

Distributed Message Service for Kafka

Getting Started

Issue 01
Date 2023-05-31



Copyright © Huawei Cloud Computing Technologies Co., Ltd. 2023. All rights reserved.

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

Trademarks and Permissions



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

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

Notice

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

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

Contents

1 Introduction.....	1
2 Step 1: Prepare the Environment.....	3
3 Step 2: Create a Kafka Instance.....	6
4 (Optional) Step 3: Create a Topic.....	11
5 Step 4: Connect to a Kafka Instance to Create and Retrieve Messages.....	13
5.1 Connecting to an Instance Without SASL.....	13
5.2 Connecting to an Instance with SASL.....	15
6 Step 5: Configure Alarm Rules.....	20

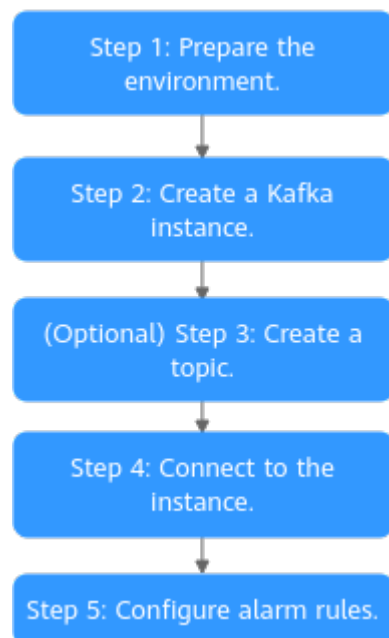
1 Introduction

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

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

Procedure

Figure 1-1 Procedure for using DMS for Kafka



1. **Prepare the environment.**

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

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

2. **Create a Kafka instance.**

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

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

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

4. Connect to the instance.

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

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

5. **Configure alarm rules.**

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

 **NOTE**

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

2 Step 1: Prepare the Environment

VPC

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

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

For details, see [Creating a VPC](#). If you already have an available VPC and subnet, you do not need to create new ones.

Note the following when creating a VPC and subnet:

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

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

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

Note the following when creating a security group:

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

Table 2-1 Security group rules

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

Direction	Protocol	Port	Source	Description
Inbound	TCP	9095	0.0.0.0/0	Access a Kafka instance through the public network (with SSL encryption).
Inbound	TCP	9093	0.0.0.0/0	Access a Kafka instance within a VPC (with SSL encryption).
Inbound	TCP	9999	0.0.0.0/0	Access Kafka Manager.
Inbound	TCP	9011	198.19.128.0/17	Access a Kafka instance across VPCs using a VPC endpoint (with or without SSL).
Inbound	TCP	9011	0.0.0.0/0	Access a Kafka instance using DNAT (with or without SSL).

 **NOTE**

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

----End

(Optional) EIP

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


For details, see [Assigning an EIP](#).

Note the following when creating EIPs:

- The EIPs must be created in the region the Kafka instance is in.
- The number of EIPs must be the same as the number of Kafka instance brokers.

ECS

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

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

For details, see [Purchasing an ECS](#). If you already have an available ECS, skip this step.

Step 2 Log in to the ECS.

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

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

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

 **NOTE**

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

Step 4 Download an open-source Kafka client.

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

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

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

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

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

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

Step 5 Run the following command to decompress the package:

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

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

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

----End

Follow-Up Procedure

[Step 2: Create a Kafka Instance](#)

3 Step 2: Create a Kafka Instance

Prerequisites

Ensure that a VPC is available. For details about how to create a VPC, see the [Virtual Private Cloud User Guide](#).

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

Procedure

Step 1 Log in to the [Kafka console](#), and click **Buy Instance** in the upper right corner.

Step 2 Select a billing mode.

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

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

Step 5 Retain the default AZ settings.

Step 6 Specify the instance name and the enterprise project.

Step 7 Configure the following instance parameters:

Specifications: Select **Default** or **Custom**.

If you select Default, specify the version, broker flavor, number of brokers, and storage space to be supported by the Kafka instance based on the site requirements.

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

2. **CPU Architecture:** The x86 architecture is supported.

3. **Broker Flavor:** Select broker specifications that best fit your business needs. For **Brokers**, specify the broker quantity.

Maximum number of partitions per broker x Number of brokers = Maximum number of partitions of an instance. If the total number of partitions of all topics exceeds the maximum number of partitions allowed for an instance, topic creation will fail.

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

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

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.

- Flavor **kafka.2u4g.cluster**: The value range of **Storage Space** is 300–300,000 GB.
- Flavor **kafka.4u8g.cluster**: The value range of **Storage Space** is 300–600,000 GB.
- Flavor **kafka.8u16g.cluster**: The value range of **Storage Space** is 300–900,000 GB.
- Flavor **kafka.12u24g.cluster**: The value range of **Storage Space** is 300–900,000 GB
- Flavor **kafka.16u32g.cluster**: The value range of **Storage Space** is 300–900,000 GB

5. **Capacity Threshold Policy**: policy used when the disk usage reaches the threshold. The capacity threshold is 95%.
 - **Automatically delete**: Messages can be created and retrieved, but 10% of the earliest messages will be deleted to ensure sufficient disk space. This policy is suitable for scenarios where no service interruption can be tolerated. Data may be lost.
 - **Stop production**: New messages cannot be created, but existing messages can still be retrieved. This policy is suitable for scenarios where no data loss can be tolerated.

Figure 3-1 Default specifications

The screenshot shows the configuration interface for a Kafka instance. It includes the following sections:

- Version:** Three tabs for 2.7 (selected), 2.3.0, and 1.1.0.
- CPU Architecture:** A dropdown menu set to x86.
- Broker Flavor:** A table with columns: Flavor Name, TPS Limit per Broker, Maximum Partitions per Broker, and Recommended Consumer Groups per Broker.

Flavor Name	TPS Limit per Broker	Maximum Partitions per Broker	Recommended Consumer Groups per Broker
<input checked="" type="radio"/> kafka.2u4g.cluster	30,000	250	4,000
<input type="radio"/> kafka.4u8g.cluster	100,000	500	4,000
<input type="radio"/> kafka.8u16g.cluster	150,000	1,000	4,000
<input type="radio"/> kafka.12u24g.cluster	200,000	1,500	4,000
<input type="radio"/> kafka.16u32g.cluster	250,000	2,000	4,000
- Brokers:** A numeric input field set to 3, with minus and plus buttons.
- Storage Space:** A dropdown menu set to Ultra-high I/O, followed by a numeric input field set to 100 and a plus button. Below it, text indicates: "Total storage space 300 GB" and "After the instance is created, you cannot change the disk type or reduce the storage space."

Ultra-high I/O | 100 | + GB

Total storage space 300 GB

After the instance is created, you cannot change the disk type or reduce the storage space. Learn more about disk types.
- Capacity Threshold Policy:** Two radio buttons: Automatically delete (selected) and Stop production.

If you select Custom, the system calculates the number of brokers and broker storage space for different flavors based on your specified peak creation traffic, retrieval traffic, number of replicas per topic, total number of

partitions, and size of messages created during the retention period. You can select one of the recommended flavors as required.

Figure 3-2 Specification calculation

Parameters

Peak Creation Traffic (MB/s) Retrieval Traffic (MB/s)

Replicas per Topic Total Partitions

Messages Created During Retention Period (GB)

Calculate

Recommended Specifications

Flavor Name	c6.2u4g.cluster	Flavor Name	c5.2u4g.cluster	Flavor Name	c6.4u8g.cluster	Flavor Name	c6.4u8g.cluster
Brokers	29	Brokers	30	Brokers	17	Brokers	30
Storage Space per Broker	1,300 GB	Storage Space per Broker	1,300 GB	Storage Space per Broker	2,300 GB	Storage Space per Broker	1,300 GB
Total storage space	Ultra-high I/O, 37,700 GB	Total storage space	High I/O, 39,000 GB	Total storage space	Ultra-high I/O, 39,100 GB	Total storage space	High I/O, 39,000 GB

Step 8 Configure the instance network parameters.

1. Select a VPC and a subnet.

NOTE

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

2. Select a security group.

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

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

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

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

1. Configure **Public Access**.

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

After enabling **Public Access**, you can enable or disable **Intra-VPC Plaintext Access**. **If it is enabled, data will be transmitted in plaintext when you connect to the instance through a private network, regardless of whether SASL_SSL is enabled. This setting cannot be changed after the instance is created.** Exercise caution. If you want to use a different setting, you must create a new instance.

Figure 3-3 Configuring public access

Public Access

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

Elastic IP Address [Create Elastic IP](#)

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

2. Configure **Kafka SASL_SSL**.

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

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

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

What are SCRAM-SHA-512 and PLAIN mechanisms?

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

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

3. Configure **Automatic Topic Creation**.

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

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

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

4. Specify tags.

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

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

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

5. Enter a description of the instance.

Step 11 Click **Buy**.

Step 12 Confirm the instance information.

Step 13 Return to the **Kafka Premium** page and check whether the instance has been created.

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

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

 **NOTE**

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

----End

Follow-Up Procedure

[\(Optional\) Step 3: Create a Topic](#)

4 (Optional) Step 3: Create a Topic

A topic is a stream of messages. If automatic topic creation is not enabled during Kafka instance creation, you need to manually create topics for creating and retrieving messages. If automatic topic creation is enabled, this step is optional. The system automatically creates a topic when a message is created. This topic has the following default settings: 3 partitions, 3 replicas, aging time 72 hours, and synchronous replication and flushing disabled.

The following describes three methods to manually create a topic.

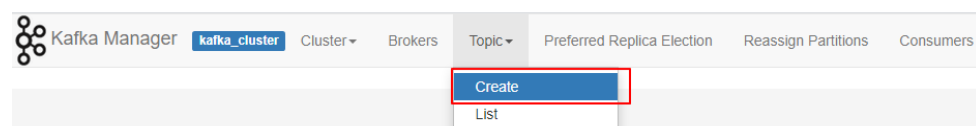
- [Method 1: Creating a Topic on the Console](#)
- [Method 2: Creating a Topic on Kafka Manager](#)
- [Method 3: Create a Topic by Using Kafka CLI](#)

Method 1: Creating a Topic on the Console

- Step 1** Log in to the [Kafka console](#), and select the region where the Kafka instance is located.
- Step 2** Click a Kafka instance.
- Step 3** On the **Topics** tab page, click **Create Topic**.
- Step 4** Enter the topic name, specify other parameters, and click **OK**.
- End

Method 2: Creating a Topic on Kafka Manager

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



NOTICE

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

Method 3: Create a Topic by Using Kafka CLI

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

NOTICE

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

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

```
./kafka-topics.sh --create --topic {topic_name} --bootstrap-server {broker_ip}:{port} --partitions {partition_num} --replication-factor {replication_num}
```
- If SASL has been enabled for the Kafka instance, perform the following steps to create a topic:
 - a. (Optional) If the SSL certificate configuration has been set, skip this step. Otherwise, perform the following operations:
Create the **ssl-user-config.properties** file in the **/config** directory of the Kafka client and add the SSL certificate configurations by referring to [Connecting to an Instance with SASL](#).
 - b. Run the following command in the *{directory where the CLI is located}*/**kafka_{version}/bin/** directory to create a topic:

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

Follow-Up Procedure

[Step 4: Connect to a Kafka Instance to Create and Retrieve Messages](#)

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

5.1 Connecting to an Instance Without SASL

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

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

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

NOTE

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

Prerequisites

- You have correctly configured security group rules. For details, see [Table 2-1](#).
- The instance connection address has been obtained.
 - For intra-VPC access, use port 9092. Obtain the instance connection address in the **Connection** section of the **Basic Information** tab page.

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

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


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

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

Instance Address (Public Network) 139.196.45:9094, 10.78.42.127:9094, 10.4.49.103:9094

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

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

Figure 5-3 Viewing the topic name

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

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

Creating Messages

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

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

Parameter description:

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

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

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

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

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

Retrieving Messages

Run the following command to retrieve messages:

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

Parameter description:

- *{connection-address}*: the address obtained in **Prerequisites**

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

The following is an example:

```
[root@ecs-kafka bin]# ./kafka-console-consumer.sh --bootstrap-server
10.3.196.45:9094,10.78.42.127:9094,10.4.49.103:9094 --topic topic-demo --group order-test --from-beginning
Kafka!
DMS
Hello
^CProcessed a total of 3 messages
[root@ecs-kafka bin]#
```

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

Follow-Up Procedure

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

Step 5: Configure Alarm Rules

5.2 Connecting to an Instance with SASL

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

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

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

NOTE

- If intra-VPC plaintext access is enabled for an instance, data is transmitted in plaintext when you connect to the instance through a private network. For details about how to connect, see [Connecting to an Instance Without SASL](#).
- Each Kafka broker allows a maximum of 1000 connections from each IP address by default. Excess connections will be rejected. You can change the limit by [modifying the Kafka parameters](#).

Prerequisites

- You have correctly configured security group rules. For details, see [Table 2-1](#).
- The instance connection address has been obtained.
 - For intra-VPC access, use port 9093. Obtain the instance connection address in the **Connection** section of the **Basic Information** tab page.

Figure 5-4 Kafka instance connection addresses for intra-VPC access with SASL

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

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

Figure 5-5 Kafka instance connection addresses for public access with SASL

Instance Address (Public Network) 139.128.145:9095,122.254.119:50:9095,119.129:9095

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

Figure 5-6 SASL mechanism in use
Connection

Username test [Reset Password](#)

Kafka SASL_SSL Enabled **Fixed for this instance**

SASL Mechanism SCRAM-SHA-512,PLAIN

- If automatic topic creation is not enabled for the Kafka instance, obtain the topic name.
You can obtain the name of the topic created in **(Optional) Step 3: Create a Topic** on the **Topics** tab page of the instance.

Figure 5-7 Viewing the topic name

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

Create Topic Delete Topic Edit Topic Reassign View Sample Code Enter a topic name

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

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

Configuring the Configuration File for Message Creation and Retrieval

Step 1 Log in to a Linux ECS.

Step 2 Map hosts to IP addresses in the **/etc/hosts** file on the ECS, so that the client can quickly parse the instance brokers.

Set IP addresses to the instance connection addresses obtained in [Prerequisites](#). Set hosts to the names of instance hosts. Specify a unique name for each host.

For example:

10.154.48.120 server01

10.154.48.121 server02

10.154.48.122 server03

- Step 3** Download **client.truststore.jks**. On the Kafka console, click the instance. On the instance details page, click **Download** next to **SSL Certificate** in the **Connection** area.

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

- Step 4** Modify the Kafka CLI configuration file based on the [SASL mechanism](#).

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

```
sasl.jaas.config=org.apache.kafka.common.security.plain.PlainLoginModule required \  
username="*****" \  
password="*****";  
sasl.mechanism=PLAIN  
  
security.protocol=SASL_SSL  
ssl.truststore.location={ssl_truststore_path}  
ssl.truststore.password=dms@kafka  
ssl.endpoint.identification.algorithm=
```

Parameter description:

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

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

```
sasl.jaas.config=org.apache.kafka.common.security.scram.ScramLoginModule required \  
username="*****" \  
password="*****";  
sasl.mechanism=SCRAM-SHA-512  
  
security.protocol=SASL_SSL  
ssl.truststore.location={ssl_truststore_path}  
ssl.truststore.password=dms@kafka  
ssl.endpoint.identification.algorithm=
```

Parameter description:

- **username** and **password**: username and password you set when enabling SASL_SSL during Kafka instance creation or when creating a SASL_SSL user.

- **ssl.truststore.location**: path for storing the certificate obtained in [Step 3](#).
- **ssl.truststore.password**: server certificate password, which must be set to **dms@kafka** and cannot be changed.
- **ssl.endpoint.identification.algorithm**: whether to verify the certificate domain name. **This parameter must be left blank, which indicates disabling domain name verification.**

----End

Creating Messages

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

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

Parameter description:

- `{connection-address}`: the address obtained in [Prerequisites](#)
- `{topic-name}`: the name of the topic created for the Kafka instance

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

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

```
[root@ecs-kafka bin]#./kafka-console-producer.sh --broker-list
10.3.196.45:9095,10.78.42.127:9095,10.4.49.103:9095 --topic topic-demo --producer.config ../config/
producer.properties
>Hello
>DMS
>Kafka!
>^C[root@ecs-kafka bin]#
```

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

Retrieving Messages

Run the following command to retrieve messages:

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

Parameter description:

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

The following is an example:

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

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

Follow-Up Procedure

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

[Step 5: Configure Alarm Rules](#)

6 Step 5: Configure Alarm Rules

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

NOTE

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



Table 6-1 Kafka instance metrics to configure alarm rules for

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

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

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

Procedure

- Step 1** Log in to the [Kafka console](#), and select the region where the Kafka instance is located.
- Step 2** Click  next to the Kafka instance name to go to the instance monitoring page of the Cloud Eye console.
- Step 3** Hover the mouse pointer over a metric and click  to create an alarm rule for the metric.
- Step 4** Specify the alarm details.
- For more information about creating alarm rules, see [Creating an Alarm Rule](#).
1. Set the alarm name and description.

2. Specify the alarm policy and alarm severity.

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

Figure 6-1 Setting the alarm policy and alarm severity

* Method Configure manually

* Alarm Policy

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

+ Add Alarm Policy You can add 0 more.

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

----End