Distributed Message Service for Kafka

User Guide

Issue 01

Date 2025-09-18





Copyright © Huawei Cloud Computing Technologies Co., Ltd. 2025. All rights reserved.

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

Trademarks and Permissions

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

Notice

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

i

Contents

1 Process of Using Katka	1
2 Permissions Management	3
2.1 Creating an IAM User and Granting DMS for Kafka Permissions	3
3 Buying a Kafka Instance	7
4 Configuring Topics	28
4.1 Creating a Kafka Topic	
4.2 Configuring Kafka Topic Permissions	34
4.3 Managing Topics	41
4.3.1 Viewing Kafka Topic Details	41
4.3.2 Viewing Kafka Topic Logs	45
4.3.3 Modifying Kafka Topic Configurations	48
4.3.4 Changing Kafka Partition Quantity	51
4.3.5 Modifying Kafka Topic Replicas	56
4.3.6 Exporting the Kafka Topic List	60
4.3.7 Reassigning Kafka Partitions	61
4.3.8 Configuring Automatic Topic Creation	73
4.3.9 Deleting a Kafka Topic	74
5 Connecting to an Instance	77
5.1 Configuring Kafka Network Connections	77
5.1.1 Kafka Network Connection Conditions	77
5.1.2 Configuring Kafka Public Access	79
5.1.3 Accessing Kafka Using a VPC Endpoint Across VPCs	87
5.1.4 Accessing Kafka in a Public Network Using DNAT	93
5.2 Configuring Kafka Access Control	98
5.2.1 Configuring Plaintext or Ciphertext Access to Kafka Instances	98
5.2.2 Generating and Replacing an SSL Kafka Certificate in JKS Format	103
5.2.3 Obtaining and Using An SSL Kafka Certificate in PEM Format	113
5.2.4 Configuring Mutual SSL Authentication for Kafka	114
5.2.5 Configuring Kafka ACL Users	122
5.3 Configuring the Kafka Client	126
5.3.1 Setting Parameters for Kafka Clients	126
5.3.2 Suggestions on Using the Kafka Client	133

5.4 Connecting to Kafka Using the Client (Plaintext Access)	135
5.5 Connecting to Kafka Using the Client (Ciphertext Access)	138
5.6 Connecting to Kafka on the Console	
6 Managing Messages	148
6.1 Viewing Kafka Messages	
6.2 Changing Kafka Message Retention Period	151
6.3 Deleting Kafka Messages	153
6.4 Diagnosing Kafka Message Accumulation	155
7 Managing Consumer Groups	159
7.1 Creating a Kafka Consumer Group	
7.2 Querying the Kafka Consumer Group List	161
7.3 Viewing Kafka Consumer Information	162
7.4 Viewing and Resetting Kafka Consumption Offsets	166
7.5 Viewing Kafka Rebalancing Logs	171
7.6 Modifying Kafka Consumer Group Description	
7.7 Configuring Automatic Consumer Group Creation for Kafka	174
7.8 Exporting Kafka Consumer Groups	175
7.9 Deleting a Kafka Consumer Group	175
7.10 Unsubscribing a Kafka Consumer Group from a Topic Topic	178
8 Managing Quotas	179
8.1 Configuring Kafka Quotas	179
8.2 Monitoring Kafka Quotas	184
9 Managing Instances	186
9.1 Viewing and Modifying Basic Information of a Kafka Instance	186
9.2 Viewing Kafka Disk Usage	192
9.3 Viewing Kafka Background Tasks	193
9.4 Viewing Sample Code of Kafka Production and Consumption	195
9.5 Modifying Kafka Instance Configuration Parameters	195
9.6 Configuring Kafka Instance Tags	201
9.7 Configuring Kafka Recycling Policies	204
9.8 Upgrading the Kafka Instance Kernel	206
9.9 Exporting the Kafka Instance List	209
9.10 Restarting a Kafka Instance	209
9.11 Deleting Kafka Instances	210
9.12 Using Kafka Manager	211
9.12.1 Accessing Kafka Manager	211
9.12.2 Resetting Kafka Manager Password	217
9.12.3 Restarting Kafka Manager	218
9.12.4 Disabling Kafka Manager	219
10 Modifying Instance Specifications	221

10.1 Modifying Cluster Kafka Instance Specifications	
10.2 Automatically Expanding the Storage Space of a Kafka Instance	229
11 Migrating Data	232
11.1 Kafka Data Migration Overview	232
11.2 Using MirrorMaker to Synchronize Data Across Clusters	238
11.3 Migrating Data Using Smart Connect	241
11.3.1 Enabling Smart Connect	241
11.3.2 Replicating Kafka Instance Data	243
11.3.3 Dumping Kafka Data to Object Storage Service (OBS)	251
11.3.4 Managing Smart Connect Tasks	253
11.3.5 Disabling Smart Connect	255
12 Testing Instance Performance	257
12.1 Kafka Production Rate and CPU Usage	
12.2 Kafka Instance TPS	
13 Applying for Increasing Kafka Quotas	278
14 Monitoring and Alarms	280
14.1 Viewing Kafka Metrics	280
14.2 Kafka Metrics	
14.3 Configuring a Kafka Alarm Rule	
15 Viewing Kafka Audit Logs	306

Process of Using Kafka

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

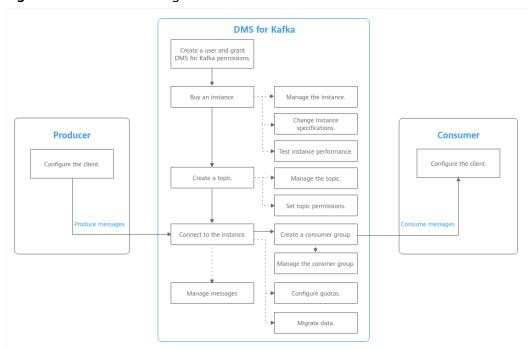


Figure 1-1 Process of using Kafka

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

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

2. Buying a Kafka Instance

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

3. Creating a Kafka Topic

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

4. Connecting to an Instance

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

2 Permissions Management

2.1 Creating an IAM User and Granting DMS for Kafka Permissions

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

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

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

This section describes the procedure for granting user permissions. **Figure 2-1** shows the process flow.

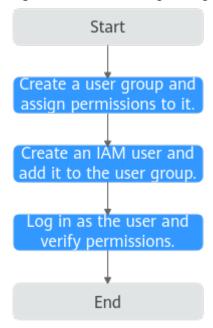
Prerequisites

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

DMS for Kafka permissions policies are based on DMS. Therefore, when assigning permissions for user groups, select DMS permissions policies.

Process Flow

Figure 2-1 Process for granting DMS for Kafka permissions



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

In the authorized region, perform the following operations:

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

Example Custom Policies

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

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

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

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

Example 1: Grant permission to delete and restart instances.

• Example 2: Grant permission to deny instance deletion.

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

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:

DMS for Kafka Resources

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

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

Table 2-1 DMS for Kafka resources and their paths

DMS for Kafka Request Conditions

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

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

Table 2-2 Predefined condition keys of DMS for Kafka

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

3 Buying a Kafka Instance

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

Preparing Instance Dependencies

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

Table 3-1 Kafka resources

Resource	Requirement	Operations
VPC and subnet	You need to configure a VPC and subnet for the Kafka instance as required. You can use the current account's existing VPC and subnet or shared ones, or create new ones. VPC owners can share the subnets in a VPC with one or multiple accounts through Resource Access Manager (RAM). Through VPC sharing, you can easily configure, operate, and manage multiple accounts' resources at low costs. For more information about VPC and subnet sharing, see VPC Sharing. Note when creating a VPC and a subnet: The VPC must be created in the same region as the Kafka instance. The Kafka instance supports IPv6 after it is enabled for the subnet. The instance with IPv6 enabled can be accessed on a client using IPv6 addresses.	For details on how to create a VPC and a subnet, see Creating a VPC and Subnet. If you need to create and use a new subnet in an existing VPC, see Creating a Subnet for an Existing VPC.
Security group	Different Kafka instances can use the same or different security groups. The security group must be in the same region as the Kafka instance. Before accessing a Kafka instance, configure security groups based on the access mode. For details, see Table 5-2.	For details on how to create a security group, see Creating a Security Group. For details on how to add rules to a security group, see Adding a Security Group Rule.

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

Notes and Constraints

- SASL_SSL cannot be manually configured for instances with IPv6 enabled.
- Ciphertext access and Smart Connect are unavailable for single-node instances.

Buying a Kafka instance

DMS for Kafka provides multiple options for you to purchase Kafka instances.

Table 3-2 Purchasing a Kafka instance

How to Purchase	Scenario
Quickly configuring a cluster Kafka instance	For a quick purchase, DMS for Kafka offers preconfigured instance specifications.
Customizing a single- node/cluster Kafka instance	For a standard purchase, you can customize a single- node or cluster Kafka instance as required.

Quick Config of a Cluster Kafka Instance

- **Step 1** Go to the **Buy Instance** page.
- **Step 2** Set basic instance configurations on the **Quick Config** page.

Parameter	Description	
Billing Mode	Yearly/Monthly is a prepaid mode. You need to pay first, and will be billed for your subscription period.	
	 Pay-per-use is a postpaid mode. You can pay after using the service, and will be billed for your usage duration. The fees are calculated in seconds and settled by hour. 	
Region	DMS for Kafka instances in different regions cannot communicate with each other over an intranet. Select a nearest location for low latency and fast access.	
AZ	An AZ is a physical region where resources use independent power supply and networks. AZs are physically isolated but interconnected through an internal network.	
	Select one AZ or at least three AZs. The AZ setting is fixed once the instance is created.	

Table 3-3 Basic instance configuration parameters

Step 3 Select the bundle.

Recommended: Select a preset DMS for Kafka bundle as required. Specify the
disk type and capacity as required. The disk type cannot be changed once
the Kafka instance is created.

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

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

The disk supports high I/O, ultra-high I/O, Extreme SSD, and General Purpose SSD types. For more information, see **Disk Types and Performance**.

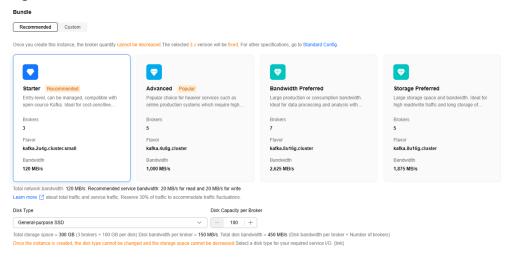
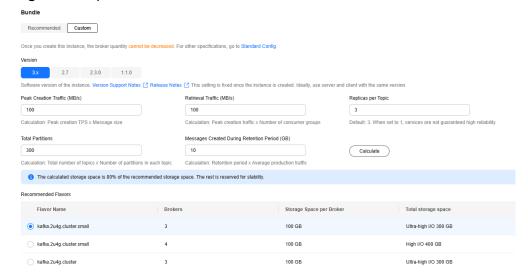


Figure 3-1 Recommended

 Custom: The system calculates Brokers and Storage Space per Broker, and provides Recommended Specifications based on your selected version and specified parameters: **Peak Creation Traffic**, **Retrieval Traffic**, **Replicas per Topic**, **Total Partitions**, and **Messages Created During Retention Period**.

Figure 3-2 Specification calculation



Step 4 Set the network information.

Table 3-4 Instance network parameters

Parameter	Description	
VPC	Select a created or shared VPC.	
	A VPC provides an isolated virtual network for your Kafka instances. You can configure and manage the network as required. To create a VPC, click Create VPC on the right. The Create VPC dialog box is displayed. For details, seeCreating a VPC.	
	After the Kafka instance is created, its VPC cannot be changed.	
Subnet	Select a created or shared subnet. To create a security group, click Create Subnet on the right. The Create Subnet dialog box is displayed. For details, see Creating a Subnet for an Existing VPC .	
	After the Kafka instance is created, its subnet cannot be changed.	
	The Kafka instance supports IPv6 after it is enabled for the subnet.	

Parameter	Description	
IPv6	This parameter is displayed after IPv6 is enabled for the subnet. The instance with IPv6 enabled can be accessed on a client using IPv6 addresses.	
	SASL_SSL cannot be manually configured for instances with IPv6 enabled.	
	The IPv6 setting is fixed once the instance is created.	
	Available in CN East2, CN South-Guangzhou, and CN East-Shanghai1 regions.	
Private IP	Select Auto or Manual .	
Addresses	Auto: The system automatically assigns an IP address from the subnet.	
	Manual: Select IP addresses from the drop-down list. If the number of selected IP addresses is less than the number of brokers, the remaining IP addresses will be automatically assigned.	
	This parameter is not displayed when IPv6 is enabled.	
Security Group	Select a created security group.	
	A security group is a set of rules for accessing a Kafka instance. You can click Create Security Group on the right. The Create Security Group dialog box is displayed. Set security group parameters by referring to Creating a Security Group .	
	Before accessing a Kafka instance on the client, configure security group rules based on the access mode. For details about security group rules, see Table 5-2 .	

Step 5 Configure the instance access mode.

Figure 3-3 Instance access mode

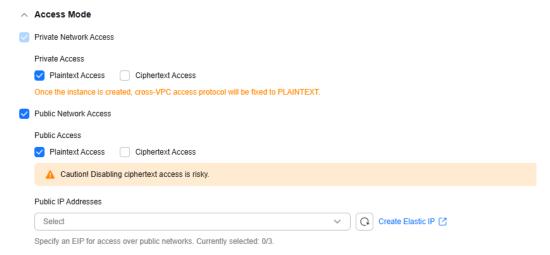


Table 3-5 Instance access mode parameters

Parameter	Sub- Parameter	Description
Private Network Access	Access Method	 There are two methods: Plaintext access: Clients connect to the Kafka instance without SASL authentication. Ciphertext access: Clients connect to the Kafka instance with SASL authentication. Enabling Ciphertext Access requires the Kafka security protocol, SSL username, password, and SASL PLAIN. Once enabled, private network access cannot be disabled. Enable plaintext or ciphertext access, or both. The value of the access mode determines that of the cross-VPC access protocol. The details are as follows: When Plaintext Access is enabled and Ciphertext Access is disabled, PLAINTEXT is used for Cross-VPC Access Protocol. When Ciphertext Access is enabled and Security Protocol is SASL_SSL, SASL_SSL is used for Cross-VPC Access Protocol. When Ciphertext Access is enabled and Security Protocol is SASL_PLAINTEXT, SASL_PLAINTEXT is used for Cross-VPC Access Protocol. Fixed once the instance is created.
Public Network Access	Access Method	 There are two methods: Plaintext access: Clients connect to the Kafka instance without SASL authentication. Ciphertext access: Clients connect to the Kafka instance with SASL authentication. Enabling Ciphertext Access requires the Kafka security protocol, SSL username, password, and SASL PLAIN. After public access is enabled, enable plaintext or ciphertext access, or both.

Parameter	Sub- Parameter	Description
	Public IP Addresses	Select the number of public IP addresses as required.
		If EIPs are insufficient, click Create Elastic IP to create EIPs. Then, return to the Kafka
		instance purchase page and click onext to Public IP Address to refresh the public IP address list.
		Kafka instances only support IPv4 EIPs.

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

Table 3-6 Ciphertext access parameters

Parameter	Value	Description
Security Protocol	SASL_SSL	SASL is used for authentication. Data is encrypted with SSL certificates for high-security transmission.
	SASL_PLAINTEX T	SASL is used for authentication. Data is transmitted in plaintext for high performance.
		SCRAM-SHA-512 authentication is recommended for plaintext transmission.
SSL Username	-	Username for a client to connect to a Kafka instance.
		A username should contain 4 to 64 characters, start with a letter, and contain only letters, digits, hyphens (-), and underscores (_).
		The username cannot be changed once ciphertext access is enabled.

Parameter	Value	Description
Password	-	Password for a client to connect to a Kafka instance.
		A password must meet the following requirements:
		Contains 8 to 32 characters.
		 Contains at least three types of the following characters: uppercase letters, lowercase letters, digits, and special characters `~!@#\$%^&*()=+\ [{}];:'",<.>? and spaces, and cannot start with a hyphen (-).
		 Cannot be the username spelled forward or backward.
SASL Mechanism	-	If PLAIN is disabled, the SCRAM-SHA-512 mechanism is used for username and password authentication.
		 If PLAIN is enabled, both the SCRAM- SHA-512 and PLAIN mechanisms are supported. You can select either of them as required.
		The SASL/PLAIN setting cannot be changed once ciphertext access is enabled.
		What are SCRAM-SHA-512 and PLAIN mechanisms?
		 SCRAM-SHA-512: uses the hash algorithm to generate credentials for usernames and passwords to verify identities. SCRAM- SHA-512 is more secure than PLAIN.
		PLAIN: a simple username and password verification mechanism.

Step 6 Configure advanced settings.

Table 3-7 Advanced configuration parameters

Parameter	Description	
Instance Name	You can customize a name that complies with the rules: 4–64 characters; starts with a letter; can contain only letters, digits, hyphens (-), and underscores (_).	
Enterprise Project	This parameter is for enterprise users. Enterprise projects facilitate project-level management and grouping of cloud resources and users. The default project is default .	

Parameter	Description	
Capacity Threshold Policy	Specify how messages are processed when the disk usage threshold (95%) is reached.	
	• Automatically delete: Messages can be produced and consumed, 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 produced, but existing messages can still be consumed. This policy is suitable for scenarios where no data loss can be tolerated.	
Smart Connect	Configure Smart Connect.	
	Smart Connect is used for data synchronization between heterogeneous systems. You can configure Smart Connect tasks to synchronize data between Kafka and another cloud service or between two Kafka instances.	
	Enabling Smart Connect creates two brokers.	
Automatic Topic	Enable automatic Kafka topic creation if needed.	
Creation	If this option is enabled, a topic will be automatically created when a message is produced in or consumed from a topic that does not exist. The default topic parameters are listed in Table 3-8 .	
	 For cluster instances, after you change the value of the log.retention.hours (retention period), default.replication.factor (replica quantity), or num.partitions (partition quantity) parameter, the value will be used in later topics that are automatically created. For example, assume that num.partitions is changed to 5, an automatically created topic has parameters listed in Table 3-8. Unavailable for single-node instances. 	

Parameter	Description	
Tags	Tags are used to identify cloud resources. When you have multiple cloud resources of the same type, you can use tags to classify them based on usage, owner, or environment.	
	If your organization has configured tag policies for DMS for Kafka, add tags to Kafka instances based on the policies. If a tag does not comply with the policies, Kafka instance creation may fail. Contact your organization administrator to learn more about tag policies.	
	• If you have predefined tags, select a predefined pair of tag key and value. You can click Create predefined tags to go to the Tag Management Service (TMS) console and view or create tags.	
	You can also create new tags by specifying Tag key and Tag value .	
	Up to 20 tags can be added to each Kafka instance. For details about the requirements on tags, see Configuring Kafka Instance Tags .	
Description	Enter a Description of the instance for 0–1024 characters.	

Table 3-8 Topic parameters

Parameter	Default Value (Single-node)	Default Value (Cluster)	Modified To (Cluster)
Partitions	1	3	5
Replicas	1	3	3
Aging Time (h)	72	72	72
Synchronous Replication	Disabled	Disabled	Disabled
Synchronous Flushing	Disabled	Disabled	Disabled
Message Timestamp	CreateTime	CreateTime	CreateTime
Max. Message Size (bytes)	10,485,760	10,485,760	10,485,760

Step 7 Specify the required duration.

This parameter is displayed only if the billing mode is yearly/monthly. If **Autorenew** is selected, the instance will be renewed automatically.

• Monthly subscriptions auto-renew for 1 month every time.

- Yearly subscriptions auto-renew for 1 year every time.
- Step 8 Click Confirm.
- **Step 9** Confirm the instance information, and read and agree to the *Huawei Cloud Customer Agreement*. If you have selected the yearly/monthly billing mode, click **Pay Now** and make the payment as prompted. If you have selected the pay-peruse mode, click **Submit**.
- **Step 10** Return to the instance list and check whether the Kafka instance has been created.

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

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

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

----End

Standard Config of a Single-node/Cluster Kafka Instance

- **Step 1** Go to the **Buy Instance** page.
- **Step 2** Set basic instance configurations on the **Standard Config** page.

Table 3-9 Basic instance configuration parameters

Parameter	Description	
Billing Mode	 Yearly/Monthly is a prepaid mode. You need to pay first, and will be billed for your subscription period. 	
	Pay-per-use is a postpaid mode. You can pay after using the service, and will be billed for your usage duration. The fees are calculated in seconds and settled by hour.	
Region	DMS for Kafka instances in different regions cannot communicate with each other over an intranet. Select a nearest location for low latency and fast access.	
AZ	An AZ is a physical region where resources use independent power supply and networks. AZs are physically isolated but interconnected through an internal network.	
	Select one AZ or at least three AZs. The AZ setting is fixed once the instance is created.	

Step 3 Configure the following instance specifications:

 Table 3-10 Instance specifications parameters

Parameter	Description	
Version	Kafka version, which can be 1.1.0, 2.7, or 3.x.	
	The version is fixed once the instance is created.	
Architecture	Select Single-node or Cluster as required.	
	Single-node instances are available only in v2.7. See Comparing Single-node and Cluster Kafka Instances.	
Broker Flavor	Select a broker flavor as required.	
	Maximum number of partitions per broker × Number of brokers = Maximum number of partitions of an instance. If the total number of partitions of all topics exceeds the upper limit of partitions, topic creation fails.	
Brokers	Specify the broker quantity.	
Disk Type	Select the disk type for Kafka data storage. The disk type cannot be changed once the Kafka instance is created.	
Disk Capacity per	Specify the disk size for Kafka data storage.	
Broker	The storage space is consumed by message replicas, logs, and metadata. Specify the storage space based on the expected service message size, the number of replicas, and the reserved disk space. Each Kafka broker reserves 33 GB disk space for storing logs and metadata.	
	Disks are formatted when an instance is created. As a result, the actual available disk space is 93% to 95% of the total disk space.	
	The disk supports high I/O, ultra-high I/O, Extreme SSD, and General Purpose SSD types. For more information, see Disk Types and Performance.	

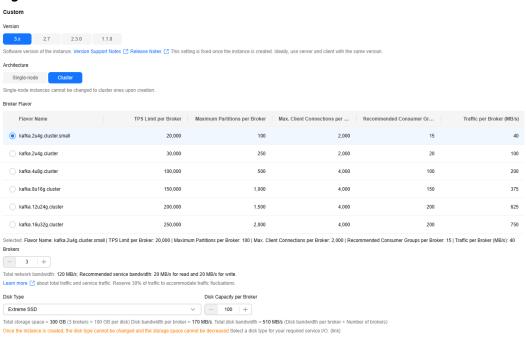


Figure 3-4 Instance flavor

Step 4 Set the network information.

Table 3-11 Instance network parameters

Parameter	Description	
VPC	Select a created or shared VPC.	
	A VPC provides an isolated virtual network for your Kafka instances. You can configure and manage the network as required. To create a VPC, click Create VPC on the right. The Create VPC dialog box is displayed. For details, see Creating a VPC .	
	After the Kafka instance is created, its VPC cannot be changed.	
Subnet	Select a created or shared subnet. To create a security group, click Create Subnet on the right. The Create Subnet dialog box is displayed. For details, see Creating a Subnet for an Existing VPC .	
	After the Kafka instance is created, its subnet cannot be changed.	
	The Kafka instance supports IPv6 after it is enabled for the subnet.	

Parameter	Description		
IPv6	This parameter is displayed after IPv6 is enabled for the subnet. The instance with IPv6 enabled can be accessed on a client using IPv6 addresses.		
	SASL_SSL cannot be manually configured for instances with IPv6 enabled.		
	The IPv6 setting is fixed once the instance is created.		
	Available in CN East2, CN South-Guangzhou, and CN East-Shanghai1 regions.		
Private IP	Select Auto or Manual .		
Addresses	Auto: The system automatically assigns an IP address from the subnet.		
	Manual: Select IP addresses from the drop-down list. If the number of selected IP addresses is less than the number of brokers, the remaining IP addresses will be automatically assigned.		
	This parameter is not displayed when IPv6 is enabled.		
Security Group	Select a created security group. A security group is a set of rules for accessing a Kafka instance. You can click Create Security Group on the right. The Create Security Group dialog box is displayed. Set security group parameters by referring to Creating a Security Group . Before accessing a Kafka instance on the client, configure security group rules based on the access mode. For details about security group rules, see Table 5-2 .		

Step 5 Configure the instance access mode.

Figure 3-5 Instance access mode

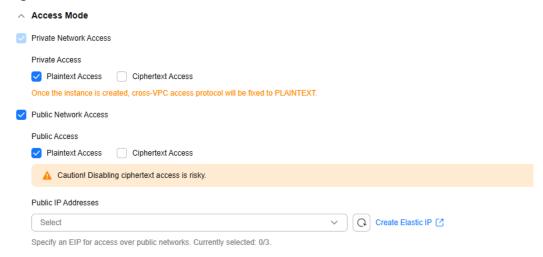


Table 3-12 Instance access mode parameters

Parameter	Sub- Parameter	Description
Private Network Access	Access Method	 There are two methods: Plaintext access: Clients connect to the Kafka instance without SASL authentication. Ciphertext access: Clients connect to the Kafka instance with SASL authentication. Enabling Ciphertext Access requires the Kafka security protocol, SSL username, password, and SASL PLAIN. Once enabled, private network access cannot be disabled. Enable plaintext or ciphertext access, or both. Ciphertext access is unavailable for single-node instances. The value of the access mode determines that of the cross-VPC access protocol. The details are as follows: When Plaintext Access is enabled and Ciphertext Access is disabled, PLAINTEXT is used for Cross-VPC Access Protocol. When Ciphertext Access is enabled and Security Protocol is SASL_SSL, SASL_SSL is used for Cross-VPC Access Protocol. When Ciphertext Access is enabled and Security Protocol is SASL_PLAINTEXT, SASL_PLAINTEXT is used for Cross-VPC Access Protocol. Fixed once the instance is created.
Public Network Access	Access Method	 There are two methods: Plaintext access: Clients connect to the Kafka instance without SASL authentication. Ciphertext access: Clients connect to the Kafka instance with SASL authentication. Enabling Ciphertext Access requires the Kafka security protocol, SSL username, password, and SASL PLAIN. After public access is enabled, enable plaintext or ciphertext access, or both. Ciphertext access is unavailable for single-node instances.

Parameter	Sub- Parameter	Description
	Public IP Addresses	Select the number of public IP addresses as required.
		If EIPs are insufficient, click Create Elastic IP to create EIPs. Then, return to the Kafka console and click C next to Public IP Address
		to refresh the public IP address list. Kafka instances only support IPv4 EIPs.

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

Table 3-13 Ciphertext access parameters

Parameter	Value	Description
Security Protocol	SASL_SSL	SASL is used for authentication. Data is encrypted with SSL certificates for high-security transmission.
	SASL_PLAINTEX T	SASL is used for authentication. Data is transmitted in plaintext for high performance.
		SCRAM-SHA-512 authentication is recommended for plaintext transmission.
SSL Username	-	Username for a client to connect to a Kafka instance.
		A username should contain 4 to 64 characters, start with a letter, and contain only letters, digits, hyphens (-), and underscores (_).
		The username cannot be changed once ciphertext access is enabled.

Parameter	Value	Description
Password	-	Password for a client to connect to a Kafka instance.
		A password must meet the following requirements:
		Contains 8 to 32 characters.
		• Contains at least three types of the following characters: uppercase letters, lowercase letters, digits, and special characters `~!@#\$%^&*()=+\ [{}];:'',<.>? and spaces, and cannot start with a hyphen (-).
		Cannot be the username spelled forward or backward.
SASL Mechanism	-	If PLAIN is disabled, the SCRAM-SHA-512 mechanism is used for username and password authentication.
		If PLAIN is enabled, both the SCRAM-SHA-512 and PLAIN mechanisms are supported. You can select either of them as required.
		The SASL/PLAIN setting cannot be changed once ciphertext access is enabled.
		What are SCRAM-SHA-512 and PLAIN mechanisms?
		 SCRAM-SHA-512: uses the hash algorithm to generate credentials for usernames and passwords to verify identities. SCRAM- SHA-512 is more secure than PLAIN.
		PLAIN: a simple username and password verification mechanism.

Step 6 Configure advanced settings.

Table 3-14 Advanced configuration parameters

Parameter	Description
Instance Name	You can customize a name that complies with the rules: 4–64 characters; starts with a letter; can contain only letters, digits, hyphens (-), and underscores (_).
Enterprise Project	This parameter is for enterprise users. Enterprise projects facilitate project-level management and grouping of cloud resources and users. The default project is default .

Parameter	Description
Capacity Threshold Policy	Specify how messages are processed when the disk usage threshold (95%) is reached.
	Automatically delete: Messages can be produced and consumed, 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 produced, but existing messages can still be consumed. This policy is suitable for scenarios where no data loss can be tolerated.
Smart Connect	Configure Smart Connect .
	Smart Connect is used for data synchronization between heterogeneous systems. You can configure Smart Connect tasks to synchronize data between Kafka and another cloud service or between two Kafka instances.
	Enabling Smart Connect creates two brokers.
	Single-node instances do not have this parameter.
Automatic Topic	Enable automatic Kafka topic creation if needed.
Creation	If this option is enabled, a topic will be automatically created when a message is produced in or consumed from a topic that does not exist. The default topic parameters are listed in Table 3-15 .
	Unavailable for single-node instances.
Tags	Tags are used to identify cloud resources. When you have multiple cloud resources of the same type, you can use tags to classify them based on usage, owner, or environment.
	If your organization has configured tag policies for DMS for Kafka, add tags to Kafka instances based on the policies. If a tag does not comply with the policies, Kafka instance creation may fail. Contact your organization administrator to learn more about tag policies.
	 If you have predefined tags, select a predefined pair of tag key and value. You can click View predefined tags to go to the Tag Management Service (TMS) console and view or create tags.
	 You can also create new tags by specifying Tag key and Tag value.
	Up to 20 tags can be added to each Kafka instance. For details about the requirements on tags, see Configuring Kafka Instance Tags .
Description	Enter a Description of the instance for 0–1024 characters.

Table 3 13 Topic parameters			
Parameter	Default Value (Single-node)	Default Value (Cluster)	Modified To (Cluster)
Partitions	1	3	5
Replicas	1	3	3
Aging Time (h)	72	72	72
Synchronous Replication	Disabled	Disabled	Disabled
Synchronous Flushing	Disabled	Disabled	Disabled
Message Timestamp	CreateTime	CreateTime	CreateTime
Max. Message Size (bytes)	10,485,760	10,485,760	10,485,760

Table 3-15 Topic parameters

Step 7 Specify the required duration.

This parameter is displayed only if the billing mode is yearly/monthly. If **Autorenew** is selected, the instance will be renewed automatically.

- Monthly subscriptions auto-renew for 1 month every time.
- Yearly subscriptions auto-renew for 1 year every time.
- **Step 8** In **Summary** on the right, view the selected instance configuration.
- Step 9 Click Confirm.
- Step 10 Confirm the instance information, and read and agree to the *Huawei Cloud Customer Agreement*. If you have selected the yearly/monthly billing mode, click Pay Now and make the payment as prompted. If you have selected the pay-per-use mode, click Submit.
- **Step 11** Return to the instance list and check whether the Kafka instance has been created.

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

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

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

----End

Purchasing a Kafka Instance with Same Configurations

To purchase another Kafka instance with the same configuration as the current one, reuse the current configuration through the **Buy Another** function.

- Step 1 Go to the Kafka console.
- **Step 2** Click on the upper left corner and select a region.

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

- **Step 3** Select a target Kafka instance and choose **More** > **Buy Another** in the **Operation** column.
- **Step 4** Adjust the automatically replicated parameter settings as required. For details, see **Standard Config of a Single-node/Cluster Kafka Instance**.

For security purposes, parameter settings involved in the following scenarios will not be replicated and a re-configuration is required:

- SSL username and password of a Kafka instance with ciphertext access enabled.
- Public IP addresses of a Kafka instance with public access enabled.
- Manually assigned IP addresses of a Kafka instance. Select Manual for Private IP Addresses again and specify private IP addresses.
- Name of a target Kafka instance.
- **Step 5** In **Summary** on the right, view the selected instance configuration.
- Step 6 Click Confirm.
- **Step 7** Confirm the instance information, and read and agree to the *Huawei Cloud Customer Agreement*. If you have selected the yearly/monthly billing mode, click **Pay Now** and make the payment as prompted. If you have selected the pay-peruse mode, click **Submit**.
- **Step 8** Return to the instance list and check whether the Kafka instance has been created.

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

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

□ NOTE

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

----End

Related Document

To purchase a Kafka instance by calling an API, see Creating a Kafka Instance.

4 Configuring Topics

4.1 Creating a Kafka Topic

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

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

The following parameters of cluster instances can be changed on the **Instance** > **Parameters** page: **log.retention.hours** (retention period), **default.replication.factor** (replica quantity), or **num.partitions** (partition quantity). The value will be used in later topics that are automatically created.

For example, assume that **num.partitions** is changed to **5**, an automatically created topic has parameters listed in **Table 4-1**.

Table 4-1 Topic parameters

Parameter	Default Value (Single-node)	Default Value (Cluster)	Modified To (Cluster)
Partitions	1	3	5
Replicas	1	3	3
Aging Time (h)	72	72	72
Synchronous Replication	Disabled	Disabled	Disabled
Synchronous Flushing	Disabled	Disabled	Disabled
Message Timestamp	CreateTime	CreateTime	CreateTime

Parameter	Default Value	Default Value	Modified To
	(Single-node)	(Cluster)	(Cluster)
Max. Message Size (bytes)	10,485,760	10,485,760	10,485,760

Notes and Constraints

- The partition quantity of topics of a single-node or cluster Kafka instance is limited. When the partition quantity limit is reached, you can no longer create topics. The partition quantity of a topic varies. For details, see Kafka Instance Specifications.
- Instances created since May 17, 2023 do not have Kafka Manager. You cannot create topics for these instances using Kafka Manager.
- For an instance with ciphertext access enabled, if allow.everyone.if.no.acl.found is set to false, topics can be created on the client only for the initial user (set when ciphertext access is enabled for the first time).
- If a topic name starts with a special character, for example, a number sign (#), monitoring data cannot be displayed.
- Due to the limitation of the Kafka kernel, topics whose names contain only
 period or underscore difference cannot be created. For example, assume that
 the Topic_1 topic is created, creating a topic named Topic.1 will fail and
 throw the Topic 'topic.1' collides with existing topics: topic_1 exception.

Creating a Kafka Topic

You can create a topic on the Kafka console, using Kafka Manager, or on a client.

Creating a Topic on the Console

- **Step 1** Log in to the **Kafka console**.
- **Step 2** Click in the upper left corner to select the region where your instance is located.
- **Step 3** Click the desired Kafka instance to view the instance details.
- **Step 4** In the navigation pane, choose **Instance** > **Topics**. Then click **Create Topic**.
- **Step 5** Enter a topic name, specify other parameters, and click **OK**.

Figure 4-1 Creating a topic (cluster instance)

Create Topic

Topic Name
topic-992096861
Default format (can be customised): Topic-Date-Four random characters (letters and digits)
Partitions ② - 3 + Cannot be decreased once set. Range: 1 to 200.
Replicas ③ - 3
Aging Time (h) ② - 72 + How long messages will be preserved in the topic. Older messages will be deleted and cannot be consumed. Range: 1 to 720.
 Synchronous Replication The topic replicates messages to all replicas before responding to the client. acks must be set to all or -1 on the client. Synchronous Flushing Enabled: Messages are written to the disk in real time for high reliability. Disabled: Messages are stored in the memory first.
Message Timestamp ②
LogAppendTime CreateTime
When the broker appends a message to the log.
Max. Message Size (bytes) ②
_ 10,485,760 +
Description (Optional)
Enter a description.

Table 4-2 Topic parameters

Parameter	Description
Topic Name	Customize a name that contains 3 to 200 characters, starts with a letter or underscore (_), and contains only letters, digits, periods (.), hyphens (-), and underscores (_).
	The name must be different from preset topics:
	consumer_offsets
	transaction_state
	•trace
	connect-status
	connect-configs
	connect-offsets
	cloud_event
	site_event
	dms_dial_test
	remote_log_metadata
	remote_log_compact_metadata
	remote_log_metadata_offset
	Once the topic is created, you cannot modify its name.
	Due to the limitation of the Kafka kernel, topics whose names contain only period or underscore difference cannot be created. For example, assume that the Topic_1 topic is created, creating a topic named Topic.1 will fail and throw the Topic 'topic.1' collides with existing topics: topic_1 exception.
Partitions	Number of partitions in the topic.
	If the number of partitions is the same as that of consumers, the larger the partitions, the higher the consumption concurrency.
	If this parameter is set to 1 , messages will be retrieved in the FIFO order.
	Value range: 1–200
Replicas	A higher number of replicas delivers higher reliability. Data is automatically backed up on each replica. When one Kafka broker becomes faulty, data is still available on other brokers. For more information, see Basic Concepts .
	If this parameter is set to 1, only one set of data is available.
	Value range: 1 to number of brokers
	NOTE If an instance node is faulty, an internal service error may be reported when you query messages in a topic with only one replica. Therefore, you are not advised using a topic with only one replica.

Parameter	Description
Aging Time (h)	The period that messages are retained for. Consumers must retrieve messages before this period ends. Otherwise, the messages will be deleted and can no longer be consumed.
	Value range: 1–720
Synchronous Replication	A message is returned to the client only after the message creation request has been received and the message has been acknowledged by all replicas.
	After enabling this, set the parameter acks to all or -1 in the configuration file or production code on the producer client.
	If there is only one replica, synchronous replication cannot be enabled.
Synchronous Flushing	A message is immediately flushed to disk once it is produced, bringing higher reliability. When this option is disabled, a message is stored in the memory instead of being immediately flushed to disk once produced.
Message	Timestamp type of a message. Options:
Timestamp	CreateTime: time when the producer created the message.
	LogAppendTime: time when the broker appended the message to the log.
Max. Message Size (bytes)	Maximum batch processing size allowed by Kafka. If message compression is enabled in the client configuration file or code of producers, this parameter indicates the size after compression.
	If this is increased and there are consumers older than 0.10.2, the consumers' fetch size must also be increased so that they can fetch record batches this large.
	Value range: 0 to 10,485,760
Description	0–200 characters.

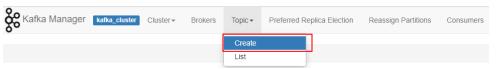
After a topic is created, view the new topic on the topic list page.

----End

Creating a Topic on Kafka Manager

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

Figure 4-2 Creating a topic on Kafka Manager



Creating a Topic on the Client

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

 For a Kafka instance with ciphertext access disabled, run the following command in the /bin directory of the Kafka client:

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

Table 4-3 Topic creation parameters

Parameter	Description
topic-name	topic name, which can be customized.
connection-address	Connection address of a Kafka instance. To obtain the address, choose Overview > Connection .
number-of-partitions	Number of partitions in the topic. To ensure performance, 200 or fewer partitions are recommended for each topic.
number-of-replicas	Number of replicas of a topic.

Example:

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

- For a Kafka instance with ciphertext access enabled, do as follows:
 - a. (Optional) If the username and password, and the SSL certificate has been configured, skip this step and go to b. Otherwise, do as follows: Create the ssl-user-config.properties file in the /config directory of the Kafka client. Add the username and password, and the SSL certificate configuration by referring to Step 3.
 - b. Run the following command in the **/bin** directory of the Kafka client: ./kafka-topics.sh --create --topic *{topic-name}* --bootstrap-server *{connection-address}* -- partitions *{number-of-partitions}* --replication-factor *{number-of-replicas}* --command-config ./ config/*{ssl-user-config.properties}*}

Table 4-4 Topic creation parameters

Parameter	Description
topic-name	topic name, which can be customized.
connection-address	Connection address of a Kafka instance. To obtain the address, choose Overview > Connection .

Parameter	Description
number-of- partitions	Number of partitions in the topic. To ensure performance, 200 or fewer partitions are recommended for each topic.
number-of-replicas	Number of replicas of a topic.
ssl-user- config.properties	Configuration file name. This file contains username, password, and SSL certificate information.

Example:

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

Related Documents

- To free from manually creating topics, enable the automatic topic creation function. For details, see Configuring Automatic Topic Creation.
- To create a topic by calling an API, see Creating a Topic for a Kafka Instance.

4.2 Configuring Kafka Topic Permissions

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

This section describes how to grant topic permissions to users after ciphertext access is enabled for a Kafka instance.

Notes and Constraints

- If parameter allow.everyone.if.no.acl.found is set to true and no topic is granted for a user, all users can subscribe to or publish messages to the topic. If permissions for a topic have been granted to one or more users, only these users can subscribe to or publish messages to the topic. The value of allow.everyone.if.no.acl.found can be modified.
- If allow.everyone.if.no.acl.found is set to false, only the initial user (set when ciphertext access is enabled for the first time) and other authorized users have the permission to subscribe to or publish messages to topics. The value of allow.everyone.if.no.acl.found can be modified.
- If both the default and individual user permissions are configured for a topic, the union of the permissions is used.
- Unavailable for single-node instances.
- Setting topic permissions in batches overwrites the previous permission settings.

 Permissions may be temporarily invalid during the configuration, throwing client error message "AuthorizationException". In this case, set a retry mechanism on the client. For details, see Suggestions on Using the Kafka Client.

Prerequisites

- Ciphertext has been enabled for the Kafka instance.
- A user is created.

Viewing Permissions of a Topic

- **Step 1** Log in to the **Kafka console**.
- **Step 2** Click on the upper left corner to select the region where your instance is located.
- **Step 3** Click the desired Kafka instance to go to the instance details page.
- **Step 4** In the navigation pane, choose **Instance** > **Topics**.
- **Step 5** Click a topic name to go to the topic details page.
- Step 6 Choose User Permissions.
- **Step 7** View all authorized users of the topic and their permissions.

Figure 4-3 User permissions of a topic

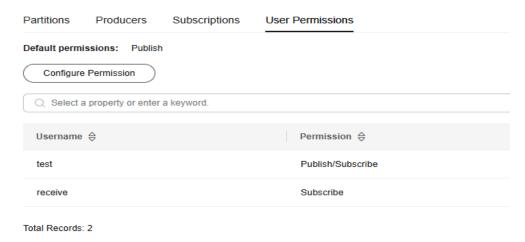


Table 4-5 User permissions

Parameter	Description
Username	Users who have the publish or subscribe permissions
Permission	Permissions of the user.

----End

Set Topic Permissions

On the console, the publish and subscribe permissions can be granted to an SASL user. The permissions can be granted for a single or multiple topics at a time.

Setting Permissions for a Topic

- **Step 1** Log in to the **Kafka console**.
- **Step 2** Click in the upper left corner to select the region where your instance is located.
- **Step 3** Click the desired Kafka instance to view the instance details.
- **Step 4** In the navigation pane, choose **Instance** > **Topics**.
- **Step 5** In the row containing the desired topic, click **Grant User Permission**.
- **Step 6** Go to the user permission page in either of the following ways:
 - In the row containing the desired topic, click **Grant User Permission**.
 - Click the desired topic name to go to the topic details page. Click **Configure Permission** in the upper right corner.
 - Click the desired topic name to go to the topic details page. Choose the **User Permissions** tab. Click **Configure Permission**.
- **Step 7** Grant topic permissions to users.
 - To grant the same permissions to all users, select **Default permissions** and then select permissions. As shown in the following figure, all users have the permission to publish messages to this topic.

Figure 4-4 Granting the same permissions to all users



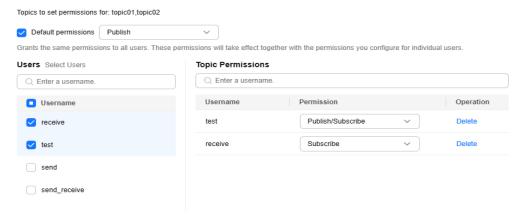
• To grant different permissions to different users, do not select **Default permissions**. In the **Users** area of the **Grant User Permission** dialog box, select target users. If there are many users, enter the username in the search box for a quick search. In the **Topic Permissions** area, configure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

Users Select Users Topic Permissions Q Enter a username Q Enter a username Username Permission Operation Username Publish Delete receive send test test Publish/Subscribe Delete Subscribe Delete send send_receive

Figure 4-5 Granting permissions to individual users

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

Figure 4-6 Granting topic permissions to users



Step 8 At the bottom of the **User Permissions** dialog box, click **Auto Enter**. The system will automatically enter **MODIFY** in the text box. Click **OK**.

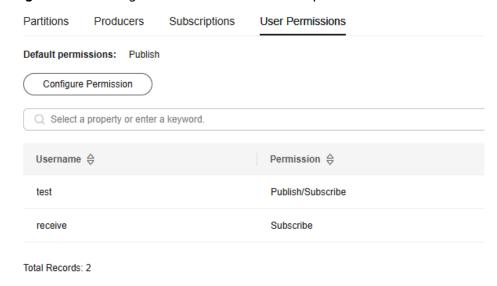
X **User Permissions** Grants the same permissions to all users. These permissions will take effect together with the permissions you comigure for individual users Users Select Users **Topic Permissions** Q Enter a username. Q Enter a username Username Username Permission Operation receive Publish/Subscribe Delete Subscribe test receive send send_receive 10 🗸 🕻 1 > 10 ∨ (1 > Total Records: 2 To confirm, enter MODIFY. Auto Enter MODIFY Cancel

Figure 4-7 Confirming the permission settings

Step 9 Verify whether the permissions are correct.

- 1. Click the desired topic name to go to the topic details page.
- 2. Choose the **User Permissions** tab.
- 3. View the configured user permissions.

Figure 4-8 Viewing authorized users and their permissions



----End

Setting Topic Permissions in Batches

- Step 1 Log in to the Kafka console.
- **Step 2** Click in the upper left corner to select the region where your instance is located.
- **Step 3** Click the desired Kafka instance to view the instance details.
- **Step 4** In the navigation pane, choose **Instance** > **Topics**.
- **Step 5** Select the topics to be configured with user permissions and click **Grant User Permission**.
- **Step 6** Set topic permissions in batches.

The permissions already set for a user are not displayed. Setting permissions in batches overwrites the previous permission settings. For example, user test already has the Publish/Subscribe permission on Topic01. When the Publish permission is set for the user in a batch permission setting, the user only has the Publish permission on Topic01.

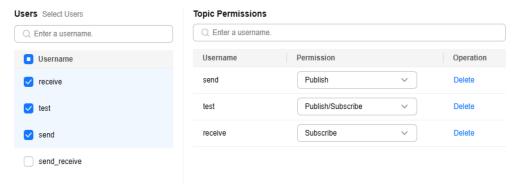
• To grant the same permissions to all users, select **Default permissions** and then select permissions. As shown in the following figure, all users have the permission to publish messages to topic01 and 02.

Figure 4-9 Granting the same permissions to all users



To grant different permissions to different users, do not select **Default permissions**. In the **Users** area of the **Grant User Permission** dialog box, select target users. If there are many users, enter the username in the search box for a quick search. In the **Topic Permissions** area, configure different permissions for the users. As shown in the following figure, only the **test**, **send**, and **receive** users can subscribe to or publish messages to topic01 and 02. The **send_receive** user cannot subscribe to or publish messages to these topics.

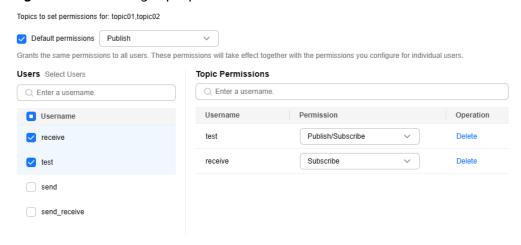
Figure 4-10 Granting permissions to multiple users



If both the default and individual user permissions are configured for a topic, the union of the permissions is used. As shown in the following figure, the test

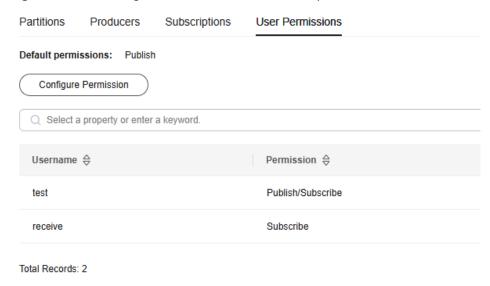
and **receive** users can subscribe to and publish messages to topic01 and 02, while other users can only publish messages to them.

Figure 4-11 Granting topic permissions to users



- **Step 7** At the bottom of the **Set Permissions** dialog box, click **Auto Enter**. The system will automatically enter **MODIFY** in the text box. Click **OK**.
- **Step 8** Verify whether the permissions are correct.
 - 1. Click the desired topic name to go to the topic details page.
 - 2. Choose the **User Permissions** tab.
 - 3. View the configured user permissions.

Figure 4-12 Viewing authorized users and their permissions



----End

Deleting Permissions for a Topic

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

- **Step 2** Click on the upper left corner to select the region where your instance is located.
- **Step 3** Click the desired Kafka instance to view the instance details.
- **Step 4** In the navigation pane, choose **Instance** > **Topics**.
- **Step 5** Go to the user permission page in either of the following ways:
 - In the row containing the desired topic, click **Grant User Permission**.
 - Click the name of the topic to go to the topic details page. Click **Configure Permission** in the upper right corner.
 - Click the name of the topic to go to the topic details page. Choose the **User Permissions** tab. Click **Configure Permission**.
- **Step 6** In the **Topic Permissions** area, in the row containing the user, click **Delete**.
- **Step 7** At the bottom of the **User Permissions** dialog box, click **Auto Enter**. The system will automatically enter **MODIFY** in the text box. Then, click **OK**.
- **Step 8** Verify whether the permissions have been deleted.
 - 1. Click the name of the topic to go to the topic details page.
 - 2. Choose the **User Permissions** tab.
 - 3. The user is deleted if it is not displayed in **User Permissions**.

----End

Related Document

To set topic permissions by calling an API, see **Granting User Permissions**.

4.3 Managing Topics

4.3.1 Viewing Kafka Topic Details

On the details page of a topic, view its basic information, partition and producer information, and subscriptions.

Procedure

- **Step 1** Log in to the **Kafka console**.
- **Step 2** Click in the upper left corner to select the region where your instance is located.
- **Step 3** Click the desired instance to go to the instance details page.
- **Step 4** In the navigation pane, choose **Instance** > **Topics**.
- **Step 5** Click a topic name to go to the topic details page.

On the topic details page, the **basic information**, **partitions**, **producers**, and **subscriptions** are displayed.

----End

Basic Information of a Topic

Figure 4-13 Basic information of a topic

Topic Name	topic-1128625419	Brokers	broker-0,broker-1,broker-2
Partitions	3	Created	Jul 03, 2025 16:24:30 GMT+08:00

Table 4-6 Basic information of a topic

Parameter	Description	
Topic Name	Name of this topic.	
Brokers	This topic has been associated with brokers.	
Partitions	Number of partitions of this topic.	
Created	Time when this topic is created. CAUTION The topic creation time is not displayed on the topic details page in any of the following cases: The topics were created on or before July 10, 2023. The topics were created automatically, by commands or code in clients, or with Kafka Manager.	

Partitions

Figure 4-14 Partitions

Partitions	Producers	Subscriptions	User Permission	ns Monitoring	Message Query			
	Partition 👄	Mini	mum Offset 🖨	First Updated ⊜		Maximum Offset ⊜	Last Updated ⊜	Messages ⊜
	0		0	Jan 02, 2025 09:59:12	? GM	0	Jan 02, 2025 09:59:12 GM	1
	1		0	-		0	**	0
	2		0	-		0		0

Table 4-7 Partition information of a topic

Parameter	Description
Partition	Partition No. of this topic.
Minimum Offset	Minimum offset of this partition.

Parameter	Description
First Updated	Time when the earliest message in this partition is updated. CAUTION First Updated displays when all messages in the partition are aged or no message is produced.
Maximum Offset	Maximum offset of this partition.
Last Updated	Time when the last message in this partition is updated.
	CAUTION Last Updated displays when all messages in the partition are aged or no message is produced.
Messages	Number of messages in this partition.

Producers



For topics created on or before Jul 10, 2023, **Producers** is not displayed on the topic details page.

Figure 4-15 Producers

Partitions	Producers	Subscriptions	User Permissions		
Broker Add	lress ⊜		Producer Address ⊜	Producer Connected ♦	
192.168.0.1	78:9092		192.168.0.21:35224	May 17, 2024 14:37:06 GMT+08:00	

Table 4-8 Producer information of a topic

Parameter	Description
Broker Address	Broker address of the Kafka instance connected to the producer.
Producer Address	Address of the producer client.
Producer Connected	Time when the producer is connected to the Kafka instance.

Subscriptions



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

Figure 4-16 Subscriptions



Table 4-9 Subscriptions of a topic

Parameter	Description
Consumer Group Name	Name of the consumer group that subscribes to this topic. Clicking a consumer group name can go to the consumer group details page and view the consumer list and consumption progress.
Status	 Current status of a consumer group. DEAD: The consumer group has no member or metadata. EMPTY: The consumer group has metadata but has no member. PREPARING_REBALANCE: The consumer group is to be rebalanced. COMPLETING_REBALANCE: All members have joined the consumer group. STABLE: Members in the consumer group can consume messages normally.
Coordinator(ID)	Broker where the Coordinator component is.
Accumulated Messages	Number of remaining messages that can be consumed in a consumer group.

Related Documents

- To query the partition information of a topic by calling an API, see Querying the Partition List of a Topic.
- To query the producer information of a topic by calling an API, see Querying the Current Producer List of a Topic.

4.3.2 Viewing Kafka Topic Logs

Topic logs record the details about leader election in topic partitions, covering the leader election time, topic partitions, and number of leader elections. This section describes how to view topic logs on the console.

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

Notes and Constraints

- Unavailable for old instances. See the console.
- By default, topic logs are retained for 7 days. To store the logs for longer, modify the log group retention period on the LTS console.
- Enabling logging will create a log group, log stream, and dashboard in LTS.
 Fees are generated based on the log volume. For details, see LTS pricing details.
- Frequently generating topic logs may affect instance performance.
- When Kafka instances store topic logs in one log group and log stream, the log group and stream of each instance contain the topic logs of all the instances.

Prerequisites

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

Enabling Topic Logging

- **Step 1** Log in to the **Kafka console**.
- **Step 2** Click in the upper left corner to select the region where your instance is located.
- **Step 3** Click the desired instance to go to the instance details page.
- **Step 4** In the left navigation pane, choose **Analysis & Diagnosis** > **Topic Logs**.
- **Step 5** Click **Enable Logging**. The **Enable Logging** dialog box is displayed.
- **Step 6** Click **OK**. The **Configure Logs** dialog box is displayed.
- **Step 7** Determine whether to enable log configuration as required and click **OK**. The **Background Tasks** page is displayed. If the logging enabling task is in the **Successful** state, topic logging is enabled successfully.
 - Disable: LTS automatically creates a log group and log stream.
 - Enable: Select the log group and log stream for storing the log file **topic.log**. You can click **View Log Group** on the right to go to the LTS console to view or create log groups and log streams.

----End

Viewing Topic Logs

- **Step 1** Log in to the **Kafka console**.
- **Step 2** Click in the upper left corner to select the region where your instance is located.
- **Step 3** Click the desired instance to go to the instance details page.
- **Step 4** In the left navigation pane, choose **Analysis & Diagnosis > Topic Logs**.
- **Step 5** On the **Logs** tab page, view topic logs.

For details about how to search for logs, see Accessing the Log Search Page.

The following is an example of topic logs:

```
"level": "INFO",
   "timestamp": "2024-12-27 17:26:13,361",
   "message": {
      "topicPartition": "topic-0",
      "targetState": "OnlinePartition",
"leaderAndIsr": "LeaderAndIsr(leader=1, leaderEpoch=3, isr=List(1, 0),
leaderRecoveryState=RECOVERED, partitionEpoch=3)",
      "partitionState": "OnlinePartition",
      "topic": "topic",
"type": "ELECT_LEADER"
  }
   "level": "INFO",
   "timestamp": "2024-12-27 17:26:13,491",
   "message": {
    "leader": "1",
      "startOffset": "0",
      "topic": "topic",
"type": "MAKE_LEADER",
      "topicPartition": "topic-0",
       "epoch": "3"
```

Table 4-10 describes the parameters.

Table 4-10 Topic log parameters

Parameter	Description
level	Level of the topic logs. The only value is INFO .
timestamp	Time when a leader is elected or determined in a topic partition.
topicPartition	Topic partition.

Parameter	Description
targetState	 Target state. The options are as follows: NewPartition: The partition is creating. OnlinePartition: The partition is working properly. OfflinePartition: The partition is discontinued. NonExistentPartition: The partition does not exist or has been deleted.
leaderAndIsr	Information of the leaderAndIsr request.
partitionState	 Partition state. The options are as follows: NewPartition: The partition is creating. OnlinePartition: The partition is working properly. OfflinePartition: The partition is discontinued. NonExistentPartition: The partition does not exist or has been deleted.
topic	Topic name.
type	Phase of a leader. The options are as follows: • ELECT_LEADER: Electing • MAKE_LEADER: Elected
leader	Partition where a leader is located.
startOffset	Offset of the first message written by a leader in an epoch. Each epoch corresponds to a startOffset.
epoch	Number of leader elections. The initial value is 0 . The epoch value increases by 1 each time a leader is elected.

----End

Disabling Topic Logging

- **Step 1** Log in to the **Kafka console**.
- **Step 2** Click in the upper left corner to select the region where your instance is located.
- **Step 3** Click the desired instance to go to the instance details page.
- **Step 4** In the left navigation pane, choose **Analysis & Diagnosis > Topic Logs**.
- **Step 5** Click **Disable Logging** in the upper right corner. The **Disable Logging** dialog box is displayed.
- **Step 6** Click **OK**. The **Background Tasks** page is displayed. The topic log function is disabled when the **Disable logging** task is in the **Successful** state.

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

----End

4.3.3 Modifying Kafka Topic Configurations

This section describes how to modify configurations in **Table 4-11** of a Kafka topic on the console.

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

Table 4-11 Kafka topic configuration parameters

Parameter	Description
Partitions	Number of partitions in a topic. For details about how to change, see Changing Kafka Partition Quantity .
Aging Time (h)	Maximum message retention. For details about how to change, see Changing Kafka Message Retention Period.
Replicas	Number of replicas of each topic partition. To modify it, see Modifying Kafka Topic Replicas .
Synchronous Replication	A message is returned to the client only after the message creation request has been received and the message has been acknowledged by all replicas.
Synchronous Flushing	 Enabled: A message is immediately flushed to disk once it is created, bringing higher reliability. Disabled: A message is stored in the memory instead of being immediately flushed to disk once created.
Message Timestamp	 Timestamp type of a message. Options: CreateTime: time when the producer created the message. LogAppendTime: time when the broker appended the message to the log.
Max. Message Size	Maximum size of messages to be processed in batches. If message compression is enabled, this parameter indicates the size after compression.
	If this value is increased and the consumer version is earlier than 0.10.2, the consumers' fetch size must also be increased so that they can obtain the latest value.
Description	Topic description.

Notes and Constraints

- If there is only one replica, **Synchronous Replication** cannot be enabled.
- After enabling synchronous replication, set acks to all or -1 on the client.
 Otherwise, this function will not take effect.
- A maximum of 50 topics can be modified in a batch at a time.

Procedure

One or more topics can be modified on the Kafka console. To modify topics in batches, their descriptions cannot be modified.

Modifying a Topic

- Step 1 Log in to the Kafka console.
- **Step 2** Click in the upper left corner to select the region where your instance is located.
- **Step 3** Click the desired instance to go to the instance details page.
- **Step 4** In the navigation pane, choose **Instance** > **Topics**.
- **Step 5** In the row containing the desired topic, click **Edit**.
- **Step 6** In the **Edit Topic** dialog box, change configurations and click **OK**.

View the reconfiguration on the **Topics** page.

----End

Modifying Topics in Batches

- Step 1 Log in to the Kafka console.
- **Step 2** Click in the upper left corner to select the region where your instance is located.
- **Step 3** Click the desired instance to go to the instance details page.
- **Step 4** In the navigation pane, choose **Instance** > **Topics**.
- **Step 5** Select the desired topics and click **Batch Edit Topic** on the upper left.
- **Step 6** In the **Batch Operations** area, select the items listed in **Table 4-12**. In the **Preview Change** area, view the unchanged and changed items. Click **OK**.

Figure 4-17 Modifying topics in batches

Batch Edit Topic

Selected settings will be synced to all selected topics. **Batch Operations** Partitions Aging Time (h) Synchronous Replication Synchronous Flushing Message Timestamp CreateTime Max. Message Size (bytes) 10485759 + **Preview Change** Max. Message Size (bytes)Before/After Topic Name Message TimestampBefore/After CreateTime 10485759 topic-doc01 10485760 LogAppendTime CreateTime 10485759 topic-doc02 LogAppendTime 10485760 CreateTime 10485759 topic-doc03 LogAppendTime 10485760 Total Records: 3 10 ∨ | < 1 >

Table 4-12 Topic configuration parameters

Parameter	Description
Synchronous Replication	Select it and enable or disable this function.
Synchronous Flushing	Select it and enable or disable this function.
Message Timestamp	Select it and select CreateTime or LogAppendTime from the drop-down list box.
Max. Message Size (bytes)	Select it and enter a value.

View the reconfiguration on the **Topics** page.

----End

Related Document

To modify topic configurations by calling an API, see **Modifying Topics of a Kafka Instance**.

4.3.4 Changing Kafka Partition Quantity

After creating a topic, you can change the number of partitions as required. Changing the number of partitions does **not** restart the instance or affect services.

Notes and Constraints

- The number of partitions can only be increased.
- Instances created since May 17, 2023 do not have Kafka Manager. You cannot modify topic partitions for these instances using Kafka Manager.
- The partition quantity of topics of a single-node or cluster Kafka instance is limited. When the partition quantity limit is reached, you can no longer create topics. The total number of partitions varies with instance specifications. For details, see Kafka Instance Specifications.
- For an instance with ciphertext access enabled, if allow.everyone.if.no.acl.found is set to false, the topic partition quantity can be modified on the client only by the initial user (set in first ciphertext access enablement).
- A maximum of 50 partitions can be modified in a batch at a time.

Procedure

You can repartition a topic on the Kafka console, using Kafka Manager, or on a client.

Repartitioning a Topic on the Console

- **Step 1** Log in to the **Kafka console**.
- **Step 2** Click in the upper left corner to select the region where your instance is located.
- **Step 3** Click the desired instance to go to the instance details page.
- **Step 4** In the navigation pane, choose **Instance** > **Topics**.
- **Step 5** In the row containing the desired topic, click **Edit**.
- **Step 6** In the **Edit Topic** dialog box, enter the number of partitions by referring to **Table 4-13**, and click **OK**.

Table 4-13 Partitions parameters

Parameter	Description
Partitions	Enter the number of partitions. The number of partitions can only be increased.
	To ensure performance, a maximum of 200 partitions is allowed for each topic on the Kafka console.

Parameter	Description
Broker	 Set the broker where the new partition is located. Auto: The Kafka service automatically allocates brokers to the new partition. Specified: Select brokers from the drop-down list box. The number of brokers cannot be less than the number of
	topic replicas.

View the partition quantity on the **Topics** page.

----End

Modifying the Number of Partitions in Batches

- **Step 1** Log in to the **Kafka console**.
- **Step 2** Click on the upper left corner to select the region where your instance is located.
- **Step 3** Click the desired instance to go to the instance details page.
- **Step 4** In the navigation pane, choose **Instance** > **Topics**.
- **Step 5** Select the desired topics and click **Batch Edit Topic** on the upper left.
- **Step 6** In the **Batch Operations** area, select **Partitions** and enter a value. In the **Preview Change** area, view the unchanged and changed item. Click **OK**.

The number of partitions is subject to the following constraints:

- To ensure performance, a maximum of 200 partitions is allowed for each topic on the Kafka console.
- This value must be set to a value greater than or equal to the maximum number of partitions among the topics. For example, when topic **Topic01** has three partitions and topic **Topic02** has six partitions, the value must be greater than or equal to 6.

Figure 4-18 Batch modifying the number of partitions

Batch Edit Topic Selected settings will be synced to all selected topics. **Batch Operations** Partitions + _ Aging Time (h) Synchronous Replication Synchronous Flushing Message Timestamp Max. Message Size (bytes) Preview Change Topic Name PartitionsBefore/After topic-doc01 2 6 topic-doc02 3 6 topic-doc03

View the partition quantity on the **Topics** page.

----End

Total Records: 3

Modifying Topic Partitions on Kafka Manager

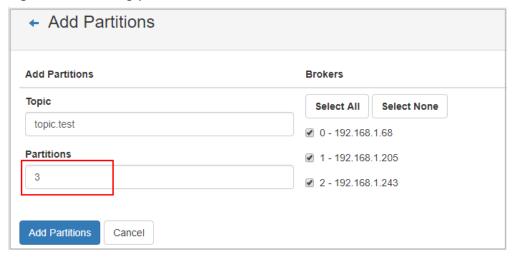
- Step 1 Log in to Kafka Manager.
- **Step 2** Choose **Topic** > **List** to view the list of topics.
- **Step 3** Click a topic to view its details.
- Step 4 Click Add Partitions.

Figure 4-19 Topic details page

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

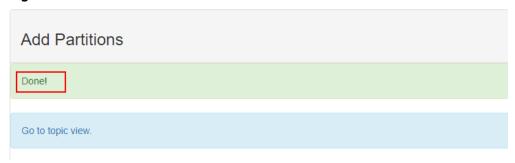
To ensure performance, 200 or less partitions are recommended for each topic.

Figure 4-20 Adding partitions



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

Figure 4-21 Partitions added



----End

Modifying Topic Partitions on the Client

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

 For a Kafka instance with ciphertext access disabled, run the following command in the /bin directory of the Kafka client:

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

Table 4-14 Topic partition quantity parameters

Parameter	Description
connection-address	Connection address of a Kafka instance. To obtain the address, choose Overview > Connection .
topic-name	Topic name.
number-of-partitions	Number of partitions in the topic.
	To ensure performance, 200 or fewer partitions are recommended for each topic.

Example:

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

- For a Kafka instance with ciphertext access enabled, do as follows:
 - a. (Optional) If the username and password, and the SSL certificate has been configured, skip this step and go to b. Otherwise, do as follows:
 Create the ssl-user-config.properties file in the /config directory of the Kafka client. Add the username and password, and the SSL certificate configuration by referring to Step 3.
 - b. Run the following command in the **/bin** directory of the Kafka client: ./kafka-topics.sh --bootstrap-server {connection-address} --topic {topic-name} --alter --partitions {number-of-partitions} --command-config ../config/{ssl-user-config.properties}

Table 4-15 Topic partition quantity parameters

Parameter	Description
connection-address	Connection address of a Kafka instance. To obtain the address, choose Overview > Connection .
topic-name	Topic name.
number-of- partitions	Number of partitions in the topic. To ensure performance, 200 or fewer partitions are recommended for each topic.

Parameter	Description
ssl-user- config.properties	Configuration file name. This file contains username, password, and SSL certificate information.

Example:

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

Related Documents

- To modify the partition quantity of a topic by calling an API, see Modifying Topics of a Kafka Instance.
- To increase the partition quantity of a topic, see **How Do I Increase the Partition Quantity?**

4.3.5 Modifying Kafka Topic Replicas

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

Notes and Constraints

- Unavailable for single-node instances.
- Reassignment tasks cannot be manually stopped. Please wait until they complete.
- If partition reassignment has been scheduled, reassignment cannot be scheduled again for any topic in this instance until this reassignment is executed.

Operation Impact

- Partition reassignment on topics with a large amount of data consumes a
 large amount of network and storage bandwidth. As a result, service requests
 may time out or the latency may increase. Therefore, you are advised to
 perform reassignment during off-peak hours. Compare the current instance
 load based on the instance specifications to decide whether the remaining
 instance capacity can support partition reassignment. Do not reassign
 partitions when there is insufficient bandwidth or when the CPU usage is
 greater than 90%. To view data volume and CPU usage of a topic, see
 Message Size and CPU Usage on the monitoring page. For details, see
 Viewing Kafka Metrics.
- A throttle refers to the upper limit of the bandwidth for replication of a topic, to ensure that other topics on the instance are not affected. Note that throttles apply to replication triggered by both normal message production and partition reassignment. If the throttle is too small, normal message production may be affected, and partition reassignment may never complete. If partitions are continuously reassigned, contact customer service.

- You cannot delete topics whose reassignment tasks have started. Otherwise, the tasks will never complete.
- After partition reassignment, the metadata of the topic changes. If the producer does not support the retry mechanism, a few requests will fail, causing some messages to fail to be produced.
- Reassignment takes longer for a topic with a large data volume. To check the
 volume, see the Message Size metric on the monitoring page by referring to
 Viewing Kafka Metrics. To reduce the amount of data to be migrated,
 decrease the topic aging time without affecting services and wait for
 messages to age. After the reassignment is complete, you can restore the
 aging time.

Prerequisite

The target broker should have sufficient disk space. To check available disk space of each broker, see **Viewing Kafka Disk Usage**. If the remaining disk capacity of the target broker is close to the amount of data to be migrated to the broker, **expand the disk capacity** before the reassignment.

Modifying Topic Replicas

The Kafka console supports two ways:

- Automatic reassignment: Batch modification is supported.
- Manual reassignment: One topic in each modification.

Modifying Replicas by Automatic Reassignment

- Step 1 Log in to the Kafka console.
- **Step 2** Click in the upper left corner to select the region where your instance is located
- **Step 3** Click the desired instance to go to the instance details page.
- **Step 4** In the navigation pane, choose **Instance** > **Topics**.
- **Step 5** Go to the **Auto** page in either of the following ways:
 - Select one or more topics and choose Reassign > Auto above the topic list.
 - In the row containing the desired topic, choose More > Reassign > Auto.
- **Step 6** Modify the replicas.

Table 4-16 Parameters of automatic reassignment

Parameter	Description
Broker Name	Select the brokers to assign the topic's partition replicas to.
Replicas	Enter the number of replicas. This number must be less than or equal to the number of brokers.

Parameter	Description
Max. Bandwidth	Enter the bandwidth. The default value is -1, indicating that the bandwidth is not limited.
	If the instance has low workload (for example, only 30 of 300 MB/s is used), you are not advised to limit the bandwidth. Otherwise, you are advised to set it to a value greater than or equal to the total production bandwidth of the to-be-reassigned topic multiplied by the maximum number of replicas of the to-be-reassigned topic.
	For details, see Calculating a Throttle.
Execute	Specify when to execute the reassignment.
	Now means to execute it immediately.
	As scheduled means to execute it at the scheduled time.

Step 7 (Optional) Click **Calculate**. **Time Required** indicates how long automatic balancing will take.

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

Step 8 Click OK.

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

Table 4-17 Checking the reassignment result

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

----End

Modifying Replicas by Manual Reassignment

- **Step 1** Log in to the **Kafka console**.
- **Step 2** Click in the upper left corner to select the region where your instance is located.
- **Step 3** Click the desired instance to go to the instance details page.
- **Step 4** In the navigation pane, choose **Instance** > **Topics**.
- **Step 5** Go to the **Manual** page in either of the following ways:
 - Select a topic and choose **Reassign** > **Manual** above the topic list. Manual reassignment does not support batch operations.
 - In the row containing the desired topic, choose **More** > **Reassign** > **Manual**.

Step 6 Modify the replicas.

- In the upper right corner of the Manual dialog box, click Delete Replica or Add Replica to reduce or increase the number of replicas for each partition of the topic.
- Under the name of the replica to be reassigned, click the broker name or vand select the target broker to migrate the replica to. Assign replicas of the same partition to different brokers.
- Specify **throttle**. The default value is **-1**, indicating that there is no throttle. If the instance has low workload (for example, only 30 of 300 MB/s is used), you are not advised to limit the bandwidth. Otherwise, you are advised to set it to a value greater than or equal to the total production bandwidth of the to-be-reassigned topic multiplied by the maximum number of replicas of the to-be-reassigned topic. For details, see **Calculating a Throttle**.
- For **Execute**, specify when to execute the reassignment. **Now** means to execute it immediately. **As scheduled** means to execute it at the scheduled time.
- **Step 7** (Optional) Click **Calculate**. **Time Required** indicates how long manual balancing will take.

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

Step 8 Click OK.

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

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

Table 4-18 Checking the reassignment result

----End

Related Document

To change the replica quantity by calling an API, see **Reassigning Replicas of a Topic for a Kafka Instance**.

4.3.6 Exporting the Kafka Topic List

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

Prerequisites

A topic has been created.

Procedure

- **Step 1** Log in to the **Kafka console**.
- **Step 2** Click on the upper left corner to select the region where your instance is located.
- **Step 3** Click the desired instance to go to the instance details page.
- **Step 4** In the navigation pane, choose **Instance** > **Topics**.
- **Step 5** Export the topic list in either of the following ways:
 - Select the desired topics and choose Export > Export selected data to an XLSX file to export specified topics.

Choose Export > Export all data to an XLSX file to export all topics.

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

----End

4.3.7 Reassigning Kafka Partitions

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

Partition reassignment is required in the following scenarios:

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

Notes and Constraints

- Unavailable for single-node instances.
- You cannot modify the partition quantity of topics whose reassignment tasks have started.
- Reassignment tasks cannot be manually stopped. Please wait until they complete.
- If partition reassignment has been scheduled, reassignment cannot be scheduled again for any topic in this instance until this reassignment is executed.

Operation Impact

- Partition reassignment on topics with a large amount of data consumes a large amount of network and storage bandwidth. As a result, service requests may time out or the latency may increase. Therefore, you are advised to perform reassignment during off-peak hours. Compare the current instance load based on the instance specifications to decide whether the remaining instance capacity can support partition reassignment. Do not reassign partitions when there is insufficient bandwidth or when the CPU usage is greater than 90%. To view data volume and CPU usage of a topic, see
 Message Size and CPU Usage on the monitoring page. For details, see
 Viewing Kafka Metrics.
- A throttle refers to the upper limit of the bandwidth for replication of a topic, to ensure that other topics on the instance are not affected. Note that throttles apply to replication triggered by both normal message production and partition reassignment. If the throttle is too small, normal message production may be affected, and partition reassignment may never complete. If partitions are continuously reassigned, contact customer service.
- You cannot delete topics whose reassignment tasks have started. Otherwise, the tasks will never complete.

- After partition reassignment, the metadata of the topic changes. If the producer does not support the retry mechanism, a few requests will fail, causing some messages to fail to be produced.
- Reassignment takes longer for a topic with a large data volume. To check the
 volume, see the Message Size metric on the monitoring page by referring to
 Viewing Kafka Metrics. To reduce the amount of data to be migrated,
 decrease the topic aging time without affecting services and wait for
 messages to age. After the reassignment is complete, you can restore the
 aging time.

Prerequisite

The target broker should have sufficient disk space. To check available disk space of each broker, see **Viewing Kafka Disk Usage**. If the remaining disk capacity of the target broker is close to the amount of data to be migrated to the broker, **expand the disk capacity** before the reassignment.

Reassigning Kafka Partitions

The Kafka console supports two ways:

- Automatic reassignment: The partitions of topics can be reassigned in batches.
- Manual reassessment: The partitions of one topic can be reassigned at a time.

Auto Reassignment

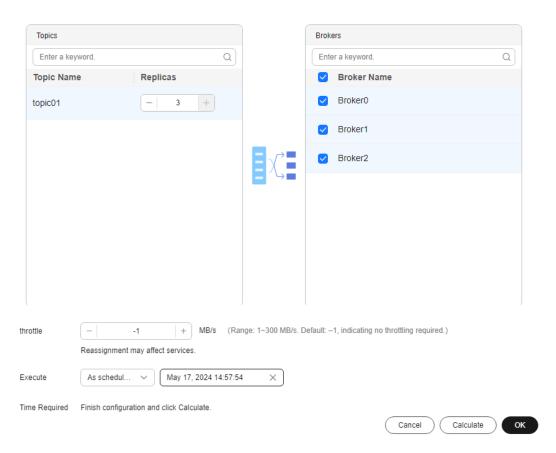
- **Step 1** Log in to the **Kafka console**.
- **Step 2** Click in the upper left corner to select the region where your instance is located.
- **Step 3** Click the desired instance to go to the instance details page.
- **Step 4** In the navigation pane, choose **Instance** > **Topics**.
- **Step 5** Reassign partitions using either of the following methods:
 - Select one or more topics and choose **Reassign** > **Auto** above the topic list.
 - In the row that contains the desired topic, choose More > Reassign > Auto.
- **Step 6** Set automatic reassignment parameters.

Table 4-19 Parameters of automatic reassignment

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

Parameter	Description
Max. Bandwidth	Enter the bandwidth. The default value is -1, indicating that the bandwidth is not limited.
	If the instance has low workload (for example, only 30 of 300 MB/s is used), you are not advised to limit the bandwidth. Otherwise, you are advised to set it to a value greater than or equal to the total production bandwidth of the to-be-reassigned topic multiplied by the maximum number of replicas of the to-be-reassigned topic. For details, see Calculating a Throttle.
Execute	Specify when to execute the reassignment.
	Now means to execute it immediately.
	As scheduled means to execute it at the scheduled time.

Figure 4-22 Setting automatic reassignment parameters **Auto**



Step 7 (Optional) Click **Calculate**. **Time Required** indicates how long automatic balancing will take.

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

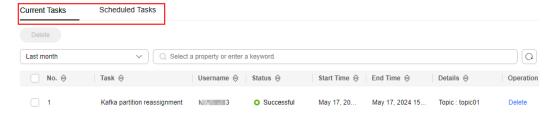
Step 8 Click OK.

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

Table 4-20 Checking the reassignment result

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

Figure 4-23 Background Tasks page



Ⅲ NOTE

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

----End

Manual Reassignment

- **Step 1** Log in to the **Kafka console**.
- **Step 2** Click in the upper left corner to select the region where your instance is located.
- **Step 3** Click the desired instance to go to the instance details page.
- **Step 4** In the navigation pane, choose **Instance** > **Topics**.
- **Step 5** Reassign partitions using either of the following methods:
 - Select a topic and choose Reassign > Manual above the topic list. Manual reassignment does not support batch operations.
 - In the row that contains the desired topic, choose More > Reassign > Manual.
- **Step 6** Set manual reassignment parameters.
 - In the upper right corner of the Manual dialog box, click Delete Replica or Add Replica to reduce or increase the number of replicas for each partition of the topic.
 - Under the name of the replica to be reassigned, click the broker name or value and select the target broker to migrate the replica to. Assign replicas of the same partition to different brokers.
 - Specify **throttle**. The default value is **-1**, indicating that there is no throttle If the instance has low workload (for example, only 30 of 300 MB/s is used), you are not advised to limit the bandwidth. Otherwise, you are advised to set it to a value greater than or equal to the total production bandwidth of the to-be-reassigned topic multiplied by the maximum number of replicas of the to-be-reassigned topic. For details, see **Calculating a Throttle**.
 - For **Execute**, specify when to execute the reassignment. **Now** means to execute it immediately. **As scheduled** means to execute it at the scheduled time.

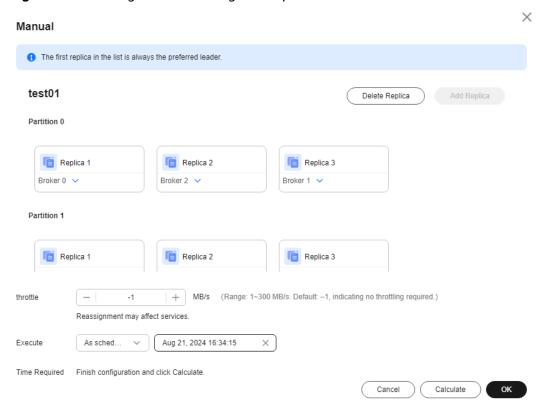


Figure 4-24 Setting manual reassignment parameters

Step 7 (Optional) Click **Calculate**. **Time Required** indicates how long manual balancing will take.

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

Step 8 Click OK.

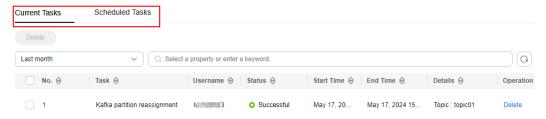
The following table lists how to check whether reassignment is complete (scheduled and non-scheduled tasks):

Table 4-21 Checking the reassignment result

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

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

Figure 4-25 Background Tasks page



◯ NOTE

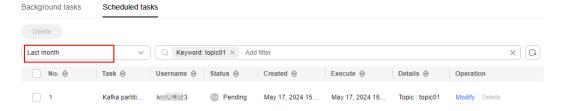
- You cannot delete topics whose reassignment tasks have started. Otherwise, the tasks will never complete.
- You cannot modify the partition quantity of topics whose reassignment tasks have started.
- Reassignment tasks cannot be manually stopped. Please wait until they complete.
- If partition reassignment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

----End

Re-scheduling Partition Reassignment

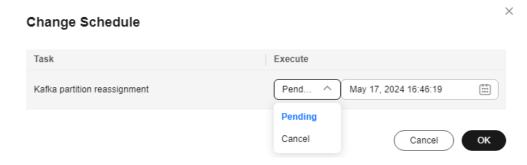
Step 1 On the **Scheduled Tasks** tab page on the **Instance** > **Background Tasks** page, click the drop-down box in the upper left corner, select a time period, enter the desired topic name in the search box, and press **Enter**.

Figure 4-26 Querying reassignment schedules



- **Step 2** In the row that contains the desired task, click **Modify**.
- **Step 3** In the **Change Schedule** dialog box, change the schedule or cancel the scheduled task.
 - To change the schedule, select a time and click **OK**.
 - To cancel the task, select **Cancel** (as shown in **Figure 4-27**) and click **OK**.

Figure 4-27 Canceling a reassignment schedule



----End

Calculating a Throttle

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

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

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

Scenario 1: Topic 1 has two partitions and two replicas, and Topic 2 has one partition and one replica. All leader replicas are on the same broker, as shown in Table 4-22. One replica needs to be added for Topic 1 and Topic 2 respectively, as shown in Table 4-23.

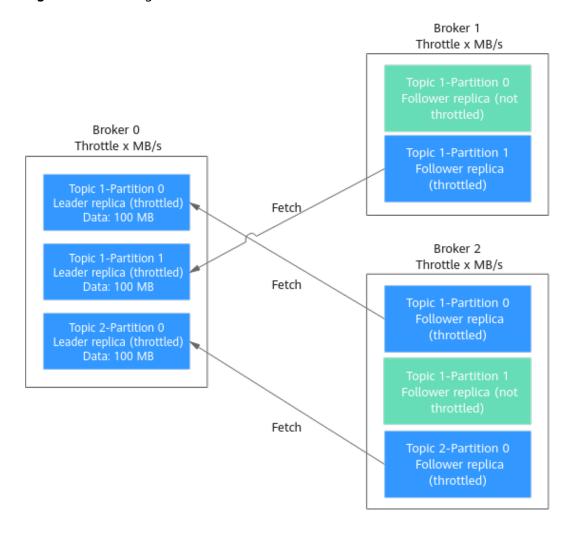
Table 4-22 Replica	distribution befo	ore reassignment
--------------------	-------------------	------------------

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

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

Table 4-23 Replica distribution after reassignment

Figure 4-28 Reassignment scenario 1



As shown in **Figure 4-28**, three replicas fetch data from Broker 0. Each replica on Broker 0 has 100 MB data. Broker 0 has only leader replicas, and Broker 1 and Broker 2 have only follower replicas.

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

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

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

Scenario 2: Topic 1 has two partitions and one replica, and Topic 2 has two partitions and one replica. Leader replicas are on different brokers, as shown in Table 4-24. One replica needs to be added for Topic 1 and Topic 2 respectively, as shown in Table 4-25.

Table 4-24 Replica distribution before reassignment

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

Table 4-25 Replica distribution after reassignment

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

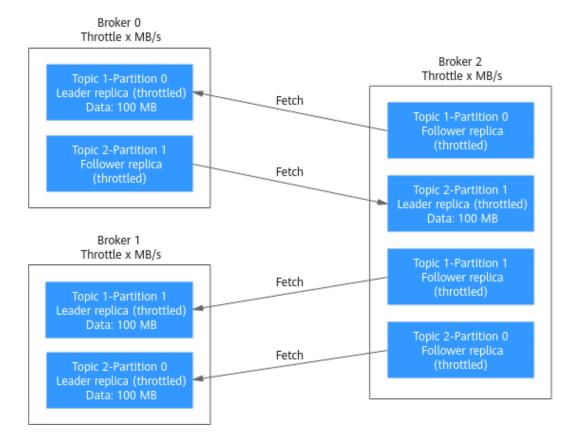


Figure 4-29 Reassignment scenario 2

As shown in **Figure 4-29**, Broker 1 has only leader replicas, and Broker 0 and Broker 2 have both leader and follower replicas. Leader and follower replicas on Broker 0 and Broker 2 are throttled separately.

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

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

Scenario 3: Both Topic 1 and Topic 2 have one partition and two replicas. All leader replicas are on the same broker. One replica needs to be added to Topic 1, as shown in Table 4-26. Messages are produced on Topic 1, causing replication, as shown in Table 4-27.

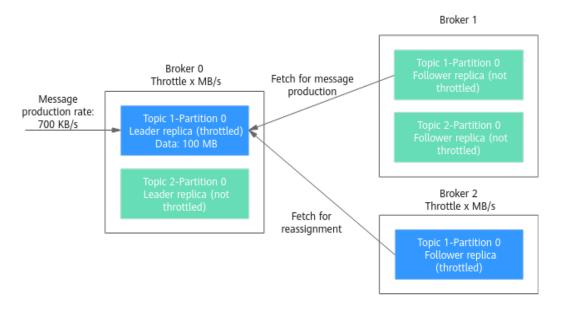
Table 4-26 Replica distribution before reassignment

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

Table 4-27 Replica distribution after reassignment

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

Figure 4-30 Reassignment scenario 3



As shown in **Figure 4-30**, one replica needs to fetch data from Broker 0 for partition reassignment, and the other replica needs to fetch data from Broker 0 for message production. Since the throttle does not distinguish between message production and partition reassignment, the traffic caused by both is limited and counted.

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

In conclusion, to complete the partition reassignment task within 200s, set the throttle to a value greater than or equal to 1.9 MB/s. The bandwidth should be set

to be greater than or equal to 2 MB/s because the limit on it on the console must be an integer.

Related Documents

- To reassign partitions by calling an API, see Reassigning Replicas of a Topic for a Kafka Instance.
- Learn about the causes and handling measures of uneven service data among partitions, see Handling Uneven Service Data.

4.3.8 Configuring Automatic Topic Creation

Automatic Topic Creation indicates that a topic will be automatically created when a message is produced in or consumed from a topic that does not exist. By default, the topic has parameters listed in **Table 4-28**.

The following parameters of cluster instances can be changed on the **Instance** > **Parameters** page: **log.retention.hours** (retention period), **default.replication.factor** (replica quantity), or **num.partitions** (partition quantity). The value will be used in later topics that are automatically created.

For example, assume that **num.partitions** is changed to **5**, an automatically created topic has parameters listed in **Table 4-28**.

Table 4-28 To	pic parameters
----------------------	----------------

Parameter	Default Value (Single-node)	Default Value (Cluster)	Modified To (Cluster)
Partitions	1	3	5
Replicas	1	3	3
Aging Time (h)	72	72	72
Synchronous Replication	Disabled	Disabled	Disabled
Synchronous Flushing	Disabled	Disabled	Disabled
Message Timestamp	CreateTime	CreateTime	CreateTime
Max. Message Size (bytes)	10,485,760	10,485,760	10,485,760

Notes and Constraints

Enabling or disabling automatic topic creation may restart the instance.

Procedure

- Step 1 Log in to the Kafka console.
- **Step 2** Click on the upper left corner to select the region where your instance is located
- **Step 3** Click the name of the desired Kafka instance to go to the **Overview** page.
- Step 4 In the Instance Information area, click or next to Automatic Topic Creation. The Confirm dialog box is displayed.
- **Step 5** Click **OK**. The **Background Tasks** page is displayed. Automatic topic creation has been configured when the task is in the **Successful** state.

----End

Related Documents

- To configure automatic topic creation by calling an API, see Configuring Automatic Topic Creation.
- To manually create a topic, see Creating a Kafka Topic.

4.3.9 Deleting a Kafka Topic

This document describes how to delete a topic.

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

Notes and Constraints

- Deleting a topic clears the topic data permanently.
- For an instance with ciphertext access enabled, if allow.everyone.if.no.acl.found is set to false, the topic can be deleted on the client only by the initial user (set in first ciphertext access enablement).

Prerequisite

The instance is in the **Running** state.

Deleting a Kafka Topic

You can delete a topic on the console or client.

Deleting a Kafka Topic (Console)

- **Step 1** Log in to the **Kafka console**.
- **Step 2** Click on the upper left corner to select the region where your instance is located.
- **Step 3** Click the desired instance to go to the instance details page.

- **Step 4** In the navigation pane, choose **Instance** > **Topics**.
- **Step 5** Delete topics using either of the following methods:
 - Select one or more topics and click **Delete Topic** in the upper left corner.
 - In the row containing the topic you want to delete, choose **More** > **Delete**.
- **Step 6** In the **Delete Topic** dialog box, click **OK**.

The topic is deleted if it is not displayed in the topic list.

----End

Deleting a Kafka Topic on the Client

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

For a Kafka instance with ciphertext access disabled, run the following command in the /bin directory of the Kafka client:

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

Table 4-29 Topic deletion parameters

Parameter	Description
connection-address	Connection address of a Kafka instance. To obtain the address, choose Overview > Connection .
topic-name	Topic name.

Example:

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

- For a Kafka instance with ciphertext access enabled, do as follows:
 - a. (Optional) If the SSL certificate has been configured, skip this step and go to b. Otherwise, do as follows:

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

b. Run the following command in the **/bin** directory of the Kafka client: ./kafka-topics.sh --bootstrap-server {connection-address} --delete --topic {topic-name} -- command-config ../config/{ssl-user-config.properties}

Table 4-30 Topic deletion parameters

Parameter	Description
connection-address	Connection address of a Kafka instance. To obtain the address, choose Overview > Connection .

Parameter	Description
topic-name	Topic name.
ssl-user- config.properties	Configuration file name. This file contains username, password, and SSL certificate information.

Example:

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

Related Documents

- To delete a topic by calling an API, see Batch Deleting Topics of a Kafka Instance.
- If a topic still exists, see Troubleshooting Topic Deletion Failures.

5 Connecting to an Instance

5.1 Configuring Kafka Network Connections

5.1.1 Kafka Network Connection Conditions

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

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

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

Table 5-1 Access modes

Mode	How To Do	Reference
Public access	To access a Kafka instance on a client using IPv4 addresses: Enable public access on the Kafka console and configure elastic IPs (EIPs). A client can connect to the Kafka instance through the EIPs.	Configuring Kafka Public Access
	To access a Kafka instance on a client using IPv6 addresses: Enable IPv6 on the Kafka console and add the IPv6 addresses into the shared bandwidth. A client can connect to the Kafka instance over a public network.	
	Configure port mapping using DNAT. The client can connect to the Kafka instance in a public network.	Accessing Kafka in a Public Network Using DNAT

Mode	How To Do	Reference
Private access	A client and a Kafka instance are interconnected when they are deployed in a VPC.	-
	When a client and a Kafka instance are deployed in different VPCs of the same region, connect the client and the Kafka instance across VPCs using a VPC endpoint.	Accessing Kafka Using a VPC Endpoint Across VPCs
	When a client and a Kafka instance are deployed in different VPCs of the same region, interconnect two VPCs using a VPC peering connection.	VPC Peering Connection

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

□ NOTE

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

Table 5-2 Security group rules

Directi on	Protoc ol	Туре	Port	Source	Description
Inboun d	ТСР	IPv4	9094	IP address or IP address group of the Kafka client	Accessing a Kafka instance over a public network (in plaintext)
Inboun d	TCP	IPv4	9092	IP address or IP address group of the Kafka client	 Accessing a Kafka instance over a private network within a VPC (in plaintext) Accessing a Kafka instance using a peering connection across VPCs (in plaintext)

Directi on	Protoc ol	Туре	Port	Source	Description
Inboun d	TCP	IPv6	9192	IP address or IP address group of the Kafka client	Accessing a Kafka instance using IPv6 addresses (without SSL) (private or public network)
Inboun d	TCP	IPv4	9095	IP address or IP address group of the Kafka client	Accessing a Kafka instance over a public network (in ciphertext)
Inboun d	TCP	IPv4	9093	IP address or IP address group of the Kafka client	 Accessing a Kafka instance over a private network within a VPC (in ciphertext) Accessing a Kafka instance using a peering connection across VPCs (in ciphertext)
Inboun d	TCP	IPv6	9193	IP address or IP address group of the Kafka client	Accessing a Kafka instance using IPv6 addresses (with SSL) (private or public network)
Inboun d	ТСР	IPv4	9011	198.19.128.0 /17	Accessing a Kafka instance using a VPC endpoint across VPCs (in ciphertext or plaintext)
Inboun d	ТСР	IPv4	9011	IP address or IP address group of the Kafka client	Accessing a Kafka instance using DNAT (in ciphertext or plaintext)

5.1.2 Configuring Kafka Public Access

A client can use IPv4 or IPv6 addresses to access a Kafka instance over a public network.

- By IPv4: On the Kafka console, enable public access and configure EIPs for the instance.
- By IPv6: Enable IPv6 in Kafka instance creation and add IPv6 addresses to the shared bandwidth to support both private and public IPv6 access.

Notes and Constraints

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

Prerequisites

- You can change the public access setting only when the Kafka instance is in the **Running** state.
- (Optional) To access a Kafka instance using IPv6 addresses, ensure that IPv6 is enabled for the Kafka instance.

Enabling Public Access to a Kafka Instance

A client can use IPv4 or IPv6 addresses to access a Kafka instance over a public network.

On the Kafka console, the procedures for enabling public IPv4 access vary depending on the content displayed in the **Connection** area on the **Overview** page.

- To enable IPv4 public access when IPv6 is enabled, see section "Enabling Public IPv4 Access (SASL Cannot Be Changed)".
- To enable IPv4 public access when IPv6 is disabled, see section "Enabling Public IPv4 Access (Plaintext or Ciphertext Access Can Be Changed)".

Enabling Public IPv4 Access (SASL Cannot Be Changed)

- **Step 1** Log in to the **Kafka console**.
- **Step 2** Click on the upper left corner to select the region where your instance is located
- **Step 3** Click the name of a Kafka instance to go to the **Overview** page.
- **Step 4** Click next to **Public Access** to enable public access. For **Elastic IP Address**, select an EIP for each broker.

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

- 1. Click **Create Elastic IP** to go to the **Buy EIP** page and purchase EIPs. For details, see **Assigning an EIP**.
- 2. After the purchase is complete, return to the public access enabling page.
- 3. Click after Elastic IP Address, select an EIP for each broker and then click
- 4. You can view the operation progress on the **Instance** > **Background Tasks** page. If the task status is **Successful**, the modification has succeeded.

Figure 5-1 Enabling public access



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

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

Directi on	Protoc ol	Туре	Port	Source	Description
Inboun d	ТСР	IPv4	9094	IP address or IP address group of the Kafka client	Accessing Kafka over a public network (without SSL)
Inboun d	TCP	IPv4	9095	IP address or IP address group of the Kafka client	Accessing Kafka over a public network (with SSL)

----End

Enabling Public IPv4 Access (Plaintext or Ciphertext Access Can Be Changed)

- Step 1 Log in to the Kafka console.
- **Step 2** Click on the upper left corner to select the region where your instance is located.
- **Step 3** Click the name of a Kafka instance to go to the **Overview** page.
- **Step 4** Click next to **Public Access** to enable public access. For **Elastic IP Address**, select an EIP for each broker.

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

- Click Create Elastic IP to go to the Buy EIP page and purchase EIPs. For details, see Assigning an EIP.
- 2. After the purchase is complete, return to the public access enabling page.
- 3. Click after Elastic IP Address, select an EIP for each broker and then click . The Background Tasks page is displayed.
- 4. If the status of the task turns to **Successful**, public access is successfully enabled.

Figure 5-2 Enabling public access



After public access is enabled, configure the access mode (plaintext or ciphertext) and security group rules listed in Table 5-4 before attempting to access Kafka. For details about accessing Kafka, see Connecting to an Instance.

Table 5-4 Kafka instance security group rules (public IPv4 access)

Directi on	Protoc ol	Туре	Port	Source	Description
Inboun d	TCP	IPv4	9094	IP address or IP address group of the Kafka client	Public plaintext access to Kafka
Inboun d	ТСР	IPv4	9095	IP address or IP address group of the Kafka client	Public ciphertext access to Kafka

----End

Enabling IPv6 Public Network Access

- **Step 1** Log in to the **Kafka console**.
- **Step 2** Click on the upper left corner to select the region where your instance is located.
- **Step 3** Click a Kafka instance to go to the **Overview** page.
- **Step 4** In the **Connection** area, obtain IPv6 **Instance Address (Private Network)**. In the **Network** area, view and record the VPC and subnet.

Connection Username Kafka SASL SSL Disabled Fixed for this instance Instance Address (Private Network) IPv4 192.168.0.169:9092,192.168.0.250:9092,192.168.0.229:9092 [2409:2001:0:aa:8fd2:7128:7cc:6105]:9192,[2409:2001:0:aa:a4db:31 37:70c9:125a]:9192,[2409:2001:0:aa:5e7d:6b09:5a9b:b9ab]:919 Public Network Access (?) Disabled Enabled Intra-VPC Plaintext Access Network ΑZ AZ1 VPC vpc-ipv6 🖸 subnet-f7ab Subnet Security Group default 🕜 🛭 🗷 IPv6 Enabled

Figure 5-3 Instance details page

- **Step 5** Click in the upper left corner of the management console and choose **Network** > **Elastic IP**. The **EIPs** page is displayed.
- **Step 6** Choose **Shared Bandwidths** in the navigation pane.
- **Step 7** Apply for a shared bandwidth. For details, see **Assigning a Shared Bandwidth**. If a shared bandwidth already exists, you do not need to apply for one again.
- **Step 8** In the row containing the shared bandwidth, click **Add Public IP Address**.
- **Step 9** Set the parameters as described in **Table 5-5** and click **OK**.

Table 5-5 Adding public IP parameters

Parameter	Description
Public IP Address	Select IPv6 Address.
VPC	Select the VPC in Step 4 from the drop-down list.
Subnet	Select the subnet in Step 4 from the drop-down list. Select all IPv6 addresses in Step 4 .

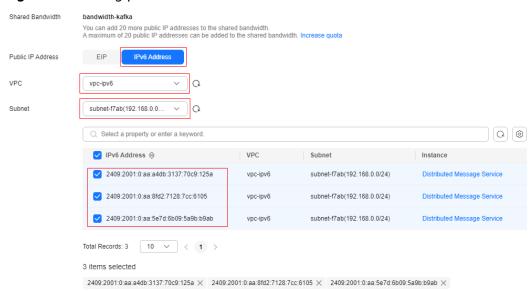


Figure 5-4 Adding public IPs

Step 10 After the shared bandwidth is configured, set a Kafka instance security group with the rules described in **Table 5-6**.

Table 5-6 Kafka instance security group rules (IPv6 access)

Directi on	Protoc ol	Туре	Port	Source	Description
Inboun d	ТСР	IPv6	9192	::/0	Accessing a Kafka instance using IPv6 addresses (without SSL encryption)
Inboun d	ТСР	IPv6	9193	::/0	Accessing a Kafka instance using IPv6 addresses (with SSL encryption)

When a client is connected to a Kafka instance over an IPv6 public network:

- The Kafka connection addresses are the IPv6 addresses in **Instance Address** (**Private Network**).
- The client NIC must be added to shard bandwidth. Shared bandwidth is using a connected network. The shared bandwidth of the client NIC and that of the Kafka instance can be different.

----End

Disabling Public Access to a Kafka Instance

On the Kafka console, the procedures for disabling public IPv4 access vary depending on the content displayed in the **Connection** area on the **Overview** page.

- To disable IPv4 public access when IPv6 is enabled, see section "Disabling Public IPv4 Access (SASL Cannot Be Changed)".
- To disable IPv4 public access when IPv6 is disabled, see section "Disabling Public IPv4 Access (Plaintext or Ciphertext Access Can Be Changed)".

Disabling Public IPv4 Access (SASL Cannot Be Changed)

- Step 1 Log in to the Kafka console.
- **Step 2** Click on the upper left corner to select the region where your instance is located.
- **Step 3** Click the name of a Kafka instance to go to the **Overview** page.
- **Step 4** Click next to **Public Access**.

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

After public access is disabled, configure security group rules listed in **Table 5-7** before attempting to access Kafka in a VPC. For details about accessing Kafka, see **Connecting to an Instance**.

Table 5-7 Kafka instance security group rules (private access)

Directi on	Protoc ol	Туре	Port	Source	Description
Inboun d	ТСР	IPv4	9092	IP address or IP address group of the Kafka client	Accessing a Kafka instance over a private network within a VPC (without SSL)
Inboun d	ТСР	IPv4	9093	IP address or IP address group of the Kafka client	Accessing a Kafka instance over a private network within a VPC (with SSL)

□ NOTE

After a security group is created, its default inbound rule allows communication among ECSs within the security group and its default outbound rule allows all outbound traffic. In this case, you can access a Kafka instance within a VPC, and do not need to add rules according to Table 5-7.

----End

Disabling Public IPv4 Access (Plaintext or Ciphertext Access Can Be Changed)

- **Step 1** Log in to the **Kafka console**.
- **Step 2** Click on the upper left corner to select the region where your instance is located.
- **Step 3** Click the name of a Kafka instance to go to the **Overview** page.
- Step 4 Before disabling public access, disable Plaintext Access and Ciphertext Access next to Public Network Access. Then click next to Public Access.
- **Step 5** Click **OK**. The **Background Tasks** page is displayed. If the status of the task turns to **Successful**, public access is successfully disabled.

After public access is disabled, configure security group rules listed in **Table 5-8** before attempting to access Kafka in a VPC. For details about accessing Kafka, see **Connecting to an Instance**.

◯ NOTE

After a security group is created, its default inbound rule allows communication among ECSs within the security group and its default outbound rule allows all outbound traffic. In this case, you can access a Kafka instance within a VPC, and do not need to add rules according to Table 5-8.

Table 5-8 Kafka instance security group rules (private access)

Directi on	Protoc ol	Туре	Port	Source	Description
Inboun d	TCP	IPv4	9092	IP address or IP address group of the Kafka client	Accessing a Kafka instance over a private network within a VPC (in plaintext)
Inboun d	ТСР	IPv4	9093	IP address or IP address group of the Kafka client	Accessing a Kafka instance over a private network within a VPC (in ciphertext)

----End

Disabling IPv6 Public Access

Remove the IPv6 addresses of a Kafka instance from the shared bandwidth. For details, see **Removing EIPs from a Shared Bandwidth**.

5.1.3 Accessing Kafka Using a VPC Endpoint Across VPCs

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

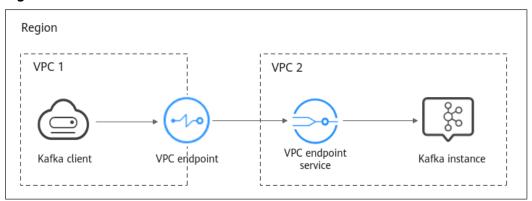
- Establish a VPC peering connection to allow two VPCs to communicate with each other. For details, see VPC Peering Connection.
- Use VPC Endpoint (VPCEP) to establish a cross-VPC connection.

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

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

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

Figure 5-5 Cross-VPC access to a Kafka instance



Is Plaintext Access or Ciphertext Access Used When a Client Accesses Kafka Across VPCs Using A VPC Endpoint?

It depends on **Cross-VPC Access Protocol**. The cross-VPC access protocol can be configured when you create a Kafka instance. After an instance is created, the setting cannot be changed.

Options:

- PLAINTEXT: There is no authentication required in such a connection and data is transmitted in plaintext.
- SASL_SSL: Clients can connect to a Kafka instance with SASL and the data will be encrypted using the SSL certificate.
- SASL_PLAINTEXT: Clients can connect to a Kafka instance with SASL and the data will be transmitted in plaintext.

Creating a VPC Endpoint Service

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

Network

- **Step 2** Click in the upper left corner to select the region where your instance is located.
- **Step 3** Click the desired instance to go to the instance details page.
- **Step 4** In the **Advanced Settings** area on the **Overview** page, obtain the listeners IP addresses and port IDs of the instance for **Cross-VPC Access**.

Figure 5-6 Cross-VPC access–related listeners IP addresses and corresponding port IDs of the Kafka instance



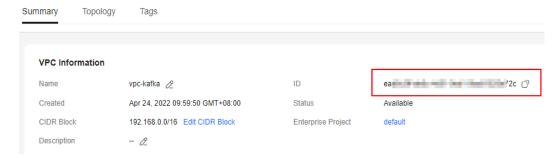
Step 5 In the **Network** area on the **Overview** page, view the VPC to which the Kafka instance belongs.

Figure 5-7 Viewing the VPC to which the Kafka instance belongs

AZ AZ1,AZ3,AZ2 VPC vpc-kafka 🖒 Subnet subnet-kafka Security Group sg-kafka 🖒 🖉 IPv6 Disabled

Step 6 Click the VPC to obtain the VPC ID on the VPC console.

Figure 5-8 Obtaining the VPC ID



Step 7 Call the VPC Endpoint API to create a VPC endpoint service. Set request parameters by referring to **Table 5-9**, and other parameters as required. For details, see **Creating a VPC Endpoint Service**.

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

Table 5-9 VPC Endpoint service creation parameters

Parameter	Description
port_id	ID of the backend resource.
	Enter a port ID obtained in Step 4.
vpc_id	ID of the VPC of the backend resource.
	Enter a VPC ID obtained in Step 6.
server_type	Resource type.
	Enter VM .
client_port	Access port.
	Enter 9011 .
server_port	Service port.
	Enter 9011 .
protocol	Port mapping protocol.
	Enter TCP.
approval_enabled	Whether approval is required.
	Entering false indicates that no approval is required and the connected VPC endpoint will be in the accepted state.
service_type	Service type.
	Enter interface .
endpoint	VPCEP endpoint obtained from Regions and Endpoints . The region must be the same as that of the Kafka instance.
project_id	project ID obtained from Obtaining a Project ID . The region must be the same as that of the Kafka instance.

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

Step 8 Repeat **Step 7** to create VPC endpoint services for other port IDs obtained in **Step 4** and record the VPC endpoint service names.

----End

(Optional) Adding a Whitelist

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

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

Buying a VPC Endpoint

- Step 1 Click in the upper left corner of the console. Then choose Network > VPC Endpoint.
- Step 2 Click Buy VPC Endpoint.
- **Step 3** Set parameters by referring to **Table 5-10**. Retain the default values for other parameters. For details, see **Buying a VPC Endpoint**.

Regions are geographic areas isolated from each other. Resources are region-specific and cannot be used across regions through internal * Billing Mode Pay-per-use ② Cloud service * Service Category * VPC Endpoint Service Name 21560-efd2-4bb8-84ed-5d707302 Service name found. Service Type: Interface Create a Private Domain Name ② vpc-kafka(192.168.0.0/16) View VPCs * Subnet subnet-kafka(192.168.0.0/... ∨ Q View Subnets Available IP Addresses: 248 * IPv4 Address Automatically assign IP address Manually specify IP address ? Access Control It is recommended that you use TMS's predefined tag function to add the same tag to different cloud resources. View predefined tags. Q Tag You can add 20 more tags. Description

Figure 5-9 VPC endpoint parameters

Table 5-10 VPC endpoint creation parameters

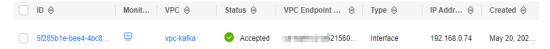
Parameter	Description		
Region	Region where the endpoint is located. Select the region that the Kafka instance is in.		
Service Category	 Cloud services: Select this option if the VPC endpoint service to be accessed is a cloud service. Find a service by name: Select this option if the VPC endpoint service to be accessed is a private service of your own. Select Find a service by name. 		

0/512

Parameter	Description
VPC Endpoint Service Name	Enter the VPC endpoint service name recorded in Step 7 and click Verify . If Service name found is displayed, proceed with subsequent operations.
(Optional) Create a Private Domain Name	If you want to access a VPC endpoint using a domain name, select Create a Private Domain Name .
VPC	VPC where the endpoint is located. Select the VPC that the Kafka client is in.
Subnet	Subnet where the endpoint is located. Select the subnet that the Kafka client is in.
IPv4 Address	IPv4 address of the endpoint. Select Automatically assign IP address.

- Step 4 Click Next.
- **Step 5** Confirm the configurations and submit the request.
- **Step 6** Go back to the VPC endpoint list and check whether the status of the created VPC endpoint has changed to **Accepted**. The **Accepted** state means that the VPC endpoint has been connected to the VPC endpoint service.

Figure 5-10 Checking the VPC endpoint status



Step 7 Click the VPC endpoint ID. On the **Summary** tab page, obtain the private IP address. You can use the private IP address to access the VPC endpoint service.

To access a VPC endpoint using a domain name, view and record the private domain name on the **Basic Information** page.



To copy a private domain name, do not carry the last period (.). Correct domain name format: **vpcep-5**xxx**7-e302-4ab4-bc6c-54**xxx**e52**.xx-xx-xx.xxx.com.

Figure 5-11 Viewing the private IP address



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

To access a VPC endpoint using a domain name, record the private domain name.

----End

Modifying Parameter advertised.listeners IP

- Step 1 Click = and choose Middleware > Distributed Message Service (for Kafka) to open the Kafka overview page.
- **Step 2** In the navigation pane, choose **Kafka Instances**.
- **Step 3** Click the desired Kafka instance to view its details.
- Step 4 In the Advanced Settings area on the Overview page, click Modify for Cross-VPC
 Access to change the value of advertised.listeners IP address/domain name to
 the private IP addresses recorded in Step 7 and Step 8. Each IP address must
 match the corresponding port ID. Otherwise, the network will be
 disconnected. After the modification, click Save.

To access an endpoint using a domain name, in the **Advanced Settings** area on the **Overview** page, click **Modify**. In the **advertised.listeners IP Address/Domain Name** column, click = to switch to the domain name mode. Change it to the private domain name recorded in **Step 7** and **Step 8**. **Each private domain name must match the corresponding port ID. Otherwise, the network will be disconnected.**

Figure 5-12 Changing the advertised.listeners IP addresses



----End

Verifying Connectivity

Check whether messages can be created and retrieved by referring to Connecting to Kafka Using the Client (Plaintext Access) or Connecting to Kafka Using the Client (Ciphertext Access).

Notes:

• The address for connecting to a Kafka instance is in the format of "advertised.listeners IP:9011". For example, the addresses for connecting to the Kafka instance shown in Figure 5-12 are 192.168.0.71:9011,192.168.0.11:9011,192.168.0.21:9011.

To access an endpoint using a domain name, the address of the Kafka instance is *domain name*:9011. For example, vpcep-5xxx7-e302-4ab4-bc6c-54xxxe52.xx-xx-xxx.com:9011.

- Configure inbound rules for the security group of the Kafka instance to allow access from 198.19.128.0/17 over port 9011.
- If a network access control list (ACL) has been configured for the subnet of this instance, configure inbound rules for the network ACL to allow access from 198.19.128.0/17 and from the subnet used by the VPC endpoint.

∩ NOTE

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

5.1.4 Accessing Kafka in a Public Network Using DNAT

Enable public access in either of the following ways:

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

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

Prerequisites

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

Step 1: Obtain Information About the Kafka Instance

- **Step 1** Log in to the **Kafka console**.
- **Step 2** Click in the upper left corner to select the region where your instance is located.
- **Step 3** Click the desired instance to go to the instance details page.
- **Step 4** In the **Connection** area on the **Overview** page, view and record the private network access addresses of the Kafka instance. In the **Network** area, view and record the VPC and subnet where the Kafka instance is located.

Connection kafka-test 19 Username Status Running a0af5cb3-c7fd-4f2d-ac1f-fe924b9d3c0c 🗇 Address (Private Network, Plaintext) 192.168.0.50:9092,... Disabled kafka.2u4g.cluster.small * 3 broker Maximum Partitions 300 AZ1,AZ3,AZ2 vpc-kafka 🕜 Smart Connect Automatic Topic Creation (?) Disabled sg-kafka 🕜 🖉 Security Group May 20, 2024 09:11:32 GMT+08:00 IPv6 - 2 default 🕜 🖉

Figure 5-13 Kafka instance information

----End

Step 2: Buy a Public NAT Gateway

- Step 1 Click in the upper left corner of the management console and choose Network > NAT Gateway. The Public NAT Gateways page is displayed.
- Step 2 Click Buy Public NAT Gateway.
- **Step 3** Set parameters by referring to **Table 5-11** and other parameters as required. For details, see **Buying a Public NAT Gateway**.

Figure 5-14 Buying a public NAT gateway

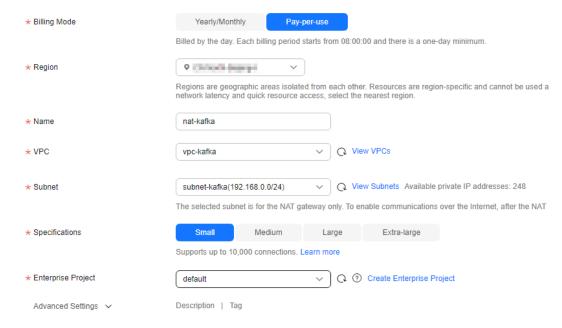


Table 2 11 rabile risk gaterial election parameters		
Parameter	Description	
Region	Region where the public NAT gateway is located. Select the region that the Kafka instance is in.	
Name	Enter a name for the public NAT gateway. Enter up to 64 characters. Only letters, digits, underscores (_), hyphens (-), and periods (.) are allowed.	
VPC	VPC where the public NAT gateway resides. Select the VPC recorded in Step 4 .	
Subnet	Subnet in the VPC where the public NAT gateway resides. Select the subnet recorded in Step 4 .	
Enterprise Project	Enterprise project that the public NAT gateway belongs to. Select as required.	

Table 5-11 Public NAT gateway creation parameters

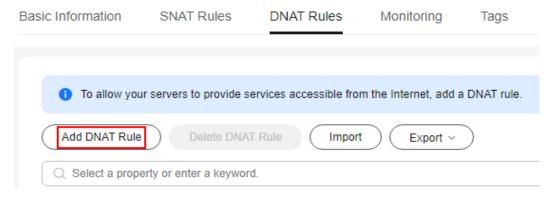
- Step 4 Click Next.
- **Step 5** Confirm the specifications. If you have selected the yearly/monthly billing mode, click **Pay Now** and make the payment as prompted. If you have selected the payper-use mode, click **Submit**.

----End

Step 3: Add a DNAT Rule

- **Step 1** On **Public NAT Gateways** page, locate the row containing the newly purchased public NAT gateway and click **Configure Rules** in the **Operation** column.
- Step 2 On the DNAT Rules tab page, click Add DNAT Rule.

Figure 5-15 Public NAT gateway details



Step 3 Set parameters by referring to **Table 5-12**. For details about more parameters, see **Adding a DNAT Rule**.

Add DNAT Rule Public NAT Gateway Name nat-kafka Direct Connect/Cloud Connect * Scenario ★ Port Type Specific port All ports * Protocol TCP ~ ★ Public IP Address Type EIP Global EIP ∨ Q View EIP ⑦ 12 = 54(1 Mbit/s | Pay-per-use | default) Bandwidth: 1 Mbit/s Billing Mode: Pay-per-use Enterprise Project: default ★ Outside Port 9011 ★ Instance Type Virtual IP address Custom ★ Private IP Address ★ Inside Port 9011 Description

Figure 5-16 Adding a DNAT rule

Table 5-12 Adding a DNAT rule

Parameter	Description	
Scenario	Select VPC . The servers in a VPC will share an EIP to provide services accessible from the Internet through the DNAT rule.	
Port Type	Select Specific port . The public NAT gateway forwards requests to your servers only from the outside port and to the inside port configured here, and only if they use the right protocol.	
Protocol	Select TCP.	
Public IP Address Type	The type of the public IP address used for accessing the Internet	
	Select EIP and select the purchased EIP from the drop-down list.	
Outside Port	Enter 9011 .	
Instance Type	Instance type for providing services over external publinetworks. Select Custom .	
Private IP Address	Enter one of the private network addresses of the Kafka instance recorded in Step 4 .	

Parameter	Description
Inside Port	Enter 9011 .

Step 4 Click OK.

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

Step 5 Create DNAT rules for other private network addresses of the Kafka instance recorded in **Step 4**. **Configure a unique EIP for each DNAT rule.**

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

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

Figure 5-17 DNAT rule list



----End

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

- Step 1 Click and choose Middleware > Distributed Message Service (for Kafka) to open the Kafka overview page.
- **Step 2** In the navigation pane, choose **Kafka Instances**.
- **Step 3** Click the desired Kafka instance to view its details.
- **Step 4** In the **Advanced Settings** area on the **Overview** page, click **Modify**.
- **Step 5** Change the values of **advertised.listeners IP Address/Domain Name** to the EIPs in the DNAT rules. Ensure that the mapping between the private network addresses and the EIPs is consistent with that recorded in **Step 6**. Then click **Save**.

Figure 5-18 Changing the advertised.listeners IP address (for DNAT access)



----End

Step 5: Verify Connectivity

Check whether messages can be created and retrieved by referring to Connecting to Kafka Using the Client (Plaintext Access) or Connecting to Kafka Using the Client (Ciphertext Access).

Notes:

- The address for connecting to a Kafka instance is in the format of "advertised.listeners IP:9011". For example, the addresses for connecting to the Kafka instance shown in Figure 5-18 are 124.xxx.xxx.167:9011,124.xxx.xxx.174:9011,124.xxx.xxx.57:9011.
- Configure security group rules for the Kafka instance to allow inbound access over port **9011**.
- Public access must be enabled on the client connected to the Kafka instance.

5.2 Configuring Kafka Access Control

5.2.1 Configuring Plaintext or Ciphertext Access to Kafka Instances

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

- Plaintext access: Clients connect to the Kafka instance without SASL authentication.
- Ciphertext access: Clients connect to the Kafka instance with SASL authentication.

Notes and Constraints

- When you change the access mode for the first time, some instances will restart. You can see the actual situation on the console. The restart takes about 75–80s. The instance will not be restarted when the access mode is changed again.
- For a single-node instance, you can only enable or disable plaintext for public network access.
- The access mode cannot be changed for instances with IPv6 enabled.

Prerequisites

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

Configuring Plaintext Access to a Kafka Instance

A client connects to a Kafka instance without SASL authentication.

Enabling Plaintext Access

- Step 1 Log in to the Kafka console.
- **Step 2** Click on the upper left corner to select the region where your instance is located.
- **Step 3** Click the name of a Kafka instance to go to the **Overview** page.
- **Step 4** An instance can be accessed in plaintext over the private network and public network. For details about how to enable plaintext access, see **Table 5-13**.

Table 5-13 Enabling plaintext access

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

----End

Disabling Plaintext Access

- Step 1 Log in to the Kafka console.
- **Step 2** Click in the upper left corner to select the region where your instance is located.
- **Step 3** Click the name of a Kafka instance to go to the **Overview** page.
- **Step 4** An instance can be accessed in plaintext over the private network and public network. For details about how to disable plaintext access, see **Table 5-14**.

Table 5-14 Disabling plaintext access

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

----End

Configuring Ciphertext Access to a Kafka Instance

A client connects to a Kafka instance with SASL authentication.

Enabling Ciphertext Access

- Step 1 Log in to the Kafka console.
- **Step 2** Click on the upper left corner to select the region where your instance is located.
- **Step 3** Click the name of a Kafka instance to go to the **Overview** page.
- **Step 4** An instance can be accessed in ciphertext over the private network and public network. For details about how to enable ciphertext access, see **Table 5-15**.

Table 5-15 Enabling ciphertext access

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

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

Table 5-16 Ciphertext access parameters

Parameter	Value	Description
Security Protocol	SASL_SSL	SASL is used for authentication. Data is encrypted with SSL certificates for high-security transmission.

Parameter	Value	Description
	SASL_PLAINTEX T	SASL is used for authentication. Data is transmitted in plaintext for high performance.
		SCRAM-SHA-512 authentication is recommended for plaintext transmission.
SASL/PLAIN	-	• If SASL/PLAIN is disabled, the SCRAM-SHA-512 mechanism is used for username and password authentication.
		• If SASL/PLAIN is enabled, both the SCRAM-SHA-512 and PLAIN mechanisms are supported. You can select either of them as required.
		The SASL/PLAIN setting cannot be changed once ciphertext access is enabled.
		What are SCRAM-SHA-512 and PLAIN mechanisms?
		 SCRAM-SHA-512: uses the hash algorithm to generate credentials for usernames and passwords to verify identities. SCRAM- SHA-512 is more secure than PLAIN.
		PLAIN: a simple username and password verification mechanism.
Username and Password	-	Username and password used by the client to connect to the Kafka instance.
		A username should contain 4 to 64 characters, start with a letter, and contain only letters, digits, hyphens (-), and underscores (_).
		A password must meet the following requirements:
		Contains 8 to 32 characters.
		 Contains at least three types of the following characters: uppercase letters, lowercase letters, digits, and special characters `~!@#\$%^&*()=+\ [{}];:'",<.>? and spaces, and cannot start with a hyphen (-).
		 Cannot be the username spelled forward or backward.
		The username cannot be changed once ciphertext access is enabled.

The Kafka security protocol, SASL/PLAIN mechanism, username, and password are required when the client accesses a Kafka instance with ciphertext access enabled. For details, see **Connecting to Kafka Using the Client (Ciphertext Access)**.

----End

Disabling Ciphertext Access

- Step 1 Log in to the Kafka console.
- **Step 2** Click in the upper left corner to select the region where your instance is located.
- **Step 3** Click the name of a Kafka instance to go to the **Overview** page.
- **Step 4** An instance can be accessed in ciphertext over the private network and public network. For details about how to disable ciphertext access, see **Table 5-17**.

Table 5-17 Disabling ciphertext access

Access Method	Disabling Plaintext Access	
Private network ciphertext access	Click next to Ciphertext Access in the Private Network Access area.	
	2. Click OK . The Background Tasks page is displayed. If the status of the task turns to Successful , ciphertext access is successfully disabled.	
Public network ciphertext access	Click next to Ciphertext Access in the Public Network Access area.	
	2. Click OK . The Background Tasks page is displayed. If the status of the task turns to Successful , ciphertext access is successfully disabled.	

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

----End

5.2.2 Generating and Replacing an SSL Kafka Certificate in JKS Format

The SSL certificate secures data transmission through encryption between a client and an instance.

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

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



Step 1: Generating a Certificate to Step 4: Verifying the Certificate describe how to make an SSL certificate applicable to the scenario where certificate domain name verification is not enabled, and how to replace with the certificate. To make an SSL certificate applicable to the scenario where certificate domain name verification is enabled, see (Optional) Making and Replacing an SSL Certificate with Domain Name Verification Enabled.

Notes and Constraints

Replacing the certificate will restart the instance. Exercise caution.

Prerequisites

- A Linux server is available. The server must install Java Development Kit
 1.8.111 or later and JAVA_HOME and PATH environment variables are configured.
- Kafka SASL SSL has been enabled for the instance.
- (Optional) To generate an SSL certificate with domain name verification enabled, obtain the connection address from the Connection area on the Kafka instance details page.

Step 1: Generating a Certificate

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

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

Enter the keystore password as prompted and record the password for later use.

The password must meet the following requirements:

- Contains 8 to 32 characters.
- Contains at least three of the following character types: letters, digits, spaces, and special characters `-!@#\$ %^&*()-_=+\|[{}]:'",<.>/? and does not start with a hyphen (-).
- Cannot be a weak password. To check whether a password is weak, enter it in Step 5.

Enter the information about the certificate owner as prompted, such as the name, company, organization, city, and country or region.

[root@ecs-kafka ~]# keytool -genkey -keystore server.keystore.jks -alias localhost -validity 3650 -keyalg RSA Enter keystore password:

Re-enter new password:

What is your first and last name?

[Unknown]: Tom

What is the name of your organizational unit?

[Unknown]: test

```
What is the name of your organization?
[Unknown]: test01
What is the name of your City or Locality?
[Unknown]: nj
What is the name of your State or Province?
[Unknown]: js
What is the two-letter country code for this unit?
[Unknown]: xx
Is CN=Tom, OU=test, O=test01, L=nj, ST=js, C=xx correct?
[no]: y
```

Step 2 Run the following command to generate a CA:

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

Enter the PEM password as prompted and record the password for later use.

The password must meet the following requirements: 4 to 1024 characters.

Enter the information about the certificate owner as prompted, such as the country or region, city, organization, company, name, and email.

```
[root@ecs-kafka ~]# openssl req -new -x509 -keyout ca-key -out ca-cert -days 3650
Generating an RSA private key
.....++++
writing new private key to 'ca-key'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
You are about to be asked to enter information that will be incorporated into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are guite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
Country Name (2 letter code) [XX]:xx
State or Province Name (full name) []:js
Locality Name (eg, city) [Default City]:nj
Organization Name (eg, company) [Default Company Ltd]:test01
Organizational Unit Name (eg, section) []:test
Common Name (eg, your name or your server's hostname) []:Tom
Email Address []:xx
[root@ecs-kafka ~]#
```

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

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

Enter the truststore password of the server certificate as prompted and record the password for later use.

The password must meet the following requirements:

- Contains 8 to 32 characters.
- Contains at least three of the following character types: letters, digits, spaces, and special characters `-!@#\$ %^&*()-_=+\|[{}]:'",<.>/? and does not start with a hyphen (-).
- Cannot be a weak password. To check whether a password is weak, enter it in **Step 5**.

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

Trust this certificate?

Step 4 Run the following command to create a client truststore certificate with the CA: keytool -keystore client.truststore.jks -alias CARoot -import -file ca-cert

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

The password must meet the following requirements:

- Contains 8 to 32 characters.
- Contains at least three of the following character types: letters, digits, spaces, and special characters `-!@#\$ %^&*()-_=+\|[{}]:'",<.>/? and does not start with a hyphen (-).
- Cannot be a weak password. To check whether a password is weak, enter it in Step 5.

Enter **y** when the following information is displayed: Trust this certificate?

Step 5 Sign the server certificate.

- Export the server certificate server.cert-file. keytool -keystore server.keystore.jks -alias localhost -certreq -file server.cert-file
 Enter the keystore password set in Step 1 as prompted.
- 2. Sign the server certificate with the CA.
 openssl x509 -req -CA ca-cert -CAkey ca-key -in server.cert-file -out server.cert-signed -days 3650 CAcreateserial

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

3. Import the CA certificate to the server keystore. keytool -keystore server.keystore.jks -alias CARoot -import -file ca-cert

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

Enter **y** when the following information is displayed: Trust this certificate?

4. Import the signed server certificate to the server keystore. keytool -keystore server.keystore.jks -alias localhost -import -file server.cert-signed Enter the keystore password set in **Step 1** as prompted.

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

The **server.keystore.jks** and **server.truststore.jks** files are used to replace the keystore and truststore files in subsequent step **Replacing a Certificate**. Store **client.truststore.jks** in a specific location on the client. Record the storage path and it is the value of **ssl.truststore.location** in the configuration file used by the client to connect to the Kafka instance.

Figure 5-19 Certificate directory

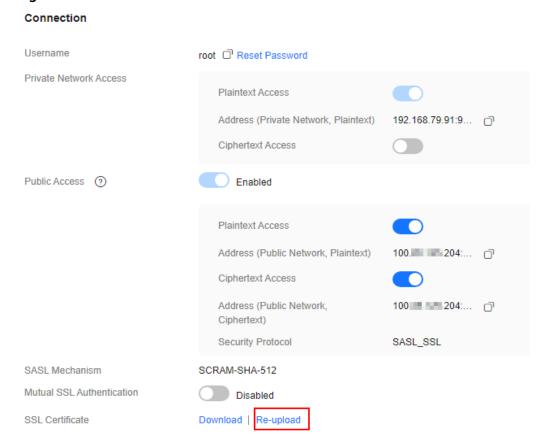
```
total 44
drwxr-xr-x
            2 root root 4096 Aug 10 15:20 ./
drwxr-xr-x 10 root root 4096 Aug
                                 8 17:04 ../
            1 root root 1322 Aug
                                 8 17:07 ca-cert
                          41 Aug
            1 root root
              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

----End

Step 2: Replacing a Certificate

- Step 1 Log in to the Kafka console.
- **Step 2** Click in the upper left corner to select the region where your instance is located.
- **Step 3** Click the desired instance to view its details.
- **Step 4** In the **Connection** area, click **Re-upload** next to **SSL Certificate**.

Figure 5-20 Connection information



Step 5 Set the parameters for replacing the SSL certificate by referring to **Table 5-18**.

Figure 5-21 Replacing the SSL certificate

Replace SSL certificate

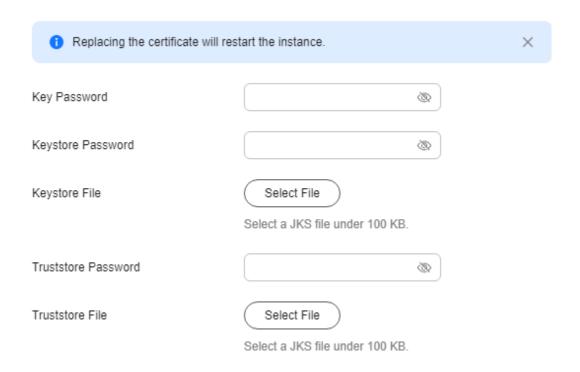


Table 5-18 Parameters for replacing the SSL certificate

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

Step 6 Click OK.

Step 7 Click OK.

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

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

----End

Step 3: Modifying Client Configuration Files

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

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

Table 5-19 Configuration file parameters

Parameter	Description	
ssl.truststore.location	Path for storing the client.truststore.jks certificate	
ssl.truststore.password	truststore password of the client certificate	
ssl.endpoint.identification.algo rithm	Whether to verify the certificate domain name This parameter must be left blank, which indicates disabling domain name verification.	

Step 4: Verifying the Certificate

Produce and consume messages by referring to Connecting to Kafka Using the Client (Ciphertext Access). The new certificate takes effect if the operation is successful.

(Optional) Making and Replacing an SSL Certificate with Domain Name Verification Enabled

MARNING

When certificate domain name verification is enabled, changing the instance address may cause a client to fail to connect to the instance. The changing operations include changing the private network address to a public network address; changing the public network address to a private network address; and increasing the number of brokers.

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

keytool -genkey -keystore server.keystore.jks -alias localhost -validity 3650 -keyalg RSA -ext SAN=IP:xxx.xxx.xx,IP:xxx.xxx,IP:xxx.xxx.xx

The IP address is the **IP address in the Kafka instance connection address**, which is obtained from **Prerequisites**. Note: The connection address of the Kafka instance contains **IP address: Port**, for example,

192.168.10.10:9093,192.168.10.11:9093,192.168.10.12:9093. The preceding command should be modified as follows:

keytool -genkey -keystore server.keystore.jks -alias localhost -validity 3650 -keyalg RSA -ext SAN=IP:192.168.10.10,IP:192.168.10.11,IP:192.168.10.12

 Enter the keystore password as prompted and record the password for later use.

The password must meet the following requirements:

- Contains 8 to 32 characters.
- Contains at least three of the following character types: letters, digits, spaces, and special characters `-!@#\$ %^&*()-_=+\|[{}]:'",<.>/? and does not start with a hyphen (-).
- Cannot be a weak password. To check whether a password is weak, enter it in Step 5.
- 2. Enter the information about the certificate owner as prompted, such as the name, company, organization, city, and country or region.

[root@ecs-kafka ~]# keytool -genkey -keystore server.keystore.jks -alias localhost -validity 3650 - keyalg RSA -ext SAN=IP:192.168.10.10,IP:192.168.10.11,IP:192.168.10.12

Enter keystore password:

Re-enter new password:

What is your first and last name?

[Unknown]: Tom

What is the name of your organizational unit?

[Unknown]: test

What is the name of your organization?

[Unknown]: test01

What is the name of your City or Locality?

[Unknown]: nj

What is the name of your State or Province?

[Unknown]: js

What is the two-letter country code for this unit?

[Unknown]: xx

Is CN=Tom, OU=test, O=test01, L=nj, ST=js, C=xx correct?

[no]:

3. Press **Enter** as prompted to set the key password to the same as the keystore password.

Enter key password for <localhost> (RETURN if same as keystore password):

Step 2 Run the following command to generate a CA private key:

openssl genrsa -out ca.key 2048

Step 3 Run the following command to generate a CA certificate:

openssl reg -new -x509 -key ca.key -out ca.crt -days 3650 -subj "/CN=KafkaTestCA"

Step 4 Run the following command to generate a server certificate:

keytool -keystore server.keystore.jks -alias localhost -certreq -file server.csr -storepass xxx

storepass indicates the password of the keystore. Enter the keystore password set in **Step 1.1**.

- **Step 5** Use the CA to sign the server certificate.
 - Run the following command to create a sans.ext file: touch sans.ext

2. Run the following command to edit the **sans.ext** file and add the following content:

vim sans.ext

Add the following content:

subjectAltName=IP:xxx.xxx.xx.xx,IP:xxx.xxx.xx,IP:xxx.xxx.xx

The IP address is the **IP address in the Kafka instance connection address**, which is obtained from **Prerequisites**.

3. Run the following command to use the CA to sign the server certificate: openssl x509 -req -CA ca.crt -CAkey ca.key -in server.csr -out server.crt -days 3650 -CAcreateserial - extfile sans.ext

The returned information is as follows:

[root@ecs-kafka ~]# openssl x509 -req -CA ca.crt -CAkey ca.key -in server.csr -out server.crt -days 3650 -CAcreateserial -extfile sans.ext Signature ok subject=C=xx, ST=js, L=nj, O=test01, OU=test, CN=Tom Getting CA Private Key

Step 6 Import the CA certificate to the server keystore.

keytool -keystore server.keystore.jks -alias CARoot -import -file ca.crt -storepass xxx -noprompt

storepass indicates the password of the keystore. Enter the keystore password set in **Step 1.1**.

The returned information is as follows:

[root@ecs-kafka ~]# keytool -keystore server.keystore.jks -alias CARoot -import -file ca.crt -storepass xxx - noprompt Certificate was added to keystore

Step 7 Import the server certificate to the server keystore.

keytool -keystore server.keystore.jks -alias localhost -import -file server.crt -storepass xxx -noprompt

storepass indicates the password of the keystore. Enter the keystore password set in **Step 1.1**.

The returned information is as follows:

[root@ecs-kafka ~]# keytool -keystore server.keystore.jks -alias localhost -import -file server.crt -storepass xxx -noprompt
Certificate reply was installed in keystore

Step 8 Create a server truststore certificate with the CA.

keytool -keystore server.truststore.jks -alias CARoot -import -file ca.crt -storepass xxx -noprompt

storepass is the password of the server truststore certificate. Set the password by complying with the following requirements and record it for later use.

The password must meet the following requirements:

- Contains 8 to 32 characters.
- Contains at least three of the following character types: letters, digits, spaces, and special characters `-!@#\$ %^&*()-_=+\|[{}]:'",<.>/? and does not start with a hyphen (-).
- Cannot be a weak password. To check whether a password is weak, enter it in Step 5.

The returned information is as follows:

[root@ecs-kafka ~]# keytool -keystore server.truststore.jks -alias CARoot -import -file ca.crt -storepass xxx - noprompt

Certificate was added to keystore

Step 9 Create a client truststore certificate with the CA.

keytool -keystore client.truststore.jks -alias CARoot -import -file ca.crt -storepass xxx -noprompt

storepass is the password of the client truststore certificate. Set the password by complying with the following requirements and record it for later use.

The password must meet the following requirements:

- Contains 8 to 32 characters.
- Contains at least three of the following character types: letters, digits, spaces, and special characters $-!@\#$ \%^*()-=+|[{}]:",<.>/?$ and does not start with a hyphen (-).
- Cannot be a weak password. To check whether a password is weak, enter it in Step 5.

The returned information is as follows:

[root@ecs-kafka ~]# keytool -keystore client.truststore.jks -alias CARoot -import -file ca.crt -storepass xxx noprompt

Certificate was added to keystore

Step 10 Export the server.keystore.jks, server.truststore.jks, and client.truststore.jks certificates to the local PC.

The **server.keystore.jks** and **server.truststore.jks** files are used to replace the keystore and truststore files in subsequent step Replacing a Certificate. Store client.truststore.jks in a specific location on the client. Record the storage path and it is the value of **ssl.truststore.location** in the configuration file used by the client to connect to the Kafka instance.

Step 11 Replace the SSL certificate on the Kafka console by referring to Step 2: Replacing a Certificate.

	Table 5-20	Parameters	for rep	lacina	the SSL	certificat
--	------------	------------	---------	--------	---------	------------

Parameter	Description
Key Password	Enter the keystore password set in Step 1.1 .
Keystore Password	Enter the keystore password set in Step 1.1 .
Keystore File	Import the server.keystore.jks certificate.
Truststore Password	Enter the server truststore password set in Step 8 .
Truststore File	Import the server.truststore.jks certificate.

Step 12 Modify the client configuration file.

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

security.protocol=SASL_SSL ssl.truststore.location=/opt/kafka_2.12-2.7.2/config/client.truststore.jks ssl.truststore.password=axxxb ssl.endpoint.identification.algorithm=https

Table 5 21 configuration file parameters			
Parameter	Description		
ssl.truststore.location	Path for storing the client.truststore.jks certificate		
ssl.truststore.password	truststore password of the client certificate		
ssl.endpoint.identification.algo rithm	Certificate domain name verification setting. To enable it, set to "https".		

Table 5-21 Configuration file parameters

Step 13 Check whether the certificate has taken effect.

Produce and consume messages by referring to **Connecting to Kafka Using the Client (Ciphertext Access)**. The new certificate takes effect if the operation is successful.

----End

5.2.3 Obtaining and Using An SSL Kafka Certificate in PEM Format

This section describes how to obtain an SSL certificate in PEM format and use it to access a Kafka instance.

Prerequisite

SASL_SSL has been enabled for the Kafka instance.

Obtaining a PEM SSL Certificate

- Step 1 Log in to the Kafka console.
- **Step 2** Click in the upper left corner to select the region where your instance is located.
- **Step 3** Click the desired instance to go to the instance details page.
- **Step 4** Click **Download** next to **Connection** > **SSL Certificate**.
- **Step 5** Decompress the Zip package to obtain the PEM SSL certificate **client.pem**.

----End

Accessing a Kafka Instance Using a PEM Certificate

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

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

If the SASL mechanism is PLAIN, configure as follows: sasl.mechanism=PLAIN

```
sasl.jaas.config=org.apache.kafka.common.security.plain.PlainLoginModule required \
username="username" \
password="password";
# If the SASL mechanism is SCRAM-SHA-512, configure as follows:
sasl.mechanism=SCRAM-SHA-512
sasl.jaas.config=org.apache.kafka.common.security.scram.ScramLoginModule required \
username="username" \
password="password";
#Set the Kafka security protocol.
security.protocol=SASL_SSL
# ssl truststore.location is the path for storing the SSL certificate. The following code uses the path format
in Windows as an example. Change the path format based on the actual running environment.
ssl.truststore.location=E:\\temp\\client.pem
# ssl.truststore.password is the server certificate password. To access a Kafka instance using a PEM
certificate, skip this parameter.
#ssl.truststore.password=dms@kafka
# ssl.endpoint.identification.algorithm indicates whether to verify the certificate domain name. This
parameter must be left blank, which indicates disabling domain name verification.
ssl.endpoint.identification.algorithm=
# Add the ssl.truststore.type parameter to specify the client certificate type to PEM.
ssl.truststore.type=PEM
```

Related Documents

Is the Same SSL Certificate Used for Different Instances?

5.2.4 Configuring Mutual SSL Authentication for Kafka

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

Enable mutual SSL authentication to achieve high security.

Mutual SSL authentication is for restricted use. It can be enabled through a ticket.

Figure 5-22 shows the overall procedure for configuring mutual SSL authentication.

Disabling mutual SSL

authentication

Start

Disable mutual SSL

Modify client configuration files.

End

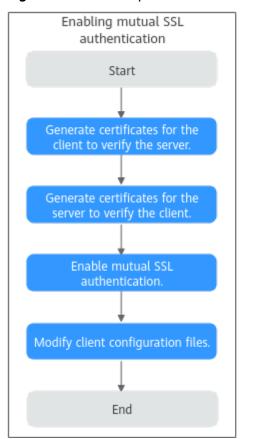


Figure 5-22 Overall procedure for configuring mutual SSL authentication

Notes and Constraints

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

Prerequisites

- A Linux server is available. The server must install Java Development Kit
 1.8.111 or later and JAVA_HOME and PATH environment variables are configured.
- Kafka SASL SSL has been enabled for the instance.

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

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

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

Enter the keystore password as prompted and record the password for later use.

The password must meet the following requirements:

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

 Cannot be a weak password. To check whether a password is weak, enter it in Step 5.

Enter the information about the certificate owner as prompted, such as the name, company, organization, city, and country or region.

```
[root@ecs-kafka ~]# keytool -genkey -keystore server.keystore.jks -alias localhost -validity 3650 -keyalg RSA
Enter keystore password:
Re-enter new password:
What is your first and last name?
 [Unknown]: Tom
What is the name of your organizational unit?
 [Unknown]: test
What is the name of your organization?
 [Unknown]: test01
What is the name of your City or Locality?
 [Unknown]: nj
What is the name of your State or Province?
[Unknown]: js
What is the two-letter country code for this unit?
 [Unknown]: xx
Is CN=Tom, OU=test, O=test01, L=nj, ST=js, C=xx correct?
[no]: y
```

Step 2 Run the following command to generate a CA:

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

Enter the PEM password as prompted and record the password for later use.

The password must meet the following requirements: 4 to 1024 characters.

Enter the information about the certificate owner as prompted, such as the country or region, city, organization, company, name, and email.

```
[root@ecs-kafka ~]# openssl req -new -x509 -keyout ca-key -out ca-cert -days 3650
Generating an RSA private key
.....++++
writing new private key to 'ca-key'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
You are about to be asked to enter information that will be incorporated into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are guite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
Country Name (2 letter code) [XX]:xx
State or Province Name (full name) []:js
Locality Name (eg, city) [Default City]:nj
Organization Name (eq. company) [Default Company Ltd]:test01
Organizational Unit Name (eg, section) []:test
Common Name (eg, your name or your server's hostname) []:Tom
Email Address []:xx
[root@ecs-kafka ~]#
```

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

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

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

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

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

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

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

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

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

Enter **y** when the following information is displayed: Trust this certificate?

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

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

Enter the password of **client.truststore.jks** as required and record the password for later use.

The password must meet the following requirements:

- Contains 8 to 32 characters.
- Contains at least three of the following character types: letters, digits, spaces, and special characters `-!@#\$ %^&*()-_=+\|[{}]:'",<.>/? and does not start with a hyphen (-).
- Cannot be a weak password. To check whether a password is weak, enter it in Step 5.

Enter **y** when the following information is displayed: Trust this certificate?

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

The **server.keystore.jks** file is used to replace the **keystore** file in the later step **Enable Mutual SSL Authentication**. **client.truststore.jks** must be stored on the client. Record the storage path and it is the value of **ssl.truststore.location** in the configuration file used by the client to connect to the Kafka instance.

----End

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

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

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

Enter the keystore password as prompted and record the password for later use.

The password must meet the following requirements:

- Contains 8 to 32 characters.
- Contains at least three of the following character types: letters, digits, spaces, and special characters `-!@#\$ %^&*()-_=+\|[{}]:'",<.>/? and does not start with a hyphen (-).
- Cannot be a weak password. To check whether a password is weak, enter it in **Step 5**.

Enter the information about the certificate owner as prompted, such as the name, company, organization, city, and country or region.

```
[root@ecs-kafka ~]# keytool -qenkey -keystore client.keystore.jks -alias localhost -validity 3650 -keyalq RSA
Enter keystore password:
Re-enter new password:
What is your first and last name?
 [Unknown]: Tom
What is the name of your organizational unit?
 [Unknown]: test
What is the name of your organization?
 [Unknown]: test01
What is the name of your City or Locality?
[Unknown]: nj
What is the name of your State or Province?
[Unknown]: is
What is the two-letter country code for this unit?
[Unknown]: xx
Is CN=Tom, OU=test, O=test01, L=nj, ST=js, C=xx correct?
[no]: y
```

Step 2 Run the following command to generate a CA:

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

Enter the PEM password as prompted and record the password for later use.

The password must meet the following requirements: 4 to 1024 characters.

Enter the information about the certificate owner as prompted, such as the country or region, city, organization, company, name, and email.

```
[root@ecs-kafka ~]# openssl req -new -x509 -keyout ca-key -out ca-cert -days 3650
Generating an RSA private key
.....++++
writing new private key to 'ca-key'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
You are about to be asked to enter information that will be incorporated into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
Country Name (2 letter code) [XX]:xx
State or Province Name (full name) []:js
Locality Name (eg, city) [Default City]:nj
Organization Name (eg, company) [Default Company Ltd]:test01
Organizational Unit Name (eg, section) []:test
Common Name (eg, your name or your server's hostname) []:Tom
Email Address []:xx
[root@ecs-kafka ~]#
```

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

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

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

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

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

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

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

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

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

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

Trust this certificate?

Step 6 Run the following command to enable the server to trust the client certificate: keytool -keystore server.truststore.jks -alias CARoot -import -file ca-cert

Enter the password of **server.truststore.jks** as prompted and record the password for later use.

The password must meet the following requirements:

- Contains 8 to 32 characters.
- Contains at least three of the following character types: letters, digits, spaces, and special characters `-!@#\$ %^&*()-_=+\|[{}]:'",<.>/? and does not start with a hyphen (-).
- Cannot be a weak password. To check whether a password is weak, enter it in Step 5.

Enter ${\bf y}$ when the following information is displayed:

Trust this certificate?

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

The **server.truststore.jks** file is used to replace the **truststore** file in the later step **Enable Mutual SSL Authentication**. **client.keystore.jks** must be stored on the client. Record the storage path and it is the value of **ssl.keystore.location** in the configuration file used by the client to connect to the Kafka instance.

----End

Step 3: Enable Mutual SSL Authentication

- Step 1 Log in to the Kafka console.
- **Step 2** Click in the upper left corner to select the region where your instance is located.
- **Step 3** Click the desired Kafka instance to go to the instance details page.
- **Step 4** In the **Connection** area, click next to **Mutual SSL Authentication**.
- **Step 5** In the displayed **Mutual SSL Authentication** dialog box, set the parameters by referring to **Table 5-22**.

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

Figure 5-23 Enabling mutual SSL authentication

Mutual SSL Authentication

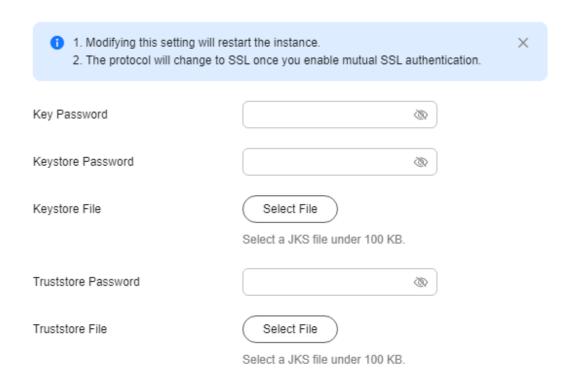


Table 5-22 Parameters for enabling mutual SSL authentication

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

Step 6 Click OK.

----End

Step 4: Modifying Client Configuration Files

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

security.protocol=SSL

ssl.truststore.location=/opt/kafka_2.12-2.7.2/config/client.truststore.jks

ssl.truststore.password=axxxb ssl.endpoint.identification.algorithm= # Add the following client certificate configurations: ssl.keystore.location=/var/private/ssl/kafka/client.keystore.jks ssl.keystore.password=txxx3 ssl.key.password=txxx3

Table 5-23 Configuration file parameters

Parameter	Description
security.protocol	Certificate protocol type. When enabling mutual SSL authentication, set this parameter to SSL .
ssl.truststore.location	Path for storing the client.truststore.jks certificate
ssl.truststore.password	password of client.truststore.jks
ssl.endpoint.identification.algo rithm	Whether to verify the certificate domain name This parameter must be left blank, which indicates disabling domain name verification.
ssl.keystore.location	Path for storing the client.keystore.jks certificate
ssl.keystore.password	password of client.keystore.jks
ssl.key.password	password of client.keystore.jks

Step 5: Verifying the Certificate

Produce and consume messages by referring to Connecting to Kafka Using the Client (Ciphertext Access). The new certificate takes effect if the operation is successful.

Disabling Mutual SSL Authentication

- Step 1 Log in to the Kafka console.
- **Step 2** Click in the upper left corner to select the region where your instance is located.
- **Step 3** Click the desired Kafka instance.
- **Step 4** In the **Connection** area, click next to **Mutual SSL Authentication**.



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

Step 5 After disabling mutual SSL authentication, modify the server certificate protocol and delete the client certificate configurations in the **consumer.properties** and **producer.properties** files on the client.

security.protocol=SASL SSL

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

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

----End

5.2.5 Configuring Kafka ACL Users

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

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

Notes and Constraints

- Single-node instances do not support user creation, user password reset, user information modification, or user deletion.
- The initial user set when ciphertext access is enabled for the first time cannot be deleted.
- Resetting a user password will interrupt services. Change the user password in the client configuration file or code as soon as possible.
- The maximum number of users that can be created for a Kafka instance is 20 or 500. Check the console for the actual limit.

Prerequisites

- Ciphertext access has been enabled for the Kafka instance.
- The Kafka instance is in the Running state.

Creating a User

- **Step 1** Log in to the **Kafka console**.
- **Step 2** Click on the upper left corner to select the region where your instance is located.
- **Step 3** Click the desired instance to go to the instance details page.

- **Step 4** In the navigation pane, choose **Instance** > **Users**.
- **Step 5** Click **Create User** in the upper left corner.
- **Step 6** Set user information by referring to **Table 5-24**.

Table 5-24 User creation parameters

Parameter	Description		
Username	The username used to access a Kafka instance, you can customize a name that complies with the rules: 4–64 characters; starts with a letter; can contain only letters, digits, hyphens (-), and underscores (_).		
Password	 The password used to access a Kafka instance. A password must meet the following requirements: Contains 8 to 32 characters. Contains at least three types of the following characters: uppercase letters, lowercase letters, digits, and special characters `~!@#\$%^&*()=+\ 		
	[{}];:'",<.>? and spaces, and cannot start with a hyphen (-).		
	Cannot be the username spelled forward or backward.		
Description	The description of a user. 0–200 characters.		

Step 7 Click OK.

View the new user on the user list page.

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

----End

Resetting a User Password

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

- Initial user: See section "Resetting the Password (for the Initial User)".
- Non-initial user: See section "Instance" > "Resetting the Password (for Non-initial Users)".

Resetting the Password (for the Initial User)

- Step 1 Log in to the Kafka console.
- **Step 2** Click in the upper left corner to select the region where your instance is located.

- **Step 3** Reset the password for the initial user in either of the following ways.
 - Choose **More** > **Reset Kafka Password** in the row containing the desired Kafka instance.
 - Click the desired Kafka instance to go to the instance details page. Choose -- > Reset Kafka Password in the upper right corner.
 - Click the desired Kafka instance to go to the instance details page. On the Overview page, click Reset Password next to Username in the Connection section.
 - Click the desired Kafka instance to go to the instance details page. On the Instance > Users page, click Reset Password in the row containing the desired user.
- **Step 4** Enter and confirm a new password, and click **OK**.
 - If the password is successfully reset, a success message is displayed.
 - If the password fails to be reset, a failure message is displayed. In this case, reset the password again. If you still fail to reset the password after multiple attempts, contact customer service.

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

----End

Resetting the Password (for Non-initial Users)

- Step 1 Log in to the Kafka console.
- **Step 2** Click on the upper left corner to select the region where your instance is located.
- **Step 3** Click the desired Kafka instance to go to the instance details page.
- **Step 4** On the **Instance** > **Users** page, click **Reset Password** in the row containing the desired user.
- **Step 5** Enter and confirm a new password, and click **OK**.
 - If the password is successfully reset, a success message is displayed.
 - If the password fails to be reset, a failure message is displayed. In this case, reset the password again. If you still fail to reset the password after multiple attempts, contact customer service.

Ⅲ NOTE

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

----End

Modifying User Information

- **Step 1** Log in to the **Kafka console**.
- **Step 2** Click oin the upper left corner to select the region where your instance is located.

- **Step 3** Click the desired instance to go to the instance details page.
- **Step 4** In the navigation pane, choose **Instance** > **Users**.
- **Step 5** In the row containing the desired user, click **Edit**.
- **Step 6** Modify the description and click **OK**.

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

----End

Deleting a User

- **Step 1** Log in to the **Kafka console**.
- **Step 2** Click on the upper left corner to select the region where your instance is located.
- **Step 3** Click the desired instance to go to the instance details page.
- **Step 4** In the navigation pane, choose **Instance** > **Users**.
- **Step 5** Delete a user in either of the following ways:
 - In the row containing the desired user, click **Delete**.
 - Select one or more users and click **Delete** above the list.
- **Step 6** In the displayed **Delete User** dialog box, click **OK** to delete the user.

The user is deleted if it is not displayed in the user list.

----End

Exporting the User List

- **Step 1** Log in to the **Kafka console**.
- **Step 2** Click in the upper left corner to select the region where your instance is located.
- **Step 3** Click the desired instance to go to the instance details page.
- **Step 4** In the navigation pane, choose **Instance** > **Users**.
- **Step 5** Export the user list in either of the following ways:
 - Select the desired users and choose Export > Export selected data to an XLSX file to export specified users.
 - Choose **Export** > **Export all data to an XLSX file** to export all users.

Note: The initial user cannot be selected manually. To export the initial user, choose **Export all data to an XLSX file**.

----End

Related Documents

- To create a user by calling an API, see Creating a User.
- To reset a user password by calling an API, see Resetting a User Password.
- To grant a user the permission to publish or subscribe to topics, see Configuring Kafka Topic Permissions.

5.3 Configuring the Kafka Client

5.3.1 Setting Parameters for Kafka Clients

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

 Table 5-25 Producer parameters

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

Paramet er	Default Value	Recommended Value	Description
retries	Client 2.1.0 and later: Integer.MAX_VALUE Earlier versions: 0		Number of times that the client resends a message. Setting this parameter to a value greater than zero will cause the client to resend any record that failed to be sent. Note that this retry is no different than if the client re-sent the record upon receiving the error. Allowing retries will potentially change the ordering of records because if two batches are sent to the same partition, and the first fails and is retried but the second succeeds, then the records in the second batch may appear first. In intermittent network disconnection scenarios, you are advised to configure the retry capability for producers as follows: Client 2.1.0 and later: You are advised to retain the default values of retries and delivery.timeout.ms, and set retry.backoff.ms to 1000. Earlier versions: You are advised to set retries to 10 and retry.backoff.ms to 1000.

Paramet er	Default Value	Recommended Value	Description
request.ti meout.m s	30,000		Maximum amount of time (in ms) the client will wait for the response of a request. If the response is not received before the timeout elapses, the client will throw a timeout exception. Setting this parameter to a large value, for example, 127000 (127s), can prevent records from failing to be sent in high-concurrency scenarios.
block.on. buffer.ful l	TRUE	TRUE	Setting this parameter to TRUE indicates that when buffer memory is exhausted, the producer must stop receiving new message records or throw an exception.
			By default, this parameter is set to TRUE . However, in some cases, non-blocking usage is desired and it is better to throw an exception immediately. Setting this parameter to FALSE will cause the producer to instead throw "BufferExhaustedException" when buffer memory is exhausted.

Paramet er	Default Value	Recommended Value	Description
batch.siz e	16,384	262,144	Default maximum number of bytes of messages that can be processed at a time. The producer will attempt to batch records together into fewer requests whenever multiple records are being sent to the same partition. This helps improve performance of both the client and the server. No attempt will be made to batch records larger than this size.
			Requests sent to brokers will contain multiple batches, one for each partition with data available to be sent.
			A smaller batch size will make batching less common and may reduce throughput (a batch size of zero will disable batching entirely). A larger batch size may use more memory as a buffer of the specified batch size will always be allocated in anticipation of additional records.

Paramet er	Default Value	Recommended Value	Description
buffer.m emory	33,554,432	67,108,864	Total bytes of memory the producer can use to buffer records waiting to be sent to the server. If records are sent faster than they can be delivered to the broker, the producer will stop sending records or throw a "block.on.buffer.full" exception. This setting should correspond roughly to the
			total memory the producer will use, but is not a rigid bound since not all memory the producer uses is used for buffering. Some additional memory will be used for compression (if compression is enabled) as well as for maintaining in-flight requests.
enable.id empoten ce	 Earlier than v3.0: false v3.0 and later: true 	If idempotence is not required, you are advised to set this parameter to false.	If you have enabled idempotence on the producer client, and produced messages, message offsets are not continuous on the consumer client or on the Instance > Message Query page on the Kafka console. This is because enabling idempotence generates some metadata control messages during message production. These control messages are produced to topics, and are invisible to consumers.

Table 5-26 Consumer parameters

Paramet er	Default Value	Recommended Value	Description
auto.com mit.enab le	TRUE	FALSE	If this parameter is set to TRUE, the offset of messages already fetched by the consumer will be periodically committed to ZooKeeper. This committed offset will be used when the process fails as the position from which the new consumer will begin.
			Constraints: If this parameter is set to FALSE , to avoid message loss, an offset must be committed to ZooKeeper after the messages are successfully consumed.
auto.offs et.reset	latest	earliest	Indicates what to do when there is no initial offset in ZooKeeper or if the current offset has been deleted. Options:
			earliest: Automatically reset to the smallest offset.
			latest: Automatically reset to the largest offset.
			none: The system throws an exception to the consumer if no offset is available.
			anything else: The system throws an exception to the consumer.
			If this parameter is set to latest, the producer may start to send messages to new partitions (if any) before the consumer resets to the initial offset. As a result, some messages will be lost.

Paramet er	Default Value	Recommended Value	Description
connecti ons.max.i dle.ms	600,000	30,000	Timeout interval (in ms) for an idle connection. The server closes the idle connection after this period of time ends. Setting this parameter to 30000 can reduce the server response failures when the network condition is poor.
max.poll. records	500	Must be less than the value of max.poll.interva l.ms.	The maximum number of messages that a consumer can pull from a broker at a time.
max.poll. interval. ms	300,000	Increase this parameter if complex and time-consuming logic exists between two polls.	The maximum interval between consumer polls, in milliseconds. If this parameter is exceeded, the consumption fails and the consumer is removed from the consumer group, triggering rebalance.
heartbea t.interval. ms	3,000	≥ 3000	Heartbeat interval between a consumer and Kafka, in milliseconds.
session.ti meout.m s	10,000	Set this parameter to at least 3 times the value of heartbeat.interval.ms.	The consumer-broker session timeout when the offset is managed by consumer group, in milliseconds.
fetch.ma x.bytes	1,000,000	max.request.size < message.max.byt es < fetch.max.bytes	The maximum bytes of a message that a consumer can pull from a broker at a time.

5.3.2 Suggestions on Using the Kafka Client

Consumers

1. Ensure that the owner thread does not exit abnormally. Otherwise, the client may fail to initiate consumption requests and the consumption will be blocked.

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

 # Set the authorization exception retry interval to 10 seconds.

 spring.kafka.listener.authorizationExceptionRetryInterval=10000

Producers

- 1. Synchronous replication: Set acks to all.
- 2. Retry message sending: Set retries to 10.
- 3. Message sending retry interval: Set **retry.backoff.ms** to **1000**.
- 4. Optimize message sending: For latency-sensitive messages, set **linger.ms** to **0**. For latency-insensitive messages, set **linger.ms** to a value ranging from **100** to **1000**.
- 5. Ensure that the producer has sufficient JVM memory to avoid blockages.
- 6. Set the timestamp to the local time. Messages will fail to age if the timestamp is a future time.
- 7. Try reusing producers. Do not create producers frequently. When idempotence is enabled (default for producer clients 3.0 and later), producing messages

- creates producer state objects on the server. Frequent creation results in too many objects to be reclaimed in time, causing server memory surges and performance deterioration. Set **enable.idempotence** to **false** if the idempotence is not required.
- 8. Catch exception **AuthorizationException**. You are advised to retry message production on the service side. Self-healing can be implemented through limited retry and backoff policies.

Topics

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

You can enable or disable automatic topic creation. Enabling this function automatically creates a topic when a message is produced in or consumed from a topic that does not exist.

Others

Maximum number of connections: 3000

Maximum size of a message: 10 MB

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

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

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

5.4 Connecting to Kafka Using the Client (Plaintext Access)

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

Notes and Constraints

For instances purchased in July 2020 and later, each Kafka broker allows a maximum of 1000 connections from each IP address by default. For instances purchased before July 2020, each Kafka broker allows a maximum of 200 connections from each IP address by default. Excess connections will be rejected. You can change the limit by referring to Modifying Kafka Instance Configuration Parameters, that is, to modify parameter max.connections.per.ip.

Prerequisites

- The network between the client and the Kafka instance is available. For details about the network requirements, see Kafka Network Connection Conditions.
- Security group rules have been properly configured.
 Before accessing a Kafka instance with SASL disabled on a client, configure proper security group rules for the instance. For details, see Table 5-2.
- The Kafka instance address has been obtained.
 Obtain the instance connection address in the Connection area on the Overview page on the Kafka console.
 - For private access within a VPC, the Kafka connection addresses are shown as follows.

Figure 5-24 Kafka instance addresses for private access within a VPC (**Instance Address (Private Network)**)

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

Figure 5-25 Kafka instance addresses for private access within a VPC (**Address (Private Network, Plaintext)**)



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

Figure 5-26 Kafka instance addresses for public access (**Instance Address** (**Public Network**))

Instance Address (Public Network) 139 45:9094,122 50:9094,119. 29:9094

Figure 5-27 Kafka instance addresses for public access (**Address (Public Network, Plaintext)**)



- If automatic topic creation is not enabled for the Kafka instance, **create a topic** before connecting to the instance.
- Compatible Kafka CLI is available. Ensure that the Kafka instance and the CLI use the same version.
 - Kafka CLI 1.1.0
 - Kafka CLI 2.7.2
 - Kafka CLI 3.4.0
- JDK v1.8.111 or later has been installed on the server, and the JAVA_HOME and PATH environment variables have been configured as follows:

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

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

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

Accessing the Instance Using CLI

The following uses Linux as an example.

Step 1 Decompress the Kafka CLI package.

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

tar -zxf {kafka_tar}

In the preceding command, {kafka_tar} indicates the name of the CLI package.

For example:

tar -zxf kafka_2.12-2.7.2.tgz

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

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

cd {kafka_tar}/bin

Step 3 Run the following command to produce messages:

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

Table 5-27 Message production parameters

Parameter	Description
Connection Address	Connection address of the Kafka instance. Obtained in Prerequisites .
Topic Name	The name of the topic created for the Kafka instance. If automatic topic creation is enabled for the Kafka instance, set this parameter to the name of a created topic or a topic that has not been created.

The following example uses connection addresses

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

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

To stop producing messages, press Ctrl+C to exit.

Step 4 Run the following command to consume messages:

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

Table 5 20 Message consumption parameters			
Parameter	Description		
Connection Address	Connection address of the Kafka instance. Obtained in Prerequisites .		
Topic Name	The name of the topic created for the Kafka instance.		
Consumer Group Name	The consumer group name set based on your service requirements. If a consumer group name starts with a special character, for example, a number sign (#), monitoring data cannot be displayed.		

Table 5-28 Message consumption parameters

Sample message consumption:

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

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

----End

Related Documents

- Troubleshooting Kafka Connection Exceptions
- Can I Change the Private Network Addresses of a Kafka Instance?

5.5 Connecting to Kafka Using the Client (Ciphertext Access)

This section describes how to access a Kafka instance in ciphertext on an open-source Kafka client. The client connects to the Kafka instance with SASL authentication. If the security protocol **SASL_SSL** is used, the client communicates with the Kafka instance in encryption, improving security.

Notes and Constraints

- For security purposes, TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256, TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256, and TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 are supported for instances created on and before March 20, 2021. TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 is also supported for instances created after March 20, 2021.
- For instances purchased in July 2020 and later, each Kafka broker allows a maximum of 1000 connections from each IP address by default. For instances purchased before July 2020, each Kafka broker allows a maximum of 200 connections from each IP address by default. Excess connections will be

rejected. You can change the limit by referring to **Modifying Kafka Instance Configuration Parameters**, that is, to modify parameter **max.connections.per.ip**.

Prerequisites

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

Before accessing a Kafka instance with ciphertext access enabled on a client, configure proper security group rules for the instance. For details, see **Table 5-2**.

• The Kafka instance address has been obtained.

Obtain the instance connection address in the **Connection** area on the **Overview** page on the Kafka console.

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

Figure 5-28 Kafka instance addresses for private access within a VPC (**Instance Address (Private Network)**)

Figure 5-29 Kafka instance addresses for private access within a VPC (**Address (Private Network, Plaintext)**)



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

Figure 5-30 Kafka instance addresses for public access (**Instance Address** (**Public Network**))



Figure 5-31 Kafka instance addresses for public access (**Address (Public Network, Ciphertext)**)



• The SASL mechanism in use is known.

In the **Connection** area on the Kafka instance details page, view **SASL Mechanism**. For instances that were created much earlier, if **SASL Mechanism** is not displayed on the instance details page, PLAIN is used by default.

Figure 5-32 SASL mechanism in use

SASL Mechanism

SCRAM-SHA-512

• The security protocol in use is known.

In the **Connection** area on the Kafka instance details page, view **Security Protocol**. For instances that were created much earlier, if **Security Protocol** is not displayed on the instance details page, SASL_SSL is used by default.

- If automatic topic creation is not enabled for the Kafka instance, **create a topic** before connecting to the instance.
- The client.jks certificate has been downloaded. Click the Kafka instance to go
 to the Basic Information tab page. Click Download next to SSL Certificate in
 the Connection area. Download and decompress the package to obtain the
 client certificate file client.jks. If you have already replaced the Kafka
 certificate with your own SSL certificate, prepare it in advance.
- Compatible Kafka CLI is available. Ensure that the Kafka instance and the CLI use the same version.
 - Kafka CLI 1.1.0
 - Kafka CLI 2.7.2
 - Kafka CLI 3.4.0
- JDK v1.8.111 or later has been installed on the server, and the JAVA_HOME and PATH environment variables have been configured as follows:

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

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

Run the **source** .bash **profile** command for the modification to take effect.

Accessing the Instance Using CLI

The following uses Linux as an example.

- **Step 1** Map hosts to IP addresses in the /etc/hosts file on the host where the client is located, so that the client can quickly parse the instance brokers.
 - Run the following command to edit the /etc/hosts file: vim /etc/hosts
 - 2. Press i and add the mapping between the host and IP address to the /etc/ hosts file.

```
ip 1 host name 1
ip 2 host name 2
ip 3 host name 3
```

Table 5-29 Parameters in the /etc/hosts file

Parameter	Description
ip	Connection address of the Kafka instance, which is obtained from Prerequisites .

Parameter	Description
host name	Name of each instance host. Host names are user-defined and must be unique.

For example:

10.154.48.120 server01 10.154.48.121 server02 10.154.48.122 server03

 Press Esc. Enter the following line and press Enter. Save the /etc/hosts file and exit.

Step 2 Decompress the Kafka CLI package.

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

tar -zxf {kafka_tar}

In the preceding command, {kafka_tar} indicates the name of the CLI package.

For example:

tar -zxf kafka_2.12-2.7.2.tgz

Step 3 Modify the configuration files of the Kafka CLI.

Configuration files: producer.properties and consumer.properties

Modify the producer.properties file.

- Go to the /config directory of the Kafka CLI. cd {kafka_tar}/config
- 2. Edit the **producer.properties** file. vim producer.properties
- 3. Press i and add the following content to the producer.properties file:
 - a. Kafka instances support two authentication mechanisms. Add the one obtained in SASL authentication mechanism to the configuration file. If both SCRAM-SHA-512 and PLAIN are enabled, use either of them in connection configurations.

PLAIN:

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

Table 5-30 SASL authentication mechanism parameters

Parameter	Description
username	Set in instance creation or user creation.
password	Set in instance creation or user creation.

SCRAM-SHA-512:

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

Table 5-31 SASL authentication mechanism parameters

Parameter	Description
username	Set in instance creation or user creation.
password	Set in instance creation or user creation.

- b. Kafka instances support two security protocols. Add the one obtained in the security protocol to the configuration file.
 - SASL_SSL:

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

Table 5-32 SASL_SSL parameters

Parameter	Description
ssl.truststore.location	Path for storing the client.jks certificate
	Even in Windows, you need to use slashes (/) for the certificate path. Do not use backslashes (\), which are used by default for paths in Windows. Otherwise, the client will fail to obtain the certificate.
ssl.truststore.password	Password of the Kafka client certificate.
	dms@kafka by default if the SSL certificate is provided from the Kafka console and cannot be changed. Configure it as required if your own client certificate is used.
ssl.endpoint.identificatio n.algorithm	Whether to verify the certificate domain name This parameter must be left blank, which indicates disabling domain name verification.

SASL_PLAINTEXT:

security.protocol=SASL_PLAINTEXT

4. Press **Esc**. Enter the following line and press **Enter**. Save the **producer.properties** file and exit.

:wq

Modify the consumer.properties file.

- Edit the consumer.properties file. vim consumer.properties
- 2. Press i and add the following content to the consumer.properties file:
 - Kafka instances support two authentication mechanisms. Add the one
 obtained in SASL authentication mechanism to the configuration file.
 If both SCRAM-SHA-512 and PLAIN are enabled, use either of them in
 connection configurations.

PLAIN:

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

Table 5-33 SASL authentication mechanism parameters

Parameter	Description
username	Set in instance creation or user creation.
password	Set in instance creation or user creation.

SCRAM-SHA-512:

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

Table 5-34 SASL authentication mechanism parameters

Parameter	Description
username	Set in instance creation or user creation.
password	Set in instance creation or user creation.

b. Kafka instances support two security protocols. Add the one obtained in the security protocol to the configuration file.

SASL_SSL:

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

Parameter	Description
ssl.truststore.location	Path for storing the client.jks certificate
	Even in Windows, you need to use slashes (/) for the certificate path. Do not use backslashes (\), which are used by default for paths in Windows. Otherwise, the client will fail to obtain the certificate.
ssl.truststore.password	Password of the Kafka client certificate.
	dms@kafka by default if the SSL certificate is provided from the Kafka console and cannot be changed. Configure it as required if your own client certificate is used.
ssl.endpoint.identificatio n.algorithm	Whether to verify the certificate domain name This parameter must be left blank, which indicates disabling domain name verification.

Table 5-35 SASL_SSL parameters

SASL_PLAINTEXT:

security.protocol=SASL_PLAINTEXT

3. Press **Esc**. Enter the following line and press **Enter**. Save the **consumer.properties** file and exit.

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

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

cd ../bin

Step 5 Run the following command to create messages:

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

Table 5-36 Message production parameters

Parameter	Description
Connection Address	Connection address of the Kafka instance. Obtained in Prerequisites .
Topic Name	The name of the topic created for the Kafka instance. If automatic topic creation is enabled for the Kafka instance, set this parameter to the name of a created topic or a topic that has not been created.

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

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

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

To stop producing 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

Table 5-37 Message consumption parameters

Parameter	Description
Connection Address	Connection address of the Kafka instance.
	Obtained in Prerequisites.
Topic Name	The name of the topic created for the Kafka instance.
Consumer Group Name	The consumer group name set based on your service requirements. If a consumer group name starts with a special character, for example, a number sign (#), monitoring data cannot be displayed.
	WARNING Ensure that the consumer group name in the command line is the same as that specified in the configuration file (consumer.properties), if any. Otherwise, message consumption may fail.
	To view and change the consumer group name in the configuration file, see Step 6.1 .

To view and change the consumer group name in the configuration file:

- Access the /config directory of the Kafka CLI. cd ../config
- Edit the consumer.properties file. vim consumer.properties
- 3. Press i to view and change the consumer group name.

group.id indicates the consumer group name. You can change the name as required.

consumer group id group.id=*test-consumer-group*

4. Press **Esc**. Enter the following line and press **Enter**. Save the **consumer.properties** file and exit.

:wc

Sample message consumption:

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

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

----End

Related Documents

- Troubleshooting Kafka Connection Exceptions
- Can I Change the Private Network Addresses of a Kafka Instance?
- Is the Same SSL Certificate Used for Different Instances?

5.6 Connecting to Kafka on the Console

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

Prerequisites

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

Producing Messages on the Console

- **Step 1** Log in to the **Kafka console**.
- **Step 2** Click in the upper left corner to select the region where your instance is located.
- **Step 3** Click the desired instance to go to the instance details page.
- **Step 4** In the navigation pane, choose **Instance** > **Topics**.
- **Step 5** Go to the message production page in either of the following ways:
 - In the row containing the desired topic, choose More > Create Message (if SASL is enabled) or click Create Message (if SASL is disabled). The Create Message dialog box is displayed.
 - Click the desired topic to go to the topic details page. Click **Create Message** in the upper right corner. The **Create Message** dialog box is displayed.
- **Step 6** Set message parameters by referring to **Table 5-38**.

Table 5-38 Message parameters

Parameter	Description
Message Body	Message content. 0 to 128,000 bytes.
Message Key	Message key.
Specify Partition	Indicates whether to enable the function of sending messages to a specified partition.
	Off: Messages are sent to partitions based on their key hash.
	On: Messages are sent to specified partitions. Requires the partition ID.

Step 7 Click OK.

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

----End

Related Document

To send specific messages to a Kafka instance on the console by calling an API, see **Producing Messages to Kafka**.

6 Managing Messages

6.1 Viewing Kafka Messages

When messages are lost or fail to be consumed, you can query the content and attributes of specific messages for troubleshooting.

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

Notes and Constraints

- 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.
- To query with content or key, due to resource and performance restrictions, a total of 200 MB and 10,000 messages can be queried, and a maximum of 10 messages can be returned.
- To query messages in all partitions, a maximum of 500 records can be returned.
- The console displays messages smaller than 4 KB. To view messages larger than 4 KB, click **Download Message**.

Procedure

- **Step 1** Log in to the **Kafka console**.
- **Step 2** Click in the upper left corner to select the region where your instance is located.
- **Step 3** Click the desired instance to go to the instance details page.
- **Step 4** Go to the message query page in either of the following ways:
 - View messages in any topic: In the left navigation pane, choose Instance > Message Query.
 - View messages in a specific topic: In the navigation pane, choose Instance > Topics. Click the name of the topic to be queried. The topic details page is displayed. Click Message Query. The Message Query tab page is displayed.

Step 5 Messages can be queried by offset or creation time.

• When the topic partition and offset of the message are known, query the message by offset. For details, see **Table 6-1**.

Table 6-1 Querying messages by offset

Parameter	Description
Topic Name	Displayed only when you choose Instance > Message Query.
	Name of the topic to be queried.
Partition	Partition where the message is located.
Offset	Offset of the message.

 When the time range of sending the message is known but the message offset is unknown, query the message by creation time. For details, see Table 6-2.

Table 6-2 Querying messages by creation time

Parameter	Description
Topic Name	Displayed only when you choose Instance > Message Query.
	Name of the topic to be queried.
Partition	Partition where the message is located.
	If no partition is specified, messages in all partitions of the topic are displayed in the query result.
	Constraints on partition -based message query:
	 Query in all partitions: Due to resource and performance limitations, a maximum of 500 messages can be returned, and the total size of all messages cannot exceed 200 MB. Process of obtaining 500 messages: The system pulls the latest 500 messages from each partition in sequence (starting from partition 0) until the total size of pulled messages exceeds 200 MB. The queried messages are sorted by time, and the latest 500 messages are returned.
	 Query in a single partition: All messages in the queried time period are returned, without the limit of 500 messages or 200 MB.
	To query more messages, shorten the query time range or specify partitions.

Parameter	Description		
Key	Enter a message key to search for messages containing it.		
	For example, a topic contains two messages whose keys are abc and abcd . Enter "abc" in the Key box. The two messages are returned.		
	Query with a key: Due to resource and performance limitations, a maximum of 10,000 messages can be searched, and the total size of all messages cannot exceed 200 MB. Only 10 messages containing the key are returned.		
Content	Multiple query conditions can be set. Query results meet all of them.		
	To set a query condition:		
	1. In the Content box, left-click and choose √ or × .		
	2. Enter a keyword in the message body and press Enter .		
	3. To set multiple conditions, repeat Step 5.a to Step 5.b .		
	Query with content: Due to resource and performance limitations, a maximum of 10,000 messages can be searched, and the total size of all messages cannot exceed 200 MB. Only 10 messages containing the keyword are returned.		
	For large records (> 20 KB per message) or a long period, dump messages for offline query.		
Created	Time when a message is created. You can set the time range of messages to be queried. WARNING If a topic contains a large amount of data, an internal service error may be reported when you query messages in a topic with only one replica. You can shorten the time range for query based on the data volume.		

Step 6 Click **Search** to query messages.

The query result is as follows.

Figure 6-1 Querying topic messages

Topic Name ⊜	Partition 😝	Offset 😂	Message Size (Bytes) ⊕ Created ⊕ Operation
topic-1081992957	0	1	8 May 20, 2024 14:12:08 GMT+08:00 View Message Body
topic-1081992957	0	0	9 May 20, 2024 14:11:57 GMT+08:00 View Message Body

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

Table 6-3 Message parameters

Step 7 Click **View Message Body**. In the displayed **View Message Body** dialog box, view the message content, including the topic name, partition, offset, creation time, and message body.

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

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

----End

Related Documents

- To view Kafka messages by calling an API, see Querying Messages.
- Why Is the Message Creation Time Displayed as Year 1970?
- Why Can't I Query Messages on the Console?

6.2 Changing Kafka Message Retention Period

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

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

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

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

The **log.retention.hours** parameter takes effect only when the aging time has not been set for the topic. For example, if the aging time of Topic01 is set to 60 hours and **log.retention.hours** is set to 72 hours, the actual aging time of Topic01 is 60 hours.

Notes and Constraints

- The retention period of single-node instances can be modified only on the **Instance** > **Topics** page.
- The retention period of a maximum of 50 topics can be modified at a time.

Changing the Kafka Message Retention Period

The retention period of single or multiple topics can be modified on the Kafka console.

Modifying the Message Retention Period of a Topic

- Step 1 Log in to the Kafka console.
- **Step 2** Click on the upper left corner to select the region where your instance is located.
- **Step 3** Click the desired instance to go to the instance details page.
- **Step 4** In the navigation pane, choose **Instance** > **Topics**.
- **Step 5** In the row containing the desired topic, click **Edit**.
- **Step 6** In the **Edit Topic** dialog box, enter the aging time (1–720) and click **OK**. View the aging time on the **Topics** page.

----End

Modifying the Message Retention Period of Multiple Topics

- **Step 1** Log in to the **Kafka console**.
- **Step 2** Click in the upper left corner to select the region where your instance is located.
- **Step 3** Click the desired instance to go to the instance details page.
- **Step 4** In the navigation pane, choose **Instance** > **Topics**.
- **Step 5** Select the desired topics and click **Batch Edit Topic** above the list.
- **Step 6** In the **Batch Operations** area, select **Aging Time (h)** and enter a value (range: 1 to 720). In the **Preview Change** area, check the aging time before and after the modification and click **OK**.

10 ∨ ⟨ 1 >

Figure 6-2 Batch modifying the aging time

Batch Edit Topic Selected settings will be synced to all selected topics. **Batch Operations** Partitions _ Aging Time (h) Synchronous Replication Synchronous Flushing Message Timestamp Max. Message Size (bytes) Preview Change Topic Name Aging Time (h)Before/After 72 topic-doc01 81 72 topic-doc02 73 Unchanged topic-doc03 72

Check the new aging time on the **Topics** page.

----End

Total Records: 3

Related Documents

- To modify the message retention period by calling an API, see **Modifying Topics of a Kafka Instance**.
- Why Do Messages Still Exist After the Retention Period Elapses?

6.3 Deleting Kafka Messages

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

Notes and Constraints

Deleting messages takes effect permanently.

Prerequisite

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

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

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

Procedure

- **Step 1** Log in to the **Kafka console**.
- **Step 2** Click on the upper left corner to select the region where your instance is located.
- **Step 3** Click the desired instance to go to the instance details page.
- **Step 4** In the navigation pane, choose **Instance** > **Topics**.
- **Step 5** Go to the message deletion page in either of the following ways:
 - In the row containing the topic whose messages are to be deleted, choose
 More > Delete Messages. The Delete Messages dialog box is displayed.
 - Click the name of the topic whose messages are to be deleted. The topic details page is displayed. Click **Delete Messages** in the upper right corner. The **Delete Messages** dialog box is displayed.
- **Step 6** Set the parameters for deleting messages, as shown in **Table 6-4**.

Figure 6-3 Deleting messages

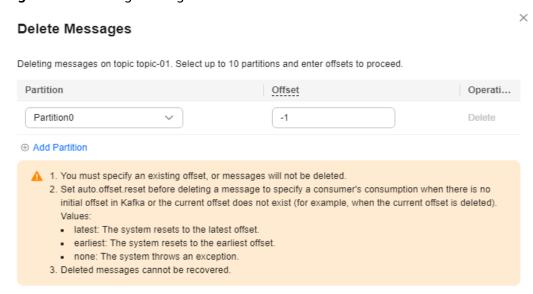


Table 6-4 Parameters for deleting a message

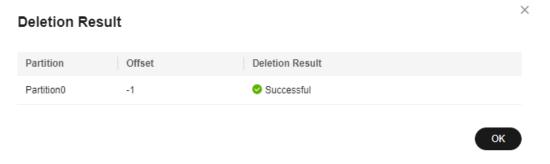
Parameter	Description
Partition	Select the ID of the partition where the message is located.

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

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

Step 7 Click **OK**. The **Deletion Result** dialog box is displayed. Click **OK** to delete the messages.

Figure 6-4 Deletion result



Step 8 Verify whether the message is deleted.

To query with offset on the **Instance** > **Message Query** page, enter the following parameters. If no message is found, it is deleted successfully.

- Partition: Enter the partition No. from **Step 6**.
- Offset: Enter a random offset if the offset from **Step 6** is **-1**. Otherwise, enter an integer smaller than the offset.

----End

Related Documents

To delete a message by calling an API, see **Deleting a Kafka Message**.

6.4 Diagnosing Kafka Message Accumulation

Unprocessed messages accumulate if the client's consumption is slower than the server's sending. Accumulated messages cannot be consumed in time.

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

Prerequisites

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

Process Flow

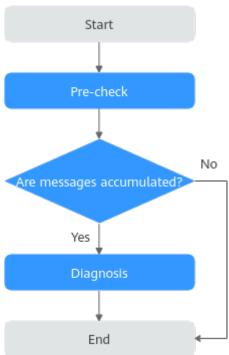


Figure 6-5 Process of accumulation diagnosis

Step 1: Pre-check

- **Step 1** Log in to the **Kafka console**.
- **Step 2** Click in the upper left corner to select the region where your instance is located.
- **Step 3** Click the desired instance to go to the instance details page.
- **Step 4** In the left navigation pane, choose **Analysis & Diagnosis** > **Accumulation Diagnosis**.
- **Step 5** Select the consumer groups and topics to be diagnosed, and click **Pre-check**.

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

Figure 6-6 Pre-check



When the total number of stacked messages in all partitions of a topic is less than 2,000, **Accumulated Messages** displays no risk. In this case, message stack diagnosis is unavailable. If there are any risks in the **Accumulated Messages** area and the consumer group is not consuming message in the assign mode, you can perform **message accumulation diagnosis**.

----End

Step 2: Diagnosis

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

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

- **Step 2** Locate the row that contains the target diagnosis record, and click **View Details**. The **Diagnosis Details page** is displayed.
- **Step 3** View the number of abnormal, failed, and normal items in the upper part of the page. In the **Diagnosed Item** area, click an abnormal item, such as **Rebalancing**, and view the possible causes, affected partitions or brokers, and handling suggestions.

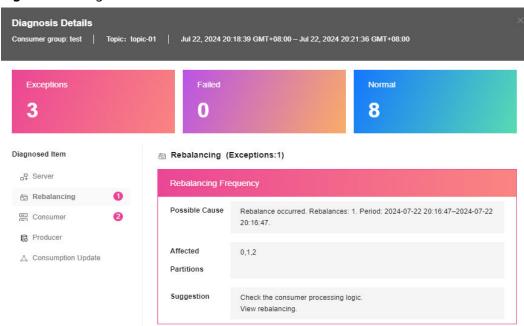


Figure 6-7 Diagnosis details

----End

Related Document

To view a diagnosis report by calling an API, see **Querying Diagnosis Report Details**.

Managing Consumer Groups

7.1 Creating a Kafka Consumer Group

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

Consumer group Group 01 Topic Consumer 0 Topic 01 Consumer 1 Partition 0 Partition 1 Group 02 Consumer 0 Topic 02 Partition 0 Consumer 1 Partition 1 Consumer 2 Consumer 3

Figure 7-1 Example consumption

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

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

This section describes how to create a consumer group on the console. This operation does not restart the Kafka instance.

Notes and Constraints

- For instances created on and after April 25, 2023, consumer groups can be created on the console.
- If **auto.create.groups.enable** is set to **true**, the consumer group status is **EMPTY**, and no offset has been submitted, the system automatically deletes the consumer group 10 minutes later.
- If **auto.create.groups.enable** is set to **false**, the system does not automatically delete consumer groups. You can manually delete them.
- If a consumer group has never committed an offset, the group will be deleted after the Kafka instance restarts.

Procedure

- **Step 1** Log in to the **Kafka console**.
- **Step 2** Click on the upper left corner to select the region where your instance is located.
- **Step 3** Click the desired instance to go to the instance details page.
- **Step 4** In the navigation pane, choose **Instance** > **Consumer Groups**.
- **Step 5** Click **Create Consumer Group**.
- **Step 6** Set consumer group parameters by referring to Table 7-1 and click **OK**.

Table 7-1 Consumer group parameters

Parameter	Description	
Consumer Group Name	Enter 3 to 64 characters, starting with a letter or underscore (_). Use only letters, digits, periods (.), hyphens (-), and underscores (_).	
	If a consumer group name starts with a special character, for example, a number sign (#), monitoring data cannot be displayed.	
Description	Enter 0 to 200 characters.	

View the new consumer group in the consumer group list.

----End

Related Documents

- To create a consumer group by calling an API, see Creating a Consumer Group.
- Why Can't I View Consumers When Instance Consumption Is Normal?

7.2 Querying the Kafka Consumer Group List

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

Viewing the Consumer Group List

A consumer group list can be viewed on the Kafka console or on a client.

Viewing the Consumer Group List (Console)

- **Step 1** Log in to the **Kafka console**.
- **Step 2** Click in the upper left corner to select the region where your instance is located.
- **Step 3** Click the desired instance to go to the instance details page.
- **Step 4** In the navigation pane, choose **Instance** > **Consumer Groups**.

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

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

Figure 7-2 Consumer group list



- **Step 5** (Optional) To query a consumer group, enter a consumer group name or status, Coordinator (ID), number of accumulated messages, description, or keyword, then press **Enter**.
- **Step 6** (Optional) To refresh the consumer group list, click \bigcirc in the upper right corner.

----End

Viewing the Consumer Group List (Kafka CLI)

 For a Kafka instance with ciphertext access disabled, run the following command in the /bin directory of the Kafka client: ./kafka-consumer-groups.sh --bootstrap-server {connection-address} --list Parameter description: **connection-address** indicates the Kafka instance address, which can be obtained in the **Connection** area on the **Overview** page on the Kafka console.

Example:

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

- For a Kafka instance with ciphertext access enabled, do as follows:
 - a. (Optional) If the username and password, and the SSL certificate has been configured, skip this step and go to b. Otherwise, do as follows:
 Create the ssl-user-config.properties file in the /config directory of the Kafka client. Add the username and password, and the SSL certificate configuration by referring to Step 3.
 - b. Run the following command in the **/bin** directory of the Kafka client: ./kafka-consumer-groups.sh --bootstrap-server {connection-address} --list --command-config ../ config/{ssl-user-config.properties}

Table 7-2 Consumer group list query parameters

Parameter	Description	
connection-address	Connection address of a Kafka instance. To obtain the address, choose Overview > Connection .	
ssl-user- config.properties	Configuration file name. This file contains username, password, and SSL certificate information.	

Example:

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

Related Document

To view a consumer group list by calling an API, see **Querying All Consumer Groups**.

7.3 Viewing Kafka Consumer Information

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

Notes and Constraints

- Due to cache reasons, the consumer connection addresses displayed on Kafka Manager may have expired. In this case, restart Kafka Manager.
- Instances created on or after May 17, 2023 do not have Kafka Manager. You cannot view consumer addresses of these instances using Kafka Manager.

Prerequisites

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

Viewing a Consumer List

A consumer list can be viewed on the Kafka console or on a client.

Viewing the Consumer List (Console)

- **Step 1** Log in to the **Kafka console**.
- **Step 2** Click in the upper left corner to select the region where your instance is located.
- **Step 3** Click the desired instance to go to the instance details page.
- **Step 4** In the navigation pane, choose **Instance** > **Consumer Groups**.
- **Step 5** Click the name of the desired consumer group.
- **Step 6** On the **Consumers** tab page, view the consumer list.
 - In the consumer list, you can view the consumer ID, consumer address, and client ID.
- **Step 7** (Optional) To query a specific consumer, enter the consumer ID in the search box and press **Enter**.

----End

Viewing the Consumer List (Kafka CLI)

• For a Kafka instance with ciphertext access disabled, run the following command in the **/bin** directory of the Kafka client:

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

Table 7-3 Consumer list query parameters

Parameter	Description	
connection-address	Connection address of a Kafka instance. To obtain the address, choose Overview > Connection .	
group-name	Consumer group name.	

Example:

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

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

- For a Kafka instance with ciphertext access enabled, do as follows:
 - a. (Optional) If the username and password, and the SSL certificate has been configured, skip this step and go to b. Otherwise, do as follows:
 Create the ssl-user-config.properties file in the /config directory of the Kafka client. Add the username and password, and the SSL certificate configuration by referring to Step 3.
 - b. Run the following command in the /bin directory of the Kafka client: ./kafka-consumer-groups.sh --bootstrap-server {connection-address} --group {group-name} --members --describe --command-config ../config/{ssl-user-config.properties}

Table 7-4 Consumer list query parameters

Parameter	Description
connection-address	Connection address of a Kafka instance. To obtain the address, choose Overview > Connection .
group-name	Consumer group name.
ssl-user- config.properties	Configuration file name. This file contains username, password, and SSL certificate information.

Example:

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

GROUP CONSUMER-ID HOST CLIENT-ID #PARTITIONS

test console-consumer-566d0c82-07d3-4d87-9a6e-f57a9bc9fc69 /192.168.0.215 console-consumer 3 [root@ecs-kafka bin]#

Viewing Consumer Connection Addresses

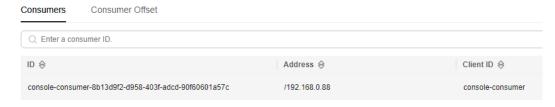
Consumer connection addresses can be viewed on the Kafka console or through Kafka Manager.

Viewing Consumer Connection Addresses (Console)

- Step 1 Log in to the Kafka console.
- **Step 2** Click in the upper left corner to select the region where your instance is located

- **Step 3** Click the desired instance to go to the instance details page.
- **Step 4** In the navigation pane, choose **Instance** > **Consumer Groups**.
- **Step 5** Click the desired consumer group.
- **Step 6** On the **Consumers** tab page, view the consumer addresses.

Figure 7-3 Consumer list

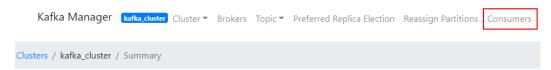


----End

Viewing Consumer Connection Addresses (Kafka Manager)

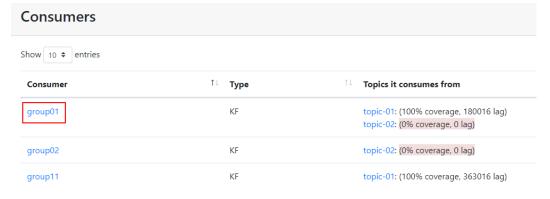
- Step 1 Log in to Kafka Manager.
- **Step 2** Click **kafka_cluster** to go to the cluster details page.
- **Step 3** On the top menu bar, choose **Consumers**.

Figure 7-4 Navigation bar



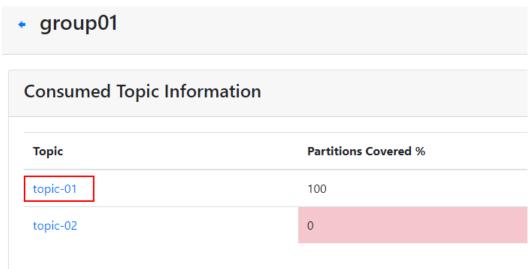
Step 4 Click the desired consumer group to view the topics that the group has subscribed to.

Figure 7-5 Consumer group list



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

Figure 7-6 Topics that the consumer group has subscribed to



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

Figure 7-7 Topic details page

Partition	LogSize	Consumer Offset	Lag	Consumer Instance Owner
0	33,333	0	33,333	consumer-1-5d096c5f-159d-468d-8b10-7961dc6f49d1 2 1&25412746
1	33,334	0	33,334	consumer-1-5d096c5f-159d-468d-8b10-7961dc6f49d1;/10_\$14.177.30
2	33,333	0	33,333	consumer-1-5d096c5f-159d-468d-8b10-7961dc6f49d1;/\$\dagger{4}\dagger{1} \dagger{1} \dagge

----End

Related Document

To view a consumer list and connection addresses by calling an API, see **Querying** a **Specified Consumer Group**.

7.4 Viewing and Resetting Kafka Consumption Offsets

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

Notes and Constraints

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

Prerequisites

The consumer offset cannot be reset on the fly. You must first stop consumption of the desired consumer group. After a client is stopped, the server considers the client offline only after the time period specified in

ConsumerConfig.SESSION_TIMEOUT_MS_CONFIG (1000 ms by default).

Viewing Consumer Offset

Consumer offset can be viewed on the Kafka console or on a client.

Viewing Consumer Offsets (Console)

- **Step 1** Log in to the **Kafka console**.
- **Step 2** Click [♥] in the upper left corner to select the region where your instance is located.
- **Step 3** Click the desired instance to go to the instance details page.
- **Step 4** In the navigation pane, choose **Instance** > **Consumer Groups**.
- **Step 5** Click the name of the desired consumer group.
- **Step 6** On the **Consumer Offset** tab page, view the list of topics that the consumer group has subscribed to, topic quantity, total number of messages accumulated in the topic, and offset of each partition.

Figure 7-8 Consumer offsets

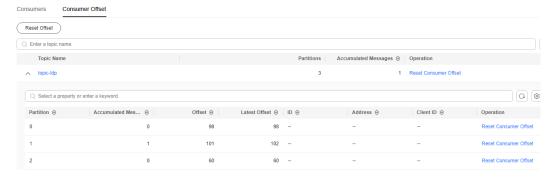


Table 7-5 Consumer offset parameters

Parameter	Description	
Topic Name	Name of a topic that the consumer group has subscribed to	
Partitions	Number of partitions in a topic	
Cumulative Messages	Number of messages that are not consumed by the consumer group in a topic	
	This parameter indicates the instantaneous value at the sampling. The value of its corresponding metric in monitoring is sampled every minute. These values may vary. For more information, see Why Is the Number of Stacked Messages Monitored as 0 when Messages Are Stacked?.	
Partition	Partition number in a topic	

Parameter	Description
Accumulated Messages	Number of messages that are not consumed by the consumer group in a partition
Offset	Offset of this partition
Latest Offset	Maximum message position of a partition
ID	ID of the consumer who consumes messages in this partition
Address	Address of the consumer who consumes messages in this partition
Client ID	Client identifier. This client is used to connect to a Kafka instance and consume messages in this partition.

- **Step 7** (Optional) To query the consumer offsets of a specific topic, enter the topic name in the search box and press **Enter**.
- **Step 8** (Optional) To export the consumption progress to the local, refer to either of the following ways.
 - Select the desired topics and choose Export > Export selected data to an XLSX file to export the consumption progress of specific topics.
 - Choose **Export** > **Export all data to an XLSX file** to export the consumption progress of all topics.

----End

Viewing Consumer Offsets (Kafka CLI)

 For a Kafka instance with ciphertext access disabled, run the following command in the /bin directory of the Kafka client:

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

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

Example:

[root@ecs-kafka bin]# ./kafka-consumer-groups.sh --bootstrap-server 192.168.xx.xx:9092,192.168.xx.xx:9092,192.168.xx.xx:9092 -- offsets -- describe -- all-groups Consumer group '_consumer-group-dial-test' has no active members. GROUP PARTITION CURRENT-OFFSET LOG-END-OFFSET LAG CLIENT-ID CONSUMER-ID HOST __consumer-group-dial-test __dms_dial_test 0 350 0 350 __consumer-group-dial-test __dms_dial_test 1 350 350 0 __consumer-group-dial-test __dms_dial_test 2 350 350 0 Consumer group 'test' has no active members. GROUP PARTITION CURRENT-OFFSET LOG-END-OFFSET LAG TOPIC CONSUMER-ID HOST

test	topic-01	0	5	5	0	-	-	-	
test	topic-01	1	3	3	0	-	-	-	
test	topic-01	2	10	10	0	-	-	-	
[root@e	cs-kafka bin]#								

- For a Kafka instance with ciphertext access enabled, do as follows:
 - a. (Optional) If the username and password, and the SSL certificate has been configured, skip this step and go to b. Otherwise, do as follows:
 Create the ssl-user-config.properties file in the /config directory of the Kafka client. Add the username and password, and the SSL certificate configuration by referring to Step 3.
 - b. Run the following command in the **/bin** directory of the Kafka client: ./kafka-consumer-groups.sh --bootstrap-server {connection-address} --offsets --describe --all-groups --command-config ../config/{ssl-user-config.properties}

Table 7-6 Consumer offset query parameters

Parameter	Description
connection-address	Connection address of a Kafka instance. To obtain the address, choose Overview > Connection .
ssl-user- config.properties	Configuration file name. This file contains username, password, and SSL certificate information.

Example:

```
[root@ecs-kafka bin]# ./kafka-consumer-groups.sh --bootstrap-server
192.168.xx.xx:9093,192.168.xx.xx:9093,192.168.xx.xx:9093 --offsets --describe --all-groups --
command-config ../config/ssl-user-config.properties
Consumer group '_consumer-group-dial-test' has no active members.
                                PARTITION CURRENT-OFFSET LOG-END-OFFSET
GROUP
                    TOPIC
LAG
           CONSUMER-ID
                           HOST
                                       CLIENT-ID
                                                          347
__consumer-group-dial-test __dms_dial_test 0
                                              347
__consumer-group-dial-test __dms_dial_test 1
                                               347
                                                          347
 _consumer-group-dial-test __dms_dial_test 2
                                               347
                                                          347
Consumer group 'test' has no active members.
                         PARTITION CURRENT-OFFSET LOG-END-OFFSET LAG
GROUP
            TOPIC
CONSUMER-ID HOST
                            CLIENT-ID
          topic-01 0
                              5
                                                  0
test
                                        3
test
          topic-01
                      1
                              3
                                                  0
          topic-01
                                        10
                                                   0
test
                      2
                              10
[root@ecs-kafka bin]#
```

Resetting Consumer Offsets

- **Step 1** Log in to the **Kafka console**.
- **Step 2** Click in the upper left corner to select the region where your instance is located.

- **Step 3** Click the desired instance to go to the instance details page.
- **Step 4** In the navigation pane, choose **Instance** > **Consumer Groups**.
- **Step 5** Click the name of the desired consumer group.
- **Step 6** On the **Consumer Offset** tab page, you can perform the following operations:
 - To reset the consumer offset of all partitions of a single topic, click **Reset Consumer Offset** in the row containing the desired topic.
 - To reset the consumer offset of a single partition of a single topic, click **Reset Consumer Offset** in the row containing the desired partition.
 - To reset the consumer offset in all partitions of all topics, click **Reset Offset**.
- **Step 7** In the displayed **Reset Consumer Offset** dialog box, set the parameters by referring to **Table 7-7**.

Table 7-7 Parameters for resetting the consumer offset

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

Step 8 Click OK.

Step 9 Click **Yes** in the confirmation dialog box. The consumer offset is reset.

On the **Consumer Offset** tab page, click before the topic whose consumer offset has been reset, and view the new value in the **Offset** column.

----End

Related Document

To reset consumer offset by calling an API, see **Resetting Consumer Group Offset** to the Specified Position.

7.5 Viewing Kafka Rebalancing Logs

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

Possible causes of rebalancing:

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

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

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

Notes and Constraints

- Rebalancing logging is not available for instances created before April 6, 2023.
- Unavailable for single-node instances.
- Rebalancing logs are stored for seven days by default. To retain them longer, see Modifying a Log Group.
- Enabling rebalancing logging will create a log group, log stream, and dashboard in LTS. Fees are generated based on the log volume. For details, see LTS pricing details.
- When Kafka instances store rebalancing logs in one log group and log stream, the log group and stream of each instance contain the rebalancing logs of all the instances.

Prerequisites

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

Enabling Rebalancing Logging

- **Step 1** Log in to the **Kafka console**.
- **Step 2** Click in the upper left corner to select the region where your instance is located.
- **Step 3** Click the desired instance to go to the instance details page.
- **Step 4** In the navigation pane, choose **Analysis & Diagnosis** > **Rebalancing Logs**.
- **Step 5** Click **Enable Logging**. The **Enable Logging** dialog box is displayed.
- **Step 6** Click **OK**. The **Configure Logs** dialog box is displayed.

- **Step 7** Determine whether to enable this function as required. Click **OK**. The **Background Tasks** page is displayed. The rebalancing log function is enabled when the **Enable logging** task is in the **Successful** state.
 - Disabled: LTS automatically creates a log group and a log stream.
 - Enabled: Select the log group and log stream that store the **coordinator.log** file. To view or create a log group and log stream, click **View Log Group** on the right to go to the LTS console.

----End

Viewing Rebalancing Logs

- Step 1 Log in to the Kafka console.
- **Step 2** Click in the upper left corner to select the region where your instance is located.
- **Step 3** Click the desired instance to go to the instance details page.
- **Step 4** In the navigation pane, choose **Analysis & Diagnosis** > **Rebalancing Logs**.
- **Step 5** On the **Dashboard** tab page, view the number of consumer group rebalancing times and reasons. On the **Logs** tab page, view rebalancing logs.

To search for logs, see Log Search.

An example rebalancing log:

```
"level":"INFO".
          "timestamp": "2023-03-23 17:23:22,906",
          "message":{
                    "leaderId": "consumer-1-177817b6-1f29-4717-8a83-dda8eaab1635",
                    "generationId":"1",
                   "reason": "Assignment received from leader for group KMOffsetCache-dms-vm-fa3cf9d6-manager-
shared-server-0 for generation 1",
                    "groupId":"KMOffsetCache-dms-vm-fa3cf9d6-manager-shared-server-0",
                    "coordinatorId":"0",
                    "type":"END REBALANCE"
                    "group":"GroupMetadata(groupId=KMOffsetCache-dms-vm-fa3cf9d6-manager-shared-server-0,
generation=1, protocolType=Some(consumer), currentState=CompletingRebalance,
members=Map(consumer-1-177817b6-1f29-4717-8a83-dda8eaab1635 ->
Member Meta data (member Id=consumer-1-177817b6-1f29-4717-8a83-dda {\bf \bar{8}} eaab 1635, client Id=consumer-1, {\bf \bar{1}} consumer-1, {\bf 
clientHost=/172.31.2.168, sessionTimeoutMs=10000, rebalanceTimeoutMs=300000,
supportedProtocols=List(range), )))"
```

Table 7-8 describes the parameters.

Table 7-8 Rebalancing parameters

Parameter	Description	
level	Level of the rebalancing logs. The only value is INFO .	
timestamp	Time of rebalancing.	
leaderId	Leader consumer ID.	

Parameter	Description	
generationId	Generation ID of the consumer group. Generation is the number of times that a consumer group performs rebalancing. It is incremented by 1 each time a rebalancing is complete.	
reason	Reason for triggering rebalancing.	
groupId	Consumer group ID.	
coordinatorId	Broker where the Coordinator component is.	
type	 Operation that triggered rebalancing. Values: JOIN_GROUP: A new consumer is added to a consumer group. OVER_CAPACITY: The group limit is exceeded. UPDATE_MEMBER: The consumer metadata is updated. PROTOCOL_CHANGE: The protocol is changed. HEARTBEAT_EXPIRED: The consumer heartbeat timed out. SYNC_GROUP: The reassignment plan is synchronized. END_REBALANCE: Rebalancing ended. LEAVE_GROUP: A consumer left a consumer group. DELETE_GROUP: A user deletes a consumer group. 	
group	Information about consumers in the consumer group.	

----End

Disabling Rebalancing Logging

- Step 1 Log in to the Kafka console.
- **Step 2** Click in the upper left corner to select the region where your instance is located.
- **Step 3** Click the desired instance to go to the instance details page.
- **Step 4** In the navigation pane, choose **Analysis & Diagnosis** > **Rebalancing Logs**.
- **Step 5** Click **Disable Logging** in the upper right corner. The **Disable Logging** dialog box is displayed.
- **Step 6** Click **OK**. The **Background Tasks** page is displayed. The rebalancing log function is disabled when the rebalancing logging task is in the **Successful** state.

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

----End

Related Document

Why Does Message Poll Often Fail During Rebalancing?

7.6 Modifying Kafka Consumer Group Description

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

Procedure

- **Step 1** Log in to the **Kafka console**.
- **Step 2** Click in the upper left corner to select the region where your instance is located.
- **Step 3** Click the desired instance to go to the instance details page.
- **Step 4** In the navigation pane, choose **Instance** > **Consumer Groups**.
- **Step 5** Go to the consumer group editing page in either of the following ways:
 - In the row containing the consumer group to be edited, click **Edit**.
 - Click the name of the consumer group whose information is to be edited. Click **Edit** in the upper right corner. The **Edit** dialog box is displayed.
- **Step 6** Modify the description and click **OK**.

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

----End

Related Document

To modify consumer group information by calling an API, see **Modifying a Specified Consumer Group**.

7.7 Configuring Automatic Consumer Group Creation for Kafka

When the **Auto Create Consumer Group** function is enabled, a consumer group will be automatically created when a consumer enters it. When this parameter is disabled, manually create a consumer group before consuming messages. Otherwise, message consumption will fail.

Notes and Constraints

- Enabled by default for new Kafka instances.
- Modifying this function does not restart Kafka instances.

Configuring Automatic Consumer Group Creation for Kafka

- Step 1 Log in to the Kafka console.
- **Step 2** Click in the upper left corner to select the region where your instance is located.
- **Step 3** Click a Kafka instance name to go to the instance details page.
- **Step 4** Modify the automatic consumer group creation function of Kafka in either of the following ways:
 - In the **Instance Information** area on the **Overview** page, configure the function.
 - Choose Instance > Parameters in the navigation pane on the left. Click Edit next to auto.create.groups.enable. Configure the function in Current Value.

----End

7.8 Exporting Kafka Consumer Groups

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

Procedure

- **Step 1** Log in to the **Kafka console**.
- **Step 2** Click in the upper left corner to select the region where your instance is located.
- **Step 3** Click the desired instance to go to the instance details page.
- **Step 4** In the navigation pane, choose **Instance** > **Consumer Groups**.
- **Step 5** Export consumer groups in either of the following ways:
 - Select the desired consumer groups and choose Export > Export selected data to an XLSX file to export specified consumer groups.
 - Choose **Export** > **Export all data to an XLSX file** to export all consumer groups.

----End

7.9 Deleting a Kafka Consumer Group

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

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

Notes and Constraints

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

Prerequisite

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

Deleting a Consumer Group

A consumer group can be deleted on the Kafka console or on a client.

Deleting a Consumer Group (Console)

- **Step 1** Log in to the **Kafka console**.
- **Step 2** Click in the upper left corner to select the region where your instance is located.
- **Step 3** Click the desired instance to go to the instance details page.
- **Step 4** In the navigation pane, choose **Instance** > **Consumer Groups**.
- **Step 5** Delete consumer groups using either of the following methods:
 - Select one or more consumer groups and click **Delete Consumer Group** above the consumer group list.
 - In the row containing the consumer group to be deleted, click **Delete**.
 - Click the consumer group to be deleted. The consumer group details page is displayed. Click **Delete** in the upper right corner.
- **Step 6** In the displayed **Delete Consumer Group** dialog box, click **OK**.

The consumer groups are deleted when they are no longer displayed in the consumer group list.

----End

Deleting a Consumer Group (Kafka Client)

The following uses Linux as an example.

For a Kafka instance with ciphertext access disabled, run the following command in the /bin directory of the Kafka client:
 ./kafka-consumer-groups.sh --bootstrap-server {connection-address} --delete --group {group-name}

Table 7-9 Consumer group deletion parameters

Parameter	Description	
connection-address	Connection address of a Kafka instance. To obtain the address, choose Overview > Connection .	
group-name	Consumer group name.	

Example:

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

- For a Kafka instance with ciphertext access enabled, do as follows:
 - a. (Optional) If the SSL certificate has been configured, skip this step and go to **b**. Otherwise, do as follows:

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

b. In the **/bin** directory of the Kafka client, run the following command: ./kafka-consumer-groups.sh --bootstrap-server *{connection-address}* --delete --group *{group-name}* --command-config ../config/*{ssl-user-config.properties}*

Table 7-10 Consumer group deletion parameters

Parameter	Description	
connection-address	Connection address of a Kafka instance. To obtain the address, choose Overview > Connection .	
group-name	Consumer group name.	
ssl-user- config.properties	Configuration file name. This file contains username, password, and SSL certificate information.	

Example:

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

Related Document

To delete a consumer group by calling an API, see **Deleting a Specified Consumer Group**.

7.10 Unsubscribing a Kafka Consumer Group from a Topic

The consumption progress in a topic can be viewed on the consumer group details page. When a consumer no longer consumes a topic, unsubscribing the consumer group from the topic can prevent false alarms caused by topic message accumulation.

Notes and Constraints

- This function may not be available for existing Kafka 1.1.0 or 2.3.0 instances. In this case, you can **upgrade the instance kernel**.
- This function will permanently delete the consumer offset in the topic. Exercise caution.

Procedure

- **Step 1** Log in to the **Kafka console**.
- **Step 2** Click on the upper left corner to select the region where your instance is located.
- **Step 3** Click the desired instance to go to the instance details page.
- **Step 4** In the navigation pane, choose **Instance** > **Consumer Groups**.
- **Step 5** Click the name of the desired consumer group.
- **Step 6** On the **Consumer Offset** tab page, select the topics to be unsubscribed, and click **Unsubscribe** on the upper left side. Up to **50** topics can be selected at a time.
- **Step 7** In the **Unsubscribe** dialog box, click **Yes**.

Unsubscribed topics are no longer displayed on the **Consumer Offset** tab page.

8 Managing Quotas

8.1 Configuring Kafka Quotas

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

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

Notes and Constraints

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

Operation Impact

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

Prerequisites

- To configure user quotas, **enable ciphertext access** on the Kafka details page and then obtain the username on the **Instance** > **Users** page on the console.
- To control client traffic, obtain the client ID from the client configuration.
- To control topic traffic, obtain the topic name from the Instance > Topics page.

Creating a Quota

The following sections describe how to create a user, client, or topic quota.

Creating a User or Client Quota

- **Step 1** Log in to the **Kafka console**.
- **Step 2** Click on the upper left corner to select the region where your instance is located.
- **Step 3** Click the desired instance to go to the instance details page.
- **Step 4** In the navigation pane, choose **Instance** > **Kafka Quotas**.
- **Step 5** Click the **User/Client** tab.
- **Step 6** In the upper left, click **Create Quota**. The **Create Quota** slide panel is displayed.
- **Step 7** Set quota parameters.

Create Quota

Figure 8-1 Creating a user/client quota

Configuring quotas may result in higher request latency, production timeout, and message losses. Learn more 🔀 Username and client ID cannot be both empty. Username Use Default test Use Default: The quota applies to all users. Client ID Enter a client ID. Use Default Use Default: The quota applies to all clients. Leave empty to apply no rate limit. However, the production limit and consumption limit cannot be both empty. (MB/s) Production Limit (MB/s) Consumption Limit

Table 8-1 Quota parameters

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

^! CAUTION

- **Username** is not displayed in the **Create Quota** dialog box for instances with ciphertext access disabled.
- Username and Client ID cannot be both empty.
- Production Limit and Consumption Limit cannot be both empty.
- **Step 8** Click **OK**. The **Background Tasks** page is displayed. If the status of the quota creation task is **Successful**, the quota has been created.

Go to the **Instance** > **Kafka Quotas** page. On the **User/Client** tab page, view the created quota in either of the following ways.

- For instances with ciphertext access disabled: Enter the name of the created quota in the search box and press **Enter**.
- For instances with ciphertext access enabled: Click **User quotas**, **Client quotas**, or **User and client quotas** in the upper left corner, select the type of the new quota, enter the quota name in the search box, and press **Enter**.

Figure 8-2 Viewing the new quota



Creating a Topic Quota

- Step 1 Log in to the Kafka console.
- **Step 2** Click on the upper left corner to select the region where your instance is located.
- **Step 3** Click the desired instance to go to the instance details page.
- **Step 4** In the navigation pane, choose **Instance** > **Kafka Quotas**.
- Step 5 Click the Topic tab.
- Step 6 In the upper left, click Create Quota. The Create Quota slide panel is displayed.
- **Step 7** Set quota parameters.

Figure 8-3 Creating a topic quota

Create Quota

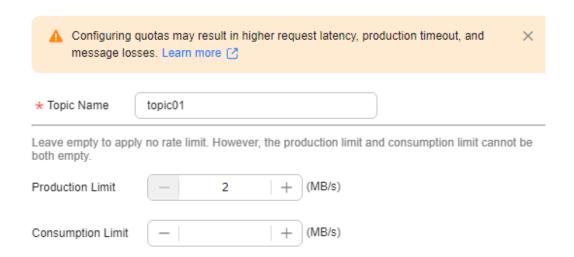


Table 8-2 Quota parameters

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



Production Limit and Consumption Limit cannot be both empty.

Step 8 Click **OK**. The **Background Tasks** page is displayed. If the status of the quota creation task is **Successful**, the quota has been created.

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

----End

Modifying a Quota

- **Step 1** Log in to the **Kafka console**.
- **Step 2** Click in the upper left corner to select the region where your instance is located.
- **Step 3** Click the desired instance to go to the instance details page.
- **Step 4** In the navigation pane, choose **Instance** > **Kafka Quotas**.
- **Step 5** In the row containing the desired quota, click **Edit**.
- **Step 6** Change the production limit or consumption limit, and click **OK**. The **Background Tasks** page is displayed. If the status of the quota modification task is **Successful**, the quota has been modified.

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

Production Limit and Consumption Limit cannot be both empty.

----End

Exporting Quotas

- Step 1 Log in to the Kafka console.
- **Step 2** Click in the upper left corner to select the region where your instance is located.
- **Step 3** Click the desired instance to go to the instance details page.
- **Step 4** In the navigation pane, choose **Instance** > **Kafka Quotas**.
- Step 5 Export quotas.
 - For specified user/client quotas: On the **User/Client** tab page, select desired user/client quotas and choose **Export** > **Export** selected data to an **XLSX** file.
 - For all user/client quotas: On the User/Client tab page, choose Export >
 Export all data to an XLSX file.
 - For specified topic quotas: On the **Topic** tab page, select desired topic quotas and choose **Export** > **Export selected data to an XLSX file**.
 - For all topic quotas: On the **Topic** tab page, choose **Export** > **Export all data to an XLSX file**.

Deleting a Quota

- **Step 1** Log in to the **Kafka console**.
- **Step 2** Click in the upper left corner to select the region where your instance is located.
- **Step 3** Click the desired instance to go to the instance details page.
- **Step 4** In the navigation pane, choose **Instance** > **Kafka Quotas**.
- **Step 5** In the row containing the desired quota, click **Delete**.
- **Step 6** Click **OK**. The **Background Tasks** page is displayed. If the status of the quota deletion task is **Successful**, the quota has been deleted.

----End

Related Documents

- To create user/client quotas by calling an API, see Creating User or Client Quotas.
- To create a topic quota by calling an API, see Creating a Topic Quota.

8.2 Monitoring Kafka Quotas

If quotas have been configured for a Kafka instance, the bandwidth usage by user/client/topic of each broker under certain quota policies can be viewed on the console.

Notes and Constraints

Unavailable for single-node instances.

Viewing Bandwidth Usage

- **Step 1** Log in to the **Kafka console**.
- **Step 2** Click in the upper left corner to select the region where your instance is located.
- **Step 3** Click the desired instance to go to the instance details page.
- **Step 4** In the navigation pane, choose **Monitoring** > **Quota Monitoring**.
- **Step 5** Set the parameters to guery bandwidth usage.

Figure 8-4 Bandwidth usage parameters

Table 8-3 Bandwidth usage query parameters

Parameter	Description	
Search By	Specify the criteria by which the bandwidth usage is to be searched.	
	Ranked: Show the specified number of users, clients, or topics that have used the most bandwidth.	
	Bandwidth: Show users, clients, or topics whose bandwidth rate is higher than your specified value.	
	Bandwidth usage: Show users, clients, or topics whose bandwidth usage is higher than your specified percentage.	
Bandwidth	Specify the bandwidth usage data source.	
From	Production: Count production bandwidth usage.	
	Consumption: Count consumption bandwidth usage.	
Dimension	Specify the bandwidth usage data dimension.	
	User/Client: Count user/client bandwidth usage.	
	Topic: Count topic bandwidth usage.	

Step 6 Click **Search** to view the bandwidth usage of users, clients, and topics of each broker.

9 Managing Instances

9.1 Viewing and Modifying Basic Information of a Kafka Instance

After creating a Kafka instance, you can view the details or modify some parameters of it on the console as required. These parameters include the instance name, description, security group, and capacity threshold policy.

Notes and Constraints

Single-node instances do not support reconfiguration of Smart Connect and private network access.

Prerequisite

You can modify basic information of a Kafka instance when the instance is in the **Running** state.

Viewing Kafka Instance Details

- **Step 1** Log in to the **Kafka console**.
- **Step 2** Click in the upper left corner to select the region where your instance is located.
- **Step 3** Specific Kafka instances can be queried using filters. You can filter instances by name, status, version, instance type, flavor, used/available space, maximum partitions, billing mode, AZ, instance ID, creation time, private connection address, public connection address, enterprise project, and resource tag.
 - Only enterprise users can filter instances by enterprise projects.
 - Resource tags can be filtered by operator. Available operators include **is equal** to =, **is not equal to !=, contains :**, and **does not contain :**.
 - Table 9-1 describes the various possible statuses of a Kafka instance.

Status Description Creating The instance is being created. The instance failed to be created. Creation failed The instance is running properly. Running Only instances in the **Running** state can provide services. Faulty The instance is not running properly. Starting The status between **Frozen** and **Running**. Restarting The instance is being restarted. Changing The instance specifications or public access configurations are being modified. Change The instance specifications or public access configurations failed failed to be modified. You cannot restart, delete, or modify an instance in the Change failed state. Contact customer service. Frozen The instance is frozen. Freezing The status between **Running** and **Frozen**. Upgrading The instance is being upgraded. Rolling The instance is being rolled back. back Binning The instance is being moved to the recycle bin. Binned The instance is in the recycle bin.

Table 9-1 Kafka instance status description

Step 4 Click the name of the desired Kafka instance and view detailed information about the instance on the **Overview** tab page.

Table 9-2 and **Table 9-3** describe the parameters for connecting to a Kafka instance. For details about other parameters, see the **Basic Information** tab page of the Kafka instance on the console.

The instance is being recovered from the recycle bin.

Table 9-2 Connection parameters (SASL_SSL cannot be changed)

Recovering

Section	Parameter	Description
Connectio n	Username	Username for accessing the instance with SASL_SSL enabled.

Section	Parameter	Description
	Kafka SASL_SSL	Whether SASL_SSL is enabled. This function is unavailable for single-node instances.
	Security Protocol	Security protocol used by the instance with SASL_SSL enabled.
	SASL Mechanism	SASL mechanism used by the instance with SASL_SSL enabled.
	SSL Certificate	Click Download to download the SSL certificate for accessing the instance.
	Instance Address	Address for connecting to the instance when public access is disabled.
	(Private Network)	The number of connection addresses is the same as that of brokers.
	Manager Address (Private Network)	Address for connecting to Kafka Manager when public access is disabled.
		Instances created since May 17, 2023 do not have this address.
	Manager Username	Username for connecting to Kafka Manager. Instances created since May 17, 2023 do not have this username.
	Public Access	Indicates whether public access has been enabled for the instance.
	Instance Address	Address for connecting to the instance when public access is enabled.
	(Public Network)	This parameter is displayed only when public access is enabled.
	Manager Address (Public Network)	Address for connecting to Kafka Manager when public access is enabled.
		This parameter is displayed only when public access is enabled.
		Instances created since May 17, 2023 do not have this address.
	Intra-VPC Plaintext Access	Whether intra-VPC plaintext access is enabled.

 Table 9-3 Connection parameters (plaintext and ciphertext access)

Sectio n	Parame ter	Sub- Parameter	Description
Conne ction	Userna me	-	Username for accessing the instance with ciphertext access enabled.
	Private Network	Plaintext Access	Indicates whether plaintext access is enabled.
	Access	Address (Private Network, Plaintext)	This parameter is displayed only after you enable Plaintext Access .
		Ciphertext Access	Indicates whether ciphertext access is enabled.
			This function is unavailable for single-node instances.
		Address (Private Network, Ciphertext)	This parameter is displayed only after you enable Ciphertext Access .
		Security Protocol	This parameter is displayed only after you enable Ciphertext Access .
	Public Network	Toggle switch	Indicates whether public access has been enabled.
	Access	Plaintext Access	This parameter is displayed only when Public Access is enabled.
			Indicates whether plaintext access is enabled.
		Address (Public Network, Plaintext)	This parameter is displayed only after you enable Plaintext Access .
		Ciphertext Access	This parameter is displayed only when Public Access is enabled.
			Indicates whether ciphertext access is enabled.
			This function is unavailable for single-node instances.
		Address (Public Network, Ciphertext)	This parameter is displayed only after you enable Ciphertext Access .

Sectio n	Parame ter	Sub- Parameter	Description
		Security Protocol	This parameter is displayed only after you enable Ciphertext Access .
	SASL Mechani sm	-	This parameter is displayed only after you enable Ciphertext Access .
	SSL Certifica	-	This parameter is displayed only when SASL_SSL is enabled.
	te		Click Download to download the SSL certificate for accessing the instance.

Modifying Basic Information of a Kafka Instance

- Step 1 Log in to the Kafka console.
- **Step 2** Click in the upper left corner to select the region where your instance is located.
- **Step 3** Click the desired instance to go to the instance details page.
- **Step 4** Modify the following parameters if needed:

Table 9-4 Modifiable Kafka parameters

Parameter	How to Modify	Result
Instance Name	Click , enter a new name, and click . Naming rules: 4–64 characters; starts with a letter; can contain only letters, digits, hyphens (-), and underscores (_).	The modification result is displayed in the upper right corner of the page.
Smart Connect	See Enabling Smart Connect and Disabling Smart Connect.	To check the progress and result of the current task, go to the Instance > Background Tasks page.
Description	Click , enter a new description, and click . 0 to 1024 characters.	The modification result is displayed in the upper right corner of the page.

Parameter	How to Modify	Result
Enterprise Project	Click , select a new enterprise project from the drop-down list, and click . Only for enterprise users. Modifying this parameter does not restart the instance.	The modification result is displayed in the upper right corner of the page.
Security Group	Click , select a new security group from the drop-down list, and click . Modifying this parameter does not restart the instance.	The modification result is displayed in the upper right corner of the page.
Private Network Access	See Configuring Plaintext or Ciphertext Access to Kafka Instances.	You will be redirected to the Background Tasks page, which displays the modification progress and result.
Public Access	See Configuring Kafka Public Access.	You will be redirected to the Background Tasks page, which displays the modification progress and result.
Capacity Threshold Policy	Click the desired policy. In the displayed Confirm dialog box, click OK . Modifying this parameter does not restart the instance. When the policies are triggered, messages may be deleted or cannot be produced.	You will be redirected to the Background Tasks page, which displays the modification progress and result.
Automatic Topic Creation	Enable/Disable this Automatic Topic Creation . In the displayed Confirm dialog box, click OK . Changing this option may restart the instance.	You will be redirected to the Background Tasks page, which displays the modification progress and result.
Auto Create Consumer Group	See Configuring Automatic Consumer Group Creation for Kafka.	You will be redirected to the Background Tasks page, which displays the modification progress and result.
Disk Autoscaling	See Automatically Expanding the Storage Space of a Kafka Instance.	The modification result is displayed in the upper right corner of the page.

Parameter	How to Modify	Result
Cross-VPC Access	See Accessing Kafka Using a VPC Endpoint Across VPCs and Accessing Kafka in a Public Network Using DNAT.	The modification result is displayed in the upper right corner of the page.

Related Documents

- To query Kafka instance information by calling an API, see Querying an Instance.
- To modify basic Kafka instance information by calling an API, see Modifying Instance Information.

9.2 Viewing Kafka Disk Usage

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

Notes and Constraints

Unavailable for single-node instances.

Procedure

- Step 1 Log in to the Kafka console.
- **Step 2** Click in the upper left corner to select the region where your instance is located.
- **Step 3** Click the name of the desired Kafka instance to go to the **Overview** page.
- **Step 4** Go to the **Monitoring > Disk Usage Statistics** page.

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

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

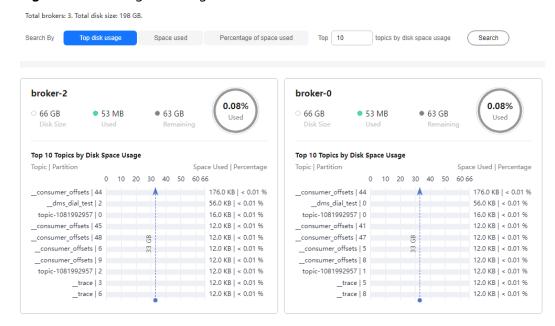


Figure 9-1 Viewing disk usage

Related Document

To view the disk usage of a topic by calling an API, see **Querying the Disk Usage Status of Topics**.

9.3 Viewing Kafka Background Tasks

After you initiate certain instance operations listed in **Table 9-5**, a background task will start for each operation. On the console, you can view the background task status and clear task information by deleting task records.

Table 9-5 Backend task list

Task Name	Description
Creating an instance	Creates a Kafka instance.
Restart Instance	Restarts a Kafka instance.
Modifying Kafka parameters	Modifies configuration parameters of Kafka.Enables/Disables automatic topic creation.
Change capacity threshold policy	Changes capacity threshold policies for a Kafka instance.
Enabling or disabling SSL	Switches between plaintext and ciphertext access.
Configure public network access	Enables/Disables public access.

Task Name	Description
Enable Smart Connect	Enables Smart Connect.
Disable Smart Connect	Disables Smart Connect.
Modify Specifications	 Expands the storage space. Adds brokers. Increases the bandwidth. Increases the broker flavor. Decreases the broker flavor.
Create Quota	Creates user/client/topic quotas.
Modify Quota	Modifies quotas.
Delete Quota	Deletes user/client/topic quotas.
Kafka partition reassignment	Reassigns partitions of a topic.
Enable logging	Enables rebalancing logging.
Disable logging	Disables rebalancing logging.
Configure topic permission	Grants permissions to users in a topic.

Procedure

- Step 1 Log in to the Kafka console.
- **Step 2** Click in the upper left corner to select the region where your instance is located.
- **Step 3** Click the name of the desired Kafka instance to go to the **Overview** page.
- **Step 4** In the navigation pane, choose **Instance** > **Background Tasks**.
- **Step 5** On the **Current Tasks** or **Scheduled Tasks** tab page, click the time drop-down list and select a time range. Enter a keyword in the search box and press **Enter**. The tasks started in the specified time range are displayed.

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

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

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

Related Document

To view a background task list by calling an API, see Listing Background Tasks.

9.4 Viewing Sample Code of Kafka Production and Consumption

Distributed Message Service for Kafka allows you to view sample Java, Go, and Python code of producing and consuming messages on the console. You can quickly complete Kafka client integration.

Procedure

- **Step 1** Log in to the **Kafka console**.
- **Step 2** Click on the upper left corner to select the region where your instance is located.
- **Step 3** Click the desired instance to go to the instance details page.
- **Step 4** In the navigation pane, choose **Instance** > **Topics**.
- **Step 5** Click **View Sample Code**. The **Sample Code** dialog box is displayed.

The sample code is available in Java, Go, and Python, and PlainText, SASL_SSL, and SASL_PLAINTEXT access modes.

- PlainText: Accessing the Kafka instance in plaintext.
- SASL_SSL: Accessing the Kafka instance in ciphertext with SASL authentication. SASL authentication uses PLAIN or SCRAM-SHA-512.
- SASL_PLAINTEXT: Accessing the Kafka instance in plaintext with SASL authentication. SASL authentication uses PLAIN or SCRAM-SHA-512.

----End

9.5 Modifying Kafka Instance Configuration Parameters

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

Kafka instances have dynamic and static parameters:

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

Notes and Constraints

- Configuration parameters of some old instances cannot be modified. Check whether your instance parameters can be modified on the console. If they cannot be modified, contact customer service.
- This function is not available for single-node instances.

Prerequisites

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

Procedure

- **Step 1** Log in to the **Kafka console**.
- **Step 2** Click in the upper left corner to select the region where your instance is located.
- **Step 3** Click the desired instance to go to the instance details page.
- **Step 4** Choose **Instance** > **Parameters**.
 - To modify a specific parameter, locate the row containing it, and click **Edit**.
 - To modify parameters in batches, click Modify.
 - To restore parameters to their default values, choose Edit > Restore Default or Modify > Restore Default.

Parameters of v1.1.0 instances are described in **Table 9-6** and **Table 9-7**. Parameters of v2.3.0/v2.7/v3.x instances are described in **Table 9-8** and **Table 9-9**.

Table 9-6 Dynamic parameters (v1.1.0 instances)

Parameter	Description	Value Range	Default Value
auto.create.groups .enable	Whether to automatically create consumer groups.	true or false	true
	You can modify this parameter on the console only for instances created on or after April 25, 2023. For instances created before April 25, 2023, the function of automatically creating consumer groups is enabled by default and cannot be disabled on the console.		

Parameter	Description	Value Range	Default Value
offsets.retention. minutes	The longest period a consumption position can be retained starts from the time of submission. Positions retained beyond this duration will be deleted. Each time a consumption position is submitted to a topic partition, its retention period resets to 0. This is a static parameter for instances created before May 1, 2023.	1,440– 30,240 Unit: minute	20,160

Table 9-7 Static parameters (v1.1.0 instances)

Parameter	Description	Value Range	Default Value
min.insync.replicas	If a producer sets the acks parameter to all (or -1), the min.insync.replicas parameter specifies the minimum number of replicas that must acknowledge a write for the write to be considered successful.	1-3	1
message.max.byte s	Maximum length of a single message.	0- 10,485,76 0 Unit: byte	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 as a last resort, even though doing so may result in data loss.	true or false	false
connections.max.i dle.ms	Idle connection timeout (in ms). Connections that are idle for the duration specified by this parameter will be closed.	5,000– 600,000 Unit: millisecon d	600,000

Parameter	Description	Value Range	Default Value
log.retention.hour s	Message retention duration of an instance, in hours. This parameter takes effect only for topics that have no aging time configured. If there is aging time configured for topics, it overrides this parameter.	1–168 Unit: hour	72
max.connections.p er.ip	The maximum number of connections allowed from each IP address. Request for new connections will be rejected once the limit is reached.	100- 20,000	1000
group.max.session .timeout.ms	Maximum session timeout for consumers. A longer timeout gives consumers more time to process messages between heartbeats but results in a longer time to detect failures.	6,000– 1,800,000 Unit: millisecon d	1,800,000
default.replication .factor	The default number of replicas configured for an automatically created topic.	1–3	3
allow.everyone.if.n o.acl.found	 When this parameter is set to true, all users can access resources without ACL rules. When this parameter is set to false, the initial user has all the permissions and other users require authorization. All the permissions cover modifying a topic, creating and deleting a topic, and changing the number of topic partitions. This parameter is displayed only when ciphertext access is enabled for the instance. This parameter cannot be modified for instances created before September 15, 2023. 	true or false	true
num.partitions	The default number of partitions configured for each automatically created topic.	1–200	3

Parameter	Description	Value Range	Default Value
group.min.session. timeout.ms	Minimum session timeout for consumers. A shorter timeout enables quicker failure detection but results in more frequent consumer heartbeating, which can overwhelm broker resources.	6,000– 300,000 Unit: millisecon d	6,000

Table 9-8 Dynamic parameters (v2.3.0/v2.7/v3.x)

Parameter	Description	Value Range	Default Value
min.insync.replicas	If a producer sets the acks parameter to all (or -1), the min.insync.replicas parameter specifies the minimum number of replicas that must acknowledge a write for the write to be considered successful.	1-3	1
message.max.byte s	Maximum length of a single message.	0- 10,485,76 0 Unit: byte	10,485,76 0
auto.create.groups .enable	Whether to automatically create consumer groups. You can modify this parameter on the console only for instances created on or after April 25, 2023. For instances created before April 25, 2023, the function of automatically creating consumer groups is enabled by default and cannot be disabled on the console.	true or false	true
max.connections.p er.ip	The maximum number of connections allowed from each IP address. Request for new connections will be rejected once the limit is reached.	100- 20,000	1000
unclean.leader.ele ction.enable	Indicates whether to allow replicas not in the ISR set to be elected as the leader as a last resort, even though doing so may result in data loss.	true or false	false

Parameter	Description	Value Range	Default Value
offsets.retention. minutes	The longest period a consumption position can be retained starts from the time of submission. Positions retained beyond this duration will be deleted. Each time a consumption position is submitted to a topic partition, its retention period resets to 0. This is a static parameter for instances created before May 1, 2023.	1,440– 30,240 Unit: minute	20,160

Table 9-9 Static parameters (v2.3.0/v2.7/v3.x)

Parameter	Description	Value Range	Default Value
connections.max.i dle.ms	Idle connection timeout (in ms). Connections that are idle for the duration specified by this parameter will be closed.	5,000- 600,000 Unit: millisecon d	600,000
log.retention.hour	Message retention duration of an instance, in hours. This parameter takes effect only for topics that have no aging time configured. If there is aging time configured for topics, it overrides this parameter.	1–168 Unit: hour	72
group.max.session .timeout.ms	Maximum session timeout for consumers. A longer timeout gives consumers more time to process messages between heartbeats but results in a longer time to detect failures.	6,000– 1,800,000 Unit: millisecon d	1,800,000
default.replication .factor	The default number of replicas configured for an automatically created topic.	1-3	3

Parameter	Description	Value Range	Default Value
allow.everyone.if.n o.acl.found	 When this parameter is set to true, all users can access resources without ACL rules. When this parameter is set to false, the initial user has all the permissions and other users require authorization. All the permissions cover modifying a topic, creating and deleting a topic, and changing the number of topic partitions. This parameter is displayed only when ciphertext access is enabled for the instance. This parameter cannot be 	true or false	true
	modified for instances created before September 15, 2023.		
num.partitions	The default number of partitions configured for each automatically created topic.	1–200	3
group.min.session. timeout.ms	Minimum session timeout for consumers. A shorter timeout enables quicker failure detection but results in more frequent consumer heartbeating, which can overwhelm broker resources.	6,000– 300,000 Unit: millisecon d	6,000

Step 5 Click Save.

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

On the **Parameters** page, view the new parameter values in the **Current Value** column.

----End

Related Document

To modify parameters of a Kafka instance by calling an API, see **Modifying Instance Configurations**.

9.6 Configuring Kafka Instance Tags

Tags facilitate Kafka instance identification and management.

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

A tag consists of a tag key and a tag value. **Table 9-10** lists the tag key and value requirements.

Table 9-10 Tag key and value requirements

Parameter	Requirements
Tag key	Cannot be left blank.
	Must be unique for the same instance.
	Can contain 1 to 128 characters.
	Can contain letters, digits, spaces, and special characters:=+-@
	Cannot start or end with a space.
	Cannot start with _sys
Tag value	Can contain 0 to 255 characters.
	Can contain letters, digits, spaces, and special characters:=+-@
	Cannot start or end with a space in instance creation.

Notes and Constraints

Up to 20 tags can be added to a Kafka instance.

Configuring Tags for a Kafka Instance

- Step 1 Log in to the Kafka console.
- **Step 2** Click on the upper left corner to select the region where your instance is located.
- **Step 3** Configure Kafka instance tags in either of the following ways:

Method 1:

- 1. Click an instance name to go to the instance details page.
- 2. In the navigation pane, choose **Instance** > **Tags**.
- 3. Click Edit Tag.
- 4. Add, edit, or delete tags as required.

Table 9-11 Tag operations

Operation	Procedure
Adding a tag	 Click Add Tag to set tags with Tag key and Tag value. If you have predefined tags, select a predefined pair of tag key and value, and click Add. A maximum of 20 tags can be added. Click OK. View the new tag on the tag list page.
Editing a tag	Modify the tag key and value, and click OK . On the tag list page, view the new tag key and value.
Deleting a tag	In the row containing the tag to be deleted, click Delete . Then, click OK to delete the tag. The tags are deleted when they are no longer displayed in the tag list.

Method 2:

- 1. In the row containing the desired Kafka instance, choose **More** > **Edit Tag**.
- 2. Add, edit, or delete tags as required.

Table 9-12 Tag operations

Operation	Procedure
Adding a tag	 Click Add Tag to set tags with Tag key and Tag value. If you have predefined tags, select a predefined pair of tag key and value, and click Add. A maximum of 20 tags can be added. Click OK.
	3. In the row containing the Kafka instance with added tags, choose More > Edit Tag . View the added tags.
Editing a tag	 Modify the tag key and value, and click OK. In the row containing the Kafka instance with edited tags, choose More > Edit Tag. View the edited tag key and value.
Deleting a tag	 In the row containing the tag to be deleted, click Delete. Then, click OK to delete the tag. In the row containing the Kafka instance whose tags were deleted, choose More > Edit Tag. The tags are deleted when they are no longer displayed.

Related Document

To add or delete Kafka instance tags by calling an API, **Batch Adding or Deleting Tags**.

9.7 Configuring Kafka Recycling Policies

If recycling is enabled, deleted instances and their data are retained in Recycle Bin, and can be recovered during the retention period. Once the retention period expires, instances in Recycle Bin will be deleted permanently.

Recycling is disabled by default.

Notes and Constraints

- Pay-per-use instance in Recycle Bin will not generate fees, but their storage will.
- Yearly/Monthly instances will be moved to Recycle Bin upon unsubscription. After that, they will not generate fees, but their storage will.
- Yearly/Monthly instances will be changed to pay-per-use ones upon successful recovery.
- Removing or unsubscribing instances in the grace or retention period deletes them permanently.

Enabling Recycle Bin

- **Step 1** Log in to the **Kafka console**.
- **Step 2** Click in the upper left corner to select the region where your instance is located.
- **Step 3** In the navigation pane, choose **Recycle Bin**.
- **Step 4** Click **Modify Recycling Policy** and the **Modify Recycling Policy** dialog box is displayed.
- Step 5 Enable Recycle Bin, specify Retention Days (1–7), and click OK.
 - ----End

Recovering Kafka Instances

- **Step 1** Log in to the **Kafka console**.
- **Step 2** Click in the upper left corner to select the region where your instance is located.
- **Step 3** In the navigation pane, choose **Recycle Bin**.

- **Step 4** Recover Kafka instances using either of the following methods:
 - Select one or more Kafka instances and click **Recover** in the upper left corner.
 - In the row containing the desired Kafka instance, click **Recover**.
- **Step 5** In the displayed **Recover Instance** dialog box, click **OK**.

It takes 3 to 10 minutes to recover an instance. You can view recovered instances on the **Kafka Instances** page.

Yearly/Monthly instances will be changed to pay-per-use ones upon successful recovery.

----End

Modifying Retention Days

- **Step 1** Log in to the **Kafka console**.
- **Step 2** Click on the upper left corner to select the region where your instance is located.
- **Step 3** In the navigation pane, choose **Recycle Bin**.
- **Step 4** Click **Modify Recycling Policy** and the **Modify Recycling Policy** dialog box is displayed.
- **Step 5** Modify the retention days (1–7) and click **OK**.

Changes to the retention period apply only to instances deleted after the changes.

----End

Exporting Instances in the Recycle Bin

- **Step 1** Log in to the **Kafka console**.
- **Step 2** Click on the upper left corner to select the region where your instance is located
- **Step 3** In the navigation pane, choose **Recycle Bin**.
- **Step 4** Export the instance list using either of the following methods:
 - Select the desired instances and choose Export > Export selected data to an XLSX file to export specified instances.
 - Choose Export > Export all data to an XLSX file to export all instances.

----End

Deleting Instances Permanently

- **Step 1** Log in to the **Kafka console**.
- **Step 2** Click in the upper left corner to select the region where your instance is located.

- **Step 3** In the navigation pane, choose **Recycle Bin**.
- **Step 4** Delete instances using either 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, click **Delete**.
- **Step 5** In the displayed **Delete Instance** dialog box, enter **DELETE** and click **OK**.

Deleting a Kafka instance in the recycle bin will **clear the instance data without any backup**. Exercise caution.

----End

Disabling Recycling

- **Step 1** Log in to the **Kafka console**.
- **Step 2** Click on the upper left corner to select the region where your instance is located.
- Step 3 In the navigation pane, choose Recycle Bin.
- **Step 4** Click **Modify Recycling Policy** and the **Modify Recycling Policy** dialog box is displayed.
- **Step 5** Disable **Recycle Bin** and click **OK**.

----End

9.8 Upgrading the Kafka Instance Kernel

Upgrade your Kafka instance kernel to use the latest kernel version. A kernel upgrade adds certain new features and resolves certain earlier issues. For example, new features may include consumer group creation on the console and topic details viewing.

Kafka instance kernel upgrades have no impact on the Kafka version. For example, if you use Kafka 2.7, you will still be using it after a kernel upgrade.

Impact of Kernel Upgrades

- Single-replica topics do not support message production and consumption during an upgrade, which will cause service interruptions.
- If a topic has multiple replicas, upgrades do not interrupt services. Brokers are restarted in sequence and workload is born on remaining brokers. Evaluate this impact and avoid peak hours.
- Brokers will be upgraded one by one. The software package and data of each broker will be updated. Upgrading the software package takes about 5 minutes. Synchronizing the data takes longer as the data volume of other brokers' leader replica becomes larger. Total upgrade duration = Software package upgrade duration of each broker + Data synchronization duration
- The monitoring process is restarted and the monitoring data is lost for each broker during the upgrade. The monitoring continues after the restart is complete.

- During an upgrade, broker restarts will cause partition leader switches, interrupting connections in seconds. The switch takes less than a minute when networks are stable. For multi-replica topics, configure retries on the producer client. To do so:
 - If you use an open-source Kafka client, configure the retries and retry.backoff.ms parameters. You are advised to set retries to 10 and retry.backoff.ms to 1000.
 - Flink: Configure the retry policy by referring to the following code:
 StreamExecutionEnvironment env = StreamExecutionEnvironment.getExecutionEnvironment();
 env.setRestartStrategy(RestartStrategies.fixedDelayRestart(10, Time.seconds(20)));

Prerequisites

The instance must be in the **Running** state.

Procedure

- **Step 1** Log in to the **Kafka console**.
- **Step 2** Click in the upper left corner to select the region where your instance is located.
- **Step 3** Upgrade a Kafka instance using either of the following methods:
 - Choose **More** > **Upgrade** in the row containing the desired instance.
 - Click the name of the Kafka instance to go to the instance details page.
 Choose --- > Upgrade in the upper right corner.
- **Step 4** Set **Execute** to **Now** or **As scheduled**.

If you select **As scheduled**, specify date and time.

Step 5 In the **Risk Check** area, check whether the items are normal.

If any risk is found, handle it as prompted and click **Recheck**. If the risk does not need to be handled, select **I understand the risks**.

Step 6 Click OK.

The method of checking the upgrade result depends on when the upgrade is executed.

Execution	Check Method
Now	 Click an instance name to go to the instance details page. Choose Instance > Background Tasks in the navigation pane.
	3. Check the upgrade task status on the Current Tasks tab page.
	 The upgrade is complete when the task is in the Successful state.
	 The upgrade failed when the task is in the Failed state. Contact customer service.
As scheduled	1. Click an instance name to go to the instance details page.
	2. Choose Instance > Background Tasks in the navigation pane.
	3. On the Scheduled tasks tab page, check whether the upgrade task is started.
	 The task has not been started when it is in the Pending state.
	 The task has been started when it is in the Successful state.
	4. Check the upgrade task status on the Current Tasks tab page.
	 The upgrade is complete when the task is in the Successful state.
	- The upgrade failed when the task is in the Failed state.

Table 9-13 Checking the upgrade result

Modifying Scheduled Upgrade Tasks

Step 1 Go to the **Scheduled tasks** tab page on the **Instance** > **Background Tasks** page, click the drop-down box in the upper left corner and select a period. Enter "upgrade" in the search box and press **Enter**.

Contact customer service.

- **Step 2** Click **Modify** in the row containing the desired task.
- **Step 3** In the **Change Schedule** dialog box, reschedule or cancel the task.
 - To reschedule: Specify a new time and click **OK**.
 - To cancel: Select Cancel and click OK.

----End

Related Document

To upgrade the kernel version of a Kafka instance by calling an API, see **Upgrading an Instance**.

9.9 Exporting the Kafka Instance List

You can export a list of instances on the DMS for Kafka console.

Procedure

- Step 1 Log in to the Kafka console.
- **Step 2** Click in the upper left corner to select the region where your instance is located.
- **Step 3** Export the instance list in either of the following ways:
 - Select the desired instances and choose Export > Export selected data to an XLSX file to export specified instances.
 - Choose Export > Export all data to an XLSX file to export all instances.

----End

9.10 Restarting a Kafka Instance

You can restart one or more Kafka instances in batches on the DMS for Kafka console.

Notes and Constraints

- When a Kafka instance is being restarted, message consumption and production requests of clients will be rejected.
- To maintain service connections during instance restart, configure the retry mechanism on the client.

Prerequisite

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

Procedure

- **Step 1** Log in to the **Kafka console**.
- **Step 2** Click [♥] in the upper left corner to select the region where your instance is located.
- **Step 3** 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 Kafka instance to be restarted, click More > Restart.
 - Click the desired Kafka instance to go to the instance details page. In the upper right corner, click •• > Restart.
- **Step 4** In the **Restart Instance** dialog box, click **OK** 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

Related Document

To restart Kafka instances by calling an API, see **Batch Restarting or Deleting Instances**.

9.11 Deleting Kafka Instances

For pay-per-use Kafka instances, you can delete one or more of them in batches on the console. For yearly/monthly Kafka instances, if you no longer need them, choose **More** > **Unsubscribe** in the **Operation** column. Kafka instances will be automatically deleted upon unsubscription.

Manage deleted instances using recycle bin policies. Deleting instances when no recycle bin policies are enabled clears instance data permanently. Recycle bin policies are disabled by default. To enable them, see **Enabling Recycle Bin**.

Prerequisites

The Kafka instance is in the **Running**, **Faulty**, **Frozen**, or **Creation failed** state.

Procedure

- Step 1 Log in to the Kafka console.
- **Step 2** Click on the upper left corner to select the region where your instance is located.
- **Step 3** Delete pay-per-use Kafka instances in either of the following ways:
 - Select one or more Kafka instances and click **Delete** in the upper left corner.
 - In the row containing the Kafka instance to be deleted, choose More > Delete.
 - Click the desired Kafka instance to go to the instance details page. In the upper right corner, choose ••• > **Delete**.
- **Step 4** In the **Delete Instance** dialog box, enter **DELETE** and click **OK** to delete the Kafka instance.

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

The Kafka instances are deleted when they are no longer displayed in the instance list.

Related Documents

- To delete a specific Kafka instance by calling an API, see **Deleting an Instance**.
- To delete Kafka instances in batches by calling an API, see Batch Restarting or Deleting Instances.

9.12 Using Kafka Manager

9.12.1 Accessing Kafka Manager

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

Instances created since May 17, 2023 do not have Kafka Manager. Kafka Manager's functions are provided on the Kafka console.

Table 9-14 Kafka Manager functions on the Kafka console

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

Prerequisites

Security group rules have been configured by referring to Table 9-15.

Table 9-15 Security group rule

Directio n	Protocol	Port	Source	Description
Inbound	ТСР	9999	IP address or IP address group of the Kafka client	Access Kafka Manager.

Logging In to Kafka Manager

Step 1 Create a Windows ECS with the same VPC and security group configurations as the Kafka instance. For details, see **Purchasing an ECS**.

If public access has been enabled, this step is optional. You can access the instance using the local browser. You do not need to create a Windows ECS.

- **Step 2** Obtain the Kafka Manager address on the instance details page.
 - If public network access has been disabled, the Kafka Manager address is Manager Address (Private Network).

Figure 9-2 Kafka Manager address (private network)

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

Figure 9-3 Kafka Manager address (public network)

Manager Address (Public Network) https://122. ■ .50:9999,https://122. ■ .36:9999 □

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

If public access is enabled, enter the Kafka Manager address in the address bar of the browser on the local PC. If public access is not enabled, log in to the ECS prepared in **Step 1** and enter the Kafka Manager address in the address bar of the browser on the ECS.

Step 4 Enter the username and password for logging in to Kafka Manager, which you set when creating the instance.

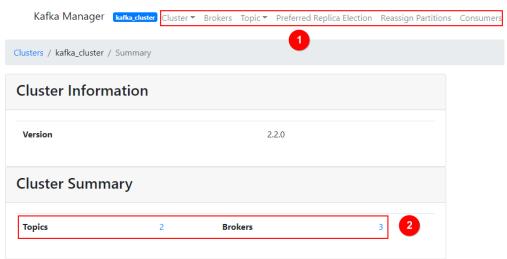
----End

Viewing Information in Kafka Manager

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

- Information about clusters
 - Click **Clusters** to view the information about clusters. **Figure 9-4** shows an example of the cluster information.
 - The top navigation bar provides the following functions, as shown in the red box 1 in the figure.
 - **Cluster**: viewing the list of clusters and cluster information.
 - Brokers: viewing information about brokers of a cluster.
 - **Topic**: viewing information about topics in a cluster.
 - Preferred Replica Election: electing the leader (preferred replica) of a topic. This operation is not recommended.
 - Reassign Partitions: reassigning partitions. This operation is not recommended.
 - Consumers: viewing the status of consumer groups in a cluster.
 - Red box 2 shows an example of the cluster information summary, including the number of topics and brokers in the cluster.

Figure 9-4 Information about clusters



Combined information about all brokers of a cluster

This page shows statistics of brokers of a cluster. **Figure 9-5** shows an example of the storage configuration.

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

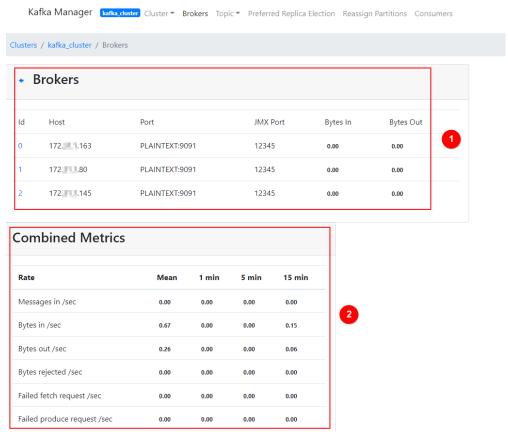


Figure 9-5 Viewing the combined information about all brokers in a cluster

Information about a specific broker

Click the ID of a broker to view its statistics. **Figure 9-6** shows an example of the storage configuration.

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

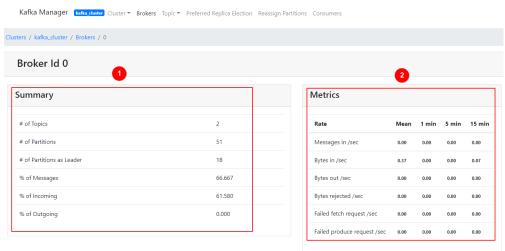


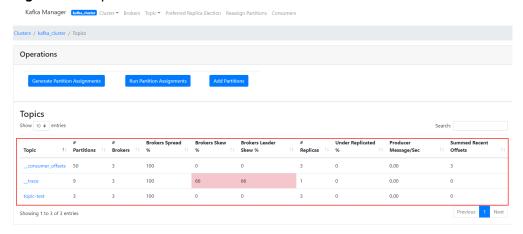
Figure 9-6 Viewing information about a broker

Topics of an instance

In the navigation bar, choose **Topic** > **List**. The displayed page shows the list of topics and information about the topics, as shown in **Figure 9-7**.

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

Figure 9-7 Topics of an instance



• Details of a topic

Click the name of a topic to view its details on the displayed page, as shown in **Figure 9-8**.

- Red box 1: basic information about the topic, including Replication,
 Number of Partitions, and Sum of Partition Offsets.
- Red box 2: information about partitions of different brokers.
- Red box 3: consumer groups of the topic. Click the name of a consumer group name to view its details.
- Red box 4: configurations of the topic. See Topic Configs.
- Red box 5: monitoring metrics of the topic.
- Red box 6: information about partitions in the topic, including Latest
 Offset, Leader of a partition, Replicas, and In Sync Replicas.

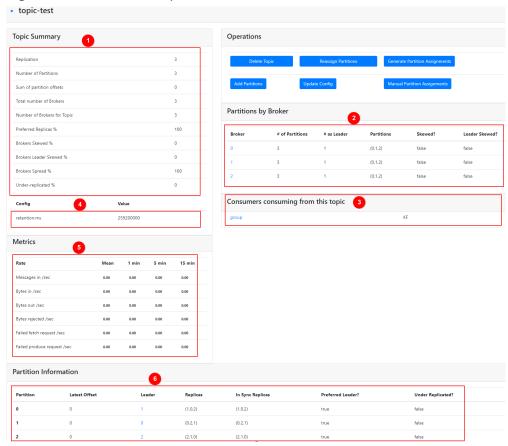


Figure 9-8 Details of a topic

List of consumers

Click **Consumers** to view the list of consumers in a cluster. Only consumer groups that have consumed messages in the last 14 days are displayed on this page.

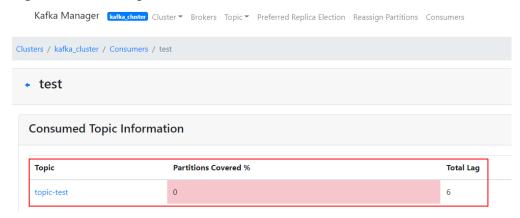
Figure 9-9 Viewing the list of consumers



Details of a specific consumer

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

Figure 9-10 Viewing consumer details



Details of topics in a consumer

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

Figure 9-11 Viewing details of a topic



9.12.2 Resetting Kafka Manager Password

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

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

Prerequisites

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

Procedure

- **Step 1** Log in to the **Kafka console**.
- **Step 2** Click in the upper left corner to select the region where your instance is located.
- **Step 3** Reset the Kafka Manager password using either of the following methods:
 - In the row containing the desired Kafka instance, choose More > Reset Manager Password.
 - Click the desired Kafka instance to go to the instance details page. Choose -- > Reset Manager Password in the upper right corner.
 - Click the desired Kafka instance to go to the instance details page. On the
 Overview page, click Reset Manager Password next to Manager Username
 in the Connection section.
- **Step 4** 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.

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

----End

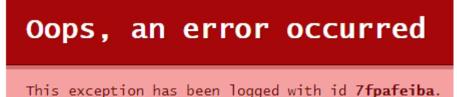
Related Document

To reset the Kafka Manager password by calling the API, see **Resetting Kafka Manager Password**.

9.12.3 Restarting Kafka Manager

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

Figure 9-12 Error information



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

Notes and Constraints

Restarting Kafka Manager does not affect services.

Procedure

- **Step 1** Log in to the **Kafka console**.
- **Step 2** Click on the upper left corner to select the region where your instance is located.
- **Step 3** Restart Kafka Manager using either of the following methods:
 - In the row containing the desired Kafka instance, choose **More** > **Restart Kafka Manager**.
 - Click the desired Kafka instance to go to the instance details page. Choose
 Restart Kafka Manager in the upper right corner.

Step 4 Click Yes.

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

----End

Related Document

To restart Kafka Manager by calling an API, see **Restarting Kafka Manager**.

9.12.4 Disabling Kafka Manager

Kafka Manager consumes memory and CPU. To free some resources, disable this function. This section describes how to disable Kafka Manager on the console.

Notes and Constraints

- Once disabled, Kafka Manager cannot be enabled.
- Disabling Kafka Manager does not restart the instance.

Procedure

- **Step 1** Log in to the **Kafka console**.
- **Step 2** Click in the upper left corner to select the region where your instance is located.
- **Step 3** Click the desired Kafka instance to go to the instance details page.
- **Step 4** On the **Overview** page, click next to **Kafka Manager** in the **Connection** area.

After Kafka Manager is disabled, the Kafka Manager connection address will not be displayed on the console, the Kafka Manager password cannot be reset, and Kafka Manager cannot be restarted.

----End

Related Document

To disable Kafka Manager by calling an API, see **Disabling Kafka Manager**.

10 Modifying Instance Specifications

10.1 Modifying Cluster Kafka Instance Specifications

After creating a Kafka instance, you can increase or decrease its specifications. **Table 10-1** lists available modification options.

Table 10-1 Specification modification options

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

Distinguishing Between Old and New Specifications

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

Figure 10-1 Instance list



Notes and Constraints

- Unavailable for single-node instances.
- After instance specifications are changed, the configuration fee changes accordingly.

Impact of Specification Modification

Table 10-2 Impact of specification modification

Modified Object	Impact
Bandwidth or broker quantity	 Increasing the bandwidth or adding brokers does not affect the original brokers or services. When you increase the bandwidth or change the broker quantity, the storage space is proportionally expanded based on the current disk space. For example, assume that the original number of brokers of an instance is 3 and the disk size of each broker is 200 GB. If the broker quantity changes to 10 and the disk size of each broker is still 200 GB, the total disk size becomes 2,000 GB. New topics are created on new brokers, and the original topics are still on the original brokers, resulting in unbalanced partitions. You can reassign partitions to
	migrate the replicas of the original topic partitions to the new brokers.
Storage space	 You can expand the storage space 20 times. Storage space expansion does not affect services.

Modified Object	Impact
Broker flavor	Single-replica topics do not support message production and consumption during this period. Services will be interrupted.
	If a topic has multiple replicas, modifying the broker flavor does not interrupt services. Brokers are restarted in sequence and workload is born by remaining brokers. Evaluate this impact and avoid peak hours.
	Broker rolling restarts will cause partition leader changes, interrupting connections for less than a minute when the network is stable. For multi-replica topics, configure the retry mechanism on the producer client. To do so:
	 If you use an open-source Kafka client, configure the retries and retry.backoff.ms parameters. You are advised to set retries to 10 and retry.backoff.ms to 1000.
	 If you use Flink, configure the retry policy by referring to the following code: StreamExecutionEnvironment env = StreamExecutionEnvironment.getExecutionEnvironment(); env.setRestartStrategy(RestartStrategies.fixedDelayRestart(10, Time.seconds(20)));
	 If the total number of partitions created for an instance is greater than the upper limit allowed by a new flavor, scaledown cannot be performed. The maximum number of partitions varies with instance specifications. For details, see Specifications. For example, if 800 partitions have been created for a kafka.4u8g.cluster*3 instance, you can no longer scale down the instance to kafka.2u4g.cluster*3 because this flavor allows only 750 partitions.
	It takes 5 to 10 minutes to modify specifications on one broker. The more brokers, the longer the modification takes.

Process of Increasing or Decreasing Broker Flavors

When you scale up or down the broker flavor, a rolling restart is performed on brokers. The following process takes three brokers as an example:

- 1. The Kafka process on Broker 0 is stopped.
- 2. The flavor of Broker 0 is scaled up or down.
- 3. The Kafka process on Broker 0 is restarted.
- 4. 1 to 3 are repeated to scale up or down the flavor of Broker 1.
- 5. 1 to 3 are repeated to scale up or down the flavor of Broker 2.

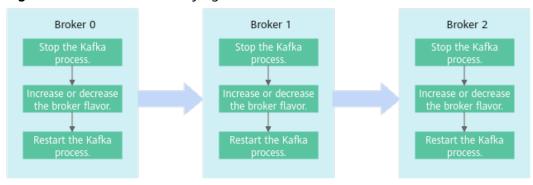


Figure 10-2 Process of modifying a broker flavor

Modifying Cluster Kafka Instance Specifications

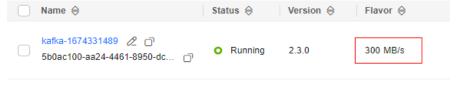
The following describes how to modify the specifications of a cluster Kafka instance.

Only one type of configurations among broker quantity/assured bandwidth, storage space, and broker flavor can be changed at a time.

Modifying the Broker Quantity or Bandwidth

- Step 1 Log in to the Kafka console.
- **Step 2** Click on the upper left corner to select the region where your instance is located.
- **Step 3** In the row containing the desired instance, click **Modify Specifications** in the **Operation** column.
- **Step 4** Specify the number of brokers or bandwidth as required.
 - Increase the bandwidth (for instances using old specifications)
 - a. Specify a new bandwidth and click Next.
 - b. Confirm the configurations and click **Submit**.
 - c. Return to the instance list and check whether the change succeeded.
 - If the instance status has changed from Changing to Running, the change succeeded. You can check the new bandwidth in the Flavor column.

Figure 10-3 Viewing the increased bandwidth



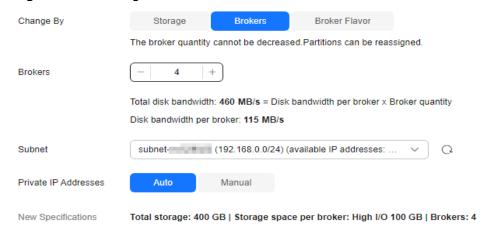
If the instance status has changed from Changing to Change failed, the change failed. Move the cursor over Change failed to check the failure cause.

Instances in the **Change failed** state cannot be restarted, modified, or deleted. After the instance status automatically changes from

Change failed to **Running**, you can continue to perform operations on the instance. If the status does not change to **Running**, contact customer service.

- d. After increasing the bandwidth, add the IP address of the new broker to the client connection configuration to improve reliability.
- Increase the broker quantity (for instances using new specifications).

Figure 10-4 Adding brokers



- a. For **Change By**, select **Brokers**.
- For Brokers, specify the broker quantity. The broker quantity range varies by instance specifications. For details, see Cluster Kafka instance specifications.
- c. If public access has been enabled, configure EIPs for the new brokers.
- d. For **Subnet**, retain the default settings.
- e. For Private IP Addresses, select Auto or Manual.
 - Auto: The system assigns an IP address from the subnet automatically.
 - Manual: Select the IP addresses for the new brokers from the dropdown list. If the number of selected IP addresses is less than the number of brokers, the remaining IP addresses will be automatically assigned.
- f. Click Next.
- g. Confirm the configurations and click **Submit**.
- h. Check the modification progress and estimated remaining time.
 - i. In the instance list, click the instance to go to the instance details page.
 - ii. In the navigation pane, choose **Instance** > **Background Tasks**. The **Current Tasks** tab page is displayed.
 - iii. Click the **Modify Specifications** task. The **Specification Modification Task Details** dialog box is displayed.
 - iv. Check the progress and estimated remaining time. In **Steps**, check the steps, start time, and end time.

- i. Check whether the modification is successful.
 - If the task is in the Successful state, the modification is successful.
 View the number of brokers in the Flavor column in the instance list.
 - If the task is in the Failed state, the modification is not successful. Move the cursor over Failed or check the cause in Steps.
 - After the modification fails, the instance is in the **Change failed** state, and cannot be restarted, modified, or deleted. After the instance status automatically changes from **Change failed** to **Running**, you can continue to perform operations. If the status does not change to **Running**, contact customer service.
- j. After adding brokers, add the IP addresses of the new brokers to the client connection configuration to improve reliability.

----End

Expanding the Storage Space

- Step 1 Log in to the Kafka console.
- **Step 2** Click in the upper left corner to select the region where your instance is located.
- **Step 3** In the row containing the desired instance, click **Modify Specifications** in the **Operation** column.
- **Step 4** Specify the required storage space.
 - Expand the storage space (for instances using old specifications)
 - a. Specify a new storage space and click **Next**.
 - b. Confirm the configurations and click **Submit**.
 - c. Return to the instance list and check whether the change succeeded.
 - If the instance status has changed from Changing to Running, the change succeeded. View the new storage space in the Used/ Available Storage Space (GB) column in the instance list.

Figure 10-5 Viewing the increased storage space

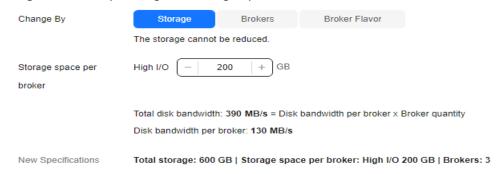


If the instance status has changed from Changing to Change failed, the change failed. Move the cursor over Change failed to check the failure cause.

Instances in the **Change failed** state cannot be restarted, modified, or deleted. After the instance status automatically changes from **Change failed** to **Running**, you can continue to perform operations on the instance. If the status does not change to **Running**, contact customer service.

Expand the storage space (for instances using new specifications)

Figure 10-6 Expanding the storage space



- a. For Change By, select Storage.
- For Storage space per broker, specify a new storage space, and click Next. The storage space range varies by instance specifications. For details, see Cluster Kafka instance specifications.
- c. Confirm the configurations and click **Submit**.
- d. Check the modification progress and estimated remaining time.
 - i. In the instance list, click the instance to go to the instance details page.
 - ii. In the navigation pane, choose Instance > Background Tasks. The Current Tasks tab page is displayed.
 - iii. Click the **Modify Specifications** task. The **Specification Modification Task Details** dialog box is displayed.
 - iv. Check the progress and estimated remaining time. In **Steps**, check the steps, start time, and end time.
- e. Check whether the modification is successful.
 - If the task is in the Successful state, the modification is successful. View the new storage space (Storage space per broker × Number of brokers) in the Used/Available Storage Space (GB) column in the instance list.
 - If the task is in the Failed state, the modification is not successful. Move the cursor over Failed or check the cause in Steps.

After the modification fails, the instance is in the **Change failed** state, and cannot be restarted, modified, or deleted. After the instance status automatically changes from **Change failed** to **Running**, you can continue to perform operations. If the status does not change to **Running**, contact customer service.

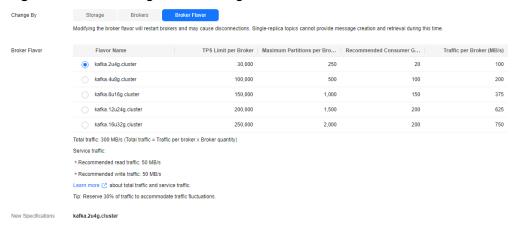
----End

Increasing or Decreasing the Broker Flavor

Step 1 Log in to the Kafka console.

- **Step 2** Click in the upper left corner to select the region where your instance is located.
- **Step 3** In the row containing the desired instance, click **Modify Specifications** in the **Operation** column.
- **Step 4** Specify the required broker flavor.

Figure 10-7 Increasing or decreasing a broker flavor



- 1. For Change By, select Broker Flavor.
- 2. Specify a new broker flavor.
- 3. In the **Risk Check** area, check for risks.

If any risk is found, handle it as prompted and then click **Recheck**. If the risks do not need to be handled, select **I understand the risks**.

- 4. Click **Next**, confirm the information, and click **Submit**.
- 5. Check the scaling progress and estimated remaining time.
 - a. In the instance list, click the instance to go to the instance details page.
 - In the navigation pane, choose Instance > Background Tasks. The Current Tasks tab page is displayed.
 - c. Click the **Modify Specifications** task. The **Specification Modification Task Details** dialog box is displayed.
 - d. Check the progress and estimated remaining time. In **Steps**, check the steps, start time, and end time.
- Check whether the modification is successful.
 - If the task is in the Successful state, the modification is successful. View the broker flavor in the Flavor column in the instance list.
 - If the task is in the Failed state, the modification is not successful. Move the cursor over Failed to view failure causes.

After the modification fails, the instance is in the **Change failed** state, and cannot be restarted, modified, or deleted. After the instance status automatically changes from **Change failed** to **Running**, you can continue to perform operations. If the status does not change to **Running**, contact customer service.

----End

Related Document

To change the specifications of a cluster Kafka instance by calling an API, see **Increasing Instance Specifications**.

10.2 Automatically Expanding the Storage Space of a Kafka Instance

Specific processing policies will be triggered when the disk usage of a Kafka instance reaches the threshold. As a result, messages may be deleted or the message production may fail. To avoid these situations, you can enable automatic disk expansion.

Notes and Constraints

- Unavailable for old instances. See the console.
- Unavailable for single-node instances.
- This function may not be completed before the disk usage reaches the threshold when the disk usage increases rapidly.
- Services will not be affected.

Prerequisite

Expanding the storage space of a Kafka instance will incur fees. Ensure that your account has sufficient balance. For details about the pricing, see **Product Pricing Details**.

Authorizing Kafka

Permission **DMS BSSAccess** is required. Otherwise, the automatic expansion fails.

- **Step 1** Log in to the **Kafka console** as the tenant account or a subaccount in user group **admin**.
- **Step 2** Click in the upper left corner to select a region.
- **Step 3** Choose **Service Authorization**.
- Step 4 Check BSS and click Update Authorization.
- **Step 5** Once the operation succeeds, agency **dms_admin_trust** is created in IAM.

Once the dms_admin_trust is created, do not modify or delete it.

----End

Enabling Automatic Storage Space Expansion

- Step 1 Log in to the Kafka console.
- **Step 2** Click on the upper left corner to select the region where your instance is located

- **Step 3** Click a Kafka instance name to go to the instance overview page.
- **Step 4** In **Disk Autoscaling** in the Instance Information area, click **∠**. The **Disk Autoscaling** dialog box is displayed.
- **Step 5** Click and set the parameters described in **Table 10-3**.

Table 10-3 Disk autoscaling parameters

Parameter	Description
Threshold	Threshold for triggering automatic disk scaling. The value range is 20% to 80%.
	When the disk usage of any node in a cluster Kafka instance reaches this threshold, the disk is automatically expanded.
Increase By	Proportion of the storage space to be expanded to the total storage space of the instance. The value range is 10% to 100%. The minimum increase is 100 GB. If the storage space to be expanded is not a multiple of 100, the value will be rounded up to 100.
	For example, when the total storage space of an instance is 300 GB and the increase is 10%, if the storage space to be expanded is $300 \times 10\% = 30$ GB, it will be rounded up to 100 GB.
Upper Limit	The upper limit of the storage space. After the total storage space of an instance reaches this limit, the instance will not automatically expand the storage space.

Step 6 Click OK.

Later, check that **Disk Autoscaling** is changed to **Enabled** in the **Instance Information** area on the Kafka instance overview page.

----End

Disabling Automatic Storage Space Expansion

- **Step 1** Log in to the **Kafka console**.
- **Step 2** Click on the upper left corner to select the region where your instance is located.
- **Step 3** Click a Kafka instance name to go to the instance overview page.
- Step 4 In Disk Autoscaling in the Instance Information area, click . The Disk Autoscaling dialog box is displayed.
- Step 5 In the Disk Autoscaling row, click and click OK.

Later, check that **Disk Autoscaling** is changed to **Disabled** in the **Instance Information** area on the Kafka instance overview page.

----End

11 Migrating Data

11.1 Kafka Data Migration Overview

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

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

Scheme Overview

Table 11-1 Migration scheme overview

Migration Scheme	Migrat ion Tool	Pros	Cons
Migrate production first, then consumption (without migrating data).	-	 This is a common migration solution in the industry. The procedure is simple and no additional plugins are required. In this scheme, the migration process is controlled by services. Orderly message consumption can be ensured. 	 There is latency during the switchover. The consumer needs to consume the original Kafka messages before consuming the new ones. After the consumption service is migrated, messages may be stacked in the new Kafka instance.
Consume messages from both Kafka instances and migrate the production later (without migrating data).	-	 The procedure is simple and no additional plugins are required. Messages are not stacked because source and target messages can be consumed at the same time. 	Early on in the migration, data is consumed from both the original and new Kafka instances, so the messages may not be consumed in the order that they are produced.
Migrate data using MirrorMaker, then consumption, then production.	Mirror Maker	The new Kafka instance synchronizes full historical data from the original Kafka instance and real-time incremental data.	After the consumer is migrated to the new Kafka instance, it may consume historical messages repeatedly. The consumer should support idempotent messages.

Migration Scheme	Migrat ion Tool	Pros	Cons
Migrate data using Smart Connect, then consumption, then production.	Smart Connec t	The new Kafka instance synchronizes full historical data from the original Kafka instance and real-time incremental data.	After the consumer is migrated to the new Kafka instance, it may consume historical messages repeatedly. The consumer should support idempotent messages.

Constraints

- When Smart Connect is used to migrate services, it consumes the source Kafka messages and produces messages to the target Kafka instance, occupying the bandwidth of the source and the target Kafka.
- To maintain performance, Smart Connect only synchronizes the source and target data in real time. The consumption progress is synchronized in batches, so the consumption progress on the source and target partitions may vary from 0 to 100.

Preparation

1. Configure the network environment.

A Kafka instance can be accessed within a VPC or over a public network. For public network access, the producer and consumer must have public access permissions, and the following security group rules must be configured.

Table 11-2 Security group rules

Directi on	Protocol	Port	Source	Description
Inboun d	ТСР	9094	IP address or IP address group of the Kafka client	Accessing a Kafka instance in a public network (in plaintext)
Inboun d	ТСР	9095	IP address or IP address group of the Kafka client	Accessing a Kafka instance in a public network (in ciphertext)

- 2. Create the target Kafka instance.
 - The specifications of the target instance cannot be lower than the original specifications. For more information, see **Buying a Kafka Instance**.
- 3. Create a topic in the target Kafka instance.

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

Migration Scheme 1: Migrating the Production First, then Consumption (Data Not Migrated)

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

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

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

----End

Migration Scheme 2: Consuming Both Messages and Migrating the Production Later (Data Not Migrated)

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

For a certain period of time, the consumption service consumes messages from both the original and new Kafka instances. Before the migration, message

consumption from the new Kafka instance has already started, so there is no latency. However, early on in the migration, data is consumed from both the original and new Kafka instances, so the messages may not be consumed in the order that they are produced. This scheme is suitable for services that require low latency but do not require strict message sequence.

- **Step 1** Start new consumer clients, set the Kafka connection addresses to that of the new Kafka instance, and consume data from the new Kafka instance.
 - Original consumer clients must continue running. Messages are consumed from both the original and new Kafka instances.
- **Step 2** Change the Kafka connection address of the producer to that of the new Kafka instance.
- **Step 3** Restart the producer client to migrate the production service to the new Kafka instance.
- **Step 4** After the production service is migrated, check whether the consumption service connected to the new Kafka instance is normal.
- **Step 5** After all data in the original Kafka is consumed, close the original consumption clients.
- **Step 6** The migration is complete.

----End

Migration Scheme 3: Migrating Data Using MirrorMaker First, then Consumption, and then Production

Use MirrorMaker to synchronize the two Kafka instances, migrate the consumer first and then the producer to the new Kafka instance.

This scheme depends on MirrorMaker. MirrorMaker synchronizes data between the original Kafka and the new Kafka. After data synchronization is complete, migrate the consumer to the new Kafka and then the producer. This scheme applies to the scenario where the producer cannot be stopped, the end-to-end latency cannot be high, but a little repeated message consumption is tolerated.

- **Step 1** Synchronize messages between two Kafka instances using MirrorMaker. For details, see **Using MirrorMaker to Synchronize Data Across Clusters**.
- **Step 2** Change the Kafka connection addresses of the consumers to those of the new Kafka instance.
- **Step 3** Restart the consumption service so that consumers can consume messages from the new Kafka instance.
- **Step 4** Check whether consumers consume messages properly from the new Kafka instance.
- **Step 5** Change the Kafka connection address of the producer to that of the new Kafka instance.
- **Step 6** Restart the producer client to migrate the production service to the new Kafka instance.

- **Step 7** After the production service is migrated, check whether the consumption service connected to the new Kafka instance is normal.
- **Step 8** The migration is complete.

----End

Migration Scheme 4: Migrating Data Using Smart Connect First, then Consumption, and then Production

Use Smart Connect to synchronize the two Kafka instances, migrate the consumer first and then the producer to the new Kafka instance.

This scheme uses Smart Connect to synchronize the source and target data in real time. However, the consumption progress is synchronized in batches. The consumption progress on the source and target partition may vary from 0 to 100. As a result, some messages are repeatedly consumed. This scheme applies to services where the message production must continue, end-to-end latency must be low, and repeated consumption can be tolerated.

- **Step 1** Create a Smart Connect task for Kafka data replication. For details, see **Replicating Kafka Instance Data**.
- **Step 2** On the **Instance** > **Message Query** page of the Kafka console, check whether the latest messages and the synchronization progress of both Kafka instances are consistent. For details, see **Viewing Kafka Messages**.
 - Yes: Go to Step 3.
 - No: Check whether the synchronized data per minute of both Kafka instances is normal. If yes, wait for the synchronization progress of both Kafka instances to be consistent, then go to Step 3.
- **Step 3** Change the Kafka connection addresses of the consumers to those of the new Kafka instance.
- **Step 4** Restart the consumption service so that consumers can consume messages from the new Kafka instance.
- **Step 5** Check whether consumers consume messages properly from the new Kafka instance.
- **Step 6** Change the Kafka connection address of the producer to that of the new Kafka instance.
- **Step 7** Restart the producer client to migrate the production service to the new Kafka instance.
- **Step 8** After the production service is migrated, check whether the consumption service connected to the new Kafka instance is normal.
- **Step 9** The migration is complete.

----End

How Do I Migrate Persisted Data Along with Services?

You can migrate consumed data from the original instance to a new instance by using Smart Connect or open-source tool MirrorMaker. This tool mirrors the

original Kafka producer and consumer into new ones and migrates data to the new Kafka instance. For details, see **Replicating Kafka Instance Data** or **Using MirrorMaker to Synchronize Data Across Clusters**.

11.2 Using MirrorMaker to Synchronize Data Across Clusters

Overview

Scenario

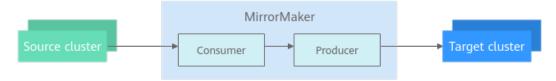
In the following scenarios, MirrorMaker can be used to synchronize data between different Kafka clusters to ensure the availability and reliability of the clusters:

- Backup and disaster recovery: An enterprise has multiple data centers. To
 prevent service unavailability caused by a fault in one data center, cluster data
 is synchronously backed up in multiple data centers.
- Cluster migration: As enterprises migrate services to the cloud, data in onpremises clusters must be synchronized with that in cloud clusters to ensure service continuity.

Principle

MirrorMaker can be used to mirror data from the source cluster to the target cluster. As shown in **Figure 11-1**, in essence, MirrorMaker first consumes data from the source cluster and then produces the consumed data to the target cluster. For more information about MirrorMaker, see **Mirroring data between clusters**.

Figure 11-1 MirrorMaker



Restrictions

- The IP addresses and port numbers of the nodes in the source cluster cannot be the same as those of the nodes in the target cluster. Otherwise, data will be replicated infinitely in a topic.
- Use MirrorMaker to synchronize data between at least two clusters. If there is only one cluster, data will be replicated infinitely in a topic.

Procedure

- **Step 1** Buy an ECS that can communicate with the source and target clusters. For details, see **Purchasing an ECS**.
- **Step 2** Log in to the ECS, install JDK, and add the following contents to .bash_profile in the home directory to configure the environment variables JAVA_HOME and

PATH. In this command, /opt/java/jdk1.8.0_151 is the JDK installation path. Change it to the path where you install JDK.

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

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

□ NOTE

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

Step 3 Download the binary software package of Kafka 3.3.1.

wget https://archive.apache.org/dist/kafka/3.3.1/kafka_2.12-3.3.1.tgz

Step 4 Decompress the binary software package.

tar -zxvf kafka_2.12-3.3.1.tgz

Step 5 Go to the binary software package directory and specify the IP addresses and ports of the source and target clusters and other parameters in the **connect-mirror-maker.properties** configuration file in the **config** directory.

```
# Specify two clusters.
clusters = A, B
A.bootstrap.servers = A_host1:A_port, A_host2:A_port, A_host3:A_port
B.bootstrap.servers = B_host1:B_port, B_host2:B_port, B_host3:B_port
# Specify the data synchronization direction. The data can be synchronized unidirectionally or bidirectionally.
A->B.enabled = true
# Specify the topics to be synchronized. Regular expressions are supported. By default, all topics are
replicated, for example, foo-.*.
A \rightarrow B.topics = .*
# If the following two configurations are enabled, clusters A and B replicate data with each other.
#B->A.enabled = true
#B->A.topics = .*
# Specify the number of replicas. If multiple topics need to be synchronized and their replica quantities are
different, create topics with the same name and replica quantity before starting MirrorMaker.
replication.factor=3
# Specify the consumer offset synchronization direction (unidirectionally or bidirectionally).
A->B.sync.group.offsets.enabled=true
# The replication factor for mm2 internal topics "heartbeats", "B.checkpoints.internal" and
# "mm2-offset-syncs.B.internal"
# In the test environment, the value can be 1. In the production environment, it is recommended that the
value be greater than 1, for example, 3.
checkpoints.topic.replication.factor=3
heartbeats.topic.replication.factor=3
offset-syncs.topic.replication.factor=3
# The replication factor for connect internal topics "mm2-configs.B.internal", "mm2-offsets.B.internal" and
# "mm2-status.B.internal"
# In the test environment, the value can be 1. In the production environment, it is recommended that the
value be greater than 1, for example, 3.
offset.storage.replication.factor=3
status.storage.replication.factor=3
config.storage.replication.factor=3
# customize as needed
# replication.policy.separator = _
# sync.topic.acls.enabled = false
# emit.heartbeats.interval.seconds = 5
# Match the topic name of the target cluster to that of the source.
# replication.policy.class = org.apache.kafka.connect.mirror.IdentityReplicationPolicy
```

- **Step 6** In the binary software package directory, start MirrorMaker to synchronize data. ./bin/connect-mirror-maker.sh config/connect-mirror-maker.properties
- **Step 7** (Optional) If a topic is created in the source cluster after MirrorMaker has been started, and the topic data needs to be synchronized, restart MirrorMaker. For details about how to restart MirrorMaker, see **Step 6**. You can also add configurations listed in **Table 11-3** to periodically synchronize new topics without restarting MirrorMaker. **refresh.topics.interval.seconds** is mandatory. Other parameters are optional.

Table 11-3 MirrorMaker configurations

Parameter	Default Value	Description
sync.topic.configs.enabled	true	Whether to monitor the source cluster for configuration changes.
sync.topic.acls.enabled	true	Whether to monitor the source cluster for ACL changes.
emit.heartbeats.enabled	true	Whether to let the connector send heartbeats periodically.
emit.heartbeats.interval.s econds	5 seconds	Heartbeat frequency.
emit.checkpoints.enabled	true	Whether to let the connector periodically send the consumer offset information.
emit.checkpoints.interval.s econds	5 seconds	Checkpoint frequency.
refresh.topics.enabled	true	Whether to let the connector periodically check for new topics.
refresh.topics.interval.seco nds	5 seconds	Frequency of checking for new topics in the source cluster.
refresh.groups.enabled	true	Whether to let the connector periodically check for new consumer groups.
refresh.groups.interval.sec onds	5 seconds	Frequency of checking for new consumer groups in the source cluster.
replication.policy.class	org.apache.kafka.c onnect.mirror.Defa ultReplicationPoli- cy	Use LegacyReplicationPolicy to imitate MirrorMaker of an earlier version.
heartbeats.topic.retention. ms	One day	Used when heartbeat topics are created for the first time.

Parameter	Default Value	Description
checkpoints.topic.retentio n.ms	One day	Used when checkpoint topics are created for the first time.
offset.syncs.topic.retentio n.ms	max long	Used when offset sync topics are created for the first time.

----End

Verifying Data Synchronization

Step 1 View the topic list in the target cluster to check whether there are source topics.

The default value of replication.policy.class is org.apache.kafka.connect.mirror.DefaultReplicationPolicy, which adds a prefix (for example, "A.") to target topic names to differentiate them from the source. This setting helps MirrorMaker avoid circular replication. To mirror the target and source topic names, set replication.policy.class to org.apache.kafka.connect.mirror.IdentityReplicationPolicy.

Step 2 Produce and consume messages in the source cluster, view the consumption progress in the target cluster, and check whether data has been synchronized from the source cluster to the target cluster.

If the target cluster is a Huawei Cloud Kafka instance, view the consumption progress on the **Instance** > **Consumer Groups** page.

----End

11.3 Migrating Data Using Smart Connect

11.3.1 Enabling Smart Connect

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

Procedure for using Smart Connect:

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

This section describes how to enable Smart Connect.

Notes and Constraints

- Enabling Smart Connect incurs additional broker fees.
 - For example, if you enable Smart Connect for a kafka.4u8g.cluster instance, at least two more kafka.4u8g brokers will be created for Smart Connect and you need to pay for them.
- Unavailable for single-node instances.

Prerequisites

- A Kafka instance has been created and is in the Running state.
- **auto.create.groups.enable** is set to **true**. If no, modify it by referring to **Modifying Kafka Instance Configuration Parameters**.

Procedure

- **Step 1** Log in to the **Kafka console**.
- **Step 2** Click on the upper left corner to select the region where your instance is located.
- **Step 3** Enable Smart Connect using one of the following methods:
 - In the row containing the desired Kafka instance, choose More > Enable Smart Connect.
 - Click the desired Kafka instance to go to the instance details page. Choose
 Enable Smart Connect in the upper right corner.
 - Click the desired Kafka instance to go to the instance details page. Click next to Smart Connect.
 - Click the desired Kafka instance to go to the instance details page. In the navigation pane, choose **Smart Connect**. Click **Enable Smart Connect**.
- **Step 4** Click , enable **Smart Connect**, set 2–16 brokers as required, and click **Next**.



By default, two brokers will be used. If synchronization traffic between two Kafka instances is estimated to be large, for example, greater than 50 MB/s, use more brokers.

Step 5 On the displayed **Enabling Smart Connect for Kafka Instance** page, ensure that **Smart Connect** is enabled and click **Submit**.

Smart Connect has been successfully enabled if the task is in the **Successful** state on the **Current Tasks** tab page of the **Instance** > **Background Tasks** page.

----End

Follow-up Operations

Proceed to Replicating Kafka Instance Data, Dumping Kafka Data to Object Storage Service (OBS), to synchronize data between DMS for Kafka and other cloud services.

Related Document

To enable Smart Connect by calling an API, see **Enabling Smart Connect (Payper-Use Instance)**.

11.3.2 Replicating Kafka Instance Data

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

Data Synchronization Rules

A Smart Connect task synchronizes the **configuration information** and **message data** of a topic. **Table 11-4** describes the rules.

Table 11-4 Synchronization rules

Sync Type	Sync Rule
Configuration information	Initialization synchronization and scheduled synchronization are available.
	 Initialization synchronization: When a Smart Connect task synchronizes configuration information to a target topic for the first time, the system determines whether to create the topic. If yes, the source topic configuration will be synchronized to the new topic. If no, the initialization synchronization of configuration information will not be triggered. Table 11-5 describes initialization synchronization parameters.
	Specify whether to synchronize the consumer offset when creating a Smart Connect task. This setting is fixed once the task is created.
	 Scheduled synchronization: After a Smart Connect task is started, the configuration information is synchronized to the target topic every minute. Table 11-5 describes scheduled synchronization parameters.
Message data	Message data is synchronized in real time .

Table 11-5 Topic parameters that can be synchronized

In Initialization	As Scheduled
Partitions	Partitions CAUTION The number of partitions can only be increased. If the number of partitions of the target topic is greater than that of the source topic, the number of partitions will not be synchronized.
Aging Time (h)	-
Synchronous Flushing	-

In Initialization	As Scheduled
Max. Message Size	-
Replicas CAUTION The replica quantity of the target topic is specified in Smart Connect task creation instead of that of the source topic.	-

Notes and Constraints

- This function is unavailable for single-node Kafka instances.
- A maximum of 18 Smart Connect tasks can be created for an instance.
- When you copy Kafka data, the two Kafka instances must be connected through the intranet. If they are in different VPCs, connect the network by referring to Accessing Kafka Using a VPC Endpoint Across VPCs or VPC Peering Connection.
- After a Smart Connect task is created, task parameters cannot be modified.
- If you have enabled Smart Connect for an instance before July 1, 2022 and Kafka data replication is not available, **disable Smart Connect** and then enable it again.
- Data can be synchronized only when Max. Message Size of the target topic is greater than or equal to 524,288 bytes. If no topic is available in the target Kafka instance, a topic will be automatically created with the same Max. Message Size as that of the source Kafka instance topic during data synchronization. In this case, ensure the Max. Message Size to be used is greater than or equal to 524,288 bytes. To modify Max. Message Size, see Modifying Kafka Topic Configurations.
- Smart Connect tasks rely on the running environments at the source and target. The service status, network connectivity, access permission, and configuration information are involved. When the environments are abnormal, Smart Connect tasks may not synchronize data properly.
- When a Smart Connect task is interrupted, generally it can be resumed to prevent data loss. Exceptionally, if source messages are aged during the synchronization interruption, data will be lost.

Prerequisites

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

Procedure

- **Step 1** Log in to the **Kafka console**.
- **Step 2** Click in the upper left corner to select the region where your instance is located.

- **Step 3** Click the desired instance to go to the instance details page.
- **Step 4** In the navigation pane, choose **Smart Connect**.
- **Step 5** On the displayed page, click **Create Task**.
- **Step 6** For **Task Name**, enter a unique Smart Connect task name. Naming rules: 4–64 characters and only letters, digits, hyphens (-), or underscores (_).
- Step 7 For Task Type, select Copy Kafka data.
- **Step 8** For **Start Immediately**, specify whether to execute the task immediately after the task is created. By default, the task is executed immediately. If you disable this option, you can enable it later in the task list.
- **Step 9** In the **Current Kafka** area, set the instance alias. Naming rules: 1–20 characters and only letters, digits, hyphens (-), or underscores (_).

The instance alias is used in the following scenarios:

- If you enable Rename Topics and select Push or Both for Sync Direction, the
 alias of the current Kafka instance will be added to the topic names of the
 peer end Kafka instance. For example, if the alias of the current Kafka
 instance is A and the topic name of the peer end Kafka instance is test, the
 renamed topic will be A.test.
- After a Smart Connect task of Kafka data replication is created, a topic named mm2-offset-syncs.peer end Kafka instance alias.internal is generated for the current Kafka instance. If the task has Sync Consumer Offset enabled and uses Pull or Both for Sync Direction, a topic named peer end Kafka instance alias.checkpoints.internal is also created for the current Kafka instance. The two topics are used to store internal data. If they are deleted, data replication will fail.
- **Step 10** In the **Peer Kafka** area, configure the following parameters.

Table 11-6 Peer Kafka parameters

Parameter	Description
Instance Alias	Naming rules: 1–20 characters and only letters, digits, hyphens (-), or underscores (_).
	The instance alias is used in the following scenarios:
	• If you enable Rename Topics and select Pull or Both for Sync Direction , the alias of the peer end Kafka instance will be added to the topic names of the current Kafka instance. For example, if the alias of the peer end Kafka instance is B and the topic name of the current Kafka instance is test01 , the renamed topic will be B.test01 .
	• After a Smart Connect task of Kafka data replication is created, if the task has Sync Consumer Offset enabled and uses Push or Both for Sync Direction, a topic named current Kafka instance alias.checkpoints.internal is also created for the peer end Kafka instance. This topic is used to store internal data. If it is deleted, data replication will fail.
Config Type	Options:
	Kafka address: Enter Kafka instance addresses. To replicate data to a target Kafka instance in another VPC, use this type.
	• Instance name: Select an existing Kafka instance. To replicate data to a target Kafka instance in the same VPC, use this type.
Instance Name	Mandatory when Instance name is used for Config Type and the Kafka instances are within a VPC.
	Select an existing Kafka instance from the drop-down list.
Kafka Address	Set this parameter when Config Type is set to Kafka address .
	Enter the IP addresses and port numbers for connecting to the Kafka instance.
	When you copy Kafka data, the two Kafka instances must be connected through the intranet. If they are in different VPCs, connect the network by referring to Accessing Kafka Using a VPC Endpoint Across VPCs or VPC Peering Connection.

Parameter	Description
Authentication	Options:
	 SASL_SSL: The Kafka instance has enabled SASL_SSL, clients can connect to it with SASL and the data will be encrypted using the SSL certificate.
	SASL_PLAINTEXT: The Kafka instance has enabled SASL_PLAINTEXT, clients can connect to it with SASL and the data will be transmitted in plaintext.
	PLAINTEXT: The instance is not using authentication.
Authentication Mechanism	Set this parameter when Authentication is set to SASL_SSL/SASL_PLAINTEXT .
	SCRAM-SHA-512: uses the hash algorithm to generate credentials for usernames and passwords to verify identities. SCRAM-SHA-512 is more secure than PLAIN.
	PLAIN: a simple username and password verification mechanism.
Username	Set this parameter when Authentication is set to SASL_SSL/SASL_PLAINTEXT .
	Set in instance creation or user creation.
Password	Set this parameter when Authentication is set to SASL_SSL/SASL_PLAINTEXT .
	Set in instance creation or user creation.

After a Smart Connect task is created, modifying the **authentication method or mechanism, or password** of the peer end instance causes the synchronization task to **fail**. In this case, delete the current Smart Connect task and create another one

Step 11 In the **Rules** area, configure the following parameters.

Table 11-7 Parameters for configuring data replication rules

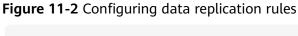
Parameter	Description
Sync Direction	 There are three synchronization directions: Pull: Replicates data from the peer Kafka instance to the current Kafka instance.
	Push: Replicates data from the current Kafka instance to the peer Kafka instance.
	Both: Bidirectional replication of Kafka instance data on both ends.

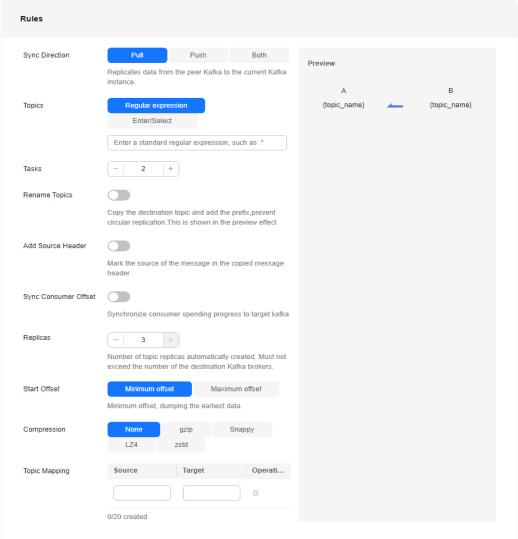
Parameter	Description
Topics	Specify the topics whose data is to be replicated.
	Regular expression: Use a regular expression to match topics.
	Enter/Select: Enter topic names. To enter multiple topic names, press Enter after entering each topic name. You can also select topics from the dropdown list. A maximum of 20 topics can be entered or selected.
	Data of topics whose names end with "internal" (for example, topic.internal) will not be synchronized.
Tasks	Number of data replication tasks. The default value is 2 . You are advised to use the default value.
	If Sync Direction is set to Both , the actual number of tasks will be twice the number of tasks you configure here.
Rename Topics	Add the alias of the source Kafka instance before the target topic name to form a new name of the target topic. For example, if the alias of the source instance is A and the target topic name is test , the renamed target topic will be A.test .
	If you select Both for Sync Direction , enable Rename Topics to prevent infinite replication.
Add Source Header	The target topic receives the replicated messages. The message header contains the message source.
	If you select Both for Sync Direction , Add Source Header is enabled by default to prevent infinite replication.

Parameter	Description
Sync Consumer Offset	Enable this option to synchronize the consumer offset to the target Kafka instance.
	After enabling Sync Consumer Offset , pay attention to the following:
	The source and target Kafka instances cannot consume messages at the same time. Otherwise, the synchronized consumer offset will be abnormal.
	The consumer offset is synchronized every minute. As a result, the consumer offset on the target end may be slightly smaller than that on the source end, and some messages are repeatedly consumed. The service logic of the consumer client must be able to handle repeated consumption.
	The offset synchronized from the source end is not the same as the offset on the target end. Instead, there is a mapping relationship. If the consumer offset is maintained by the consumer client, the consumer client does not obtain the consumer offset from the target Kafka instance after switching consumption from the source Kafka instance to the target Kafka instance. As a result, the offset may be incorrect or the consumer offset may be reset.
Replicas	Number of topic replicas when a topic is automatically created in the peer instance. The value of this parameter cannot exceed the number of brokers in the peer instance.
	This parameter takes precedence over the default.replication.factor parameter set in the peer instance.
Start Offset	Options:
	Minimum offset: dumping the earliest data
	Maximum offset: dumping the latest data
Compression	Compression algorithm to use for copying messages.
Topic Mapping	Customize the target topic name.
	Maximum mappings: 20. Rename Topic and Topic Mapping cannot be configured at the same time.

Precautions:

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





- **Step 12** (Optional) In the lower right corner of the page, click **Check** to test the connectivity between the Kafka instances.
 - If "Connectivity check passed." is displayed, the Kafka instances are connected.
- **Step 13** Click **Create**. The Smart Connect task list page is displayed. The message "Task *xxx* was created successfully." is displayed in the upper right corner of the page.
 - After a Smart Connect task of Kafka data replication is created, Kafka automatically creates the following topics:
 - A topic named mm2-offset-syncs.peer end Kafka instance alias.internal is generated for the current Kafka instance. If the task has Sync Consumer Offset enabled and uses Pull or Both for Sync Direction, a topic named peer end Kafka instance alias.checkpoints.internal is also created for the current

Kafka instance. The two topics are used to store internal data. If they are deleted, data replication will fail.

 After a Smart Connect task of Kafka data replication is created, if the task has Sync Consumer Offset enabled and uses Push or Both for Sync Direction, a topic named current Kafka instance alias.checkpoints.internal is also created for the peer end Kafka instance. This topic is used to store internal data. If it is deleted, data replication will fail.

----End

Related Document

To configure data replication between Kafka instances by calling an API, **Creating a Smart Connect Task**.

11.3.3 Dumping Kafka Data to Object Storage Service (OBS)

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

Data in the source Kafka instance is synchronized to the dumping file in real time.

Notes and Constraints

- This function is unavailable for single-node instances.
- A maximum of 18 Smart Connect tasks can be created for an instance.
- After a Smart Connect task is created, task parameters cannot be modified.

Prerequisites

- You have enabled Smart Connect.
- A Kafka instance has been created and is in the **Running** state.
- The OBS bucket must be created in the same region as the Kafka instance.

Procedure

- **Step 1** Log in to the **Kafka console**.
- **Step 2** Click in the upper left corner to select the region where your instance is located.
- **Step 3** Click the desired instance to go to the instance details page.
- **Step 4** In the navigation pane, choose **Smart Connect**.
- **Step 5** On the displayed page, click **Create Task**.
- **Step 6** For **Task Name**, enter a unique Smart Connect task name. Naming rules: 4–64 characters and only letters, digits, hyphens (-), or underscores (_).
- Step 7 For Task Type, select Dumping.
- **Step 8** For **Start Immediately**, specify whether to execute the task immediately after the task is created. By default, the task is executed immediately. If you disable this option, you can enable it later in the task list.

- **Step 9** In the **Source** area, retain the default setting.
- **Step 10** In the **Topics** area, set parameters based on the following table.

Table 11-8 Topic parameters

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

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

Table 11-9 Target parameters

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

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

□ NOTE

- Do not use the key of a message as the dumping file name.
- OBS file directory format: *bucket-name*/ *file-directory*/ *topic-name*/ *time*. For example, obs-kafka/smartconnect/topic01/2025/01/01.
- OBS file name format: *topic-name+partition No.+start offset*, for example, **topic01+0+0000000004**.
- OBS file generation rule: One file is generated for each specified data dump period.
- **Step 12** Click **Create**. The Smart Connect task list page is displayed. The message "Task *xxx* was created successfully." is displayed in the upper right corner of the page.

----End

Related Document

To dump Kafka data to OBS by calling an API, see Creating a Smart Connect Task.

11.3.4 Managing Smart Connect Tasks

View, delete, start, pause, or restart a Smart Connect task.

Notes and Constraints

Unavailable for single-node instances.

Prerequisite

A Smart Connect task has been created.

Viewing Smart Connect Tasks

- **Step 1** Log in to the **Kafka console**.
- **Step 2** Click in the upper left corner to select the region where your instance is located.
- **Step 3** Click the desired instance to go to the instance details page.

- **Step 4** In the navigation pane, choose **Smart Connect**.
- **Step 5** Click a Smart Connect task name to go to the details page.
- **Step 6** View the basic information, source, and target of the Smart Connect task.

□ NOTE

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

----End

Deleting a Smart Connect Task

- Step 1 Log in to the Kafka console.
- **Step 2** Click in the upper left corner to select the region where your instance is located.
- **Step 3** Click the desired instance to go to the instance details page.
- **Step 4** In the navigation pane, choose **Smart Connect**.
- **Step 5** In the row containing the Smart Connect task to be deleted, click **Delete**.
- Step 6 Click OK.

The task is deleted when it is no longer displayed in the Smart Connect task list.

----End

Starting or Pausing a Smart Connect Task

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

- **Step 1** Log in to the **Kafka console**.
- **Step 2** Click on the upper left corner to select the region where your instance is located.
- **Step 3** Click the desired instance to go to the instance details page.
- **Step 4** In the navigation pane, choose **Smart Connect**.
- **Step 5** Perform the required operation:
 - To start a Smart Connect task, click **Start** in the row that contains the task.
 - To pause a Smart Connect task, click **Pause** in the row containing the task, then click **OK** in the dialog box that is displayed.

The task is started or paused when it is in the **Running** or **Paused** state on the Smart Connect tasks page.

----End

Restarting a Smart Connect Task

- Step 1 Log in to the Kafka console.
- **Step 2** Click in the upper left corner to select the region where your instance is located.
- **Step 3** Click the desired instance to go to the instance details page.
- **Step 4** In the navigation pane, choose **Smart Connect**.
- **Step 5** In the row containing the desired Smart Connect task, click **Restart**.

Precautions:

- Modifying the source or target parameters after a Smart Connect task is created may cause the restart to fail.
- Restarting a Smart Connect task resets the synchronization progress and the synchronization task will be restarted.

Step 6 Click OK.

Once the task is restarted, a success message is displayed in the upper area of the page.

----End

Related Documents

- To view Smart Connect task details by calling an API, see Querying Smart Connect Task Details.
- To delete a Smart Connect task by calling an API, see Deleting a Smart Connect Task.
- To pause a Smart Connect task by calling an API, see Pausing a Smart Connect Task.

11.3.5 Disabling Smart Connect

Disable Smart Connect and resources can be freed.

Disabling Smart Connect does not affect services.

Notes and Constraints

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

Prerequisites

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

Procedure

- Step 1 Log in to the Kafka console.
- **Step 2** Click on the upper left corner to select the region where your instance is located.
- **Step 3** Disable Smart Connect using either of the following methods:
 - In the row containing the desired Kafka instance, choose More > Disable Smart Connect.
 - Click the desired Kafka instance to go to the instance details page. Choose -- Disable Smart Connect in the upper right corner.
- **Step 4** Click to disable Smart Connect. Then click **Next**.
- **Step 5** Ensure that **Smart Connect** is disabled and click **Submit**.

Smart Connect has been successfully disabled if the task is in the **Successful** state on the **Current Tasks** tab page of the **Instance** > **Background Tasks** page.

----End

Related Document

To disable Smart Connect by calling an API, see **Disabling Smart Connect (Payper-Use Instance)**.

12 Testing Instance Performance

12.1 Kafka Production Rate and CPU Usage

This section describes performance tests on Distributed Message Service (DMS) for Kafka. The performance is measured by the message production rate on the client side and CPU usage on the server side. The tests cover the following scenarios:

- Scenario 1 (batch size): same Kafka instance, same topics, different message size settings
- Scenario 2 (cross-AZ or intra-AZ production): same Kafka instance, same topics, different AZ settings for the client and server
- Scenario 3 (number of replicas): same Kafka instance, different numbers of replicas
- Scenario 4 (synchronous or asynchronous replication): same Kafka instance, topics with different replication settings

Table 12-1 Test parameters

Partitio ns	Replicas	Synchronous Replication	batch.size	Cross-AZ Production
3	1	No	1 KB	No
3	1	No	16 KB	No
3	1	No	1 KB	Yes
3	3	Yes	1 KB	No
3	3	No	1 KB	No

Environment

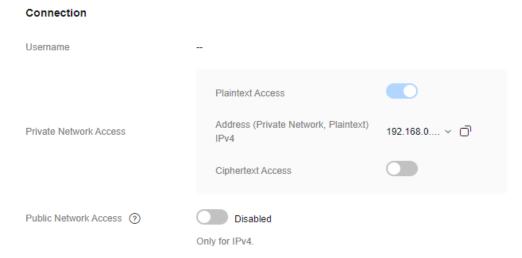
Perform the following steps to set up the test environment.

 Purchase a Kafka instance with parameters specified as follows and retain the default settings for other ones. For details about how to purchase one, see Buying a Kafka Instance.

Table 12-2

Parameter	Example Value
Region	CN-Hong Kong
AZ	AZ1
Version	2.7
Architecture	Cluster
Broker Flavor	kafka.2u4g.cluster
Brokers	3
Storage Space per Broker	Ultra-high I/O, 200 GB
VPC	Select a VPC.
Subnet	Select a subnet.
Security Group	Select a security group.
Access Mode	Default value
Instance Name	kafka-test
Enterprise Project	default

After the purchase, obtain **Address (Private Network, Plaintext)** on the instance details page.



 Create three topics with parameters specified as follows for the purchased Kafka instance. For details, see Creating a Kafka Topic.

- Topic-01: 3 partitions, 1 replica, asynchronous replication
- Topic-02: 3 partitions, 3 replicas, asynchronous replication
- Topic-03: 3 partitions, 3 replicas, synchronous replication
- 3. Obtain the test tool.

Obtain Kafka CLI 2.7.2.

4. Purchase a server for the client.

Buy two ECSs with the following configurations. For details about how to purchase an ECS, see **Purchasing an ECS**.

- One ECS is 4 vCPUs | 8 GB, runs Linux, and is configured with the same region, AZ, VPC, subnet, and security group as the Kafka instance.
- The other ECS is 4 vCPUs | 8 GB, runs Linux, and is configured with the same region, VPC, subnet, and security group but a different AZ from the Kafka instance.

Perform the following operations on the ECSs:

 Install Java JDK and configure the environment variables JAVA_HOME and PATH.

export JAVA_HOME=/root/jdk1.8.0_231 export PATH=\$JAVA_HOME/bin:\$PATH

 Download Kafka CLI 2.7.2 and decompress it. tar -zxf kafka_2.12-2.7.2.tgz

Script

./kafka-producer-perf-test.sh --producer-props bootstrap.servers={Connection address} acks=1 batch.size={batch.size} linger.ms=0 --topic {Topic name} --num-records {num-records} --record-size 1024 --throughput 102400

Table 12-3 Script parameters

Parameter	Description
bootstrap.servers	Address of the Kafka instance obtained in 1.
acks	Message synchronization policy. acks=1 indicates asynchronous replication, and acks=-1 indicates synchronous replication.
batch.size	Size of messages sent in each batch, in bytes.
linger.ms	Interval between two batches.
topic	Topic name set in 2.
num-records	Total number of messages to be sent.
record-size	Size of each message.
throughput	Number of messages sent per second.

Procedure

Scenario 1: Varied Batch Sizes

1. Log in to the client server, go to the **kafka_2.12-2.7.2/bin** directory, and run the following scripts.

Set **batch.size** to 1 KB, and run the following script:

./kafka-producer-perf-test.sh --producer-props bootstrap.servers=192.168.0.69:9092,192.168.0.42:9092,192.168.0.66:9092 acks=1 batch.size=1024 linger.ms=0 --topic Topic-01 --num-records 8000000 --record-size 1024 --throughput 102400

Result:

8000000 records sent, 34128.673632 records/sec (33.33 MB/sec), 879.91 ms avg latency, 4102.00 ms max latency, 697 ms 50th, 2524 ms 95th, 2888 ms 99th, 4012 ms 99.9th.

Message production rate: 34,128 records/second

Set batch.size to 16 KB, and run the following script:

/kafka-producer-perf-test.sh --producer-props bootstrap.servers=192.168.0.69:9092,192.168.0.42:9092,192.168.0.66:9092 acks=1 batch.size=16384 linger.ms=0 --topic Topic-01 --num-records 100000000 --record-size 1024 --throughput 102400

Result:

100000000 records sent, 102399.318430 records/sec (100.00 MB/sec), 4.72 ms avg latency, 914.00 ms max latency, 1 ms 50th, 5 ms 95th, 162 ms 99th, 398 ms 99.9th.

Message production rate: 102,399 records/second

- 2. Log in to the **Kafka console**.
- 3. Click the name of the test instance to go to the instance details page.
- 4. Choose **Monitoring** > **Details** in the navigation pane.
- 5. On the **By Broker** tab page, view the CPU usage of the server node.



Figure 12-1 broker-0 CPU usage (batch.size = 1 KB)

CPU usage: 58.10%

CPU Usage ②

Max Min
24.10 0

30

20

10

19:30 19:37 19:43 19:50 19:57 20:03

Figure 12-2 broker-0 CPU usage (batch.size = 16 KB)

CPU usage: 24.10%

Figure 12-3 broker-1 CPU usage (batch.size = 1 KB)



CPU usage: 56.70%

Figure 12-4 broker-1 CPU usage (batch.size = 16 KB)



CPU usage: 25%



Figure 12-5 broker-2 CPU usage (batch.size = 1 KB)

CPU usage: 53.30%

Figure 12-6 broker-2 CPU usage (batch.size = 16 KB)



CPU usage: 23.30%

Scenario 2: Cross-AZ or Intra-AZ Production

 Log in to the client server, go to the kafka_2.12-2.7.2/bin directory, and run the following scripts.

Configure the same AZ for the client and the instance, and run the following script:

./kafka-producer-perf-test.sh --producer-props bootstrap.servers=192.168.0.69:9092,192.168.0.42:9092,192.168.0.66:9092 acks=1 batch.size=1024 linger.ms=0 --topic Topic-01 --num-records 8000000 --record-size 1024 --throughput 102400

Result:

8000000 records sent, 34128.673632 records/sec (33.33 MB/sec), 879.91 ms avg latency, 4102.00 ms max latency, 697 ms 50th, 2524 ms 95th, 2888 ms 99th, 4012 ms 99.9th.

Message production rate: 34,128 records/second

Configure different AZs for the client and the instance, and run the following script:

/kafka-producer-perf-test.sh --producer-props bootstrap.servers=192.168.0.69:9092,192.168.0.42:9092,192.168.0.66:9092 acks=1 batch.size=1024 linger.ms=0 --topic Topic-01 --num-records 4000000 --record-size 1024 --throughput 102400

Result:

4000000 records sent, 8523.042044 records/sec (8.32 MB/sec), 3506.20 ms avg latency, 11883.00 ms max latency, 1817 ms 50th, 10621 ms 95th, 11177 ms 99th, 11860 ms 99.9th.

Message production rate: 8523 records/second

- 2. Log in to the **Kafka console**.
- 3. Click the name of the test instance to go to the instance details page.
- 4. Choose **Monitoring** > **Details** in the navigation pane.
- 5. On the **By Broker** tab page, view the CPU usage of the server node.

Figure 12-7 broker-0 CPU usage (same AZ)



CPU usage: 58.10%

Figure 12-8 broker-0 CPU usage (different AZs)



CPU usage: 17.20%

CPU Usage ②

Max Min
56.70 0

40

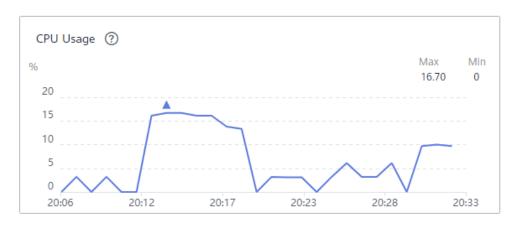
20

17:10 17:22 17:34 17:46 17:58 18:10

Figure 12-9 broker-1 CPU usage (same AZ)

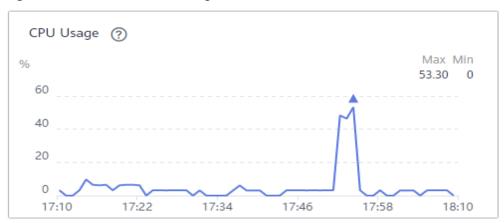
CPU usage: 56.70%

Figure 12-10 broker-1 CPU usage (different AZs)



CPU usage: 16.70%

Figure 12-11 broker-2 CPU usage (same AZ)



CPU usage: 53.30%



Figure 12-12 broker-2 CPU usage (different AZs)

CPU usage: 18.80%

Scenario 3: Varied Numbers of Replicas

 Log in to the client server, go to the kafka_2.12-2.7.2/bin directory, and run the following scripts.

For the **one-replica** topic, run the following script:

/kafka-producer-perf-test.sh --producer-props bootstrap.servers=192.168.0.69:9092,192.168.0.42:9092,192.168.0.66:9092 acks=1 batch.size=1024 linger.ms=0 --topic Topic-01 --num-records 8000000 --record-size 1024 --throughput 102400

Result:

8000000 records sent, 34128.673632 records/sec (33.33 MB/sec), 879.91 ms avg latency, 4102.00 ms max latency, 697 ms 50th, 2524 ms 95th, 2888 ms 99th, 4012 ms 99.9th.

Message production rate: 34,128 records/second

For the **three-replica** topic, run the following script:

./kafka-producer-perf-test.sh --producer-props bootstrap.servers=192.168.0.69:9092,192.168.0.42:9092,192.168.0.66:9092 acks=1 batch.size=1024 linger.ms=0 --topic Topic-02 --num-records 4000000 --record-size 1024 --throughput 102400

Result:

4000000 records sent, 14468.325219 records/sec (14.13 MB/sec), 2069.99 ms avg latency, 7911.00 ms max latency, 846 ms 50th, 6190 ms 95th, 6935 ms 99th, 7879 ms 99.9th.

Message production rate: 14,468 records/second

- 2. Log in to the **Kafka console**.
- 3. Click the name of the test instance to go to the instance details page.
- 4. Choose **Monitoring** > **Details** in the navigation pane.
- 5. On the **By Broker** tab page, view the CPU usage of the server node.

CPU Usage ②

Max Min
58.10 0

60

40

20

17:10 17:22 17:34 17:46 17:58 18:10

Figure 12-13 broker-0 CPU usage (one replica)

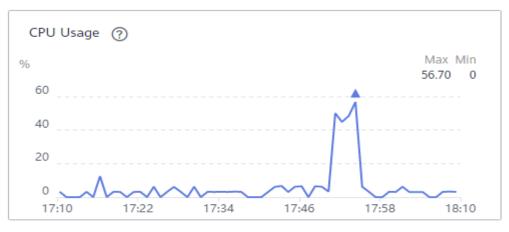
CPU usage: 58.10%

Figure 12-14 broker-0 CPU usage (three replicas)



CPU usage: 86.70%

Figure 12-15 broker-1 CPU usage (one replica)



CPU usage: 56.70%

CPU Usage ②

Max Min 80.60 0

90

60

30

0

09:30

09:36

09:41

09:47

09:53

09:58

Figure 12-16 broker-1 CPU usage (three replicas)

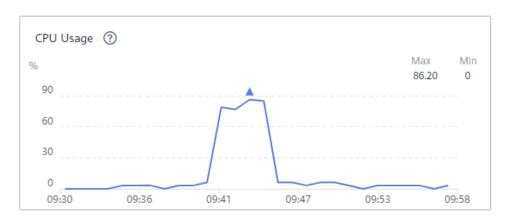
CPU usage: 80.60%

Figure 12-17 broker-2 CPU usage (one replica)



CPU usage: 53.30%

Figure 12-18 broker-2 CPU usage (three replicas)



CPU usage: 86.20%

Scenario 4: Synchronous/Asynchronous Replication

 Log in to the client server, go to the kafka_2.12-2.7.2/bin directory, and run the following scripts.

For **asynchronous replication**, run the following script:

./kafka-producer-perf-test.sh --producer-props bootstrap.servers=192.168.0.69:9092,192.168.0.42:9092,192.168.0.66:9092 acks=1 batch.size=1024 linger.ms=0 --topic Topic-02 --num-records 4000000 --record-size 1024 --throughput 102400

Result:

4000000 records sent, 14468.325219 records/sec (14.13 MB/sec), 2069.99 ms avg latency, 7911.00 ms max latency, 846 ms 50th, 6190 ms 95th, 6935 ms 99th, 7879 ms 99.9th.

Message production rate: 14,468 records/second

For **synchronous replication**, run the following script:

/kafka-producer-perf-test.sh --producer-props bootstrap.servers=192.168.0.69:9092,192.168.0.42:9092,192.168.0.66:9092 acks=-1 batch.size=1024 linger.ms=0 --topic Topic-03 --num-records 1000000 --record-size 1024 --throughput 102400

Result:

1000000 records sent, 3981.937930 records/sec (3.89 MB/sec), 7356.98 ms avg latency, 19013.00 ms max latency, 6423 ms 50th, 14381 ms 95th, 18460 ms 99th, 18975 ms 99.9th.

Message production rate: 3981 records/second

- 2. Log in to the **Kafka console**.
- 3. Click the name of the test instance to go to the instance details page.
- 4. Choose **Monitoring** > **Details** in the navigation pane.
- 5. On the **By Broker** tab page, view the CPU usage of the server node.

Figure 12-19 broker-0 CPU usage (asynchronous replication)



CPU usage: 86.70%

CPU Usage ②

Max Min
60 0

40

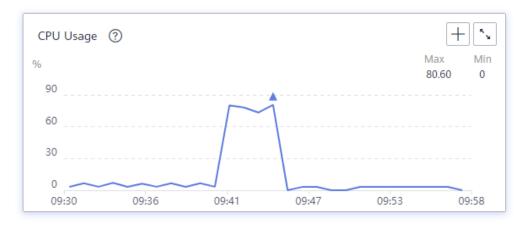
20

0
08:38 08:50 09:02 09:14 09:26 09:38

Figure 12-20 broker-0 CPU usage (synchronous replication)

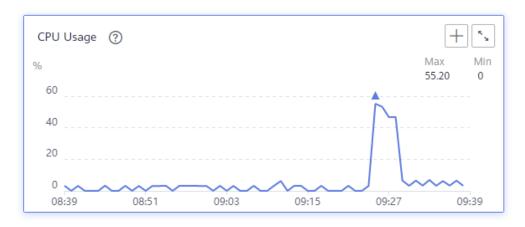
CPU usage: 60%

Figure 12-21 broker-1 CPU usage (asynchronous replication)



CPU usage: 80.60%

Figure 12-22 broker-1 CPU usage (synchronous replication)



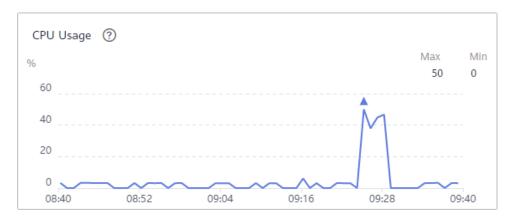
CPU usage: 55.20%

CPU Usage ? Min Max % 86.20 0 90 60 30 0 09:30 09:36 09:41 09:47 09:53 09:58

Figure 12-23 broker-2 CPU usage (asynchronous replication)

CPU usage: 86.20%

Figure 12-24 broker-2 CPU usage (synchronous replication)



CPU usage: 50%

Result

Table 12-4 Testing results

Par titi on s	Re plic as	Synchr onous Replic ation	batch. size	Cross- AZ Produ ction	Message Producti on Rate on the Client Side (Records /Second)	CPU Usage on the Server Side (broker- 0)	CPU Usage on the Server Side (broker -1)	CPU Usage on the Server Side (broke r-2)
3	1	No	1 KB	No	34,128	58.10%	56.70%	53.30%
3	1	No	16 KB	No	102,399	24.10%	25.00%	23.30%
3	1	No	1 KB	Yes	8,523	17.20%	16.70%	18.80%

Par titi on s	Re plic as	Synchr onous Replic ation	batch. size	Cross- AZ Produ ction	Message Producti on Rate on the Client Side (Records /Second)	CPU Usage on the Server Side (broker- 0)	CPU Usage on the Server Side (broker -1)	CPU Usage on the Server Side (broke r-2)
3	3	Yes	1 KB	No	3981	60.00%	55.20%	50.00%
3	3	No	1 KB	No	14,468	86.70%	80.60%	86.20%

Based on the test results, the following conclusions are drawn (for reference only):

- When the **batch.size** of production requests is 16 times larger, the message production rate increases, and the CPU usage decreases.
- Compared with cross-AZ production, intra-AZ production significantly increases message production rate and CPU usage.
- When the number of replicas changes from 1 to 3, the message production rate decreases significantly, and the CPU usage increases.
- Compared with synchronous replication, asynchronous replication increases the message production rate and the CPU usage.

12.2 Kafka Instance TPS

TPS tests can be performed in the following scenarios:

- Scenario 1 (whether SASL is enabled): same topic, different SASL settings
- Scenario 2 (synchronous or asynchronous replication): same instance, topics with different replication settings
- Scenario 3 (synchronous or asynchronous flushing): same instance, topics with different flushing settings
- Scenario 4 (disk type): same topic, instances with different disk types
- Scenario 5 (number of partitions): same instance, topics with different number of partitions

Environment

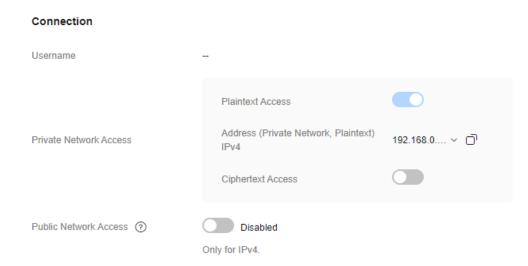
Perform the following steps to set up the test environment.

1. Purchase Kafka instances with parameters specified in **Table 12-5**. For more information, see **Buying a Kafka Instance**.

Table 12-5 Instance parameters

Instance Name	Brokers	Broker Flavor	SASL	Storage space per broker
kafka-01	3	kafka.2u4g.clust er	Yes	Ultra-high I/O
kafka-02	3	kafka.4u8g.clust er	Yes	Ultra-high I/O
kafka-03	3	kafka.8u16g.clu ster	Yes	Ultra-high I/O
kafka-04	3	kafka.12u24g.cl uster	Yes	Ultra-high I/O
kafka-05	3	kafka.16u32g.cl uster	Yes	Ultra-high I/O
kafka-06	3	kafka.2u4g.clust er	No	Ultra-high I/O
kafka-07	3	kafka.4u8g.clust er	No	Ultra-high I/O
kafka-08	3	kafka.8u16g.clu ster	No	Ultra-high I/O
kafka-09	3	kafka.12u24g.cl uster	No	Ultra-high I/O
kafka-10	3	kafka.16u32g.cl uster	No	Ultra-high I/O
kafka-11	3	kafka.2u4g.clust er	No	High I/O
kafka-12	3	kafka.4u8g.clust er	No	High I/O
kafka-13	3	kafka.8u16g.clu ster	No	High I/O
kafka-14	3	kafka.12u24g.cl uster	No	High I/O
kafka-15	3	kafka.16u32g.cl uster	No	High I/O

After the purchase, obtain **Address (Private Network, Plaintext)** on the instance details page.



 Create topics with parameters specified in Table 12-6 for each instance purchased above. For details about how to create topics, see Creating a Kafka Topic.

Table 12-6 Topic parameters

Topic Name	Synchronous Replication	Synchronous Flushing	Replicas	Partitions
topic-01	No	No	3	30
topic-02	Yes	No	3	30
topic-03	No	Yes	3	30
topic-04	No	No	3	3
topic-05	No	No	3	12
topic-06	No	No	3	100

3. Obtain the test tool.

Obtain Kafka CLI v2.7.2.

4. Purchase a server for the client.

Buy a Linux ECS (with the same region, AZ, VPC, subnet, and security group as the Kafka instance). For details about how to purchase an ECS, see **Purchasing an ECS**.

Perform the following operations on the ECSs:

 Install Java JDK and configure the environment variables JAVA_HOME and PATH.

export JAVA_HOME=/root/jdk1.8.0_231 export PATH=\$JAVA_HOME/bin:\$PATH

 Download Kafka CLI v2.7.2 and decompress it. tar -zxf kafka_2.12-2.7.2.tgz

Script

./kafka-producer-perf-test.sh --producer-props bootstrap.servers={Connection address} acks=1 batch.size=16384 linger.ms=10 --topic {Topic name} --num-records 10000000 --record-size 1024 --throughput -1 --producer.config ../config/producer.properties

Table 12-7 Script parameters

Parameter	Description
bootstrap.servers	Address of the Kafka instance obtained in 1.
acks	Message synchronization policy. acks=1 indicates asynchronous replication, and acks=-1 indicates synchronous replication.
batch.size	Size of messages sent in each batch, in bytes.
linger.ms	Interval between two batches.
topic	Topic name set in 2.
num-records	Total number of messages to be sent.
record-size	Size of each message.
throughput	Number of messages sent per second.

Result

The test results of the five test scenarios are as follows:

Test scenarios:

- Scenario 1 (whether SASL is enabled): same topic (30 partitions, 3 replicas, asynchronous replication, and asynchronous flushing), instances with SASL enabled or disabled.
- Scenario 2 (synchronous/asynchronous replication): same instance (ultra-high I/O, three brokers, SASL disabled), topics with different replication settings, and number of producer processes is three.
- Scenario 3 (synchronous/asynchronous replication flushing): same instance (ultra-high I/O, three brokers, SASL disabled), topics with different flushing settings.
- Scenario 4 (different disk types): same topic (30 partitions, 3 replicas, asynchronous replication, and asynchronous flushing) with different disk types.
- Scenario 5 (different numbers of partitions): same instance (ultra-high I/O, three brokers, SASL disabled), topics with different number of partitions.

Scenario 1 (Whether SASL Is Enabled)

Table 12-8 Test results

Instance Flavor	Storage space per broker	Brokers	TPS (SASL Enabled)	TPS (SASL Disabled)
kafka.2u4g.clust er	Ultra-high I/O	3	100,000	280,000
kafka.4u8g.clust er	Ultra-high I/O	3	170,000	496,000
kafka.8u16g.clus ter	Ultra-high I/O	3	200,000	730,000
kafka.12u24g.clu ster	Ultra-high I/O	3	320,000	790,000
kafka.16u32g.clu ster	Ultra-high I/O	3	360,000	1,000,000

Conclusion: When messages are produced to Kafka instances with the same flavor and topic but different access modes, instances without SASL show higher TPS than those with SASL.

Scenario 2 (Synchronous/Asynchronous Replication)

Table 12-9 Test results

Instance Flavor	Synchron ous Flushing	Replicas	Partitions	TPS (Synchron ous Replicatio n)	TPS (Asynchro nous Replicatio n)
kafka.2u4g.clus ter	No	3	30	100,000	280,000
kafka.4u8g.clus ter	No	3	30	230,000	496,000
kafka.8u16g.clu ster	No	3	30	342,000	730,000
kafka.12u24g.cl uster	No	3	30	383,000	790,000
kafka.16u32g.cl uster	No	3	30	485,000	1,000,000

Conclusion: When messages are produced to different topics of a Kafka instance, topics with asynchronous replication show higher TPS than those with synchronous replication when other topic parameters are the same.

Scenario 3 (Whether Synchronous Flushing Is Enabled)

Table 12-10 Test results

Instance Flavor	Synchron ous Replicati on	Replicas	Partitions	TPS (Synchron ous Flushing)	TPS (Asynchro nous Flushing)
kafka.2u4g.clus ter	No	3	30	30,000	280,000
kafka.4u8g.clus ter	No	3	30	32,500	496,000
kafka.8u16g.clu ster	No	3	30	36,100	730,000
kafka.12u24g.cl uster	No	3	30	37,400	790,000
kafka.16u32g.cl uster	No	3	30	40,400	1,000,000

Conclusion: When messages are produced to different topics of a Kafka instance, topics with asynchronous flushing show significantly higher TPS than those with synchronous flushing when other topic parameters are the same.

Scenario 4 (Different Disk Types)

Table 12-11 Test results

Instance Flavor	Brokers	SASL	TPS (High I/O)	TPS (Ultra-High I/O)
kafka.2u4g.clust er	3	No	110,000	250,000
kafka.4u8g.clust er	3	No	135,000	380,000
kafka.8u16g.clus ter	3	No	213,000	480,000
kafka.12u24g.clu ster	3	No	240,000	577,000
kafka.16u32g.clu ster	3	No	280,000	840,000

Conclusion: When messages are produced to the same topics of Kafka instances with the same flavor but different disk types, instances with ultra-high I/O disks show higher TPS than those with high I/O disks.

Scenario 5 (Different Number of Partitions)

Table 12-12 Test results

Instance Flavor	Synch ronou s Flushi ng	Synch ronou s Replic ation	Replic as	TPS (3 Partitions)	TPS (12 Partitions)	TPS (100 Partitions)
kafka.2u4g.cl uster	No	No	3	250,000	260,000	250,000
kafka.4u8g.cl uster	No	No	3	330,000	280,000	260,000
kafka.8u16g.cl uster	No	No	3	480,000	410,000	340,000
kafka.12u24g. cluster	No	No	3	570,000	750,000	520,000
kafka.16u32g. cluster	No	No	3	840,000	1,000,000	630,000

Conclusion: When messages are produced to topics with different partition quantities of a Kafka instance, instances with more partitions show higher performance when other parameters are the same. However, performance reaches a peak and then deteriorates when partitions continue to increase.

13 Applying for Increasing Kafka Quotas

What Is a Quota?

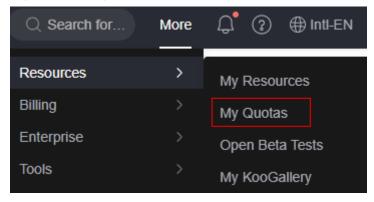
A quota is a limit on the quantity or capacity of a certain type of service resources that you can use, for example, the maximum number of Kafka instances that you can create.

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

How Do I View My Quota?

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

Figure 13-1 My Quotas



4. On the **Quotas** page, view the used and total quotas of resources.

If a quota cannot meet your needs, apply for a higher quota by performing the following operations.

How Do I Increase My Quota?

- 1. Log in to the **Kafka console**.
- In the upper right corner of the page, choose Resources > My Quotas.
 The Service Quota page is displayed.
- 3. Click Increase Quota.
- On the Create Service Ticket page, set the parameters.
 In the Problem Description area, enter the required quota and the reason for the quota adjustment.
- 5. Read the agreements and confirm that you agree to them, and then click **Submit**.

14 Monitoring and Alarms

14.1 Viewing Kafka Metrics

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

Constraints

On the **Monitoring > Monitoring Details** page of the Kafka console, you can select a maximum of 50 resources from the drop-down list at a time. If you need to view the monitoring data of more than 50 resources, do so in batches.

Prerequisite

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

Procedure

- Step 1 Log in to the Kafka console.
- **Step 2** Click in the upper left corner to select the region where your instance is located.
- **Step 3** View the instance metrics in either of the following ways:

Table 14-1 Viewing metrics

Data Type	Check Method	
All	 In the row containing the desired instance, click View Metric. On the Cloud Eye console, view the metrics. Metric data is reported to Cloud Eye every minute. Click a Kafka instance name to go to the instance details page. In the navigation pane, choose 	
	Monitoring > Monitoring Details. On the displayed page, view the monitoring data. The data is updated every minute.	
Specific	Click a Kafka instance name to go to the instance details page. In the navigation pane, choose Instance > Topics . Click the desired topic to go to the topic details page. On the monitoring page, view the monitoring data by topic. The data is updated every minute.	
Specified by consumer group	Click a Kafka instance name to go to the instance details page. In the navigation pane, choose Instance > Consumer Groups . Click the desired consumer group to go to the consumer group details page. On the monitoring page, view the monitoring data by consumer group. The data is updated every minute.	

Click the following dimensions to view monitoring data:

- Single-node instance: **By Instance**, **By Broker**, **By Topic**, or **By Consumer Group**.
- Cluster instance: By Instance, By Broker, By Topic, By Consumer Group, or By Smart Connect.

----End

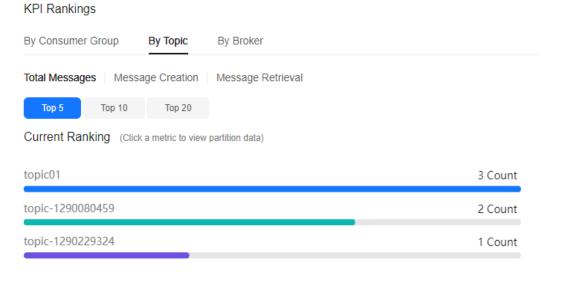
Viewing Top Kafka KPI Data

- Step 1 Log in to the Kafka console.
- **Step 2** Click on the upper left corner to select the region where your instance is located.
- **Step 3** Click the desired instance to go to the instance details page.
- **Step 4** Choose **Monitoring** > **Overview** in the navigation pane.
- **Step 5** In the **KPI Rankings** area, check the top 5, top 10, and top 20 data by consumer group, topic, and broker.

Table 14-2 KPI list

Dimension	Key Metric
By Consumer Group	Accumulated MessagesConsumer Retrieved Messages
By Topic	 Total Messages Message Creation Message Retrieval
Broker	CPU UsageNetwork Bandwidth UsageDisk Read SpeedDisk Write Speed

Figure 14-1 KPI ranking



----End

Checking the Monitoring View of Kafka Production Inbound Traffic

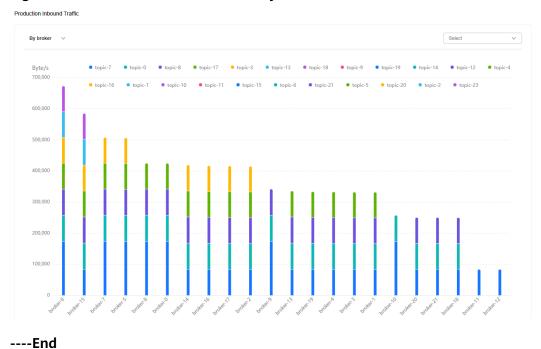
This function is unavailable for some existing instances. Contact customer service.

- **Step 1** Log in to the **Kafka console**.
- **Step 2** Click in the upper left corner to select the region where your instance is located.
- **Step 3** Click the desired instance to go to the instance details page.
- **Step 4** In the navigation pane, choose **Monitoring** > **Overview**.
- **Step 5** On **KPI Rankings** > **By Dimension** tab page, check the production inbound traffic.

Table 14-3 Production inbound traffic

Dimension	Description
By topic	Topics are sorted by production inbound traffic in descending order, and the traffic is displayed by broker in different colors. Hovering your mouse over a topic bar displays the production inbound traffic of the broker.
	Select one or more brokers from the drop-down box in the upper right corner to view the broker production inbound traffic.
	A maximum of top 20 topics can be displayed.
By broker	Brokers are sorted by production inbound traffic in descending order, and the traffic is displayed by topic in different colors. Hovering your mouse over a broker bar displays the production inbound traffic of all topics.
	Select one or more topics from the drop-down box in the upper right corner to view the topic production inbound traffic.

Figure 14-2 Production inbound traffic by broker



Related Document

Why Can't I View the Monitoring Data?

14.2 Kafka Metrics

Introduction

This section describes metrics reported by DMS for Kafka to Cloud Eye as well as their namespaces and dimensions. You can use the Cloud Eye console or APIs to query the Kafka metrics and alarms, or view Kafka instance metrics on the Monitoring Details page of the DMS for Kafka console.

Namespace

SYS.DMS

Instance Metrics

Table 14-4 Instance metrics

Metri c ID	Metr ic Nam e	Description	Value Range	Unit	Con vers ion Rul e	Monitor ed Object (Dimens ion)	Monitori ng Period (Raw Data)
curren t_part itions	Partit ions	Number of used partitions in the instance	0- 100,00 0	Count	N/A	Kafka instance	1 minute
curren t_topi cs	Topic s	Number of created topics in the instance	0- 100,00 0	Count	N/A	Kafka instance	1 minute
group _msgs	Accu mula ted Mess ages	Total number of accumulated messages in all consumer groups of the instance	0- 1,000,0 00,000	Count	N/A	Kafka instance	1 minute

Metri c ID	Metr ic Nam e	Description	Value Range	Unit	Con vers ion Rul e	Monitor ed Object (Dimens ion)	Monitori ng Period (Raw Data)
instan ce_byt es_in_ rate	Mess age Prod uctio n	Number of bytes produced in the instance per second Some instances do not support this metric. Check whether your instance supports it on the console.	0- 1,000,0 00	Byte/s	102 4(IE C)	Kafka instance	1 minute
instan ce_byt es_out _rate	Mess age Cons umpt ion	Number of bytes consumed from the instance per second Some instances do not support this metric. Check whether your instance supports it on the console.	0- 1,000,0 00	Byte/s	102 4(IE C)	Kafka instance	1 minute
curren t_part itions_ usage	Curre nt Partit ions Usag e	Current Partitions Usage Some instances do not support this metric. Check whether your instance supports it on the console.	0-100	%	N/A	Kafka instance	1 minute

Broker Metrics

Enabling Smart Connect for a Kafka instance creates two or more brokers. On the **By Broker** tab page, select "connector" for **Node Type** for Smart Connect broker metrics, or select "broker" for **Node Type** for Kafka instance broker metrics.

Metrics of Smart Connect brokers: disk capacity usage, memory usage, JVM heap memory usage, node alive status, and connections.

Table 14-5 Broker metrics

Metri c ID	Metr ic Nam e	Description	Value Range	Unit	Con vers ion Rul e	Monitor ed Object (Dimens ion)	Monitori ng Period (Raw Data)
broker _data _size	Mess age Size	Total size of messages in the broker	0- 5,000,0 00,000, 000	Byte	102 4(IE C)	Kafka instance broker	1 minute
broker _mess ages_i n_rate	Mess age Creat ion Rate	Number of messages created per second	0- 500,00 0	Count/ s	N/A	Kafka instance broker	1 minute
broker _bytes _out_r ate	Mess age Retri eval	Number of bytes retrieved per second	0- 500,00 0,000	Byte/s	102 4(IE C)	Kafka instance broker	1 minute
broker _bytes _in_ra te	Mess age Creat ion	Number of bytes created per second	0- 500,00 0,000	Byte/s	102 4(IE C)	Kafka instance broker	1 minute
broker _fetch _mea n	Aver age Mess age Retri eval Proce ssing Dura tion	Average time that the broker spends processing message retrieval requests	0- 10,000	ms	N/A	Kafka instance broker	1 minute

Metri c ID	Metr ic Nam e	Description	Value Range	Unit	Con vers ion Rul e	Monitor ed Object (Dimens ion)	Monitori ng Period (Raw Data)
broker _prod uce_m ean	Aver age Mess age Creat ion Proce ssing Dura tion	Average time that the broker spends processing message creation requests	0- 10,000	ms	N/A	Kafka instance broker	1 minute
broker _cpu_ core_l oad	Aver age Load per CPU Core	Average load of each CPU core of the Kafka VM	0-20	N/A	N/A	Kafka instance broker	1 minute
broker _disk_ usage	Disk Capa city Usag e	Disk usage of the Kafka VM	0–100	%	N/A	Kafka instance broker	1 minute
broker _mem ory_us age	Mem ory usag e of the broke r VM	Memory usage of the Kafka VM	0–100	%	N/A	Kafka instance broker	1 minute
broker _heap _usag e	JVM Heap Mem ory Usag e of JVM	Heap memory usage of the Kafka JVM	0-100	%	N/A	Kafka instance broker	1 minute

Metri c ID	Metr ic Nam e	Description	Value Range	Unit	Con vers ion Rul e	Monitor ed Object (Dimens ion)	Monitori ng Period (Raw Data)
broker _alive	Brok er Alive	Whether the Kafka broker is alive This metric is supported by instances purchased in April 2020 or later.	• 1: alive • 0: not alive	N/A	N/A	Kafka instance broker	1 minute
broker _conn ection s	Conn ectio ns	Total number of TCP connections on the Kafka broker This metric is supported by instances purchased in April 2020 or later.	0- 65,535	Count	N/A	Kafka instance broker	1 minute
broker _cpu_ usage	CPU Usag e	CPU usage of the Kafka VM This metric is supported by instances purchased in April 2020 or later.	0–100	%	N/A	Kafka instance broker	1 minute
broker _disk_ read_ await	Aver age Disk Read Time	Average time for each disk I/O read in the monitoring period This metric is supported for instances purchased in June 2020 or later.	> 0	ms	N/A	Kafka instance broker	1 minute

Metri c ID	Metr ic Nam e	Description	Value Range	Unit	Con vers ion Rul e	Monitor ed Object (Dimens ion)	Monitori ng Period (Raw Data)
broker _disk_ write_ await	Aver age Disk Write Time	Average time for each disk I/O write in the monitoring period This metric is supported for instances purchased in June 2020 or later.	> 0	ms	N/A	Kafka instance broker	1 minute
broker _total _bytes _in_ra te	Inbo und Traffi c	Inbound traffic per second This metric is supported for instances purchased in June 2020 or later.	0- 1,000,0 00,000	Byte/s	102 4(IE C)	Kafka instance broker	1 minute
broker _total _bytes _out_r ate	Outb ound Traffi c	Outbound traffic per second This metric is supported for instances purchased in June 2020 or later.	0- 1,000,0 00,000	Byte/s	102 4(IE C)	Kafka instance broker	1 minute
broker _disk_ read_r ate	Disk Read Spee d	Read traffic on the disk This metric is supported for instances purchased on or after May 16, 2022.	≥ 0	Byte/s	102 4(IE C)	Kafka instance broker	1 minute

Metri c ID	Metr ic Nam e	Description	Value Range	Unit	Con vers ion Rul e	Monitor ed Object (Dimens ion)	Monitori ng Period (Raw Data)
broker _disk_ write_ rate	Disk Write Spee d	Write traffic on the disk This metric is supported for instances purchased on or after May 16, 2022.	≥ 0	Byte/s	102 4(IE C)	Kafka instance broker	1 minute
netwo rk_ba ndwid th_us age	Netw ork Band width Usag e	Network bandwidth usage This metric is supported only: For instances purchased since July 9 2023. For instances purchased before July 9 2023, this metric is supported for brokers if they are added since July 9 2023.	0-100	%	N/A	Kafka instance broker	1 minute

Topic Metrics

Table 14-6 Topic metrics

Metri c ID	Metr ic Nam e	Description	Value Range	Unit	Con vers ion Rul e	Monitor ed Object (Dimens ion)	Monitori ng Period (Raw Data)
topic_ bytes_ in_rat e	Mess age Creat ion	Number of bytes created per second This metric is available only when Monitoring Type is set to Basic monitoring on the By Topic tab page.	0- 500,00 0,000	Byte/s	102 4(IE C)	Topic in a Kafka instance	1 minute
topic_ bytes_ out_ra te	Mess age Retri eval	Number of bytes retrieved per second This metric is available only when Monitoring Type is set to Basic monitoring on the By Topic tab page.	0- 500,00 0,000	Byte/s	102 4(IE C)	Topic in a Kafka instance	1 minute
topic_ data_s ize	Mess age Size	Total size of messages in the queue This metric is available only when Monitoring Type is set to Basic monitoring on the By Topic tab page.	0- 5,000,0 00,000, 000	Byte	102 4(IE C)	Topic in a Kafka instance	1 minute

Metri c ID	Metr ic Nam e	Description	Value Range	Unit	Con vers ion Rul e	Monitor ed Object (Dimens ion)	Monitori ng Period (Raw Data)
topic_ messa ges	Total Mess ages	Total number of messages in the queue This metric is available only when Monitoring Type is set to Basic monitoring on the By Topic tab page.	≥ 0	Count	N/A	Topic in a Kafka instance	1 minute
topic_ messa ges_in _rate	Mess age Creat ion Rate	Number of messages created per second This metric is available only when Monitoring Type is set to Basic monitoring on the By Topic tab page.	0- 500,00 0	Count/ s	N/A	Topic in a Kafka instance	1 minute
partiti on_m essag es	Partit ion Mess ages	Total number of messages in the partition This metric is available only when Monitoring Type is set to Partition monitoring on the By Topic tab page.	≥ 0	Count	N/A	Topic in a Kafka instance	1 minute

Metri c ID	Metr ic Nam e	Description	Value Range	Unit	Con vers ion Rul e	Monitor ed Object (Dimens ion)	Monitori ng Period (Raw Data)
produ ced_m essag es	Creat ed Mess ages	Number of messages that have been created This metric is available only when Monitoring Type is set to Partition monitoring on the By Topic tab page.	≥ 0	Count	N/A	Topic in a Kafka instance	1 minute
partiti on_m essag es_in_ rate	Partit ion mess age prod uctio n rate	Number of messages created per second in a partition. This metric is available only when Monitoring Type is set to Partition monitoring on the By Topic tab page. Some instances do not support this metric. Contact customer service.	≥ 0	Count/s	N/A	Topic in a Kafka instance	1 minute

Consumer Group Metrics

Table 14-7 Consumer group metrics

Metri c ID	Metr ic Nam e	Description	Value Range	Unit	Con vers ion Rul e	Monitor ed Object (Dimens ion)	Monitori ng Period (Raw Data)
messa ges_c onsu med	Retri eved Mess ages	Number of messages that have been retrieved in the consumer group This metric is available only when Topic is set to a specific topic name and Monitoring Type is set to Partition monitoring on the By Consumer Group tab page.	≥ 0	Count	N/A	Consume r group of a Kafka instance	1 minute
messa ges_re maine d	Num ber of accu mula tions Avail able Mess ages	Number of messages that can be retrieved in the consumer group This metric is available only when Topic is set to a specific topic name and Monitoring Type is set to Partition monitoring on the By Consumer Group tab page.	≥ 0	Count	N/A	Consume r group of a Kafka instance	1 minute

Metri c ID	Metr ic Nam e	Description	Value Range	Unit	Con vers ion Rul e	Monitor ed Object (Dimens ion)	Monitori ng Period (Raw Data)
topic_ messa ges_re maine d	Topic Avail able Mess ages	Number of remaining messages that can be retrieved from the specified topic in the consumer group This metric is available only when Topic is set to a specific topic name and Monitoring Type is set to Basic monitoring on the By Consumer Group tab page.	0 to 2 ⁶³ –1	Count	N/A	Consume r group of a Kafka instance	1 minute
topic_ messa ges_c onsu med	Topic Retri eved Mess ages	Number of messages that have been retrieved from the specified topic in the consumer group This metric is available only when Topic is set to a specific topic name and Monitoring Type is set to Basic monitoring on the By Consumer Group tab page.	0 to 2 ⁶³ –1	Count	N/A	Consume r group of a Kafka instance	1 minute

Metri c ID	Metr ic Nam e	Description	Value Range	Unit	Con vers ion Rul e	Monitor ed Object (Dimens ion)	Monitori ng Period (Raw Data)
consu mer_ messa ges_re maine d	Cons umer Avail able Mess ages	Number of remaining messages that can be retrieved in the consumer group This metric is available only when Topic is set to All topics on the By Consumer Group tab page.	0 to 2 ⁶³ –1	Count	N/A	Consume r group of a Kafka instance	1 minute
consu mer_ messa ges_c onsu med	Cons umer Retri eved Mess ages	Number of messages that have been retrieved in the consumer group This metric is available only when Topic is set to All topics on the By Consumer Group tab page.	0 to 2 ⁶³ –1	Count	N/A	Consume r group of a Kafka instance	1 minute

Metri c ID	Metr ic Nam e	Description	Value Range	Unit	Con vers ion Rul e	Monitor ed Object (Dimens ion)	Monitori ng Period (Raw Data)
messa ges_c onsu med_ per_m in	Partit ion Cons umpt ion Rate	Number of messages consumed from the specified queue partition in the consumer group every minute This metric is available only when Topic is set to a specific topic name and Monitoring Type is set to Partition monitoring on the By Consumer Group tab page. Some instances do not support this metric. Check whether your instance supports it on the console.	0- 30,000, 000	Count/ min	N/A	Consume r group of a Kafka instance	1 minute

Metri c ID	Metr ic Nam e	Description	Value Range	Unit	Con vers ion Rul e	Monitor ed Object (Dimens ion)	Monitori ng Period (Raw Data)
topic_ messa ges_c onsu med_ per_m in	Queu e Cons umpt ion Rate	Number of messages consumed from the specified queue in the consumer group every minute This metric is available only when Topic is set to a specific topic name and Monitoring Type is set to Basic monitoring on the By Consumer Group tab page. Some instances do not support this metric. Check whether your instance supports it on the console.	0- 30,000, 000	Count/ min	N/A	Consume r group of a Kafka instance	1 minute

Metri c ID	Metr ic Nam e	Description	Value Range	Unit	Con vers ion Rul e	Monitor ed Object (Dimens ion)	Monitori ng Period (Raw Data)
consu mer_ messa ges_c onsu med_ per_m in	Cons umer Grou p Cons umpt ion Rate	Number of messages consumed from the consumer group every minute This metric is available only when Topic is set to All topics on the By Consumer Group tab page. Some instances do not support this metric. Check whether your instance supports it on the console.	0- 30,000, 000	Count/ min	N/A	Consume r group of a Kafka instance	1 minute

Smart Connect Metrics

Smart Connect metrics are only available for cluster instances.

Table 14-8 Smart Connect metrics

Metri c ID	Metr ic Nam e	Description	Value Range	Unit	Con vers ion Rul e	Monitor ed Object (Dimens ion)	Monitori ng Period (Raw Data)
kafka _wait_ synchr onize_ data	Kafk a Data to Sync	Data to synchronize in the Kafka migration task	≥ 0	Count	N/A	Smart Connect task of a Kafka instance	1 minute

Metri c ID	Metr ic Nam e	Description	Value Range	Unit	Con vers ion Rul e	Monitor ed Object (Dimens ion)	Monitori ng Period (Raw Data)
kafka _sync hroniz e_rate	Kafk a Data Sync ed per Minu te	Data synchronized per minute in the Kafka migration task	≥ 0	Count	N/A	Smart Connect task of a Kafka instance	1 minute
task_s tatus	Task Statu s	Status of the current task	• 0: abn orm al • 1: nor mal	N/A	N/A	Smart Connect task of a Kafka instance	1 minute
messa ge_del ay	Mess age Dela y	Time elapsed between when a message is sent from the source and received by the target	≥ 0	ms	N/A	Smart Connect task of a Kafka instance	1 minute

Precautions:

- A Smart Connect task that bidirectionally copies Kafka data is split into two tasks for monitoring: *Smart Connect task name_source_0* and *Smart Connect task name_source_1*.
- If all messages in a topic have aged before the next synchronization, there is no Kafka data to be synchronized. However, since the Kafka data synchronization metric uses the offset value that contains aged data, **Kafka Data Synced per Minute** will display the number of aged messages.

Dimension

Кеу	Value
kafka_instance_id	Kafka instance ID, for example, 6a0df74a-fxxx-497bfdda6c8e.

Кеу	Value
kafka_broker	Kafka instance node. The node ID starts from 0 and increases in ascending order. For example, if an instance has three nodes, the nodes are broker-0, broker-1, and broker-2.
kafka_topics	Topic name, for example, topic-01 .
kafka_partitions	Partition number in a topic, for example, 0 .
kafka_groups_partitions	Partition number in a topic consumed by a consumer group, for example, 0 .
kafka_groups_topics	Topic name consumed by a consumer group, for example, topic-01 .
kafka_groups	Consumer group name of a Kafka instance, for example, group-01 .
connector_task	Smart Connect task name, for example, smart-connect-01.

Note, the following dimension level is described:

- **kafka_partitions** is a sub-dimension of **kafka_topics**.
- **kafka_groups_topics** and **kafka_groups_partitions** are sub-dimensions of **kafka_groups**.

14.3 Configuring a Kafka Alarm Rule

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

Table 14-9 Alarm policies and handling of Kafka instances

Metric ID	Metric	Moni tored Objec t	Alarm Policy	Description	Handling Suggestion
broker _disk_u sage	Disk Capacit y Usage	Broke r	Alarm threshold: original value > 80% Number of consecutive periods: 1 Alarm severity: critical	Disk usage of the Kafka VM	Modify the instance storage space. For details, see Modifying Instance Specifications.
broker _mem ory_us age	Memor y Usage	Broke r	Alarm threshold: original value > 90% Number of consecutive periods: 3 Alarm severity: critical	Memory usage of the Kafka VM.	Modify the instance bandwidth or the number of brokers. For details, see Modifying Instance Specifications.
current _partiti ons	Partitio ns	Insta nce	Alarm threshold: original value > 90% of the maximum allowed number of partitions. The partition limit varies depending on instance specifications. For details, see Specification s. Number of consecutive periods: 1 Alarm severity: major	Number of used partitions in the instance.	If new topics are required, modify the instance bandwidth or the number of brokers, or split the service to multiple instances. For details about how to modify the instance bandwidth or the number of brokers, see Modifying Instance Specifications.

Metric ID	Metric	Moni tored Objec t	Alarm Policy	Description	Handling Suggestion
broker _cpu_u sage	CPU Usage	Broke r	Alarm threshold: original value > 90% Number of consecutive periods: 3 Alarm severity: major	CPU usage of the Kafka VM.	Check whether the metric has been approaching or exceeding the alarm threshold for a long time. If yes, modify the instance bandwidth or the number of brokers. For details, see Modifying Instance Specifications.
group_ msgs	Accum ulated Messa ges	Insta nce	Alarm threshold: original value > 90% of the upper limit. The upper limit is customized. Number of consecutive periods: 1 Alarm severity: major	Total number of accumulated messages in all consumer groups of the instance	Delete idle consumer groups, if any. You can also accelerate message retrieval, for example, by increasing the number of consumers.
topic_ messa ges_re maine d	Topic Availab le Messa ges	Consu mer group	Alarm threshold: original value > 90% of the upper limit. The upper limit is customized. Number of consecutive periods: 1 Alarm severity: major	Number of remaining messages that can be retrieved from the specified topic in the consumer group.	Check whether the consumer code logic is correct, for example, by checking whether the consumer stops consuming messages due to an exception. You can also accelerate message retrieval, for example, by adding topic consumers. Ensure that the number of partitions is greater than or equal to the number of consumers.

Metric ID	Metric	Moni tored Objec t	Alarm Policy	Description	Handling Suggestion
task_st atus	Task Status	Smart Conn ect task	Alarm threshold: Raw value < 1 Number of consecutive periods: 3 Alarm severity: major	Whether Smart Connect tasks are in the normal state.	Pause the Smart Connect task, then restart it, and check whether the task status is recovered. If the fault persists, contact customer service.

Configuring Kafka Alarm Rules

The following section describes how to configure alarm rules for a specified Kafka instance.

- Step 1 Log in to the Kafka console.
- **Step 2** Click in the upper left corner to select the region where your instance is located.
- **Step 3** Go to the monitoring page in either of the following ways:
 - Click **View Metric** in the row containing the desired Kafka instance.
 - Click the desired Kafka instance to go to the instance details page. Choose
 Monitoring > Monitoring Details in the navigation pane.
- **Step 4** 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 5** Specify the alarm details.

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

- 1. Set the alarm name and description.
- 2. Set the alarm policy.

As shown in the following figure, if the original disk capacity usage is equal to or higher than 80% once, an alarm is generated. If the alarm is not handled on time, an alarm notification is sent.

Figure 14-3 Setting the alarm policy



3. Set the alarm notification configurations.

If you enable **Alarm Notification**, specify **Notification Recipient** and **Notification Policies**.

4. Click Create.

----End

Viewing Alarm Rules of a Kafka Instance

The following procedure describes how to query all alarm rules of a specified Kafka instance.

- Step 1 Log in to the Kafka console.
- **Step 2** Click in the upper left corner to select the region where your instance is located.
- Step 3 Click in the upper left corner of the console, choose Management & Governance > Cloud Eye.
- **Step 4** In the navigation pane, choose **Cloud Service Monitoring**.
- **Step 5** Search for "Distributed Message Service" and press **Enter**.
- **Step 6** Click **Distributed Message Service DMS**. The **Details** page is displayed.
- **Step 7** In the row containing the desired Kafka instance, choose **More** > **View Alarm Rule**. All alarm rules of this Kafka instance are displayed in the **View Alarm Rule** dialog box.

----End

15 Viewing Kafka Audit Logs

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

Prerequisite

CTS has been enabled.

DMS for Kafka Operations Supported by CTS

Table 15-1 DMS for Kafka operations that can be recorded by CTS

Operation	Resource Type	Trace Name
Successfully creating an order for creating an instance	kafka	createDMSInstanceOrderSuccess
Successfully creating an instance	kafka	createDMSInstanceTaskSuccess
Failing to create an order for creating an instance	kafka	createDMSInstanceOrderFailure
Failing to create an instance	kafka	createDMSInstanceTaskFailure
Successfully deleting an instance that failed to be created	kafka	deleteDMSCreateFailureInstan- cesSuccess
Failing to delete an instance that failed to be created	kafka	deleteDMSCreateFailureInstan- cesFailure
Successfully deleting an instance	kafka	deleteDMSInstanceTaskSuccess

Operation	Resource Type	Trace Name
Failing to delete an instance	kafka	deleteDMSInstanceTaskFailure
Deleting multiple instance tasks at a time	kafka	batchDeleteDMSInstanceTask
Successfully submitting a request to delete multiple instances at a time	kafka	batchDeleteDMSInstanceSuccess
Successfully deleting multiple instances at a time	kafka	batchDeleteDMSInstanceTask- Success
Failing to submit a request to delete multiple instances at a time	kafka	batchDeleteDMSInstanceFailure
Failing to delete multiple instances at a time	kafka	batchDeleteDMSInstanceTask- Failure
Successfully submitting a request to modify an instance order	kafka	modifyDMSInstanceOrderSuccess
Failing to submit a request to modify an instance order	kafka	modifyDMSInstanceOrderFailure
Successfully submitting a request to scale up an instance	kafka	extendDMSInstanceSuccess
Successfully scaling up an instance	kafka	extendDMSInstanceTaskSuccess
Failing to submit a request to scale up an instance	kafka	extendDMSInstanceFailure
Failing to scale up an instance	kafka	extendDMSInstanceTaskFailure
Successfully submitting a request to reset instance password	kafka	resetDMSInstancePasswordSuccess

Operation	Resource Type	Trace Name
Failing to submit a request to reset instance password	kafka	resetDMSInstancePasswordFai- lure
Successfully submitting a request to restart an instance	kafka	restartDMSInstanceSuccess
Successfully restarting an instance	kafka	restartDMSInstanceTaskSuccess
Failing to submit a request to restart an instance	kafka	restartDMSInstanceFailure
Failing to restart an instance	kafka	restartDMSInstanceTaskFailure
Successfully submitting a request to restart multiple instances at a time	instance	batchRestartDMSInstanceSuc- cess
Successfully restarting multiple instances at a time	kafka	batchRestartDMSInstanceTask- Success
Failing to submit a request to restart multiple instances at a time	instance	batchRestartDMSInstanceFailure
Failing to restart multiple instances at a time	kafka	batchRestartDMSInstanceTask- Failure
Successfully submitting a request to modify instance information	kafka	modifyDMSInstanceInfoSuccess
Successfully modifying instance information	kafka	modifyDMSInstanceInfoTaskSuccess
Failing to submit a request to modify instance information	kafka	modifyDMSInstanceInfoFailure
Failing to modify instance information	kafka	modifyDMSInstanceInfoTaskFai- lure
Successfully deleting a background task	kafka	deleteDMSBackendJobSuccess

Operation	Resource Type	Trace Name
Failing to delete a background task	kafka	deleteDMSBackendJobFailure
Successfully enabling Smart Connect	kafka	CreateConnectorSuccess
Failing to enable Smart Connect	kafka	createConnectorFailure
Successfully creating a Smart Connect task	kafka	createConnectorTaskSuccess
Failing to create a Smart Connect task	kafka	createConnectorTaskFailure
Successfully modifying a Smart Connect task	kafka	putConnectorTaskSuccess
Failing to modify a Smart Connect task	kafka	putConnectorTaskFailure
Successfully deleting a Smart Connect task	kafka	deleteConnectorTaskSuccess
Failing to delete a Smart Connect task	kafka	deleteConnectorTaskFailure
Successfully restarting a Smart Connect task	kafka	restartConnectorSuccess
Failing to restart a Smart Connect task	kafka	restartConnectorFailure
Successfully pausing a Smart Connect task	kafka	pauseConnectorSuccess
Failing to pause a Smart Connect task	kafka	pauseConnectorFailure
Successfully starting a Smart Connect task	kafka	resumeConnectorSuccess
Failing to start a Smart Connect task	kafka	resumeConnectorFailure
Connectivity check succeeded	kafka	ValidateConnectorConnectivity- Success
Connectivity check failed	kafka	ValidateConnectorConnectivity- Failure
Successfully disabling Smart Connect	kafka	DeleteConnectorSuccess
Failing to disable Smart Connect	kafka	DeleteConnectorFailure

Operation	Resource Type	Trace Name
Successfully creating a connector	kafka	createConnectorNodeTaskSuc- cess
Failing to create a connector	kafka	createConnectorNodeTaskFai- lure
Successfully deleting a connector	kafka	deleteConnectorNodeTaskSuc- cess
Failing to delete a connector	kafka	deleteConnectorNodeTaskFai- lure
Successfully freezing an instance	kafka	freezeDMSInstanceTaskSuccess
Failing to freeze an instance	kafka	freezeDMSInstanceTaskFailure
Successfully unfreezing an instance	kafka	unfreezeDMSInstanceTaskSuc- cess
Failing to unfreeze an instance	kafka	unfreezeDMSInstanceTaskFai- lure
Successfully creating a topic for a Kafka instance	kafka	Kafka_create_topicSuccess
Failing to create a topic for a Kafka instance	kafka	Kafka_create_topicFailure
Successfully deleting a topic from a Kafka instance	kafka	Kafka_delete_topicsSuccess
Failing to delete a topic for a Kafka instance	kafka	Kafka_delete_topicsFailure
Successfully enabling automatic topic creation	kafka	enable_auto_topicSuccess
Failing to enable automatic topic creation	kafka	enable_auto_topicFailure
Successfully modifying a topic	kafka	Kafka_alter_topicsSuccess
Failing to modify a topic	kafka	Kafka_alter_topicsFailure

Operation	Resource Type	Trace Name
Successfully reassigning partitions	kafka	kafka_reassignmentTaskSuccess
Failing to reassign partitions	kafka	kafka_reassignmentTaskFailure
Successfully submitting a partition reassignment request	kafka	kafka_reassignmentSuccess
Failing to submit a partition reassignment request	kafka	kafka_reassignmentFailure
Successfully resetting the consumer offset	kafka	Kafka_reset_consumer_offsetSuc cess
Failing to reset the consumer offset	kafka	Kafka_reset_consumer_offsetFail ure
Successfully deleting consumer groups in batches	kafka	Kafka_batch_delete_groupSucce ss
Failing to delete consumer groups in batches	kafka	Kafka_batch_delete_groupFailur e
Successfully creating a user	kafka	createUserSuccess
Failing to create a user	kafka	createUserFailure
Successfully deleting a user	kafka	deleteUserSuccess
Failing to delete a user	kafka	deleteUserFailure
Successfully updating user policies	kafka	updateUserPoliciesTaskSuccess
Failing to update user policies	kafka	updateUserPoliciesTaskFailure
Successfully recovering an instance from Recycle Bin	kafka	out_recycleTaskSuccess
Failing to recover an instance from Recycle Bin	kafka	out_recycleTaskFailure

Viewing Audit Logs

See Viewing CTS Traces in the Trace List.