TDMQ for Apache Pulsar

Getting Started

Product Documentation

Contents

Getting Started

Use SDK to Receive/Send Ordinary Messages

Resource Creation and Preparation

Last updated：2024-06-28 11:29:49

Overview

This document describes how to create resources such as cluster and topic in the TDMQ for Apache Pulsar console and what you need to do in the console before running a client.

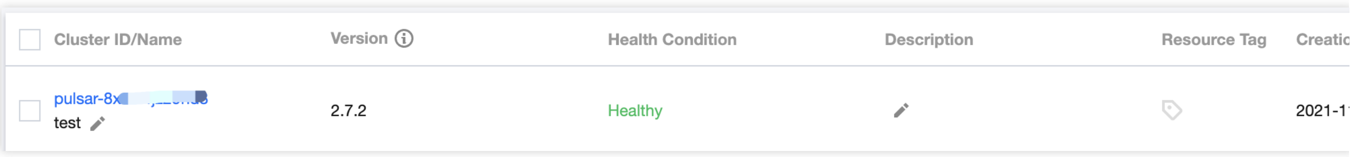
Prerequisites

You have signed up for a Tencent Cloud account as instructed in Signing Up.

Directions

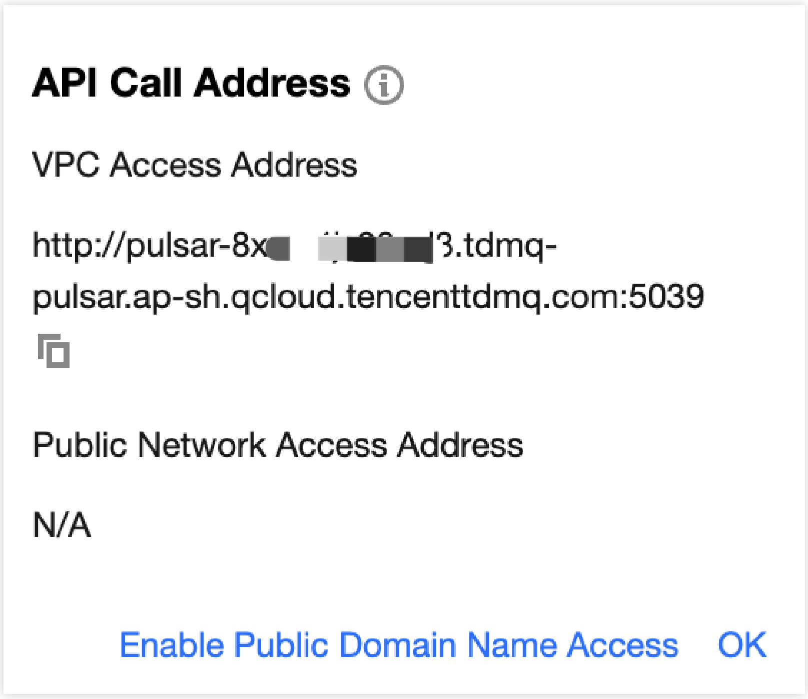
Step 1. Create a cluster and configure the network

1. Log in to the TDMQ for Apache Pulsar console, enter the **Cluster** page, and select the target region.
2. Click **Create Cluster** to create a cluster.
3. Click **Access Address** in the **Operation** column of the created cluster.



Cluster ID/Name	Version	Health Condition	Description	Resource Tag	Creatic
pulsar-8x[redacted] test	2.7.2	Healthy			2021-1

You can get the access address in the following ways:
Clusters on v2.7.1 or later
Clusters on v2.6.1
You can directly get the access address.



API Call Address

VPC Access Address

http://pulsar-8x[redacted]3.tdmq-pulsar.ap-sh.qcloud.tencenttdmq.com:5039

Public Network Access Address

N/A

On the access point list page, click **Create** to create a VPC access point (in the same VPC as the resource on which the client runs).

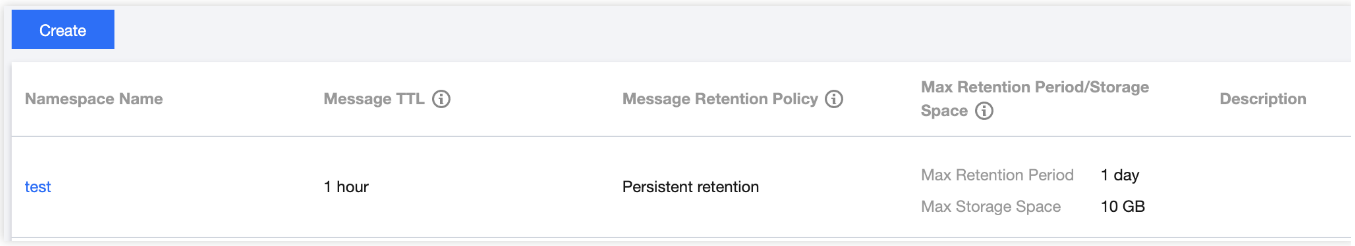
Note:

For more information on clusters, see Cluster Management.

The VPC access address does not support cross-region access. Make sure that the client and virtual cluster are in the same region.

Step 2. Create a namespace

On the Namespace page in the console, select the region and the cluster just created and click **Create** to create a namespace.

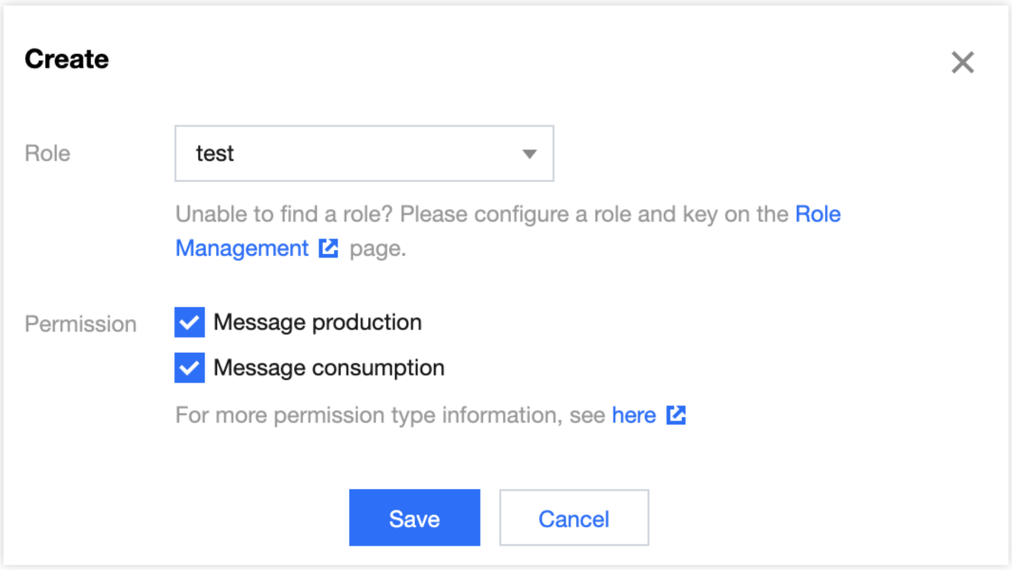


Namespace Name	Message TTL	Message Retention Policy	Max Retention Period/Storage Space		Description
test	1 hour	Persistent retention	Max Retention Period	1 day	
			Max Storage Space	10 GB	

Step 3. Create a role and configure permissions

1. On the Role Management page in the console, select the region and the cluster just created and click **Create** to enter the **Create Role** page.
2. Enter the role name and remarks and click **Submit**.

3. Enter the Namespace page and click **Configure Permission** in the **Operation** column of the namespace just created to open its permission list.
4. On the **Configure Permissions** page, click **Add Role**, add the role just created, and assign the production and consumption permissions.



Create

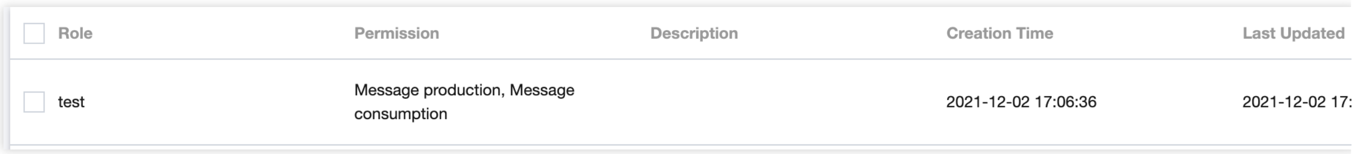
Role: test

Unable to find a role? Please configure a role and key on the Role Management page.

Permission:
- ✓ Message production
- ✓ Message consumption

For more permission type information, see here

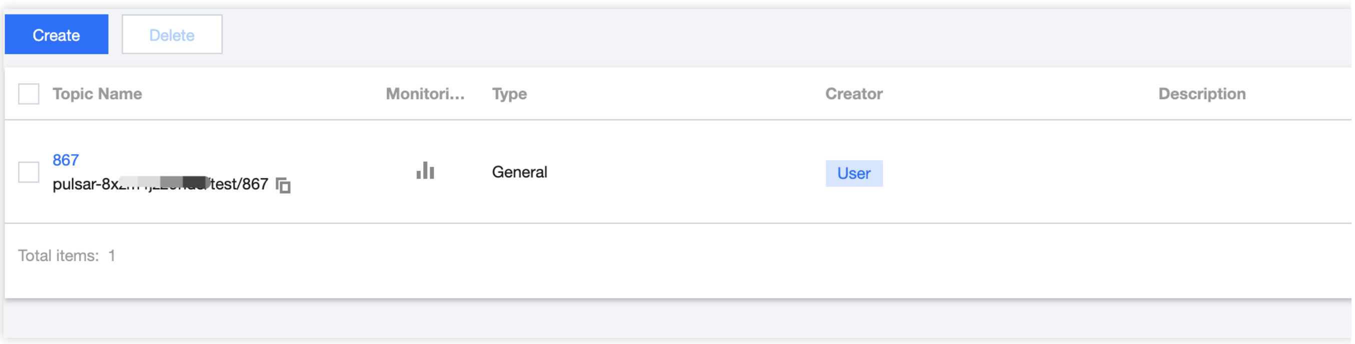
5. If the following is displayed, the permissions are configured successfully.



Role	Permission	Description	Creation Time	Last Updated
test	Message production, Message consumption		2021-12-02 17:06:36	2021-12-02 17:

Step 4. Create a topic and subscription

1. On the Topic page, select the target region, cluster, and namespace and click **Create** to create a topic.
2. Click **Create Subscription** in the **Operation** column to create a subscription for the topic just created.
3. Click **More** > **View Subscription/Consumer** in the **Operation** column to view the subscription just created.



Topic Name	Monitori...	Type	Creator	Description
867 pulsar-8x[redacted]/test/867		General	User	

Total items: 1

Downloading and Running Demo

Last updated：2024-07-04 16:08:41

Overview

This document describes how to download the demo, perform a simple test, and run a client after you purchase the TDMQ for Apache Pulsar and CVM services.

Notes

The following takes the Java client as an example. For clients in other languages, see SDK Overview.

Prerequisites

You have purchased a CVM instance.

Directions

1. Download the demo here and configure relevant parameters.

About Maven dependencies

The dependencies in the `pom.xml` file are configured according to Pulsar's official dependencies. For more information, see Pulsar Java client.

```
<!-- in your <properties> block -->
<pulsar.version>2.7.2</pulsar.version>
<!-- in your <dependencies> block -->
<dependency>
    <groupId>org.apache.pulsar</groupId>
    <artifactId>pulsar-client</artifactId>
    <version>${pulsar.version}</version>
</dependency>
```

Create a client

Access sample for cluster on v2.7.1 or later

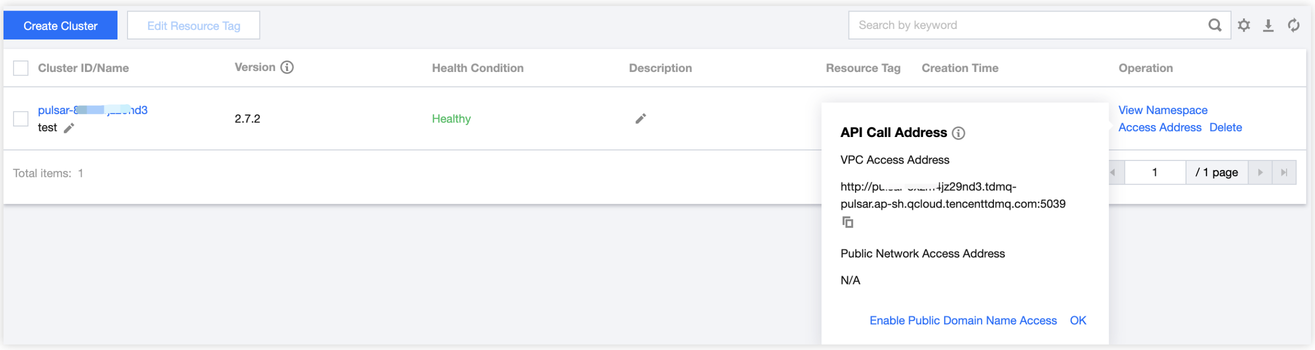
Access sample for cluster on v2.6.1

```
// One Pulsar client corresponds to one client connection
// In principle, one process corresponds to one client. Try to avoid repeated creat
// For the practical tutorial of clients and producers/consumers, see [Client Conne

PulsarClient client = PulsarClient.builder()
        // Replace it with the cluster access address displayed on the **Cluster**
        .serviceUrl("http://pulsar-..tencenttdmq.com:8080")
        // Replace it with the role token displayed on the **Role Management** page
        .authentication(AuthenticationFactory.token("eyJr"))
        .build();

System.out.println(">> pulsar client created.");
```

`serviceUrl` is the access address, which can be viewed and copied on the Cluster page in the console.



Cluster ID/Name	Version	Health Condition	Description	Resource Tag	Creation Time	Operation
pulsar-[redacted]nd3 test	2.7.2	Healthy				View Namespace Access Address Delete

Total items: 1

1 / 1 page

API Call Address

VPC Access Address

http://pu[redacted]4jz29nd3.tdmq-pulsar.ap-sh.qcloud.tencenttdmq.com:5039

Public Network Access Address

N/A

Enable Public Domain Name Access OK

`token` is the role token, which can be copied on the **Role Management** page.

Note:

Token leakage may lead to data leakage; therefore, you should keep your token confidential.

```
PulsarClient client = PulsarClient.builder()
    .serviceUrl("pulsar://...:6000/")// Access address, which can be copied from th
    .listenerName("custom:pulsar-/vpc-/subnet-")// Replace the value of `custom:` w
    .authentication(AuthenticationFactory.token("eyJr"))// Replace it with the role
    .build();
System.out.println(">> pulsar client created.");
```

`serviceUrl` is the access address, which can be viewed and copied on the Cluster > **Access Point** page in the console.

`listenerName` is the `custom:` plus the route ID (NetModel), which can be viewed and copied on the Cluster > **Access Point** page in the console.
`token` is the role token, which can be copied on the **Role Management** page.

Note:

Token leakage may lead to data leakage; therefore, you should keep your token confidential.

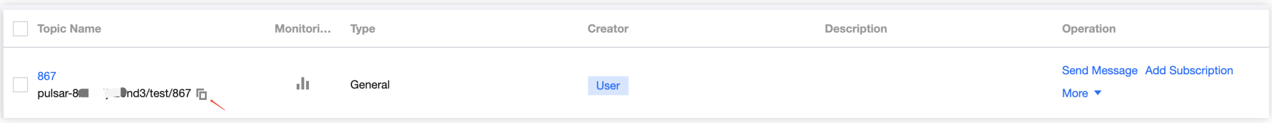
Create a consumer process

```
Consumer<byte[]> consumer = client.newConsumer()
                // Complete path of the topic in the format of `persistent://cluste
```

```
                    .topic("persistent://pulsar-****/namespace/topicName")
                    // You need to create a subscription on the topic details page in t
                    .subscriptionName("subscriptionName")
                    // Declare the exclusive mode as the consumption mode
                    .subscriptionType(SubscriptionType.Exclusive)
                    // Configure consumption starting at the earliest offset; otherwise
                    .subscriptionInitialPosition(SubscriptionInitialPosition.Earliest)
                    .subscribe();
          System.out.println(">> pulsar consumer created.");
```

Note:

You need to enter the complete path of the topic name, i.e., `persistent://clusterid/namespace/Topic` , where the `clusterid/namespace/topic` part can be copied directly from the Topic page in the console.



Topic Name	Monitori...	Type	Creator	Description	Operation
867 pulsar-8[redacted]nd3/test/867		General	User		Send Message Add Subscription More

You need to enter the subscription name in the `subscriptionName` parameter, which can be viewed on the **Consumption Management** page.

Create a producer process

```
Producer<byte[]> producer = client.newProducer()
                // Complete path of the topic in the format of `persistent://cluste
                .topic("persistent://pulsar-****/namespace/topicName")
                .create();
        System.out.println(">> pulsar producer created.");
```

Note:

You need to enter the complete path of the topic name, i.e., `persistent://clusterid/namespace/Topic` , where the `clusterid/namespace/topic` part can be copied directly from the Topic page in the console.

Produce a message

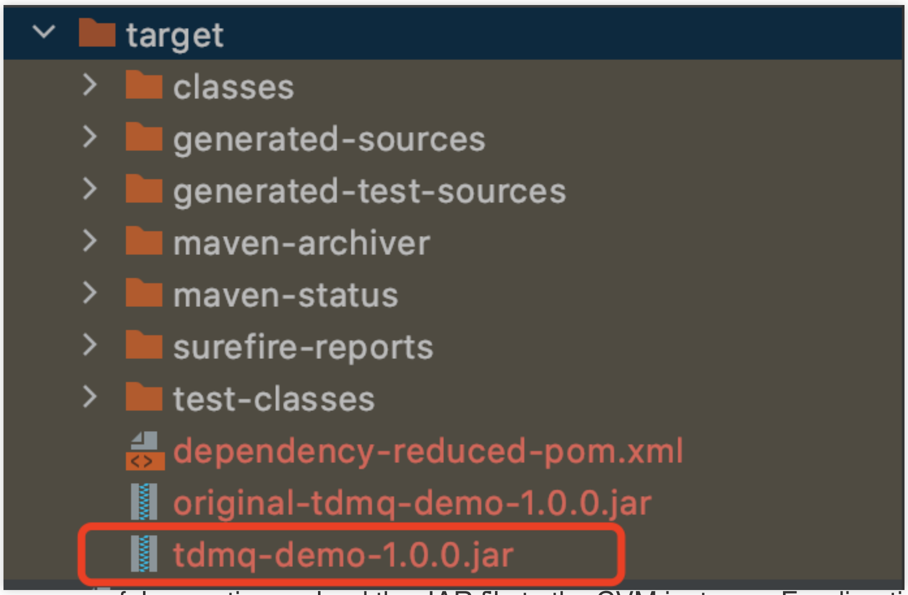
```
for (int i = 0; i < 5; i++) {
            String value = "my-sync-message-" + i;
            // Send the message
            MessageId msgId = producer.newMessage().value(value.getBytes()).send();
            System.out.println("deliver msg " + msgId + ",value:" + value);
        }
        // Close the producer
        producer.close();
```

Consume a message

```
        }
```

```
for (int i = 0; i < 5; i++) {
            // Receive a message corresponding to the current offset
            Message<byte[]> msg = consumer.receive();
            MessageId msgId = msg.getMessageId();
            String value = new String(msg.getValue());
            System.out.println("receive msg " + msgId + ",value:" + value);
            // Messages must be acknowledged once received; otherwise, the offset w
            consumer.acknowledge(msg);
        }
```

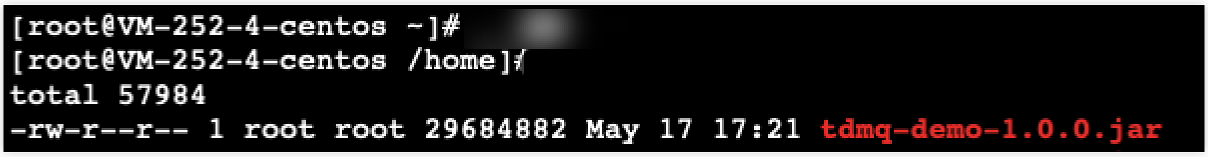
2. Run the `mvn clean package` command in the directory of `pom.xml` or use the features of the IDE to package the entire project and generate an executable JAR file in the `target` directory.



```
target
  classes
  generated-sources
  generated-test-sources
  maven-archiver
  maven-status
  surefire-reports
  test-classes
  dependency-reduced-pom.xml
  original-tdmq-demo-1.0.0.jar
  tdmq-demo-1.0.0.jar
```

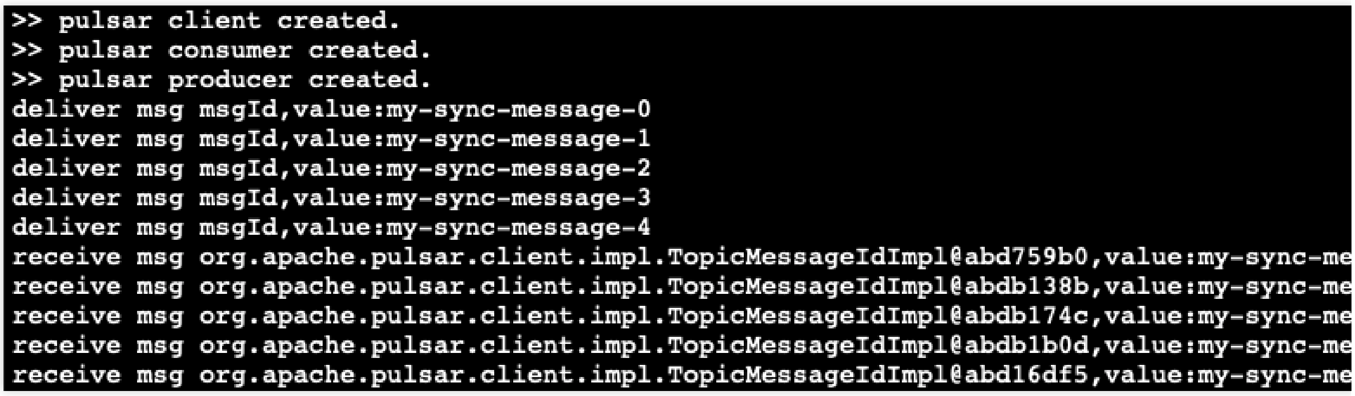
3. After successful execution, upload the JAR file to the CVM instance. For directions, see Copying Local Files to CVMs.

4. Log in to the CVM instance and enter the directory of the JAR file just uploaded, where you can see that the file has been uploaded to the CVM instance.



```
[root@VM-252-4-centos ~]# [redacted]
[root@VM-252-4-centos /home]# [redacted]
total 57984
-rw-r--r-- 1 root root 29684882 May 17 17:21 tdmq-demo-1.0.0.jar
```

Run the `java -jar tdmq-demo-1.0.0.jar` command to run the demo and view the execution logs.

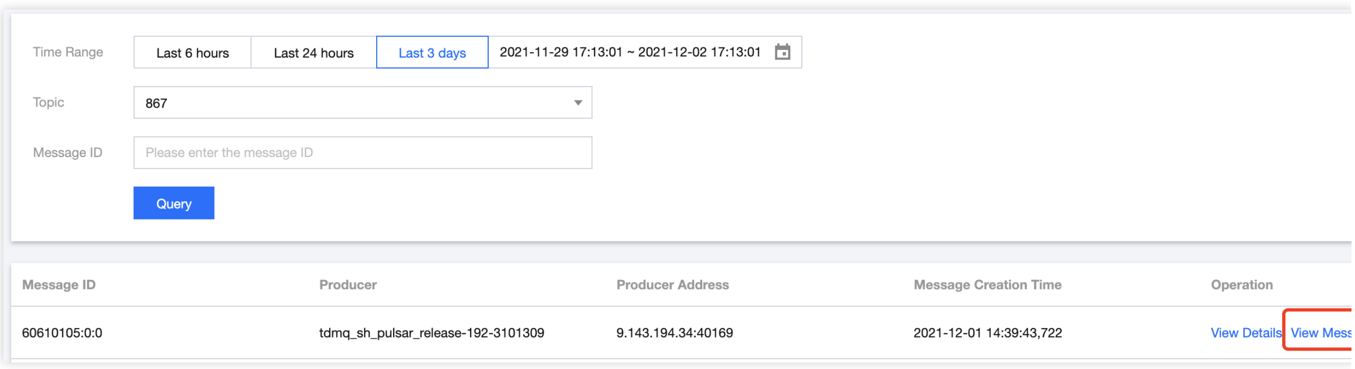


```
>> pulsar client created.
>> pulsar consumer created.
>> pulsar producer created.
deliver msg msgId,value:my-sync-message-0
deliver msg msgId,value:my-sync-message-1
deliver msg msgId,value:my-sync-message-2
deliver msg msgId,value:my-sync-message-3
deliver msg msgId,value:my-sync-message-4
receive msg org.apache.pulsar.client.impl.TopicMessageIdImpl@abd759b0,value:my-sync-me
receive msg org.apache.pulsar.client.impl.TopicMessageIdImpl@abdb138b,value:my-sync-me
receive msg org.apache.pulsar.client.impl.TopicMessageIdImpl@abdb174c,value:my-sync-me
receive msg org.apache.pulsar.client.impl.TopicMessageIdImpl@abdb1b0d,value:my-sync-me
receive msg org.apache.pulsar.client.impl.TopicMessageIdImpl@abd16df5,value:my-sync-me
```

5. Log in to the TDMQ for Apache Pulsar console, click **Topic** > **Topic Name** to enter the consumption management page, and click the triangle below a subscription name to view the production and consumption records.
6. Enter the Message Query page to view the message trace after running the demo.

Note:

You can only query the trace of one single message. If you enable the batch feature in the producer, only the first message in a batch can be queried for its trace.



Time Range: Last 6 hours | Last 24 hours | Last 3 days | 2021-11-29 17:13:01 ~ 2021-12-02 17:13:01

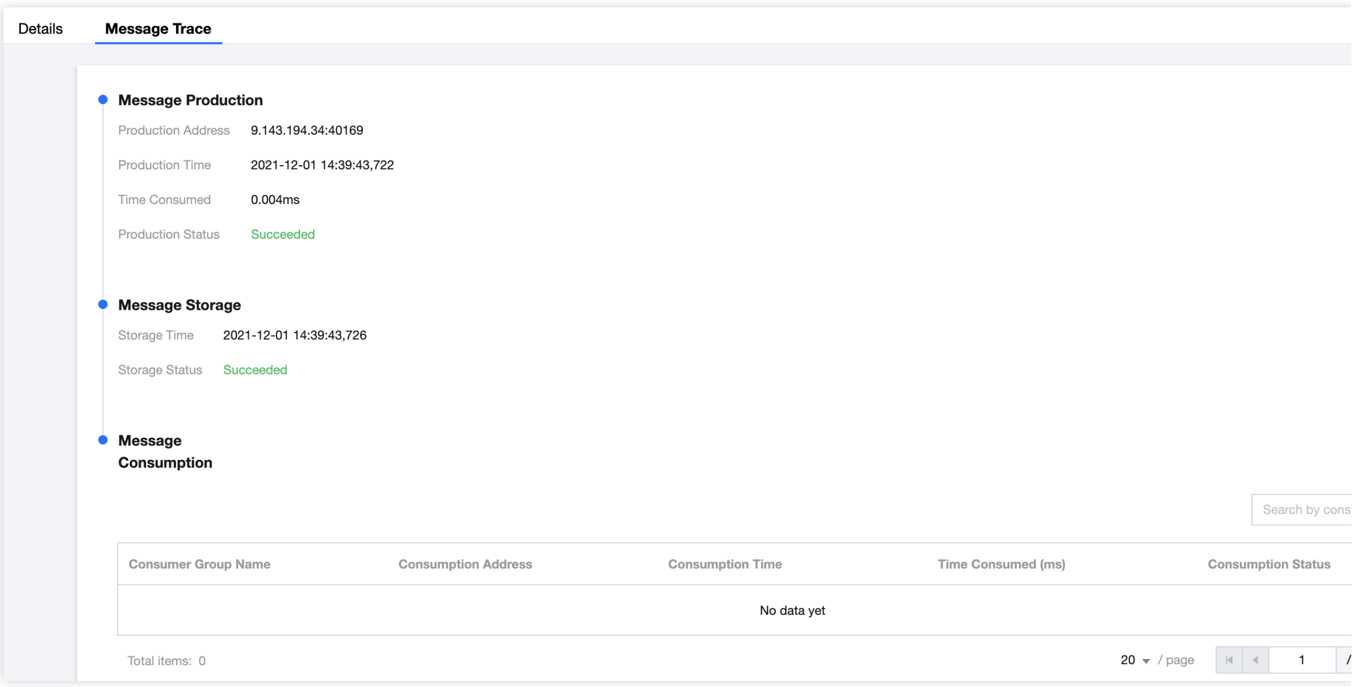
Topic: 867

Message ID: Please enter the message ID

Query

Message ID	Producer	Producer Address	Message Creation Time	Operation
60610105:0:0	tdmq_sh_pulsar_release-192-3101309	9.143.194.34:40169	2021-12-01 14:39:43,722	View Details View Mess

The message trace is as follows:



Details | Message Trace

Message Production

Production Address	9.143.194.34:40169
Production Time	2021-12-01 14:39:43,722
Time Consumed	0.004ms
Production Status	Succeeded

Message Storage

Storage Time	2021-12-01 14:39:43,726
Storage Status	Succeeded

Message Consumption

Search by cons

Consumer Group Name	Consumption Address	Consumption Time	Time Consumed (ms)	Consumption Status
No data yet				

Total items: 0 20 / page 1 /