# 消息队列 RabbitMQ 版

## 产品文档

# 目录

消息队列 RabbitMQ 版 4
- 产品简介 4
  - 产品概述 4
  - 产品优势 5
  - 应用场景 6
  - 使用限制 11
  - 相关概念 13
    - 基础概念 13
    - Exchange 16
  - 开源对比 20
- 快速入门 21
  - 资源创建与准备 21
  - 使用 SDK 收发消息 28
- 操作指南 32
  - 集群管理 32
    - 创建集群 32
    - 查看集群 34
    - 升配集群 36
    - 删除集群 37
    - 访问原生控制台 39
    - 添加路由策略 40
  - 节点管理 42
  - Vhost 管理 43
  - Exchange 管理 46
  - Queue 管理 50
  - 路由关系 58
  - 用户与权限管理 60
  - 监控告警 64
    - 查看监控指标 64
    - 配置告警 69
  - 策略列表 72
    - 策略管理 72
    - 默认镜像策略 78
  - 插件管理 82
  - 访问管理 CAM 84
  - 上云迁移方案 89
    - 迁移方案概述 89
    - 步骤1：购买云上实例 90
    - 步骤2：迁移元数据上云 91
    - 步骤3：开启双读写 95
- 开发指南 97
  - 延时消息 97
- 最佳实践 100
  - RabbitMQ 客户端实践教程 100
  - RabbitMQ 消息可靠性实践教程 103
  - 其他注意事项 104
- 常见问题 105
  - 常见问题 105
- 通用参考 106
  - RabbitMQ 客户端接口说明 106
- SDK文档 107
  - Spring Boot Starter 接入 107
  - Spring Cloud Stream 接入 111
  - Java SDK 115
  - Go SDK 119
  - Python SDK 123
  - PHP SDK 127
  - C++ SDK 132
- API文档 136
  - （tdmq） 136
    - 版本（2020-02-17） 136
      - API 概览 136
      - 调用方式 142
        - 接口签名v1 142
        - 接口签名v3 149
        - 请求结构 158
        - 返回结果 159
        - 公共参数 162
      - CMQ管理相关接口 164
        - 回溯cmq队列 164
      - RocketMQ 主题相关接口 166
        - 创建RocketMQ主题 166
        - 删除RocketMQ主题 168
        - 获取RocketMQ主题列表 170
        - 更新RocketMQ主题信息 173
      - RocketMQ 命名空间相关接口 175
        - 创建RocketMQ命名空间 175
        - 删除RocketMQ命名空间 177
        - 获取RocketMQ命名空间列表 179
        - 更新RocketMQ命名空间 181
      - RocketMQ 平滑迁移相关接口 183
        - 平滑迁移：导入消费者组列表 183
        - 平滑迁移：导入topic列表 185
      - RocketMQ 消息相关接口 187
        - rocketmq消息详情 187
        - 查询RocketMQ消息轨迹 190
        - rocketmq 消息查询 192
        - 导出RocketMQ消息详情 195
        - 重置RocketMQ消费位点 198

- 发送RocketMQ消息 200
- Rocketmq消费验证 202
- RocketMQ 消费组相关接口 204
  - 创建RocketMQ消费组 204
  - 删除RocketMQ消费组 206
  - 获取在线消费端详情 208
  - 获取指定消费组下当前客户端的连接情况 210
  - 获取RocketMQ消费组列表 212
  - 获取RocketMQ消费组订阅关系 215
  - 获取指定消费组下订阅的主题列表 217
  - 更新RocketMQ消费组信息 219
- RocketMQ 角色授权相关接口 221
  - 创建RocketMQ环境角色授权 221
  - 创建RocketMQ角色 224
  - 批量删除RocketMQ环境角色授权 226
  - 批量删除RocketMQ角色 228
  - 获取RocketMQ命名空间角色列表 230
  - 获取RocketMQ角色列表 232
  - 修改RocketMQ环境角色授权 234
  - RocketMQ角色修改 236
- RocketMQ 集群相关接口 238
  - 创建RocketMQ集群 238
  - 删除RocketMQ集群 240
  - 获取单个RocketMQ集群信息 242
  - 获取RocketMQ的集群列表 244
  - 查询RocketMQ实例公网接入点信息 246
  - 查询RocketMQ专享实例列表 248
  - 更新RocketMQ集群信息 250
- 主题相关接口 252
  - 新增主题 252
  - 删除主题 255
  - 查询主题列表 257
  - 修改主题 260
- 其他接口 262
  - 删除角色 262
  - 获取租户VPC绑定关系 264
  - 发送消息 266
- 命名空间相关接口 268
  - 创建命名空间 268
  - 删除命名空间 271
  - 获取命名空间属性 273
  - 获取命名空间角色列表 276
  - 获取命名空间列表 278
  - 修改命名空间属性 280
- 消息相关接口 282
  - 确认消息 282
  - 接收消息 284
  - 批量发送消息 286
  - 发送单条消息 289
- 环境角色授权相关接口 291
  - 创建环境角色授权 291
  - 创建角色 293
  - 删除环境角色授权 295
  - 获取角色列表 297
  - 修改环境角色授权 299
  - 角色修改 301
  - 修改资源对应的项目（包含新增） 303
- 生产消费相关接口 305
  - 创建订阅关系 305
  - 删除订阅关系 308
  - 消费订阅列表 310
  - 消息回溯 313
- 集群相关接口 315
  - 创建集群 315
  - 删除集群 317
  - 获取专享集群列表 319
  - 获取集群详情 321
  - 获取集群列表 323
  - 更新集群信息 325
- 数据结构 327
- 错误码 372

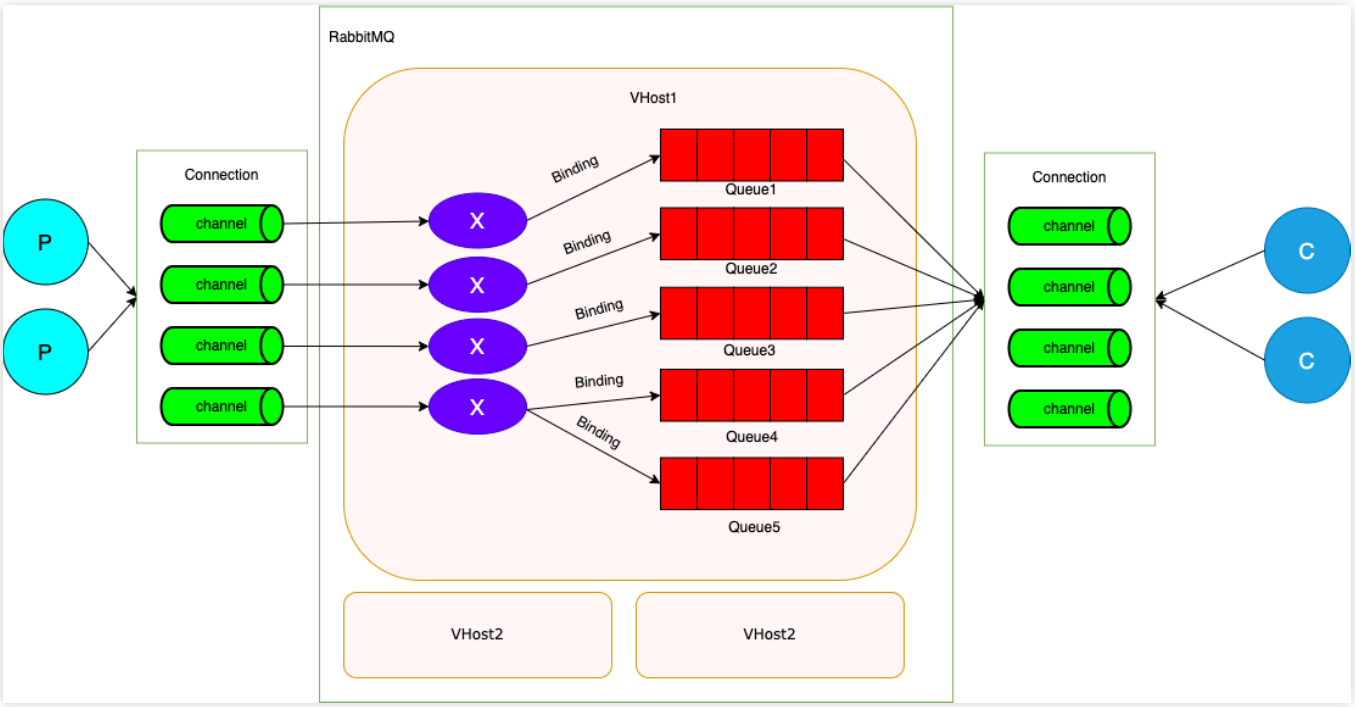
# 产品简介
# 产品概述

消息队列 RabbitMQ 版（TDMQ for RabbitMQ，简称 TDMQ RabbitMQ 版）是一款自主研发的消息队列服务，支持 AMQP 0-9-1 协议，完全兼容开源 RabbitMQ 的各个组件与概念，同时具备计算存储分离，灵活扩容的底层优势。TDMQ RabbitMQ 版拥有极为灵活的路由来适应各类业务的消息投递规则，能有缓冲上游流量压力的能力，保证消息系统的稳定运行。常用于系统间的异步通信和服务解耦，减轻不同服务之间的依赖，广泛应用于金融，政务等行业的分布式系统中。

## 技术架构

TDMQ RabbitMQ 版的基本概念如下：

- Connection：生产者、消费者连接到 RabbitMQ Broker 的 TCP 连接。
- Channel：建立在 Connection 之上的虚拟连接，RabbitMQ 处理的每个 AMQP 命令都是通过 Channel 完成的。
- Vhost：用作逻辑隔离，不同 Vhost 之间的 Exchange 和 Queue 相互隔离，互不干扰。
- Exchange（X）：接收来自生产者的消息并将消息路由到 Queue 的组件。
- Queue：存储消息的缓冲区，供消费者消费消息。
- 生产者（P）：向Exchange发送消息。
- 消费者（C）：从Queue拉取消息进行消费。



# 产品优势

## 兼容开源

支持 AMQP 0-9-1 版本标准协议，完全支持开源 RabbitMQ 社区和 Queue、Exchange、Vhost 组件。一键迁移开源 RabbitMQ 元数据，实现迁移上云零成本。

## 功能完备

TDMQ RabbitMQ 版支持原生 RabbitMQ 的各类消息模型。支持死信交换机与备用交换机，用户无需担心由于消息过期、路由失败等因素造成的消息丢失。

## 稳定可靠

持久化机制确保了 TDMQ RabbitMQ 版的高可靠性。设置 Exchange、Queue、消息的持久化，保证服务重启后元数据与消息内容不丢失。消息采用三副本存储策略，某台物理机故障时，能够实现数据的快速迁移，保证用户数据3个备份可用，服务可用性达99.95%。

## 高扩展性

TDMQ RabbitMQ 版相比于开源 RabbitMQ 支持更高的队列数量，可扩展能力强，底层系统可根据业务规模自动弹性伸缩、扩容集群规模，对用户透明。

## 易用免运维

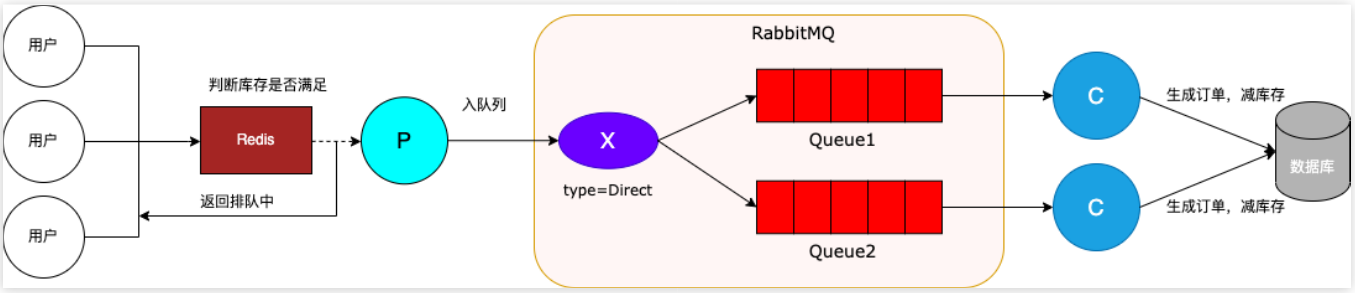
提供 API 访问接口，支持开源所有语言和版本的 SDK。提供云平台整套运维服务，实时监控告警，帮助用户快速发现并解决问题，保证服务的可用性。

# 应用场景

## 秒杀场景

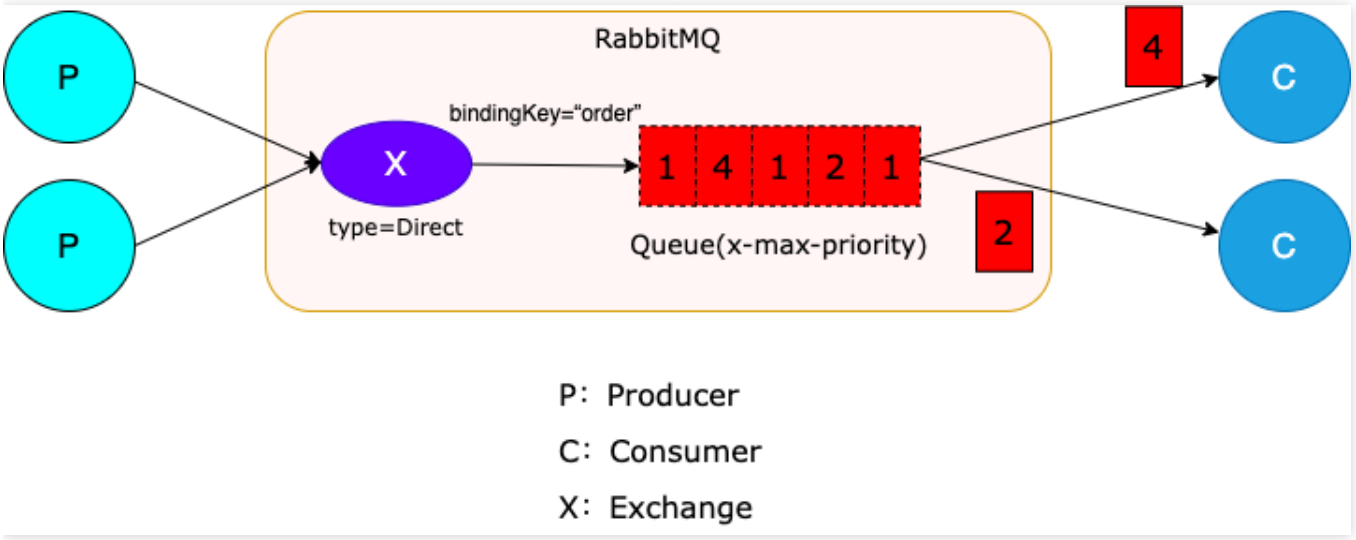
秒杀是电商系统中一个非常常见的场景。解决方案比较多，但是使用消息队列 RabbitMQ 版是一个比较好的做法。
如果是复杂的扣减库存（如涉及商品信息本身或牵连其他系统），则建议使用数据库进行库存数量的扣减，可以使用异步的方式来应对这种高并发的库存更新。

1. 在用户下单时，不立刻生成订单，而是将所有订单依次放入队列。
2. 下单模块依据自身的处理速度，从队列中依次获取订单进行“下单扣库存”操作。
3. 在订单生成成功后，用户即可进行支付操作了。
   这种方式是针对“秒杀”场景的，依据“先到先得”原则来保证公平公正，所有用户都可以抢购，然后等待订单处理，最后生成订单（如果库存不足，则生成订单失败）。




## 优先级消息

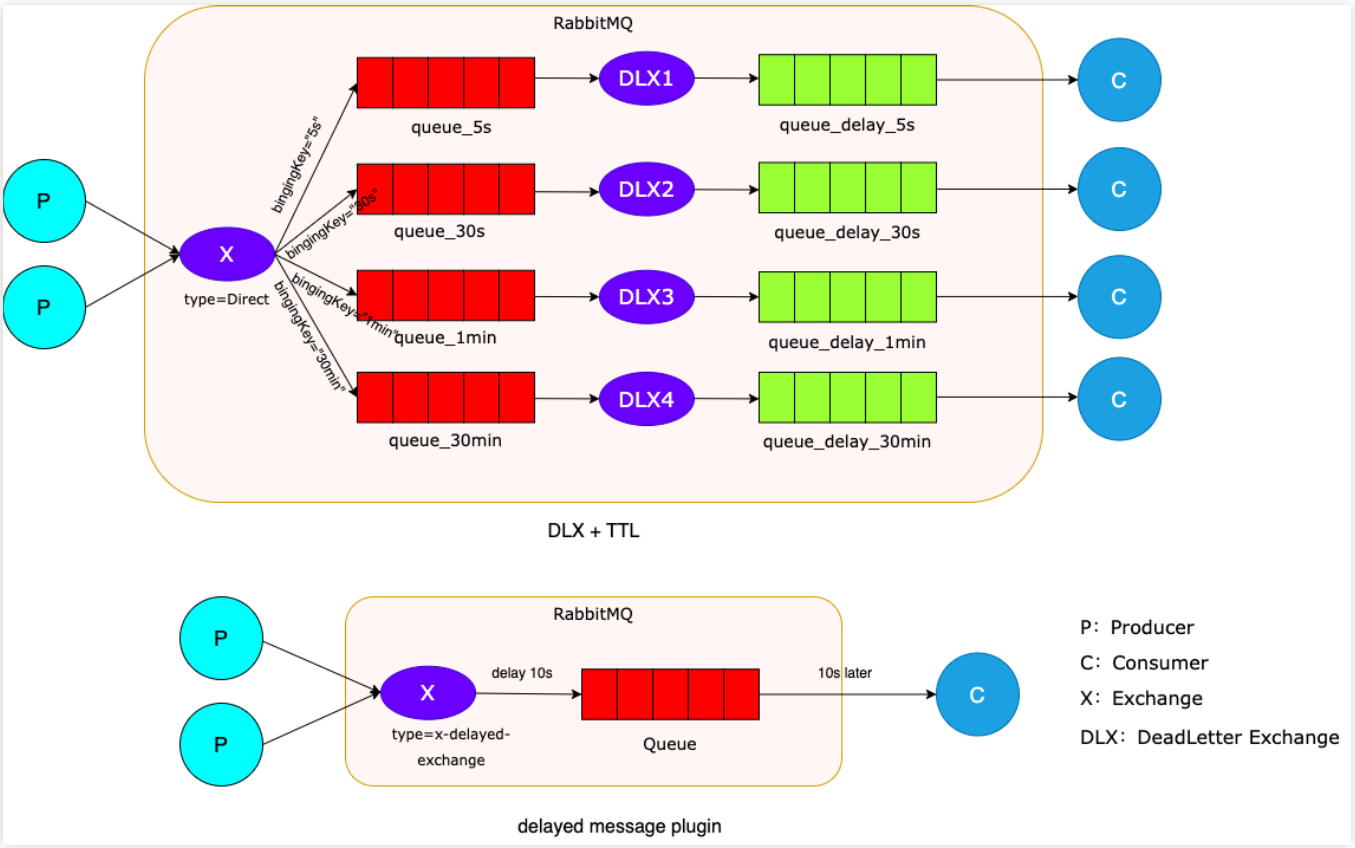
在消费消息时，如果消息的重要程度不同，重要性高的消息希望被优先消费，这时可以使用消息队列 RabbitMQ 版优先级队列的能力，让优先级高的消息优先被消费。
参考如下业务场景：
例如系统中有订单催付的场景，客户在电商系统下的订单，系统会及时将订单推送给客户。如果在设定的时间内未付款那么就会给客户推送一条短信提醒。但是，电商系统会分大客户和小客户，大客户的订单催收消息需要优先处理，其他的小客户的催收相对优先级会低一些。
使用消息队列 RabbitMQ 版的优先级队列可以很好的支持这个场景，让优先级高的消息不会积压太久。如果发现是大客户的订单给一个相对比较高的优先级，优先被处理；否则就是默认优先级。




# 延迟消息场景

实际业务系统中，有发送延迟消息的需求。如果自己实现延迟逻辑，可靠性和延迟精度很难得到保障。使用消息中间件，如消息队列 RabbitMQ 版可以很好的处理此类需求。延迟消息的使用场景有：

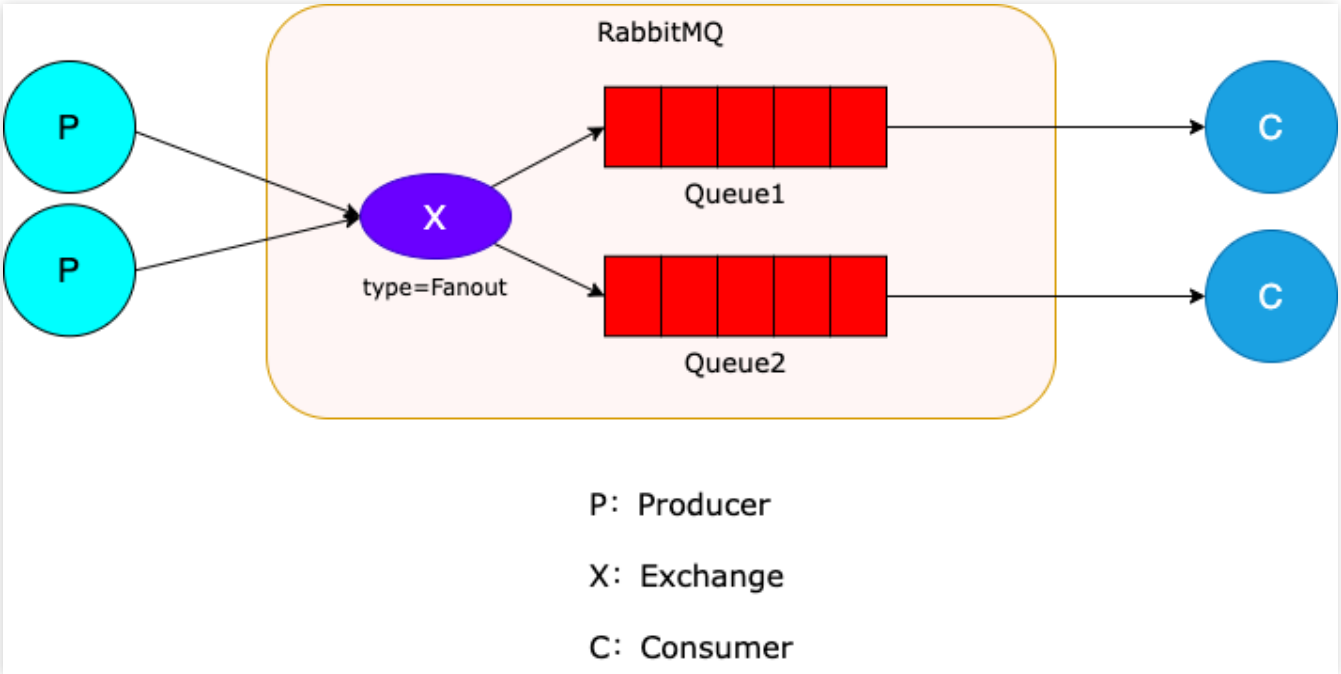
- 订单系统中，用户下单后有30分钟时间进行支付。如果30分钟内没有支付成功，那么这个订单将进行异常处理。这里可以使用消息队列 RabbitMQ 版延迟消息能力来处理这些超时订单。
- 物联网系统中，用户希望通过手机遥控智能设备在指定的时间工作，这时就可以将指令发到延迟队列，当指令设定的时间到了，再将控制指令推送到智能设备。




# 消息广播

很多业务系统需要将信息广播给下游的系统。如果通过 RPC 的方式，耦合比较重，对上游业务系统的压力也比较大。此时，可以使用消息队列 RabbitMQ 版的 Fanout Exchange 来处理此类需求。可以使用的场景：

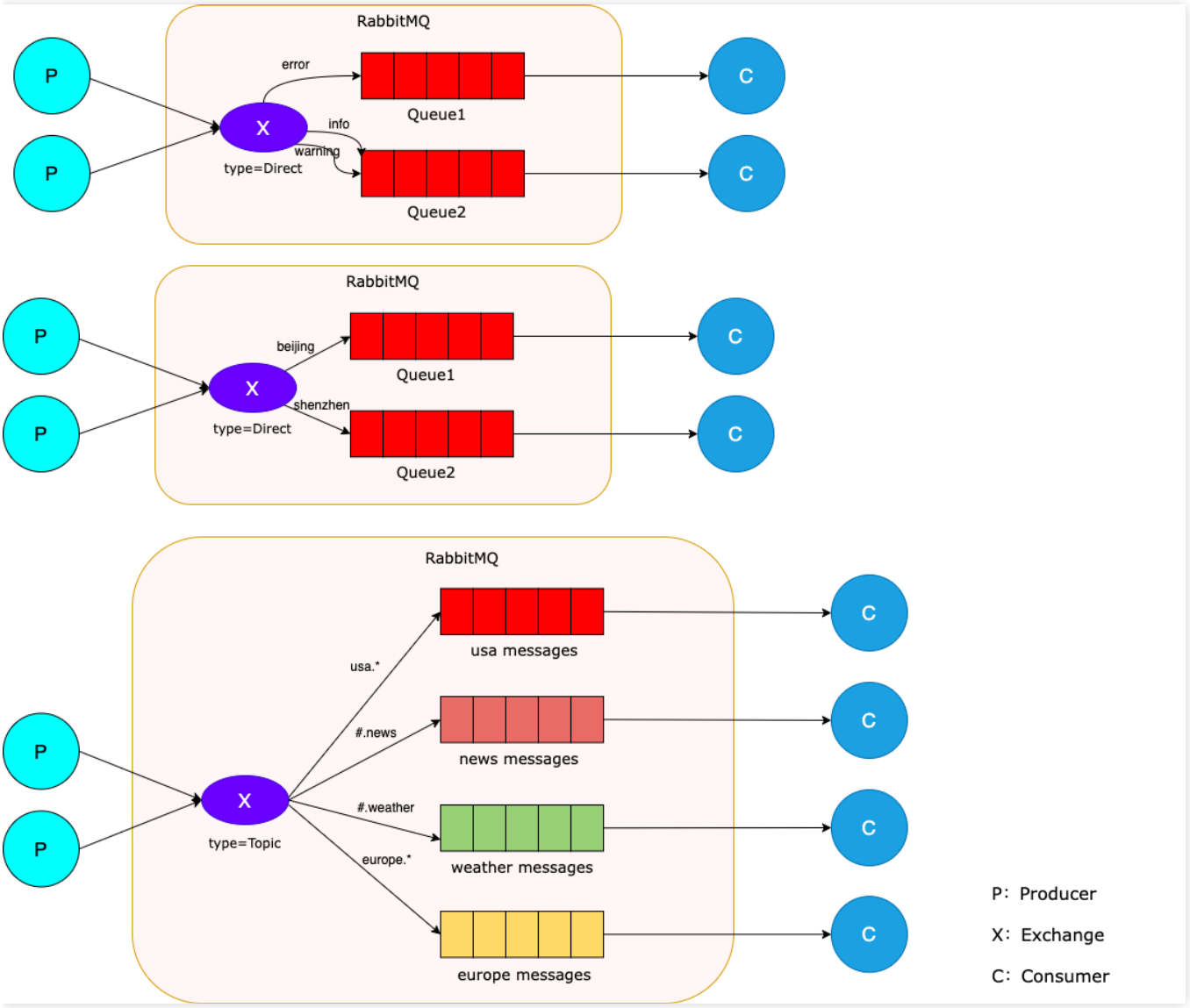
- 大型多人在线游戏（MMO）可以将其用于排行榜更新或其他全局事件；
- 体育新闻网站可以使用扇形交换机向客户端近乎实时的分发比分信息；
- 分布式系统使用它来广播各种状态和配置更新；
- 群聊可以使用它在参与者之间分发消息。




# 灵活路由场景

随着微服务架构的流行，服务拆分得较细，服务的数据会以消息的形式，使用精心设计的分发策略发送到不同的队列中去。这时可以充分的利用消息队列 RabbitMQ 版灵活的消息路由的能力，将消息分发到目标 Queue 中。可以使用的场景：

- 日志处理场景，可以将日志按类型投递到不同的 Queue。例如 Error 单独一个处理队列，优先处理。
- 电商物流系统的物流信息按地域分发给不同的消费端进行处理。
- 复杂路由场景。


RabbitMQ
P
P
X
type=Direct
error
info
warning
Queue1
Queue2
C
C
RabbitMQ
P
P
X
type=Direct
beijing
shenzhen
Queue1
Queue2
C
C
RabbitMQ
P
P
X
type=Topic
usa.*
#.news
#.weather
europe.*
usa messages
news messages
weather messages
europe messages
C
C
C
C
P：Producer
X：Exchange
C：Consumer

# 使用限制

本文列举了 TDMQ RabbitMQ 版专享集群对集群和字符的限制，请您在使用产品时注意不要超出对应的限制值，避免出现异常。

## 集群

| 限制类型 | 限制说明 |
| --- | --- |
| 集群名称长度 | 3个字符 - 64个字符 |
| 集群名称规范 | 不能为空，只能包含数字、字母、“-”和“_” |

## Vhost

| 限制类型 | 限制说明 |
| --- | --- |
| Vhost 名称长度 | 1个字符 - 64个字符 |
| Vhost 名称规范 | 不能为空，只能包含字母、数字、“.”、“-”及“_” |
| Vhost 数量 | 每个集群限制20个 |

## Exchange

| 限制类型 | 限制说明 |
| --- | --- |
| Exchange 名称长度 | 1个字符 - 64个字符 |
| Exchange 名称规范 | 不能为空，只能包含字母、数字、“.”、“-”及“_” |
| Exchange 路由类型 | 可选项包含direct，fanout，topic及headers，创建时必选其一 |
| Exchange 数量 | 每个 Vhost 限制 1000个 |

## Queue

| 限制类型 | 限制说明 |
| --- | --- |
| Queue 名称长度 | 1个字符 - 64个字符 |
| Queue 名称规范 | 不能为空，只能包含字母、数字、“.”、“-”及“_” |
| Queue 数量 | 每个 Vhost 限制 1000个 |

## Channel

| 限制类型 | 限制说明 |
| --- | --- |
| Connection 通道数 | 每个 Connection 的最大通道数为1024，如果用户在这个连接上创建的通道数达到这个限制则无法再创建通道 |

# 相关概念
# 基础概念

本文主要介绍了在使用 TDMQ RabbitMQ 版中常见的名词及解释说明。

## Binding

RabbitMQ 中通过 Binding 将 Exchange 与 Queue 关联起来，这样 RabbitMQ 就知道如何正确地将消息路由到指定的 Queue了。

## Binding key

- 在绑定（Binding）Exchange 与 Queue 的同时，一般会指定一个 binding key；生产者将消息发送给 Exchange 时，一般会指定一个 routing key；当 binding key 与 routing key 相匹配时，消息将会被路由到对应的 Queue 中。
- 在绑定多个 Queue 到同一个 Exchange 的时候，这些 Binding 允许使用相同的 binding key。
- binding key 并不是在所有情况下都生效，它依赖于 Exchange Type，例如 fanout 类型的 Exchange 就会无视 binding key，而是将消息路由到所有绑定到该 Exchange 的 Queue。

## Channel

在客户端的每个物理 TCP 连接里，可建立多个 Channel，每个 Channel 代表一个会话任务。

## Connection

TCP 连接，生产者或消费者与 TDMQ RabbitMQ 版间的物理 TCP 连接。

## Exchange

生产者将消息发送到 Exchange，由 Exchange 将消息路由到一个或多个 Queue 中（或者丢弃）。Exchange 根据消息的属性或内容路由消息。

## Exchange Types

RabbitMQ 常用的 Exchange Type 有 fanout、direct、topic 、header 等。

- Fanout：fanout 类型的 Exchange 会把所有发送到该 Exchange 的消息路由到所有与它绑定的 Queue 中。
- Direct：Direct 类型的 Exchange 会把消息路由到那些 binding key 与 routing key 完全匹配的 Queue 中。
- Topic：该类型与 direct 类型相似，Topic 类型 Exchange 支持多条件匹配和模糊匹配，即使用 Routing Key 模式匹配和字符串比较的方式将消息路由至与其绑定的 Queue 中。
- Header：与 Routing Key 无关，匹配机制是匹配消息中的 Headers 属性信息。在绑定 Queue 与 Headers Exchange 之前声明一个 map 键值对，通过这个map 对象实现 Queue 和 Exchange 的绑定。当消息发送到 RabbitMQ 时会取到该消息的 Headers 与 Exchange 绑定时指定的键值对进行匹配；如果完全匹配则消息会路由到该队列，否则不会路由到该队列。

# Message acknowledgment

消息回执，消费者在消费完消息后发送一个回执给 RabbitMQ，RabbitMQ 收到回执后才将消息从队列中删除，如果没有收到回执并检测到消费者的 RabbitMQ 连接断开，那么会将该消息发送给其他消费者进行处理。

# Message durability

消息持久化，即 RabbitMQ 服务重启的情况下，也不会丢失消息，我们可以将 Queue 与 Message 都设置为（durable），这样可以保证绝大部分情况下我们的 RabbitMQ 消息不会丢失。

# Prefetch count

消费者在开启 acknowledge 的情况下，对接收到的消息可以根据业务的需要异步对消息进行确认，prefetch 允许为每个 consumer 指定最大的 unacked messages 数目。例如：设置prefetchCount=10，则 Queue 每次给每个消费者（假设一共2个AB）发送10条消息，消费者无需马上回应，消费者将10条消息缓存本地客户端；当一个消费者处理完1条消息后 Queue 会再给该消费者发送一条消息。如果两个消费者都没有回复任何一条 ack，则 queue 就不会继续发送消息。

# Queue

Queue（队列）是 RabbitMQ 的内部对象，用于存储消息。每个消息都会被投入到一个或多个 Queue 里。

# Routing key

- 生产者在将消息发送到 Exchange 的时候，一般会指定一个 routing key，来指定这个消息的路由规则，而这

个 routing key 需要与 Exchange Type 及 binding key 联合使用才能最终生效。

- 在 Exchange Type 与 binding key 固定的情况下（在正常使用时一般这些内容都是固定配置好的），我们的生产者就可以在发送消息给 Exchange 时，通过指定 routing key 来决定消息流向哪里。

# Vhost

虚拟主机（Virtual Host，Vhost），用作逻辑隔离，分别管理各自的 Exchange、Queue 和 Binding，使得应用安全的运行在不同的 Vhost 实例上，相互之间不会干扰。一个实例下可以有多个 Vhost，一个 Vhost 里面可以有若干个 Exchange 和 Queue。生产者和消费者连接消息队列 RabbitMQ 版需要指定一个 Vhost。

# Exchange

本文介绍 TDMQ RabbitMQ 版中 Exchange 的概念、类型和使用方式。

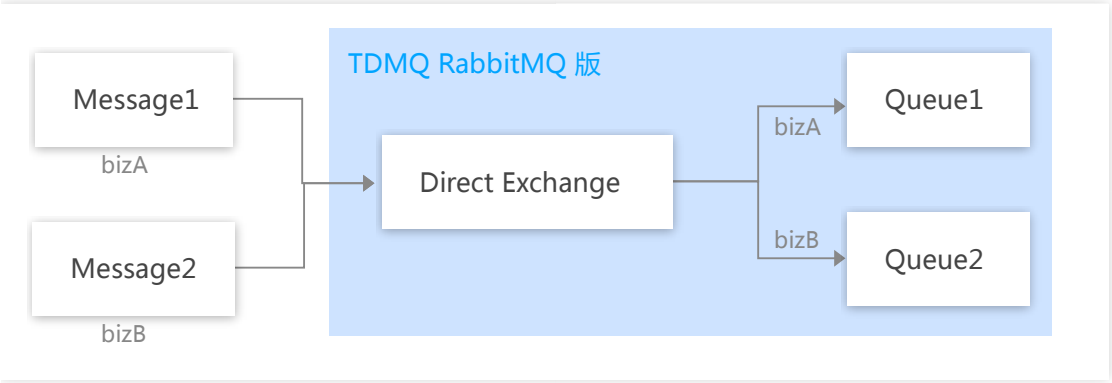
## 概念

Exchange 是 TDMQ RabbitMQ 版的消息路由代理，Producer 将消息发送到 Exchange 中，Exchange 根据消息的属性或内容将消息路由到一个或多个 Queue 中（或者丢弃），Consumer 从 Queue 中拉取消息进行消费。
TDMQ RabbitMQ 版目前支持 Direct、Fanout、Topic 和 Header 四种类型的 Exchange。

- Direct：该类型 Exchange 会把消息路由到 RoutingKey 和 BindingKey 完全匹配的 Queue 中。
- Fanout：该类型 Exchange 会将消息路由到所有与其绑定的 Queue 中。
- Topic：该类型 Exchange 支持多条件匹配和模糊匹配，即使用 RoutingKey 模式匹配和字符串比较的方式将消息路由至与其绑定的Queue中。
- Header：该类型 Exchange 与 Routing Key 无关，匹配机制是匹配消息中的 Headers 属性信息。在绑定 Queue 与 Headers Exchange 之前声明一个map键值对，通过这个map对象实现消息队列和交换机的绑定。

## Direct Exchange

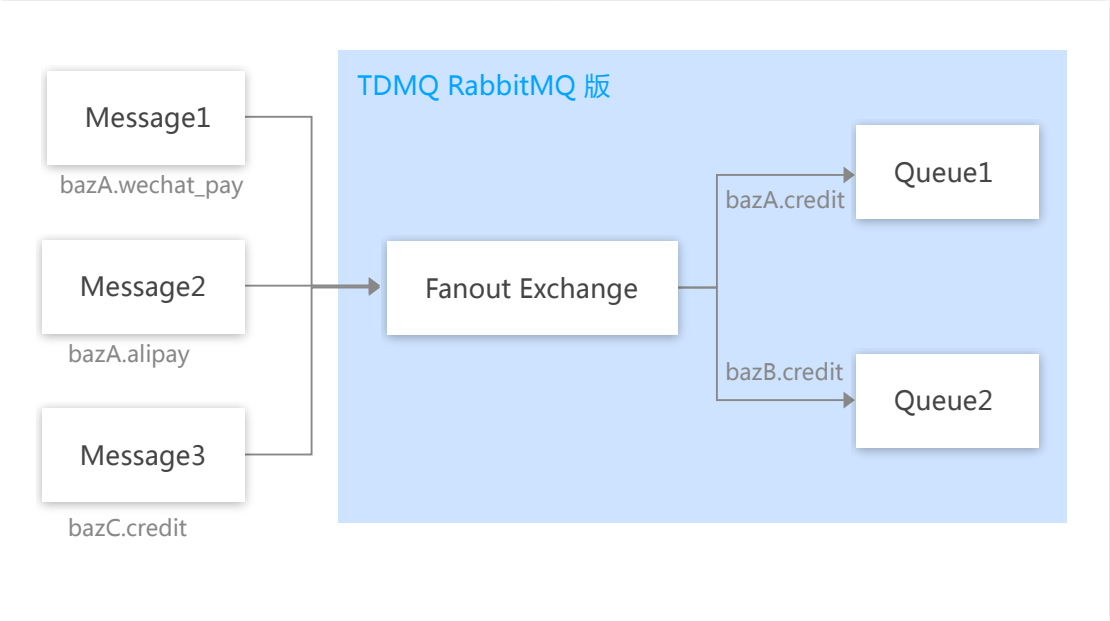
路由规则：Direct Exchange 会把消息路由到 RoutingKey 和 BindingKey 完全匹配的 Queue 中。
应用场景：该类型 Exchange 适用于通过简单字符标识符过滤消息的场景，常用于单播路由。
使用示例：




| Message | Routing Key | Binding Key | Queue |
| --- | --- | --- | --- |
| Message 1 | bizA | bizA | Queue 1 |
| Message 2 | bizB | bizB | Queue 2 |

## Fanout Exchange

路由规则： 该类型 Exchange 会将消息路由到所有与其绑定的 Queue 中。
应用场景：该类型 Exchange 适用于广播消息的场景。例如分发系统使用 Fanout Exchange 来广播各种状态和配置更新。
使用示例

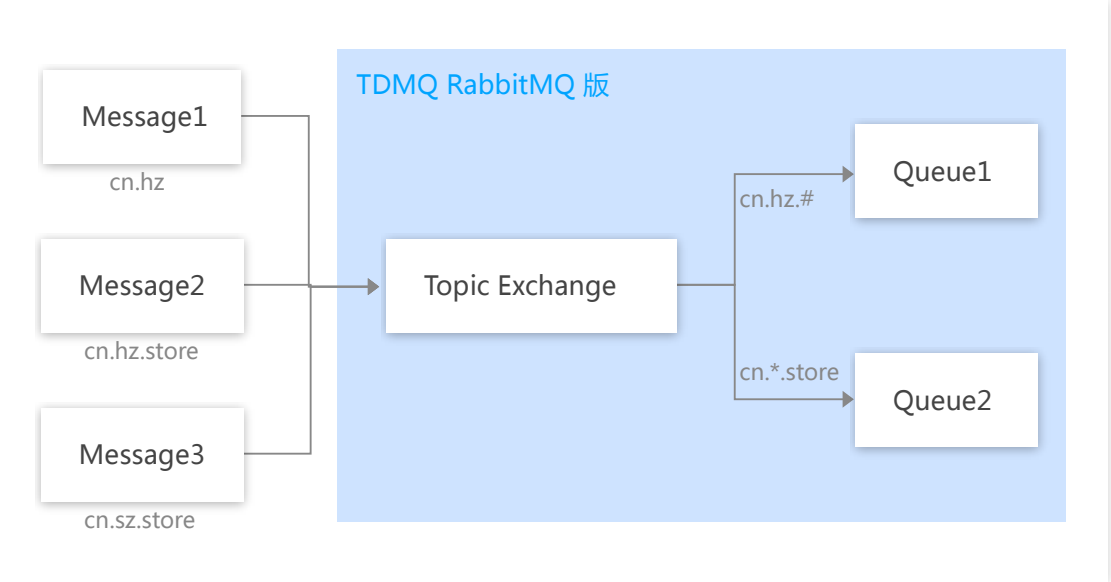



| Message | Routing Key | Binding Key | Queue |
| --- | --- | --- | --- |
| Message 1 | bazA.wechat_pay | bazA.credit , bazB.credit | Queue 1 , Queue 2 |
| Message 2 | bazA.alipay | bazA.credit , bazB.credit | Queue 1 , Queue 2 |
| Message 3 | bazC.credit | bazA.credit , bazB.credit | Queue 1 , Queue 2 |

# Topic Exchange

路由规则：该类型 Exchange 支持多条件匹配和模糊匹配，即使用 Routing Key 模式匹配和字符串比较的方式将消息路由至与其绑定的 Queue 中。

- Topic Exchange 支持的通配符包括星号“*”和井号“#”。
- 星号“*”代表一个英文单词，例如 sh。
- 井号“#”代表零个、一个或多个英文单词，单词间用英文句号"."分隔，例如 cn.hz。
  应用场景
  该类型 Exchange 常用于多播路由，例如需要使用 Topic Exchange 分发有关于特定地理位置的数据。
  使用示例

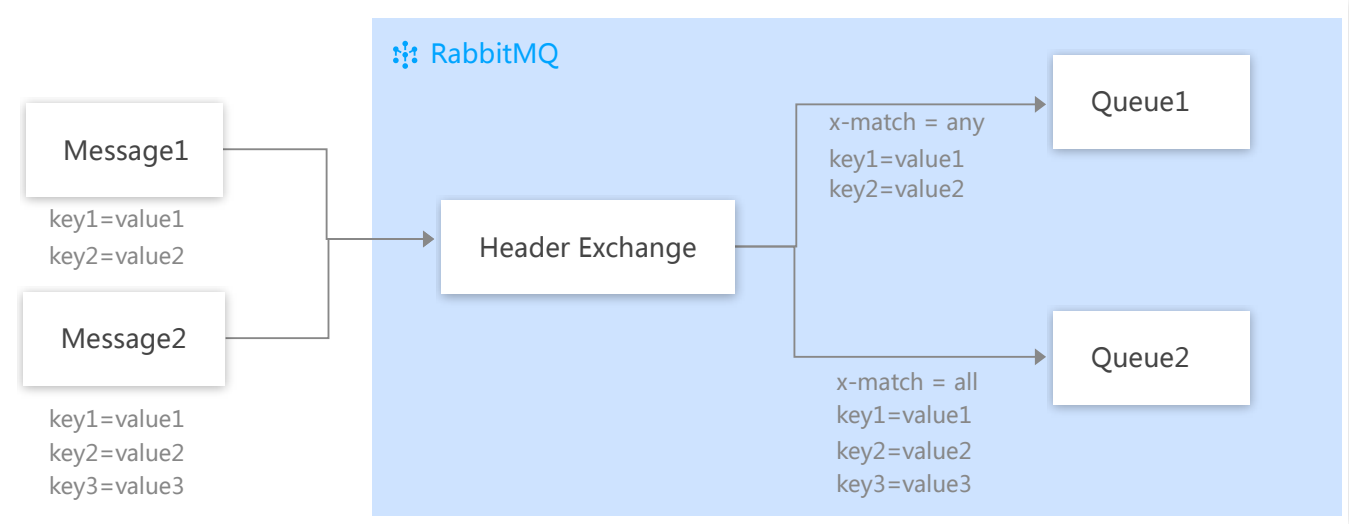



| Message | Routing Key | Binding Key | Queue |
| --- | --- | --- | --- |
| Message 1 | cn.hz | cn.hz.# | Queue 1 |
| Message 2 | cn.hz.store | cn.hz.# 、 cn.*.store | Queue 1、Queue 2 |
| Message 3 | cn.sz.store | cn.*.store | Queue 2 |

# Header Exchange

与 Routing Key 无关，匹配机制是匹配消息中的 Headers 属性信息。在绑定 Queue 与 Headers Exchange 之前声明一个map键值对，通过这个map对象实现Queue 和 Exchange 的绑定。当消息发送到 RabbitMQ 时会取到该消息的 Headers 与 Exchange 绑定时指定的键值对进行匹配；如果完全匹配则消息会路由到该队列，否则不会路由到该队列。
匹配规则x-match有下列两种类型：

- x-match = all ：表示所有的键值对都匹配才能接受到消息
- x-match = any ：表示只要有键值对匹配就能接受到消息




| Message | 消息Headers属性 | Binding Headers 属性 | Queue |
| --- | --- | --- | --- |
| Message 1 | key1=value1<br>key2=value2 | x-match = any<br>key1=value1<br>key2=value2 | Queue 1 |
| Message 2 | key1=value1<br>key2=value2<br>key3=value3 | x-match = any<br>key1=value1<br>key2=value2 、<br><br>x-match = all<br>key1=value1<br>key2=value2<br>key3=value3 | Queue 1、<br>Queue 2 |

# 开源对比

本文介绍使用TDMQRabbitMQ 版对比自建 RabbitMQ 具备的优势。

| 模块 | 功能描述 | TDMQ RabbitMQ 版 | 自建 RabbitMQ | 总结 |
| --- | --- | --- | --- | --- |
| 高可用 | 跨AZ高可用部署 | 支持跨 AZ 高可用部署，具备成熟的故障恢复方案。 | 支持但较为繁琐，需要自行设计部署方案和参数。 | **TDMQ RabbitMQ 有成熟标准化的高可用方案。**省去人工设计配置方案的工作，也避免方案出错的风险。 |
| 可扩展性 | 在线升级 | 支持产品化水平/垂直扩容，按需升级集群规格。 | 不支持，需要手动操作，可能影响业务。 | **TDMQ RabbitMQ 比开源可扩展性更强。**可根据业务需求自动化扩容，无需额外关注底层资源。 |
| 可运维性 | 开箱即用 | 成熟化的部署、监控方案，开箱即用。 | 开源控制台可以看到部分监控指标 | TDMQ RabbitMQ 比开源可运维性更高。<br>支持开箱即用，提供丰富的监控指标、告警，界面化组件运维。这些都是开源 RabbitMQ 不具备的特性。 |
| | 监控告警 | 支持实例、节点、Vhost、Queue、Exchange 等多维度监控告警。 | 开源控制台可以看到部分监控指标，若需要集成节点 CPU 利用率等其他指标，需要自行搭建并维护开源监控系统。 | |
| 安全合规性 | 管控操作权限控制 | 控制台操作基于 CAM 控制权限，安全合规。 | 管控操作权限难以精细化控制。 | TDMQ RabbitMQ 比开源安全合规性更强。管控流操作都配套有权限管控审计，确保所有操作可追溯。且完全兼容开源的访问控制能力 |
| | 操作可追溯 | 管控操作对接云审计，可回溯。 | 不支持。 | |

# 快速入门
# 资源创建与准备

## 操作场景

本文介绍通过 RabbitMQ 控制台创建 Vhost、Exchange 和 Queue 等资源的操作步骤，了解运行一个客户端之前所需要进行的资源准备。

## 操作步骤

### 步骤1：新建集群

1. 登录 RabbitMQ 控制台。
2. 在左侧导航栏选择集群管理 > 集群列表，单击新建集群，进入购买页面。
3. 在购买页面，选择需要创建的实例规格，完成集群创建。
4. 在集群列表页，单击集群的“ID”，进入基本信息页面，在客户端接入模块，查看或创建服务端的连接信息。

### 步骤2：新建 Vhost

1. 单击刚刚创建好的集群的“ID”，进入基本信息页面。
2. 选择顶部 Vhost 页签，单击新建进入创建 Vhost 页面。
3. 在新建 Vhost 对话框，设置 Vhost 的相关属性配置。
4. 单击提交完成所在集群 Vhost 的创建。

## 步骤3：创建用户并授权

每个集群下都默认有一个名为“admin”的用户，您可以为这个默认用户配置权限，也可以重新新建用户。

1. 在页面顶部选择用户与权限页签，在用户管理页面，单击新建用户。
2. 填写好用户名和密码，单击提交，完成用户创建。

3. 在用户与权限页面，选择权限列表页签，进入权限列表，单击配置权限。
4. 在权限配置页面，选择好需要配置权限的 Vhost 和用户，并设置好权限规则。
   权限规则设置支持使用正则表达式匹配资源。例如，勾选配置且在输入框内输入 test.* ，则表示授权给该用户当前 Vhost 下，所有名称以 test 开头的资源的配置权限。
5. 单击提交，完成权限配置。

## 步骤4：创建 Exchange

1. 在左侧导航栏选择集群管理 > Exchange，选择刚刚创建好的集群和 Vhost，单击新建。
2. 填写 Exchange 名称，选择路由类型，可选填其他高级参数。
3. 单击提交，完成 Exchange 创建。

## 步骤5：创建 Queue

1. 在左侧导航栏选择集群管理 > Queue 页签，选择刚刚创建好的集群和 Vhost，单击新建。
2. 填写 Queue 名称，选择 Queue类型，所在节点，可选填常用参数和其他高级选项。
3. 在最后一步，单击提交，完成 Queue 创建。

## 步骤6：绑定路由关系

1. 在 Vhost 列表页，刚刚创建好的 Vhost 的“ID”，进入基本信息页面。
2. 在页面顶部选择路由关系页签，单击新建。
   源 Exchange 选择刚刚创建的 Exchange，填写绑定key，绑定类型选择 Queue，绑定目标选择刚刚创建好的 Queue。
3. 单击提交，完成路由关系绑定。

新增绑定关系
当前 Vhost
(AMQP default vhost)
源 Exchange *
绑定 key *
只能包含字母、数字、短划线（-）、下划线（_）、英文句号（.）、at符号（@）、井号（#）、星号（*），长度限制1~255字符。
绑定目标类型
Exchange
Queue
绑定目标 *
提交
关闭

# 使用 SDK 收发消息

## 操作场景

本文以调用 Java SDK 为例介绍通过开源 SDK 实现消息收发的操作过程，帮助您更好地理解消息收发的完整过程。

## 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 pom.xml 添加以下依赖：

```
<!-- in your <dependencies> block -->
<dependency>
    <groupId>com.rabbitmq</groupId>
    <artifactId>amqp-client</artifactId>
    <version>5.17.1</version>
</dependency>
<dependency>
    <groupId>org.slf4j</groupId>
    <artifactId>slf4j-api</artifactId>
    <version>1.7.30</version> <!-- 使用最新版本 -->
</dependency>
<dependency>
    <groupId>ch.qos.logback</groupId>
    <artifactId>logback-classic</artifactId>
    <version>1.2.3</version> <!-- 使用最新版本 -->
</dependency>
```

### 步骤2：生产消息

编译并运行 MessageProducer.java。

```
import com.rabbitmq.client.Channel;
import com.rabbitmq.client.Connection;
import com.rabbitmq.client.ConnectionFactory;
import com.tdmq.demo.cloud.Constant;

/**

 * 消息生产者
  */
  public class MessageProducer {

   /**
```

```
 * 交换机名称
  */
  private static final String EXCHANGE_NAME = "exchange_name";

  public static void main(String[] args) throws Exception {
   // 连接工厂
   ConnectionFactory factory = new ConnectionFactory();
   // 设置服务地址 (完整复制控制台接入点地址)
   factory.setUri("amqp://***");
   // 设置Virtual Hosts (开源 RabbitMQ 控制台复制完整Vhost名称)
   factory.setVirtualHost(VHOST_NAME);
   // 设置用户名 (开源 RabbitMQ 控制台中Vhost的配置权限中的user名称)
   factory.setUsername(USERNAME);
   // 设置密码 (对应user的密钥)
   factory.setPassword("****");
   // 获取连接、建立通道
   try (Connection connection = factory.newConnection(); Channel channel = connection.createChannel()) {
      // 绑定消息交换机 (EXCHANGE_NAME必须在消息队列RabbitMQ版控制台上已存在，并且Exchange的类型与控制台上的类型一致)
      channel.exchangeDeclare(EXCHANGE_NAME, "fanout");
      for (int i = 0; i < 10; i++) {
         String message = "this is rabbitmq message " + i;
         // 发布消息到交换机，交换机自动将消息投递到相应队列
         channel.basicPublish(EXCHANGE_NAME, "", null, message.getBytes());
         System.out.println(" [producer] Sent '" + message + "'");
      }
   } catch (Exception e) {
      e.printStackTrace();
   }
}
}
```

| 参数 | 说明 |
| --- | --- |
| EXCHANGE_NAME | Exchange 名称，在控制台 Exchange 列表获取。 |
| factory.setUri | 集群接入地址，在集群基本信息页面的客户端接入模块获取。<br> |
| factory.setVirtualHost | Vhost 名称，在控制台 Vhost 列表获取。 |
| factory.setUsername | 用户名称，填写在控制台创建的用户名称。 |
| factory.setPassword | 用户密码，填写在控制台创建用户时填写的密码。 |

## 步骤3：消费消息

编译并运行 MessageConsumer.java。

```
import com.rabbitmq.client.AMQP;
import com.rabbitmq.client.Channel;
import com.rabbitmq.client.Connection;
import com.rabbitmq.client.ConnectionFactory;
```

```
import com.rabbitmq.client.DefaultConsumer;
import com.rabbitmq.client.Envelope;
import com.tdmq.demo.cloud.Constant;

import java.io.IOException;
import java.nio.charset.StandardCharsets;

/**

 * 消息消费者
  */
  public class MessageConsumer1 {

  /**

   * 队列名称
    */
    public static final String QUEUE_NAME = "queue_name";

  /**

   * 交换机名称
    */
    private static final String EXCHANGE_NAME = "exchange_name";

  public static void main(String[] args) throws Exception {
      // 连接工厂
      ConnectionFactory factory = new ConnectionFactory();
      // 设置服务地址 (完整复制控制台接入点地址)
      factory.setUri("amqp://***");
      // 设置Virtual Hosts (开源 RabbitMQ 控制台中复制完整Vhost名称)
      factory.setVirtualHost(VHOST_NAME);
      // 设置用户名 (开源 RabbitMQ 控制台中Vhost的配置权限中的user名称)
      factory.setUsername(USERNAME);
      // 设置密码 (对应user的密钥)
      factory.setPassword("****");
      // 获取连接
      Connection connection = factory.newConnection();
      // 建立通道
      Channel channel = connection.createChannel();
      // 绑定消息交换机
      channel.exchangeDeclare(EXCHANGE_NAME, "fanout");
      // 声明队列信息
      channel.queueDeclare(QUEUE_NAME, true, false, false, null);
      // 绑定消息交换机 (EXCHANGE_NAME必须在消息队列RabbitMQ版控制台上已存在，并且Exchange的类型与控制台上的类型一致)
      channel.queueBind(QUEUE_NAME, EXCHANGE_NAME, "");
      System.out.println(" [Consumer1] Waiting for messages.");
      // 订阅消息
      channel.basicConsume(QUEUE_NAME, false, "ConsumerTag", new DefaultConsumer(channel) {
          @Override
          public void handleDelivery(String consumerTag, Envelope envelope,
                                     AMQP.BasicProperties properties, byte[] body)
                  throws IOException {
              //接收到的消息，进行业务逻辑处理。
              System.out.println("Received :  " + new String(body, StandardCharsets.UTF_8) + ", deliveryTag: " + envelope.getDeliveryTag
() + ", messageId: " + properties.getMessageId());
              channel.basicAck(envelope.getDeliveryTag(), false);
          }
      });
  }
```

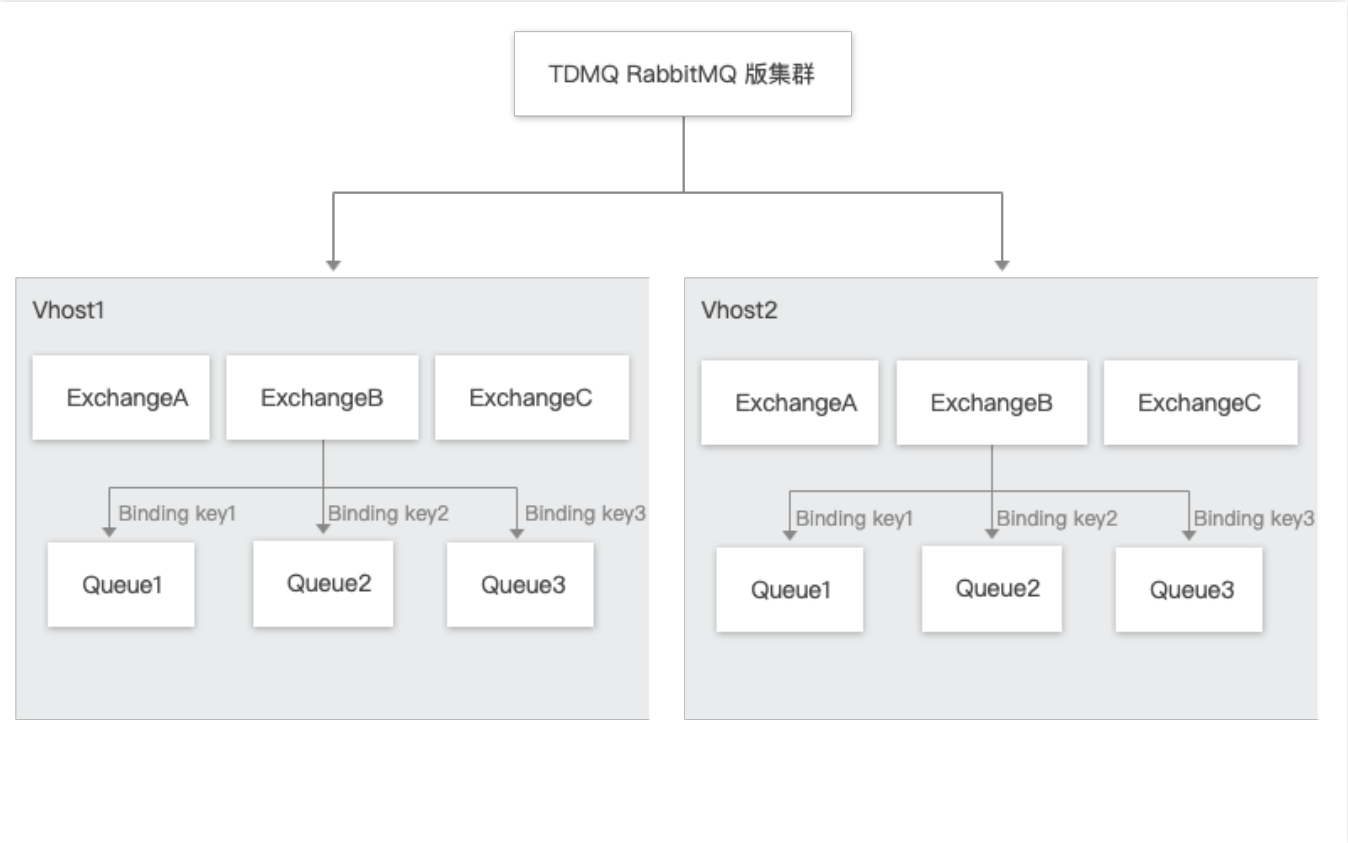
```
}
```

| 参数 | 说明 |
| --- | --- |
| QUEUE_NAME | Queue 名称，在控制台 Queue 列表获取。 |
| EXCHANGE_NAME | Exchange 名称，在控制台 Exchange 列表获取。 |
| factory.setUri | 集群接入地址，在集群基本信息页面的客户端接入模块获取。<br> |
| factory.setVirtualHost | Vhost 名称，在控制台 Vhost 列表获取。 |
| factory.setUsername | 用户名称，填写在控制台创建的用户名称。 |
| factory.setPassword | 用户密码，填写在控制台创建用户时填写的密码。 |

# 操作指南
# 集群管理
# 创建集群

## 操作场景

集群是 TDMQ RabbitMQ 版中的一个资源维度，不同集群的 Vhost、Exchange、Queue 等完全隔离。常见的使用方式如：开发测试环境使用一个专门集群，生产环境使用一个专门的集群。
本文主要介绍在 TDMQ RabbitMQ 版控制台中创建专享集群的操作步骤。




## 操作步骤

1. 登录 RabbitMQ 控制台。

2. 在左侧导航栏选择集群管理 > 集群列表，单击新建集群，进入集群新建页面。

3. 在集群新建页面，选择需要创建的实例规格。

| 参数 | 是否必选 | 说明 |
| --- | --- | --- |
| RabbitMQ 版本 | 是 | 支持 3.8.30 和 3.11.8 两个版本。 |
| 可用区 | 是 | 根据实际需要选择可用区。支持跨可用区部署。 |
| 节点规格 | 是 | 根据您的业务需要选择合适的节点规格。 |
| 节点数量 | 是 | 根据您的业务需要选择合适的节点数量。 |
| 单节点存储规格 | 是 | 消息存储单独计费，在购买的存储范围内可以无限制保留消息，实际存储消耗可以按等于消息生产流量×保留时间来进行估算。 |
| 私有网络 | 是 | 将新购集群接入点域名绑定至指定的私有网络。 |
| 集群名称 | 是 | 填写集群名称， 3-64个字符，只能包含数字、字母、“-”和“_”。 |
| 镜像队列 | 否 | 建议开启镜像队列保证可用性。开启后会在实例详情-策略列表默认生成一条策略，可以删除，或自定义新策略覆盖该条默认策略。详情参考策略管理。 |

4. 选中我已阅读并同意消息队列 RabbitMQ 版服务条款，然后创建集群。

5. 完成上述步骤后，等待3-5分钟即可在集群管理列表页面看到创建好的集群。

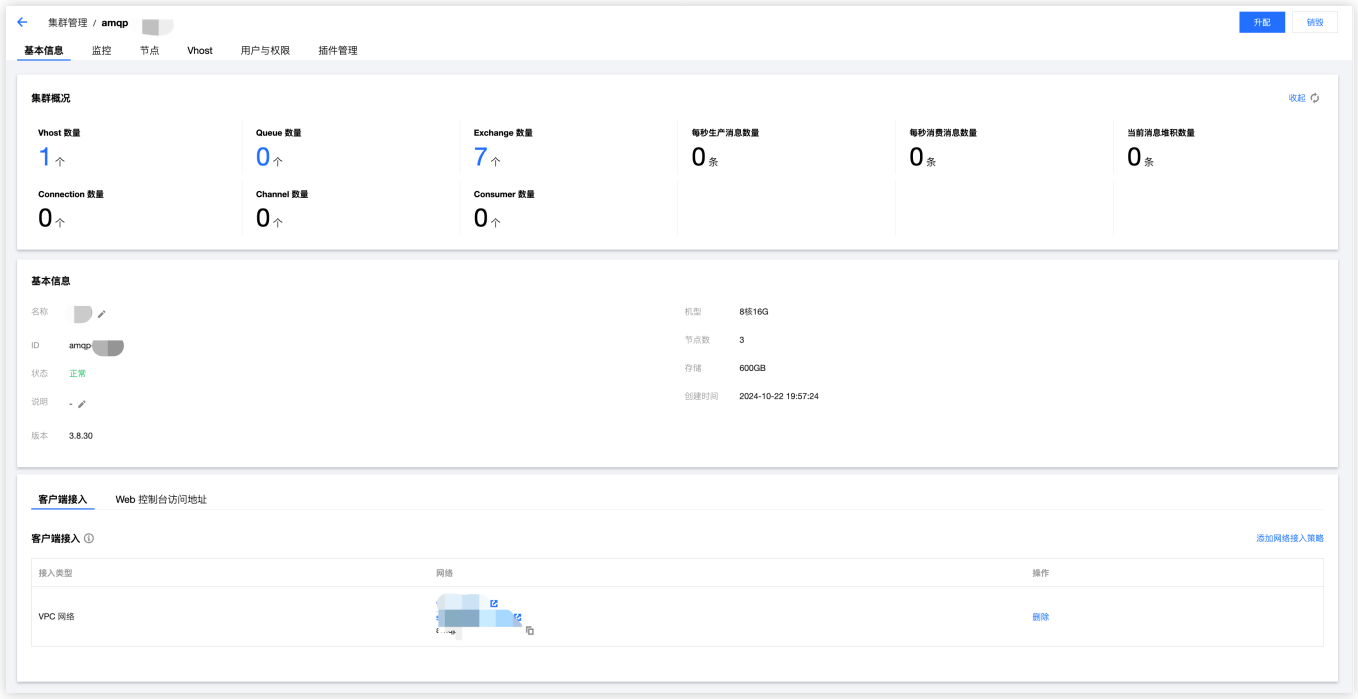
# 查看集群

## 操作场景

本文档指导您使用消息队列 TDMQ RabbitMQ 版时在控制台上查看集群的配置信息和健康状态。

## 操作步骤

1. 登录 RabbitMQ 控制台。
2. 在左侧导航栏选择 集群管理 > 集群列表，单击目标集群的 ID，在基本信息页，可查看集群的资源概况、健康状态、基本信息、接入方式、Web 控制台访问地址等信息。




## 健康状态说明

TDMQ RabbitMQ 版对每个集群均设置有巡检程序，巡检程序会检查该集群的节点状态、磁盘使用率、内存使用率等指标，当这些指标超过一定的阈值后会产生不同的健康状态。说明如下：

| 指标 | 阈值（N） | 状态描述 |
| --- | --- | --- |
| 节点未启动 | - | 异常 |
| 节点连接异常 | - | 异常 |

| 指标 | 阈值（N） | 状态描述 |
| --- | --- | --- |
| 磁盘剩余空间 | N ≥ 100GB | 健康 |
| | 50MB ≤ N < 100GB | 告警 |
| | N < 50MB | 异常 |
| 内存使用率 | N ≤ 60% | 健康 |
| | 60% < N ≤ 90% | 告警 |
| | N > 90% | 异常 |
| socket使用率 | N ≤ 90% | 健康 |
| | N > 90% | 异常 |
| erlang进程使用率 | N ≤ 90% | 健康 |
| | N > 90% | 异常 |
| fd使用率 | N ≤ 90% | 健康 |
| | N > 90% | 异常 |

# 升配集群

## 操作场景

如当前的集群规格不满足您的业务需求，您可以在控制台上提升您的节点规格、节点数量和单节点存储规格。

> 说明：
>
> 当前仅支持提升节点规格、节点数量和节点存储规格，暂不支持降配。

## 操作步骤

1. 登录 RabbitMQ 控制台。
2. 在左侧导航栏选择集群管理 > 集群列表，在集群管理列表页，单击操作列的升配。
3. 选择目标节点规格后，单击确认调整。
   - 目标节点规格：变更节点规格将会涉及资源变更及开源控制台重启，建议先在控制台配置多节点镜像队列后再进行升配，避免升配过程中服务中断及数据丢失。
   - 目标节点数量：支持基数节点数量（根据项目合同确定可选节点）。
   - 单节点存储规格：调整单节点存储后，新的存储规格对集群内所有的节点都生效。
   - 变配时间：可以选择立即执行或者自定义时间（推荐选择夜晚执行，减少对业务的影响）。

# 删除集群

## 操作场景

用户不再需要 TDMQ RabbitMQ 版集群时，可以删除该集群。
TDMQ RabbitMQ 版集群的生命周期是指集群从启动到释放所经历的状态。通过对集群从启动到删除期间的合理的管理，可确保运行于集群上的应用程序能高效经济地提供服务。
集群有以下状态：

| 状态名 | 状态属性 | 状态描述 |
| --- | --- | --- |
| 创建中 | 中间状态 | 集群创建后，进入运行中之前的状态。 |
| 正在运行 | 稳定状态 | 集群正常运行状态，表明您的节点状况、磁盘利用率等指标都处于正常范围内。 |
| 删除中 | 中间状态 | 集群受控制台或通过 API 执行删除操作。 |
| 创建失败 | 中间状态 | 集群受控制台或通过 API 执行购买操作扣费成功但分配集群失败，如遇到这种情况请联系客服进行处理。 |
| 删除失败 | 稳定状态 | 集群被手动删除或者在到期7天后未进行续费，TDMQ RabbitMQ版执行资源释放时失败。 |

## 操作步骤

### 手动删除

针对正在运行的集群，您可以选择手动删除。 操作步骤如下：

1. 登录 RabbitMQ 控制台。
2. 在左侧导航栏选择集群管理 > 集群列表，在集群管理列表页，单击操作列的销毁。
3. 在删除的确认弹框中，二次确认后，即可删除集群。

> 说明：
>
> 删除后，该集群下的所有配置都会被清空，且无法恢复，请谨慎操作。

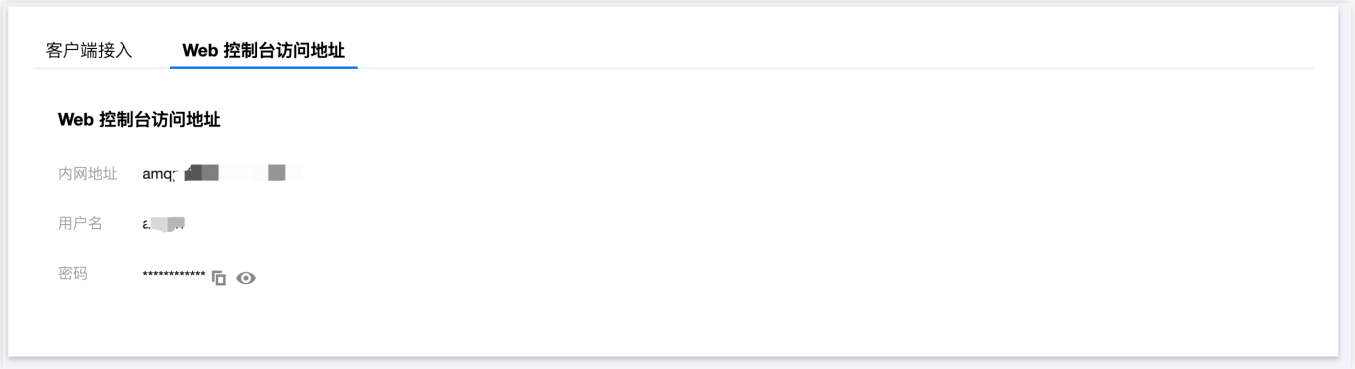
# 访问原生控制台

## 操作场景

本文档指导您使用消息队列 TDMQ RabbitMQ 版时登录访问 RabbitMQ 原生控制台。

## 操作步骤

1. 登录 RabbitMQ 控制台。
2. 在左侧导航栏选择集群管理 > 集群列表，选择好地域后，单击目标集群的 ID，进入集群基本信息页面。
3. 在其他信息 > Web 控制台访问地址 模块，您可以查看内网地址，用户名和密码。并使用该地址和账密访问开源控制台：



# 添加路由策略

## 操作场景

本文档指导您使用消息队列 TDMQ RabbitMQ 版时，通过控制台配置路由接入规则。

## 操作步骤

> 说明：
>
> 一个集群最多创建5条路由。

### VPC网络

操作场景：
您购买集群时选择私有网络并选择了相应的 VPC 环境（例如 VPC A），表示仅能所选择的 VPC A 访问您的 TDMQ RabbitMQ 版服务（生产数据、消费数据等）；若后续使用过程中发现其他 VPC 环境（例如 VPC B）有需求访问 VPC A 内的 TDMQ RabbitMQ 版服务，则可以通过配置接入方式，选择 VPC 网络的路由策略。
操作步骤：

1. 登录 RabbitMQ 控制台。
2. 在左侧导航栏选择集群管理 > 集群列表，选择好地域后，单击目标集群的 ID，进入集群基本信息页面。
3. 在 客户端接入 模块，单击右上角的添加网络接入策略。
4. 在弹窗中，路由类型选择 VPC 网络，选择好 VPC 和子网。

> 说明：
>
> 选择 VPC 网络接入时，支持指定 IP，当变更接入方式时可以通过指定 IP 来保持 IP 不发生变化。

5. 单击提交，完成策略添加。

# 节点管理

## 操作场景

- 节点管理页面展示了当前集群的所有节点列表以及节点状态指标。
- 本文档指导您使用消息队列 TDMQ RabbitMQ 版时在控制台上查看节点列表。

## 前提条件

已 创建集群。

## 操作步骤

### 查看节点

1. 登录 RabbitMQ 控制台。
2. 在左侧导航栏选择集群管理 > 集群列表，选择好地域后，单击目标集群的 ID.
3. 在基本信息页，选择顶部 节点 页签，可以查看当前集群的节点信息。

> 说明：
>
> 在节点列表页，单击操作列的查看监控，可以查看对应的节点的详细监控信息。

# Vhost 管理

## 操作场景

虚拟主机（Virtual Host，简称 Vhost）是 TDMQ RabbitMQ 版中的一个资源管理概念，用作逻辑隔离，不同 Vhost 之间的 Exchange 和 Queue 相互隔离，互不干扰。
用户不同的业务场景一般都可以通过 Vhost 做隔离，并且针对不同的业务场景设置专门的配置，例如消息保留时间。
本文档指导您使用消息队列 TDMQ RabbitMQ 版时，创建多个 Vhost，以便在同一个集群下将 TDMQ RabbitMQ 版应用于不同的场景。

> 说明：
>
> 同一个 Vhost 下的 Exchange 和 Queue 的名称唯一。

## 前提条件

已 创建集群。

## 操作步骤

### 创建 Vhost

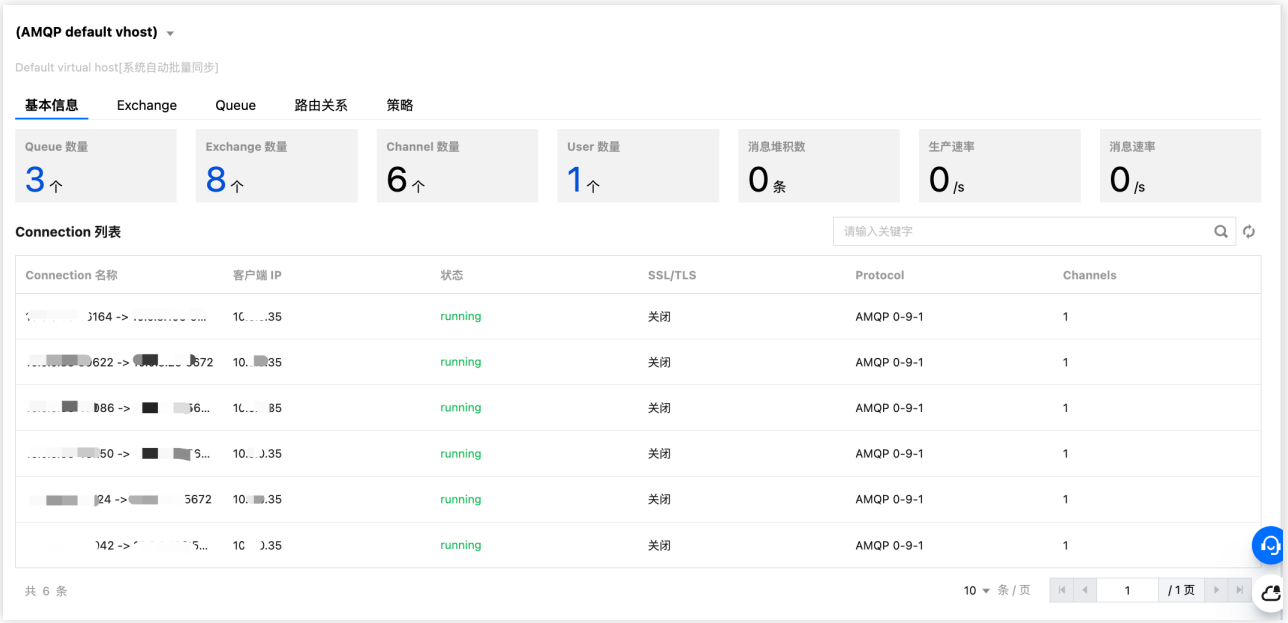
1. 登录 RabbitMQ 控制台。
2. 在左侧导航栏选择集群管理 > Vhost，单击新建进入创建 Vhost 页面。
3. 在新建 Vhost 对话框，设置 Vhost 的相关属性配置。
    - Vhost 名称：设置 Vhost 的名称（创建后不可修改），1-64个字符，只能包含字母、数字、'.'、'-'及'_'
    - 镜像队列：建议开启镜像队列保证可用性。此镜像队列可以在 RabbitMQ 集群中的多个节点上复制队列中的消息，确保在某个节点发生故障时，队列中的消息不会丢失。（单节点集群无法开启）具体参数请见默认镜像策略。
    - 说明：Vhost 的备注说明
4. 单击提交完成所在集群 Vhost 的创建。

后续步骤：接下来就可以在该 Vhost 中 创建 Exchange 和 Queue 进行消息的生产和消费了。

## 查看 Vhost

在 Vhost 列表页，单击要查看的 Vhost 的 ID，进入 Vhost 基本信息页面。

- 概览
  - Queue 数量：展示当前 Vhost 下的 Queue 数量。
  - Exchange 数量：展示当前 Vhost 下的 Exchange 数量。
  - Channel 数量：展示当前 Vhost 下的 Channel 数量
  - User 数量：展示当前 Vhost 的用户数量。
  - 堆积消息数：展示当前 Vhost 的堆积消息数。
  - 生产速率：展示当前 Vhost 的生产速率。
  - 消费速率：展示当前 Vhost 的消费速率。
- Connection 列表
  - 展示当前 Vhost 下的 connection 情况和相应 connection 的 channel 数量




## 修改 Vhost

如果需要重新修改编辑 Vhost 说明，可以通过以下步骤操作：

1. 在 Vhost 列表页，单击操作列的编辑，进入编辑页面。
2. 修改说明，单击提交完成修改。

## 删除 Vhost

如果想删掉创建的 Vhost，可以通过以下步骤操作：

1. 在 Vhost 列表页，单击操作列的删除。
2. 在删除的确认弹框中，单击确定，即可删除 Vhost。

> 注意
>
> Vhost 删除后，该 Vhost 下的所有资源将会被清空，且无法恢复。

# Exchange 管理

## 操作场景

生产者将消息发送到 Exchange 中，Exchange 根据消息的属性或内容将消息路由到一个或多个 Queue 中（或者丢弃），Consumer 从 Queue 中拉取消息进行消费。
该任务指导您使用消息队列 TDMQ RabbitMQ 版时在控制台上创建，删除和查询 Exchange。

## 前提条件

已创建好对应的 Vhost（参见 Vhost 管理）。

## 操作步骤

### 创建 Exchange

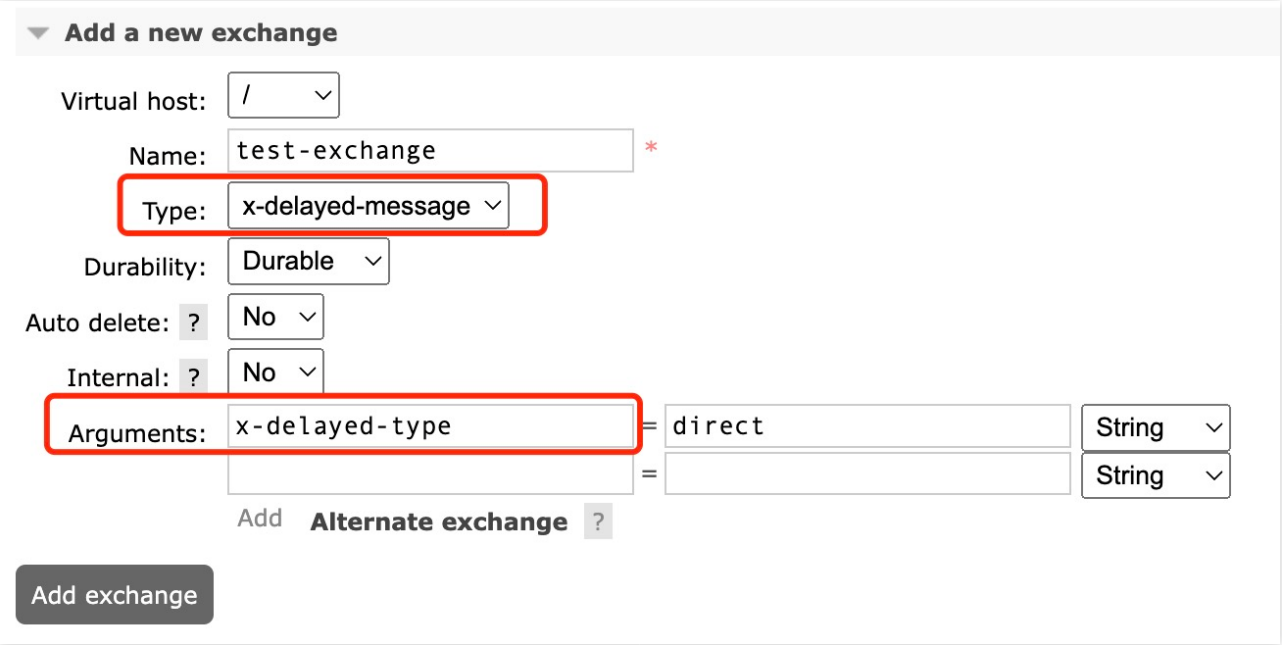
1. 登录 RabbitMQ 控制台。
2. 在左侧导航栏选择集群管理 >**Exchange 页签，选择 Vhost 后，单击**新建进入创建 Exchange 页面。
3. 在新建 Exchange 对话框中，填写以下信息。

- Exchange 名称：填写 Exchange 名称（创建后不可修改），1-64个字符，只能包含字母、数字、“.”、“-”及“_”
- 路由类型：选择路由类型，包括：Direct、Fanout、Topic 和 headers，路由类型选择后不可修改（关于路由类型的详细说明，请参见 Exchange ）
  - Direct：该类型 Exchange 会把消息路由到 RoutingKey 和 BindingKey 完全匹配的 Queue 中
  - Fanout：该类型 Exchange 会将消息路由到所有与其绑定的 Queue 中
  - Topic：该类型 Exchange 支持多条件匹配和模糊匹配，即使用 Routing Key 模式匹配和字符串比较的方式将消息路由至与其绑定的 Queue 中。
  - Headers：与 Routing Key 无关，匹配机制是匹配消息中的 Headers 属性信息。在绑定 Queue 与 Headers Exchange 之前声明一个map键值对，通过这个 map 对象实现消息队列和交换机的绑定。当消息发送到 RabbitMQ 时会取到该消息的 Headers 与 Exchange 绑定时指定的键值对进行匹配；如果完全匹配则消息会路由到该队列，否则不会路由到该队列。
  - X-Delayed-Message：这是一种自定义的 Exchange 类型，用于实现延迟消息传递。当将消息发送到 X-Delayed-Message 类型的交换机时，可以为消息设置一个延迟时间。RabbitMQ 会在延迟时间结束后，将消息路由到与其绑定的队列中。这种类型的交换机允许您在发送消息时控制消息何时被传递给消费者，从而实现

消息的定时和延迟处理。创建此类型 Exchange 方法请看下方说明。

- Durable：如果设置为 true，该Exchange在服务重启后仍然存在；如果设置为false，该 Exchange 在服务重启后消失，需要重建。
- AutoDelete：如果设置为 true，当最后一个绑定到该 Exchange 上的队列被删除后，自动删除该 Exchange。
- Internal：如果设置为 true，则该 Exchange 不能直接被 Producer 使用，而只能与其它 Exchange 绑定。
- Exchange 说明：填写 Exchange 的说明信息，最多128个字符。
- 添加备用 Exchange：可选填，默认不使用备用 Exchange。发送到主 Exchange 的消息如果无法被路由，会被发送到此处指定的备用 Exchange。

> 说明:
>
> 目前控制台暂不支持创建 X-Delayed-Message 路由类型，如有使用需要，请先在开源控制台 - Exchanges 中创建（登录访问 RabbitMQ 开源控制台的操作步骤详情可参见 访问原生控制台）。在创建时，Arguments 需要指定 x-delayed-type，示例如下：




> 需要注意的是，X-Delayed-Message 类型并不是 RabbitMQ 的内置类型，需要通过开启插件 rabbitmq_delayed_message_exchange 来实现，详情请参见 插件管理。

4. 单击提交，在 Exchange 列表中即可看见创建好的 Exchange。

## 编辑 Exchange

1. 在 Exchange 列表中，找到需要编辑的 Exchange ，单击操作栏中的编辑。
2. 在弹出的对话框中可以对 Exchange 的说明进行编辑。
3. 单击提交即完成对 Exchange 的编辑。

## 删除 Exchange

1. 在 Exchange 列表中，找到需要删除的 Exchange ，单击操作列中的删除。
2. 在弹出的提示框中，单击删除，完成删除。

> 注意：
>
> Exchange 删除后，该 Exchange 下的所有配置将会被清空，且无法恢复。

# Queue 管理

## 操作场景

队列（Queue）用于存储消息，每个消息都会被投入到一个或多个 Queue 里，Producer 生产消息并最终投递到 Queue 中，Consumer 可以从 Queue 中拉取消息进行消费。
多个 Consumer 可以订阅同一个 Queue，这时 Queue 中的消息会被平均分摊给多个 Consumer 进行处理，而不是每个 Consumer 都收到所有的消息并处理。
该任务指导您使用消息队列 TDMQ RabbitMQ 版时在控制台上创建，删除和查询 Queue。

## 前提条件

已创建好对应的 Vhost（参见 Vhost 管理）。

## 操作步骤

### 创建 Queue

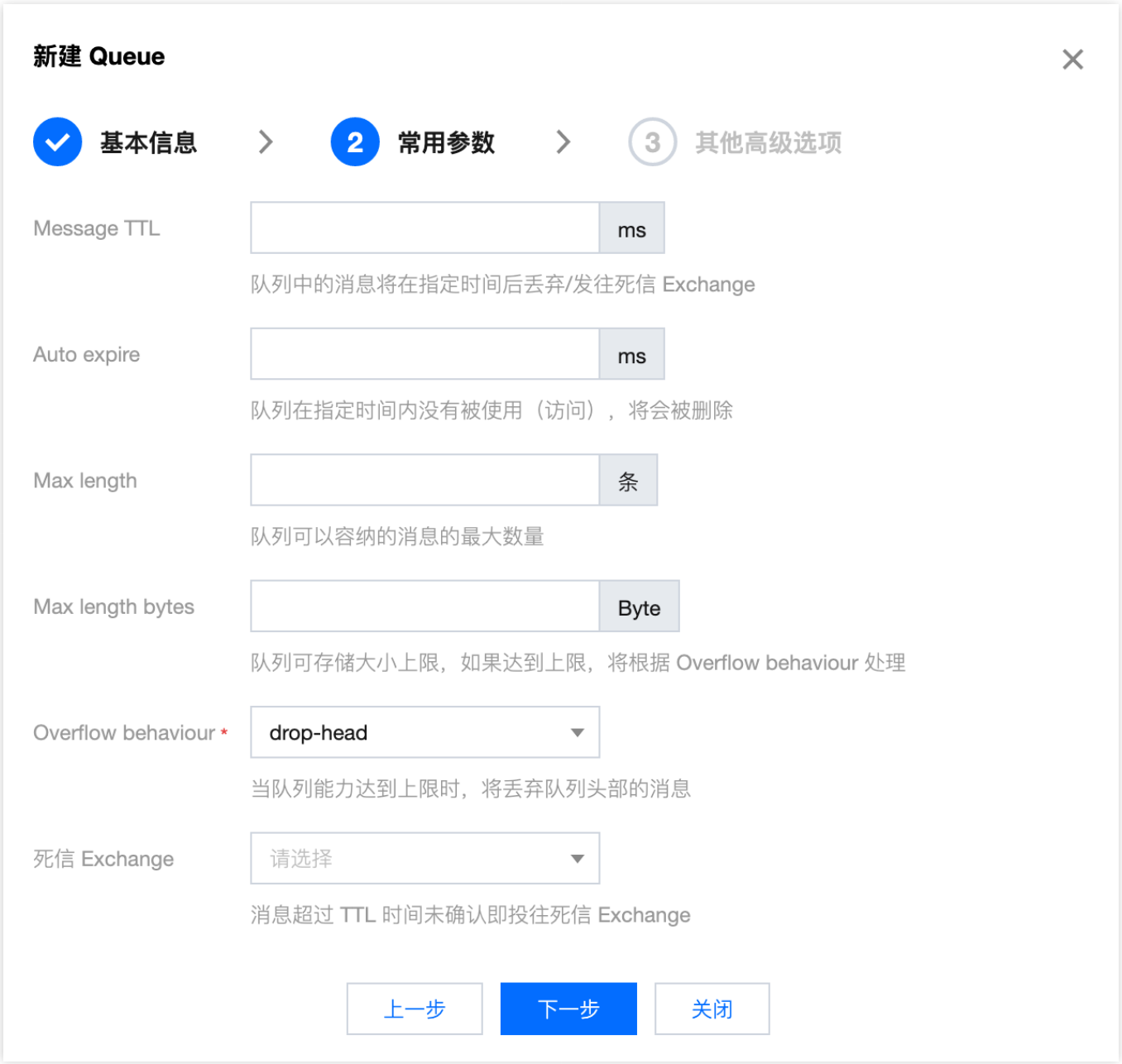
> 说明：
>
> 支持创建普通队列或Quorum队列，可以单击以下页签查看不同类型队列的创建方式。

普通队列

1. 登录 RabbitMQ 控制台。
2. 在左侧导航栏选择集群管理 >**Queue 页签，选择 Vhost 后，单击*新建进入创建 Queue 页面。
3. 填写 Queue 基本信息。

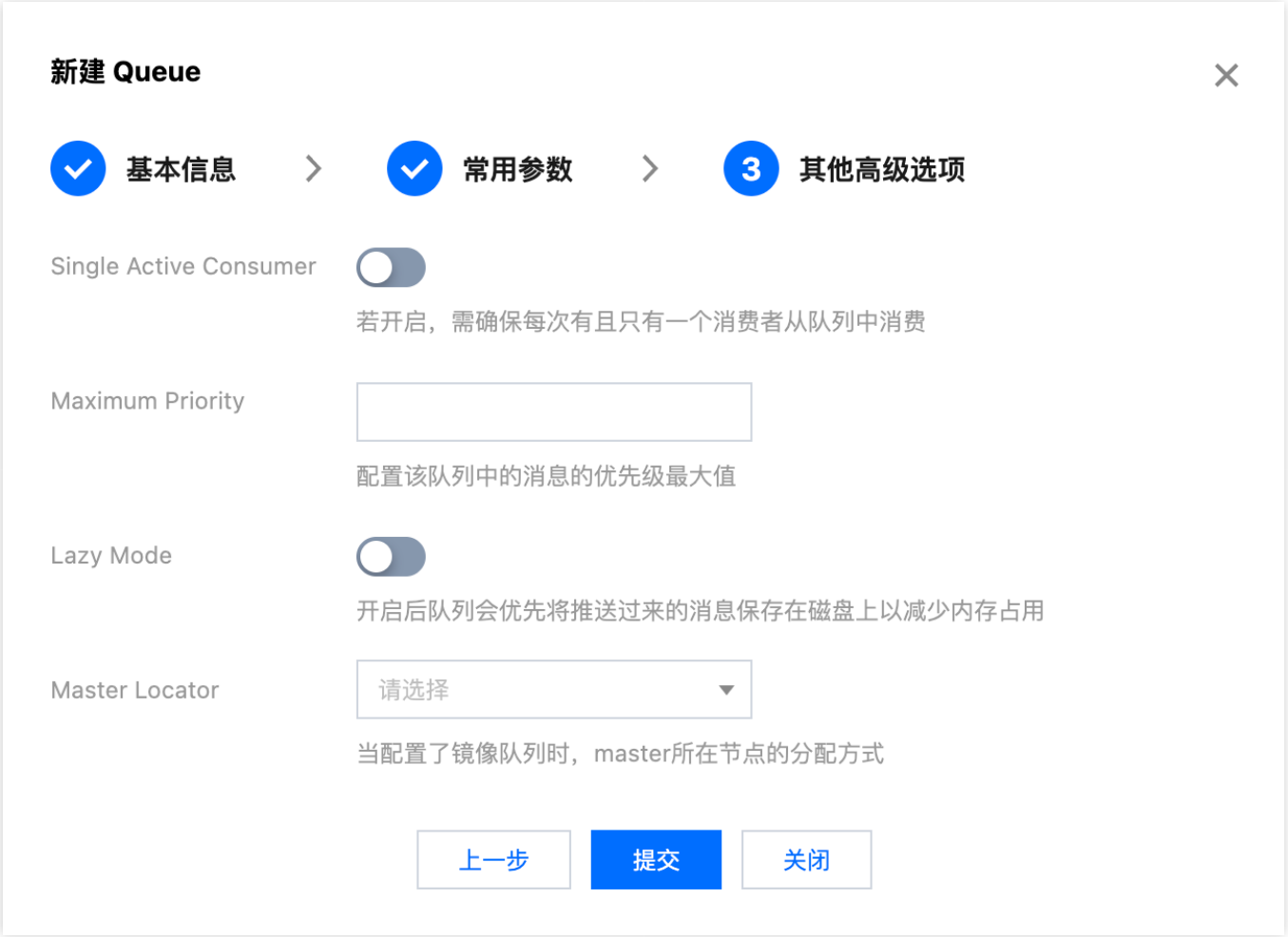
- Queue 名称：填写 Queue 名称（创建后不可修改），3-64个字符，只能包含字母、数字、“-”及“_”
- 类型：普通队列。
- Durable：设置队列是否执行持久化。
- 节点：选择队列所在节点。
- AutoDelete：开启后，最后一个消费者取消订阅后立即删除该 Queue。
- Queue 说明：填写 Queue 说明，最多128个字符。

4. 单击下一步，填写常用参数。




- Message TTL：队列中的消息将在指定时间后丢弃/发往死信 Exchange。
- Auto expire：队列在指定时间内没有被使用（访问），将会被删除。
- Max length：队列可以容纳的消息的最大数量。
- Max length bytes：队列可存储大小上限，如果达到上限，将根据 Overflow behaviour处理。
- Overflow behaviour：当队列能力达到上限时，将丢弃队列头部的消息。
- 死信 Exchange：消息超过 TTL 时间未确认即投往死信 Exchange。

5. 单击下一步，设置其他高级选项。




- Single active consumer：若开启，需确保每次有且只有一个消费者从队列中消费。
- Maximum priority：配置该队列中的消息的优先级最大值。
- Lazy mode：开启后队列会优先将推送过来的消息保存在磁盘上以减少内存占用。
- Master locator：当配置了镜像队列时，master 所在节点的分配方式。
  - min-masters：当配置了镜像队列时，选择托管队列master数量最少的节点，作为当前队列的master所在的节点。
  - client-local：当配置了镜像队列时，选择声明队列的客户端所连接到的节点，作为当前队列的master所在的节点。
  - random：当配置了镜像队列时，选择一个随机节点作为当前队列的master所在的节点。

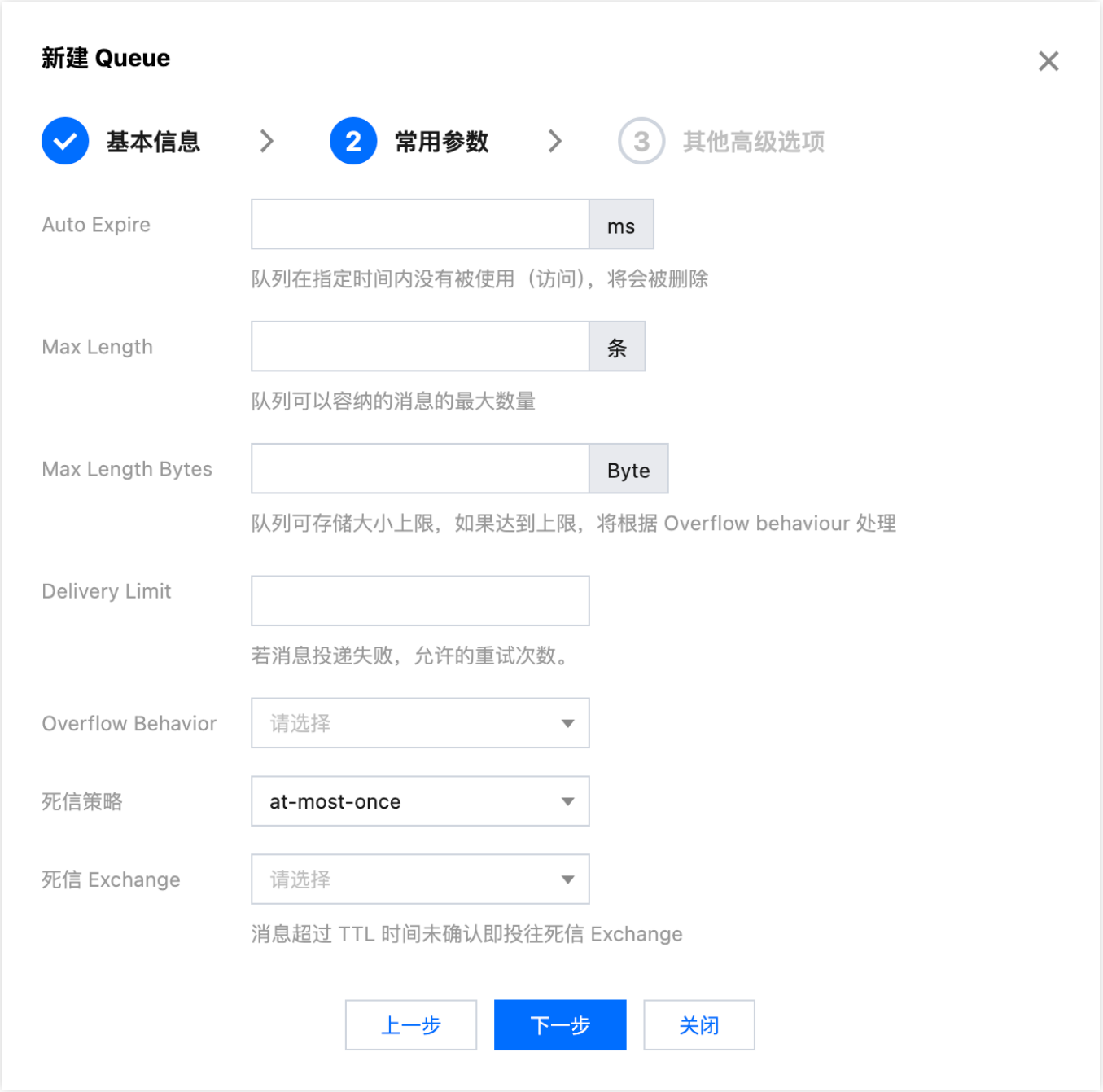
6. 单击提交，完成 Queue 创建。

   Quorum队列

7. 登录 RabbitMQ 控制台。
8. 在左侧导航栏选择集群管理 > Queue，选择 Vhost 后，单击新建进入创建 Queue 页面。
9. 填写 Queue 基本信息。

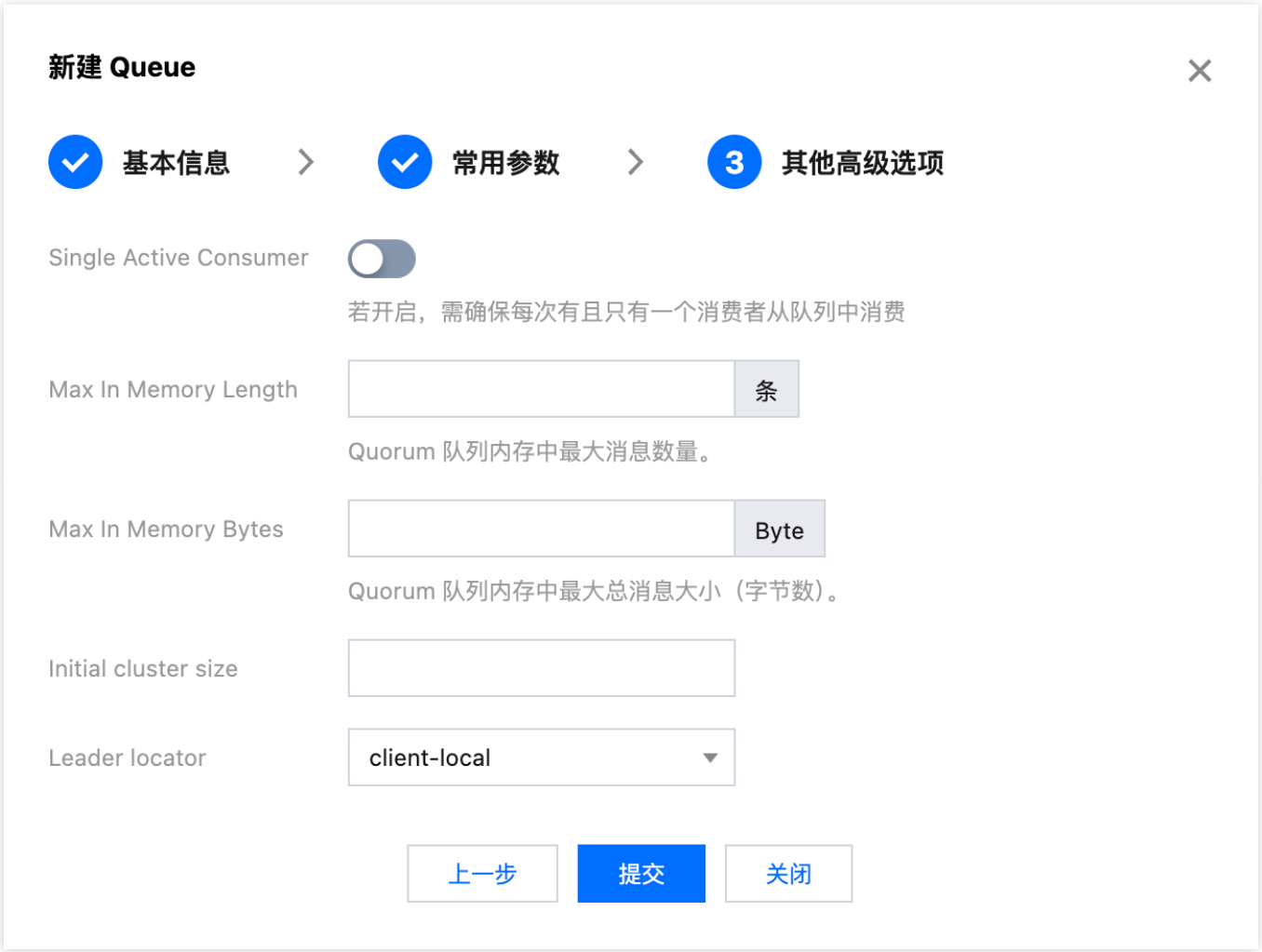
- Queue 名称：填写 Queue 名称（创建后不可修改），3-64个字符，只能包含字母、数字、“-”及“_”
- 类型：Quorum队列。
- 节点：选择队列所在节点。
- Queue 说明：填写 Queue 说明，最多128个字符。

10. 单击下一步，填写常用参数。




- Auto expire：队列在指定时间内没有被使用（访问），将会被删除。
- Max length：队列可以容纳的消息的最大数量。
- Max length bytes：队列可存储大小上限，如果达到上限，将根据 Overflow behaviour处理。
- Delivery Limit：若队列内消息投递失败，允许的重试次数。
- Overflow behaviour：当队列能力达到上限时，将丢弃队列头部的消息。
- 死信策略：可选 at-most-once 和 at-least-once。只有 overflow behavior 选择了 reject-publish 时，死信策略才可以选择 at-least-once。
- 死信 Exchange：消息超过 TTL 时间未确认即投往死信 Exchange。

11. 单击下一步，设置其他高级选项。



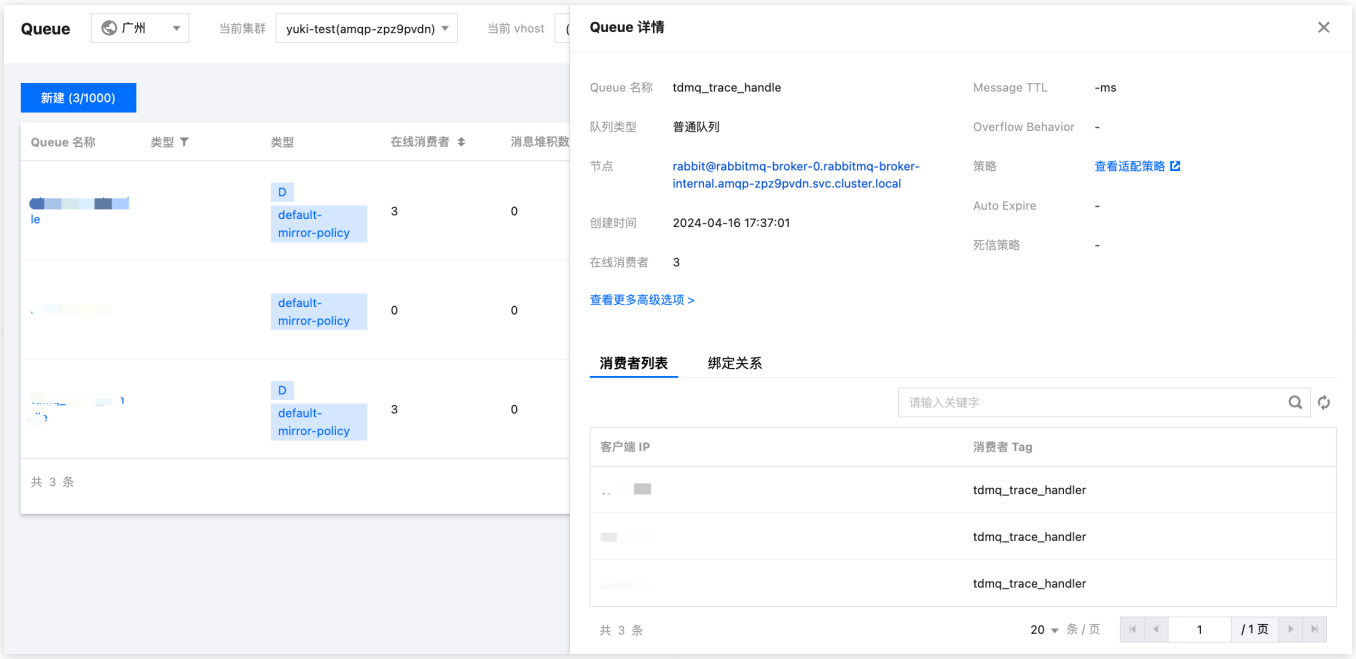

- Single active consumer：若开启，需确保每次有且只有一个消费者从队列中消费。
- Max in memory length：Quorum队列内存中最大消息数量。
- Max in memory bytes：Quorum队列中最大总消息大小（字节数）。
- Initial Cluster Size：Quorum队列初始集群大小。
- Leader Locator：可选 client-local 和 balanced。如果网络延迟对性能影响较大，可以选择 client-local 策略；如果需要平衡各个节点的负载，可以选择 balanced 策略。

12. 单击提交，完成 Queue 创建。

## 查看 Queue 详情

在 Queue 列表，单击 Queue 的“ID”，可查看该 Queue 的详情。
您可以看到：

- 基本信息：展示队列类型、在线消费者、死信 Exchange、AutoDelete 等信息。点击更多高级选项，可以查看该 Queue 所有参数设置信息。
- 消费者列表：展示订阅该 Queue 的消费者信息。
- 绑定关系：展示与该 Queue 绑定的路由关系。




## 编辑 Queue

1. 在 Queue 列表中，单击目标 Queue 操作列的编辑。
2. 在弹窗中，对 Queue 信息进行编辑。
3. 单击提交，完成修改。

## 删除 Queue

1. 在 Queue 列表中，找到需要删除的 Queue，单击操作列的删除。
2. 在弹出的提示框中，单击删除，完成删除。

> 注意：
>
> Queue 删除后，该 Queue 下的所有配置将会被清空，且无法恢复。

# 路由关系

## 操作场景

本文档为您介绍如何在 TDMQ 控制台建立或解除 Exchange 和 Queue 之间的路由关系。

## 前提条件

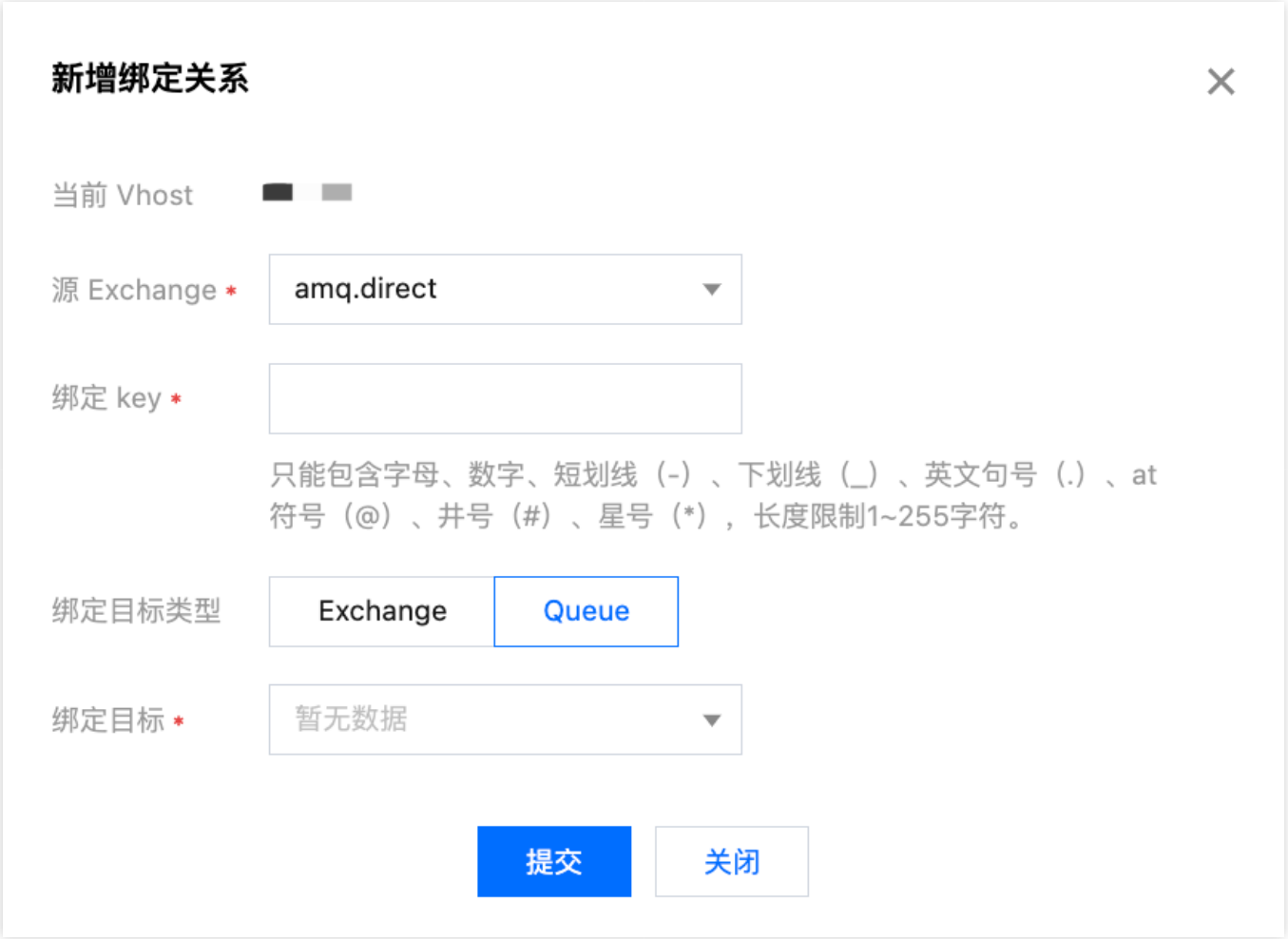
- 已 已创建 Exchange。
- 已 创建 Queue。

## 操作步骤

### 新建绑定关系

绑定路由关系有多个入口：

- 入口一：在左侧导航栏选择集群管理 >*Exchange 列表，单击目标 Exchange 操作栏的*绑定路由。
- 入口二：在左侧导航栏选择集群管理 >*Queue 列表，单击目标 Queue 操作栏的*绑定路由。
- 入口三：在左侧导航栏选择集群管理 >*Vhost，单击目标 Vhost 的“ID”，在页面上方选择 Exchange 或者 Queue 页签，单击目标 Queue 操作栏的*绑定路由。
  在新增绑定关系弹窗中，设置好源 Exchange、绑定 Key，绑定目标类型和绑定目标，单击提交，完成绑定关系创建。




## 解除绑定关系

1. 在路由关系列表，找到需要解除绑定的路由关系，单击操作列的解除绑定。
2. 在弹出的窗口中单击删除，完成路由关系解除。

> 注意：
>
> 路由删除后，该路由将不再提供服务，且无法恢复。

# 用户与权限管理

## 名词解释

- 用户是指在 TDMQ RabbitMQ 版集群内部做权限划分的最小单位，您可以通过为用户配置权限为其赋予不同 Vhost 下的配置和读写权限。
- 用户密码：用户可以通过在客户端中添加用户名和密码来访问 TDMQ RabbitMQ 版集群进行消息的生产消费。
- 权限指的是用户对该 Vhost 下 Exchange，Queue的操作权限，包括配置权限，读写权限。配置权限会影响 Exchange，Queue的声明和删除。读写权限影响从 Queue 里读取消息，向 Exchange 发送消息以及Queue 和 Exchange 的绑定(bind)操作。

## 使用限制

单集群下用户数量上限为20个。

## 使用场景

- 用户需要安全地使用 TDMQ RabbitMQ 版进行消息的生产消费。
- 用户需要对不同的 Vhost 设置不同用户的生产消费权限。
  例如：一个公司有 A 部门和 B 部门，A 部门的系统产生交易数据，B 部门的系统根据这些交易数据做数据分析和展示。那么遵循权限最小化原则，可以创建两个用户，A 部门用户只授予往交易系统 Vhost 中生产消息的权限，B 部门则只授予消费消息的权限。这样可以很大程度避免由于权限不清带来的数据混乱、业务脏数据等问题。

## 操作步骤

### 新增用户

每个集群下都默认有一个名为“admin”的用户，您可以为这个默认用户配置权限，也可以重新新建用户。

1. 登录 RabbitMQ 控制台。
2. 在左侧导航栏选择集群管理 > 集群列表，选择好地域后，单击目标集群的 ID，进入基本信息页面。
3. 在页面顶部选择用户与权限页签，在用户管理页面，单击新建用户。
4. 在新建用户页面，填写用户名称密码和说明：
   - 用户名称：不能为空，不能只输入“.”，1-64个字符，只能包含字母、数字、“.”、“-”及“_”。
   - 用户密码：不能为空，8-64个字符，至少要包含小写字母、大写字母、数字、特殊字符【()`~!@#$%^&*_=|{}[]:;',.?/】中的两项。

- 角色：选择用户角色。

| 角色 | 权限说明 |
| --- | --- |
| none | 无法登录 Web 控制台，通常是普通的生产者和消费者。 |
| management | 可以登录 Web 控制台；<br>可以查看其名下的 Vhost，以及其中的 queue、exchange 和 binding；<br>可以查看和关闭其名下的 channe l和 connection。 |
| policymaker | 在 “management” 所有权限的基础上：<br>可以查看、修改、删除其名下的 Vhost 的策略和参数。 |
| monitoring | 在“management”所有权限的基础上：<br>可以查看所有 Vhost、connection 和 channel 列表；<br>可以查看节点相关信息（如磁盘使用情况、内存使用情况、进程数等）。 |
| administrator | 超级管理员，在 “policymaker” 和 “monitoring” 所有权限的基础上：<br>可以创建和删除 Vhost；<br>可以查看、创建和删除用户和权限；<br>关闭其他用户的 connection。 |

- 说明（选填）：填写用户说明。

5. 单击提交，完成当前集群的用户创建。

## 配置权限

1. 在用户与权限页面，选择权限列表页签，进入权限列表，单击配置权限。
2. 在权限配置页面，选择好需要配置权限的 Vhost 和用户，并设置好权限规则。
   权限规则设置支持使用正则表达式匹配资源。例如，勾选“配置”且在输入框内输入 “test-.*”，则表示授权给该用户当前 Vhost 下，所有名称以 “test-” 开头的资源的配置权限。

3. 单击提交，完成权限配置。
4. 将用户名和密码添加到客户端的参数中。如何在客户端代码中添加密钥参数请参考 RabbitMQ 的 SDK文档，对应其中的 Username 和 password。
5. 检查权限是否生效。您可以运行配置好的客户端访问对应 Vhost 中的 Exchange 和 Queue 资源，按照刚刚配置的权限进行生产或消费，看是否会产生没有权限的报错信息，如果没有即代表配置成功。

## 删除权限

删除权限前请确保当前业务已经没有使用该用户进行消息的生产消费再进行此项操作，否则可能会出现客户端无法生产消费而导致的异常。

1. 在用户与权限列表页面，找到需要删除权限的权限，单击操作列的删除。
2. 在删除的弹框中，单击确认，即可删除该权限。

# 监控告警
# 查看监控指标

## 操作场景

TDMQ RabbitMQ 版支持监控您账户下创建的资源，包括集群、节点、Vhost、Queue 和 Exchange，帮助您实时掌握资源状态，针对可能存在的问题及时处理，保障其稳定运行。
本文为您介绍通过 RabbitMQ 控制台查看监控指标的操作方法和监控指标的含义。

## 监控指标

TDMQ RabbitMQ 版专享集群支持查看集群、节点、Vhost、Queue 和 Exchange 五个维度的监控数据，支持的监控指标如下：
集群

| 分类 | 监控指标 | 单位 | 监控指标含义 |
| --- | --- | --- | --- |
| 基本信息 | 连接数量 | Count | 当前打开的连接数量 |
| | 通道数量 | Count | 当前打开的通道数量 |
| | 队列数量 | Count | 当前可用队列总数 |
| | 消费者数量 | Count | 当前在线消费者总数 |
| | 堆积消息数量 | Count | ready 状态(堆积未投递)的消息总数 |
| | 推荐 TPS 上限（生产+消费） | Count/s | 集群未开启镜像队列时的 TPS 上限推荐值 |
| 生产消费 | 生产确认速率 | Count/s | client 消息生产成功后，broker 确认回包速率 |
| | 每秒生产消息数量 | Count/s | client 端生产消息速率 |
| | 消费未确认消息数量 | Count | 已投递至消费者但未确认的消息总数 |
| | 消费确认速率 | Count/s | 消费者确认的消息速率 |

| 分类 | 监控指标 | 单位 | 监控指标含义 |
| --- | --- | --- | --- |
| | 每秒消费消息数量 | Count/s | 总体每秒消费消息速率，包括 autoAck=false 和 autoAck=true 两种情况 |
| | 重投递速率 | Count/s | channel 中重投递到消费者的消息速率 |
| | 消息丢弃速率 | Count/s | mandatory=false 的情况下发送至 exchange，并且没有符合的路由条件导致消息丢弃(drop)速率 |
| | 延迟消息数量 | Count | 当前集群下延迟消息数量 |

节点

| 分类 | 监控指标 | 单位 | 监控指标含义 |
| --- | --- | --- | --- |
| 基本信息 | 连接数量 | Count | 当前打开的连接数量 |
| | 通道数量 | Count | 当前打开的通道数量 |
| | 队列数量 | Count | 当前可用队列总数 |
| | 消费者数量 | Count | 当前在线消费者总数 |
| | 堆积消息数量 | Count | ready 状态(堆积未投递)的消息总数 |
| | CPU利用率 | % | 节点 CPU 利用率 |
| | 内存利用率 | % | 节点内存利用率 |
| | 磁盘利用率 | % | 节点磁盘利用率 |
| | 内存占用 | GBytes | 节点内存占用大小 |
| 生产消费 | 消费未确认消息数量 | Count | 已投递至消费者但未确认的消息总数 |
| | 消费确认速率 | Count/s | 消费者确认的消息速率 |
| | 每秒消费消息数量 | Count/s | 总体每秒消费消息速率，包括 autoAck = false 和 autoAck = true 两种情况 |
| | 重投递速率 | Count/s | channel 中重投递到消费者的消息速率 |
| | 消息丢弃速率 | Count/s | mandatory = false 的情况下发送至 exchange，并且没有符合的路由条件导致消息丢弃(drop)速率 |

| 分类 | 监控指标 | 单位 | 监控指标含义 |
| --- | --- | --- | --- |
| | 延迟消息数量 | Count | 当前节点下延迟消息数量 |

Vhost

| 分类 | 监控指标 | 单位 | 监控指标含义 |
| --- | --- | --- | --- |
| 基本信息 | 消费者数量 | Count | 当前在线消费者总数 |
| | 堆积消息数量 | Count | ready 状态(堆积未投递)的消息总数 |
| 生产消费 | 生产确认速率 | Count/s | client 消息生产成功后，broker 确认回包速率 |
| | 每秒生产消息数量 | Count/s | client 端生产消息速率 |
| | 消费未确认消息数量 | Count | 已投递至消费者但未确认的消息总数 |
| | 消费确认速率 | Count/s | 消费者确认的消息速率 |
| | 每秒消费消息数量 | Count/s | 总体每秒消费消息速率，包括 autoAck=false 和 autoAck=true 两种情况 |
| | 重投递速率 | Count/s | channel 中重投递到消费者的消息速率 |
| | 消息丢弃速率 | Count/s | mandatory = false 的情况下发送至 exchange，并且没有符合的路由条件导致消息丢弃(drop)速率 |

Queue

| 分类 | 监控指标 | 单位 | 监控指标含义 |
| --- | --- | --- | --- |
| 基本信息 | 消费者数量 | Count | 当前在线消费者总数 |
| | 堆积消息数量 | Count | ready 状态(堆积未投递)的消息总数 |
| 生产消费 | 消费未确认消息数量 | Count | 已投递至消费者但未确认的消息总数 |
| | 消费确认速率 | Count/s | 消费者确认的消息速率 |
| | 重投递速率 | Count/s | channel 中重投递到消费者的消息速率 |

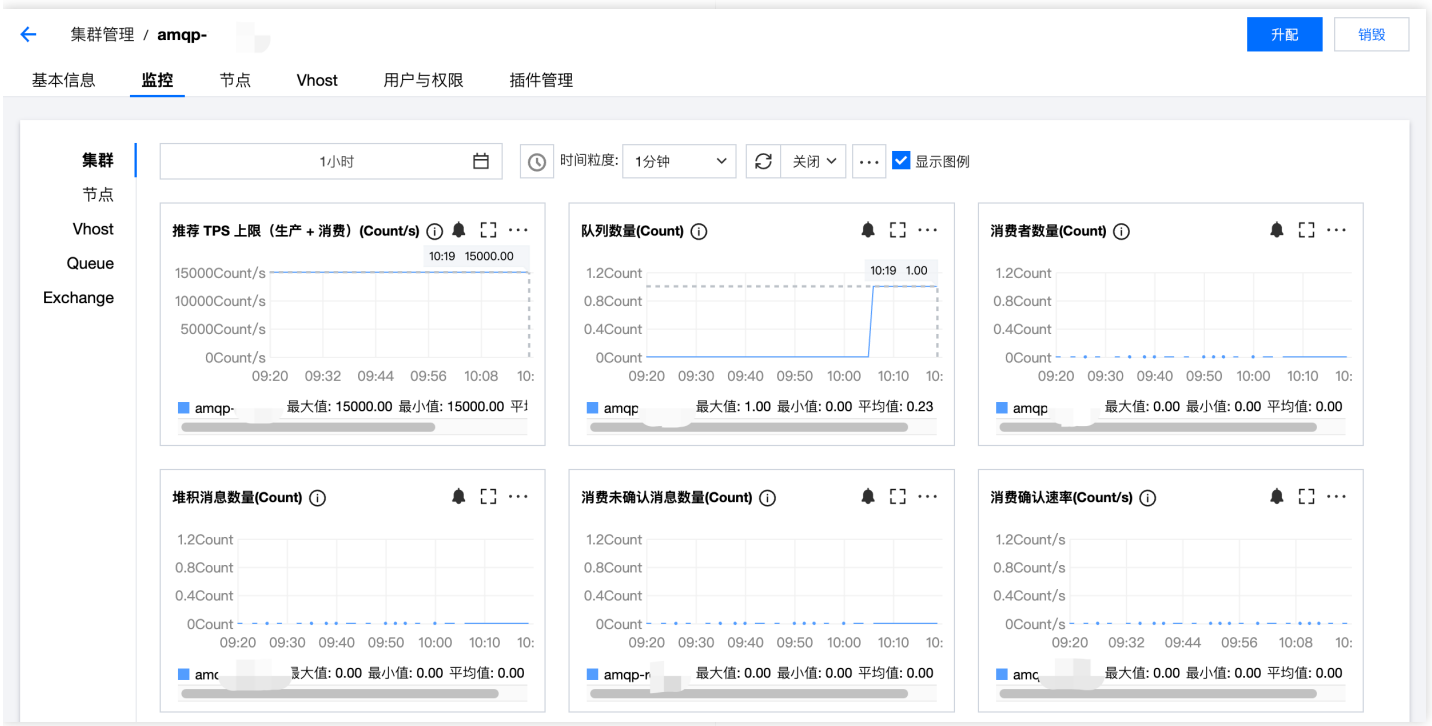
Exchange

| 分类 | 监控指标 | 单位 | 监控指标含义 |
| --- | --- | --- | --- |
| 生产消费 | 延迟消息数量 | Count | 当前 Exchange 下延迟消息数量。 |

# 查看监控数据

1. 登录 RabbitMQ 控制台。
2. 在左侧导航栏选择集群管理 > 集群列表，选择好地域后，单击目标集群的 ID，进入基本信息页面。
3. 在集群详情页顶部，选择监控页签，进入监控页面。
4. 选择要查看的资源页签，选择要查看的资源，并设置好时间范围后，查看对应的监控数据。

| 图标 | 说明 |
| --- | --- |
| | 单击可查看监控指标同环比，支持同比、环比和自定义日期对比。 |
| | 单击可刷新获取最新的监控数据，支持设置30s、5min、30min和1h时间间隔自动刷新监控数据。 |
| | 勾选后可在图表上显示图例信息。 |


集群管理 / amqp-
升配
销毁
基本信息
监控
节点
Vhost
用户与权限
插件管理
集群
节点
Vhost
Queue
Exchange
1小时
时间粒度:
1分钟
关闭
显示图例
推荐 TPS 上限（生产 + 消费）(Count/s)
15000Count/s
10000Count/s
5000Count/s
0Count/s
09:20 09:32 09:44 09:56 10:08
amqp-
最大值: 15000.00 最小值: 15000.00
队列数量(Count)
1.2Count
0.8Count
0.4Count
0Count
09:20 09:30 09:40 09:50 10:00 10:10
10:19 1.00
amqp
最大值: 1.00 最小值: 0.00 平均值: 0.23
消费者数量(Count)
1.2Count
0.8Count
0.4Count
0Count
09:20 09:30 09:40 09:50 10:00 10:10
amqp
最大值: 0.00 最小值: 0.00 平均值: 0.00
堆积消息数量(Count)
1.2Count
0.8Count
0.4Count
0Count
09:20 09:30 09:40 09:50 10:00 10:10
最大值: 0.00 最小值: 0.00 平均值: 0.00
消费未确认消息数量(Count)
1.2Count
0.8Count
0.4Count
0Count
09:20 09:30 09:40 09:50 10:00 10:10
amqp-r
最大值: 0.00 最小值: 0.00 平均值: 0.00
消费确认速率(Count/s)
1.2Count/s
0.8Count/s
0.4Count/s
0Count/s
09:20 09:32 09:44 09:56 10:08
最大值: 0.00 最小值: 0.00 平均值: 0.00

# 配置告警

## 操作场景

TDMQ RabbitMQ 版支持监控您账户下创建的资源，包括集群、节点、Vhost、Queue 和 Exchange，帮助您实时掌握资源状态。您可以为监控指标配置告警规则，当监控指标达到设定的报警阈值时，云监控平台可以通过邮件、短信等方式通知您，帮助您及时应对异常情况。

## 配置告警规则

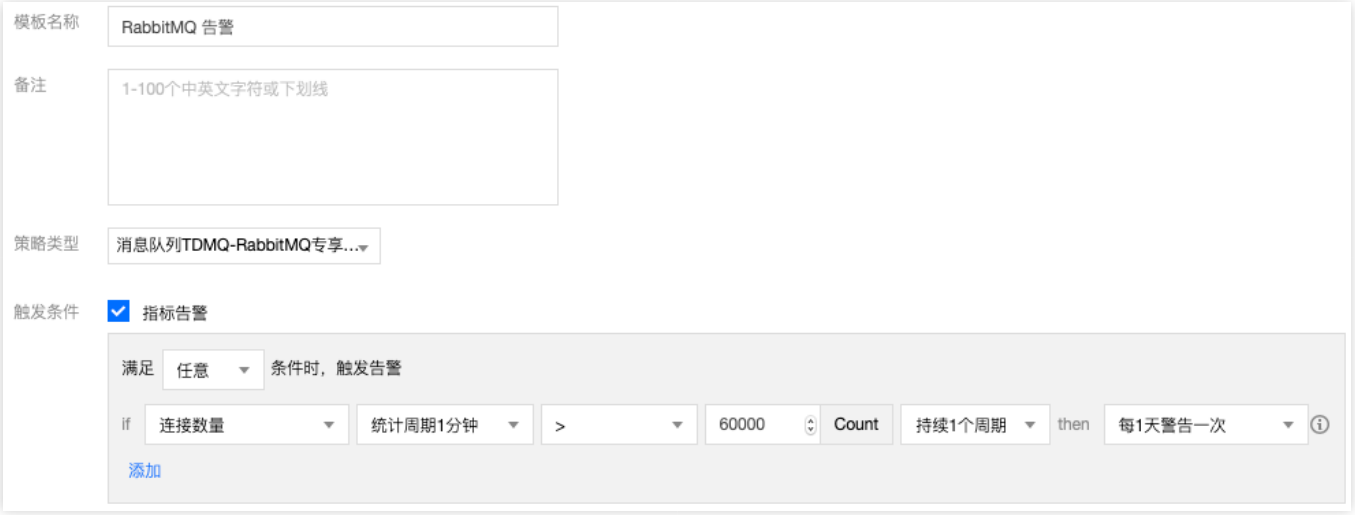
### 新建告警规则

您可以为监控指标配置告警规则，当监控指标达到设定的报警阈值时，云监控平台可以通过邮件、短信等方式通知您，帮助您及时应对异常情况。

1. 进入“云监控”产品页面。
2. 在“告警配置 > 告警策略”页面，单击“新建”，选择好策略类型和要设置告警的实例，设置好告警规则和告警通知模板。
    - 监控类型：选择云产品监控。
    - 策略类型：选择消息队列TDMQ / RabbitMQ。
    - 告警对象：选择需要配置告警策略的 RabbitMQ 资源。
    - 触发条件：支持选择模板和手动配置，默认选择手动配置，新建模板可在“触发条件模板”页面操作。
    - 通知模板：选择通知模板，也可以新建通知模板，设置告警接收对象和接收渠道。
3. 单击完成，完成配置。

### 新建触发条件模板

1. 进入“云监控”产品页面。
2. 在左侧导航栏中，单击触发条件模板，进入触发条件列表页面。
3. 在触发条件模板页单击新建。
4. 在新建模板页，配置策略类型。
    - 策略类型：选择消息队列 TDMQ / RabbitMQ。
    - 触发条件：勾选“指标告警”，设置告警策略。
5. 确认无误后，单击保存。




6. 返回新建告警策略页，单击刷新，就会出现刚配置的告警策略模板。

# 告警配置建议

本节为您介绍在使用 TDMQ RabbitMQ 版过程中需要重点关注的一些指标及其告警建议配置：

| 指标 | 维度 | 告警建议配置 | 详细说明 |
| --- | --- | --- | --- |
| 磁盘利用率（%） | 节点 | 统计粒度1分钟，“磁盘利用率”数值 > 80%，持续5个数据点，每30分钟告警一次 | 磁盘使用率过高会导致节点没有足够的磁盘空间容纳消息分配到该节点上，从而导致消息无法落盘。建议在平均磁盘使用率超过80%时及时清理数据或扩容集群。 |
| 内存利用率（%） | 节点 | 统计粒度1分钟，“内存利用率”数值 > 50%，持续5个数据点，每30分钟告警一次 | 内存利用率过高会阻塞消息生产。建议在内存利用率超过50%时及时加快消费、对生产进行流控或扩容集群。 |
| CPU利用率（%） | 节点 | 统计粒度1分钟，“CPU利用率”数值 > 70%，持续5个数据点，每30分钟告警一次 | CPU 利用率过高会影响消息生产速度。建议在 CPU 利用率超过70%时及时扩容。 |
| 堆积消息数量（Count） | 节点 | 统计粒度5分钟，“堆积消息数量”数值 > 业务预期堆积消息数量，持续5个数据点，每30分钟告警一次 | 堆积过多的消息会导致 Broker 节点磁盘使用率迅速上涨，无法再接入更多消息。需要进行扩容。 |
| 节点存活状况（%） | 节点 | 统计粒度1分钟，“节点存活状况”数值 = 1，持续3个数据点，每15分钟告警一次 | 节点存活状况异常/宕机会导致消息丢失，尤其是在没有开启持久化或镜像队列的情况下。同时会增加其余节点的负载，从而导致集群性能下降。建议结合其他指标和告警信息检查原因。 |

> 说明：
>
> - 指标：例如“连接数量”，选择统计粒度为1分钟，则在1分钟内，生产时延平均耗时连续N个数据点超过阈值，就

会出发告警。

- 告警频次：例如“每30分钟警告一次”，指每30分钟内，连续多个统计周期指标都超过了阈值，如果有一次告警，30分钟内就不会再次进行告警，直到下一个30分钟，如果指标依然超过阈值，才会再次告警。

# 策略列表
# 策略管理

## 操作场景

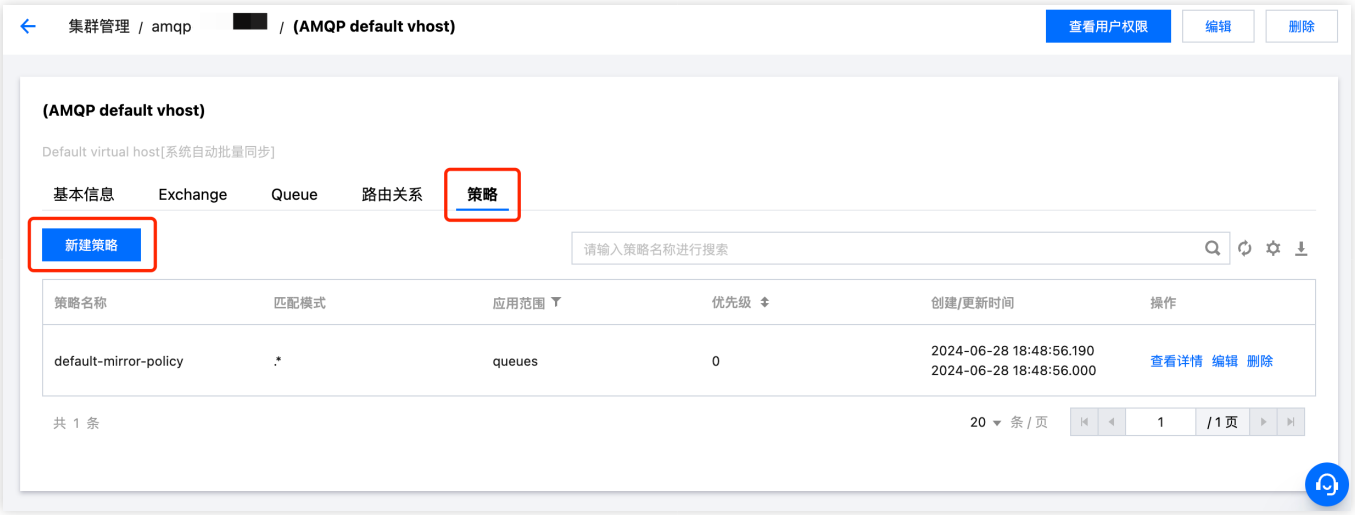
在消息队列 RabbitMQ 中，除了强制属性，例如 durable、Exclusive 等，在创建 Queue 或者 Exchange 时还可以配置一些可选的属性来获得不同的功能，比如：x-message-ttl、x-expires、x-max-length 等。
但是，通过 RabbitMQ 客户端为 Queue 或者 Exchange 设定的属性参数一旦设置成功就不能再改变，除非将原来的 Queue 或者 Exchange 删除，重新创建新的 Queue 或者 Exchange 。
策略是一种特殊的运行时参数的用法，支持动态地修改一些属性参数，策略针对 Vhost 级别，一条策略可以匹配一个或多个 Queue 或者 Exchange ，便于批量管理。这就解决了 RabbitMQ 客户端创建的交换器和队列不能修改的问题，也大大提高了应用的灵活性。

## 操作步骤

### 新建策略

在新建集群时，若开启了镜像队列，在控制台 策略 页签下默认会有一条策略，可以删除，您也可以重新新建改策略。

1. 登录 RabbitMQ 控制台。
2. 在左侧导航栏选择集群管理 > Vhost，选择好地域后，单击目标 Vhost 的 ID，进入基本信息页面。
3. 单击策略 > 新建策略。



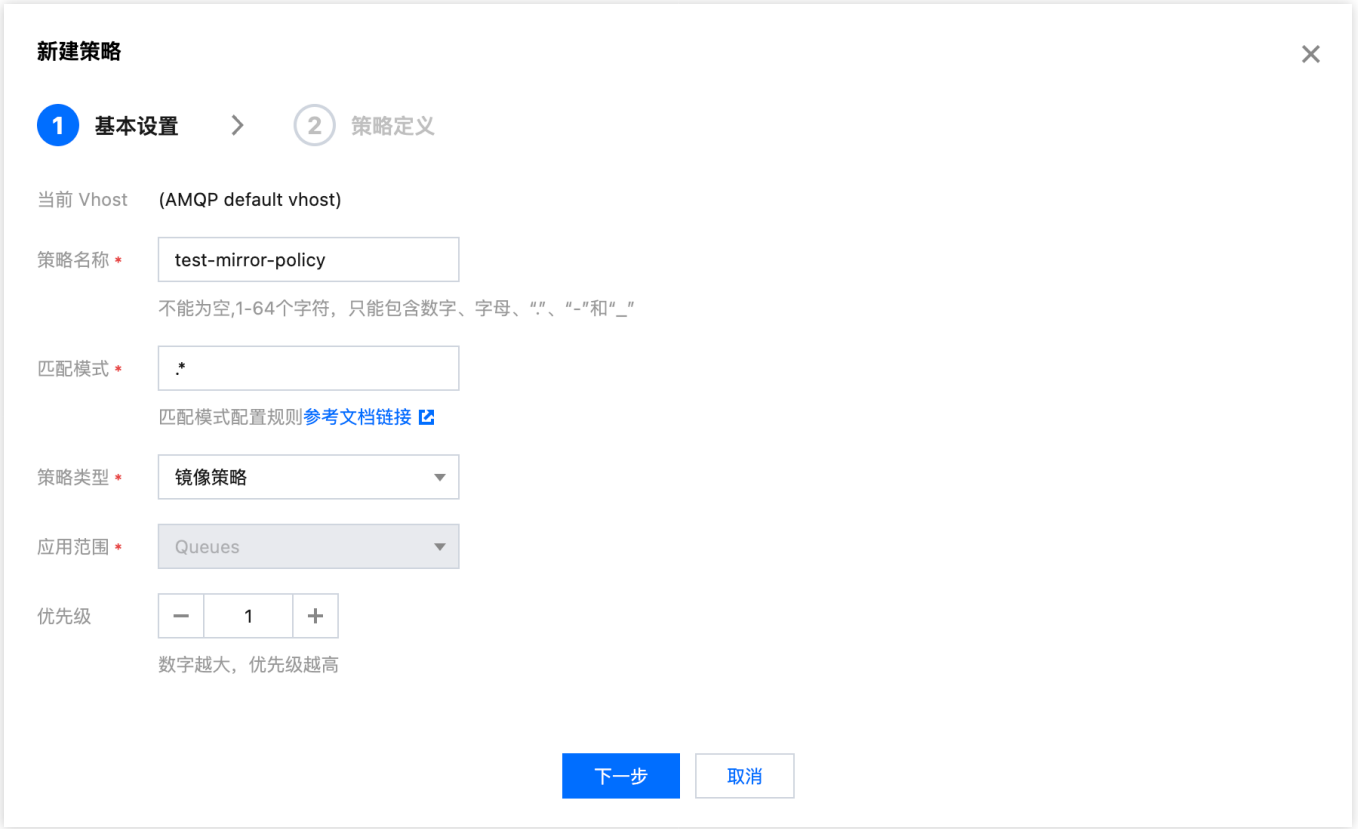

4. 填写策略基本信息。控制台支持新建镜像策略或者自定义策略，具体说明如下：
   镜像策略
   基本设置

- 当前 Vhost：表示正在给哪个 Vhost 创建镜像策略。

- 策略名称：1-64个字符，只能包含数字、字母、“.”、“-”和“_”。
- 匹配模式：一个正则表达式，用来匹配相关的 Queue。常用匹配模式的正则表达式可参考下表：

| 正则表达式 | 含义 |
| --- | --- |
| .* | 将匹配该 Vhost 下的所有 Queue。 |
| ^test.* | 将匹配该 Vhost 下名称以“test”开头的所有 Queue。 |
| *.test.* | 将匹配该 Vhost 下名称包含“test”的所有 Queue。 |
| .*test$ | 将匹配该 Vhost 下名称以“test”结尾的所有 Queue。 |

- 策略类型：选择 镜像策略。
- 应用范围：用来指定当前 Policy 生效的范围。镜像策略仅支持在 Queue 中生效。
- 优先级：定义策略的优先级。如果有多个策略作用于同一个 Queue，那么优先级数字最大的那个 Policy 才会有用。




策略定义

- 镜像模式：镜像队列的模式，有效值为 all/exactly/nodes。
  - all：表示在集群中所有的节点上进行镜像。
  - exactly: 表示在指定个数的节点上进行镜像，节点的个数由镜像参数指定。

  - nodes: 表示在指定的节点上进行镜像，节点名称通过镜像参数指定。

- 镜像参数：表示消息将被同步到的节点。
  - 当镜像模式选择 all 时，该项不用填。
  - 当镜像模式选择 exactly 时，镜像参数可选至少 1 个，最多等于当前集群节点数。
  - 当镜像模式选择 nodes 时，镜像参数可按照节点名称，选到具体哪些节点。

- 消息同步方式：镜像队列中消息的同步方式，可选 automatic 或者 manual。

- 主节点退出处理：当主节点优雅退出时，是否允许选举未同步的镜像为 master。

- 主节点故障处理：当主节点故障/失败时，是否允许选举未同步的镜像为 master。为保证可用性，建议保持为“允许选择所有镜像”。

自定义策略
基本设置

- 当前Vhost：表示正在给哪个 Vhost 创建镜像策略

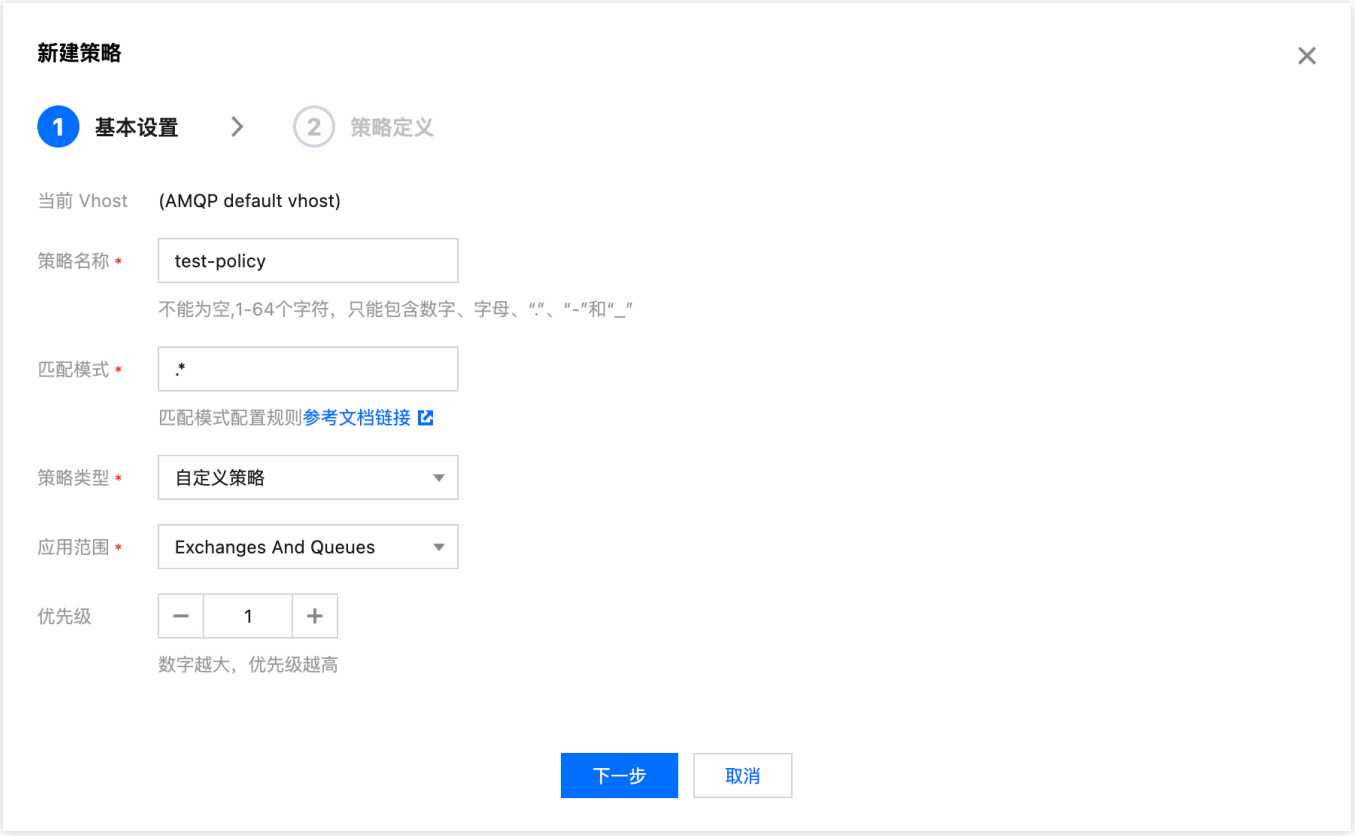
- 策略名称：1-64个字符，只能包含数字、字母、“.”、“-”和“_”

- 匹配模式：一个正则表达式，用来匹配相关的 Queue 或者 Exchange。常用匹配模式的正则表达式可参考下表：

| 正则表达式 | 含义 |
| --- | --- |

| | |
|---|---|
| .* | 将匹配该 Vhost 下的所有 Queue 或者 Exchange。 |
| ^test.* | 将匹配该 Vhost 下名称以“test”开头的所有 Queue 或者 Exchange。 |
| .*test.* | 将匹配该 Vhost 下名称包含“test”的所有 Queue 或者 Exchange。 |
| .*test$ | 将匹配该 Vhost 下名称以“test”结尾的所有 Queue 或者 Exchange。 |

- 策略类型：选择 镜像策略。
- 应用范围：用来指定当前 Policy 生效的范围
  - Exchanges And Queues：表示作用于与 Pattern 所匹配的所有 Queue 或者 Exchange。
  - Queues：表示作用于与 Pattern 所匹配的所有 Queue。
  - Exchanges：表示作用于与 Pattern 所匹配的所有 Exchange。
- 优先级：定义策略的优先级。如果有多个策略作用于同一个 Queue 或者 Exchange，那么优先级数字最大的那个 Policy 才会有用。




策略定义

- 您可以自己选择所需要的策略定义字段，如下图。

- 目前云控制台支持的字段如下表。如需更多字段，请到开源控制台创建。

| 字段分类 | 字段名称 | 含义 | 数据类型限制 |
| --- | --- | --- | --- |
| Queues (All Types) | Max Length | 队列中允许的最大消息数。当队列中的消息数达到此限制时，根据Overflow Behaviour 设置，将会删除旧消息或拒绝新消息。 | Number |
| Queues (All Types) | Max Length Bytes | 队列中允许的最大消息字节总数。当队列中的消息字节总数达到此限制时，根据Overflow Behaviour 设置，将会删除旧消息或拒绝新消息。 | Number |
| Queues (All Types) | Overflow Behaviour | 当队列达到最大长度或最大字节限制时的处理方式。合法值为：<br>◦ drop-head：删除队列头部的旧消息<br>◦ reject-publish：拒绝新发布的消息 | String |
| Queues (All Types) | Auto Expire | 队列的自动过期时间。单位为毫秒。当队列在此时间内没有被访问（例如：没有发布消息、消费消息或检查队列状态），队列将被删除。 | Number |
| Queues (All Types) | Dead letter exchange | 死信交换机。当消息因为超过TTL、达到队列最大长度或被消费者拒绝而被删除时，这些消息将被发送到指定的死信交换机。 | String |
| Queues (All Types) | Dead letter routing key | 死信路由键。当消息被发送到死信交换机时，可以使用此路由键进行路由。 | String |
| Queues [Classic] | Message TTL | 消息的生存时间（Time to Live）。单位为毫秒。这个值定义了消息在队列中可以存活的最长时间。当消息在队列中的存活时间超过这个值时，消息将被删除。如果消息被消费者消费（并确认）或者被重新发布到其他队列，那么这个计时器将被重置。 | Number |
| Queues [Classic] | Lazy mode | 惰性模式。启用此模式后，RabbitMQ会尽可能将队列中的消息存储到磁盘，以减少内存使用。 | String ("lazy") |

| | | | |
|---|---|---|---|
| Queues [Quorum] | Max in memory length | 每个消费者在确认之前允许消费的最大未确认消息数。 | Number |
| Queues [Quorum] | Max in memory bytes | 队列中允许的最大内存字节总数。当队列中的消息占用的内存字节总数达到此限制时，RabbitMQ会尝试将超出限制的消息写入磁盘，以减少内存使用。请注意，这个设置对于启用了Lazy mode的队列可能不适用，因为在惰性模式下，RabbitMQ默认会尽可能将消息存储到磁盘。 | Number |
| Queues [Quorum] | Delivery limit | 每个消费者在确认之前允许消费的最大未确认消息数。 | Number |
| Exchanges | Alternate exchange | 备用交换机。当消息无法被路由到任何队列（例如，没有匹配的路由键或队列）时，这些消息将被发送到指定的备用交换机。 | String |

5. 单击完成，完成策略创建，可以在策略列表看见已创建好的策略。

## 编辑策略

1. 在策略列表中，单击目标策略操作列的编辑。
2. 在弹窗中，对策略信息进行编辑。
3. 单击提交，完成修改。

## 删除策略

1. 在策略列表中，找到需要删除的策略，单击操作列的删除。
2. 在弹出的提示框中，单击删除，完成删除。

# 默认镜像策略

## 策略说明

为了提高消息队列 RabbitMQ 集群的可靠性和容错能力，在用户新建 RabbitMQ 集群或者新建 Vhost 时（该集群节点数至少为3），为用户提供开启“默认镜像队列”的选项。此镜像队列可以在 RabbitMQ 集群中的多个节点上复制队列中的消息，确保在某个节点发生故障时，队列中的消息不会丢失。
以下是消息队列 RabbitMQ 为用户提供的“默认镜像队列”策略参数详细说明：

| 参数名 | 配置参数 | 参数说明 |
| --- | --- | --- |
| Name | pay-mirror-policy | 策略名称，用于标识和引用该策略。 |
| Pattern | .* | 策略的匹配模式，采用正则表达式语法。点号(.)表示匹配任意字符，星号(*)表示匹配前面的字符零次或多次，所以.*表示匹配任意名称的队列。 |
| Apply to | Queues | 策略的应用对象，设置为 Queues 表示策略应用于队列。 |
| Priority | 0 | 策略的优先级。如果一个队列匹配多个策略，那么优先级高的策略将会被应用，0 表示最低优先级。 |
| ha-mode | exactly | 镜像队列的复制模式。<br>• exactly：表示队列的消息会被复制到指定数量的节点<br>• all：表示队列的消息会被复制到所有节点。<br><br>选择 exactly 可以在保证可用性的同时，减少网络和存储的开销，提高性能。 |
| ha-params | 3 | 镜像队列的复制参数。当 ha-mode 设置为 exactly 时，这里需要设置复制的节点数量。默认设置为3，即使将来扩展到5个节点，性能仍然可以保持在一个较好的水平。 |
| ha-promote-on-failure | always | 节点故障时的镜像队列提升策略。always 表示无论节点故障的原因如何，都会将镜像队列提升为主队列。when-synced 表示只有当节点故障后重新同步时，才会将镜像队列提升为主队列。默认设置为 always，以确保在任何故障情况下都能保持服务的可用性。 |
| ha-promote-on-shutdown | when-synced | 节点正常关闭时的镜像队列提升策略。<br>• always：表示无论节点关闭的原因如何，都会将镜像队列提升为主队列。<br>• when-synced：表示只有当节点关闭后重新同步时，才会将镜像队 |

| | | |
|---|---|---|
| | | 列提升为主队列。<br><br>默认设置为 when-synced，以避免不必要的提升操作。 |
| ha-sync-mode | manual | 镜像队列的同步模式。<br>• automatic：表示在节点启动或重新连接到集群时，自动将镜像队列与主队列同步。<br>• manual：表示需要手动触发同步操作，将镜像队列与主队列同步。<br><br>默认设置为manual，以避免堆积消息时自动同步影响集群性能。 |

通过配置默认的镜像队列策略，可以在保证 RabbitMQ 集群可靠性的同时，优化性能和资源利用。用户可以根据自己的需求和场景，进一步调整这些参数，也可以删除并重新新建策略。

# 操作步骤

## 开启默认镜像策略

如果在创建集群时选择了开启镜像队列，则该集群的“默认Vhost”会自动创建默认镜像策略。为Vhost配置镜像策略还有以下两种方式：

1、新建 Vhost 时开启镜像队列：集群列表 > Vhost > 新建 Vhost > 开启镜像队列。

2、在 Vhost列表 > Vhost详情页 > 策略 页签，点击新建策略，选择配置“镜像策略”，详情可以参考策略管理。

# 限制

消息队列 RabbitMQ 只允许3个或以上节点的集群开启“默认镜像队列”，主要是为了确保集群的高可用性和容错能力。在一个拥有3个或更多节点的集群中，镜像队列可以在多个节点上复制消息，这样既可以分担每个节点的负载，提高性能，又可以在某个节点发生故障时保证服务的正常运行。同时，这也提供了更多的灵活性，允许我们根据实际需求灵活配置镜像队列的参数。因此，这个限制是为了提供更稳定、更可靠的服务。

# 插件管理

插件管理模块对 TDMQ RabbitMQ 版集群支持的插件进行统一查看和管理。
本文介绍如何在 TDMQ RabbitMQ 版控制台上查看支持的插件。

## 查看插件

1. 登录 RabbitMQ 控制台。
2. 在左侧导航栏选择集群管理，选择好地域，单击需要查看的集群的“ID”，进入集群详情页。
3. 在集群详情页顶部，选择插件管理页签，进入插件管理列表。

TDMQ RabbitMQ 版集群创建后，默认提供并开启的插件：

| 插件名称 | 插件描述 |
| --- | --- |
| rabbitmq_delayed_message_exchange | rabbitmq_delayed_message_exchange插件允许您在RabbitMQ中实现延迟消息的功能。这意味着您可以将消息发送到一个特殊的交换机，并指定一个延迟时间，在延迟时间过后，消息会被发送到响应的队列。 |
| rabbitmq_management | rabbitmq_management 插件提供了一个基于Web的用户界面和一组 HTTP API，用于管理和监控 RabbitMQ 服务器。 |
| rabbitmq_peer_discovery_k8s | rabbitmq_peer_discovery_k8s 插件允许 RabbitMQ 节点在 Kubernetes 环境中自动发现并连接到其他RabbitMQ 节点，从而形成一个集群。 |

> 注意：
>
> TDMQ RabbitMQ 对 “rabbitmq_delayed_message_exchange” 延时消息插件做了默认关闭，原因为该插件存在以下风险和限制：
>
> 1. 当前插件的设计不适用于大量延迟消息（未调度的消息达数十万甚至数百万条）的场景，生产环境请谨慎评估消

息量级，避免非预期的长时间延迟、消息丢失等问题。

2. 延时消息在每个节点上只有一个持久化副本，如果节点无法正常运行（例如由于消息堆积导致持续 OOM 后重启且无法恢复），则该节点上的延时消息无法被消费端消费。
3. 延时交换机不支持设置 mandatory，生产者无法通过 basic.return 事件感知到无法路由的消息，因此发送延时消息前请务必保证对应的交换机、队列、路由关系存在。
   风险和限制原文可参见 RabbitMQ 延时消息插件的官方使用限制说明。

另外，可以参见开发指南中描述的两种延时消息的实现方式。

# 访问管理 CAM

## CAM 基本概念

主账号通过给子账号绑定策略实现授权，策略设置可精确到 [API，资源，用户/用户组，允许/拒绝，条件] 维度。

账户

- 主账号：拥有云平台所有资源，可以访问其任何资源。
- 子账号：包括子用户和协作者。
- 子用户： 由主账号创建，完全归属于创建该子用户的主账号。
- 协作者：本身拥有主账号身份，被添加作为当前主账号的协作者，则为当前主账号的子账号之一，可切换回主账号身份。
- 身份凭证：包括登录凭证和访问证书两种，登录凭证指用户登录名和密码，访问证书指云 API 密钥（SecretId 和 SecretKey）。

资源与权限

- 资源：资源是云服务中被操作的对象，在 TDMQ RabbitMQ 版中，资源有集群、Vhost、Exchange、Queue、routeRelation 等。
- 权限：权限是指允许或拒绝某些用户执行某些操作。默认情况下，主账号拥有其名下所有资源的访问权限，而子账号没有主账号下任何资源的访问权限。
- 策略：策略是定义和描述一条或多条权限的语法规范。主账号通过将策略关联到用户/用户组完成授权。
  单击查看更多 CAM 文档

## 子账号使用 RabbitMQ

为了保证子账号能够顺利使用 RabbitMQ，主账号需要对子账户进行授权。
主账号登录 访问管理控制台，在子账号列表中找到对应的子账号，单击操作列的授权。
RabbitMQ 为子账号提供了两种预设策略：QcloudTDMQReadOnlyAccess 和 QcloudTDMQFullAccess，前者仅能查看控制台的相关信息，后者可以在产品控制台进行读写等相关操作。

# 支持资源级授权的 API 列表

TDMQ RabbitMQ版支持资源级授权，您可以指定子账号拥有特定资源的接口权限。
支持资源级授权的接口列表如下：

| API名 | API描述 | 资源类型 | 资源六段式示例 |
| --- | --- | --- | --- |
| DeleteAMQPCluster | 删除 AMQP 集群 | cluster | qcs::tdmq:${region}:uin/${uin}:cluster/${clusterId} |
| ModifyAMQPCluster | 修改 AMQP 集群 | cluster | qcs::tdmq:${region}:uin/${uin}:cluster/${clusterId} |
| CreateAMQPVHost | 创建 AMQP VHost | cluster | qcs::tdmq:${region}:uin/${uin}:cluster/${clusterId} |
| DescribeAMQPClusters | 查询 AMQP 集群列表 | cluster | qcs::tdmq:${region}:uin/${uin}:cluster/${clusterId} |
| DescribeAMQPCluster | 获取单个 AMQP 集群信息 | cluster | qcs::tdmq:${region}:uin/${uin}:cluster/${clusterId} |
| CreateAMQPExchange | 创建 AMQP Exchange | vHost | qcs::tdmq:${region}:uin/${uin}:vHost/${clusterId}/${vHostId} |
| ModifyAMQPVHost | 修改 AMQP VHost | vHost | qcs::tdmq:${region}:uin/${uin}:vHost/${clusterId}/${vHostId} |

| API名 | API描述 | 资源类型 | 资源六段式示例 |
| --- | --- | --- | --- |
| DeleteAMQPVHost | 删除 AMQP VHost | vHost | qcs::tdmq:${region}:uin/${uin}:vHost/${clusterId}/${vHostId} |
| CreateAMQPQueue | 创建 AMQP 队列 | vHost | qcs::tdmq:${region}:uin/${uin}:vHost/${clusterId}/${vHostId} |
| CreateAMQPRouteRelation | 创建 AMQP 路由关系 | vHost | qcs::tdmq:${region}:uin/${uin}:vHost/${clusterId}/${vHostId} |
| DescribeAMQPVHostConnections | 查询 AMQP VHost 连接列表 | vHost | qcs::tdmq:${region}:uin/${uin}:vHost/${clusterId}/${vHostId} |
| DescribeAMQPVHosts | 查询 AMQP VHost 列表 | vHost | qcs::tdmq:${region}:uin/${uin}:vHost/${clusterId}/${vHostId} |
| DeleteAMQPExchange | 删除 AMQP Exchange | exchange | qcs::tdmq:${region}:uin/${uin}:exchange/${clusterId}/${vHostId}/${exchangeName} |
| ModifyAMQPExchange | 修改 AMQP Exchange | exchange | qcs::tdmq:${region}:uin/${uin}:exchange/${clusterId}/${vHostId}/${exchangeName} |
| DescribeAMQPExchanges | 查询 AMQP Exchange 列表 | exchange | qcs::tdmq:${region}:uin/${uin}:exchange/${clusterId}/${vHostId}/${exchangeName} |
| DeleteAMQPQueue | 删除 AMQP 队列 | queue | qcs::tdmq:${region}:uin/${uin}:queue/${clusterId}/${vHostId}/${queueName} |
| DescribeAMQPQueueConsumers | 获取指定队列下消费者列表 | queue | qcs::tdmq:${region}:uin/${uin}:queue/${clusterId}/${vHostId}/${queueName} |
| ModifyAMQPQueue | 修改 AMQP 队列 | queue | qcs::tdmq:${region}:uin/${uin}:queue/${clusterId}/${vHostId}/${queueName} |
| DescribeAMQPQueues | 查询 AMQP 队列列表 | queue | qcs::tdmq:${region}:uin/${uin}:queue/${clusterId}/${vHostId}/${queueName} |
| DescribeAMQPRouteRelations | 查询 AMQP 路由关系列表 | routeRelation | qcs::tdmq:${region}:uin/${uin}:routeRelation/${clusterId}/${vHostId}/${routeRelationId} |
| DeleteAMQPRouteRelation | 删除 AMQP 路由关系 | routeRelation | qcs::tdmq:${region}:uin/${uin}:routeRelation/${clusterId}/ |

| API名 | API描述 | 资源类型 | 资源六段式示例 |
| --- | --- | --- | --- |
| | | | ${vHostId}/${routeRelationId} |

# 不支持资源级授权的 API 列表

| API 名 | API 描述 | 资源六段式 |
| --- | --- | --- |
| CreateAMQPCluster | 创建 AMQP 集群 | * |
| DescribeAMQPCreateQuota | 获取用户配额 | * |

# 授权方案示例

## 全读写策略

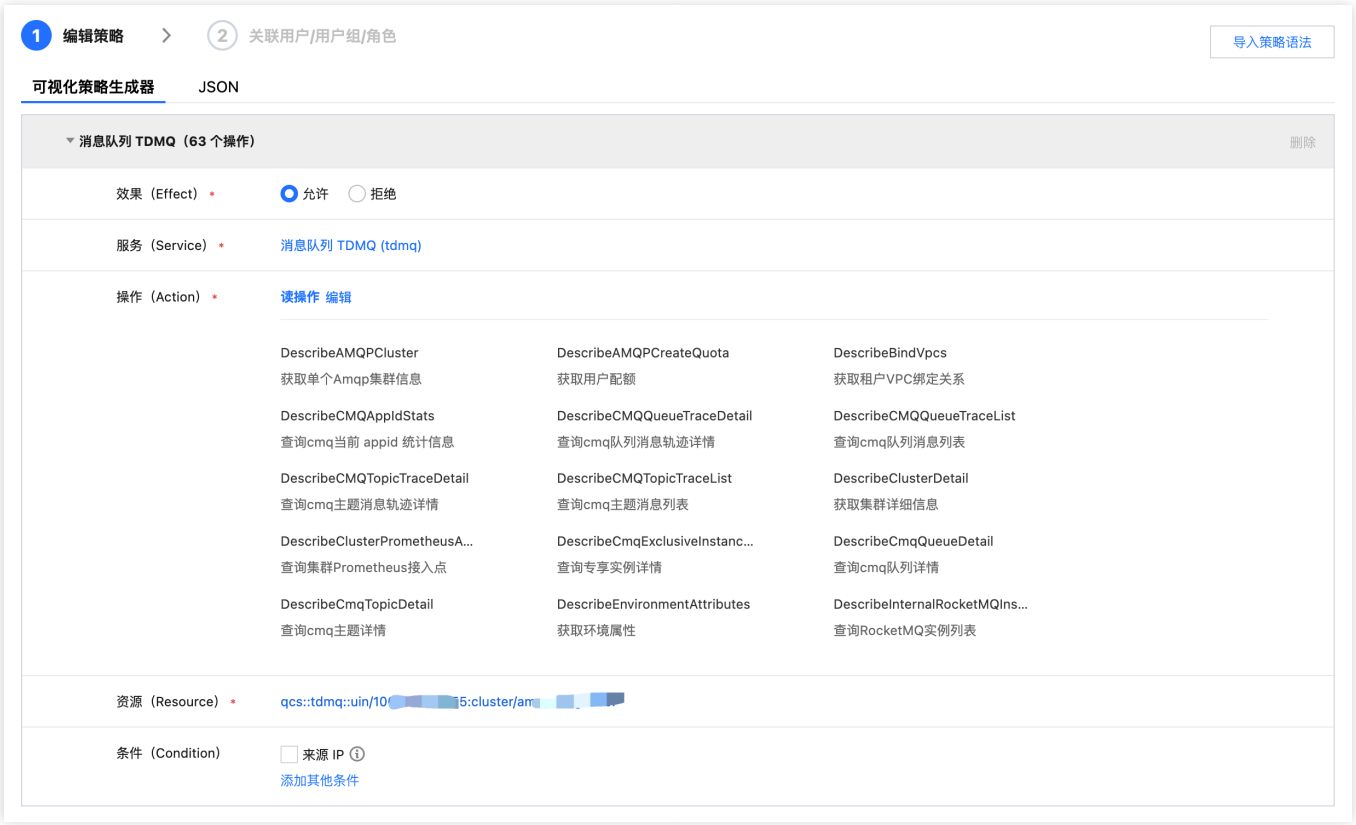
授权一个子用户以 TDMQ RabbitMQ 版服务的完全管理权限（创建、管理等全部操作）。

1. 登录 访问管理控制台。
2. 在左侧菜单栏中，单击 策略。
3. 在策略列表中，单击新建自定义策略。
4. 在选择创建策略方式的弹窗中，选择按策略生成器创建。
5. 在编辑策略页面，单击右上角导入策略语法。
6. 在导入策略语法页面，搜索“TDMQ”，在搜索结果中勾选 QcloudTDMQFullAccess，单击确定。
7. 在编辑策略页面，单击下一步，填写策略名称和描述，选择您要关联的用户/用户组。
8. 单击完成，完成策略创建与授权。

## 资源只读策略

以授权单个集群的只读权限为例。

1. 登录 访问管理控制台。
2. 在左侧菜单栏中，单击策略。
3. 在策略列表中，单击新建自定义策略。
4. 在选择创建策略方式的弹窗中，选择按策略生成器创建，填写策略信息。




| 参数 | 说明 |
| --- | --- |
| 效果（Effect） | 选择允许 |
| 服务（Service） | 选择消息队列 TDMQ (tdmq) |
| 操作（Action） | 选择读操作 |
| 资源（Resource） | 选择特定资源，点击添加自定义资源六段式<br>- 地域：选择资源所在地域<br>- 账户：系统自动填充<br>- 资源前缀：cluster<br>- 填写您要授权的集群 ID |
| 条件（Condition） | 仅当请求来自指定 IP 地址范围内时才允许访问指定操作 |

5. 单击下一步，填写策略名称和描述，选择您要关联的用户/用户组。

6. 单击完成，完成策略创建与授权。

# 上云迁移方案
# 迁移方案概述

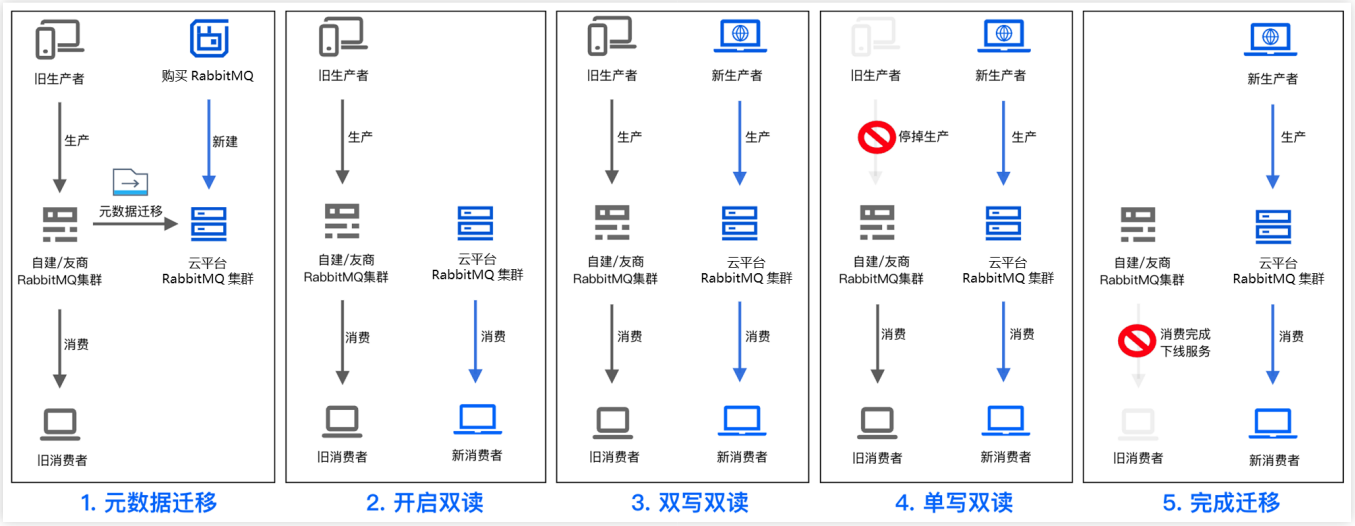
## 操作场景

本文档为您总体介绍将自建/其他云产品 RabbitMQ 集群迁移到云平台消息队列 RabbitMQ 集群的可行方案。

## 方案说明：双写双消费

该方案整体简单清晰、便于操作，且无数据积压，可以保证消息被及时消费。
步骤如下：

1. 购买云平台上的 RabbitMQ 集群。初始状态时，读写均在旧集群，新集群就绪，完成 RabbitMQ 的元数据迁移。
2. 上线新的消费端到新集群，开启双读模式。
3. 上线新的生产端，将流量逐步灰度切换到新集群，开启双写双读模式。
4. 停掉旧集群生产流量，暂时保留旧集群消费端，进入到单写双读模式。
5. 等待旧集群上的数据消费完后，确认没有积压消息后再下线旧的消费服务，完成迁移过程。



# 步骤1：购买云上实例

1. 登录 RabbitMQ 控制台。
2. 在左侧导航栏选择 rabbitmq 下的集群管理，单击新建集群，进入购买页面。
3. 在购买页面，选择创建的实例规格。
4. 选中我已阅读并同意消息队列 RabbitMQ 版服务条款，然后创建集群。
5. 等待3-5分钟即可在集群管理列表页面看到创建好的集群。

# 步骤2：迁移元数据上云

## 操作场景

该任务指导您将开源 RabbitMQ 的元数据迁移至消息队列 RabbitMQ 版。

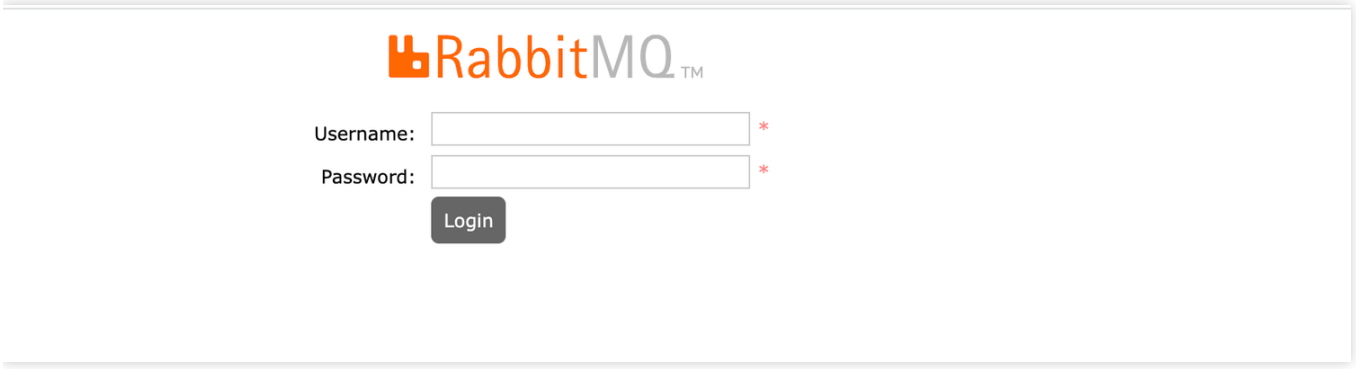
## 前提条件

向消息队列 RabbitMQ 版导入元数据文件时，您需要先从开源 RabbitMQ 导出元数据文件。

## 操作步骤

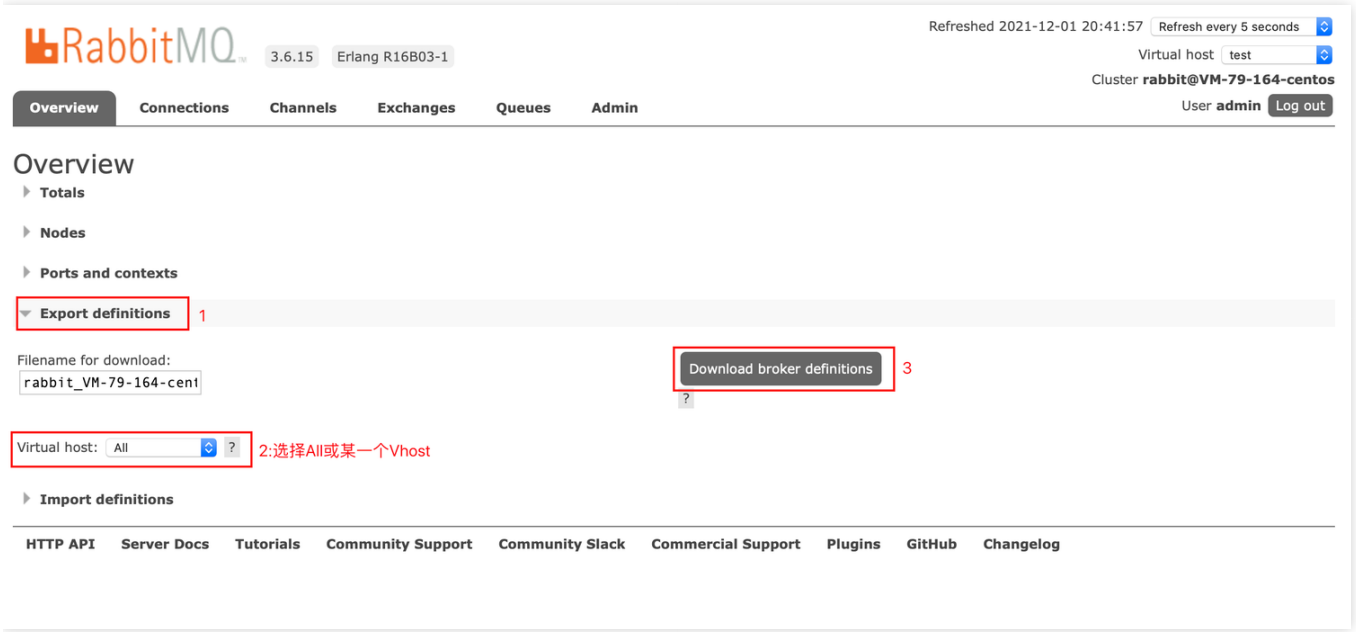
### 从自建RabbitMQ集群导出元数据

1. 打开浏览器登录到自建开源 RabbitMQ 控制台。




enter image description here

2. 在 Overview 页面下方，单击 Export definitions。在输入框中填写目标文件名后，在“Virtual host”处选择“All”或某个 Vhost，单击右侧的 Download broker definitions，导出全部 Vhosts 或某个 Vhost 的元数据文件。




enter image description here

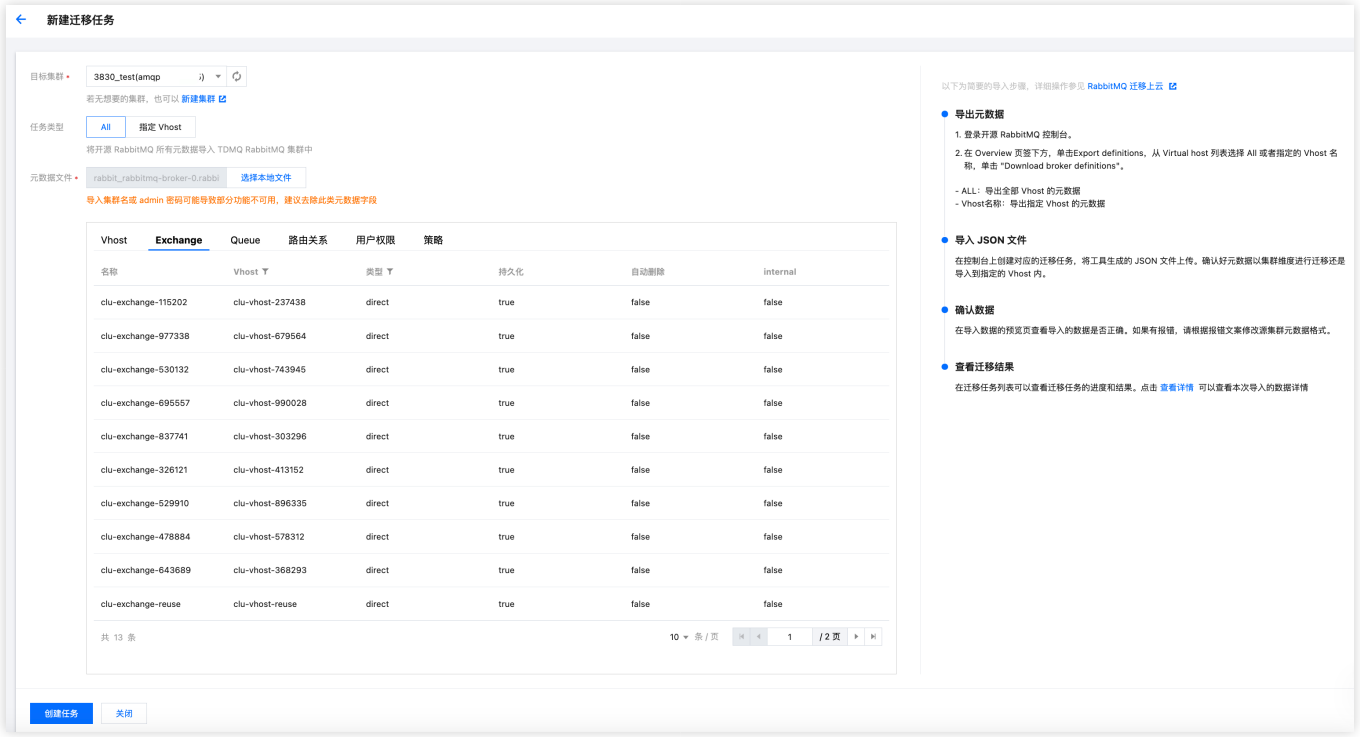
3. 查看导出的元数据的内容

enter image description here

将元数据导入到RabbitMQ集群

1. 打开控制台，单击迁移上云。

2. 在迁移上云任务列表页，单击新建任务。



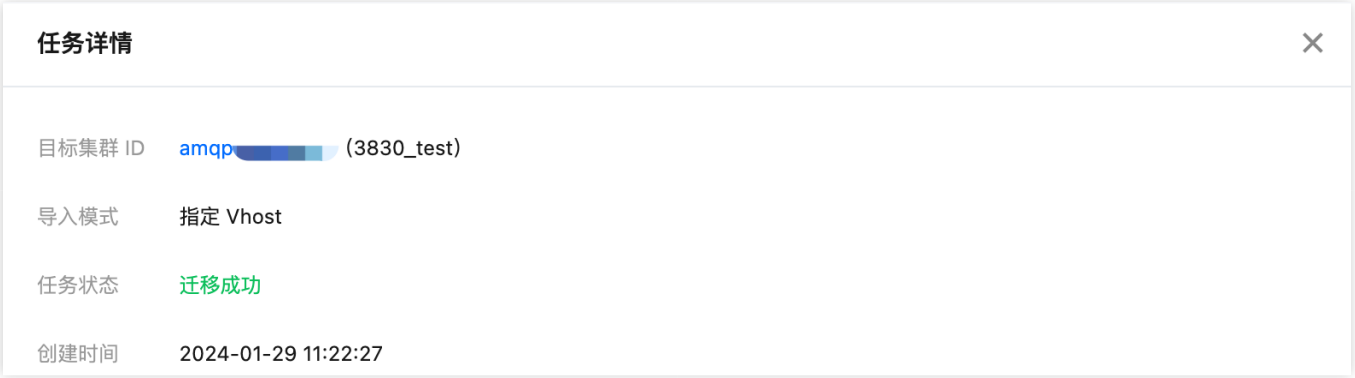

- 目标集群：选择元数据导入的目标 TDMQ RabbitMQ 版集群。
- 任务类型
  All：将开源 RabbitMQ 集群所有元数据导入 TDMQ RabbitMQ 版集群中；
  指定Vhost：将开源 RabbitMQ 集群指定 Vhost 元数据导入 TDMQ RabbitMQ 版集群指定的 Vhost 中。
- 元数据文件：选择本地的元数据文件。

  > 说明：
  >
  > 导入集群名称或 admin 密码可能导致部分功能不可用，建议去除此类元数据字段。

3. 在导入数据的预览页查看导入的数据是否正确。如果有报错，请根据报错文案修改源集群元数据格式。

4. 单击创建任务。在迁移上云任务列表页会生成一条任务记录。

5. 单击查看详情可以查看本次迁移任务的详情。


任务详情
目标集群 ID
amqp
(3830_test)
导入模式
指定 Vhost
任务状态
迁移成功
创建时间
2024-01-29 11:22:27

# 步骤3：开启双读写

## 操作场景

本文主要介绍使用双写双消费方案，将自建 RabbitMQ 集群的服务切换到云平台消息队列 RabbitMQ 中的方法。

## 方案：双写双消费模式

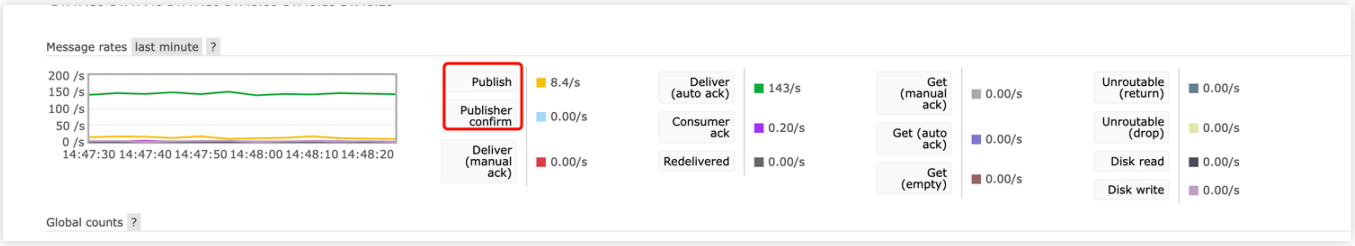
### 前提条件

1. 已购买RabbitMQ 实例
2. 已将自建 RabbitMQ 集群元数据迁移到 RabbitMQ。

### 操作步骤

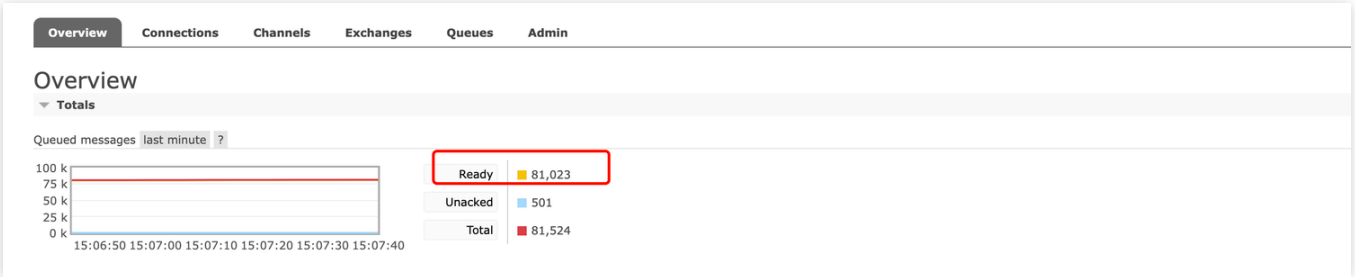
1. 切换消费者集群中部分节点的接入信息，将这部分消费者接入到新版 RabbitMQ 集群。切换的这部分消费者将消费新版 RabbitMQ 集群中的消息，剩余消费者继续消费原 RabbitMQ 集群中的消息。
2. 切换生产者集群中部分节点的接入信息，将这部分生产者接入到新版 RabbitMQ。切换的这部分生产者将发送消息到新版 RabbitMQ 集群中；剩余的生产者还是将消息发送到旧版 RabbitMQ 集群中。为了防止消息的重复或丢失，可以事先做好消息消费的幂等逻辑。
3. 将剩余的生产者全部接入到新的 RabbitMQ 集群上。此时所有消息将全部被发送到新版 RabbitMQ 集群中。

- **Tips 1：**可以在自建集群的 RabbitMQ 社区管控台，确认自建集群的生产流量已经停止。




enter image description here

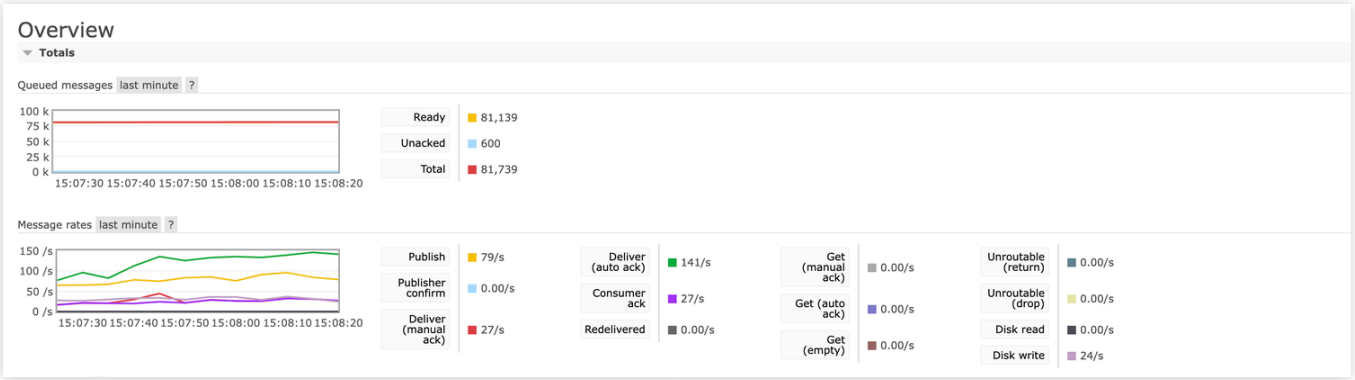
- **Tips 2：**在自建集群的 RabbitMQ 社区管控台上，可以确认自建 RabbitMQ 集群的堆积消息在下降。




enter image description here

4. 检查原 RabbitMQ 集群是否有堆积的消息没有消费，确认原 RabbitMQ 集群没有堆积或者未处理的消息后，将剩余的消费者全部接入到新版 RabbitMQ 集群上。完成整个数据流的迁移操作。

- **Tips：**确认RabbitMQ 集群的消息写入和消费，并确保消息无堆积。




enter image description here

> 注意：
>
> - 如您没有按照上述顺序进行切换，如先切换生产者再切换消费者，可能导致消息丢失的情况。
> - 切换剩余消费者之前，请确保原 RabbitMQ 集群中的消息已全部消费完，否则可能会导致消费遗漏。

# 可能存在的问题

## 顺序问题

由于集群切换，切换过程中，消息顺序问题无法保证；切换的过程中会导致局部乱序。

## 消息重复

理论上不会重复，极端情况下会发生，例如切换过程中，消费者消费了消息，但是还未给服务端（原 RabbitMQ 集群）ACK，这时会导致该消息进入重试队列导致重复消费。对消息做幂等处理逻辑可以规避这个问题。

## 消费延迟

读写切换过程中，由于分区重新分配，需要进行队列和消费者客户端间的 rebalance，可能导致短暂的消费延迟现象。遇到该情况无需额外操作，切换完成后即可恢复。

# 开发指南
# 延时消息

本文主要介绍消息队列 TDMQ RabbitMQ 版中延迟消息的概念、使用场景和使用方式。

## 名词解释

延时消息：消息在发送至服务端后，实际业务并不希望消费端马上收到这条消息，而是推迟一段时间后再被消费，这类消息统称为延时消息。

## 使用场景

- 场景 1：对于消息的生产消费时间有要求的场景。例如在电商系统中，若用户下单后 30 分钟不支付，自动取消订单。
- 场景 2：通过消息触发延时任务的场景。用户登录 App 浏览特定商品 20 分钟后还没下单，自动推送商品评测信息的消息并发放商品相关优惠券。

## 实现方式

### 方式一：通过设置消息的过期时间和死信队列实现延时消息

原理概述

- 发送端通过设置消息过期时间，触发过期未消费的消息被投递到死信交换机下的死信队列。
- 消费端通过消费死信队列中的消息，实现延时消息的消费。

使用示例

代码仅示例使用，交换机类型、routingKey 等参数请结合实际场景替换为业务预期的值。

- 发送端代码示例如下，消费端订阅死信队列即可。

```
// 声明死信交换机
channel.exchangeDeclare("${dlxExchangeName}", "direct", true);
// 声明用于发送延时消息的交换机
channel.exchangeDeclare("${delayExchangeName}", "direct", true);
// 声明用于发送延时消息的队列，并指定其死信交换机
Map<String, Object> args = new HashMap<>();
args.put("x-dead-letter-exchange", "${dlxExchangeName}");
```

```
channel.queueDeclare("${delayQueueName}", true, false, false, args);
channel.queueBind("${delayQueueName}", "${delayExchangeName}", "");

// 声明死信队列
channel.queueDeclare("${delayQueueName}", true, false, false, null);
channel.queueBind("${dlxQueueName}", "${dlxExchangeName}", "");

// 发送延时消息
int delayInSeconds = 10; // 消息延时 10s
AMQP.BasicProperties props = new AMQP.BasicProperties.Builder().expiration(String.valueOf(dela
yInSeconds * 1000));
channel.basicPublish("${delayExchangeName}", "", props.build(), "delayed payload".getBytes());
```

- 参数说明如下

| 参数 | 说明 |
| --- | --- |
| ${dlxExchangeName} | 用于延迟消息的死信交换机名称，请替换为可在控制台 Exchange 列表查询到的名称。 |
| ${delayExchangeName} | 实际发送延时消息的交换机名称，请替换为可在控制台 Exchange 列表查询到的名称。 |
| x-dead-letter-exchange | 队列参数 key，用于设置其对应的死信交换机。 |
| ${dlxQueueName} | 用于延迟消息的死信队列名称 |
| ${delayQueueName} | 实际发送延时消息的队列名称 |

## 方式二：通过内置的 rabbitmq_delayed_message_exchange 插件实现延时消息

使用限制

> 说明：
>
> 更多详细信息请参考 RabbitMQ 官方插件使用限制说明。

- 当前插件的设计不适用于大量延迟消息（未调度的消息达数十万甚至数百万条）的场景，生产环境请谨慎评估消息量级，避免非预期的长时间延迟、消息丢失等问题。
- 延时消息在每个节点上只有一个持久化副本，如果节点无法正常运行（例如由于消息堆积导致持续 OOM 后重启且无法恢复），则该节点上的延时消息无法被消费端消费。
- 延时交换机不支持设置 mandatory ，生产者无法通过 basic.return 事件感知到无法路由的消息，因此发送延时消息前请务必保证对应的交换机、队列、路由关系存在。

使用示例

TDMQ RabbitMQ 版延时消息的使用方式和 RabbitMQ 官方支持的延时插件的使用方式完全一致，方便业务进行无改造迁移。

1. 声明 Exchange 并指定 Exchange 的路由类型。

```
// 声明延时交换机
Map<String, Object> args = new HashMap<String, Object>();
args.put("x-delayed-type", "direct");
channel.exchangeDeclare("${delayedExchangeName}", "x-delayed-message", true, false, args);
```

   - 参数说明如下：

| 参数 | 说明 |
| --- | --- |
| x-delayed-type | Exchange 的类型，指定路由规则。取值说明如下：<br>• direct<br>• fanout<br>• topic<br><br>具体说明请参见 Exchange 管理。 |
| ${delayedExchangeName} | Exchange 的名称，请替换为可在控制台 Exchange 列表查询到的名称。 |
| x-delayed-message | 指定 Exchange 类型，以支持投递延时消息。 |

2. 发送延时消息。在消息的 Header 属性中增加一个键为 x-delay，值为毫秒数的键值对，并且指定发送的目标 Exchange 为上一步已声明的 Exchange。

```
byte[] messageBodyBytes = "delayed payload".getBytes("UTF-8");
Map<String, Object> headers = new HashMap<String, Object>();
headers.put("x-delay", 4000); // 消息延时 4s
AMQP.BasicProperties.Builder props = new AMQP.BasicProperties.Builder().headers(headers);
channel.basicPublish("${delayedExchangeName}", "", props.build(), messageBodyBytes);
```

   当消息到达 Exchange 后，会在4000毫秒后投递到对应的 Queue。

# 最佳实践
# RabbitMQ 客户端实践教程

## 请不要每次发送消息都创建连接/通道

RabbitMQ 中创建连接是一个耗时/耗资源的操作，每个连接会使用至少 100KB 的内存，连接数过多会增大 Broker 的内存压力。应该在程序启动时创建连接，每次发送消息时复用此长连接，提升发送性能和减少服务端内存占用。
通道是一种更加轻量的通信方式，建议尽可能多的使用通道来复用连接。但最好不要跨线程并发使用同一个通道，因为很多 RabbitMQ 客户端的通道实现并不是线程安全的。

## 为生产者设置合理的发送超时时间

RabbitMQ 不同语言和版本的客户端设置了不同的默认发送时间，部分客户端默认超时时间过长，例如 580 秒或 900 秒。网络异常时，过长的发送超时时间会阻塞发送端线程，甚至进一步造成雪崩效应，建议根据业务场景设置合理的超时时间，3 秒是一个推荐的值。

## 生产者和消费者使用独立的连接

由于 RabbitMQ 有独特的流控机制，如果生产者和消费者复用同一个物理连接，而消费流量大触发了流控，可能会导致生产者被流控导致发送慢或超时，因此建议生产者和消费者初始化时使用不同的物理连接，避免互相影响。

## 消费者不建议开启自动消息确认

RabbitMQ 服务端提供了至少投递一次的消费语义，以确保消息正确地触达到下游业务系统。一旦消费端开启了自动确认消息，服务端将消息推送给消费方后，就自动确认删除消息，即使消费端处理消息时出现了异常也不会重试，可能会导致业务上漏处理消息。

## 消费者幂等处理消息

RabbitMQ 服务端提供了至少投递一次的消费语义，极端场景下有可能重复投递消息，因此建议关键业务处理消息时一定要做幂等处理，即使收到重复消息，也不会产生负面业务影响。
业务幂等处理可以通过在消息中加入唯一业务标识，消费端消费时检查此类标识和消息状态等，根据业务需求处理重复消息，保证即使重复接受消息也不会产生业务上的负面影响。

# 限制队列长度，避免大量堆积消息

过长的队列（大量消息堆积）会占用大量内存，消耗更多服务端系统资源，不仅在运行时需要更长的时间来进行状态同步，并且会导致服务端 Broker 启动恢复时间大大增加。
较短的队列会提供更快的处理速度和系统性能。
因此需要客户端尽可能提高消费能力、队列维度限制 max-length 等手段来保证队列尽可能短。

# 使用 Consume 还是 Get 消费消息？

Get 是一种基于轮询的拉模式消费方式，每消费一条消息都需要向 Broker 发送一个请求，如果队列中没有消息，可能会导致大量无效空拉导致资源占用。而 Consume 一次可以接收一批消息，并且由服务端根据实际情况推送消息，在绝大多数情况下都应该使用 Consume 而不是 Get 来消费消息。
如果业务处理必须使用 Get 消费消息，应关注业务层面的 Get 机制，避免持续 Get Empty 空拉（队列中已经无待消费的消息，但消费端一直持续 Get）导致的服务端 CPU 负载异常高。

# 消费者设置合理的 Prefetch Count

Prefetch 设置是消费端为了提升消费吞吐，预先将消息推送到消费端缓存，降低消费等待和延迟的机制。但如果 Prefetch Count 如果过大或不限制，会导致大量消息缓存在消费端，并且服务端 Broker 也在内存维护 unack 消息的状态，占用大量资源；如果消息持续 unack 状态中，这些消息也无法被其他空闲的消费者消费，表现为消费延迟增加或消费者负载不均衡。
建议根据业务消费速率，将 Prefetch Count 设置在合理范围。

# 消费者设置合理的异常处理策略

消费端消费消息时，遇到了无法处理的异常，未设置自动确认的消息会触发消息重试。如果异常持续无法正常退出，会触发消息被无限重试，不仅会对 Broker 端造成较高的负载压力，也会导致后续的消息无法被合理消费。

# 客户端重连机制确认

服务端 Broker 在例如 OOM、容器母机故障等极端场景下会自愈重启，日常业务自行操作集群升配等场景也会触发 Broker 重启，为避免 Broker 重启期间客户端持续连接异常，请确保客户端实现了自动重连机制。

# 客户端 SDK 请不要关闭 heartbeat 设置

heartbeat 在服务端和客户端分别有一个配置值（服务端为 60 秒），最终生效的 heartbeat 由服务端和客户端协商决定，且不同语言/版本的客户端协商机制不同。客户端设置 heartbeat=0 即关闭心跳检测，会导致服务端无法自动剔除长期无数据交互的无用连接，最终可能引起非预期的连接泄露。

# RabbitMQ 消息可靠性实践教程

## 消息持久化

为了确保队列元数据和队列中的消息在 Broker 重启后不丢失，建议将队列设置为 durable、消息设置为 persistent，这样队列在接收到消息后会立即将其持久化到磁盘上。
非持久化的消息也会占用更多的服务端内存资源，极端情况下引起服务端内存负载过高。

## 发送端 Confirm

Confirm 机制可以确保消息被成功地发送到 Broker。但如果发送消息时不设置 mandatory，无论消息是否成功路由到目标队列，Broker 都会回应 confirm 给发送端。如果设置了 mandatory（延迟交换机不支持设置 mandatory），消息无法路由时 Broker 会将消息原路返回客户端，客户端可以通过实现 basic.return 的处理来感知这些无法路由的消息；消息可以成功路由到目标队列时，Broker 才会回应 confirm 给发送端。

## 消费端 Acknowledgement

消费端 ACK 机制可以确保客户端收到消息，提供了 at least once 级别的消费语义保证，确保消息被正确处理后才能被删除。但这也需要客户端做好幂等，避免重复消费消息引发错误，而且未被 ACK 的消息会堆积在内存中导致客户端和服务端内存占用增加。

## 开启镜像队列

镜像队列通过将队列数据复制到集群内其他 Broker 上来保证队列的高可用。配置镜像队列策略可能会增加 Broker 的启动时长和资源占用，但可以确保队列在单个 Broker 故障时仍处于可用状态，尽量确保不丢失消息。
配置镜像队列策略时，应避免设置 ha-sync-mode=automatic，该配置会引起服务端 Broker 重启后自动全量同步队列数据（无论队列数据是否被同步过）。如果同步的队列堆积数据过多，最终会导致 Broker 同步数据时间过长、持续占用内存资源等问题，且队列同步完成前队列处于不可用状态，对业务可用性、服务端稳定性都有严重影响。

# 其他注意事项

## rabbitmq_delayed_message_exchange 插件实现的延迟消息

1. 当前插件的设计不适用于大量延迟消息（未调度的消息达数十万甚至数百万条）的场景，生产环境请谨慎评估消息量级，避免非预期的长时间延迟、消息丢失等问题。
2. 延时消息在每个节点上只有一个持久化副本，如果节点无法正常运行（例如由于消息堆积导致持续 OOM 后重启且无法恢复），则该节点上的延时消息无法被消费端消费。
3. 延时交换机不支持设置 mandatory ，生产者无法通过 basic.return 事件感知到无法路由的消息，因此发送延时消息前请务必保证对应的交换机、队列、路由关系存在。
   综上所述，我们强烈建议您不要使用这一插件，转而使用死信队列来间接实现延时消息。如果您在了解了这个插件的若干缺陷后仍然要使用，那我们强烈推荐您将延迟消息数量保持在一个尽可能低的水平，避免触发内存负载高的问题。

## 网络分区

1. 网络分区是在使用 RabbitMQ 时不得不面对的一个问题。网络分区会带来集群状态的不一致，即使是网络恢复后，RabbitMQ 仍需通过重启 Broker 的方式来重新同步状态。RabbitMQ 目前使用的 autoheal 模式，它会自动决出一个获胜的分区，然后重启非信任分区内的 Broker。
2. 建议客户端做好以下措施，尽可能减小网络分区带来的负面影响：
   - 消息发送方，考虑发送消息时配合使用 mandatory 机制并有处理 basic.return 的能力，以便及时处理网络分区时可能出现的消息路由失败。
   - 消息消费方，网络分区出现/处理过程中，可能会伴有消息重复，消费端需要做好幂等处理。

## 告警配置

云平台提供了集群/节点等多维度的监控指标，详情请见监控告警。
强烈建议重点关注节点 CPU、内存、磁盘利用率、堆积消息数量等几个指标并配置告警，避免服务端持续负载过高影响集群稳定性。

# 常见问题
# 常见问题

## 什么是消息队列 TDMQ RabbitMQ 版?

消息队列 TDMQ RabbitMQ 版（TDMQ for RabbitMQ，简称 TDMQ RabbitMQ 版）是一款自主研发的消息队列服务，支持 AMQP 0-9-1 协议，完全兼容开源 RabbitMQ 的各个组件与概念，同时具备计算存储分离，灵活扩缩容的底层优势。
TDMQ RabbitMQ 版拥有极为灵活的路由来适应各类业务的消息投递规则，能有缓冲上游流量压力的能力，保证消息系统的稳定运行。常用于系统间的异步通信和服务解耦，减轻不同服务之间的依赖，广泛应用于金融、政务等行业的分布式系统中。

## 如何使用 TDMQ RabbitMQ 版？

您可以参见 快速入门 和 操作指南，快速上手并使用 TDMQ RabbitMQ 版。

## TDMQ RabbitMQ 版有哪些应用场景？

详细说明请参见 应用场景。

# 通用参考
# RabbitMQ 客户端接口说明

RabbitMQ 客户端支持的接口列表如下：

| 接口 | 说明 |
| --- | --- |
| exchangeDeclare | 声明一个 Exchange，没有则创建。 |
| exchangeDeclareNoWait | 声明一个 Exchange，没有则异步创建。 |
| exchangeDeclarePassive | 声明一个 Exchange，没有则报异常。 |
| exchangeDelete | 删除 Exchange。 |
| exchangeUnbindNoWait | 异步取消绑定。 |
| queueDeclare | 声明一个 Queue，没有则创建。 |
| queueDeclareNoWait | 声明一个 Queue，没有则异步创建。 |
| queueDeclarePassive | 声明一个 Queue，没有则报异常。 |
| queueDelete | 删除 Queue。 |
| queueBind | 声明 Queue 和某一 Exchange 的绑定关系，没有则绑定。 |
| queueBindNoWait | 声明 Queue 和某一 Exchange 的绑定关系，没有则异步绑定。 |
| queuePurge | 重置消费位点，从最新消费（原生 RabbitMQ 相应为删除消息）。 |
| queueUnbind | 取消绑定。 |
| queueUnbindNoWait | 异步取消绑定。 |

# SDK文档
# Spring Boot Starter 接入

## 操作场景

本文以调用 Spring Boot Starter SDK 为例介绍通过开源 SDK 实现消息收发的操作过程，帮助您更好地理解消息收发的完整过程。

## 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：添加依赖

在 pom.xml 中添加依赖。

```
<!--rabbitmq-->
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-amqp</artifactId>
</dependency>
```

### 步骤2：准备配置

1. 在配置文件中加入 RabbitMQ 配置信息（以 yaml 配置为例）。

```
spring:
  rabbitmq:
     # host地址可在控制台中获取 或 使用Rabbitmq服务地址
    host: amqp-xx.rabbitmq.x.xxxxtdmq.com
    port: 5672
    # 要使用的角色名称 可在角色管理控制台获取
    username: admin
    # 角色密钥
    password: eyJrZXlJZ....
    # Vhost全称，可在集群控制台Vhost tab页获取
    virtual-host: amqp-xxx|Vhost
```

| 参数 | 说明 |
| --- | --- |
| host | 集群接入地址，在集群基本信息页面的客户端接入模块获取。 |

| 参数 | 说明 |
| --- | --- |
| | |
| port | 集群接入地址端口，在集群基本信息页面的客户端接入模块获取。 |
| username | 用户名称，填写在控制台创建的用户名称。 |
| password | 用户密码，填写在控制台创建用户时填写的密码。 |
| virtual-host | Vhost 名称，在控制台 Vhost 列表获取。 |

2. 创建配置文件加载程序（以 Fanout交换机 为例）。

> 说明
>
> 其他类型的交换机配置可参见具体 Demo 示例。

```
/**
 * Fanout交换机配置
 */
@Configuration
public class FanoutRabbitConfig {

    /**
     * 交换机
     */
    @Bean
    public FanoutExchange fanoutExchange() {
        return new FanoutExchange("fanout-logs", true, false);
    }


    /**
     * 消息队列
     */
    @Bean
    public Queue fanoutQueueA() {
        return new Queue("ps_queue", true);
    }

    @Bean
    public Queue fanoutQueueB() {
```

```
        // 可通过这种方式绑定死信队列
        Map<String, Object> requestParam = new HashMap<>();
        requestParam.put("x-dead-letter-exchange", "my-deadLetter-exchange");
        // 设置消息过期时长
        requestParam.put("x-message-ttl", 60000);
        return new Queue("ps_queue1", true, false,true, requestParam);
    }

    /**
     * 绑定消息队列与交换机
     */
    @Bean
    public Binding bindingFanoutA() {
        return BindingBuilder.bind(fanoutQueueA())
                .to(fanoutExchange());
    }

    @Bean
    public Binding bindingFanoutB() {
        return BindingBuilder.bind(fanoutQueueB())
                .to(fanoutExchange());
    }
}
```

| 参数 | 说明 |
| --- | --- |
| fanout-logs | 绑定的 Exchange 名称，在控制台 Exchange 列表获取。 |
| ps_queue | 与 Exchange 绑定的第一个 Queue名称，在控制台 Queue 列表获取。 |
| my-deadLetter-exchange | 死信 Exchange 名称，在控制台 Exchange 列表获取。 |
| ps_queue1 | 与 Exchange 绑定的第二个 Queue 名称，在控制台 Queue 列表获取。 |

### 步骤3：发送消息

创建并编译消息发送程序 DemoApplication.java，使用 RabbitTemplate 发送即可。

```
@Autowired
private RabbitTemplate rabbitTemplate;

public String send() {
    String msg = "This is a new message.";
    // 发送消息
    // 参数说明：参数1：交换机名称，在控制台 Exchange 列表获取。  参数2：routing key  参数3：消息内容
    rabbitTemplate.convertAndSend("direct_logs", "", msg);
    return "success";
}
```

### 步骤4：消费消息

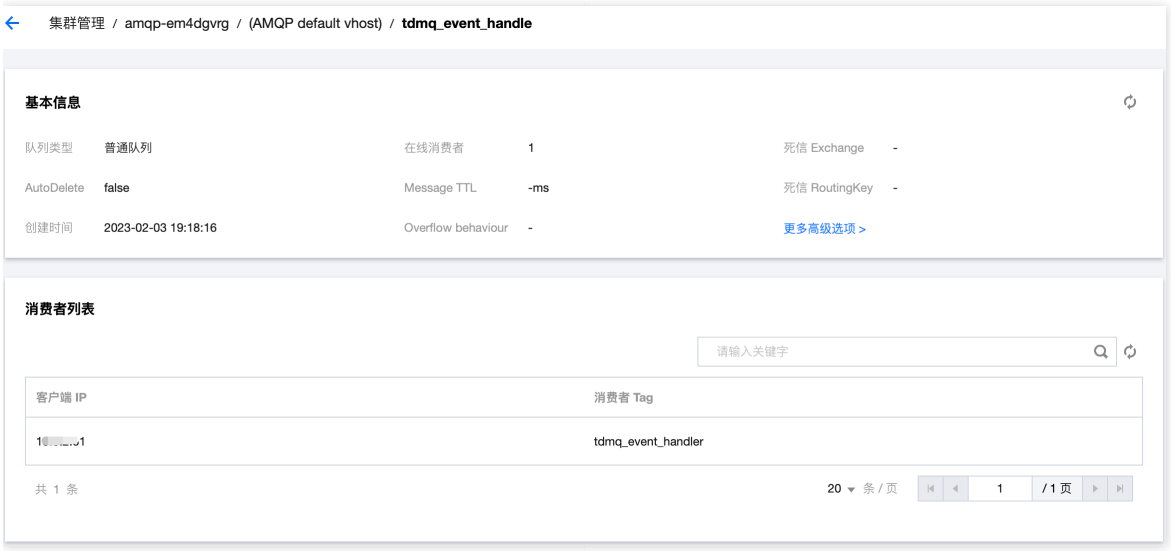
创建并编译消息接收程序 FanoutReceiver.java。（以 Fanout交换机 为例。）

```
@Component
public class FanoutReceiver {
 // 注册一个listener监听指定消息队列
 @RabbitHandler
```

```
    @RabbitListener(queues = "ps_queue")  //与 Exchange 绑定的 Queue名称，在控制台 Queue 列表获取。
    public void listenerPsQueue(String msg) {
        // 业务处理...
        System.out.println("(ps_queue) receiver message. [" + msg + "]");
    }
}
```

## 步骤5：查看消息

如果您想确认消息是否成功发送至 TDMQ RabbitMQ 版，可以在控制台 集群管理 > Queue 的基本信息页面查看接入的消费者情况。




> 说明
>
> 其他使用示例请参见 Spring AMQP 官网。

# Spring Cloud Stream 接入

## 操作场景

本文以调用 Spring Cloud Stream SDK 为例介绍通过开源 SDK 实现消息收发的操作过程，帮助您更好地理解消息收发的完整过程。

## 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：添加依赖

在 pom.xml 中添加 Stream RabbitMQ 相关依赖。

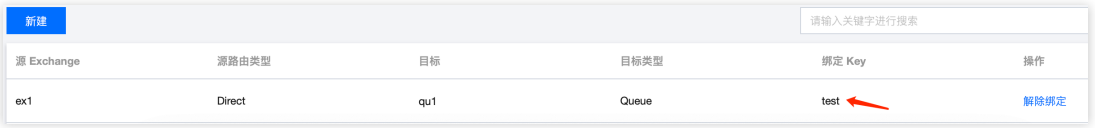
```
<dependency>
  <groupId>org.springframework.cloud</groupId>
  <artifactId>spring-cloud-starter-stream-rabbit</artifactId>
</dependency>
```

### 步骤2：准备配置

1. 在配置文件中进行相应配置（以 direct 交换机配置为例）。

```
spring:
  application:
  name: application-name
  cloud:
  stream:
    rabbit:
      bindings:
        # 输出channel名称
        output:
          # 生产者配置信息
          producer:
            # 生产者使用的交换机类型  如果已存在交换机名称，该类型必须与交换机类型一致
            exchangeType: direct
            # 用于指定 routing key 表达式
            routing-key-expression: headers["routeTo"] # 该值表示使用头信息的routeTo字段作为 routing key
            queueNameGroupOnly: true
        # 输入channel名称
        input:
          # 消费者配置信息
          consumer:
            # 消费者使用的交换机类型  如果已存在交换机名称，该类型必须与交换机类型一致
            exchangeType: direct
            # 消费者消息队列绑定的 routing key
            bindingRoutingKey: info,waring,error
            # 该配置会对上面的 routing key 进行处理
            bindingRoutingKeyDelimiter: ","  # 该配置表示：使用,切割上面配置的routing key
```

```
        # 消息确认模式  具体查看AcknowledgeMode
        acknowledge-mode: manual
        queueNameGroupOnly: true
    bindings:
      # 输出channel名称
     output: #通道的名称
      destination: direct_logs #要使用的exchange名称
      content-type: application/json
      default-binder: dev-rabbit
     # 输入channel名称
     input: #通道的名称
      destination: direct_logs #要使用的exchange名称
      content-type: application/json
      default-binder: dev-rabbit
      group: route_queue1 # 要使用的消息队列名称
    binders:
     dev-rabbit:
      type: rabbit
      environment:
        spring:
          rabbitmq:
            host: amqp-xxx.rabbitmq.xxx.xxxxtdmq.com #集群接入地址，在集群管理页面操作列的获取接入地址获取。
            port: 5672
            username: admin #角色名称
            password: password #角色密钥
            virtual-host: vhostnanme #Vhost名称
```

| 参数 | 说明 |
| --- | --- |
| bindingRoutingKey | 消费者消息队列绑定的 routing key，消息的路由规则，在控制台绑定关系列表的绑定 Key列获取。<br>  |
| direct_log | Exchange 名称，在控制台 Exchange 列表获取。 |
| route_queue1 | Queue名称，在控制台 Queue 列表获取。 |
| host | 集群接入地址，集群接入地址，在集群基本信息页面的客户端接入模块获取。<br> |
| port | 集群接入地址端口，在集群管理页面操作列的获取接入地址获取。 |
| username | 用户名称，填写在控制台创建的用户名称。 |
| password | 用户密码，填写在控制台创建用户时填写的密码。 |

| 参数 | 说明 |
| --- | --- |
| virtual-host | Vhost 名称，在控制台 Vhost 列表获取。 |

2. 创建配置文件加载程序。

- OutputMessageBinding.java

```
public interface OutputMessageBinding {
  /**
    * 要使用的通道名称(输出channel名称)
    */
  String OUTPUT = "output";

  @Output(OUTPUT)
  MessageChannel output();
}
```

- InputMessageBinding.java

```
public interface InputMessageBinding {

  /**
    * 要使用的通道名称
    */
  String INPUT = "input";

  @Input(INPUT)
  SubscribableChannel input();
}
```

### 步骤3：发送消息

创建并编译消息发送程序 IMessageSendProvider.java。

```
// 引入配置类
@EnableBinding(OutputMessageBinding.class)
public class MessageSendProvider {

    @Autowired
    private OutputMessageBinding outputMessageBinding;

    public String sendToDirect() {
        outputMessageBinding.output().send(MessageBuilder.withPayload("[info] This is a new message.[" + System.currentTimeMillis() +
"]").setHeader("routeTo", "info").build());
        outputMessageBinding.output().send(MessageBuilder.withPayload("[waring] This is a new waring message.[" + System.currentTime
Millis() + "]").setHeader("routeTo", "waring").build());
        outputMessageBinding.output().send(MessageBuilder.withPayload("[error] This is a new error message.[" + System.currentTimeMilli
s() + "]").setHeader("routeTo", "error").build());
        return "success";
    }

    public String sendToFanout() {
        for (int i = 0; i < 3; i++) {
            outputMessageBinding.output().send(MessageBuilder.withPayload("This is a new message" + i).build());
        }
        return "success";
    }
```

```
}
```

在要发送消息的类中注入 MessageSendProvider 即可进行发送消息。

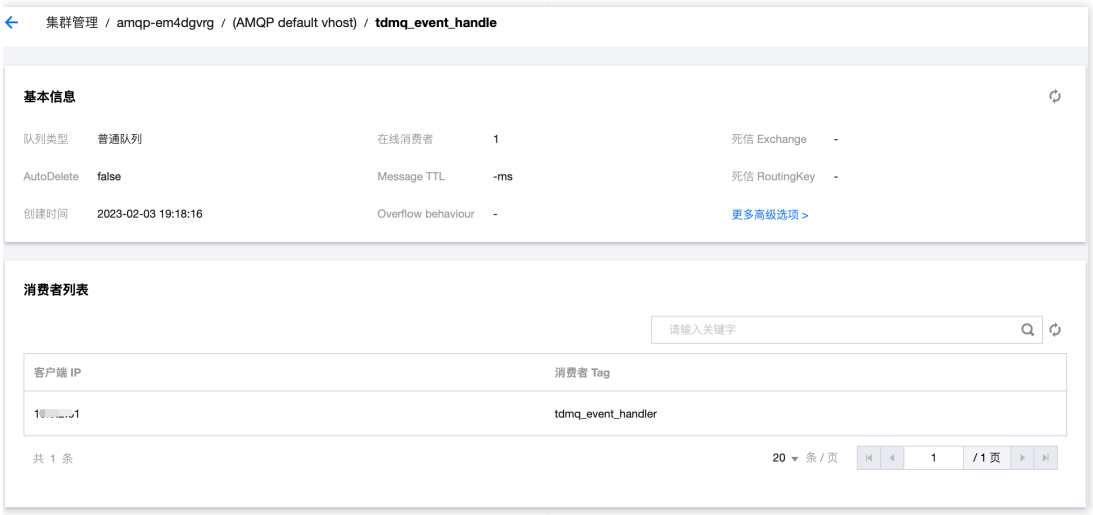
### 步骤4：消费消息

创建并编译消息消费程序 MessageConsumer.java。可配置多个通道，可对不同消息队列的监听。

```
@Service
@EnableBinding(InputMessageBinding.class)
public class MessageConsumer {

    @StreamListener(InputMessageBinding.INPUT)
    public void test(Message<String> message) throws IOException {
        Channel channel = (com.rabbitmq.client.Channel) message.getHeaders().get(AmqpHeaders.CHANNEL);
        Long deliveryTag = (Long) message.getHeaders().get(AmqpHeaders.DELIVERY_TAG);
        channel.basicAck(deliveryTag, false);
        String payload = message.getPayload();
        System.out.println(payload);
    }
}
```

### 步骤5：查看消息

如果您想确认消息是否成功发送至 TDMQ RabbitMQ 版，可以在控制台 集群管理 > Queue 基本信息页面查看接入的消费者情况。




> 说明
>
> 上述是基于 RabbitMQ 的发布订阅模型的一个简单示例，可根据实际使用进行不同配置，具体可参见 Demo 示例 或 Spring cloud stream 官网。

# Java SDK

## 操作场景

本文以调用 Java SDK 为例介绍通过开源 SDK 实现消息收发的操作过程，帮助您更好地理解消息收发的完整过程。

## 前提条件

- 完成集群创建与准备
- 完成Vhost、Queue和用户创建。Vhost 打开 Trace 插件。
- 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 pom.xml 添加以下依赖：

```
<!-- in your <dependencies> block -->
<dependency>
    <groupId>com.rabbitmq</groupId>
    <artifactId>amqp-client</artifactId>
    <version>5.13.0</version>
</dependency>
```

### 步骤2：生产消息

创建并编译运行 MessageProducer.java。

```
import com.rabbitmq.client.Channel;
import com.rabbitmq.client.Connection;
import com.rabbitmq.client.ConnectionFactory;

/**
 * rabbitmq Hello World 模型示例
 * 消息生产者
 */
public class MessageProducer {

    /**
     * 消息队列名称
     */
    public static final String QUEUE_NAME = "hello-world-queue";


    public static void main(String[] args) throws Exception {
        // 连接工厂
        ConnectionFactory factory = new ConnectionFactory();
        // 设置服务地址
        factory.setUri("amqp://*****");
        // 设置Virtual Hosts
```

```
        factory.setVirtualHost("VHOST-NAME");
        // 设置用户名
        factory.setUsername("USERNAME");
        // 设置密码
        factory.setPassword("PASSWORD");
        Connection connection = null;
        Channel channel = null;
        try {
            // 获取连接
            connection = factory.newConnection();
            // 建立通道
            channel = connection.createChannel();
            // 绑定消息队列
            channel.queueDeclare(QUEUE_NAME, true, false, false, null);
            String message = "Hello World!";
            // 发布消息 (hello world 消息模型无需指定交换机类型)
            channel.basicPublish("", QUEUE_NAME, null, message.getBytes());
            System.out.println(" [Producer(hello world)] Sent '" + message + "'");
        } catch (Exception e) {
            e.printStackTrace();
        } finally {
            // 释放资源
            if (channel != null) {
                channel.close();
            }
            if (connection != null) {
                connection.close();
            }
        }
    }
}
```

| 参数 | 说明 |
| --- | --- |
| QUEUE_NAME | Queue名称，在控制台 Queue 列表获取。 |
| factory.setUri | 集群接入地址，集群接入地址，在集群基本信息页面的 客户端接入 模块获取。<br> <br>客户端接入 添加路由策略<br>接入类型 访问策略 带宽 网络 操作<br>VPC 网络 - - vpc-atw25e81 subnet-0vbttohk amqp:// [redacted] 删除<br>公网域名接入 白名单：[redacted]（默认禁止所有IP访问） 修改 3Mbps amqp://[redacted]:5672 删除 |
| factory.setVirtualHost | Vhost 名称，在控制台 Vhost 列表获取。 |
| factory.setUsername | 用户名称，填写在控制台创建的用户名称。 |
| factory.setPassword | 用户密码，填写在控制台创建用户时填写的密码。 |

步骤3：消费消息

创建并编译运行 MessageConsumer.java。

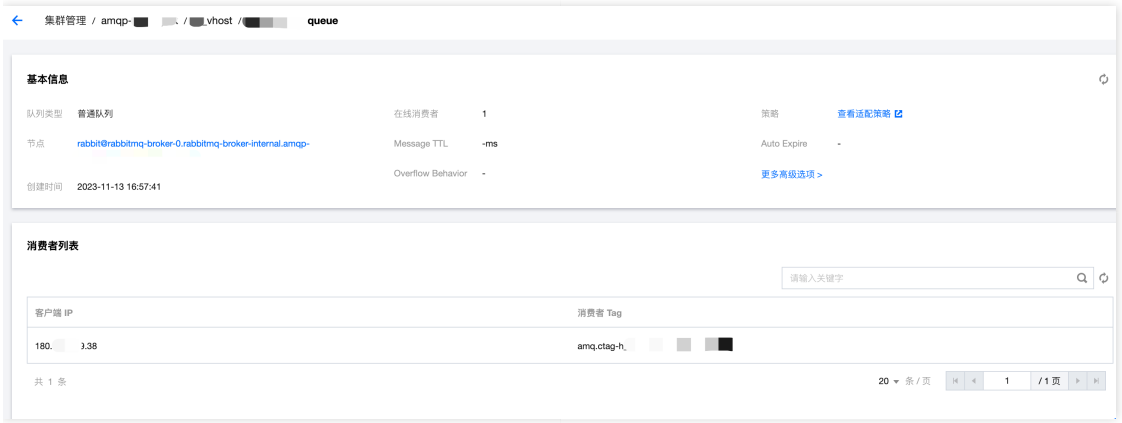
```
import com.rabbitmq.client.Channel;
import com.rabbitmq.client.Connection;
import com.rabbitmq.client.ConnectionFactory;
import com.rabbitmq.client.DeliverCallback;

import java.nio.charset.StandardCharsets;

/**
 * rabbitmq Hello World 模型示例
 * 消息消费者
 */
public class MessageConsumer {

    /**
     * 消息队列名称
     */
    public static final String QUEUE_NAME = "hello-world-queue";

    public static void main(String[] args) throws Exception {
        // 连接工厂
        ConnectionFactory factory = new ConnectionFactory();
        // 设置服务地址
        factory.setUri("amqp://*****");
        // 设置Virtual Hosts
        factory.setVirtualHost("VHOST-NAME");
        // 设置用户名
        factory.setUsername("USERNAME");
        // 设置密码
        factory.setPassword("PASSWORD");
        // 获取连接、建立通道
        Connection connection = factory.newConnection();
        Channel channel = connection.createChannel();
        // 绑定消息队列
        channel.queueDeclare(QUEUE_NAME, true, false, false, null);
        System.out.println(" [Consumer(hello world)] Waiting for messages.");
        // 消息处理回调；当收到消息后会执行该消息处理逻辑
        DeliverCallback deliverCallback = (consumerTag, delivery) -> {
            String message = new String(delivery.getBody(), StandardCharsets.UTF_8);
            System.out.println(" [Consumer(hello world)] Received '" + message + "'");
        };
        // 订阅消息 (第二个参数为true表示收到消息自动确认)
        channel.basicConsume(QUEUE_NAME, true, deliverCallback, consumerTag -> {
            System.out.println("consumerTag = " + consumerTag);
        });
    }
}
```

| 参数 | 说明 |
| --- | --- |
| QUEUE_NAME | Queue名称，在控制台 Queue 列表获取。 |
| factory.setUri | 集群接入地址，在集群基本信息页面的 客户端接入 模块获取。 |

| 参数 | 说明 |
| --- | --- |
| | |
| factory.setVirtualHost | Vhost 名称，在控制台 Vhost 列表获取。 |
| factory.setUsername | 用户名称，填写在控制台创建的用户名称。 |
| factory.setPassword | 用户密码，填写在控制台创建用户时填写的密码。 |

## 步骤4：查看消费者

可以在控制台 集群管理 > Queue 基本信息页面查看接入的消费者情况。




> 说明
>
> 上述是基于 RabbitMQ 的发布订阅模型的一个简单示例。其他示例可参见 Demo 或 RabbitMQ 官网 实例。

# Go SDK

## 操作场景

本文以调用 Go SDK 为例介绍通过开源 SDK 实现消息收发的操作过程，帮助您更好地理解消息收发的完整过程。

## 前提条件

- 完成资源创建与准备
- 安装 Go
- 下载 Demo

## 操作步骤

1. 执行如下命令在客户端环境安装所需包。

   ```
   go get "github.com/rabbitmq/amqp091-go"
   ```

2. 安装完成后，即可引入到您的 GO 工程文件中。

   ```
   import (amqp "github.com/rabbitmq/amqp091-go")
   ```

3. 引入之后即可在您的项目中使用客户端。

## 使用示例

1. 建立连接和通信信道。

   ```
   // 所需参数
   const (
       host     = "amqp-xx.rabbitmq.x.xxxxxtdmq.com" // 服务接入地址
       username = "roleName" // 角色控制台对应的角色名称
       password = "eyJrZX..." // 角色对应的密钥
       vhost    = "amqp-xx|Vhost" // 要使用的Vhost全称
   )
   // 创建连接
   conn, err := amqp.Dial("amqp://" + username + ":" + password + "@" + host + ":5672/" + vhost)
   failOnError(err, "Failed to connect to RabbitMQ")
   defer func(conn *amqp.Connection) {
       err := conn.Close()
       if err != nil {
       }
   }(conn)

   // 建立通道
   ch, err := conn.Channel()
   ```

```
failOnError(err, "Failed to open a channel")
defer func(ch *amqp.Channel) {
    err := ch.Close()
    if err != nil {
    }
}(ch)
```

| 参数 | 说明 |
| --- | --- |
| host | 集群接入地址，在集群基本信息页面的客户端接入模块获取。<br> |
| username | 用户名称，填写在控制台创建的用户名称。 |
| password | 用户密码，填写在控制台创建用户时填写的密码。 |
| vhost | Vhost 名称，在控制台 Vhost 列表获取。 |

2. 声明交换机。

```
// 声明交换机（名称和类型需要与存在的交换机保持一致）
    err = ch.ExchangeDeclare(
        "logs-exchange",   // 交换机名称
        "fanout", // 交换机类型
        true,     // durable
        false,    // auto-deleted
        false,    // internal
        false,    // no-wait
        nil,      // arguments
    )
    failOnError(err, "Failed to declare a exchange")
```

3. 发布消息。

消息可发给交换机，也可以直接发到指定队列（hello world 和 work queues 消息模型)。

- 发布消息到交换机：

```
// 消息内容
body := "this is new message."
// 发布消息到交换机
err = ch.Publish(
```

```
    "logs-exchange",        // exchange
    "", // routing key  （根据使用的交换机类型可选择的是否需要routing key），如果不选择交换机，该参数为消息队列名称
    false,      // mandatory
    false,      // immediate
    amqp.Publishing{
        ContentType: "text/plain",
        Body:        []byte(body),
    })
failOnError(err, "Failed to publish a message")
```

- 发布消息到指定队列：

```
// 发布消息到指定的消息队列
err = ch.Publish(
"",         // exchange
queue.Name, // routing key
false,      // mandatory
false,      // immediate
amqp.Publishing{
    ContentType: "text/plain",
    Body:        []byte(body),
})
failOnError(err, "Failed to publish a message")
```

4. 订阅消息。

```
// 创建消费者并消费指定消息队列中的消息
msgs, err := ch.Consume(
    "message-queue", // message-queue
    "",          // consumer
    false,        // 设置为非自动确认(可根据需求自己选择)
    false,        // exclusive
    false,        // no-local
    false,        // no-wait
    nil,          // args
)
failOnError(err, "Failed to register a consumer")

// 获取消息队列中的消息
forever := make(chan bool)
go func() {
    for d := range msgs {
        log.Printf("Received a message: %s", d.Body)
        t := time.Duration(1)
        time.Sleep(t * time.Second)
        // 手动回复ack
        d.Ack(false)
    }
}()
log.Printf(" [Consumer] Waiting for messages.")
<-forever
```

5. 消费者使用 routing key。

```
// 需要在消息队列中指定 交换机 和 routing key
err = ch.QueueBind(
    q.Name, // queue name
    "routing_key",     // routing key
    "topic_demo", // exchange
    false,
    nil,
)
failOnError(err, "Failed to bind a queue")
```

> 说明
>
> 详细使用示例可参见 Demo 或 RabbitMQ 官网。

# Python SDK

## 操作场景

本文以调用 Python SDK 为例介绍通过开源 SDK 实现消息收发的操作过程，帮助您更好地理解消息收发的完整过程。

## 前提条件

- 完成资源创建与准备
- 安装 Python
- 安装 pip
- 下载 Demo

## 操作步骤

### 步骤1：添加依赖

1. 根据 RabbitMQ 官网推荐使用 pika，首先要在客户端使用环境中安装 pika。

   ```
   python -m pip install pika --upgrade
   ```

2. 在创建客户端时导入 pika。

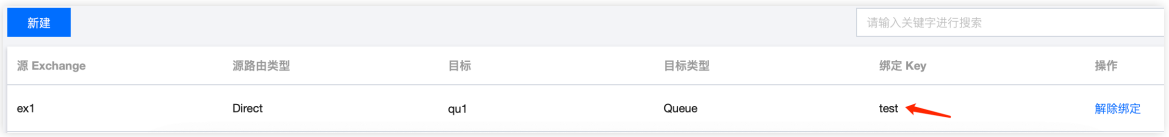
   ```
   import pika
   ```

### 步骤2：生产消息

创建并编译运行生产消息程序 messageProducer.py。

```
import pika

# 使用用户名和密码创建登录凭证对象
credentials = pika.PlainCredentials('rolename', 'eyJr***')
# 创建连接
connection = pika.BlockingConnection(pika.ConnectionParameters(
    host='amqp-xx.rabbitmq.x.com', port=5672, virtual_host='amqp-xxx|Vhostname', credentials=credentials))
# 建立信道
channel = connection.channel()
# 声明交换机
channel.exchange_declare(exchange='direct_exchange', exchange_type="direct")

routingKeys = ['aaa.bbb.ccc', 'aaa.bbb.ddd', 'aaa.ccc.zzz', "xxx.yyy.zzz"]

for routingKey in routingKeys:
    # 发送消息到指定的交换机
    # 不指定交换机的情况下发送消息，需要指定消息队列，参数routing_key在使用指定交换机时，表示routing_key，不指定交换机时代表消息队列名称
    channel.basic_publish(exchange='direct_exchange',
```

```
                routing_key=routingKey,
                body=(routingKey + 'This is a new direct message.').encode(),
                properties=pika.BasicProperties(
                    delivery_mode=2,  # 设置消息持久化
                ))
    print('send success msg to rabbitmq')
connection.close()
```

| 参数 | 说明 |
| --- | --- |
| rolename | 用户名称，填写在控制台创建的用户名称。 |
| eyJr*** | 用户密码，填写在控制台创建用户时填写的密码。 |
| host | 集群接入地址，在集群基本信息页面的客户端接入模块获取。<br> |
| port | 集群接入地址端口，在集群基本信息页面的客户端接入模块获取。 |
| virtual_host | Vhost 名称，在控制台 Vhost 列表获取。 |
| direct_exchange | Exchange 名称，在控制台 Exchange 列表获取。 |
| routingKeys | 消息的路由规则，在控制台 绑定关系列表的绑定 Key列获取。<br>  |

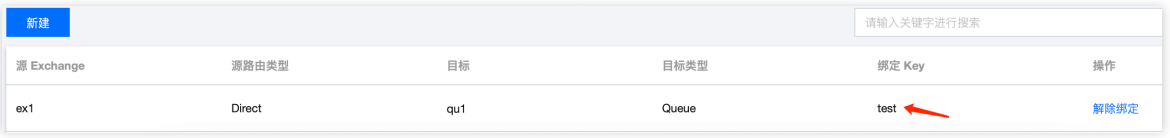
## 步骤3：消费消息

创建并编译运行消费消息程序 messageConsumer.py。

```
import os
import pika
import sys

def main():
    # 使用用户名和密码创建登录凭证对象
    credentials = pika.PlainCredentials('rolename', 'eyJr***')
    # 创建连接
    connection = pika.BlockingConnection(pika.ConnectionParameters(
        host='amqp-xx.rabbitmq.x.com', port=5672, virtual_host='amqp-xxx|Vhostname', credentials=credentials))
```

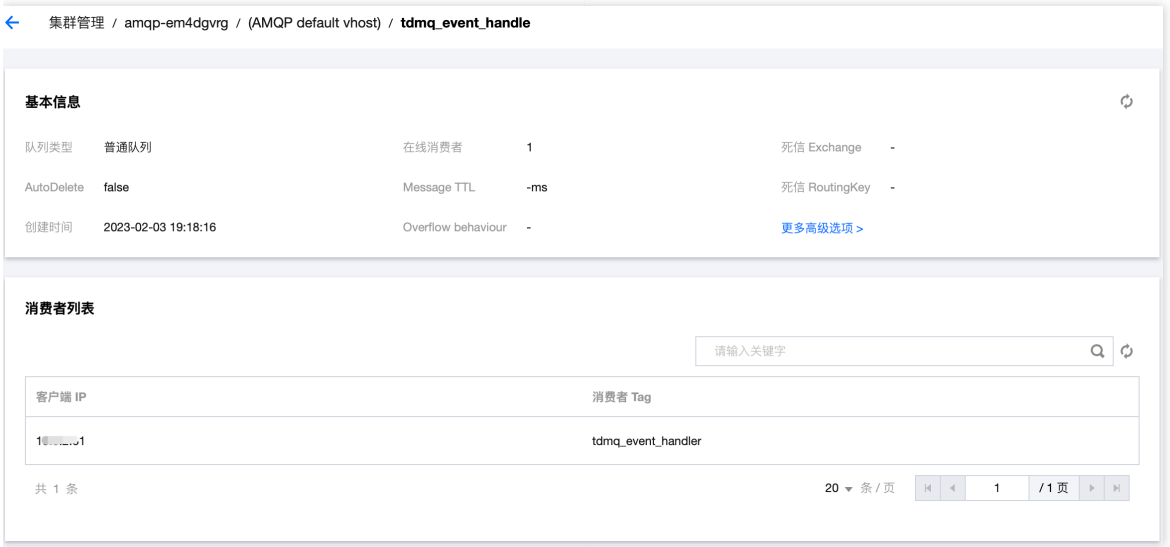
```
    # 建立信道
    channel = connection.channel()
    # 声明消息队列
    channel.queue_declare(queue='route_queue1', exclusive=True, durable=True)
    # 绑定消息队列到交换机，并指定 routing key
    routing_keys = ['aaa.bbb.ccc', 'aaa.bbb.ddd']
    for routingKey in routing_keys:
        channel.queue_bind(exchange='direct_exchange', queue="route_queue1", routing_key=routingKey)
    # 设置只接受一个未确认消息
    channel.basic_qos(prefetch_count=1)

    # 消息消费逻辑
    def callback(ch, method, properties, body):
        print(" [Consumer1(Direct 'aaa.bbb.ccc'/'aaa.bbb.ddd')] Received (%r)" % body)
        # 手动回复ACK
        ch.basic_ack(delivery_tag=method.delivery_tag)

    # 创建消费者，消费消息队列中的消息
    channel.basic_consume(queue='route_queue1',
                          on_message_callback=callback,
                          auto_ack=False)  # 设置为非自动确认

    print(" [Consumer1(Direct 'aaa.bbb.ccc'/'aaa.bbb.ddd')] Waiting for messages. To exit press CTRL+C")
    channel.start_consuming()

if __name__ == '__main__':
    try:
        main()
    except KeyboardInterrupt:
        print('Interrupted')
        try:
            sys.exit(0)
        except SystemExit:
            os._exit(0)
```

| 参数 | 说明 |
| --- | --- |
| rolename | 用户名称，填写在控制台创建的用户名称。 |
| eyJr*** | 用户密码，填写在控制台创建用户时填写的密码。 |
| host | 集群接入地址，在集群基本信息页面的获取接入地址获取。<br> <br>客户端接入 添加路由策略<br>接入类型 / 访问策略 / 带宽 / 网络 / 操作<br>VPC 网络 - - vpc-atw25e81 subnet-0vbttohk amqp:// 2 删除<br>公网域名接入 白名单：1（默认禁止所有IP访问） 修改 3Mbps amqp://4 4:5672 删除 |

| 参数 | 说明 |
| --- | --- |
| port | 集群接入地址端口，在集群基本信息页面的获取接入地址获取。 |
| virtual_host | Vhost 名称，在控制台 Vhost 列表获取。 |
| direct_exchange | Exchange 名称，在控制台 Exchange 列表获取。 |
| route_queue1 | Queue名称，在控制台 Queue 列表获取。 |
| routingKey | 消息的路由规则，在控制台 绑定关系列表的绑定 Key列获取。<br>  <br><br>img |

## 步骤4：查看消息

如果您想确认消息是否成功发送至 TDMQ RabbitMQ 版，可以在控制台 集群管理 > Queue 基本信息页面查看接入的消费者情况。




> 说明
>
> 完整示例或其他使用可参见 Demo 或者 RabbitMQ 官方使用文档。

# PHP SDK

## 操作场景

本文以调用 PHP SDK 为例介绍通过开源 SDK 实现消息收发的操作过程，帮助您更好地理解消息收发的完整过程。

## 前提条件

- 安装 PHP 5.6 或以上版本
- 安装 PEAR
- 下载 Demo

## 操作步骤

### 步骤1：安装 php-amqplib 库

根据 RabbitMQ 官网推荐使用 php-amqplib Client，首先需要在项目中引入 php-amqplib 库。

1. 在项目中添加 composer.json 文件。

```
{
    "require": {
        "php-amqplib/php-amqplib": ">=3.0"
    }
}
```

2. 使用 Composer 进行安装。

```
composer.phar install
```

或者使用下述命令：

```
composer install
```

3. 创建客户端需要引入库文件，在客户端文件中引入库文件。

```
require_once('../vendor/autoload.php');
```

完成上述步骤即可创建连接与服务端进行交互。

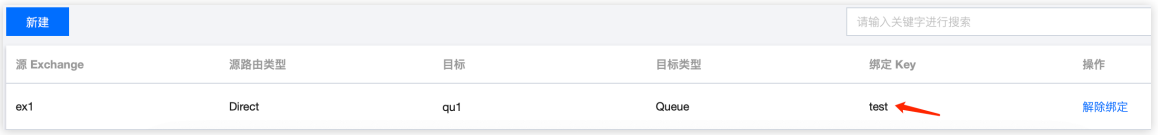
### 步骤2：发送消息

创建并编译生产消息程序（以 direct 类型交换机为例）。

```
require_once('../vendor/autoload.php');
```

```
use PhpAmqpLib\Connection\AMQPStreamConnection;
use PhpAmqpLib\Message\AMQPMessage;

$exchange_name = 'exchange_name';
$exchange_type = 'direct';

// 创建连接
$connection = new AMQPStreamConnection(
    $host,
    $port,
    $username,
    $password,
    $vhost,
    false,
    'PLAIN');
// 建立通道
$channel = $connection->channel();
// 声明交换机
$channel->exchange_declare($exchange_name, $exchange_type, false, true, false);

// 设定消息路由key
$routing_keys = array('info', 'waring', 'error');

for ($x = 0; $x < count($routing_keys); $x++) {
    // 消息内容
    $msg = new AMQPMessage('This is a direct[' . $routing_keys[$x] . '] message!');
    // 发送消息到指定的交换机并设置routing key
    $channel->basic_publish($msg, $exchange_name, $routing_keys[$x]);

    echo " [Producer(Routing)] Sent '" . $msg->body . "'\n";
}
// 资源释放
$channel->close();
$connection->close();
```

| 参数 | 说明 |
| --- | --- |
| $exchange_name | Exchange 名称，在控制台 Exchange 列表获取。 |
| $exchange_type | 类型需与上述 Exchange 的类型保持一致。 |
| $host | 集群接入地址，在集群基本信息页面的客户端接入模块获取。<br> <br> |
| $port | 集群接入地址中的端口号。 |

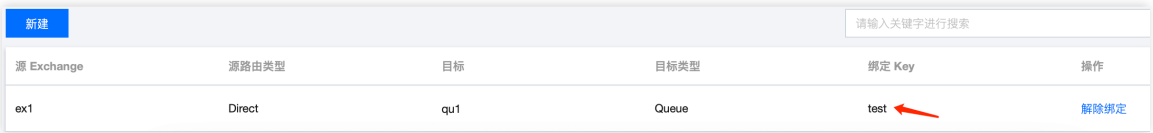
| 参数 | 说明 |
| --- | --- |
| $username | 用户名称，填写在控制台创建的用户名称。 |
| $password | 用户密码，填写在控制台创建用户时填写的密码。 |
| $vhost | Vhost 名称，在控制台 Vhost 列表获取。 |
| $routing_keys[$x] | 消费者消息队列绑定的 routing key，消息的路由规则，在控制台绑定关系列表的绑定 Key列获取。<br>  <br><br>img |

### 步骤3：消费消息

创建并编译消费消息程序。

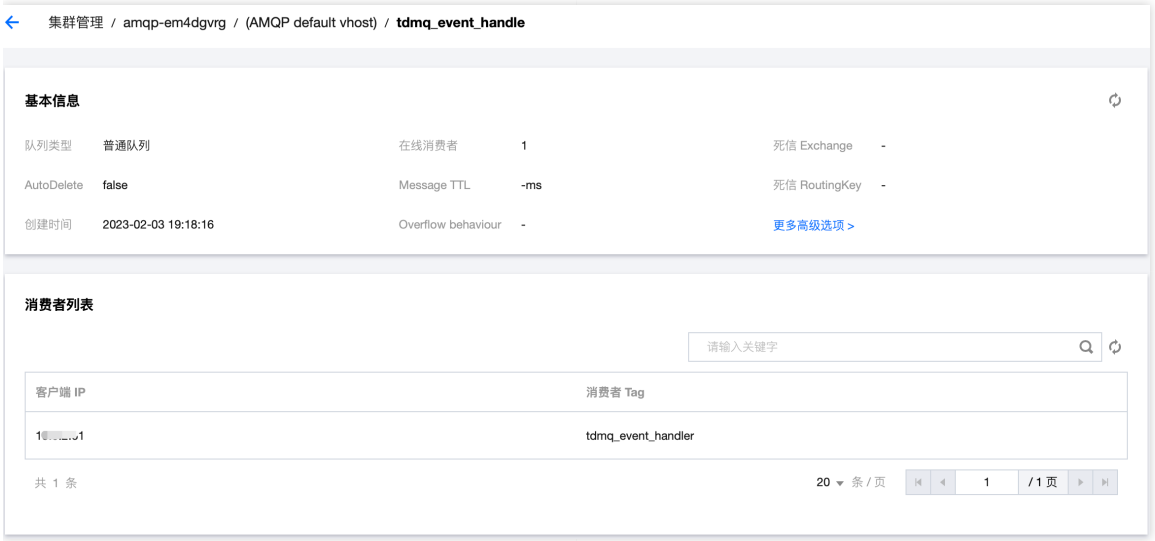
```
<?php

require_once('../vendor/autoload.php');
require_once('../Constant.php');

use PhpAmqpLib\Connection\AMQPStreamConnection;

$exchange_name = 'exchange_name';
$exchange_type = 'direct';
$queue_name = 'route_queue1';

// 创建链接
$connection = new AMQPStreamConnection(
    $host,
    $port,
    $username,
    $password,
    $vhost,
    false,
    'PLAIN');
// 建立通道
$channel = $connection->channel();
// 声明交换机
$channel->exchange_declare($exchange_name, $exchange_type, false, true, false);
// 声明消息队列
$channel->queue_declare($queue_name, false, true, false, false);

// 设定队列路由key
$routing_keys = array('info', 'waring', 'error');
for ($x = 0; $x < count($routing_keys); $x++) {
    // 绑定消息队列到指定交换机并设置routingKey
    $channel->queue_bind($queue_name, $exchange_name, $routing_keys[$x]);
}

echo " [Consumer1(Routing: info/waring/error)] Waiting for messages. To exit press CTRL+C\n";

// 消息回调（消息消费逻辑）
```

```
$callback = function ($msg) {
    echo ' [Consumer1(Routing: info/waring/error)] Received ', $msg->body, "\n";
};
// 创建消费者监听指定消息队列
$channel->basic_consume($queue_name, '', false, true, false, false, $callback);

while ($channel->is_open()) {
    $channel->wait();
}
// 关闭资源
$channel->close();
$connection->close();
```

| 参数 | 说明 |
| --- | --- |
| $exchange_name | Exchange 名称，可在控制台 Exchange 列表获取。 |
| $exchange_type | 类型需与上述exchange的类型保持一致。 |
| $queue_name | Queue名称，可在控制台 Queue 列表获取。 |
| $host | 集群接入地址，在集群基本信息页面的客户端接入模块获取。<br> <br> |
| $port | 集群接入地址中的端口号。 |
| $username | 用户名称，填写在控制台创建的用户名称。 |
| $password | 用户密码，填写在控制台创建用户时填写的密码。 |
| $vhost | Vhost 名称，在控制台 Vhost 列表获取。 |
| $routing_keys[$x] | 消息队列支持的routing key。消费者消息队列绑定的 routing key，消息的路由规则，在控制台绑定关系列表的绑定Key列获取。<br>  <br><br>img |

## 步骤4：查看消息

如果您想确认消息是否成功发送至 TDMQ RabbitMQ 版，可以在控制台 集群管理 > Queue 基本信息页面查看接入的消费者情况。




> 说明
>
> 完整示例或其他使用可参见 Demo 或者 RabbitMQ 官方使用文档。

# C++ SDK

## 操作场景

本文以调用 C++ SDK 为例介绍通过开源 SDK 实现消息收发的操作过程，帮助您更好地理解消息收发的完整过程。

## 前提条件

- 完成资源创建与准备
- 下载 Demo

## 操作步骤

### 步骤1：环境准备

1. 安装客户端相关的库 C and C++ 库， 本文以 AMQP-CPP 为例。
2. 导入动态库和头文件。

### 步骤2：生产消息

1. 建立连接。

```
auto evbase = event_base_new();
LibEventHandlerMyError hndl(evbase);

// 建立连接
AMQP::TcpConnection connection(&hndl, AMQP::Address(
    "amqp://admin:eyJrZXlJZC...@amqp-xxx.rabbitmq.ap-sh.public.xxxxxtdmq.com:5672/amqp-xxx|vhost-cpp"));
    // 服务地址格式为  amqp://username:password@host:port/vhost
// 建立通道
AMQP::TcpChannel channel(&connection);
channel.onError([&evbase](const char *message) {
    std::cout << "Channel error: " << message << std::endl;
    event_base_loopbreak(evbase);
});
```

| 参数 | 说明 |
| --- | --- |
| host | 集群接入地址，在集群基本信息页面的客户端接入模块获取。 |

| 参数 | 说明 |
| --- | --- |
| | |
| port | 集群接入地址中的端口号。 |
| username | 用户名称，填写在控制台创建的用户名称。 |
| password | 用户密码，填写在控制台创建用户时填写的密码。 |
| vhost | Vhost 名称，在控制台 Vhost 列表获取。 |

2. 发送消息。

```
// 声明交换机
channel.declareExchange(exchange_name, AMQP::ExchangeType::direct);

// 发送消息到交换机
channel.publish(exchange_name, routing_key, "Hello client this is a info message");

event_base_dispatch(evbase);
event_base_free(evbase);
```

| 参数 | 说明 |
| --- | --- |
| exchange_name | Exchange 名称，可在控制台 Exchange 列表获取。 |
| routing_key | 消息队列支持的routing key |

## 步骤3：消费消息

1. 建立连接。

```
ConnHandler handler;
// 建立连接
AMQP::TcpConnection connection(handler, AMQP::Address(host, part, AMQP::Login(username, password), vhost));
// 建立通道
AMQP::TcpChannel channel(&connection);
channel.onError([&handler](const char *message) {
        std::cout << "Channel error: " << message << std::endl;
        handler.Stop();
});
```

| 参数 | 说明 |
| --- | --- |
| host | 集群接入地址，在集群基本信息页面的客户端接入模块获取。<br> <br> |
| port | 集群接入地址中的端口号。 |
| username | 用户名称，填写在控制台创建的用户名称。 |
| password | 用户密码，填写在控制台创建用户时填写的密码。 |
| vhost | Vhost 名称，在控制台 Vhost 列表获取。 |

2. 声明交换机、消息队列，并将消息队列绑定到交换机。

```
// 声明交换机
channel.declareExchange(exchange_name, AMQP::ExchangeType::direct);
// 声明消息队列
channel.declareQueue(queue_name, AMQP::durable);
// 绑定消息队列到交换机
channel.bindQueue(exchange_name, queue_name, routing_key);
```

| 参数 | 说明 |
| --- | --- |
| exchange_name | Exchange 名称，可在控制台 Exchange 列表获取。 |
| queue_name | 消息队列名称，可在控制台 Queue 列表获取。 |
| routing_key | 消息队列支持的 routing key |

3. 订阅消息。

```
// 订阅消息
channel.consume(queue_name)
        .onReceived
        (
        [&channel](const AMQP::Message &msg, uint64_t tag, bool redelivered) {
                // 处理消息
                std::cout << "Received: " << msg.body() << std::endl;
                // 回复ack， 消费失败可回复reject
                channel.ack(tag);
        }
);
```

```
handler.Start();
```

| 参数 | 说明 |
| --- | --- |
| queue_name | 消息队列名称，可在控制台 Queue 列表获取。 |

> 说明
>
> 完整示例或其他使用可参考 Demo 或者 AMQP-CPP 和 AMQP-CPP 示例

# API文档
# （tdmq）
# 版本（2020-02-17）
# API 概览

## API版本

V3

## CMQ管理相关接口

| 接口名称 | 接口功能 |
| --- | --- |
| RewindCmqQueue | 回溯cmq队列 |

## RocketMQ 主题相关接口

| 接口名称 | 接口功能 |
| --- | --- |
| CreateRocketMQTopic | 创建RocketMQ主题 |
| DeleteRocketMQTopic | 删除RocketMQ主题 |
| DescribeRocketMQTopics | 获取RocketMQ主题列表 |
| ModifyRocketMQTopic | 更新RocketMQ主题信息 |

## RocketMQ 命名空间相关接口

| 接口名称 | 接口功能 |
| --- | --- |
| CreateRocketMQNamespace | 创建RocketMQ命名空间 |
| DeleteRocketMQNamespace | 删除RocketMQ命名空间 |

| 接口名称 | 接口功能 |
| --- | --- |
| DescribeRocketMQNamespaces | 获取RocketMQ命名空间列表 |
| ModifyRocketMQNamespace | 更新RocketMQ命名空间 |

# RocketMQ 平滑迁移相关接口

| 接口名称 | 接口功能 |
| --- | --- |
| ImportRocketMQConsumerGroups | 平滑迁移：导入消费者组列表 |
| ImportRocketMQTopics | 平滑迁移：导入topic列表 |

# RocketMQ 消息相关接口

| 接口名称 | 接口功能 |
| --- | --- |
| DescribeRocketMQMsg | rocketmq消息详情 |
| DescribeRocketMQMsgTrace | 查询RocketMQ消息轨迹 |
| DescribeRocketMQTopicMsgs | rocketmq 消息查询 |
| ExportRocketMQMessageDetail | 导出RocketMQ消息详情 |
| ResetRocketMQConsumerOffSet | 重置RocketMQ消费位点 |
| SendRocketMQMessage | 发送RocketMQ消息 |
| VerifyRocketMQConsume | Rocketmq消费验证 |

# RocketMQ 消费组相关接口

| 接口名称 | 接口功能 |
| --- | --- |
| CreateRocketMQGroup | 创建RocketMQ消费组 |
| DeleteRocketMQGroup | 删除RocketMQ消费组 |
| DescribeRocketMQConsumerConnectionDetail | 获取在线消费端详情 |

| 接口名称 | 接口功能 |
| --- | --- |
| DescribeRocketMQConsumerConnections | 获取指定消费组下当前客户端的连接情况 |
| DescribeRocketMQGroups | 获取RocketMQ消费组列表 |
| DescribeRocketMQSubscriptions | 获取RocketMQ消费组订阅关系 |
| DescribeRocketMQTopicsByGroup | 获取指定消费组下订阅的主题列表 |
| ModifyRocketMQGroup | 更新RocketMQ消费组信息 |

# RocketMQ 角色授权相关接口

| 接口名称 | 接口功能 |
| --- | --- |
| CreateRocketMQEnvironmentRole | 创建RocketMQ环境角色授权 |
| CreateRocketMQRole | 创建RocketMQ角色 |
| DeleteRocketMQEnvironmentRoles | 批量删除RocketMQ环境角色授权 |
| DeleteRocketMQRoles | 批量删除RocketMQ角色 |
| DescribeRocketMQEnvironmentRoles | 获取RocketMQ命名空间角色列表 |
| DescribeRocketMQRoles | 获取RocketMQ角色列表 |
| ModifyRocketMQEnvironmentRole | 修改RocketMQ环境角色授权 |
| ModifyRocketMQRole | RocketMQ角色修改 |

# RocketMQ 集群相关接口

| 接口名称 | 接口功能 |
| --- | --- |
| CreateRocketMQCluster | 创建RocketMQ集群 |
| DeleteRocketMQCluster | 删除RocketMQ集群 |
| DescribeRocketMQCluster | 获取单个RocketMQ集群信息 |
| DescribeRocketMQClusters | 获取RocketMQ的集群列表 |
| DescribeRocketMQPublicAccessPoint | 查询RocketMQ实例公网接入点信息 |

| 接口名称 | 接口功能 |
| --- | --- |
| DescribeRocketMQVipInstances | 查询RocketMQ专享实例列表 |
| ModifyRocketMQCluster | 更新RocketMQ集群信息 |

# 主题相关接口

| 接口名称 | 接口功能 |
| --- | --- |
| CreateTopic | 新增主题 |
| DeleteTopics | 删除主题 |
| DescribeTopics | 查询主题列表 |
| ModifyTopic | 修改主题 |

# 其他接口

| 接口名称 | 接口功能 |
| --- | --- |
| DeleteRoles | 删除角色 |
| DescribeBindVpcs | 获取租户VPC绑定关系 |
| SendMsg | 发送消息 |

# 命名空间相关接口

| 接口名称 | 接口功能 |
| --- | --- |
| CreateEnvironment | 创建命名空间 |
| DeleteEnvironments | 删除命名空间 |
| DescribeEnvironmentAttributes | 获取命名空间属性 |
| DescribeEnvironmentRoles | 获取命名空间角色列表 |
| DescribeEnvironments | 获取命名空间列表 |

| 接口名称 | 接口功能 |
| --- | --- |
| ModifyEnvironmentAttributes | 修改命名空间属性 |

# 消息相关接口

| 接口名称 | 接口功能 |
| --- | --- |
| AcknowledgeMessage | 确认消息 |
| ReceiveMessage | 接收消息 |
| SendBatchMessages | 批量发送消息 |
| SendMessages | 发送单条消息 |

# 环境角色授权相关接口

| 接口名称 | 接口功能 |
| --- | --- |
| CreateEnvironmentRole | 创建环境角色授权 |
| CreateRole | 创建角色 |
| DeleteEnvironmentRoles | 删除环境角色授权 |
| DescribeRoles | 获取角色列表 |
| ModifyEnvironmentRole | 修改环境角色授权 |
| ModifyRole | 角色修改 |
| TpoModifyResourceProject | 修改资源对应的项目（包含新增） |

# 生产消费相关接口

| 接口名称 | 接口功能 |
| --- | --- |
| CreateSubscription | 创建订阅关系 |
| DeleteSubscriptions | 删除订阅关系 |

| 接口名称 | 接口功能 |
| --- | --- |
| DescribeSubscriptions | 消费订阅列表 |
| ResetMsgSubOffsetByTimestamp | 消息回溯 |

# 集群相关接口

| 接口名称 | 接口功能 |
| --- | --- |
| CreateCluster | 创建集群 |
| DeleteCluster | 删除集群 |
| DescribeBindClusters | 获取专享集群列表 |
| DescribeClusterDetail | 获取集群详情 |
| DescribeClusters | 获取集群列表 |
| ModifyCluster | 更新集群信息 |

# 调用方式
# 接口签名v1

TCloudFinanceZone API 会对每个访问请求进行身份验证，即每个请求都需要在公共请求参数中包含签名信息（ Signature ）以验证请求者身份。
签名信息由安全凭证生成，安全凭证包括 SecretId 和 SecretKey；若用户还没有安全凭证，请前往云API密钥页面申请，否则无法调用云API接口。

## 1. 申请安全凭证

在第一次使用云API之前，请前往云API密钥页面申请安全凭证。
安全凭证包括 SecretId 和 SecretKey：

- SecretId 用于标识 API 调用者身份
- SecretKey 用于加密签名字符串和服务器端验证签名字符串的密钥。
- 用户必须严格保管安全凭证，避免泄露。

申请安全凭证的具体步骤如下：

1. 登录TCloudFinanceZone管理中心控制台。
2. 前往云API密钥的控制台页面
3. 在云API密钥页面，点击【新建】即可以创建一对SecretId/SecretKey

注意：开发商帐号最多可以拥有两对 SecretId / SecretKey。

## 2. 生成签名串

有了安全凭证SecretId 和 SecretKey后，就可以生成签名串了。以下是生成签名串的详细过程：

假设用户的 SecretId 和 SecretKey 分别是：

- SecretId: AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE
- SecretKey: Gu5t9xGARNpq86cd98joQYCN3EXAMPLE

注意：这里只是示例，请根据用户实际申请的 SecretId 和 SecretKey 进行后续操作！

以云服务器查看实例列表(DescribeInstances)请求为例，当用户调用这一接口时，其请求参数可能如下:

| 参数名称 | 中文 | 参数值 |
| --- | --- | --- |

| 参数名称 | 中文 | 参数值 |
| --- | --- | --- |
| Action | 方法名 | DescribeInstances |
| SecretId | 密钥Id | AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE |
| Timestamp | 当前时间戳 | 1465185768 |
| Nonce | 随机正整数 | 11886 |
| Region | 实例所在区域 | shjr |
| InstanceIds.0 | 待查询的实例ID | ins-09dx96dg |
| Offset | 偏移量 | 0 |
| Limit | 最大允许输出 | 20 |
| Version | 接口版本号 | 2017-03-12 |

## 2.1. 对参数排序

首先对所有请求参数按参数名的字典序（ ASCII 码）升序排序。注意：1）只按参数名进行排序，参数值保持对应即可，不参与比大小；2）按 ASCII 码比大小，如 InstanceIds.2 要排在 InstanceIds.12 后面，不是按字母表，也不是按数值。用户可以借助编程语言中的相关排序函数来实现这一功能，如 php 中的 ksort 函数。上述示例参数的排序结果如下:

```
{
    'Action' : 'DescribeInstances',
    'InstanceIds.0' : 'ins-09dx96dg',
    'Limit' : 20,
    'Nonce' : 11886,
    'Offset' : 0,
    'Region' : 'shjr',
    'SecretId' : 'AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE',
    'Timestamp' : 1465185768,
    'Version': '2017-03-12',
}
```

使用其它程序设计语言开发时，可对上面示例中的参数进行排序，得到的结果一致即可。

## 2.2. 拼接请求字符串

此步骤生成请求字符串。
将把上一步排序好的请求参数格式化成“参数名称”=“参数值”的形式，如对 Action 参数，其参数名称为 "Action" ，参数值为 "DescribeInstances" ，因此格式化后就为 Action=DescribeInstances 。
注意：“参数值”为原始值而非url编码后的值。

然后将格式化后的各个参数用"&"拼接在一起，最终生成的请求字符串为:

```
Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=shjr&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE&Timestamp=1465185768&Version=2017-03-12
```

## 2.3. 拼接签名原文字符串

此步骤生成签名原文字符串。
签名原文字符串由以下几个参数构成:

1. 请求方法: 支持 POST 和 GET 方式，这里使用 GET 请求，注意方法为全大写。
2. 请求主机:查看实例列表(DescribeInstances)的请求域名为：cvm.finance.cloud.tencent.com。实际的请求域名根据接口所属模块的不同而不同，详见各接口说明。
3. 请求路径: 当前版本云API的请求路径固定为 / 。
4. 请求字符串: 即上一步生成的请求字符串。

签名原文串的拼接规则为: 请求方法 + 请求主机 +请求路径 + ? + 请求字符串

示例的拼接结果为：

```
GETcvm.finance.cloud.tencent.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=shjr&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE&Timestamp=1465185768&Version=2017-03-12
```

## 2.4. 生成签名串

此步骤生成签名串。
首先使用 HMAC-SHA1 算法对上一步中获得的签名原文字符串进行签名，然后将生成的签名串使用 Base64 进行编码，即可获得最终的签名串。

具体代码如下，以 PHP 语言为例:

```
$secretKey = 'Gu5t9xGARNpq86cd98joQYCN3EXAMPLE';
$srcStr = 'GETcvm.finance.cloud.tencent.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=shjr&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE&Timestamp=1465185768&Version=2017-03-12';
$signStr = base64_encode(hash_hmac('sha1', $srcStr, $secretKey, true));
echo $signStr;
```

最终得到的签名串为:

```
EliP9YW3pW28FpsEdkXt/+WcGeI=
```

使用其它程序设计语言开发时，可用上面示例中的原文进行签名验证，得到的签名串与例子中的一致即可。

# 3. 签名串编码

生成的签名串并不能直接作为请求参数，需要对其进行 URL 编码。

如上一步生成的签名串为 EliP9YW3pW28FpsEdkXt/+WcGeI= ，最终得到的签名串请求参数（Signature）为：EliP9YW3pW28FpsEdkXt%2f%2bWcGeI%3d，它将用于生成最终的请求 URL。

注意：如果用户的请求方法是 GET，或者请求方法为 POST 同时 Content-Type 为 application/x-www-form-urlencoded，则发送请求时所有请求参数的值均需要做 URL 编码，参数键和=符号不需要编码。非 ASCII 字符在 URL 编码前需要先以 UTF-8 进行编码。

注意：有些编程语言的 http 库会自动为所有参数进行 urlencode，在这种情况下，就不需要对签名串进行 URL 编码了，否则两次 URL 编码会导致签名失败。

注意：其他参数值也需要进行编码，编码采用 RFC 3986。使用 %XY 对特殊字符例如汉字进行百分比编码，其中“X”和“Y”为十六进制字符（0-9 和大写字母 A-F），使用小写将引发错误。

# 4. 签名失败

根据实际情况，存在以下签名失败的错误码，请根据实际情况处理

| 错误代码 | 错误描述 |
| --- | --- |
| AuthFailure.SignatureExpire | 签名过期 |
| AuthFailure.SecretIdNotFound | 密钥不存在 |
| AuthFailure.SignatureFailure | 签名错误 |
| AuthFailure.TokenFailure | token 错误 |
| AuthFailure.InvalidSecretId | 密钥非法（不是云 API 密钥类型） |

# 5. 签名演示

在实际调用 API 3.0 时，推荐使用配套的TCloudFinanceZone SDK 3.0 ，SDK 封装了签名的过程，开发时只关注产品提供的具体接口即可。详细信息参见 SDK 中心。当前支持的编程语言有：

- Python
- Java

- PHP
- Go
- Node

为了更清楚的解释签名过程，下面以实际编程语言为例，将上述的签名过程具体实现。请求的域名、调用的接口和参数的取值都以上述签名过程为准，代码只为解释签名过程，并不具备通用性，实际开发请尽量使用 SDK 。

最终输出的 url 可能为： https://cvm.finance.cloud.tencent.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=shjr&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE&Signature=EliP9YW3pW28FpsEdkXt%2F%2BWcGeI%3D&Timestamp=1465185768&Version=2017-03-12

注意：由于示例中的密钥是虚构的，时间戳也不是系统当前时间，因此如果将此 url 在浏览器中打开或者用 curl 等命令调用时会返回鉴权错误：签名过期。为了得到一个可以正常返回的 url ，需要修改示例中的 SecretId 和 SecretKey 为真实的密钥，并使用系统当前时间戳作为 Timestamp 。

注意：在下面的示例中，不同编程语言，甚至同一语言每次执行得到的 url 可能都有所不同，表现为参数的顺序不同，但这并不影响正确性。只要所有参数都在，且签名计算正确即可。

注意：以下代码仅适用于 API 3.0，不能直接用于其他的签名流程，即使是旧版的 API ，由于存在细节差异也会导致签名计算错误，请以对应的实际文档为准。

### Java

```
import java.io.UnsupportedEncodingException;
import java.net.URLEncoder;
import java.util.Random;
import java.util.TreeMap;
import javax.crypto.Mac;
import javax.crypto.spec.SecretKeySpec;
import javax.xml.bind.DatatypeConverter;

public class CloudAPIDemo {
    private final static String CHARSET = "UTF-8";

    public static String sign(String s, String key, String method) throws Exception {
        Mac mac = Mac.getInstance(method);
        SecretKeySpec secretKeySpec = new SecretKeySpec(key.getBytes(CHARSET), mac.getAlgorithm());
        mac.init(secretKeySpec);
        byte[] hash = mac.doFinal(s.getBytes(CHARSET));
        return DatatypeConverter.printBase64Binary(hash);
    }

    public static String getStringToSign(TreeMap<String, Object> params) {
        StringBuilder s2s = new StringBuilder("GETcvm.finance.cloud.tencent.com/?");
```

```
        // 签名时要求对参数进行字典排序，此处用TreeMap保证顺序
        for (String k : params.keySet()) {
            s2s.append(k).append("=").append(params.get(k).toString()).append("&");
        }
        return s2s.toString().substring(0, s2s.length() - 1);
    }

    public static String getUrl(TreeMap<String, Object> params) throws UnsupportedEncodingException
{
        StringBuilder url = new StringBuilder("https://cvm.finance.cloud.tencent.com/?");
        // 实际请求的url中对参数顺序没有要求
        for (String k : params.keySet()) {
            // 需要对请求串进行urlencode，由于key都是英文字母，故此处仅对其value进行urlencode
            url.append(k).append("=").append(URLEncoder.encode(params.get(k).toString(), CHARSET)).app
end("&");
        }
        return url.toString().substring(0, url.length() - 1);
    }

    public static void main(String[] args) throws Exception {
        TreeMap<String, Object> params = new TreeMap<String, Object>(); // TreeMap可以自动排序
        // 实际调用时应当使用随机数，例如：params.put("Nonce", new Random().nextInt(java.lang.Intege
r.MAX_VALUE));
        params.put("Nonce", 11886); // 公共参数
        // 实际调用时应当使用系统当前时间，例如：  params.put("Timestamp", System.currentTimeMillis() /
1000);
        params.put("Timestamp", 1465185768); // 公共参数
        params.put("SecretId", "AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE"); // 公共参数
        params.put("Action", "DescribeInstances"); // 公共参数
        params.put("Version", "2017-03-12"); // 公共参数
        params.put("Region", "shjr"); // 公共参数
        params.put("Limit", 20); // 业务参数
        params.put("Offset", 0); // 业务参数
        params.put("InstanceIds.0", "ins-09dx96dg"); // 业务参数
        params.put("Signature", sign(getStringToSign(params), "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE
", "HmacSHA1")); // 公共参数
        System.out.println(getUrl(params));
    }
}
```

## Python

注意：如果是在 Python 2 环境中运行，需要先安装 requests 依赖包：  pip install requests 。

```
# -*- coding: utf8 -*-
import base64
```

```
import hashlib
import hmac
import time

import requests

secret_id = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE"
secret_key = "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE"

def get_string_to_sign(method, endpoint, params):
    s = method + endpoint + "/?"
    query_str = "&".join("%s=%s" % (k, params[k]) for k in sorted(params))
    return s + query_str

def sign_str(key, s, method):
    hmac_str = hmac.new(key.encode("utf8"), s.encode("utf8"), method).digest()
    return base64.b64encode(hmac_str)

if __name__ == '__main__':
    endpoint = "cvm.finance.cloud.tencent.com"
    data = {
        'Action' : 'DescribeInstances',
        'InstanceIds.0' : 'ins-09dx96dg',
        'Limit' : 20,
        'Nonce' : 11886,
        'Offset' : 0,
        'Region' : 'shjr',
        'SecretId' : secret_id,
        'Timestamp' : 1465185768, # int(time.time())
        'Version': '2017-03-12'
    }
    s = get_string_to_sign("GET", endpoint, data)
    data["Signature"] = sign_str(secret_key, s, hashlib.sha1)
    print(data["Signature"])
    # 此处会实际调用，成功后可能产生计费
    # resp = requests.get("https://" + endpoint, params=data)
    # print(resp.url)
```

# 接口签名v3

TCloudFinanceZone API 会对每个访问请求进行身份验证，即每个请求都需要在公共请求参数中包含签名信息（Signature）以验证请求者身份。
签名信息由安全凭证生成，安全凭证包括 SecretId 和 SecretKey；若用户还没有安全凭证，请前往云API密钥页面申请，否则无法调用云API接口。

## 1. 申请安全凭证

在第一次使用云API之前，请前往云API密钥页面申请安全凭证。
安全凭证包括 SecretId 和 SecretKey：

- SecretId 用于标识 API 调用者身份
- SecretKey 用于加密签名字符串和服务器端验证签名字符串的密钥。
- 用户必须严格保管安全凭证，避免泄露。

申请安全凭证的具体步骤如下：

1. 登录TCloudFinanceZone管理中心控制台。
2. 前往云API密钥的控制台页面
3. 在云API密钥页面，点击【新建】即可以创建一对SecretId/SecretKey

注意：开发商帐号最多可以拥有两对 SecretId / SecretKey。

## 2. TC3-HMAC-SHA256 签名方法

注意：对于GET方法，只支持 Content-Type: application/x-www-form-urlencoded 协议格式。对于POST方法，目前支持 Content-Type: application/json 以及 Content-Type: multipart/form-data 两种协议格式，json 格式默认所有业务接口均支持，multipart 格式只有特定业务接口支持，此时该接口不能使用 json 格式调用，参考具体业务接口文档说明。

下面以云服务器查询广州区实例列表作为例子，分步骤介绍签名的计算过程。我们仅用到了查询实例列表的两个参数：Limit 和 Offset，使用 GET 方法调用。

假设用户的 SecretId 和 SecretKey 分别是：AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE 和 Gu5t9xGARNpq86cd98joQYCN3EXAMPLE

### 2.1. 拼接规范请求串

按如下格式拼接规范请求串（CanonicalRequest）：

```
CanonicalRequest =
    HTTPRequestMethod + '\n' +
    CanonicalURI + '\n' +
    CanonicalQueryString + '\n' +
    CanonicalHeaders + '\n' +
    SignedHeaders + '\n' +
    HashedRequestPayload
```

- HTTPRequestMethod：HTTP 请求方法（GET、POST ），本示例中为 GET；
- CanonicalURI：URI 参数，API 3.0 固定为正斜杠（/）；
- CanonicalQueryString：发起 HTTP 请求 URL 中的查询字符串，对于 POST 请求，固定为空字符串，对于 GET 请求，则为 URL 中问号（?）后面的字符串内容，本示例取值为：Limit=10&Offset=0。注意：CanonicalQueryString 需要经过 URL 编码。
- CanonicalHeaders：参与签名的头部信息，至少包含 host 和 content-type 两个头部，也可加入自定义的头部参与签名以提高自身请求的唯一性和安全性。拼接规则：1）头部 key 和 value 统一转成小写，并去掉首尾空格，按照 key:value\n 格式拼接；2）多个头部，按照头部 key（小写）的字典排序进行拼接。此例中为： content-type:application/x-www-form-urlencoded\nhost:cvm.finance.cloud.tencent.com\n
- SignedHeaders：参与签名的头部信息，说明此次请求有哪些头部参与了签名，和 CanonicalHeaders 包含的头部内容是一一对应的。content-type 和 host 为必选头部。拼接规则：1）头部 key 统一转成小写；2）多个头部 key（小写）按照字典排序进行拼接，并且以分号（;）分隔。此例中为： content-type;host
- HashedRequestPayload：请求正文的哈希值，计算方法为 Lowercase(HexEncode(Hash.SHA256(RequestPayload)))，对 HTTP 请求整个正文 payload 做 SHA256 哈希，然后十六进制编码，最后编码串转换成小写字母。注意：对于 GET 请求，RequestPayload 固定为空字符串，对于 POST 请求，RequestPayload 即为 HTTP 请求正文 payload。

根据以上规则，示例中得到的规范请求串如下（为了展示清晰，\n 换行符通过另起打印新的一行替代）：

```
GET
/
Limit=10&Offset=0
content-type:application/x-www-form-urlencoded
host:cvm.finance.cloud.tencent.com

content-type;host
e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855
```

## 2.2. 拼接待签名字符串

按如下格式拼接待签名字符串：

```
StringToSign =
    Algorithm + \n +
```

```
RequestTimestamp + \n +
CredentialScope + \n +
HashedCanonicalRequest
```

- Algorithm：签名算法，目前固定为 TC3-HMAC-SHA256；
- RequestTimestamp：请求时间戳，即请求头部的 X-TC-Timestamp 取值，如上示例请求为 1539084154；
- CredentialScope：凭证范围，格式为 Date/service/tc3_request，包含日期、所请求的服务和终止字符串（tc3_request）。Date 为 UTC 标准时间的日期，取值需要和公共参数 X-TC-Timestamp 换算的 UTC 标准时间日期一致；service 为产品名，必须与调用的产品域名一致，例如 cvm。如上示例请求，取值为 2018-10-09/cvm/tc3_request；
- HashedCanonicalRequest：前述步骤拼接所得规范请求串的哈希值，计算方法为 Lowercase(HexEncode(Hash.SHA256(CanonicalRequest)))。

注意：

> 1. Date 必须从时间戳 X-TC-Timestamp 计算得到，且时区为 UTC+0。如果加入系统本地时区信息，例如东八区，将导致白天和晚上调用成功，但是凌晨时调用必定失败。假设时间戳为 1551113065，在东八区的时间是 2019-02-26 00:44:25，但是计算得到的 Date 取 UTC+0 的日期应为 2019-02-25，而不是 2019-02-26。
> 2. Timestamp 必须是当前系统时间，且需确保系统时间和标准时间是同步的，如果相差超过五分钟则必定失败。如果长时间不和标准时间同步，可能导致运行一段时间后，请求必定失败（返回签名过期错误）。

根据以上规则，示例中得到的待签名字符串如下（为了展示清晰，\n 换行符通过另起打印新的一行替代）：

```
TC3-HMAC-SHA256
1539084154
2018-10-09/cvm/tc3_request
91c9c192c14460df6c1ffc69e34e6c5e90708de2a6d282cccf957dbf1aa7f3a7
```

## 2.3. 计算签名

1）计算派生签名密钥，伪代码如下

```
SecretKey = "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE"
SecretDate = HMAC_SHA256("TC3" + SecretKey, Date)
SecretService = HMAC_SHA256(SecretDate, Service)
SecretSigning = HMAC_SHA256(SecretService, "tc3_request")
```

- SecretKey：原始的 SecretKey；
- Date：即 Credential 中的 Date 字段信息，如上示例，为2018-10-09；
- Service：即 Credential 中的 Service 字段信息，如上示例，为 cvm；

2）计算签名，伪代码如下

```
Signature = HexEncode(HMAC_SHA256(SecretSigning, StringToSign))
```

- SecretSigning：即以上计算得到的派生签名密钥；
- StringToSign：即步骤2计算得到的待签名字符串；

## 2.4. 拼接 Authorization

按如下格式拼接 Authorization：

```
Authorization =
    Algorithm + ' ' +
    'Credential=' + SecretId + '/' + CredentialScope + ', ' +
    'SignedHeaders=' + SignedHeaders + ', '
    'Signature=' + Signature
```

- Algorithm：签名方法，固定为 TC3-HMAC-SHA256；
- SecretId：密钥对中的 SecretId；
- CredentialScope：见上文，凭证范围；
- SignedHeaders：见上文，参与签名的头部信息；
- Signature：签名值

根据以上规则，示例中得到的值为：

```
TC3-HMAC-SHA256 Credential=AKIDEXAMPLE/Date/service/tc3_request, SignedHeaders=content-type;host, Signature=5da7a33f6993f0614b047e5df4582db9e9bf4672ba50567dba16c6ccf174c474
```

最终完整的调用信息如下：

```
https://cvm.finance.cloud.tencent.com/?Limit=10&Offset=0

Authorization: TC3-HMAC-SHA256 Credential=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE/2018-10-09/cvm/tc3_request, SignedHeaders=content-type;host, Signature=5da7a33f6993f0614b047e5df4582db9e9bf4672ba50567dba16c6ccf174c474
Content-Type: application/x-www-form-urlencoded
Host: cvm.finance.cloud.tencent.com
X-TC-Action: DescribeInstances
X-TC-Version: 2017-03-12
X-TC-Timestamp: 1539084154
X-TC-Region: shjr
```

# 3. 签名失败

根据实际情况，存在以下签名失败的错误码，请根据实际情况处理

| 错误代码 | 错误描述 |
| --- | --- |
| AuthFailure.SignatureExpire | 签名过期 |
| AuthFailure.SecretIdNotFound | 密钥不存在 |
| AuthFailure.SignatureFailure | 签名错误 |
| AuthFailure.TokenFailure | token 错误 |
| AuthFailure.InvalidSecretId | 密钥非法（不是云 API 密钥类型） |

# 4. 签名演示

Java

```
import java.io.BufferedReader;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.URL;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.Map;
import java.util.TimeZone;
import java.util.TreeMap;
import javax.crypto.Mac;
import javax.crypto.spec.SecretKeySpec;
import javax.net.ssl.HttpsURLConnection;
import javax.xml.bind.DatatypeConverter;

import org.apache.commons.codec.digest.DigestUtils;

public class CloudAPITC3Demo {
    private final static String CHARSET = "UTF-8";
    private final static String ENDPOINT = "cvm.finance.cloud.tencent.com";
    private final static String PATH = "/";
    private final static String SECRET_ID = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE";
    private final static String SECRET_KEY = "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE";
    private final static String CT_X_WWW_FORM_URLENCODED = "application/x-www-form-urlencode
d";
    private final static String CT_JSON = "application/json";
```

```
    private final static String CT_FORM_DATA = "multipart/form-data";

    public static byte[] sign256(byte[] key, String msg) throws Exception {
        Mac mac = Mac.getInstance("HmacSHA256");
        SecretKeySpec secretKeySpec = new SecretKeySpec(key, mac.getAlgorithm());
        mac.init(secretKeySpec);
        return mac.doFinal(msg.getBytes(CHARSET));
    }

    public static void main(String[] args) throws Exception {
        String service = "cvm";
        String host = "cvm.finance.cloud.tencent.com";
        String region = "shjr";
        String action = "DescribeInstances";
        String version = "2017-03-12";
        String algorithm = "TC3-HMAC-SHA256";
        String timestamp = "1539084154";
        //String timestamp = String.valueOf(System.currentTimeMillis() / 1000);
        SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");
        // 注意时区，否则容易出错
        sdf.setTimeZone(TimeZone.getTimeZone("UTC"));
        String date = sdf.format(new Date(Long.valueOf(timestamp + "000")));

        // ************* 步骤 1：拼接规范请求串 *************
        String httpRequestMethod = "GET";
        String canonicalUri = "/";
        String canonicalQueryString = "Limit=10&Offset=0";
        String canonicalHeaders = "content-type:application/x-www-form-urlencoded\n" + "host:" + host
+ "\n";
        String signedHeaders = "content-type;host";
        String hashedRequestPayload = DigestUtils.sha256Hex("");
        String canonicalRequest = httpRequestMethod + "\n" + canonicalUri + "\n" + canonicalQueryStri
ng + "\n"
                + canonicalHeaders + "\n" + signedHeaders + "\n" + hashedRequestPayload;
        System.out.println(canonicalRequest);

        // ************* 步骤 2：拼接待签名字符串 *************
        String credentialScope = date + "/" + service + "/" + "tc3_request";
        String hashedCanonicalRequest = DigestUtils.sha256Hex(canonicalRequest.getBytes(CHARSET));
        String stringToSign = algorithm + "\n" + timestamp + "\n" + credentialScope + "\n" + hashedCan
onicalRequest;
        System.out.println(stringToSign);

        // ************* 步骤 3：计算签名 *************
        byte[] secretDate = sign256(("TC3" + SECRET_KEY).getBytes(CHARSET), date);
        byte[] secretService = sign256(secretDate, service);
        byte[] secretSigning = sign256(secretService, "tc3_request");
```

```
        String signature = DatatypeConverter.printHexBinary(sign256(secretSigning, stringToSign)).toLow
erCase();
        System.out.println(signature);

        // ************* 步骤 4：拼接 Authorization *************
        String authorization = algorithm + " " + "Credential=" + SECRET_ID + "/" + credentialScope + ", "
                + "SignedHeaders=" + signedHeaders + ", " + "Signature=" + signature;
        System.out.println(authorization);

        TreeMap<String, String> headers = new TreeMap<String, String>();
        headers.put("Authorization", authorization);
        headers.put("Host", host);
        headers.put("Content-Type", CT_X_WWW_FORM_URLENCODED);
        headers.put("X-TC-Action", action);
        headers.put("X-TC-Timestamp", timestamp);
        headers.put("X-TC-Version", version);
        headers.put("X-TC-Region", region);
    }
}
```

## Python

```
# -*- coding: utf-8 -*-
import hashlib, hmac, json, os, sys, time
from datetime import datetime

# 密钥参数
secret_id = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE"
secret_key = "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE"

service = "cvm"
host = "cvm.finance.cloud.tencent.com"
endpoint = "https://" + host
region = "shjr"
action = "DescribeInstances"
version = "2017-03-12"
algorithm = "TC3-HMAC-SHA256"
timestamp = 1539084154
date = datetime.utcfromtimestamp(timestamp).strftime("%Y-%m-%d")
params = {"Limit": 10, "Offset": 0}

# ************* 步骤 1：拼接规范请求串 *************
http_request_method = "GET"
canonical_uri = "/"
canonical_querystring = "Limit=10&Offset=0"
ct = "x-www-form-urlencoded"
```

```
payload = ""
if http_request_method == "POST":
    canonical_querystring = ""
    ct = "json"
    payload = json.dumps(params)
canonical_headers = "content-type:application/%s\nhost:%s\n" % (ct, host)
signed_headers = "content-type;host"
hashed_request_payload = hashlib.sha256(payload.encode("utf-8")).hexdigest()
canonical_request = (http_request_method + "\n" +
                     canonical_uri + "\n" +
                     canonical_querystring + "\n" +
                     canonical_headers + "\n" +
                     signed_headers + "\n" +
                     hashed_request_payload)
print(canonical_request)

# ************* 步骤 2：拼接待签名字符串 *************
credential_scope = date + "/" + service + "/" + "tc3_request"
hashed_canonical_request = hashlib.sha256(canonical_request.encode("utf-8")).hexdigest()
string_to_sign = (algorithm + "\n" +
                  str(timestamp) + "\n" +
                  credential_scope + "\n" +
                  hashed_canonical_request)
print(string_to_sign)

# ************* 步骤 3：计算签名 *************
# 计算签名摘要函数
def sign(key, msg):
    return hmac.new(key, msg.encode("utf-8"), hashlib.sha256).digest()
secret_date = sign(("TC3" + secret_key).encode("utf-8"), date)
secret_service = sign(secret_date, service)
secret_signing = sign(secret_service, "tc3_request")
signature = hmac.new(secret_signing, string_to_sign.encode("utf-8"), hashlib.sha256).hexdigest()
print(signature)

# ************* 步骤 4：拼接 Authorization *************
authorization = (algorithm + " " +
                 "Credential=" + secret_id + "/" + credential_scope + ", " +
                 "SignedHeaders=" + signed_headers + ", " +
                 "Signature=" + signature)
print(authorization)

# 公共参数添加到请求头部
headers = {
    "Authorization": authorization,
    "Host": host,
    "Content-Type": "application/%s" % ct,
```

```
    "X-TC-Action": action,
    "X-TC-Timestamp": str(timestamp),
    "X-TC-Version": version,
    "X-TC-Region": region,
}
```

# 请求结构

## 1. 服务地址

地域（Region）是指物理的数据中心的地理区域。TCloudFinanceZone交付验证不同地域之间完全隔离，保证不同地域间最大程度的稳定性和容错性。为了降低访问时延、提高下载速度，建议您选择最靠近您客户的地域。

您可以通过 API接口 查询地域列表 查看完成的地域列表。

## 2. 通信协议

TCloudFinanceZone API 的所有接口均通过 HTTPS 进行通信，提供高安全性的通信通道。

## 3. 请求方法

支持的 HTTP 请求方法:

- POST（推荐）
- GET

POST 请求支持的 Content-Type 类型：

- application/json（推荐），必须使用 TC3-HMAC-SHA256 签名方法。
- application/x-www-form-urlencoded，必须使用 HmacSHA1 或 HmacSHA256 签名方法。
- multipart/form-data（仅部分接口支持），必须使用 TC3-HMAC-SHA256 签名方法。

GET 请求的请求包大小不得超过 32 KB。POST 请求使用签名方法为 HmacSHA1、HmacSHA256 时不得超过 1 MB。POST 请求使用签名方法为 TC3-HMAC-SHA256 时支持 10 MB。

## 4. 字符编码

均使用UTF-8编码。

# 返回结果

## 正确返回结果

以云服务器的接口查看实例状态列表 (DescribeInstancesStatus) 2017-03-12 版本为例，若调用成功，其可能的返回如下为：

```
{
    "Response": {
        "TotalCount": 0,
        "InstanceStatusSet": [],
        "RequestId": "b5b41468-520d-4192-b42f-595cc34b6c1c"
    }
}
```

- Response 及其内部的 RequestId 是固定的字段，无论请求成功与否，只要 API 处理了，则必定会返回。
- RequestId 用于一个 API 请求的唯一标识，如果 API 出现异常，可以联系我们，并提供该 ID 来解决问题。
- 除了固定的字段外，其余均为具体接口定义的字段，不同的接口所返回的字段参见接口文档中的定义。此例中的 TotalCount 和 InstanceStatusSet 均为 DescribeInstancesStatus 接口定义的字段，由于调用请求的用户暂时还没有云服务器实例，因此 TotalCount 在此情况下的返回值为 0， InstanceStatusSet 列表为空。

## 错误返回结果

若调用失败，其返回值示例如下为：

```
{
    "Response": {
        "Error": {
            "Code": "AuthFailure.SignatureFailure",
            "Message": "The provided credentials could not be validated. Please check your signature is cor
rect."
        },
        "RequestId": "ed93f3cb-f35e-473f-b9f3-0d451b8b79c6"
    }
}
```

- Error 的出现代表着该请求调用失败。Error 字段连同其内部的 Code 和 Message 字段在调用失败时是必定返回的。
- Code 表示具体出错的错误码，当请求出错时可以先根据该错误码在公共错误码和当前接口对应的错误码列表里面查找对应原因和解决方案。

- Message 显示出了这个错误发生的具体原因，随着业务发展或体验优化，此文本可能会经常保持变更或更新，用户不应依赖这个返回值。
- RequestId 用于一个 API 请求的唯一标识，如果 API 出现异常，可以联系我们，并提供该 ID 来解决问题。

# 公共错误码

返回结果中如果存在 Error 字段，则表示调用 API 接口失败。 Error 中的 Code 字段表示错误码，所有业务都可能出现的错误码为公共错误码，下表列出了公共错误码。

| 错误码 | 错误描述 |
| --- | --- |
| AuthFailure.InvalidSecretId | 密钥非法（不是云 API 密钥类型）。 |
| AuthFailure.MFAFailure | MFA 错误。 |
| AuthFailure.SecretIdNotFound | 密钥不存在。 |
| AuthFailure.SignatureExpire | 签名过期。 |
| AuthFailure.SignatureFailure | 签名错误。 |
| AuthFailure.TokenFailure | token 错误。 |
| AuthFailure.UnauthorizedOperation | 请求未 CAM 授权。 |
| DryRunOperation | DryRun 操作，代表请求将会是成功的，只是多传了 DryRun 参数。 |
| FailedOperation | 操作失败。 |
| InternalError | 内部错误。 |
| InvalidAction | 接口不存在。 |
| InvalidParameter | 参数错误。 |
| InvalidParameterValue | 参数取值错误。 |
| LimitExceeded | 超过配额限制。 |
| MissingParameter | 缺少参数错误。 |
| NoSuchVersion | 接口版本不存在。 |
| RequestLimitExceeded | 请求的次数超过了频率限制。 |
| ResourceInUse | 资源被占用。 |
| ResourceInsufficient | 资源不足。 |

| 错误码 | 错误描述 |
| --- | --- |
| ResourceNotFound | 资源不存在。 |
| ResourceUnavailable | 资源不可用。 |
| UnauthorizedOperation | 未授权操作。 |
| UnknownParameter | 未知参数错误。 |
| UnsupportedOperation | 操作不支持。 |
| UnsupportedProtocol | http(s)请求协议错误，只支持 GET 和 POST 请求。 |
| UnsupportedRegion | 接口不支持所传地域。 |

# 公共参数

公共参数是用于标识用户和接口鉴权目的的参数，如非必要，在每个接口单独的接口文档中不再对这些参数进行说明，但每次请求均需要携带这些参数，才能正常发起请求。

## 签名方法 v3

使用 TC3-HMAC-SHA256 签名方法时，公共参数需要统一放到 HTTP Header 请求头部中，如下：

| 参数名称 | 类型 | 必选 | 描述 |
| --- | --- | --- | --- |
| X-TC-Action | String | 是 | 操作的接口名称。取值参考接口文档中输入参数公共参数 Action 的说明。例如云服务器的查询实例列表接口，取值为 DescribeInstances。 |
| X-TC-Region | String | 是 | 地域参数，用来标识希望操作哪个地域的数据。接口接受的地域取值参考接口文档中输入参数公共参数 Region 的说明。注意：某些接口不需要传递该参数，接口文档中会对此特别说明，此时即使传递该参数也不会生效。 |
| X-TC-Timestamp | Integer | 是 | 当前 UNIX 时间戳，可记录发起 API 请求的时间。例如 1529223702。注意：如果与服务器时间相差超过5分钟，会引起签名过期错误。 |
| X-TC-Version | String | 是 | 操作的 API 的版本。取值参考接口文档中入参公共参数 Version 的说明。例如云服务器的版本 2017-03-12。 |
| Authorization | String | 是 | HTTP 标准身份认证头部字段，例如：<br>TC3-HMAC-SHA256 Credential=AKIDEXAMPLE/Date/service/tc3_request, SignedHeaders=content-type;host, Signature=fe5f80f77d5fa3beca038a248ff027d0445342fe2855ddc963176630326f1024<br>其中，<br>- TC3-HMAC-SHA256：签名方法，目前固定取该值；<br>- Credential：签名凭证，AKIDEXAMPLE 是 SecretId；Date 是 UTC 标准时间的日期，取值需要和公共参数 X-TC-Timestamp 换算的 UTC 标准时间日期一致；service为产品名，必须与调用的产品域名一致，例如cvm；<br>- SignedHeaders：参与签名计算的头部信息，content-type 和 host 为必选头部；<br>- Signature：签名摘要。 |
| X-TC-Token | String | 否 | 临时证书所用的 Token ，需要结合临时密钥一起使用。临时密钥和 Token 需要到访问管理服务调用接口获取。长期密钥不需要 Token。 |

## 签名方法 v1

使用 HmacSHA1 和 HmacSHA256 签名方法时，公共参数需要统一放到请求串中，如下

| 参数名称 | 类型 | 必选 | 描述 |
| --- | --- | --- | --- |
| Action | String | 是 | 操作的接口名称。取值参考接口文档中输入参数公共参数 Action 的说明。例如云服务器的查询实例列表接口，取值为 DescribeInstances。 |
| Region | String | 是 | 地域参数，用来标识希望操作哪个地域的数据。接口接受的地域取值参考接口文档中输入参数公共参数 Region 的说明。注意：某些接口不需要传递该参数，接口文档中会对此特别说明，此时即使传递该参数也不会生效。 |

| 参数名称 | 类型 | 必选 | 描述 |
| --- | --- | --- | --- |
| Timestamp | Integer | 是 | 当前 UNIX 时间戳，可记录发起 API 请求的时间。例如1529223702，如果与当前时间相差过大，会引起签名过期错误。 |
| Nonce | Integer | 是 | 随机正整数，与 Timestamp 联合起来，用于防止重放攻击。 |
| SecretId | String | 是 | 在云API密钥上申请的标识身份的 SecretId，一个 SecretId 对应唯一的 SecretKey ，而 SecretKey 会用来生成请求签名 Signature。 |
| Signature | String | 是 | 请求签名，用来验证此次请求的合法性，需要用户根据实际的输入参数计算得出。具体计算方法参见接口鉴权文档。 |
| Version | String | 是 | 操作的 API 的版本。取值参考接口文档中入参公共参数 Version 的说明。例如云服务器的版本 2017-03-12。 |
| SignatureMethod | String | 否 | 签名方式，目前支持 HmacSHA256 和 HmacSHA1。只有指定此参数为 HmacSHA256 时，才使用 HmacSHA256 算法验证签名，其他情况均使用 HmacSHA1 验证签名。 |
| Token | String | 否 | 临时证书所用的 Token ，需要结合临时密钥一起使用。临时密钥和 Token 需要到访问管理服务调用接口获取。长期密钥不需要 Token 。 |

# 地域列表

地域（Region）是指物理的数据中心的地理区域。TCloudFinanceZone交付验证不同地域之间完全隔离，保证不同地域间最大程度的稳定性和容错性。为了降低访问时延、提高下载速度，建议您选择最靠近您客户的地域。

您可以通过 API接口 查询地域列表 查看完成的地域列表。

# CMQ管理相关接口
# 回溯cmq队列

## 1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

回溯cmq队列

默认接口请求频率限制：20次/秒。

接口更新时间：2021-09-15 16:59:19。

接口既验签名又鉴权。

## 2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

| 参数名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Action | 是 | 否 | String | 公共参数，本接口取值：RewindCmqQueue |
| Version | 是 | 否 | String | 公共参数，本接口取值：2020-02-17 |
| Region | 是 | 否 | String | 公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表 |
| QueueName | 是 | 否 | String | 队列名字，在单个地域同一帐号下唯一。队列名称是一个不超过64个字符的字符串，必须以字母为首字符，剩余部分可以包含字母、数字和横划线(-)。<br>示例值： test-queue |
| StartConsumeTime | 是 | 否 | Uint64 | 设定该时间，则（Batch）receiveMessage接口，会按照生产消息的先后顺序消费该时间戳以后的消息。<br>示例值： 1582108595000 |

## 3. 输出参数

| 参数名称 | 类型 | 描述 |
| --- | --- | --- |

| 参数名称 | 类型 | 描述 |
| --- | --- | --- |
| RequestId | String | 唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。 |

# 4. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见公共错误码。

# RocketMQ 主题相关接口
# 创建RocketMQ主题

## 1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

创建RocketMQ主题

默认接口请求频率限制：20次/秒。

接口更新时间：2025-11-18 22:21:43。

接口既验签名又鉴权。

## 2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

| 参数名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Action | 是 | 否 | String | 公共参数，本接口取值：CreateRocketMQTopic |
| Version | 是 | 否 | String | 公共参数，本接口取值：2020-02-17 |
| Region | 是 | 否 | String | 公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表 |
| Topic | 是 | 是 | String | 主题名称，3-64个字符，只能包含字母、数字、“-”及“_”<br>示例值： order-topic |
| Namespaces | 是 | 是 | Array of String | 主题所在的命名空间，目前支持在单个命名空间下创建主题<br>示例值： ["ns01"] |
| Type | 是 | 是 | String | 主题类型，可选值为<br>Normal, GlobalOrder, PartitionedOrder, Transaction, DelayScheduled。<br>Transaction仅在专享版支持。<br>示例值： Normal |
| ClusterId | 是 | 是 | String | 集群ID<br>示例值： rocketmq-4k4orqgq |
| Remark | 否 | 是 | String | 主题说明，最大128个字符<br>示例值： info-mark |
| PartitionNum | 否 | 是 | Int64 | 分区数，全局顺序无效<br>示例值： 1 |

## 3. 输出参数

| 参数名称 | 类型 | 描述 |
| --- | --- | --- |
| RequestId | String | 唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。 |

# 4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

| 错误码 | 描述 |
| --- | --- |
| FailedOperation.CreateTopic | |
| ResourceInUse | |
| InvalidParameter | |
| InternalError | |
| LimitExceeded | |
| ResourceNotFound | |
| FailedOperation | |
| ResourceNotFound.Cluster | |
| ResourceNotFound.Namespace | |
| InternalError.IllegalMessage | |
| ResourceUnavailable.FundRequired | |
| InvalidParameter.Partition | |

# 删除RocketMQ主题

## 1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

删除RocketMQ主题

默认接口请求频率限制：20次/秒。

接口更新时间：2025-11-18 22:21:43。

接口既验签名又鉴权。

## 2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

| 参数名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Action | 是 | 否 | String | 公共参数，本接口取值：DeleteRocketMQTopic |
| Version | 是 | 否 | String | 公共参数，本接口取值：2020-02-17 |
| Region | 是 | 否 | String | 公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表 |
| ClusterId | 是 | 是 | String | 集群ID<br>示例值： rocketmq-2p9vx3ax9jxg |
| NamespaceId | 是 | 是 | String | 命名空间名称<br>示例值： test_namespace |
| Topic | 是 | 是 | String | 主题名称<br>示例值： test_topic |

## 3. 输出参数

| 参数名称 | 类型 | 描述 |
| --- | --- | --- |
| RequestId | String | 唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。 |

# 4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

| 错误码 | 描述 |
| --- | --- |
| FailedOperation.DeleteTopics | |
| InvalidParameter | |
| InternalError | |
| ResourceNotFound | |
| FailedOperation | |

# 获取RocketMQ主题列表

## 1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

获取RocketMQ主题列表

默认接口请求频率限制：20次/秒。

接口更新时间：2025-11-18 22:21:43。

接口只验签名不鉴权。

## 2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

| 参数名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Action | 是 | 否 | String | 公共参数，本接口取值：DescribeRocketMQTopics |
| Version | 是 | 否 | String | 公共参数，本接口取值：2020-02-17 |
| Region | 是 | 否 | String | 公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表 |
| DataOwnerUin | 否 | 是 | Int64 | 查询指定主账号uin的资源<br>示例值： 7621238456 |
| TagFilters | 否 | 是 | Array of TagFilter | 标签过滤器<br>示例值：查看 |
| Limit | 是 | 是 | Uint64 | 查询限制数<br>示例值： 10 |
| ClusterId | 是 | 是 | String | 集群ID<br>示例值： rocketmq-4k4orqgq |
| NamespaceId | 是 | 是 | String | 命名空间<br>示例值： test_ns |
| FilterName | 否 | 是 | String | 按主题名称搜索，支持模糊查询<br>示例值： test_topic |

| 参数名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| FilterGroup | 否 | 是 | String | 按订阅消费组名称过滤<br>示例值： test_group |
| DataRegion | 否 | 是 | String | 查询指定region的资源<br>示例值： ap-beijing |
| Offset | 是 | 是 | Uint64 | 查询偏移量<br>示例值： 0 |
| FilterType | 否 | 是 | Array of String | 按主题类型过滤查询结果，可选择<br>Normal, GlobalOrder, PartitionedOrder, Transaction<br>示例值： ["Normal"] |

# 3. 输出参数

| 参数名称 | 类型 | 描述 |
| --- | --- | --- |
| TotalCount | Uint64 | 总记录数<br>示例值： 100 |
| Topics | Array of RocketMQTopic | 主题信息列表<br>示例值：查看 |
| RequestId | String | 唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。 |

# 4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

| 错误码 | 描述 |
| --- | --- |
| InvalidParameter | |
| InternalError | |
| ResourceUnavailable | |
| ResourceNotFound | |
| FailedOperation | |
| ResourceNotFound.BrokerCluster | |

| 错误码 | 描述 |
| --- | --- |
| ResourceNotFound.Cluster | |
| InternalError.IllegalMessage | |

# 更新RocketMQ主题信息

## 1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

更新RocketMQ主题信息

默认接口请求频率限制：20次/秒。

接口更新时间：2025-11-18 22:21:43。

接口既验签名又鉴权。

## 2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

| 参数名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Action | 是 | 否 | String | 公共参数，本接口取值：ModifyRocketMQTopic |
| Version | 是 | 否 | String | 公共参数，本接口取值：2020-02-17 |
| Region | 是 | 否 | String | 公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表 |
| PartitionNum | 否 | 是 | Int64 | 分区数，全局类型无效，不可小于当前分区数<br>示例值： 3 |
| ClusterId | 是 | 是 | String | 集群ID<br>示例值： rocketmq-2p9vx3ax9jxg |
| NamespaceId | 是 | 是 | String | 命名空间名称<br>示例值： test_namespace |
| Topic | 是 | 是 | String | 主题名称<br>示例值： test_topic |
| Remark | 是 | 是 | String | 说明信息，最大128个字符<br>示例值： 测试修改 |

## 3. 输出参数

| 参数名称 | 类型 | 描述 |
| --- | --- | --- |
| RequestId | String | 唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。 |

# 4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

| 错误码 | 描述 |
| --- | --- |
| ResourceInUse | |
| InvalidParameter | |
| InternalError | |
| ResourceNotFound | |
| FailedOperation | |
| InvalidParameter.Partition | |

# RocketMQ 命名空间相关接口
# 创建RocketMQ命名空间

## 1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

创建RocketMQ命名空间

默认接口请求频率限制：20次/秒。

接口更新时间：2025-11-18 22:21:43。

接口既验签名又鉴权。

## 2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

| 参数名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Action | 是 | 否 | String | 公共参数，本接口取值：CreateRocketMQNamespace |
| Version | 是 | 否 | String | 公共参数，本接口取值：2020-02-17 |
| Region | 是 | 否 | String | 公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表 |
| ClusterId | 是 | 是 | String | 集群ID<br>示例值： rocketmq-4k4orqgq |
| NamespaceId | 是 | 是 | String | 命名空间名称，3-64个字符，只能包含字母、数字、“-”及“_”<br>示例值： test_ns |
| Ttl | 否 | 是 | Uint64 | 已废弃<br>示例值： 259200000 |
| RetentionTime | 否 | 是 | Uint64 | 已废弃<br>示例值： 259200000 |
| Remark | 否 | 是 | String | 说明，最大128个字符<br>示例值： test remark info. |

# 3. 输出参数

| 参数名称 | 类型 | 描述 |
| --- | --- | --- |
| RequestId | String | 唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。 |

# 4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

| 错误码 | 描述 |
| --- | --- |
| ResourceInUse | |
| InvalidParameter | |
| InternalError | |
| ResourceInsufficient | |
| LimitExceeded | |
| ResourceNotFound | |
| FailedOperation | |
| ResourceNotFound.Cluster | |
| FailedOperation.SetTTL | |
| InternalError.IllegalMessage | |
| ResourceUnavailable.FundRequired | |
| FailedOperation.InstanceNotReady | |

# 删除RocketMQ命名空间

## 1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

删除RocketMQ命名空间

默认接口请求频率限制：20次/秒。

接口更新时间：2025-11-18 22:21:43。

接口既验签名又鉴权。

## 2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

| 参数名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Action | 是 | 否 | String | 公共参数，本接口取值：DeleteRocketMQNamespace |
| Version | 是 | 否 | String | 公共参数，本接口取值：2020-02-17 |
| Region | 是 | 否 | String | 公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表 |
| ClusterId | 是 | 是 | String | 集群ID<br>示例值： rocketmq-4k4orqgq |
| NamespaceId | 是 | 是 | String | 命名空间名称<br>示例值： prod |

## 3. 输出参数

| 参数名称 | 类型 | 描述 |
| --- | --- | --- |
| RequestId | String | 唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。 |

## 4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

| 错误码 | 描述 |
| --- | --- |
| InvalidParameter | |
| InternalError | |
| ResourceNotFound | |
| FailedOperation | |
| FailedOperation.RoleInUse | |

# 获取RocketMQ命名空间列表

## 1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

获取RocketMQ命名空间列表

默认接口请求频率限制：20次/秒。

接口更新时间：2025-11-18 22:21:43。

接口只验签名不鉴权。

## 2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

| 参数名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Action | 是 | 否 | String | 公共参数，本接口取值：DescribeRocketMQNamespaces |
| Version | 是 | 否 | String | 公共参数，本接口取值：2020-02-17 |
| Region | 是 | 否 | String | 公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表 |
| Offset | 是 | 是 | Uint64 | 偏移量<br>示例值： 0 |
| Limit | 是 | 是 | Uint64 | 限制数目<br>示例值： 10 |
| NameKeyword | 否 | 是 | String | 按名称搜索<br>示例值： ordername |
| DataRegion | 否 | 是 | String | 查询指定region的资源<br>示例值： ap-beijing |
| DataOwnerUin | 否 | 是 | Int64 | 查询指定主账号uin的资源<br>示例值： 7261234356 |
| ClusterId | 是 | 是 | String | 集群ID<br>示例值： rocketmq-4k4orqgq |

# 3. 输出参数

| 参数名称 | 类型 | 描述 |
| --- | --- | --- |
| Namespaces | Array of RocketMQNamespace | 命名空间列表<br>示例值：查看 |
| TotalCount | Uint64 | 总条数<br>示例值： 100 |
| RequestId | String | 唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。 |

# 4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

| 错误码 | 描述 |
| --- | --- |
| InvalidParameter | |
| InternalError | |
| ResourceUnavailable | |
| ResourceNotFound | |
| FailedOperation | |
| ResourceNotFound.Cluster | |
| InternalError.IllegalMessage | |

# 更新RocketMQ命名空间

## 1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

更新RocketMQ命名空间

默认接口请求频率限制：20次/秒。

接口更新时间：2025-11-18 22:21:43。

接口既验签名又鉴权。

## 2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

| 参数名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Action | 是 | 否 | String | 公共参数，本接口取值：ModifyRocketMQNamespace |
| Version | 是 | 否 | String | 公共参数，本接口取值：2020-02-17 |
| Region | 是 | 否 | String | 公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表 |
| PublicAccessEnabled | 否 | 是 | Bool | 是否开启公网访问<br>示例值： false |
| ClusterId | 是 | 是 | String | 集群ID<br>示例值： rocketmq-4k4orqgq |
| NamespaceId | 是 | 是 | String | 命名空间名称，3-64个字符，只能包含字母、数字、“-”及“_”<br>示例值： prod |
| Ttl | 否 | 是 | Uint64 | 已废弃<br>示例值： 100000 |
| RetentionTime | 否 | 是 | Uint64 | 已废弃<br>示例值： 100000 |

| 参数名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Remark | 否 | 是 | String | 说明，最大128个字符<br>示例值： info-remark |

# 3. 输出参数

| 参数名称 | 类型 | 描述 |
| --- | --- | --- |
| RequestId | String | 唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。 |

# 4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

| 错误码 | 描述 |
| --- | --- |
| ResourceInUse | |
| InvalidParameter | |
| InternalError | |
| ResourceInsufficient | |
| LimitExceeded | |
| ResourceNotFound | |
| FailedOperation | |
| ResourceNotFound.Cluster | |
| FailedOperation.SetTTL | |
| ResourceUnavailable.FundRequired | |

# RocketMQ 平滑迁移相关接口
# 平滑迁移：导入消费者组列表

## 1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

输入迁移任务id和要导入的Group，导入后台

默认接口请求频率限制：10000次/秒。

接口更新时间：2025-11-18 22:21:43。

接口既验签名又鉴权。

## 2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

| 参数名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Action | 是 | 否 | String | 公共参数，本接口取值：ImportRocketMQConsumerGroups |
| Version | 是 | 否 | String | 公共参数，本接口取值：2020-02-17 |
| Region | 是 | 否 | String | 公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表 |
| Groups | 是 | 是 | Array of RocketMQGroupConfig | 导入topic<br>示例值：查看 |
| TaskId | 是 | 是 | String | 任务id<br>示例值： 20157405-8691-4972-b7ec-52fd0839b7a9t |

## 3. 输出参数

| 参数名称 | 类型 | 描述 |
| --- | --- | --- |

| 参数名称 | 类型 | 描述 |
| --- | --- | --- |
| RequestId | String | 唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。 |

# 4. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见公共错误码。

# 平滑迁移：导入topic列表

## 1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

导入topic列表

默认接口请求频率限制：10000次/秒。

接口更新时间：2025-11-18 22:21:43。

接口既验签名又鉴权。

## 2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

| 参数名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Action | 是 | 否 | String | 公共参数，本接口取值：ImportRocketMQTopics |
| Version | 是 | 否 | String | 公共参数，本接口取值：2020-02-17 |
| Region | 是 | 否 | String | 公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表 |
| Topics | 是 | 是 | Array of RocketMQTopicConfig | 导入topic<br>示例值：查看 |
| TaskId | 是 | 是 | String | 任务ID<br>示例值： a50e97f6-8d2b-4fa8-b1f8-ba47bc50780e |

## 3. 输出参数

| 参数名称 | 类型 | 描述 |
| --- | --- | --- |
| RequestId | String | 唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。 |

# 4. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见公共错误码。

# RocketMQ 消息相关接口
# rocketmq消息详情

## 1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

rocketmq消息详情

默认接口请求频率限制：20次/秒。

接口更新时间：2025-11-18 22:21:43。

接口既验签名又鉴权。

## 2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

| 参数名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Action | 是 | 否 | String | 公共参数，本接口取值：DescribeRocketMQMsg |
| Version | 是 | 否 | String | 公共参数，本接口取值：2020-02-17 |
| Region | 是 | 否 | String | 公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表 |
| EnvironmentId | 是 | 是 | String | 命名空间<br>示例值： test_ns |
| MsgId | 是 | 是 | String | 消息id<br>示例值：092BAE5A1656070DEA4E276DF0760089 |
| QueryDlqMsg | 否 | 是 | Bool | 查询死信时该值为true，只对Rocketmq有效<br>示例值： false |
| TopicName | 是 | 是 | String | 主题，查询死信时传groupId<br>示例值： test_topic |

| 参数名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| PulsarMsgId | 是 | 是 | String | pulsar消息id<br>示例值：<br>092BAE5A1656070DEA4E276DF0760089 |
| QueryDeadLetterMessage | 否 | 是 | Bool | 查询死信时该值为true，只对Rocketmq有效<br>示例值： false |
| Offset | 否 | 是 | Int64 | 分页Offset<br>示例值： 0 |
| Limit | 否 | 是 | Int64 | 分页Limit<br>示例值： 20 |
| FilterTrackGroup | 否 | 是 | String | 根据消费组名称过滤消费详情<br>示例值： test_group |
| ClusterId | 是 | 是 | String | 集群id<br>示例值： rocketmq-4k4orqgq |

# 3. 输出参数

| 参数名称 | 类型 | 描述 |
| --- | --- | --- |
| MessageTracksCount | Int64 | 消费组消费情况列表总数<br>示例值： 100 |
| Body | String | 消息体<br>示例值： order-id-2ksn3b23 |
| Properties | String | 详情参数<br>示例值： {"type":"normal"} |
| ProduceTime | String | 生产时间<br>示例值： 2021-11-23 21:21:26 |
| MsgId | String | 消息id<br>示例值：<br>092BAE5A1656070DEA4E276DF0760089 |
| ProducerAddr | String | 生产者地址<br>示例值： 9.43.174.90 |
| MessageTracks | Array of RocketMQMessageTrack | 消费组消费情况列表<br>示例值：查看 |

| 参数名称 | 类型 | 描述 |
| --- | --- | --- |
| ShowTopicName | String | 详情页展示的topic名称<br>示例值： test_topic |
| RequestId | String | 唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。 |

# 4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

| 错误码 | 描述 |
| --- | --- |
| FailedOperation | |
| ResourceNotFound.Cluster | |

# 查询RocketMQ消息轨迹

## 1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

查询消息轨迹

默认接口请求频率限制：10000次/秒。

接口更新时间：2025-11-18 22:21:43。

接口既验签名又鉴权。

## 2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

| 参数名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Action | 是 | 否 | String | 公共参数，本接口取值：DescribeRocketMQMsgTrace |
| Version | 是 | 否 | String | 公共参数，本接口取值：2020-02-17 |
| Region | 是 | 否 | String | 公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表 |
| ClusterId | 是 | 是 | String | 集群id<br>示例值： rocketmq-4k4orqgq |
| EnvironmentId | 是 | 是 | String | 命名空间<br>示例值： test_ns |
| TopicName | 是 | 是 | String | 主题，rocketmq查询死信时值为groupId<br>示例值： test_topic |
| MsgId | 是 | 是 | String | 消息id<br>示例值：092BAE5A1656070DEA4E276DF0760089 |
| GroupName | 否 | 是 | String | 消费组、订阅<br>示例值： test_group |

| 参数名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| QueryDLQMsg | 否 | 是 | Bool | 查询死信时该值为true<br>示例值： false |
| QueryDeadLetterMessage | 否 | 是 | String | 查询死信时该值为true<br>示例值： false |

# 3. 输出参数

| 参数名称 | 类型 | 描述 |
| --- | --- | --- |
| Result | Array of TraceResult | 轨迹详情列表<br>示例值：查看 |
| ShowTopicName | String | 消息轨迹页展示的topic名称<br>示例值： test_topic |
| RequestId | String | 唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。 |

# 4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

| 错误码 | 描述 |
| --- | --- |
| FailedOperation | |
| ResourceNotFound.Cluster | |

# rocketmq 消息查询

## 1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

rocketmq 消息查询

默认接口请求频率限制：20次/秒。

接口更新时间：2025-11-18 22:21:43。

接口既验签名又鉴权。

## 2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

| 参数名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Action | 是 | 否 | String | 公共参数，本接口取值：DescribeRocketMQTopicMsgs |
| Version | 是 | 否 | String | 公共参数，本接口取值：2020-02-17 |
| Region | 是 | 否 | String | 公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表 |
| QueryDlqMsg | 否 | 是 | Bool | 死信查询时该值为true，只对Rocketmq有效<br>示例值： false |
| NumOfLatestMsg | 否 | 是 | Int64 | 查询最近N条消息 最大不超过1024，默认-1为其他查询条件<br>示例值： 20 |
| Tag | 否 | 是 | String | TAG表达式<br>示例值： test_tag |
| QueryDeadLetterMessage | 否 | 是 | Bool | 死信查询时该值为true，只对Rocketmq有效<br>示例值： false |
| ClusterId | 是 | 是 | String | 集群 ID<br>示例值： rocketmq-4k4orqgq |

| 参数名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| EnvironmentId | 是 | 是 | String | 命名空间<br>示例值： test_namespace |
| TopicName | 是 | 是 | String | 主题名称，查询死信时为groupId<br>示例值： test_topic |
| StartTime | 是 | 是 | String | 开始时间<br>示例值： 2021-12-22 12:22:22 |
| EndTime | 是 | 是 | String | 结束时间<br>示例值： 2021-12-22 12:22:22 |
| Limit | 否 | 是 | Uint64 | 查询限额<br>示例值： 20 |
| TaskRequestId | 否 | 是 | String | 标志一次分页事务<br>示例值： 2e5ecbb3-f2d6-49d3-9c82-8648f8ed8748 |
| MsgId | 否 | 是 | String | 消息 ID<br>示例值：092BAE5A1656070DEA4E276DF0760089 |
| MsgKey | 否 | 是 | String | 消息 key<br>示例值： test_key |
| Offset | 否 | 是 | Uint64 | 查询偏移<br>示例值： 0 |

# 3. 输出参数

| 参数名称 | 类型 | 描述 |
| --- | --- | --- |
| TotalCount | Uint64 | 总数<br>示例值： 20 |
| TopicMsgLogSets | Array of RocketMQMsgLog | 消息列表<br>示例值：查看 |
| TaskRequestId | String | 标志一次分页事务<br>示例值： 2e5ecbb3-f2d6-49d3-9c82-8648f8ed8748 |
| RequestId | String | 唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。 |

# 4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

| 错误码 | 描述 |
| --- | --- |
| FailedOperation | |
| ResourceNotFound.BrokerCluster | |
| ResourceNotFound.Cluster | |

# 导出RocketMQ消息详情

## 1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

导出RocketMQ消息详情

默认接口请求频率限制：10000次/秒。

接口更新时间：2025-11-18 22:21:43。

接口既验签名又鉴权。

## 2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

| 参数名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Action | 是 | 否 | String | 公共参数，本接口取值：ExportRocketMQMessageDetail |
| Version | 是 | 否 | String | 公共参数，本接口取值：2020-02-17 |
| Region | 是 | 否 | String | 公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表 |
| DeadLetterMsg | 否 | 是 | Bool | 是否死信消息<br>示例值： false |
| ClusterId | 是 | 是 | String | 集群id<br>示例值： rocketmq-a5nzx52ab7jb |
| EnvironmentId | 是 | 是 | String | 应用命名空间<br>示例值： test_ns |
| TopicName | 是 | 是 | String | Topic名称<br>如果是死信消息 isDlqMsg=true<br>示例值： test_topic |
| MsgId | 是 | 是 | String | 消息id<br>示例值： 7F0000013AEC18B4AAC278B640D30002 |

| 参数名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| IncludeMsgBody | 是 | 是 | Bool | 是否包含消息体<br>示例值： false |

## 3. 输出参数

| 参数名称 | 类型 | 描述 |
| --- | --- | --- |
| ReConsumeTimes | Uint64 | 消息重试次数<br>示例值： 0 |
| MsgBody | String | Base64编码格式字符串<br>示例值： This is a normal message for Apache RocketMQ |
| MsgBodyCRC | Int64 | 消息内容的CRC32 Code<br>示例值： 0 |
| MsgBodySize | Uint64 | 消息体大小（单位K）<br>当大于2048时不返回消息<br>示例值： 1024 |
| MsgId | String | 消息id<br>示例值： 7F0000013AEC18B4AAC278B640D30002 |
| BornTimestamp | Int64 | 消息生成时间戳<br>示例值： 1729478721 |
| Properties | String | 消息属性<br>示例值： {"orderId":"1002328497239845"} |
| StoreTimestamp | Int64 | 消息存储时间戳<br>示例值： 1729478721 |
| BornHost | String | 消息生产客户端地址<br>示例值： 10.2.139.175:8080 |
| MsgTag | String | 消息Tag<br>示例值： test_tag |
| MsgKey | String | 消息Key<br>示例值： test_key |
| RequestId | String | 唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。 |

# 4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

| 错误码 | 描述 |
| --- | --- |
| FailedOperation | |
| ResourceNotFound.Cluster | |
| InternalError.FailedOperation | |

# 重置RocketMQ消费位点

## 1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

重置指定Group的消费位点到指定时间戳

默认接口请求频率限制：20次/秒。

接口更新时间：2025-11-18 22:21:43。

接口既验签名又鉴权。

## 2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

| 参数名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Action | 是 | 否 | String | 公共参数，本接口取值：ResetRocketMQConsumerOffSet |
| Version | 是 | 否 | String | 公共参数，本接口取值：2020-02-17 |
| Region | 是 | 否 | String | 公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表 |
| RetryFlag | 否 | 是 | Bool | 重置的是否是retry topic<br>示例值： false |
| ClusterId | 是 | 是 | String | 集群ID<br>示例值： rocketmq-2p9vx3ax9jxg |
| NamespaceId | 是 | 是 | String | 命名空间名称<br>示例值： test_namespace |
| GroupId | 是 | 是 | String | 消费组名称<br>示例值： test_group |
| Type | 是 | 是 | Uint64 | 重置方式，0表示从最新位点开始，1表示从指定时间点开始<br>示例值： 1 |

| 参数名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Topic | 否 | 是 | String | 主题名称<br>示例值： test_topic |
| ResetTimestamp | 否 | 是 | Uint64 | 重置指定的时间戳，仅在 Type 为1是生效，以毫秒为单位<br>示例值： 1697945990000 |

# 3. 输出参数

| 参数名称 | 类型 | 描述 |
| --- | --- | --- |
| RequestId | String | 唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。 |

# 4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

| 错误码 | 描述 |
| --- | --- |
| ResourceInUse | |
| InvalidParameter | |
| InternalError | |
| ResourceNotFound | |
| FailedOperation | |

# 发送RocketMQ消息

## 1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

发送 RocketMQ 消息，该接口仅用于控制台发送少量测试消息，不保证SLA，且云 API 存在限流，在真实业务场景下，请使用 RocketMQ SDK 发送消息。

默认接口请求频率限制：20次/秒。

接口更新时间：2025-11-18 22:21:43。

接口既验签名又鉴权。

## 2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

| 参数名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Action | 是 | 否 | String | 公共参数，本接口取值：SendRocketMQMessage |
| Version | 是 | 否 | String | 公共参数，本接口取值：2020-02-17 |
| Region | 是 | 否 | String | 公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表 |
| MsgKey | 否 | 是 | String | 消息key信息<br>示例值： test_key |
| MsgTag | 否 | 是 | String | 消息tag信息<br>示例值： test_tag |
| ClusterId | 是 | 是 | String | 集群id<br>示例值： rocketmq-4k4orqgq |
| NamespaceId | 是 | 是 | String | 命名空间<br>示例值： test_namespace |
| TopicName | 是 | 是 | String | topic名称<br>示例值： test_topic |
| MsgBody | 是 | 是 | String | 信息内容<br>示例值： |

| 参数名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| | | | | This is a normal message for Apache RocketMQ |

# 3. 输出参数

| 参数名称 | 类型 | 描述 |
| --- | --- | --- |
| Result | Bool | 发送结果<br>示例值： true |
| MsgId | String | 消息ID<br>示例值： 7F0000013AEC18B4AAC278B640D30002 |
| RequestId | String | 唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。 |

# 4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

| 错误码 | 描述 |
| --- | --- |
| ResourceNotFound.Topic | |
| FailedOperation | |
| ResourceNotFound.Cluster | |
| ResourceUnavailable.FundRequired | |

# Rocketmq消费验证

## 1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

Rocketmq消费验证

默认接口请求频率限制：20次/秒。

接口更新时间：2025-11-18 22:21:43。

接口既验签名又鉴权。

## 2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

| 参数名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Action | 是 | 否 | String | 公共参数，本接口取值：VerifyRocketMQConsume |
| Version | 是 | 否 | String | 公共参数，本接口取值：2020-02-17 |
| Region | 是 | 否 | String | 公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表 |
| ClientId | 是 | 是 | String | 客户端ID<br>示例值： 127.0.0.1@25760#13749780392400 |
| TopicName | 是 | 是 | String | 主题名称<br>示例值： test_topic |
| ClusterId | 是 | 是 | String | 集群id<br>示例值： rocketmq-4k4orqgq |
| NamespaceId | 是 | 是 | String | 命名空间<br>示例值： test_namespace |
| GroupId | 是 | 是 | String | 消费组ID<br>示例值： test_group |
| MsgId | 是 | 是 | String | 消息id<br>示例值： 092BAE5A1656070DEA4E276DF0760089 |

# 3. 输出参数

| 参数名称 | 类型 | 描述 |
| --- | --- | --- |
| RequestId | String | 唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。 |

# 4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

| 错误码 | 描述 |
| --- | --- |
| FailedOperation | |
| ResourceNotFound.Cluster | |

# RocketMQ 消费组相关接口
# 创建RocketMQ消费组

## 1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

创建RocketMQ消费组

默认接口请求频率限制：20次/秒。

接口更新时间：2025-11-18 22:21:43。

接口既验签名又鉴权。

## 2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

| 参数名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Action | 是 | 否 | String | 公共参数，本接口取值：CreateRocketMQGroup |
| Version | 是 | 否 | String | 公共参数，本接口取值：2020-02-17 |
| Region | 是 | 否 | String | 公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表 |
| RetryMaxTimes | 否 | 是 | Uint64 | Group最大重试次数<br>示例值： 16 |
| GroupId | 是 | 是 | String | Group名称，8~64个字符<br>示例值： test_group |
| Namespaces | 是 | 是 | Array of String | 命名空间，目前只支持单个命名空间<br>示例值： ["test_namespace"] |
| ReadEnable | 是 | 是 | Bool | 是否开启消费<br>示例值： true |
| BroadcastEnable | 是 | 是 | Bool | 是否开启广播消费<br>示例值： true |

| 参数名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| ClusterId | 是 | 是 | String | 集群ID<br>示例值： rocketmq-4k4orqgq |
| Remark | 否 | 是 | String | 说明信息，最长128个字符<br>示例值： 测试创建 |
| GroupType | 否 | 是 | String | Group类型（TCP/HTTP）<br>示例值： TCP |

# 3. 输出参数

| 参数名称 | 类型 | 描述 |
| --- | --- | --- |
| RequestId | String | 唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。 |

# 4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

| 错误码 | 描述 |
| --- | --- |
| ResourceInUse | |
| InvalidParameter | |
| InternalError | |
| LimitExceeded | |
| FailedOperation | |
| ResourceNotFound.Cluster | |
| InternalError.IllegalMessage | |

# 删除RocketMQ消费组

## 1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

删除RocketMQ消费组

默认接口请求频率限制：20次/秒。

接口更新时间：2025-11-18 22:21:43。

接口既验签名又鉴权。

## 2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

| 参数名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Action | 是 | 否 | String | 公共参数，本接口取值：DeleteRocketMQGroup |
| Version | 是 | 否 | String | 公共参数，本接口取值：2020-02-17 |
| Region | 是 | 否 | String | 公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表 |
| ClusterId | 是 | 是 | String | 集群ID<br>示例值： rocketmq-2p9vx3ax9jxg |
| NamespaceId | 是 | 是 | String | 命名空间名称<br>示例值： test_namespace |
| GroupId | 是 | 是 | String | 消费组名称<br>示例值： test_group |

## 3. 输出参数

| 参数名称 | 类型 | 描述 |
| --- | --- | --- |
| RequestId | String | 唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。 |

# 4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

| 错误码 | 描述 |
| --- | --- |
| ResourceInUse | |
| InvalidParameter | |
| InternalError | |
| ResourceNotFound | |
| FailedOperation | |
| ResourceNotFound.Cluster | |
| InternalError.IllegalMessage | |

# 获取在线消费端详情

## 1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

获取在线消费端详情

默认接口请求频率限制：20次/秒。

接口更新时间：2025-11-18 22:21:43。

接口既验签名又鉴权。

## 2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

| 参数名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Action | 是 | 否 | String | 公共参数，本接口取值：DescribeRocketMQConsumerConnectionDetail |
| Version | 是 | 否 | String | 公共参数，本接口取值：2020-02-17 |
| Region | 是 | 否 | String | 公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表 |
| NamespaceId | 是 | 是 | String | 命名空间名称<br>示例值： test_ns |
| GroupId | 是 | 是 | String | 消费组名称<br>示例值： test_group |
| ClientId | 是 | 是 | String | 消费端实例ID<br>示例值： local@23645@0@30ey5aef8w |
| Offset | 是 | 是 | Uint64 | 偏移量<br>示例值： 0 |
| Limit | 是 | 是 | Uint64 | 限制数目<br>示例值： 20 |
| FilterType | 否 | 是 | Array of String | 按主题类型过滤查询结果，可选择<br>Normal, GlobalOrder, PartitionedOrder, Retry, Transaction, DeadLetter<br>示例值： ["Normal"] |
| ClusterId | 是 | 是 | String | 集群ID<br>示例值： rocketmq-2p9vx3ax9jxg |

## 3. 输出参数

| 参数名称 | 类型 | 描述 |
| --- | --- | --- |
| TotalCount | Uint64 | 总条数<br>示例值： 100 |
| Details | Array of RocketMQConsumerTopic | 消费端主题信息列表<br>示例值：查看 |
| RequestId | String | 唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。 |

# 4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

| 错误码 | 描述 |
| --- | --- |
| InvalidParameter | |
| InternalError | |
| ResourceNotFound | |
| FailedOperation | |

# 获取指定消费组下当前客户端的连接情况

## 1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

获取指定消费组下当前客户端的连接情况

默认接口请求频率限制：20次/秒。

接口更新时间：2025-11-18 22:21:43。

接口既验签名又鉴权。

## 2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

| 参数名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Action | 是 | 否 | String | 公共参数，本接口取值：DescribeRocketMQConsumerConnections |
| Version | 是 | 否 | String | 公共参数，本接口取值：2020-02-17 |
| Region | 是 | 否 | String | 公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表 |
| SortOrder | 否 | 是 | String | 查询结果排序规则，ASC为升序，DESC为降序<br>示例值： ASC |
| ClusterId | 是 | 是 | String | 集群ID<br>示例值： rocketmq-2p9vx3ax9jxg |
| NamespaceId | 是 | 是 | String | 命名空间名称<br>示例值： test_ns |
| GroupId | 是 | 是 | String | 消费组ID<br>示例值： test_group |
| Offset | 是 | 是 | Uint64 | 偏移量<br>示例值： 0 |
| Limit | 是 | 是 | Uint64 | 限制数目<br>示例值： 20 |

| 参数名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| SortedBy | 否 | 是 | String | 对查询结果排序，此为排序字段，目前支持Accumulative（消息堆积量）<br>示例值： Accumulative |

## 3. 输出参数

| 参数名称 | 类型 | 描述 |
| --- | --- | --- |
| GroupDetail | RocketMQGroup | 订阅组信息<br>示例值：查看 |
| TotalCount | Uint64 | 总数目<br>示例值： 100 |
| Connections | Array of RocketMQConsumerConnection | 在线消费者信息<br>示例值：查看 |
| RequestId | String | 唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。 |

## 4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

| 错误码 | 描述 |
| --- | --- |
| InvalidParameter | |
| InternalError | |
| ResourceUnavailable | |
| ResourceNotFound | |
| FailedOperation | |
| ResourceNotFound.Cluster | |

# 获取RocketMQ消费组列表

## 1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

获取RocketMQ消费组列表

默认接口请求频率限制：20次/秒。

接口更新时间：2025-11-18 22:21:43。

接口只验签名不鉴权。

## 2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

| 参数名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Action | 是 | 否 | String | 公共参数，本接口取值：DescribeRocketMQGroups |
| Version | 是 | 否 | String | 公共参数，本接口取值：2020-02-17 |
| Region | 是 | 否 | String | 公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表 |
| TagFilters | 否 | 是 | Array of TagFilter | 标签过滤器<br>示例值：查看 |
| NamespaceId | 是 | 是 | String | 命名空间<br>示例值： test_ns |
| Offset | 是 | 是 | Uint64 | 偏移量<br>示例值： 0 |
| Limit | 是 | 是 | Uint64 | 限制条数<br>示例值： 20 |
| FilterGroup | 否 | 是 | String | 按消费组名称查询消费组，支持模糊查询<br>示例值： test_group |
| FilterOneGroup | 否 | 是 | String | 订阅组名称，指定此参数后将只返回该订阅组信息<br>示例值： test_group |

| 参数名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Types | 否 | 是 | Array of String | group类型<br>示例值： ["TCP"] |
| ClusterId | 是 | 是 | String | 集群ID<br>示例值： rocketmq-4k4orqgq |
| FilterTopic | 否 | 是 | String | 主题名称，输入此参数可查询该主题下所有的订阅组<br>示例值： test_topic |
| SortedBy | 否 | 是 | String | 按照指定字段排序，可选值为 subscribeNum: 订阅 Topic 个数<br>示例值： subscribeNum |
| SortOrder | 否 | 是 | String | 按升序或降序排列，可选值为asc，desc<br>示例值： asc |

# 3. 输出参数

| 参数名称 | 类型 | 描述 |
| --- | --- | --- |
| TotalCount | Uint64 | 总数量<br>示例值： 100 |
| Groups | Array of RocketMQGroup | 订阅组列表<br>示例值：查看 |
| RequestId | String | 唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。 |

# 4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

| 错误码 | 描述 |
| --- | --- |
| ResourceInUse | |
| InvalidParameter | |
| InternalError | |

| 错误码 | 描述 |
| --- | --- |
| ResourceUnavailable | |
| ResourceNotFound | |
| FailedOperation | |
| ResourceNotFound.Cluster | |

# 获取RocketMQ消费组订阅关系

## 1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

用于获取RocketMQ消费组订阅关系数据

默认接口请求频率限制：20次/秒。

接口更新时间：2025-11-18 22:21:43。

接口既验签名又鉴权。

## 2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

| 参数名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Action | 是 | 否 | String | 公共参数，本接口取值：DescribeRocketMQSubscriptions |
| Version | 是 | 否 | String | 公共参数，本接口取值：2020-02-17 |
| Region | 是 | 否 | String | 公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表 |
| Group | 是 | 是 | String | 消费组名称<br>示例值： test_group |
| Offset | 是 | 是 | Int64 | 查询起始位置<br>示例值： 0 |
| Limit | 是 | 是 | Int64 | 查询限制条数<br>示例值： 20 |
| ClusterId | 是 | 是 | String | 集群ID<br>示例值： rocketmq-4k4orqgq |
| Namespace | 是 | 是 | String | 命名空间名称<br>示例值： test_namespace |

## 3. 输出参数

| 参数名称 | 类型 | 描述 |
| --- | --- | --- |
| TotalCount | Int64 | 总条数<br>示例值： 10 |
| Subscriptions | Array of RocketMQSubscription | 订阅关系列表<br>示例值：查看 |
| RequestId | String | 唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。 |

# 4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

| 错误码 | 描述 |
| --- | --- |
| ResourceUnavailable | |
| FailedOperation | |
| ResourceNotFound.Cluster | |

# 获取指定消费组下订阅的主题列表

## 1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

获取指定消费组下订阅的主题列表

默认接口请求频率限制：20次/秒。

接口更新时间：2025-11-18 22:21:43。

接口只验签名不鉴权。

## 2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

| 参数名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Action | 是 | 否 | String | 公共参数，本接口取值：DescribeRocketMQTopicsByGroup |
| Version | 是 | 否 | String | 公共参数，本接口取值：2020-02-17 |
| Region | 是 | 否 | String | 公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表 |
| ClusterId | 是 | 是 | String | 集群ID<br>示例值： rocketmq-4k4orqgq |
| NamespaceId | 是 | 是 | String | 命名空间名称<br>示例值： test_ns |
| GroupId | 是 | 是 | String | 消费组名称<br>示例值： test_group |
| Offset | 否 | 是 | Uint64 | 偏移量<br>示例值： 0 |
| Limit | 否 | 是 | Uint64 | 限制条数<br>示例值： 10 |

# 3. 输出参数

| 参数名称 | 类型 | 描述 |
| --- | --- | --- |
| TotalCount | Uint64 | 总条数<br>示例值： 100 |
| Topics | Array of String | 主题列表<br>示例值： ["test_topic"] |
| RequestId | String | 唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。 |

# 4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

| 错误码 | 描述 |
| --- | --- |
| InvalidParameter | |
| InternalError | |
| ResourceNotFound | |
| FailedOperation | |
| ResourceNotFound.Cluster | |

# 更新RocketMQ消费组信息

## 1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

更新RocketMQ消费组信息

默认接口请求频率限制：20次/秒。

接口更新时间：2025-11-18 22:21:43。

接口既验签名又鉴权。

## 2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

| 参数名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Action | 是 | 否 | String | 公共参数，本接口取值：ModifyRocketMQGroup |
| Version | 是 | 否 | String | 公共参数，本接口取值：2020-02-17 |
| Region | 是 | 否 | String | 公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表 |
| BroadcastEnable | 否 | 是 | Bool | 是否开启广播消费<br>示例值： true |
| RetryMaxTimes | 否 | 是 | Uint64 | 最大重试次数<br>示例值： 16 |
| ClusterId | 是 | 是 | String | 集群ID<br>示例值： rocketmq-2p9vx3ax9jxg |
| NamespaceId | 是 | 是 | String | 命名空间<br>示例值： test_namespace |
| GroupId | 是 | 是 | String | 消费组名称<br>示例值： test_group |
| Remark | 否 | 是 | String | 说明信息，最长128个字符<br>示例值： 测试消费组 |

| 参数名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| ReadEnable | 否 | 是 | Bool | 是否开启消费<br>示例值： true |

# 3. 输出参数

| 参数名称 | 类型 | 描述 |
| --- | --- | --- |
| RequestId | String | 唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。 |

# 4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

| 错误码 | 描述 |
| --- | --- |
| ResourceInUse | |
| InvalidParameter | |
| InternalError | |
| ResourceNotFound | |
| FailedOperation | |

# RocketMQ 角色授权相关接口
# 创建RocketMQ环境角色授权

## 1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

创建环境角色授权

默认接口请求频率限制：10000次/秒。

接口更新时间：2025-11-18 22:21:43。

接口既验签名又鉴权。

## 2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

| 参数名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Action | 是 | 否 | String | 公共参数，本接口取值：CreateRocketMQEnvironmentRole |
| Version | 是 | 否 | String | 公共参数，本接口取值：2020-02-17 |
| Region | 是 | 否 | String | 公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表 |
| RoleName | 是 | 是 | String | 角色名称。<br>示例值： test_role_name |
| Permissions | 是 | 是 | Array of String | 授权项，最多只能包含produce、consume两项的非空字符串数组。<br>示例值： ["produce"] |
| ClusterId | 是 | 是 | String | 必填字段，集群的ID<br>示例值： rocketmq-2p9vx3ax9jxg |
| DetailedPerms | 否 | 是 | Array of DetailedRolePerm | Topic&Group维度权限配置<br>示例值：查看 |

| 参数名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| EnvironmentId | 是 | 是 | String | 命名空间<br>示例值： test_namespace |

## 3. 输出参数

| 参数名称 | 类型 | 描述 |
| --- | --- | --- |
| EnvironmentId | String | 环境（命名空间）名称。<br>示例值： test_namespace |
| RoleName | String | 角色名称。<br>示例值： test_role_name |
| Permissions | Array of String | 授权项。<br>示例值： ["produce"] |
| RequestId | String | 唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。 |

## 4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

| 错误码 | 描述 |
| --- | --- |
| InvalidParameterValue.InvalidParams | |
| AuthFailure.UnauthorizedOperation | |
| ResourceNotFound.Environment | |
| MissingParameter.NeedMoreParams | |
| ResourceNotFound.Role | |
| ResourceInUse.EnvironmentRole | |
| FailedOperation.CreateEnvironmentRole | |
| FailedOperation.UpdateEnvironmentRole | |
| ResourceInUse | |

| 错误码 | 描述 |
| --- | --- |
| FailedOperation | |
| ResourceNotFound.Cluster | |

# 创建RocketMQ角色

## 1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

创建角色

默认接口请求频率限制：10000次/秒。

接口更新时间：2025-11-18 22:21:43。

接口既验签名又鉴权。

## 2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

| 参数名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Action | 是 | 否 | String | 公共参数，本接口取值：CreateRocketMQRole |
| Version | 是 | 否 | String | 公共参数，本接口取值：2020-02-17 |
| Region | 是 | 否 | String | 公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表 |
| RoleName | 是 | 是 | String | 角色名称，不支持中字以及除了短线和下划线外的特殊字符且长度必须大于0且小等于32。<br>示例值： test_role_name |
| ClusterId | 是 | 是 | String | 必填字段，集群Id<br>示例值： rocketmq-2p9vx3ax9jxg |
| Remark | 否 | 是 | String | 备注说明，长度必须大等于0且小等于128。<br>示例值： 测试角色 |
| PermType | 否 | 是 | String | 角色授权类型（集群：Cluster; 主题或消费组：TopicAndGroup）<br>示例值： Cluster |

## 3. 输出参数

| 参数名称 | 类型 | 描述 |
| --- | --- | --- |
| RoleName | String | 角色名称<br>示例值： test_role_name |
| Token | String | 角色token<br>示例值：<br>eyJrZXlJZCI6InN1bmdvxxxxx0X3JvbGVfMyJ9.dbHR8m6gc4L4vZUrodhW_O9bDulZQ6lraNswNLtcUcY |
| Remark | String | 备注说明<br>示例值： 测试角色 |

| 参数名称 | 类型 | 描述 |
| --- | --- | --- |
| RequestId | String | 唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。 |

# 4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

| 错误码 | 描述 |
| --- | --- |
| InvalidParameterValue.InvalidParams | |
| InternalError.SystemError | |
| AuthFailure.UnauthorizedOperation | |
| MissingParameter.NeedMoreParams | |
| ResourceInUse.Role | |
| FailedOperation.CreateRole | |
| FailedOperation | |
| ResourceNotFound.BrokerCluster | |
| ResourceNotFound.Cluster | |
| FailedOperation.CreateSecretKey | |
| FailedOperation.SaveSecretKey | |
| ResourceUnavailable.FundRequired | |

# 批量删除RocketMQ环境角色授权

## 1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

删除环境角色授权。

默认接口请求频率限制：10000次/秒。

接口更新时间：2025-11-18 22:21:43。

接口既验签名又鉴权。

## 2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

| 参数名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Action | 是 | 否 | String | 公共参数，本接口取值：DeleteRocketMQEnvironmentRoles |
| Version | 是 | 否 | String | 公共参数，本接口取值：2020-02-17 |
| Region | 是 | 否 | String | 公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表 |
| EnvironmentId | 是 | 是 | String | 环境（命名空间）名称。<br>示例值： test_namespace |
| RoleNames | 是 | 是 | Array of String | 角色名称数组。<br>示例值： ["test_role_name"] |
| ClusterId | 是 | 是 | String | 集群的ID<br>示例值： rocketmq-mq-4k4orqgq |

## 3. 输出参数

| 参数名称 | 类型 | 描述 |
| --- | --- | --- |

| 参数名称 | 类型 | 描述 |
| --- | --- | --- |
| EnvironmentId | String | 环境（命名空间）名称。<br>示例值： test_namespace |
| RoleNames | Array of String | 角色名称<br>示例值： ["test_role_name"] |
| RequestId | String | 唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。 |

# 4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

| 错误码 | 描述 |
| --- | --- |
| InvalidParameterValue.InvalidParams | |
| AuthFailure.UnauthorizedOperation | |
| MissingParameter.NeedMoreParams | |
| ResourceNotFound.EnvironmentRole | |
| FailedOperation.DeleteEnvironmentRoles | |
| FailedOperation | |
| ResourceNotFound.Cluster | |

# 批量删除RocketMQ角色

## 1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

删除角色，支持批量。

默认接口请求频率限制：10000次/秒。

接口更新时间：2025-11-18 22:21:43。

接口既验签名又鉴权。

## 2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

| 参数名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Action | 是 | 否 | String | 公共参数，本接口取值：DeleteRocketMQRoles |
| Version | 是 | 否 | String | 公共参数，本接口取值：2020-02-17 |
| Region | 是 | 否 | String | 公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表 |
| RoleNames | 是 | 是 | Array of String | 角色名称数组。<br>示例值： ["test_role_name"] |
| ClusterId | 是 | 是 | String | 必填字段，集群Id<br>示例值： rocketmq-2p9vx3ax9jxg |

## 3. 输出参数

| 参数名称 | 类型 | 描述 |
| --- | --- | --- |
| RoleNames | Array of String | 成功删除的角色名称数组。<br>示例值： ["test_role_name"] |
| RequestId | String | 唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。 |

# 4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

| 错误码 | 描述 |
| --- | --- |
| InvalidParameterValue.InvalidParams | |
| AuthFailure.UnauthorizedOperation | |
| MissingParameter.NeedMoreParams | |
| ResourceNotFound.Role | |
| FailedOperation.DeleteRoles | |
| ResourceInUse.EnvironmentRole | |
| FailedOperation | |
| ResourceNotFound.Cluster | |

# 获取RocketMQ命名空间角色列表

## 1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

获取命名空间角色列表

默认接口请求频率限制：10000次/秒。

接口更新时间：2025-11-18 22:21:43。

接口只验签名不鉴权。

## 2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

| 参数名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Action | 是 | 否 | String | 公共参数，本接口取值：DescribeRocketMQEnvironmentRoles |
| Version | 是 | 否 | String | 公共参数，本接口取值：2020-02-17 |
| Region | 是 | 否 | String | 公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表 |
| Limit | 否 | 是 | Int64 | 返回数量，不填则默认为10，最大值为20。<br>示例值： 20 |
| RoleName | 否 | 是 | String | 角色名称<br>示例值： test_role |
| Filters | 否 | 是 | Array of Filter | RoleName按照角色名进行过滤，精确查询。类型：String必选：否<br>示例值：查看 |
| ClusterId | 是 | 是 | String | 必填字段，RocketMQ集群的ID<br>示例值： rocketmq-ge3n43v3wj |
| EnvironmentId | 否 | 是 | String | 命名空间<br>示例值： test_ns |

| 参数名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Offset | 否 | 是 | Int64 | 起始下标，不填默认为0。<br>示例值： 0 |

## 3. 输出参数

| 参数名称 | 类型 | 描述 |
| --- | --- | --- |
| TotalCount | Int64 | 记录数。<br>示例值： 10 |
| EnvironmentRoleSets | Array of EnvironmentRole | 命名空间角色集合。<br>示例值：查看 |
| RequestId | String | 唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。 |

## 4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

| 错误码 | 描述 |
| --- | --- |
| InvalidParameterValue.InvalidParams | |
| AuthFailure.UnauthorizedOperation | |
| ResourceNotFound.Environment | |
| MissingParameter.NeedMoreParams | |
| ResourceNotFound.Role | |
| ResourceNotFound.BrokerCluster | |
| ResourceNotFound.Cluster | |

# 获取RocketMQ角色列表

## 1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

获取角色列表

默认接口请求频率限制：10000次/秒。

接口更新时间：2025-11-18 22:21:43。

接口既验签名又鉴权。

## 2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

| 参数名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Action | 是 | 否 | String | 公共参数，本接口取值：DescribeRocketMQRoles |
| Version | 是 | 否 | String | 公共参数，本接口取值：2020-02-17 |
| Region | 是 | 否 | String | 公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表 |
| Filters | 否 | 是 | Array of Filter | RoleName按照角色名进行过滤，精确查询。类型：String必选：否<br>示例值：查看 |
| Offset | 是 | 是 | Int64 | 起始下标，不填默认为0。<br>示例值： 0 |
| Limit | 是 | 是 | Int64 | 返回数量，不填则默认为10，最大值为20。<br>示例值： 20 |
| ClusterId | 是 | 是 | String | 必填字段，集群Id<br>示例值： rocketmq-2p9vx3ax9jxg |
| RoleName | 否 | 是 | String | 角色名称，模糊查询<br>示例值： test_role_name |

# 3. 输出参数

| 参数名称 | 类型 | 描述 |
| --- | --- | --- |
| TotalCount | Int64 | 记录数。<br>示例值： 4 |
| RoleSets | Array of Role | 角色数组。<br>示例值：查看 |
| RequestId | String | 唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。 |

# 4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

| 错误码 | 描述 |
| --- | --- |
| InvalidParameterValue.InvalidParams | |
| InternalError.SystemError | |
| AuthFailure.UnauthorizedOperation | |
| MissingParameter.NeedMoreParams | |
| ResourceNotFound.BrokerCluster | |
| ResourceNotFound.Cluster | |

# 修改RocketMQ环境角色授权

## 1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

修改环境角色授权。

默认接口请求频率限制：10000次/秒。

接口更新时间：2025-11-18 22:21:43。

接口既验签名又鉴权。

## 2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

| 参数名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Action | 是 | 否 | String | 公共参数，本接口取值：ModifyRocketMQEnvironmentRole |
| Version | 是 | 否 | String | 公共参数，本接口取值：2020-02-17 |
| Region | 是 | 否 | String | 公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表 |
| DetailedPerms | 否 | 是 | Array of DetailedRolePerm | Topic&Group维度权限配置<br>示例值：查看 |
| EnvironmentId | 是 | 是 | String | 环境（命名空间）名称。<br>示例值： test_namespace |
| RoleName | 是 | 是 | String | 角色名称。<br>示例值： test_role_name |
| Permissions | 是 | 是 | Array of String | 授权项，最多只能包含produce、consume两项的非空字符串数组。<br>示例值： ["produce"] |
| ClusterId | 是 | 是 | String | 必填字段，集群的ID<br>示例值： rocketmq-2p9vx3ax9jxg |

# 3. 输出参数

| 参数名称 | 类型 | 描述 |
| --- | --- | --- |
| EnvironmentId | String | 环境（命名空间）名称。<br>示例值： test_namespace |
| RoleName | String | 角色名称。<br>示例值： test_role_name |
| Permissions | Array of String | 授权项。<br>示例值： ["produce","consume"] |
| RequestId | String | 唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。 |

# 4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

| 错误码 | 描述 |
| --- | --- |
| InvalidParameterValue.InvalidParams | |
| AuthFailure.UnauthorizedOperation | |
| MissingParameter.NeedMoreParams | |
| ResourceNotFound.EnvironmentRole | |
| FailedOperation.UpdateEnvironmentRole | |
| FailedOperation | |
| ResourceNotFound.Cluster | |

# RocketMQ角色修改

## 1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

角色修改

默认接口请求频率限制：10000次/秒。

接口更新时间：2025-11-18 22:21:43。

接口既验签名又鉴权。

## 2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

| 参数名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Action | 是 | 否 | String | 公共参数，本接口取值：ModifyRocketMQRole |
| Version | 是 | 否 | String | 公共参数，本接口取值：2020-02-17 |
| Region | 是 | 否 | String | 公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表 |
| RoleName | 是 | 是 | String | 角色名称，不支持中字以及除了短线和下划线外的特殊字符且长度必须大于0且小等于32。<br>示例值： test_role_name |
| ClusterId | 是 | 是 | String | 必填字段，集群Id<br>示例值： rocketmq-2p9vx3ax9jxg |
| Remark | 否 | 是 | String | 备注说明，长度必须大等于0且小等于128。<br>示例值： 测试修改 |
| PermType | 否 | 是 | String | 权限类型，默认按集群授权（Cluster：集群级别；TopicAndGroup：主题&消费组级别）<br>示例值： Cluster |

## 3. 输出参数

| 参数名称 | 类型 | 描述 |
| --- | --- | --- |
| RoleName | String | 角色名称<br>示例值： test_role_name |
| Remark | String | 备注说明<br>示例值： 测试修改 |
| RequestId | String | 唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。 |

# 4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

| 错误码 | 描述 |
| --- | --- |
| InvalidParameterValue.InvalidParams | |
| AuthFailure.UnauthorizedOperation | |
| MissingParameter.NeedMoreParams | |
| ResourceNotFound.Role | |
| FailedOperation.UpdateRole | |
| ResourceNotFound.BrokerCluster | |
| ResourceNotFound.Cluster | |
| ResourceUnavailable.FundRequired | |

# RocketMQ 集群相关接口
# 创建RocketMQ集群

## 1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

此接口用于创建一个RocketMQ集群

默认接口请求频率限制：20次/秒。

接口更新时间：2025-11-18 22:21:43。

接口只验签名不鉴权。

## 2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

| 参数名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Action | 是 | 否 | String | 公共参数，本接口取值：CreateRocketMQCluster |
| Version | 是 | 否 | String | 公共参数，本接口取值：2020-02-17 |
| Region | 是 | 否 | String | 公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表 |
| Name | 是 | 是 | String | 集群名称，3-64个字符，只能包含字母、数字、“-”及“_”<br>示例值： test_name |
| Remark | 否 | 是 | String | 集群描述，128个字符以内<br>示例值： remark info |
| TagList | 否 | 是 | Array of Tag | 标签列表<br>示例值：查看 |

## 3. 输出参数

| 参数名称 | 类型 | 描述 |
| --- | --- | --- |

| 参数名称 | 类型 | 描述 |
| --- | --- | --- |
| ClusterId | String | 集群ID<br>示例值： rocketmq-a52qova7x97b |
| RequestId | String | 唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。 |

# 4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

| 错误码 | 描述 |
| --- | --- |
| ResourceInUse | |
| InvalidParameter | |
| InternalError | |
| ResourceInsufficient | |
| LimitExceeded | |
| ResourceUnavailable | |
| ResourceNotFound | |
| FailedOperation | |
| LimitExceeded.Clusters | |
| FailedOperation.CreateCluster | |
| ResourceUnavailable.FundRequired | |
| ResourceUnavailable.CreateFailed | |

# 删除RocketMQ集群

## 1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

删除RocketMQ集群

默认接口请求频率限制：20次/秒。

接口更新时间：2025-11-18 22:21:43。

接口既验签名又鉴权。

## 2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

| 参数名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Action | 是 | 否 | String | 公共参数，本接口取值：DeleteRocketMQCluster |
| Version | 是 | 否 | String | 公共参数，本接口取值：2020-02-17 |
| Region | 是 | 否 | String | 公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表 |
| ClusterId | 是 | 是 | String | 待删除的集群Id。<br>示例值： rocketmq-xxxxx |

## 3. 输出参数

| 参数名称 | 类型 | 描述 |
| --- | --- | --- |
| RequestId | String | 唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。 |

## 4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

| 错误码 | 描述 |
| --- | --- |
| InvalidParameter | |
| InternalError | |
| ResourceNotFound | |
| FailedOperation | |
| FailedOperation.DeleteCluster | |
| ResourceNotFound.Cluster | |

# 获取单个RocketMQ集群信息

## 1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

获取单个RocketMQ集群信息

默认接口请求频率限制：20次/秒。

接口更新时间：2025-11-18 22:21:43。

接口既验签名又鉴权。

## 2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

| 参数名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Action | 是 | 否 | String | 公共参数，本接口取值：DescribeRocketMQCluster |
| Version | 是 | 否 | String | 公共参数，本接口取值：2020-02-17 |
| Region | 是 | 否 | String | 公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表 |
| ClusterId | 是 | 是 | String | 集群ID<br>示例值： rocketmq-rd3545bkkj49 |

## 3. 输出参数

| 参数名称 | 类型 | 描述 |
| --- | --- | --- |
| ClusterInfo | RocketMQClusterInfo | 集群信息<br>示例值：查看 |
| ClusterConfig | RocketMQClusterConfig | 集群配置<br>示例值：查看 |

| 参数名称 | 类型 | 描述 |
| --- | --- | --- |
| ClusterStats | RocketMQClusterRecentStats | 集群最近使用量，即将废弃，请使用腾讯云可观测平台获取相关数据。<br>示例值：查看 |
| RequestId | String | 唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。 |

# 4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

| 错误码 | 描述 |
| --- | --- |
| InvalidParameter | |
| ResourceNotFound | |
| FailedOperation | |
| ResourceNotFound.Cluster | |
| InternalError.IllegalMessage | |

# 获取RocketMQ的集群列表

## 1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

获取RocketMQ集群列表

默认接口请求频率限制：20次/秒。

接口更新时间：2025-11-18 22:21:43。

接口只验签名不鉴权。

## 2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

| 参数名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Action | 是 | 否 | String | 公共参数，本接口取值：DescribeRocketMQClusters |
| Version | 是 | 否 | String | 公共参数，本接口取值：2020-02-17 |
| Region | 是 | 否 | String | 公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表 |
| Offset | 是 | 是 | Uint64 | 偏移量<br>示例值： 0 |
| Limit | 是 | 是 | Uint64 | 限制数目<br>示例值： 10 |
| IdKeyword | 否 | 是 | String | 按照集群ID关键字搜索<br>示例值： rocketmq-a52qova7x97b |
| NameKeyword | 否 | 是 | String | 按照集群名称关键字搜索<br>示例值： test_name |
| ClusterIdList | 否 | 是 | Array of String | 集群ID列表过滤<br>示例值： ["rocketmq-a52qova7x97b"] |
| IsTagFilter | 否 | 是 | Bool | 标签过滤查找时，需要设置为true<br>示例值： true |

| 参数名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Filters | 否 | 是 | Array of Filter | 过滤器。目前支持标签过滤。<br>示例值：查看 |

# 3. 输出参数

| 参数名称 | 类型 | 描述 |
| --- | --- | --- |
| ClusterList | Array of RocketMQClusterDetail | 集群信息<br>示例值：查看 |
| TotalCount | Uint64 | 总条数<br>示例值： 100 |
| RequestId | String | 唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。 |

# 4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

| 错误码 | 描述 |
| --- | --- |
| InvalidParameter | |
| InternalError | |
| ResourceNotFound | |
| FailedOperation | |

# 查询RocketMQ实例公网接入点信息

## 1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

接口用于查询RocketMQ实例的公网接入信息

默认接口请求频率限制：20次/秒。

接口更新时间：2025-11-18 22:21:43。

接口既验签名又鉴权。

## 2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

| 参数名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Action | 是 | 否 | String | 公共参数，本接口取值：DescribeRocketMQPublicAccessPoint |
| Version | 是 | 否 | String | 公共参数，本接口取值：2020-02-17 |
| Region | 是 | 否 | String | 公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表 |
| InstanceId | 是 | 是 | String | 集群ID，当前只支持专享集群<br>示例值： rocketmq-a52qova7x97b |

## 3. 输出参数

| 参数名称 | 类型 | 描述 |
| --- | --- | --- |
| Status | Int64 | 公网接入点状态：<br>0， 已开启<br>1， 已关闭<br>2，开启中<br>3，关闭中<br>4，修改中<br>示例值： 0 |

| 参数名称 | 类型 | 描述 |
| --- | --- | --- |
| PayStatus | Int64 | 支付状态：<br>0, 未知<br>1，正常<br>2，欠费<br>示例值： 1 |
| AccessUrl | String | 接入点地址<br>示例值： tdmq-rocketmq.ap-gz.public.tencenttdmq.com:9876 |
| Rules | Array of PublicAccessRule | 安全访问规则列表<br>示例值：查看 |
| Bandwidth | Int64 | 带宽<br>示例值： 2 |
| PayMode | Int64 | 付费模式<br>示例值： 0 |
| BillingFlow | Bool | 公网是否按流量计费<br>示例值： false |
| RequestId | String | 唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。 |

# 4. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见公共错误码。

# 查询RocketMQ专享实例列表

## 1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

查询用户已购的RocketMQ专享实例列表

默认接口请求频率限制：20次/秒。

接口更新时间：2025-11-18 22:21:43。

接口只验签名不鉴权。

## 2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

| 参数名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Action | 是 | 否 | String | 公共参数，本接口取值：DescribeRocketMQVipInstances |
| Version | 是 | 否 | String | 公共参数，本接口取值：2020-02-17 |
| Region | 是 | 否 | String | 公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表 |
| Filters | 否 | 是 | Array of Filter | 查询条件过滤器，支持的查询条件如下：<br>instanceIds - 实例ID<br>instanceName - 实例名称<br>status - 实例状态<br>示例值：查看 |
| Limit | 否 | 是 | Uint64 | 查询数目上限，默认20<br>示例值： 20 |
| Offset | 否 | 是 | Uint64 | 查询起始位置<br>示例值： 1 |

## 3. 输出参数

| 参数名称 | 类型 | 描述 |
| --- | --- | --- |

| 参数名称 | 类型 | 描述 |
| --- | --- | --- |
| TotalCount | Uint64 | 未分页的总数目<br>示例值： 10 |
| Instances | Array of RocketMQVipInstance | 实例信息列表<br>示例值：查看 |
| RequestId | String | 唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。 |

# 4. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见公共错误码。

# 更新RocketMQ集群信息

## 1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

更新RocketMQ集群信息

默认接口请求频率限制：20次/秒。

接口更新时间：2025-11-18 22:21:43。

接口既验签名又鉴权。

## 2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

| 参数名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Action | 是 | 否 | String | 公共参数，本接口取值：ModifyRocketMQCluster |
| Version | 是 | 否 | String | 公共参数，本接口取值：2020-02-17 |
| Region | 是 | 否 | String | 公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表 |
| ClusterId | 是 | 是 | String | RocketMQ集群ID<br>示例值： rocketmq-a52qova7x97b |
| ClusterName | 否 | 是 | String | 3-64个字符，只能包含字母、数字、“-”及“_”<br>示例值： test_name |
| Remark | 否 | 是 | String | 说明信息，不超过128个字符<br>示例值： remark info |
| PublicAccessEnabled | 否 | 是 | Bool | 是否开启HTTP公网访问<br>示例值： false |

## 3. 输出参数

| 参数名称 | 类型 | 描述 |
| --- | --- | --- |
| RequestId | String | 唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。 |

# 4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

| 错误码 | 描述 |
| --- | --- |
| ResourceInUse | |
| InvalidParameter | |
| InternalError | |
| ResourceInsufficient | |
| LimitExceeded | |
| ResourceNotFound | |
| FailedOperation | |
| ResourceNotFound.Cluster | |
| ResourceUnavailable.FundRequired | |

# 主题相关接口
# 新增主题

## 1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

新增指定分区、类型的消息主题

默认接口请求频率限制：200次/秒。

接口更新时间：2021-07-19 19:21:10。

接口既验签名又鉴权。

## 2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

| 参数名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Action | 是 | 否 | String | 公共参数，本接口取值：CreateTopic |
| Version | 是 | 否 | String | 公共参数，本接口取值：2020-02-17 |
| Region | 是 | 否 | String | 公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表 |
| EnvironmentId | 是 | 否 | String | 环境（命名空间）名称。<br>示例值： default |
| TopicName | 是 | 否 | String | 主题名，不支持中字以及除了短线和下划线外的特殊字符且不超过64个字符。<br>示例值： test_topic |
| Partitions | 是 | 否 | Uint64 | 0：非分区topic，无分区；非0：具体分区topic的分区数，最大不允许超过128。<br>示例值： 2 |
| Remark | 否 | 否 | String | 备注，128字符以内。<br>示例值： 2个分区的普通消息队列 |
| TopicType | 是 | 否 | Uint64 | 0： 普通消息；<br>1 ：全局顺序消息； |

| 参数名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| | | | | 2 ：局部顺序消息；<br>3 ：重试队列；<br>4 ：死信队列。<br>示例值： TopicType 和 PulsarTopicType 两者不能同时为空，有一个必选 |
| ClusterId | 否 | 否 | String | Pulsar 集群的ID<br>示例值： pulsar-5r59xd4vnx |

# 3. 输出参数

| 参数名称 | 类型 | 描述 |
| --- | --- | --- |
| EnvironmentId | String | 环境（命名空间）名称。<br>示例值： default |
| TopicName | String | 主题名。<br>示例值： test_topic |
| Partitions | Uint64 | 0：非分区topic，无分区；非0：具体分区topic的分区数。<br>示例值： 2 |
| Remark | String | 备注，128字符以内。<br>示例值： 2个分区的普通消息队列 |
| TopicType | Uint64 | 0： 普通消息；<br>1 ：全局顺序消息；<br>2 ：局部顺序消息；<br>3 ：重试队列；<br>4 ：死信队列；<br>5 ：事务消息。<br>示例值： 0 |
| RequestId | String | 唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。 |

# 4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

| 错误码 | 描述 |
| --- | --- |
| InvalidParameterValue.InvalidParams | |

| 错误码 | 描述 |
| --- | --- |
| InternalError.SystemError | |
| AuthFailure.UnauthorizedOperation | |
| ResourceNotFound.Environment | |
| ResourceInUse.Topic | 重名，主题已存在。 |
| FailedOperation.CreateTopic | |
| MissingParameter.NeedMoreParams | |
| LimitExceeded.Topics | 实例下主题数量超过限制。 |
| InvalidParameterValue | 参数取值错误。 |
| FailedOperation | |
| ResourceNotFound.BrokerCluster | |
| ResourceNotFound.Cluster | |

# 删除主题

## 1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

批量删除topics

默认接口请求频率限制：20次/秒。

接口更新时间：2021-08-04 10:41:16。

接口既验签名又鉴权。

## 2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

| 参数名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Action | 是 | 否 | String | 公共参数，本接口取值：DeleteTopics |
| Version | 是 | 否 | String | 公共参数，本接口取值：2020-02-17 |
| Region | 是 | 否 | String | 公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表 |
| TopicSets | 是 | 否 | Array of TopicRecord | 主题集合，每次最多删除20个。<br>示例值：查看 |
| ClusterId | 否 | 否 | String | pulsar集群Id。<br>示例值： pulsar-xk3ne8k2qkp8 |
| EnvironmentId | 否 | 否 | String | 环境（命名空间）名称。<br>示例值： devNs |
| Force | 否 | 否 | Bool | 是否强制删除，默认为false<br>示例值： false |

## 3. 输出参数

| 参数名称 | 类型 | 描述 |
| --- | --- | --- |

| 参数名称 | 类型 | 描述 |
| --- | --- | --- |
| TopicSets | Array of TopicRecord | 被删除的主题数组。<br>示例值：查看 |
| RequestId | String | 唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。 |

# 4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

| 错误码 | 描述 |
| --- | --- |
| InvalidParameterValue.InvalidParams | |
| InternalError.SystemError | |
| AuthFailure.UnauthorizedOperation | |
| ResourceNotFound.Environment | |
| ResourceNotFound.Topic | |
| FailedOperation.DeleteTopics | |
| OperationDenied.DefaultEnvironment | 默认环境不允许操作。 |
| MissingParameter.NeedMoreParams | |
| FailedOperation | |
| ResourceNotFound.Cluster | |

# 查询主题列表

## 1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

获取环境下主题列表

默认接口请求频率限制：20次/秒。

接口更新时间：2023-04-18 15:22:05。

接口既验签名又鉴权。

## 2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

| 参数名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Action | 是 | 否 | String | 公共参数，本接口取值：DescribeTopics |
| Version | 是 | 否 | String | 公共参数，本接口取值：2020-02-17 |
| Region | 是 | 否 | String | 公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表 |
| EnvironmentId | 是 | 否 | String | 环境（命名空间）名称。<br>示例值： devNs |
| TopicName | 否 | 否 | String | 主题名模糊匹配。<br>示例值： topic1 |
| Offset | 否 | 否 | Uint64 | 起始下标，不填默认为0。<br>示例值： 0 |
| Limit | 否 | 否 | Uint64 | 返回数量，不填则默认为10，最大值为20。<br>示例值： 20 |
| TopicType | 否 | 否 | Uint64 | topic类型描述：<br>0：普通消息；<br>1：全局顺序消息；<br>2：局部顺序消息；<br>3：重试队列；<br>4：死信队列； |

| 参数名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| | | | | 5：事务消息。<br>示例值： 1 |
| ClusterId | 否 | 否 | String | Pulsar 集群的ID<br>示例值： cluster1 |
| Filters | 否 | 否 | Array of Filter | * TopicName<br>按照主题名字查询，精确查询。<br>类型：String<br>必选：否<br>示例值：查看 |
| InternalReqSource | 否 | 否 | String | 内部业务标识<br>示例值： rocketmq-xxxxx |

## 3. 输出参数

| 参数名称 | 类型 | 描述 |
| --- | --- | --- |
| TopicSets | Array of Topic | 主题集合数组。<br>示例值：查看 |
| TotalCount | Uint64 | 主题数量。<br>示例值： 1 |
| RequestId | String | 唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。 |

## 4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

| 错误码 | 描述 |
| --- | --- |
| InvalidParameterValue.InvalidParams | |
| InternalError.SystemError | |
| AuthFailure.UnauthorizedOperation | |
| ResourceNotFound.Environment | |
| MissingParameter.NeedMoreParams | |

| 错误码 | 描述 |
| --- | --- |
| ResourceNotFound.BrokerCluster | |
| ResourceNotFound.Cluster | |

# 修改主题

## 1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

修改主题备注和分区数

默认接口请求频率限制：50次/秒。

接口更新时间：2021-06-16 18:02:21。

接口既验签名又鉴权。

## 2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

| 参数名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Action | 是 | 否 | String | 公共参数，本接口取值：ModifyTopic |
| Version | 是 | 否 | String | 公共参数，本接口取值：2020-02-17 |
| Region | 是 | 否 | String | 公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表 |
| EnvironmentId | 是 | 否 | String | 环境（命名空间）名称。<br>示例值： default |
| TopicName | 是 | 否 | String | 主题名。<br>示例值： test_topic |
| Partitions | 是 | 否 | Uint64 | 分区数，必须大于或者等于原分区数，若想维持原分区数请输入原数目，修改分区数仅对非全局顺序消息起效果，不允许超过128个分区。<br>示例值： 3 |
| Remark | 否 | 否 | String | 备注，128字符以内。<br>示例值： 修改分区为3 |
| ClusterId | 否 | 否 | String | Pulsar 集群的ID<br>示例值： pulsar-5r59xd4vnx |

# 3. 输出参数

| 参数名称 | 类型 | 描述 |
| --- | --- | --- |
| Partitions | Uint64 | 分区数<br>示例值： 3 |
| Remark | String | 备注，128字符以内。<br>示例值： 修改分区为3 |
| RequestId | String | 唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。 |

# 4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

| 错误码 | 描述 |
| --- | --- |
| InvalidParameterValue.InvalidParams | |
| InternalError.SystemError | |
| AuthFailure.UnauthorizedOperation | |
| ResourceNotFound.Topic | |
| FailedOperation.UpdateTopic | 主题更新失败。 |
| MissingParameter.NeedMoreParams | |
| FailedOperation.GetTopicPartitionsFailed | 获取主题分区数失败。 |
| FailedOperation | |

# 其他接口
# 删除角色

## 1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

删除角色，支持批量。

默认接口请求频率限制：20次/秒。

接口更新时间：2021-10-08 16:07:08。

接口既验签名又鉴权。

## 2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

| 参数名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Action | 是 | 否 | String | 公共参数，本接口取值：DeleteRoles |
| Version | 是 | 否 | String | 公共参数，本接口取值：2020-02-17 |
| Region | 是 | 否 | String | 公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表 |
| RoleNames | 是 | 否 | Array of String | 角色名称数组。<br>示例值： ["test_role_1","test_role_2"] |
| ClusterId | 否 | 否 | String | 必填字段，集群Id<br>示例值： pulsar-xk3ne8k2qkp8 |

## 3. 输出参数

| 参数名称 | 类型 | 描述 |
| --- | --- | --- |
| RoleNames | Array of String | 成功删除的角色名称数组。<br>示例值： ["test_role_1","test_role_2"] |

| 参数名称 | 类型 | 描述 |
| --- | --- | --- |
| RequestId | String | 唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。 |

# 4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

| 错误码 | 描述 |
| --- | --- |
| InvalidParameterValue.InvalidParams | |
| AuthFailure.UnauthorizedOperation | |
| MissingParameter.NeedMoreParams | |
| ResourceNotFound.Role | |
| FailedOperation.DeleteRoles | |
| ResourceInUse.EnvironmentRole | |
| FailedOperation | |

# 获取租户VPC绑定关系

## 1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

获取租户VPC绑定关系

默认接口请求频率限制：20次/秒。

接口更新时间：2021-01-13 15:38:57。

接口既验签名又鉴权。

## 2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

| 参数名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Action | 是 | 否 | String | 公共参数，本接口取值：DescribeBindVpcs |
| Version | 是 | 否 | String | 公共参数，本接口取值：2020-02-17 |
| Region | 是 | 否 | String | 公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表 |
| Offset | 否 | 否 | Uint64 | 起始下标，不填默认为0。<br>示例值： 1 |
| Limit | 否 | 否 | Uint64 | 返回数量，不填则默认为10，最大值为20。<br>示例值： 1 |
| ClusterId | 否 | 否 | String | Pulsar 集群的ID<br>示例值： pulsar-xk3ne8k2qkp8 |

## 3. 输出参数

| 参数名称 | 类型 | 描述 |
| --- | --- | --- |
| TotalCount | Uint64 | 记录数。<br>示例值： 1 |

| 参数名称 | 类型 | 描述 |
| --- | --- | --- |
| VpcSets | Array of VpcBindRecord | Vpc集合。<br>示例值：查看 |
| RequestId | String | 唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。 |

# 4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

| 错误码 | 描述 |
| --- | --- |
| InvalidParameterValue.NeedMoreParams | 必要参数没有传递。 |
| InvalidParameterValue.InvalidParams | |
| InternalError.SystemError | |
| AuthFailure.UnauthorizedOperation | |
| MissingParameter.NeedMoreParams | |
| ResourceNotFound.BrokerCluster | |

# 发送消息

## 1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

此接口仅用于测试发生消息，不能作为现网正式生产使用

默认接口请求频率限制：20次/秒。

接口更新时间：2021-06-21 17:11:02。

接口既验签名又鉴权。

## 2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

| 参数名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Action | 是 | 否 | String | 公共参数，本接口取值：SendMsg |
| Version | 是 | 否 | String | 公共参数，本接口取值：2020-02-17 |
| Region | 是 | 否 | String | 公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表 |
| EnvironmentId | 是 | 否 | String | 环境（命名空间）名称。<br>示例值： userNamespace |
| TopicName | 是 | 否 | String | 主题名称，如果是分区topic需要指定具体分区，如果没有指定则默认发到0分区，例如：my_topic-partition-0。<br>示例值： cloud_test |
| MsgContent | 是 | 否 | String | 消息内容，不能为空且大小不得大于5242880个byte。<br>示例值： one_msg |
| ClusterId | 否 | 否 | String | Pulsar 集群的ID<br>示例值： pulsar-k3ex2kxanuol |

## 3. 输出参数

| 参数名称 | 类型 | 描述 |
| --- | --- | --- |
| Result | Bool | 结果。<br>示例值： true |
| MsgId | String | 消息ID。<br>示例值： 50169152:0:0:-1 |
| RequestId | String | 唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。 |

# 4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

| 错误码 | 描述 |
| --- | --- |
| InvalidParameterValue.InvalidParams | |
| InternalError.SystemError | |
| AuthFailure.UnauthorizedOperation | |
| ResourceNotFound.Topic | |
| MissingParameter.NeedMoreParams | |
| FailedOperation.GetTopicPartitionsFailed | 获取主题分区数失败。 |
| FailedOperation.SendMsgFailed | 发送消息失败。 |
| FailedOperation | |

# 命名空间相关接口
# 创建命名空间

## 1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

用于在用户账户下创建消息队列 Tdmq 命名空间

默认接口请求频率限制：20次/秒。

接口更新时间：2021-07-27 16:20:00。

接口既验签名又鉴权。

## 2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

| 参数名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Action | 是 | 否 | String | 公共参数，本接口取值：CreateEnvironment |
| Version | 是 | 否 | String | 公共参数，本接口取值：2020-02-17 |
| Region | 是 | 否 | String | 公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表 |
| EnvironmentId | 是 | 否 | String | 环境（命名空间）名称，不支持中字以及除了短线和下划线外的特殊字符且不超过16个字符。<br>示例值： devNamespace |
| MsgTTL | 是 | 否 | Uint64 | 未消费消息过期时间，单位：秒，最小60，最大1296000，（15天）。<br>示例值： 60 |
| Remark | 否 | 否 | String | 说明，128个字符以内。<br>示例值： demo |
| ClusterId | 否 | 否 | String | Pulsar 集群的ID<br>示例值： pulsar-ge3n43v3wj |

| 参数名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| RetentionPolicy | 否 | 否 | RetentionPolicy | 消息保留策略<br>示例值：查看 |

# 3. 输出参数

| 参数名称 | 类型 | 描述 |
| --- | --- | --- |
| EnvironmentId | String | 环境（命名空间）名称。<br>示例值： devNamespace |
| MsgTTL | Uint64 | 未消费消息过期时间，单位：秒。<br>示例值： 60 |
| Remark | String | 说明，128个字符以内。<br>示例值： demo |
| NamespaceId | String | 命名空间ID<br>示例值： namespace-ak7jexmrz2 |
| RequestId | String | 唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。 |

# 4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

| 错误码 | 描述 |
| --- | --- |
| InvalidParameterValue.InvalidParams | |
| InternalError.SystemError | |
| AuthFailure.UnauthorizedOperation | |
| FailedOperation.CreateEnvironment | 环境创建失败。 |
| OperationDenied.DefaultEnvironment | 默认环境不允许操作。 |
| MissingParameter.NeedMoreParams | |
| LimitExceeded.Environments | 实例下环境数量超过限制。 |
| FailedOperation | |

| 错误码 | 描述 |
| --- | --- |
| ResourceNotFound.Cluster | |
| InvalidParameterValue.TTL | 无效的消息TTL值。 |
| LimitExceeded.Namespaces | 实例下命名空间数量超过限额。 |
| ResourceInUse.Namespace | 重名，命名空间已存在。 |
| FailedOperation.CreateNamespace | 创建命名空间失败。 |

# 删除命名空间

## 1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

批量删除租户下的命名空间

默认接口请求频率限制：20次/秒。

接口更新时间：2023-04-19 18:37:26。

接口既验签名又鉴权。

## 2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

| 参数名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Action | 是 | 否 | String | 公共参数，本接口取值：DeleteEnvironments |
| Version | 是 | 否 | String | 公共参数，本接口取值：2020-02-17 |
| Region | 是 | 否 | String | 公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表 |
| EnvironmentIds | 是 | 否 | Array of String | 环境（命名空间）数组，每次最多删除20个。<br>示例值： [test1,test2] |
| ClusterId | 否 | 否 | String | Pulsar 集群的ID<br>示例值： pulsar-ge3n43v3wj |

## 3. 输出参数

| 参数名称 | 类型 | 描述 |
| --- | --- | --- |
| EnvironmentIds | Array of String | 成功删除的环境（命名空间）数组。<br>示例值： [test1,test2] |
| RequestId | String | 唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。 |

# 4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

| 错误码 | 描述 |
| --- | --- |
| InvalidParameterValue.InvalidParams | |
| InternalError.SystemError | |
| AuthFailure.UnauthorizedOperation | |
| ResourceNotFound.Environment | |
| FailedOperation.DeleteEnvironments | 环境删除失败。 |
| OperationDenied.DefaultEnvironment | 默认环境不允许操作。 |
| MissingParameter.NeedMoreParams | |
| InvalidParameter | |
| FailedOperation.RoleInUse | |
| FailedOperation.TopicInUse | 必须先清除关联主题数据才能继续操作。 |
| FailedOperation.DeleteNamespace | 删除命名空间失败。 |
| ResourceNotFound.Cluster | |

# 获取命名空间属性

## 1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

获取指定命名空间的属性

默认接口请求频率限制：20次/秒。

接口更新时间：2021-07-12 16:00:36。

接口既验签名又鉴权。

## 2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

| 参数名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Action | 是 | 否 | String | 公共参数，本接口取值：DescribeEnvironmentAttributes |
| Version | 是 | 否 | String | 公共参数，本接口取值：2020-02-17 |
| Region | 是 | 否 | String | 公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表 |
| EnvironmentId | 是 | 否 | String | 环境（命名空间）名称。<br>示例值： default |
| ClusterId | 否 | 否 | String | Pulsar 集群的ID<br>示例值： pulsar-ge3n43v3wj |

## 3. 输出参数

| 参数名称 | 类型 | 描述 |
| --- | --- | --- |
| MsgTTL | Uint64 | 未消费消息过期时间，单位：秒，最大1296000（15天）。<br>示例值： 60 |

| 参数名称 | 类型 | 描述 |
| --- | --- | --- |
| RateInByte | Uint64 | 消费速率限制，单位：byte/秒，0：不限速。<br>示例值： 0 |
| RateInSize | Uint64 | 消费速率限制，单位：个数/秒，0：不限速。<br>示例值： 0 |
| RetentionHours | Uint64 | 已消费消息保存策略，单位：小时，0：消费完马上删除。<br>示例值： 0 |
| RetentionSize | Uint64 | 已消费消息保存策略，单位：G，0：消费完马上删除。<br>示例值： 0 |
| EnvironmentId | String | 环境（命名空间）名称。<br>示例值： default |
| Replicas | Uint64 | 副本数。<br>示例值： 2 |
| Remark | String | 备注。<br>示例值： demo |
| RequestId | String | 唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。 |

# 4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

| 错误码 | 描述 |
| --- | --- |
| InternalError.GetAttributesFailed | 获取属性失败。 |
| InvalidParameterValue.InvalidParams | |
| InternalError.SystemError | |
| AuthFailure.UnauthorizedOperation | |
| ResourceNotFound.Environment | |
| FailedOperation.GetEnvironmentAttributesFailed | 获取环境属性失败。 |
| MissingParameter.NeedMoreParams | |
| InternalError.BrokerService | Broker服务异常。 |

# 获取命名空间角色列表

## 1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

获取命名空间角色列表

默认接口请求频率限制：20次/秒。

接口更新时间：2021-10-08 16:06:38。

接口既验签名又鉴权。

## 2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

| 参数名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Action | 是 | 否 | String | 公共参数，本接口取值：DescribeEnvironmentRoles |
| Version | 是 | 否 | String | 公共参数，本接口取值：2020-02-17 |
| Region | 是 | 否 | String | 公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表 |
| EnvironmentId | 否 | 否 | String | 必填字段，环境（命名空间）名称。<br>示例值： default |
| Offset | 否 | 否 | Int64 | 起始下标，不填默认为0。<br>示例值： 0 |
| Limit | 否 | 否 | Int64 | 返回数量，不填则默认为10，最大值为20。<br>示例值： 10 |
| ClusterId | 否 | 否 | String | 必填字段，Pulsar 集群的ID<br>示例值： pulsar-ge3n43v3wj |
| RoleName | 否 | 否 | String | 角色名称<br>示例值： role |
| Filters | 否 | 否 | Array of Filter | * RoleName<br>按照角色名进行过滤，精确查询。<br>类型：String |

| 参数名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| | | | | 必选：否<br>示例值：查看 |

## 3. 输出参数

| 参数名称 | 类型 | 描述 |
| --- | --- | --- |
| TotalCount | Int64 | 记录数。<br>示例值： 1 |
| EnvironmentRoleSets | Array of EnvironmentRole | 命名空间角色集合。<br>示例值：查看 |
| RequestId | String | 唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。 |

## 4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

| 错误码 | 描述 |
| --- | --- |
| InvalidParameterValue.InvalidParams | |
| AuthFailure.UnauthorizedOperation | |
| ResourceNotFound.Environment | |
| MissingParameter.NeedMoreParams | |
| ResourceNotFound.Role | |
| ResourceNotFound.BrokerCluster | |
| ResourceNotFound.Cluster | |

# 获取命名空间列表

## 1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

获取租户下命名空间列表

默认接口请求频率限制：20次/秒。

接口更新时间：2023-04-20 15:35:11。

接口既验签名又鉴权。

## 2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

| 参数名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Action | 是 | 否 | String | 公共参数，本接口取值：DescribeEnvironments |
| Version | 是 | 否 | String | 公共参数，本接口取值：2020-02-17 |
| Region | 是 | 否 | String | 公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表 |
| EnvironmentId | 否 | 否 | String | 命名空间名称，模糊搜索。<br>示例值： default |
| Offset | 否 | 否 | Uint64 | 起始下标，不填默认为0。<br>示例值： 0 |
| Limit | 否 | 否 | Uint64 | 返回数量，不填则默认为10，最大值为20。<br>示例值： 5 |
| ClusterId | 否 | 否 | String | Pulsar 集群的ID<br>示例值： pulsar-ge3n43v3wj |
| Filters | 否 | 否 | Array of Filter | * EnvironmentId<br>按照名称空间进行过滤，精确查询。<br>类型：String<br>必选：否<br>示例值：查看 |

# 3. 输出参数

| 参数名称 | 类型 | 描述 |
| --- | --- | --- |
| TotalCount | Uint64 | 命名空间记录数。<br>示例值： 1 |
| EnvironmentSet | Array of Environment | 命名空间集合数组。<br>示例值：查看 |
| RequestId | String | 唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。 |

# 4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

| 错误码 | 描述 |
| --- | --- |
| InvalidParameterValue.InvalidParams | |
| InternalError.SystemError | |
| AuthFailure.UnauthorizedOperation | |
| MissingParameter.NeedMoreParams | |
| ResourceNotFound.Cluster | |
| FailedOperation | |

# 修改命名空间属性

## 1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

修改指定命名空间的属性值

默认接口请求频率限制：20次/秒。

接口更新时间：2021-07-28 11:01:03。

接口既验签名又鉴权。

## 2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

| 参数名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Action | 是 | 否 | String | 公共参数，本接口取值：ModifyEnvironmentAttributes |
| Version | 是 | 否 | String | 公共参数，本接口取值：2020-02-17 |
| Region | 是 | 否 | String | 公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表 |
| EnvironmentId | 是 | 否 | String | 命名空间名称。<br>示例值： default |
| MsgTTL | 是 | 否 | Uint64 | 未消费消息过期时间，单位：秒，最大1296000。<br>示例值： 60 |
| Remark | 否 | 否 | String | 备注，字符串最长不超过128。<br>示例值： demo |
| ClusterId | 否 | 否 | String | 集群ID<br>示例值： pulsar-ge3n43v3wj |
| RetentionPolicy | 否 | 否 | RetentionPolicy | 消息保留策略<br>示例值：查看 |

# 3. 输出参数

| 参数名称 | 类型 | 描述 |
| --- | --- | --- |
| EnvironmentId | String | 命名空间名称。<br>示例值： default |
| MsgTTL | Uint64 | 未消费消息过期时间，单位：秒。<br>示例值： 60 |
| Remark | String | 备注，字符串最长不超过128。<br>示例值： demo |
| NamespaceId | String | 命名空间ID<br>示例值： namespace-ak27qjs68x |
| RequestId | String | 唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。 |

# 4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

| 错误码 | 描述 |
| --- | --- |
| InvalidParameterValue.InvalidParams | |
| InternalError.SystemError | |
| AuthFailure.UnauthorizedOperation | |
| ResourceNotFound.Environment | |
| FailedOperation.UpdateEnvironment | 环境更新失败。 |
| OperationDenied.DefaultEnvironment | 默认环境不允许操作。 |
| MissingParameter.NeedMoreParams | |
| ResourceNotFound.Cluster | |
| InvalidParameterValue.TTL | 无效的消息TTL值。 |
| FailedOperation.SetTTL | |
| ResourceNotFound.Namespace | |

# 消息相关接口
# 确认消息

## 1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

根据提供的 MessageID 确认指定 topic 中的消息

默认接口请求频率限制：1000次/秒。

接口更新时间：2021-06-24 14:51:05。

接口既验签名又鉴权。

## 2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

| 参数名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Action | 是 | 否 | String | 公共参数，本接口取值：AcknowledgeMessage |
| Version | 是 | 否 | String | 公共参数，本接口取值：2020-02-17 |
| Region | 是 | 否 | String | 公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表 |
| MessageId | 是 | 否 | String | 用作标识消息的唯一的ID（可从 receiveMessage 的返回值中获得）<br>示例值： 5:7:-1 |
| AckTopic | 是 | 否 | String | Topic 名字（可从 receiveMessage 的返回值中获得）这里尽量需要使用topic的全路径，即：tenant/namespace/topic。如果不指定，默认使用的是：public/default<br>示例值： persistent://tenant/namespace/my-topic |
| SubName | 是 | 否 | String | 订阅者的名字，可以从receiveMessage的返回值中获取到。这里尽量与receiveMessage中的订阅者保持一致，否则没办法正确ack 接收回来的消息。<br>示例值： my-sub |

# 3. 输出参数

| 参数名称 | 类型 | 描述 |
| --- | --- | --- |
| ErrorMsg | String | 如果为“”，则说明没有错误返回<br>示例值： "ack error" |
| RequestId | String | 唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。 |

# 4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

| 错误码 | 描述 |
| --- | --- |
| InvalidParameterValue.TopicNotFound | 上传的topic name错误。 |
| InvalidParameter.TenantNotFound | 上传的 tenant name 错误。 |

# 接收消息

## 1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

接收发送到指定 topic 中的消息

默认接口请求频率限制：1000次/秒。

接口更新时间：2021-06-24 14:50:21。

接口既验签名又鉴权。

## 2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

| 参数名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Action | 是 | 否 | String | 公共参数，本接口取值：ReceiveMessage |
| Version | 是 | 否 | String | 公共参数，本接口取值：2020-02-17 |
| Region | 是 | 否 | String | 公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表 |
| Topic | 是 | 否 | String | 接收消息的topic的名字, 这里尽量需要使用topic的全路径，如果不指定，即：tenant/namespace/topic。默认使用的是：public/default<br>示例值： persistent://tenant/namespace/my-topic |
| SubscriptionName | 是 | 否 | String | 订阅者的名字<br>示例值： my-sub |
| ReceiverQueueSize | 否 | 否 | Int64 | 默认值为1000，consumer接收的消息会首先存储到receiverQueueSize这个队列中，用作调优接收消息的速率<br>示例值： 1000 |
| SubInitialPosition | 否 | 否 | String | 默认值为：Latest。用作判定consumer初始接收消息的位置，可选参数为：Earliest, Latest<br>示例值： Latest |

# 3. 输出参数

| 参数名称 | 类型 | 描述 |
| --- | --- | --- |
| MessageID | String | 用作标识消息的唯一主键<br>示例值： {"Response":{"MessageId":"26365:1475:0"}} |
| MessagePayload | String | 接收消息的内容<br>示例值： my-payload |
| AckTopic | String | 提供给 Ack 接口，用来Ack哪一个topic中的消息<br>示例值： my-topic |
| ErrorMsg | String | 返回的错误信息，如果为空，说明没有错误<br>示例值： sssss |
| SubName | String | 返回订阅者的名字，用来创建 ack consumer时使用<br>示例值： my-sub |
| RequestId | String | 唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。 |

# 4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

| 错误码 | 描述 |
| --- | --- |
| InvalidParameterValue.TopicNotFound | 上传的topic name错误。 |
| InvalidParameter.TenantNotFound | 上传的 tenant name 错误。 |
| InvalidParameter.TokenNotFound | 没有获取到正确的 token。 |
| FailedOperation.ReceiveTimeout | 接收消息超时，请重试。 |
| FailedOperation.ReceiveError | 接收消息出错。 |

# 批量发送消息

## 1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

批量发送消息

默认接口请求频率限制：1000次/秒。

接口更新时间：2021-06-24 14:46:50。

接口既验签名又鉴权。

## 2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

| 参数名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Action | 是 | 否 | String | 公共参数，本接口取值：SendBatchMessages |
| Version | 是 | 否 | String | 公共参数，本接口取值：2020-02-17 |
| Region | 是 | 否 | String | 公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表 |
| Topic | 是 | 否 | String | 消息要发送的topic的名字, 这里尽量需要使用topic的全路径，即：tenant/namespace/topic。如果不指定，默认使用的是：public/default<br>示例值： persistent://tenant/namespace/my-topic |
| Payload | 是 | 否 | String | 需要发送消息的内容<br>示例值： hello tdmq |
| StringToken | 否 | 否 | String | String 类型的 token，可以不填，系统会自动获取<br>示例值： abcdefghijklmnopqrstovwxyz |
| ProducerName | 否 | 否 | String | producer 的名字，要求全局是唯一的，如果不设置，系统会自动生成 |

| 参数名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| | | | | 示例值： producer1 |
| SendTimeout | 否 | 否 | Int64 | 单位：s。消息发送的超时时间。默认值为：30s<br>示例值： 30 |
| MaxPendingMessages | 否 | 否 | Int64 | 内存中允许缓存的生产消息的最大数量，默认值：1000条<br>示例值： 1000 |
| BatchingMaxMessages | 否 | 否 | Int64 | 每一个batch中消息的最大数量，默认值：1000条/batch<br>示例值： 1000 |
| BatchingMaxPublishDelay | 否 | 否 | Int64 | 每一个batch最大等待的时间，超过这个时间，不管是否达到指定的batch中消息的数量和大小，都会将该batch发送出去，默认：10ms<br>示例值： 10L |
| BatchingMaxBytes | 否 | 否 | Int64 | 每一个batch中最大允许的消息的大小，默认：128KB<br>示例值： 128*1024 |

# 3. 输出参数

| 参数名称 | 类型 | 描述 |
| --- | --- | --- |
| MessageId | String | 消息的唯一标识<br>示例值： {"Response":{"MessageId":"26365:1475:0"}} |
| ErrorMsg | String | 错误消息，返回为 ""，代表没有错误<br>示例值： "error" |
| RequestId | String | 唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。 |

# 4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

| 错误码 | 描述 |
| --- | --- |

| 错误码 | 描述 |
| --- | --- |
| InvalidParameterValue.TopicNotFound | 上传的topic name错误。 |
| InvalidParameter.TenantNotFound | 上传的 tenant name 错误。 |
| InvalidParameter.TokenNotFound | 没有获取到正确的 token。 |
| FailedOperation.CreateProducerError | 创建producer出错。 |
| FailedOperation.CreatePulsarClientError | 创建TDMQ client的出错。 |

# 发送单条消息

## 1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

发送单条消息

默认接口请求频率限制：1000次/秒。

接口更新时间：2021-06-24 14:46:14。

接口既验签名又鉴权。

## 2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

| 参数名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Action | 是 | 否 | String | 公共参数，本接口取值：SendMessages |
| Version | 是 | 否 | String | 公共参数，本接口取值：2020-02-17 |
| Region | 是 | 否 | String | 公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表 |
| StringToken | 否 | 否 | String | Token 是用来做鉴权使用的，可以不填，系统会自动获取<br>示例值： abcdefghijklmnopqrstovwxyz |
| Topic | 是 | 否 | String | 消息要发送的topic的名字, 这里尽量需要使用topic的全路径，即：tenant/namespace/topic。如果不指定，默认使用的是：public/default<br>示例值： persistent://tenant/namespaces/my-topic |
| Payload | 是 | 否 | String | 要发送的消息的内容<br>示例值： hello tdmq |
| ProducerName | 否 | 否 | String | 设置 producer 的名字，要求全局唯一，用户不配置，系统会随机生成<br>示例值： producer |

| 参数名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| SendTimeout | 否 | 否 | Int64 | 设置消息发送的超时时间，默认为30s<br>示例值： 30 |
| MaxPendingMessages | 否 | 否 | Int64 | 内存中缓存的最大的生产消息的数量，默认为1000条<br>示例值： 1000 |

# 3. 输出参数

| 参数名称 | 类型 | 描述 |
| --- | --- | --- |
| MessageId | String | 消息的messageID, 是全局唯一的，用来标识消息的元数据信息<br>示例值： {"Response":{"MessageId":"26365:1475:0"}} |
| ErrorMsg | String | 返回的错误消息，如果返回为“”，说明没有错误<br>示例值： "error" |
| RequestId | String | 唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。 |

# 4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

| 错误码 | 描述 |
| --- | --- |
| InvalidParameterValue.TopicNotFound | 上传的topic name错误。 |
| InvalidParameter.TenantNotFound | 上传的 tenant name 错误。 |
| InvalidParameter.TokenNotFound | 没有获取到正确的 token。 |
| FailedOperation.CreateProducerError | 创建producer出错。 |
| FailedOperation.CreatePulsarClientError | 创建TDMQ client的出错。 |

# 环境角色授权相关接口
# 创建环境角色授权

## 1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

创建环境角色授权

默认接口请求频率限制：20次/秒。

接口更新时间：2023-04-19 18:38:36。

接口既验签名又鉴权。

## 2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

| 参数名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Action | 是 | 否 | String | 公共参数，本接口取值：CreateEnvironmentRole |
| Version | 是 | 否 | String | 公共参数，本接口取值：2020-02-17 |
| Region | 是 | 否 | String | 公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表 |
| EnvironmentId | 是 | 否 | String | 环境（命名空间）名称。<br>示例值： devDemo |
| RoleName | 是 | 否 | String | 角色名称。<br>示例值： test_role_1 |
| Permissions | 是 | 否 | Array of String | 授权项，最多只能包含produce、consume两项的非空字符串数组。<br>示例值： ["produce"] |
| ClusterId | 否 | 否 | String | 必填字段，集群的ID<br>示例值： pulsar-xk3ne8k2qkp8 |

## 3. 输出参数

| 参数名称 | 类型 | 描述 |
| --- | --- | --- |
| EnvironmentId | String | 环境（命名空间）名称。<br>示例值： devDemo |
| RoleName | String | 角色名称。<br>示例值： test_role_1 |
| Permissions | Array of String | 授权项。<br>示例值： ["produce"] |
| RequestId | String | 唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。 |

# 4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

| 错误码 | 描述 |
| --- | --- |
| InvalidParameterValue.InvalidParams | |
| AuthFailure.UnauthorizedOperation | |
| ResourceNotFound.Environment | |
| MissingParameter.NeedMoreParams | |
| ResourceNotFound.Role | |
| ResourceInUse.EnvironmentRole | |
| FailedOperation.CreateEnvironmentRole | |
| FailedOperation | |
| ResourceNotFound.Cluster | |
| FailedOperation.UpdateEnvironmentRole | |

# 创建角色

## 1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

创建角色

默认接口请求频率限制：20次/秒。

接口更新时间：2023-04-18 15:54:23。

接口既验签名又鉴权。

## 2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

| 参数名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Action | 是 | 否 | String | 公共参数，本接口取值：CreateRole |
| Version | 是 | 否 | String | 公共参数，本接口取值：2020-02-17 |
| Region | 是 | 否 | String | 公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表 |
| RoleName | 是 | 否 | String | 角色名称，不支持中字以及除了短线和下划线外的特殊字符且长度必须大于0且小等于32。<br>示例值： test_role_123 |
| Remark | 否 | 否 | String | 备注说明，长度必须大等于0且小等于128。<br>示例值： devDemo |
| ClusterId | 否 | 否 | String | 必填字段，集群Id<br>示例值： pulsar-xk3ne8k2qkp8 |
| EnvironmentRoleSets | 否 | 否 | Array of EnvironmentRoleSet | 批量操作环境<br>示例值：查看 |

## 3. 输出参数

| 参数名称 | 类型 | 描述 |
| --- | --- | --- |
| RoleName | String | 角色名称<br>示例值： test_role_123 |
| Token | String | 角色token<br>示例值：<br>eyJrZXlJZCI6InN1bmdvxxxxx0X3JvbGVfMyJ9.dbHR8m6gc4L4vZUrodhW_O9bDulZQ6lraNswNLtcUcY |
| Remark | String | 备注说明<br>示例值： devDemo |
| EnvironmentRoleSets | Array of EnvironmentRoleSet | 批量操作环境信息<br>示例值：查看 |
| RequestId | String | 唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。 |

## 4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

| 错误码 | 描述 |
| --- | --- |
| InvalidParameterValue.InvalidParams | |
| InternalError.SystemError | |
| AuthFailure.UnauthorizedOperation | |
| MissingParameter.NeedMoreParams | |
| ResourceInUse.Role | |
| FailedOperation.CreateRole | |
| FailedOperation | |
| ResourceNotFound.BrokerCluster | |
| ResourceNotFound.Cluster | |
| FailedOperation.CreateSecretKey | |
| FailedOperation.SaveSecretKey | |

# 删除环境角色授权

## 1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

删除环境角色授权。

默认接口请求频率限制：20次/秒。

接口更新时间：2021-10-08 16:09:02。

接口既验签名又鉴权。

## 2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

| 参数名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Action | 是 | 否 | String | 公共参数，本接口取值：DeleteEnvironmentRoles |
| Version | 是 | 否 | String | 公共参数，本接口取值：2020-02-17 |
| Region | 是 | 否 | String | 公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表 |
| EnvironmentId | 是 | 否 | String | 环境（命名空间）名称。<br>示例值： devNs |
| RoleNames | 是 | 否 | Array of String | 角色名称数组。<br>示例值： ["test_role"] |
| ClusterId | 否 | 否 | String | 必填字段，集群的ID<br>示例值： pulsar-xk3ne8k2qkp8 |

## 3. 输出参数

| 参数名称 | 类型 | 描述 |
| --- | --- | --- |
| EnvironmentId | String | 环境（命名空间）名称。<br>示例值： devNs |

| 参数名称 | 类型 | 描述 |
| --- | --- | --- |
| RoleNames | Array of String | 角色名称数组。<br>示例值： ["test_role"] |
| RequestId | String | 唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。 |

# 4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

| 错误码 | 描述 |
| --- | --- |
| InvalidParameterValue.InvalidParams | |
| AuthFailure.UnauthorizedOperation | |
| MissingParameter.NeedMoreParams | |
| ResourceNotFound.EnvironmentRole | |
| FailedOperation.DeleteEnvironmentRoles | |

# 获取角色列表

## 1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

获取角色列表

默认接口请求频率限制：20次/秒。

接口更新时间：2021-10-08 16:05:57。

接口既验签名又鉴权。

## 2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

| 参数名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Action | 是 | 否 | String | 公共参数，本接口取值：DescribeRoles |
| Version | 是 | 否 | String | 公共参数，本接口取值：2020-02-17 |
| Region | 是 | 否 | String | 公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表 |
| RoleName | 否 | 否 | String | 角色名称，模糊查询<br>示例值： devRole1 |
| Offset | 否 | 否 | Int64 | 起始下标，不填默认为0。<br>示例值： 0 |
| Limit | 否 | 否 | Int64 | 返回数量，不填则默认为10，最大值为20。<br>示例值： 10 |
| ClusterId | 否 | 否 | String | 必填字段，集群Id<br>示例值： pulsar-xk3ne8k2qkp8 |
| Filters | 否 | 否 | Array of Filter | * RoleName<br>按照角色名进行过滤，精确查询。<br>类型：String<br>必选：否<br>示例值：查看 |

# 3. 输出参数

| 参数名称 | 类型 | 描述 |
| --- | --- | --- |
| TotalCount | Int64 | 记录数。<br>示例值： 4 |
| RoleSets | Array of Role | 角色数组。<br>示例值：查看 |
| RequestId | String | 唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。 |

# 4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

| 错误码 | 描述 |
| --- | --- |
| InvalidParameterValue.InvalidParams | |
| InternalError.SystemError | |
| AuthFailure.UnauthorizedOperation | |
| MissingParameter.NeedMoreParams | |
| ResourceNotFound.BrokerCluster | |
| ResourceNotFound.Cluster | |

# 修改环境角色授权

## 1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

修改环境角色授权。

默认接口请求频率限制：20次/秒。

接口更新时间：2021-10-09 10:09:49。

接口既验签名又鉴权。

## 2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

| 参数名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Action | 是 | 否 | String | 公共参数，本接口取值：ModifyEnvironmentRole |
| Version | 是 | 否 | String | 公共参数，本接口取值：2020-02-17 |
| Region | 是 | 否 | String | 公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表 |
| EnvironmentId | 是 | 否 | String | 环境（命名空间）名称。<br>示例值： default |
| RoleName | 是 | 否 | String | 角色名称。<br>示例值： test_role |
| Permissions | 是 | 否 | Array of String | 授权项，最多只能包含produce、consume两项的非空字符串数组。<br>示例值： ["produce"] |
| ClusterId | 否 | 否 | String | 必填字段，集群的ID<br>示例值： pulsar-xk3ne8k2qkp8 |

## 3. 输出参数

| 参数名称 | 类型 | 描述 |
| --- | --- | --- |
| EnvironmentId | String | 环境（命名空间）名称。<br>示例值： default |
| RoleName | String | 角色名称。<br>示例值： test_role |
| Permissions | Array of String | 授权项。<br>示例值： ["produce","consume"] |
| RequestId | String | 唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。 |

# 4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

| 错误码 | 描述 |
| --- | --- |
| InvalidParameterValue.InvalidParams | |
| AuthFailure.UnauthorizedOperation | |
| MissingParameter.NeedMoreParams | |
| ResourceNotFound.EnvironmentRole | |
| FailedOperation.UpdateEnvironmentRole | |

# 角色修改

## 1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

角色修改

默认接口请求频率限制：20次/秒。

接口更新时间：2025-10-15 16:57:42。

接口既验签名又鉴权。

## 2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

| 参数名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Action | 是 | 否 | String | 公共参数，本接口取值：ModifyRole |
| Version | 是 | 否 | String | 公共参数，本接口取值：2020-02-17 |
| Region | 是 | 否 | String | 公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表 |
| RoleName | 是 | 否 | String | 角色名称，不支持中字以及除了短线和下划线外的特殊字符且长度必须大于0且小等于32。<br>示例值： test_role |
| Remark | 否 | 否 | String | 备注说明，长度必须大等于0且小等于128。<br>示例值： 更新角色 |
| ClusterId | 否 | 否 | String | 必填字段，集群Id<br>示例值： pulsar-xk3ne8k2qkp8 |
| EnvironmentRoleSets | 否 | 否 | Array of EnvironmentRoleSet | 批量操作环境<br>示例值：查看 |

| 参数名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| UnbindAllEnvironment | 否 | 否 | Bool | 是否全部解绑环境<br>示例值： false |

# 3. 输出参数

| 参数名称 | 类型 | 描述 |
| --- | --- | --- |
| RoleName | String | 角色名称<br>示例值： test_role |
| Remark | String | 备注说明<br>示例值： Remark1 |
| RequestId | String | 唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。 |

# 4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

| 错误码 | 描述 |
| --- | --- |
| InvalidParameterValue.InvalidParams | |
| AuthFailure.UnauthorizedOperation | |
| MissingParameter.NeedMoreParams | |
| ResourceNotFound.Role | |
| FailedOperation.UpdateRole | |
| ResourceNotFound.BrokerCluster | |

# 修改资源对应的项目（包含新增）

## 1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

新增或修改资源对应的项目，新增时 oldProjectId 为空

默认接口请求频率限制：20次/秒。

接口更新时间：2021-11-11 22:10:21。

接口只验签名不鉴权。

## 2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

| 参数名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Action | 是 | 否 | String | 公共参数，本接口取值：TpoModifyResourceProject |
| Version | 是 | 否 | String | 公共参数，本接口取值：2020-02-17 |
| Region | 是 | 否 | String | 公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表 |
| ResourceProjectInfos | 是 | 否 | Array of ResourceProjectInfo | ResourceProjectInfo 数组，用于新增或修改资源对应的权限<br>示例值：查看 |

## 3. 输出参数

| 参数名称 | 类型 | 描述 |
| --- | --- | --- |
| ResourceProjectInfos | Array of ResourceProjectInfo | ResourceProjectInfo 数组，新增或修改成功的资源<br>示例值：查看 |

| 参数名称 | 类型 | 描述 |
| --- | --- | --- |
| RequestId | String | 唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。 |

# 4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

| 错误码 | 描述 |
| --- | --- |
| FailedOperation | |
| InternalError | |
| InternalError.CloudAPIError | 云API错误。 |

# 生产消费相关接口
# 创建订阅关系

## 1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

创建一个主题的订阅关系

默认接口请求频率限制：200次/秒。

接口更新时间：2021-07-27 11:43:25。

接口既验签名又鉴权。

## 2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

| 参数名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Action | 是 | 否 | String | 公共参数，本接口取值：CreateSubscription |
| Version | 是 | 否 | String | 公共参数，本接口取值：2020-02-17 |
| Region | 是 | 否 | String | 公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表 |
| EnvironmentId | 是 | 否 | String | 环境（命名空间）名称。<br>示例值： default |
| TopicName | 是 | 否 | String | 主题名称。<br>示例值： cloud_test |
| SubscriptionName | 是 | 否 | String | 订阅者名称，不支持中字以及除了短线和下划线外的特殊字符且不超过150个字符。<br>示例值： cloud_sub |
| Remark | 否 | 否 | String | 备注，128个字符以内。<br>示例值： 创建订阅关系 |
| IsIdempotent | 是 | 否 | Bool | 是否幂等创建，若否不允许创建同名的订阅关系。<br>示例值： true |

| 参数名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| ClusterId | 否 | 否 | String | Pulsar 集群的ID<br>示例值： pulsar-5r59xd4vnx |
| AutoCreatePolicyTopic | 否 | 否 | Bool | 是否自动创建死信和重试主题，True 表示创建，False表示不创建，默认自动创建死信和重试主题。<br>示例值： true |
| PostFixPattern | 否 | 否 | String | 指定死信和重试主题名称规范，LEGACY表示历史命名规则，COMMUNITY表示Pulsar社区命名规范<br>示例值： LEGACY |

## 3. 输出参数

| 参数名称 | 类型 | 描述 |
| --- | --- | --- |
| Result | Bool | 创建结果。<br>示例值： true |
| RequestId | String | 唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。 |

## 4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

| 错误码 | 描述 |
| --- | --- |
| InvalidParameterValue.InvalidParams | |
| InternalError.SystemError | |
| InternalError.Retry | 重试可以成功。 |
| AuthFailure.UnauthorizedOperation | |
| ResourceNotFound.Topic | |
| ResourceInUse.Subscription | 重名，订阅关系已存在。 |
| FailedOperation.GetTopicPartitionsFailed | 获取主题分区数失败。 |

| 错误码 | 描述 |
| --- | --- |
| FailedOperation.CreateSubscription | 创建订阅关系失败。 |
| LimitExceeded.Topics | 实例下主题数量超过限制。 |
| LimitExceeded.Subscriptions | 实例下订阅者数量超过限制。 |
| FailedOperation | |
| ResourceNotFound.BrokerCluster | |

# 删除订阅关系

## 1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

删除订阅关系

默认接口请求频率限制：20次/秒。

接口更新时间：2021-08-04 10:44:23。

接口既验签名又鉴权。

## 2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

| 参数名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Action | 是 | 否 | String | 公共参数，本接口取值：DeleteSubscriptions |
| Version | 是 | 否 | String | 公共参数，本接口取值：2020-02-17 |
| Region | 是 | 否 | String | 公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表 |
| SubscriptionTopicSets | 是 | 否 | Array of SubscriptionTopic | 订阅关系集合，每次最多删除20个。<br>示例值：查看 |
| ClusterId | 否 | 否 | String | pulsar集群Id。<br>示例值： pulsar-xk3ne8k2qkp8 |
| EnvironmentId | 否 | 否 | String | 环境（命名空间）名称。<br>示例值： devNs |
| Force | 否 | 否 | Bool | 是否强制删除，默认为false<br>示例值： false |

# 3. 输出参数

| 参数名称 | 类型 | 描述 |
| --- | --- | --- |
| SubscriptionTopicSets | Array of SubscriptionTopic | 成功删除的订阅关系数组。<br>示例值：查看 |
| RequestId | String | 唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。 |

# 4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

| 错误码 | 描述 |
| --- | --- |
| InvalidParameterValue.InvalidParams | |
| InternalError.SystemError | |
| AuthFailure.UnauthorizedOperation | |
| ResourceNotFound.Topic | |
| ResourceNotFound.Subscription | 订阅关系不存在。 |
| ResourceInUse.Subscription | 重名，订阅关系已存在。 |
| MissingParameter.NeedMoreParams | |
| FailedOperation.GetTopicPartitionsFailed | 获取主题分区数失败。 |
| FailedOperation.DeleteSubscriptions | 删除订阅关系失败。 |
| ResourceNotFound.BrokerCluster | |
| ResourceNotFound.Cluster | |
| OperationDenied.ConsumerRunning | 订阅仍在消费中。 |

# 消费订阅列表

## 1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

查询指定环境和主题下的订阅者列表

默认接口请求频率限制：20次/秒。

接口更新时间：2021-06-29 16:20:55。

接口既验签名又鉴权。

## 2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

| 参数名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Action | 是 | 否 | String | 公共参数，本接口取值：DescribeSubscriptions |
| Version | 是 | 否 | String | 公共参数，本接口取值：2020-02-17 |
| Region | 是 | 否 | String | 公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表 |
| EnvironmentId | 是 | 否 | String | 环境（命名空间）名称。<br>示例值： devNs |
| TopicName | 是 | 否 | String | 主题名称。<br>示例值： devtopic |
| Offset | 否 | 否 | Uint64 | 起始下标，不填默认为0。<br>示例值： 0 |
| Limit | 否 | 否 | Uint64 | 返回数量，不填则默认为10，最大值为20。<br>示例值： 10 |
| SubscriptionName | 否 | 否 | String | 订阅者名称，模糊匹配。<br>示例值： devSub |

| 参数名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Filters | 否 | 否 | Array of FilterSubscription | 数据过滤条件。<br>示例值：查看 |
| ClusterId | 否 | 否 | String | Pulsar 集群的ID<br>示例值： pulsar-b843bz38z5mz |

# 3. 输出参数

| 参数名称 | 类型 | 描述 |
| --- | --- | --- |
| SubscriptionSets | Array of Subscription | 订阅者集合数组。<br>示例值：查看 |
| TotalCount | Uint64 | 数量。<br>示例值： 1 |
| RequestId | String | 唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。 |

# 4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

| 错误码 | 描述 |
| --- | --- |
| InvalidParameterValue.InvalidParams | |
| InternalError.SystemError | |
| AuthFailure.UnauthorizedOperation | |
| ResourceNotFound.Environment | |
| ResourceNotFound.Topic | |
| MissingParameter.NeedMoreParams | |
| FailedOperation | |
| ResourceNotFound.BrokerCluster | |
| ResourceNotFound.Cluster | |

| 错误码 | 描述 |
| --- | --- |
| FailedOperation.DescribeSubscription | 查询订阅数据失败。 |

# 消息回溯

## 1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

根据时间戳进行消息回溯，精确到毫秒

默认接口请求频率限制：20次/秒。

接口更新时间：2021-06-21 11:31:34。

接口既验签名又鉴权。

## 2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

| 参数名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Action | 是 | 否 | String | 公共参数，本接口取值：ResetMsgSubOffsetByTimestamp |
| Version | 是 | 否 | String | 公共参数，本接口取值：2020-02-17 |
| Region | 是 | 否 | String | 公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表 |
| EnvironmentId | 是 | 否 | String | 命名空间名称。<br>示例值： default |
| TopicName | 是 | 否 | String | 主题名称。<br>示例值： topic |
| Subscription | 是 | 否 | String | 订阅者名称。<br>示例值： subscription |
| ToTimestamp | 是 | 否 | Uint64 | 时间戳，精确到毫秒。<br>示例值： 1730690396152 |
| ClusterId | 否 | 否 | String | Pulsar 集群的ID<br>示例值： pulsar-ge3n43v3wj |

# 3. 输出参数

| 参数名称 | 类型 | 描述 |
| --- | --- | --- |
| Result | Bool | 结果。<br>示例值： true |
| RequestId | String | 唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。 |

# 4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

| 错误码 | 描述 |
| --- | --- |
| InvalidParameterValue.InvalidParams | |
| InternalError.SystemError | |
| AuthFailure.UnauthorizedOperation | |
| ResourceNotFound.Topic | |
| ResourceNotFound.Subscription | 订阅关系不存在。 |
| MissingParameter.NeedMoreParams | |
| FailedOperation.ResetMsgSubOffsetByTimestampFailed | 消息回溯设置失败。 |
| FailedOperation | |
| ResourceNotFound.BrokerCluster | |

# 集群相关接口
# 创建集群

## 1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

创建用户的集群

默认接口请求频率限制：20次/秒。

接口更新时间：2023-04-19 11:29:14。

接口既验签名又鉴权。

## 2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

| 参数名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Action | 是 | 否 | String | 公共参数，本接口取值：CreateCluster |
| Version | 是 | 否 | String | 公共参数，本接口取值：2020-02-17 |
| Region | 是 | 否 | String | 公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表 |
| ClusterName | 是 | 否 | String | 集群名称，不支持中字以及除了短线和下划线外的特殊字符且不超过16个字符。<br>示例值： devName |
| BindClusterId | 否 | 否 | Uint64 | 用户专享物理集群ID，如果不传，则默认在公共集群上创建用户集群资源。<br>示例值： sh_cluster_1 |
| Remark | 否 | 否 | String | 说明，128个字符以内。<br>示例值： demo |
| Tags | 否 | 否 | Array of Tag | 集群的标签列表<br>示例值：查看 |
| ProjectId | 否 | 否 | String | 集群对应的项目id<br>示例值： 0 |

| 参数名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| BindClusterName | 否 | 否 | String | 绑定集群名<br>示例值： sh_cluster_1 |

# 3. 输出参数

| 参数名称 | 类型 | 描述 |
| --- | --- | --- |
| ClusterId | String | 集群ID<br>示例值： pulsar-ge3n43v3wj |
| RequestId | String | 唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。 |

# 4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

| 错误码 | 描述 |
| --- | --- |
| FailedOperation | |
| ResourceInUse.Cluster | 集群已存在。 |
| InternalError.SystemError | |
| FailedOperation.CreateCluster | |
| ResourceUnavailable | |
| AuthFailure.UnauthorizedOperation | |
| LimitExceeded.Clusters | |
| MissingParameter.NeedMoreParams | |
| ResourceUnavailable.CreateFailed | |

# 删除集群

## 1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

删除集群

默认接口请求频率限制：20次/秒。

接口更新时间：2021-07-20 14:46:19。

接口既验签名又鉴权。

## 2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

| 参数名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Action | 是 | 否 | String | 公共参数，本接口取值：DeleteCluster |
| Version | 是 | 否 | String | 公共参数，本接口取值：2020-02-17 |
| Region | 是 | 否 | String | 公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表 |
| ClusterId | 是 | 否 | String | 集群Id，传入需要删除的集群Id。<br>示例值： pulsar-ge3n43v3wj |

## 3. 输出参数

| 参数名称 | 类型 | 描述 |
| --- | --- | --- |
| ClusterId | String | 集群的ID<br>示例值： pulsar-ge3n43v3wj |
| RequestId | String | 唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。 |

## 4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

| 错误码 | 描述 |
| --- | --- |
| OperationDenied.DefaultEnvironment | 默认环境不允许操作。 |
| FailedOperation | |
| FailedOperation.NamespaceInUse | 必须先清除关联命名空间才能继续操作。 |
| FailedOperation.RoleInUse | |
| FailedOperation.VpcInUse | 必须先清除关联VPC路由数据才能继续操作。 |
| FailedOperation.DeleteCluster | |
| ResourceNotFound.Cluster | |

# 获取专享集群列表

## 1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

获取用户绑定的专享集群列表

默认接口请求频率限制：20次/秒。

接口更新时间：2021-06-09 15:35:05。

接口只验签名不鉴权。

## 2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

| 参数名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Action | 是 | 否 | String | 公共参数，本接口取值：DescribeBindClusters |
| Version | 是 | 否 | String | 公共参数，本接口取值：2020-02-17 |
| Region | 是 | 否 | String | 公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表 |

## 3. 输出参数

| 参数名称 | 类型 | 描述 |
| --- | --- | --- |
| TotalCount | Int64 | 专享集群的数量<br>示例值： 1 |
| ClusterSet | Array of BindCluster | 专享集群的列表<br>示例值：查看 |
| RequestId | String | 唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。 |

## 4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

| 错误码 | 描述 |
| --- | --- |
| ResourceUnavailable.SystemUpgrade | 系统升级。 |
| UnsupportedOperation | |
| UnauthorizedOperation | 未授权操作。 |
| ResourceUnavailable | |
| ResourcesSoldOut | 资源售罄。 |
| FailedOperation | |
| UnknownParameter | 未知参数错误。 |

# 获取集群详情

## 1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

获取集群的详细信息

默认接口请求频率限制：20次/秒。

接口更新时间：2021-06-09 15:51:04。

接口既验签名又鉴权。

## 2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

| 参数名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Action | 是 | 否 | String | 公共参数，本接口取值：DescribeClusterDetail |
| Version | 是 | 否 | String | 公共参数，本接口取值：2020-02-17 |
| Region | 是 | 否 | String | 公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表 |
| ClusterId | 是 | 否 | String | 集群的ID<br>示例值： pulsar-ge3n43v3wj |

## 3. 输出参数

| 参数名称 | 类型 | 描述 |
| --- | --- | --- |
| ClusterSet | Cluster | 集群的详细信息<br>示例值：查看 |
| RequestId | String | 唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。 |

## 4. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见公共错误码。

# 获取集群列表

## 1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

获取集群列表

默认接口请求频率限制：20次/秒。

接口更新时间：2023-04-18 17:18:12。

接口只验签名不鉴权。

## 2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

| 参数名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Action | 是 | 否 | String | 公共参数，本接口取值：DescribeClusters |
| Version | 是 | 否 | String | 公共参数，本接口取值：2020-02-17 |
| Region | 是 | 否 | String | 公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表 |
| Offset | 否 | 否 | Uint64 | 起始下标，不填默认为0。<br>示例值： 0 |
| Limit | 否 | 否 | Uint64 | 返回数量，不填则默认为10，最大值为20。<br>示例值： 1 |
| ClusterName | 否 | 否 | String | Pulsar 集群的名称<br>示例值： devCluster |
| ClusterIdList | 否 | 否 | Array of String | 集群ID列表过滤<br>示例值： ["cluster001"] |

## 3. 输出参数

| 参数名称 | 类型 | 描述 |
| --- | --- | --- |

| 参数名称 | 类型 | 描述 |
| --- | --- | --- |
| TotalCount | Int64 | 集群列表数量<br>示例值： 1 |
| ClusterSet | Array of Cluster | 集群信息列表<br>示例值：查看 |
| RequestId | String | 唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。 |

# 4. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见公共错误码。

# 更新集群信息

## 1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

更新集群信息

默认接口请求频率限制：20次/秒。

接口更新时间：2021-07-20 14:45:34。

接口既验签名又鉴权。

## 2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

| 参数名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Action | 是 | 否 | String | 公共参数，本接口取值：ModifyCluster |
| Version | 是 | 否 | String | 公共参数，本接口取值：2020-02-17 |
| Region | 是 | 否 | String | 公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表 |
| ClusterId | 是 | 否 | String | Pulsar 集群的ID，需要更新的集群Id。<br>示例值： pulsar-ge3n43v3wj |
| ClusterName | 否 | 否 | String | 更新后的集群名称。<br>示例值： devName |
| Remark | 否 | 否 | String | 说明信息。<br>示例值： demo |

## 3. 输出参数

| 参数名称 | 类型 | 描述 |
| --- | --- | --- |
| ClusterId | String | Pulsar 集群的ID<br>示例值： pulsar-ge3n43v3wj |

| 参数名称 | 类型 | 描述 |
| --- | --- | --- |
| RequestId | String | 唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。 |

# 4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

| 错误码 | 描述 |
| --- | --- |
| MissingParameter | 缺少参数错误。 |
| FailedOperation | |
| InvalidParameterValue.ClusterNameDuplication | 与现有集群名称重复。 |

# 数据结构

## Cluster

集群信息集合

被如下接口引用：DescribeClusterDetail、DescribeClusters

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| ClusterId | 是 | 否 | String | 集群Id。<br>示例值： 1000 |
| ClusterName | 是 | 否 | String | 集群名称。<br>示例值： pulsar-47emajijmazk |
| Remark | 是 | 否 | String | 说明信息。<br>示例值： devInfo |
| EndPointNum | 是 | 否 | Int64 | 接入点数量<br>示例值： 0 |
| CreateTime | 是 | 否 | String | 创建时间<br>示例值： 2021-04-21 19:00:00 |
| Healthy | 是 | 否 | Int64 | 集群是否健康，1表示健康，0表示异常<br>示例值： 1 |
| HealthyInfo | 是 | 是 | String | 集群健康信息<br>示例值： healthy |
| Status | 是 | 否 | Int64 | 集群状态，0:创建中，1:正常，2:删除中，3:已删除，5:创建失败，6: 删除失败<br>示例值： 1 |
| MaxNamespaceNum | 是 | 否 | Int64 | 最大命名空间数量<br>示例值： 100 |
| MaxTopicNum | 是 | 否 | Int64 | 最大Topic数量<br>示例值： 8000 |
| MaxQps | 是 | 否 | Int64 | 最大QPS<br>示例值： 1000 |
| MessageRetentionTime | 是 | 否 | Int64 | 消息保留时间<br>示例值： 100000000 |
| MaxStorageCapacity | 是 | 否 | Int64 | 最大存储容量<br>示例值： 1024 |
| Version | 是 | 是 | String | 集群版本<br>示例值： 2.5.1 |
| PublicEndPoint | 是 | 是 | String | 公网访问接入点<br>示例值： http://pulsar-45epqe3j7jvz.sap-8iutas0k.tdmq.ap-gz.qcloud.xxx.com:8080 |
| VpcEndPoint | 是 | 是 | String | VPC访问接入点<br>示例值： http://pulsar-45epqe3j7jvz.sap-8iutas0k.tdmq.ap-gz.qcloud.xxx.com:8080 |
| NamespaceNum | 是 | 是 | Int64 | 命名空间数量<br>示例值： 2 |
| UsedStorageBudget | 是 | 是 | Int64 | 已使用存储限制，MB为单位<br>示例值： 0 |
| ProjectId | 否 | 是 | String | 资源对应项目id<br>示例值： 251198839 |

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| ProjectName | 否 | 是 | String | 资源对应项目名称<br>示例值： product01 |
| PayMode | 是 | 否 | Int64 | 计费模型0:按量计费，1：包年包月<br>示例值： 0 |
| MaxPublishRateInMessages | 否 | 是 | Int64 | 最大生产消息速率，以条数为单位 注意：此字段可能返回 null，表示取不到有效值。<br>示例值： 1 |
| PclusterName | 否 | 是 | String | 集群名字<br>示例值： test-Id |
| PublicAccessEnabled | 否 | 是 | Bool | 是否开启公网访问，不填时默认开启 注意：此字段可能返回 null，表示取不到有效值。<br>示例值： true |
| MaxDispatchRateInBytes | 否 | 是 | Int64 | 命名空间最大消费带宽，byte为单位 注意：此字段可能返回 null，表示取不到有效值。<br>示例值： 1 |
| Tags | 否 | 是 | Array of Tag | 标签<br>示例值：查看 |
| TopicNum | 否 | 否 | Int64 | topic数量<br>示例值： 1 |
| InternalEndPoint | 否 | 是 | String | 内部接入点<br>示例值： pulsar://1.1.1.1:6650 |
| MaxDispatchRateInMessages | 否 | 是 | Int64 | 最大推送消息速率，以条数为单位 注意：此字段可能返回 null，表示取不到有效值。<br>示例值： 1 |
| InternalPulsarEndPoint | 否 | 是 | String | 内部pulsar-sdk接入地址<br>示例值： pulsar://tdmq.ap-gz.xxx.com:6650 |
| MaxPublishRateInBytes | 否 | 是 | Int64 | 最大生产消息速率，以字节为单位 注意：此字段可能返回 null，表示取不到有效值。<br>示例值： 1 |
| MaxMessageDelayInSeconds | 否 | 是 | Int64 | 最长消息延时，以秒为单位 注意：此字段可能返回 null，表示取不到有效值。<br>示例值： 1 |
| InternalHttpEndPoint | 否 | 是 | String | 内部http接入地址<br>示例值： http://tdmq.ap-gz.xxx.com:8080 |

## SDUTrafficRule

单元化流量路径

被如下接口引用：NotifyDeploymentUnitArchChanged

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| SDUId | 否 | 否 | String | SDU ID<br>示例值： 无 |
| CloudSpaceId | 否 | 否 | String | 云空间 ID<br>示例值： 无 |
| GDUId | 否 | 否 | String | GDU ID<br>示例值： 无 |

## RocketMQConsumerTopic

消费者详情中的主题信息

被如下接口引用：DescribeRocketMQConsumerConnectionDetail

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Accumulative | 是 | 否 | Int64 | 消息堆积数<br>示例值： 100 |
| LastConsumptionTime | 是 | 否 | Uint64 | 最后消费时间，以毫秒为单位<br>示例值： 1697945990000 |
| SubRule | 是 | 是 | String | 订阅规则<br>示例值： test_tag |
| Topic | 是 | 否 | String | 主题名称<br>示例值： test_topic |
| Type | 是 | 否 | String | 主题类型，Normal表示普通，GlobalOrder表示全局顺序，PartitionedOrder表示局部顺序，Transaction表示事务，Retry表示重试，DeadLetter表示死信<br>示例值： Normal |
| PartitionNum | 是 | 否 | Uint64 | 分区数<br>示例值： 3 |

## DisasterRecoveryUnit

容灾单元

被如下接口引用：NotifyDisasterRecoveryUnit

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| RemoteDisasterRecoveryUnitId | 否 | 是 | String | 异地灾备单元 id<br>示例值： unit-01 |
| Status | 否 | 是 | String | 容灾状态NORMAL—正常(恢复)LOCAL_DISASTER_RECOVERY—同城容灾 REMOTE_DISASTER_RECOVERY—异地容灾<br>示例值： NORMAL |
| CloudSpaceId | 否 | 是 | String | 云空间 ID(非必填)<br>示例值： cloud-01 |
| RemoteDisasterRecoveryCloudSpaceId | 否 | 是 | String | 异地灾备云空间 id<br>示例值： cloud-01 |
| CloudId | 否 | 是 | String | 云空间名称<br>示例值： clould-01 |
| RegionId | 否 | 是 | String | 地域 ID<br>示例值： ap-chengdu |
| UnitId | 否 | 是 | String | 单元 ID<br>示例值： unit-01 |
| LocalDisasterRecoveryUnitId | 否 | 是 | String | 同城灾备单元 id<br>示例值： unit-01 |
| CurrentDisasterRecoveryUnitId | 否 | 是 | String | 当前容灾单元 Id<br>示例值： unit-01 |
| ExcludeZones | 否 | 是 | Array of String | 排除流量的可用区 ID<br>示例值： [az1Id,az2Id] |

## ProducerLog

消息生产信息

被如下接口引用：DescribeMsgTrace

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| MsgId | 是 | 否 | String | 消息ID。<br>示例值： 6054156:22067:16 |
| ProducerName | 否 | 否 | String | 生产者名称。<br>示例值： pulsar-7dqvnqwjgzm-2156-9 |
| ProduceTime | 是 | 否 | String | 消息生产时间。<br>示例值： 2024-10-08 14:45:52,636 |
| ProduceUseTime | 是 | 否 | Uint64 | 生产耗时（秒）。<br>示例值： 2 |
| Status | 是 | 否 | String | 状态。<br>示例值： success |

# CloudSpace

单元化云空间

被如下接口引用：NotifyDeploymentUnitArchChanged

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| CloudSpaceName | 否 | 否 | String | 云空间名称<br>示例值： 无 |
| CloudId | 否 | 否 | String | TCS ID<br>示例值： 无 |
| CloudName | 否 | 否 | String | TCS 名称<br>示例值： 无 |
| RegionId | 否 | 否 | String | 地域<br>示例值： 无 |
| GDUs | 否 | 否 | Array of GDU | GDU<br>示例值：查看 |
| Gateways | 否 | 否 | Array of Gateway | 单元化网关<br>示例值：查看 |
| Extensions | 否 | 否 | Array of Extension | 组件扩展信息<br>示例值：查看 |
| CustomerIdMappingServiceURL | 否 | 否 | String | 客户号映射服务地址<br>示例值： 无 |
| CloudSpaceId | 否 | 否 | String | 云空间 ID<br>示例值： 无 |
| SDUs | 否 | 否 | Array of SDU | SDU<br>示例值：查看 |
| SDUTrafficRules | 否 | 否 | Array of SDUTrafficRule | 流量路径<br>示例值：查看 |
| RegionName | 否 | 否 | String | 地域名<br>示例值： 无 |
| UnbindGatewayIds | 否 | 否 | Array of String | 待解绑的网关 Id， rocketmq目前没有使用这个字段<br>示例值： 无 |

# AgentNodeOpt

代理节点

被如下接口引用：DescribeAgentNodesOpt

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| ClusterName | 是 | 否 | String | 集群名称<br>示例值： tdmq_txy_gz_01 |
| ModuleName | 是 | 否 | String | 模块名称<br>示例值： ZOOKEEPER_BOOKIE |
| AgentId | 是 | 否 | String | 代理ID<br>示例值： 10.73.13.46-fqLzhQ6B |
| AgentIp | 是 | 否 | String | 代理IP<br>示例值： 10.73.13.46 |
| ProcId | 是 | 否 | Int64 | 协议ID<br>示例值： 8451 |
| AgentVersion | 是 | 否 | String | 代理版本<br>示例值： 0.0.1-SNAPSHOT |
| JavaVersion | 是 | 否 | String | JDK版本<br>示例值： 1.8.0_275 |

## PartitionsTopic

分区topic

被如下接口引用：DescribeTopics

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| AverageMsgSize | 是 | 是 | String | 最后一次间隔内发布消息的平均byte大小。<br>示例值： 10 |
| ConsumerCount | 是 | 是 | String | 消费者数量。<br>示例值： 10 |
| LastConfirmedEntry | 是 | 是 | String | 被记录下来的消息总数。<br>示例值： 10 |
| LastLedgerCreatedTimestamp | 是 | 是 | String | 最后一个ledger创建的时间。<br>示例值： 2024-10-08 08:04:13,809 |
| MsgRateIn | 是 | 是 | String | 本地和复制的发布者每秒发布消息的速率。<br>示例值： 10 |
| MsgRateOut | 是 | 是 | String | 本地和复制的消费者每秒分发消息的数量之和。<br>示例值： 10 |
| MsgThroughputIn | 是 | 是 | String | 本地和复制的发布者每秒发布消息的byte。<br>示例值： 10 |
| MsgThroughputOut | 是 | 是 | String | 本地和复制的消费者每秒分发消息的byte。<br>示例值： 10 |
| NumberOfEntries | 是 | 是 | String | 被记录下来的消息总数。<br>示例值： 10 |
| Partitions | 是 | 是 | Int64 | 子分区id。<br>示例值： 2 |
| ProducerCount | 是 | 是 | String | 生产者数量。<br>示例值： 2 |
| TotalSize | 是 | 是 | String | 以byte计算的所有消息存储总量。<br>示例值： 2 |

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| TopicType | 是 | 是 | Uint64 | topic类型描述。<br>示例值： 0 |

# RocketMQTopicConfig

RocketMQ主题配置信息

被如下接口引用：CreateRocketMQMigrationTask、ImportRocketMQTopics

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Namespace | 是 | 否 | String | 命名空间<br>示例值： test_ns |
| TopicName | 是 | 否 | String | 主题名称<br>示例值： test_topic |
| Type | 是 | 否 | String | 主题类型：<br>Normal，普通<br>PartitionedOrder, 分区顺序<br>Transaction，事务消息<br>DelayScheduled，延迟/定时消息<br>示例值： Normal |
| Partitions | 是 | 否 | Int64 | 分区个数<br>示例值： 3 |
| Remark | 否 | 否 | String | 备注信息<br>示例值： remark info |

# RocketMQRouterRuleInfoItem

RocketMQ Router输出的规则信息

被如下接口引用：DeleteRocketMQRouterRule、DescribeRocketMQRouterRuleDetail、DescribeRocketMQRouterRules、UpdateRocketMQRouterRule

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| TenMQTTTarget | 否 | 是 | RouterTencentMQTTTarget | 腾讯云MQTT目标集群信息<br>示例值：查看 |
| RemarkName | 否 | 是 | String | 规则备注<br>示例值： - |
| SourceType | 否 | 是 | String | 源类型。<br>OPEN_SOURCE_ROCKETMQ：开源rocketmq<br>ALI_ROCKETMQ：阿里云rocketmq<br>TENCENT_ROCKETMQ：腾讯云rocketmq<br>ALI_MNS：阿里云mns<br>示例值： TENCENT_ROCKETMQ |
| TenRocketMQSource | 否 | 是 | RouterTencentRocketMQSource | 腾讯云rocketmq源信息<br>示例值：查看 |
| Status | 否 | 是 | String | 规则状态。<br>UNKNOWN：未知<br>START_FAIL：启动/重启失败<br>JUST_BE_CREATED：已创建<br>JUST_BE_RE_CREATING：正在重建<br>PAUSED：已暂停<br>RUNNING：运行中<br>SHUTDOWN：已关闭<br><br>示例值： RUNNING |

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| CreateTime | 否 | 是 | Int64 | 创建时间, 毫秒级时间戳<br>示例值： 1694056404492 |
| AliRocketMQTarget | 否 | 是 | RouterRocketMQTarget | 阿里云rocketmq目标信息<br>示例值：查看 |
| AliMessageServiceSource | 否 | 是 | RouterMessageServiceSource | 阿里云mns源信息<br>示例值：查看 |
| TenRocketMQTarget | 否 | 是 | RouterTencentRocketMQTarget | 腾讯云rocketmq目标信息<br>示例值：查看 |
| AliasName | 否 | 是 | String | 别名<br>示例值： - |
| RuleId | 否 | 是 | String | 规则id<br>示例值： router-xxxx |
| AliMessageServiceTarget | 否 | 是 | RouterMessageServiceTarget | 阿里云mns目标信息<br>示例值：查看 |
| ApacheRocketMQTarget | 否 | 是 | RouterRocketMQTarget | 开源rocketmq目标信息<br>示例值：查看 |
| TaskStatusReason | 否 | 是 | String | 任务状态产生原因.<br>UNKNOWN: 未知<br>CleanTaskBySystem: 自动被系统清理<br>CleanTaskByRouterOffline: 自动被系统清理<br>CreateTask: 创建任务失败<br>RestartTask : 重启任务失败<br>CheckBeforeStartTask: 启动任务前参数检查失败<br>DeleteByUser: 用户删除了任务<br>PausedByUser：用户暂停了任务<br>ParamCheckFail：启动/重启时参数检查失败<br>ProducerFail: 生产失败<br>ProducerError: 生产异常<br>示例值： PausedByUser |
| UpdateTime | 否 | 是 | Int64 | 更新时间，毫秒级时间戳<br>示例值： 1694056404492 |
| TenMQTTSource | 否 | 是 | RouterTencentMQTTSource | 腾讯云MQTT源集群信息<br>示例值：查看 |
| TargetType | 否 | 是 | String | 目标类型，和 SourceType 字段一样<br>示例值： TENCENT_ROCKETMQ |
| AliRocketMQSource | 否 | 是 | RouterRocketMQSource | 阿里云rocketmq源信息<br>示例值：查看 |
| ApacheRocketMQSource | 否 | 是 | RouterRocketMQSource | 开源rocketmq源信息<br>示例值：查看 |
| SyncType | 否 | 是 | String | 数据同步类型。<br>Topic: 按topic同步<br>示例值： Topic |

# AMQPVHostConnection

Vhost客户端连接情况

被如下接口引用：DescribeAMQPVHostConnections

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| ClientAddr | 是 | 否 | String | 客户端实例的地址和端口<br>示例值： 192.168.0.1:5555 |

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Protocol | 是 | 否 | String | 客户端协议<br>示例值： AMQP 0.9.1 |
| Security | 是 | 否 | Bool | 是否支持SSL/TLS<br>示例值： true |
| ChannelNum | 是 | 否 | Uint64 | 信道数<br>示例值： 2 |
| ReceiveRate | 是 | 否 | Uint64 | 接受速率<br>示例值： 102 |
| SendRate | 是 | 否 | Uint64 | 发送速率<br>示例值： 97 |

# Topic

主题实例

被如下接口引用：DescribeTopics

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| AverageMsgSize | 是 | 是 | String | 最后一次间隔内发布消息的平均byte大小。<br>示例值： 10 |
| ConsumerCount | 是 | 是 | String | 消费者数量。<br>示例值： 120 |
| LastConfirmedEntry | 是 | 是 | String | 被记录下来的消息总数。<br>示例值： 100 |
| LastLedgerCreatedTimestamp | 是 | 是 | String | 最后一个ledger创建的时间。<br>示例值： 2024-06-19 14:15:54 |
| MsgRateIn | 是 | 是 | String | 本地和复制的发布者每秒发布消息的速率。<br>示例值： 130 |
| MsgRateOut | 是 | 是 | String | 本地和复制的消费者每秒分发消息的数量之和。<br>示例值： 140 |
| MsgThroughputIn | 是 | 是 | String | 本地和复制的发布者每秒发布消息的byte。<br>示例值： 160 |
| MsgThroughputOut | 是 | 是 | String | 本地和复制的消费者每秒分发消息的byte。<br>示例值： 120 |
| NumberOfEntries | 是 | 是 | String | 被记录下来的消息总数。<br>示例值： 100 |
| Partitions | 是 | 是 | Int64 | 分区数<=0：topic下无子分区。<br>示例值： 2 |
| ProducerCount | 是 | 是 | String | 生产者数量。<br>示例值： 2 |
| TotalSize | 是 | 是 | String | 以byte计算的所有消息存储总量。<br>示例值： 2 |
| SubTopicSets | 是 | 是 | Array of PartitionsTopic | 分区topic里面的子分区。<br>示例值：查看 |
| TopicType | 是 | 是 | Uint64 | topic类型描述：<br>0：普通消息；<br>1：全局顺序消息；<br>2：局部顺序消息；<br>3：重试队列；<br>4：死信队列； |

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| | | | | 5：事务消息。<br>示例值： 0 |
| EnvironmentId | 是 | 是 | String | 环境（命名空间）名称。<br>示例值： devNs |
| TopicName | 是 | 是 | String | 主题名称。<br>示例值： devTopic |
| Remark | 是 | 是 | String | 说明，128个字符以内。<br>示例值： devRemark |
| CreateTime | 是 | 是 | Datetime | 创建时间。<br>示例值： 2024-06-19 14:15:54 |
| UpdateTime | 是 | 是 | Datetime | 最近修改时间。<br>示例值： 2024-06-19 14:15:54 |
| ProducerLimit | 是 | 是 | String | 生产者上限。<br>示例值： 5000 |
| ConsumerLimit | 是 | 是 | String | 消费者上限。<br>示例值： 5000 |
| PulsarTopicType | 否 | 是 | Int64 | 0: 非持久非分区 1: 非持久分区 2: 持久非分区 3: 持久分区<br>示例值： 1 |
| ReplicationStatus | 否 | 是 | String | topic所在集群的消息同步状态OriginReplicationOpen 源集群开启消息同步OriginReplicationClose 源集群关闭消息同步TargetReplicationOpen 目标集群开启消息同步TargetReplicationClose 目标集群关闭消息同步<br>示例值： success |
| MsgTTL | 否 | 是 | Uint64 | 未消费消息过期时间，单位：秒<br>示例值： 60 |

## TagFilter

标签过滤器

被如下接口引用：DescribeRocketMQGroups、DescribeRocketMQTopics

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| TagKey | 是 | 否 | String | 标签键名称<br>示例值： test_key |
| TagValues | 是 | 否 | Array of String | 标签值列表<br>示例值： test_value |

## GDU

单元化GDU

被如下接口引用：NotifyDeploymentUnitArchChanged

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Status | 否 | 否 | String | 当前容灾状态<br>示例值： 无 |
| RemoteDisasterRecoveryCloudSpaceId | 否 | 否 | String | 异地灾备云空间 id<br>示例值： 无 |
| GDUId | 否 | 否 | String | GDU ID<br>示例值： 无 |

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| ZoneId | 否 | 否 | String | 可用区<br>示例值： 无 |
| LocalDisasterRecoveryUnitId | 否 | 否 | String | 同城灾备单元 id<br>示例值： 无 |
| UnitEnvType | 否 | 否 | String | 单元环境类型- prod，gray<br>示例值： 无 |
| ExcludeZones | 否 | 否 | Array of String | 排除的可用区Id信息, V4新增<br>示例值： 无 |
| GDUName | 否 | 否 | String | GDU 名称<br>示例值： 无 |
| ZoneName | 否 | 否 | String | 可用区名<br>示例值： 无 |
| RemoteDisasterRecoveryUnitId | 否 | 否 | String | 异地灾备单元 id<br>示例值： 无 |

## AMQPRouteRelation

AMQP路由关系

被如下接口引用：DescribeAMQPRouteRelations

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| RouteRelationId | 是 | 否 | String | 路由关系ID<br>示例值： test |
| SourceExchange | 是 | 否 | String | 源Exchange<br>示例值： xxx |
| DestType | 是 | 否 | String | 目标类型:Queue |
| DestValue | 是 | 否 | String | 目标值<br>示例值： queue1 |
| RoutingKey | 是 | 否 | String | 绑定key<br>示例值： xxxc |
| SourceExchangeType | 是 | 否 | String | 源路由类型:Direct |
| CreateTime | 是 | 否 | Uint64 | 创建时间，以毫秒为单位<br>示例值： 10000 |
| UpdateTime | 是 | 否 | Uint64 | 修改时间，以毫秒为单位<br>示例值： 修改时间 |
| Remark | 是 | 是 | String | 说明信息<br>示例值： 备注 |

## RocketMQNamespace

RocketMQ命名空间信息

被如下接口引用：DescribeInternalRocketMQNamespaces、DescribeRocketMQNamespaces

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Ttl | 是 | 否 | Uint64 | 已废弃，未消费消息的保留时间，以毫秒单位，范围60秒到15天<br>示例值： 100000 |

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| RetentionTime | 是 | 否 | Uint64 | 消息持久化后保留的时间，以毫秒单位<br>示例值： 10000 |
| Remark | 是 | 是 | String | 说明<br>示例值： 备注 |
| PublicEndpoint | 是 | 是 | String | 公网接入点地址<br>示例值： http://ip:port |
| VpcEndpoint | 是 | 是 | String | VPC接入点地址<br>示例值： http://ip:port |
| InternalEndpoint | 是 | 是 | String | 内部接入点地址<br>示例值： http://ip:port |
| UnitType | 否 | 否 | String | 单元类型<br>示例值： SDU |
| UnitStatus | 否 | 否 | String | 容灾状态<br>示例值： NOMORL |
| UnitEnv | 否 | 否 | String | 环境类型<br>示例值： prod |
| NamespaceId | 是 | 否 | String | 命名空间名称，3-64个字符，只能包含字母、数字、“-”及“_”<br>示例值： prod |

# RocketMQClusterDetail

租户RocketMQ集群详细信息

被如下接口引用：DescribeRocketMQClusters

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Status | 是 | 是 | Int64 | 集群状态，0:创建中，1:正常，2:销毁中，3:已删除，4: 隔离中，5:创建失败，6: 删除失败<br>示例值： 0 |
| Info | 是 | 否 | RocketMQClusterInfo | 集群基本信息<br>示例值：查看 |
| Config | 是 | 否 | RocketMQClusterConfig | 集群配置信息<br>示例值：查看 |
| Tags | 是 | 是 | Array of Tag | 标签<br>示例值：查看 |

# AMQPClusterRecentStats

AMQP集群近期使用量

被如下接口引用：DescribeAMQPCluster

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| QueueNum | 是 | 否 | Uint64 | Queue数量<br>示例值： 20 |
| ProducedMsgNum | 是 | 否 | Uint64 | 消息生产数<br>示例值： 11000 |
| AccumulativeMsgNum | 是 | 否 | Uint64 | 消息堆积数<br>示例值： 22000 |

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| ExchangeNum | 是 | 否 | Uint64 | Exchange数量<br>示例值： 20 |

## TemplateItemInfoOpt

模板配置项

被如下接口引用：DescribeTemplateItemOpt

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| ItemKey | 是 | 否 | String | 配置属性名<br>示例值： key |
| ItemDefaultValue | 是 | 是 | String | 配置项默认值<br>示例值： value |
| ItemValue | 是 | 是 | String | 模板配置项值<br>示例值： value |
| ItemDesc | 是 | 是 | String | 描述<br>示例值： desc |
| IsDynamic | 是 | 否 | Bool | 是否动态配置，0需要重启，1不需要重启<br>示例值： 0 |

## ServerLog

服务方信息

被如下接口引用：DescribeMsgTrace

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| SaveTime | 是 | 否 | String | 存储时间。<br>示例值： 2024-10-08 11:38:48,068 |
| Status | 是 | 否 | String | 状态。<br>示例值： success |

## RocketMQMessageTrack

Rocketmq消息消费track信息

被如下接口引用：DescribeRocketMQMsg

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| ConsumeStatus | 是 | 否 | String | 消费状态,<br>CONSUMED: 已消费<br>CONSUMED_BUT_FILTERED: 已过滤<br>NOT_CONSUME: 未消费<br>ENTER_RETRY: 进入重试队列<br>ENTER_DLQ: 进入死信队列<br>UNKNOWN: 查询不到消费状态<br>示例值： CONSUMED |
| TrackType | 是 | 否 | String | 消息track类型<br>示例值： CONSUMED_BUT_FILTERED |
| ExceptionDesc | 是 | 是 | String | 异常信息<br>示例值： NO Exception |

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| TraceConsumeSource | 否 | 否 | String | 消费状态来源，枚举值如下：<br><br>- DIFF_OFFSET：通过服务端offset计算<br>- TRACE_REPORT：通过上报的轨迹判断<br>示例值： TRACE_REPORT |
| Group | 是 | 否 | String | 消费者组<br>示例值： test_group |

## ConfigTypeInfoOpt

集群模块配置类型描述

被如下接口引用：DescribeClusterConfigTypeOpt

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| ComponentName | 是 | 否 | String | 模块组件名称<br>示例值： BROKER |
| ConfigName | 是 | 否 | String | 模块配置文件名称<br>示例值： broker.conf |
| ModuleName | 是 | 否 | String | 配置模块名<br>示例值： BROKER |

## RocketMQGroupConfig

RocketMQ消费组配置信息

被如下接口引用：CreateRocketMQMigrationTask、ImportRocketMQConsumerGroups

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| ConsumeBroadcastEnable | 是 | 否 | Bool | 是否开启广播消费<br>示例值： false |
| ConsumeEnable | 是 | 否 | Bool | 是否开启消费<br>示例值： false |
| Remark | 否 | 否 | String | 备注信息<br>示例值： remark info |
| ConsumerGroupType | 否 | 否 | String | 协议类型，支持以下枚举值<br>TCP;<br>HTTP;<br>示例值： TCP |
| Namespace | 是 | 否 | String | 命名空间<br>示例值： test_ns |
| GroupName | 是 | 否 | String | 消费组名称<br>示例值： test_group |

## AMQPClusterInfo

AMQP集群基本信息

被如下接口引用：DescribeAMQPCluster、DescribeAMQPClusters

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| ClusterId | 是 | 否 | String | 集群ID |

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| | | | | 示例值： amqp-xxxxx |
| ClusterName | 是 | 否 | String | 集群名称<br>示例值： test |
| Region | 是 | 否 | String | 地域信息<br>示例值： 某地区 |
| CreateTime | 是 | 否 | Uint64 | 创建时间，毫秒为单位<br>示例值： 100000000 |
| Remark | 是 | 是 | String | 集群说明信息<br>示例值： 备注 |
| PublicEndPoint | 是 | 是 | String | 公网接入地址<br>示例值： xxxx |
| VpcEndPoint | 是 | 是 | String | VPC接入地址<br>示例值： xxxxx |

# TopicRecord

主题关键信息

被如下接口引用：DeleteTopics

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| EnvironmentId | 是 | 否 | String | 环境（命名空间）名称。<br>示例值： devNs |
| TopicName | 是 | 否 | String | 主题名称。<br>示例值： devTopic |

# Filter

过滤参数

被如下接口引用：DescribeEnvironmentRoles、DescribeEnvironments、DescribeInternalRocketMQInstances、DescribeInternalRocketMQNamespaces、DescribeRocketMQClusters、DescribeRocketMQEnvironmentRoles、DescribeRocketMQMigrateTopicAccessInfo、DescribeRocketMQMigrationTaskDetail、DescribeRocketMQMigrationTasks、DescribeRocketMQRoles、DescribeRocketMQVipInstances、DescribeRoles、DescribeTopics

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Name | 否 | 否 | String | 过滤参数的名字<br>示例值： paramName |
| Values | 否 | 否 | Array of String | 数值<br>示例值： ["a"] |

# ConfigTemplateItemOpt

模板配置项

被如下接口引用：CreateTemplateItemsOpt

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| ItemKey | 是 | 否 | String | 配置属性<br>示例值： key |
| ItemValue | 是 | 否 | String | 属性值<br>示例值： value |

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| ItemDesc | 否 | 否 | String | 配置项备注<br>示例值： desc |
| IsDynamic | 否 | 否 | Bool | 是否动态配置<br>示例值： true |

## RocketMQMigrationTask

RocketMQ迁移任务信息

被如下接口引用：DescribeRocketMQMigrationTasks

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| TopicNum | 是 | 否 | Int64 | 主题总数<br>示例值： 10 |
| GroupNum | 是 | 否 | Int64 | 消费组总数<br>示例值： 10 |
| CreateTime | 是 | 否 | Uint64 | 创建时间，单位毫秒<br>示例值： 1000 |
| Status | 是 | 否 | Int64 | 任务状态：<br>0，迁移中<br>1，迁移成功<br>2，迁移完成，只有部分数据完成迁移<br>示例值： 0 |
| TaskId | 是 | 否 | String | 任务id<br>示例值： xxxx |
| Namespace | 是 | 是 | String | 导入的命名空间<br>示例值： xxxx |
| ClusterId | 是 | 否 | String | 集群id<br>示例值： rocketmq-xxx |
| Type | 是 | 否 | Int64 | 任务类型：<br>0，集群迁移<br>1，导入指定的命名空间<br>示例值： 0 |

## BusinessSystem

单元化客户映射服务地址

被如下接口引用：NotifyDeploymentUnitArchChanged

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| BusinessSystemName | 否 | 否 | String | 业务系统名称<br>示例值： 无 |
| BusinessSystemType | 否 | 否 | String | 业务系统类型<br>示例值： 无 |
| BusinessSystemId | 否 | 否 | String | 业务系统 ID<br>示例值： 无 |

## RocketMQTopicRouteAccessInfo

访问主题路由的信息

被如下接口引用：DescribeRocketMQMigrateTopicAccessInfo

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| LastAccessTimestamp | 否 | 是 | Uint64 | 最近访问的时间戳，毫秒为单位<br>示例值： 无 |
| ClientIP | 否 | 是 | String | 客户端地址<br>示例值： 无 |

# RocketMQClusterConfig

RocketMQ集群配置

被如下接口引用：DescribeRocketMQCluster、DescribeRocketMQClusters

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| MaxNamespaceNum | 是 | 否 | Uint64 | 最大命名空间数量<br>示例值： 10 |
| UsedNamespaceNum | 是 | 否 | Uint64 | 已使用命名空间数量<br>示例值： 3 |
| MaxTopicNum | 是 | 否 | Uint64 | 最大Topic数量<br>示例值： 1000 |
| UsedTopicNum | 是 | 否 | Uint64 | 已使用Topic数量<br>示例值： 8 |
| UsedGroupNum | 是 | 否 | Uint64 | 已使用Group数量<br>示例值： 10 |
| MaxRetentionTime | 是 | 否 | Uint64 | 消息最大保留时间，以毫秒为单位<br>示例值： 100000 |
| MaxTpsLimit | 否 | 否 | Int64 | TPS限额<br>示例值： 4000 |
| MaxTpsPerNamespace | 是 | 否 | Uint64 | 单命名空间TPS上限<br>示例值： 8000 |
| MaxGroupNum | 是 | 否 | Uint64 | 最大Group数量<br>示例值： 10000 |
| MaxLatencyTime | 是 | 否 | Uint64 | 消息最长延时，以毫秒为单位<br>示例值： 100000 |
| MaxQueuesPerTopic | 是 | 是 | Uint64 | 单个主题最大队列数<br>示例值： 16 |
| TopicDistribution | 是 | 是 | Array of RocketMQTopicDistribution | topic分布<br>示例值：查看 |
| MaxRoleNum | 否 | 否 | Int64 | 最大角色数量<br>示例值： 100 |

# AMQPClusterDetail

租户AMQP集群详细信息

被如下接口引用：DescribeAMQPClusters

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Info | 是 | 否 | AMQPClusterInfo | 集群基本信息<br>示例值：查看 |

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Config | 是 | 否 | AMQPClusterConfig | 集群配置信息<br>示例值：查看 |

## RouteRecord

路由记录

被如下接口引用：DescribeRoute

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| VpcId | 是 | 是 | String | 私有网络<br>示例值： xxx |
| SubnetId | 是 | 是 | String | 子网id<br>示例值： xxx |
| RouterId | 是 | 否 | String | 路由id<br>示例值： pulsar-xxx |
| Vip | 是 | 否 | String | vip地址<br>示例值： 10.10.10.10 |
| Vport | 是 | 否 | Uint64 | vport记录<br>示例值： 6000 |
| Remark | 是 | 是 | String | 备注信息<br>示例值： xxx |
| NetType | 是 | 否 | Uint64 | 路由类型 1: 公网 2：vpc 4：支撑网<br>示例值： 1 |
| Id | 否 | 否 | Uint64 | 路由Id（int 类型）<br>示例值： route-ID |

## Tag

标签的key/value的类型

被如下接口引用：CreateCluster、CreateRocketMQCluster、CreateRocketMQGroupV2、DescribeClusterDetail、DescribeClusters、DescribeRocketMQClusters、DescribeRocketMQConsumerConnections、DescribeRocketMQGroups、DescribeRocketMQTopic、DescribeRocketMQTopics

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| TagKey | 是 | 是 | String | 标签的key的值<br>示例值： test_tag_key |
| TagValue | 是 | 是 | String | 标签的Value的值<br>示例值： test_tag_value1 |

## Environment

命名空间信息

被如下接口引用：DescribeEnvironments

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| EnvironmentId | 是 | 否 | String | 命名空间名称<br>示例值： devNs |
| Remark | 是 | 否 | String | 说明<br>示例值： devRemark |

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| MsgTTL | 是 | 否 | Int64 | 未消费消息过期时间，单位：秒，最大1296000（15天）<br>示例值： 2 |
| CreateTime | 是 | 否 | Datetime | 创建时间<br>示例值： 2020-02-17 00:00:00 |
| UpdateTime | 是 | 否 | Datetime | 最近修改时间<br>示例值： 2020-02-17 00:00:00 |
| NamespaceId | 是 | 否 | String | 命名空间ID<br>示例值： devName1 |
| NamespaceName | 是 | 否 | String | 命名空间名称<br>示例值： devName1 |
| TopicNum | 是 | 是 | Int64 | Topic数量<br>示例值： 2 |
| RetentionPolicy | 是 | 是 | RetentionPolicy | 消息保留策略<br>示例值：查看 |

## RouterRocketMQSource

rocketmq router的rocketmq源信息

被如下接口引用：CreateRocketMQRouterRule、DeleteRocketMQRouterRule、DescribeRocketMQRouterRuleDetail、DescribeRocketMQRouterRules、UpdateRocketMQRouterRule

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Namespace | 否 | 是 | String | 4.X的命名空间，没有使用则留空，5.X也留空。<br>naemspace 需要包含MQ_INST那部分. 比如：<br>MQ_INST_rocketmxxx57d53rnn5_tiger_namespace<br>示例值： - |
| Topic | 是 | 是 | String | topic名字<br>示例值： - |
| SecretKey | 是 | 是 | String | secret key<br>示例值： - |
| SourceConsumeConcurrentThreadCount | 否 | 是 | Int64 | 消费者并发数，0 ~ 100<br>示例值： 1 |
| AccessKey | 是 | 是 | String | access key<br>示例值： - |
| FilterExpression | 是 | 是 | String | 过滤表达式, 参考：https://github.com/apache/rocketmq/blob/develop/common/src/main/java/org/apache/rocketmq/common/filter/ExpressionType.java<br>示例值： * |
| FilterFromTimestampMs | 是 | 是 | Int64 | 过滤时间，毫秒时间戳<br>示例值： 1691983398481 |
| FilterType | 是 | 是 | String | 过滤类型，参考：https://github.com/apache/rocketmq/blob/develop/common/src/main/java/org/apache/rocketmq/common/filter/ExpressionType.java<br>示例值： TAG |
| Group | 是 | 是 | String | 消费者组名。会使用该消费者来消费消息<br>示例值： xxx-group |
| NameServerAddressList | 是 | 是 | String | namesrv地址列表，也可以是云厂商提供的接入点信息<br>示例值： 127.0.0.1:9876 |

# RouterMessageServiceSource

rocketmq router mns 源信息

被如下接口引用：CreateRocketMQRouterRule、DeleteRocketMQRouterRule、DescribeRocketMQRouterRuleDetail、DescribeRocketMQRouterRules、UpdateRocketMQRouterRule

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| SourceConsumeConcurrentThreadCount | 否 | 是 | Int64 | mns消费并发数<br>示例值： 1 |
| FilterFromTimestampMs | 否 | 否 | Int64 | 过滤时间戳，毫秒级时间戳<br>示例值： 1694055357311 |
| AccessKey | 是 | 是 | String | access key<br>示例值： - |
| EnableDecodeBody | 否 | 是 | String | 是否自动解码mns body。<br>ON：自动解码，复制后的消息内容为解码后的明文。<br>OFF：不用解码，保持Mns消息体原始状态<br>示例值： OFF |
| Endpoint | 是 | 是 | String | mns接入点。<br>需要和router内网联通，一般是公网接入点<br>示例值： http://1111111.mns.cn-shenzhen.aliyuncs.com |
| Queue | 是 | 是 | String | mns queue名字。<br>可以在https://mns.console.aliyun.com/region/cn-shenzhen/queues 看到<br>示例值： - |
| SecretKey | 是 | 是 | String | secret key<br>示例值： - |

# AMQPQueueDetail

AMQP 队列信息

被如下接口引用：DescribeAMQPQueues

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Name | 是 | 否 | String | Queue名称<br>示例值： test |
| Remark | 是 | 是 | String | 说明<br>示例值： 备注 |
| DestBindedNum | 是 | 是 | Uint64 | 被绑定数<br>示例值： 1 |
| CreateTime | 是 | 是 | Uint64 | 创建时间，以毫秒为单位<br>示例值： 100000 |
| UpdateTime | 是 | 是 | Uint64 | 创建时间，以毫秒为单位<br>示例值： 100000 |
| OnlineConsumerNum | 是 | 是 | Uint64 | 在线消费者数<br>示例值： 20 |
| Tps | 是 | 是 | Uint64 | 每秒钟的事务数<br>示例值： 120 |
| AccumulativeMsgNum | 是 | 是 | Uint64 | 消息堆积数<br>示例值： 90 |
| AutoDelete | 是 | 是 | Bool | 是否自动删除<br>示例值： true |

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| DeadLetterExchange | 是 | 是 | String | 死信交换机<br>示例值： test |
| DeadLetterRoutingKey | 是 | 是 | String | 死信交换机路由键<br>示例值： test |

## SendConfigInfoOpt

下发配置结果

被如下接口引用：SendAllModulesConfigOpt、SendConfigOpt

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| NodeIp | 是 | 否 | String | 节点ip地址<br>示例值： 192.168.1.1 |
| ClusterName | 是 | 否 | String | 集群名<br>示例值： test-cluster |
| ModuleName | 是 | 是 | String | 模块名<br>示例值： BROKER |
| ConfigVersion | 是 | 是 | String | 配置版本<br>示例值： 1 |
| Status | 是 | 否 | Bool | 下发状态<br>示例值： true |

## ConfigModuleInfoOpt

配置模块

被如下接口引用：DescribeModuleListOpt

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| ModuleName | 是 | 否 | String | 模块名<br>示例值： broker |
| ComponentName | 是 | 否 | String | 组件名<br>示例值： BROKER |
| ConfigName | 是 | 否 | String | 配置文件名<br>示例值： broker.conf |
| IsShell | 是 | 否 | Bool | 是否脚本文件<br>示例值： false |

## RocketMQClusterInfo

RocketMQ集群基本信息

被如下接口引用：DescribeRocketMQCluster、DescribeRocketMQClusters

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| PublicEndPoint | 是 | 否 | String | 公网接入地址<br>示例值： rocketmq.access.public.com:9867 |
| InternalEndpoint | 是 | 是 | String | TCP内部接入地址<br>示例值： namespace.internal:port |

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| HttpInternalEndpoint | 是 | 是 | String | HTTP协议内部接入地址<br>示例值： http://ip:port |
| InstanceStatus | 否 | 是 | Int64 | 实例状态，0表示创建中，1表示正常，2表示隔离中，3表示已销毁，4 - 异常, 5 - 发货失败，6 - 变配中，7 - 变配失败<br>示例值： 1 |
| AdminSecretKey | 否 | 是 | String | Admin SK<br>示例值： sk4ea181d5da |
| RocketMQFlag | 是 | 是 | Bool | Rocketmq集群标识<br>示例值： false |
| PublicClbId | 否 | 是 | String | 公网CLB实例ID<br>示例值： clbId |
| Remark | 是 | 是 | String | 集群说明信息<br>示例值： info-remark |
| HttpPublicEndpoint | 是 | 是 | String | HTTP协议公网接入地址<br>示例值： http://ip:port |
| AdminFeatureEnabled | 否 | 否 | Bool | 是否开启集群Admin能力<br>示例值： false |
| ClusterName | 是 | 否 | String | 集群名称<br>示例值： rocket-order-instance |
| IsolateTime | 是 | 是 | Int64 | 欠费停服时间，毫秒为单位<br>示例值： 1729478721 |
| ZoneIds | 否 | 是 | Array of Int64 | 集群节点所在的可用区，若该集群为跨可用区集群，则包含该集群节点所在的多个可用区。<br>示例值： [100001] |
| NamePattern | 否 | 是 | String | 名称格式<br>示例值： STARAND |
| IsFrozen | 否 | 否 | Bool | 是否已冻结<br>示例值： false |
| SupportNamespaceEndpoint | 是 | 是 | Bool | 是否支持命名空间接入点<br>示例值： true |
| HttpVpcEndpoint | 是 | 是 | String | HTTP协议VPC接入地址<br>示例值： http://ip:port |
| AdminAccessKey | 否 | 是 | String | Admin AK<br>示例值： ak5pe5ae0cca |
| Vip | 否 | 是 | String | vip<br>示例值： vip |
| AutoCreateTopicEnabled | 否 | 否 | Bool | 是否开启自动创建主题<br>示例值： false |
| EnableDeletionProtection | 否 | 否 | Bool | 是否开启删除保护<br>示例值： false |
| Vpcs | 是 | 是 | Array of VpcConfig | VPC信息<br>示例值：查看 |
| IsVip | 是 | 是 | Bool | 是否为专享实例<br>示例值： false |
| Status | 是 | 是 | Int64 | 计费状态，1表示正常，2表示已停服，3表示已销毁<br>示例值： 1 |
| AclEnabled | 是 | 是 | Bool | 是否开启ACL鉴权，专享实例支持关闭<br>示例值： true |

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| SupportMigration | 否 | 是 | Bool | 是否支持迁移<br>示例值： false |
| ZoneId | 否 | 是 | Int64 | 集群所属可用区，表明集群归属的可用区<br>示例值： 100001 |
| ClusterId | 是 | 否 | String | 集群ID<br>示例值： rocketmq-4k4orqgq |
| Region | 是 | 否 | String | 地域信息<br>示例值： 华南地区 |
| CreateTime | 是 | 否 | Uint64 | 创建时间，毫秒为单位<br>示例值： 1729478721 |
| VpcEndPoint | 是 | 否 | String | VPC接入地址<br>示例值： rocketmq.resource.vpc.com:5010 |
| VpcId | 否 | 是 | String | 所属VPC<br>示例值： mqvpc296 |

## RocketMQConsumerClient

消费者客户端信息

被如下接口引用：DescribeRocketMQConsumerClients

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| ClientId | 是 | 是 | String | 客户端ID<br>示例值： 127.0.0.1@25760#13749780392400 |
| ClientAddr | 是 | 是 | String | 客户端地址<br>示例值： 127.0.0.1:6133 |

## RocketMQTopic

RocketMQ主题信息

被如下接口引用：DescribeRocketMQTopic、DescribeRocketMQTopics

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| InstanceId | 是 | 是 | String | 实例ID<br>示例值： rocketmq-2p9vx3ax9jxg |
| Namespace | 是 | 是 | String | 命名空间<br>示例值： test_namespace |
| LastUpdateTime | 否 | 是 | Int64 | 最后写入时间，单位为秒<br>示例值： 1727420571 |
| SubscriptionCount | 否 | 是 | Int64 | 订阅数量<br>示例值： 10 |
| Type | 是 | 否 | String | 主题的类别，为枚举类型，Normal，GlobalOrder，PartitionedOrder，Transaction，Retry及DeadLetter<br>示例值： Normal |
| GroupNum | 是 | 否 | Uint64 | 订阅组数量<br>示例值： 5 |
| Remark | 是 | 是 | String | 说明<br>示例值： 备注信息 |

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| SubscriptionData | 否 | 是 | Array of RocketMQSubscription | 订阅关系列表<br>示例值：查看 |
| TagList | 否 | 否 | Array of Tag | 绑定的标签列表<br>示例值：查看 |
| Name | 是 | 否 | String | 主题名称<br>示例值： test_topic |
| PartitionNum | 是 | 否 | Uint64 | 读写分区数<br>示例值： 3 |
| CreateTime | 是 | 否 | Uint64 | 创建时间，以毫秒为单位<br>示例值： 100000 |
| UpdateTime | 是 | 否 | Uint64 | 创建时间，以毫秒为单位<br>示例值： 100000 |

## ConsumersSchedule

消费进度详情

被如下接口引用：DescribeSubscriptions

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Partitions | 是 | 是 | Uint64 | 当前分区id。<br>示例值： 1 |
| NumberOfEntries | 是 | 是 | Uint64 | 消息数量。<br>示例值： 1000 |
| MsgBacklog | 是 | 是 | Uint64 | 消息积压数量。<br>示例值： 2 |
| MsgRateOut | 是 | 否 | String | 消费者每秒分发消息的数量之和。<br>示例值： 0.0 |
| MsgThroughputOut | 是 | 否 | String | 消费者每秒消息的byte。<br>示例值： 0.0 |
| MsgRateExpired | 是 | 是 | String | 超时丢弃比例。<br>示例值： 0.0 |

## AMQPQueueConsumer

AMQP队列消费者信息

被如下接口引用：DescribeAMQPQueueConsumers

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| ClientAddr | 是 | 否 | String | 消费者实例的地址和端口<br>示例值： 192.168.0.1:5555 |
| ClientTag | 是 | 否 | String | 消费者标签<br>示例值： consumer1 |
| CreateTime | 是 | 否 | Uint64 | 创建时间，以毫秒为单位<br>示例值： 100000 |

## RocketMQRoleConfig

RocketMQ角色配置信息

被如下接口引用：CreateRocketMQMigrationTask

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Remark | 否 | 否 | String | 备注<br>示例值： remark info |
| PermType | 否 | 否 | String | 权限类型，默认按集群授权（Cluster：集群级别；TopicAndGroup：主题&消费组级别）<br>示例值： Cluster |
| DetailedRolePerms | 否 | 否 | Array of DetailedRolePerm | Topic和Group维度权限配置<br>示例值：查看 |
| RoleName | 否 | 否 | String | 角色名，对应SecretKey<br>示例值： test_role |
| RoleToken | 否 | 否 | String | accessKey<br>示例值： akTd0eozCSH78BYp1E |
| EnvironmentId | 否 | 否 | String | 命名空间<br>示例值： test_ns |
| Permissions | 否 | 否 | Array of String | 角色权限<br>示例值： [ "produce", "consume" ] |

## SDU

单元化SDU

被如下接口引用：NotifyDeploymentUnitArchChanged

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| ZoneId | 否 | 否 | String | 可用区<br>示例值： 无 |
| ZoneName | 否 | 否 | String | 可用去名<br>示例值： 无 |
| RemoteDisasterRecoveryCloudSpaceId | 否 | 否 | String | 异地灾备云空间 id<br>示例值： 无 |
| UnitEnvType | 否 | 否 | String | 单元环境类型- prod，gray<br>示例值： 无 |
| SDUId | 否 | 否 | String | SDU ID<br>示例值： 无 |
| SDUName | 否 | 否 | String | SDU 名称<br>示例值： 无 |
| LocalDisasterRecoveryUnitId | 否 | 否 | String | 同城灾备单元 id<br>示例值： 无 |
| RemoteDisasterRecoveryUnitId | 否 | 否 | String | 异地灾备单元 id<br>示例值： 无 |
| Status | 否 | 否 | String | 当前容灾状态<br>示例值： 无 |

## RocketMQRouterRuleInfo

rocketmq router规则基本信息

被如下接口引用：CreateRocketMQRouterRule

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| SourceType | 是 | 否 | String | 源类型。<br>OPEN_SOURCE_ROCKETMQ：开源rocketmq<br>ALI_ROCKETMQ：阿里云rocketmq<br>TENCENT_ROCKETMQ：腾讯云rocketmq<br>TENCENT_MQTT：腾讯云MQTT<br>ALI_MNS：阿里云mns<br>示例值： TENCENT_ROCKETMQ |
| TargetType | 是 | 否 | String | 目标类型。<br>枚举和SourceType字段一样<br>示例值： TENCENT_ROCKETMQ |
| AliMessageServiceSource | 否 | 否 | RouterMessageServiceSource | 阿里云mns源信息<br>示例值：查看 |
| AliMessageServiceTarget | 否 | 否 | RouterMessageServiceTarget | 阿里云mns目标信息<br>示例值：查看 |
| TenRocketMQTarget | 否 | 否 | RouterTencentRocketMQTarget | 腾讯云rocketmq目标信息<br>示例值：查看 |
| AliasName | 否 | 否 | String | 别名<br>示例值： - |
| TenMQTTTarget | 否 | 否 | RouterTencentMQTTTarget | 腾讯云 MQTT 目标集群信息<br>示例值：查看 |
| RemarkName | 否 | 否 | String | 规则备注，创建后任务后可以修改<br>示例值： - |
| AliRocketMQSource | 否 | 否 | RouterRocketMQSource | 阿里云rocketmq源信息<br>示例值：查看 |
| AliRocketMQTarget | 否 | 否 | RouterRocketMQTarget | 阿里云rocketmq目标信息<br>示例值：查看 |
| TenRocketMQSource | 否 | 否 | RouterTencentRocketMQSource | 腾讯云rocketmq源信息<br>示例值：查看 |
| TenMQTTSource | 否 | 否 | RouterTencentMQTTSource | 腾讯云 MQTT 源集群信息<br>示例值：查看 |

## TraceResult

消息轨迹结果

被如下接口引用：DescribeRocketMQMsgTrace

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Stage | 是 | 否 | String | 阶段<br>示例值： produce |
| Data | 是 | 否 | String | 内容详情<br>示例值： {"", ""} |

## PublicAccessRule

公网访问安全规则

被如下接口引用：DescribeRocketMQPublicAccessPoint

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| IpRule | 是 | 否 | String | ip网段信息 |

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| | | | | 示例值： 0.0.0.0/0 |
| Allow | 是 | 否 | Bool | 允许或者拒绝<br>示例值： true |
| Remark | 否 | 是 | String | 备注信息<br>示例值： test remark info |

## RouterTencentRocketMQSource

rocketmq router腾讯云源

被如下接口引用：CreateRocketMQRouterRule、DeleteRocketMQRouterRule、DescribeRocketMQRouterRuleDetail、DescribeRocketMQRouterRules、UpdateRocketMQRouterRule

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| SourceConsumeConcurrentThreadCount | 否 | 是 | Int64 | 消费者并发数<br>示例值： 1 |
| ClusterId | 是 | 是 | String | rocketmq实例id<br>示例值： rocketmq-xxxxx |
| SourceRegion | 否 | 是 | String | 源地域<br>示例值： ap-beijing |
| FilterExpression | 是 | 是 | String | 过滤表达式, 参考：https://github.com/apache/rocketmq/blob/develop/common/src/main/java/org/apache/rocketmq/common/filter/ExpressionType.java<br>示例值： - |
| FilterFromTimestampMs | 是 | 是 | Int64 | 过滤时间，毫秒时间戳<br>示例值： 1691983398481 |
| FilterType | 是 | 是 | String | 过滤类型，参考：https://github.com/apache/rocketmq/blob/develop/common/src/main/java/org/apache/rocketmq/common/filter/ExpressionType.java<br>示例值： TAG |
| Namespace | 否 | 是 | String | 4.X的命名空间，没有使用则留空，5.X也留空。 naemspace 需要包含MQ_INST那部分. 比如：MQ_INST_rocketmqka57d53rnn5_tiger_namespace<br>示例值： - |
| Topic | 是 | 是 | String | topic名字<br>示例值： - |

## EditableTarget

RocketMQ Router可编辑的目标信息

被如下接口引用：UpdateRocketMQRouterRule

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| AccessKey | 否 | 否 | String | 新的access key<br>示例值： - |
| SecretKey | 否 | 否 | String | 新的secret key<br>示例值： - |

## Role

角色实例

被如下接口引用：DescribeRocketMQRoles、DescribeRoles

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Secretkey | 否 | 是 | String | secret key<br>示例值： Zl34bUZYqqsdsdfYaxlrZ |
| PermType | 否 | 否 | String | 授权类型（Cluster：集群；TopicAndGroup：主题或消费组）<br>示例值： Cluster |
| RoleName | 是 | 否 | String | 角色名称。<br>示例值： test_role_name |
| Token | 是 | 否 | String | 角色token值。<br>示例值：<br>eyJrZXlJZCI6InN1bmdvxxxxx0X3JvbGVfMyJ9.dbHR8m6gc4L4vZUrodhW_O9bDulZQ6lraNswNLtcUcY |
| Remark | 是 | 否 | String | 备注说明。<br>示例值： 测试账号 |
| CreateTime | 是 | 否 | Datetime | 创建时间。<br>示例值： 2020-09-22 00:00:00 |
| UpdateTime | 是 | 否 | Datetime | 更新时间。<br>示例值： 2020-09-22 00:00:00 |

# SubscriptionTopic

订阅关系

被如下接口引用：DeleteSubscriptions

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| EnvironmentId | 是 | 否 | String | 环境（命名空间）名称。<br>示例值： devNs |
| TopicName | 是 | 否 | String | 主题名称。<br>示例值： devTopic |
| SubscriptionName | 是 | 否 | String | 订阅名称。<br>示例值： devSub |

# ConfigBaseInfoOpt

集群配置基础信息

被如下接口引用：CreateConfigOpt、DescribeAllVersionConfigsOpt、DescribeClusterConfigOpt

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| ClusterName | 是 | 否 | String | 集群名称<br>示例值： tdmq_txy_gz_01 |
| ModuleName | 是 | 否 | String | 模块名称<br>示例值： BROKER |
| ConfigVersion | 是 | 否 | String | 配置版本<br>示例值： 1 |
| RootPath | 是 | 是 | String | 程序路径<br>示例值： /usr |
| DataPath | 是 | 是 | String | 数据路径<br>示例值： /usr |

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| ConfigPath | 是 | 是 | String | 配置路径<br>示例值： /usr |
| ConfigName | 是 | 是 | String | 配置文件名<br>示例值： broker.conf |

## DetailedRolePerm

Topic&Group维度的权限配置

被如下接口引用：CreateRocketMQEnvironmentRole、CreateRocketMQGeneralRole、CreateRocketMQMigrationTask、ModifyRocketMQEnvironmentRole

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Remark | 否 | 否 | String | 资源备注<br>示例值： test remark info. |
| Resource | 是 | 否 | String | 权限对应的资源<br>示例值： test_topic |
| PermWrite | 是 | 否 | Bool | 是否开启生产权限<br>示例值： true |
| PermRead | 是 | 否 | Bool | 是否开启消费权限<br>示例值： false |
| ResourceType | 是 | 否 | String | 授权资源类型，枚举值如下：<br>- Topic：主题维度<br>- Group：消费组维度<br>- Cluster：集群维度（默认值）<br>示例值： Topic |

## MsgLog

消息日志

被如下接口引用：DescribeTopicMsgs

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| MsgId | 否 | 是 | String | 消息ID。<br>示例值： 931087:81:0 |
| ProducerName | 否 | 是 | String | 生产者名称。<br>示例值： devProducerName |
| ProduceTime | 否 | 是 | String | 生产时间。<br>示例值： 2024-10-08 08:04:13,809 |
| ProducerAddr | 否 | 是 | String | 生产客户端地址。<br>示例值： 10.0.0.120:38168 |

## ConsumerLog

消费日志

被如下接口引用：DescribeMsgTrace

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| MsgId | 是 | 否 | String | 消息ID。<br>示例值： 1:1:1 |

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| ConsumerGroup | 是 | 否 | String | 消费组。<br>示例值： group |
| ConsumerName | 是 | 否 | String | 消费组名称。<br>示例值： name |
| ConsumeTime | 是 | 否 | String | 消费时间。<br>示例值： 2021-11-02 17:53:00,12,654 |
| ConsumerAddr | 是 | 否 | String | 消费者客户端地址。<br>示例值： 1.1.1.1 |
| ConsumeUseTime | 是 | 否 | Uint64 | 消费耗时（毫秒）。<br>示例值： 12 |
| Status | 是 | 否 | String | 消费状态。<br>示例值： 0 |

## DeploymentUnitHealthStatusResult

DeploymentUnitHealthStatus返回值

被如下接口引用：DescribeDeploymentUnitHealthStatus

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Enabled | 否 | 是 | Bool | 是否启用单元化<br>示例值： 无 |
| Data | 否 | 是 | String | 扩展信息<br>示例值： 无 |

## Subscription

订阅者

被如下接口引用：DescribeSubscriptions

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| TopicName | 是 | 否 | String | 主题名称。<br>示例值： devTopic |
| EnvironmentId | 是 | 否 | String | 环境（命名空间）名称。<br>示例值： devNs |
| ConnectedSince | 是 | 是 | String | 消费者开始连接的时间。<br>示例值： 2024-09-12T16:01:59.677+08:00 |
| ConsumerAddr | 是 | 是 | String | 消费者地址。<br>示例值： 127.0.0.1 |
| ConsumerCount | 是 | 是 | String | 消费者数量。<br>示例值： 11 |
| ConsumerName | 是 | 是 | String | 消费者名称。<br>示例值： devConsumerName |
| MsgBacklog | 是 | 是 | String | 堆积的消息数量。<br>示例值： 1 |
| MsgRateExpired | 是 | 是 | String | 于TTL，此订阅下没有被发送而是被丢弃的比例。<br>示例值： 1 |

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| MsgRateOut | 是 | 是 | String | 消费者每秒分发消息的数量之和。<br>示例值： 1 |
| MsgThroughputOut | 是 | 是 | String | 消费者每秒消息的byte。<br>示例值： 1 |
| SubscriptionName | 是 | 是 | String | 订阅名称。<br>示例值： devSubscriptionName |
| ConsumerSets | 是 | 是 | Array of Consumer | 消费者集合。<br>示例值：查看 |
| IsOnline | 是 | 是 | Bool | 是否在线。<br>示例值： true |
| ConsumersScheduleSets | 是 | 是 | Array of ConsumersSchedule | 消费进度集合。<br>示例值：查看 |
| Remark | 是 | 是 | String | 备注。<br>示例值： devRemark |
| CreateTime | 是 | 是 | Datetime | 创建时间。<br>示例值： 2023-10-05 14:30:00 |
| UpdateTime | 是 | 是 | Datetime | 最近修改时间。<br>示例值： 2023-12-05 14:30:00 |
| BlockedSubscriptionOnUnackedMsgs | 否 | 是 | Bool | 是否由于未 ack 数到达上限而被 block 注意：此字段可能返回 null，表示取不到有效值。<br>示例值： true |
| SubType | 否 | 是 | String | 订阅类型，Exclusive，Shared，Failover， Key_Shared，空或NULL表示未知， 注意：此字段可能返回 null，表示取不到有效值。<br>示例值： Shared |
| MaxUnackedMsgNum | 否 | 是 | Int64 | 未 ack 消息数上限 注意：此字段可能返回 null，表示取不到有效值。<br>示例值： 5000 |

## RouterTencentMQTTSource

待复制的 MQTT 源集群信息

被如下接口引用：CreateRocketMQRouterRule、DeleteRocketMQRouterRule、DescribeRocketMQRouterRuleDetail、DescribeRocketMQRouterRules、UpdateRocketMQRouterRule

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| SourceConsumeConcurrentThreadCount | 否 | 是 | Int64 | 消费者并发数，0 ~ 100<br>示例值： 0 |
| FilterFromTimestampMs | 否 | 是 | Int64 | 过滤时间，毫秒时间戳<br>示例值： 0 |
| Endpoint | 否 | 是 | String | MQTT集群接入点信息<br>示例值： http://mqtt-sk3bsed7.com |
| ClusterId | 是 | 是 | String | MQTT 集群ID<br>示例值： mqtt-demo123 |
| Topic | 是 | 是 | String | 主题名<br>示例值： topic_test |
| SourceRegion | 是 | 是 | String | 地域<br>示例值： ap-guangzhou |

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| UserName | 是 | 是 | String | 用户名<br>示例值： admin |
| Password | 否 | 是 | String | 用户密码<br>示例值： test-qwe |

## AMQPClusterConfig

AMQP集群配置

被如下接口引用：DescribeAMQPCluster、DescribeAMQPClusters

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| MaxTpsPerVHost | 是 | 否 | Uint64 | 单Vhost TPS上限<br>示例值： 8000 |
| MaxConnNumPerVHost | 是 | 否 | Uint64 | 单Vhost客户端连接数上限<br>示例值： 8000 |
| MaxVHostNum | 是 | 否 | Uint64 | 最大Vhost数量<br>示例值： 10 |
| MaxExchangeNum | 是 | 否 | Uint64 | 最大exchange数量<br>示例值： 100 |
| MaxQueueNum | 是 | 否 | Uint64 | 最大Queue数量<br>示例值： 100 |
| MaxRetentionTime | 是 | 否 | Uint64 | 消息最大保留时间，以毫秒为单位<br>示例值： 100000 |
| UsedVHostNum | 是 | 否 | Uint64 | 已使用Vhost数量<br>示例值： 3 |
| UsedExchangeNum | 是 | 否 | Uint64 | 已使用exchange数量<br>示例值： 3 |
| UsedQueueNum | 是 | 否 | Uint64 | 已使用queue数量<br>示例值： 8 |

## EnvironmentRoleSet

角色批量绑定命名空间信息

被如下接口引用：CreateRole、ModifyRole

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| EnvironmentId | 是 | 否 | String | 命名空间Id<br>示例值： devNs |
| Permissions | 是 | 否 | Array of String | 权限枚举值<br>示例值： ["produce"] |
| ClusterId | 否 | 是 | String | 绑定的集群Id<br>示例值： pulsar-jmw4o94a3z82 |
| RoleName | 否 | 是 | String | 角色名字<br>示例值： role-a |

## DeliveryInfoOpt

配置下发状态信息

被如下接口引用：DescribeDeliveryStatusOpt

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| DeliveryIp | 是 | 否 | String | 代理节点IP<br>示例值： 10.73.13.97 |
| DeliveryTimes | 是 | 是 | Int64 | 下发时间<br>示例值： 1 |
| DeliveryStatus | 是 | 否 | Int64 | 下发状态（实际大小1字节） 0 下发中 1 陈工 2失败<br>示例值： 2 |
| ConfigVersion | 是 | 否 | String | 版本<br>示例值： 1 |
| AddTime | 是 | 否 | Int64 | 新增时间<br>示例值： 1111 |

# Gateway

单元化网关

被如下接口引用：NotifyDeploymentUnitArchChanged

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| BusinessSystemId | 否 | 否 | String | 业务系统 ID<br>示例值： 无 |
| GatewayId | 否 | 否 | String | 网关 ID<br>示例值： 无 |
| Vip | 否 | 否 | String | 网关 vip<br>示例值： 无 |
| Port | 否 | 否 | String | 网关端口<br>示例值： 无 |

# RocketMQConsumerConnection

在线消费者情况

被如下接口引用：DescribeRocketMQConsumerConnections

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Language | 是 | 否 | String | 消费者应用的语言版本<br>示例值： JAVA |
| Accumulative | 是 | 否 | Int64 | 消息堆积量<br>示例值： 100 |
| Version | 是 | 否 | String | 消费端版本<br>示例值： V4.9.7 |
| ClientId | 是 | 否 | String | 消费者实例ID<br>示例值： rocketmq-2p9vx3ax9jxg |
| ClientAddr | 是 | 否 | String | 消费者实例的地址和端口<br>示例值： 192.168.0.1:5555 |

# RocketMQSubscription

RocketMQ消费组订阅信息

被如下接口引用：DescribeRocketMQSubscriptions、DescribeRocketMQTopic、DescribeRocketMQTopics

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| ConsumerGroup | 否 | 是 | String | 消费组名称<br>示例值： test_group |
| ClientProtocol | 否 | 是 | String | 协议类型<br>示例值： TCP/HTTP |
| Type | 是 | 是 | String | 主题类型：<br>Normal 普通,<br>GlobalOrder 全局顺序,<br>PartitionedOrder 局部顺序,<br>Transaction 事务消息,<br>DelayScheduled 延时消息,<br>Retry 重试,<br>DeadLetter 死信<br>示例值： DeadLetter |
| IsOnline | 否 | 是 | Bool | 是否在线<br>示例值： false |
| ConsumeType | 否 | 是 | Int64 | 消费类型<br>0: 广播消费<br>1: 集群消费<br>示例值： 1 |
| Topic | 是 | 否 | String | 主题名称<br>示例值： test_topic |
| ConsumerLag | 是 | 是 | Int64 | 消费堆积数量<br>示例值： 260 |
| ClusterId | 否 | 是 | String | 实例ID<br>示例值： rocketmq-2p9vx3ax9jxg |
| ClientSubscriptionInfos | 否 | 是 | Array of ClientSubscriptionInfo | 客户端订阅详情<br>示例值：查看 |
| Consistency | 否 | 是 | Int64 | 订阅一致性<br>示例值： 1 |
| LastUpdateTime | 否 | 是 | Int64 | 最后消费进度更新时间，秒为单位<br>示例值： 1727420571 |
| MaxRetryTimes | 否 | 是 | Int64 | 最大重试次数<br>示例值： 10 |
| PartitionNum | 是 | 是 | Int64 | 分区数<br>示例值： 3 |
| ExpressionType | 是 | 是 | String | 过滤模式，TAG，SQL<br>示例值： TAG |
| SubString | 是 | 是 | String | 过滤表达式<br>示例值： test_tag |
| Status | 是 | 是 | Int64 | 订阅状态：<br>0，订阅关系一致<br>1，订阅关系不一致<br>2，未知<br>示例值： 0 |

# RouterTencentMQTTTarget

MQTT 目标集群信息

被如下接口引用：CreateRocketMQRouterRule、DeleteRocketMQRouterRule、DescribeRocketMQRouterRuleDetail、DescribeRocketMQRouterRules、UpdateRocketMQRouterRule

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Endpoint | 否 | 是 | String | MQTT集群接入点信息<br>示例值： http://mqtt-sk3bsed7.com |
| ClusterId | 是 | 是 | String | MQTT 集群ID<br>示例值： mqtt-demo123 |
| Topic | 是 | 是 | String | 主题名<br>示例值： topic_test |
| SourceRegion | 是 | 是 | String | 地域<br>示例值： ap-guangzhou |
| UserName | 是 | 是 | String | 用户名<br>示例值： admin |
| Password | 否 | 是 | String | 用户密码<br>示例值： text-qwe |
| TargetProduceConcurrentThreadCount | 否 | 是 | Int64 | 消费者并发数，0 ~ 100<br>示例值： 0 |
| FilterFromTimestampMs | 否 | 是 | Int64 | 过滤时间，毫秒时间戳<br>示例值： 0 |

## Consumer

消费者

被如下接口引用：DescribeSubscriptions

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| ConnectedSince | 是 | 是 | String | 消费者开始连接的时间。<br>示例值： 2024-04-11T23:03:47.328+08:00 |
| ConsumerAddr | 是 | 是 | String | 消费者地址。<br>示例值： 30.11.19.46:40940 |
| ConsumerName | 是 | 是 | String | 消费者名称。<br>示例值： devName1 |
| ClientVersion | 是 | 是 | String | 消费者版本。<br>示例值： 2.7.1 |
| Partition | 否 | 是 | Int64 | Partition数量<br>示例值： 0 |

## ConsumerLogs

消费信息

被如下接口引用：DescribeMsgTrace

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| TotalCount | 是 | 是 | Uint64 | 记录数。<br>示例值： 1 |
| ConsumerLogSets | 是 | 是 | Array of ConsumerLog | 消费日志。<br>示例值：查看 |

# RocketMQMigrationSubTask

RocketMQ数据迁移子任务信息

被如下接口引用：DescribeRocketMQMigrationTaskDetail

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Name | 是 | 否 | String | 主题或消费组名称<br>示例值： xxxx |
| Code | 是 | 否 | Int64 | 执行结果码：<br>0，成功<br>1，已存在，跳过<br>2，额度已满<br>3，名称不符合规范<br>4，系统错误<br>5，不合法的分区数<br>示例值： 0 |
| Type | 是 | 否 | Int64 | 元数据类型：<br>0，主题<br>1，消费组<br>示例值： 0 |
| Namespace | 是 | 否 | String | 命名空间<br>示例值： xxxx |

# RocketMQTopicDistribution

RocketMQtopic分布情况

被如下接口引用：DescribeRocketMQCluster、DescribeRocketMQClusters

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| TopicType | 是 | 否 | String | topic类型<br>示例值： Normal |
| Count | 是 | 否 | Uint64 | topic数量<br>示例值： 10 |

# InternalRocketMQInstance

RocketMQ实例信息内部展示

被如下接口引用：DescribeInternalRocketMQInstances

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| AdminFeatureEnabled | 否 | 否 | Bool | 是否开启Admin功能<br>示例值： true |
| AppId | 是 | 否 | Uint64 | 实例所属用户APPID<br>示例值： 100000 |
| Uin | 是 | 否 | String | 实例所属用户UIN<br>示例值： 12343444 |
| RateLimit | 是 | 否 | Uint64 | 限流值<br>示例值： 4000 |
| MaxRetention | 是 | 否 | Int64 | 最大可调整消息保留时间，小时为单位<br>示例值： 168 |

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| ClusterId | 是 | 否 | String | 实例所属物理集群<br>示例值： rocketmq-n5degomgrwja |
| MaxQueuesPerTopic | 是 | 否 | Uint64 | 单节点主题最大可调整队列数<br>示例值： 16 |
| IsVIP | 是 | 否 | Bool | 是否专享实例<br>示例值： true |
| MinRetention | 是 | 否 | Int64 | 最小可调整消息保留时间，小时为单位<br>示例值： 24 |
| AclEnabled | 是 | 否 | Bool | 是否开启ACL鉴权<br>示例值： true |
| InstanceId | 是 | 否 | String | 实例ID<br>示例值： rocketmq-n5degomgrwja |
| PayStatus | 是 | 否 | String | 实例付费状态，OK正常，ISOLATED隔离中，DESTROYED已销毁<br>示例值： OK |
| MaxNamespaces | 是 | 否 | Uint64 | 最大可创建命名空间数量<br>示例值： 1 |
| MaxGroups | 是 | 否 | Uint64 | 最大可创建消费组数量<br>示例值： 100 |
| Retention | 是 | 否 | Int64 | 消息保留时间，小时为单位<br>示例值： 72 |
| AutoCreateTopicEnabled | 否 | 否 | Bool | 是否开启自动创建Topic<br>示例值： true |
| Name | 是 | 否 | String | 实例名称<br>示例值： test_name |
| MaxTopics | 是 | 否 | Uint64 | 最大可创建主题数量<br>示例值： 10 |

# RocketMQMsgLog

rocketmq消息日志

被如下接口引用：DescribeRocketMQTopicMsgs

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| DeadLetterResendTimes | 是 | 是 | Int64 | 死信重发次数<br>示例值： 10 |
| ResendSuccessCount | 是 | 是 | Int64 | 死信重发成功次数<br>示例值： 10 |
| MsgId | 是 | 否 | String | 消息id<br>示例值： 092BAE5A1656070DEA4E276DF0760089 |
| MsgTag | 是 | 是 | String | 消息tag<br>示例值： test_tag |
| MsgKey | 是 | 是 | String | 消息key<br>示例值： test_key |
| ProducerAddr | 是 | 否 | String | 客户端地址<br>示例值： 9.43.174.90 |
| ProduceTime | 是 | 否 | String | 消息发送时间<br>示例值： 2023-10-08 15:45:16 |

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| PulsarMsgId | 是 | 否 | String | pulsar消息id<br>示例值： 092BAE5A1656070DEA4E276DF0760089 |

## Extension

单元化组件扩展信息

被如下接口引用：NotifyDeploymentUnitArchChanged

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Data | 否 | 否 | String | 扩展数据<br>示例值： 无 |
| Service | 否 | 否 | String | 组件名称，例如 tsf<br>示例值： 无 |

## RouterMessageServiceTarget

rocketmq router mns 目标信息

被如下接口引用：CreateRocketMQRouterRule、DeleteRocketMQRouterRule、DescribeRocketMQRouterRuleDetail、DescribeRocketMQRouterRules、UpdateRocketMQRouterRule

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| SecretKey | 是 | 是 | String | secret key<br>示例值： - |
| TargetProduceConcurrentThreadCount | 否 | 是 | Int64 | 发送mns的并发数<br>示例值： 1 |
| Topic | 否 | 是 | String | topic名字。可以在https://mns.console.aliyun.com/region/cn-shenzhen/topics 看到<br>示例值： - |
| AccessKey | 是 | 是 | String | access key<br>示例值： - |
| Endpoint | 是 | 是 | String | mns接入点，需要和router内网联通，一般是公网接入点<br>示例值： http://1111111.mns.cn-shenzhen.aliyuncs.com |
| Queue | 否 | 是 | String | mns queue名字。可以在https://mns.console.aliyun.com/region/cn-shenzhen/queues 看到<br>示例值： - |
| SendWithBase64 | 否 | 是 | String | 是否用base64编码发送mns消息。<br>ON：发送base64编码的消息体到mns<br><br>OFF：将原始消息体发送到mns<br>示例值： OFF |

## ConfigDictItemOpt

配置项字典

被如下接口引用：DescribeConfigDictOpt

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| ItemKey | 是 | 否 | String | 配置项Key<br>示例值： clientPort |

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| ItemDefaultValue | 是 | 否 | String | 配置项默认值<br>示例值： 2181 |
| ItemDesc | 是 | 否 | String | 配置项描述<br>示例值： Item |
| IsDynamic | 是 | 否 | Bool | 是否动态生成<br>示例值： false |

## AMQPExchange

AMQP Exchange信息

被如下接口引用：DescribeAMQPExchanges

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Name | 是 | 否 | String | Exchange名称<br>示例值： test |
| Type | 是 | 否 | String | Exchange的类别，为枚举类型:Direct, Fanout, Topic<br>示例值： Default |
| SourceBindedNum | 是 | 否 | Uint64 | 主绑定数<br>示例值： 1 |
| Remark | 是 | 是 | String | 说明<br>示例值： 备注 |
| DestBindedNum | 是 | 否 | Uint64 | 被绑定数<br>示例值： 1 |
| CreateTime | 是 | 否 | Uint64 | 创建时间，以毫秒为单位<br>示例值： 100000 |
| UpdateTime | 是 | 否 | Uint64 | 创建时间，以毫秒为单位<br>示例值： 100000 |
| Internal | 是 | 否 | Bool | 是否为内部Exchange(以amq.前缀开头的)<br>示例值： false |

## RocketMQClusterRecentStats

RocketMQ近期使用量

被如下接口引用：DescribeRocketMQCluster

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| TopicNum | 是 | 否 | Uint64 | Topic数量<br>示例值： 20 |
| ApiRequestNum | 是 | 否 | Uint64 | API请求数<br>示例值： 538000 |
| ProducedMsgNum | 是 | 否 | Uint64 | 消息生产数<br>示例值： 11000 |
| ConsumedMsgNum | 是 | 否 | Uint64 | 消息消费数<br>示例值： 11200 |
| AccumulativeMsgNum | 是 | 否 | Uint64 | 消息堆积数<br>示例值： 22000 |

# ResourceProjectInfo

新增或修改资源对应的项目，新增时 oldProjectid 为空，修改时oldProjectId 为当前资源对应的 projectId

被如下接口引用：TpoModifyResourceProject

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| ResourceId | 是 | 否 | String | 资源id<br>示例值： pulasr-11123 |
| OldProjectId | 否 | 否 | String | 旧项目id<br>示例值： pr-106602ce |
| NewProjectId | 是 | 否 | String | 新项目id<br>示例值： pr-106602ce7 |

# RocketMQGroup

RocketMQ消费组信息

被如下接口引用：DescribeRocketMQConsumerConnections、DescribeRocketMQGroups

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Name | 是 | 否 | String | 消费组名称<br>示例值： test_group |
| TotalAccumulative | 是 | 否 | Int64 | 总堆积数量<br>示例值： 380 |
| ClientProtocol | 是 | 否 | String | 客户端协议<br>示例值： tcp |
| RetryMaxTimes | 是 | 是 | Uint64 | 重试次数<br>示例值： 16 |
| SubscribeTopicNum | 否 | 否 | Int64 | 订阅的主题个数<br>示例值： 10 |
| ConsumerNum | 是 | 否 | Uint64 | 在线消费者数量<br>示例值： 2 |
| TPS | 是 | 否 | Uint64 | 消费TPS<br>示例值： 100 |
| ConsumptionMode | 是 | 否 | Int64 | 0表示集群消费模式，1表示广播消费模式，-1表示未知<br>示例值： 0 |
| RetryPartitionNum | 是 | 是 | Uint64 | 重试队列分区数<br>示例值： 2 |
| UpdateTime | 是 | 否 | Uint64 | 修改时间，以毫秒为单位<br>示例值： 1729478721 |
| Remark | 是 | 是 | String | 说明信息<br>示例值： 备注信息 |
| ConsumerType | 是 | 是 | String | 消费者类型，枚举值ACTIVELY, PASSIVELY<br>示例值： PUSH |
| BroadcastEnabled | 是 | 否 | Bool | 是否开启广播消费<br>示例值： false |
| ReadEnabled | 是 | 否 | Bool | 是否允许消费<br>示例值： true |
| CreateTime | 是 | 否 | Uint64 | 创建时间，以毫秒为单位<br>示例值： 1729478721 |

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| GroupType | 是 | 是 | String | Group类型<br>示例值： TCP |
| InstanceId | 是 | 是 | String | 实例ID<br>示例值： rocketmq-4k4orqgq |
| Namespace | 是 | 是 | String | 命名空间<br>示例值： test_namespace |
| TagList | 否 | 否 | Array of Tag | 绑定的标签列表<br>示例值：查看 |

## BindCluster

用户专享集群信息

被如下接口引用：DescribeBindClusters

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| ClusterId | 是 | 否 | Int64 | 物理集群的ID<br>示例值： pulsar-mw4qmgaq298g |
| ClusterName | 是 | 否 | String | 物理集群的名称<br>示例值： devClusterName |

## AMQPVHost

vhostd信息

被如下接口引用：DescribeAMQPVHosts

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| VHostId | 是 | 否 | String | 命名空间名称，3-64个字符，只能包含字母、数字、“-”及“_”<br>示例值： prod |
| MsgTtl | 是 | 否 | Uint64 | 未消费消息的保留时间，以毫秒单位，范围60秒到15天<br>示例值： 100000 |
| Remark | 是 | 是 | String | 备注<br>示例值： 备注 |
| CreateTime | 是 | 否 | Uint64 | 创建时间，以毫秒为单位<br>示例值： 100000 |
| UpdateTime | 是 | 否 | Uint64 | 更新时间，以毫秒为单位<br>示例值： 100000 |
| Username | 是 | 否 | String | 用户名<br>示例值： xxxxx |
| Password | 是 | 否 | String | 密码<br>示例值： xxxxxx |

## VpcBindRecord

vcp绑定记录

被如下接口引用：DescribeBindVpcs

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| UniqueVpcId | 是 | 否 | String | 租户Vpc Id<br>示例值： vpc-8jjiuayds |
| UniqueSubnetId | 是 | 否 | String | 租户Vpc子网Id<br>示例值： subnet-1i12ofyz |
| RouterId | 是 | 否 | String | 路由Id<br>示例值： 124323 |
| Ip | 是 | 否 | String | Vpc的Id<br>示例值： 10.0.0.120 |
| Port | 是 | 否 | Uint64 | Vpc的Port<br>示例值： 8080 |
| Remark | 是 | 是 | String | 说明，128个字符以内<br>示例值： devRemark |

## VpcConfig

VPC配置信息

被如下接口引用：DescribeRocketMQCluster、DescribeRocketMQClusters

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| VpcId | 是 | 否 | String | vpc的id<br>示例值： vpc-9dlrd5h1 |
| SubnetId | 是 | 否 | String | 子网id<br>示例值： subnet-jadmas |

## ClientSubscriptionInfo

客户端订阅详情，可用于辅助判断哪些客户端订阅关系不一致

被如下接口引用：DescribeRocketMQSubscriptions、DescribeRocketMQTopic、DescribeRocketMQTopics

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| SubString | 是 | 是 | String | 订阅表达式<br>示例值： test_tag |
| ExpressionType | 是 | 是 | String | 订阅方式<br>示例值： TAG |
| ClientId | 是 | 否 | String | 客户端ID<br>示例值： rocketmq-2p9vx3ax9jxg |
| ClientAddr | 是 | 是 | String | 客户端IP<br>示例值： 1.1.1.12 |
| Topic | 是 | 是 | String | 订阅主题<br>示例值： test_topic |

## FilterSubscription

过滤订阅列表

被如下接口引用：DescribeSubscriptions

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| ConsumerHasCount | 否 | 否 | Bool | 是否仅展示包含真实消费者的订阅。<br>示例值： false |
| ConsumerHasBacklog | 否 | 否 | Bool | 是否仅展示消息堆积的订阅。<br>示例值： false |
| ConsumerHasExpired | 否 | 否 | Bool | 是否仅展示存在消息超期丢弃的订阅。<br>示例值： false |
| SubscriptionNames | 否 | 否 | Array of String | 按照订阅名过滤，精确查询。<br>示例值： false |

## EnvironmentRole

环境角色集合

被如下接口引用：DescribeEnvironmentRoles、DescribeRocketMQEnvironmentRoles

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| CreateTime | 是 | 否 | Datetime | 创建时间。<br>示例值： 2023-07-18 17:06:57 |
| UpdateTime | 是 | 否 | Datetime | 更新时间。<br>示例值： 2023-07-18 17:06:57 |
| EnvironmentId | 是 | 否 | String | 环境（命名空间）。<br>示例值： test_namespace |
| RoleName | 是 | 否 | String | 角色名称。<br>示例值： test_role_name |
| Permissions | 是 | 否 | Array of String | 授权项，最多只能包含produce、consume两项的非空字符串数组。<br>示例值： ["produce"] |
| RoleDescribe | 是 | 否 | String | 角色描述。<br>示例值： 测试角色 |

## RocketMQVipInstance

RocketMQ专享实例信息

被如下接口引用：DescribeRocketMQVipInstances

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| Retention | 是 | 否 | Int64 | 实例消息保留时间，小时为单位<br>示例值： 72 |
| InstanceId | 是 | 否 | String | 实例id<br>示例值： rmq-4k4orqgq1 |
| ConfigDisplay | 是 | 否 | String | 实例配置规格名称<br>示例值： 基础型 |
| MaxStorage | 是 | 否 | Uint64 | 存储容量，GB为单位<br>示例值： 100 |
| Remark | 是 | 否 | String | 备注信息<br>示例值： remark-info |
| SpecName | 是 | 否 | String | 实例配置ID<br>示例值： vip-basic-1 |

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| CanVerticalChange | 是 | 否 | Bool | 是否可以垂直变配<br>示例值： false |
| InstanceName | 是 | 否 | String | 实例名称<br>示例值： order-instance |
| MaxBandWidth | 是 | 否 | Uint64 | 峰值带宽，Mbps为单位<br>示例值： 1200 |
| AclEnabled | 是 | 否 | Bool | 是否开启ACL鉴权<br>示例值： true |
| Status | 是 | 否 | Uint64 | 实例状态，0表示创建中，1表示正常，2表示隔离中，3表示已销毁，4 - 异常, 5 - 发货失败，6 - 变配中，7 - 变配失败<br>示例值： 0 |
| ExpireTime | 是 | 否 | Uint64 | 实例到期时间，毫秒为单位<br>示例值： 100000 |
| AutoRenewFlag | 是 | 否 | Uint64 | 自动续费标记，0表示默认状态(用户未设置，即初始状态即手动续费)， 1表示自动续费，2表示明确不自动续费(用户设置)<br>示例值： 0 |
| PayMode | 是 | 否 | Uint64 | 0-后付费，1-预付费<br>示例值： 0 |
| DestroyTime | 否 | 否 | Uint64 | 销毁时间<br>示例值： 369182637452345 |
| InstanceVersion | 是 | 否 | String | 实例版本<br>示例值： 4.9.3 |
| NodeCount | 是 | 否 | Uint64 | 节点数量<br>示例值： 5 |
| MaxTps | 是 | 否 | Uint64 | 峰值TPS<br>示例值： 10000 |
| MaxRetention | 是 | 否 | Int64 | 最大可设置消息保留时间，小时为单位<br>示例值： 168 |
| MinRetention | 是 | 否 | Int64 | 最小可设置消息保留时间，小时为单位<br>示例值： 24 |

# RouterRocketMQTarget

rocketmq router任务的目标信息

被如下接口引用：CreateRocketMQRouterRule、DeleteRocketMQRouterRule、DescribeRocketMQRouterRuleDetail、DescribeRocketMQRouterRules、UpdateRocketMQRouterRule

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| TargetProduceConcurrentThreadCount | 否 | 是 | Int64 | 生产者并发数，可以不用填，可以在规则创建后修改<br>示例值： 1 |
| AccessKey | 是 | 是 | String | access key<br>示例值： - |
| NameServerAddressList | 是 | 是 | String | namesrv地址列表，也可以是云厂商提供的接入点信息<br>示例值： 127.0.0.1:9876 |
| Namespace | 否 | 是 | String | 4.X的命名空间，没有使用则留空，5.X也留空。 naemspace 需要包含MQ_INST那部分. 比如：MQ_INST_rocketmqka57d53rnn5_tiger_namespace<br>示例值： - |

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| SecretKey | 是 | 是 | String | secret key<br>示例值： - |
| Topic | 是 | 是 | String | topic名字<br>示例值： - |

## EditableSource

RocketMQ Router可编辑的规则源属性

被如下接口引用：UpdateRocketMQRouterRule

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| AccessKey | 否 | 否 | String | 新的access key<br>示例值： - |
| SecretKey | 否 | 否 | String | 新的secret key<br>示例值： - |
| NewFilterExpression | 否 | 否 | String | 新的RocketMQ过滤规则表达式<br>示例值： - |
| NewFilterFromTimestampMs | 否 | 否 | String | 新的过滤时间，毫秒级时间戳<br>示例值： - |

## RetentionPolicy

消息保留策略

被如下接口引用：CreateEnvironment、DescribeEnvironments、ModifyEnvironmentAttributes

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| TimeInMinutes | 是 | 否 | Int64 | 消息保留时长<br>示例值： 0 |
| SizeInMB | 是 | 否 | Int64 | 消息保留大小<br>示例值： 0 |

## RouterTencentRocketMQTarget

rocketmq router腾讯云目标

被如下接口引用：CreateRocketMQRouterRule、DeleteRocketMQRouterRule、DescribeRocketMQRouterRuleDetail、DescribeRocketMQRouterRules、UpdateRocketMQRouterRule

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| ClusterId | 是 | 是 | String | rocketmq实例id<br>示例值： rocketmq-xxxxx |
| TargetRegion | 否 | 是 | String | 目标region<br>示例值： ap-beijing |
| Namespace | 否 | 是 | String | 4.X的命名空间，没有使用则留空，5.X也留空。 naemspace 需要包含MQ_INST那部分. 比如：MQ_INST_rocket2mq3ka57d53rnn5_tiger_namespace<br>示例值： - |
| Topic | 是 | 是 | String | topic名字<br>示例值： - |

| 名称 | 必选 | 允许NULL | 类型 | 描述 |
| --- | --- | --- | --- | --- |
| TargetProduceConcurrentThreadCount | 否 | 是 | Int64 | 消费者并发数<br>示例值： 1 |

# 错误码

## 功能说明

如果返回结果中存在 Error 字段，则表示调用 API 接口失败。例如：

```
{
  "Response": {
    "Error": {
      "Code": "AuthFailure.SignatureFailure",
      "Message": "The provided credentials could not be validated. Please check your signature is cor
rect."
    },
    "RequestId": "ed93f3cb-f35e-473f-b9f3-0d451b8b79c6"
  }
}
```

Error 中的 Code 表示错误码，Message 表示该错误的具体信息。

## 错误码列表

公共错误码

| 错误码 | 说明 |
| --- | --- |
| AuthFailure.InvalidSecretId | 密钥非法（不是云 API 密钥类型）。 |
| AuthFailure.MFAFailure | MFA 错误。 |
| AuthFailure.SecretIdNotFound | 密钥不存在。 请在控制台检查密钥是否已被删除或者禁用，如状态正常，请检查密钥是否填写正确，注意前后不得有空格。 |
| AuthFailure.SignatureExpire | 签名过期。Timestamp 和服务器时间相差不得超过五分钟，请检查本地时间是否和标准时间同步。 |
| AuthFailure.SignatureFailure | 签名错误。 签名计算错误，请对照调用方式中的接口鉴权文档检查签名计算过程。 |
| AuthFailure.TokenFailure | token 错误。 |
| AuthFailure.UnauthorizedOperation | 请求未 CAM 授权。 |
| DryRunOperation | DryRun 操作，代表请求将会是成功的，只是多传了 DryRun 参数。 |

| 错误码 | 说明 |
| --- | --- |
| FailedOperation | 操作失败。 |
| InternalError | 内部错误。 |
| InvalidAction | 接口不存在。 |
| InvalidParameter | 参数错误。 |
| InvalidParameterValue | 参数取值错误。 |
| LimitExceeded | 超过配额限制。 |
| MissingParameter | 缺少参数错误。 |
| NoSuchVersion | 接口版本不存在。 |
| RequestLimitExceeded | 请求的次数超过了频率限制。 |
| ResourceInUse | 资源被占用。 |
| ResourceInsufficient | 资源不足。 |
| ResourceNotFound | 资源不存在。 |
| ResourceUnavailable | 资源不可用。 |
| UnauthorizedOperation | 未授权操作。 |
| UnknownParameter | 未知参数错误。 |
| UnsupportedOperation | 操作不支持。 |
| UnsupportedProtocol | http(s)请求协议错误，只支持 GET 和 POST 请求。 |
| UnsupportedRegion | 接口不支持所传地域。 |

## 业务错误码

| 错误码 | 说明 |
| --- | --- |
| FailedOperation.DescribeProducers | 查询生产者数据失败。 |
| FailedOperation.CreateSubscription | 创建订阅关系失败。 |
| ResourceInsufficient | |
| ResourceInUse.Role | |

| 错误码 | 说明 |
| --- | --- |
| FailedOperation.DeleteCluster | |
| InvalidParameter | |
| FailedOperation.CreatePulsarClientError | 创建TDMQ client的出错。 |
| InternalError.Retry | 重试可以成功。 |
| InternalError.CloudAPIError | 云API错误。 |
| FailedOperation.UpdateEnvironmentRole | |
| InvalidParameterValue.InvalidParams | |
| ResourceNotFound.BrokerCluster | |
| LimitExceeded.Clusters | |
| InvalidParameterValue.ClusterNameDuplication | 与现有集群名称重复。 |
| LimitExceeded.Environments | 实例下环境数量超过限制。 |
| InvalidParameter.TenantNotFound | 上传的 tenant name 错误。 |
| FailedOperation.CallTrade | |
| FailedOperation.DeleteTopics | |
| OperationDenied.ConsumerRunning | 订阅仍在消费中。 |
| FailedOperation.NamespaceInUse | 必须先清除关联命名空间才能继续操作。 |
| FailedOperation.CreateSecretKey | |
| FailedOperation.CreateTopic | |
| MissingParameter.NeedMoreParams | |
| InternalError.GetAttributesFailed | 获取属性失败。 |
| ResourceNotFound.MsgProduceLog | 消息生产日志不存在。 |
| ResourceNotFound.Namespace | |
| ResourceInUse.Topic | 重名，主题已存在。 |
| ResourceNotFound.MigrationTask | |
| UnsupportedOperation | |

| 错误码 | 说明 |
| --- | --- |
| LimitExceeded.Topics | 实例下主题数量超过限制。 |
| ResourceNotFound.Msg | 找不到指定消息。 |
| FailedOperation.SetTTL | |
| ResourceUnavailable | |
| LimitExceeded.Namespaces | 实例下命名空间数量超过限额。 |
| FailedOperation.CreateCluster | |
| ResourceInUse.VpcRoute | vpc路由已存在。 |
| FailedOperation.ReceiveTimeout | 接收消息超时，请重试。 |
| FailedOperation.UpdateRole | |
| FailedOperation.ResetMsgSubOffsetByTimestampFailed | 消息回溯设置失败。 |
| FailedOperation.CreateNamespace | 创建命名空间失败。 |
| ResourceUnavailable.FundRequired | |
| InvalidParameterValue.NeedMoreParams | 必要参数没有传递。 |
| FailedOperation.RouterRule | |
| ResourceUnavailable.SystemUpgrade | 系统升级。 |
| UnknownParameter | 未知参数错误。 |
| InvalidParameterValue | 参数取值错误。 |
| ResourceInUse.Namespace | 重名，命名空间已存在。 |
| FailedOperation.CreateEnvironment | 环境创建失败。 |
| FailedOperation.DeleteRoles | |
| ResourceInUse.BindVpc | vpc绑定关系已存在。 |
| InternalError.BrokerService | Broker服务异常。 |
| FailedOperation.DeleteNamespace | 删除命名空间失败。 |
| ResourceNotFound.BindVpc | vpc绑定关系不存在。 |
| FailedOperation.GetTopicPartitionsFailed | 获取主题分区数失败。 |

| 错误码 | 说明 |
| --- | --- |
| InternalError.IllegalMessage | |
| FailedOperation.ReceiveError | 接收消息出错。 |
| UnauthorizedOperation | 未授权操作。 |
| LimitExceeded.Subscriptions | 实例下订阅者数量超过限制。 |
| FailedOperation.CreateRole | |
| ResourceInUse.EnvironmentRole | |
| ResourceNotFound.Subscription | 订阅关系不存在。 |
| InvalidParameter.Partition | |
| ResourceNotFound.Environment | |
| FailedOperation.UpdateTopic | 主题更新失败。 |
| LimitExceeded | |
| MissingParameter | 缺少参数错误。 |
| InternalError.SystemError | |
| FailedOperation.CreateEnvironmentRole | |
| FailedOperation.CreateBindVpc | 创建vpc绑定关系失败。 |
| FailedOperation | |
| ResourceNotFound.Instance | |
| ResourceInUse.SupportRoute | 支撑路由已存在。 |
| ResourceNotFound.Topic | |
| AuthFailure.UnauthorizedOperation | |
| ResourceInUse | |
| FailedOperation.DescribeSubscription | 查询订阅数据失败。 |
| FailedOperation.DeleteSubscriptions | 删除订阅关系失败。 |
| FailedOperation.SendMsgFailed | 发送消息失败。 |
| FailedOperation.CreateProducerError | 创建producer出错。 |

| 错误码 | 说明 |
| --- | --- |
| ResourceUnavailable.CreateFailed | |
| FailedOperation.UpdateEnvironment | 环境更新失败。 |
| InvalidParameterValue.TopicNotFound | 上传的topic name错误。 |
| InvalidParameter.TokenNotFound | 没有获取到正确的 token。 |
| ResourceNotFound | |
| OperationDenied.DefaultEnvironment | 默认环境不允许操作。 |
| FailedOperation.GetEnvironmentAttributesFailed | 获取环境属性失败。 |
| InternalError.VpcService | VPC服务异常。 |
| FailedOperation.TopicInUse | 必须先清除关联主题数据才能继续操作。 |
| InternalError.FailedOperation | |
| UnsupportedOperation.InstanceDowngrade | |
| FailedOperation.VpcInUse | 必须先清除关联VPC路由数据才能继续操作。 |
| InternalError | |
| ResourceNotFound.EnvironmentRole | |
| InvalidParameterValue.TTL | 无效的消息TTL值。 |
| ResourcesSoldOut | 资源售罄。 |
| FailedOperation.DeleteEnvironmentRoles | |
| InvalidParameterValue.AtLeastOne | |
| FailedOperation.DeleteBindVpc | 删除vpc绑定关系失败。 |
| FailedOperation.DeleteEnvironments | 环境删除失败。 |
| ResourceInUse.Cluster | 集群已存在。 |
| ResourceNotFound.Namspace | |
| FailedOperation.SaveSecretKey | |
| ResourceNotFound.Role | |
| ResourceInUse.Subscription | 重名，订阅关系已存在。 |

| 错误码 | 说明 |
| --- | --- |
| FailedOperation.RoleInUse | |
| FailedOperation.InstanceNotReady | |
| ResourceNotFound.Cluster | |