

CMQ Topic C++ SDK linux 版安装使用指南

[CMQ Queue C++ SDK 安装使用指南下载](#)

依赖环境搭建

CMQ Topic C++ SDK 依赖于 curl 库和 cmake，在使用 C++ SDK 之前，需要配置 curl 库，curl 库有如下安装方式：

curl 安装

在线安装

1. RedHat 系操作系统：RedHat、Centos 等使用如下命令：
yum install curl
2. Debian 系操作系统：Debian、Ubuntu 等使用如下命令：
apt-get install curl libcurl3 libcurl3-dev

离线安装

请从[这里](#)下载，并参考 [libcurl 安装指南](#) 安装，典型的安装方式如下：

```
./configure
```

```
make
```

```
make install
```

如果请求采用 https 方式，安装时需打开 ssl 支持，典型的安装方式如下：

```
./configure --with-ssl=/openssl_path
```

```
make
```

```
make install
```

cmake 安装

在线安装

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

离线安装

请从[这里](#)下载 cmake。典型的安装方式如下：

```
tar xzf cmake-2.8.11.tar.gz
```

```
cd cmake-2.8.11
```

```
./configure
```

```
make
```

make install

2. C++ SDK 下载与配置

2.1 云 api 密钥使用说明

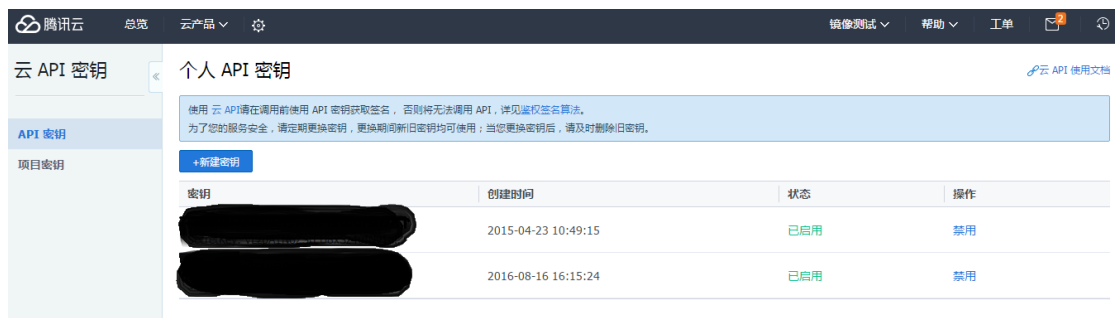
使用 SDK 首先需要用户的云 api 密钥，云 api 密钥是对用户身份的合法性验证。云 api 密钥的方法如下：
登录[腾讯云](#)，输入登陆账号和密码，如下图，点击右上角管理中心，进入管理中心。



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 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 C++ SDK 下载

下载最新版 [c++ 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 C++ SDK 的使用

首先使用 cmake make 生成库文件和 sample

```
./cmake .  
./make
```

下面的代码也 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。

```
// for cmq sdk
```

```

#include "cmq_common.h"
#include <iostream>
#include <vector>
#include <fstream>

using namespace std;
using namespace cmq;

int main()
{
    //从腾讯云官网查询的云API 密钥信息
    string secretId="";
    string secretKey="";
    string endpoint="";
    try
    {
        // create account and topic
        cout<<"Init account"<<endl;
        Account account(endpoint, secretId, secretKey)
        string topicName = "topic-test";
        account.createTopic(topicName, 65536);
        cout<< "create topic " <<endl;
        //get topic meta
        Topic topic = account.getTopic(topicName);
        //cout<< "get topic meta " <<endl;
        TopicMeta topicMeta;
        topic.GetTopicAttributes(topicMeta);
        cout <<"get topic meta " <<endl ;
        cout<< "topic maxMsgSize is " << topicMeta.m_maxMsgSize <<endl;
        cout<<"create subscription"<<endl;
        string subscriptionName="sub-test";
        // your endpoint and protocol
        string endpoint = "";
        string protocol = "";
        vector<string> vfilterTag;
        account.createSubscribe(topicName,subscriptionName,endpoint,protocol);
        cout<<"create subscription"<<endl;
        string msg = "this is a test message";
        string msgId;
        topic.PublishMessage(msg,msgId);
        cout<<"Publish message" <<endl;
        // batch publish message
        vector< string > vmsg;
        for (unsigned int i = 0; i < 5; ++i)

```

```

    {
        string ss;
        ss = "this is a test message ";
        vmsg.push_back(ss);
    }
    vector<string> vmsgId;
    topic.BatchPublishMessage(vmsg, vmsgId);
    cout<< "batch publish message "<<<<endl;
    // list topic
    // create subscription
    vector< pair<string, string > > vTopic;
    int total(0);
    account.listTopic(total, vTopic);
    Subscription sub = account.getSubscription(topicName, subscriptionName);
    //set and get subscription meta
    SubscriptionMeta subMeta;
    sub.getSubscriptionAttributes(subMeta);
    cout<<"get sub meta"<<<<endl ;
    sub.setSubscriptionAttributes(subMeta);
    cout<<"set sub meta"<<<<endl ;
    vector< vector<string> > vsublist;
    account.listSubscription(topicName,total, vsublist);
    account.deleteSubscribe(topicName,subscriptionName);
    account.deleteTopic(topicName);
    cout<<"delete sub and topic"<<<<endl;
}
catch(CMQServerNetworkException &e)
{
    cout<<"Server Network exception,http status: "<<<<e.getStatus()<<<< endl;
    return -1;
}
catch(CMQServerException &e)
{
    cout<<"Server Action failed,code:"<<<<
e.getCode()<<<< ",message:"<<<<e.getMessage()<<<< ",requestId:"<<<<e.getRequestId()<<<< endl;
    return -1;
}
catch(CMQClientException &e)
{
    cout<<"CMQ Client Exceptrion:"<<<<e.what()<<<< endl;
    return -2;
}
}

```