SSL 证书

实践教程

腾讯云

【联系我们】

我们致力于为您提供个性化的售前购买咨询服务，及相应的技术售后服务，任何问题请联系 4009100100或95716。

文档目录

实践教程
SSL 证书角色策略配置说明

最近更新时间：2024-01-17 15:00:32

由于腾讯云 SSL 证书支持将证书部署到不同的腾讯云云服务，需要访问您账号下的云服务资源。因此当您在腾讯云SSL证书控制台，使用**查询证书关联资源、证书部署、证书更新、证书托管**等功能时，需要您对 SSL 证书角色进行授权，通过策略控制 SSL 证书访问范围。

> **说明：**
> 服务（相关）角色是由腾讯云服务预定义，经用户授权后相应服务即可通过扮演服务相关角色对用户资源进行访问操作。

角色使用场景

能够申请担任角色的对象我们称它为角色载体。目前，腾讯云角色载体分为三类：腾讯云账号、已支持角色功能的产品服务、身份提供商。对应的场景如下：

- 您要向您账号中的用户授予临时的资源访问权限，或者是向另一个腾讯云主账号内的用户授予您账户中的资源访问权限。
- 您可能需要允许腾讯云产品服务对您的资源拥有访问权限，但不希望将长期密钥嵌入在产品服务中，因为这样存在难以轮换密钥以及被截取后泄露导致的安全问题。
- 您要向您账号中的资源（例如容器实例，云服务器实例等）绑定角色，使云资源对不同的云服务具有不同的访问权限。详情请参见 基于资源的服务角色 。

SSL 证书在访问管理控制台的角色名称

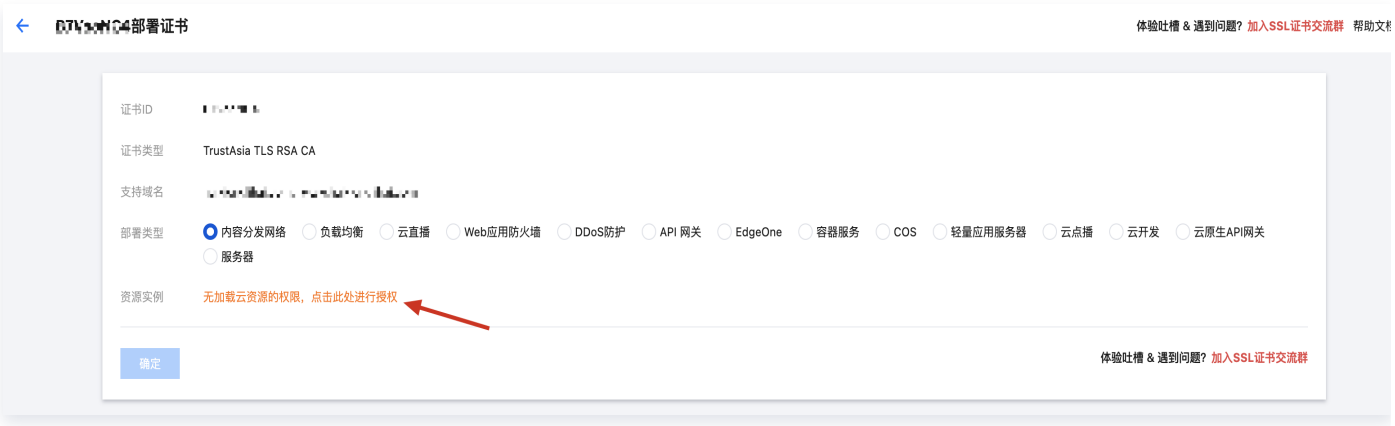
您可参考下列角色描述，根据自己的使用场景需要进行授权。例如，当您需要使用**查询证书关联资源、证书部署、证书更新、证书托管功能时，可以提前进行角色授权，**方便您后续使用。详细信息请参见 SSL 证书 。

CAM 中产品名	角色名称	角色类型	备注
SSL证书	SSL_QCSLinkedRoleInCertificateWaf	服务相关角色	使用**一键 HTTPS** 时授权。
SSL证书	SSL_QCSLinkedRoleInCertificateDependence	服务相关角色	使用**自动添加 DNS** 时授权。
SSL证书	SSL_QCSLinkedRoleInReplaceLoadCertificate	服务相关角色	使用**查询证书关联资源、证书部署、证书更新、证书托管**时授权。 点击此处前往访问管理控制台进行角色授权>>

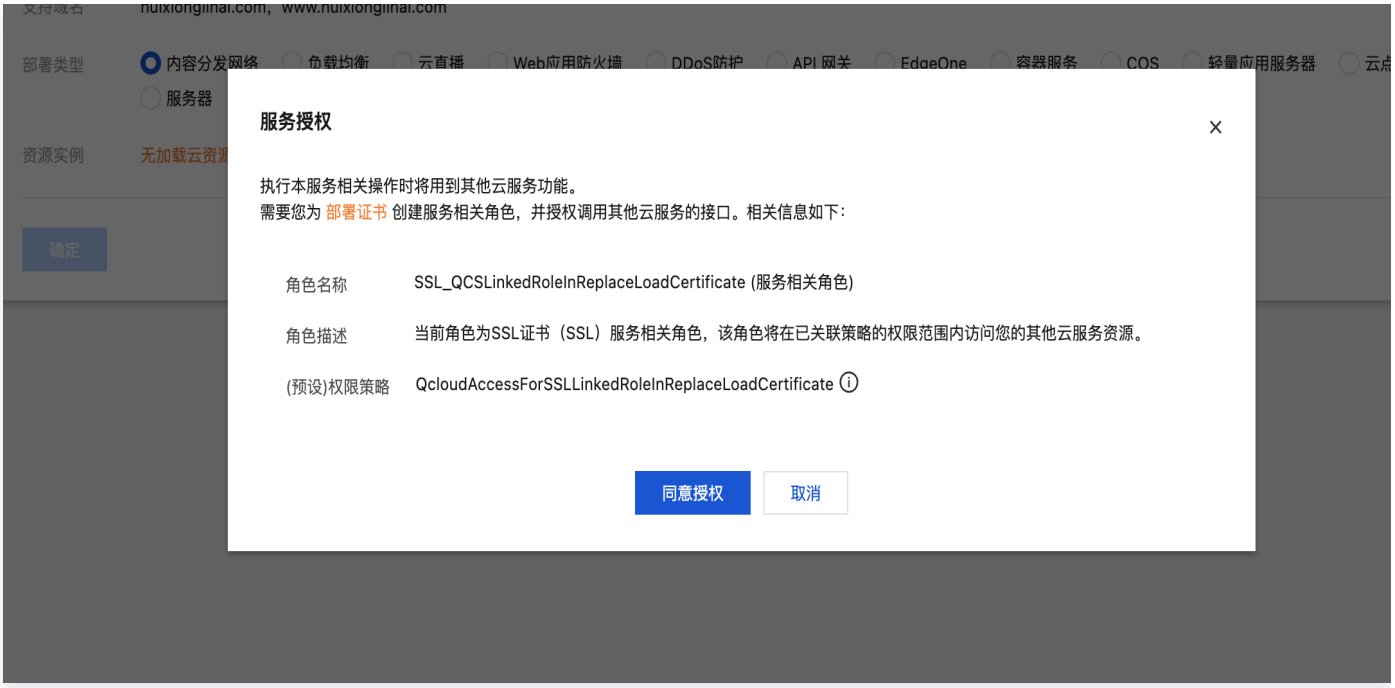
SSL证书	SSL_QCSLinkedRoleInCertificateCloudMonitor	服务相关角色	使用**可视化监控**时授权。
SSL证书	SSL_QCSLinkedRoleInDescribeDeployedResources	服务相关角色	使用**查询证书关联资源**时授权（旧接口）。

为 SSL 证书角色进行授权

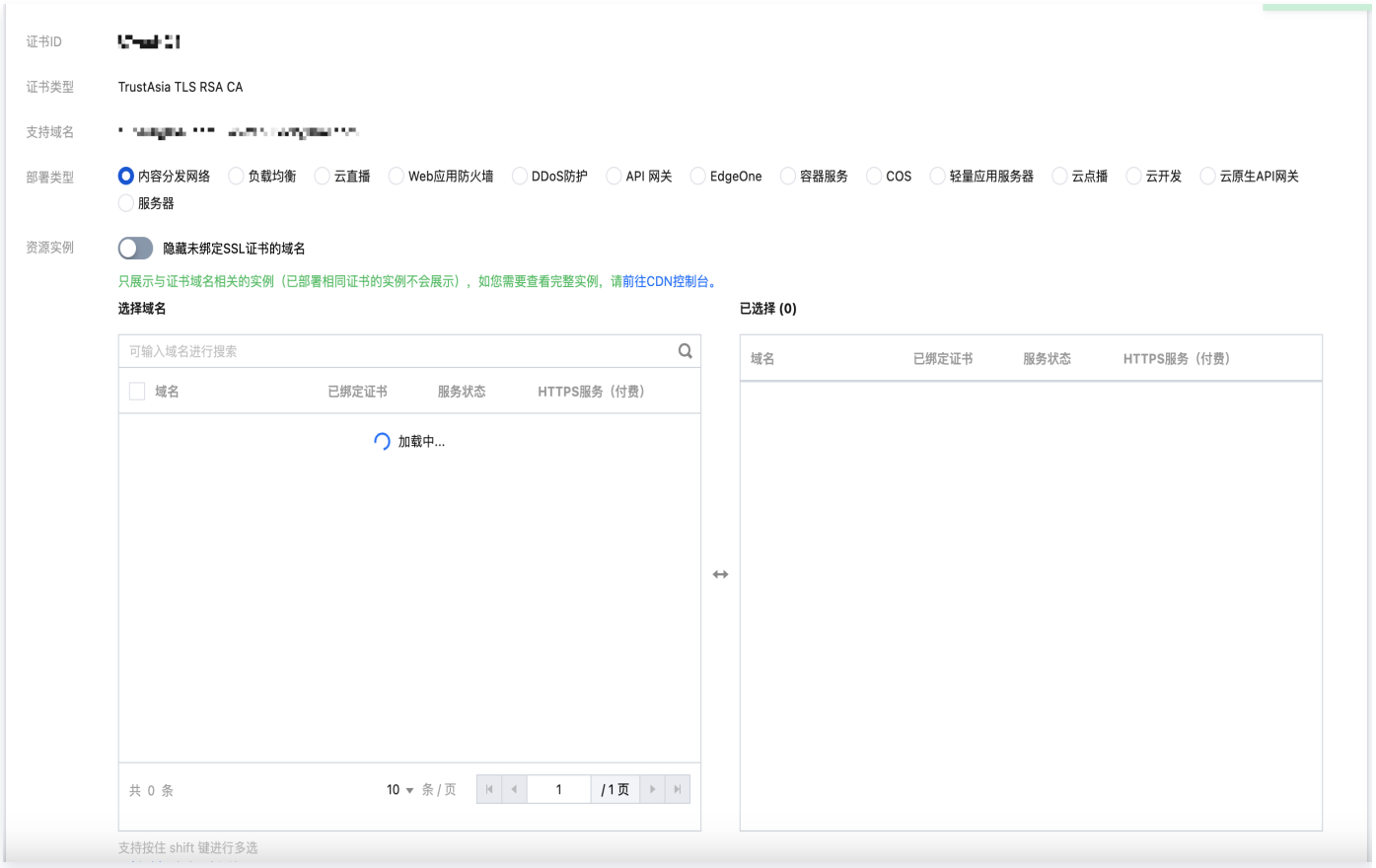
1. 当您使用证书部署时，若缺少角色授权，控制台页面会有相关提示，如下图所示：



2. 单击**点击此处授权**，在弹出的窗口中，单击**同意授权**进行授权。



3. 完成授权后，即可查询账号下云服务的资源。



如何为子账号赋予扮演角色策略?

详细信息，请参见 为子账号赋予扮演角色策略 。

腾讯云实现全站 HTTPS 方案

最近更新时间：2025-05-18 08:39:52

概述

腾讯云和权威的数字证书授权（CA）机构和专家级证书代理商合作，支持域名型、企业型、企业增强型 SSL 证书的申请和上传管理，并且腾讯云 CDN、负载均衡服务均支持 SSL 证书的快速部署。
您可以在腾讯云上一站式实现全站 HTTPS，下面详细说明如何操作。

前提条件

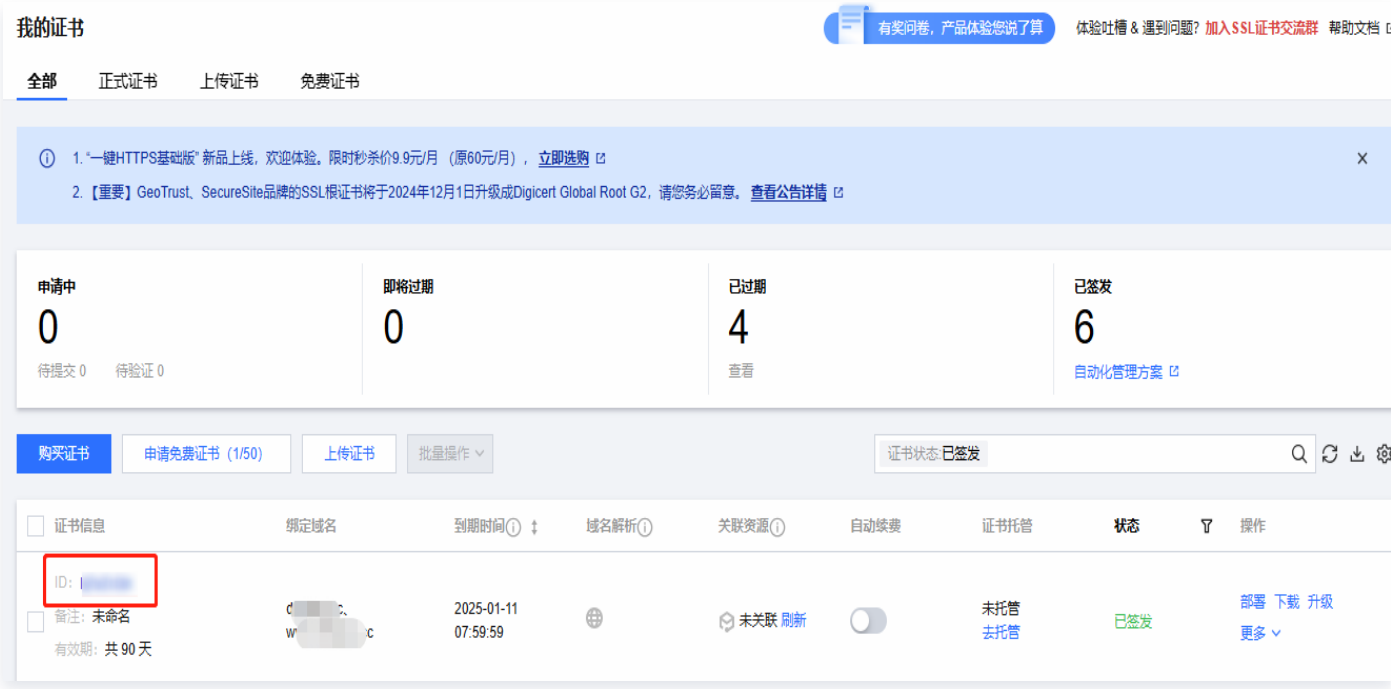
已登录 SSL 证书控制台，成功申请获取证书（参考 如何免费申请域名型证书）。

操作步骤

部署证书到负载均衡

> **注意：**
> 操作之前，请确认您的 负载均衡控制台 是否有实例，若没有实例，请您先创建实例。

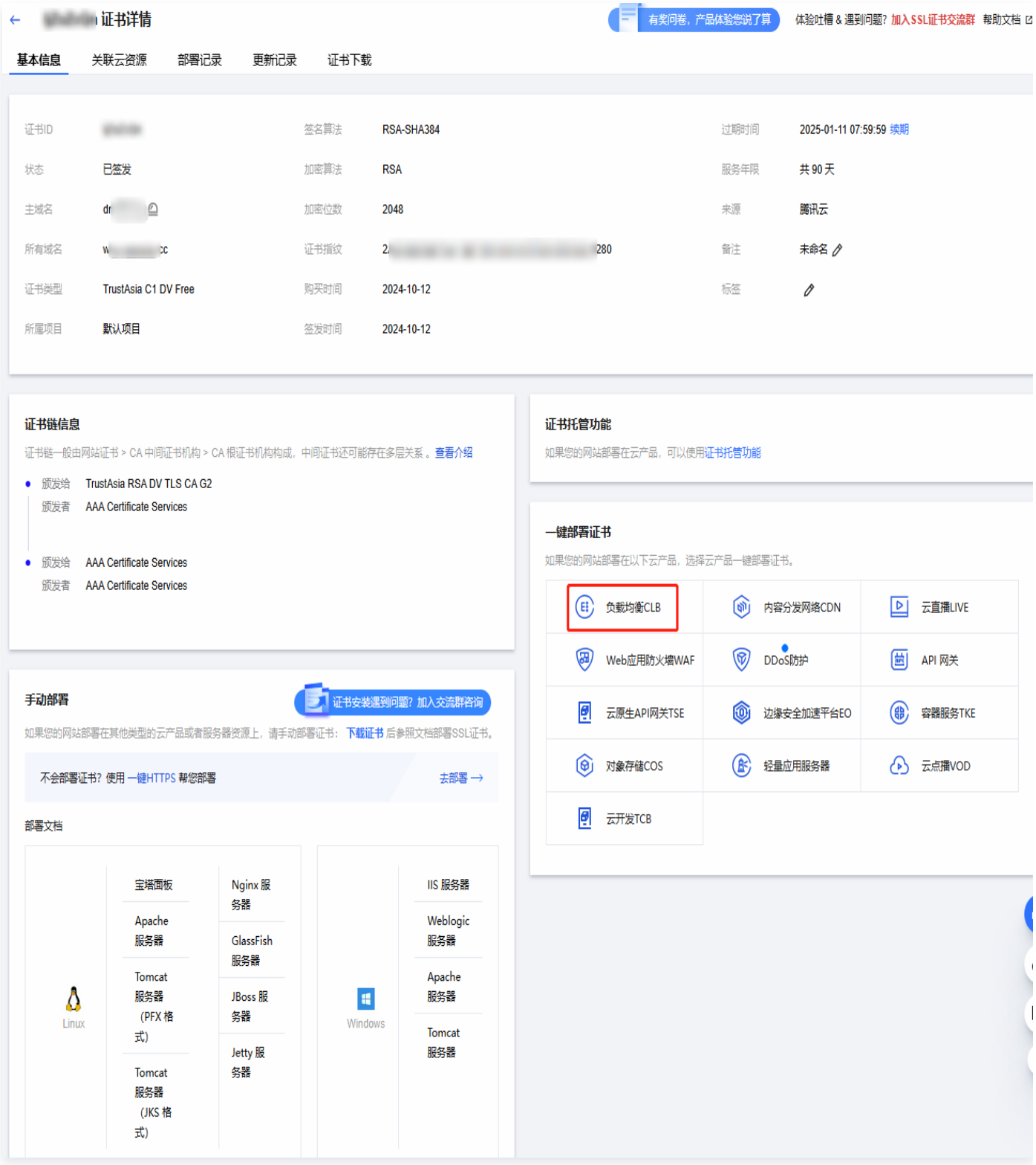
1. 在我的证书页面，在**证书信息**列表中，选择您需要部署的证书，并单击**证书 ID**，如下图所示：



2. 在**证书详情**页面，选择**基本信息**页签，在**一键部署证书**模块中， 单击**负载均衡 CLB**。如下图所示：

> **注意：**

> 目前不支持华南地区-深圳金融。



3. 在弹出的**部署证书**窗口中，选择**负载均衡 CLB**，选择资源实例以及监听器资源。如下图所示：

> **说明：**

> 如果您的负载均衡（CLB）资源未创建监听器资源，请参见 负载均衡监听器概述 并选择您需要的类型进行配置监听器。

4. 单击**确定**，即可操作成功。如下图所示：

部署证书到内容分发网络 CDN

> **注意：**
>
> - 域名需要已经接入 CDN，且状态为**部署中**或**已启动**，关闭状态的域名无法部署证书，具体操作请参考 接入域名 。

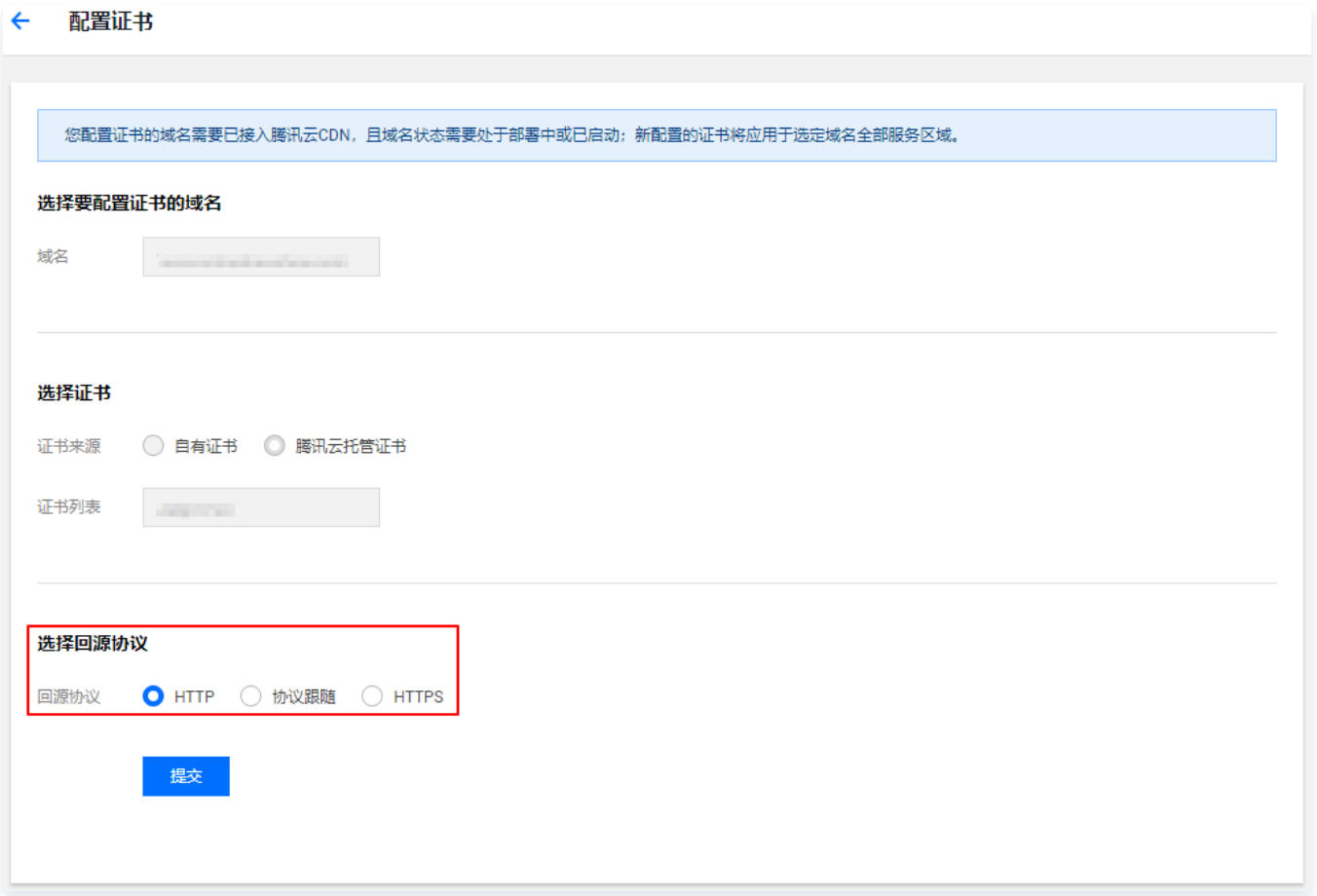
- COS 或 数据万象开启 CDN 加速后，默认的 .file.myqcloud.com 或 .image.myqcloud.com 域名无法配置证书。
- SVN 托管源暂时无法配置证书。

1. 在 **我的证书** 页面，单击**已签发**页签，选择您需要部署的证书，并单击**证书 ID**，如下图所示：

2. 进入**证书详情**页面，单击**一键部署**。

3. 在弹出的**选择部署类型**窗口中，选择**内容分发网络**，并单击**确定**。

4. 跳转到 CDN 控制台，进入**配置证书**详情页，已显示对应的域名、证书来源以及证书 ID。
5. 选择回源协议方式，您可以选择 CDN 节点回源站获取资源时的回源方式，如下图所示：



- 选择 **HTTP** 回源配置成功后，用户至 CDN 节点请求支持 HTTPS/HTTP，CDN 节点回源站请求均为 HTTP。
- 选择 **协议跟随** 回源配置，您的源站需要部署一张证书，否则将导致 HTTPS 模式回源失败。配置成功后，用户至 CDN 节点请求为 HTTP 时，CDN 节点回源请求也为 HTTP；用户至 CDN 节点请求为 HTTPS 时，CDN 节点回源请求也为 HTTPS。
- 若域名源站修改 HTTPS 端口为非 443 端口，会导致配置失败。
- COS 源或 FTP 源域名仅支持 HTTP 回源。

6. 配置成功后，您可以在**证书管理**页面看到已经配置成功的域名以及证书情况，如下图所示：

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：2024-12-03 11:38:42

概述

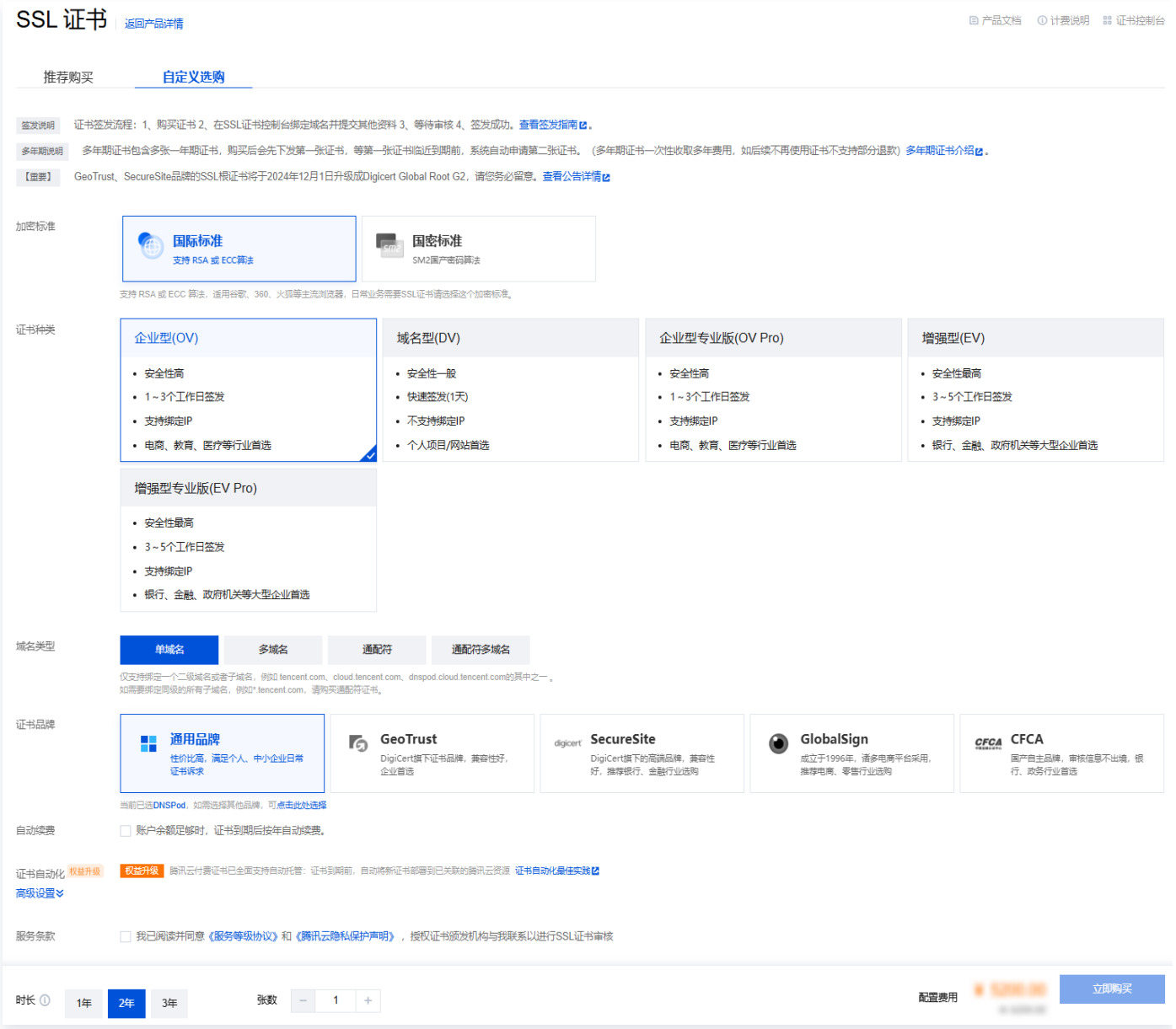
多年期证书是腾讯云 SSL 证书提供的自动审核交付功能，在腾讯云购买1年以上的多年期证书并完成审核后，腾讯云将在前一个 SSL 证书有效期到期前一个月为您自动审核信息并颁发第二张 SSL 证书，无需您重新申请，简化 SSL 证书产品申请和续费时的繁琐流程。
同时，腾讯云 SSL 证书支持云资源托管能力，可自动将新 SSL 证书部署至原 SSL 证书已部署的腾讯云云资源，例如负载均衡、内容分发网络等。
本文档将指导您如何通过两者相结合实现证书签发和资源绑定的全自动交付能力，帮助您实现从多年期证书申请到部署的全自动化。

> **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，如下图所示：



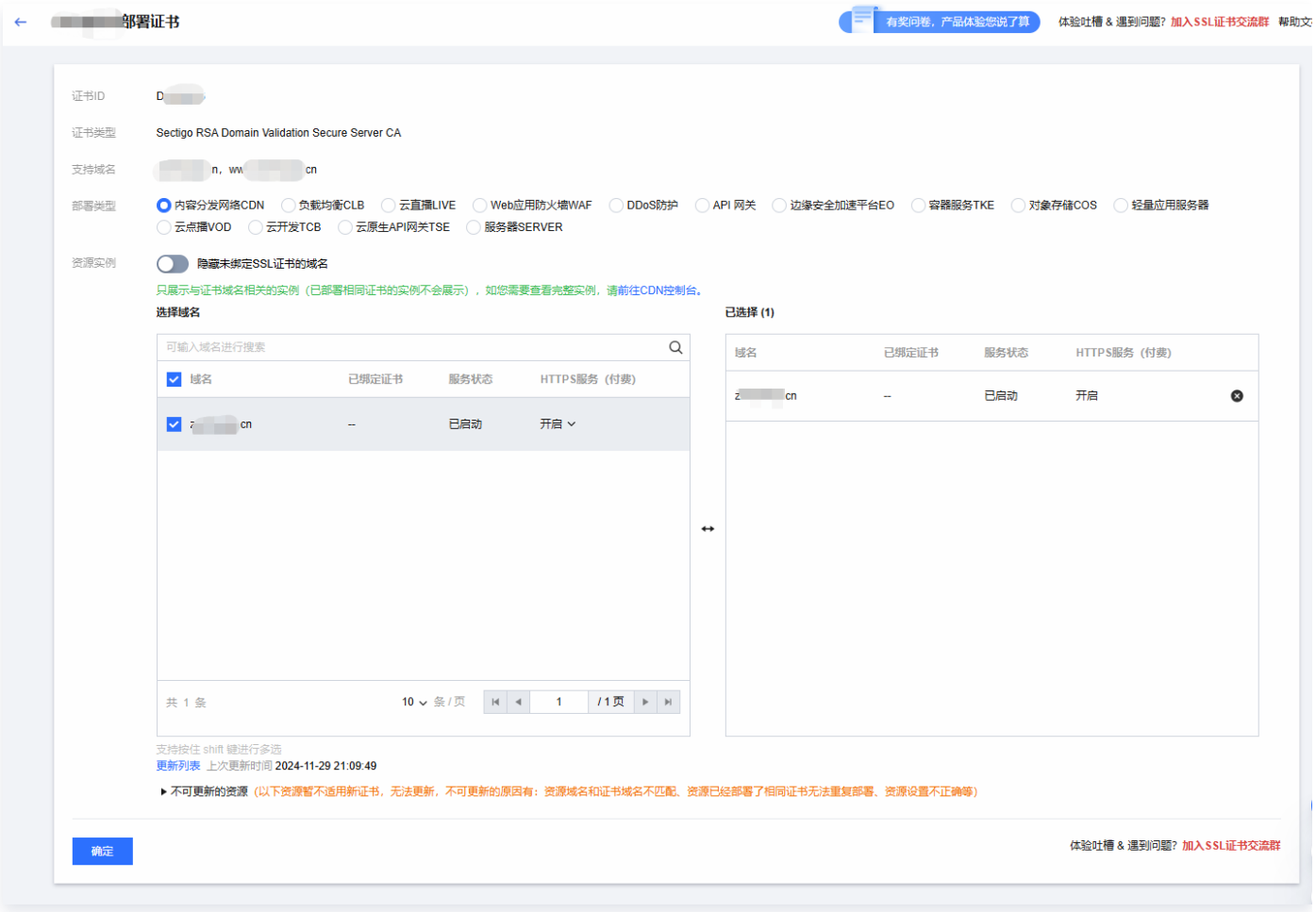
3. 完成购买后，您可按照 SSL 证书申请流程（例如：DV（域名型）SSL 证书提交流程）完成 SSL 证书的申请。

步骤2：SSL 证书部署至云资源

完成申请证书并颁发后，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**，如下图所示：

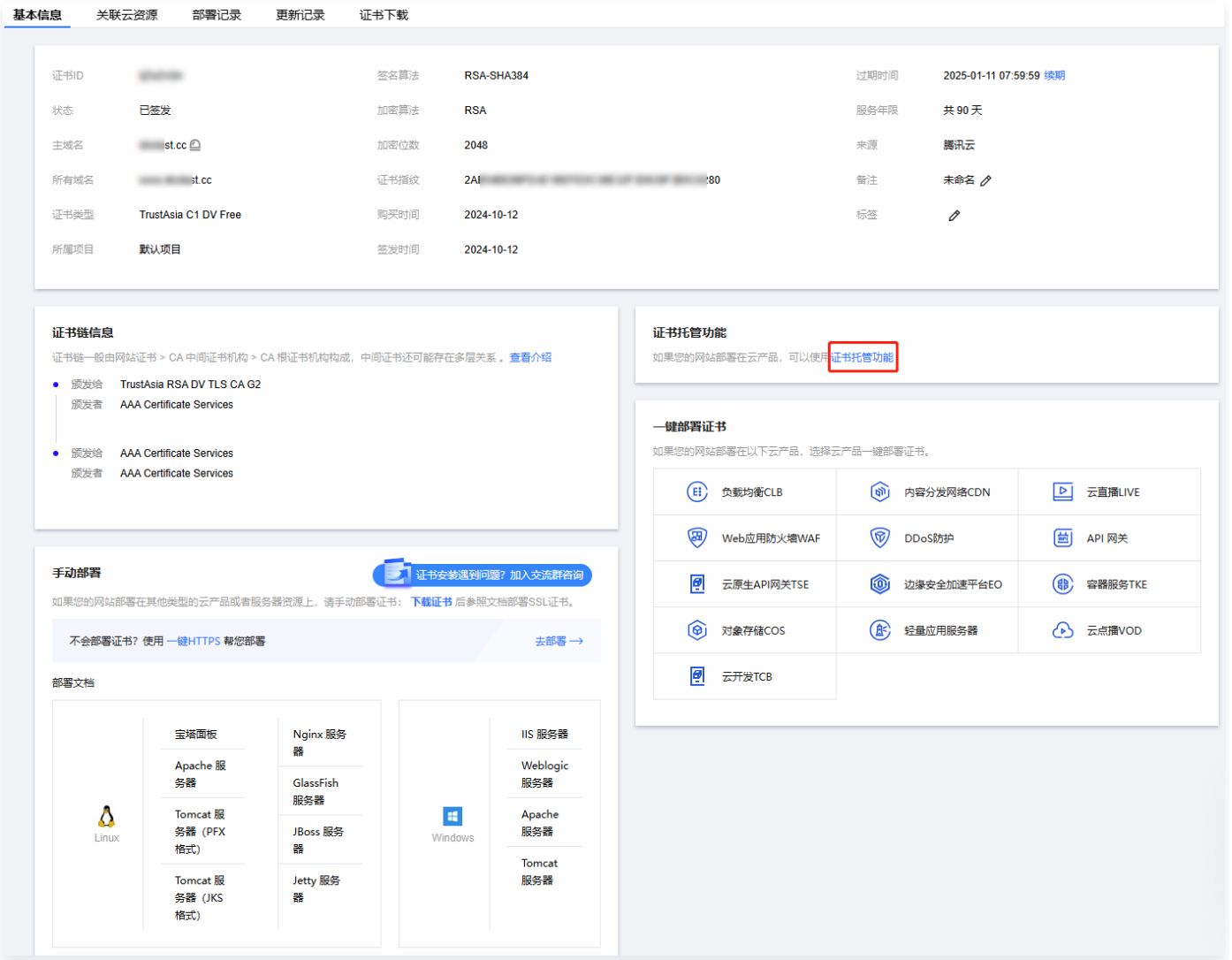
2. 在**部署证书**页面中，选择您需部署类型并勾选对应资源实例。如下图所示：



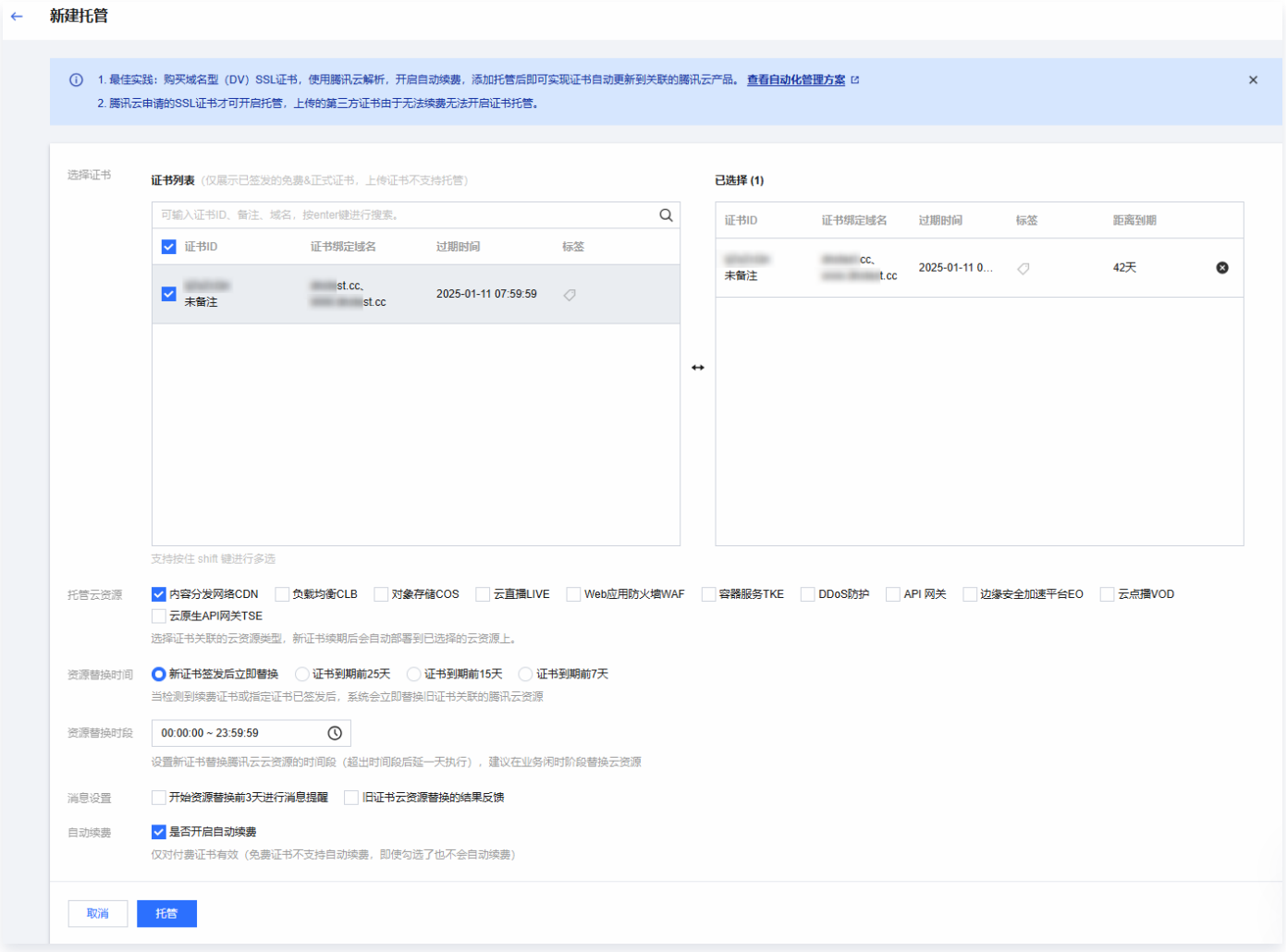
3. 单击**确定**，SSL 证书即可成功部署至对应云资源中。

步骤3：开启云资源托管

1. 单击目标**证书名称**，进入**证书详情**页面。
2. 在**基本信息**模块的**证书托管功能**处，单击**证书托管功能**，如下图所示：



3. 在弹出的**新建托管**窗口中，勾选您需开启的云资源，如下图所示：



4. 单击**托管**，即可完成操作。

苹果 ATS 特性服务器配置指南

最近更新时间：2022-10-11 14:29:45

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:!ADH:!RC4;
    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:!MD5:!ADH:!RC4
        </VirtualHost>
</IfModule>
```

Tomcat 证书配置

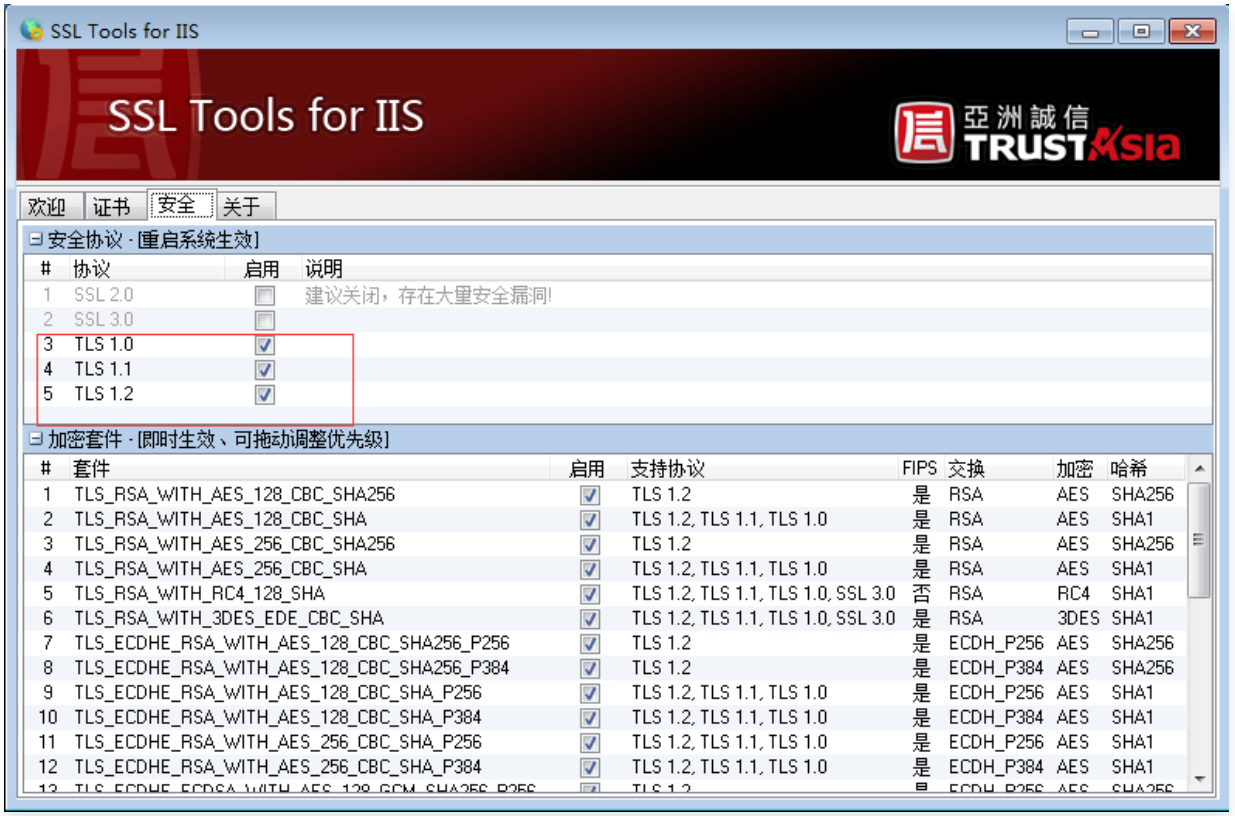
更新 `%TOMCAT_HOME%\conf\server.xml` 文件如下：

```
<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:!MD5:!ADH:!RC4" />
```

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：

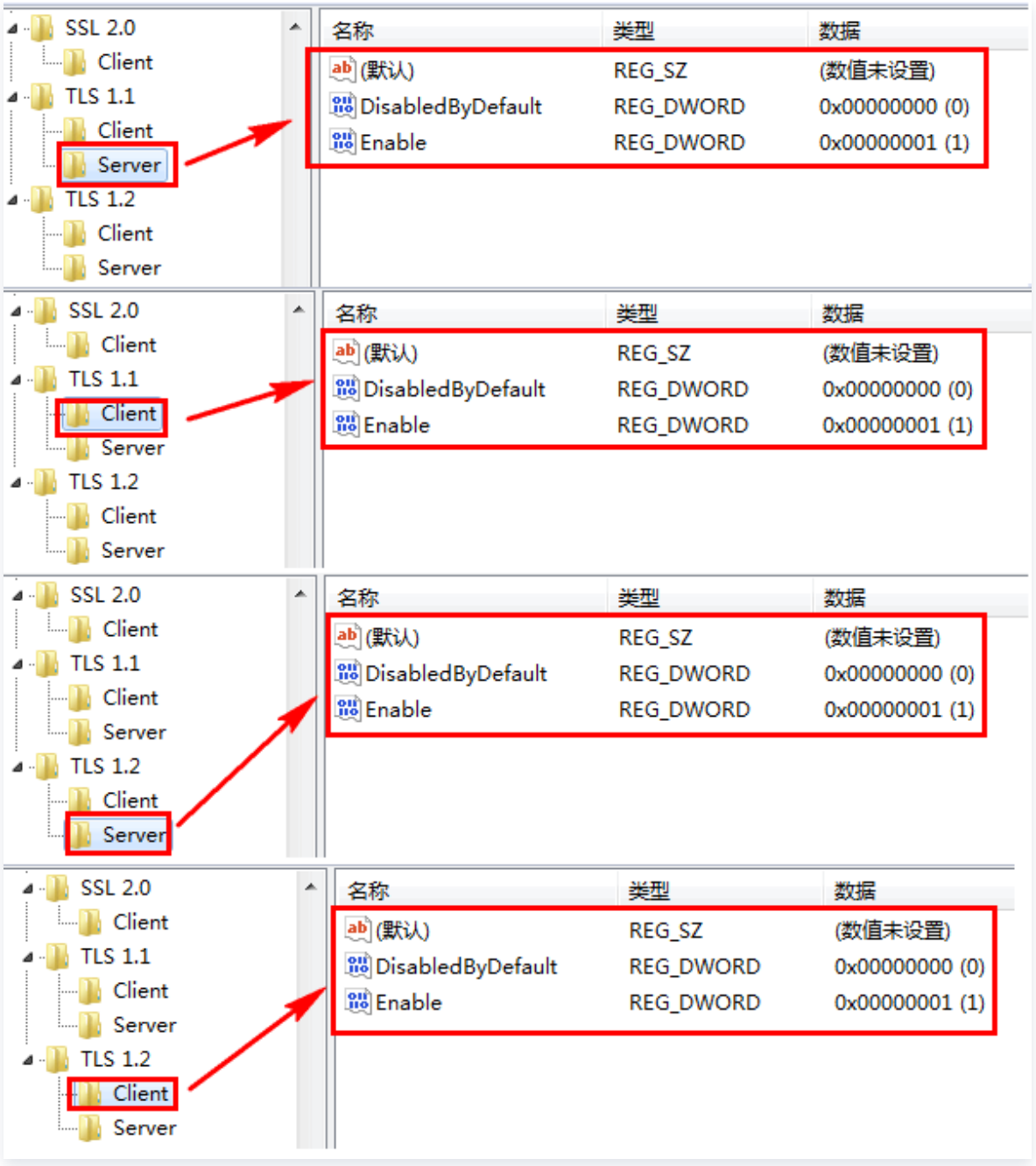


- 勾选三个 TLS 协议并重启系统即可。
- 如果检查到 PFS 不支持，在加密套件中选中带 ECDHE 和 DHE 就可以了。

方法二

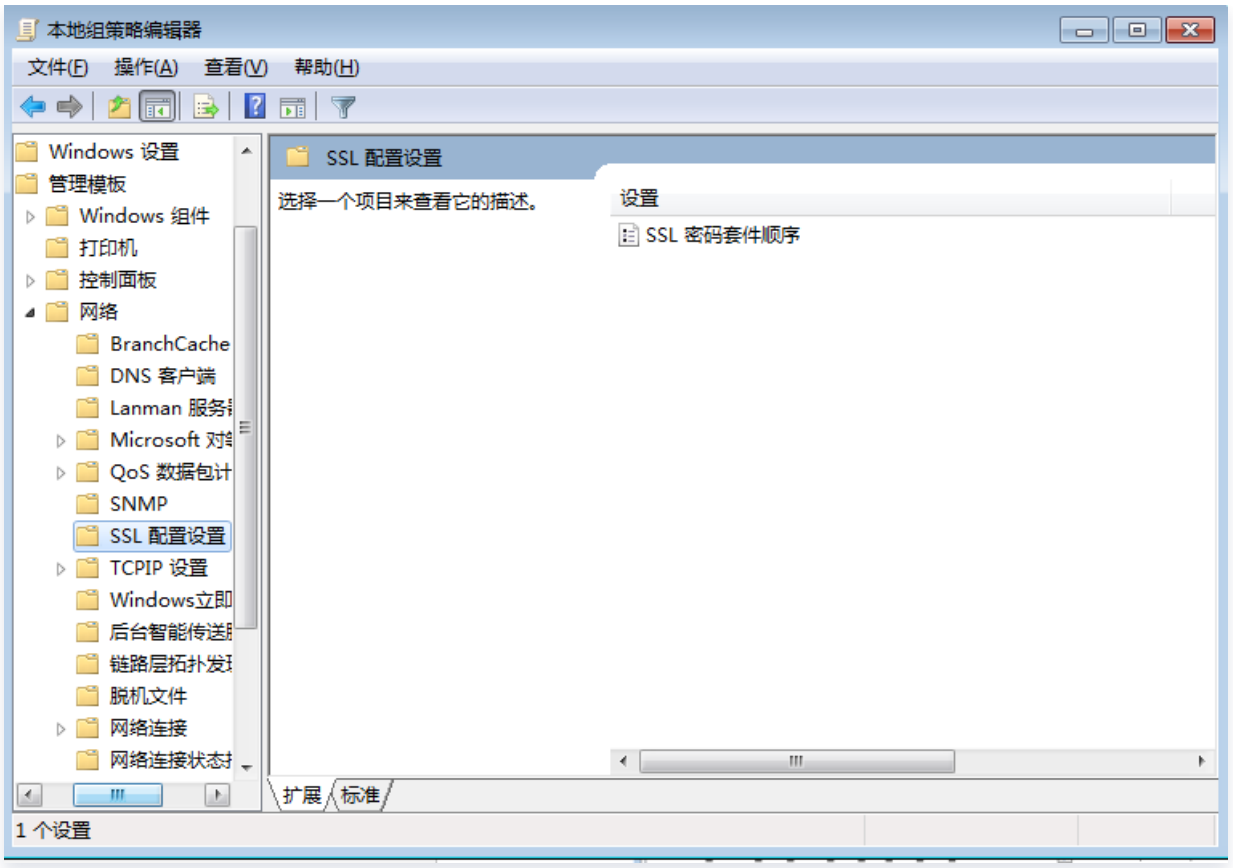
1. 开始——运行，输入 `regedit` 。
2. 找到 HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols 右键->新建->项->新建 TLS 1.1，TLS 1.2。
3. TLS 1.1 和 TLS 1.2 右键->新建->项->新建 Server，Client。
4. 在新建的 Server 和 Client 中新建如下的项（DWORD 32位值），总共4个。如下图所示：
   - DisabledByDefault [Value = 0]

- Enabled [Value = 1]



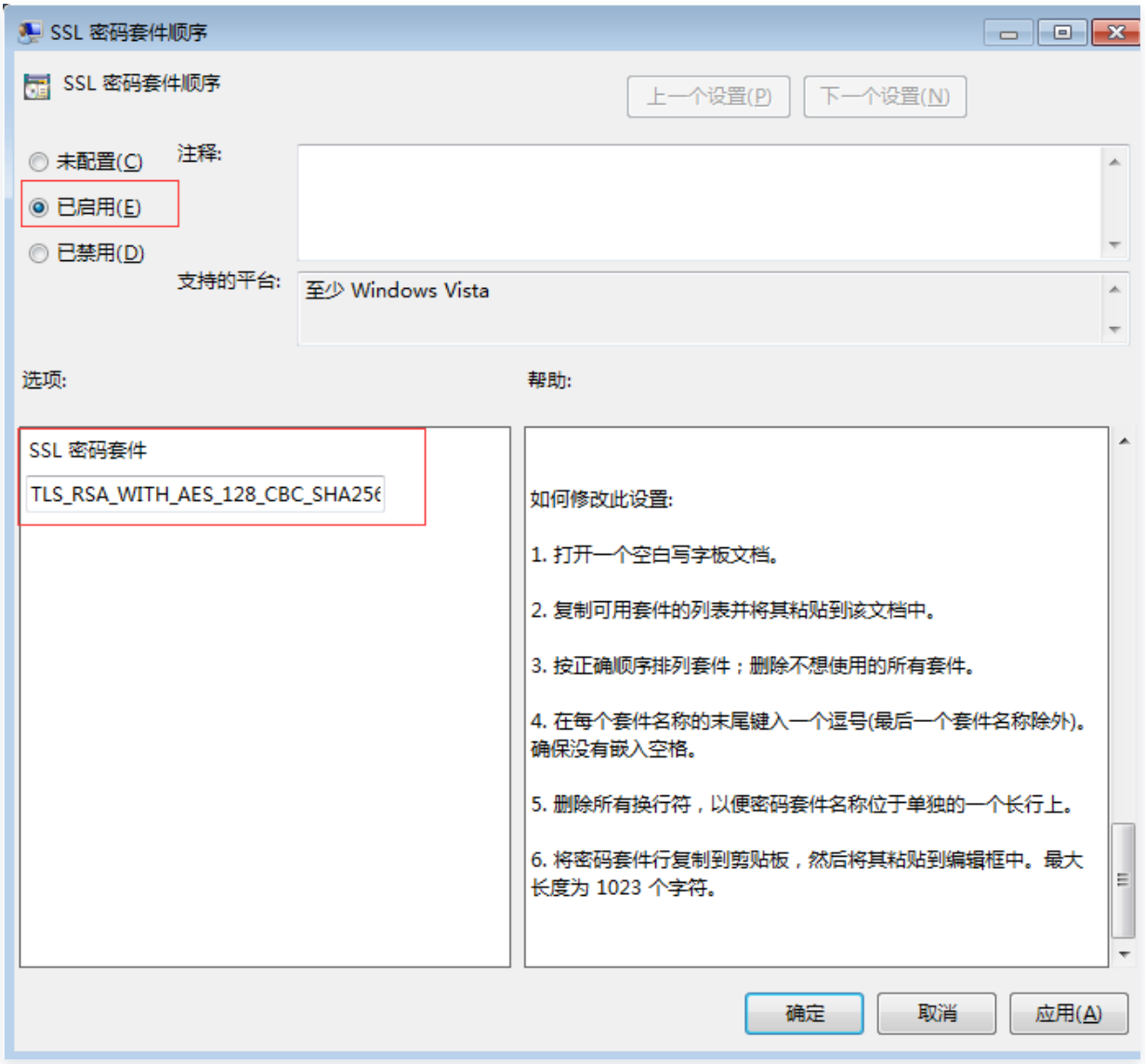
5. 完成后重启系统。
6. 加密套件调整。开始菜单——运行，输入 `gpedit.msc` 进行加密套件调整，在此操作之前需要先开启 TLS1.2 协议。如下图所示：

> **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。


本地组策略编辑器
文件(F) 操作(A) 查看(V) 帮助(H)
Windows 设置
管理模板
Windows 组件
打印机
控制面板
网络
BranchCache
DNS 客户端
Lanman 服务
Microsoft 对
QoS 数据包计
SNMP
SSL 配置设置
TCPIP 设置
Windows立即
后台智能传送
链路层拓扑发
脱机文件
网络连接
网络连接状态
SSL 配置设置
选择一个项目来查看它的描述。
设置
SSL 密码套件顺序
扩展 标准
1 个设置

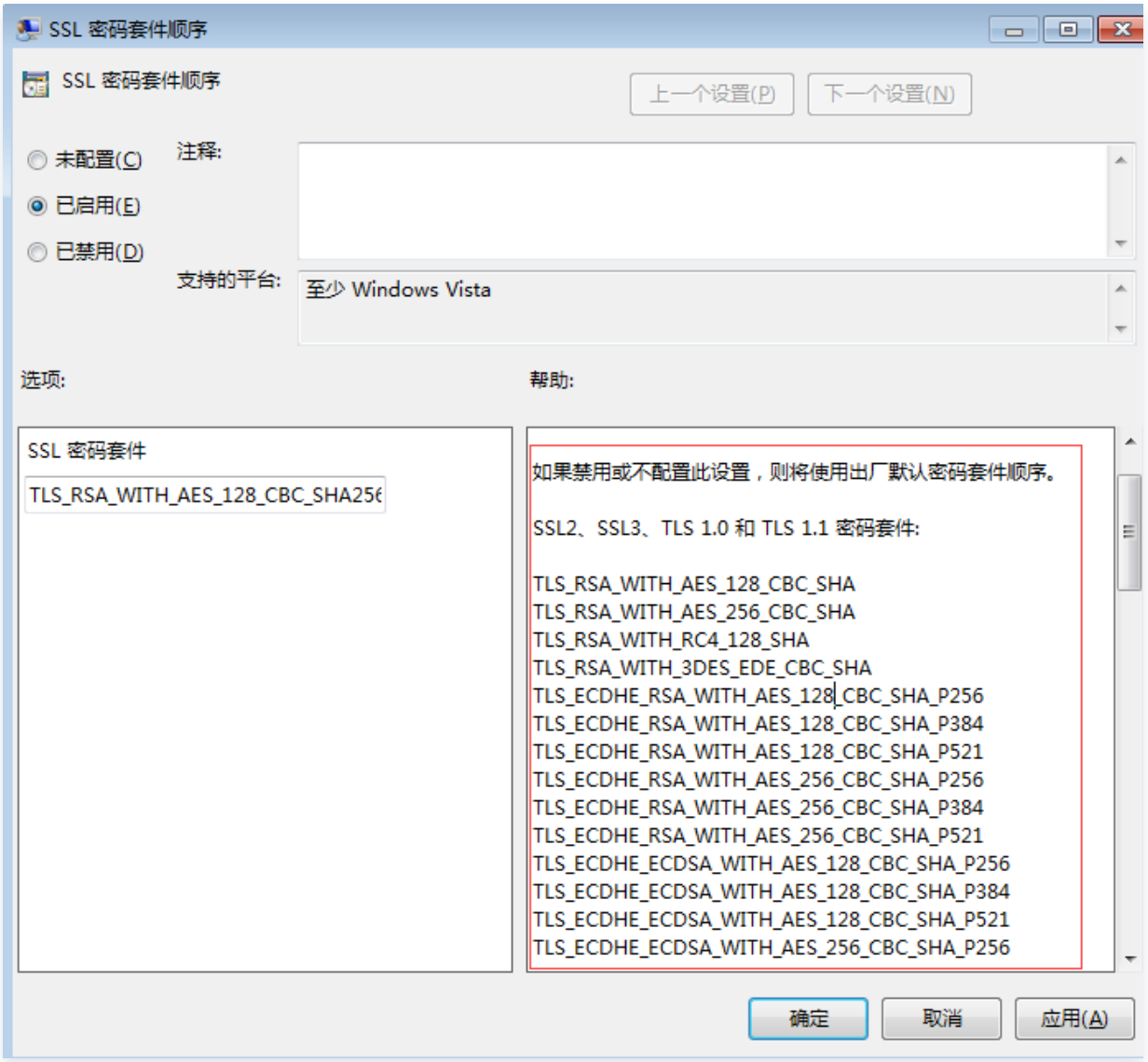
3

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：2024-09-04 14:51:11

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**页面。
2. 在**我的域名**中，单击**添加域名**。如下图所示：

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。如下图所示：

> **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。

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。如下图所示：

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 提示框中的**立即申请**。如下图所示：

2. 在SSL 证书控制台，单击申请免费证书，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：2024-07-01 14:20:11

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

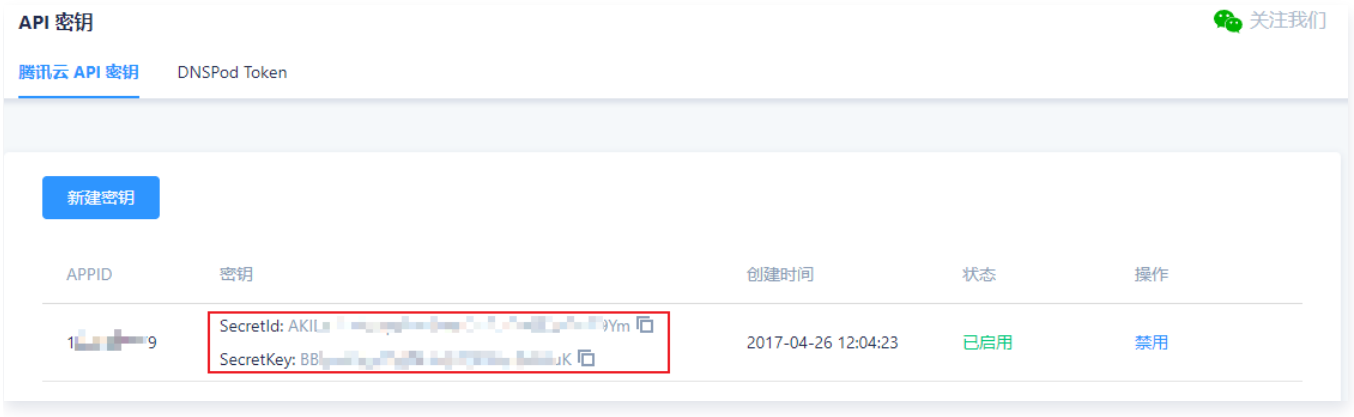
- 拥有群晖（Synology）NAS 管理员权限的账号。
- 拥有腾讯云 DNSPod 账号并完成 实名认证 。
- 群晖（Synology） NAS 拥有公网 IP 地址。
- 拥有1个可用域名并且解析托管在 DNSPod 。

> **说明**
> 若无可用域名，您可前往腾讯云 域名注册购买页 购买您心仪的域名。
> 若您的域名解析未托管在 DNSPod，您可参考 其他平台解析域名平滑转入 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。如下图所示：

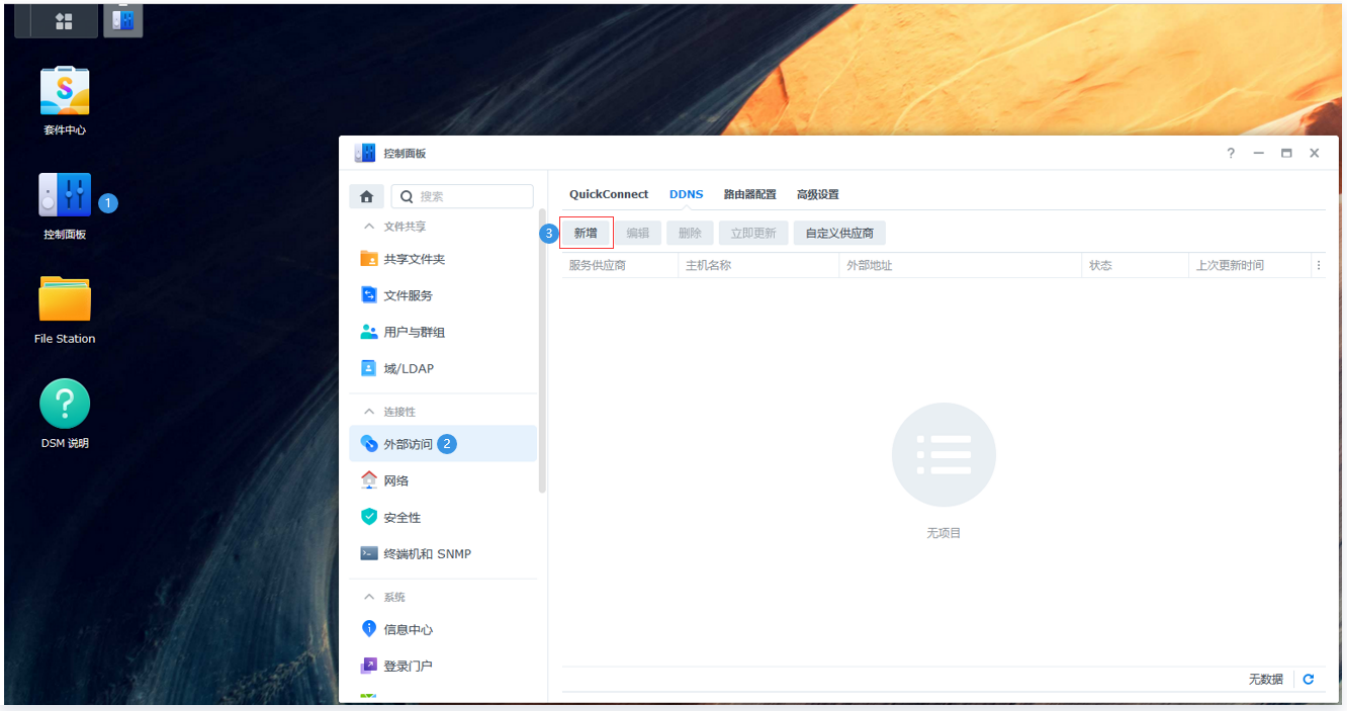


> **注意**

- 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**。如下图所示：



2. 在弹出的 “添加 DDNS” 窗口中，填写相关信息。如下图所示：

- **服务供应商**：请选择**腾讯云**。
- **主机名称**：请填写您的**主机名称**。
- **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**
> - 需要先在**腾讯云 SSL 控制台**进行服务授权，未授权无法自动申请证书。详情请参见 SSL 证书角色策略配置说明 。
> - 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。如下图所示：

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。如下图所示：

   - 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：2023-05-22 16:55:02

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 最新版本，如需安装，请前往 Python 官网 进行下载。
- 已安装 PyCharm 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
```

```
C:\Users\                                    Python\Python310\Scripts>pip list
Package                  Version
------------------------ ---------
certifi                  2021.10.8
charset-normalizer       2.0.12
idna                     3.3
pip                      22.0.4
requests                 2.27.1
setuptools               58.1.0
tencentcloud-sdk-python  3.0.611
urllib3                  1.26.9
```

> ⚠ 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-python
```

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,os
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
TencentCloudSDKException
from tencentcloud.ssl.v20191205 import ssl_client, models

start = time()
# 请使用环境变量获取。不要硬编码
secretId = os.getenv('TENCENT_CLOUD_SECRET_ID')
secretKey = os.getenv('TENCENT_CLOUD_SECRET_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，请直接按回车键
 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()
```

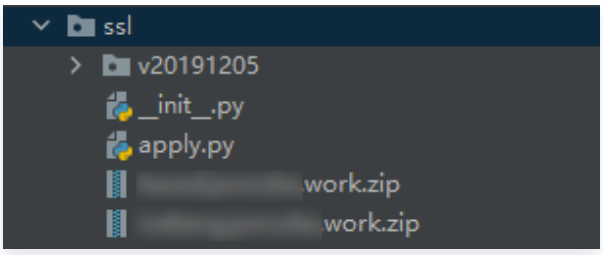
```
print('本次代码执行共耗时：', round(end - start, 2), 's')
```

结果展示

1. 申请批量证书。如下图所示：

```
C:\Users\                                  Python\Python310\python.exe E:/tencent/tencentcloud-sdk-python/tencentcloud/ssl/apply.py
要申请证书的域名:                          .work
要申请证书的域名:                          .work
要申请证书的域名:
域名:                                      .work资料已提交，五秒钟后自动验证
域名:                                      .work验证成功！准备下载证书
域名:                                      .work资料已提交，五秒钟后自动验证
域名:                                      .work验证成功！准备下载证书
本次代码执行共耗时： 220.87 s
```

2. 下载证书内容。如下图所示：



补全 SSL 证书链

最近更新时间：2024-10-17 16:35:03

通常情况下 PC 端浏览器都可以通过 Authority Info Access（权威信息访问）的 URL 链接获得中间证书，但在部分 Android 系统的浏览器上访问时会出现证书不可信或无法访问等问题。
主要原因在于部分 Android 系统的浏览器并不支持通过 Authority Info Access（权威信息访问）的 URL 链接获得中间证书，这时您需要把证书链文件按照 SSL 证书链的结构合并为一个文件重新部署到服务器上，浏览器在与服务器连接时将会下载用户证书和中间证书，使您的浏览器访问时显示为可信证书。SSL 证书链结构如下所示：

```
-----BEGIN CERTIFICATE-----
网站证书
-----END CERTIFICATE-----

-----BEGIN CERTIFICATE-----

CA 中间证书机构
-----END CERTIFICATE-----

-----BEGIN CERTIFICATE-----

CA 根证书机构

-----END CERTIFICATE-----
```

说明

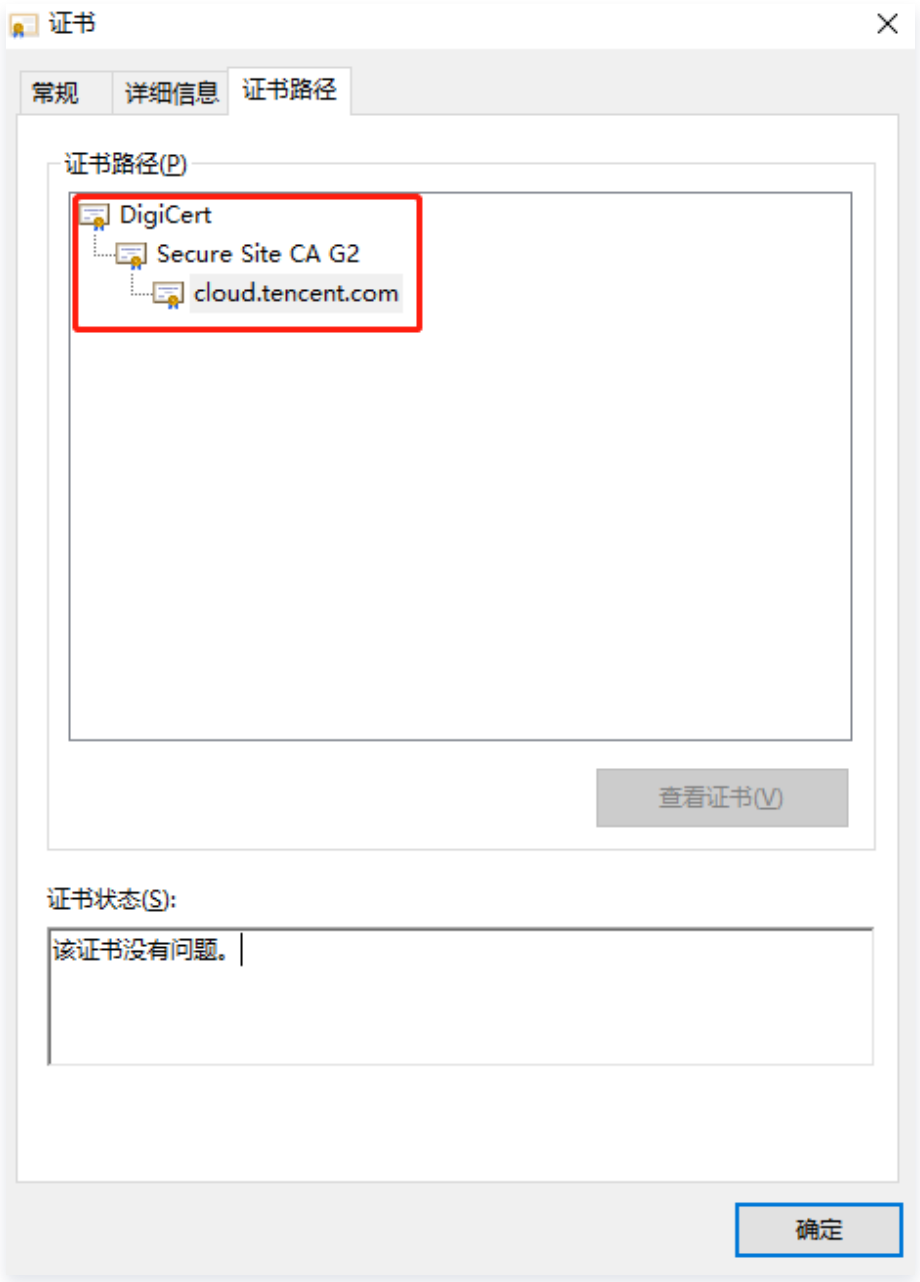
- SSL 证书链的结构，一般是由**网站证书 > CA 中间证书机构 > CA 根证书机构**构成，中间证书还可能存在多层关系。
- 腾讯云提供的 SSL 国际标准证书为完整的证书链，无需进行补齐即可正常使用。

如何查看 SSL 证书链?

1. 打开 Chrome 浏览器访问安装部署 SSL 证书成功的网站。此处以 Chrome 浏览器为例。

2. 在浏览器地址栏单击 小锁图标，并在弹出的信息卡片中，单击**证书**。如下图所示：

3. 在弹出的 “证书” 信息窗口中，单击**证书路径**，即可查看 SSL 证书链。如下图所示：



安装 OpenSSL

最近更新时间：2024-10-22 17:13:32

OpenSSL 是用于安全通信的著名开源密码学工具包，包括主要的密码算法、常见密码和证书封装功能。

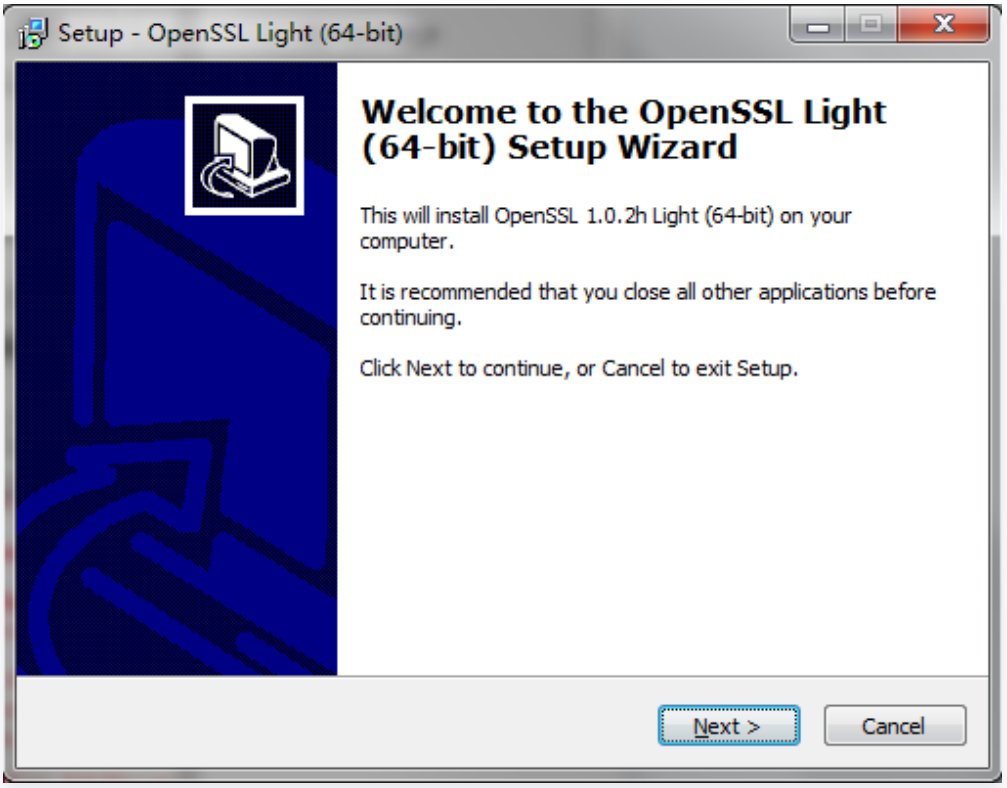
OpenSSL 官网

官方下载地址： 请单击此处 。

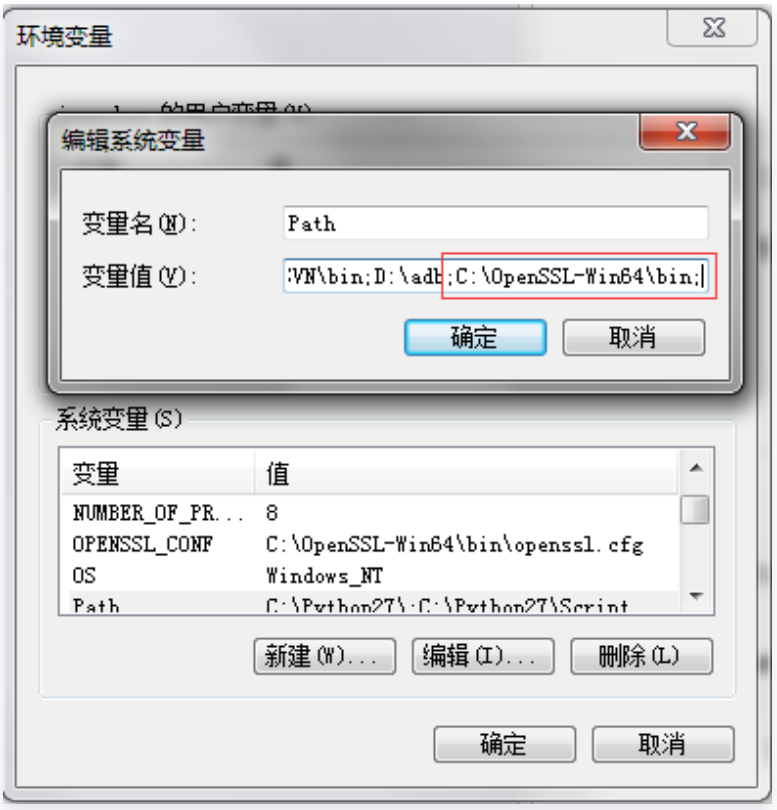
Windows 安装方法

OpenSSL 官网没有提供 Windows 版本的安装包，可以选择其他开源平台提供的工具。请单击此处，以该工具为例，安装步骤和使用方法如下：

1. 选择32位或者64位合适的版本下载，例如 `Win64OpenSSL_Light-1_0_2h.exe` 。如下图所示：



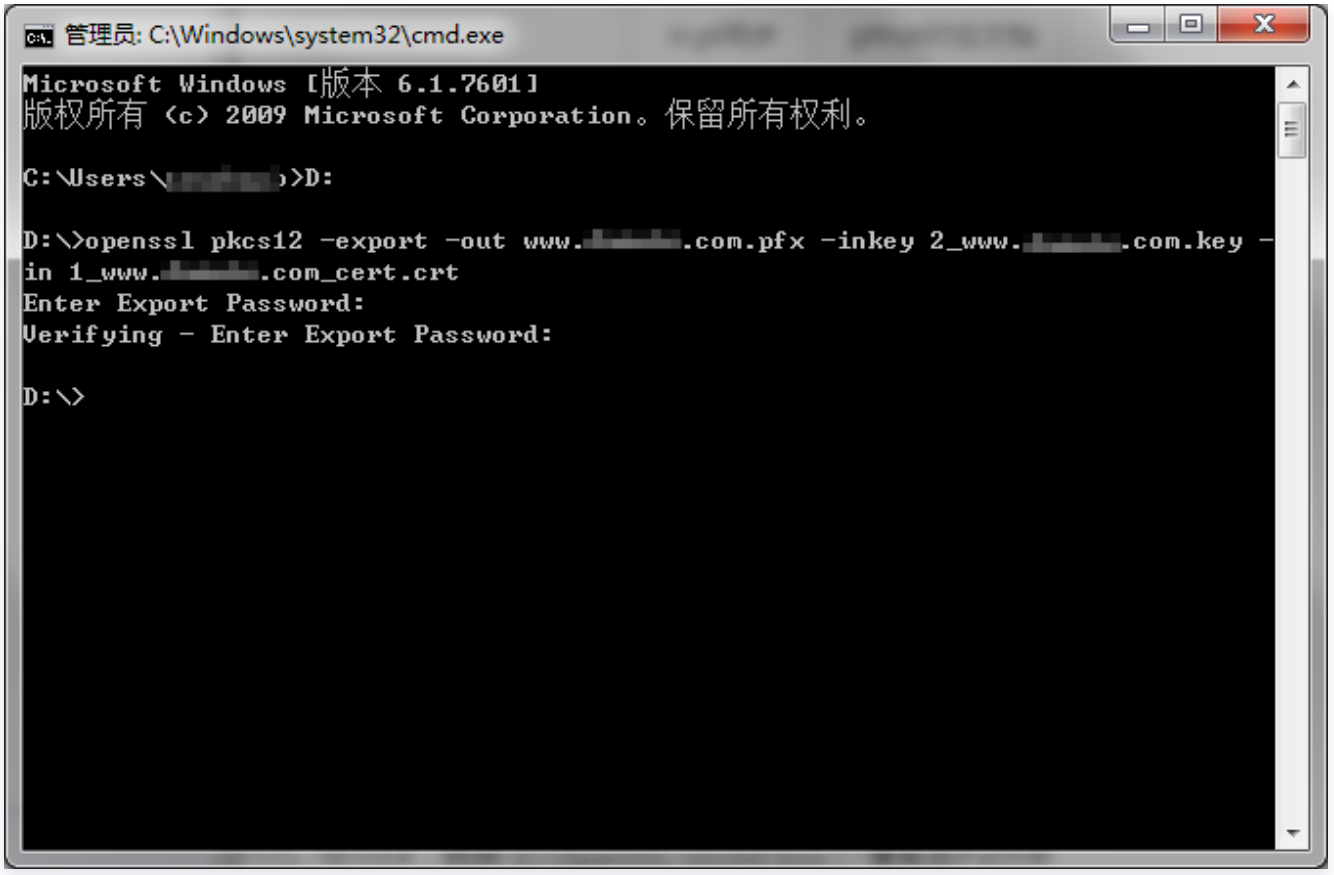
2. 设置环境变量，例如，工具安装在 `C:\OpenSSL-Win64` ，则将 `C:\OpenSSL-Win64\bin;` 复制到 Path 中。如下图所示：



3. **打开命令行程序 cmd（以管理员身份运行），进入** `2_www.tencent.com.key` **、** `1_www.tencent.com_cert.crt` **文件所在目录，运行以下命令。**

```
openssl pkcs12 -export -out www.tencent.com.pfx -inkey
2_www.tencent.com.key -in 1_www.tencent.com_cert.crt
```

例如，key 和 crt 文件保存在 D:\，运行情况如下：



> **注意：**
> Export Password 不需要的情况下，请直接回车不进行输入。

4. 在 D:\ 已生成的 `www.tencent.com.pfx` 文件，可以继续完成在 IIS 管理器中的证书安装。

HTTPS 双向认证指南

最近更新时间：2024-10-29 15:50:42

名词定义

双向认证（又称客户端认证或 SSL/TLS 双向认证）要求客户端和服务器在建立 HTTPS 连接时都要进行身份验证。

概述

HTTPS 双向认证通常用于需要高度安全性的场景，其中服务器和客户端都需要验证对方的身份。此时，**除了配置服务器的证书之外，还需要配置客户端的证书，**以实现通信双方的双向认证功能。

应用场景

1. 企业内部系统：用于内部系统之间的通信，例如在企业内部的微服务架构中，各个服务之间需要进行双向身份验证。
2. 银行和金融机构：用于客户端与银行服务器之间的通信，确保客户端和服务器都是合法且受信任的。
3. 政府机构通信：政府部门间或政府与合作伙伴之间的通信，确保通信双方的身份是可靠的。
4. 医疗保健领域：用于医疗信息系统中，确保只有经过授权的客户端可以访问和传输敏感的健康信息。

基本流程介绍

1. 生成服务器端证书。
2. 配置服务器。
   - 在 Web 服务器（如Apache、Nginx）上安装服务器证书和私钥。
3. 生成客户端证书。
4. 配置服务器以接受客户端证书。
   - 在服务器上配置双向认证以要求客户端提供有效的证书。
5. 配置客户端。
   - 安装客户端证书。
6. 测试连接。
   - 客户端尝试连接到服务器。
   - 服务器验证客户端证书，客户端验证服务器证书。

操作步骤

本章节以腾讯云证书为例，指导您如何在 Nginx 服务器中部署双向认证。

> **说明：**
> - 本文档以证书名称 `server.cloud.tencent.com` 和 `client.cloud.tencent.com` 为例。

- Nginx 版本以 `nginx/1.18.0` 为例。
- 当前服务器的操作系统为 CentOS 7，由于操作系统的版本不同，详细操作步骤略有区别。
- 安装 SSL 证书前，请您在 Nginx 服务器上开启 HTTPS 默认端口 `443`，避免证书安装后无法启用 HTTPS。具体请参见 服务器如何开启443端口?
- SSL 证书文件上传至服务器方法请参见 如何将本地文件拷贝到云服务器。
- 如果您的证书部署在腾讯云负载均衡，请参见 SSL 单向认证和双向认证说明 。

步骤一：购买服务器证书

1. 登录 SSL 证书购买页，购买服务器证书。若您无需指定证书品牌，可进入**推荐购买**选购性价比最高的 SSL 证书。

步骤二：配置服务器

1. 请在 SSL 证书控制台 中选择您需要使用的证书并单击**下载**。
2. **分别**下载服务端 `server.cloud.tencent.com` 和客户端 `client.cloud.tencent.com` 证书。如下图所示：


证书下载
请根据您的服务器类型选择证书下载：
服务器类型
操作
Tomcat（pfx格式）
帮助 | 下载
Tomcat（JKS格式）
帮助 | 下载
Apache
帮助 | 下载
Nginx（适用大部分场景）
下载
腾讯云宝塔面板
下载
IIS
帮助 | 下载
其他
帮助 | 下载
根证书下载
帮助 | 下载
网站代理 HTTPS 服务
不用安装证书，不用纠结各种安全套件的选择，不用担心私钥泄漏，网站代理 HTTPS 服务器 立即使用 助您解决网站 HTTPS 问题。
体验吐槽 & 遇到问题? 加入官方交流群
取消

3. 下载客户端 `client.cloud.tencent.com` 证书的根证书。如下图所示：

4. 将下载得到的证书文件 `server.cloud.tencent.com_bundle.crt` 、 `server.cloud.tencent.com.key` 和 `client.cloud.tencent.com_root.crt` 上传至服务器。
5. 编辑 Nginx 根目录下的 `conf/nginx.conf` 文件。修改内容如下：

> **说明：**
> 由于版本问题，配置文件可能存在不同的写法。例如：Nginx 版本为 `nginx/1.15.0` 以上请使用 `listen 443 ssl` 代替 `listen 443` 和 `ssl on` 。

```
server {
      #SSL 默认访问端口号为 443
      listen 443 ssl;
      #请填写绑定证书的域名
      server_name server.cloud.tencent.com;
      #请填写证书文件的相对路径或绝对路径
      ssl_certificate server.cloud.tencent.com_bundle.crt;
      #请填写私钥文件的相对路径或绝对路径
      ssl_certificate_key server.cloud.tencent.com.key;
      ssl_session_timeout 5m;
      #请按照以下协议配置
      ssl_protocols TLSv1.2 TLSv1.3;
      #请按照以下套件配置，配置加密套件，写法遵循 openssl 标准。
      ssl_ciphers ECDHE-RSA-AES128-GCM-SHA256:HIGH:!aNULL:!MD5:!RC4:!DHE;
      ssl_prefer_server_ciphers on;
      #开启客户端验证
      ssl_verify_client on;
      #请填写客户端根证书文件的相对路径或绝对路径
      ssl_client_certificate client.cloud.tencent.com_root.crt;
      #证书验证深度。腾讯云免费证书建议设置为2
      ssl_verify_depth 2;

      location / {
      #网站主页路径。此路径仅供参考，具体请您按照实际目录操作。
      #例如，您的网站主页在 Nginx 服务器的 /etc/www 目录下，则请修改 root 后面的
html 为 /etc/www。
      root html;
      index  index.html index.htm;
      }
  }
```

6. 在 Nginx 根目录下，通过执行以下命令验证配置文件问题。

```
./sbin/nginx -t
```

   - 若存在，请您重新配置或者根据提示修改存在问题。
   - 若不存在，请执行 步骤7 。

7. 在 Nginx 根目录下，通过执行以下命令重启 Nginx。

```
./sbin/nginx -s reload
```

测试结果

直接使用 `curl` 命令进行访问。

```
curl https://server.cloud.tencent.com
```

返回报错 `400 Bad Request` 。如下图所示：

```
~ curl  https://server.cloud.tencent.com/
<html>
<head><title>400 No required SSL certificate was sent</title></head>
<body>
<center><h1>400 Bad Request</h1></center>
<center>No required SSL certificate was sent</center>
<hr><center>nginx</center>
</body>
</html>
~
```

使用 `curl` 命令带上受信任的客户端证书和私钥进行访问。

```
curl  https://server.cloud.tencent.com --cert
/tmp/other_client.cloud.tencent.com_bundle.crt --key
/tmp/other_client.cloud.tencent.com.key
```

返回成功。如下图:

```
~ curl  https://server.cloud.tencent.com --cert /tmp/client.cloud.tencent.com_bundle.crt --key /tmp/client.cloud.tencent.com.key
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
    body {
        width: 35em;
        margin: 0 auto;
        font-family: Tahoma, Verdana, Arial, sans-serif;
    }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
~
```