SSL 证书

最佳实践

产品文档

腾讯云

文档目录

最佳实践
多年期证书实现证书签发和资源绑定全自动方案

最近更新时间：2024-03-06 17:49:08

概述

多年期证书是腾讯云 SSL 证书提供的自动审核交付功能，在腾讯云购买1年以上的多年期证书并完成审核后，腾讯云将在前一个 SSL 证书有效期到期前一个月为您自动审核信息并颁发第二张 SSL 证书，无需您重新申请，简化 SSL 证书产品申请时的繁琐流程。
同时，腾讯云 SSL 证书支持云资源托管能力，可自动将新 SSL 证书部署至原 SSL 证书已部署的腾讯云云资源，例如负载均衡、内容分发网络等。
本文档将指导您如何通过两者相结合实现证书签发和资源绑定的全自动交付能力，帮助您实现从多年期证书申请到部署的全自动化。
说明：
本文以 GeoTrust 品牌 OV 型多年期证书、腾讯云云资源以内容分发网络（CDN）为例。

操作步骤

步骤1：购买多年期证书

1. 登录 SSL 证书购买页。
2. 根据您的需求选择并购买支持多年期的 SSL 证书。
3. 完成购买后，您可按照SSL 证书申请流程 完成 SSL 证书的申请。

步骤2：SSL 证书部署至云资源

申请完成证书并颁发后，您可以使用 SSL 证书一键部署功能将证书部署至腾讯云云资源，例如内容分发网络（CDN）。

1. 登录 SSL 证书控制台，选择需部署的多年期证书，单击**部署**。
2. 在弹出的**选择部署类型**窗口中，选择您需部署类型并勾选对应资源实例。
3. 单击**确定**，SSL 证书即可成功部署至对应云资源中。

步骤3：开启云资源托管

1. 单击申请的**证书名称**，进入**证书详情**管理页面。

2. 在**基本信息**模块的云资源托管处，单击**查看**。
3. 在弹出的**云资源托管**窗口中，勾选您需开启的云资源。
4. 单击**确定**，即可完成操作。

苹果ATS特性服务器配置指南

最近更新时间：2024-03-06 17:49:10

注意：

需要配置符合 PFS 规范的加密套餐，目前推荐配置：

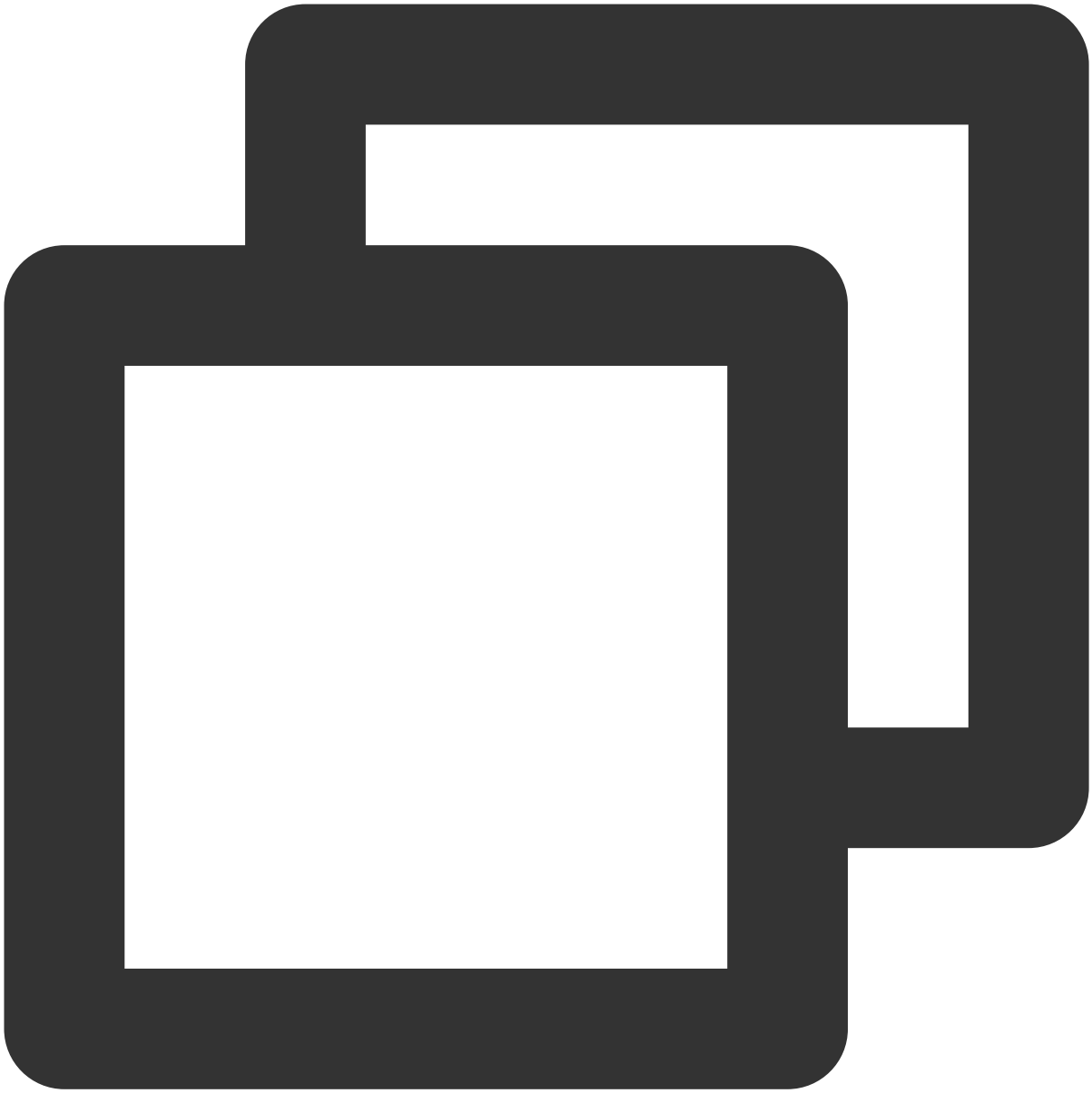
```
ECDHE-RSA-AES128-GCM-SHA256:ECDHE:ECDH:AES:HIGH:!NULL:!aNULL:!MD5:!ADH:!RC4
```

需要在服务端 TLS 协议中启用 TLS1.2，目前推荐配置：

```
TLSv1 TLSv1.1 TLSv1.2
```

Nginx 证书配置

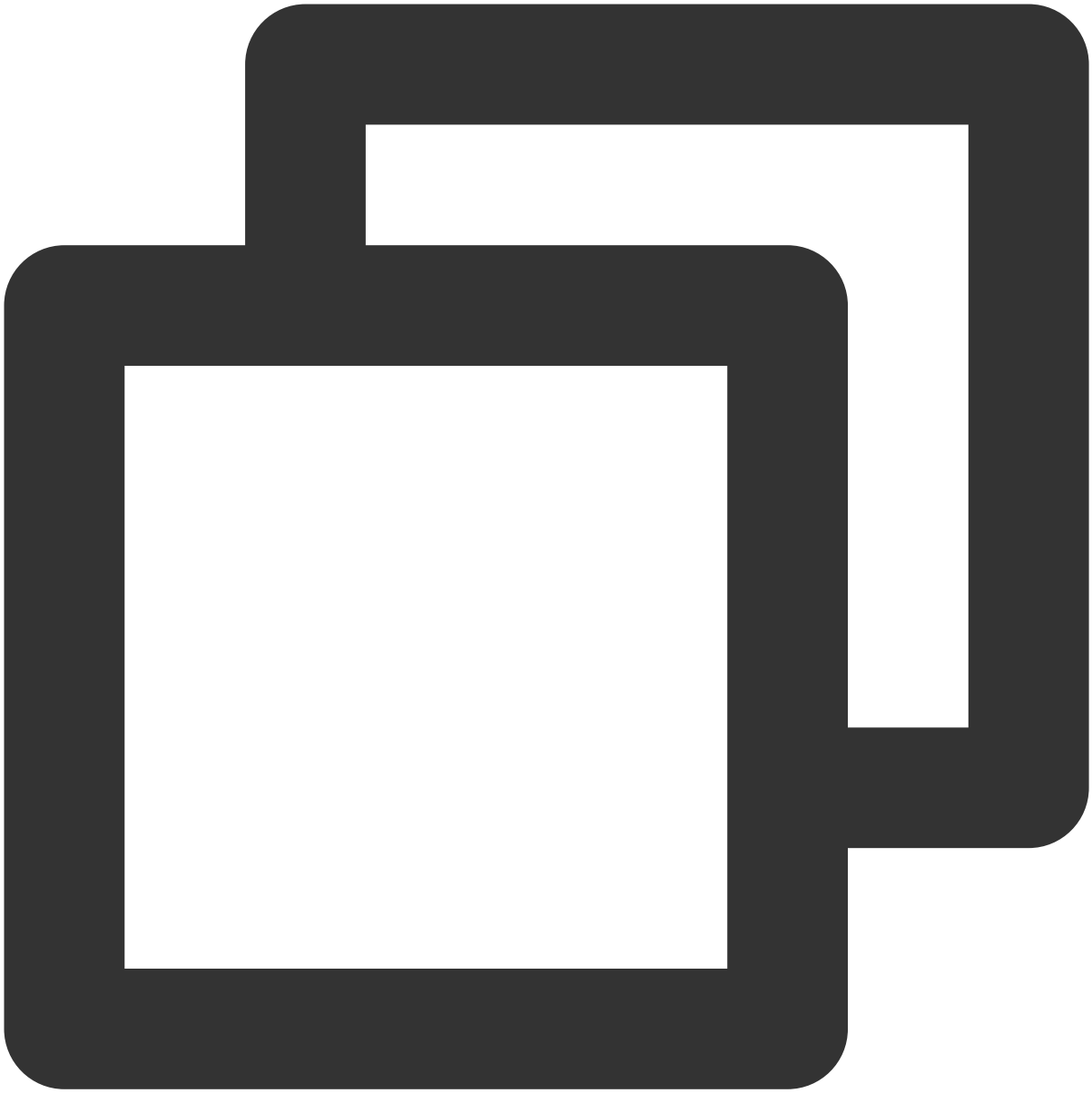
更新 Nginx 根目录下 `conf/nginx.conf` 文件如下：



```
server {
    ssl_ciphers ECDHE-RSA-AES128-GCM-SHA256:ECDHE:ECDH:AES:HIGH:!NULL:!aNULL:!MD5:!
    ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
}
```

Apache 证书配置

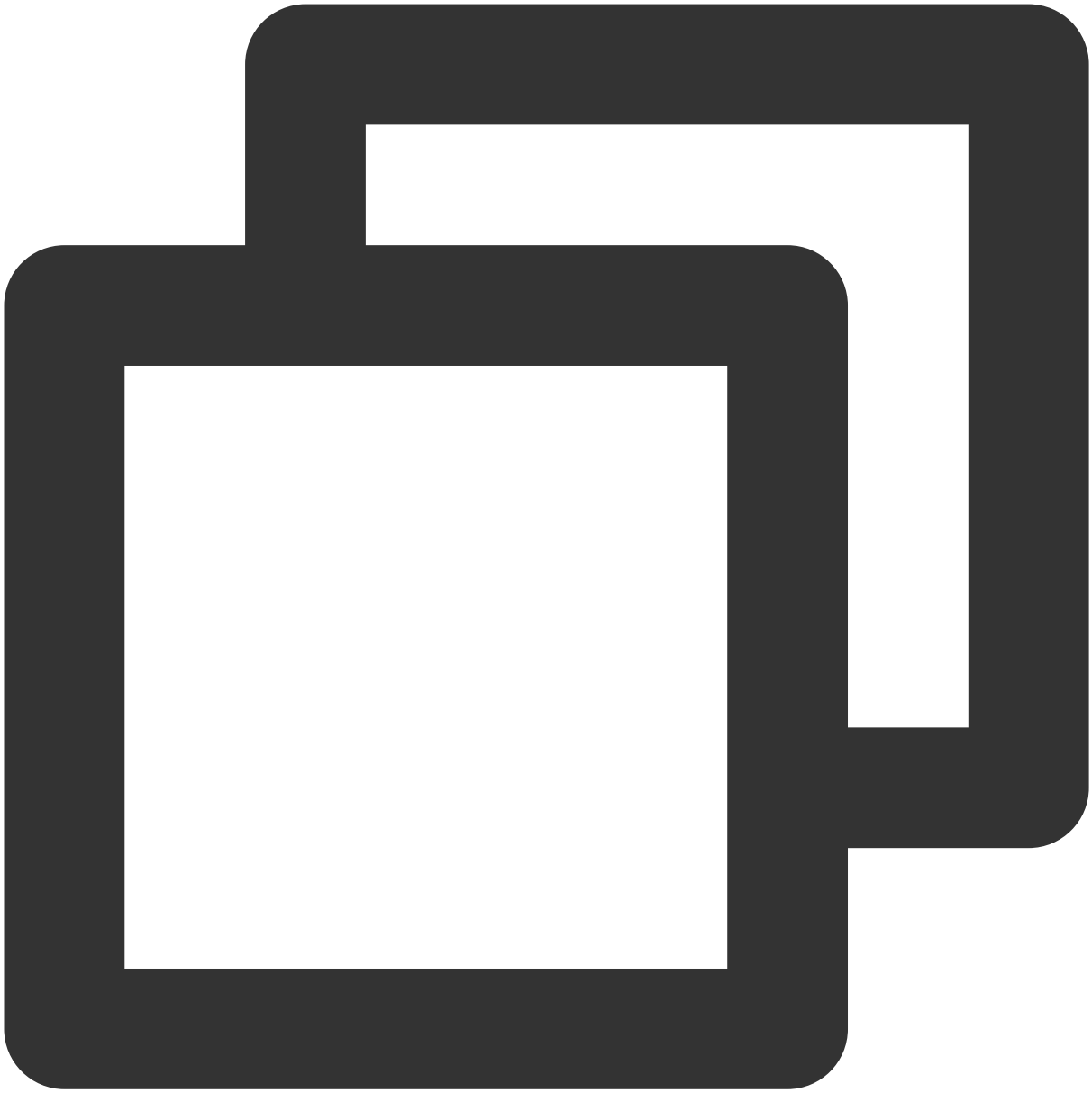
更新 Apache 根目录下 `conf/httpd.conf` 文件如下：



```
<IfModule mod_ssl.c>
        <VirtualHost *:443>
        SSLProtocol TLSv1 TLSv1.1 TLSv1.2
        SSLCipherSuite ECDHE-RSA-AES128-GCM-SHA256:ECDHE:ECDH:AES:HIGH:!NULL:!aNULL
        </VirtualHost>
</IfModule>
```

Tomcat 证书配置

更新 `%TOMCAT_HOME%\\conf\\server.xml` 文件如下：



```
<Connector port="443" protocol="HTTP/1.1" SSLEnabled="true"
    scheme="https" secure="true"
    SSLProtocol="TLSv1+TLSv1.1+TLSv1.2"
    SSLCipherSuite="ECDHE-RSA-AES128-GCM-SHA256:ECDHE:ECDH:AES:HIGH:!NULL:!aNULL:!M
```

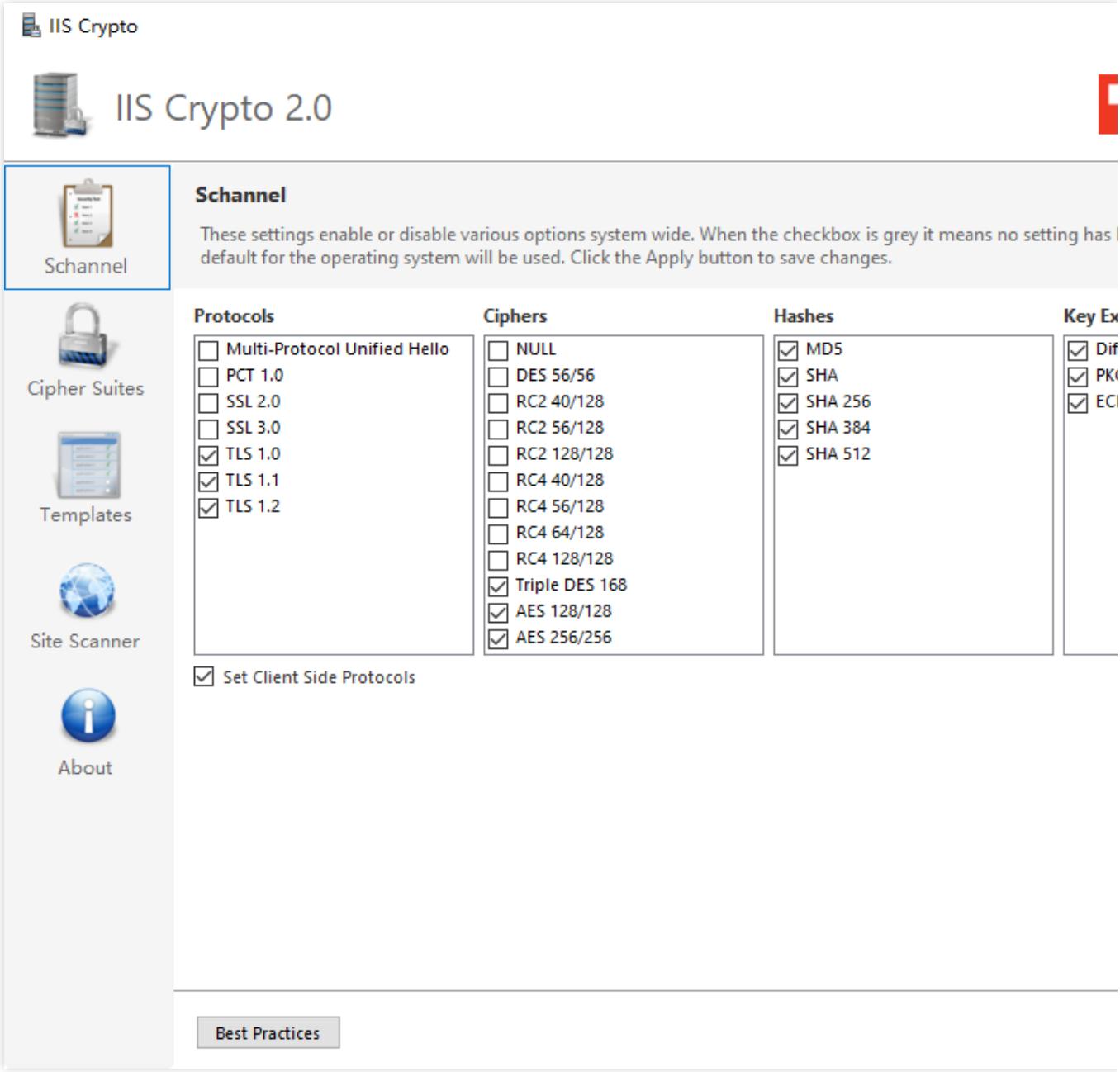
IIS 证书配置

方法一

Windows 2008及更早的版本不支持 TLS1.2 协议，因此无法调整 2008R2 TLS1.2 协议，默认是关闭的，需要启用此协议达到 ATS 要求。

以 2008 R2 为例，导入证书后没有对协议及套件做任何的调整。

证书导入后检测到套件是支持 ATS 需求的，但协议 TLS1.2 没有被启用，ATS 需要 TLS1.2 的支持。可使用的ssltools工具（亚洲诚信提供，单击下载）启用 TLS1.2 协议。如下图所示：



勾选三个 TLS 协议并重启系统即可。

如果检查到 PFS 不支持，在加密套件中选中带 ECDHE 和 DHE 就可以了。

方法二

1. 开始——运行，输入 `regedit` 。

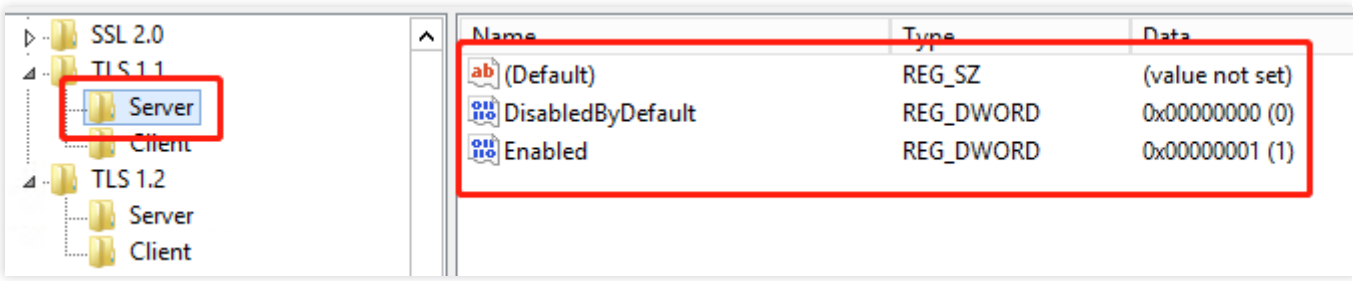
2. 找到 HKEY_LOCAL_MACHINE\\SYSTEM\\CurrentControlSet\\Control\\SecurityProviders\\SCHANNEL\\Protocols 右键->新建->项->新建 TLS 1.1，TLS 1.2。

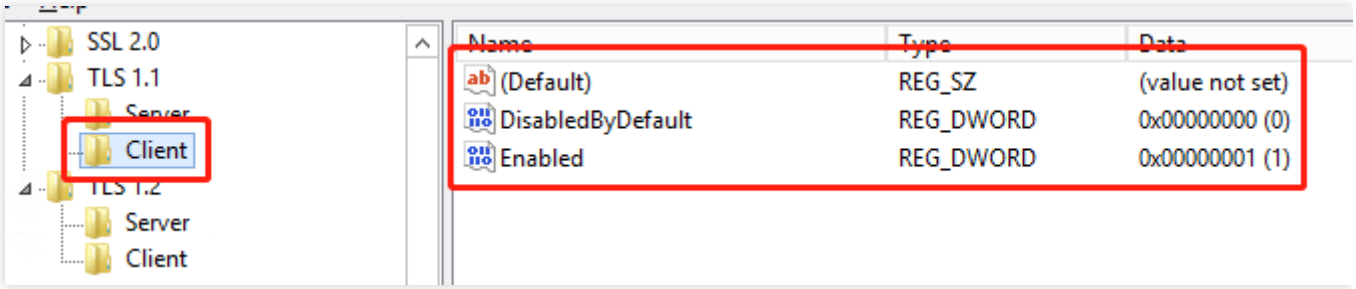
3. TLS 1.1 和 TLS 1.2 右键->新建->项->新建 Server，Client。

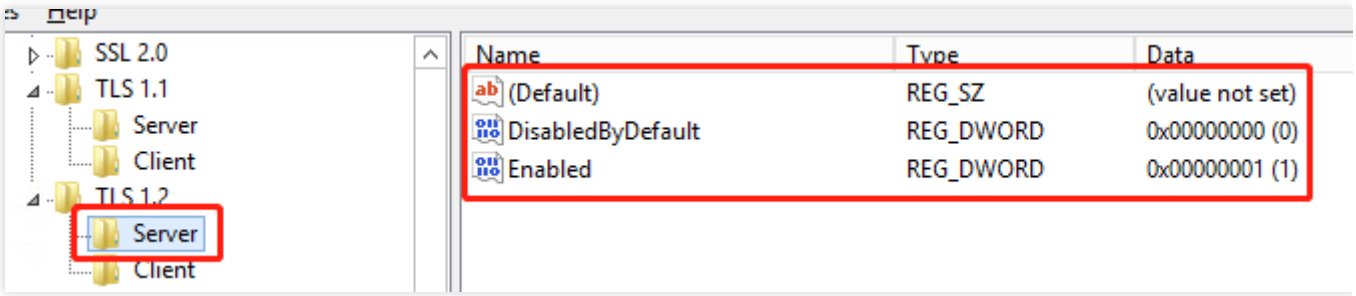
4. 在新建的 Server 和 Client 中新建如下的项（DWORD 32位值），总共4个。如下图所示：

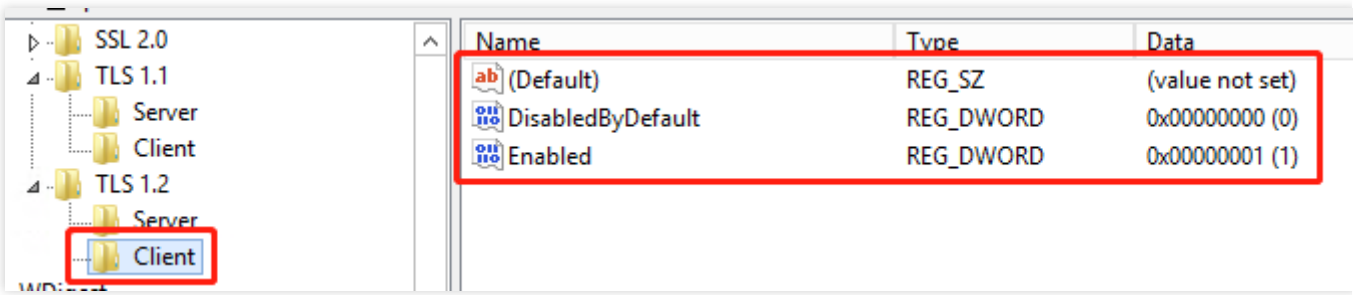
DisabledByDefault [Value = 0]

Enabled [Value = 1]







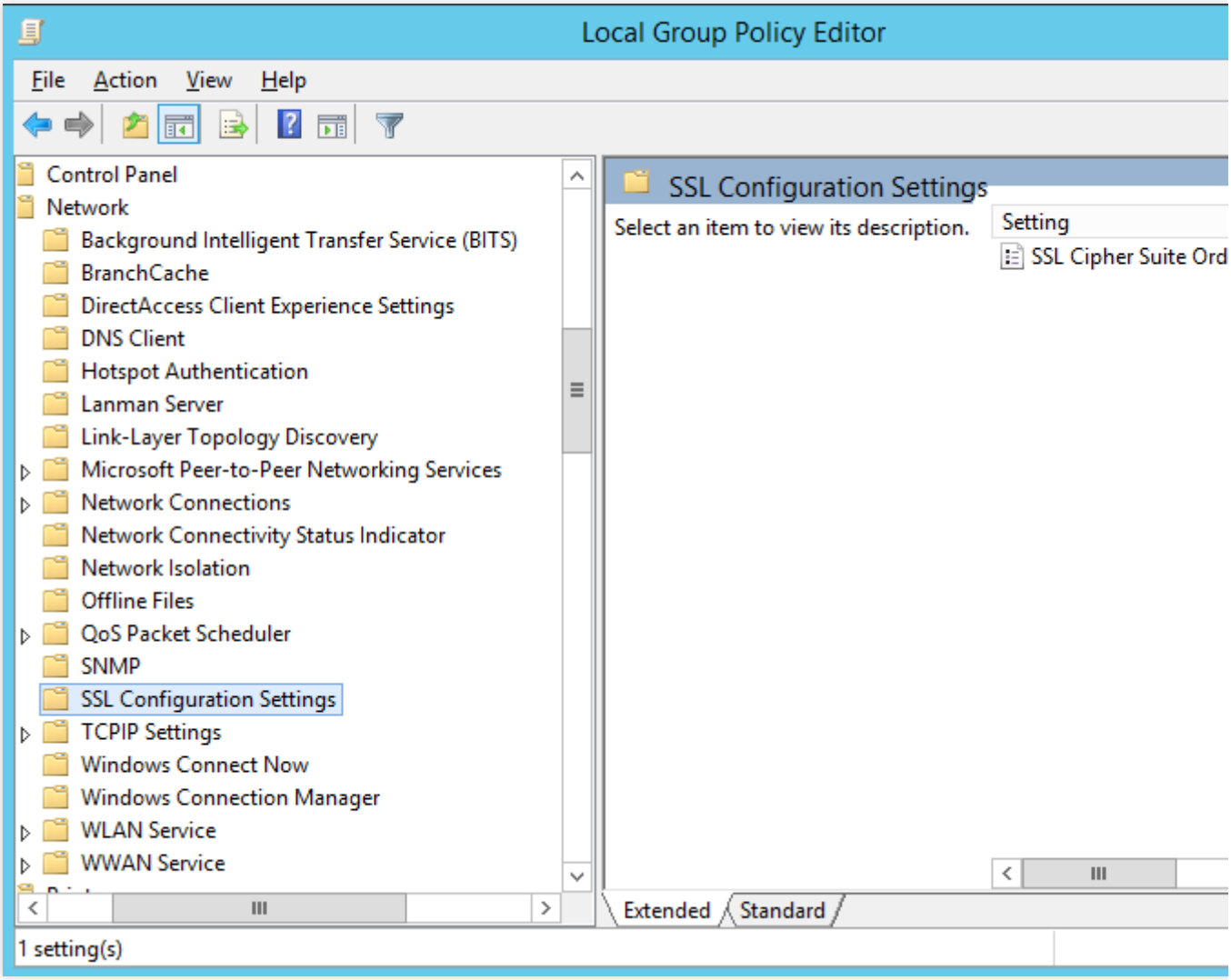


5. 完成后重启系统。

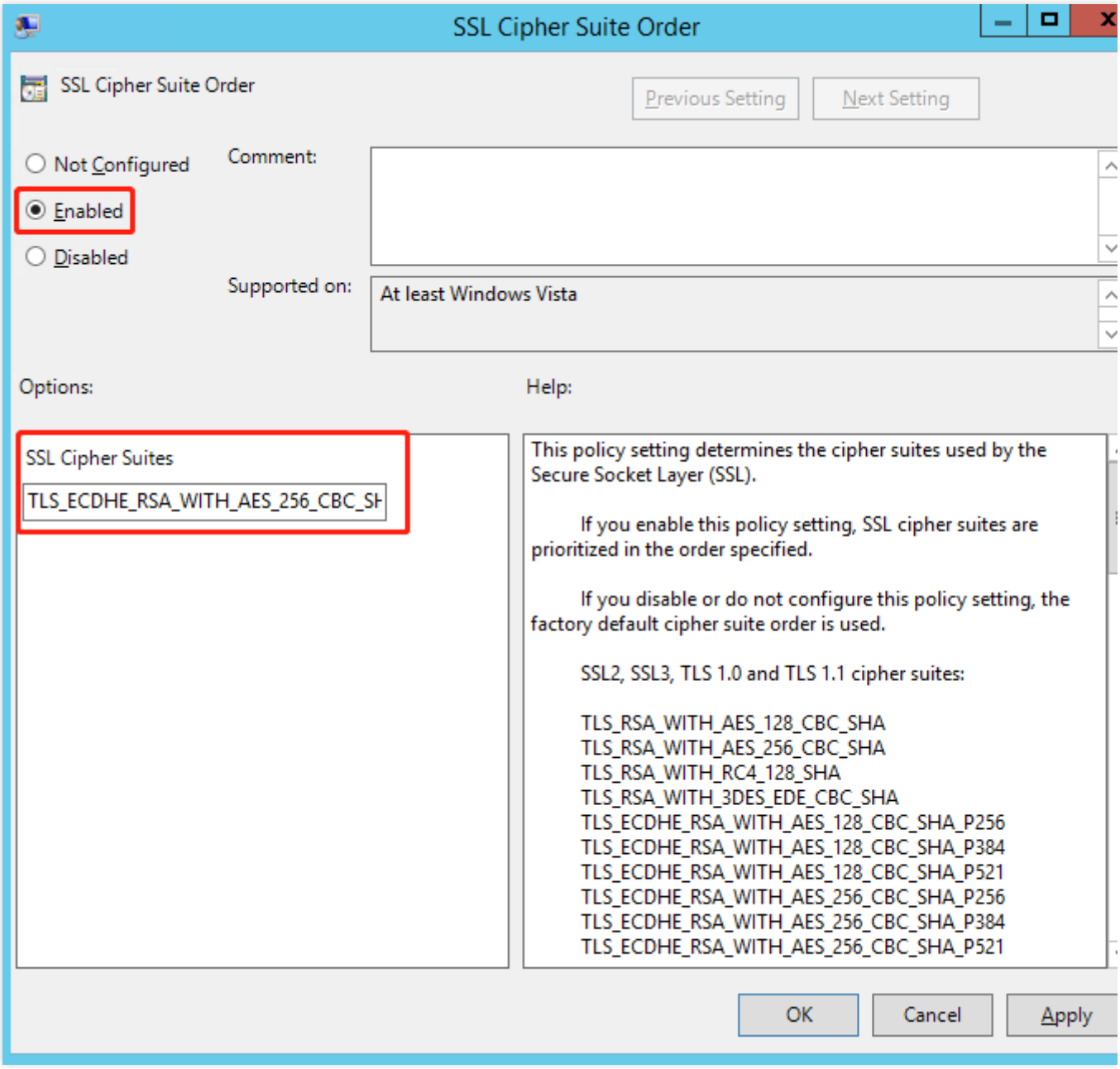
6. 加密套件调整。开始菜单——运行，输入 `gpedit.msc` 进行加密套件调整，在此操作之前需要先开启 TLS1_2 协议。如下图所示：

注意：

对于前向保密加密套件不支持的话可通过组策略编辑器进行调整。



7. 双击 SSL 密码套件顺序，填写如下内容。如下图所示：



设置为 “已启用”。

把支持的 ECDHE 加密套件加入 SSL 密码套件中，以逗号（,）分隔。

填写套件步骤如下：

a. 打开一个空白写字板文档。

b. 复制下图中右侧可用套件的列表并将其粘贴到该文档中。

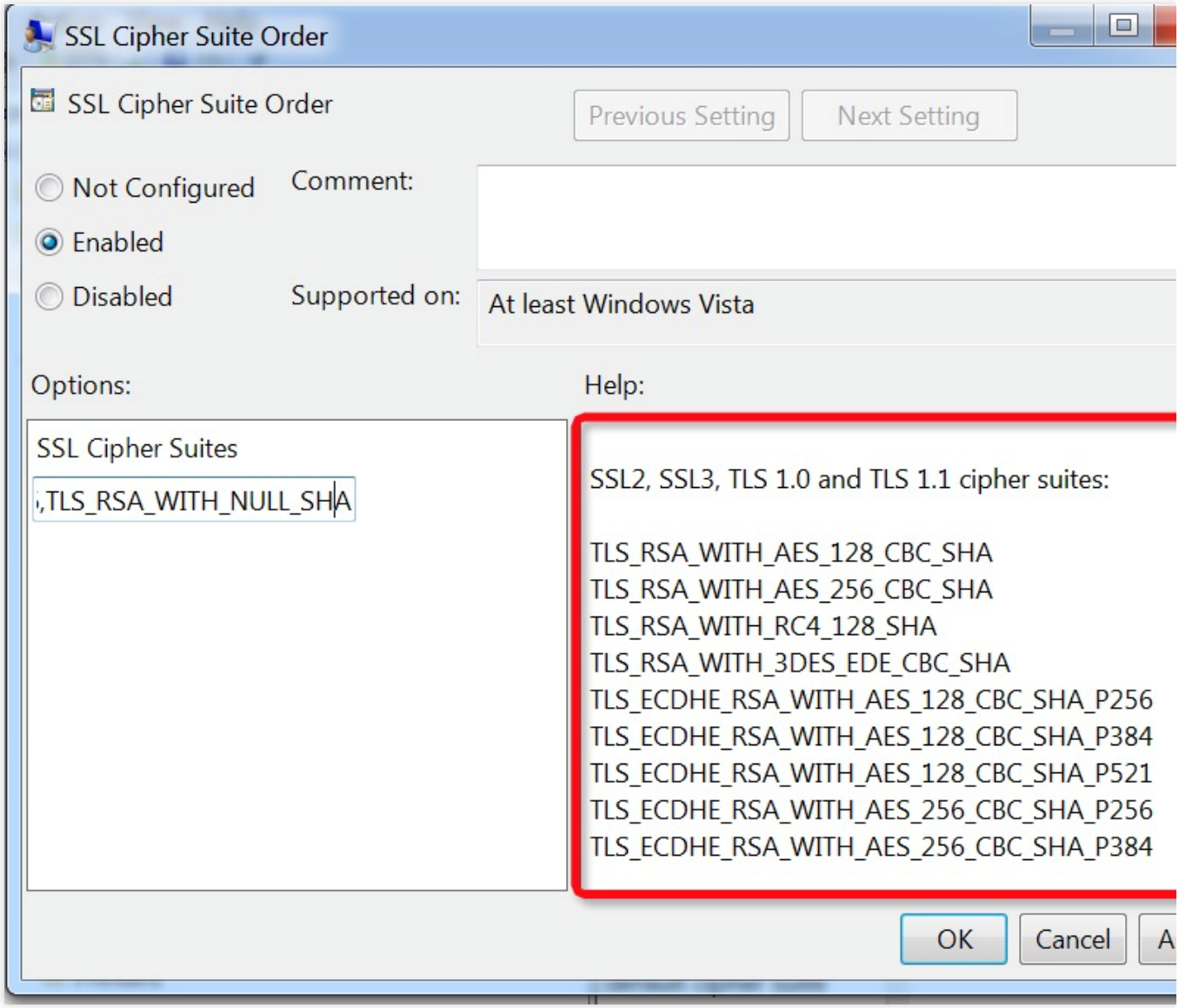
c. 按正确顺序排列套件；删除不想使用的所有套件。

d. 在每个套件名称的末尾键入一个逗号（最后一个套件名称除外）。确保没有嵌入空格。

e. 删除所有换行符，以便密码套件名称位于单独的一个长行上。

f. 将密码套件行复制到剪贴板，然后将其粘贴到编辑框中。最大长度为1023个字符。

8. 填写完成。如下图所示：



可将以下套件加入密码套件中：

TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA

TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA

TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256

TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384

TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256

TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384

推荐套件组合：

TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA_P256

TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA_P384

TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA_P521

TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA_P256

TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA_P384

TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA_P521
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256_P256
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256_P384
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256_P521
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384_P256
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384_P384
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384_P521
TLS_DHE_RSA_WITH_AES_256_GCM_SHA384
TLS_RSA_WITH_AES_128_CBC_SHA
TLS_RSA_WITH_AES_256_CBC_SHA
TLS_RSA_WITH_3DES_EDE_CBC_SHA
TLS_RSA_WITH_AES_128_CBC_SHA256
TLS_RSA_WITH_AES_256_CBC_SHA256
TLS_RSA_WITH_AES_128_GCM_SHA256
TLS_RSA_WITH_AES_256_GCM_SHA384

DNSPod 一键申请免费 SSL 证书

最近更新时间：2024-03-06 17:49:10

概述

如果您需要快速颁发腾讯云免费 SSL 证书，您可以使用 DNSPod 管理控制台一键申请功能，即可为您快速申请免费 SSL 证书。
以下操作将为您介绍如何快速申请免费 SSL 证书。

前提条件

已在域名注册商处注册域名。

操作指南

说明：
若您的域名已在 DNSPod 管理控制台正常托管，您可以跳过以下步骤1与步骤2操作。

步骤1：在 DNSPod 管理控制台添加域名

1. 登录 DNSPod 管理控制台，进入**我的域名**管理页面。
2. 在**我的域名**中，单击**添加域名**。
3. 在展开的输入框中，输入您需要添加的二级域名并单击**确认**，即可成功添加。

说明：
DNSPod 暂时不支持添加二级域名以外的其他子域名，如只支持 `dnspod.cn` 二级域名，不支持 `bbs.dnspod.cn` 三级域名。
若提示域名已被其他用户添加，请参见 域名取回相关 进行处理。

步骤2：修改域名 DNS 服务器

若您添加的域名提示 “未正确设置 DNS 服务器”，则需要将域名的 DNS 服务器修改为提示的 DNSPod 所属 DNS 服务器，DNSPod 才可进行解析托管。

说明：
DNSPod 将会根据您的域名注册商信息查询对应设置文档。您可以单击提示框，并查看对应设置文档完成修改操作。
若无对应设置文档或无法查询，则建议您前往您域名的注册商处进行咨询。

若您的域名为腾讯云注册并在当前登录的 DNSPod 账号下，则支持一键修改为正确的 DNS 服务器，单击**一键修改**并等待生效即可。

步骤3：一键申请免费 SSL 证书

1. 选择您需要申请免费 SSL 证书的域名，单击操作栏 **SSL**。
2. 在弹出的**申请 SSL 证书**窗口中，选择 **SSL 证书免费版**，并单击**免费申请**。如下图所示：

说明：

免费证书仅支持二级域名与其子域名，不支持通配符域名。若您需要通配符域名证书，请使用付费版 SSL 证书。

3. DNSPod 将自动为您进行域名验证，您只需等待腾讯云颁发证书即可。

说明：

腾讯云 SSL 证书将在1个工作日内完成审核，审核结果将以短信、邮件及站内信的方式通知您。

群晖（Synology）NAS 启用腾讯云 DDNS 并安装免费证书

最近更新时间：2024-03-06 17:49:08

操作场景

本文档指导您在群晖（Synology）NAS上启用腾讯云提供的 DDNS（动态域名服务）。启用后，您可以在拥有公网 IP 地址的群晖（Synology） NAS 上使用域名外网访问群晖（Synology）NAS。

说明：

本过程中仅购买域名可能收取一定的费用，启用 DDNS 及申请证书均免费。

前提条件

拥有群晖（Synology）NAS 管理员权限的账号。

拥有腾讯云 DNSPod 账号并完成 实名认证 。

群晖（Synology） NAS 拥有公网 IP 地址。

拥有1个可用域名并且解析托管在 DNSPod。

操作步骤

步骤1：获取 API 密钥信息

登录 腾讯云 API 密钥 ，获取您的腾讯云 API **SecretId** 及 **SecretKey** 密钥信息。

注意：

您的 API 密钥代表您的账号身份和所拥有的权限，使用腾讯云 API 密钥可以操作您名下的所有腾讯云资源。

为了您的财产和服务安全，请妥善保存和定期更换密钥，请勿通过任何方式（如 GitHub）上传或者分享您的密钥信息。

步骤2：群晖（Synology）NAS 配置 DDNS

1. 使用具有管理员权限的账号登入您的群晖（Synology）NAS，依次单击**控制面板 > 外部访问 > DDNS > 新增**。

2. 在弹出的 “添加 DDNS” 窗口中，填写相关信息。

服务供应商：请选择**腾讯云**。

主机名称：请填写您的**域名名称**。

用户名/电子邮件：请填写您获取到的 **SecretId** 信息。

密码/密钥：请填写您获取到的 **SecretKey** 信息。
从 Tencent Cloud 获取证书，并将其设置为默认证书：勾选选项后，可自动为您申请腾讯云 TrustAsia SSL 免费证书并替换 NAS 的默认 SSL 证书。
说明：
单击**测试联机**，测试是否能联机成功。如状态栏显示为**正常**，则代表联机成功。
3. 单击**确定**，即可完成设置。等待解析生效后，即可使用域名访问您的群晖（Synology）NAS。
说明：
解析生效时间一般需要 10分钟，请耐心等待。

步骤3：手动更新 DDNS（可选）

1. 完成设置后，单击**立即更新**，系统将为您更新最新的 DDNS 解析记录，并确认状态是否显示为正常。
2. 返回 我的域名 管理页面，单击您的域名，即可查看记录值是否已变更为您的公网 IP 地址。
若已变更，则设置成功。
若未变更，请根据以下常见问题进行排查。

常见问题

完成设置后域名还是无法正常访问？

请检查您的 IP 地址是否为公网 IP。您可直接在外网环境下使用浏览器访问群晖（Synology）NAS 获取的 IP 地址，若可访问，即为公网 IP。
完成设置后，需要等待解析生效才可正常访问，解析生效时间一般需要10分钟，请耐心等待。 解析生效后，您可使用 `ping 域名` 命令检查返回的 IP 地址是否为您的公网 IP 地址。

手动更新后解析记录值未变更？

请检查您填写的 **SecretId** 及 **SecretKey** 密钥信息是否正确。

使用 Python 调用 API 批量申请免费证书并下载至本地

最近更新时间：2024-03-06 17:49:08

概述

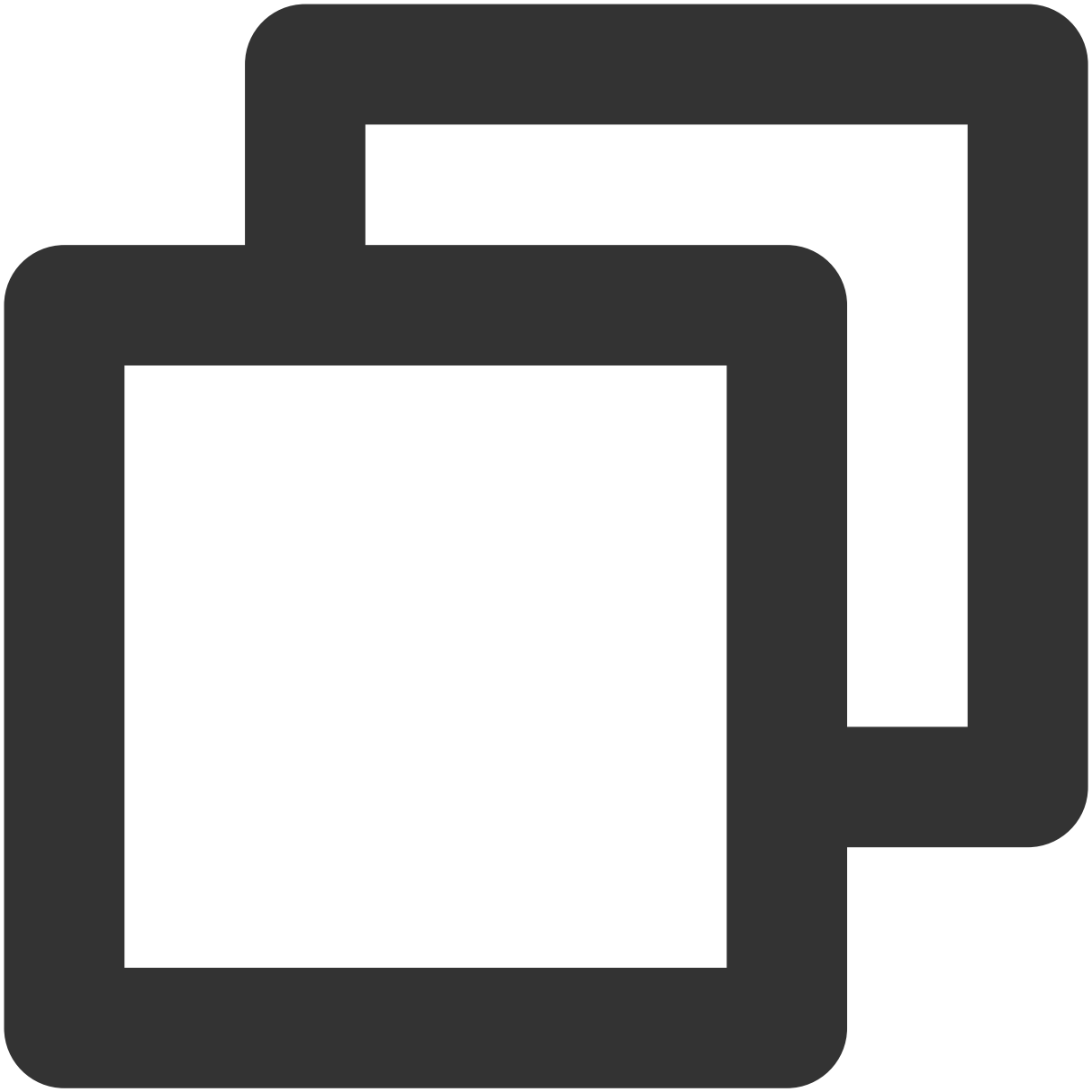
本文将指导您介绍如何使用腾讯云 API 批量申请证书并下载证书。

前提条件

子用户创建并授权云 API 与 SSL 证书全部权限。
已安装 Python 版本最新版本，如需安装，请前往 Python 官网 进行下载。
已安装 PyCharm 版本最新版本，如需安装，请前往 PyCharm 官网 进行下载。
注意：
为了保障您的账户以及云上资产的安全，请谨慎保管 SecretId 与 SecretKey 并定期更新。
创建子账号请参考 创建子账号并授权。

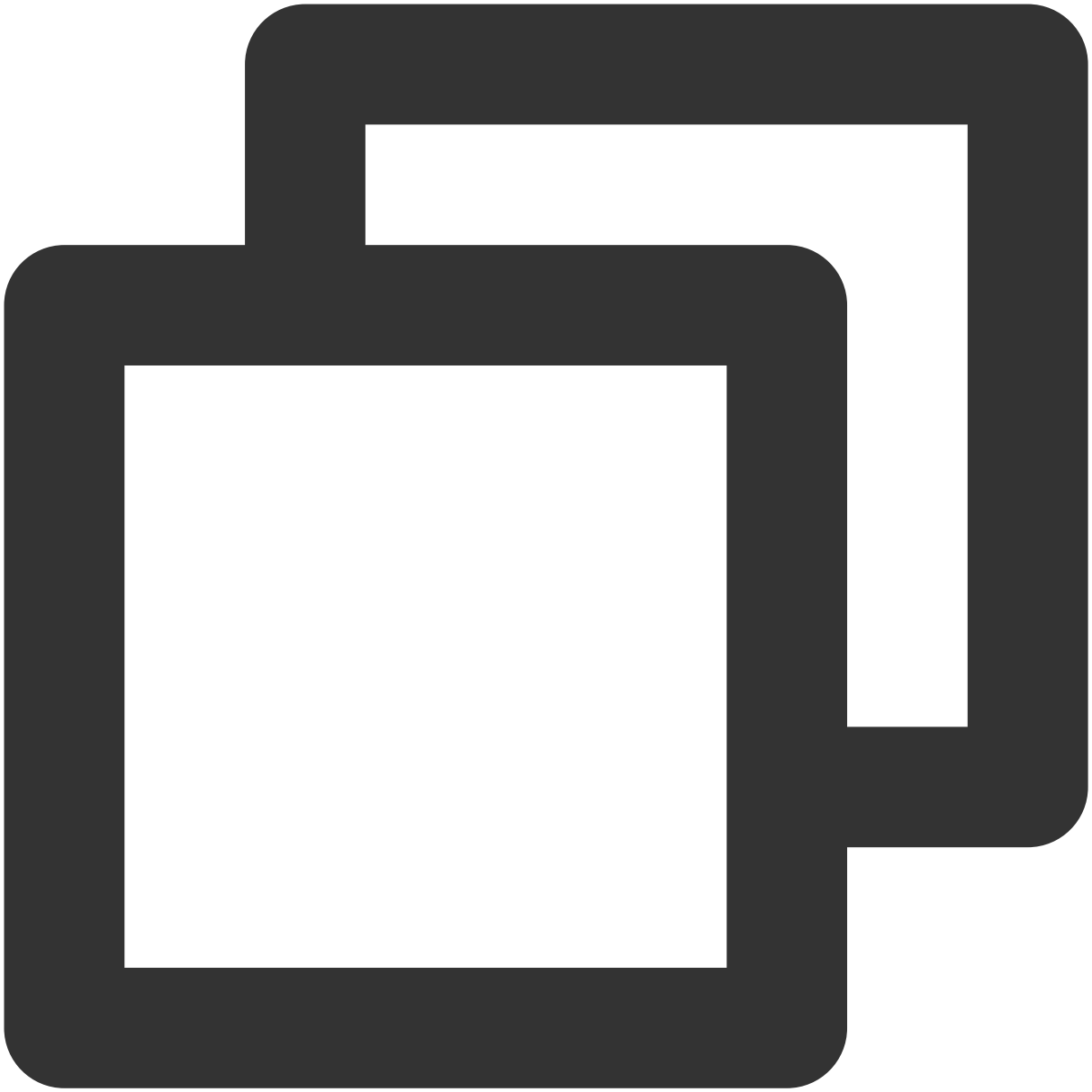
操作步骤

1. 打开命令提示符，查看 Python 版本。命令行如下：

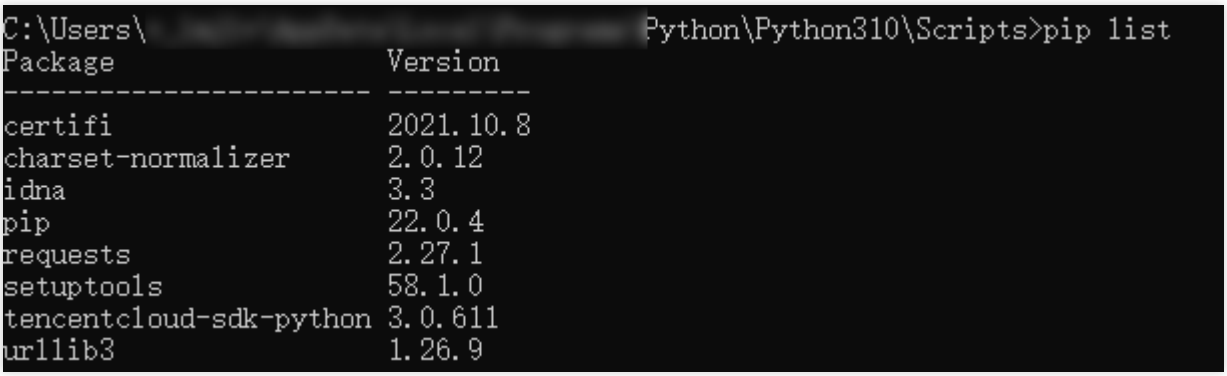


```
python -V
```

2. 查看 Python 目前已经安装的第三方模块，命令行如下：



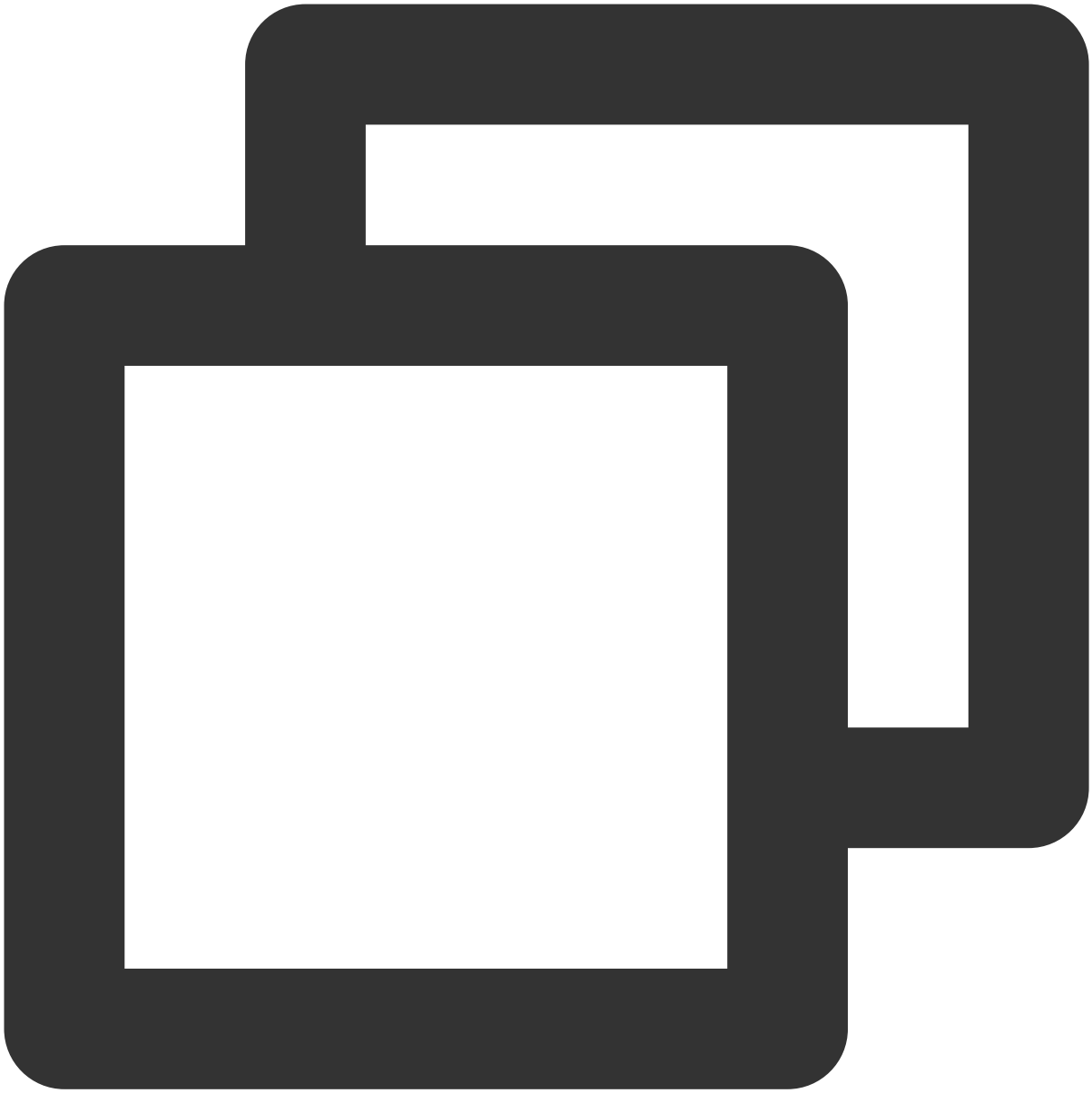
```
pip list
```



注意：

例如缺少 requests，可通过 `pip install requests` 安装该模块。

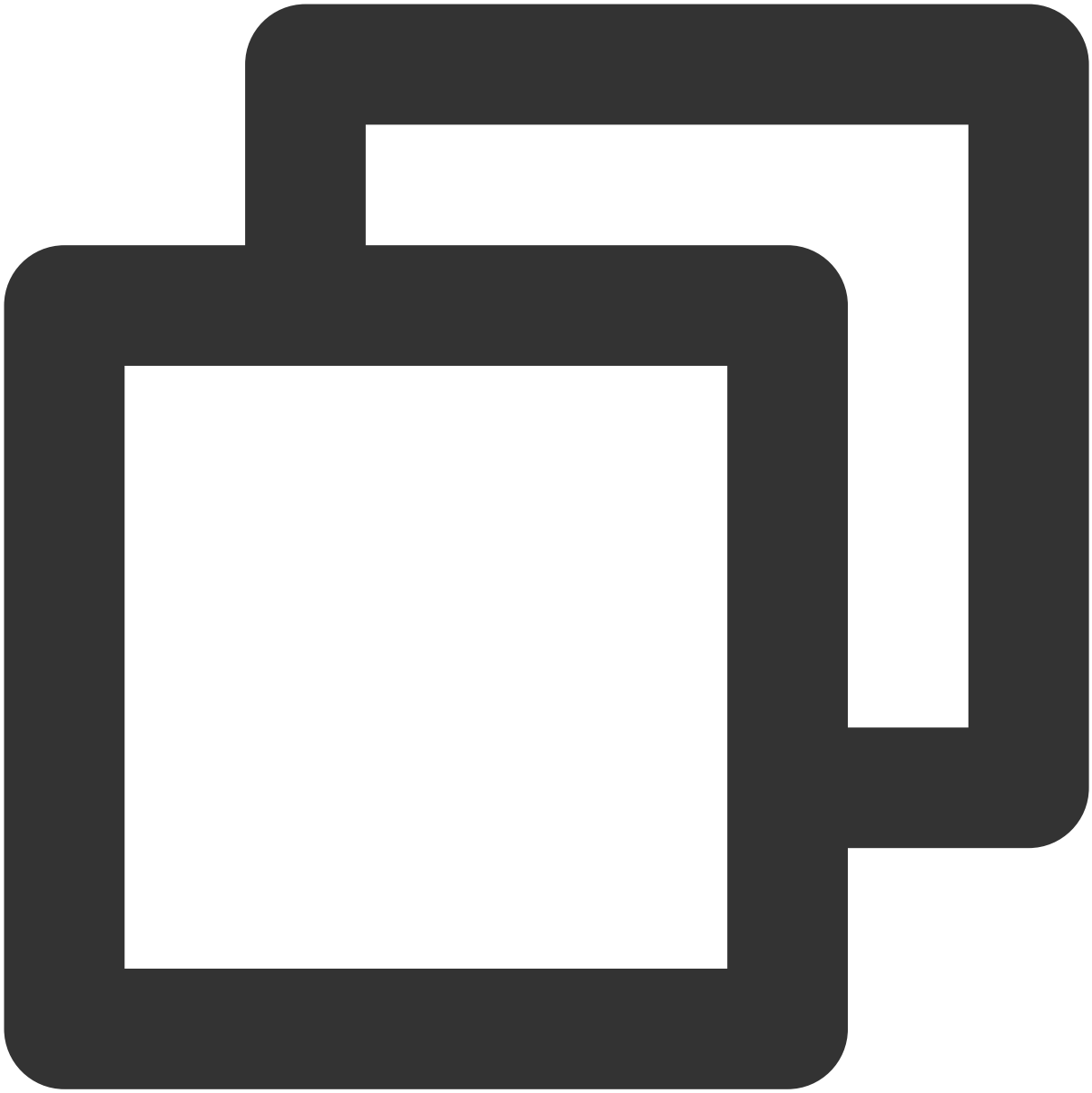
3. 通过 pip 安装腾讯云 Python SDK。命令行如下：



```
pip install -i https://mirrors.tencent.com/pypi/simple/ --upgrade tencentcloud-sdk-
```

4. 前往 Github 仓库 或者 Gitee 仓库 下载最新代码至本地，并进行解压。

5. 打开 PyCharm，导入最新的代码文件，进入 `tencentcloud-sdk-python/tencentcloud/ssl` 目录下并创建新的 Python 文件，例如 `apply.py` 。添加以下代码并执行。



```
import json,base64
from time import time,sleep
from tencentcloud.common import credential
from tencentcloud.common.profile.client_profile import ClientProfile
from tencentcloud.common.profile.http_profile import HttpProfile
from tencentcloud.common.exception.tencent_cloud_sdk_exception import TencentCloudS
from tencentcloud.ssl.v20191205 import ssl_client, models

start = time()
#SecretId 请填写您的 API 密钥ID, SecretKey 请填写您的 API 密钥KEY
cred = credential.Credential("SecretId", "SecretKey")
```

```
httpProfile = HttpProfile()
httpProfile.endpoint = "ssl.tencentcloudapi.com"
clientProfile = ClientProfile()
clientProfile.httpProfile = httpProfile
domain_name = []
while True:
 domain = input('要申请证书的域名：')#输入您需要申请的证书绑定的域名，如不需要继续申请，请直接按
 if domain == '':
     break
 else:
     domain_name.append(domain)

for i in range(len(domain_name)):
 client = ssl_client.SslClient(cred, "", clientProfile)
 try:

     req = models.ApplyCertificateRequest()
     params = {
         "DvAuthMethod": "DNS_AUTO",
         "DomainName": domain_name[i]
     }
     req.from_json_string(json.dumps(params))

     resp = client.ApplyCertificate(req)
     response = json.loads(resp.to_json_string())
     print('域名：{0}资料已提交，五秒钟后自动验证'.format(domain_name[i]))
     certid = response['CertificateId']
     sleep(5)
     try:
         req1 = models.CompleteCertificateRequest()
         params1 = {
             "CertificateId": certid
         }
         req1.from_json_string(json.dumps(params1))

         resp1 = client.CompleteCertificate(req1)
         response1 = json.loads(resp1.to_json_string())
         print('域名：{0}验证成功！准备下载证书'.format(domain_name[i]))
         try:
             req2 = models.DownloadCertificateRequest()
             params2 = {
                 "CertificateId": certid
             }
             req2.from_json_string(json.dumps(params2))

             resp2 = client.DownloadCertificate(req2)
             response2 = json.loads(resp2.to_json_string())
```

```
            # print(response2['Content'])
            content = response2['Content']
            with open("{0}.zip".format(domain_name[i]), "wb") as f:

                f.write(base64.b64decode(content))
                f.close()
        except TencentCloudSDKException as err:
            print(err)
    except TencentCloudSDKException as err:
        print(err)
except TencentCloudSDKException as err:
    print(err)
end = time()
print('本次代码执行共耗时：', round(end - start, 2), 's')
```

结果展示

1. 申请批量证书。
2. 下载证书内容。如下图所示：