kafka VM. Unit: % Alarm handling: When an alarm is generated for this metric, modify the instance <b>bandwidth</b> . For details, see <b>Modifying Instance</b> <b>Specifications</b> .

Table 15-6	Kafka	instance	metrics to	configure	alarm	rules	for
				5			

### 15.7.3 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.

# 15.7.4 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.



Date	Description
2022-08-12	This issue is the first official release.