SSL Certificates

Certificate Installation

Copyright Notice

Trademark Notice

Service Notice

This document provides an overview of the as-is details of Tencent Cloud's products and services in their entirety or part. The descriptions of certain products and services may be subject to adjustments from time to time.

The commercial contract concluded by you and Tencent Cloud will provide the specific types of Tencent Cloud products and services you purchase and the service standards. Unless otherwise agreed upon by both parties, Tencent Cloud does not make any explicit or implied commitments or warranties regarding the content of this document.

Contact Us

We are committed to providing personalized pre-sales consultation and technical after-sale support. Don't hesitate to contact us at 4009100100 or 95716 for any inquiries or concerns.

Contents

Certificate Installation
Installing an SSL Certificate on a Tencent Cloud Service
How to Automatically Deploy SSL Certificates to Tencent Cloud Resources

Last updated: 2023-09-28 10:24:24

Automatic Certificate Deployment

If you need to deploy the certificate on Tencent Cloud resources such as CDN, Cloud Load Balancer (CLB), Light Application Server, Object Storage (COS), Web Application Firewall, etc., you can use the automatic deployment feature provided by the SSL Certificate Console to automatically deploy the certificate to the cloud resources.

Cloud Resource Type	Automatic Deployment Supported	Automatic Deployment Guide
Cloud Load Balancer (CLB)	This feature is supported.	Guide to Installing and Deploying SSL Certificates to Cloud Load Balancer (CLB)
Content Delivery Network (CDN)	This feature is supported.	Guide to Installing and Deploying SSL Certificates to Content Delivery Network (CDN)
Web Application Firewall (WAF)	This feature is supported.	Guide to Installing and Deploying SSL Certificates to the Web Application Firewall (WAF)
Cloud Object Storage (COS)	This feature is supported.	Guide to Installing and Deploying SSL Certificates to Object Storage (COS)
Tencent Kubernetes Engine (Ingress)	This feature is supported.	Installing and Deploying SSL Certificate to Container Service Ingress
Tencent Cloud Lighthouse	This feature is supported.	Installing and Deploying SSL Certificates to Light Application Servers (LightHouse)

(Lighthouse)		
Anti-DDoS	This feature is supported.	Guide to Installing and Deploying SSL Certificates to Anti-DDoS
Cloud Streaming Services (CSS)	This feature is supported.	Guide to Installing and Deploying SSL Certificates to Cloud Live Broadcasting (CSS)
VOD	This feature is supported.	Coming soon
Cloud Native API Gateway	This feature is supported.	Guide to Installing and Deploying SSL Certificates to API Gateway
EdgeOne	This feature is supported.	Coming soon
Website Construction	Developing	Manual Deployment Guide: Website Construction > SSL Certificate Deployment

Installing an SSL Certificate in CDN

Last updated: 2023-09-28 10:29:57

Scenario

This document describes how to deploy an SSL certificate to CDN.

Preparations

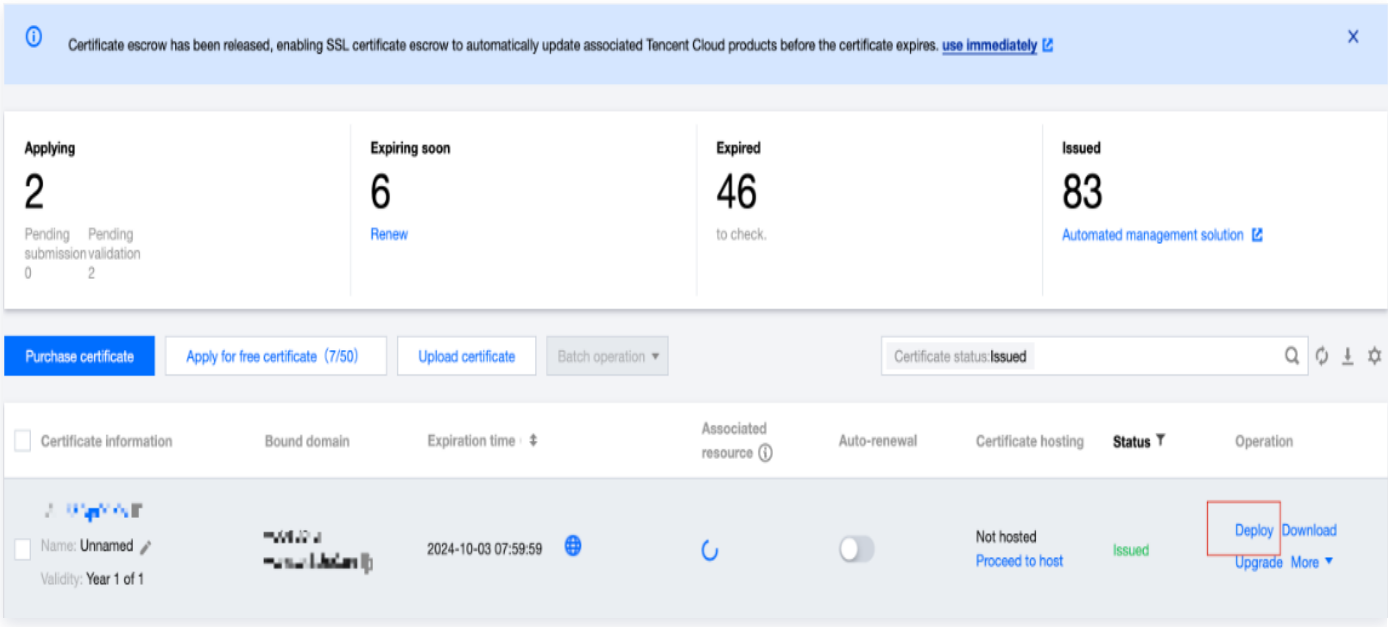
You have successfully applied for and issued an SSL certificate on Tencent Cloud.

Instructions

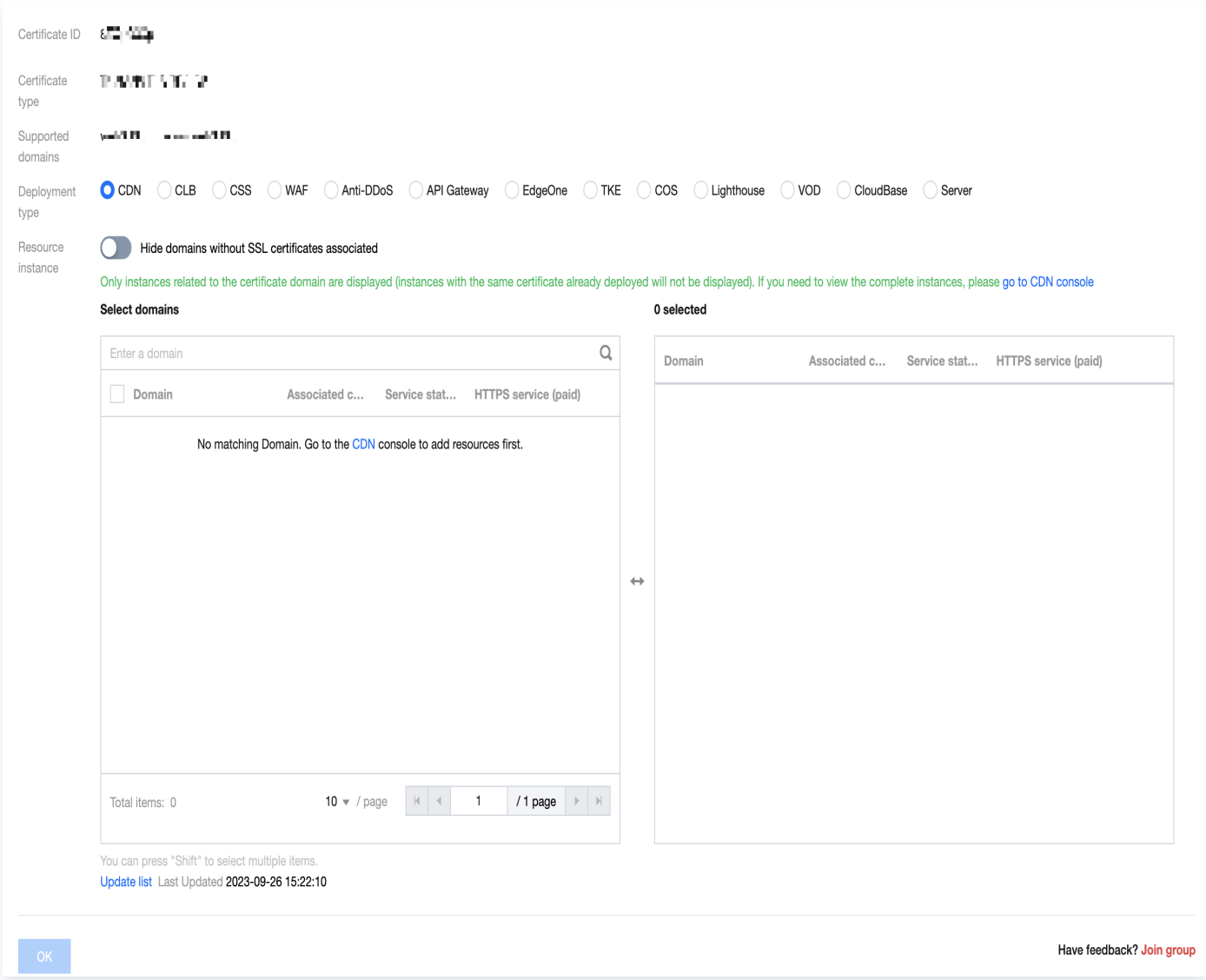
Note

- The domain name must be connected to the CDN and be in the "Deploying" or "Enabled" state. Certificates cannot be deployed to domain names in the "Disabled" state. For specific operations, please refer to Domain Access.
- Once CDN acceleration is activated for COS or CI, the default domain names `.file.myqcloud.com` or `.image.myqcloud.com` cannot be configured with certificates.
- Currently, certificates cannot be configured for SVN hosted origins.

1. Log in to the **SSL Certificate Service console**, navigate to **Certificate List**, select the certificate you wish to deploy, and click **Deploy**.



2. On the **Certificate Deployment** page, select **Content Delivery Network** and choose the CDN resources as needed, as shown below:



3. Click **OK** and wait for the deployment.

How to Enable HTTPS on CDN

> **Note:**
> After the SSL certificate is deployed, **you must enable the HTTPS service in the CDN console to implement HTTPS,** for more information, see HTTPS Configuration Instructions.

Selecting an Installation Type for an SSL Certificate

Last updated: 2023-09-28 16:54:15

Manually Installing a Certificate

You can choose an appropriate method to install a certificate based on the encryption standard of your certificate and your server type.

Note

- With the one-click HTTPS feature, you can upgrade from HTTP to HTTPS without the need for complex SSL certificate deployment. For more details, please refer to One-click HTTPS.
- Currently, 15 methods are available for installing a certificate.

Certificate type	Server System	Certificate Installation Method
International standard certificate (RSA/ECC)	Linux system	Installation of SSL Certificate on Baota Panel
		Apache Server Certificate Installation
		Nginx Server Certificate Installation
		Installation and Deployment of SSL Certificates on Tomcat Server (.jks format)
		Installation and Deployment of SSL Certificate on Tomcat Server (PFX format)
		Installation of GlassFish Server Certificate
		JBoss Server Certificate Installation
		Jetty Server Certificate Installation
	For Windows systems	IIS Server Certificate Installation
		Deploying SSL Certificate on Nginx Server (Windows)
		Weblogic Server Certificate Installation

		Apache Server SSL Certificate Installation and Deployment (Windows)
		Installation and Deployment of SSL Certificate on Tomcat Server (.jks format) (Windows)
Chinese national cryptography standard certificate (SM2)	Linux system	Installation of National Cryptographic Certificate on Apache Server
		Nginx For Linux National Cryptography Certificate Installation
	For Windows systems	Installation of National Cryptographic Certificate for Nginx on Windows Server

Root Certificate Download

If your service needs to be accessed through a non-browser client, you will need to download and install a root certificate. For more details, please refer to SSL Root Certificate Download.

Deploying the Certificate to a Cloud Service

The certificate can be deployed to cloud services in the following ways. Please choose the method that best suits your needs.

- Guide to Deploying SSL Certificates to Cloud Load Balancer (CLB)
- Guide to Deploying SSL Certificates to Content Delivery Network (CDN)
- Guide to Deploying SSL Certificates to the Web Application Firewall (WAF)
- Guide to Deploying SSL Certificates to Anti-DDoS
- Guide to Deploying SSL Certificates to Cloud Live Broadcasting (CSS)
- Deploying SSL Certificate to Container Service Ingress
- Tencent Cloud's Full-site HTTPS Solution

Installation of International Standard Certificates
Installing an SSL Certificate on an Apache Server (Linux)

Last updated: 2023-09-28 16:58:51

Scenario

This document describes how to install an SSL certificate on an Apache server.

Note

- The certificate name `cloud.tencent.com` is used as an example.
- The Apache version used as an example is `Apache/2.4.6`. The default port is `80`. You can download it from the Apache official website. If you need to use other versions, please contact us.
- The current server OS is CentOS 7. Detailed steps vary slightly with the OS version.
- Before installing an SSL certificate, enable port 443 on the Apache server to ensure that HTTPS can be enabled after the certificate is installed. For more information, see How Do I Enable Port 443 for a VM?.
- For detailed directions on how to upload SSL certificate files to a server, see Copying Local Files to CVMs.

Preparations

- A remote file copy tool, such as WinSCP, has been prepared (it is recommended to download the latest version from the official website). If you are deploying to a Tencent Cloud server, it is recommended to use the server's file upload feature. For more information, see Uploading Files to the Cloud Server.
- Install the remote login tool such as PuTTY or Xshell.
- The Apache service has been installed and configured on the current server.
- The data required to install the SSL certificate includes:

Name	Note
Server IP address	The server IP address, which is used to connect the PC to the server.

Username	The username used to log in to the server.
Password	The password used to log in to the server.

> **Note**
> For a CVM instance purchased on the Tencent Cloud official website, log in to the CVM console to get the server IP address, username, and password.

Instructions

Certificate Installation

1. Please navigate to the SSL Certificate Management Console and select the certificate you wish to install, then click **Download**.
2. In the "Certificate Download" window that appears, select **Apache** for the server type, click **Download** and decompress the `cloud.tencent.com` certificate file package to a local directory. After decompression, you can obtain the relevant type of certificate files. This includes the `cloud.tencent.com_apache` folder:
   - **Folder Name:** `cloud.tencent.com_apache`
   - **Folder content:**
     - `root_bundle.crt` : Certificate file
     - `cloud.tencent.com.crt` : Certificate file
     - `cloud.tencent.com.key` : Private key file
   - **CSR file content:** `cloud.tencent.com.csr` file

   > **Note**
   > The CSR file, either uploaded by you or generated online by the system during the certificate application, is provided to the CA. This file can be disregarded during installation.

3. Log in to the Apache server using "WinSCP", a tool for copying files between local and remote computers.

   > **Note**
   > - For detailed directions, see Uploading files via WinSCP to a Linux CVM from Windows.
   > - We recommend using the file upload feature of the Cloud Virtual Machine (CVM) for deployment to Tencent Cloud CVM. For more details, please refer to

Uploading Files to CVM.

4. Copy the obtained certificate file `root_bundle.crt`, the certificate file `cloud.tencent.com.crt`, and the private key file `cloud.tencent.com.key` from the local directory to the `/etc/httpd/ssl` directory on the Apache server.

> **Note**
> If there is no `/etc/httpd/ssl` directory, you can create it using the `mkdir /etc/httpd/ssl` command line.

5. Log in to the Apache server remotely. For instance, using the "PuTTY" tool.

> **Note**
> For a first-time Apache server installation, directories such as `conf.d`, `conf`, and `conf.modules.d` are located by default in the `/etc/httpd` directory.

6. Locate the configuration statement `Include conf.modules.d/*.conf` (used to load the SSL configuration directory) in the httpd.conf configuration file in the `/etc/httpd/conf` directory, and ensure that this configuration statement is not commented out. If it is commented out, remove the comment symbol (`#`) at the beginning of the line and save the configuration file.
7. Locate the configuration statement `LoadModule ssl_module modules/mod_ssl.so` (used to load the SSL module) in the 00-ssl.conf configuration file in the `/etc/httpd/conf.modules.d` directory. Ensure that this configuration statement is not commented out. If it is, remove the comment symbol (`#`) at the beginning of the line and save the configuration file.

> **Note**
> Given the variations in operating system versions, directory structures also differ. Please search according to your actual OS version. If you cannot find the configuration statements `LoadModule ssl_module modules/mod_ssl.so` and `Include conf.modules.d/*.conf` in the above configuration files, please verify if the mod_ssl.so module has been installed. If the mod_ssl.so module is not installed, you can install it by executing the `yum install mod_ssl` command.

8. Edit the ssl.conf configuration file in the `/etc/httpd/conf.d` directory. Make the following changes:

```
<VirtualHost 0.0.0.0:443>
```

```
    DocumentRoot "/var/www/html"
    #Enter the certificate name
    ServerName cloud.tencent.com
    #Enable SSL functionality
    SSLEngine on
    #Path of the certificate file
    SSLCertificateFile /etc/httpd/ssl/cloud.tencent.com.crt
    #Path to the private key file
    SSLCertificateKeyFile /etc/httpd/ssl/cloud.tencent.com.key
    #Path to the certificate chain file
    SSLCertificateChainFile /etc/httpd/ssl/root_bundle.crt
</VirtualHost>
```

9. Restart the Apache server and then you can access it through `https://cloud.tencent.com`.
   - If the security lock icon is displayed in the browser, the certificate has been installed successfully. The details are as shown below:

   - In case of a website access exception, troubleshoot the issue by referring to the following FAQs:
     - Unable to access the website via HTTPS
     - After deploying the SSL certificate, the browser indicates "Website connection is not secure"
     - Is your site indicating an insecure connection?
     - What should I do if I am prompted that HTTPS is not secure after reapplying for deployment upon expiration of the SSL certificate?
     - After deploying the SSL certificate on the server, a 404 error occurs when accessing resources

(Optional) Security configuration for automatic redirect from HTTP to HTTPS

If you need to automatically redirect HTTP requests to HTTPS, you can set it up using the following steps:

1. Edit the httpd.conf configuration file in the `/etc/httpd/conf` directory.

> **Note**
> - Different versions of Apache have different directory structures. For specifics, please refer to the official Apache rewrite documentation.

- The location of the httpd.conf configuration file is not unique. You can search for it one by one according to `/etc/httpd/*`.

2. Please confirm whether the configuration file contains `LoadModule rewrite_module modules/mod_rewrite.so`.
   - If it exists, please remove the comment symbol (`#`) in front of `LoadModule rewrite_module modules/mod_rewrite.so` and proceed to Step 4.
   - If it does not exist, please proceed to Step 3.
3. Please create a new *.conf file in `/etc/httpd/conf.modules.d`, for example, 00-rewrite.conf. Add the following content to the new file:

```
LoadModule rewrite_module modules/mod_rewrite.so
```

4. Add the following content to the httpd.conf configuration file:

```
<Directory "/var/www/html">
# Addition
RewriteEngine on
RewriteCond %{SERVER_PORT} !^443$
RewriteRule ^(.*)?$ https://%{SERVER_NAME}%{REQUEST_URI} [L,R]
</Directory>
```

5. Restart the Apache server and then you can access it through `http://cloud.tencent.com`.

Note

If anything goes wrong during this process, please contact us.

Installing an SSL Certificate on an Nginx Server

Last updated: 2023-09-28 17:10:01

The following video shows you how to install an SSL certificate on an Nginx server:
Watch video

Scenario

This document describes how to install an SSL certificate on an Nginx server.

Note

- The certificate name `cloud.tencent.com` is used as an example.
- The Nginx version `nginx/1.18.0` is used as an example.
- The current server OS is CentOS 7. Detailed steps vary slightly with the OS version.
- Before installing an SSL certificate, please enable the default HTTPS port `443` on the Nginx server to ensure that HTTPS can be enabled after the certificate is installed. For more information, refer to How Do I Enable Port 443 on a Server?
- For detailed directions on how to upload SSL certificate files to a server, see Copying Local Files to CVMs.

Preparations

- A remote file copy tool such as WinSCP has been prepared. It is recommended to download the latest version from the official website.
  If you need to deploy to Tencent Cloud Server, it is suggested to use the file upload function of the cloud server. For more details, please refer to Uploading Files to Cloud Server.
- Install the remote login tool such as PuTTY or Xshell.
- The Nginx service, which includes the `http_ssl_module` module, has been installed and configured on the current server.
- The data required to install the SSL certificate includes:

Name	Note
Server IP address	The server IP address, which is used to connect the PC to the server.

Username	The username used to log in to the server.
Password	The password used to log in to the server.

> **Note**
> For a CVM instance purchased on the Tencent Cloud official website, log in to the CVM console to get the server IP address, username, and password.

Instructions

Certificate Installation

1. Please navigate to the SSL Certificate Service Console and select the certificate you wish to install, then click **Download**.
2. In the "Certificate Download" window that appears, select **Nginx** as the server type, click **Download** and decompress the `cloud.tencent.com` certificate file package to a local directory. After decompression, you can obtain the relevant type of certificate files, which includes the `cloud.tencent.com_nginx` folder:
   - **Folder Name:** `cloud.tencent.com_nginx`
   - **Folder content:**
     - `cloud.tencent.com_bundle.crt` : Certificate file
     - `cloud.tencent.com_bundle.pem` : Certificate file (this file can be ignored)
     - `cloud.tencent.com.key` : Private key file
     - `cloud.tencent.com.csr` : CSR file

     > **Note**
     > The CSR file, either uploaded by you or generated online by the system during the certificate application, is provided to the CA. This file can be disregarded during installation.
3. Log in to the Nginx server using "WinSCP", a tool for copying files between local and remote computers.

> **Note**
> - For detailed directions, see Uploading files via WinSCP to a Linux CVM from Windows .
> - We recommend using the file upload feature of the Cloud Virtual Machine (CVM) for deployment to Tencent Cloud CVM. For more details, please refer to Uploading Files to CVM .

4. Copy the obtained certificate file `cloud.tencent.com_bundle.crt` and the private key file `cloud.tencent.com.key` from the local directory to the `/etc/nginx` directory on the Nginx server (this is the default installation directory for Nginx, please operate according to the actual situation).
5. Log in to the Nginx server remotely, for instance, using the "PuTTY" tool.
6. Edit the `nginx.conf` file in the Nginx root directory. The modifications are as follows:

Note

- If the following content is not found, it can be manually added. You can run the command nginx -t to find the path to the nginx configuration file.

  As illustrated below:

```
# nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
#
```

- You can edit this file by running the `vim /etc/nginx/nginx.conf` command.
- Due to version differences, configuration files may vary. For instance, if the Nginx version is `nginx/1.15.0` or higher, use `listen 443 ssl` instead of `listen 443` and `ssl on`.

```
server {
    #The default port for SSL access is 443.
    listen 443 ssl;
    #Please enter the domain name to bind the certificate to
    server_name cloud.tencent.com;
    #Please specify the relative or absolute path of the certificate file.
    ssl_certificate  cloud.tencent.com_bundle.crt;
    Please enter the relative or absolute path of the private key file
    ssl_certificate_key cloud.tencent.com.key;
    ssl_session_timeout 5m;
    Please configure according to the following protocol
    ssl_protocols TLSv1.2 TLSv1.3;
    #Please follow the cipher suite configuration below, adhering to the OpenSSL
standard.
    ssl_ciphers ECDHE-RSA-AES128-GCM-SHA256:HIGH:!aNULL:!MD5:!RC4:!DHE;
    ssl_prefer_server_ciphers on;
    location / {
```

```
        # Website home path. This path is for reference only, please operate
according to the actual directory.
        #For instance, if your website's homepage is in the /etc/www directory on the
Nginx server, please change the html after root to /etc/www.
        root html;
        index index.html index.htm;
    }
}
```

7. Execute the following command to verify the configuration file issues.

```
nginx -t
```

   - If issues exist, please reconfigure or modify according to the provided suggestions.
   - If it does not exist, please proceed to Step 8.

8. Reload Nginx by executing the following command.

```
nginx -s reload
```

9. Upon successful reloading, you can access the server via `https://cloud.tencent.com`.

(Optional) Security configuration for automatic redirect from HTTP to HTTPS

If you need to automatically redirect HTTP requests to HTTPS, you can set it up using the following steps:

1. Choose the following configuration methods according to your actual needs:
   - Add a JS script to the page.
   - Add redirection in the backend program.
   - Implement redirection through the web server.
   - Nginx supports the rewrite function. If you did not remove pcre during compilation, you can add `return 301 https://$host$request_uri;` in the HTTP server to redirect the default port 80 request to HTTPS. Modify the following content:

     Note
     - Uncommented configuration statements can be configured as shown below.
     - Due to version differences, configuration files may vary. For instance, if the Nginx version is `nginx/1.15.0` or higher, use `listen 443 ssl` instead of `listen 443` and `ssl on`.

```
server {
 #The default port for SSL access is 443.
 listen 443 ssl;
 #Please enter the domain name to bind the certificate to
 server_name cloud.tencent.com;
 #Please specify the relative or absolute path of the certificate file.
 ssl_certificate  cloud.tencent.com_bundle.crt;
 Please enter the relative or absolute path of the private key file
 ssl_certificate_key cloud.tencent.com.key;
 ssl_session_timeout 5m;
 #Please follow the cipher suite configuration below, adhering to the OpenSSL
standard.
 ssl_ciphers ECDHE-RSA-AES128-GCM-
SHA256:ECDHE:ECDH:AES:HIGH:!NULL:!aNULL:!MD5:!ADH:!RC4;
 Please configure according to the following protocol
 ssl_protocols TLSv1.2 TLSv1.3;
 ssl_prefer_server_ciphers on;
 location / {
   # Website home path. This path is for reference only, please operate
according to the actual directory.
   #For instance, if your website's homepage is in the /etc/www directory on the
Nginx server, please change the html after root to /etc/www.
   root html;
   index index.html index.htm;
 }
}
server {
 listen 80;
 #Please enter the domain name to bind the certificate to
 server_name cloud.tencent.com;
 #Convert HTTP domain requests to HTTPS
 return 301 https://$host$request_uri;
}
```

2. Execute the following command to verify the configuration file issues.

```
nginx -t
```

   - If issues exist, please reconfigure or modify according to the provided suggestions.
   - If it does not exist, please proceed to Step 3.

3. Reload Nginx by executing the following command.

```
nginx -s reload
```

4. Upon successful reloading, you can access the server via `https://cloud.tencent.com`.
   - If the security lock icon is displayed in the browser, the certificate has been installed successfully. The details are as shown below:

   - In case of a website access exception, troubleshoot the issue by referring to the following FAQs:
     - Unable to access the website via HTTPS
     - After deploying the SSL certificate, the browser indicates "Website connection is not secure"
     - Is your site indicating an insecure connection?
     - What should I do if I am prompted that HTTPS is not secure after reapplying for deployment upon expiration of the SSL certificate?
     - After deploying the SSL certificate on the server, a 404 error occurs when accessing resources

> **Note**
> If anything goes wrong during this process, please contact us.

SSL Certificate Installation and Deployment on Nginx Server (Windows)

Last updated: 2023-09-28 17:15:09

Scenario

This guide illustrates how to install an SSL certificate on a **Windows operating system**.

Note:

- The certificate name cloud.tencent.com is used as an exemplar in this document.
- The version of Nginx used as an example in this document is nginx/1.24.0.
- The operating system of the current server is **Windows Server 2022**. Detailed procedures may vary slightly due to differences in operating system versions.
- Before installing the SSL certificate, please ensure that the default HTTPS port 443 is enabled on your server to prevent any issues with enabling HTTPS after the certificate is installed.

Instructions

Certificate Installation

1. Please navigate to the SSL Certificate Service Console, select the certificate you wish to install, and click **Download**.
2. In the "Certificate Download" window that appears, select **Nginx** as the server type, click **Download** and decompress the `cloud.tencent.com` certificate file package to a local directory. After decompression, you can obtain the relevant type of certificate files, which includes the `cloud.tencent.com_nginx` folder:
   - **Folder Name:** `cloud.tencent.com_nginx`
   - **Folder content:**
     - `cloud.tencent.com_bundle.crt` : Certificate file
     - `cloud.tencent.com_bundle.pem` : Certificate file (this file can be ignored)
     - `cloud.tencent.com.key` : Private key file
     - `cloud.tencent.com.csr` : CSR file

> **Note:**
> The CSR file, either uploaded by you or generated online by the system during the certificate application, is provided to the CA. This file can be disregarded during installation.

3. Copy the obtained certificate file `cloud.tencent.com_bundle.crt` and the private key file `cloud.tencent.com.key` from the local directory to the server's `C:\nginx\` directory (this can be any directory that does not contain Chinese characters or spaces, please operate according to the actual situation).
4. Edit the `conf\nginx.conf` file located in the same directory as nginx.exe. The modifications are as follows:

```
server {
    # The default port for SSL access is 443.
    listen 443 ssl;
    #Please enter the domain name to which the certificate is bound.
    server_name cloud.tencent.com;
    #Please enter the absolute path of the certificate file. This path is for reference only, please operate according to the actual directory.
    ssl_certificate C:\\nginx\\certificates\\cloud.tencent.com_bundle.crt;
    #Please enter the absolute path of the private key file. This path is for reference only, please operate according to the actual directory.
    ssl_certificate_key C:\\nginx\\certificates\\cloud.tencent.com.key;
    ssl_session_timeout 5m;
    # Please configure according to the following protocol
    ssl_protocols TLSv1.2 TLSv1.3;
    #Please follow the suite configuration below to set up the cipher suite, adhering to the OpenSSL standard.
    ssl_ciphers ECDHE-RSA-AES128-GCM-SHA256:HIGH:!aNULL:!MD5:!RC4:!DHE;
    ssl_prefer_server_ciphers on;
    location / {
        # Website home path. This path is for reference only, please operate according to the actual directory.
        root C:\\html;
        index  index.html index.htm;
    }
 }
```

5. Execute the following command to verify the configuration file issues.

```
.\nginx.exe -t
```

- If issues exist, please reconfigure or modify according to the provided suggestions.
- If it does not exist, please proceed to Step 6.

6. Initiate Nginx by executing the following command.

```
start .\nginx.exe
```

7. Upon successful startup, you can access the server via `https://cloud.tencent.com`.
   - If the security lock icon is displayed in the browser, the certificate has been installed successfully.

   - In case of a website access exception, troubleshoot the issue by referring to the following FAQs:
     - Unable to access the website via HTTPS
     - After deploying the SSL certificate, the browser indicates "Website connection is not secure"
     - Is your site indicating an insecure connection?
     - What should I do if I am prompted that HTTPS is not secure after reapplying for deployment upon expiration of the SSL certificate?
     - After deploying the SSL certificate on the server, a 404 error occurs when accessing resources

> ⚠ **Note:**
> If anything goes wrong during this process, please contact us.

Installing an SSL Certificate (JKS Format) on a Tomcat Server (Linux)

Last updated: 2023-10-07 16:12:43

Scenario

This document describes how to install an SSL certificate (JKS format) on a Tomcat server.

Note

- The certificate name `cloud.tencent.com` is used as an example.
- The `tomcat-9.0.56` version is used as an example.
- The current server OS is CentOS 7. Detailed steps vary slightly by OS.
- Before installing an SSL certificate, please enable port 443 on the Tomcat server to ensure that HTTPS can be enabled after the certificate is installed. For more information, refer to How Do I Enable Port 443 for a VM?.
- For detailed directions on how to upload SSL certificate files to a server, see Copying Local Files to CVMs.

Preparations

- A remote file copy tool such as WinSCP has been prepared. It is recommended to download the latest version from the official website.
  If you need to deploy to Tencent Cloud CVM, it is suggested to use the file upload feature of the CVM. For more information, see Uploading Files to CVM.
- Install the remote login tool such as PuTTY or Xshell.
- The Tomcat service has been installed and configured on the current server.
- The data required to install the SSL certificate includes:

Name	Note
Server IP address	The server IP address, which is used to connect the PC to the server.
Username	The username used to log in to the server.
Password	The password used to log in to the server.

Note

- For a CVM instance purchased on the Tencent Cloud official website, log in to the CVM console to get the server IP address, username, and password.
- If you have chosen the "Paste CSR" method when applying for an SSL certificate, or if the brand of the certificate you purchased is Wotrus, then the download of the JKS certificate file is not provided. You will need to manually convert the format to generate a keystore. The operation method is as follows:
  - Access the conversion tool.
  - Upload the certificate and private key files from the Nginx folder to the conversion tool, fill in the keystore password, click submit, and convert it into a JKS format certificate.
  - The current Tomcat service is installed by default in the /usr directory. For instance, if the Tomcat folder is named Tomcat-9.0.56, then its configuration file directory would be: /usr/Tomcat-9.0.56/conf.

- If you have chosen the "Paste CSR" method when applying for an SSL certificate, or if the brand of the certificate you purchased is Wotrus, then the download of the JKS certificate file is not provided. You will need to manually convert the format to generate a keystore. The operation method is as follows:
- Access the conversion tool.
- Upload the certificate and private key files from the Nginx folder to the conversion tool, input the keystore password, click **Submit**, and convert them into a JKS format certificate.
- The current Tomcat service is installed by default in the /usr directory. For instance, if the Tomcat folder is named Tomcat-9.0.56, then its configuration file directory would be: `/usr/Tomcat-9.0.56/conf`.

Instructions

Certificate Installation

1. Please navigate to the SSL Certificate Service Console and select the certificate you wish to install, then click **Download**.
2. In the "Certificate Download" window that appears, select **JKS** for the server type, click **Download**, and decompress the `cloud.tencent.com` certificate file package to a local directory. After decompression, you can obtain the relevant type of certificate files. This includes the `cloud.tencent.com_jks` folder:
   - **Folder Name:** `cloud.tencent.com_jks`
   - **Folder content:**
     - `cloud.tencent.com.jks` : Key Store

- `keystorePass.txt` password file (If a private key password has been set, there will be no `keystorePass.txt` password file)

3. Utilize WinSCP (a tool for transferring files between a local and a remote computer) to log into the Tomcat server. Then, copy the obtained keystore file `cloud.tencent.com.jks` from the local directory to the Tomcat configuration file directory `/usr/Tomcat-9.0.56/conf`.

> **Note**
> - For detailed directions, see Uploading files via WinSCP to a Linux CVM from Windows.
> - We recommend using the file upload feature of the Cloud Virtual Machine (CVM) for deployment to CVM. For more details, please refer to Uploading Files to CVM.

4. In the `/usr/Tomcat-9.0.56/conf` directory, add the following content to the `server.xml` file:

```
<Connector port="443" protocol="HTTP/1.1" SSLEnabled="true"
 maxThreads="150" scheme="https" secure="true"
#Path of the certificate
 keystoreFile="Tomcat installation directory/conf/cloud.tencent.com.jks"
#Keystore password
 keystorePass="******"
 clientAuth="false"/>
```

The main parameters of the configuration file are described as below:

- **keystoreFile**: The location of the keystore file. You can specify an absolute path or a path relative to the <CATALINA_HOME> (Tomcat installation directory) environment variable. If this item is not set, Tomcat will read a file named ".keystore" from the current operating system user's home directory by default.
- **KeystorePass**: This is the keystore password, which specifies the password for the keystore. If you have set a private key password when applying for the certificate, please enter the private key password. If you haven't set a private key password when applying for the certificate, please enter the password found in the keystorePass.txt file in the Tomcat folder.
- **clientAuth**: If set to true, it indicates that Tomcat requires all SSL clients to present a security certificate for identity verification.

For detailed content, refer to the `server.xml` file:

> **Note**
> To avoid format issues, you are not advised to copy the content of `server.xml` directly.

```
<?xml version="1.0" encoding="UTF-8"?>
<Server port="8005" shutdown="SHUTDOWN">
  <Listener className="org.apache.catalina.startup.VersionLoggerListener" />
  <Listener className="org.apache.catalina.core.AprLifecycleListener"
SSLEngine="on" />
  <Listener className="org.apache.catalina.core.JreMemoryLeakPreventionListener"
/>
  <Listener
className="org.apache.catalina.mbeans.GlobalResourcesLifecycleListener" />
  <Listener
className="org.apache.catalina.core.ThreadLocalLeakPreventionListener" />
<GlobalNamingResources>
  <Resource name="UserDatabase" auth="Container"
        type="org.apache.catalina.UserDatabase"
        description="User database that can be updated and saved"
        factory="org.apache.catalina.users.MemoryUserDatabaseFactory"
        pathname="conf/tomcat-users.xml" />
</GlobalNamingResources>
 <Service name="Catalina">
    <Connector port="80" protocol="HTTP/1.1" connectionTimeout="20000"
redirectPort="8443" />
    <Connector port="443" protocol="HTTP/1.1"
         maxThreads="150" SSLEnabled="true" scheme="https" secure="true"
         clientAuth="false"
          keystoreFile="Tomcat installation directory/conf/cloud.tencent.com.jks"
          keystorePass="**" />
    <Connector port="8009" protocol="AJP/1.3" redirectPort="8443" />
  <Engine name="Catalina" defaultHost="cloud.tencent.com">
    <Realm className="org.apache.catalina.realm.LockOutRealm">
    <Realm className="org.apache.catalina.realm.UserDatabaseRealm"
         resourceName="UserDatabase"/>
    </Realm>
   <Host name="cloud.tencent.com"  appBase="webapps"
    unpackWARs="true" autoDeploy="true" >
    <Context path="" docBase ="Knews" />
    <Valve className="org.apache.catalina.valves.AccessLogValve"
directory="logs"
       prefix="localhost_access_log" suffix=".txt"
       pattern="%h %l %u %t &amp;quot;%r&amp;quot; %s %b" />
   </Host>
  </Engine>
 </Service>
</Server>
```

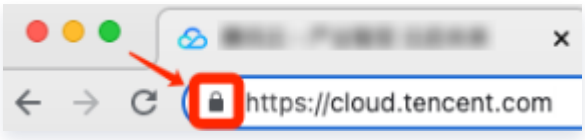
5. Verify if the Tomcat server is running.
   - If already started, you need to execute the following commands in sequence in the `bin` directory of the Tomcat installation directory (for example, `/usr/Tomcat-9.0.56/bin`) to shut down and restart the Tomcat service.

```
./shutdown.sh (Shut down the Tomcat service)
./startup.sh (Start the Tomcat service)
```

   - If not already running, you need to execute the following command in the `bin` directory of the Tomcat installation directory (for example, `/usr/Tomcat-9.0.56/bin`) to start the Tomcat service.

```
./startup.sh
```

6. Upon successful startup, you can access it through `https://cloud.tencent.com`.
   - If the security lock icon is displayed in the browser, the certificate has been installed successfully. The details are as shown below:



   - In case of a website access exception, troubleshoot the issue by referring to the following FAQs:
     - Unable to access the website via HTTPS
     - After deploying the SSL certificate, the browser indicates "Website connection is not secure"
     - Is your site indicating an insecure connection?
     - What should I do if I am prompted that HTTPS is not secure after reapplying for deployment upon expiration of the SSL certificate?
     - After deploying the SSL certificate on the server, a 404 error occurs when accessing resources

(Optional) Security configuration for automatic redirect from HTTP to HTTPS

If you need to automatically redirect HTTP requests to HTTPS, you can set it up using the following steps:

1. Edit the `web.xml` file in the `conf` directory of the Tomcat installation directory (for example, `/usr/Tomcat-9.0.56/conf`), and locate the `</welcome-file-list>` tag.

2. Please add a new line after the closing tag </welcome-file-list> and insert the following content:

```
<login-config>
 <!-- Authorization setting for SSL -->
 <auth-method>CLIENT-CERT</auth-method>
 <realm-name>Client Cert Users-only Area</realm-name>
</login-config>
<security-constraint>
<!-- Authorization setting for SSL -->
<web-resource-collection>
  <web-resource-name>SSL</web-resource-name>
  <url-pattern>/*</url-pattern>
</web-resource-collection>
<user-data-constraint>
  <transport-guarantee>CONFIDENTIAL</transport-guarantee>
</user-data-constraint>
</security-constraint>
```

3. In the Tomcat installation directory, modify the `server.xml` file in the `conf` directory (for example, `/usr/Tomcat-9.0.56/conf`). Change the `redirectPort` parameter to the port of the SSL connector, which is port 443. See the following example:

```
<Connector port="80" protocol="HTTP/1.1"
 connectionTimeout="20000"
 redirectPort="443" />
```

> **Note**
> This modification operation allows non-SSL connectors to be redirected to SSL connectors.

4. In the Tomcat installation directory `/bin` (for example, `/usr/Tomcat-9.0.56/bin`), execute the following command to shut down the Tomcat service.

```
./shutdown.sh
```

5. Execute the following command to ascertain whether there are any issues with the configuration.

```
./configtest.sh
```

- If issues exist, please reconfigure or modify according to the provided suggestions.
- If it does not exist, please proceed to the next step.

6. Execute the following command to start the Tomcat service, then you can access it through `http://cloud.tencent.com` .

```
./startup.sh
```

Installing an SSL Certificate (JKS Format) on a Tomcat Server

Last updated: 2023-10-08 10:56:42

Scenario

This document describes how to install an SSL certificate (JKS format) on a Tomcat server.

Note

- The certificate name `cloud.tencent.com` is used as an example.
- The `tomcat-9.0.56` version is used as an example.
- The current server OS is Windows Server 2016 Chinese. Detailed steps vary slightly with the OS.
- Before installing an SSL certificate, please enable port 443 on the Tomcat server to ensure that HTTPS can be enabled after the certificate is installed. For more information, refer to How Do I Enable Port 443 for a VM? .
- For detailed directions on how to upload SSL certificate files to a server, see Copying Local Files to CVMs .

Preparations

- The Tomcat service has been installed and configured on the current server.
- The data required to install the SSL certificate includes:

Name	Note
Server IP address	The server IP address, which is used to connect the PC to the server.
Username	The username used to log in to the server.
Password	The password used to log in to the server.

Note

- For a CVM instance purchased on the Tencent Cloud official website, log in to the CVM console to get the server IP address, username, and password.
- If you have chosen the "Paste CSR" method when applying for an SSL certificate, or if the brand of the certificate you purchased is Wotrus, then the download of the

JKS certificate file is not provided. You will need to manually convert the format to generate a keystore. The operation method is as follows:

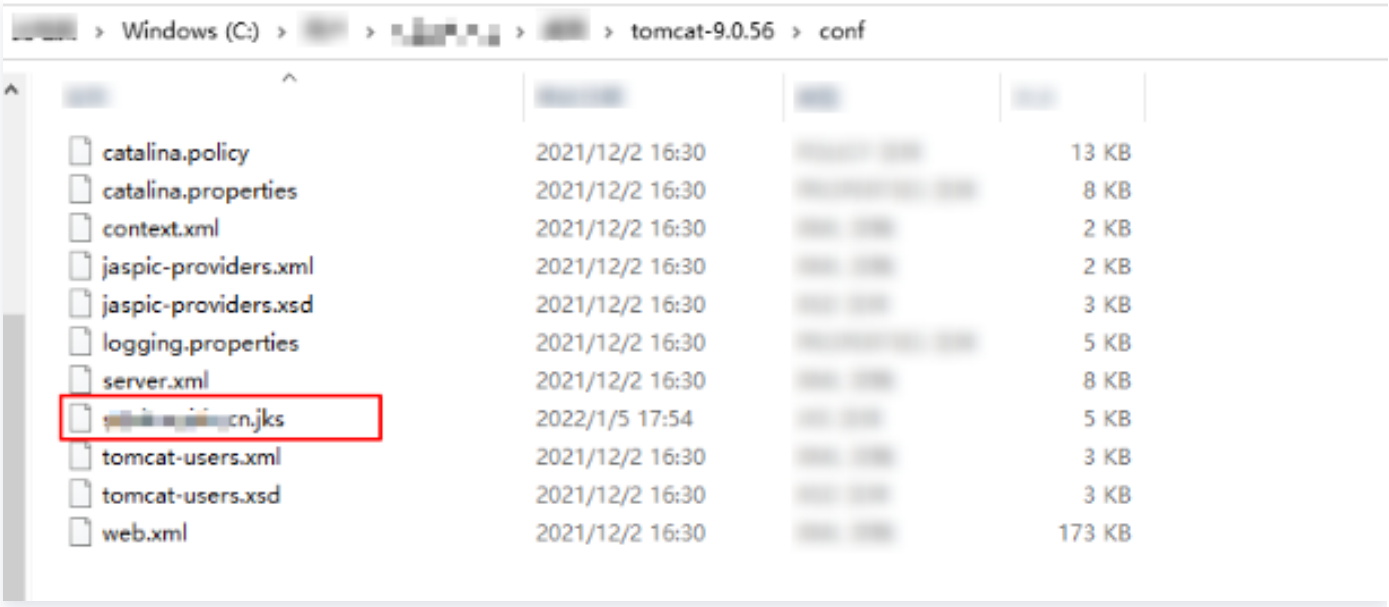
- Access the conversion tool.
- Upload the certificate and private key files from the Nginx folder to the conversion tool, fill in the keystore password, click submit, and convert it into a JKS format certificate.

- If you have chosen the "Paste CSR" method when applying for an SSL certificate, or if the brand of the certificate you purchased is Wotrus, then the download of the JKS certificate file is not provided. You will need to manually convert the format to generate a keystore. The operation method is as follows:
- Access the conversion tool.
- Upload the certificate and private key files from the Nginx folder to the conversion tool, input the keystore password, click **Submit**, and convert them into a JKS format certificate.

Instructions

Certificate Installation

1. Please navigate to the SSL Certificate Service Console and select the certificate you wish to install, then click **Download**.
2. In the "Certificate Download" window that appears, select **JKS** for the server type, click **Download**, and decompress the `cloud.tencent.com` certificate file package to a local directory. After decompression, you can obtain the relevant type of certificate files. This includes the `cloud.tencent.com_jks` folder:
   - **Folder Name:** `cloud.tencent.com_jks`
   - **Folder content:**
     - `cloud.tencent.com.jks` : Key Store
     - `keystorePass.txt` password file (If a private key password has been set, there will be no `keystorePass.txt` password file)
3. Copy the obtained `cloud.tencent.com.jks` keystore file to the `conf` directory in the Tomcat installation directory, as shown below:



4. In the `conf` directory, edit the `server.xml` file and add the following content:

```
<Connector port="443" protocol="HTTP/1.1" SSLEnabled="true"
 maxThreads="150" scheme="https" secure="true"
Path of the certificate
 keystoreFile="Tomcat installation directory/conf/cloud.tencent.com.jks"
Keystore password
 keystorePass="**"
 clientAuth="false"/>
```

For detailed content, refer to the `server.xml` file:

> **Note**
>
> To avoid format issues, you are not advised to copy the content of `server.xml` directly.

```
 <?xml version="1.0" encoding="UTF-8"?>
  <Server port="8005" shutdown="SHUTDOWN">
 <Listener className="org.apache.catalina.startup.VersionLoggerListener" />
 <Listener className="org.apache.catalina.core.AprLifecycleListener"
SSLEngine="on" />
 <Listener
className="org.apache.catalina.core.JreMemoryLeakPreventionListener" />
 <Listener
className="org.apache.catalina.mbeans.GlobalResourcesLifecycleListener" />
```

```
 <Listener
className="org.apache.catalina.core.ThreadLocalLeakPreventionListener" />
 <GlobalNamingResources>
 <Resource name="UserDatabase" auth="Container"
         type="org.apache.catalina.UserDatabase"
         description="User database that can be updated and saved"
         factory="org.apache.catalina.users.MemoryUserDatabaseFactory"
         pathname="conf/tomcat-users.xml" />
  </GlobalNamingResources>
  <Service name="Catalina">
    <Connector port="80" protocol="HTTP/1.1" connectionTimeout="20000"
redirectPort="8443" />
    <Connector port="443" protocol="HTTP/1.1"
          maxThreads="150" SSLEnabled="true" scheme="https" secure="true"
          clientAuth="false"
           keystoreFile="Tomcat installation directory/conf/cloud.tencent.com.jks"
           keystorePass="**" />
 <Connector port="8009" protocol="AJP/1.3" redirectPort="8443" />
<Engine name="Catalina" defaultHost="cloud.tencent.com">
  <Realm className="org.apache.catalina.realm.LockOutRealm">
    <Realm className="org.apache.catalina.realm.UserDatabaseRealm"
          resourceName="UserDatabase"/>
  </Realm>
 <Host name="cloud.tencent.com"  appBase="webapps"
    unpackWARs="true" autoDeploy="true" >
    <Context path="" docBase ="Knews" />
 <Valve className="org.apache.catalina.valves.AccessLogValve" directory="logs"
       prefix="localhost_access_log" suffix=".txt"
       pattern="%h %l %u %t &amp;quot;%r&amp;quot; %s %b" />
  </Host>
 </Engine>
  </Service>
</Server>
```

The main parameters of the configuration file are described as below:

- **keystoreFile**: The location of the keystore file. You can specify an absolute path or a path relative to the <CATALINA_HOME> (Tomcat installation directory) environment variable. If this item is not set, Tomcat will read a file named ".keystore" from the current operating system user's home directory by default.
- **KeystorePass**: This is the keystore password, which specifies the password for the keystore. If you have set a private key password when applying for the certificate, please enter the private key password. If you haven't set a private key password when

applying for the certificate, please enter the password found in the keystorePass.txt file in the Tomcat folder.

- **clientAuth**: If set to true, it indicates that Tomcat requires all SSL clients to present a security certificate for identity verification.

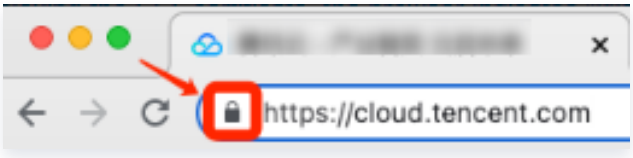
5. Verify if the Tomcat server is running.
   - If already started, you need to execute the following bat scripts in the `bin` directory of the Tomcat installation directory to shut down and restart the Tomcat service.

```
shutdown.bat (Shut down the Tomcat server)
startup.bat (Start the Tomcat server)
```

   - If not already running, you need to execute the following bat script in the `bin` directory of the Tomcat installation directory to start the Tomcat service.

```
startup.bat
```

6. Upon successful startup, you can access it through `https://cloud.tencent.com` .
   - If the security lock icon is displayed in the browser, the certificate has been installed successfully. The details are as shown below:

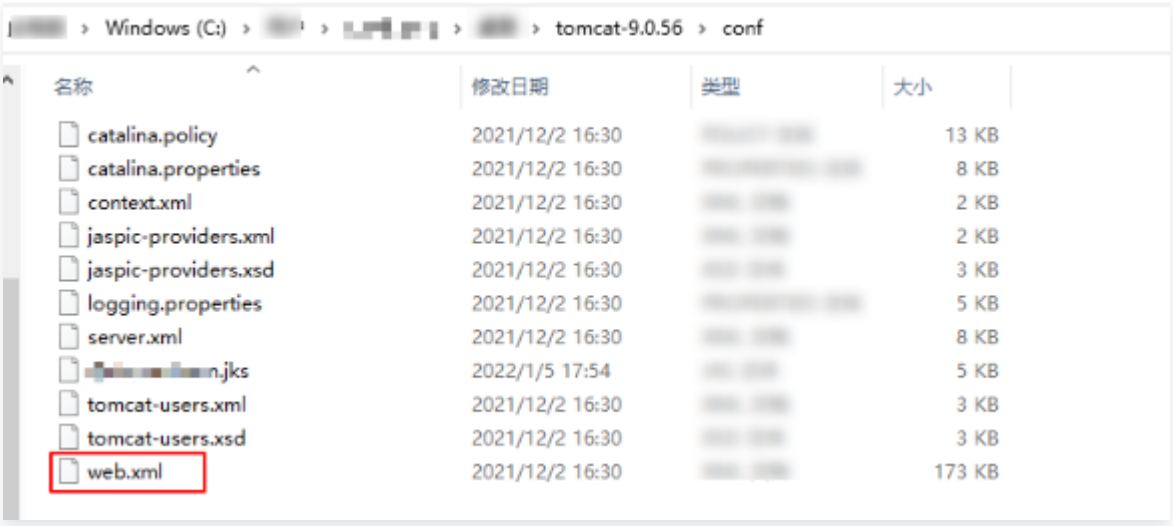


   - In case of a website access exception, troubleshoot the issue by referring to the following FAQs:
     - Unable to access the website via HTTPS
     - After deploying the SSL certificate, the browser indicates "Website connection is not secure"
     - Is your site indicating an insecure connection?
     - What should I do if I am prompted that HTTPS is not secure after reapplying for deployment upon expiration of the SSL certificate?
     - After deploying the SSL certificate on the server, a 404 error occurs when accessing resources

(Optional) Security configuration for automatic redirect from HTTP to HTTPS

If you need to automatically redirect HTTP requests to HTTPS, you can set it up using the following steps:

1. Navigate to the `conf` directory in the Tomcat installation directory, edit the `web.xml` file, and locate the </welcome-file-list> tag. As shown in the image below:



2. Please add a new line after the closing tag </welcome-file-list> and insert the following content:

```
<login-config>
 <!-- Authorization setting for SSL -->
 <auth-method>CLIENT-CERT</auth-method>
 <realm-name>Client Cert Users-only Area</realm-name>
 </login-config>
 <security-constraint>
 <!-- Authorization setting for SSL -->
 <web-resource-collection>
 <web-resource-name>SSL</web-resource-name>
 <url-pattern>/*</url-pattern>
 </web-resource-collection>
 <user-data-constraint>
 <transport-guarantee>CONFIDENTIAL</transport-guarantee>
 </user-data-constraint>
 </security-constraint>
```

3. In the Tomcat installation directory, modify the `server.xml` file and change the redirectPort parameter to the port of the SSL connector, which is port 443. As shown below:

```
<Connector port="80" protocol="HTTP/1.1"
 connectionTimeout="20000"
 redirectPort="443" />
```

> **Note**
> This modification operation allows non-SSL connectors to be redirected to SSL connectors.

4. Execute the following bat script in the Tomcat installation directory `/bin` to stop the Tomcat server.

```
shutdown.bat
```

5. Execute the following command to ascertain whether there are any issues with the configuration.

```
configtest.bat
```

   - If issues exist, please reconfigure or modify according to the provided suggestions.
   - If it does not exist, please proceed to the next step.

6. Execute the following bat script to start the Tomcat server, then you can access it through `http://cloud.tencent.com` .

```
startup.bat
```

Installing an SSL Certificate (PFX Format) on a Tomcat Server

Last updated: 2023-10-08 11:00:10

Scenario

This document describes how to install an SSL certificate (PFX format) on a Tomcat server.

Note

- The certificate name `cloud.tencent.com` is used as an example.
- The `tomcat9.0.40` version is used as an example.
- The current server OS is CentOS 7. Detailed steps vary slightly with the OS version.
- If you need to install an SSL certificate in JKS format on a Tomcat server, please refer to: Installing and Deploying an SSL Certificate on a Tomcat Server (JKS Format).
- Before installing an SSL certificate, please enable port 443 on the Tomcat server to ensure that HTTPS can be enabled after the certificate is installed. For more information, refer to: How Do I Enable Port 443 for a VM?
- For detailed directions on how to upload SSL certificate files to a server, see Copying Local Files to CVMs.

Preparations

- A remote file copy tool such as WinSCP has been prepared. It is recommended to download the latest version from the official website.
  If you need to deploy to Tencent Cloud Server, it is suggested to use the file upload function of the cloud server. For more details, please refer to Uploading Files to Cloud Server.
- Install the remote login tool such as PuTTY or Xshell.
- The Tomcat service has been installed and configured on the current server.
- The data required to install the SSL certificate includes:

Name	Note
Server IP address	The server IP address, which is used to connect the PC to the server.
Username	The username used to log in to the server.

Password	The password used to log in to the server.

Note

- For a CVM instance purchased on the Tencent Cloud official website, log in to the CVM console to get the server IP address, username, and password.
- The current Tomcat server is installed in the `/usr` directory. For instance, if the Tomcat folder name is `tomcat9.0.40`, then `/usr/*/conf` actually refers to `/usr/tomcat9.0.40/conf`.

Instructions

Certificate Installation

1. Please navigate to the SSL Certificate Service Console and select the certificate you wish to install, then click **Download**.
2. In the "Certificate Download" window that appears, select **Tomcat** as the server type, click **Download** and decompress the `cloud.tencent.com` certificate file package to a local directory. After decompression, you can obtain the relevant type of certificate files. This includes the `cloud.tencent.com_tomcat` folder:
   - **Folder Name:** `cloud.tencent.com_tomcat`
   - **Folder content:**
     - `cloud.tencent.com.pfx`: Certificate file
     - `keystorePass.txt` password file (If a private key password has been set, there will be no `keystorePass.txt` password file)
3. Utilize WinSCP (a tool for transferring files between a local and a remote computer) to access the Tomcat server.

   Note

   - For detailed directions, see Uploading files via WinSCP to a Linux CVM from Windows.
   - We recommend using the file upload feature of the Cloud Virtual Machine (CVM) for deployment to Tencent Cloud CVM. For more details, please refer to Uploading Files to CVM.
4. Copy the obtained certificate file `cloud.tencent.com.pfx` from the local directory to the `/usr/*/conf` directory.
5. Remotely log in to the Tomcat server, for instance, using the "PuTTY" tool.

6. Edit the `server.xml` file in the `/usr/*/conf` directory. Choose one of the following methods based on your actual needs:

> **Note**
> When using Method 1 for configuration, Tomcat will automatically select the SSL implementation for you. If you are unable to complete the subsequent configuration using Method 1, it may be because your environment does not support this implementation. You can manually select SSL for configuration using Method 2, based on your environment attributes.

Method 1: Automatic SSL selection

Modify the `server.xml` file by adding the Connector attributes as follows:

```
<Connector port="443"
protocol="HTTP/1.1"
    SSLEnabled="true"
    scheme="https"
    secure="true"
    keystoreFile="/usr/*/conf/cloud.tencent.com.pfx" <!-- Path where the
certificate is saved -->
    keystoreType="PKCS12"
    keystorePass="Certificate Password"  <! -- Please replace with the content in
the keystorePass.txt password file.-->
    clientAuth="false"
    SSLProtocol="TLSv1.1+TLSv1.2+TLSv1.3"

ciphers="TLS_RSA_WITH_AES_128_CBC_SHA,TLS_RSA_WITH_AES_256_CBC_SHA,T
LS_ECDHE_RSA_WITH_AES_128_CBC_SHA,TLS_ECDHE_RSA_WITH_AES_128_CBC_
SHA256,TLS_RSA_WITH_AES_128_CBC_SHA256,TLS_RSA_WITH_AES_256_CBC_SH
A256"/>
```

Method 2: Manually select SSL

Modify the `server.xml` file by adding the Connector attributes as follows:

```
<Connector
    protocol="org.apache.coyote.http11.Http11NioProtocol"
    port="443" maxThreads="200"
```

```
    scheme="https" secure="true" SSLEnabled="true"
    keystoreFile="/usr/*/conf/cloud.tencent.com.pfx" keystorePass="Certificate
Password" <! --Replace pfx with the path where the certificate is saved, and
replace the certificate password with the content in the keystorePass.txt
password file.-->
    clientAuth="false" sslProtocol="TLS"/>
```

The main parameters of the configuration file are described as below:

- **keystoreFile**: The location of the certificate file. You can specify an absolute path or a path relative to the <CATALINA_HOME> (Tomcat installation directory) environment variable. If this item is not set, Tomcat will read a file named ".keystore" from the current operating system user's home directory by default.
- **keystorePass**: This is the password for the password file, which specifies the keystore password. If a private key password was set when applying for the certificate, please enter the private key password. If no private key password was set when applying for the certificate, please enter the password found in the keystorePass.txt file within the `cloud.tencent.com_tomcat` folder.
- **clientAuth**: If set to true, it indicates that Tomcat requires all SSL clients to present a security certificate for identity verification.

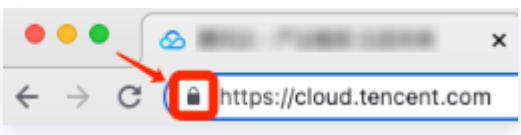
7. Verify if the Tomcat server is running.
   - If already running, you need to execute the following commands in the `/usr/*/bin` directory to shut down and restart the Tomcat server.

     ```
     ./shutdown.sh (Shut down the Tomcat server)
     ./startup.sh (Start the Tomcat server)
     ```

   - If not already running, you need to execute the following command in the `/usr/*/bin` directory to start the Tomcat server.

     ```
     ./startup.sh
     ```

8. Upon successful startup, you can access it through `https://cloud.tencent.com` .
   - If the security lock icon is displayed in the browser, the certificate has been installed successfully. The details are as shown below:



- In case of a website access exception, troubleshoot the issue by referring to the following FAQs:
  - Unable to access the website via HTTPS
  - After deploying the SSL certificate, the browser indicates "Website connection is not secure"
  - Is your site indicating an insecure connection?
  - What should I do if I am prompted that HTTPS is not secure after reapplying for deployment upon expiration of the SSL certificate?
  - After deploying the SSL certificate on the server, a 404 error occurs when accessing resources

(Optional) Security configuration for automatic redirect from HTTP to HTTPS

If you need to automatically redirect HTTP requests to HTTPS, you can set it up using the following steps:

1. In the `/usr/*/conf` directory, locate the </welcome-file-list> tag in the `web.xml` file.
2. Please add a new line after the </welcome-file-list> end tag, and append the following content.

```
<login-config>
 <!-- Authorization setting for SSL -->
 <auth-method>CLIENT-CERT</auth-method>
 <realm-name>Client Cert Users-only Area</realm-name>
</login-config>
<security-constraint>
 <!-- Authorization setting for SSL -->
 <web-resource-collection >
    <web-resource-name >SSL</web-resource-name>
    <url-pattern>/*</url-pattern>
 </web-resource-collection>
 <user-data-constraint>
    <transport-guarantee>CONFIDENTIAL</transport-guarantee>
 </user-data-constraint>
</security-constraint>
```

3. In the `/usr/*/conf` directory, modify the redirectPort parameter in the `server.xml` file to the port of the SSL connector, which is port 443. As shown below:

```
<Connector port="80" protocol="HTTP/1.1"
 connectionTimeout="20000"
```

```
redirectPort="443" />
```

Note
This modification operation allows non-SSL connectors to be redirected to SSL connectors.

4. In the `/usr/*/bin` directory, execute the following command to shut down the Tomcat server.

```
./shutdown.sh
```

5. Execute the following command to ascertain whether there are any issues with the configuration.

```
./configtest.sh
```

   - If issues exist, please reconfigure or modify according to the provided suggestions.
   - If it does not exist, please proceed to the next step.

6. Execute the following command to start the Tomcat server, then you can access it through `http://cloud.tencent.com` .

```
./startup.sh
```

Installing an SSL Certificate on a GlassFish Server

Last updated: 2023-10-08 11:10:15

Scenario

This document describes how to install an SSL certificate on a GlassFish server.

Note

- The certificate name `cloud.tencent.com` is used as an example.
- The `glassfish-4.0` version is used as an example.
- The current server OS is CentOS 7. Detailed steps vary slightly with the OS version.
- Before installing the SSL certificate, please ensure that port "443" is open on your GlassFish server to avoid any issues enabling HTTPS after the certificate is installed. For more information, refer to How to open port 443 on a server?
- For detailed directions on how to upload SSL certificate files to a server, see Copying Local Files to CVMs.

Preparations

- A remote file copy tool such as WinSCP has been prepared. It is recommended to download the latest version from the official website.
  If you need to deploy to Tencent Cloud Server, it is suggested to use the file upload function of the cloud server. For more details, please refer to Uploading Files to Cloud Server.
- Install the remote login tool such as PuTTY or Xshell.
- The GlassFish service has been installed and configured on the current server.
- The data required to install the SSL certificate includes:

Name	Note
Server IP address	The server IP address, which is used to connect the PC to the server.
Username	The username used to log in to the server.
Password	The password used to log in to the server.

Note

- For a CVM instance purchased on the Tencent Cloud official website, log in to the CVM console to get the server IP address, username, and password.
- If you have selected the "Paste CSR" method when applying for the SSL certificate, or if the certificate brand you purchased is Wotrus, then the certificate file formats (.pfx and .jks) supported by Tomcat are not provided for download. You will need to manually convert the format to generate a keystore. The procedure is as follows:
  - Access the conversion tool.
  - Upload the certificate and private key files from the Nginx folder to the conversion tool, input the keystore password, click **Submit**, and convert them into a JKS format certificate.
- The GlassFish service is installed in the `/usr/share` directory.

Instructions

1. Please navigate to the SSL Certificate Service Console and select the certificate you wish to install, then click **Download**.
2. In the "Certificate Download" window that appears, select **Apache** and **JKS** as the server types, click **Download** and decompress the `cloud.tencent.com` certificate file package to a local directory. After decompression, you can obtain the relevant type of certificate files. This includes the `cloud.tencent.com_apache` folder and the `cloud.tencent.com_jks` folder:
   - **Folder Name:** `cloud.tencent.com_apache`
     - `cloud.tencent.com.crt` Certificate file
     - `cloud.tencent.com.key` Private key file
   - **CSR file content:** `cloud.tencent.com.csr` file

     Note
     The CSR file, either uploaded by you or generated online by the system during the certificate application, is provided to the CA. This file can be disregarded during installation.
3. Remotely log in to the GlassFish server, for instance, using the "PuTTY" tool.
4. Navigate to the `/usr/share/glassfish4/glassfish/bin` directory and run the `./asadmin` command. To change the management password for the domain, execute the `change-master-password --savemasterpassword=true domain1` command, as shown in the following figure:

> **Note**
> - The default installation path for domain1 is `/usr/share/glassfish4/glassfish/domains`. Please fill in the domain name according to the actual situation.
> - The default password is 'changeit'. After pressing enter, please input the new password. The new password should be the **private key password** set when applying for the certificate.
> - If you haven't set a private key password when applying for the certificate, enter the password in the `keystorePass.txt` file in the `cloud.tencent.com_jks` folder.

5. Execute the command `mkdir temp` in the `/usr/share` directory to create a temp folder.
6. Utilize "WinSCP" (a tool for copying files between local and remote computers) to log into the GlassFish server, and copy the `cloud.tencent.com.crt` certificate file and `cloud.tencent.com.key` private key file from the local directory to the temp folder.

> **Note**
> - For detailed directions, see Uploading files via WinSCP to a Linux CVM from Windows.
> - We recommend using the file upload feature of the Cloud Virtual Machine (CVM) for deployment to Tencent Cloud CVM. For more details, please refer to Uploading Files to CVM.

7. Execute the following command in the temp directory to generate a PKCS12 file. You will be prompted to enter a password. Please enter the newly set password, which is the private key password, as shown below:

```
openssl pkcs12 -export -in cloud.tencent.com.crt -inkey cloud.tencent.com.key -out mycert.p12 -name s1as
```

8. Execute the command `ls -l` in the temp directory to confirm whether the PKCS12 file contains the certificate you applied for.
9. In the `temp` folder, run the following command to generate the `keystore.jks` file:

```
keytool -importkeystore -destkeystore keystore.jks -srckeystore mycert.p12 -srcstoretype PKCS12 -alias s1as
```

10. In the `temp` directory, execute the following command to generate the `cacert.jks` file. If prompted for a password, enter the newly set password, which is the private key

password. As shown below:

```
keytool -importcert -trustcacerts -destkeystore cacerts.jks -file
cloud.tencent.com.crt -alias s1as
```

If the system asks whether to trust the certificate, enter **yes** as shown in the following figure.

```
Trust this certificate? [no]:  yes
Certificate was added to keystore
[root@VM_4_2_centos Apache]#
```

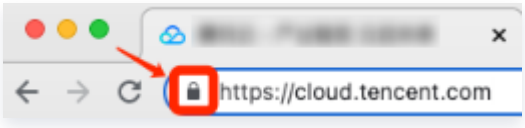
11. Replace the `keystore.jks` and `cacert.jks` files in the `domain1/config` directory with the files generated in steps 9 and 10.
12. In the `/usr/share/glassfish4/glassfish/domains/domain1/config` directory, modify the port numbers in the `domain.xml` file as shown below:

```
<network-listeners>
    <network-listener port="80" protocol="http-listener-1" transport="tcp"
name="http-listener-1" thread-pool="http-thread-pool"></network-listener>
    <network-listener port="443" protocol="http-listener-2" transport="tcp"
name="http-listener-2" thread-pool="http-thread-pool"></network-listener>
    <network-listener port="4848" protocol="admin-listener" transport="tcp"
name="admin-listener" thread-pool="admin-thread-pool"></network-listener>
  </network-listeners>
```

13. Start the GlassFish server and then you can access it through `https://cloud.tencent.com` .

```
[root@VM_4_2_centos ~]# cd /usr/share/glassfish4/glassfish/bin/
[root@VM_4_2_centos bin]# ./asadmin
Use "exit" to exit and "help" for online help.
asadmin> start-domain domain1
```

- If the security lock icon is displayed in the browser, the certificate has been installed successfully. The details are as shown below:



- In case of a website access exception, troubleshoot the issue by referring to the following FAQs:
  - Unable to access the website via HTTPS
  - After deploying the SSL certificate, the browser indicates "Website connection is not secure"

- Is your site indicating an insecure connection?
- What should I do if I am prompted that HTTPS is not secure after reapplying for deployment upon expiration of the SSL certificate?
- After deploying the SSL certificate on the server, a 404 error occurs when accessing resources

Note

If anything goes wrong during this process, please contact us.

Installing an SSL Certificate on a JBoss Server

Last updated: 2023-10-08 11:17:00

Scenario

This document describes how to install an SSL certificate on a JBoss server.

Note

- The certificate name `cloud.tencent.com` is used as an example.
- The `jboss-7.1.1` version is used as an example.
- The current server OS is CentOS 7. Detailed steps vary slightly with the OS version.
- Before installing an SSL certificate, please enable port 443 on the JBoss server to ensure that HTTPS can be enabled after the certificate is installed. For more information, refer to How Do I Enable Port 443 for a VM?
- For detailed directions on how to upload SSL certificate files to a server, see Copying Local Files to CVMs.

Preparations

- A remote file copy tool such as WinSCP has been prepared. It is recommended to download the latest version from the official website.
  If you need to deploy to Tencent Cloud Server, it is suggested to use the file upload function of the cloud server. For more details, please refer to Uploading Files to Cloud Server.
- Install the remote login tool such as PuTTY or Xshell.
- The JBoss service has been installed and configured on the current server.
- The data required to install the SSL certificate includes:

Name	Note
Server IP address	The server IP address, which is used to connect the PC to the server.
Username	The username used to log in to the server.
Password	The password used to log in to the server.

> **Note**
> - For a CVM instance purchased on the Tencent Cloud official website, log in to the CVM console to get the server IP address, username, and password.
> - If you have chosen the "Paste CSR" method when applying for an SSL certificate, or if the brand of the certificate you purchased is Wotrus, then the download of the JKS certificate file is not provided. You will need to manually convert the format to generate a keystore. The operation method is as follows:
>   - Access the conversion tool.
>   - Upload the certificate and private key files from the Nginx folder to the conversion tool, fill in the keystore password, click submit, and convert it into a JKS format certificate.
>   - The JBoss server is currently installed in the /usr/local directory.

- If you have chosen the "Paste CSR" method when applying for an SSL certificate, or if the brand of the certificate you purchased is Wotrus, then the download of the JKS certificate file is not provided. You will need to manually convert the format to generate a keystore. The operation method is as follows:
- Access the conversion tool.
- Upload the certificate and private key files from the Nginx folder to the conversion tool, input the keystore password, click **Submit**, and convert them into a JKS format certificate.
- The JBoss server is currently installed in the `/usr/local` directory.

Instructions

1. Please navigate to the SSL Certificate Service Console and select the certificate you wish to install, then click **Download**.
2. In the "Certificate Download" window that appears, select **JKS** for the server type, click **Download**, and decompress the `cloud.tencent.com` certificate file package to a local directory. After decompression, you can obtain the relevant type of certificate files. This includes the `cloud.tencent.com_jks` folder:
   - **Folder Name:** `cloud.tencent.com_jks`
   - **Folder content:**
     - `cloud.tencent.com.jks` Key Store
     - `keystorePass.txt` password file (If a private key password has been set, there will be no `keystorePass.txt` password file)
3. Remotely log in to the JBoss server. For instance, log in using the "PuTTY" tool.
4. Proceed to the certificate deployment step. In the

`/usr/local/jboss-7.1.1/standalone/configuration` directory, execute the `mkdir cert` command to create a cert folder.

5. Utilize WinSCP (a tool for copying files between a local and a remote computer) to log into the JBoss server. Then, transfer the obtained keystore file `cloud.tencent.com.jks` from the local directory to the cert folder.

> **Note**
> - For detailed directions, see Uploading files via WinSCP to a Linux CVM from Windows.
> - We recommend using the file upload feature of the Cloud Virtual Machine (CVM) for deployment to Tencent Cloud CVM. For more details, please refer to Uploading Files to CVM.

6. In the `/usr/local/jboss-7.1.1/standalone/configuration` directory, modify the port configuration in the `standalone.xml` file as shown below:
   - Part 1:

```
<interfaces>
    <interface name="management">
        <inet-address value="${jboss.bind.address.management:127.0.0.1}"/>
    </interface>
            <!--Enable Remote Access-->
    <interface name="public">
        <inet-address value="${jboss.bind.address:0.0.0.0}"/>
    </interface>
    <interface name="unsecure">
        <inet-address value="${jboss.bind.address.unsecure:127.0.0.1}"/>
    </interface>
 </interfaces>
 <socket-binding-group name="standard-sockets" default-interface="public"
port-offset="${jboss.socket.binding.port-offset:0}">
    <socket-binding name="management-native" interface="management"
port="${jboss.management.native.port:9999}"/>
    <socket-binding name="management-http" interface="management"
port="${jboss.management.http.port:9990}"/>
    <socket-binding name="management-https" interface="management"
port="${jboss.management.https.port:9443}"/>
    <socket-binding name="ajp" port="8009"/>
            <!--Modify HTTP port-->
    <socket-binding name="http" port="80"/>
            <!--Modify HTTPS port-->
    <socket-binding name="https" port="443"/>
```

```
    <socket-binding name="osgi-http" interface="management"
port="8090"/>
    <socket-binding name="remoting" port="4447"/>
    <socket-binding name="txn-recovery-environment" port="4712"/>
    <socket-binding name="txn-status-manager" port="4713"/>
    <outbound-socket-binding name="mail-smtp">
       <remote-destination host="localhost" port="25"/>
    </outbound-socket-binding>
 </socket-binding-group>
```

Changes required are as follows:

- **Enable Remote Access:** Adjust `${jboss.bind.address:127.0.0.1}` to `${jboss.bind.address:0.0.0.0}`.
- **Modify HTTP Port:** Adjust the port from 8080 to 80.
- **Modify HTTPS Port:** Adjust port 8443 to 443.

- Part Two: Adding Certificate-Related Configuration.

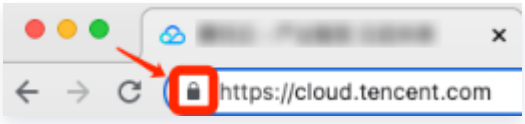
```
<subsystem xmlns="urn:jboss:domain:web:1.1" default-virtual-
server="default-host" native="false">
        <connector name="http" protocol="HTTP/1.1" scheme="http" socket-
binding="http"/>
        <connector name="https" protocol="HTTP/1.1" scheme="https" socket-
binding="https" secure="true">
            <ssl name="https" password="**" certificate-key-
file="../standalone/configuration/cert/cloud.tencent.com.jks" cipher-
suite="TLS_RSA_WITH_AES_256_CBC_SHA,TLS_DHE_RSA_WITH_AES_256_CBC_
SHA,TLS_DHE_DSS_WITH_AES_128_CBC_SHA,SSL_RSA_WITH_3DES_EDE_CBC_S
HA,SSL_DHE_RSA_WITH_3DES_EDE_CBC_SHA,SSL_DHE_DSS_WITH_3DES_EDE_
CBC_SHA" protocol="TLSv1,TLSv1.1,TLSv1.2"/>
        </connector>
        <virtual-server name="default-host" enable-welcome-root="true">
            <alias name="localhost"/>
            <alias name="example.com"/>
        </virtual-server>
    </subsystem>
```

7. Navigate to the `/usr/local/jboss-7.1.1/bin` directory and execute the startup command `./standalone.sh` to ensure a successful launch, as shown below:

```
[root@VM_4_2_centos ~]# cd /usr/local/jboss-7.1.1/bin
[root@VM_4_2_centos bin]# ./standalone.sh
```

8. Once the certificate is deployed, you can access it at `https://cloud.tencent.com`.

- If the security lock icon is displayed in the browser, the certificate has been installed successfully. The details are as shown below:



- In case of a website access exception, troubleshoot the issue by referring to the following FAQs:
  - Unable to access the website via HTTPS
  - After deploying the SSL certificate, the browser indicates "Website connection is not secure"
  - Is your site indicating an insecure connection?
  - What should I do if I am prompted that HTTPS is not secure after reapplying for deployment upon expiration of the SSL certificate?
  - After deploying the SSL certificate on the server, a 404 error occurs when accessing resources

Note

If anything goes wrong during this process, please contact us.

Installing an SSL Certificate on a Jetty Server

Last updated: 2023-10-08 11:20:01

Scenario

This document describes how to install an SSL certificate on a Jetty server.

Note

- The certificate name `cloud.tencent.com` is used as an example.
- The `jetty-distribution-9.4.28.v20200408` version is used as an example.
- The current server OS is CentOS 7. Detailed steps vary slightly with the OS version.
- Before installing an SSL certificate, please enable port 443 on the Jetty server to ensure that HTTPS can be enabled after the certificate is installed. For more details, refer to How Do I Enable Port 443 for a VM?
- For detailed directions on how to upload SSL certificate files to a server, see Copying Local Files to CVMs.

Preparations

- A remote file copy tool such as WinSCP has been prepared. It is recommended to download the latest version from the official website.
  If you need to deploy to Tencent Cloud Server, it is suggested to use the file upload function of the cloud server. For more details, please refer to Uploading Files to Cloud Server.
- Install the remote login tool such as PuTTY or Xshell.
- The Jetty service has been installed and configured on the current server.
- The data required to install the SSL certificate includes:

Name	Note
Server IP address	The server IP address, which is used to connect the PC to the server.
Username	The username used to log in to the server.
Password	The password used to log in to the server.

Note

- For a CVM instance purchased on the Tencent Cloud official website, log in to the CVM console to get the server IP address, username, and password.
- If you have chosen the "Paste CSR" method when applying for an SSL certificate, or if the brand of the certificate you purchased is Wotrus, then the download of the JKS certificate file is not provided. You will need to manually convert the format to generate a keystore. The operation method is as follows:
  - Access the conversion tool.
  - Upload the certificate and private key files from the Nginx folder to the conversion tool, input the keystore password, click **Submit**, and convert them into a JKS format certificate.
- The Jetty server is currently installed in the `/usr/local/jetty` directory.

Instructions

1. Please navigate to the SSL Certificate Service Console and select the certificate you wish to install, then click **Download**.
2. In the "Certificate Download" window that appears, select **JKS** for the server type, click **Download**, and decompress the `cloud.tencent.com` certificate file package to a local directory. After decompression, you can obtain the relevant type of certificate files. This includes the `cloud.tencent.com_jks` folder:
   - **Folder Name:** `cloud.tencent.com_jks`
   - **Folder content:**
     - `cloud.tencent.com.jks` Key Store
     - `keystorePass.txt` password file (If a private key password has been set, there will be no `keystorePass.txt` password file)
3. Remotely log in to the Jetty server. For instance, log in using the "PuTTY" tool.
4. Proceed to the certificate deployment steps. In the `/usr/local/jetty/jetty-distribution-9.4.28.v20200408/etc` directory, execute the command `mkdir cert` to create a cert folder.
5. Utilize WinSCP (a tool for transferring files between a local and a remote computer) to access the Jetty server. Subsequently, transfer the obtained `cloud.tencent.com.jks` keystore file from the local directory to the cert folder.

Note

- For detailed directions, see Uploading files via WinSCP to a Linux CVM from Windows.
- We recommend using the file upload feature of the Cloud Virtual Machine (CVM) for deployment to Tencent Cloud CVM. For more details, please refer to Uploading Files to CVM.

6. Modify the `jetty-ssl-context.xml` file in the `/usr/local/jetty/jetty-distribution-9.4.28.v20200408/etc` directory as follows:

Note

- **KeyStorePath:** Please enter the path where the certificate is stored for the default value.
- **KeyStorePassword:** The default value is 'default'. Enter the keystore password, which specifies the password for the keystore. If you set a private key password when applying for the certificate, enter the private key password. If you didn't set a private key password when applying for the certificate, enter the password from the keystorePass.txt file in the `cloud.tencent.com_jks` folder.
- **KeyManagerPassword:** Please enter the password from the keystorePass.txt file in the `cloud.tencent.com_jks` folder.
- **TrustStorePath:** Please enter the path where the certificate is stored for the default value.

```
<?xml version="1.0"?><!DOCTYPE Configure PUBLIC "-//Jetty//Configure//EN"
"http://www.eclipse.org/jetty/configure_9_3.dtd">
<!--
=============================================================
=========== --><!-- SSL ContextFactory configuration                    -->
<!--
=============================================================
=========== -->
<!--
 To configure Includes / Excludes for Cipher Suites or Protocols see tweak-ssl.xml
example at
 https://www.eclipse.org/jetty/documentation/current/configuring-
ssl.html#configuring-sslcontextfactory-cipherSuites
-->
<Configure id="sslContextFactory"
class="org.eclipse.jetty.util.ssl.SslContextFactory$Server">
 <Set name="Provider"><Property name="jetty.sslContext.provider"/></Set>
```

```
  <Set name="KeyStorePath"><Property name="jetty.base" default="."
/>/<Property name="jetty.sslContext.keyStorePath" deprecated="jetty.keystore"
default="etc/cert/cloud.tencent.com.jks"/></Set>
  <Set name="KeyStorePassword"><Property
name="jetty.sslContext.keyStorePassword" deprecated="jetty.keystore.password"
default="4d5jtdq238j1l"/></Set>
  <Set name="KeyStoreType"><Property name="jetty.sslContext.keyStoreType"
default="JKS"/></Set>
  <Set name="KeyStoreProvider"><Property
name="jetty.sslContext.keyStoreProvider"/></Set>
  <Set name="KeyManagerPassword"><Property
name="jetty.sslContext.keyManagerPassword"
deprecated="jetty.keymanager.password" default="4d5jtdq238j1l"/></Set>
  <Set name="TrustStorePath"><Property name="jetty.base" default="."
/>/<Property name="jetty.sslContext.trustStorePath" deprecated="jetty.truststore"
default="etc/cert/cloud.tencent.com.jks"/></Set>
  <Set name="TrustStorePassword"><Property
name="jetty.sslContext.trustStorePassword"
deprecated="jetty.truststore.password"/></Set>
  <Set name="TrustStoreType"><Property
name="jetty.sslContext.trustStoreType"/></Set>
  <Set name="TrustStoreProvider"><Property
name="jetty.sslContext.trustStoreProvider"/></Set>
  <Set name="EndpointIdentificationAlgorithm"><Property
name="jetty.sslContext.endpointIdentificationAlgorithm"/></Set>
  <Set name="NeedClientAuth"><Property
name="jetty.sslContext.needClientAuth" deprecated="jetty.ssl.needClientAuth"
default="false"/></Set>
  <Set name="WantClientAuth"><Property
name="jetty.sslContext.wantClientAuth" deprecated="jetty.ssl.wantClientAuth"
default="false"/></Set>
  <Set name="useCipherSuitesOrder"><Property
name="jetty.sslContext.useCipherSuitesOrder" default="true"/></Set>
  <Set name="sslSessionCacheSize"><Property
name="jetty.sslContext.sslSessionCacheSize" default="-1"/></Set>
  <Set name="sslSessionTimeout"><Property
name="jetty.sslContext.sslSessionTimeout" default="-1"/></Set>
  <Set name="RenegotiationAllowed"><Property
name="jetty.sslContext.renegotiationAllowed" default="true"/></Set>
  <Set name="RenegotiationLimit"><Property
name="jetty.sslContext.renegotiationLimit" default="5"/></Set>
  <Set name="SniRequired"><Property name="jetty.sslContext.sniRequired"
default="false"/></Set>
  <!-- Example of how to configure a PKIX Certificate Path revocation Checker
```

```
  <Call id="pkixPreferCrls"
class="java.security.cert.PKIXRevocationChecker$Option" name="valueOf">
<Arg>PREFER_CRLS</Arg></Call>
  <Call id="pkixSoftFail" class="java.security.cert.PKIXRevocationChecker$Option"
name="valueOf"><Arg>SOFT_FAIL</Arg></Call>
  <Call id="pkixNoFallback"
class="java.security.cert.PKIXRevocationChecker$Option" name="valueOf">
<Arg>NO_FALLBACK</Arg></Call>
  <Call class="java.security.cert.CertPathBuilder" name="getInstance">
 <Arg>PKIX</Arg>
 <Call id="pkixRevocationChecker" name="getRevocationChecker">
  <Call name="setOptions">
   <Arg>
    <Call class="java.util.EnumSet" name="of">
     <Arg><Ref refid="pkixPreferCrls"/></Arg>
     <Arg><Ref refid="pkixSoftFail"/></Arg>
     <Arg><Ref refid="pkixNoFallback"/></Arg>
    </Call>
   </Arg>
  </Call>
 </Call>
  </Call>
  <Set name="PkixCertPathChecker"><Ref refid="pkixRevocationChecker"/>
</Set>
  -->
</Configure>
```

7. In the `/usr/local/jetty/jetty-distribution-9.4.28.v20200408/etc` directory, modify the port to 443 in the `jetty-ssl.xml` file as shown below:

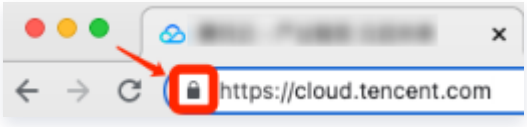
```
 <Call  name="addConnector">
 <Arg>
  <New id="sslConnector" class="org.eclipse.jetty.server.ServerConnector">
   <Arg name="server"><Ref refid="Server" /></Arg>
   <Arg name="acceptors" type="int"><Property name="jetty.ssl.acceptors"
deprecated="ssl.acceptors" default="-1"/></Arg>
   <Arg name="selectors" type="int"><Property name="jetty.ssl.selectors"
deprecated="ssl.selectors" default="-1"/></Arg>
   <Arg name="factories">
    <Array type="org.eclipse.jetty.server.ConnectionFactory">
     <!-- uncomment to support proxy protocol
     <Item>
      <New class="org.eclipse.jetty.server.ProxyConnectionFactory"/>
     </Item>-->
```

```
      </Array>
    </Arg>
    <Set name="host"><Property name="jetty.ssl.host" deprecated="jetty.host" />
</Set>
    <Set name="port"><Property name="jetty.ssl.port" deprecated="ssl.port"
default="443" /></Set>
    <Set name="idleTimeout"><Property name="jetty.ssl.idleTimeout"
deprecated="ssl.timeout" default="30000"/></Set>
    <Set name="acceptorPriorityDelta"><Property
name="jetty.ssl.acceptorPriorityDelta" deprecated="ssl.acceptorPriorityDelta"
default="0"/></Set>
    <Set name="acceptQueueSize"><Property name="jetty.ssl.acceptQueueSize"
deprecated="ssl.acceptQueueSize" default="0"/></Set>
    <Get name="SelectorManager">
      <Set name="connectTimeout"><Property name="jetty.ssl.connectTimeout"
default="15000"/></Set>
    </Get>
  </New>
 </Arg>
  </Call>
```

8. In the `/usr/local/jetty/jetty-distribution-9.4.28.v20200408` directory, add the following content to the `start.ini` file:

```
etc/jetty-ssl.xml
etc/jetty-ssl-context.xml
etc/jetty-https.xml
```

9. Once the certificate is successfully deployed, execute the start command `java -jar start.jar` in the Jetty root directory. You can then access the server via `https://cloud.tencent.com`.
   - If the security lock icon is displayed in the browser, the certificate has been installed successfully. The details are as shown below:



   - In case of a website access exception, troubleshoot the issue by referring to the following FAQs:
     - Unable to access the website via HTTPS
     - After deploying the SSL certificate, the browser indicates "Website connection is not secure"
     - Is your site indicating an insecure connection?

- What should I do if I am prompted that HTTPS is not secure after reapplying for deployment upon expiration of the SSL certificate?
- After deploying the SSL certificate on the server, a 404 error occurs when accessing resources

Supports and Limits

Upon successful deployment of the certificate, if accessing `https://cloud.tencent.com` displays the following:

Solution: You can copy the ROOT file from the `/usr/local/jetty/jetty-distribution-9.4.28.v20200408/demo-base/webapps` directory to the `/usr/local/jetty/jetty-distribution-9.4.28.v20200408/webapps` directory, restart Jetty, and access should be successful.

> **Note**
> If anything goes wrong during this process, please contact us.

Installing SSL Certificate on Spring Boot

Last updated: 2023-10-08 11:23:04

Scenario

This document provides guidance on how to configure and install a PFX format SSL certificate for Spring Boot.

Note:

- The certificate name `cloud.tencent.com` is used as an illustrative example in this document.
- The `3.0.2` version of Spring Boot is used as an example.
- The current server OS is CentOS 7. Detailed steps vary slightly with the OS version.
- If you need to install an SSL certificate in JKS format on a Tomcat server, please refer to: Installing and Deploying an SSL Certificate on a Tomcat Server (JKS Format).
- Before installing an SSL certificate, please enable port 443 on the Tomcat server to ensure that HTTPS can be enabled after the certificate is installed. For more information, refer to: How Do I Enable Port 443 for a VM?

Preparations

- A remote file copy tool such as WinSCP has been prepared (it is recommended to download the latest version from the official website). If you need to deploy to Tencent Cloud CVM, it is suggested to use the file upload feature of the CVM. For more details, please refer to Uploading Files to CVM.
- Install the remote login tool such as PuTTY or Xshell.
- The data required to install the SSL certificate includes:

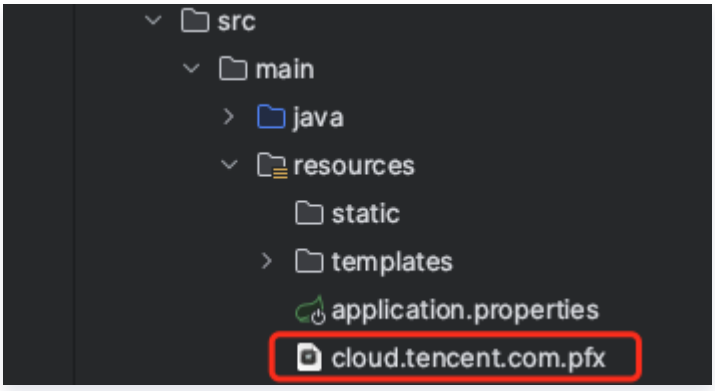
Name	Note
Server IP address	The server IP address, which is used to connect the PC to the server.
Username	The username used to log in to the server.
Password	The password used to log in to the server.

Instructions

1. Please navigate to the SSL Certificate Service Console and select the certificate you wish to install, then click **Download**.
2. In the pop-up **Certificate Download** window, select **Tomcat** as the server type, click to download and unzip the `cloud.tencent.com` certificate file package to a local directory. After decompression, you can obtain the relevant type of certificate files. This includes the `cloud.tencent.com_tomcat` folder:
   - Folder name: `cloud.tencent.com_tomcat`
   - Folder content:
   - `cloud.tencent.com.pfx` Certificate file
   - `keystorePass.txt` password file (If a private key password has been set, there will be no `keystorePass.txt` password file)
3. Utilize WinSCP (a tool for transferring files between a local and a remote computer) to access the Tomcat server.

> **Note:**
> For detailed directions, see Uploading files via WinSCP to a Linux CVM from Windows.
> We recommend using the file upload feature of the Cloud Virtual Machine (CVM) for deployment to Tencent Cloud. For more details, please refer to Uploading Files to CVM.

4. Copy the obtained `cloud.tencent.com.pfx` certificate file from the local directory to the `src/main/resources` directory of the Spring Boot project, as shown in the following figure:



5. Log in to the Spring Boot server remotely. For instance, using the "PuTTY" tool for login.
6. Edit the `application.properties` or `application.yml` file of the Spring Boot project. Choose one of the following methods based on your actual requirements:
   - Method 1

Modify the `application.properties` file by adding the following content:

```
server.address=cloud.tencent.comserver.port = 443server.ssl.key-store = classpath:cloud.tencent.com.pfxserver.ssl.key-store-password = ***server.ssl.keyStoreType = PKCS12
```

- Method 2

  Modify the `application.yml` file by adding the following content:

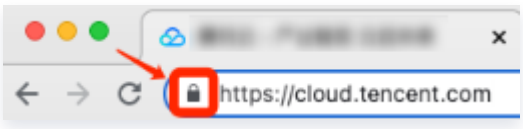
```
server:  address: cloud.tencent.com  port: 443  ssl:    key-alias: cloud.tencent.com    key-store-password: ***    key-store-type: PKCS12    key-store: classpath:cloud.tencent.com.pfx
```

  The main parameters of the configuration file are described as below:

  - **Classpath:** Refers to the project's `src/main/java` and `src/main/resources` paths. Files stored under these two paths can be referenced using the classpath as the path.
  - **Key-store:** The location where the certificate file is stored, i.e., the path where the certificate file is stored in Step 4.
  - **key-store-password:** The password for the password file, specifying the password for the keystore.
    - If a private key password was set when applying for the certificate, please enter the private key password;
    - If a private key password was not set when applying for the certificate, please enter the password found within the keystorePass.txt file in the `cloud.tencent.com_tomcat` directory.
  - **Key-alias:** This is the key alias. For certificates downloaded from the SSL Certificate Service Console, the alias is the domain to which the certificate is bound, such as `cloud.tencent.com`.

7. Restart the Spring Boot service. If the startup is successful, you can access it through `https://cloud.tencent.com`.

   If the security lock icon is displayed in the browser, the certificate has been installed successfully. The details are as shown below:



SSL Root Certificate Download

Last updated: 2023-10-08 11:28:54

Overview

A root certificate is a public key certificate belonging to a certificate authority (CA) and serves as the starting point of the SSL certificate trust chain. This document will guide you on how to download a root certificate.

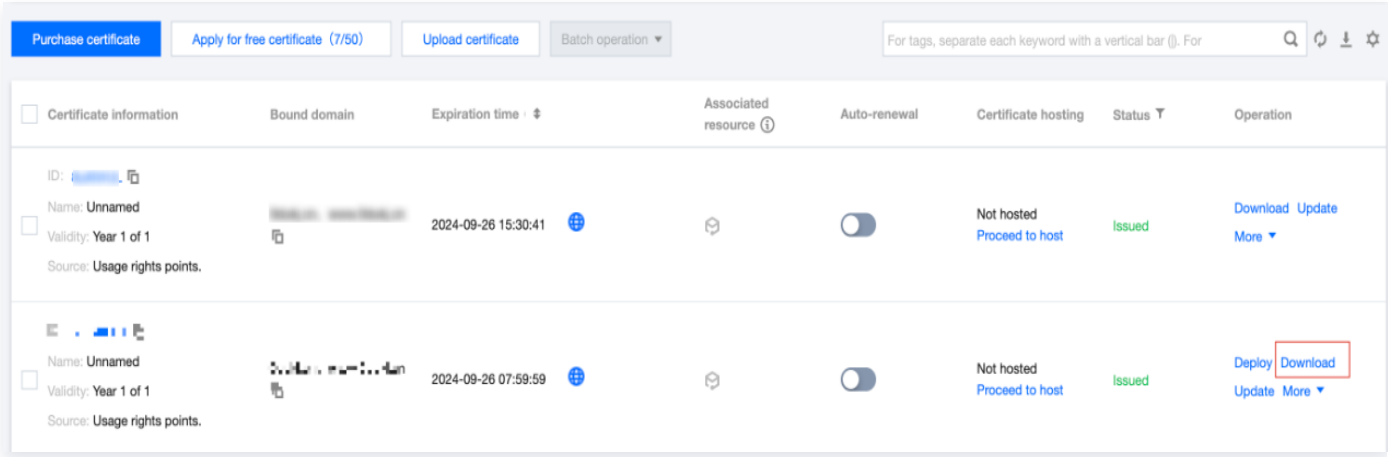
Use Cases

- If your business is accessed through a browser, such as a website, there is no need to concern yourself with the root certificate. This is because the root certificate is already built into the browser. You only need to install the issued SSL certificate on the web service to enable HTTPS communication between the browser and the web service. For more details, please refer to How to select a deployment type for an SSL certificate? .
- If your business requires non-browser clients to access your services, you will need to install the root certificate in the corresponding client, as these clients do not have pre-installed root certificates. This ensures normal communication between the client and the server.

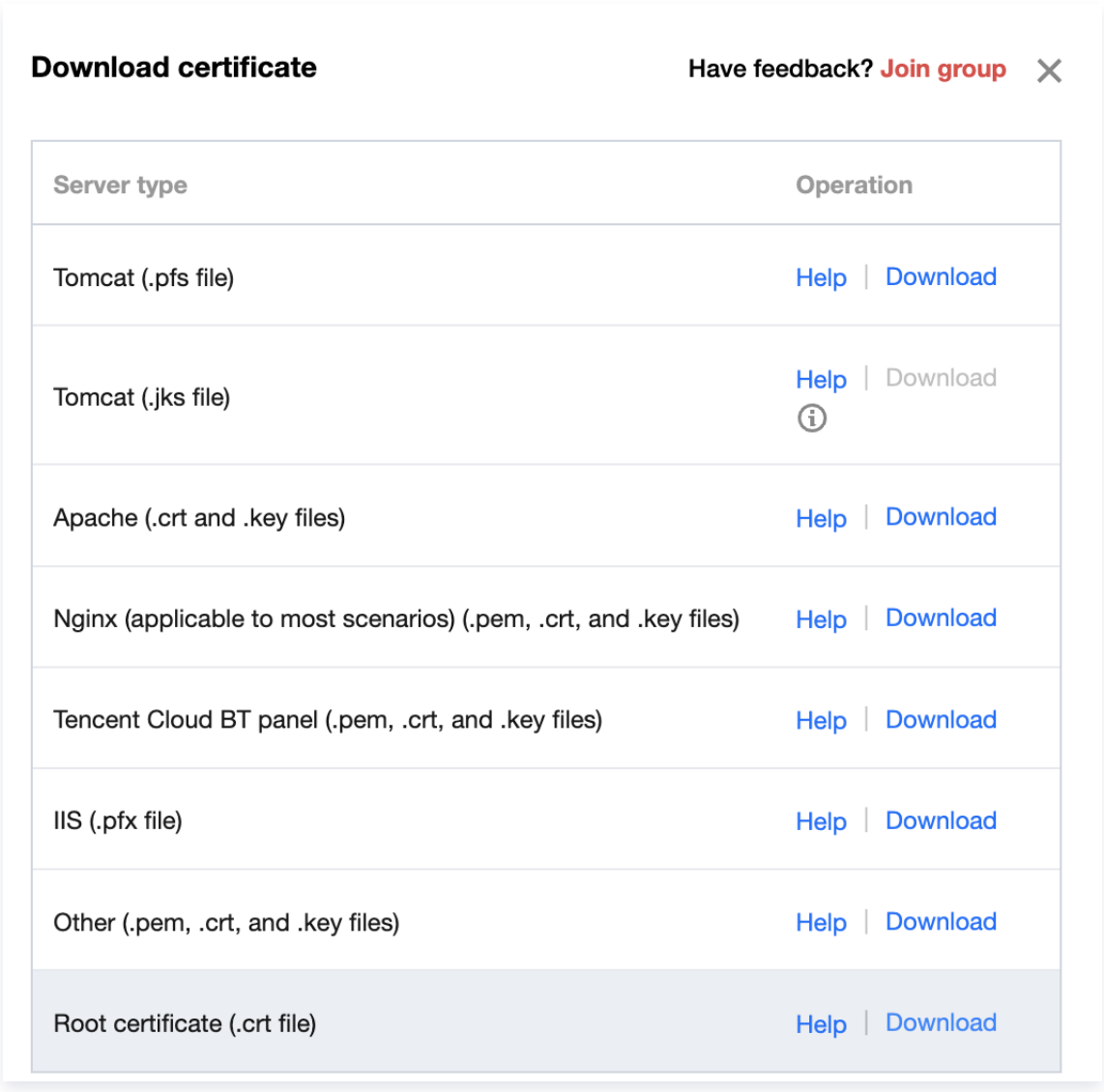
> ⚠ Note
> - Installing a root certificate in non-browser clients may affect your business due to reasons such as root certificate expiration or policy changes. It is recommended to use methods like the system default trust store for client verification.
> - Given the variety of client service types, please consult your client manufacturer or developer for guidance on installing the root certificate.

Instructions

1. Log in to the Tencent Cloud SSL Certificate Service console .
2. Select the SSL certificate for the root certificate you need to download and click **Download** as shown below:



3. In the "Download Certificate" pop-up window, click **Download** to download the root certificate, as shown below:



Installation of Chinese SM (SM2) SSL Certificate
Wotrus
Installation and Deployment of Wotrus National Cryptography Standard SSL Certificate on Apache Server

Last updated: 2023-10-08 11:36:33

Scenario

This document provides guidance on how to install a National Cryptography Standard SSL certificate on an Apache server.

Note

- The National Cryptography Standard SSL certificate is currently only supported on Apache servers in a Linux environment.
- The certificate name `cloud.tencent.com` is used as an example.
- It is suggested to use Apache version `apache-2.4.46` or `apache-2.4.48`. You can download it from the Apache official website or click here for a quick download of `apache-2.4.48`. If you need to use other versions, please contact us.
- The current server OS is CentOS 7. Detailed steps vary slightly with the OS version.
- Before installing the SSL certificate, please ensure that port 443 is enabled on your Apache server to avoid any issues enabling HTTPS after the certificate is installed. For more details, refer to How Do I Enable Port 443 for a VM?
- For detailed directions on how to upload SSL certificate files to a server, see Copying Local Files to CVMs.

Preparations

- A remote file copy tool such as WinSCP has been prepared. It is recommended to download the latest version from the official website. If you need to deploy to Tencent

Cloud Server, it is suggested to use the file upload function of the cloud server. For more details, please refer to Uploading Files to Cloud Server.

- Install the remote login tool such as PuTTY or Xshell.
- A National Cryptography Standard (SM2) SSL certificate has been purchased.
- The data required to install the SSL certificate includes:

Name	Note
Server IP address	The server IP address, which is used to connect the PC to the server.
Username	The username used to log in to the server.
Password	The password used to log in to the server.

Note
For a CVM instance purchased on the Tencent Cloud official website, log in to the CVM console to get the server IP address, username, and password.

Instructions

Environment configuration

Note
- To install the National Cryptography Standard SSL certificate on an Apache server, the server must have the relevant environment support module. The following will guide you through the process of compiling and configuring an Apache server that supports the National Cryptography Standard SSL certificate.
- The directories mentioned in the following steps are the directories of the test environment. Determine their specific paths based on your actual environment and needs.

1. Log in to the Apache server remotely. For instance, using the "PuTTY" tool.
2. **Install Compilation Tools**: If your system is brand new, please first install the C++ development environment on the server to provide support for compilation. You can use the following command for installation.

```
yum install -y gcc
yum install -y gcc-c++
```

3. **Download and compile apr** (using `apr 1.7.0` as an example). You can download apr to the server and compile it by entering the following command on the server. Due to different operating system versions, the detailed operation steps may vary slightly.

```
#Switch to the /usr/local/ directory
cd /usr/local/
#Downloading apr v1.7.0
wget -c http://mirrors.tencent.com/apache/apr/apr-1.7.0.tar.gz
#Decompress the downloaded apr 1.7.0 package
tar -zvxf apr-1.7.0.tar.gz
#Enter the decompressed apr 1.7.0 folder and specify the compilation directory
path.
cd apr-1.7.0/
./configure --prefix=/usr/local/apr
#Compile and Install APR
make && make install
```

4. **Download and compile the apr-util** (It is recommended to use the `apr-util-1.5` version, with `apr-util-1.5` as an example).

```
#Switch to the /usr/local/ directory
cd /usr/local/
#Download apr-util-1.5.4
wget -c http://archive.apache.org/dist/apr/apr-util-1.5.4.tar.gz
#Decompress the downloaded apr-util-1.5.4 package
tar -zvxf apr-util-1.5.4.tar.gz
#Enter the decompressed apr-util-1.5.4 folder and specify the compilation directory
path.
cd /usr/local/apr-util-1.5.4/
./configure --prefix=/usr/local/apr-util --with-apr=/usr/local/apr
#Compile and Install apr-util
make && make install
```

> **Note**
> If you encounter an error message `#include <expat.h> ^ compilation terminated.` when executing the make command, please enter the command `yum install -y expat-devel` to install the dependency library.

5. **Install pcre**. You can proceed with the installation in two ways.
   - It is recommended to use `yum` for installation.

```
yum install -y pcre-devel
```

- Compile and install.

```
#Switch to the /usr/local/ directory
cd /usr/local/
#Download pcre-8.43
wget -c https://ftp.pcre.org/pub/pcre/pcre-8.43.tar.gz
#Decompress the downloaded pcre-8.43 package
tar -zvxf pcre-8.43.tar.gz
#Enter the decompressed pcre-8.43 folder and specify the compilation
directory path.
cd pcre-8.43/
./configure --prefix=/usr/local/pcre
#Compile and Install PCRE
make && make install
```

6. **Apache Server Installation:** After the above three files have been compiled and installed, please download the Apache National Cryptography version and the National Cryptography module to the `/usr/local` directory for compilation and installation.

> **Note**
> - Please do not modify the National Cryptography module filename `wotrus_ssl.tar.gz` during decompression and installation, as it may lead to installation errors.
> - If you cannot find related files such as pcre, apr-util, or apr during the installation of Apache, please add files like `/pcre/bin`, `/apr-util/bin`, or `/apr/bin` to the system path.

```
#Download the Apache httpd-2.4.48 compressed package
wget -c http://mirrors.tencent.com/apache/httpd/httpd-2.4.48.tar.gz
#Downloading the National Cryptography Module
wget -c https://www.wotrus.com/download/wotrus_ssl.tar.gz
#Decompress the downloaded wotrus_ssl package
tar -zvxf wotrus_ssl.tar.gz
#Decompress the downloaded httpd-2.4.48 package
tar -zvxf httpd-2.4.48.tar.gz
#Enter the decompressed httpd-2.4.48 folder and specify the compilation directory
path.
cd httpd-2.4.48/
```

```
./configure --prefix=/usr/local/httpd --enable-so --enable-ssl --enable-cgi --enable-
rewrite --enable-modules=most --enable-mpms-shared=all --with-mpm=prefork --
with-zlib --with-apr=/usr/local/apr --with-apr-util=/usr/local/apr-util --with-
ssl=/usr/local/wotrus_ssl2.0
#Compile and Install Apache
make && make install
```

Installation of National Cryptography Standard Certificate

1. You have downloaded and decompressed the `cloud.tencent.com` certificate file package to a local directory from the SSL Certificate Service Console. After decompression, you can obtain the relevant type of certificate files. This includes the Apache folder and CSR file:
   - **Folder Name**: Apache
   - **Folder content**:
     - `1_root_sign_bundle.crt` Certificate file
     - `2_root_encrypt_bundle.crt` Certificate file
     - `3_cloud.tencent.com_sign.crt` Certificate file
     - `4_cloud.tencent.com_encrypt.crt` Certificate file
     - `5_cloud.tencent.com.key` : Private key file
   - **CSR File Content**:
     - `cloud.tencent.com_sign.csr` file
     - `cloud.tencent.com_encrypt.csr` file

> **Note**
> The CSR file, either uploaded by you or generated online by the system during the certificate application, is provided to the CA. This file can be disregarded during installation.

2. Log in to the Apache server using "WinSCP", a tool for copying files between local and remote computers.

> **Note**
> - For detailed directions, see Uploading files via WinSCP to a Linux CVM from Windows.
> - We recommend using the file upload feature of the Cloud Virtual Machine (CVM) for deployment to Tencent Cloud CVM. For more details, please refer to Uploading Files to CVM.

3. Navigate to the `/usr/local/httpd/conf` directory, create a new `cert` directory, and copy the obtained `1_root_sign_bundle.crt` certificate file, `2_root_encrypt_bundle.crt` certificate file, `3_cloud.tencent.com_sign.crt` certificate file, `4_cloud.tencent.com_encrypt.crt` certificate file, and `5_cloud.tencent.com.key` private key file from the local directory to the `/usr/local/httpd/conf/cert` directory on the Apache server.
4. Navigate to the `/usr/local/httpd/conf` directory and edit the `httpd.conf` file following these steps:
   4.1 Please add `ServerName (your domain name):80` below `#ServerName www.example.com:80` .
   4.2 Please remove the `#` before `LoadModule ssl_module modules/mod_ssl.so` .
   4.3 Please add the `Include conf/ssl.conf` file content below `#Include conf/extra/httpd-ssl.conf` , then save and exit.
5. In the `/usr/local/httpd/conf` directory, create a new `ssl.conf` file and add the following configuration:

```
Listen 443
<VirtualHost *:443>
#Enter the certificate name
ServerName cloud.tencent.com
#Enter the Website File Path
DocumentRoot Website Root Directory
#Enable SSL
SSLEngine on
# SM2 Certificate Sign Configuration
SSLCertificateFile /usr/local/httpd/conf/cert/3_cloud.tencent.com_sign.crt
SSLCertificateKeyFile /usr/local/httpd/conf/cert/5_cloud.tencent.com.key
SSLCertificateChainFile /usr/local/httpd/conf/cert/1_root_sign_bundle.crt
# SM2 Certificate Encrypt Configuration
SSLCertificateFile /usr/local/httpd/conf/cert/4_cloud.tencent.com_encrypt.crt
SSLCertificateKeyFile /usr/local/httpd/conf/cert/5_cloud.tencent.com.key
SSLCertificateChainFile /usr/local/httpd/conf/cert/2_root_encrypt_bundle.crt
# The .key in the sign and encrypt configurations is the same.
Configure the following protocols
SSLProtocol all -SSLv2 -SSLv3
#Configure the cipher suite according to the OpenSSL standard.
SSLCipherSuite SM2-WITH-SMS4-
SM3:ECDH:AESGCM:HIGH:MEDIUM:!RC4:!DH:!MD5:!aNULL:!eNULL
SSLHonorCipherOrder on
<Directory "website root directory">
Options -Indexes -FollowSymLinks +ExecCGI
AllowOverride None
Order allow,deny
```

```
Allow from all
Require all granted
</Directory>
</VirtualHost>
```

> **Note**
> The above configuration content is for reference only. Please configure the specific certificate name, certificate directory, `Directory`, etc., according to the actual environment.

6. Execute the following command to validate configuration file issues.

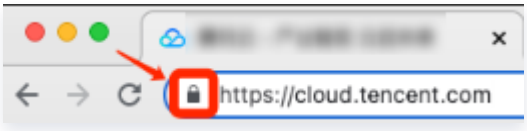
```
/usr/local/httpd/bin/httpd -t
```

   - If `Syntax OK` is displayed, it indicates that the configuration is correct and the Apache server can be started.
   - If the message is not `Syntax OK`, please reconfigure or modify according to the prompts to resolve the issue.

7. Execute the following command to restart the Apache server, then you can access it through `https://cloud.tencent.com`.

```
/usr/local/httpd/bin/httpd -k restart
```

   - If the security lock icon is displayed in the browser, the certificate has been installed successfully. The details are as shown below:



   - In case of a website access exception, troubleshoot the issue by referring to the following FAQs:
     - Unable to access the website via HTTPS
     - After deploying the SSL certificate, the browser indicates "Website connection is not secure"
     - Is your site indicating an insecure connection?
     - What should I do if I am prompted that HTTPS is not secure after reapplying for deployment upon expiration of the SSL certificate?
     - After deploying the SSL certificate on the server, a 404 error occurs when

accessing resources

Optional dual installation of International Standard Certificate and National Cryptography Standard Certificate

If you need to resolve browser compatibility issues by installing both an international standard certificate and a National Cryptography Standard certificate, you can do so by following these steps:

> **Note**
> Tencent Cloud offers free DV SSL certificates to users who have purchased the National Cryptography Standard DNSPod certificate to smoothly resolve browser compatibility issues. For certificate application, please refer to Free DV SSL certificates.

1. Utilize WinSCP (a tool for copying files between a local and a remote computer) to transfer the obtained international standard certificate files `1_root_bundle.crt`, `2_cloud.tencent.com.crt`, and the private key file `3_cloud.tencent.com.key` from the local directory to the `/usr/local/httpd/conf` directory on the Apache server.

   > **Note**
   > - For detailed directions, see Uploading files via WinSCP to a Linux CVM from Windows.
   > - We recommend using the file upload feature of the Cloud Virtual Machine (CVM) for deployment to Tencent Cloud CVM. For more details, please refer to Uploading Files to CVM.

2. Edit the `ssl.conf` file in the `/usr/local/httpd/conf` directory.
3. Locate `SSLEngine on` and start a new line beneath it, incorporating the following details:

```
SSLCertificateFile /usr/local/httpd/conf/cert/2_cloud.tencent.com.crt
SSLCertificateKeyFile /usr/local/httpd/conf/cert/3_cloud.tencent.com.key
SSLCertificateChainFile /usr/local/httpd/conf/cert/1_root_bundle.crt
```

   > **Note**
   > The above configuration content is for reference only. The specific certificate name and certificate directory should be configured according to the actual environment.

4. Execute the following command to validate configuration file issues.

```
/usr/local/httpd/bin/httpd -t
```

- If `Syntax OK` is displayed, it indicates that the configuration is correct and the Apache server can be started.
- If the message is not `Syntax OK`, please reconfigure or modify according to the prompts to resolve the issue.

5. Restarting the Apache server can resolve browser compatibility issues.

> **Note**
> If anything goes wrong during this process, please contact us.

Installation and Deployment of Nginx For Linux National Cryptography Standard SSL Certificate (Wotrus)

Last updated: 2023-10-08 14:26:46

Scenario

This document provides guidance on how to install a National Cryptography Standard SSL certificate on an Nginx server.

Note

- The Nginx version `nginx/1.18.0` is used as an example.
- The certificate name `cloud.tencent.com` is used as an example.
- The current server OS is CentOS 7. Detailed steps vary slightly with the OS version.
- Before installing the SSL certificate, please enable port "443" on the Nginx server to ensure that HTTPS can be enabled after the certificate installation. For more details, refer to How to Open Port 443 on a Server?
- For detailed directions on how to upload SSL certificate files to a server, see Copying Local Files to CVMs.

Preparations

- A remote file copy tool such as WinSCP has been prepared. It is recommended to download the latest version from the official website. If you need to deploy to Tencent Cloud Server, it is suggested to use the file upload function of the cloud server. For more details, please refer to Uploading Files to Cloud Server.
- Install the remote login tool such as PuTTY or Xshell.
- A National Cryptography Standard (SM2) SSL certificate has been purchased.
- The data required to install the SSL certificate includes:

Name	Note
Server IP address	The server IP address, which is used to connect the PC to the server.
Username	The username used to log in to the server.

Password	The password used to log in to the server.

Note
For a CVM instance purchased on the Tencent Cloud official website, log in to the CVM console to get the server IP address, username, and password.

Instructions

Environment configuration

Note
- To install a National Cryptography Standard SSL certificate on an Nginx server, the server must have the relevant environment support module. The following will guide you through the process of compiling and configuring an Nginx server that supports National Cryptography Standard SSL certificates.
- The directories mentioned in the following steps are the directories of the test environment. Determine their specific paths based on your actual environment and needs.

1. Log in to the Nginx server remotely, for instance, using the "PuTTY" tool.
2. **Install Compilation Tools**: If your system is brand new, please first install the C++ development environment, pcre-devel, and zlib-devel software on the server to provide environmental support for compilation. You can use the following commands for installation.

```
#Setting up C++ Development Environment
yum install -y gcc gcc-c++
#Install pcre-devel
yum install pcre-devel -y
#Install zlib-devel
yum install zlib-devel -y
```

3. **Download and compile Nginx**: You can download the Nginx National Cryptography version and module to the server and compile and install them by entering commands in the following order on the server. Due to differences in operating system versions, the detailed operation steps may vary slightly.

Note
- Using `nginx-1.18.0` as an example, the directory is `/usr/local`. Please determine

according to your actual environment and needs.

- Please do not modify the National Cryptography module filename `wotrus_ssl.tar.gz` during decompression and installation, as it may lead to installation errors.

```
#Switch to the /usr/local/ directory
cd /usr/local/
#Download nginx-1.18.0
wget -c http://nginx.org/download/nginx-1.18.0.tar.gz
#Downloading the SM2 National Cryptography Module
wget -c https://www.wotrus.com/download/wotrus_ssl.tar.gz
#Decompress the downloaded nginx-1.18.0 package.
tar -zvxf nginx-1.18.0.tar.gz
#Decompress the downloaded National Cryptography SM2 module package.
tar -zvxf wotrus_ssl.tar.gz
#Enter the decompressed nginx-1.18.0 folder.
cd nginx-1.18.0/
#Specify the compilation directory path and the module to be compiled. Additional modules can be added as needed.
./configure --prefix=/usr/local/nginx --with-http_stub_status_module --with-stream --with-http_ssl_module --with-stream_ssl_module --with-openssl=/usr/local/wotrus_ssl2.0
#Compile and Install Nginx
make && make install
```

4. If you encounter an error `make[1]: *** [/usr/local/wotrus_ssl2.0/.openssl/include/openssl/ssl.h] Error 127` during the compilation process, you need to navigate to the `nginx-1.18.0/auto/lib/openssl` directory and edit the `conf` file. The following content needs to be modified:

```
CORE_INCS="$CORE_INCS $OPENSSL/.openssl/include"
CORE_DEPS="$CORE_DEPS $OPENSSL/.openssl/include/openssl/ssl.h"
CORE_LIBS="$CORE_LIBS $OPENSSL/.openssl/lib/libssl.a"
CORE_LIBS="$CORE_LIBS $OPENSSL/.openssl/lib/libcrypto.a"
```

Modify as follows:

```
CORE_INCS="$CORE_INCS $OPENSSL/include"
CORE_DEPS="$CORE_DEPS $OPENSSL/include/openssl/ssl.h"
CORE_LIBS="$CORE_LIBS $OPENSSL/lib/libssl.a"
CORE_LIBS="$CORE_LIBS $OPENSSL/lib/libcrypto.a"
```

5. After saving the file, execute `make clean` to clear the compilation configuration. Then, re-enter the `nginx-1.18.0` folder and execute `./configure --prefix=/usr/local/nginx --with-http_stub_status_module --with-stream --with-http_ssl_module --with-stream_ssl_module --with-openssl=/usr/local/wotrus_ssl2.0` and `make && make install`.

Installation of National Cryptography Standard Certificate

1. You have downloaded and decompressed the `cloud.tencent.com` certificate file package from the SSL Certificate Service Console to a local directory. After decompression, you can obtain the relevant type of certificate files. This includes the Nginx directory and CSR files:
   - **Folder Name:** Nginx
   - **Folder content:**
     - `1_cloud.tencent.com_sign_bundle.crt` Certificate file
     - `2_cloud.tencent.com_encrypt_bundle.crt` Certificate file
     - `3_cloud.tencent.com.key` Private key file
   - **CSR File Content:**
     - `cloud.tencent.com_sign.csr` file
     - `cloud.tencent.com_encrypt.csr` file

> **Note**
> The CSR file, either uploaded by you or generated online by the system during the certificate application, is provided to the CA. This file can be disregarded during installation.

2. Log in to the Nginx server using "WinSCP", a tool for copying files between local and remote computers.

> **Note**
> - For detailed directions, see Uploading files via WinSCP to a Linux CVM from Windows.
> - We recommend using the file upload feature of the Cloud Virtual Machine (CVM) for deployment to Tencent Cloud CVM. For more details, please refer to Uploading Files to CVM.

3. Navigate to the `/usr/local/nginx/conf` directory, create a new `sm2` directory, and copy the obtained certificate files `1_cloud.tencent.com_sign_bundle.crt`,

`2_cloud.tencent.com_encrypt_bundle.crt`, and the private key file `3_cloud.tencent.com.key` from the local directory to this `sm2` directory.

4. Navigate to the `/usr/local/nginx/conf` directory, edit the `nginx.conf` file, and add the following configuration:

```
server {
listen 443 ssl;
server_name domain.com;
ssl_certificate /usr/local/nginx/conf/sm2/1_cloud.tencent.com_sign_bundle.crt;
ssl_certificate_key /usr/local/nginx/conf/sm2/3_cloud.tencent.com.key;
ssl_certificate /usr/local/nginx/conf/sm2/2_cloud.tencent.com_encrypt_bundle.crt;
ssl_certificate_key /usr/local/nginx/conf/sm2/3_cloud.tencent.com.key;
#First, configure the signing certificate, then the encryption certificate. The private key for both the signing and encryption certificates is the same!
ssl_session_timeout 5m;
ssl_protocols TLSv1.1 TLSv1.2 TLSv1.3;
ssl_ciphers ECC-SM4-SM3:ECDH:AESGCM:HIGH:MEDIUM:!RC4:!DH:!MD5:!aNULL:!eNULL;
ssl_prefer_server_ciphers on;
location / {
 root html;
 index index.html index.htm;
}
}
```

> **Note**
> The above configuration content is for reference only. Please configure the specific certificate name, certificate directory, `location`, and other settings according to your actual environment.

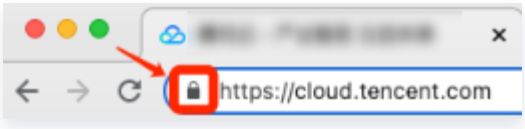
5. Please verify the configuration file issues by executing the following command.

```
/usr/local/nginx/sbin/nginx -t
```

   - If `Syntax OK` is displayed, it indicates that the configuration is correct and the Nginx server can be started.
   - If the message is not `Syntax OK`, please reconfigure or modify according to the prompts to resolve the issue.

6. Restart the Nginx server and then you can access it through `https://cloud.tencent.com`.
   - If the security lock icon is displayed in the browser, the certificate has been installed

successfully. The details are as shown below:



- In case of a website access exception, troubleshoot the issue by referring to the following FAQs:
  - Unable to access the website via HTTPS
  - After deploying the SSL certificate, the browser indicates "Website connection is not secure"
  - Is your site indicating an insecure connection?
  - What should I do if I am prompted that HTTPS is not secure after reapplying for deployment upon expiration of the SSL certificate?
  - After deploying the SSL certificate on the server, a 404 error occurs when accessing resources

Optional dual installation of International Standard Certificate and National Cryptography Standard Certificate

If you need to resolve browser compatibility issues by installing both an international standard certificate and a National Cryptography Standard certificate, you can do so by following these steps:

> **Note**
> Tencent Cloud offers free DV SSL certificates to users who have purchased the National Cryptography Standard DNSPod certificate to smoothly resolve browser compatibility issues. For certificate application, please refer to Free DV SSL certificates.

1. Utilize WinSCP (a tool for copying files between a local and a remote computer) to transfer the `1_root_bundle.crt` certificate file and the `2_cloud.tencent.com.key` private key file from the Nginx folder in the obtained international standard certificate zip package, from the local directory to the `/usr/local/nginx/conf/sm2` directory on the Nginx server.

   > **Note**
   > - For detailed directions, see Uploading files via WinSCP to a Linux CVM from Windows.
   > - We recommend using the file upload feature of the Cloud Virtual Machine (CVM) for deployment to Tencent Cloud CVM. For more details, please refer to Uploading Files to CVM.

2. Edit the `ssl.conf` file in the `/usr/local/nginx/conf` directory.
3. Locate `server_name cloud.tencent.com` and start a new line beneath it, then add the following content:

```
ssl_certificate /usr/local/nginx/conf/sm2/1_cloud.tencent.com_bundle.crt;
ssl_certificate_key /usr/local/nginx/conf/sm2/2_cloud.tencent.com.key;
```

> **Note**
> The above configuration content is for reference only. Please configure the specific certificate name and certificate directory according to the actual environment.

4. Execute the following command to validate configuration file issues.

```
/usr/local/nginx/sbin/nginx -t
```

   - If `Syntax OK` is displayed, it indicates that the configuration is correct and the Nginx server can be started.
   - If the message is not `Syntax OK`, please reconfigure or modify according to the prompts to resolve the issue.
5. Restarting the Nginx server can resolve browser compatibility issues.

> **Note**
> If anything goes wrong during this process, please contact us.

DNSPod Installation of National Cryptographic SSL Certificate for Nginx on Linux (DNSPod)

Last updated: 2023-10-08 14:41:20

Scenario

This document guides you on how to install the DNSPod SSL certificate, compliant with the national encryption standard, on an Nginx server.

Note:

- The Nginx version used as an example in this document is nginx/1.18.0.
- The certificate name cloud.tencent.com is used as an example in this document.
- The current server OS is CentOS 7. Detailed steps vary slightly with the OS version.
- Before installing the SSL certificate, please ensure that port 443 is enabled on your Nginx server to avoid any issues enabling HTTPS after the certificate installation. For detailed instructions, refer to How to Open Port 443 on a Server?.
- For detailed directions on how to upload SSL certificate files to a server, see Copying Local Files to CVMs.

Preparations

- A remote file copy tool, such as WinSCP, has been prepared (it is recommended to download the latest version from the official website). If you need to deploy to Tencent Cloud's CVM, it is suggested to use the file upload feature of the CVM. For more details, please refer to Uploading Files to CVM.
- Install the remote login tool such as PuTTY or Xshell.
- You have already purchased the DNSPod SSL certificate compliant with the national encryption standard (SM2).
- The data required to install the SSL certificate includes:

Name	Note
Server IP address	The server IP address, which is used to connect the PC to the server.

Password	The password used to log in to the server.
Username	The username used to log in to the server.

Note:
For a CVM instance purchased on the Tencent Cloud official website, log in to the CVM console to get the server IP address, username, and password.

Instructions

Environment configuration

Note:
- The DNSPod SSL certificate, compliant with the national encryption standard, is installed on an Nginx server, which requires a supporting environment module. The following sections will guide you through the process of compiling and configuring an Nginx server that supports the DNSPod SSL certificate compliant with the national encryption standard.
- The directories mentioned in the following steps are the directories of the test environment. Determine their specific paths based on your actual environment and needs.

1. Log in to the Apache server remotely. For instance, using the **PuTTY tool** for login.
2. Install the compilation dependency packages. If your system is brand new, please first install the following dependency packages on the server to provide environmental support for compilation. You can use the following command for installation.

```
yum install -y gcc gcc-c++ wget make perl pcre-devel zlib-devel
```

3. Download and compile Nginx. You can download the Nginx national encryption version and the national encryption module to the server and compile and install them by entering the following commands in sequence on the server. Due to differences in operating system versions, the detailed operation steps may vary slightly.

Note:
The DNSPod national encryption module filename `gmssl_openssl_2021.1011.tar.gz` should not be altered during extraction and installation, as it may lead to installation errors.

> The directory for nginx-1.18.0 is `/usr/local`, please determine according to your actual environment and requirements.

3.1 Download the Nginx version compliant with the national encryption standard and the corresponding encryption module.

```
# Navigate to the /usr/local/ directory
cd /usr/local/
# Download nginx-1.18.0
wget -c http://nginx.org/download/nginx-1.18.0.tar.gz
# Downloading the DNSPod National Encryption Module
wget -c https://certificate-1258344699.cos.ap-guangzhou.myqcloud.com/public/gmssl_openssl_2021.1011.tar.gz
# Decompress the downloaded nginx-1.18.0 package
tar -zvxf nginx-1.18.0.tar.gz
# Decompress the downloaded gmssl_openssl_2021.1011 package.
tar -zvxf gmssl_openssl_2021.1011.tar.gz
```

3.2 Edit the `/usr/local/nginx-1.18.0/auto/lib/openssl/conf` file, pointing to the static gmssl, with the following modifications:

Before modification:

```
CORE_INCS="$CORE_INCS $OPENSSL/.openssl/include"
CORE_DEPS="$CORE_DEPS $OPENSSL/.openssl/include/openssl/ssl.h"
CORE_LIBS="$CORE_LIBS $OPENSSL/.openssl/lib/libssl.a"
CORE_LIBS="$CORE_LIBS $OPENSSL/.openssl/lib/libcrypto.a"
```

After modification:

```
CORE_INCS="$CORE_INCS $OPENSSL/include"
CORE_DEPS="$CORE_DEPS $OPENSSL/include/openssl/ssl.h"
CORE_LIBS="$CORE_LIBS $OPENSSL/lib/libssl.a"
CORE_LIBS="$CORE_LIBS $OPENSSL/lib/libcrypto.a"
```

3.3 Compile and install Nginx.

```
# Navigate to the /usr/local/ directory
cd /usr/local/
# Specify the path to the compilation directory and the module to be compiled. Additional modules can be added as needed.
```

```
./configure --prefix=/usr/local/nginx --without-http_gzip_module --with-
http_ssl_module --with-http_stub_status_module --with-http_v2_module --with-
file-aio --with-openssl="/usr/local/gmssl" --with-cc-opt="-
I/usr/local/gmssl/include" --with-ld-opt="-lm"
# Compilation and Installation
make && make install
```

Installation of National Cryptography Standard Certificate

1. You have already downloaded and decompressed the `cloud.tencent.com` certificate file package to a local directory from the SSL Certificate Service Console. After decompression, you can obtain the relevant type of certificate files. This includes the Nginx folder and CSR files:
   - Folder name: `cloud.tencent.com_nginx`
   - Folder content:
     - `cloud.tencent.com.key` private key file
     - `cloud.tencent.com._sign_bundle.crt` Signature certificate file
     - `cloud.tencent.com._sign.key` Signature certificate private key file
     - `cloud.tencent.com._sign_bundle.pem` (This file can be disregarded)
     - `cloud.tencent.com._encrypt_bundle.crt` Encrypted certificate file
     - `cloud.tencent.com._encrypt.key` Encryption certificate private key file
     - `cloud.tencent.com._encrypt_bundle.pem` (This file can be disregarded)
   - CSR file content:
     - `cloud.tencent.com._sign.csr` file

     > **Note:**
     > The CSR file, either uploaded by you or generated online by the system during the certificate application, is provided to the CA. This file can be disregarded during installation.

2. Log in to the Apache server using "WinSCP", a tool for copying files between local and remote computers.

   > **Note:**
   > - For detailed directions, see Uploading files via WinSCP to a Linux CVM from Windows.
   > - We recommend using the file upload feature of the Cloud Virtual Machine (CVM) for deployment to Tencent Cloud. For more details, please refer to Uploading

> Files to CVM.

3. Navigate to the `/usr/local/nginx/conf` directory and create a new directory named 'cert'.

```
cd /usr/local/nginx/conf/
mkdir cert
```

Copy the obtained private key file `cloud.tencent.com.key`, signature certificate file `cloud.tencent.com._sign_bundle.crt`, signature certificate private key file `cloud.tencent.com._sign.key`, encryption certificate file `cloud.tencent.com._encrypt_bundle.crt`, and encryption certificate private key file `cloud.tencent.com._encrypt.key` from the local directory to the `/usr/local/nginx/conf/cert` directory on the Nginx server.

4. Navigate to the `/usr/local/nginx/conf` directory, edit the `nginx.conf` file, and add the following configuration:

```
server
{
  listen 443 ssl;
  ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
  # SM2 Encryption Cipher Suite
  ssl_ciphers ECC-SM4-GCM-SM3:ECC-SM4-CBC-SM3:ECDHE-RSA-AES128-GCM-
SHA256:AES128-SHA:DES-CBC3-SHA;
  ssl_verify_client off;
  # Signature Certificate/Private Key
  ssl_certificate /usr/local/nginx/conf/cert/cloud.tencent.com._sign_bundle.crt;
  ssl_certificate_key /usr/local/nginx/conf/cert/cloud.tencent.com._sign.key;
  # Encryption Certificate/Private Key
  ssl_certificate /usr/local/nginx/conf/cert/cloud.tencent.com._encrypt_bundle.crt;
  ssl_certificate_key /usr/local/nginx/conf/cert/cloud.tencent.com._encrypt.key;
  location /
  {
    root html;
    index index.html index.htm;
  }
}
```

5. Configure the national encryption standard license permission.

Please download the national encryption standard DNSPod certificate license file (filename: `gmssl_xxxx.lic`) from the SSL Certificate Service Console to your local directory, and then upload it to the `/etc` directory on the Nginx server.

6. You can verify the configuration file issues by executing the following command.

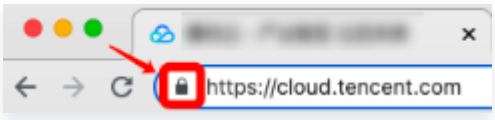
```
/usr/local/nginx/sbin/nginx -t
```

   - If the message "Syntax OK" is displayed, it indicates that the configuration is correct and the Nginx service can be initiated.
   - If the prompt is not 'Syntax OK', please reconfigure or modify according to the prompt to resolve the issue.

7. Run the following command to start the Nginx service, and then you can access it through `https://cloud.tencent.com` .

```
cd /usr/local/nginx/sbin
./nginx
```

   - If the security lock icon is displayed in the browser, the certificate has been installed successfully. The details are as shown below:



   - In case of a website access exception, troubleshoot the issue by referring to the following FAQs:
     - Unable to access the website via HTTPS
     - After deploying the SSL certificate, the browser indicates "Website connection is not secure"
     - Is your site indicating an insecure connection?
     - What should I do if I am prompted that HTTPS is not secure after reapplying for deployment upon expiration of the SSL certificate?
     - After deploying the SSL certificate on the server, a 404 error occurs when accessing resources

Optional dual installation of International Standard Certificate and National Cryptography Standard Certificate

If you need to resolve browser compatibility issues by installing both an international standard certificate and a National Cryptography Standard certificate, you can do so by following these steps:

> **Note:**

> Tencent Cloud offers free DV SSL certificates to users who have purchased the DNSPod certificate compliant with the national encryption standard, effectively resolving browser compatibility issues. For certificate application, please refer to Free DV SSL Certificate.

1. Utilize WinSCP (a tool for copying files between a local and a remote computer) to transfer the obtained international standard certificate files `1_root_bundle.crt`, `2_cloud.tencent.com.crt`, and the private key file `3_cloud.tencent.com.key` from the local directory to the `/usr/local/nginx/conf/cert` directory on the Nginx server.

   > **Note:**
   > - For detailed directions, see Uploading files via WinSCP to a Linux CVM from Windows.
   > - We recommend using the file upload feature of the Cloud Virtual Machine (CVM) for deployment to Tencent Cloud. For more details, please refer to Uploading Files to CVM.

2. Edit the `nginx.conf` file located in the `/usr/local/nginx/conf/` directory and add the following content:

```
SSLCertificateFile /usr/local/nginx/conf/cert/2_cloud.tencent.com.crt
SSLCertificateKeyFile /usr/local/nginx/conf/cert/3_cloud.tencent.com.key
SSLCertificateChainFile /usr/local/nginx/conf/cert/1_root_bundle.crt
```

3. Execute the following command to validate configuration file issues.

```
/usr/local/nginx/sbin/nginx -t
```

   - If the message "Syntax OK" is displayed, it indicates that the configuration is correct and the Nginx service can be initiated.
   - If the prompt is not 'Syntax OK', please reconfigure or modify according to the prompt to resolve the issue.

4. Reloading the Nginx service configuration can resolve browser compatibility issues.

```
/usr/local/nginx/sbin/nginx -s reload
```

> **Note:**
> If anything goes wrong during this process, please contact us.

Installation and Deployment of the National Encryption Standard SSL Certificate on Apache Server (DNSPod)

Last updated: 2023-10-08 15:14:07

Scenario

This document provides guidance on how to install the DNSPod SSL certificate, compliant with the national encryption standard, on an Apache server.

Note:

- The DNSPod SSL certificate, compliant with the national encryption standard, is currently only supported on Apache servers in a Linux environment.
- The certificate name cloud.tencent.com is used as an example in this document.
- It is recommended to use Apache versions apache-2.4.46 or apache-2.4.48. You can download these from the Apache official website or click here for a quick download of apache-2.4.48.
- The current server OS is CentOS 7. Detailed steps vary slightly with the OS version.
- Before installing the SSL certificate, please ensure that port 443 is enabled on your Apache server to prevent any issues with enabling HTTPS after the certificate is installed. For more details, refer to How Do I Enable Port 443 for a VM?
- For detailed directions on how to upload SSL certificate files to a server, see Copying Local Files to CVMs.

Preparations

- A remote file copy tool such as WinSCP has been installed (Download the latest version from the official website).

  We recommend using the file upload feature of the Cloud Virtual Machine (CVM) for deployment to Tencent Cloud. For more details, please refer to Uploading Files to CVM.
- Install the remote login tool such as PuTTY or Xshell (Download the latest version from the official website).
- You have already purchased the DNSPod SSL certificate compliant with the national encryption standard (SM2).

The data required to install the SSL certificate includes:

Name	Note
Server IP address	The server IP address, which is used to connect the PC to the server.
Username	The username used to log in to the server.
Password	The password used to log in to the server.

Note:
For a CVM instance purchased on the Tencent Cloud official website, log in to the CVM console to get the server IP address, username, and password.

Instructions

Environment configuration

Note:
- The DNSPod SSL certificate, compliant with the national encryption standard, is installed on an Apache server, which requires the relevant environment support module. The following text will guide you through the process of compiling and configuring an Apache server that supports the DNSPod SSL certificate, compliant with the national encryption standard.
- The directories mentioned in the following steps are the directories of the test environment. Determine their specific paths based on your actual environment and needs.

1. Log in to the Apache server remotely. For instance, using the PuTTY tool for login.
2. Install the compilation dependency package: If your system is brand new, please first install the following dependency packages on the server to provide environmental support for compilation. You can use the following command for installation.

```
yum install -y gcc gcc-c++ wget make perl pcre-devel expat-devel bison bison-devel flex flex-devel
```

3. Download and compile the installation of apr (taking apr 1.7.0 as an example). You can

download apr to the server and compile the installation by entering the following command on the server. Due to differences in operating system versions, the detailed operation steps may vary slightly.

```
# Navigate to the /usr/local/ directory
cd /usr/local/
# Download apr v1.7.0
wget -c http://mirrors.tencent.com/apache/apr/apr-1.7.0.tar.gz
# Decompress the downloaded apr 1.7.0 compressed package
tar -zvxf apr-1.7.0.tar.gz
# Enter the decompressed apr 1.7.0 folder and specify the path to the compilation directory.
cd apr-1.7.0/
./configure --prefix=/usr/local/apr
# Compilation and Installation of APR
make && make install
```

4. Download and compile the installation of apr-util (using apr-util-1.6 version as an example).

```
# Navigate to the /usr/local/ directory
cd /usr/local/
# Download apr-util-1.6.1
wget -c http://archive.apache.org/dist/apr/apr-util-1.6.1.tar.gz
# Decompress the downloaded apr-util-1.6.1 compressed package
tar -zvxf apr-util-1.6.1.tar.gz
# Enter the decompressed apr-util-1.6.1 folder and specify the path of the compilation directory.
cd /usr/local/apr-util-1.6.1/
./configure --prefix=/usr/local/apr-util --with-apr=/usr/local/apr
# Compilation and Installation of apr-util
make && make install
```

5. Apache Service Installation: After completing the above compilation and installation steps, please download the Apache National Encryption Edition and the National Encryption Module to the `/usr/local` directory for compilation and installation.

> **Note:**
> Please do not modify the DNSPod national encryption module filename `gmssl_openssl_2021.1011.tar.gz` during decompression and installation, as it may lead to installation errors.

> If you cannot find related files such as pcre, apr-util, or apr during the Apache installation process, please add files like `/pcre/bin`, `/apr-util/bin`, or `/apr/bin` to the system path.

```
# Navigate to the /usr/local/ directory
cd /usr/local/
# Download the Apache httpd-2.4.54 compressed package
wget -c http://mirrors.tencent.com/apache/httpd/httpd-2.4.54.tar.gz
# Downloading the DNSPod National Encryption Module
wget -c https://certificate-1258344699.cos.ap-
guangzhou.myqcloud.com/public/gmssl_openssl_2021.1011.tar.gz
# Decompress the downloaded gmssl_openssl_2021.1011 package.
tar -zvxf gmssl_openssl_2021.1011.tar.gz
# Decompress the downloaded httpd-2.4.54 compressed package.
tar -zvxf httpd-2.4.54.tar.gz
# Enter the decompressed httpd-2.4.54 folder and specify the path to the
compilation directory.
cd httpd-2.4.54/
./configure --prefix=/usr/local/httpd --enable-so --enable-ssl --enable-cgi --enable-
rewrite --enable-modules=most --enable-mpms-shared=all --with-mpm=prefork --
with-zlib --with-apr=/usr/local/apr --with-apr-util=/usr/local/apr-util --with-
ssl=/usr/local/gmssl LDFLAGS=-lm
# Edit build/config_vars.mk to change SSL to static linking
vi build/config_vars.mk
Locate ab_LIBS = -lssl -lcrypto -lrt -lcrypt -lpthread -ldl
Replace with: ab_LIBS = /usr/local/gmssl/lib/libssl.a /usr/local/gmssl/lib/libcrypto.a -
lssl -lcrypto -lrt -lcrypt -lpthread -ldl
# Compile and Install Apache
make install
```

Installation of National Cryptography Standard Certificate

1. The cloud.tencent.com certificate package has been downloaded and decompressed from the SSL Certificate Service Console to a local directory.

   After decompression, you can obtain relevant types of certificate files. This includes the Apache folder and the CSR file:

   - Folder name: `cloud.tencent.com_apache`
   - Folder content:
     - `cloud.tencent.com.key` private key file
     - `cloud.tencent.com._encrypt.crt` Encrypted certificate file

- `cloud.tencent.com._encrypt.key` Encryption certificate private key file
- `root_encrypt_bundle.crt` Encrypted Certificate Chain File
- `cloud.tencent.com._sign.crt` Signature certificate file
- `cloud.tencent.com._sign.key` Signature certificate private key file
- `root_sign_bundle.crt` Signature certificate chain file

- CSR file content:

  `cloud.tencent.com._sign.csr` file

> **Note:**
> The CSR file, either uploaded by you or generated online by the system during the certificate application, is provided to the CA. This file can be disregarded during installation.

2. Log in to the Apache server using "WinSCP", a tool for copying files between local and remote computers.

> **Note:**
> - For detailed directions, see Uploading files via WinSCP to a Linux CVM from Windows.
> - We recommend using the file upload feature of the Cloud Virtual Machine (CVM) for deployment to Tencent Cloud. For more details, please refer to Uploading Files to CVM.

3. Navigate to the /usr/local/httpd/conf directory and create a new directory named cert.

```
cd /usr/local/httpd/conf/
mkdir cert
```

Copy the obtained private key file `cloud.tencent.com.key`, encrypted certificate file `cloud.tencent.com._encrypt.crt`, encrypted certificate private key file `cloud.tencent.com._encrypt.key`, encrypted certificate chain file `root_encrypt_bundle.crt`, signature certificate file `cloud.tencent.com._sign.crt`, signature certificate private key file `cloud.tencent.com._sign.key`, and signature certificate chain file `root_sign_bundle.crt` from the local directory to the `/usr/local/httpd/conf/cert` directory on the Apache server.

4. Navigate to the `/usr/local/httpd/conf` directory and edit the `httpd.conf` file following these steps:

   4.1 Please add `ServerName (your domain):80` under `#ServerName www.example.com:80`.

4.2 Please remove the `#` before `LoadModule ssl_module modules/mod_ssl.so`.

4.3 Please remove the `#` before `Include conf/extra/httpd-ssl.conf`.

5. Edit the `/usr/local/httpd/conf/extra/httpd-ssl.conf` file and make the following modifications:

```
# Edit File
vi /usr/local/httpd/conf/extra/httpd-ssl.conf
```

5.1 Comment out all configuration lines containing `SSLSessionCache`, that is, prepend them with `#`.

5.2 Comment out the default certificate and key by adding a pound sign (#) at the beginning, as shown below:

```
# SSLCertificateFile "/usr/local/httpd/conf/server.crt"
# SSLCertificateKeyFile "/usr/local/httpd/conf/server.key"
```

5.3 Add the following configuration content to the file:

```
# Configuring the Algorithm
SSLCipherSuite HIGH:ECC-SM4-SM3:ECDHE-SM4-SM3
# Signing Certificate/Private Key/Certificate Chain
SSLCertificateFile "/usr/local/httpd/conf/cert/cloud.tencent.com._sign.crt"
SSLCertificateKeyFile "/usr/local/httpd/conf/cert/cloud.tencent.com._sign.key"
SSLCertificateChainFile "/usr/local/httpd/conf/cert/root_sign_bundle.crt"
# Encryption Certificate/Private Key/Certificate Chain
SSLCertificateFile "/usr/local/httpd/conf/cert/cloud.tencent.com._encrypt.crt"
SSLCertificateKeyFile
"/usr/local/httpd/conf/cert/cloud.tencent.com._encrypt.key"
SSLCertificateChainFile "/usr/local/httpd/conf/cert/root_encrypt_bundle.crt"
```

6. Configuring the national encryption standard license permission.

   Please download the national encryption license file (filename: `gmssl_xxxx.lic`) from the SSL Certificate Service Console DNSPod certificate details page to your local directory, and then upload it to the `/etc` directory on the Apache server.

7. You can verify the configuration file issues by executing the following command.

```
/usr/local/httpd/bin/httpd -t
```

   - If "Syntax OK" is displayed, it indicates that the configuration is correct and the Apache

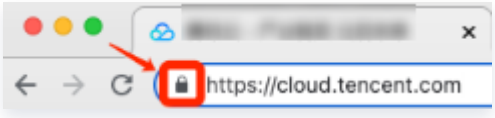
service can be started.

- If the prompt is not 'Syntax OK', please reconfigure or modify according to the prompt to resolve the issue.

8. Execute the following command to start the Apache service, then you can access it through `https://cloud.tencent.com` .

```
/usr/local/httpd/bin/httpd -k start
```

- If the security lock icon is displayed in the browser address bar, it indicates that the certificate has been successfully installed, as shown in the following figure:



- In case of a website access exception, troubleshoot the issue by referring to the following FAQs:
  - Unable to access the website via HTTPS
  - After deploying the SSL certificate, the browser indicates "Website connection is not secure"
  - Is your site indicating an insecure connection?
  - What should I do if I am prompted that HTTPS is not secure after reapplying for deployment upon expiration of the SSL certificate?
  - After deploying the SSL certificate on the server, a 404 error occurs when accessing resources

Optional dual installation of International Standard Certificate and National Cryptography Standard Certificate

If you need to resolve browser compatibility issues by installing both an international standard certificate and a National Cryptography Standard certificate, you can do so by following these steps:

> **Note:**
> Tencent Cloud offers free DV SSL certificates to users who have purchased the DNSPod certificate compliant with the national encryption standard, effectively resolving browser compatibility issues. For certificate application, please refer to Free DV SSL Certificate .

1. Utilize WinSCP (a tool for copying files between a local and a remote computer), to transfer

the obtained international standard certificate files `1_root_bundle.crt`, `2_cloud.tencent.com.crt`, and the private key file `3_cloud.tencent.com.key` from the local directory to the `/usr/local/httpd/conf/cert` directory on the Apache server.

> **Note:**
> - For detailed directions, see Uploading files via WinSCP to a Linux CVM from Windows.
> - We recommend using the file upload feature of the Cloud Virtual Machine (CVM) for deployment to Tencent Cloud. For more details, please refer to Uploading Files to CVM.

2. Edit the `httpd-ssl.conf` file located in the `/usr/local/httpd/conf/extra/` directory and add the following content:

```
SSLCertificateFile /usr/local/httpd/conf/cert/2_cloud.tencent.com.crt
SSLCertificateKeyFile /usr/local/httpd/conf/cert/3_cloud.tencent.com.key
SSLCertificateChainFile /usr/local/httpd/conf/cert/1_root_bundle.crt
```

3. Execute the following command to validate configuration file issues.

```
/usr/local/httpd/bin/httpd -t
```

   - If the message "Syntax OK" appears, it indicates that the configuration is correct and the Apache server can be launched.
   - If the prompt is not 'Syntax OK', please reconfigure or modify according to the prompt to resolve the issue.

4. Restarting the Apache service can resolve browser compatibility issues.

```
/usr/local/httpd/bin/httpd -k restart
```

> **Note:**
> If anything goes wrong during this process, please contact us.

FAQs

Last updated: 2023-10-08 16:06:55

Potential issues encountered during certificate installation, refer to the following for specific problems.

- How to resolve the issue when prompted with “Call to WAF failed: The domain name has not yet been ICP filed” during one-click HTTPS?
- How Do I Enable Port 443 for a VM?
- Why am I receiving a 'Connection Not Secure' warning when accessing the site?
- How to Configure the TLS Protocol Version for an SSL Certificate?
- How to Configure an Apple ATS Certificate?