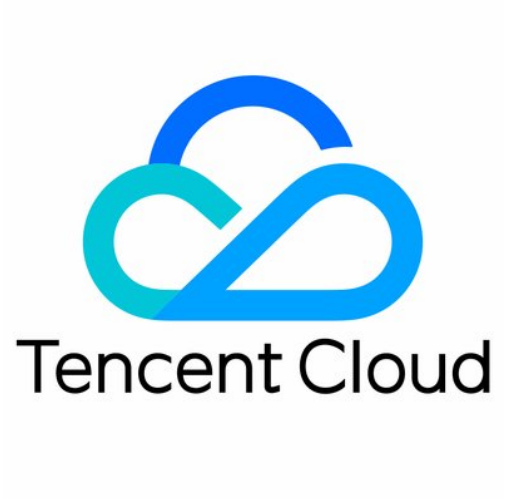
SSL Certificate Service

Best Practices

Product Documentation



Contents

Best Practices
Automatic Solution for Implementing and Issuing Multi-Year Certificates and Binding Resources

Last updated：2024-03-06 17:49:08

Overview

The multi-year certificate is an automatic SSL certificate review and delivery feature powered by Tencent Cloud. If you buy a multi-year certificate for two years or more and complete the review, Tencent Cloud will automatically review information and issue the next SSL certificate for you one month before your existing SSL certificate expires, simplifying the application process.
Tencent Cloud also supports SSL certificate management by cloud resources, where a new SSL certificate can be deployed as the original one in cloud resources such as Tencent Cloud CLB and CDN.
This document describes how to realize the automatic certificate issuance and resource binding of a multi-year certificate by combining the above two features.

Note:

Take the GeoTrust OV multi-year certificate and the Tencent Cloud CDN as examples here.

Directions

Step 1. Purchase a multi-year certificate

1. Log in to the SSL Certificate Service buy page.
2. Select and purchase a multi-year SSL certificate based on your needs.
3. Complete the SSL certificate application process.

Step 2. Deploy the SSL certificate to cloud resources

After the certificate is obtained, you can deploy it to Tencent Cloud resources (such as CDN) using the quick SSL certificate deployment feature.

1. Log in to the SSL Certificate Service console, select a target multi-year certificate, and click **Deploy**.
2. In the “Select deployment type” pop-up window, select a target type and corresponding resource instance.
3. Click **OK** to deploy the SSL certificate to selected cloud resources.

Step 3. Enable certificate management by cloud resources

1. Click a **certificate name** to go to the "Certificate Details" page.
2. In the Basic Info module, click **View** to check certificate management by cloud resources.
3. In the “Management by cloud resource” pop-up window, select target cloud resources.
4. Click **OK** to complete the operation.

Apple ATS Server Configuration

Last updated：2024-03-06 17:49:08

Note:

You need to configure cipher suites compliant with PFS specifications. The recommended configuration is:

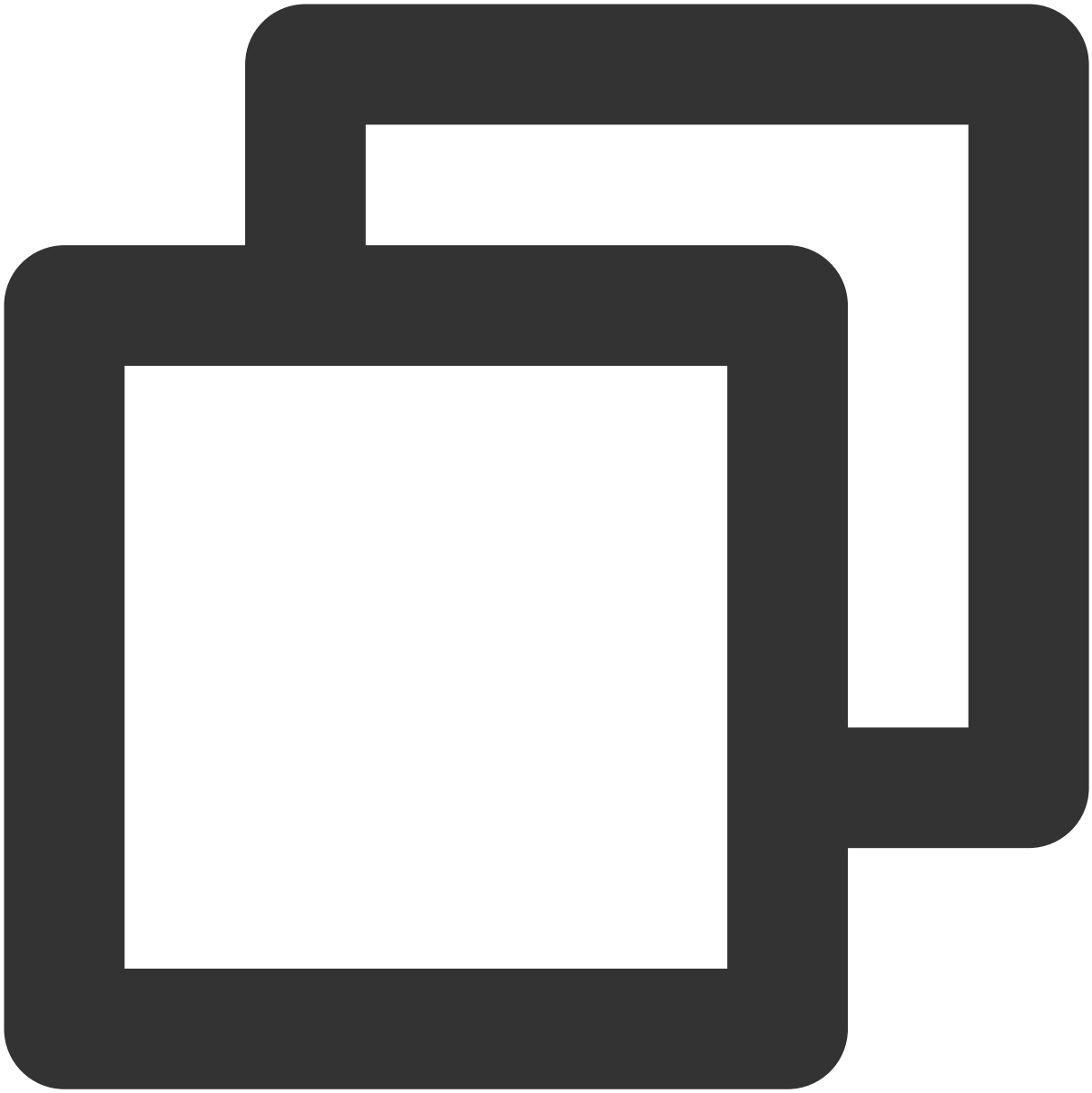
```
ECDHE-RSA-AES128-GCM-SHA256:ECDHE:ECDH:AES:HIGH:!NULL:!aNULL:!MD5:!ADH:!RC4
```

You need to enable the TLS1.2 protocol on the server. The recommended configuration is:

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

Nginx certificate configuration

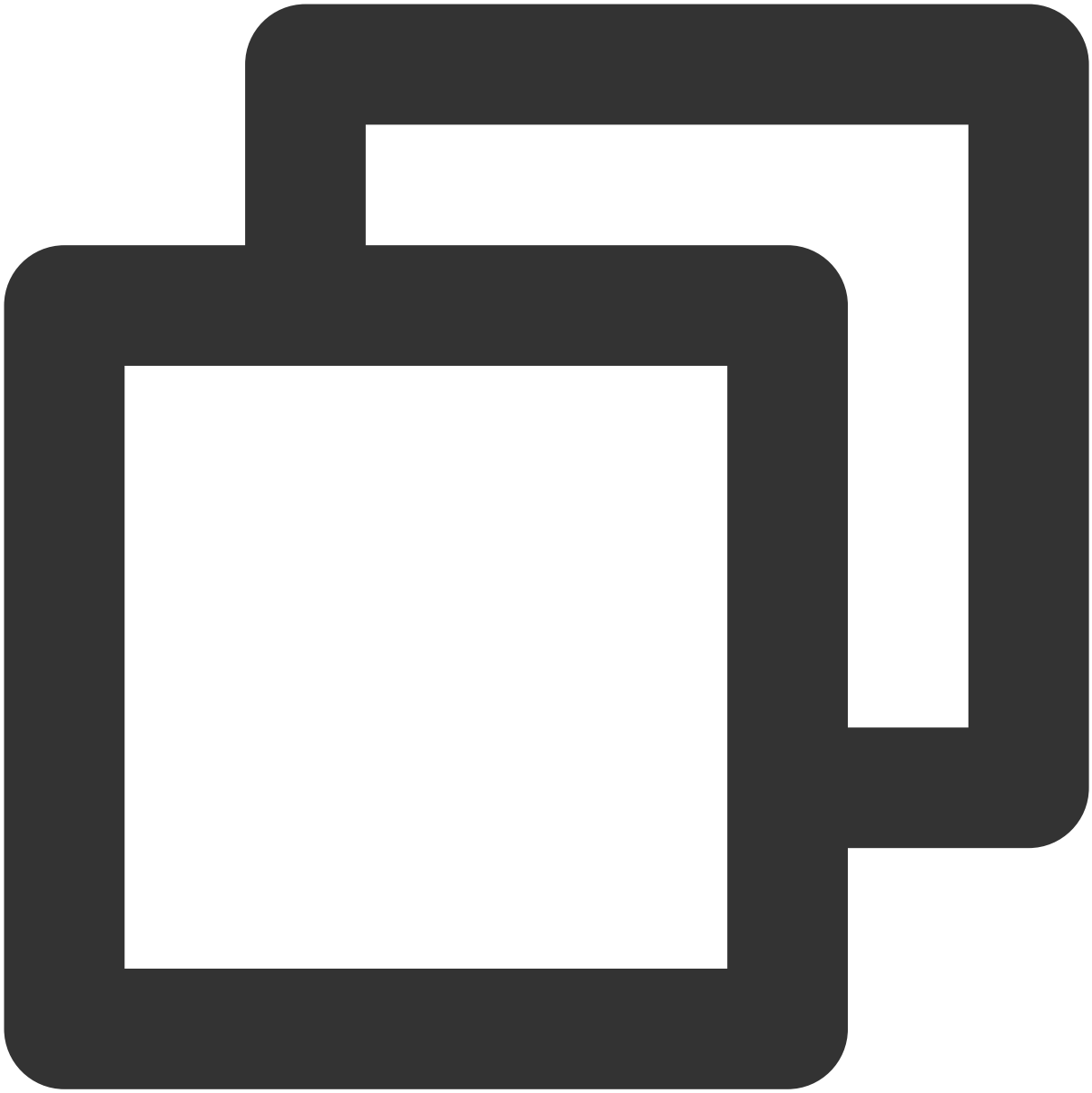
Update the `conf/nginx.conf` file in the Nginx root directory as follows:



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

Apache certificate configuration

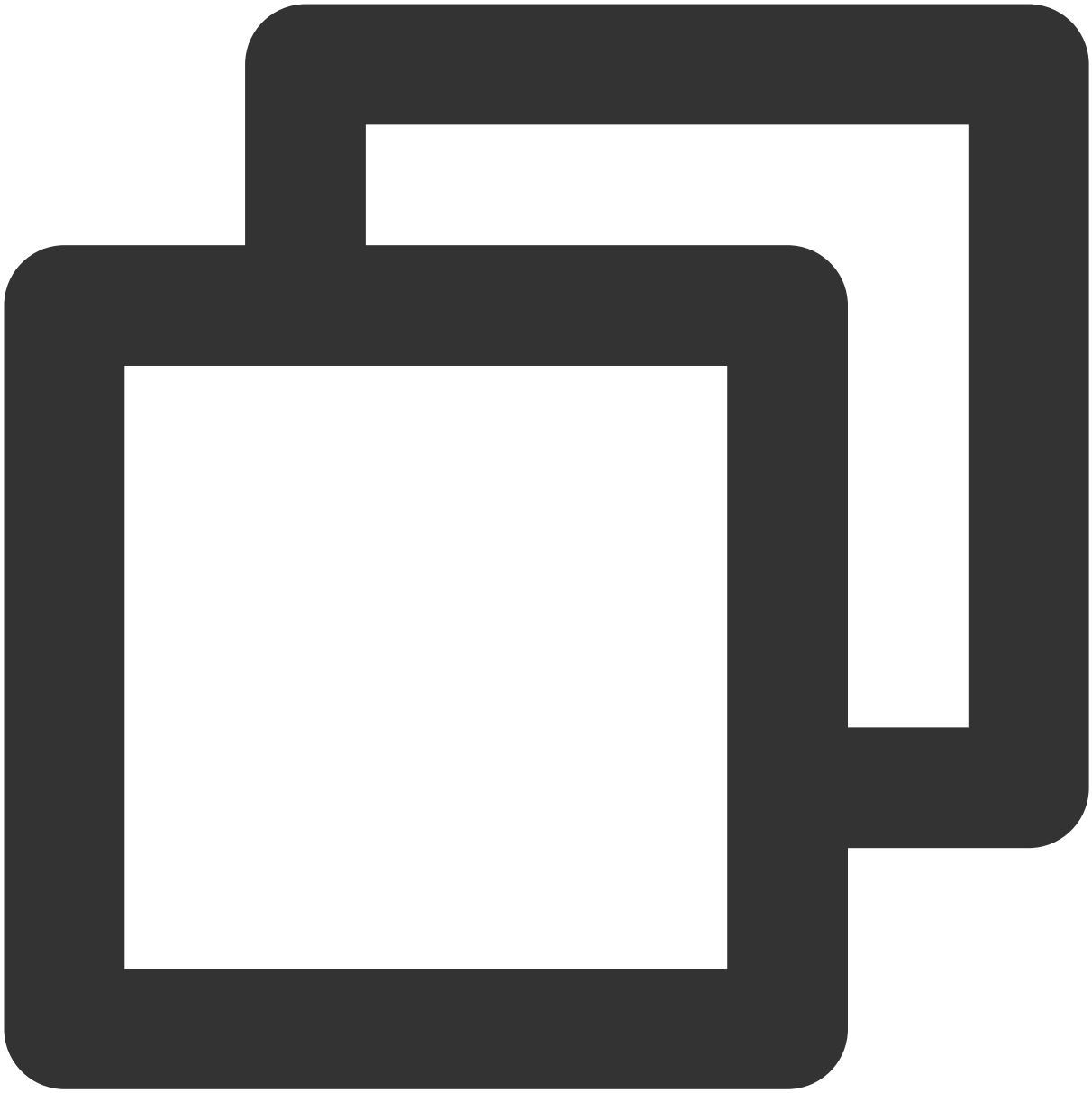
Update the `conf/httpd.conf` file in the Apache root directory as follows:



```
<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 certificate configuration

Update the `%TOMCAT_HOME%\\conf\\server.xml` file as follows:

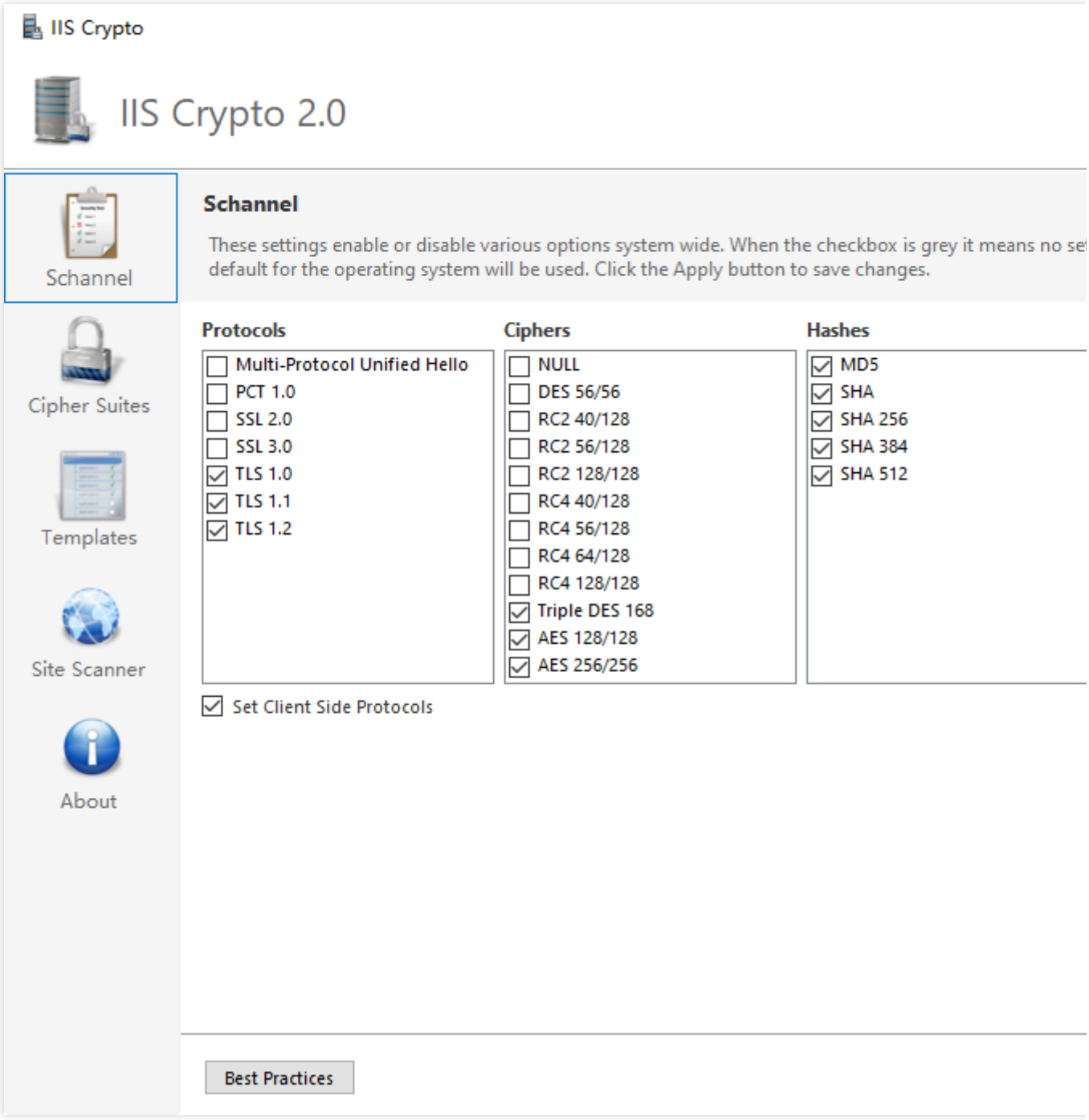


```
<Connector port="443" protocol="HTTP/1.1" SSLEnabled="true"
    scheme="https" secure="true"
    SSLProtocol="TLSv1+TLSv1.1+TLSv1.2"
    SSLCipherSuite="ECDHE-RSA-AES128-GCM-SHA256:ECDHE:ECDH:AES:HIGH:!NULL:!aNULL:!M
```

IIS certificate configuration

Method 1

Windows Server 2008 and earlier versions do not support the TLS1.2 protocol. Therefore, SSL tools are disabled on those versions. To address this issue, enable the TLS1.2 protocol to meet the ATS requirements.
Taking Windows Server 2008 R2 as an example, there is no adjustment to protocols and cipher suites after the certificate is imported.
The cipher suites will support ATS requirements after the certificate is imported but the TLS1.2 protocol required for ATS is not enabled. You can use ssltools (click to download) to enable the TLS1.2 protocol, as shown below:



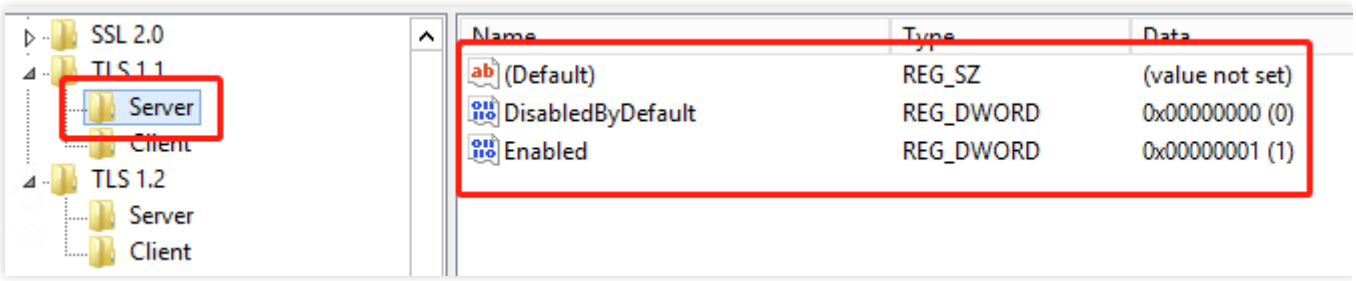
Select the 3 TLS protocols, and restart the system.
If PFS is not supported, select ECDHE and DHE in **Cipher Suites**.

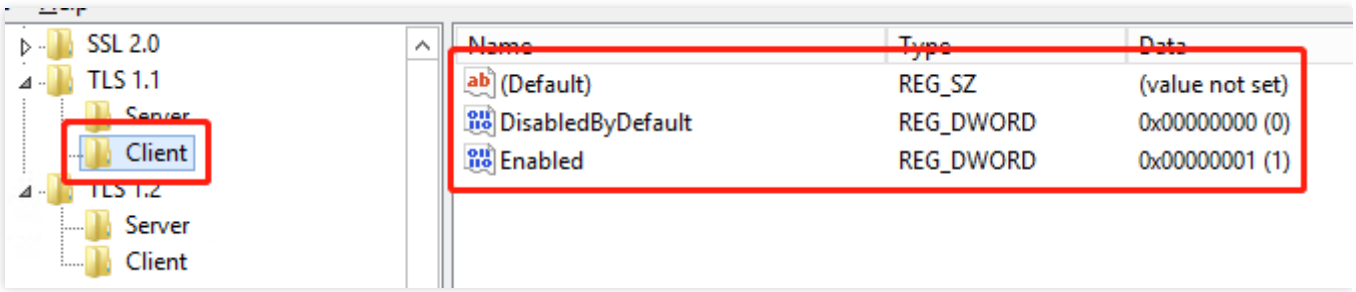
Method 2

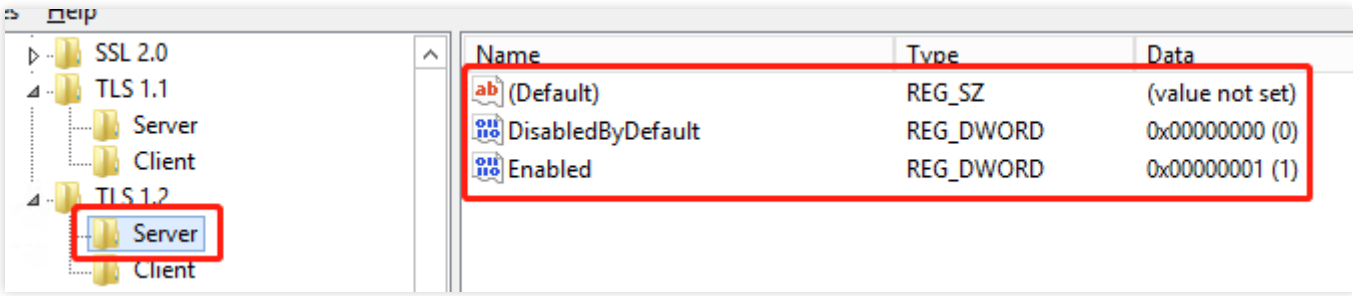
1. Choose **Start** -> **Run**. Enter `regedit` .
2. Find `HKEY_LOCAL_MACHINE\\SYSTEM\\CurrentControlSet\\Control\\SecurityProviders\\SCHANNEL\\Protocols` , right-click it, and then choose **New** -> **Item** -> **Create TLS 1.1, TLS 1.2**.
3. Right-click TLS 1.1 and TLS 1.2, and choose **New** -> **Item** -> **Create Server, Client**.
4. Create the following items (4 in total, DWORD 32-bit value) in the new servers and clients.

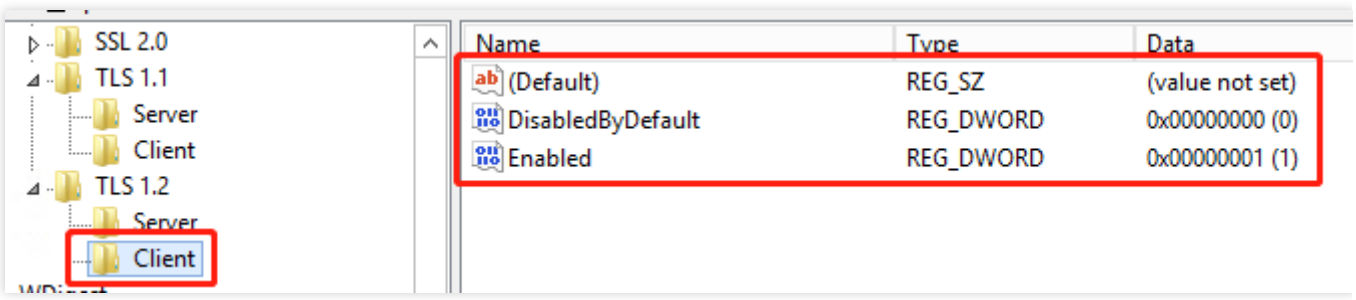
DisabledByDefault [Value = 0]

Enabled [Value = 1]







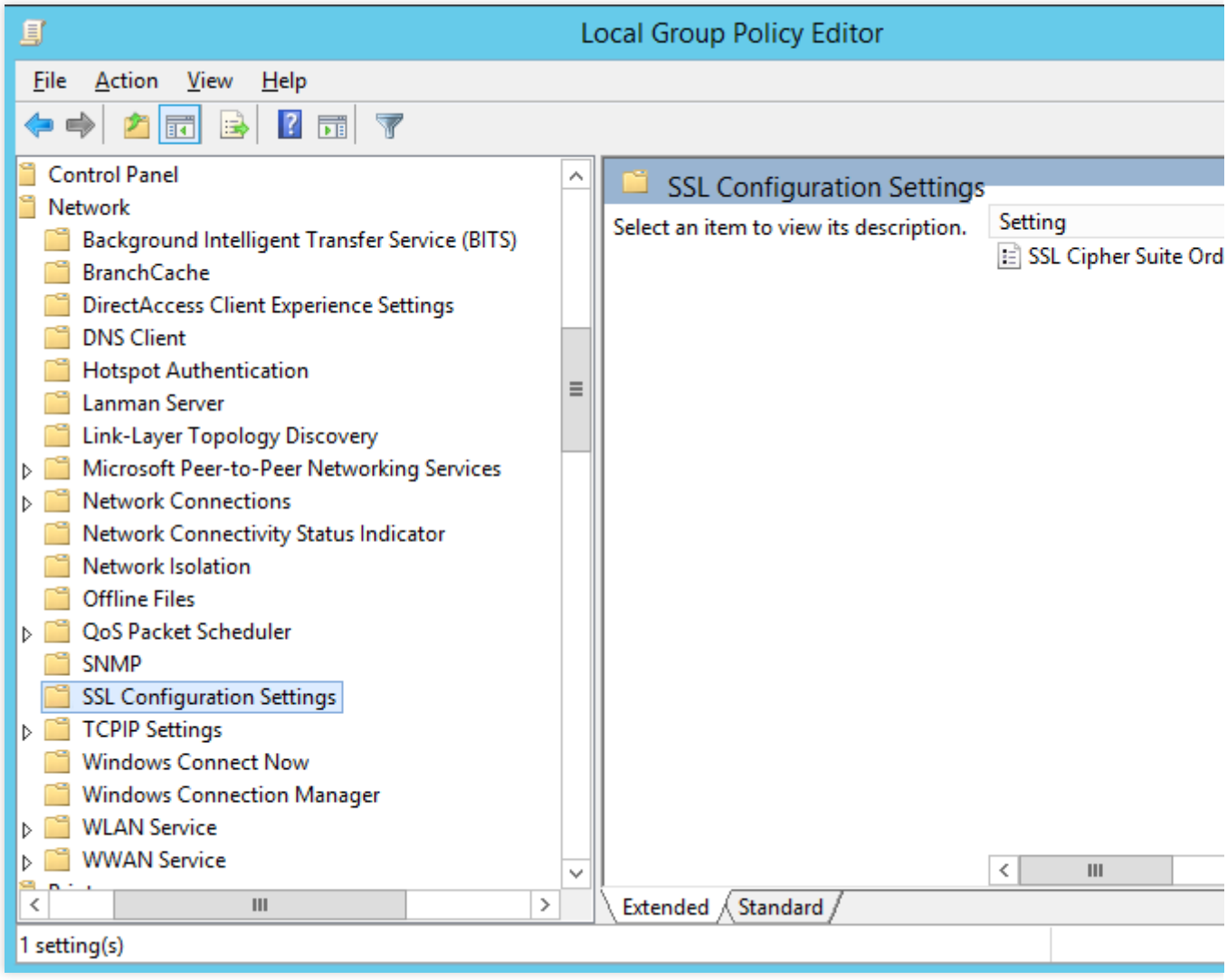


5. Restart the system.

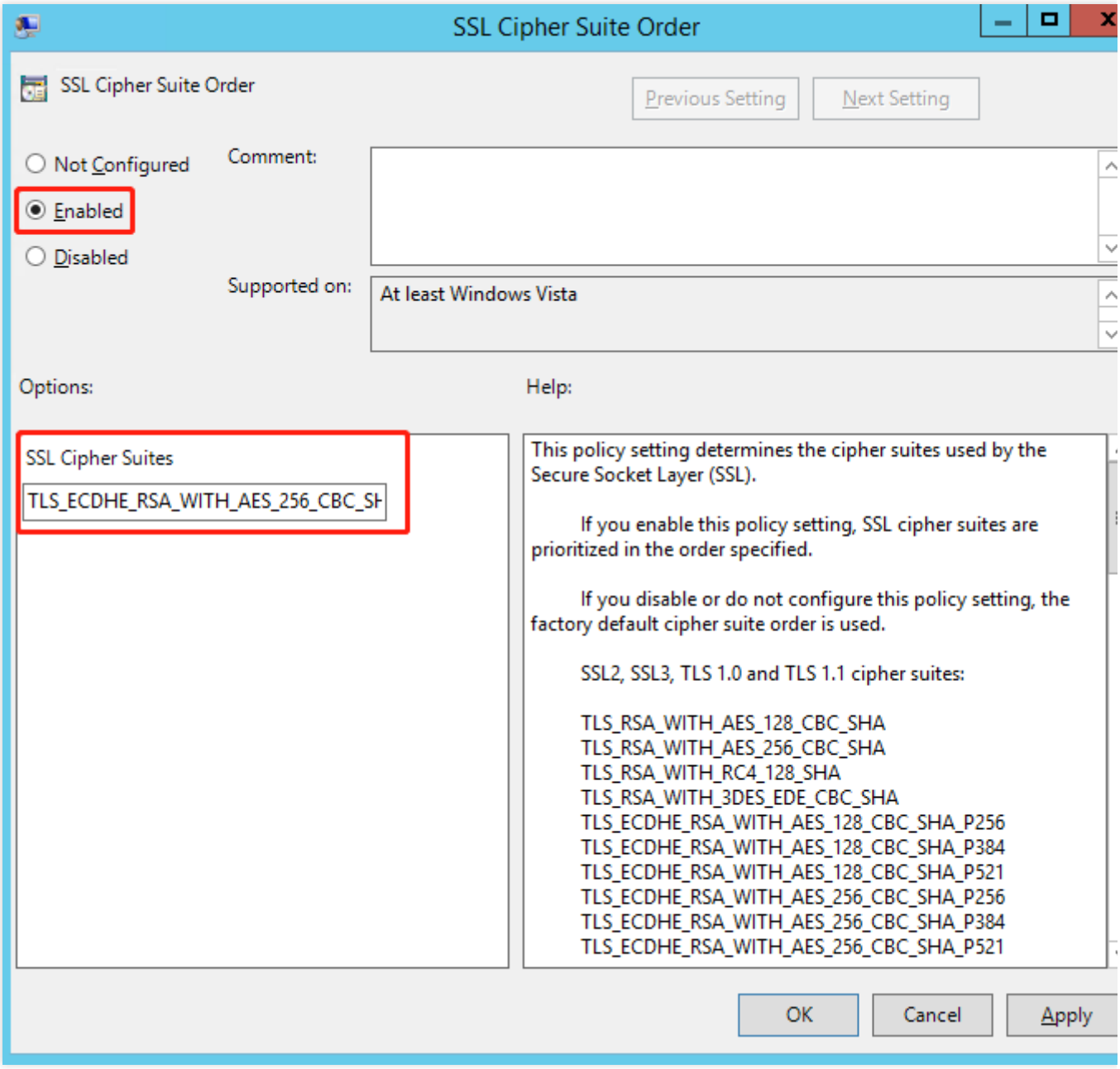
6. Adjust the cipher suites: choose **Start** -> **Run**, and enter `gpedit.msc` for the cipher suite adjustments after enabling the TLS1.2 protocol.

Note:

Adjustments can be made through the Group Policy Editor if PFS is not supported by the cipher suites.



7. Double-click **SSL Cipher Suite Order** and enter information, as shown in the following figure:



Select `Enabled` .

Add the supported ECDHE cipher suites to the SSL cipher suites, separated by commas (,).

Enter the cipher suite information as follows:

a. Open a blank WordPad document.

b. Copy the list of available suites on the right in the figure below and paste it into the document.

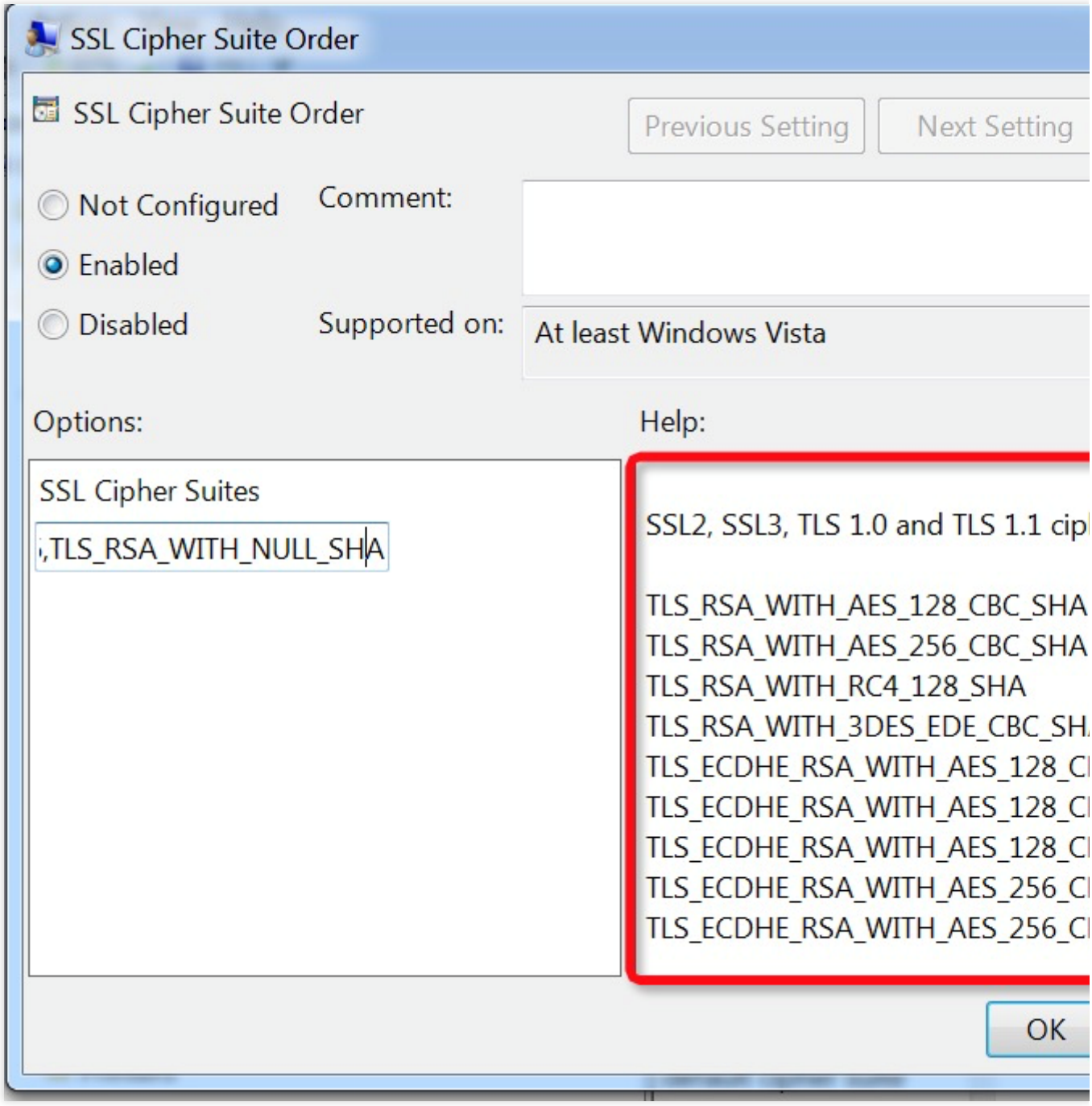
c. Sort the suites in the correct order and delete any suites you do not want to use.

d. Type a comma at the end of each suite name (except for the last one). Make sure no space is entered.

e. Remove all the line breaks so that the cipher suite names are in a single, long line.

f. Copy the cipher suite line to the clipboard and paste it into the edit box. You can enter up to 1,023 characters.

8. After the cipher suite information is entered, the content in the window is updated, as shown in the following figure:



The following suites can be added to the cipher suite:

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

The following suite combination is recommended:

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

Quickly Applying for a Free SSL Certificate via DNSPod

Last updated：2024-03-06 17:49:08

Overview

The DNSPod console allows you to quickly apply for a free SSL certificate.
This document describes how to quickly apply for one.

Prerequisites

You have registered a domain at your registrar.

Directions

Note:
You can skip steps 1 and 2 if your domain has been hosted in the DNSPod console.

Step 1. Add a domain in the DNSPod console

1. Log in to the DNSPod console and go to the **My Domains** page.
2. On the **My Domains** page, click **Add Domain**.
3. In the displayed input box, enter the target second-level domain and click **OK**.

Note:
DNSPod does not support adding subdomains other than second-level domains. For example, it supports the second-level domain `dnspod.cn` but not the third-level domain `bbs.dnspod.cn`.
If you are prompted that the domain has been added by another user, see Domain Retrieval.

Step 2. Modify the DNS server of the domain

If "DNS Servers Not Correctly Set" is prompted for the added domain, you need to change the DNS server of the domain to that of DNSPod to allow for DNS query and hosting in DNSPod.

Note:
DNSPod will query the corresponding settings document based on your registrar information. You can click the prompt box and view the settings document to complete the change.
If the settings document is unavailable or cannot be queried, we recommend that you contact your registrar.

If your domain is registered at Tencent Cloud and under the current DNSPod account, you can click **One-Click Modification** to quickly change to the correct DNS server.

Step 3. Quickly apply for a free SSL certificate

1. Select the target domain and click **SSL** in the **Operation** column.

2. In the **Apply for SSL certificate** pop-up window, select **Free SSL Certificate** and click **Apply (Free)** as shown below:

Note:

Only second-level domains and their subdomains are supported for a free certificate. If you need to use a wildcard domain, get a paid SSL certificate.

3. DNSPod will validate your domain automatically, and you only need to wait for the certificate issue.

Note:

Tencent Cloud will complete the SSL certificate review within one business day and notify you of the result through SMS, email, and Message Center.

Enabling Tencent Cloud DDNS and Installing Free Certificates for Synology NAS

Last updated：2024-03-06 17:49:10

Overview

This document describes how to enable Tencent Cloud's dynamic DNS (DDNS) in Synology NAS, so as to access Synology NAS with a public IP over the public network by using a domain.

Note:

In the process, fees may be incurred by domain purchase, but enabling DDNS and applying for a certificate are free of charge.

Prerequisites

You have a Synology NAS account with admin permissions.

You have a DNSPod account and have completed identity verification.

The Synology NAS has a public IP.

You have an available domain hosted with DNSPod.

Directions

Step 1. Get the API key information

On the TencentCloud API key page, get the **SecretId** and **SecretKey**.

Note:

Your API key represents your account identity and granted permissions, with which all Tencent Cloud resources under your account can be manipulated.

For the security of your assets and services, store your keys safely and change them regularly. Do not upload or share them via any method (such as GitHub).

Step 2. Configure DDNS in the Synology NAS

1. Log in to your Synology NAS with an admin account and click **Control Panel** > **External Access** > **DDNS** > **Add**.
2. In the **Add DDNS** pop-up window, enter the information.

Service Provider: Select **Tencent Cloud**.

Hostname: Enter your **domain**.
Username/Email: Enter the obtained **SecretId**.
Password/Key: Enter the obtained **SecretKey**.
Get a certificate from Tencent Cloud and set it as default: After this option is selected, the system will automatically apply for a free TrustAsia SSL certificate for you and replace the default NAS SSL certificate with it.
Note:
Click
Test Connection
to test the connection. If the
Status
is
Normal
, the connection is established successfully.
3. Click **OK**. Wait for the DNS record to take effect. Then, you can use the domain to access your Synology NAS.
Note:
The DNS record usually takes 10 minutes to take effect.

Step 3. Manually update the DDNS (optional)

1. After configuring the settings, click **Update Now**, and the system will update the DDNS record. Then, check whether the **Status** is **Normal**.
2. Go back to the **My Domains** page and click your domain to check whether the record value has changed to your public IP.
If so, the settings are successfully applied.
If not, troubleshoot according to the following FAQs.

FAQs

What should I do if the domain cannot be accessed after the settings are configured?

Check whether your IP is a public IP. Specifically, access the IP obtained by the Synology NAS via the browser in the public network environment. If the access succeeds, the IP is a public IP.
After configuring the settings, you need to wait for the DNS record to take effect (which usually takes 10 minutes) before access. Then, run the `ping domain` command to check whether the returned IP is your public IP.

What should I do if the record value does not change after the manual update?

Check whether the **SecretId** and **SecretKey** are entered correctly.

Batch Applying for and Downloading Free Certificates Using Python-based API Calls

Last updated：2024-03-06 17:49:08

Overview

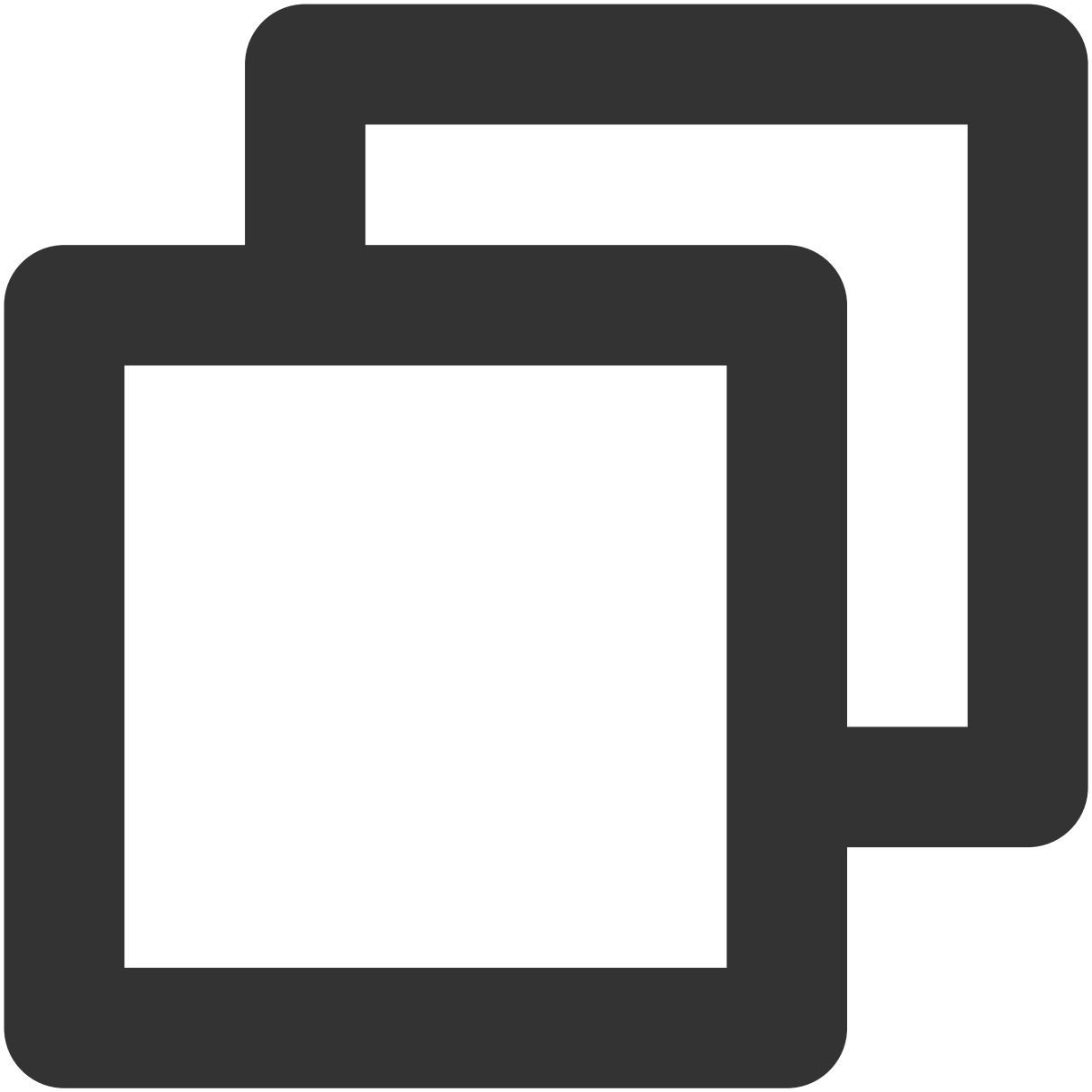
This document describes how to batch apply for and download certificates using Tencent Cloud APIs.

Preparations

Create a sub-account and authorize it with all permissions associated with cloud APIs and SSL certificates.
Install the latest version of Python. Download the package via here if necessary.
Install the latest version of PyCharm. Download it via here if necessary.
Note:
To keep your account and cloud assets under it secure, properly keep and regularly update `SecretId` and `SecretKey` .
Create a sub-account as instructed in Creating and Authorizing Sub-account.

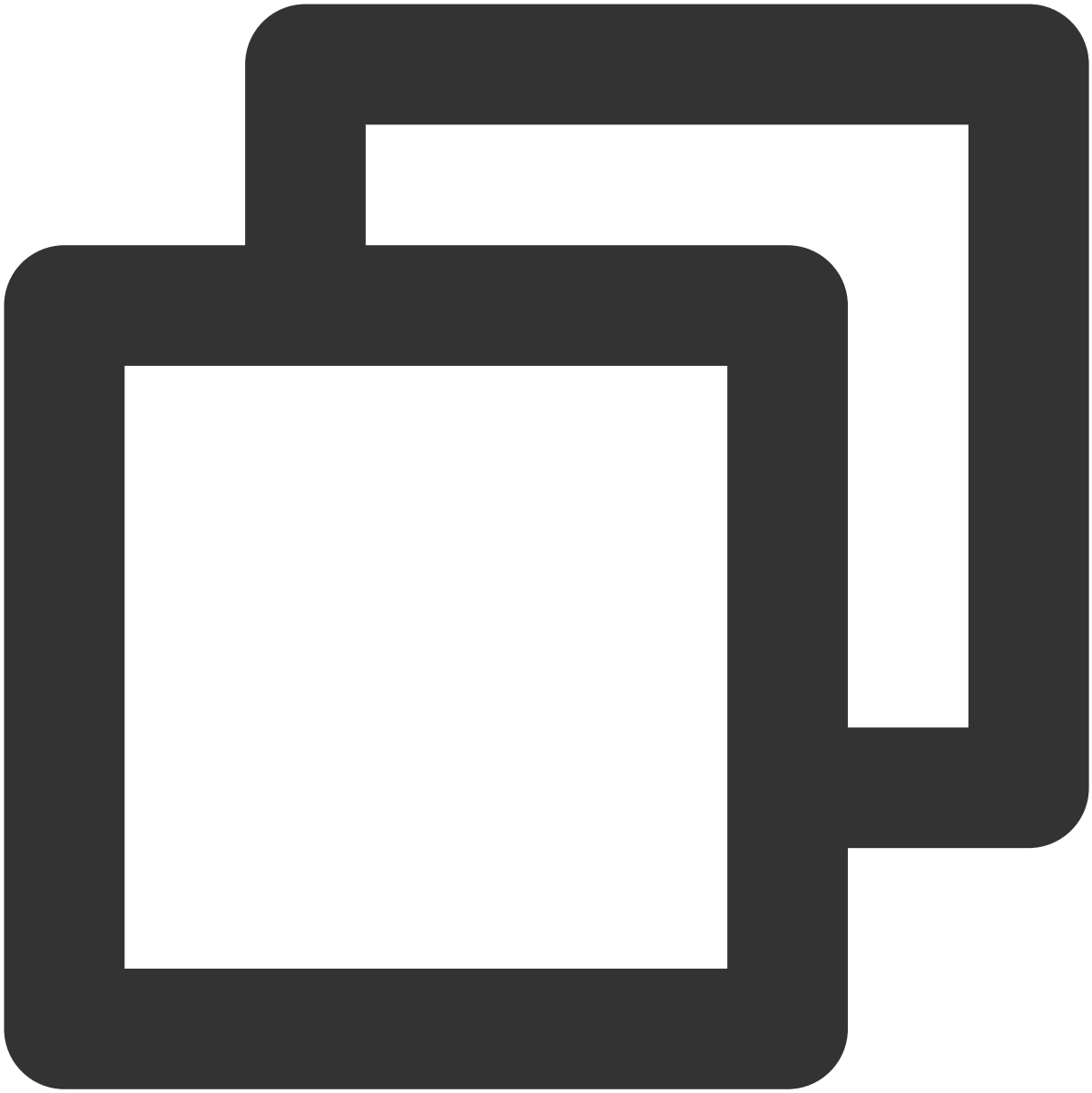
Directions

1. Open the command prompt window and view the Python script with the following command:

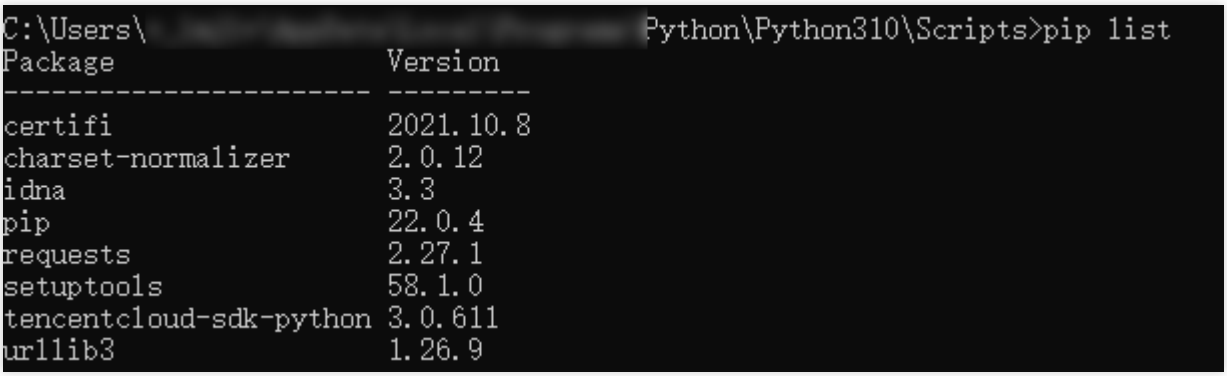


```
python -V
```

2. View installed third-party modules for Python with the following command:



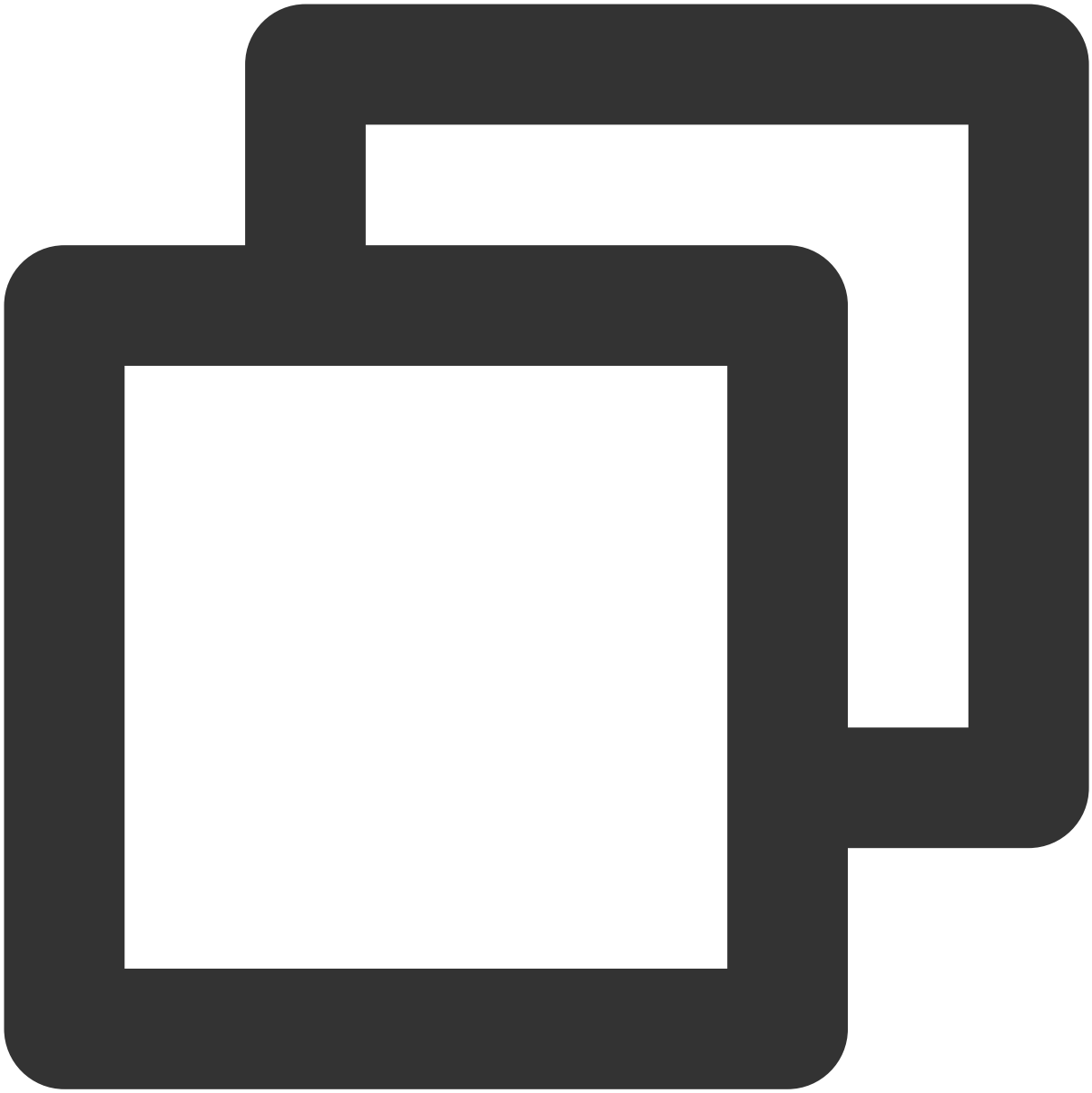
```
pip list
```



Note:

For example, if `requests` is missing, install it with `pip install requests`.

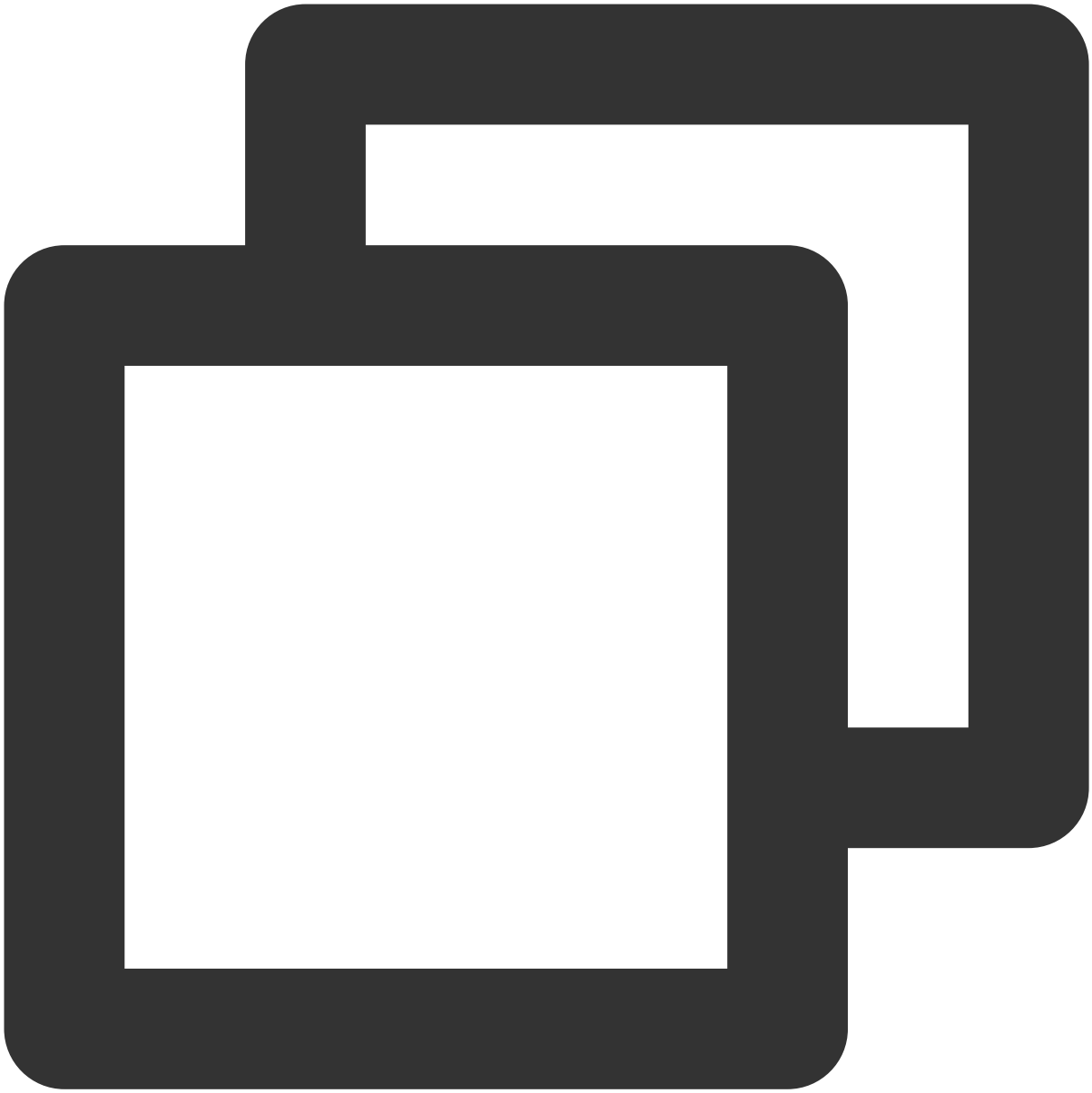
3. Use pip to install Tencent Cloud Python SDK with the following command:



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

4. Download the latest code from Github repository or Gitee repository and decompress it.

5. Open PyCharm, import the latest code file, create a new .py file under the `tencentcloud-sdk-python/tencentcloud/ssl` directory, such as `apply.py`, add the following code in the file and run.



```
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: Your API SecretID; SecretKey: Your API SecretKey.
cred = credential.Credential("SecretId", "SecretKey")
```

```
httpProfile = HttpProfile()
httpProfile.endpoint = "ssl.tencentcloudapi.com"
clientProfile = ClientProfile()
clientProfile.httpProfile = httpProfile
domain_name = []
while True:
 domain = input('the domain for which a certificate is applied')#Enter the domain t
 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('domain: {0}material submitted, auto-verification in 5s'.format(domain_n
     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('doman: {0}verified successfully. Prepare to download the certificat
         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('This code execution takes', round(end - start, 2), 's')
```

Result display

1. Apply for certificates in batches.
2. Download certificates.