

消息队列 RocketMQ 版

快速入门



腾讯云

【 版权声明 】

©2013–2025 腾讯云版权所有

本文档（含所有文字、数据、图片等内容）完整的著作权归腾讯云计算（北京）有限责任公司单独所有，未经腾讯云事先明确书面许可，任何主体不得以任何形式复制、修改、使用、抄袭、传播本文档全部或部分内容。前述行为构成对腾讯云著作权的侵犯，腾讯云将依法采取措施追究法律责任。

【 商标声明 】



及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。未经腾讯云及有关权利人书面许可，任何主体不得以任何方式对前述商标进行使用、复制、修改、传播、抄录等行为，否则将构成对腾讯云及有关权利人商标权的侵犯，腾讯云将依法采取措施追究法律责任。

【 服务声明 】

本文档意在向您介绍腾讯云全部或部分产品、服务的当时的相关概况，部分产品、服务的内容可能不时有所调整。您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或默示的承诺或保证。

【 联系我们 】

我们致力于为您提供个性化的售前购买咨询服务，及相应的技术售后服务，任何问题请联系 4009100100或 95716。

文档目录

快速入门

快速入门概述

创建资源

5.x 集群使用 5.x SDK 收发普通消息

5.x 集群使用 4.x SDK 收发普通消息

4.x 集群使用 4.x SDK 收发普通消息

运行 RocketMQ 客户端（可选）

快速入门

快速入门概述

最近更新时间：2025-06-10 16:17:02

本文为您介绍快速使用消息队列 RocketMQ 版实现普通消息收发的流程和注意事项。

使用说明

- 消息队列 RocketMQ 版支持的消息类型有普通消息、定时与延时消息、顺序消息和事务消息四种，本文以介绍收发普通消息为例，其他类型的消息介绍请参见 [消息类型](#)。

❗ 说明：

不同消息类型的 Topic 不能混用，因此您创建的普通消息的 Topic，不能用于收发其他类型的消息。

- 消息队列 RocketMQ 版支持 TCP 协议和 HTTP 协议接入，因此，建议您分别为两种协议创建对应类型的 Group，本文以介绍 TCP 协议为例。

❗ 说明：

若多个消费者使用同一个 Group 消费消息，其中部分消费者使用 TCP 协议，部分消费者使用 HTTP 协议，可能会导致消费失败、部分消息重复或丢失。

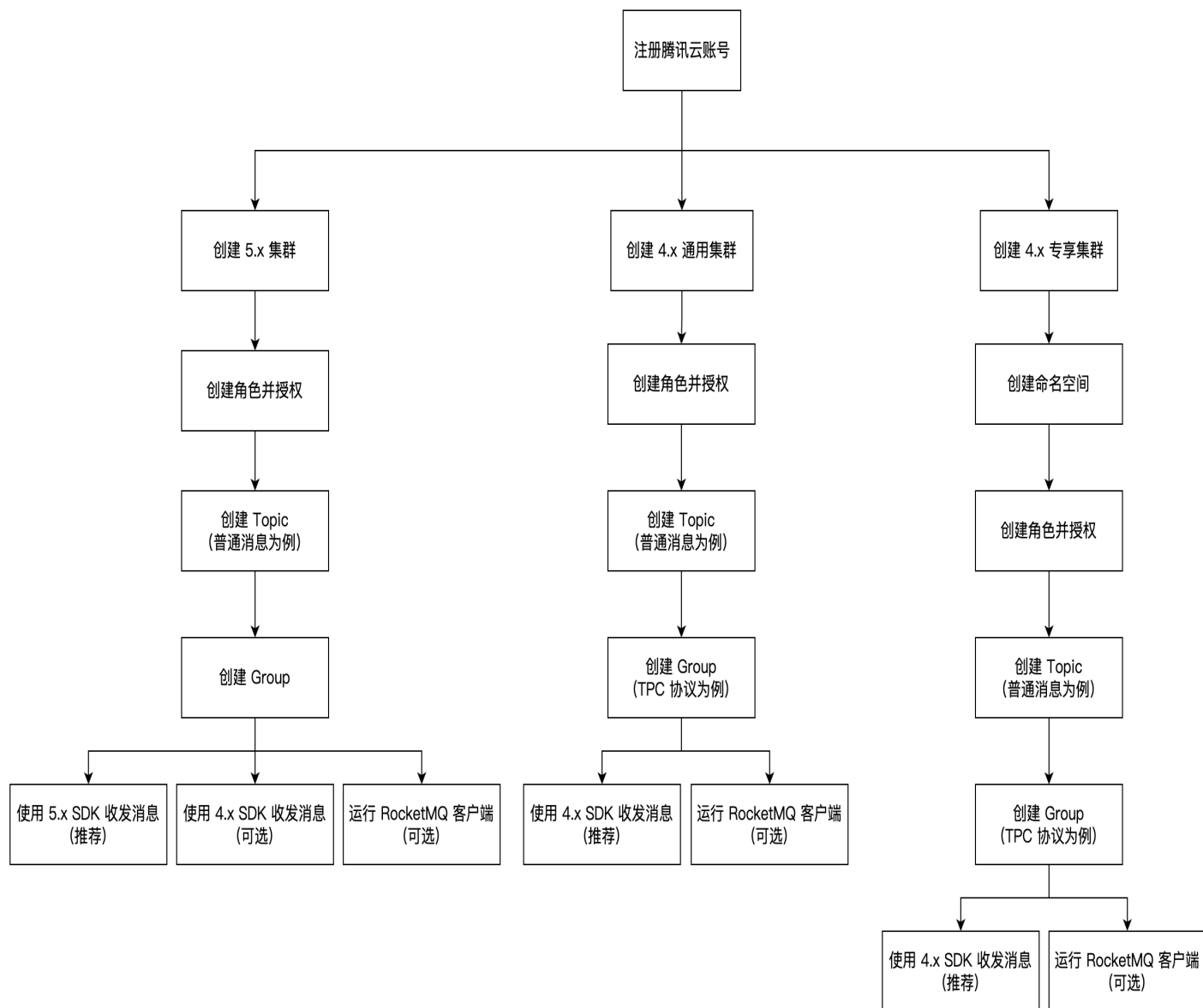
- TCP 协议和 HTTP 协议均支持公网和 VPC 网络接入地址，生产环境默认推荐 VPC 网络，公网接入地址默认不开通，如需开通公网访问，专享集群可以通过 [调整公网带宽](#) 来开启/关闭公网访问。建议公网访问只在测试、调试等不影响生产环境的场景下使用。

❗ 说明：

TCP 协议和 HTTP 协议在各地域均可提供支持，若您当前的实例所在地域未支持 HTTP 协议且有使用需求，可以 [提交工单](#) 申请。

使用流程

根据集群类型的不同，快速入门的操作流程也有所差异，使用消息队列 RocketMQ 版收发消息的流程如下图所示。



创建资源

最近更新时间：2025-06-10 16:17:02

操作场景

在使 SDK 收发消息前，您需要在消息队列 RocketMQ 版控制台中创建集群、Topic、Group 等资源，运行客户端时需要配置相关的资源信息。

前提条件

已 [注册腾讯云账号](#)。

操作步骤

5.x 集群

步骤1. 新建集群并获取接入点信息

1.

登录 [RocketMQ 控制台](#)，进入[集群管理](#)页面，选择好目标地域。
2.

单击[新建集群](#)，选择好集群规格，单击[立即购买](#)，创建一个集群。

您可以参考如下示例值快速创建一个集群。具体参数的含义说明，请参见 [新建集群](#)。

参数	填写示例
集群版本	选择 5.x
计费模式	5.x 架构集群当前支持按量计费和包年包月两种模式。
地域	选择与您的业务最靠近的地域，处于不同地域的云产品内网不通，购买后不能更换，请您谨慎选择。例如，广州地域的云服务器无法通过内网访问上海地域的集群。
部署方式	保持默认值。支持单可用区部署和多可用区部署两种方式，多可用区部署可选择两个可用区进行部署。
集群规格	基础版。当前支持标准版、专业版和铂金版，不同集群类型在性能规格和功能上有差异，详细请参见 产品系列 。
TPS 规格	保持默认值。TPS 规格包含生产消息和消费消息的总和，单条消息以 4KB 为单位对消息进行折算，特殊类型的消息按照特定比例进行折算。
Topic 个数	保持默认值。

Group 个数	默认 Topic 和 Group 的比例为 1 比 10。
消息保留时间	保持默认值。
私有网络	授权将新购集群接入点域名绑定至私有网络（VPC）。
公网访问	保持关闭。开启公网带宽后会新增单独的费用，开通后可在集群管理页关闭。
集群名称	填写集群名称，3-64个字符，只能包含数字、字母、“-”和“_”。
标签	可不选。标签用于从不同维度对资源分类管理。
协议条款	勾选我已阅读并同意《消息队列 RocketMQ 版服务条款》。

3. 在集群列表页面，单击创建好的集群ID，在集群基本信息页面的接入信息模块，获取集群的接入点信息。

接入信息

私有网络

私有网络

子网

vpc-

subnet-

公网访问

已关闭

内网接入地址

rmq-

.rocketmq.gz.qcloud.tencenttdmq.com:8080

支撑网接入地址

rmq-

a.rocketmq.gz.internal.tencenttdmq.com:8080

关闭

步骤2. 配置集群权限

1. 单击 步骤1 中创建好的集群的“ID”，进入集群基本信息页面。
2. 在页面上方选择集群权限页签，单击添加角色，创建一个角色并为其配置生产和消费权限。

说明：

5.x 集群支持在集群管理页面关闭 ACL 权限开关。但是为了保证客户的集群安全，在关闭 ACL 开关前会首先校验客户的公网开关是否打开，如果已经打开，需要再关闭公网开关后再关闭 ACL 权限。

3. 在集群权限列表页面，获取密钥 AccessKey 和 SecretKey。

集群权限							
添加角色 ACL 权限 ☒ 请输入关键字进行搜索							
角色	AccessKey	SecretKey	权限	说明	创建时间	最近更新时间	操作
	ak 6 455ddca	复制	消费消息, 生产消息	-	2025-05-09 16:35:30	2025-05-09 16:35:30	查看角色 编辑 删除

步骤3. 创建 Topic

1. 在集群详情页面，选择 Topic 页签，进入 Topic 列表页。
2. 单击**新建**，填写好 Topic 名称，消息类型选择**普通消息**，其他选项可保持默认，单击**提交**，创建一个 Topic。

说明：

本文以收发普通消息为例进行说明，因此，您参考上述步骤创建的普通消息的Topic，不能用于收发其他类型的消息。

基本信息

集群监控

Topic

Group

集群权限

新建 (1 / 50)

请输入 Topic 名称进行检索

Q

🔄

⚙️

⌵

☐ Topic 名称	监控	类型 ▾	队列数量	订阅 Group 数	说明	操作
☐ topic1		普通消息	3	-	-	发送测试消息 编辑 删除

共 1 条

20 ▾ 条 / 页

⏪

⏩

1

/ 1 页

⏪

⏩

步骤4. 新建 Group

1. 在集群详情页面，选择 Group 页签，进入 Group 列表页。
2. 单击**新建**，填写好 Group 名称，其他选项可保持默认，单击**提交**，创建一个 Group。

基本信息

集群监控

Topic

Group

集群权限

新建 (1 / 500)

请输入关键字进行搜索

🔍 ⌂ ⚙ ⬇

☐ Group 名称 ⓘ	监控	消费者信息	最大重试次数	投递顺序性 ▼	说明	操作
☐ group1		在线消费者 0 TPS 0 总堆积 0	16	并发投递		重置 offset 编辑 删除

共 1 条

20 ▼ 条 / 页

⏮ ⏪

1

⏩ ⏭

/ 1 页

4.x 集群

步骤1. 新建集群并获取接入点信息

1. 登录 [RocketMQ 控制台](#)，进入[集群管理](#)页面，选择好目标地域。
2. 单击[新建集群](#)，选择好集群规格，单击[立即购买](#)，创建一个集群。

您可以参考如下示例值快速创建一个集群。具体参数的含义说明，请参见 [新建集群](#)。

参数	填写示例
集群版本	选择 4.x
集群类型	● 专享集群：独占物理资源，数据安全，使用几乎无限制。按照不同的节点规格按需自由购买。 ● 通用集群：独占物理资源，数据安全，使用几乎无限制。按照 TPS 范围进行售卖，固定节点规格。
计费模式	当前仅支持包年包月-预付费计费模式。
迁移上云组件	关闭
地域	选择与您的业务最靠近的地域，处于不同地域的云产品内网不通，购买后不能更换，请您谨慎选择。例如，广州地域的云服务器无法通过内网访问上海地域的集群。
部署方式	保持默认值。支持单可用区部署和多可用区部署两种方式，多可用区部署可选择两个可用区进行部署。
集群规格	根据您的业务需求选择不同的集群规格，此处可保持默认值， ● 专享集群按照不同的节点规格进行售卖。 ● 通用集群按照 TPS 范围进行售卖，固定节点规格。
Topic 个数	保持默认值
Group 个数	默认 Topic 和 Group 的比例为 1 比 10 。
私有网络	授权将新购集群接入点域名绑定至私有网络（VPC）。
公网访问	保持关闭。开启公网带宽后会新增单独的费用，开通后可在集群管理页关闭。
集群名称	填写集群名称，3-64个字符，只能包含数字、字母、“-”和“_”。

标签	可不选。标签用于从不同维度对资源分类管理。
协议条款	勾选我已阅读并同意《消息队列 RocketMQ 版服务条款》。

3. 在集群列表页面，单击创建好的集群ID，在集群基本信息页面的接入信息模块，获取集群的接入点信息。

接入信息

私有网络

私有网络

子网

vpc-

subnet-

公网访问

已关闭 

TCP 内网接入地址

http://rocketmq---evvo.rmqnamesrv-5xbgrkqr2evo.rocketmq.gz.qcloud.tencenttdmq.com:9876 

HTTP 内网接入地址

http://rocketmq---evvo.tdmq-rocketmq.rest.ap-gz.qcloud.tencenttdmq.com:5576 

支撑网接入点

已关闭 [打开](#)

步骤2. 配置权限

4.x 通用集群和专享集群在权限设置上的步骤略有不同，您可以参考以下页签查看对应的说明。

通用集群

- 单击 [步骤1](#) 中创建好的集群的“ID”，进入集群基本信息页面。
- 在页面上方选择角色管理页签，单击添加角色，创建一个角色并为其配置生产和消费权限。
- 在集群权限列表页面，获取密钥 AccessKey 和角色 SecretKey。

基本信息 集群监控 Topic Group 角色管理

添加角色

请输入角色名称搜索    

角色 (SecretKey)	密钥 (AccessKey)	权限	说明	创建时间	最近更新时间	操作
  or 复制		生产消息, 消费消息	-	2025-05-20 15:00:15	2025-05-20 15:00:15	查看角色 编辑 删除

专享集群

创建命名空间

1. 单击 [步骤1](#) 中创建好的集群的“ID”，进入集群基本信息页面。
2. 选择顶部的命名空间页签，单击新建，设置好命名空间名称和描述信息，创建一个命名空间。

创建角色并授权

1. 在集群详情页面，选择角色管理页签，单击添加角色，创建一个角色。
2. 在角色管理列表页面，获取密钥 AccessKey 和角色 SecretKey。

角色管理						
基本信息 集群监控 命名空间 Topic Group 角色管理						
添加角色 请输入角色名称搜索 Q ↺ ⚙ ⬇ 🔍						
角色 (SecretKey)	密钥 (AccessKey)	说明	创建时间	最近更新时间	操作	
	复制	-	2024-01-11 17:57:39	2025-03-06 10:59:24	查看角色 编辑 删除	

3. 在页面上方选择命名空间页签，找到刚创建好的命名空间，单击操作栏的配置权限。
4. 在配置权限页面，单击添加角色，为刚刚创建的角色添加生产消费权限。

新建

×

角色

super

找不到角色？请先到[角色管理](#)配置角色及密钥

权限

☒ 生产消息
 ☒ 消费消息

关于权限类型的详细说明请参考[权限说明](#)

保存

取消

步骤3. 创建 Topic

1. 在集群详情页面，选择 Topic 页签，进入 Topic 列表页。
2. 单击新建，填写好 Topic 名称，消息类型选择普通消息，其他选项可保持默认，单击提交，创建一个 Topic。

🔔 说明：

本文以收发普通消息为例进行说明，因此，您参考上述步骤创建的普通消息的Topic，不能用于收发其他类型的消息。

基本信息

集群监控

Topic

Group

集群权限

新建 (1 / 50)

请输入 Topic 名称进行检索

Topic 名称	监控	类型	队列数量	订阅 Group 数	说明	操作
topic1		普通消息	3	-	-	发送测试消息 编辑 删除

共 1 条

20 条 / 页

1 / 1 页

步骤4. 新建 Group

- 在集群详情页面，选择 **Group** 页签，进入 Group 列表页。
- 单击**新建**，填写好 Group 名称，协议类型选择 **TCP**，其他选项可保持默认，单击**提交**，创建一个 Group。

说明：

消息队列 RocketMQ 版支持 HTTP 协议和 TCP 协议，因此，建议您分别为两种协议创建对应类型的 Group，若多个消费者使用同一个 Group 消费消息，其中部分消费者使用 TCP 协议，部分消费者使用 HTTP 协议，可能会导致消费失败、部分消息重复或丢失。

基本信息

集群监控

Topic

Group

角色管理

新建 (2/4010)

请输入关键字进行检索

Group 名称	监控	消费者信息	消费模式	协议类型	说明	操作
		在线消费者 0 TPS 0 总堆积 0	未知	TCP	-	消费者详情 重置位点 编辑 删除

5.x 集群使用 5.x SDK 收发普通消息

最近更新时间：2025-06-10 16:17:02

操作场景

消息队列 RocketMQ 版支持多种语言的 SDK 收发不同类型的消息，本文以调用 Java SDK 为例介绍通过 5.x SDK 连接消息队列 RocketMQ 版服务端实现普通消息收发的操作过程，

前提条件

- [完成资源创建与准备](#)
- [安装1.8或以上版本 JDK](#)
- [安装2.5或以上版本 Maven](#)
- [下载 Demo](#)

操作步骤

步骤1：安装 Java 依赖库

在 Java 项目中引入相关依赖，以 Maven 工程为例，在 pom.xml 添加以下依赖：

```
<dependencies>
    <dependency>
        <groupId>org.apache.rocketmq</groupId>
        <artifactId>rocketmq-client-java</artifactId>
        <version>5.0.6</version>
    </dependency>
    <dependency>
        <groupId>org.slf4j</groupId>
        <artifactId>slf4j-api</artifactId>
        <version>1.7.7</version>
    </dependency>
</dependencies>
```

步骤2. 生产消息

在已创建的 Java 工程中，创建发送普通消息程序并运行。

```
import java.io.IOException;
import java.nio.charset.StandardCharsets;
import java.util.concurrent.CompletableFuture;
```

```
import java.util.concurrent.ExecutorService;
import java.util.concurrent.Executors;
import org.apache.rocketmq.client.apis.ClientConfiguration;
import org.apache.rocketmq.client.apis.ClientException;
import org.apache.rocketmq.client.apis.ClientServiceProvider;
import org.apache.rocketmq.client.apis.SessionCredentialsProvider;
import org.apache.rocketmq.client.apis.StaticSessionCredentialsProvider;
import org.apache.rocketmq.client.apis.message.Message;
import org.apache.rocketmq.client.apis.producer.Producer;
import org.apache.rocketmq.client.apis.producer.SendReceipt;
import org.apache.rocketmq.client.java.example.AsyncProducerExample;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;

public class NormalMessageSyncProducer {
    private static final Logger log =
        LoggerFactory.getLogger(NormalMessageSyncProducer.class);

    private NormalMessageSyncProducer() {
    }

    public static void main(String[] args) throws ClientException,
        IOException {
        final ClientServiceProvider provider =
            ClientServiceProvider.loadService();

        // 添加配置的ak和sk
        String accessKey = "yourAccessKey"; //ak
        String secretKey = "yourSecretKey"; //sk
        SessionCredentialsProvider sessionCredentialsProvider =
            new StaticSessionCredentialsProvider(accessKey, secretKey);

        // 填写腾讯云提供的接入地址
        String endpoints = "rmq-xxx.rocketmq.xxx.tencentttdmq.com:8081";
        ClientConfiguration clientConfiguration =
            ClientConfiguration.newBuilder()
                .setEndpoints(endpoints)
                .enableSsl(false)
                .setCredentialProvider(sessionCredentialsProvider)
                .build();

        String topic = "yourNormalTopic";
        // 通常在一个客户端内无需创建过多的生产者。
        final Producer producer = provider.newProducerBuilder()
```

```
.setClientConfiguration(clientConfiguration)
// 设置主题名，此处的设置非必须，但是推荐设置，以便生产者可以在正式发送
消息前，预先抓取消息路由。
.setTopics(topic)
// 如生产者未初始化可能会报 M {@link ClientException} 的错误。
.build();
// 此处定义消息主体。
byte[] body = "This is a normal message for Apache
RocketMQ".getBytes(StandardCharsets.UTF_8);
String tag = "yourMessageTagA";
final Message message = provider.newMessageBuilder()
    // Set topic for the current message.
    .setTopic(topic)
    // 在 topic 下进行的消息二级分类，区别同一个主题内不同的消息。
    .setTag(tag)
    // 消息键，除消息 ID 外可以区别不同消息的其他途径。
    .setKeys("yourMessageKey-1c151062f96e")
    .setBody(body)
    .build();
try {
    final SendReceipt sendReceipt = producer.send(message);
    log.info("Send message successfully, messageId={}",
sendReceipt.getMessageId());
} catch (Throwable t) {
    log.error("Failed to send message", t);
}
// 发送完成后，如无别的需要可以关闭生产者客户端。
producer.close();
}
```

参数

accessKey

说明

角色密钥，在控制台的集群权限页面 AccessKey 列复制。

基本信息

集群监控

Topic

Group

集群权限

添加角色

ACL 权限

请输入关键字进行搜索

角色	AccessKey	SecretKey	权限	说明	创建时间	最近更新时间	操作
	ak1-455ddca6		消费消息, 生产消息	-	2025-05-09 16:35:30	2025-05-09 16:35:30	查看角色 编辑 删除

secretKey	角色名称，在控制台的集群权限页面 SecretKey 列复制。				
endpoints	集群接入地址，在控制台集群基本信息页面的接入信息模块获取。 接入信息 私有网络 <table><tr><td>私有网络</td><td>子网</td></tr><tr><td>vpc-</td><td>subnet-</td></tr></table> 公网访问 已关闭 内网接入地址 rmq- .rocketmq.gz.qcloud.tencentttdmq.com:8080 支撑网接入地址 rmq- a.rocketmq.gz.internal.tencentttdmq.com:8080 关闭	私有网络	子网	vpc-	subnet-
私有网络	子网				
vpc-	subnet-				

步骤3. 消费消息

在已创建的 Java 工程中，创建订阅普通消息程序并运行。

TDMQ RocketMQ 5.x 系列支持两种消费模式，分别为 Push Consumer 和 Simple Consumer，以下代码示例以 Push Consumer 为例。

```
import java.io.IOException;
import java.util.Collections;
import org.apache.rocketmq.client.apis.ClientConfiguration;
import org.apache.rocketmq.client.apis.ClientException;
import org.apache.rocketmq.client.apis.ClientServiceProvider;
import org.apache.rocketmq.client.apis.SessionCredentialsProvider;
import org.apache.rocketmq.client.apis.StaticSessionCredentialsProvider;
import org.apache.rocketmq.client.apis.consumer.ConsumeResult;
import org.apache.rocketmq.client.apis.consumer.FilterExpression;
import org.apache.rocketmq.client.apis.consumer.FilterExpressionType;
```



```
import org.apache.rocketmq.client.apis.consumer.PushConsumer;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;

public class NormalPushConsumer {
    private static final Logger log =
        LoggerFactory.getLogger(NormalPushConsumer.class);

    private NormalPushConsumer() {}

    public static void main(String[] args) throws ClientException,
        IOException, InterruptedException {
        final ClientServiceProvider provider =
            ClientServiceProvider.loadService();

        // 添加配置的 ak 和 sk
        String accessKey = "yourAccessKey"; //ak
        String secretKey = "yourSecretKey"; //sk
        SessionCredentialsProvider sessionCredentialsProvider =
            new StaticSessionCredentialsProvider(accessKey, secretKey);

        // 填写腾讯云提供的接入地址
        String endpoints = "rmq-xxx.rocketmq.xxxtencenttdmq.com:8081";
        ClientConfiguration clientConfiguration =
            ClientConfiguration.newBuilder()
                .setEndpoints(endpoints)
                .enableSsl(false)
                .setCredentialProvider(sessionCredentialsProvider)
                .build();

        String tag = "*";
        FilterExpression filterExpression = new FilterExpression(tag,
            FilterExpressionType.TAG);

        String consumerGroup = "yourConsumerGroup";
        String topic = "yourTopic";
        // 通常在一个客户端内无需创建过多的消费者。
        PushConsumer pushConsumer = provider.newPushConsumerBuilder()
            .setClientConfiguration(clientConfiguration)
            // 设置消费者组名称。
            .setConsumerGroup(consumerGroup)
            // 设置消费者订阅名称
            .setSubscriptionExpressions(Collections.singletonMap(topic,
                filterExpression))
```

```
.setMessageListener(messageView -> {
    // 处理消息并返回消息消费结果。
    log.info("Consume message={}", messageView);
    return ConsumeResult.SUCCESS;
})
.build();
// 生产环境无需阻塞主线程。
Thread.sleep(Long.MAX_VALUE);
// 消费完成后，如无别的需要可以关闭消费者客户端。
pushConsumer.close();
}
```

参数	说明
accessKey	角色密钥，在控制台的集群权限页面 AccessKey 列复制。 
secretKey	角色名称，在控制台的集群权限页面 SecretKey 列复制。
endpoint	集群接入地址，在控制台集群基本信息页面的接入信息模块获取。 
consumerGroup	消费者组名称，在控制台 Group 管理页面复制。

topic

基本信息

集群监控

Topic

Group

集群权限

新建 (1 / 500)

请输入关键字进行搜索

☐	Group 名称 ①	监控	消费者信息	最大重试次数	投递顺序性 ▾	说明	操作
☐	group1		在线消费者 0 TPS 0 总堆积 0	16	并发投递		重置 offset 编辑 删除

共 1 条

20 条 / 页

Topic 的名称，在控制台 Topic 管理页面复制。

基本信息

集群监控

Topic

Group

集群权限

新建 (1 / 50)

请输入 Topic 名称进行检索

☐	Topic 名称	监控	类型 ▾	队列数量	订阅 Group 数	说明	操作
☐	topic1		普通消息	3	-	-	发送测试消息 编辑 删除

共 1 条

20 条 / 页

步骤4. 查看消息详情

发送完成消息后会得到一个消息 ID（messageID），您可以在控制台的消息查询 > 综合查询页面查询刚刚发送的消息，以及该消息的详情和轨迹等信息。

综合查询

广州

集群

Topic

时间范围

查询方式

消息 ID

近 100 条

近30分钟

近1小时

近6小时

近24小时

2025-05-21 15:06:46 ~ 2025-05-21 15:36:46

查询全部

按消息 ID 查询

按消息 Key 查询

按消息 Tag 查询

15314CC.....58715B285C

查询

批量导出

☐	消息 ID	消息 Tag	消息 Key	生产者地址	消息创建时间	操作
☐	15314CC.....58715B285C				2025-05-21 15:36:21.083	查看详情 查看消息轨迹 消费验证 导出消息

5.x 集群使用 4.x SDK 收发普通消息

最近更新时间：2025-06-10 16:17:02

操作场景

RocketMQ 5.0 提供了全新的基于 gRPC 协议的 5.x SDK，新版本 SDK 更加轻量化，多语言支持更好，建议优先使用。同时，消息队列 RocketMQ 5.x 系列也支持存量业务继续使用 4.x SDK 访问，本文以调用 4.x Java SDK 为例介绍通过开源 SDK 实现普通消息收发的操作过程。

前提条件

- [完成资源创建与准备](#)
- [安装1.8或以上版本 JDK](#)
- [安装2.5或以上版本 Maven](#)
- [下载 Demo](#)

操作步骤

步骤1：安装 Java 依赖库

在 Java 项目中引入相关依赖，以 Maven 工程为例，在 pom.xml 添加以下依赖：

```
<!-- in your <dependencies> block -->
<dependency>
  <groupId>org.apache.rocketmq</groupId>
  <artifactId>rocketmq-client</artifactId>
  <version>4.9.7</version>
</dependency>

<dependency>
  <groupId>org.apache.rocketmq</groupId>
  <artifactId>rocketmq-acl</artifactId>
  <version>4.9.7</version>
</dependency>
```

步骤2. 生产消息

在已创建的 Java 工程中，创建发送普通消息程序并运行。

```
// 实例化消息生产者Producer
DefaultMQProducer producer = new DefaultMQProducer(
```

```
        groupName,
        new AclClientRPCHook(new SessionCredentials(accessKey,
secretKey)) // ACL权限
    );
    // 设置NameServer的地址,地址就是形如xxx.tencentttdmq.com:8080 这样的接入地
    址。
    producer.setNamesrvAddr(nameserver);
    // 启动Producer实例
    producer.start();

    for (int i = 0; i < 10; i++) {
        // 创建消息实例,设置topic和消息内容。
        Message msg = new Message(topic_name, ("Hello RocketMQ " +
i).getBytes(RemotingHelper.DEFAULT_CHARSET));
        // 发送消息
        SendResult sendResult = producer.send(msg);
        System.out.printf("%s%n", sendResult);
    }
```

参数	说明																
accessKey	角色密钥，在控制台的集群权限页面 AccessKey 列复制。 基本信息 集群监控 Topic Group 集群权限 添加角色 ACL 权限 请输入关键字进行搜索 <table><tr><th>角色</th><th>AccessKey</th><th>SecretKey</th><th>权限</th><th>说明</th><th>创建时间</th><th>最近更新时间</th><th>操作</th></tr><tr><td></td><td>ak1-6455ddca</td><td></td><td>消费消息, 生产消息</td><td>-</td><td>2025-05-09 16:35:30</td><td>2025-05-09 16:35:30</td><td>查看角色 编辑 删除</td></tr></table>	角色	AccessKey	SecretKey	权限	说明	创建时间	最近更新时间	操作		ak1-6455ddca		消费消息, 生产消息	-	2025-05-09 16:35:30	2025-05-09 16:35:30	查看角色 编辑 删除
角色	AccessKey	SecretKey	权限	说明	创建时间	最近更新时间	操作										
	ak1-6455ddca		消费消息, 生产消息	-	2025-05-09 16:35:30	2025-05-09 16:35:30	查看角色 编辑 删除										
secretKey	角色名称，在控制台的集群权限页面 SecretKey 列复制。																
nameserver	集群接入地址，在控制台集群基本信息页面的接入信息模块获取。																

接入信息

私有网络

私有网络

子网

vpc-

subnet-

公网访问

已关闭

内网接入地址

rmq- .rocketmq.gz.qcloud.tencenttdmq.com:8080

支撑网接入地址

rmq- a.rocketmq.gz.internal.tencenttdmq.com:8080 关闭

Topic 的名称，在控制台 Topic 管理页面复制。

基本信息

集群监控

Topic

Group

集群权限

新建 (1 / 50)

请输入 Topic 名称进行检索

Topic 名称

监控

类型

队列数量

订阅 Group 数

说明

操作

topic1

普通消息

3

-

-

发送测试消息 编辑 删除

共 1 条

20 条 / 页

topic_name

步骤3. 消费消息

在已创建的 Java 工程中，创建订阅普通消息程序并运行。以下代码示例以 Push Consumer 为例，其他的可以参考更详细 4.x 的使用文档。

```
// 实例化消费者
DefaultMQPushConsumer pushConsumer = new DefaultMQPushConsumer(
    groupName,
    new AclClientRPCHook(new SessionCredentials(accessKey,
secretKey))); //ACL权限
// 设置NameServer的地址
pushConsumer.setNamesrvAddr(nameserver);
// 订阅topic
pushConsumer.subscribe(topic_name, "*");
// 注册回调实现类来处理从broker拉取回来的消息
pushConsumer.registerMessageListener((MessageListenerConcurrently)
(msgs, context) -> {
    // 消息处理逻辑
    System.out.printf("%s Receive New Messages: %s %n",
Thread.currentThread().getName(), msgs);
    // 标记该消息已经被成功消费，根据消费情况，返回处理状态
```

```
return ConsumeConcurrentlyStatus.CONSUME_SUCCESS;

});
// 启动消费者实例
pushConsumer.start();
```

参数	说明																
accessKey	角色密钥，在控制台的集群权限页面 AccessKey 列复制。 基本信息 集群监控 Topic Group 集群权限 添加角色 ACL 权限 请输入关键字进行搜索 <table><tr><th>角色</th><th>AccessKey</th><th>SecretKey</th><th>权限</th><th>说明</th><th>创建时间</th><th>最近更新时间</th><th>操作</th></tr><tr><td></td><td>ak-455ddca</td><td>复制</td><td>消费消息, 生产消息</td><td>-</td><td>2025-05-09 16:35:30</td><td>2025-05-09 16:35:30</td><td>查看角色 编辑 删除</td></tr></table>	角色	AccessKey	SecretKey	权限	说明	创建时间	最近更新时间	操作		ak-455ddca	复制	消费消息, 生产消息	-	2025-05-09 16:35:30	2025-05-09 16:35:30	查看角色 编辑 删除
角色	AccessKey	SecretKey	权限	说明	创建时间	最近更新时间	操作										
	ak-455ddca	复制	消费消息, 生产消息	-	2025-05-09 16:35:30	2025-05-09 16:35:30	查看角色 编辑 删除										
secretKey	角色名称，在控制台的集群权限页面 SecretKey 列复制。																
nameserver	集群接入地址，在控制台集群基本信息页面的接入信息模块获取。 接入信息 私有网络 私有网络 子网 vpc- subnet- 公网访问 已关闭 内网接入地址 rmq-.rocketmq.gz.qcloud.tencenttdmq.com:8080 支撑网接入地址 rmq-a.rocketmq.gz.internal.tencenttdmq.com:8080																
groupName	消费者组名称，在控制台 Group 管理页面复制。 基本信息 集群监控 Topic Group 集群权限 新建 (1 / 500) 请输入关键字进行搜索 <table><tr><th>Group 名称</th><th>监控</th><th>消费者信息</th><th>最大重试次数</th><th>投递顺序性</th><th>说明</th><th>操作</th></tr><tr><td>group1</td><td></td><td>在线消费者 0 TPS 0 总堆积 0</td><td>16</td><td>并发投递</td><td></td><td>重置 offset 编辑 删除</td></tr></table> 共 1 条 20 条 / 页	Group 名称	监控	消费者信息	最大重试次数	投递顺序性	说明	操作	group1		在线消费者 0 TPS 0 总堆积 0	16	并发投递		重置 offset 编辑 删除		
Group 名称	监控	消费者信息	最大重试次数	投递顺序性	说明	操作											
group1		在线消费者 0 TPS 0 总堆积 0	16	并发投递		重置 offset 编辑 删除											

topic_name

Topic 的名称，在控制台 Topic 管理页面复制。

基本信息 集群监控 Topic Group 集群权限

新建 (1 / 50)

请输入 Topic 名称进行检索

Topic 名称	监控	类型	队列数量	订阅 Group 数	说明	操作
topic1		普通消息	3	-	-	发送测试消息 编辑 删除

共 1 条

20 条 / 页

步骤4. 查看消息详情

发送完成消息后会得到一个消息 ID（messageID），您可以在控制台的消息查询 > 综合查询页面查询刚刚发送的消息，以及该消息的详情和轨迹等信息。

综合查询 广州

集群 5.x

Topic

时间范围

近 100 条 近30分钟 近1小时 近6小时 近24小时

 2025-05-21 15:06:46 ~ 2025-05-21 15:36:46

查询方式

查询全部 按消息 ID 查询 按消息 Key 查询 按消息 Tag 查询

消息 ID 15314CC.....58715B285C

查询

批量导出

消息 ID	消息 Tag	消息 Key	生产者地址	消息创建时间	操作
15314C.....5B285C				2025-05-21 15:36:21.083	查看详情 查看消息轨迹 消费验证 导出消息

4.x 集群使用 4.x SDK 收发普通消息

最近更新时间：2025-06-10 16:17:02

操作场景

消息队列 RocketMQ 版支持多种语言的 SDK 收发不同类型的消息，本文以调用 Java SDK 为例介绍通过 4.x SDK 连接消息队列 RocketMQ 版服务端实现普通消息收发的操作过程，

前提条件

- [完成资源创建与准备](#)
- [安装1.8或以上版本 JDK](#)
- [安装2.5或以上版本 Maven](#)
- [下载 Demo](#)

操作步骤

步骤1：安装 Java 依赖库

在 Java 项目中引入相关依赖，以 Maven 工程为例，在 pom.xml 添加以下依赖：

说明：
依赖版本要求 $\geq 4.9.3$ 。

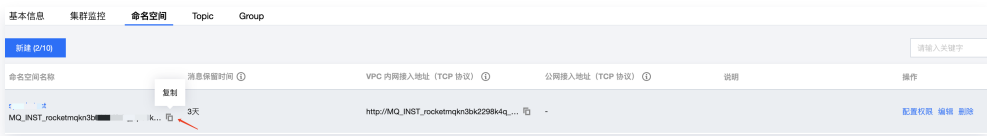
```
<!-- in your <dependencies> block -->
<dependency>
  <groupId>org.apache.rocketmq</groupId>
  <artifactId>rocketmq-client</artifactId>
  <version>4.9.3</version>
</dependency>

<dependency>
  <groupId>org.apache.rocketmq</groupId>
  <artifactId>rocketmq-acl</artifactId>
  <version>4.9.3</version>
</dependency>
```

步骤2：生产消息

1. 创建消息生产者

```
// 实例化消息生产者Producer
DefaultMQProducer producer = new DefaultMQProducer(
    namespace,
    groupName,
    new AclClientRPCHook(new SessionCredentials(accessKey,
secretKey))
    // ACL权限
);
// 设置NameServer的地址
producer.setNamesrvAddr(nameserver);
// 启动Producer实例
producer.start();
```

参数	说明
namespace	命名空间的名称，在控制台命名空间页面复制。如果您使用的是 4.x 通用集群，此处填写集群 ID 即可 
groupName	生产者组名称，在控制台集群管理中 Group 页签中复制。
namesrv	集群接入地址，在控制台集群基本信息页面的接入信息模块获取。

er

接入信息

私有网络

私有网络

子网

vpc-

subnet-

公网访问

已关闭

TCP 内网接入地址

http://rocketmq-vo.rmqlnamesrv-5xbgrkqr2evo.rocketmq.gz.qcloud.tencenttdmq.com:9876

HTTP 内网接入地址

http://rocketmq-vo.tdmq-rocketmq.rest.ap-gz.qcloud.tencenttdmq.com:5576

支撑网接入点

已关闭 [打开](#)

secretKey

角色名称，在控制台的角色管理页面 SecretKey 列复制。

基本信息

集群监控

命名空间

Topic

Group

角色管理

添加角色

请输入角色名称搜索

角色 (SecretKey)	密钥 (AccessKey)	说明	创建时间	最近更新时间	操作
	复制	-	2024-01-11 17:57:39	2025-03-06 10:59:24	查看角色 编辑 删除

accessKey

角色密钥，在控制台的角色管理页面 AccessKey 列复制。

2. 发送消息

发送消息有多种方式：同步发送、异步发送、单向发送等。

同步发送

```
for (int i = 0; i < 10; i++) {
    // 创建消息实例，设置topic和消息内容
    Message msg = new Message(topic_name, "TAG", ("Hello RocketMQ
" + i).getBytes(RemotingHelper.DEFAULT_CHARSET));
    // 发送消息
    SendResult sendResult = producer.send(msg);
    System.out.printf("%s\n", sendResult);
}
```

参数

说明

topic_name

Topic 的名称，在控制台 Topic 管理页面复制。

基本信息

集群监控

Topic

Group

集群权限

新建 (1 / 50)

请输入 Topic 名称进行检索

☐ Topic 名称	监控	类型 ▾	队列数量	订阅 Group 数	说明	操作
☐ topic1		普通消息	3	-	-	发送测试消

共 1 条

20 ▾ 条 / 页

⏮

⏪

1

TAG

用来设置消息的 TAG。

异步发送

```
// 设置发送失败后不重试
producer.setRetryTimesWhenSendAsyncFailed(0);
// 设置发送消息的数量
int messageCount = 10;
final CountDownLatch countDownLatch = new
CountDownLatch(messageCount);
for (int i = 0; i < messageCount; i++) {
    try {
        final int index = i;
        // 创建消息实体，设置topic和消息内容
```

```

        Message msg = new Message(topic_name, "TAG", ("Hello
rocketMq " + index).getBytes(RemotingHelper.DEFAULT_CHARSET));
        producer.send(msg, new SendCallback() {
            @Override
            public void onSuccess(SendResult sendResult)
            {
                // 消息发送成功逻辑
                countDownLatch.countDown();
                System.out.printf("%-10d OK %s %n",
index, sendResult.getMsgId());
            }

            @Override
            public void onException(Throwable e) {
                // 消息发送失败逻辑
                countDownLatch.countDown();
                System.out.printf("%-10d Exception %s
%n", index, e);
                e.printStackTrace();
            }
        });
    } catch (Exception e) {
        e.printStackTrace();
    }
}
countDownLatch.await(5, TimeUnit.SECONDS);

```

参
数

说明

t
o
p
i
c
-
n
a
m
e

Topic 的名称，在控制台 Topic 管理页面复制。

基本信息

集群监控

Topic

Group

集群权限

新建 (1 / 50)

请输入 Topic 名称进行检索

☐ Topic 名称	监控	类型 ▾	队列数量	订阅 Group 数	说明	操作
☐ topic1		普通消息	3	-	-	发送测试消

共 1 条

20 ▾ 条 / 页

1

TAG

用来设置消息的 TAG。

单向发送

```
for (int i = 0; i < 10; i++) {  
    // 创建消息实例，设置topic和消息内容  
    Message msg = new Message(topic_name, "TAG", ("Hello RocketMQ  
" + i).getBytes(RemotingHelper.DEFAULT_CHARSET));  
    // 发送单向消息  
    producer.sendOneway(msg);  
}
```

参数

说明

topic_name

Topic 的名称，在控制台 Topic 管理页面复制。

基本信息

集群监控

Topic

Group

集群权限

新建 (1 / 50)

请输入 Topic 名称进行检索

☐ Topic 名称	监控	类型 ▼	队列数量	订阅 Group 数	说明	操作
☐ topic1		普通消息	3	-	-	发送测试消

共 1 条

20 ▼ 条 / 页

⏮

⏪

1

TAG

用来设置消息的 TAG。

❗ 说明：

批量发送及其他情况可参见 [Demo](#) 或 [社区文档](#)。

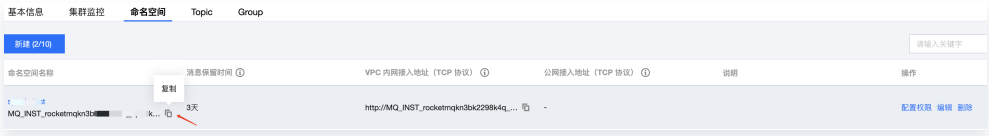
步骤3：消费消息

1. 创建消费者

TDMQ RocketMQ 版支持 push 和 pull 两种消费模式。

push 消费者

```
// 实例化消费者
DefaultMQPushConsumer pushConsumer = new DefaultMQPushConsumer(
    namespace,
    groupName,
    new AclClientRPCHook(new SessionCredentials(accessKey,
secretKey))); //ACL权限
// 设置NameServer的地址
pushConsumer.setNamesrvAddr(nameserver);
```

参数	说明
namespace	命名空间的名称，在控制台命名空间页面复制。如果您使用的是 4.x 通用集群，此处填写集群 ID 
groupName	生产者组名称，在控制台集群管理中 Group 页签中复制。
nameserver	集群接入地址，在控制台集群基本信息页面的接入信息模块获取。

ver

接入信息

私有网络

私有网络

子网

vpc-   

subnet-   

公网访问

已关闭 

TCP 内网接入地址

http://rocketmq-   vo.rmnamesrv-5xbgrkqr2evo.rocketmq.gz.qcloud.tencenttdmq.com:9876 

HTTP 内网接入地址

http://rocketmq-   vo.tdmq-rocketmq.rest.ap-gz.qcloud.tencenttdmq.com:5576 

支撑网接入点

已关闭 [打开](#)

secretKey

角色名称，在控制台的角色管理页面 SecretKey 列复制。

基本信息 集群监控 命名空间 Topic Group 角色管理					
添加角色		请输入角色名称搜索			
角色 (SecretKey)	密钥 (AccessKey)	说明	创建时间	最近更新时间	操作
 	复制	-	2024-01-11 17:57:39	2025-03-06 10:59:24	查看角色

accessKey

角色密钥，在控制台的角色管理页面 AccessKey 列复制。

pull 消费者

```
// 实例化消费者
DefaultLitePullConsumer pullConsumer = new
DefaultLitePullConsumer(
```



```
namespace,
groupName,
    new AclClientRPCHook(new SessionCredentials(accessKey,
secretKey)));
// 设置NameServer的地址
pullConsumer.setNamesrvAddr(nameserver);
// 设置从第一个偏移量开始消费
pullConsumer.setConsumeFromWhere(ConsumeFromWhere.CONSUME_FROM_FIR
ST_OFFSET);
```

参数	说明												
namespace	命名空间的名称，在控制台命名空间页面复制。如果您使用的是 4.x 通用集群，此处填写集群 ID 基本信息 集群监控 命名空间 Topic Group 新建 (2/10) 请输入关键字 <table><tr><th>命名空间名称</th><th>消息保留时间 ①</th><th>VPC 内网接入地址 (TCP 协议) ①</th><th>公网接入地址 (TCP 协议) ①</th><th>说明</th><th>操作</th></tr><tr><td>MQ_INST_rocketmqn3b... (k...)</td><td>3天</td><td></td><td>http://MQ_INST_rocketmqn3b2298k4q...</td><td></td><td>配置权限 编辑 删除</td></tr></table>	命名空间名称	消息保留时间 ①	VPC 内网接入地址 (TCP 协议) ①	公网接入地址 (TCP 协议) ①	说明	操作	MQ_INST_rocketmqn3b... (k...)	3天		http://MQ_INST_rocketmqn3b2298k4q...		配置权限 编辑 删除
命名空间名称	消息保留时间 ①	VPC 内网接入地址 (TCP 协议) ①	公网接入地址 (TCP 协议) ①	说明	操作								
MQ_INST_rocketmqn3b... (k...)	3天		http://MQ_INST_rocketmqn3b2298k4q...		配置权限 编辑 删除								
groupname	生产者组名称，在控制台集群管理中 Group 页签中复制。												
nameserver	集群接入地址，在控制台集群基本信息页面的接入信息模块获取。												

接入信息

私有网络

私有网络	子网
vpc-	subnet-

公网访问

已关闭

TCP 内网接入地址

http://rocketmq-vo.rmqnamesrv-5xbgrkqr2evo.rocketmq.gz.qcloud.tencenttdmq.com:9876

HTTP 内网接入地址

http://rocketmq-vo.tdmq-rocketmq.rest.ap-gz.qcloud.tencenttdmq.com:5576

支撑网接入点

已关闭 [打开](#)

SecretKey

角色名称，在控制台的角色管理页面 SecretKey 列复制。

基本信息	集群监控	命名空间	Topic	Group	角色管理
添加角色		请输入角色名称搜索			
角色 (SecretKey)	密钥 (AccessKey)	说明	创建时间	最近更新时间	操作
	复制	-	2024-01-11 17:57:39	2025-03-06 10:59:24	查看角色

AccessKey

角色密钥，在控制台的角色管理页面 AccessKey 列复制。

说明：

更多消费类型可参见 [Demo](#) 或 [RocketMQ 官方文档](#) 。

2. 订阅消息

根据消费模式不同，订阅方式也有所区别。

push 订阅

```
// 订阅topic
pushConsumer.subscribe(topic_name, "*");
// 注册回调实现类来处理从broker拉取回来的消息
pushConsumer.registerMessageListener((MessageListenerConcurrently)
(msgs, context) -> {
    // 消息处理逻辑
    System.out.printf("%s Receive New Messages: %s %n",
Thread.currentThread().getName(), msgs);
    // 标记该消息已经被成功消费，根据消费情况，返回处理状态
    return ConsumeConcurrentlyStatus.CONSUME_SUCCESS;
});
// 启动消费者实例
pushConsumer.start();
```

参数

说明

topic_name

Topic 的名称，在控制台 Topic 管理页面复制。

基本信息

集群监控

Topic

Group

集群权限

新建 (1 / 50)

请输入 Topic 名称进行检索

☐ Topic 名称	监控	类型 ▾	队列数量	订阅 Group 数	说明
☐ topic1		普通消息	3	-	-

共 1 条

20 条 / 页

"*"

订阅表达式如果为 null 或*表达式表示订阅全部，同时支持 "tag1 || tag2 || tag3" 标识订阅多个

pull 订阅

```
// 订阅topic
pullConsumer.subscribe(topic_name, "*");
// 启动消费者实例
pullConsumer.start();
```

```
try {
    System.out.printf("Consumer Started.%n");
    while (true) {
        // 拉取消息
        List<MessageExt> messageExts = pullConsumer.poll();
        System.out.printf("%s%n", messageExts);
    }
} finally {
    pullConsumer.shutdown();
}
```

参数

topic-name

"*"

"

说明

Topic 的名称，在控制台 Topic 管理页面复制。

基本信息

集群监控

Topic

Group

集群权限

新建 (1 / 50)

请输入 Topic 名称进行检索

☐ Topic 名称	监控	类型 ▾	队列数量	订阅 Group 数	说明
☐ topic1		普通消息	3	-	-

共 1 条

20 ▾ 条 / 页

订阅表达式如果为 null 或*表达式表示订阅全部，同时支持 "tag1 || tag2 || tag3" 标识订阅多个

步骤4. 查看消息详情

发送完成消息后会得到一个消息 ID（messageID），您可以在控制台的消息查询 > 综合查询页面查询刚刚发送的消息，以及该消息的详情和轨迹等信息。

综合查询

广州

集群

Topic

时间范围

近 100 条

近30分钟

近1小时

近6小时

近24小时

2025-05-21 15:06:46 ~ 2025-05-21 15:36:46

查询方式

查询全部

按消息 ID 查询

按消息 Key 查询

按消息 Tag 查询

消息 ID

15314CC.....58715B285C

查询

批量导出

☐	消息 ID	消息 Tag	消息 Key	生产者地址	消息创建时间	操作
☐	15314CC.....58715B285C				2025-05-21 15:36:21.083	查看详情 查看消息轨迹 消费验证 导出消息

运行 RocketMQ 客户端（可选）

最近更新时间：2025-06-10 16:17:02

操作场景

该任务指导您在购买 RocketMQ 服务后，使用 RocketMQ API。在腾讯云服务器上搭建 RocketMQ 环境后，本地下载并解压 RocketMQ 工具包，并对 RocketMQ API 进行简单测试。

操作步骤

步骤1：安装 JDK 环境

1. 检查 Java 安装。

打开终端，执行如下命令：

```
java -version
```

如果输出 Java 版本号，说明 Java 安装成功；如果没有安装 Java，请 [下载安装 Java 软件开发套件（JDK）](#)。

2. 设置 Java 环境。

设置 `JAVA_HOME` 环境变量，并指向您机器上的 Java 安装目录。

以 Java JDK 1.8.0_20 版本为例，操作系统的输出如下：

操作系统	输出
Windows	Set the environment variable JAVA_HOME to C:\Program Files\Java\jdkjdk1.8.0_20
Linux	export JAVA_HOME=/usr/local/java-current
Mac OSX	export JAVA_HOME=/Library/Java/Home

将 Java 编译器地址添加到系统路径中：

操作系统	输出
Windows	将字符串 “;C:\Program Files\Java\jdk1.8.0_20\bin” 添加到系统变量 “Path” 的末尾
Linux	export PATH=\$PATH:\$JAVA_HOME/bin/

Mac OSX not required

使用上面提到的 `java -version` 命令验证 Java 安装。

步骤2：下载 RocketMQ 工具包

下载并解压 RocketMQ 安装包。（[RocketMQ 安装包官网下载地址](#)）

步骤3：RocketMQ API 测试

通过 CLI 命令生产和消费消息，去到 `./benchmark` 目录下。单条消息发送，可以使用云上管控台的发送功能。

1. 打开终端启动消费者。

```
sh consumer.sh -t "topic的名称" -n "rocketmq接入地址" -g "group的名称" -a true -ak xxx -sk yyy
```

说明：

- 将 `rocketmq接入点` 替换成 VPC 网络访问的域名与端口，如果在本地地上测试，不能联通 VPC，可以替换成公网的（注意放开对应的安全组），在控制台实例详情页面的接入方式模块获取。

基本信息	集群监控	Topic	Group	集群权限						
收发 TPS 峰值上限 ① 500	消息堆积数 一条			消息存储空间 GB						
接入信息										
私有网络 <table border="1"> <tr> <td>私有网络</td> <td>子网</td> </tr> <tr> <td>vpc-rorzgplx</td> <td>subnet-5cy9bras</td> </tr> </table>					私有网络	子网	vpc-rorzgplx	subnet-5cy9bras		
私有网络	子网									
vpc-rorzgplx	subnet-5cy9bras									
公网访问带宽 1Mbps(按小时带宽)										
公网安全策略 <table border="1"> <thead> <tr> <th>来源</th> <th>策略</th> <th>备注</th> </tr> </thead> <tbody> <tr> <td>0.0.0.0/0</td> <td>允许</td> <td>-</td> </tr> </tbody> </table> 编辑公网安全策略					来源	策略	备注	0.0.0.0/0	允许	-
来源	策略	备注								
0.0.0.0/0	允许	-								
公网接入地址 <code>rmq-w-.j.rocketmq.gz.public.tencenttdmq.com:8080</code>										
内网接入地址 <code>rmq-w-.j.rocketmq.gz.qcloud.tencenttdmq.com:8080</code>										

- topic：在控制台 Topic 管理 页面获取的 topic 的名称。

- group: 在控制台 **Group 管理** 页面获取的 group 的名称。

2. 另外开一个终端窗口启动生产者。

```
sh producer.sh -t "topic的名称" -w 1 -s 1024 -n "rocketmq接入地址" -a
true -ak xxx -sk yyy -m true
```

说明:

- 将 **rocketmq接入点** 替换成 VPC 网络访问的域名与端口，如果在本地测试，不能联通 VPC，可以替换成公网的（注意放开对应的安全组），在控制台实例详情页面的**接入方式**模块获取。

基本信息

集群监控

Topic

Group

集群权限

收发 TPS 峰值上限 ⓘ

500

消息堆积数

— 条

消息存储空间

— GB

接入信息

私有网络

私有网络	子网
vpc-rorzgplx	subnet-5cy9bras

公网访问带宽

1Mbps(按小时带宽) ✎

公网安全策略

来源	策略	备注
0.0.0.0/0	允许	-

编辑公网安全策略

公网接入地址

rmq-w[模糊]j.rocketmq.gz.public.tencenttdmq.com:8080

内网接入地址

rmq-w[模糊]j.rocketmq.gz.qcloud.tencenttdmq.com:8080

- topic: 将 XXXX 替换成 topic 名称，在控制台 **Topic 管理** 页面获取。

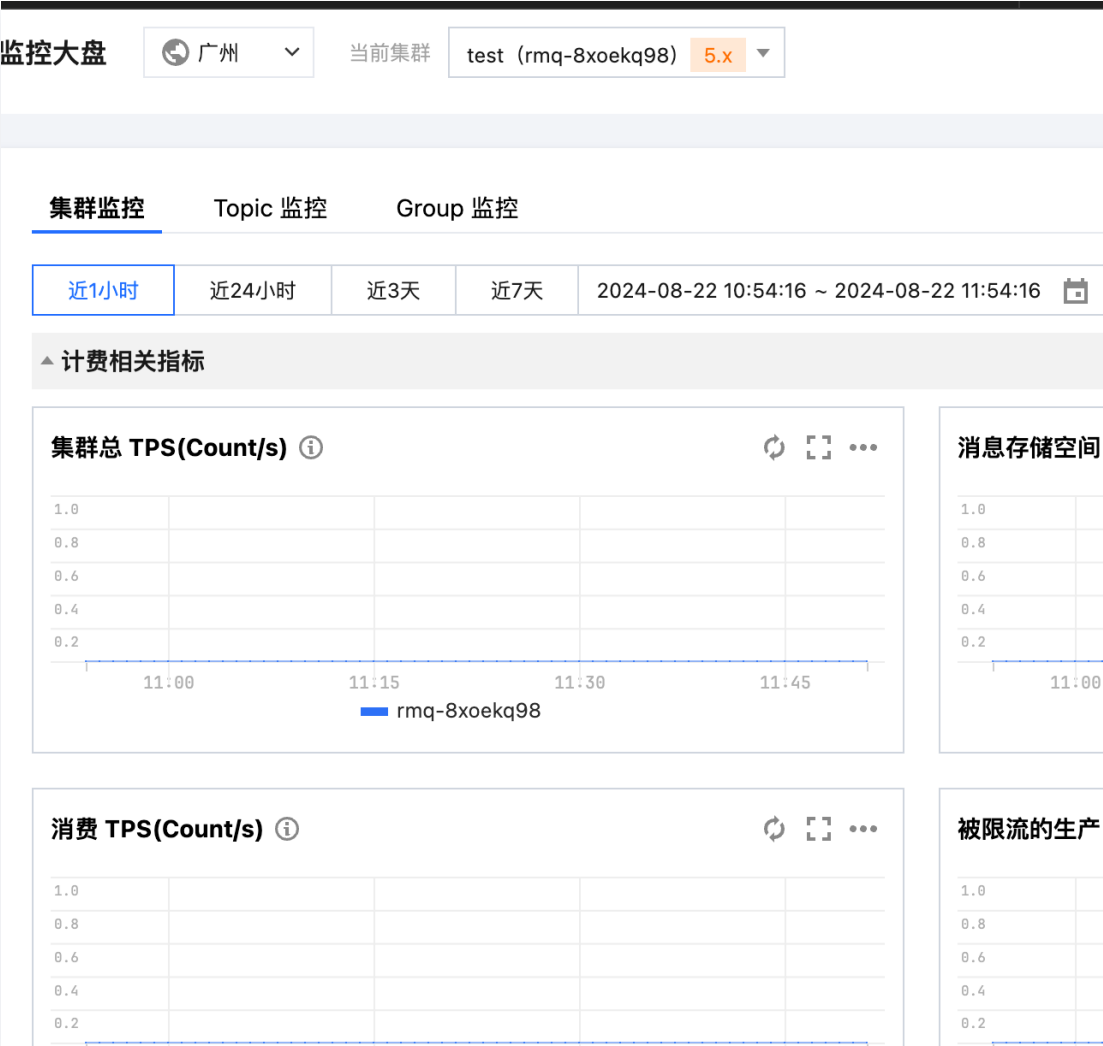
启动后即可看到生产消费正常生产：


```
ownloads/rocketmq-all-5.3.0-bin-release/benchmark
sh producer.sh -t topic0901 -w 1 -s 1024 -n "http://MQ_INST_rocketmqw4...tdmq-rocketmq.ap-gz.public.tencenttdmq.com:98
ld61xLX...n0.eyJzdWIiOi...CJ9.pQwCSlCqqfTvXALbJOELe-uw62A8F-Wa_e08aeYFix4

ownloads/rocketmq-all-5.3.0-bin-release/benchmark
Create RAMDisk /Volumes/RAMDisk for gc logging on Darwin OS.
penJDK 64-Bit Server VM warning: ignoring option PermSize=128m; support was removed in 8.0
penJDK 64-Bit Server VM warning: ignoring option MaxPermSize=320m; support was removed in 8.0
penJDK 64-Bit Server VM warning: UseCMSCompactAtFullCollection is deprecated and will likely be removed in a future release.
h producer.sh -t topic0901 -w 1 -s 1024 -n "http://MO INST rocketmqw4noo7awvqm2...public.tencenttdmq.com:9876
61xL)...SlCqqfTvXALbJOELe-uw62A8F-Wa_e08aeYFix4
t: 1,...Enable: true, messageQuantity: 0, delay
nable: false
compressEnable: false, reportInterval: 10000
urrent Time: 2024-08-22 11:43:05,547 | Send TPS: 27 | Max RT(ms): 499 | Average RT(ms): 36.244 | Send Failed: 0 | Response Failed: 0
urrent Time: 2024-08-22 11:43:15,545 | Send TPS: 28 | Max RT(ms): 499 | Average RT(ms): 35.294 | Send Failed: 0 | Response Failed: 0

Consumer Started.
Current Time: 2024-08-22 11:49:50,844 | Consume TPS: 0 | AVG(B2C) RT(ms): NaN | AVG(S2C) RT(ms): NaN | MAX(B2C) RT(ms): 0 | MAX(S2C) RT(ms): 0 | Consume Fail: 0
Current Time: 2024-08-22 11:49:51,344 | Consume TPS: 10 | AVG(B2C) RT(ms): 401853.542 | AVG(S2C) RT(ms): 401726.458 | MAX(B2C) RT(ms): 407031 | MAX(S2C) RT(ms): 406806 |
```

也可以在监控大盘查看监控指标：



3. 在 RocketMQ 控制台消息查询页面，查询刚刚发送的消息内容。

时间范围

近 100 条

近30分钟

近1小时

近6小时

近24小时

近3天

2024-08-22 11:17:14 ~ 2024-08-22 11:47:14

集群

test (rocketmq-4.x)

命名空间

Topic

查询方式

查询全部

按消息 ID 查询

按消息 Key 查询

按消息 Tag 查询

查询

批量导出

☐	消息 ID	消息 Tag	消息 Key	生产者地址	消息创建时间
☐					2024-08-22 11:42:57,200
☐					2024-08-22 11:42:57,059
☐					2024-08-22 11:42:56,931

消息详情如下:

消息查询 / 9

详情消息轨迹

基本信息

Topic

ID

00019

生产者地址

消息创建时间

2024-08-22 11:42:57,200

消息体

Qbdtqitmz0ZOOK38VhkVLhYrtJoHbEU02kY7nbFLXBrlMDFFnlpAcdXMBLCt90gaEn8oPE7HXnLzaXKAN8i9Dgu2uillPTaT5q7Soluv42IMq
hpfThTJfz5awN1zv3Vetxai94yeEaUc5bw2rLL0oUFUw4g0LwfORlIFayyMg5Y7sbp9weCg6ku5qybTYPwKtSiujB8FcSG0OmPwTrFRAhZ
ntm7vbaPeLcxRYlinVQALBaO6bOIX8OslS1PgpWulzTb7mmmUm2NM1Zopi5iDiJeBQHTs6vnyP200zJirujyUad1CoA5SKkolk2f65bPvXL
mRJP4zSjfdDemtrMBqNMUgcXXZHvjpnGN0vjNe1F7r9D15qCwelGZSuC4N0UGisjc6tJ432uliViWtnVUyQUKlJz6Sg8cq3gZQ9erUeeyOvl
pKUCUpDAF2QzKoowPXrXqXs8R5vy8Z3QtVnbY709FXvZuoxSoOm21NllwWJlooq8q7adWlmytKmiaRAdfbERKOkCinFumle6mwC68v
OxcNnuNsWWU0VAs7frtRo0HGK1KJe6K2YG6N1QF0JRykEk2bbQ1LR8RRHaOnGPrpgyftxbWYOVSoLcxW93yBCVJFRYjgCBy4VD3ux
GQCldlZ3yGsLQod1T7P3EZ9pSLEkZEaMOFireWGb5BwmYZWNqy6fGpz7w8YpJkXpUI38h2H9DKrQXnKgJwiTwkZne63jsNknELJz6XK
UCFrRhNGW9iITcg1NDBmQ1w632rTtk2DQ2tVuzcP12hvjldcmih9qsCFym5WRHV4vYB5SNgrZdz229gyloxxwTGagy7bnBNsdxO1LYiuD
Mdeol02YHFBlluTld3yZ8sqGrixowD8IA1byUt27eBgf0RLWLHa1SYWDt5LJEWNTIVG9KtzmCo2yRDNMbsKyhUJsVuMyI3xppOXCbbdGfD
VySkuKkaq3zTy2yZCq6DGHJLbB57dmKrW55xXUWGOn7qf7JPAfo5heDxB05t5n6Q2n3MnzThwiTER

详情参数

```
{  "TRACE_ON": "true",  "MSG_REGION": "gz",  "UNIQ_KEY": "FD07B51ACC660000A",  "CLUSTER": "gz_...",  "INSTANCE_ID": "MQ_INST_rocketmq"}
```