Distributed Message Service for Kafka

Getting Started

Issue	01
Date	2024-11-27

Contents

1 Getting Started with Kafka for Message Production and Consumption

This section takes the example of creating a Kafka instance with SASL enabled and accessing it on the client (private network, within a virtual private cloud (VPC)) for message production and consumption to get you quickly started with Distributed Message Service (DMS) for Kafka.

Figure 1-1 Procedure for using DMS for Kafka
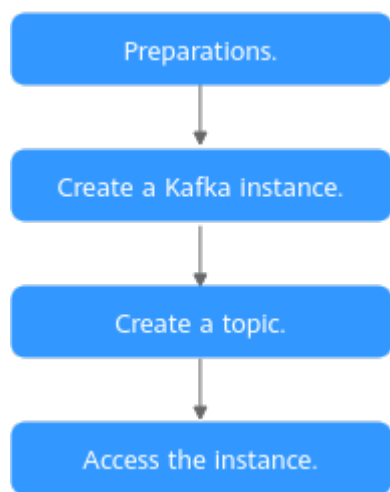
1. **Step 1: Preparations**

   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 producing and consuming messages.

2. **Step 2: Create a Kafka Instance**

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

3. **Step 3: Create a Topic**

   Topics store messages created by producers and subscribed by consumers.

   This section uses the example of creating a topic on the console.

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

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

Step 1: Preparations

Step 1 Register with Huawei Cloud and complete real-name authentication.

For details, see **Registering a HUAWEI ID and Enabling HUAWEI CLOUD Services** and **Real-Name Authentication**.

If you already have a Huawei account and have completed real-name authentication, skip this step.

Step 2 Grant Kafka instance permissions.

To achieve fine-grained management of your cloud resources, create Identity and Access Management (IAM) user groups and users and grant specified permissions to the users. For more information, see **Creating a User and Granting DMS for Kafka Permissions**.

Step 3 Create a VPC and subnet.

Before creating a Kafka instance, ensure that a VPC and a subnet are available. For details about how to create a VPC and a subnet, see **Creating a VPC**.

NOTICE

The VPC must be created in the same region as the Kafka instance.

Step 4 Create a security group and add security group rules.

Before creating a Kafka instance, ensure that a security group is available. For details about how to create a security group, see **Creating a Security Group**.

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

Table 1-1 Security group rules

Direction	Protocol	Port	Source address	Description
Inbound	TCP	9093	0.0.0.0/0	Accessing a Kafka instance over a private network within a VPC (with 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 using the private network within a VPC, you do not need to add the rules described in **Table 1-1**.

Step 5 Construct a client for message production and consumption.

This section uses a Linux elastic cloud server (ECS) as the client. Before creating a Kafka instance, create an ECS with an EIP, install the JDK, configure environment variables, and download an open-source Kafka client.

1. Log in to the console, click ☰ in the upper left corner, click **Elastic Cloud Server** under **Computing**, and then create an ECS.

    For details about how to create an ECS, see **Purchasing an ECS**. If you already have an available ECS, skip this step.

2. Log in to an ECS as user **root**.

3. Install Java JDK and configure the environment variables **JAVA_HOME** and **PATH**.

    a. Download a JDK.

        ☐ 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**.

    b. Decompress the JDK.
        ```
        tar -zxvf jdk-8u321-linux-x64.tar.gz
        ```
        Change **jdk-8u321-linux-x64.tar.gz** to your JDK version.

    c. Open the **.bash_profile** file.
        ```
        vim ~/.bash_profile
        ```

    d. Add the following content:
        ```
        export JAVA_HOME=/root/jdk1.8.0_321
        export PATH=$JAVA_HOME/bin:$PATH
        ```
        Change **/root/jdk1.8.0_321** to the path where you install JDK.

    e. Press **Esc**. Enter the following line and press **Enter**. Save the **.bash_profile** file and exit.
        ```
        :wq
        ```

    f. Run the following command to make the change take effect:
        ```
        source .bash_profile
        ```

    g. Check whether the JDK is installed.
        ```
        java -version
        ```
        If the following message is returned, the JDK is installed.
        ```
        java version "1.8.0_321"
        ```

4. Download an open-source Kafka client.
    ```
    wget https://archive.apache.org/dist/kafka/2.7.2/kafka_2.12-2.7.2.tgz
    ```

5. Run the following command to decompress the package:
    ```
    tar -zxf kafka_2.12-2.7.2.tgz
    ```

----End

Step 2: Create a Kafka Instance

Step 1 Go to the **Buy Instance page**.

Step 2 Specify the basic instance settings. For details, see **Table 1-2**.

Table 1-2 Basic instance settings

Parameter	Description
Billing Mode	Select **Pay-per-use**, which 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 in different regions cannot communicate with each other over an intranet. Select a nearest location for low latency and fast access.
Project	Projects isolate compute, storage, and network resources across geographical regions. For each region, a preset project is available. Select EU-Dublin (default).
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 **AZ1**.
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 (_). Enter **kafka-test**.
Enterprise Project	This parameter is for enterprise users. An enterprise project manages project resources in groups. Enterprise projects are logically isolated. Select **default**.
Specifications	Select **Default** to create a cluster Kafka instance.
Version	Kafka version. Cannot be changed once the instance is created. Select **2.7**.
CPU Architecture	**x86** Retain the default value.
Broker Flavor	Select a broker flavor as required. Select **kafka.2u4g.cluster**.
Brokers	Specify the number of brokers as required. Enter **3**.
Storage Space per Broker	Select the disk type and specify the disk size as required. Total storage space = Storage space per broker × Broker quantity. After the instance is created, you cannot change the disk type. Select **Ultra-high I/O** and enter **100**.

Parameter	Description
Capacity Threshold Policy	Select **Automatically delete**: When the disk reaches the disk capacity threshold (95%), messages can still be produced and consumed, but the earliest 10% of messages will be deleted to ensure sufficient disk space. Use this policy for services intolerant of interruptions. However, data may be lost.

Step 3 Configure the instance network. For details, see **Table 1-3**.

Table 1-3 Configuring instance network

Parameter	Description
VPC	The VPC and subnet cannot be changed once the instance is created. Select the VPC and subnet prepared in **Step 3**.
Security Group	Select the security group prepared in **Step 4**.

Step 4 Set the instance access mode. For details, see **Table 1-4**.

Table 1-4 Setting the instance access mode

Parameter	Description
Kafka SASL_SSL	When this parameter is enabled, SASL authentication is required when a client connects to the Kafka instance. Fixed once the instance is created. Enable it.
SASL/PLAIN	If this parameter is enabled, both the SCRAM-SHA-512 and PLAIN mechanisms are supported. You can select either of them to connect to Kafka instances on the client. Enable it.
Username	Username for the client to connect to Kafka instances. A username should contain 4 to 64 characters, start with a letter, and contain only letters, digits, hyphens (-), and underscores (_). Enter "test".

Parameter	Description
Password	Password for the client to connect to Kafka instances. 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 forwards or backwards.

Step 5 Skip **Advanced Settings**.

Step 6 Click **Buy**.

Step 7 Confirm the instance information, read and agree to the *Huawei Cloud Customer Agreement*, and then submit the request.

Step 8 Return to the **DMS for Kafka** page and check whether the instance has been created.

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

● If the instance is created successfully, its status changes to **Running**.

● If the instance is in the **Failed** state, delete it 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.

Step 9 After the instance is created, click its name to go to the instance details page.

Step 10 In the **Connection** area, view and record the connection address.

Figure 1-2 Kafka instance addresses (private network) for intra-VPC access

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

----End

Step 3: Create a Topic

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

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

Step 3 Click **Create Topic**.

Step 4 Enter the topic name, specify other parameters by referring to **Table 1-5**, and click **OK**.

Table 1-5 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 Cannot be changed once the topic is created. Enter **topic-01**.
Partitions	If the number of partitions is the same as that of consumers, the larger the partitions, the higher the consumption concurrency. Enter **3**.
Replicas	Data is automatically backed up to each replica. When one Kafka broker becomes faulty, data is still available. A higher number of replicas delivers higher reliability. Enter **3**.
Aging Time (h)	How long messages will be preserved in the topic. Messages older than this period cannot be consumed. They will be deleted, and can no longer be consumed. Enter **72**.
Synchronous Replication	Skip it. When this option is disabled, leader replicas are independent from follower replica synchronization. They receive messages and write them to local logs, then immediately send the successfully written ones to the client.
Synchronous Flushing	Skip it. When this option is disabled, messages are produced and stored in memory instead of written to the disk immediately.
Message Timestamp	Select **CreateTime**: time when the producer created the message.
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. Enter **10,485,760**.

----End

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

Step 1 Prepare the file for production and consumption configuration.

1. Log in to a Linux ECS.

2. Download the **client.truststore.jks** certificate and upload it to the **/root** directory on the ECS.

   To obtain the certificate: On the Kafka console, click the Kafka instance to go to the **Basic Information** page. Click **Download** next to **SSL Certificate** in the **Connection** area. Decompress the package to obtain the client certificate file **client.truststore.jks**.

   📖 NOTE

   /root is the path for storing the certificate. Change it to the actual path if needed.

3. Go to the **/config** directory on the Kafka client.
   ```
   cd kafka_2.12-2.7.2/config
   ```

4. Add the following commands in both the **consumer.properties** and **producer.properties** files (PLAIN is used as an example).
   ```
   sasl.jaas.config=org.apache.kafka.common.security.plain.PlainLoginModule required \
   username="**********" \
   password="**********";
   sasl.mechanism=PLAIN

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

   Description:

   – **username** and **password** are specified when enabling SASL_SSL during instance creation.

   – **ssl.truststore.location** is the path for storing the certificate obtained in **Step 1.2**.

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

   – **ssl.endpoint.identification.algorithm** decides whether to verify the certificate domain name. In this example, **leave this parameter blank, which indicates disabling domain name verification**.

Step 2 Go to the **/bin** directory on the Kafka client.
```
cd ../bin
```

Step 3 Produce messages.
```
./kafka-console-producer.sh --broker-list ${connection address} --topic ${topic name} --producer.config ../
config/producer.properties
```

Description:

● *{connection address}*: the connection address obtained in **Step 10**

● *{topic name}*: the topic name obtained in **Step 4**

For example, **192.xxx.xxx.xxx:9093**, **192.xxx.xxx.xxx:9093**, **192.xxx.xxx.xxx:9093** are the connection addresses of the Kafka instance.

After running this command, you can send messages to the Kafka instance by entering the information as prompted and pressing **Enter**. Each line of content will be sent as a message.

```
[root@ecs-kafka bin]#./kafka-console-producer.sh --broker-list
192.xxx.xxx.xxx:9093,192.xxx.xxx.xxx:9093,192.xxx.xxx.xxx:9093  --topic topic-01 --producer.config ../config/
producer.properties
>Hello
>DMS
>Kafka!
>^C[root@ecs-kafka bin]#
```

Press **Ctrl+C** to cancel.

Step 4 Consume messages.

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

Description:

- *{connection address}*: the connection address obtained in **Step 10**
- *{topic name}*: the topic name obtained in **Step 4**

Sample:

```
[root@ecs-kafka bin]#  ./kafka-console-consumer.sh --bootstrap-server
192.xxx.xxx.xxx:9093,192.xxx.xxx.xxx:9093,192.xxx.xxx.xxx:9093 --topic topic-01 --from-beginning --
consumer.config ../config/consumer.properties
Hello
Kafka!
DMS
^CProcessed a total of 3 messages
[root@ecs-kafka bin]#
```

Press **Ctrl+C** to cancel.

----End

Related Information

- Learn more about the **basic concepts of Kafka**.
- Learn more about **Distributed Message Service for Kafka Pricing**.
- To view messages on the console, see **Querying Messages**.
- To view consumption offsets, see **Resetting the Consumer Offset**.
- To view monitoring metrics of a Kafka instance, see **Viewing Metrics**.

2 Common Practices

You can use the common practices provided by DMS for Kafka to meet your service requirements.

Table 2-1 Common practices

Practice	Description
Using MirrorMaker to Synchronize Data Across Clusters	MirrorMaker can synchronize data from the source cluster to the target cluster.
Configuring an Alarm Rule for Accumulated Messages	Configure an alarm rule so that the SMN service will notify you by SMS or email when the number of accumulated messages in a consumer group exceeds the threshold.