

CMQ Topic Python SDK linux 版安装使用指南

[CMQ Queue python SDK 安装使用指南下载](#)

1. 依赖环境搭建

CMQ python sdk 依赖于 python2.x，目前不支持 python3，因此需要用户首先安装 python2，请从[这里](#)选择 python2 下载安装。也可以在线安装。

在线安装

1. RedHat 系操作系统：RedHat、Centos 等使用如下命令：
`yum install python`
2. Debian 系操作系统：Debian、Ubuntu 等使用如下命令：
`apt-get install python`

2. Python SDK 下载与配置

2.1 [云 api 密钥](#)使用说明

使用 SDK 首先需要用户的云 api 密钥，云 api 密钥是对用户身份的合法性验证。云 api 密钥的方法如下：
登录[腾讯云](#)，输入登陆账号和密码，如下图，点击右上角管理中心，进入管理中心。



在管理中心界面，左上角云产品中打开云 api 密钥选项



即可看到目前正在使用的 secretId 和对应的 secretKey, 这里可以管理密钥: 生成密钥, 启用密钥, 禁用密钥。



2.2 endpoint

endpoint 是使用 CMQ 服务的访问地址, 同时 endpoint 中也包含了使用的协议, 这里对 endpoint 进行一个简单的说明:

endpoint 的格式如下:

内网: `http://cmq-topic-region.api.tencentyun.com`

外网: `http(s)://cmq-topic-region.api.qcloud.com`

2.2.1 region 说明

region 需要使用具体地域进行替换, 有如下三个地区: gz(广州), sh(上海), bj(北京), 划分三个地域, 有助于不同地域的用户就近选择, 提供更好的服务。公共参数中的 region 值要与域名的 region 值保持一致, 如果出现不一致的情况, 以域名的 region 值为准, 将请求发往域名 region 所指定的地域。

2.2.2 内外网区别

如果业务进程也部署在腾讯云的 CVM 子机上, 强烈建议使用同地域的内网 endpoint 原因如下:

- 1) 同地域内网的时延更低;
- 2) 目前消息队列对于公网下行流量是要收取流量费用的, 用内网可以节省这部分的费用。

外网域名请求既支持 http, 也支持 https。内网请求仅支持 http。

举个例子: 如果使用腾讯云北京地区的云主机: 那么建议使用北京地域的 cmq, 这样可以取得较低的时延, 同时使用内网, 可以减少使用费用, 因此 endpoint=`http://cmq-queue-bj.api.tencentyun.com`。

2.3 python SDK 下载

下载最新版 [python sdk](#)。解压到你的工作目录（任意目录），填入上述的 secret id, secret key 和 endpoint。参考 sample 中的例子，就可以利用 CMQ 创建主题，发布消息了。

如果有错误返回，请参考官网[错误码](#)说明排查问题，或在微信交流群上

3.CMQ python SDK 的使用

下面的代码也是 sdk 中的 sample，从创建主题，获取主题属性，发布消息，创建订阅，设置订阅属性，删除订阅，主题等操作演示了整个消息队列主题订阅操作的全过程。

```
import sys
import os
sys.path.insert(0, os.path.dirname(os.path.abspath(__file__)) + "/..")

import time
import logging
from cmq.account import Account
from cmq.cmq_exception import *
from cmq.topic import *
from cmq.subscription import *
# 从腾讯云官网查看云 api 的密钥信息
secretId = '您的 secretId'
secretKey = '您的 secretKey'
# 使用广州地域的消息服务
endpoint = 'http://cmq-queue-gz.api.tencentyun.com'
try:
# 初始化 my_account
# Account 类对象不是线程安全的，如果多线程使用，需要每个线程单独初始化 Account 类对象
    my_account = Account(endpoint, secretId, secretKey, debug=True)
    my_account.set_log_level(logging.DEBUG)
    topic_name = sys.argv[1] if len(sys.argv) > 1 else "Topic-test"
    my_topic = my_account.get_topic(topic_name)
# create topic
    topic_meta = TopicMeta()
    my_topic.create(topic_meta)
# set and get topic meta
    topic_meta.maxMsgSize = 32768
    my_topic.set_attributes(topic_meta)
    topic_meta = my_topic.get_attributes();
# list topic
    totalCount, topicList, next_off = my_account.list_topic();
# public message without tags
```

```

    message = Message()
    message.msgBody = "this is a test message"
    my_topic.publish_message(message)
# public message with tags
    message = Message("this is a test message", ["test", "york"])
    my_topic.publish_message(message)
# batch publish message without tags
    vmsg = []
    for i in range(6):
        message = Message()
        message.msgBody = "this is a test message"
        vmsg.append(message)

    my_topic.batch_publish_message(vmsg)
# batch publish message with tags
    vmsg = []
    for i in range(6):
        message = Message("this is a test message", ["test"])
        vmsg.append(message)
    my_topic.batch_publish_message(vmsg)

# create subscription
    subscription_name = "subsc-test"
    my_sub = my_account.get_subscription(topic_name, subscription_name)
    subscription_meta = SubscriptionMeta()
# please set your endpoint protocol here
    subscription_meta.Endpoint = "your endpoint "
    subscription_meta.Protocol = "http"
    my_sub.create(subscription_meta)
# list SubscriptionList
    totalCount, SubscriptionMetaipctionList, next_off =
my_topic.list_subscription()
# set subscription meta
    subscription_meta = my_sub.get_attributes()
    my_sub.set_attributes(subscription_meta)
# delete sub and topic_name
    my_sub.delete()
    my_topic.delete
except CMQExceptionBase, e:
    print "Exception:%s\n" % e

```

