

Alibaba Cloud

Message Queue for Apache
Kafka
Quick-start

Document Version: 20201110

Legal disclaimer

Alibaba Cloud reminds you to carefully read and fully understand the terms and conditions of this legal disclaimer before you read or use this document. If you have read or used this document, it shall be deemed as your total acceptance of this legal disclaimer.

1. You shall download and obtain this document from the Alibaba Cloud website or other Alibaba Cloud-authorized channels, and use this document for your own legal business activities only. The content of this document is considered confidential information of Alibaba Cloud. You shall strictly abide by the confidentiality obligations. No part of this document shall be disclosed or provided to any third party for use without the prior written consent of Alibaba Cloud.
2. No part of this document shall be excerpted, translated, reproduced, transmitted, or disseminated by any organization, company or individual in any form or by any means without the prior written consent of Alibaba Cloud.
3. The content of this document may be changed because of product version upgrade, adjustment, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and an updated version of this document will be released through Alibaba Cloud-authorized channels from time to time. You should pay attention to the version changes of this document as they occur and download and obtain the most up-to-date version of this document from Alibaba Cloud-authorized channels.
4. This document serves only as a reference guide for your use of Alibaba Cloud products and services. Alibaba Cloud provides this document based on the "status quo", "being defective", and "existing functions" of its products and services. Alibaba Cloud makes every effort to provide relevant operational guidance based on existing technologies. However, Alibaba Cloud hereby makes a clear statement that it in no way guarantees the accuracy, integrity, applicability, and reliability of the content of this document, either explicitly or implicitly. Alibaba Cloud shall not take legal responsibility for any errors or lost profits incurred by any organization, company, or individual arising from download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, take responsibility for any indirect, consequential, punitive, contingent, special, or punitive damages, including lost profits arising from the use or trust in this document (even if Alibaba Cloud has been notified of the possibility of such a loss).
5. By law, all the contents in Alibaba Cloud documents, including but not limited to pictures, architecture design, page layout, and text description, are intellectual property of Alibaba Cloud and/or its affiliates. This intellectual property includes, but is not limited to, trademark rights, patent rights, copyrights, and trade secrets. No part of this document shall be used, modified, reproduced, publicly transmitted, changed, disseminated, distributed, or published without the prior written consent of Alibaba Cloud and/or its affiliates. The names owned by Alibaba Cloud shall not be used, published, or reproduced for marketing, advertising, promotion, or other purposes without the prior written consent of Alibaba Cloud. The names owned by Alibaba Cloud include, but are not limited to, "Alibaba Cloud", "Aliyun", "HiChina", and other brands of Alibaba Cloud and/or its affiliates, which appear separately or in combination, as well as the auxiliary signs and patterns of the preceding brands, or anything similar to the company names, trade names, trademarks, product or service names, domain names, patterns, logos, marks, signs, or special descriptions that third parties identify as Alibaba Cloud and/or its affiliates.
6. Please directly contact Alibaba Cloud for any errors of this document.

Document conventions



Style	Description	Example
 Danger	A danger notice indicates a situation that will cause major system changes, faults, physical injuries, and other adverse results.	 Danger: Resetting will result in the loss of user configuration data.
 Warning	A warning notice indicates a situation that may cause major system changes, faults, physical injuries, and other adverse results.	 Warning: Restarting will cause business interruption. About 10 minutes are required to restart an instance.
 Notice	A caution notice indicates warning information, supplementary instructions, and other content that the user must understand.	 Notice: If the weight is set to 0, the server no longer receives new requests.
 Note	A note indicates supplemental instructions, best practices, tips, and other content.	 Note: You can use Ctrl + A to select all files.
>	Closing angle brackets are used to indicate a multi-level menu cascade.	Click Settings > Network > Set network type .
Bold	Bold formatting is used for buttons, menus, page names, and other UI elements.	Click OK .
Courier font	Courier font is used for commands	Run the <code>cd /d C:/window</code> command to enter the Windows system folder.
<i>Italic</i>	Italic formatting is used for parameters and variables.	<code>bae log list --instanceid</code> <i>Instance_ID</i>
[] or [a b]	This format is used for an optional value, where only one item can be selected.	<code>ipconfig [-all -t]</code>
{ } or {a b}	This format is used for a required value, where only one item can be selected.	<code>switch {active stand}</code>

Table of Contents

1.Overview	05
2.Step 1: Obtain the access permissions	06
3.Step 2: Purchase and deploy an instance	07
3.1. Connect Message Queue for Apache Kafka to a VPC	07
4.Step 3: Create resources	09
5.Step 4: Use the SDK to send and subscribe to messages	12
5.1. Use the default endpoint to send and subscribe to messag.....	12

1. Overview

This topic helps you get started with Message Queue for Apache Kafka.

Process

Quick start for Message Queue for Apache Kafka



Procedure

Perform the following steps:

1. [Step 1: Obtain the access permissions](#)
2. Step 2: Purchase and deploy an instance.
 - [Connect Message Queue for Apache Kafka to a VPC](#)
3. [Step 3: Create resources](#)
4. Step 4: Use the SDK to send and subscribe to messages.
 - [Use the default endpoint to send and subscribe to messages](#)

2.Step 1: Obtain the access permissions


To enable the Message Queue for Apache Kafka instance to access resources in other Alibaba Cloud services, you must first grant the Message Queue for Apache Kafka instance the permissions to access resources in your Alibaba Cloud services.

Prerequisites

An Alibaba Cloud account is created and real-name verification is completed. For more information, see [Create your Alibaba Cloud account](#).

Context

To facilitate quick activation of the service, Message Queue for Apache Kafka creates a default Resource Access Management (RAM) user for your account to access Alibaba Cloud resources.

 **Note** RAM is the access control system of Alibaba Cloud. You can control access to your resources by creating RAM users and roles and configuring corresponding permissions for them. For more information about RAM, see [Grant permissions to RAM users](#).

Procedure

1. Log on to the .
2. On the [Cloud Resource Access Authorization](#) page, click **Confirm Authorization Policy**.

Result

Then, the page jumps to the Message Queue for Apache Kafka console.

What's next

Purchase a Message Queue for Apache Kafka instance and deploy it based on the network type.

- [Connect Message Queue for Apache Kafka to a VPC](#)

3.Step 2: Purchase and deploy an instance

3.1. Connect Message Queue for Apache Kafka to a VPC

If you need to connect Message Queue for Apache Kafka only to a virtual private cloud (VPC), you can purchase and deploy a Message Queue for Apache Kafka instance of the VPC type. You cannot change the network type after you select it.

Prerequisites

- Message Queue for Apache Kafka is granted the permission to access resources of other cloud services. For more information, see [Step 1: Obtain the access permissions](#).
- A VPC is created. For more information, see [Create a VPC](#).

Step 1: Purchase a Message Queue for Apache Kafka instance

1. Log on to the [Message Queue for Apache Kafka console](#).
2. In the top navigation bar, select the region where you want to purchase an instance.
3. In the left-side navigation pane, click **Overview**.
4. On the **Overview** page, click **Purchase Instance**.
5. On the page that appears, set **Network** to **VPC**. Select appropriate configurations as required. Then, click **Buy Now**.
6. Confirm and pay for the order.

Step 2: Obtain the VPC information

1. Log on to the [VPC console](#).
2. Select the region where the VPC is deployed.
3. In the left-side navigation pane, click **VSwitches**.
4. On the **VSwitches** page, click the specified instance. View the vSwitch ID and VPC ID on the **VSwitch Details** page.

Step 3: Deploy the instance

1. In the Message Queue for Apache Kafka console, click **Overview** in the left-side navigation pane. Find the instance in the **Undeployed** state and then click **Deploy**.
2. In the **Deploy** dialog box, deploy the instance.
(Optional)
 - i. Select the specified VPC ID from the **VPC ID** drop-down list.
 - ii. Select the specified vSwitch ID. from the **VSwitch ID** drop-down list.
After you select the vSwitch ID, the system automatically selects a zone where the vSwitch is located.

- iii. (Optional) If the instance is the Professional Edition, you can select whether to deploy the instance across zones. Deployment across zones ensures high disaster recovery capabilities and can withstand breakdowns in data centers.

- iv. Click **Deploy**.

The instance enters the **Deploying** state. The instance deployment takes about 10 to 30 minutes.

Step 4: View the endpoint

1. In the Message Queue for Apache Kafka console, click **Instance Details** in the left-side navigation pane. In the **Basic Information** section, view the value of Default Endpoint.

What's next

[Step 3: Create resources](#)

4.Step 3: Create resources

Before you use Message Queue for Apache Kafka to send and subscribe to messages, you must create resources in the Message Queue for Apache Kafka console. Otherwise, you cannot pass authentication or use the management and maintenance functions. Resources here refers to topics and consumer groups.

Prerequisites

Message Queue for Apache Kafka has been purchased and deployed based on the network type:

- [Connect Message Queue for Apache Kafka to a VPC](#)

Context


After the Message Queue for Apache Kafka instance is deployed, you must create topics and consumer groups.

- A topic is the first-level identifier for classifying messages in Message Queue for Apache Kafka. For example, you can create a topic named Topic_Trade for transaction messages. The first step to using Message Queue for Apache Kafka is to create a topic for your application.
- A consumer group identifies a type of consumer. The consumers in a group subscribe to and consume a type of message with the same consumption logic. The relationship between consumer groups and topics is N:N. One consumer group can subscribe to multiple topics, and one topic can be subscribed to by multiple consumer groups.

Step 1: Create a topic

Perform the following steps to create a topic:

1. Log on to the [Message Queue for Apache Kafka console](#).
2. In the top navigation bar, select a region.

 **Notice** You must create the topic in the region where the application is located. This is the region where the Elastic Compute Service (ECS) instance for the application is deployed. A topic cannot be used across regions. For example, if a topic is created in the China (Beijing) region, the message producer and consumer must run on ECS instances in the China (Beijing) region.


3. In the left-side navigation pane, click **Topics**.
4. On the **Topics** page, select an instance and click **Create Topic**.
5. In the **Create Topic** dialog box, set topic properties and then click **Create**.

- o For Standard Edition instances:

In the **Create Topic** dialog box, enter the topic name, select an instance, enter the description and tags, select the number of partitions, and then click **Create**.

- o For Professional Edition instances:

In the **Create Topic** dialog box, enter the topic name, select an instance, enter the description and tags, and select the number of partitions. Click **Advanced Settings**, select **Storage Engine** and **Message Type**, and then click **Create**.

Parameter	Description
Storage Engine	<p>Message Queue for Apache Kafka supports the following storage engines:</p> <ul style="list-style-type: none"> ■ Cloud Storage: accesses Alibaba Cloud disks at the underlying layer and stores three replicas in distributed mode. This storage engine features low latency, high performance, data persistence, and high reliability. ■ Local Storage: stores three replicas in distributed mode as an Apache Kafka in-sync replica (ISR) set. <div style="border: 1px solid #add8e6; padding: 10px; margin: 10px 0;"> <p> Notice Only Message Queue for Apache Kafka Professional Edition instances running Apache Kafka 2.2.0 support local storage. For more information about how to upgrade an instance edition, see Upgrade the instance version.</p> </div> <p>For more information about storage engines, see Storage engine comparison.</p>
cleanup.policy	<p>If you select local storage, you must configure a log cleanup policy. Message Queue for Apache Kafka supports the following log cleanup policies:</p> <ul style="list-style-type: none"> ■ delete: the default message cleanup policy. If the remaining disk space is sufficient, messages are retained for the maximum retention period. If disk usage exceeds 85%, earlier messages are deleted to ensure service availability. ■ compact: the Apache Kafka log compaction policy. In log compaction, if multiple messages have the same key, only the message with the latest key value is retained. This policy is applicable to scenarios where the system recovers from a system failure or the cache is reloaded after a system restart. For example, when you use Kafka Connect or Confluent Schema Registry, you must store the system status information or configuration information in a log-compacted topic. <div style="border: 1px solid #add8e6; padding: 10px; margin: 10px 0;"> <p> Notice Log-compacted topics are generally used only in some ecosystem components, such as Kafka Connect or Confluent Schema Registry. Do not enable log compaction on the topics for sending and subscribing to messages in other components. For more information, see Message Queue for Apache Kafka demos.</p> </div>

Parameter	Description
Message Type	<p>Message Queue for Apache Kafka supports the following message types:</p> <ul style="list-style-type: none">■ Normal message: Messages with the same key are stored in the same partition in the order they are sent. If an instance in the cluster fails, the messages may be out of order.■ Partitionally ordered message: Messages with the same key are stored in the same partition in the order they are sent. If an instance in the cluster fails, the messages are still stored in the same partition in the order they are sent. However, some messages in the partition cannot be sent until the partition is restored.

After the topic is created, it appears in the list on the **Topics** page.

Step 2: Create a consumer group

Perform the following steps to create a consumer group:

1. In the left-side navigation pane of the Message Queue for Apache Kafka console, click **Consumer Groups**.
2. On the **Consumer Groups** page, click an instance and then click **Create Consumer Group**.
3. In the **Create Consumer Group** dialog box, enter the consumer group name, select an instance, enter tags, and then click **Create**.

After the consumer group is created, it appears in the list on the **Consumer Groups** page.

What's next

Use SDKs to send and subscribe to messages based on the network type:

- [Use the default endpoint to send and subscribe to messages](#)

5. Step 4: Use the SDK to send and subscribe to messages

5.1. Use the default endpoint to send and subscribe to messages

This topic describes how a Java client uses the SDK for Java to connect to the default endpoint of Message Queue for Apache Kafka and send and subscribe to messages in a virtual private cloud (VPC).


Prerequisites

- [Step 3: Create resources](#)
- JDK 1.8 or later is installed. For more information, see [Download JDK](#).
- Maven 2.5 or later is installed. For more information, see [Download Maven](#).

Install Java dependencies

1. Add the following dependencies to the *pom.xml* file:

```
<dependency>
  <groupId>org.apache.kafka</groupId>
  <artifactId>kafka-clients</artifactId>
  <version>0.10.2.2</version>
</dependency>
<dependency>
  <groupId>org.slf4j</groupId>
  <artifactId>slf4j-log4j12</artifactId>
  <version>1.7.6</version>
</dependency>
```

 **Note** We recommend that you keep the version of the client consistent with that of the broker. That is, the client library version must be consistent with the major version of the Message Queue for Apache Kafka instance. You can obtain the major version of the Message Queue for Apache Kafka instance on the [Instance Details](#) page in the Message Queue for Apache Kafka console.

Prepare configurations

1. Create a Log4j configuration file *log4j.properties*.

```
# Licensed to the Apache Software Foundation (ASF) under one or more
# contributor license agreements. See the NOTICE file distributed with
# this work for additional information regarding copyright ownership.
# The ASF licenses this file to You under the Apache License, Version 2.0
# (the "License"); you may not use this file except in compliance with
# the License. You may obtain a copy of the License at
#
# http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.

log4j.rootLogger=INFO, STDOUT

log4j.appender.STDOUT=org.apache.log4j.ConsoleAppender
log4j.appender.STDOUT.layout=org.apache.log4j.PatternLayout
log4j.appender.STDOUT.layout.ConversionPattern=[%d] %p %m (%c)%n
```

2. Create a Kafka configuration file.

```
## Set the endpoint to the default endpoint on the Instance Details page in the Message Queue for Apache Kafka console.
bootstrap.servers=xxxxxxxxxxxxxxxxxxxxxxxxx

## Configure the topic. You can create the topic in the Message Queue for Apache Kafka console.
topic=alikafka-topic-demo

## Configure the consumer group. You can create the consumer group in the Message Queue for Apache Kafka console.
group.id=CID-consumer-group-demo
```

3. Create a profile loader named *JavaKafkaConfigurer.java*.

```
public class JavaKafkaConfigurer {
    private static Properties properties;
    public synchronized static Properties getKafkaProperties() {
        if (null != properties) {
            return properties;
        }
        // Obtain the content of the kafka.properties file.
        Properties kafkaProperties = new Properties();
        try {
            kafkaProperties.load(KafkaProducerDemo.class.getClassLoader().getResourceAsStream("kafka.pr
operties"));
        } catch (Exception e) {
            // If the file cannot be loaded, exit the program.
            e.printStackTrace();
        }
        properties = kafkaProperties;
        return kafkaProperties;
    }
}
```

SDKs for other languages

For information about how to use the SDKs for other languages, see [Overview](#).