SSL Certificates

Certificate Installation

Product Documentation

Service Statement

This document is intended to provide users with general information about Tencent Cloud's products and services only and does not form part of Tencent Cloud's terms and conditions. Tencent Cloud's products or services are subject to change. Specific products and services and the standards applicable to them are exclusively provided for in Tencent Cloud's applicable terms and conditions.

Contents

Certificate Installation

Installing an SSL Certificate on a Tencent Cloud Service

Installing an SSL Certificate in CDN

Last updated：2025-06-03 11:42:18

Overview

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

Prerequisites

You have logged in to the SSL Certificate Service console and obtained the certificate successfully.

Directions

Note:

The domain should be already connected to CDN and in "deploying" or "enabled" status. You cannot deploy a certificate for a disabled domain. For detailed directions, see Adding Domain Names.

If CDN acceleration is enabled for COS or CI, certificates cannot be configured for the domain `.file.myqcloud.com` or `.image.myqcloud.com` by default.

Currently, certificates cannot be configured for SVN hosted origins.

1. Click the **Issued** tab, select the target certificate, and click **Certificate Details**.
2. On the **Certificate Details** page, click **Quick Deployment**.
3. In the **Select a deployment type** pop-up window, select **CDN** and click **OK**.
4. Go to the **Configure certificate** page in the CDN console, which displays the corresponding **domain name**, **certificate source**, and **certificate ID**.
5. Select the origin-pull protocol. You can select the origin-pull method for getting resources from the origin by the CDN node.

If **HTTP** Origin-pull is selected, the requests sent from users to CDN nodes support HTTPS/HTTP, and the requests sent from CDN nodes to the origin server all use HTTP.

If you have selected **Follow protocol** for origin-pull, the origin server must have a valid certificate deployed; otherwise, origin-pull may fail. When the deployment is complete, the requests sent from CND nodes to the origin server follows the same protocol as the requests sent from users to CDN nodes, using either HTTP or HTTPS.

If the HTTPS port on the domain name's origin server is modified to a port number other than 443, the configuration will fail.

Domain names connected with the COS origin or FTP origin only support using HTTP as the origin-pull method.

6. After the configuration is completed, you can view the configured domain and certificate on the **Certificate Management** page.

Installation of International Standard Certificates

Installing an SSL Certificate on an Nginx Server

Last updated：2024-03-06 17:38:42

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

Overview

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/1.18.0` version is used as an example.

The current server OS is CentOS 7. Detailed steps vary slightly by OS.

Before you install an SSL certificate, enable the default HTTPS port `443` on the Nginx server so 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.

Prerequisites

Install the remote file copy tool such as WinSCP. The latest official version is recommended.

We recommend that you use CVM's file upload feature for deployment to CVM.

Install the remote login tool such as PuTTY or Xshell. The latest official version is recommended.

Install the Nginx service containing `http_ssl_module` module in the current server.

The data required to install the SSL certificate includes the following:

Name	Description
Server IP address	IP address of the server, 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.

Directions

Installing the certificate

1. Log in to the SSL Certificate Service console, and click **Download** for the certificate you need to install.
2. In the pop-up window, select **Nginx** for the server type, click **Download**, and decompress the `cloud.tencent.com` certificate file package to the local directory.

After decompression, you can get the certificate file of the corresponding type, which includes the `cloud.tencent.com_nginx` folder.

Folder: `cloud.tencent.com_nginx`

Files in the folder:

`cloud.tencent.com_bundle.crt` : Certificate file

`cloud.tencent.com_bundle.pem` : Certificate file (optional)

`cloud.tencent.com.key` : Private key file

`cloud.tencent.com.csr` : CSR file

Note:

You can upload the CSR file when applying for a certificate or have it generated online by the system. It is provided to the CA and irrelevant to the installation.

3. Log in to the Nginx server using WinSCP (a tool copying files between a local computer and a remote computer).

Note:

For detailed directions, see Uploading files via WinSCP to a Linux CVM from Windows.

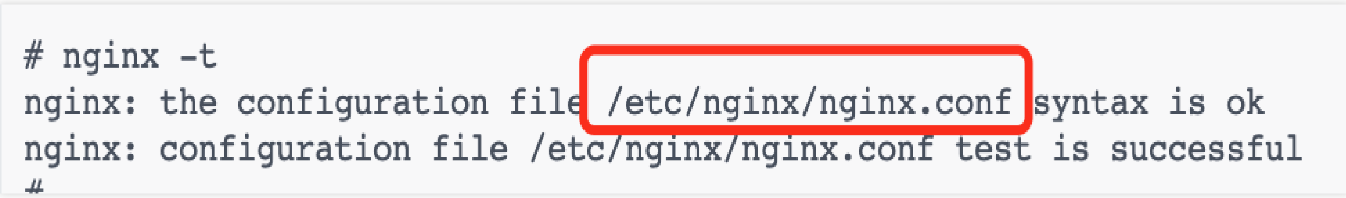
We recommend that you use CVM's file upload feature for deployment to CVM.

4. Copy the `cloud.tencent.com_bundle.crt` certificate file and `cloud.tencent.com.key` private key file from the local directory to the `/etc/nginx` directory (this is the default Nginx installation directory and needs to be adjusted as needed) of the Nginx server.
5. Log in to the Nginx server remotely with such a login tool as PuTTY.
6. Edit the `nginx.conf` configuration file in the Nginx root directory as follows:

Note:

If you cannot find the following content, manually add it. Run the `nginx -t` command to find the path of the Nginx configuration file.

As shown below:



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

This operation can edit the file by running `vim /etc/nginx/nginx.conf` .

The configuration file may be written differently on different versions; for example, use `listen 443 ssl` instead of `listen 443` and `ssl on` on `nginx/1.15.0` or later.

```
server {
     # The default SSL access port is 443
     listen 443 ssl;
     # Enter the domain name bound to the certificate
     server_name cloud.tencent.com;
     # Enter the relative or absolute path of the certificate file
     ssl_certificate cloud.tencent.com_bundle.crt;
     # Enter the relative or absolute path of the private key file
     ssl_certificate_key cloud.tencent.com.key;
     ssl_session_timeout 5m;
     # Configure the following protocols
     ssl_protocols TLSv1.2 TLSv1.3;
     # Configure the cipher suite according to the OpenSSL standard
     ssl_ciphers ECDHE-RSA-AES128-GCM-SHA256:HIGH:!aNULL:!MD5:!RC4:!DHE;
     ssl_prefer_server_ciphers on;
     location / {
         # Path to the website homepage. This example is for reference only.
You need to set it to the actual path.
         # For example, if your website homepage is under the "/etc/www" path
of the Nginx server, change the "html" behind "root" to "/etc/www".
         root html;
         index  index.html index.htm;
     }
 }
```

7. Run the following command to check whether there is a problem with the configuration file.

```
nginx -t
```

If yes, reconfigure or fix the problem as prompted.

If not, proceed to step 8.

8. Run the following command to reload the Nginx server.

```
nginx -s reload
```

9. If the server is reloaded successfully, you can access it through `https://cloud.tencent.com` .

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

To redirect HTTP requests to HTTPS, complete the following settings:

1. Select one of the following configuration methods based on your actual needs:

Add a JavaScript script to the page.

Add redirect in the backend program.

Redirect through a web server.

Nginx supports rewrite. If you did not remove PCRE during the compilation, you can add `return 301 https://$host$request_uri;` to the HTTP server to redirect requests made to the default HTTP port 80 to HTTPS.

Note:

Uncommented configuration statements can be configured as follows.

The configuration file may be written differently on different versions; for example, use `listen 443 ssl` instead of `listen 443` and `ssl on` on `nginx/1.15.0` or later.

```
  server {
   # The default SSL access port is 443
   listen 443 ssl;
   # Enter the domain name bound to the certificate
   server_name cloud.tencent.com;
   # Enter the relative or absolute path of the certificate file
   ssl_certificate  cloud.tencent.com_bundle.crt;
   # Enter the relative or absolute path of the private key file
   ssl_certificate_key cloud.tencent.com.key;
   ssl_session_timeout 5m;
   # Configure the cipher suite according to the OpenSSL standard
   ssl_ciphers ECDHE-RSA-AES128-GCM-
SHA256:ECDHE:ECDH:AES:HIGH:!NULL:!aNULL:!MD5:!ADH:!RC4;
   # Configure the following protocols
   ssl_protocols TLSv1.2 TLSv1.3;
   ssl_prefer_server_ciphers on;
   location / {
     # Path to the website homepage. This example is for reference only. You
need to set it to the actual path.
     # For example, if your website homepage is under the "/etc/www" path of
the Nginx server, change the "html" behind "root" to "/etc/www".
     root html;
     index index.html index.htm;
   }
  }
  server {
   listen 80;
   # Enter the domain name bound to the certificate
   server_name cloud.tencent.com;
   # Redirect requests made to an HTTP domain name to HTTPS
   return 301 https://$host$request_uri;
  }
```

2. Run the following command to check whether there is a problem with the configuration file.

```
nginx -t
```

If yes, reconfigure or fix the problem as prompted.

If not, proceed to step 3.

3. Run the following command to reload the Nginx server.

```
nginx -s reload
```

4. If the server is reloaded successfully, 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.

In case of a website access exception, troubleshoot the issue by referring to the following FAQs:

Website Inaccessible After an SSL Certificate is Deployed

"Your Connection is Not Secure" is Displayed After the SSL Certificate is Installed

Why Does the Website Prompt "Connection Is Untrusted"?

404 Error After the SSL Certificate is Deployed on IIS

Note:

If anything goes wrong during this process, contact us.

Installing an SSL Certificate on an Apache Server (Linux)

Last updated：2024-03-06 17:38:42

Overview

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/2.4.6` version is used as an example. The default port is `80`. You can download it from the Apache official website. If you need to use another version, contact us.

The current server OS is CentOS 7. Detailed steps vary slightly by OS.

Before you install an SSL certificate, enable port `443` on the Apache server so 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.

Prerequisites

A remote file copy tool such as WinSCP has been installed. Please download the latest version from the official website.

We recommend that you use CVM's file upload feature for deployment to CVM.

Install the remote login tool such as PuTTY or Xshell. The latest official version is recommended.

Install the Apache service on the current server.

The data required to install the SSL certificate includes the following:

Name	Description
Server IP address	IP address of the server, 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.

Directions

Installing the certificate

1. Log in to the SSL Certificate Service console, and click **Download** for the certificate you need to install.

2. In the pop-up window, select **Apache** for the server type, click **Download**, and decompress the `cloud.tencent.com` certificate file package to the local directory.

After decompression, you can get the certificate file of the corresponding type, which includes the `cloud.tencent.com_apache` folder.

Folder: `cloud.tencent.com_apache`

Files in the folder:

`root_bundle.crt` : certificate file

`cloud.tencent.com.crt` : certificate file

`cloud.tencent.com.key` : Private key file

CSR file: `cloud.tencent.com.csr` file

Note:

You can upload the CSR file when applying for a certificate or have it generated online by the system. It is provided to the CA and irrelevant to the installation.

3. Log in to the Apache server using WinSCP (a tool copying files between a local computer and a remote computer).

Note:

For detailed directions, see Uploading files via WinSCP to a Linux CVM from Windows.

We recommend that you use CVM's file upload feature for deployment to CVM.

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

Note:

If the `/etc/httpd/ssl` directory does not exist, run the `mkdir /etc/httpd/ssl` command to create it.

5. Log in to the Apache server remotely by using a login tool such as PuTTY.

Note:

For a newly installed Apache server, the `conf.d` , `conf` , and `conf.modules.d` directories are under the `/etc/httpd` directory by default.

6. In the `httpd.conf` configuration file under `/etc/httpd/conf` , find the `Include conf.modules.d/*.conf` configuration statement (for loading the SSL configuration directory) and check whether it is commented out. If so, remove the comment symbol (`#`) from the first line and save the configuration file.

7. In the `00-ssl.conf` configuration file under `/etc/httpd/conf.modules.d` , find the `LoadModule ssl_module modules/mod_ssl.so` configuration statement (for loading the SSL module) and check whether it is commented out. If so, remove the comment symbol (`#`) from the first line and save the configuration file.

Note:

The directory structure varies by OS version. Find files in accordance with the actual OS version.

If `LoadModule ssl_module modules/mod_ssl.so` and `Include conf.modules.d/*.conf` configuration statements cannot be found in the configuration files above, check whether the `mod_ssl.so` module has been installed, and if not, run the `yum install mod_ssl` command to install it.

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

```
<VirtualHost 0.0.0.0:443>
     DocumentRoot "/var/www/html"
     # Enter the certificate name
     ServerName cloud.tencent.com
     # Enable SSL
     SSLEngine on
     # Path of the certificate file
     SSLCertificateFile /etc/httpd/ssl/cloud.tencent.com.crt
     # Path of the private key file
     SSLCertificateKeyFile /etc/httpd/ssl/cloud.tencent.com.key
     # Path of 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.

In case of a website access exception, troubleshoot the issue by referring to the following FAQs:

Website Inaccessible After an SSL Certificate is Deployed

"Your Connection is Not Secure" is Displayed After the SSL Certificate is Installed

Why Does the Website Prompt "Connection Is Untrusted"?

404 Error After the SSL Certificate is Deployed on IIS

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

To redirect HTTP requests to HTTPS, complete the following settings:

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

Note:

The directory structure varies by Apache version. For more information, see Apache Module mod_rewrite.

The `httpd.conf` configuration file is located in more than one directory. You can filter them by `/etc/httpd/*` .

2. Check whether `LoadModule rewrite_module modules/mod_rewrite.so` is in it.

If so, remove the comment symbol (`#`) in front of `LoadModule rewrite_module modules/mod_rewrite.so` and proceed to step 4.

If not, proceed to step 3.

3. C

reat
e a `*.conf` file such as `00-rewrite.conf` in `/etc/httpd/conf.modules.d` and add the following content to it:

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

4. Ad
d the follo
wing to the `httpd.conf` configuration file:

```
<Directory "/var/www/html">
# Add the following:
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, contact us.

Installing an SSL Certificate on an Apache Server (Windows)

Last updated：2024-03-06 17:38:42

Overview

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/2.4.53` version is used as an example. The default port is `80` . You can download it from the Apache official website. If you need to use another version, contact us.

The current server OS is Windows Server 2012 R2. Detailed steps vary slightly by OS.

Before you install an SSL certificate, enable port `443` on the Apache server so 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.

Prerequisites

Install the Apache service on the current server.

The data required to install the SSL certificate includes the following:

Name	Description
Server IP address	IP address of the server, 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.

Directions

Step 1. Upload the certificate file

1. Log in to the SSL Certificate Service console, and click **Download** for the certificate you need to install.
2. In the pop-up window, select **Apache** for the server type, click **Download**, and decompress the `cloud.tencent.com` certificate file package to the local directory.

After decompression, you can get the certificate file of the corresponding type, which includes the `cloud.tencent.com_apache` file.

Folder: `cloud.tencent.com_apache`

Files in the folder:

`root_bundle.crt` : certificate file

`cloud.tencent.com.crt` : certificate file

`cloud.tencent.com.key` : Private key file

`cloud.tencent.com.csr` : CSR file

Note:

You can upload the CSR file when applying for a certificate or have it generated online by the system. It is provided to the CA and irrelevant to the installation.

3. Log in to the Apache server via the RDP port.

Note:

For detailed directions, see Uploading Files from Linux to Windows CVM using RDP.

We recommend that you use CVM's file upload feature for deployment to CVM.

4. Copy the `root_bundle.crt` certificate file, `cloud.tencent.com.crt` certificate file, and `cloud.tencent.com.key` private key file from the local directory to the `ssl.crt` and `ssl.key` folders under the `\\conf` directory of the Apache server, respectively.

SSL Certificate File	Folder
root_bundle.crt	ssl.crt
cloud.tencent.com.crt	
cloud.tencent.com.key	ssl.key

Step 2. Configure the file

1. Open the `httpd.conf` file in the `conf` directory of the Apache server with a text editor and delete the `#` before the following fields.

```
#LoadModule ssl_module modules/mod_ssl.so
#Include conf/extra/httpd-ssl.conf
```

2. Open the `httpd-ssl.conf` file in the `conf\\extra` directory of the Apache server with a text editor.

3. Modify the `httpd-ssl.conf` file and set the following field parameters to the paths of the uploaded certificate files as shown below:

```
SSLCertificateFile "C:/apache/conf/ssl.crt/cloud.tencent.com.crt"
SSLCertificateKeyFile "C:/apache/conf/ssl.key/cloud.tencent.com.key"
SSLCACertificateFile "C:/apache/conf/ssl.crt/root_bundle.crt"
```

4. 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.

In case of a website access exception, troubleshoot the issue by referring to the following FAQs:

Website Inaccessible After an SSL Certificate is Deployed

"Your Connection is Not Secure" is Displayed After the SSL Certificate is Installed

Why Does the Website Prompt "Connection Is Untrusted"?

404 Error After the SSL Certificate is Deployed on IIS

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

1. Open the `httpd.conf` file in the `conf` directory of the Apache server with a text editor and delete the `#` before the following fields.

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

2. Configure the fields in the website running directory. For example, add the following content to the `<Directory "C:/xampp/htdocs">` field:

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

3. Restart the Apache server and then you can access it through both `https://intl.cloud.tencent.com/` (which will be automatically redirected to `https://intl.cloud.tencent.com/`) and `https://intl.cloud.tencent.com/` .

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

Last updated：2024-03-06 17:42:38

Overview

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 you install an SSL certificate, enable port `443` on the Tomcat server so 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.

Prerequisites

Install the remote file copy tool such as WinSCP. The latest official version is recommended.

We recommend that you use CVM's file upload feature for deployment to CVM.

Install the remote login tool such as PuTTY or Xshell. The latest official version is recommended.

The Tomcat service has been installed and configured on the server.

The data required to install the SSL certificate includes the following:

Name	Description
Server IP address	IP address of the server, 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 **By pasting** method when applying for the SSL certificate, or your certificate brand is WoTrus, the option to download the JKS certificate file is not provided. Instead, you need to manually convert the format to generate a keystore as follows:

Access the conversion tool.

Upload the certificate and private key files in the Nginx folder to the conversion tool, enter the keystore password, click **Submit**, and convert the certificate to a `.jks` certificate.

Currently, the Tomcat service is installed in the `/usr` directory by default. For example, if the Tomcat folder is `Tomcat-9.0.56`, the configuration file directory will be `/usr/Tomcat-9.0.56/conf`.

If you have selected the **Paste CSR** method when applying for the SSL certificate, or your certificate brand is Wotrus, the option to download the JKS certificate file is not provided. Instead, you need to manually convert the format to generate a keystore as follows:

Access the conversion tool.

Upload the certificate and private key files in the Nginx folder to the conversion tool, enter the keystore password, click **Submit**, and convert the certificate to a .jks certificate.

Currently, the Tomcat service is installed in the `/usr` directory by default. For example, if the Tomcat folder is `Tomcat-9.0.56`, the configuration file directory will be `/usr/Tomcat-9.0.56/conf`.

Directions

Installing the certificate

1. Log in to the SSL Certificate Service console, and click **Download** for the certificate you need to install.
2. In the pop-up window, select **JKS** for the server type, click **Download**, and decompress the `cloud.tencent.com` certificate file package to the local directory.

After decompression, you can get the certificate file of the corresponding type, which includes the `cloud.tencent.com_jks` folder.

Folder: `cloud.tencent.com_jks`

Files in the folder:

`cloud.tencent.com.jks`: keystore file

`keystorePass.txt`: password file (if you have set a private key password, this file will not be generated)

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

Note:

For detailed directions, see Uploading files via WinSCP to a Linux CVM from Windows.

We recommend that you use CVM's file upload feature for deployment to CVM.

4. Add the following content to the `server.xml` file in the `/usr/Tomcat-9.0.56/conf` directory:

```
<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 (Tomcat installation directory) environment variable. If this parameter is not set, Tomcat will read the file named `.keystore` from the user directory of the current OS user.

keystorePass: keystore password. If you set a private key password when applying for the certificate, enter the private key password; otherwise, enter the password in the `keystorePass.txt` file in the Tomcat folder.

clientAuth: If it is set to true, Tomcat requires all SSL clients to provide a security certificate for identity verification.

For details about the `server.xml` file, see below:

Note:

Do not copy the content of the `server.xml` file; otherwise, the format will be incorrect.

```
  <?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="
   <Listener className="org.apache.catalina.core.JreMemoryLeakPreventionListener"
   <Listener className="org.apache.catalina.mbeans.GlobalResourcesLifecycleListene
   <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"  redirec
       <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.j
                 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="log
           prefix="localhost_access_log" suffix=".txt"
           pattern="%h %l %u %t &quot;%r&quot; %s %b" />
      </Host>
    </Engine>
  </Service>
</Server>
```

5. Check whether the Tomcat server is started.

If so, you need to run the following commands to shut down and restart the Tomcat service in the `bin` directory (for example, `/usr/Tomcat-9.0.56/bin`) of the Tomcat installation directory.

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

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

```
./startup.sh
```

6. If the server is started successfully, 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.

In case of a website access exception, troubleshoot the issue by referring to the following FAQs:

Website Inaccessible After an SSL Certificate is Deployed

"Your Connection is Not Secure" is Displayed After the SSL Certificate is Installed

Why Does the Website Prompt "Connection Is Untrusted"?

404 Error After the SSL Certificate is Deployed on IIS

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

To redirect HTTP requests to HTTPS, complete the following settings:

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

2. Insert a new line after `</welcome-file-list>` and add the following:

```
<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. Edit the `server.xml` file in the `conf` directory (for example, `/usr/Tomcat-9.0.56/conf`) of the Tomcat installation directory by changing the `redirectPort` parameter to the port of the SSL connector, i.e., port `443`, as shown below:

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

Note:

This change allows a non-SSL connector to redirect to an SSL connector.

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

```
./shutdown.sh
```

5. Run the following command to confirm whether there is a problem with the configuration:

```
./configtest.sh
```

If yes, reconfigure or fix the problem as prompted.

If no, proceed to the next step.

6. Run the following command to start the Tomcat service. In this way, you can access it through `http://cloud.tencent.com`.

```
./startup.sh
```

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

Last updated：2024-03-06 17:38:42

Overview

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.

Tomcat 9.0.56 is used as an example.

The current server OS is Windows Server 2016 Chinese. Detailed steps vary slightly with the OS.

Before you install an SSL certificate, enable port 443 on the Tomcat server so that HTTPS can be enabled after the certificate is installed. For more information, please see How Do I Enable Port 443 for a VM?

For more information about how to upload SSL certificate files to a server, please see Copying Local Files to CVMs.

Prerequisites

The Tomcat service has been installed and configured on the server.

The data required to install the SSL certificate includes the following:

Item	Description
Server IP address	IP address of the server, 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 obtain the server IP address, username, and password.

If you have selected the **Paste CSR** method when applying for the SSL certificate, or your certificate brand is Wotrus, the option to download the JKS certificate file is not provided. Instead, you need to manually convert the format to generate a keystore as follows:

Access the conversion tool.

Upload the certificate and private key files in the Nginx folder to the conversion tool, enter the keystore password, click **Submit**, and convert the certificate to a .jks certificate.

Directions

Certificate Installation

1. Log in to the SSL Certificate Service console, and click **Download** for the certificate you need to install.
2. In the pop-up window, select **JKS** for the server type, click **Download**, and decompress the `cloud.tencent.com` certificate file package to the local directory.

After decompression, you can get the certificate file of the corresponding type, which includes the `cloud.tencent.com_jks` folder.

Folder: `cloud.tencent.com_jks`

Files in the folder:

`cloud.tencent.com.jks` : keystore file

`keystorePass.txt` : password file (if you have set a private key password, this file will not be generated)

3. Copy the keystore file `cloud.tencent.com.jks` to the `conf` directory of the Tomcat installation directory.
4. Edit the `server.xml` file in the `conf` directory by adding the following:

```
<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 details about the `server.xml` file, see below:

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"  redirectPo
     <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 &quot;%r&quot; %s %b" />
   </Host>
 </Engine>
  </Service>
</Server>
```

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

keystoreFile: 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 parameter is not set, Tomcat reads the file named ".keystore" from the user directory of the current OS user.

keystorePass: keystore password. If you set a private key password when applying for the certificate, enter the private key password; otherwise, enter the password in the `keystorePass.txt` file in the Tomcat folder.

clientAuth: If it is set to true, Tomcat requires all SSL clients to provide a security certificate for identity verification.

5. Confirm whether the Tomcat server is started.

If the Tomcat server is already started, you need to run the following .bat scripts in sequence in the `bin` directory of the Tomcat installation directory to shut down and restart it:

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

If the Tomcat server is not started, you need to run the following .bat script in the `bin` directory of the Tomcat installation directory to start it:

```
startup.bat
```

6. If the server is started successfully, you can access it through ``https://intl.cloud.tencent.com/`.

If the browser address bar displays the security lock logo, it means that the certificate is installed successfully.

If the website access is abnormal, you can refer to the following solutions to common problems:

Website Inaccessible After an SSL Certificate is Deployed

“Your Connection is Not Secure” is Displayed After the SSL Certificate is Installed”

Why Does the Website Prompt “Connection Is Untrusted"?

404 Error After the SSL Certificate is Deployed on IIS

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

You can redirect HTTP requests to HTTPS by configuring the following settings:

1. Edit the `web.xml` file in the `conf` directory of the Tomcat installation directory and find the `<\\/welcome-file-list>` tag.
2. Insert a new line after `<\\/welcome-file-list>` and add the following:

```
<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. Edit the `server.xml` file in the Tomcat installation directory by changing the `redirectPort` parameter to the port of the SSL connector, i.e., port 443, as shown below:

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

Note:

This change allows a non-SSL connector to redirect to an SSL connector.

4. Run the following .bat script in the `/bin` directory of the Tomcat installation directory to shut down the Tomcat server:

```
shutdown.bat
```

5. Run the following command to confirm whether there is a problem with the configuration:

```
configtest.bat
```

If yes, reconfigure or fix the problem as prompted.

If no, proceed to the next step.

6. Run the following .bat script to start the Tomcat server. In this way, you can access it through `https://intl.cloud.tencent.com/` .

```
startup.bat
```

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

Last updated：2024-03-06 17:38:42

Overview

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.

If you need to install an SSL certificate (JKS format) on a Tomcat server, see Installing an SSL Certificate (JKS Format) on a Tomcat Server.

Before you install an SSL certificate, enable port 443 on the Tomcat server so that HTTPS can be enabled after the certificate is installed. For more information, please see How Do I Enable Port 443 for a VM?

For more information about how to upload SSL certificate files to a server, see Copying Local Files to CVMs.

Prerequisites

A remote file copy tool such as WinSCP has been installed. Please download the latest version from the official website.

A remote login tool such as PuTTY or Xshell has been installed. Please download the latest version from the official website.

The Tomcat service has been installed and configured on the server.

The data required to install the SSL certificate includes the following:

Item	Description
Server IP address	IP address of the server, 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 obtain the server IP address, username, and password.
Currently, the Tomcat server is installed in the `/usr` directory. For example, if the Tomcat folder name is `tomcat9.0.40`, `/usr/*/conf` is actually `/usr/tomcat9.0.40/conf`.

Directions

Certificate Installation

1. Log in to the SSL Certificate Service console, and click **Download** for the certificate you need to install.
2. In the pop-up window, select **Tomcat** for the server type, click **Download**, and decompress the `cloud.tencent.com` certificate file package to the local directory.
After decompression, you can get the certificate file of the corresponding type, which includes the `cloud.tencent.com_tomcat` folder.
Folder: `cloud.tencent.com_tomcat`
Files in the folder:
`cloud.tencent.com.pfx`: certificate file
`keystorePass.txt`: password file (if you have set a private key password, this file will not be generated)
3. Log in to the Tomcat server using WinSCP (a tool copying files between a local computer and a remote computer).
4. Copy the obtained `cloud.tencent.com.pfx` certificate file from the local directory to the `/usr/*/conf` directory.
5. Remotely log in to the Tomcat server using a login tool such as PuTTY.
6. Edit the `server.xml` file in the `/usr/*/conf` directory by using either of the following methods as needed:
Note:
If you use method 1, Tomcat automatically selects an SSL implementation mode for you. If you are unable to complete the subsequent configuration according to method 1, it may be because your environment does not support the implementation mode. In that case, you can use method 2 to manually select an SSL implementation mode based on your environment properties.
Method 1: automatically selecting an SSL implementation mode
Method 2: manually selecting an SSL implementation mode
Modify the value of the `Connector` attribute in the `server.xml` file to the following:

```
<Connector port="443"
protocol="HTTP/1.1"
    SSLEnabled="true"
    scheme="https"
    secure="true"
    keystoreFile="/usr/*/conf/cloud.tencent.com.pfx" # Path of the certificate file
    keystoreType="PKCS12"
    keystorePass="Certificate password"  # Replace the value 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,TLS_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_SHA256"/>
```

Modify the value of the `Connector` attribute in the `server.xml` file to the following:

```
<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 of the certificate file, and replace `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: 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 parameter is not set, Tomcat reads the file named ".keystore" from the user directory of the current OS user.

keystorePass: password in the password file, i.e., keystore password. If you have set a private key password when applying for the certificate, enter the private key password; otherwise, enter the password in the `keystorePass.txt` file in the `cloud.tencent.com_tomcat` folder.

clientAuth: If it is set to true, Tomcat requires all SSL clients to provide a security certificate for identity verification.

7. Confirm whether the Tomcat server is started.

If the Tomcat server is already started, you need to run the following commands in sequence in the `/usr/*/bin` directory to shut down and restart it.

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

If the Tomcat server is not started, you need to run the following command in the `/usr/*/bin` directory to start it.

```
./startup.sh
```

8. If the server is started successfully, you can access it through `https://cloud.tencent.com` .

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

You can redirect HTTP requests to HTTPS by configuring the following settings:

1. Edit the `web.xml` file in the `/usr/*/conf` directory and find the `<\\/welcome-file-list>` tag.

2. Insert a new line after `<\\/welcome-file-list>` and add the following:

```
<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. Edit the `server.xml` file in the `/usr/*/conf` directory by changing the `redirectPort` parameter to the port of the SSL connector, i.e., port 443, as shown below:

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

Note:

This change allows a non-SSL connector to redirect to an SSL connector.

4. Shut down the Tomcat server by running the following command in the `/usr/*/bin` directory:

```
./shutdown.sh
```

5. Run the following command to confirm whether there is a problem with the configuration:

```
./configtest.sh
```

If yes, reconfigure or fix the problem as prompted.

If no, proceed to the next step.

6. Run the following command to start the Tomcat server. In this way, you can access it through `http://cloud.tencent.com`.

```
./startup.sh
```

Installing an SSL Certificate on a GlassFish Server

Last updated：2024-03-06 17:38:41

Overview

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

Description

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 by OS.

Before you install an SSL certificate, enable port `443` on the GlassFish server so 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.

Prerequisites

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

We recommend that you use CVM's file upload feature for deployment to CVM.

A remote login tool such as PuTTY or Xshell has been installed. Download the latest version from the official website.

The GlassFish service has been installed and configured on the current server.

The data required to install the SSL certificate includes:

Name	Description
Server IP address	IP address of the server, 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.

Notes

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 **By pasting** method when applying for the SSL certificate, or your certificate brand is WoTrus, the Tomcat option to download the `.pfx` or `.jks` certificate file is not provided. Instead, you need to manually convert the format to generate a keystore as follows:

Access the conversion tool.

Upload the certificate and private key files in the Nginx folder to the conversion tool, enter the keystore password, click **Submit**, and convert the certificate to a `.jks` certificate.

The GlassFish service is installed in the `/usr/share` directory.

Directions

1. Log in to the SSL Certificate Service console and click **Download** for the certificate you need to install.

2. In the pop-up window, select **Apache** and **JKS** for the server type, click **Download**, and decompress the `cloud.tencent.com` certificate file package to a local directory.

After decompression, you can get the certificate files of the corresponding types, which include the `cloud.tencent.com_apache` and `cloud.tencent.com_jks` folders.

Folder: `cloud.tencent.com_apache`

`cloud.tencent.com.crt` : Certificate file

`cloud.tencent.com.key` : Private key file

CSR file: `cloud.tencent.com.csr` file

Description

You can upload the CSR file when applying for a certificate or have it generated online by the system. It is provided to the CA and irrelevant to the installation.

3. Remotely log in to the GlassFish server.

4. Go to the `/usr/share/glassfish4/glassfish/bin` directory, run the `./asadmin` command, and run the `change-master-password --savemasterpassword=true domain1` command to change the domain administrator password as shown below:

Notes

The default installation directory of the `domain1` service is `/usr/share/glassfish4/glassfish/domains` . Enter the domain according to the actual situation.

The default password is `changeit` . Press **Enter** and enter the new password, which should be the **private key password** you 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. In the `/usr/share` directory, run the `mkdir temp` command to create the `temp` folder.

6. Use WinSCP (a tool for copying files between a local computer and a remote computer) to log in to the GlassFish server. Then, copy the certificate file `cloud.tencent.com.crt` and the private key file `cloud.tencent.com.key` from the local directory to the `temp` folder.

Description

For detailed directions, see Uploading files via WinSCP to a Linux CVM from Windows.

We recommend that you use CVM's file upload feature for deployment to CVM.

7. In the `temp` folder, run the following command to generate the `PKCS12` file. When the system prompts you for a password during the process, enter the new password, which is the private key password.

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

8. In the `temp` folder, run the `ls -l` command to check whether the `PKCS12` file contains the certificate you applied for.

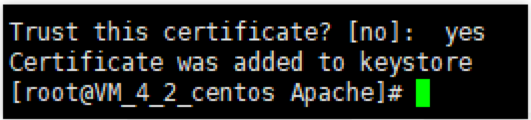
9. In the `temp` folder, run the following command to generate the `keystore.jks` file:

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

10. In the `temp` folder, run the following command to generate the `cacert.jks` file. When the system prompts you for a password during this process, enter the new password, which is the private key password.

```
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.



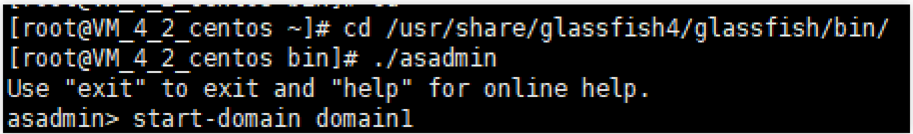
```
Trust this certificate? [no]:  yes
Certificate was added to keystore
[root@VM_4_2_centos Apache]# 
```

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, change the port numbers in the `domain.xml` file.

```
<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.

In case of a website access exception, troubleshoot the issue by referring to the following FAQs:

Website Inaccessible After an SSL Certificate is Deployed

"Your Connection is Not Secure" is Displayed After the SSL Certificate is Installed

Why Does the Website Prompt "Connection Is Untrusted"?

What Should I Do If I Am Prompted That HTTPS Is Not Secure After Reapplying for Deployment upon Expiration of the SSL Certificate?

404 Error After the SSL Certificate is Deployed on IIS

Notes

If anything goes wrong during this process, contact us.

Installing an SSL Certificate on a JBoss Server

Last updated：2024-03-06 17:38:42

Overview

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 by OS.

Before you install an SSL certificate, enable port `443` on the JBoss server so 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.

Prerequisites

Install the remote file copy tool such as WinSCP. The latest official version is recommended.

We recommend that you use CVM's file upload feature for deployment to CVM.

Install the remote login tool such as PuTTY or Xshell. The latest official version is recommended.

The JBoss service has been installed and configured on the current server.

The data required to install the SSL certificate includes the following:

Name	Description
Server IP address	IP address of the server, 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 **By pasting** method when applying for the SSL certificate, or your certificate brand is WoTrus, the option to download the JKS certificate file is not provided. Instead, you need to manually convert the format to generate a keystore as follows:

Access the conversion tool.

Upload the certificate and private key files in the Nginx folder to the conversion tool, enter the keystore password, click **Submit**, and convert the certificate to a `.jks` certificate.

The JBoss service is installed in the `/usr/local` directory.

If you have selected the **Paste CSR** method when applying for the SSL certificate, or your certificate brand is Wotrus, the option to download the JKS certificate file is not provided. Instead, you need to manually convert the format to generate a keystore as follows:

Access the conversion tool.

Upload the certificate and private key files in the Nginx folder to the conversion tool, enter the keystore password, click **Submit**, and convert the certificate to a .jks certificate.

The JBoss service is installed in the `/usr/local` directory.

Directions

1. Log in to the SSL Certificate Service console, and click **Download** for the certificate you need to install.
2. In the pop-up window, select **JKS** for the server type, click **Download**, and decompress the `cloud.tencent.com` certificate file package to the local directory.

After decompression, you can get the certificate file of the corresponding type, which includes the `cloud.tencent.com_jks` folder.

Folder: `cloud.tencent.com_jks`

Files in the folder:

`cloud.tencent.com.jks` : keystore file

`keystorePass.txt` : password file (if you have set a private key password, this file will not be generated)

3. Remotely log in to the JBoss server. For example, you can use PuTTY for remote login.
4. In the `/usr/local/jboss-7.1.1/standalone/configuration` directory, run the `mkdir cert` command to create the `cert` folder.
5. Use WinSCP (a tool for copying files between a local computer and a remote computer) to log in to the JBoss server and copy the 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 that you use CVM's file upload feature for deployment to CVM.

6. In the `/usr/local/jboss-7.1.1/standalone/configuration` directory, change the port configuration and add certificate configuration in the `standalone.xml` file.

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="${jboss.bind.address.unsecure:127.0.0.1}"/>
     </interface>
 </interfaces>
 <socket-binding-group name="standard-sockets" default-interface="public" port-offs
     <socket-binding name="management-native" interface="management" port="${jboss.
     <socket-binding name="management-http" interface="management" port="${jboss.ma
     <socket-binding name="management-https" interface="management" port="${jboss.m
     <socket-binding name="ajp" port="8009"/>
             <!--Change the HTTP port-->
     <socket-binding name="http" port="80"/>
             <!--Change the 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:

Enabling remote access: change `${jboss.bind.address:127.0.0.1}` to `${jboss.bind.address:0.0.0.0}` .

Changing the HTTP port: change port 8080 to 80.

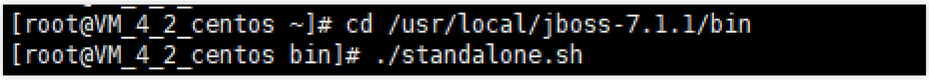
Changing the HTTPS port: change port 8443 to 443.

Part 2: adding certificate configuration

```
<subsystem xmlns="urn:jboss:domain:web:1.1" default-virtual-server="default-host" n
         <connector name="http" protocol="HTTP/1.1" scheme="http" socket-binding="h
         <connector name="https" protocol="HTTP/1.1" scheme="https" socket-binding=
             <ssl name="https" password="******" certificate-key-file="../standalon
         </connector>
         <virtual-server name="default-host" enable-welcome-root="true">
             <alias name="localhost"/>
```

```
                <alias name="example.com"/>
            </virtual-server>
        </subsystem>
```

7. Go to the `/usr/local/jboss-7.1.1/bin` directory and run the `./standalone.sh` command to start the JBoss server.



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

8. The certificate is deployed and you can access the website through `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:

Website Inaccessible After an SSL Certificate is Deployed

"Your Connection is Not Secure" is Displayed After the SSL Certificate is Installed

Why Does the Website Prompt "Connection Is Untrusted"?

404 Error After the SSL Certificate is Deployed on IIS

Note:

If anything goes wrong during this process, contact us.

Installing an SSL Certificate on a Jetty Server

Last updated：2024-03-06 17:38:42

Overview

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 by OS.

Before you install an SSL certificate, enable port `443` on the Jetty server so 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.

Prerequisites

Install the remote file copy tool such as WinSCP. The latest official version is recommended.

We recommend that you use CVM's file upload feature for deployment to CVM.

Install the remote login tool such as PuTTY or Xshell. The latest official version is recommended.

The Jetty service has been installed and configured on the current server.

The data required to install the SSL certificate includes the following:

Name	Description
Server IP address	IP address of the server, 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 **By pasting** method when applying for the SSL certificate, or your certificate brand is WoTrus, the option to download the JKS certificate file is not provided. Instead, you need to manually convert the format to generate a keystore as follows:

Access the conversion tool.

Upload the certificate and private key files in the Nginx folder to the conversion tool, enter the keystore password, click **Submit**, and convert the certificate to a `.jks` certificate.

The Jetty service is installed in the `/usr/local/jetty` directory.

Directions

1. Log in to the SSL Certificate Service console, and click **Download** for the certificate you need to install.

2. In the pop-up window, select **JKS** for the server type, click **Download**, and decompress the `cloud.tencent.com` certificate file package to the local directory.

After decompression, you can get the certificate file of the corresponding type, which includes the `cloud.tencent.com_jks` folder.

Folder: `cloud.tencent.com_jks`

Files in the folder:

`cloud.tencent.com.jks` : keystore file

`keystorePass.txt` : password file (if you have set a private key password, this file will not be generated)

3. Remotely log in to the Jetty server. For example, you can use PuTTY for remote login.

4. In the `/usr/local/jetty/jetty-distribution-9.4.28.v20200408/etc` directory, run the `mkdir cert` command to create the `cert` folder.

5. Use WinSCP (a tool for copying files between a local computer and a remote computer) to log in to the Jetty server and copy the 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 that you use CVM's file upload feature for deployment to CVM.

6. In the `/usr/local/jetty/jetty-distribution-9.4.28.v20200408/etc` directory, modify the configuration in the `jetty-ssl-context.xml` file.

Note:

KeyStorePath: Set the default value to the certificate path.

KeyStorePassword: Set the default value to the keystore password. If you have set a private key password when applying for the certificate, enter the private key password; otherwise, enter the password in the `keystorePass.txt` file in the `cloud.tencent.com_jks` folder.

KeyManagerPassword: Set the value to the password in the `keystorePass.txt` file in the `cloud.tencent.com_jks` folder.

TrustStorePath: Set the default value to the certificate path.

```
<?xml version="1.0"?><!DOCTYPE Configure PUBLIC "-//Jetty//Configure//EN" "http://w
<!-- ============================================================= --><!-- SSL Cont
<!--
  To configure Includes / Excludes for Cipher Suites or Protocols see tweak-ssl.xml
  https://www.eclipse.org/jetty/documentation/current/configuring-ssl.html#configur
-->
<Configure id="sslContextFactory" class="org.eclipse.jetty.util.ssl.SslContextFacto
  <Set name="Provider"><Property name="jetty.sslContext.provider"/></Set>
  <Set name="KeyStorePath"><Property name="jetty.base" default="." />/<Property nam
  <Set name="KeyStorePassword"><Property name="jetty.sslContext.keyStorePassword" d
  <Set name="KeyStoreType"><Property name="jetty.sslContext.keyStoreType" default="
  <Set name="KeyStoreProvider"><Property name="jetty.sslContext.keyStoreProvider"/>
  <Set name="KeyManagerPassword"><Property name="jetty.sslContext.keyManagerPasswor
  <Set name="TrustStorePath"><Property name="jetty.base" default="." />/<Property n
  <Set name="TrustStorePassword"><Property name="jetty.sslContext.trustStorePasswor
  <Set name="TrustStoreType"><Property name="jetty.sslContext.trustStoreType"/></Se
  <Set name="TrustStoreProvider"><Property name="jetty.sslContext.trustStoreProvide
  <Set name="EndpointIdentificationAlgorithm"><Property name="jetty.sslContext.endp
  <Set name="NeedClientAuth"><Property name="jetty.sslContext.needClientAuth" depre
  <Set name="WantClientAuth"><Property name="jetty.sslContext.wantClientAuth" depre
  <Set name="useCipherSuitesOrder"><Property name="jetty.sslContext.useCipherSuites
  <Set name="sslSessionCacheSize"><Property name="jetty.sslContext.sslSessionCacheS
  <Set name="sslSessionTimeout"><Property name="jetty.sslContext.sslSessionTimeout"
  <Set name="RenegotiationAllowed"><Property name="jetty.sslContext.renegotiationAl
  <Set name="RenegotiationLimit"><Property name="jetty.sslContext.renegotiationLimi
  <Set name="SniRequired"><Property name="jetty.sslContext.sniRequired" default="fa
  <!-- Example of how to configure a PKIX Certificate Path revocation Checker
  <Call id="pkixPreferCrls" class="java.security.cert.PKIXRevocationChecker$Option"
  <Call id="pkixSoftFail" class="java.security.cert.PKIXRevocationChecker$Option" n
  <Call id="pkixNoFallback" class="java.security.cert.PKIXRevocationChecker$Option"
  <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, change the port number to 443 in the `jetty-ssl.xml` file.

```
  <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" deprecat
      <Arg name="selectors" type="int"><Property name="jetty.ssl.selectors" deprecat
      <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" /></S
      <Set name="port"><Property name="jetty.ssl.port" deprecated="ssl.port" default
      <Set name="idleTimeout"><Property name="jetty.ssl.idleTimeout" deprecated="ssl
      <Set name="acceptorPriorityDelta"><Property name="jetty.ssl.acceptorPriorityDe
      <Set name="acceptQueueSize"><Property name="jetty.ssl.acceptQueueSize" depreca
      <Get name="SelectorManager">
        <Set name="connectTimeout"><Property name="jetty.ssl.connectTimeout" default
      </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. In the Jetty root directory, run the `java -jar start.jar` command to start the Jetty 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.

In case of a website access exception, troubleshoot the issue by referring to the following FAQs:

Website Inaccessible After an SSL Certificate is Deployed

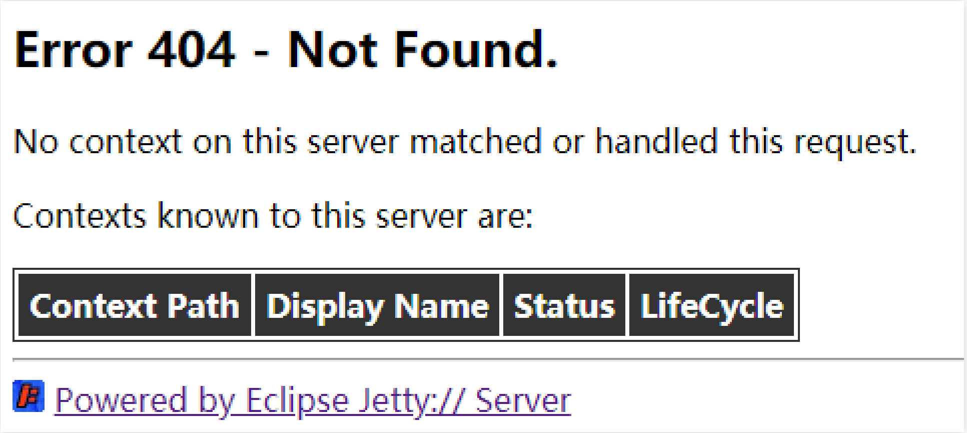
"Your Connection is Not Secure" is Displayed After the SSL Certificate is Installed

Why Does the Website Prompt "Connection Is Untrusted"?

404 Error After the SSL Certificate is Deployed on IIS

Reminders

After the certificate is deployed, the following error message may be displayed when you access `https://cloud.tencent.com`:



Error 404 - Not Found.

No context on this server matched or handled this request.

Contexts known to this server are:

Context Path	Display Name	Status	LifeCycle

Powered by Eclipse Jetty:// Server

If the error message is displayed, 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, and then restart the Jetty server.

Note:

If anything goes wrong during this process, contact us.

Installing an SSL Certificate on an IIS Server

Last updated：2024-03-06 17:38:42

Overview

This document describes how to install an SSL certificate in IIS.

Note:

The certificate name `cloud.tencent.com` is used as an example. The actual name in your certificate shall prevail.

Windows Server 2012 R2 is used as an example. Detailed steps vary slightly by OS.

Before you install an SSL certificate, enable port `443` on the IIS server so 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.

Directions

Installing the certificate

1. Log in to the SSL Certificate Service console, and click **Download** for the certificate you need to install.
2. In the pop-up window, select **IIS** for the server type, click **Download**, and decompress the `cloud.tencent.com` certificate file package to a local directory.

After decompression, you can get the certificate file of the corresponding type, which contains the `cloud.tencent.com.iis` folder.

Folder: `cloud.tencent.com.iis`

Files in the folder:

`cloud.tencent.com.pfx` : certificate file

`keystorePass.txt` : password file (if you have set a private key password, this file will not be generated)

3. Open the IIS Manager, select the computer name, and double-click **Server Certificates**.
4. In the **Actions** column on the right of the **Server Certificates** window, click **Import**.
5. In the **Import Certificate** pop-up window, select the path where the certificate file is stored, enter the password, and click **OK** as shown below:

Note:

If you have set a private key password when applying for the certificate, enter the private key password; otherwise, enter the password in the `keystorePass.txt` file in the `cloud.tencent.com.iis` folder.

If you forgot your private key password, submit a ticket to have the certificate deleted and reapply for one under the domain.

6. Select the name of a site in **Sites** and click **Bindings** in the **Actions** column on the right.
7. In the **Site Bindings** pop-up window, click **Add**.
8. In the **Add Site Binding** window, set **Type** to **https**, **IP address** to **All Unassigned**, and **Port** to **443**, enter the domain of your current certificate in **Host name**, specify the corresponding SSL certificate, and click **OK**.
9. Then, you can see the newly added content in the **Site Bindings** window.
10. Access the website through `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:

Website Inaccessible After an SSL Certificate is Deployed

"Your Connection is Not Secure" is Displayed After the SSL Certificate is Installed

Why Does the Website Prompt "Connection Is Untrusted"?

404 Error After the SSL Certificate is Deployed on IIS

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

Note:

For normal redirect, edit the rule in the following steps. If you have other needs, you can set it on your own.

During the redirect from HTTP to HTTPS, if your website element contains external links or uses the HTTP protocol, the entire webpage is not completely based on HTTPS. In this case, some browsers may prompt for risk such as "this link is unsecure" due to those factors. You can view the error cause by clicking **Details** on the unsecure page.

1. Open the IIS Manager.
2. Select the name of a site in **Sites** and double-click to open **URL Rewrite**.

Note:

Download and install the URL Rewrite module before performing this step.

3. Go to the **URL Rewrite** page and click **Add Rule(s)** in the **Actions** column on the right.
4. In the **Add Rule(s)** pop-up window, select **Blank rule** and click **OK**.
5. Go to the **Edit Inbound Rule** page.

Name: Enter **Forced HTTPS**.

Match URL: Enter `(.*)` in **Pattern**.

Conditions: Click

⌄ to expand and click **Add** to pop up the **Add Condition** window.

Condition input: `{HTTPS}` .

Check if input string: Select "Matches the Pattern" by default.

Pattern: Enter `^OFF$` .

Action: Enter the following parameters.

Action Type: Select "Redirect".

Redirect URL: `https://{HTTP_HOST}/{R:1}` .

Redirect Type: Select "See Other (303)".

6. Click **Apply** in the **Actions** column to save.

7. Return to the **Sites** page and click **Restart** in the **Manage Website** column on the right. Then, the website can be accessed through `http://cloud.tencent.com` .

Note:

If anything goes wrong during this process, contact us.

Installing a Certificate on WebLogic Servers

Last updated：2024-03-06 17:38:42

Overview

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

Note:

The example certificate name used in this document is `cloud.tencent.com`. Please use the actual name of the certificate applied to your environment.

The example WebLogic version used in this document is 14.1.1.

The example operating system used in this document is Windows Server 2012 R2. The operational steps may vary slightly depending on the operating system.

Before installing an Weblogic certificate, enable port 443 on the WebLogic server so that HTTPS can be enabled after the certificate is installed. For more information, see How do I Enable Port 443 for a VM?.

For details about how to upload SSL certificate files to a server, see Copying Local Files to CVMs.

Directions

Note:

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. Download the certificate package for the domain name `cloud.tencent.com` from the SSL Certificate Service console and decompress it to a local directory.

After decompression, you can obtain the certificate files of the relevant types, including the `Tomcat` folder and the CSR file:

Folder name: `Tomcat`

Folder content:

`cloud.tencent.com.jks` : certificate file

`keystorePass.txt` : password file (if you have set a private key password, this file will not be generated)

CSR file content: `cloud.tencent.com.csr` file

Note:

The CSR file is uploaded by you or generated online by the system when you apply for the certificate and is provided to the CA. It is irrelevant to the installation.

If you selected **Paste CSR** when applying for the SSL certificate or purchased the SSL certificate from Wotrus, the option to download the Tomcat certificate file is not provided. Instead, you need to manually convert the format to generate a keystore by using the conversion tool.

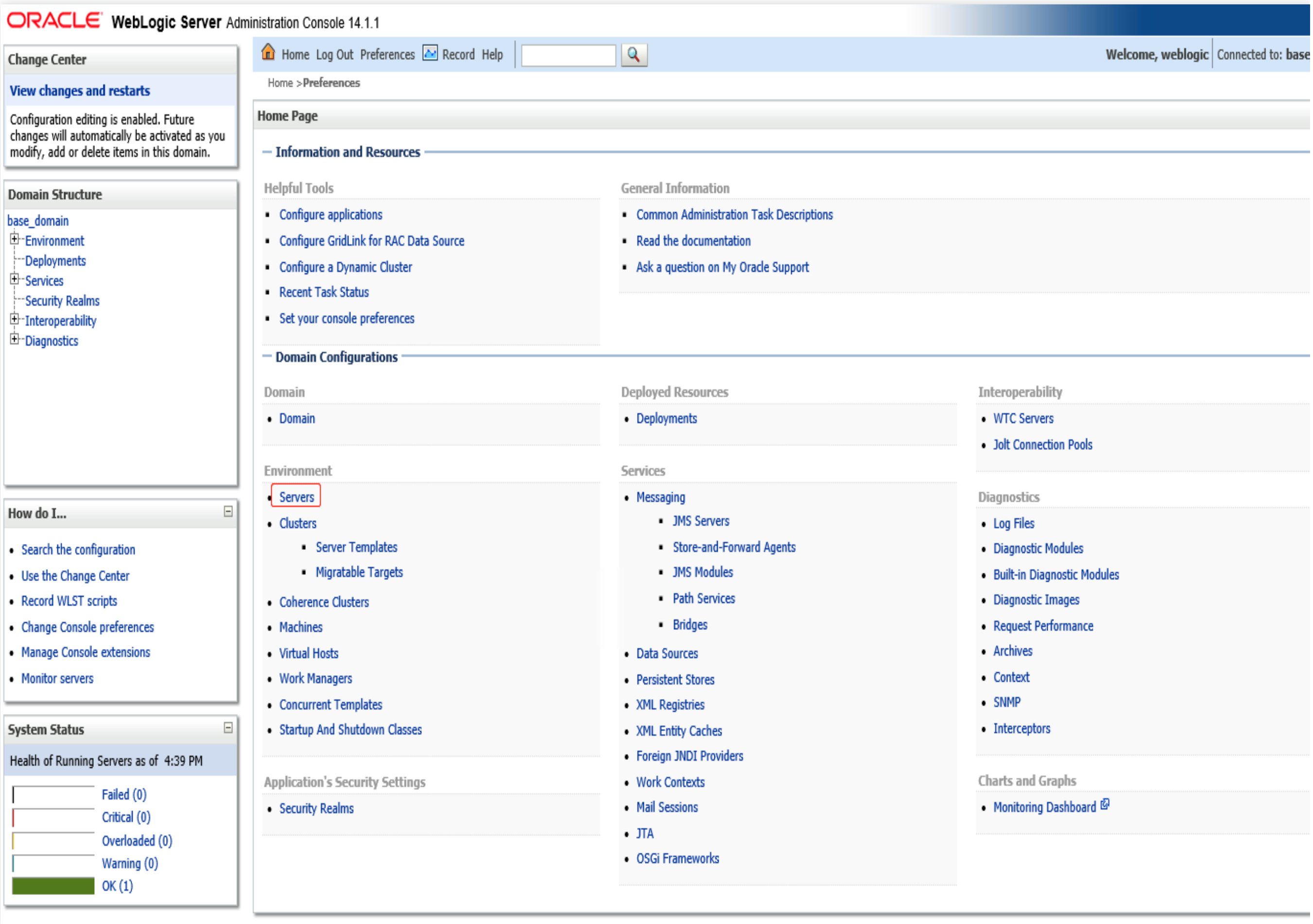
If the Tomcat certificate file is not provided, upload the certificate and private key files in the `Nginx` folder to the conversion tool, enter the keystore password, and click **Submit** to convert the certificate to a .jks certificate.

2. Log in to the server and create a folder. For example, `temp` in the C drive.

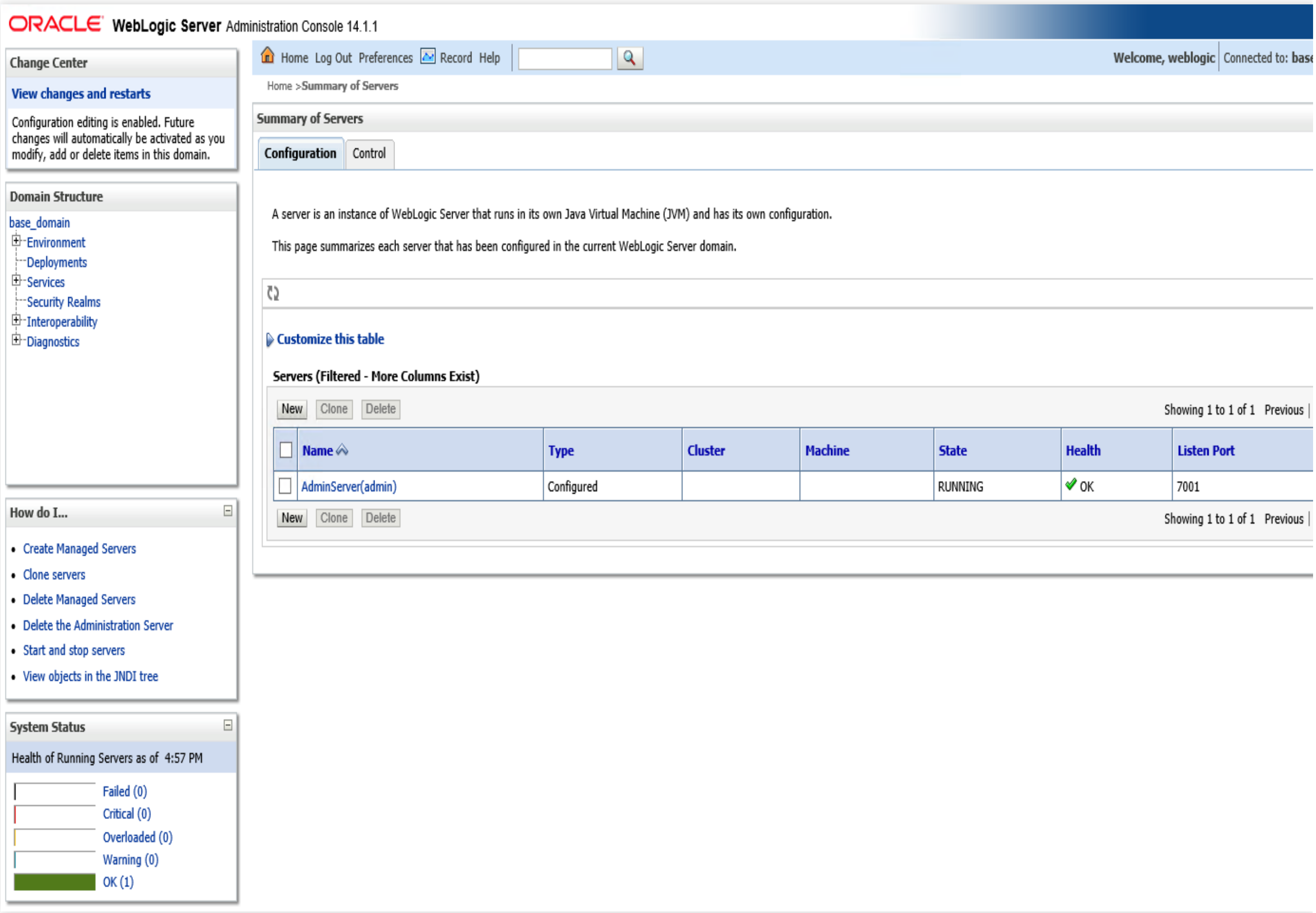
3. Decompress the certificate and password files and upload them to the `temp` folder.

4. Log in to the WebLogic Server Administration Console (default address: `http://localhost:7001/console`) by entering your username and password.

5. Choose **Domain Configurations** -> **Servers**.



6. On the **Summary of Deployments** page, select a target server such as `AdminiServer` .



ORACLE WebLogic Server Administration Console 14.1.1

Home Log Out Preferences Record Help | Welcome, weblogic | Connected to: base

Home >Summary of Servers

Change Center

View changes and restarts

Configuration editing is enabled. Future changes will automatically be activated as you modify, add or delete items in this domain.

Domain Structure

base_domain
- Environment
- Deployments
- Services
- Security Realms
- Interoperability
- Diagnostics

How do I...

- Create Managed Servers
- Clone servers
- Delete Managed Servers
- Delete the Administration Server
- Start and stop servers
- View objects in the JNDI tree

System Status

Health of Running Servers as of 4:57 PM

- Failed (0)
- Critical (0)
- Overloaded (0)
- Warning (0)
- OK (1)

Summary of Servers

Configuration | Control

A server is an instance of WebLogic Server that runs in its own Java Virtual Machine (JVM) and has its own configuration.

This page summarizes each server that has been configured in the current WebLogic Server domain.

Customize this table

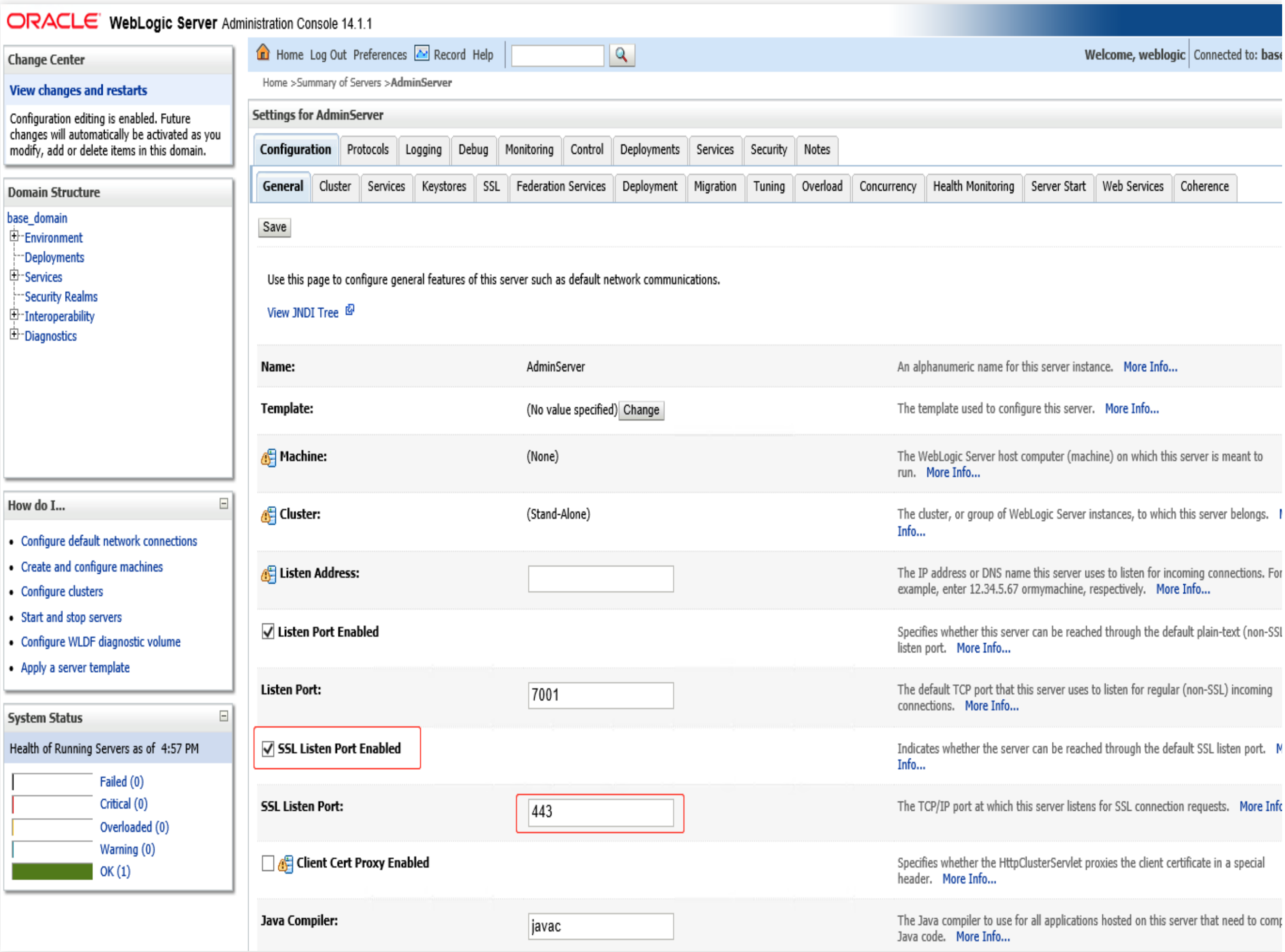
Servers (Filtered - More Columns Exist)

New Clone Delete — Showing 1 to 1 of 1 Previous |

□	Name	Type	Cluster	Machine	State	Health	Listen Port
□	AdminServer(admin)	Configured			RUNNING	✓ OK	7001

New Clone Delete — Showing 1 to 1 of 1 Previous |

7. On the configuration management page for `AdminiServer`, select **SSL Listening Port Enabled**, set **SSL Listening Port** to `443`, and click **Save**.

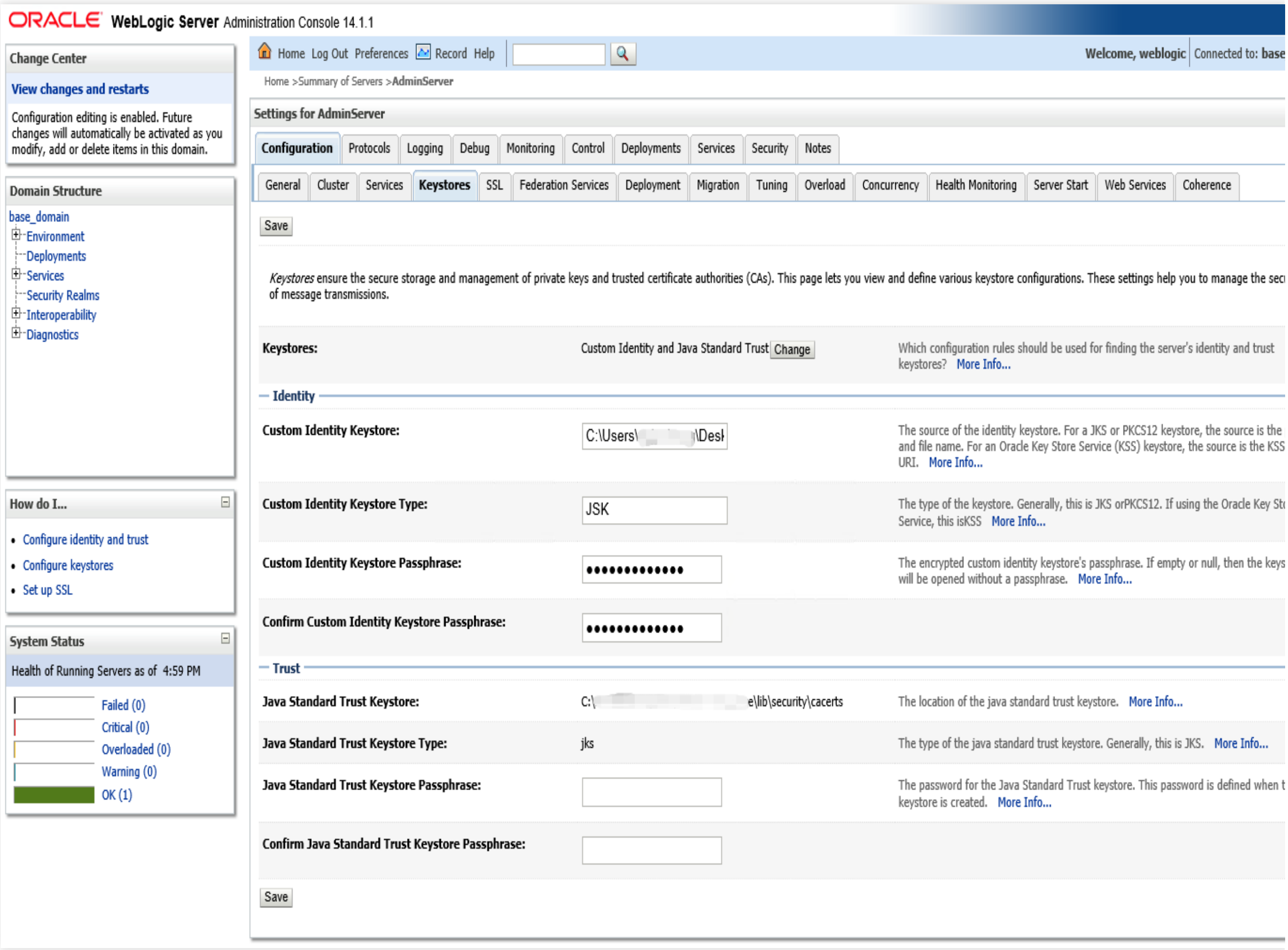

ORACLE WebLogic Server Administration Console 14.1.1
Home Log Out Preferences Record Help
Welcome, weblogic Connected to: base
Home >Summary of Servers >AdminServer
Change Center
View changes and restarts
Domain Structure
base_domain
Environment
Deployments
Services
Security Realms
Interoperability
Diagnostics
How do I...
Configure default network connections
Create and configure machines
Configure clusters
Start and stop servers
Configure WLDF diagnostic volume
Apply a server template
System Status
Health of Running Servers as of 4:57 PM
Failed (0)
Critical (0)
Overloaded (0)
Warning (0)
OK (1)
Settings for AdminServer
Configuration Protocols Logging Debug Monitoring Control Deployments Services Security Notes
General Cluster Services Keystores SSL Federation Services Deployment Migration Tuning Overload Concurrency Health Monitoring Server Start Web Services Coherence
Save
View JNDI Tree
Change

Configuration editing is enabled. Future changes will automatically be activated as you modify, add or delete items in this domain.

Use this page to configure general features of this server such as default network communications.

Name:	AdminServer	An alphanumeric name for this server instance. More Info...
Template:	(No value specified)	The template used to configure this server. More Info...
Machine:	(None)	The WebLogic Server host computer (machine) on which this server is meant to run. More Info...
Cluster:	(Stand-Alone)	The cluster, or group of WebLogic Server instances, to which this server belongs. M Info...
Listen Address:		The IP address or DNS name this server uses to listen for incoming connections. For example, enter 12.34.5.67 ormymachine, respectively. More Info...
☑ Listen Port Enabled		Specifies whether this server can be reached through the default plain-text (non-SSL listen port. More Info...
Listen Port:	7001	The default TCP port that this server uses to listen for regular (non-SSL) incoming connections. More Info...
☑ SSL Listen Port Enabled		Indicates whether the server can be reached through the default SSL listen port. M Info...
SSL Listen Port:	443	The TCP/IP port at which this server listens for SSL connection requests. More Info
□ Client Cert Proxy Enabled		Specifies whether the HttpClusterServlet proxies the client certificate in a special header. More Info...
Java Compiler:	javac	The Java compiler to use for all applications hosted on this server that need to comp Java code. More Info...

8. On the configuration management page for `AdminiServer` , click the **Keystores** tab, set the parameters, and click **Save**.


ORACLE WebLogic Server Administration Console 14.1.1
Home Log Out Preferences Record Help
Welcome, weblogic Connected to: base
Change Center
View changes and restarts
Domain Structure
base_domain
Environment
Deployments
Services
Security Realms
Interoperability
Diagnostics
How do I...
Configure identity and trust
Configure keystores
Set up SSL
System Status
Health of Running Servers as of 4:59 PM
Failed (0)
Critical (0)
Overloaded (0)
Warning (0)
OK (1)
Home >Summary of Servers >AdminServer
Configuration Protocols Logging Debug Monitoring Control Deployments Services Security Notes
General Cluster Services Keystores SSL Federation Services Deployment Migration Tuning Overload Concurrency Health Monitoring Server Start Web Services Coherence
Save
Change
More Info...
Identity
Trust
Save

Configuration editing is enabled. Future changes will automatically be activated as you modify, add or delete items in this domain.

Settings for AdminServer

Keystores ensure the secure storage and management of private keys and trusted certificate authorities (CAs). This page lets you view and define various keystore configurations. These settings help you to manage the secu of message transmissions.

Keystores:	Custom Identity and Java Standard Trust	Which configuration rules should be used for finding the server's identity and trust keystores? More Info...
Custom Identity Keystore:	C:\Users\ \Desl	The source of the identity keystore. For a JKS or PKCS12 keystore, the source is the and file name. For an Oracle Key Store Service (KSS) keystore, the source is the KSS URI. More Info...
Custom Identity Keystore Type:	JSK	The type of the keystore. Generally, this is JKS orPKCS12. If using the Oracle Key Sto Service, this isKSS More Info...
Custom Identity Keystore Passphrase:	••••••••••••••	The encrypted custom identity keystore's passphrase. If empty or null, then the keys will be opened without a passphrase. More Info...
Confirm Custom Identity Keystore Passphrase:	••••••••••••••	
Java Standard Trust Keystore:	C:\ e\lib\security\cacerts	The location of the java standard trust keystore. More Info...
Java Standard Trust Keystore Type:	jks	The type of the java standard trust keystore. Generally, this is JKS. More Info...
Java Standard Trust Keystore Passphrase:		The password for the Java Standard Trust keystore. This password is defined when t keystore is created. More Info...
Confirm Java Standard Trust Keystore Passphrase:		

Set the parameters as follows:

Keystores: select **Custom Identity and Java Standard Trust**.

Custom Identity Keystore: enter the path to your JKS certificate file. For example, `C:\\temp\\cloud.tencent.com.jks` .

Custom Identity Keystore Type: enter `JKS` .

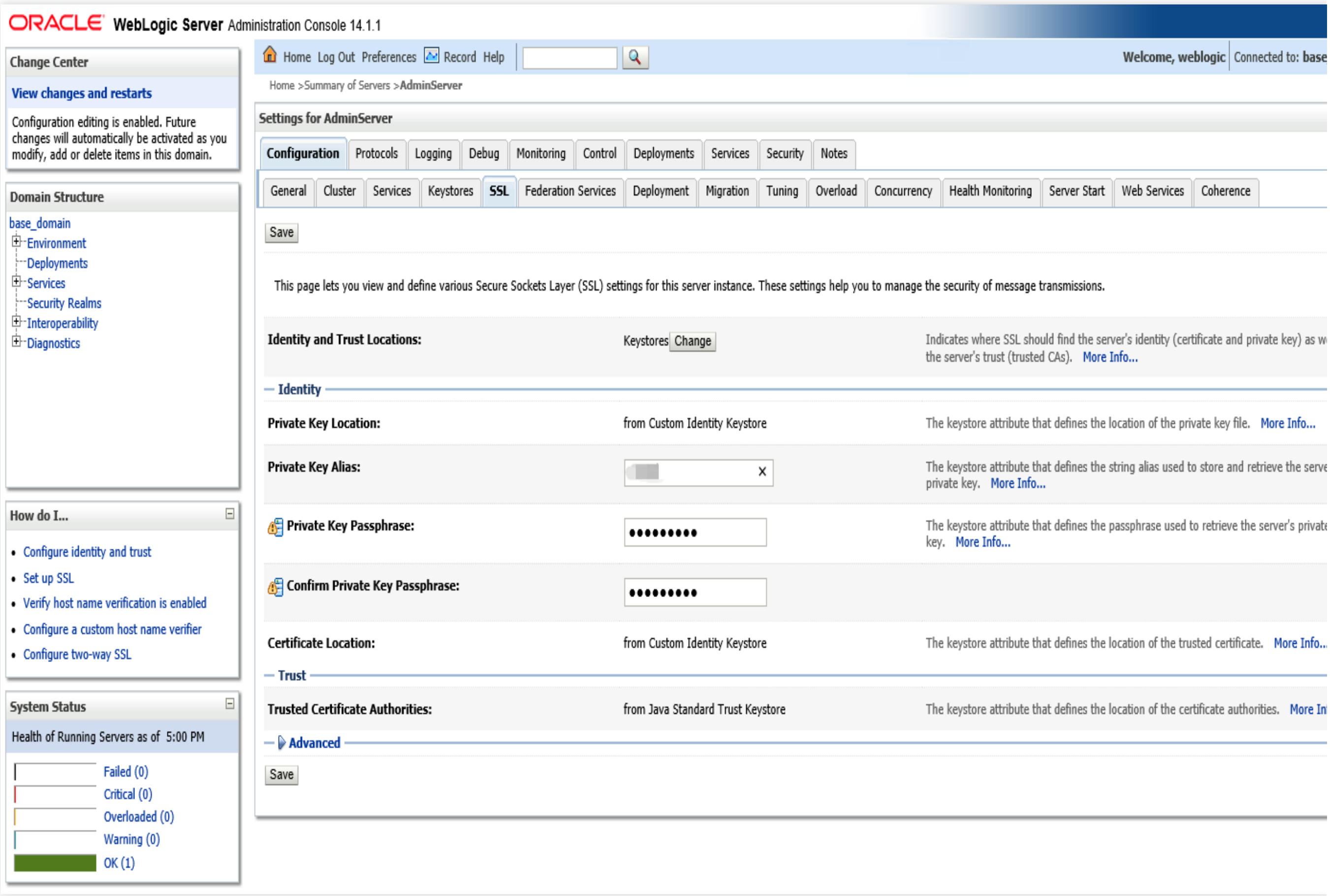
Custom Identity Keystore Passphrase: enter your JKS password.

Confirm Custom Identity Keystore Passphrase: re-enter your JKS password.

Note:

Custom Identity Keystore Passphrase and **Confirm Custom Identity Keystore Passphrase** are left empty by default and can be set to your JKS password. The settings of these 2 parameters do not affect the use of your certificate.

9. On the configuration management page for `AdminiServer` , click the **SSL** tab, set the parameters, and click **Save**.



Configuration editing is enabled. Future changes will automatically be activated as you modify, add or delete items in this domain.

Health of Running Servers as of 5:00 PM

Settings for AdminServer

This page lets you view and define various Secure Sockets Layer (SSL) settings for this server instance. These settings help you to manage the security of message transmissions.

Identity and Trust Locations: Keystores — Indicates where SSL should find the server's identity (certificate and private key) as w the server's trust (trusted CAs).

Private Key Location: from Custom Identity Keystore — The keystore attribute that defines the location of the private key file.

Private Key Alias: — The keystore attribute that defines the string alias used to store and retrieve the serve private key.

Private Key Passphrase: ••••••••• — The keystore attribute that defines the passphrase used to retrieve the server's private key.

Confirm Private Key Passphrase: •••••••••

Certificate Location: from Custom Identity Keystore — The keystore attribute that defines the location of the trusted certificate.

Trusted Certificate Authorities: from Java Standard Trust Keystore — The keystore attribute that defines the location of the certificate authorities.

Set the parameters as follows:

Identity and Trust Locations: set it to `KEYSTORES` .

Private Key Alias: enter the JKS alias.

Private Key Passphrase: if you set a private password when applying for a certificate, enter the private password. Otherwise, leave this parameter empty.

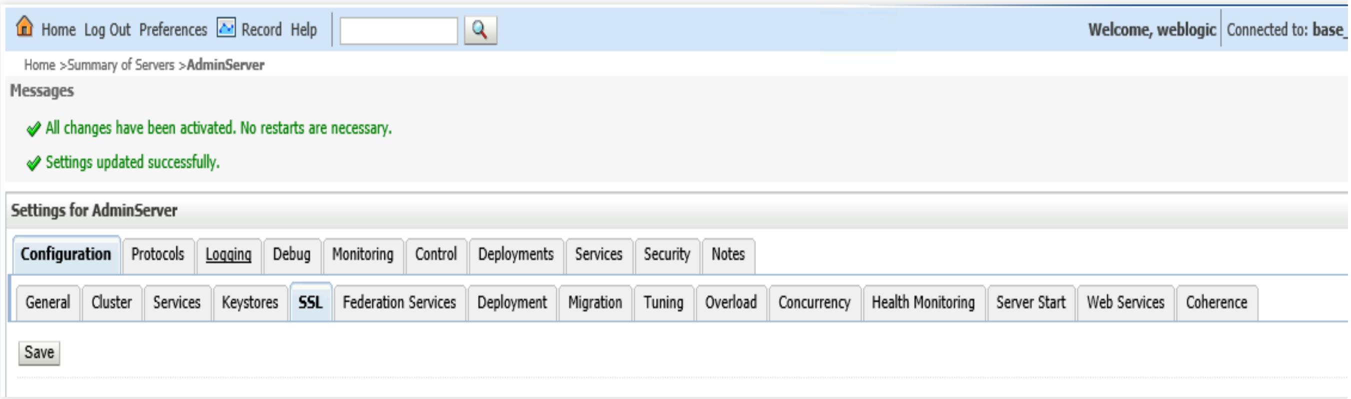
Confirm Private Key Passphrase: re-enter the private password.

Note:

For WebLogic 10.3.6-12c, select **JSSE** in the advanced settings area on the SSL settings page.

WebLogic versions earlier than 10.3.6 do not support SHA2 certificates. Please upgrade and try again.

10. Click **Save** after modifying the necessary information. The modifications are automatically activated, eliminating the need for restart.


Home Log Out Preferences Record Help
Welcome, weblogic Connected to: base_
Home >Summary of Servers >AdminServer

Messages

✓ All changes have been activated. No restarts are necessary.

✓ Settings updated successfully.

Settings for AdminServer

Configuration Protocols Logging Debug Monitoring Control Deployments Services Security Notes
General Cluster Services Keystores SSL Federation Services Deployment Migration Tuning Overload Concurrency Health Monitoring Server Start Web Services Coherence
Save

11. Access `https://cloud.tencent.com` .

Selecting an Installation Type for an SSL Certificate

Last updated：2024-03-06 17:38:42

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:

The quick HTTPS feature helps you upgrade from HTTP to HTTPS without tedious SSL certificate deployment.

Currently, 15 methods are available for installing a certificate.

Certificate Type	Server System	Certificate Installation Method
International standard certificate (RSA/ECC)	Linux	Installing an SSL Certificate on an Apache Server (Linux)
		Installing an SSL Certificate on an Nginx Server
		Installing an SSL Certificate (JKS Format) on a Tomcat Server
		Installing an SSL Certificate (PFX Format) on a Tomcat Server
		Installing an SSL Certificate on a GlassFish Server
		Installing an SSL Certificate on a JBoss Server
		Installing an SSL Certificate on a Jetty Server
	Windows	Installing a Certificate on IIS Servers
		Installing a Certificate on WebLogic Servers
		Installing an SSL Certificate on an Apache Server (Windows)
		Installing an SSL Certificate (JKS Format) on a Tomcat Server