阿里云
SSL证书

最佳实践

文档版本：20190805

法律声明

阿里云提醒您在阅读或使用本文档之前仔细阅读、充分理解本法律声明各条款的内容。如果您阅读或使用本文档，您的阅读或使用行为将被视为对本声明全部内容的认可。

1. 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档，且仅能用于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息，您应当严格遵守保密义务；未经阿里云事先书面同意，您不得向任何第三方披露本手册内容或提供给任何第三方使用。
2. 未经阿里云事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。
3. 由于产品版本升级、调整或其他原因，本文档内容有可能变更。阿里云保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在阿里云授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过阿里云授权渠道下载、获取最新版的用户文档。
4. 本文档仅作为用户使用阿里云产品及服务的参考性指引，阿里云以产品及服务的”现状“、“有缺陷”和“当前功能”的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引，但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的，阿里云不承担任何法律责任。在任何情况下，阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害，包括用户使用或信赖本文档而遭受的利润损失，承担责任（即使阿里云已被告知该等损失的可能性）。
5. 阿里云网站上所有内容，包括但不限于著作、产品、图片、档案、资讯、资料、网站架构、网站画面的安排、网页设计，均由阿里云和/或其关联公司依法拥有其知识产权，包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意，任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外，未经阿里云事先书面同意，任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称（包括但不限于单独为或以组合形式包含”阿里云”、Aliyun”、“万网”等阿里云和/或其关联公司品牌，上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司）。
6. 如若发现本文档存在任何错误，请与阿里云取得直接联系。

通用约定

格式	说明	样例
	该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。	 **禁止:** 重置操作将丢失用户配置数据。
	该类警示信息可能导致系统重大变更甚至故障，或者导致人身伤害等结果。	 **警告:** 重启操作将导致业务中断，恢复业务所需时间约10分钟。
	用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。	 **说明:** 您也可以通过按Ctrl + A选中全部文件。
>	多级菜单递进。	设置 > 网络 > 设置网络类型
粗体	表示按键、菜单、页面名称等UI元素。	单击 **确定**。
`courier` 字体	命令。	执行 `cd /d C:/windows` 命令，进入Windows系统文件夹。
##	表示参数、变量。	`bae log list --instanceid` *`Instance_ID`*
[]或者[a|b]	表示可选项，至多选择一个。	`ipconfig` *`[-all|-t]`*
{}或者{a|b}	表示必选项，至多选择一个。	`swich` *`{stand | slave}`*

目录

1 Ubuntu系统Apache 2部署SSL证书

本文档为您介绍了如何在Ubuntu系统以及Apache2中安装阿里云SSL证书。

环境准备

操作系统：Ubuntu

Web服务器：Apache 2

前提条件

- 已从SSL证书控制台下载Apache服务器证书。
- 已安装Open SSL。

操作步骤

1. 运行以下命令在*apache2*目录下创建*ssl*目录。

```
mkdir /etc/apache2/ssl
```

2. 运行以下命令将下载的阿里云证书文件复制到*ssl*目录中。

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

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

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

3. 运行以下命令启用SSL模块。

```
sudo a2enmod ssl
```

```
root@ :~# sudo a2enmod ssl
Considering dependency setenvif for ssl:
Module setenvif already enabled
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Module socache_shmcb already enabled
Module ssl already enabled
```

SSL模块启用后可执行`ls /etc/apache2/sites-available`查看目录下生成的*default-ssl.conf*文件。

说明:

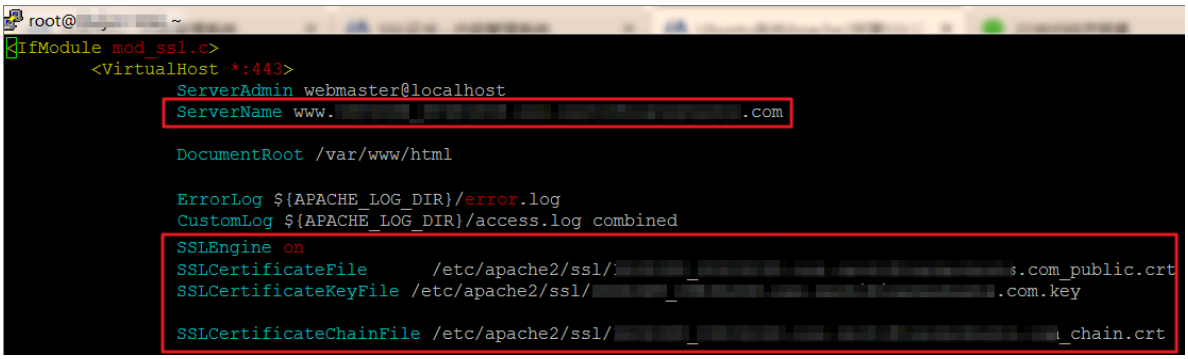
443端口是网络浏览端口，主要用于HTTPS服务。SSL模块启用后会自动放行443端口。若443端口未自动放行，可执行`vi /etc/apache2/ports.conf`并添加`Listen 443`手动放行。

4. 运行以下命令修改SSL配置文件*default-ssl.conf*。

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

在*default-ssl.conf*文件中找到以下参数进行修改后保存并退出。

```
<IfModules mod_ssl.c>
<VirtualHost *:443>
ServerName   #修改为证书绑定的域名www.YourDomainName.com。
SSLCertificateFile /etc/apache2/ssl/www.YourDomainName_public.crt
 #将/etc/apache2/ssl/www.YourDomainName.com_public.crt替换为证书文件路径+证书文件名。
SSLCertificateKeyFile /etc/ssl/apache2/www.YourDomainName.com.key
 #将/etc/apache2/ssl/www.YourDomainName.com.key替换为证书秘钥文件路径+证书秘钥文件名。
SSLCertificateChainFile /etc/apache2/ssl/www.YourDomainName.
com_chain.crt  #将/etc/apache2/ssl/www.YourDomainName.com_chain.crt替换为证书链文件路径+证书链文件名。
```



/sites-available：该目录存放的是可用的虚拟主机；*/sites-enabled*：该目录存放的是已经启用的虚拟主机。

说明：

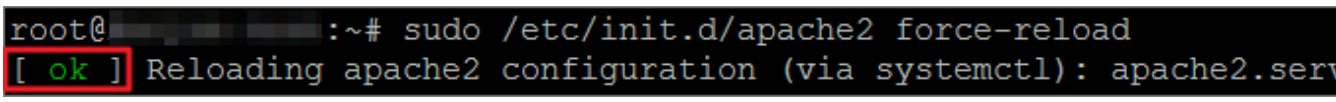
*default-ssl.conf*文件可能存放在*/etc/apache2/sites-available*或*/etc/apache2/sites-enabled*目录中。

5. 运行以下命令把*default-ssl.conf*映射至*/etc/apache2/sites-enabled*文件夹中建立软链接、实现二者之间的自动关联。

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

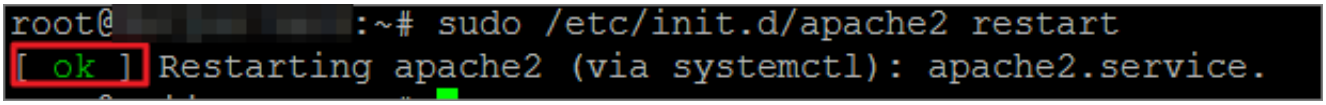
6. 运行以下命令重新加载Apache 2配置文件。

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



7. 运行以下命令重启Apache 2服务。

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



后续操作

Apache 2服务重启成功后，您可在浏览器中输入*https://www.YourDomainName.com*验证证书安装结果。浏览器地址栏显示绿色的小锁标识说明证书安装成功。

安装证书相关文档：

- 在Tomcat服务器上安装SSL证书
- 在Apache服务器上安装SSL证书
- 我获取到的数字证书如何配置在自己的Apache中?
- 在Nginx/Tengine服务器上安装证书
- 在IIS服务器上安装证书
- CentOS系统Tomcat 8.5/9部署SSL证书
- Jetty服务器配置SSL证书

2 CentOS系统Tomcat 8.5/9部署SSL证书

本文档介绍了CentOS系统下Tomcat 8.5或9部署SSL证书的操作说明。

环境准备

操作系统：CentOS 7.6 64位

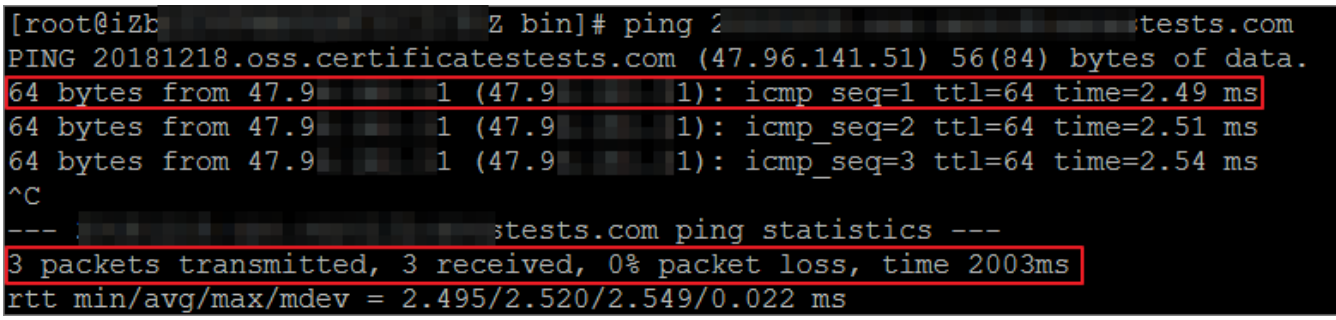
Web服务器：Tomcat 8.5或9

> 说明:
> Tomcat服务器需要提前安装JDK环境变量，请前往Tomcat官网查看推荐的JDK兼容配置。

前提条件

- 已从阿里云SSL证书服务控制台下载Tomcat服务器证书（包含PFX格式证书文件和TXT格式密码文件）。
- 您申请SSL证书时绑定的域名已完成DNS解析、实现了该域名指向您Tomcat服务器的IP地址。

  域名解析设置完成后执行`ping www.yourdomain.com`命令，如果返回了您所设置解析的主机IP地址，说明解析成功。

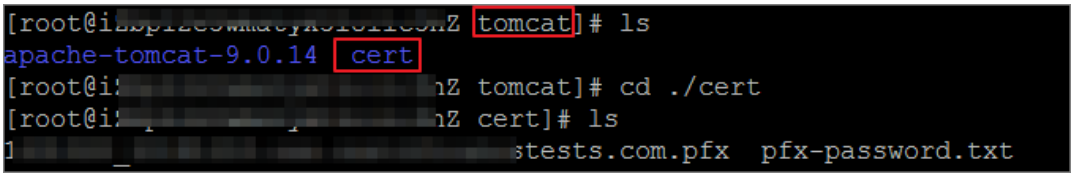


操作步骤

1. 解压Tomcat证书。

   > 说明:
   > 每次下载证书都会产生新的密码，该密码仅匹配本次下载的证书。如果需要更新证书文件，同时也要更新匹配的密码。

2. 在Tomcat安装目录下新建cert目录，将下载的证书和密码文件拷贝到cert目录下。



3. 打开*Tomcat/conf/server.xml*，在*server.xml*文件中找到以下参数并进行修改。

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

 #找到以上参数，去掉<!- - 和 - ->这对注释符并修改为如下参数，对HTTPS默认端口进行配置：
 <Connector port="80" protocol="HTTP/1.1"   #将Connector port修改为80。
               connectionTimeout="20000"
               redirectPort="443" />    #将redirectPort修改为SSL默认端口443，让HTTPS请求转发到443端口。
```

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

    #找到以上参数，去掉<!- - 和 - ->这对注释符并修改为如下参数：
    <Connector port="443"   #将Tomcat中默认的HTTPS端口Connector port 8443修改为443。8443端口不可通过域名直接访问、需要在域名后加上端口号；443端口是HTTPS的默认端口，可通过域名直接访问，无需在域名后加端口号。
          protocol="org.apache.coyote.http11.Http11NioProtocol"   #server.xml文件中Connector port有两种运行模式（NIO和APR），请选择NIO模式（也就是protocol="org.apache.coyote.http11.Http11NioProtocol"）这一段进行配置。
          maxThreads="150"
          SSLEnabled="true">
        <SSLHostConfig>
           <Certificate       certificateKeystoreFile="/usr/local/tomcat/cert/证书域名.pfx"   #此处certificateKeystoreFile代表证书文件的路径，请用您证书的路径+文件名替换证书域名.pfx，例如：certificateKeystoreFile="/usr/local/tomcat/cert/abc.com.pfx"
             certificateKeystorePassword="证书密码"   #此处certificateKeystorePassword为SSL证书的密码，请用您证书密码文件pfx-password.txt中的密码替换，例如：certificateKeystorePassword="bMNML1Df"
             certificateKeystoreType="PKCS12" />   #证书类型为PFX格式时，certificateKeystoreType修改为PKCS12。
        </SSLHostConfig>
    </Connector>
```

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

#找到以上参数，去掉<!- - 和 - ->这对注释符并修改为如下参数：
<Connector port="8009" protocol="AJP/1.3" redirectPort="443" />  #将redirectPort修改为443，让HTTPS请求转发到443端口。
```

4. （可选步骤）在server.xml文件最底部添加以下内容，实现HTTP自动跳转为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>
```

5. 保存server.xml文件配置。
6. 重启Tomcat服务。
   a. 在Tomcat下的bin目录中执行./shutdown.sh关闭Tomcat服务。

```
[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. 在Tomcat下的bin目录中执行./startup.sh开启Tomcat服务。

```
[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.
```

后续操作

Tomcat服务重启成功后，您可在浏览器中输入您SSL证书绑定的域名*https://www.YourDomainName.com*验证证书安装结果。浏览器地址栏显示绿色的小锁标识说明证书安装成功。

安装证书相关文档：

- 在Tomcat服务器上安装SSL证书
- 在Apache服务器上安装SSL证书
- Ubuntu系统Apache 2部署SSL证书
- 我获取到的数字证书如何配置在自己的Apache中?
- 在Nginx/Tengine服务器上安装证书
- 在IIS服务器上安装证书
- Jetty服务器配置SSL证书