Alibaba Cloud SSL证书（CA证书服务、数据安全）

Best Practices

Issue: 20200507

Legal disclaimer

Alibaba Cloud reminds you to carefully read and fully understand the terms and conditions of this legal disclaimer before you read or use this document. If you have read or used this document, it shall be deemed as your total acceptance of this legal disclaimer.

1. You shall download and obtain this document from the Alibaba Cloud website or other Alibaba Cloud-authorized channels, and use this document for your own legal business activities only. The content of this document is considered confidential information of Alibaba Cloud. You shall strictly abide by the confidentiality obligations. No part of this document shall be disclosed or provided to any third party for use without the prior written consent of Alibaba Cloud.
2. No part of this document shall be excerpted, translated, reproduced, transmitted, or disseminated by any organization, company, or individual in any form or by any means without the prior written consent of Alibaba Cloud.
3. The content of this document may be changed due to product version upgrades, adjustments, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and the updated versions of this document will be occasionally released through Alibaba Cloud-authorized channels. You shall pay attention to the version changes of this document as they occur and download and obtain the most up-to-date version of this document from Alibaba Cloud-authorized channels.
4. This document serves only as a reference guide for your use of Alibaba Cloud products and services. Alibaba Cloud provides the document in the context that Alibaba Cloud products and services are provided on an "as is", "with all faults" and "as available" basis. Alibaba Cloud makes every effort to provide relevant operational guidance based on existing technologies. However, Alibaba Cloud hereby makes a clear statement that it in no way guarantees the accuracy, integrity, applicability, and reliability of the content of this document, either explicitly or implicitly. Alibaba Cloud shall not bear any liability for any errors or financial losses incurred by any organizations, companies, or individuals arising from their download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, bear responsibility for any indirect, consequential, exemplary, incidental, special, or punitive damages, including lost profits arising from the use or trust in this document, even if Alibaba Cloud has been notified of the possibility of such a loss.

5. By law, all the contents in Alibaba Cloud documents, including but not limited to pictures, architecture design, page layout, and text description, are intellectual property of Alibaba Cloud and/or its affiliates. This intellectual property includes, but is not limited to, trademark rights, patent rights, copyrights, and trade secrets. No part of this document shall be used, modified, reproduced, publicly transmitted, changed, disseminated, distributed, or published without the prior written consent of Alibaba Cloud and/or its affiliates. The names owned by Alibaba Cloud shall not be used, published, or reproduced for marketing, advertising, promotion, or other purposes without the prior written consent of Alibaba Cloud. The names owned by Alibaba Cloud include, but are not limited to, "Alibaba Cloud", "Aliyun", "HiChina", and other brands of Alibaba Cloud and/or its affiliates, which appear separately or in combination, as well as the auxiliary signs and patterns of the preceding brands, or anything similar to the company names, trade names, trademarks, product or service names, domain names, patterns, logos, marks, signs, or special descriptions that third parties identify as Alibaba Cloud and/or its affiliates.
6. Please contact Alibaba Cloud directly if you discover any errors in this document.

Document conventions

Style	Description	Example
	A danger notice indicates a situation that will cause major system changes, faults, physical injuries, and other adverse results.	 **Danger:** Resetting will result in the loss of user configuration data.
	A warning notice indicates a situation that may cause major system changes, faults, physical injuries, and other adverse results.	 **Warning:** Restarting will cause business interruption. About 10 minutes are required to restart an instance.
	A caution notice indicates warning information, supplementary instructions, and other content that the user must understand.	 **Notice:** If the weight is set to 0, the server no longer receives new requests.
	A note indicates supplemental instructions, best practices, tips, and other content.	 **Note:** You can use Ctrl + A to select all files.
>	Closing angle brackets are used to indicate a multi-level menu cascade.	Click **Settings** > **Network** > **Set network type**.
Bold	Bold formatting is used for buttons, menus, page names, and other UI elements.	Click **OK**.
`Courier font`	Courier font is used for commands.	Run the `cd /d C:/window` command to enter the Windows system folder.
Italic	Italic formatting is used for parameters and variables.	`bae log list --instanceid` *Instance_ID*
[] or [a|b]	This format is used for an optional value, where only one item can be selected.	`ipconfig [-all|-t]`

Style	Description	Example
{} or {a|b}	This format is used for a required value, where only one item can be selected.	switch {active|stand}

Contents

1 Deploy SSL certificate on Ubuntu Apache2

This manual describes how to install the Alibaba Cloud SSL certificate in Apache2 on Ubuntu.

Environment

OS: Ubuntu

Web server: Apache2

Prerequisites

- The Apache server certificate is downloaded from the Alibaba Cloud SSL certificate services console.
- Open SSL is installed.

Steps

1. In apache2 directory, execute the following command to create ssl directory.

   `mkdir /etc/apache2/ssl`

2. Execute the following command to copy the downloaded Alibaba Cloud certificate file to ssl directory.

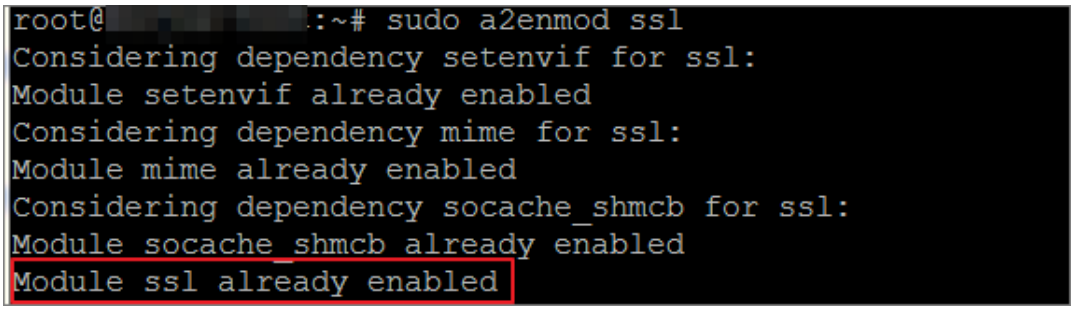
   `cp -r YourDomainName_public.crt /etc/apache2/ssl`

   `cp -r YourDomainName_chain.crt /etc/apache2/ssl`

   `cp -r YourDomainName.key /etc/apache2/ssl`

3. Execute the following command to enable the SSL module.

   sudo a2enmod ssl



   After the SSL module is enabled, you can execute ls /etc/apache2/sites-available and view the default-ssl.conf file created in the directory.

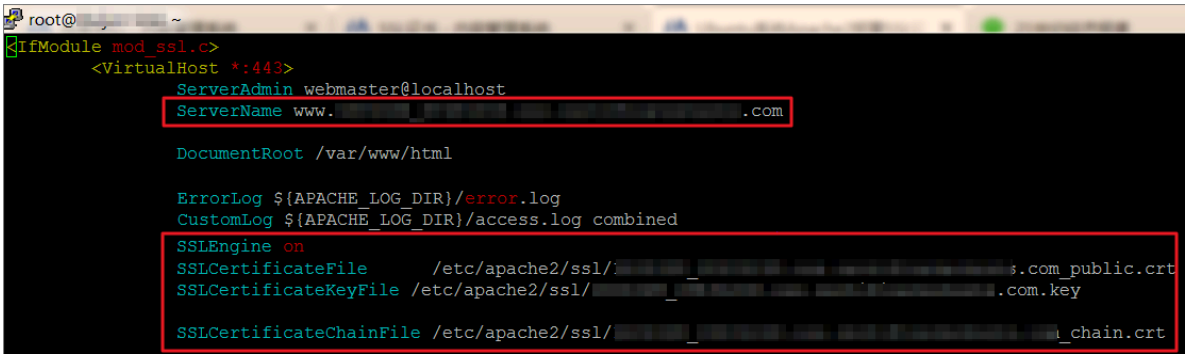
> **Note:**
> Port 443 is a network browsing port that is used primarily for HTTPS services. After the SSL module is enabled, port 443 is automatically released. If port 443 is not automatically released, you can execute vi /etc/apache2/ports.conf and add Listen 443 to manually release it.

4. Execute the following command to modify the configuration file default-ssl.conf for certificate installation.

   vi /etc/apache2/sites-available/default-ssl.conf

   Indefault-ssl.conffile, find the following parameters and modify the parameters. After modification is complete, click :wq to save and exit.

```
<IfModules mod_ssl.c>
<VirtualHost *:443>
ServerName #change to the domain as www.YourDomainName.com bound by the certificate.
SSLCertificateFile /etc/apache2/ssl/www.YourDomainName_public.crt #replace/etc/apache2/ssl/www.YourDomainName.com_public.crt with certificate file path+certificate file name.
SSLCertificateKeyFile /etc/apache2/ssl/www.YourDomainName.com.key  #replace /etc/apache2/ssl/www.YourDomainName.com.key with certificate key file path+certificate key file name.
SSLCertificateChainFile /etc/apache2/ssl/www.YourDomainName.com_chain.crt  #replace/etc/apache2/ssl/www.YourDomainName.com_chain.crt with certificate chain file path+certificate chain file name.
```



/sites-available: This directory stores available virtual machine host; /sites-enabled: This directory stores enabled virtual machine host.

Note:

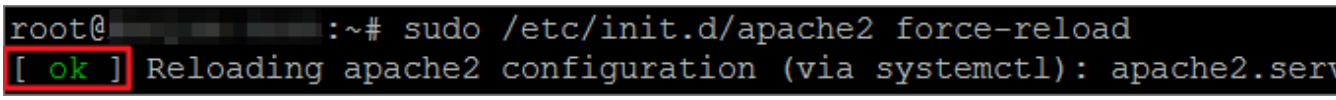
default-ssl.conf This file may be stored at /etc/apache2/sites-available or /etc/apache2/sites-enabled.

5. Map default-ssl.conf to /etc/apache2/sites-enabled folder, create soft links in order to automatically link the two folders.

```
sudo ln -s /etc/apache2/sites-available/default-ssl.conf /etc/apache2/sites-enabled/001-ssl.conf
```

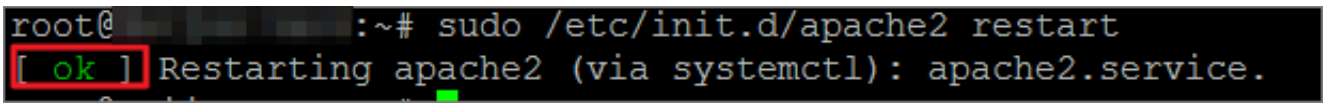
6. Reload the Apache2 configuration file.

```
sudo /etc/init.d/apache2 force-reload
```



7. Execute the following command to restart the Apache2 service.

```
sudo /etc/init.d/apache2 restart
```



What to do next

Apache2 service is reloaded successfully. You can enter https://www.YourDomainName.com in your explorer to validate certificate installation result.

2 Deploy SSL certificates on CentOS-based Tomcat 8.5 or 9.0

This topic describes how to deploy SSL certificates on a Tomcat 8.5 or Tomcat 9.0 server that runs CentOS.

Environment

Operating system: CentOS 7.6, 64-bit

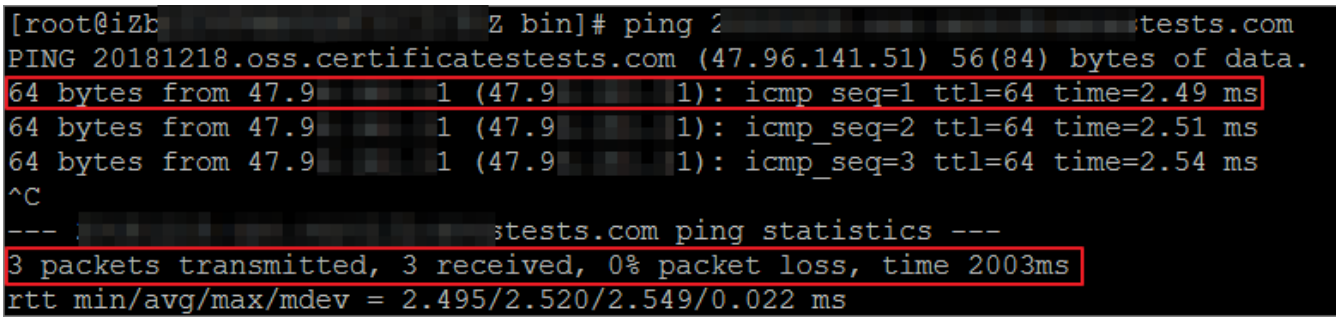
Web server: Tomcat 8.5 or Tomcat 9.0

Note:
JDK environment variables must be installed on the Tomcat server first. You can view the recommended JDK compatible configuration on the Tomcat official website.

Prerequisites

- You have downloaded the Tomcat server certificate from the SSL Certificates Service console. The Tomcat server certificate includes the certificate file in PFX format and the password file in TXT format.
- You have completed domain name resolution for the domain name that you bound to your SSL certificate when you applied for this certificate. You have also pointed this domain name to the IP address of your Tomcat server.

  Run the **ping www.yourdomain.com** command after the domain name resolution is configured. If the IP address of your Tomcat server is returned, the resolution is successful.



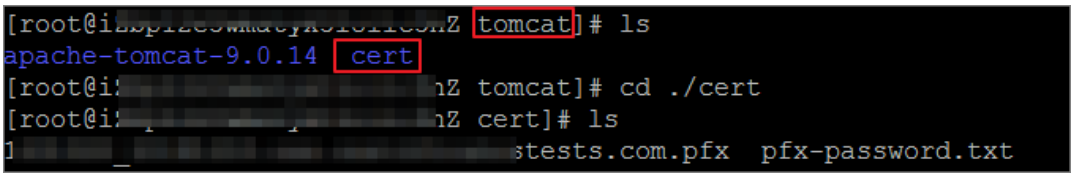
Procedure

1. Decompress the Tomcat server certificate.

   Note:

   A new password file is generated each time you download the certificate. The password is valid only for the currently downloaded certificate. If you want to update the certificate, you must update the password at the same time.

2. Create the **cert** directory in the **Tomcat** installation directory and copy the downloaded certificate and password files to the **cert** directory.



   Note:

   To install a JKS certificate, run the following command to convert a PFX certificate to a JKS certificate:

```
keytool -importkeystore -srckeystore domain name.pfx -destkeystore domain name.jks -srcstoretype PKCS12 -deststoretype JKS
```

3. Open Tomcat/conf/server.xml, find the following parameters in the server.xml file, and modify these parameters.

```
<Connector port="8080" protocol="HTTP/1.1"
           connectionTimeout="20000"
           redirectPort="8443" />

 #Find the preceding parameters, remove the <! - - and - -> annotation symbols, and modify the parameters as follows to configure the default HTTPS port:
 <Connector port="80" protocol="HTTP/1.1"   #Set Connector port to 80.
           connectionTimeout="20000"
           redirectPort="443" />    #Set redirectPort to the default SSL port 443 to redirect HTTPS requests to this port.
```

```
   <Connector port="8443"
       protocol="org.apache.coyote.http11.Http11NioProtocol"
       maxThreads="150"
       SSLEnabled="true">
     <SSLHostConfig>
       <Certificate      certificateKeystoreFile="cert/keystore.pfx"
        certificateKeystorePassword="XXXXXXX"
                 certificateKeystoreType="PKCS12" />

   #Find the preceding parameters, remove the <! - - and - -> annotation symbols, and modify the parameters as follows:
```

```
    <Connector port="443"   # Change Connector port from 8443 to the default Tomcat
 HTTPS port 443. Port 8443 cannot be directly accessed through the domain name
. Therefore, you must append a port number to the domain name. Port 443 is the
 default HTTPS port. You can directly access this port through the domain name
without the need to append a port number to the domain name.
        protocol="org.apache.coyote.http11.Http11NioProtocol"   #Connector port in the
server.xml file has two modes: NIO and APR. In this deployment, the NIO mode is used
. The protocol="org.apache.coyote.http11.Http11NioProtocol" specifies the NIO mode.
        maxThreads="150"
        SSLEnabled="true">
      <SSLHostConfig>
        <Certificate      certificateKeystoreFile="/usr/local/tomcat/cert/Certificate
Domain Name.pfx"   #The certificateKeystoreFile parameter specifies the path of the
certificate file. Use your certificate path and file name to replace Certificate Domain
Name.pfx, for example, certificateKeystoreFile="/usr/local/tomcat/cert/abc.com.pfx".
         certificateKeystorePassword="Password"   #The certificateKeystorePassword
parameter specifies the password for the SSL certificate. Use your certificate password
 in pfx-password.txt to replace it, for example, certificateKeystorePassword="
bMNML1Df".
         certificateKeystoreType="PKCS12" />   #When the certificate type is PFX, set
certificateKeystoreType to PKCS12.
      </SSLHostConfig>
   </Connector>
```

```
<Connector port="8009" protocol="AJP/1.3" redirectPort="8443" />

#Find the preceding parameters, remove the <! - - and - -> annotation symbols, and
modify the parameters as follows:
<Connector port="8009" protocol="AJP/1.3" redirectPort="443" />  #Set redirectPort to
443 to redirect HTTPS requests to this port.
```

4. Save the configuration in the server.xml file.
5. (Optional) Add the following content at the bottom of the web.xml file to automatically redirect HTTP requests to HTTPS:

```
<security-constraint>
       <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>
```

6. Restart the Tomcat service.

   a. Run **./shutdown.sh** in the bin directory of Tomcat to disable the Tomcat service.

```
[root@iZ                    Z bin]# ./shutdown.sh
Using CATALINA_BASE:   /usr/local/tomcat/apache-tomcat-9.0.14
Using CATALINA_HOME:   /usr/local/tomcat/apache-tomcat-9.0.14
Using CATALINA_TMPDIR: /usr/local/tomcat/apache-tomcat-9.0.14/temp
Using JRE_HOME:        /usr/local/java/jdk-11.0.2
Using CLASSPATH:       /usr/local/tomcat/apache-tomcat-9.0.14/bin/bootstrap.jar:/usr/local
ache-tomcat-9.0.14/bin/tomcat-juli.jar
NOTE: Picked up JDK_JAVA_OPTIONS:  --add-opens=java.base/java.lang=ALL-UNNAMED --add-opens
/java.io=ALL-UNNAMED --add-opens=java.rmi/sun.rmi.transport=ALL-UNNAMED
[root@iZ                    nZ bin]# ps -ef|grep java
root       939    843  0 16:37 pts/2    00:00:00 grep --color=auto java
```

   b. Run **./startup.sh** in the bin directory of Tomcat to enable the Tomcat service.

```
[root@iZ                    nZ bin]# ./startup.sh
Using CATALINA_BASE:   /usr/local/tomcat/apache-tomcat-9.0.14
Using CATALINA_HOME:   /usr/local/tomcat/apache-tomcat-9.0.14
Using CATALINA_TMPDIR: /usr/local/tomcat/apache-tomcat-9.0.14/temp
Using JRE_HOME:        /usr/local/java/jdk-11.0.2
Using CLASSPATH:       /usr/local/tomcat/apache-tomcat-9.0.14/bin/bootstrap.jar:/usr/loca
ache-tomcat-9.0.14/bin/tomcat-juli.jar
Tomcat started.
```

What to do next

After the Tomcat service is restarted, enter https://www.YourDomainName.com in the address bar of your browser to verify the certificate installation result. It is the domain name that you bound to your SSL certificate. If the green lock icon appears in the address bar of your browser, the certificate is installed.

References

- Install SSL certificates on Tomcat servers
- #unique_6
- Deploy SSL certificate on Ubuntu Apache2
- #unique_7
- #unique_8
- #unique_9
- #unique_10