ALIBABA CLOUD

阿里云

SSL证书服务

证书安装

文档版本：20220315

阿里云

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
危险	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。	危险 重置操作将丢失用户配置数据。
警告	该类警示信息可能会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，恢复业务时间约十分钟。
注意	用于警示信息、补充说明等，是用户必须了解的内容。	注意 权重设置为0，该服务器不会再接受新请求。
说明	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。	单击**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。	在**结果确认**页面，单击**确定**。
Courier字体	命令或代码。	执行 `cd /d C:/window` 命令，进入Windows系统文件夹。
斜体	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。	`switch {active|stand}`

目录

1.SSL证书安装指南

通过数字证书管理服务购买并签发SSL证书后，您可以将已签发的证书下载到本地，并根据需要将证书安装到要使用证书的环境。阿里云SSL证书适用于安装到Web服务器、部署到支持的阿里云产品。

前提条件

已通过数字证书管理服务购买并签发SSL证书。相关操作，请参见选择购买方式购买SSL证书、提交证书申请。

证书安装场景

场景	说明	操作流程概览
安装到Web服务器	表示在提供Web服务的服务器上配置SSL证书，并开启HTTPS监听，实现客户端与服务端之间的HTTPS通信。 不同类型的Web服务器支持配置的证书格式不同。为了便于您安装证书，数字证书管理服务提供了适用于各种主流Web服务器（例如，Nginx、Spring Boot、Apache Tomcat、Apache（httpd）、Internet Information Services）的证书压缩包，供您直接下载使用（无需手动转换证书格式）。	1. 通过数字证书管理服务控制台下载已签发的证书到本地。 您可以根据Web服务器的类型，下载对应格式的证书文件。具体操作，请参见下载证书到本地。 2. 将下载的证书文件上传到Web服务器，并修改服务器的相关配置，开启HTTPS监听。 不同Web服务器需要修改的配置不同，数字证书管理服务提供了主流Web服务器安装SSL证书的方法介绍。更多信息，请参见在服务器上安装证书。
部署到阿里云产品	表示通过数字证书管理服务控制台，将SSL证书快速应用到阿里云产品的指定资源上，简化证书配置操作。 目前只有以下阿里云产品支持通过数字证书管理服务部署证书：**Web应用防火墙（WAF）、对象存储OSS、传统负载均衡CLB（原SLB）、应用负载均衡ALB、内容分发网络（CDN）、安全加速、全站加速、视频直播、DDoS高防、视频点播API网关、全球加速GA、函数计算、云服务器ECS**。	通过数字证书管理服务控制台的**部署**操作，将已签发的证书一键应用到指定的阿里云资源上。具体操作，请参见部署证书到阿里云产品。

下载证书到本地

如果您已经通过数字证书管理服务购买并签发了SSL证书，可以执行以下步骤，将已签发的阿里云SSL证书下载到本地。

> **注意** 为了保证数据安全性，您上传到数字证书管理服务进行统一管理的第三方证书不支持下载。

1. 登录数字证书管理服务控制台。
2. 在左侧导航栏，单击**SSL证书**。
3. 通过单击对应页签，选择要操作的证书类型：
   - **证书管理**页签：表示操作付费版SSL证书。

- **免费证书**页签：表示操作免费版SSL证书。

不同类型证书的下载操作相同，下文操作描述以下载付费版SSL证书为例。

4. 在证书列表，定位到要下载的证书，单击**操作**列下的**下载**。

> **说明** 只有状态为**已签发**、**即将过期**、**已过期**的证书支持**下载**操作，否则证书**操作**列下不会显示**下载**按钮。

5. 在**证书下载**面板，下载适用于您的Web服务器的证书。

数字证书管理服务已经将您的证书自动转换成适用于不同Web服务器的格式并压缩，您只需根据服务器类型，单击对应的**下载**按钮，即可将满足该类型服务器配置需求的证书压缩包下载到本地。

证书下载示例：

- 如果您的Web服务器类型是Apache Tomcat（支持配置PFX格式、JKS格式的证书）、Spring Boot，根据要配置的证书类型进行操作：
  - 单击**Tomcat**后的**下载**，下载PFX格式的证书压缩包。
  - 单击**JKS**后的**下载**，下载JKS格式的证书压缩包。
- 如果您的Web服务器类型是Apache（httpd）、Nginx、Internet Information Services（IIS），分别单击**Apache**、**Nginx**、**IIS**后的**下载**。
- 如果您的Web服务器类型不在Apache Tomcat、Apache（httpd）、Nginx、IIS范围，单击**其他**后的**下载**。
- 如果您需要将证书安装到阿里云产品上，推荐您使用证书部署功能。如果要安装证书的阿里云产品暂不支持证书部署，建议您下载适用于Nginx服务器的证书压缩包（即单击**Nginx**后的**下载**）。
- 如果您需要在App、Java等客户端上安装根证书，单击**根证书下载**后的**下载**。

> **说明** 对于通过客户端浏览器访问的Web服务，无需关注根证书，因为根证书已经内置在客户端浏览器。

完成下载后，证书压缩包将会被下载到当前浏览器的默认下载目录。您可以在下载目录中查看已下载的证书压缩包，并解压获得对应的证书文件。

在服务器上安装证书

通过数字证书管理服务控制台下载证书到本地后，您还需要将已下载的证书上传到Web服务器并修改服务器的相关配置，才能使SSL证书生效。

不同Web服务器安装SSL证书的具体操作不同。以下内容介绍了在主流Web服务器上安装SSL证书的方法，供您参考。

> **说明** 如果您的Web服务器类型不在以下范围，或者您不熟悉Web服务器配置操作，您可以登录数字证书管理服务控制台，将鼠标移动至左侧导航栏下方**有问题？找专家！**处，使用钉钉扫描二维码加入钉钉技术支持群，获取操作指导。

Web服务器类型	证书安装方法
Nginx、Tengine	在Nginx（或Tengine）服务器上安装证书
Spring Boot	在Spring Boot上启用HTTPS
Apache Tomcat 7（及以下版本）	• 安装PFX格式证书 • 安装JKS格式证书
Apache Tomcat 8（及以上版本）	在CentOS系统Tomcat 8.5或9上部署SSL证书
Apache（httpd）	在Apache服务器上安装SSL证书
Apache 2	在Ubuntu系统Apache 2部署SSL证书
IIS	在IIS服务器上安装SSL证书
Jetty	在Jetty服务器上安装SSL证书
GlassFish	在GlassFish服务器上安装SSL证书

2.安装证书到服务器

2.1. 在Spring Boot上启用HTTPS

您可以通过配置SSL证书为Spring Boot启用HTTPS，实现网络通信数据的加密传输。本文介绍在Spring Boot上启用HTTPS的具体步骤。

前提条件

- SSL证书的加密算法为RSA或ECC，并且证书为**已签发**状态。

  > **说明** 如果您证书的加密算法为SM2（国密算法），该证书无法配置在Spring Boot上。您需要吊销该证书并重新申请加密算法为RSA或ECC的证书。关于吊销证书和申请证书的操作，请参见吊销SSL证书和提交证书申请。

- 您的Spring Boot已经开启了443端口（HTTPS服务的默认端口）。
- 已准备好远程登录工具（例如PuTTY、Xshell），用于登录您的Web应用服务器。

操作步骤

1. 登录SSL证书控制台。
2. 在左侧导航栏，单击**SSL证书**。
3. 定位到要下载的证书，单击**操作**列下的**下载**。
4. 在**证书下载**面板，单击**Tomcat**服务器**操作**列下的**下载**。

   > **说明** Spring Boot支持配置PFX和JKS格式的证书。单击**Tomcat操作**列的**下载**即可下载PFX格式证书，本文以PFX格式证书为例介绍如何在Spring Boot上启用HTTPS。如果您需要配置JKS格式的证书，请单击**JKS操作**列的**下载**。

   该操作会将Tomcat服务器证书压缩包下载到本地，并保存在浏览器的默认下载位置。
5. 访问浏览器的默认下载目录，解压已下载的证书压缩包文件。
   解压后您将会获得以下文件。

   - 证书文件（*domain name.pfx*）
   - 密码文件（*pfx-password.txt*）

   > **说明**
   > - 本文中证书名称以*domain name*为示例。
   > - 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。

6. 登录您的Spring Boot应用服务器。
7. 将解压后的证书文件和密码文件拷贝到Spring Boot项目的根目录*src/main/resources/*下。

> **说明** 如果您修改过Spring Boot项目的目录，您需要将证书文件和密码文件拷贝到与配置文件*application.properties*或*application.yml*相同的目录下。

8. 修改配置文件*application.properties*或*application.yml*。
   - 参考以下示例配置*application.properties*中的参数：

```
server.port = 443    #HTTPS协议默认端口号为443，需要使用其他端口时，您可以在此处自定义。
server.ssl.key-store: classpath = <domain name.pfx>    #您需要使用实际的证书名称替换domain name.pfx。
server.ssl.key-store-password = ********    #填写pfx-password.txt文件内的密码。
server.ssl.keyStoreType = PKCS12
```

   - 参考以下示例配置*application.yml*中的参数：

```
server:
  port: 443    #HTTPS协议默认端口号为443，需要使用其他端口时，您可以在此处自定义。
  ssl:
    key-alias: tomcat
    key-store-password: ********    #填写pfx-password.txt文件内的密码。
    key-store-type: PKCS12
    key-store: classpath:<domain name.pfx>    #您需要使用实际的证书名称替换domain name.pfx。
```

9. 执行 `mvn spring-boot:run` 命令重启Spring Boot服务。

后续步骤

验证是否已启用HTTPS协议。

配置完成后，您可通过访问证书绑定的域名验证是否已启用HTTPS协议。

```
https://yourdomain    #需要将yourdomain替换成证书绑定的域名。
```

- 如果网页地址栏出现小锁标志，表示已启用HTTPS协议。
- 如果无法通过HTTPS正常访问网站，请确认您服务器的443端口是否已开启。如果443端口正常开启，仍无法通过HTTPS正常访问网站，您可以提交工单处理。

2.2. 在Apache服务器上安装SSL证书

阿里云SSL证书服务支持下载证书安装到Apache服务器，从而使Apache服务器支持HTTPS安全访问。本文介绍了证书安装的具体操作。

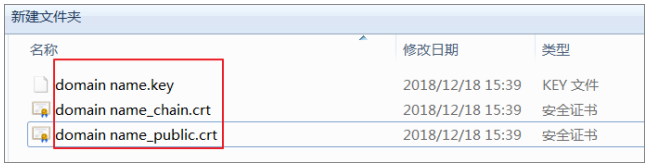
前提条件

- 已在数字证书管理服务控制台完成证书的签发和下载。具体操作，请参见下载证书到本地。
- 您的Apache服务器上已经开启了443端口（HTTPS服务的默认端口）。
- 您的Apache服务器上已安装了*mod_ssl.so*模块（启用SSL功能）。

操作步骤

> **注意** 本文档证书名称以domain name为示例，例如：证书文件名称为domain name_public.crt，证书链文件名称为domain name_chain.crt，证书密钥文件名称为domain name.key。

1. 解压已下载保存到本地的Apache证书文件。
   解压后的文件夹中有3个文件：
   - 证书文件：以.crt为后缀或文件类型。
   - 证书链文件：以.crt为后缀或文件类型。
   - 密钥文件：以.key为后缀或文件类型。



> **说明**
> - 申请证书时如果**CSR生成方式**选择的是**手动填写**或**选择已有的CSR**，未选择**系统生成**，则证书下载压缩包中将不包含.key文件（密钥文件）。
> - *.crt*扩展名的证书文件采用Base64-encoded的PEM格式文本文件，可根据需要修改成*.pem*等扩展名。关于证书格式转换的具体操作，请参见如何转换证书格式？。

2. 在Apache安装目录中新建*cert*目录，并将解压的Apache证书、证书链文件和密钥文件拷贝到*cert*目录中。
   如果需要安装多个证书，需在Apache安装目录中新建对应数量的*cert*目录，用于存放不同的证书 。

> **说明** 如果申请证书时**CSR生成方式**选择了**手动填写**，请将手动生成创建的密钥文件拷贝到*cert*目录中并命名为*domain name.key*。

3. 修改*httpd.conf*配置文件。
   i. 在Apache安装目录*Apache/conf/*下，打开*httpd.conf*文件。

> **说明** *Apache/conf/*为Apache的默认安装目录，如果修改过该路径，您需要在修改后的路径下查找*httpd.conf*文件。

   ii. 在*httpd.conf*文件中找到以下参数，按照下文中注释内容进行配置。

```
#LoadModule ssl_module modules/mod_ssl.so  #删除行首的配置语句注释符号"#"加载mod_ssl.so模块启用SSL服务，Apache默认是不启用该模块的。
#Include conf/extra/httpd-ssl.conf  #删除行首的配置语句注释符号"#"。
```

> **说明** 如果您在*httpd.conf*文件中没有找到以上参数，请确认您的Apache服务器中是否已经安装*mod_ssl.so*模块。可执行 `yum install -y mod_ssl` 命令安装*mod_ssl.so*模块。

   iii. 保存*httpd.conf*文件并退出。
4. 修改*httpd-ssl.conf*配置文件。

i. 在*Apache/conf/extra/*目录下，打开*httpd-ssl.conf*文件。

> **说明** 根据操作系统的不同，*http-ssl.conf*文件也可能存放在*conf.d/ssl.conf/*目录下。

ii. 在*httpd-ssl.conf*文件中找到以下参数，按照下文中注释内容进行配置。

```
<VirtualHost *:443>
    ServerName   #修改为申请证书时绑定的域名www.YourDomainName1.com。

    DocumentRoot  /data/www/hbappserver/public
    SSLEngine on
    SSLProtocol all -SSLv2 -SSLv3 # 添加SSL协议支持协议，去掉不安全的协议。
    SSLCipherSuite HIGH:!RC4:!MD5:!aNULL:!eNULL:!NULL:!DH:!EDH:!EXP:+MEDIUM   # 修改加密套件。
    SSLHonorCipherOrder on
    SSLCertificateFile cert/domain name1_public.crt   # 将domain name1_public.crt替换成您证书文件名。
    SSLCertificateKeyFile cert/domain name1.key   # 将domain name1.key替换成您证书的密钥文件名。
    SSLCertificateChainFile cert/domain name1_chain.crt  # 将domain name1_chain.crt替换成您证书的密钥文件名；证书链开头如果有#字符，请删除。
</VirtualHost>
#如果证书包含多个域名，复制以上参数，并将ServerName修改为第二个域名。
<VirtualHost *:443>
    ServerName   #修改为申请证书时绑定的第二个域名www.YourDomainName2.com。

    DocumentRoot  /data/www/hbappserver/public
    SSLEngine on
    SSLProtocol all -SSLv2 -SSLv3 # 添加SSL协议支持协议，去掉不安全的协议。
    SSLCipherSuite HIGH:!RC4:!MD5:!aNULL:!eNULL:!NULL:!DH:!EDH:!EXP:+MEDIUM   # 修改加密套件。
    SSLHonorCipherOrder on
    SSLCertificateFile cert/domain name2_public.crt   # 将domain name2替换成您申请证书时的第二个域名。
    SSLCertificateKeyFile cert/domain name2.key   # 将domain name2替换成您申请证书时的第二个域名。
    SSLCertificateChainFile cert/domain name2_chain.crt  # 将domain name2替换成您申请证书时的第二个域名；证书链开头如果有#字符，请删除。
</VirtualHost>
```

> **注意** 请关注您的浏览器版本是否支持SNI功能。如果不支持，多域名证书配置将无法生效。

iii. 保存*httpd-ssl.conf*文件并退出。

5. （可选）修改*httpd.conf*文件，设置HTTP请求自动跳转HTTPS。

   在*httpd.conf*文件中的 `<VirtualHost *:80> </VirtualHost>` 中间，添加以下重定向代码。

```
RewriteEngine on
RewriteCond %{SERVER_PORT} !^443$
RewriteRule ^(.*)$ https://%{SERVER_NAME}$1 [L,R]
```

6. 重启Apache服务器使SSL配置生效。

在Apache的*bin*目录下执行以下步骤：

i. 执行 `apachectl -k stop` 停止Apache服务。

ii. 执行 `apachectl -k start` 开启Apache服务。

后续操作

证书安装完成后，您可通过访问证书的绑定域名验证该证书是否安装成功。

```
https://yourdomain    #需要将yourdomain替换成证书绑定的域名。
```

如果网页地址栏出现小锁标志，表示证书已经安装成功。

证书安装完成后，如果网站无法通过HTTPS正常访问，需确认您安装证书的服务器443端口是否已开启或被其他工具拦截。如果您使用的是阿里云ECS服务器，请前往ECS控制台**安全组**页面配置放行443端口。

2.3. Tomcat服务器安装SSL证书

2.3.1. 安装PFX格式证书

您可以将已签发的SSL证书下载并安装到Tomcat服务器。Tomcat服务器支持安装PFX格式和JKS两种格式的证书，您可以根据Tomcat版本选择要下载的证书格式。本文介绍了安装PFX格式证书的具体步骤。

前提条件

- 已登录您的Tomcat服务器。
- 您的Tomcat服务器上已经开启了443端口（HTTPS服务的默认端口）。
- 已安装OpenSSL工具。访问OpenSSL官网，下载并安装OpenSSL工具。
- 已下载Tomcat服务器所需要的证书文件。关于下载证书的具体操作，请参见下载证书到本地。

> **注意**
>
> - 如果您在提交证书申请时，未将**CSR生成方式**设置为**系统生成**，则您下载的证书压缩包中不包含TXT密码文件。您必须选择**其他**类型服务器，下载CRT格式的证书，并使用OpenSSL工具生成PFX格式的证书文件。
> - 如果您拥有其他证书，可使用OpenSSL工具将您的证书文件转化为PFX格式。

版本说明

本文以安装在Linux操作系统中的Tomcat 7为例。

操作步骤

1. 解压已下载保存到本地的Tomcat证书文件。

   解压后您将看到文件夹中有以下文件：

   - 证书文件（*domain name.pfx*）

     > **说明** 本文中证书名称以*domain name*为示例。

   - 密码文件（*pfx-password.txt*）

> **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，创建*cert*目录，将解压的证书和密码文件拷贝到*cert*目录下。

> **说明** Tomcat安装目录与您的服务器环境有关。您可以使用 `sudo find / -name *tomcat*` 命令，查询Tomcat的安装目录。

3. 修改配置文件*server.xml*（路径：*Tomcat安装目录/conf/server.xml*），并保存。

   您可以从以下方式中选择一种进行操作：

> **注意** 使用方式一配置SSL连接器时，Tomcat将自动为您选择SSL的实现方式。如果您按照方式一无法完成后续配置，可能是因为您的环境不支持自动选定的SSL实现方式。这种情况下，您可以根据环境属性，使用方式二手动指定SSL的实现方式。

   - 方式一：Tomcat服务器自动选择SSL的实现方式。

     修改SSL连接器的属性为以下内容：

```
<Connector port="443"   #port属性根据实际情况修改（HTTPS默认端口为443）。如果使用其他端口号，则您需要使用https://domain name:port的方式来访问您的网站。
    protocol="HTTP/1.1"
    SSLEnabled="true"
    scheme="https"
    secure="true"
    keystoreFile="Tomcat安装目录/cert/domain name.pfx" #证书名称前需加上证书的绝对路径，请使用您证书的文件名替换domain name。
    keystoreType="PKCS12"
    keystorePass="证书密码"  #请替换为密码文件pfx-password.txt中的内容。
    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"/>
```

   - 方式二：您手动指定SSL的实现方式。

     您可以在*server.xml*中移除以下代码的注释，指定使用JSSE实现方式：

```
<Connector
           protocol="org.apache.coyote.http11.Http11NioProtocol"
           port="443" maxThreads="200"
           scheme="https" secure="true" SSLEnabled="true"
           keystoreFile="Tomcat安装目录/cert/domain name.pfx" keystorePass="证书密码"
           clientAuth="false" sslProtocol="TLS"/>
```

4. （可选）配置*web.xml*文件，开启HTTP强制跳转HTTPS。

   在文件*</welcome-file-list>*后添加以下内容：

```
<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>项目名称</web-resource-name> #请将该参数替换为您的项目名称。
        <url-pattern>/*</url-pattern>
    </web-resource-collection>
    <user-data-constraint>
        <transport-guarantee>CONFIDENTIAL</transport-guarantee>
    </user-data-constraint>
</security-constraint>
```

5. 重启Tomcat服务。
   i. 执行以下命令，关闭Tomcat服务：

   ```
   ./shutdown.sh
   ```

   ii. 执行以下命令，开启Tomcat服务：

   ```
   ./startup.sh
   ```

后续操作

证书安装完成后，您可以通过访问证书绑定域名的方式验证证书是否安装成功：

```
https://domain name   #请将domain name替换成证书绑定的域名。
```

- 如果网页地址栏出现小锁标志，表示证书已安装成功。
- 如果无法通过HTTPS正常访问网站，请确认您安装证书的服务器的443端口是否已开启。

2.3.2. 安装JKS格式证书

阿里云SSL证书服务支持下载证书安装到Tomcat服务器上。Tomcat支持安装PFX格式和JKS两种格式的证书，您可以根据Tomcat版本选择要下载的证书格式。本文介绍了安装JKS格式证书的具体步骤。

前提条件

- 已登录您的Tomcat服务器。
- 您的Tomcat服务器上已经开启了443端口（HTTPS服务的默认端口）。
- 已安装OpenSSL工具。访问OpenSSL官网，下载并安装OpenSSL工具。
- 已下载Tomcat服务器所需要的证书文件。关于下载证书的具体操作，请参见下载证书到本地。

> 注意
> - 如果您在提交证书申请时，未将**CSR生成方式**设置为**系统生成**，则您下载的证书压缩包中不包含TXT密码文件。您必须选择**其他**类型服务器，下载CRT格式的证书，并使用OpenSSL工具生成PFX格式的证书文件。
> - 如果您拥有其他证书，可使用OpenSSL工具将您的证书文件转化为PFX格式。

版本说明

本文以安装在Linux操作系统中的Tomcat 7为例。

操作步骤

1. 解压已下载保存到本地的Tomcat证书文件。

   解压后您将看到文件夹中有以下文件：

   - 证书文件（*domain name.pfx*）

     > **说明** 本文中证书名称以*domain name*为示例。

   - 密码文件（*pfx-password.txt*）

     > **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. 将PFX格式的证书转换成JKS格式。

   i. 输入以下Java JDK命令：

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

   > **说明** Windows系统中，需在 `%JAVA_HOME%/jdk/bin` 目录下执行该命令。

   ii. 回车后输入PFX证书密码，即密码文件*pfx-password.txt*中的内容。

   > **说明** JKS证书密码等同于PFX证书密码。两个密码不同时会导致Tomcat重启失败。

3. 在**Tomcat**安装目录下新建**cert**目录，将转化后的证书文件和密码文件拷贝到**cert**目录下 。
4. 参考以下步骤修改配置文件*server.xml*。

   i. 访问*Tomcat安装目录/conf/server.xml*目录，打开*server.xml*文件。

ii. 去掉*server.xml*中以下内容的注释。

```
<Connector  port="8443"
protocol="HTTP/1.1"
  port="8443" SSLEnabled="true"
  maxThreads="150" scheme="https" secure="true"
  clientAuth="false" sslProtocol="TLS" />
```

iii. 参照以下内容修改 `<Connector port="443"` 标签内容。

```
<Connector port="443"   #port属性根据实际情况修改（HTTPS默认端口为443）。如果使用其他端口号，则您需要使用https://yourdomain:port的方式来访问您的网站。
    protocol="HTTP/1.1"
    SSLEnabled="true"
    scheme="https"
    secure="true"
    keystoreFile="Tomcat安装目录/cert/domain name.jks" #证书名称前需加上证书的绝对路径，请使用您证书的文件名替换domain name。
    keystorePass="证书密码"   #此处请替换为您证书密码文件pfx-password.txt中的内容。
    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"/>
```

iv. 保存*server.xml*文件。

5. （可选）配置*web.xml*文件，开启HTTP强制跳转HTTPS。

   在文件*</welcome-file-list>*后添加以下内容：

```
<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>项目名称</web-resource-name> #请将该参数替换为您的项目名称。
        <url-pattern>/*</url-pattern>
    </web-resource-collection>
    <user-data-constraint>
        <transport-guarantee>CONFIDENTIAL</transport-guarantee>
    </user-data-constraint>
</security-constraint>
```

6. 重启Tomcat服务。

   i. 执行以下命令，关闭Tomcat服务：

```
./shutdown.sh
```

   ii. 执行以下命令，开启Tomcat服务：

```
./startup.sh
```

后续操作

证书安装完成后，您可以通过访问证书绑定域名的方式验证证书是否安装成功：

```
https://domain name    #请将domain name替换成证书绑定的域名。
```

- 如果网页地址栏出现小锁标志，表示证书已安装成功。
- 如果无法通过HTTPS正常访问网站，请确认您安装证书的服务器的443端口是否已开启。

2.4. 在Nginx（或Tengine）服务器上安装证书

您可以将已签发的SSL证书安装到Nginx（或Tengine）服务器上。本文介绍了下载SSL证书并在Nginx（或Tengine）服务器上安装证书的具体操作。

前提条件

- 已经通过SSL证书服务完成证书签发。更多信息，请参见提交证书申请。
- 已准备好远程登录工具（例如PuTTY、Xshell），用于登录您的Web服务器。

版本说明

本文以CentOS 8操作系统、Nginx 1.14.1服务器系统为例进行说明。由于服务器系统版本不同，您在操作过程中使用的命令可能会略有区别。

步骤1：下载证书到本地

1. 登录SSL证书控制台。
2. 在左侧导航栏，单击**SSL证书**。
3. 定位到要下载的证书，单击**操作**列下的**下载**。
4. 在**证书下载**面板，定位到**Nginx**服务器，单击**操作**列下的**下载**。
   该操作会将Nginx服务器证书压缩包下载到本地，并保存在浏览器的默认下载位置。
5. 打开浏览器的默认下载位置，解压已下载的Nginx证书压缩包文件。
   解压后您将会获得以下文件：

> 注意 本文中出现证书文件名称的地方，统一使用*cert-file-name*为例进行描述。例如，本文中用到的证书文件为*cert-file-name.pem*，证书私钥文件为*cert-file-name.key*。在实际操作过程中，您必须使用真实的证书文件名称替换示例代码中的*cert-file-name*。获取证书文件名称的具体操作，请参见下载证书到本地。

   - PEM格式的证书文件。

     PEM格式的证书文件是采用Base64编码的文本文件，您可以根据需要将证书文件修改成其他格式。关于证书格式的更多信息，请参见主流数字证书都有哪些格式。
   - KEY格式的证书私钥文件。

> **注意** 如果您在申请证书时将**CSR生成方式**设置为**手动填写**，则下载的证书文件压缩包中不会包含KEY文件，您需要手动创建证书私钥文件。

步骤2：在Nginx服务器上安装证书

在Nginx独立服务器、Nginx虚拟主机上安装证书的具体操作不同，请根据您的实际环境，选择对应的安装步骤。

参考以下步骤，在Nginx独立服务器上安装证书：

1. 登录Nginx服务器。

   例如，您可以使用远程登录工具（例如，PuTTY、Xshell）登录服务器。

2. 执行以下命令，在Nginx安装目录（默认为*/usr/local/nginx/conf*）下创建一个用于存放证书的目录，将其命名为*cert*。

   ```
   cd /usr/local/nginx/conf  #进入Nginx默认安装目录。如果您修改过默认安装目录，请根据实际配置进行调整。
   mkdir cert  #创建证书目录，命名为cert。
   ```

3. 使用远程登录工具（例如，PuTTY、Xshell）附带的本地文件上传功能，将本地证书文件和私钥文件上传到Nginx服务器的证书目录（示例中为*/usr/local/nginx/conf/cert*）。

   > **注意** 如果您在申请证书时将**CSR生成方式**设置为**手动填写**，请将您手动创建的证书私钥文件上传至*/usr/local/nginx/conf/cert*目录。

4. 编辑Nginx配置文件（*nginx.conf*），修改与证书相关的配置内容。

   i. 执行以下命令，打开配置文件。

   > **注意** *nginx.conf*默认保存在*/usr/local/nginx/conf*目录下。如果您修改过*nginx.conf*的位置，请将 `/usr/local/nginx/conf/nginx.conf` 替换成修改后的位置。

   ```
   vim /usr/local/nginx/conf/nginx.conf
   ```

   ii. 按**i**键进入编辑模式。

iii. 在配置文件中定位到HTTP协议代码片段（`http{}`），并在HTTP协议代码里面添加以下server配置（如果server配置已存在，按照以下注释内容修改相应配置即可）。

使用示例代码前，请注意替换以下内容：

- *yourdomain*：替换成证书绑定的域名。

  如果您购买的是单域名证书，需要修改为单域名（例如`www.aliyundoc.com`）；如果您购买的是通配符域名证书，则需要修改为通配符域名（例如`*.aliyundoc.com`）。

- `cert-file-name.pem`：替换成您在步骤3上传的证书文件的名称。
- `cert-file-name.key`：替换成您在步骤3上传的证书私钥文件的名称。

```
#以下属性中，以ssl开头的属性表示与证书配置有关。
server {
    listen 443 ssl;
    #配置HTTPS的默认访问端口为443。
    #如果未在此处配置HTTPS的默认访问端口，可能会造成Nginx无法启动。
    #如果您使用Nginx 1.15.0及以上版本，请使用listen 443 ssl代替listen 443和ssl on。
    server_name yourdomain; #需要将yourdomain替换成证书绑定的域名。
    root html;
    index index.html index.htm;
    ssl_certificate cert/cert-file-name.pem;  #需要将cert-file-name.pem替换成已上传的证书文件的名称。
    ssl_certificate_key cert/cert-file-name.key; #需要将cert-file-name.key替换成已上传的证书私钥文件的名称。
    ssl_session_timeout 5m;
    ssl_ciphers ECDHE-RSA-AES128-GCM-SHA256:ECDHE:ECDH:AES:HIGH:!NULL:!aNULL:!MD5:!ADH:!RC4;
    #表示使用的加密套件的类型。
    ssl_protocols TLSv1.1 TLSv1.2 TLSv1.3; #表示使用的TLS协议的类型。
    ssl_prefer_server_ciphers on;
    location / {
        root html;  #Web网站程序存放目录。
        index index.html index.htm;
    }
}
```

iv. （可选）设置HTTP请求自动跳转HTTPS。

如果您希望所有的HTTP访问自动跳转到HTTPS页面，则可以在需要跳转的HTTP站点下添加以下 `rewrite` 语句。

> **注意** 以下代码片段需要放置在*nginx.conf*文件中 `server {}` 代码段后面，即设置HTTP请求自动跳转HTTPS后，*nginx.conf*文件中会存在两个 `server {}` 代码段。

```
server {
    listen 80;
    server_name yourdomain; #需要将yourdomain替换成证书绑定的域名。
    rewrite ^(.*)$ https://$host$1; #将所有HTTP请求通过rewrite指令重定向到HTTPS。
    location / {
        index index.html index.htm;
    }
}
```

> **警告** 如果您使用的是阿里云ECS服务器，必须在ECS管理控制台的**安全组**页面，配置放行80端口和443端口，否则网站访问可能出现异常。关于如何配置安全组，请参见添加安全组规则。

v. 修改完成后，按**Esc**键、输入**:wq!**并按**Enter**键，保存修改后的配置文件并退出编辑模式。

5. 执行以下命令，重启Nginx服务。

```
cd /usr/local/nginx/sbin  #进入Nginx服务的可执行目录。
./nginx -s reload  #重新载入配置文件。
```

如果重启Nginx服务时收到报错，您可以使用以下方法进行排查：

- 收到 `the "ssl" parameter requires ngx_http_ssl_module` 报错：您需要重新编译Nginx并在编译安装的时候加上 `--with-http_ssl_module` 配置。
- 收到 `"/cert/3970497_pic.certificatestests.com.pem":BIO_new_file() failed (SSL: error:02001002:system library:fopen:No such file or directory:fopen('/cert/3970497_pic.certificatestests.com.pem','r') error:2006D080:BIO routines:BIO_new_file:no such file)` 报错：您需要去掉证书相对路径最前面的 `/` 。例如，您需要去掉 `/cert/`*cert-file-name*`.pem` 最前面的 `/` ，使用正确的相对路径 `cert/`*cert-file-name*`.pem` 。

参考以下步骤，在Nginx虚拟主机上安装证书：

在不同的虚拟主机上安装证书，您需要执行不同的操作步骤。如果您使用的是阿里云的云虚拟主机，具体操作，请参见开启HTTPS加密访问。如果您使用的是其他品牌的虚拟主机，请参考对应的虚拟主机安装证书的操作指南。

步骤3：验证是否安装成功

证书安装完成后，您可通过访问证书的绑定域名验证该证书是否安装成功。

```
https://yourdomain   #需要将yourdomain替换成证书绑定的域名。
```

如果网页地址栏出现小锁标志，表示证书已经安装成功。

如果验证时出现访问异常，请参照下表进行排查。

异常现象	可能原因	处理方法
通过HTTPS无法正常访问您的网站。	安装证书的Nginx服务器的443端口未开放或被其他工具拦截。	• 如果您使用的是阿里云ECS服务器，请前往ECS管理控制台的**安全组**页面，配置开放443端口。 关于如何配置安全组，请参见添加安全组规则。 • 如果您使用的不是阿里云ECS服务器，请参照对应的服务器安全设置指南，配置开放服务器的443端口。
收到网站提示“您与网站之间的连接未完全安全”。	您的网站代码中调用的是HTTP协议。	您需要在网站代码中把HTTP协议修改为HTTPS协议。 **说明** 不同网站代码的实现逻辑可能存在差异，请您根据具体情况进行修改。如果需要更多支持，请提交工单。
收到网站提示“该网站未根据工信部相关法律进行备案”。	• 您的网站未完成备案，未在接入商处完成备案接入。 • 您的网站内容与备案信息不符、备案信息不准确、网站存在不适宜传播的内容等。	• 如果您使用的是阿里ICP云备案系统，请前往阿里云备案系统进行网站备案。 • 如果您使用的不是阿里云ICP备案系统，请前往备案服务商的系统进行网站备案。

2.5. 在IIS服务器上安装SSL证书

您可以在IIS（Internet Information Services）服务器上安装SSL证书，确保Web服务支持HTTPS安全访问。本文以安装在Windows Server 2012 R2操作系统上的IIS 8为例，介绍如何为IIS服务器安装SSL证书。

前提条件

- 您的Web服务器类型是IIS。

  如果您使用其他类型的Web服务器，请参见在服务器上安装证书，根据您的Web服务器类型，选择对应的证书安装指导。

- 您要安装的证书已通过CA中心审核并签发。

  在CA中心审核前，您需要先提交申请，详细信息，请参见提交证书申请。如果审核失败，请提交工单咨询。

步骤1：将SSL证书（IIS）下载到服务器

1. 连接到装有Windows Server 2012 R2操作系统的服务器。

   如果您使用ECS云服务器，您可以通过多种方式远程连接，详细信息，请参见连接方式概述ECS远程连接操作指南。

2. 将已签发的SSL证书（IIS）下载到服务器。

   说明 您也可以先将SSL证书（IIS）下载到任意一台计算机，然后将已下载的证书上传到服务器。

   i. 登录SSL证书控制台。

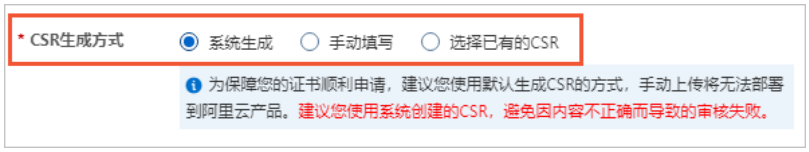
ii. 在左侧导航栏，单击**SSL证书**。

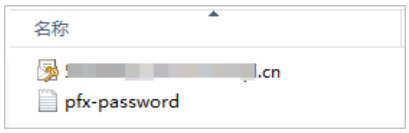
iii. 定位到已签发的SSL证书，单击**操作**列下的**下载**。

iv. 在**证书下载**面板，单击**IIS**服务器类型后的**下载**。

SSL证书（IIS）压缩包将会自动下载到当前浏览器的默认下载文件保存目录。

v. 解压缩已下载的SSL证书（IIS）压缩包。
根据您在提交证书申请时选择的**CSR生成方式**，解压缩获得的文件不同，具体如下表所示。关于CSR（Certificate Signing Request）的更多介绍，请参见申请证书时需要提交的信息。



CSR生成方式	证书压缩包包含的文件
系统生成或**选择已有的CSR**	包括以下文件（如下图所示）： ■ 证书文件（PFX格式）：以 *证书ID_证书绑定域名* 方式命名。 ■ 私钥文件（TXT格式）：名称为*pfx-password*，内容为证书的密码。 **注意** 每次下载证书时都会产生新的密码，该密码仅匹配本次下载的证书文件。 
手动填写	只包括证书文件（PEM格式），如下图所示。证书文件以 *证书ID_证书绑定域名*.pem 方式命名。 

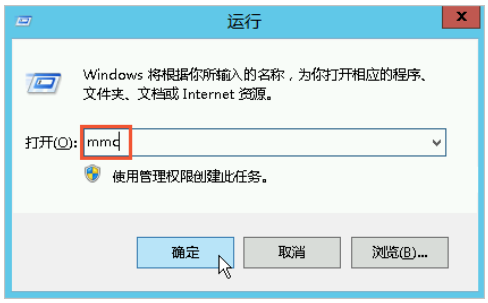
3. 如果您在上一步获得了PEM格式的证书文件，必须将证书文件（连同您手动生成CSR时获得的私钥文

件）格式转换成PFX格式；如果您已经获得PFX格式的证书文件，请跳过该步骤。

您可以使用OpenSSL工具转换证书格式。更多信息，请参见证书格式转换。

步骤2：导入证书

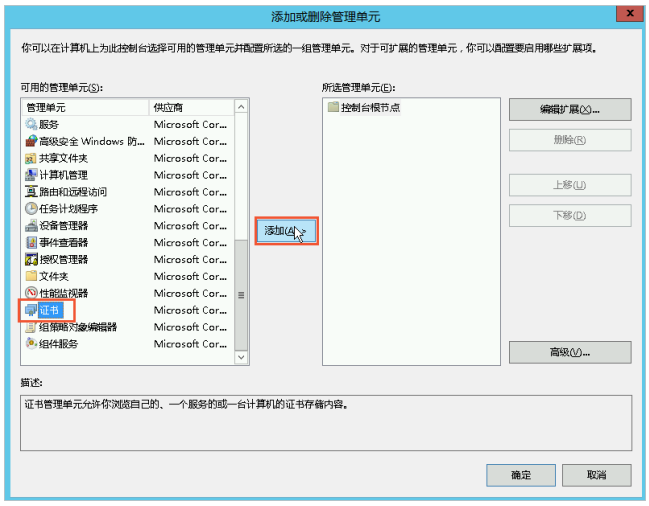
1. 在服务器按Win+R键，打开**运行**。
2. 输入*mmc*，单击**确定**。



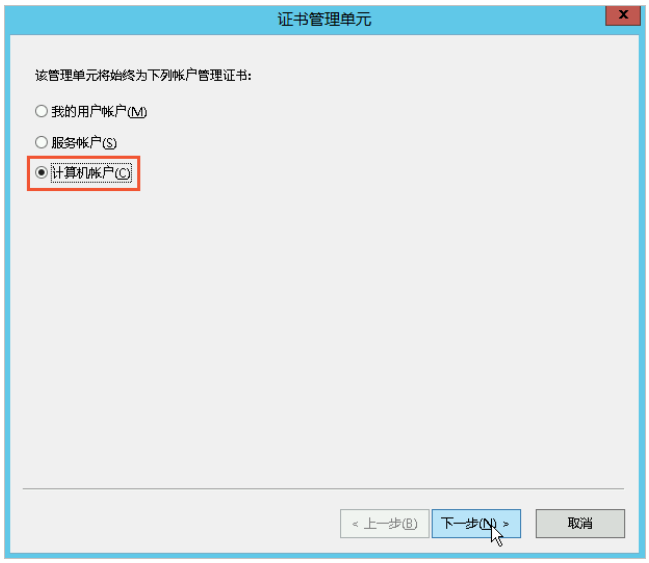
该操作将打开Windows服务器控制台（MMC，Microsoft Management Console）。

3. 为本地计算机添加证书管理单元。
   i. 在控制台的顶部菜单栏，选择**文件 > 添加/删除管理单元**。

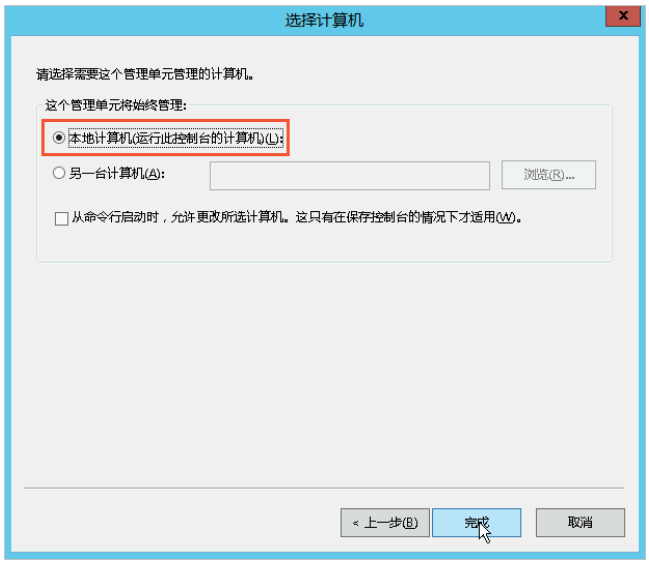
   ii. 在**添加或删除管理单元**对话框，从左侧**可用的管理单元**列表中选择**证书**，单击**添加**。



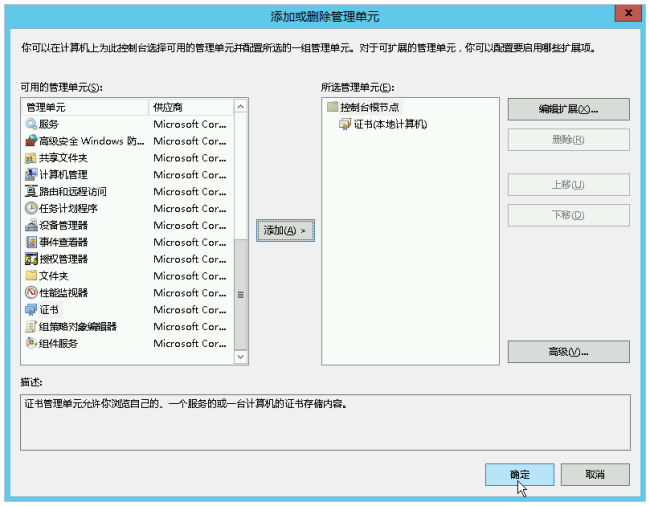
iii. 在**证书管理单元**对话框，选择**计算机账户**，单击**下一步**。



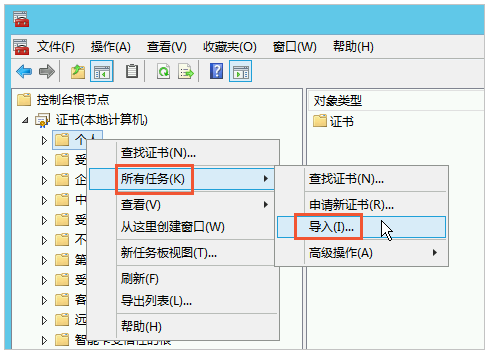
iv. 在**选择计算机**对话框，选择**本地计算机（运行此控制台的计算机）**，单击**完成**。



v. 在**添加或删除管理单元**对话框，单击**确定**。



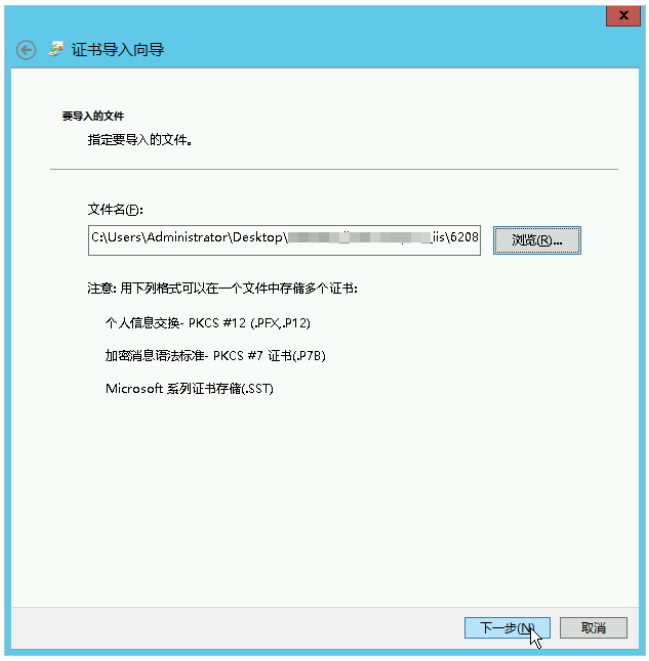
4. 在控制台左侧导航栏，展开**控制台根节点 > 证书（本地计算机）**，然后将光标放置在**个人**上并打开右键菜单，选择**所有任务 > 导入**。



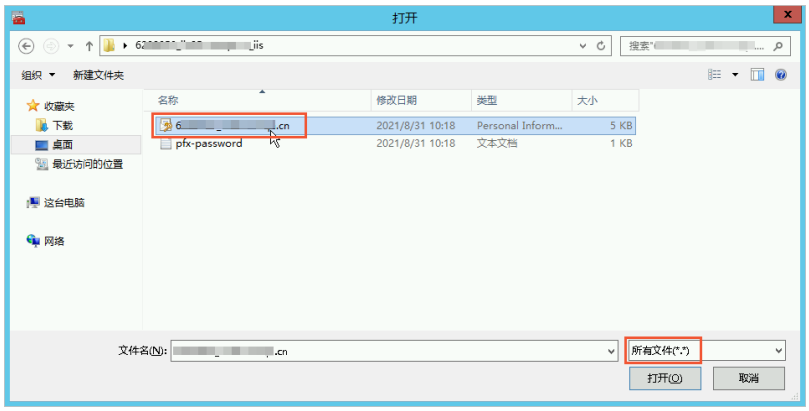
5. 完成证书导入向导。

   i. **欢迎使用证书导入向导**：单击**下一步**。

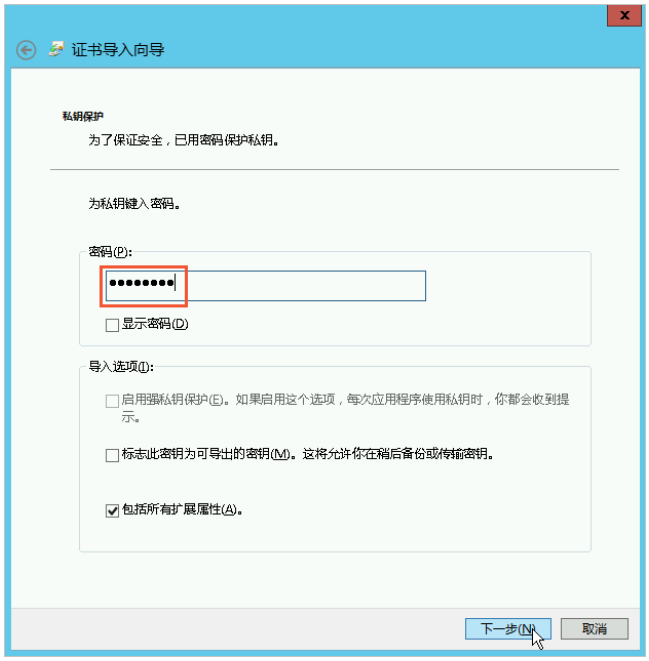
ii. **要导入的文件**：单击**浏览**，打开PFX格式的证书文件，单击**下一步**。



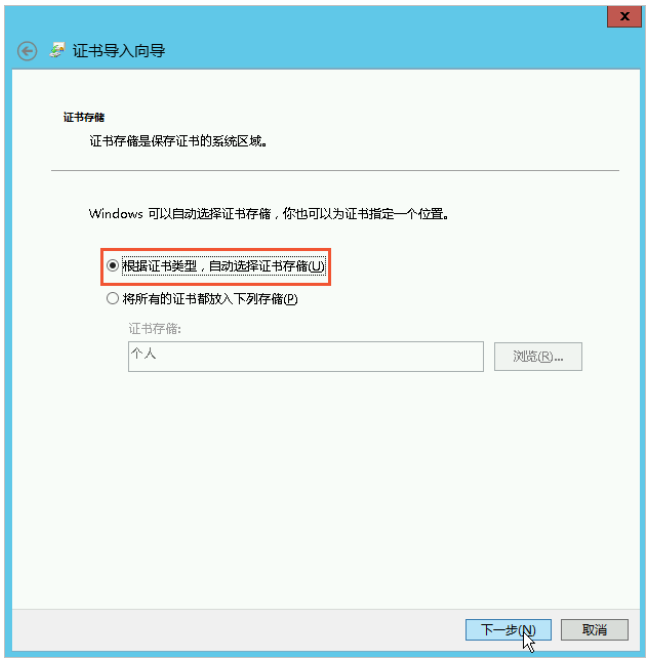
> **注意** 在打开文件时，您必须先将文件类型设置为**所有文件（*）**，然后再选择证书文件。



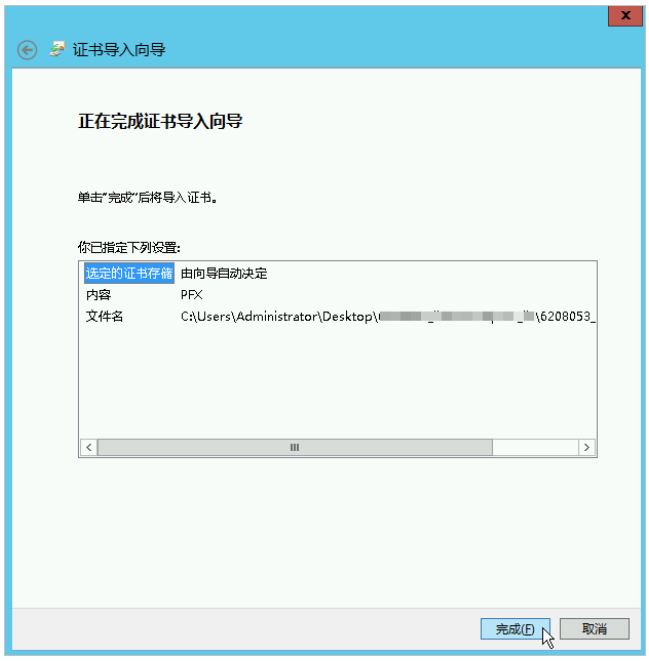
iii. **私钥保护**：打开TXT格式的私钥文件，复制文件内容，并将内容粘贴在**密码**输入框，单击**下一步**。



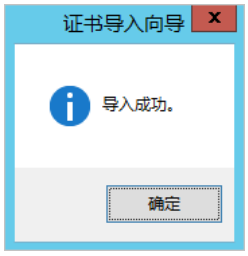
iv. **证书存储**：选择**根据证书类型，自动选择证书存储**，单击**下一步**。



v. **正在完成证书导入向导**：单击**完成**。

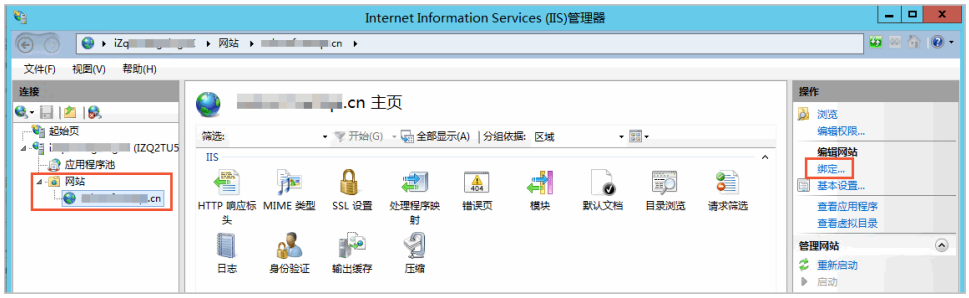


vi. 收到**导入成功**提示后，单击**确定**。

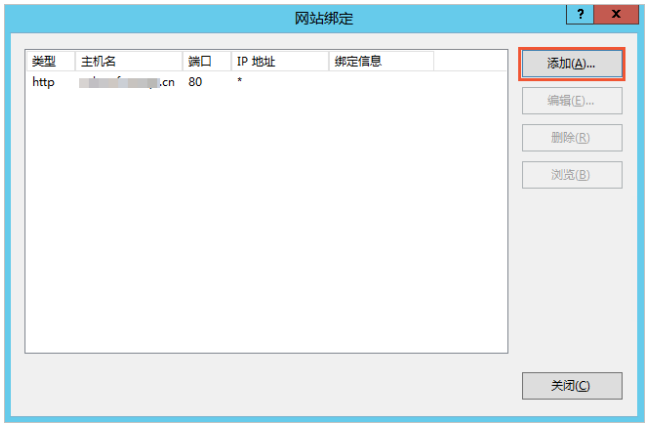


步骤3：为网站绑定证书

1. 打开IIS管理器。
2. 在左侧**连接**导航栏，展开主机，单击**网站**，选择对应的域名。
3. 在右侧**操作**导航栏，单击**绑定**。



4. 在**网站绑定**对话框，单击**添加**。



5. 在**添加网站绑定**对话框，完成网站的相关配置，并单击**确定**。

网站绑定的具体配置如下：

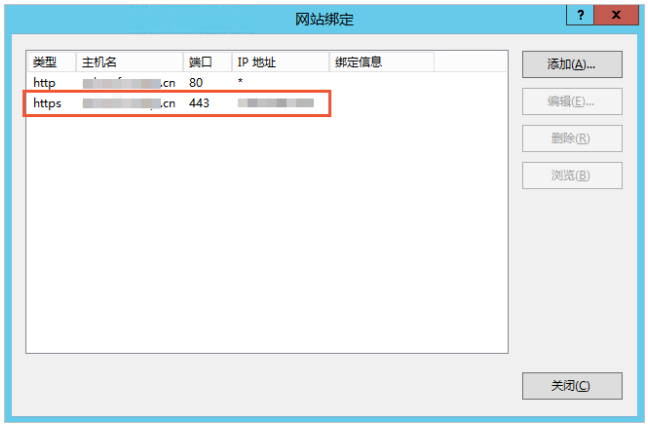
- **类型**：选择**https**。
- **IP地址**：选择服务器的IP地址。
- **端口**：默认为**443**，无需修改。

> **说明** 如果您设置了其他端口（例如8443），则网站用户通过浏览器访问网站时，必须在网站域名后输入端口号（以8443为例，用户必须在地址栏输入 `https://domain_name:8443` ）才能访问网站。使用默认端口443时无该限制，用户在浏览器地址栏输入 `https://domain_name` ，即可访问网站。

- **主机名**：填写网站域名。
- **SSL证书**：选择已导入的证书（alias）。

  alias是阿里云SSL证书的友好名称。如果您已导入多个阿里云SSL证书，可以单击**选择**，在**选择证书**对话框，通过域名搜索对应的证书。

完成配置后，您可以在**网站绑定**列表查看已添加的**https**类型网站绑定。



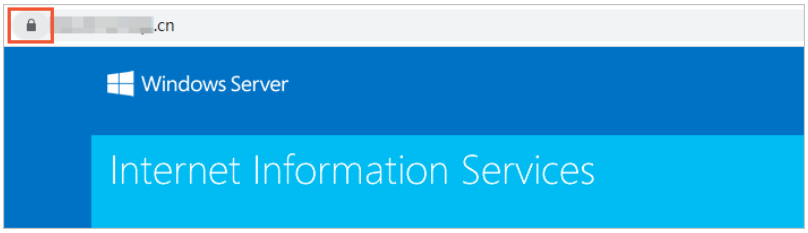
6. 在**网站绑定**对话框，单击**关闭**。

步骤4：验证证书在IIS上是否安装成功

打开计算机的浏览器，在地址栏输入安装的证书所绑定的域名，验证证书在IIS服务器上是否安装成功。

如果您能够获得响应且地址栏的前部出现

图标（如下图所示），表示成功建立了HTTPS连接，证书已经安装成功。



2.6. 在GlassFish服务器上安装SSL证书

阿里云数字证书管理服务支持下载证书并安装到GlassFish服务器上，本文介绍了证书安装的具体操作。

前提条件

已完成购买、申请和下载证书。

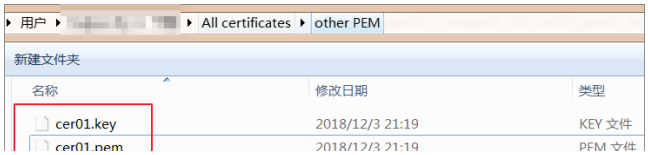
- 关于证书购买的具体操作，请参见购买SSL证书服务。
- 关于证书申请的具体操作，请参见提交证书申请。
- 您需要下载**服务器类型**为**其他**的证书。关于证书下载的具体操作，请参见下载证书到本地。

背景信息

本文中证书名称以**cer01**为示例，如证书文件名称为**cer01.pem**，证书密钥文件名称为**cer01.key**。

操作步骤

1. 解压已下载保存到本地的证书文件。您将看到文件中有一个证书文件（以.pem为后缀或文件类型）和一个密钥文件（以.txt为后缀或文件类型）。



2. 输入以下两行命令将证书和密钥文件转换成JKS格式。

```
openssl pkcs12 -export -in cer01.pem -inkey cer01.key -out temp.p12 -passout pass:changeit -name s1as
#请用您的证书名称替换命令行中cer01.pem、用您的证书密钥替换cer01.key。转换证书格式时设置的密码必须和您GlassFish服务器中自带的证书密码一致，该证书默认密码是changeit。
```

```
keytool -importkeystore -srckeystore temp.p12 -srcstoretype PKCS12 -srcstorepass changeit -deststoretype JKS -destkeystore ./GlassFish5/GlassFish/domains/domain1/config/keystore.jks -deststorepass changeit -alias s1as
#转换证书格式时设置的密码必须和您GlassFish服务器中自带的证书密码一致，该证书默认密码是changeit。
```

3. 重启GlassFish服务。

```
./Glassfish5/bin/asadmin restart-domain
```

后续步骤

证书安装完成后，您可通过访问证书的绑定域名验证该证书是否安装成功。

```
https://yourdomain    #需要将yourdomain替换成证书绑定的域名。
```

如果网页地址栏出现小锁标志，表示证书已经安装成功。

证书安装完成后，如果网站无法通过HTTPS正常访问，需确认您安装证书的服务器443端口是否已开启或被其他工具拦截。如果您使用的是阿里云ECS服务器，请前往ECS控制台**安全组**页面配置放行443端口。

2.7. 在Jetty服务器上安装SSL证书

阿里云SSL证书服务支持下载证书安装到Jetty服务器，从而使Jetty服务器支持HTTPS安全访问。本文介绍了证书安装的具体操作。

1. Jetty服务器版本确认。建议使用Jetty 9.2.22及以上版本。
2. 从阿里云下载*tomcat*格式的证书。非系统生成的*CSR*需要生成*pfx*证书密匙对文件，转换命令如下。

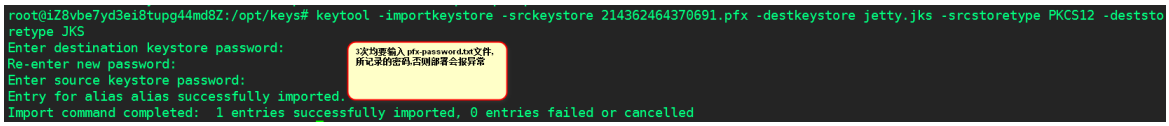
```
openssl pkcs12 -export -out 214362464370691.pfx -inkey 214362464370691.key -in 214362464370691.pem
```

3. 转换*pfx*的证书密匙对文件为*jks*格式，转换命令如下：

> **说明** Windows环境注意在*%JAVA_HOME%/jdk/bin*目录中执行。

```
keytool -importkeystore -srckeystore 密匙对文件.pfx -destkeystore 证书名称.jks -srcstoretype PKCS12 -deststoretype JKS
```

回车后输入两次要设置的*jks*格式证书密码，然后输入一次*pfx*证书密码。三次密码必须输入*pfx-password.txt*记录的密码。*jks*密码与*pfx*证书密码相同，否则可能会导致Jetty服务器启动失败。



4. 配置Jetty的SSL。

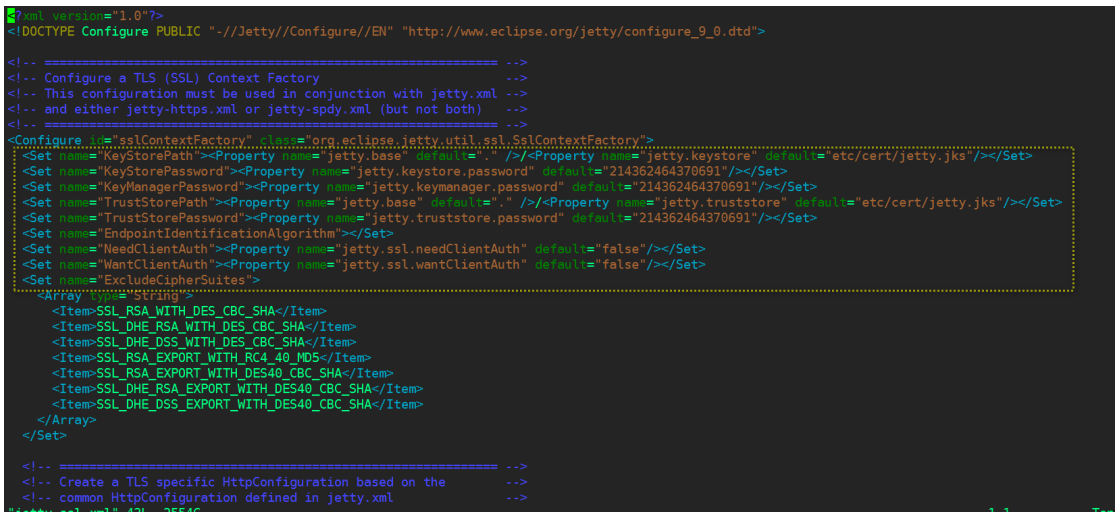
i. 确保Jetty的HTTP页面可正常访问。

ii. 拷贝证书。进入Jetty服务器目录下的etc，新建存放jks格式证书的目录，并复制jks格式证书至当前目录。

```
# pwd
/opt/jetty9222/etc
# mkdir cert
# cd cert/
# cp ../../../keys/jetty.jks .
# ls
jetty.jks
```

```
root@iZ8vbe7yd3ei8tupg44md8Z:/opt/jetty9222/etc# pwd
/opt/jetty9222/etc
root@iZ8vbe7yd3ei8tupg44md8Z:/opt/jetty9222/etc# mkdir cert
root@iZ8vbe7yd3ei8tupg44md8Z:/opt/jetty9222/etc# cd cert/
root@iZ8vbe7yd3ei8tupg44md8Z:/opt/jetty9222/etc/cert# cp ../../../keys/jetty.jks .
root@iZ8vbe7yd3ei8tupg44md8Z:/opt/jetty9222/etc/cert# ls
jetty.jks
```

iii. 编辑Jetty服务器目录中的etc中的*jetty-ssl.xml*文件，设置证书相关参数（密码设置均为*pfx-password.txt*所记录的密码）。



```
<?xml version="1.0"?>
<!DOCTYPE Configure PUBLIC "-//Jetty//Configure//EN" "http://www.eclipse.org/jetty/
configure_9_0.dtd">
<!-- ============================================================= -->
<!-- Configure a TLS (SSL) Context Factory                         -->
<!-- This configuration must be used in conjunction with jetty.xml -->
<!-- and either jetty-https.xml or jetty-spdy.xml (but not both)   -->
<!-- ============================================================= -->
<Configure id="sslContextFactory" class="org.eclipse.jetty.util.ssl.SslContextFacto
ry">
```

```
  <Set name="KeyStorePath"><Property name="jetty.base" default="." />/<Property nam
e="jetty.keystore" default="etc/cert/jetty.jks"/></Set>
  <Set name="KeyStorePassword"><Property name="jetty.keystore.password" default="21
4362464370691"/></Set>
<?xml version="1.0"?>
<!DOCTYPE Configure PUBLIC "-//Jetty//Configure//EN" "http://www.eclipse.org/jetty/
configure_9_0.dtd">
<!-- ============================================================= -->
<!-- Configure a TLS (SSL) Context Factory                         -->
<!-- This configuration must be used in conjunction with jetty.xml -->
<!-- and either jetty-https.xml or jetty-spdy.xml (but not both)   -->
<!-- ============================================================= -->
<Configure id="sslContextFactory" class="org.eclipse.jetty.util.ssl.SslContextFacto
ry">
  <Set name="KeyStorePath"><Property name="jetty.base" default="." />/<Property nam
e="jetty.keystore" default="etc/cert/jetty.jks"/></Set>
  <Set name="KeyStorePassword"><Property name="jetty.keystore.password" default="21
4362464370691"/></Set>
  <Set name="KeyManagerPassword"><Property name="jetty.keymanager.password" default
="214362464370691"/></Set>
  <Set name="TrustStorePath"><Property name="jetty.base" default="." />/<Property n
ame="jetty.truststore" default="etc/cert/jetty.jks"/></Set>
  <Set name="TrustStorePassword"><Property name="jetty.truststore.password" default
="214362464370691"/></Set>
  <Set name="EndpointIdentificationAlgorithm"></Set>
  <Set name="NeedClientAuth"><Property name="jetty.ssl.needClientAuth" default="fal
se"/></Set>
  <Set name="WantClientAuth"><Property name="jetty.ssl.wantClientAuth" default="fal
se"/></Set>
  <Set name="ExcludeCipherSuites">
    <Array type="String">
      <Item>SSL_RSA_WITH_DES_CBC_SHA</Item>
      <Item>SSL_DHE_RSA_WITH_DES_CBC_SHA</Item>
      <Item>SSL_DHE_DSS_WITH_DES_CBC_SHA</Item>
      <Item>SSL_RSA_EXPORT_WITH_RC4_40_MD5</Item>
      <Item>SSL_RSA_EXPORT_WITH_DES40_CBC_SHA</Item>
      <Item>SSL_DHE_RSA_EXPORT_WITH_DES40_CBC_SHA</Item>
      <Item>SSL_DHE_DSS_EXPORT_WITH_DES40_CBC_SHA</Item>
    </Array>
  </Set>
  <!-- ============================================================= -->
  <!-- Create a TLS specific HttpConfiguration based on the        -->
  <!-- common HttpConfiguration defined in jetty.xml               -->
  <!-- Add a SecureRequestCustomizer to extract certificate and    -->
  <!-- session information                                         -->
  <!-- ============================================================= -->
  <New id="sslHttpConfig" class="org.eclipse.jetty.server.HttpConfiguration">
    <Arg><Ref refid="httpConfig"/></Arg>
    <Call name="addCustomizer">
      <Arg><New class="org.eclipse.jetty.server.SecureRequestCustomizer"/></Arg>
    </Call>
  </New>
</Configure>
```

iv. 编辑Jetty服务器目录中的etc中的*jetty-https.xml*文件，配置HTTPS所使用的443端口。

```
<Call id="httpsConnector" name="addConnector">
  <Arg>
    <New class="org.eclipse.jetty.server.ServerConnector">
      <Arg name="server"><Ref refid="Server" /></Arg>
      <Arg name="acceptors" type="int"><Property name="ssl.acceptors" default="-1"/></Arg>
      <Arg name="selectors" type="int"><Property name="ssl.selectors" default="-1"/></Arg>
      <Arg name="factories">
        <Array type="org.eclipse.jetty.server.ConnectionFactory">
          <Item>
            <New class="org.eclipse.jetty.server.SslConnectionFactory">
              <Arg name="next">http/1.1</Arg>
              <Arg name="sslContextFactory"><Ref refid="sslContextFactory"/></Arg>
            </New>
          </Item>
          <Item>
            <New class="org.eclipse.jetty.server.HttpConnectionFactory">
              <Arg name="config"><Ref refid="sslHttpConfig"/></Arg>
            </New>
          </Item>
        </Array>
      </Arg>
      <Set name="host"><Property name="jetty.host" /></Set>
      <Set name="port"><Property name="https.port" default="443" /></Set>
      <Set name="idleTimeout"><Property name="https.timeout" default="30000"/></Set>
      <Set name="soLingerTime"><Property name="https.soLingerTime" default="-1"/></Set>
      <Set name="acceptorPriorityDelta"><Property name="ssl.acceptorPriorityDelta" default="0"/></Set>
      <Set name="selectorPriorityDelta"><Property name="ssl.selectorPriorityDelta" default="0"/></Set>
      <Set name="acceptQueueSize"><Property name="https.acceptQueueSize" default="0"/></Set>
    </New>
  </Arg>
```

```
<?xml version="1.0"?>
<!DOCTYPE Configure PUBLIC "-//Jetty//Configure//EN" "http://www.eclipse.org/jetty/
configure_9_0.dtd">
<!-- ============================================================= -->
<!-- Configure a HTTPS connector.                                  -->
<!-- This configuration must be used in conjunction with jetty.xml -->
<!-- and jetty-ssl.xml.                                            -->
<!-- ============================================================= -->
<Configure id="Server" class="org.eclipse.jetty.server.Server">
  <!-- ============================================================= -->
  <!-- Add a HTTPS Connector.                                        -->
  <!-- Configure an o.e.j.server.ServerConnector with connection     -->
  <!-- factories for TLS (aka SSL) and HTTP to provide HTTPS.        -->
  <!-- All accepted TLS connections are wired to a HTTP connection.-->
  <!--                                                               -->
  <!-- Consult the javadoc of o.e.j.server.ServerConnector,          -->
  <!-- o.e.j.server.SslConnectionFactory and                         -->
  <!-- o.e.j.server.HttpConnectionFactory for all configuration      -->
  <!-- that may be set here.                                         -->
  <!-- ============================================================= -->
  <Call id="httpsConnector" name="addConnector">
    <Arg>
      <New class="org.eclipse.jetty.server.ServerConnector">
        <Arg name="server"><Ref refid="Server" /></Arg>
        <Arg name="acceptors" type="int"><Property name="ssl.acceptors" default="-1
"/></Arg>
        <Arg name="selectors" type="int"><Property name="ssl.selectors" default="-1
"/></Arg>
        <Arg name="factories">
          <Array type="org.eclipse.jetty.server.ConnectionFactory">
            <Item>
              <New class="org.eclipse.jetty.server.SslConnectionFactory">
                <Arg name="next">http/1.1</Arg>
                <Arg name="sslContextFactory"><Ref refid="sslContextFactory"/></Arg
```

```
>
              </New>
            </Item>
            <Item>
              <New class="org.eclipse.jetty.server.HttpConnectionFactory">
                <Arg name="config"><Ref refid="sslHttpConfig"/></Arg>
              </New>
            </Item>
          </Array>
        </Arg>
        <Set name="host"><Property name="jetty.host" /></Set>
        <Set name="port"><Property name="https.port" default="443" /></Set>
        <Set name="idleTimeout"><Property name="https.timeout" default="30000"/></S
et>
        <Set name="soLingerTime"><Property name="https.soLingerTime" default="-1"/>
</Set>
        <Set name="acceptorPriorityDelta"><Property name="ssl.acceptorPriorityDelta
" default="0"/></Set>
        <Set name="selectorPriorityDelta"><Property name="ssl.selectorPriorityDelta
" default="0"/></Set>
        <Set name="acceptQueueSize"><Property name="https.acceptQueueSize" default=
"0"/></Set>
      </New>
    </Arg>
  </Call>
</Configure>
```

v. 编辑Jetty服务器目录中的*start.ini*文件，按需求更改端口号，并设置启动加载*jetty-https.xml*，*jetty-ssl.xml*。

```
jetty.port=80
jetty.dump.stop=
etc/jetty-ssl.xml
etc/jetty-https.xml
```

vi. 重启Jetty，验证HTTPS访问是否正常。

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
- 在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（或Tengine）服务器上安装证书
- 在IIS服务器上安装SSL证书
- 在CentOS系统Tomcat 8.5或9上部署SSL证书

2.8. 在Ubuntu系统Apache 2部署SSL证书

本文介绍了如何在Ubuntu系统以及Apache 2中安装阿里云SSL证书。

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。

环境准备

- 操作系统：Ubuntu
- Web服务器：Apache 2

操作步骤

1. 执行以下命令，在 `apache2` 目录下创建 `ssl` 目录。

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

2. 执行以下命令，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. 执行以下命令，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. 执行以下命令，修改SSL配置文件*default-ssl.conf*。

> **说明** *default-ssl.conf*文件可能存放在*/etc/apache2/sites-available*或*/etc/apache2/sites-enabled*目录中。*/sites-available*目录存放的是可用的虚拟主机，*/sites-enabled*目录存放的是已经启用的虚拟主机。

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

在*default-ssl.conf*文件中找到以下参数，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。
SSLCertificateFile /etc/apache2/ssl/YourDomainName_public.crt    #将/etc/apache2/ssl/YourDomainName_public.crt替换为证书文件路径+证书文件名。
SSLCertificateKeyFile /etc/ssl/apache2/YourDomainName.key    #将/etc/apache2/ssl/YourDomainName.key替换为证书密钥文件路径+证书密钥文件名。
SSLCertificateChainFile /etc/apache2/ssl/YourDomainName_chain.crt  #将/etc/apache2/ssl/YourDomainName_chain.crt替换为证书链文件路径+证书链文件名。
```

5. 执行以下命令，把*default-ssl.conf*映射至*/etc/apache2/sites-enabled*文件夹中建立软链接，实现两者之间的自动关联。

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

6. 执行以下命令，重新加载Apache 2配置文件。

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

```
root@          :~# sudo /etc/init.d/apache2 force-reload
[ ok ] Reloading apache2 configuration (via systemctl): apache2.service.
```

7. 执行以下命令，重启Apache 2服务。

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

```
root@          :~# sudo /etc/init.d/apache2 restart
[ ok ] Restarting apache2 (via systemctl): apache2.service.
```

8. 在浏览器中输入 `https://<YourDomainName>` ，验证证书安装结果。

   浏览器地址栏显示小锁标识说明证书安装成功。

2.9. 在CentOS系统Tomcat 8.5或9上部署SSL证书

本文介绍了CentOS系统下Tomcat 8.5或9部署SSL证书的具体操作。

前提条件

- 已从阿里云数字证书管理服务控制台下载Tomcat服务器证书（包含PFX格式证书文件和TXT格式密码文件）。关于下载证书的具体操作，请参见下载证书到本地。
- 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 <yourdomainName>**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。

```
[root@iZb          Z bin]# ping 2          tests.com
PING 20181218.oss.certificatestests.com (47.96.141.51) 56(84) bytes of data.
64 bytes from 47.9      1 (47.9      1): icmp_seq=1 ttl=64 time=2.49 ms
64 bytes from 47.9      1 (47.9      1): icmp_seq=2 ttl=64 time=2.51 ms
64 bytes from 47.9      1 (47.9      1): icmp_seq=3 ttl=64 time=2.54 ms
^C
---          stests.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 2.495/2.520/2.549/0.022 ms
```

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。

操作步骤

1. 解压已下载的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。需要更新证书文件时，需同步更新匹配的密码。

2. 将解压后的证书和密码文件拷贝到Tomcat的**conf**目录下。

   > **说明** 如果需要安装JKS格式证书，可使用以下命令将PFX格式证书转化成JKS格式。
   >
   > ```
   > keytool -importkeystore -srckeystore domain name.pfx -destkeystore domain name.jks
   > -srcstoretype PKCS12 -deststoretype JKS
   > ```

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. 保存*server.xml*文件配置。
5. （可选）在*web.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>
```

6. 执行 `./shutdown.sh` 和 `./startup.sh` 命令重启Tomcat服务。

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://<YourDomainName>*验证证书安装结果。浏览器地址栏显示小锁标识说明证书安装成功。

3.部署证书到阿里云产品

由于控制台功能调整，数字证书管理服务已不支持在数字证书管理服务控制台上将证书部署到阿里云产品中，本文档后期将不再维护并下线，请知悉。

如果您需要将已签发的SSL证书部署到阿里云产品，您可以前往对应云产品的控制台配置证书，也可以在数字证书管理服务控制台下载证书后，将证书安装到服务器中。关于安装SSL证书的更多信息，请参见SSL证书安装指南。

4.下载CA根证书和中间证书

为了保证客户端和服务端通过HTTPS成功通信，您在安装SSL证书时，也需要安装CA根证书和中间证书。本文介绍如何获取CA根证书和中间证书。

根证书

使用场景

如果您的业务用户通过浏览器访问您的Web业务，则您无需关注根证书，因为CA根证书已经内置在浏览器。您只需在Web服务器安装经CA签发的SSL证书，即可实现浏览器与服务端的HTTPS通信。关于安装SSL证书的相关操作，请参见SSL证书安装指南。

如果您的业务用户通过Java等客户端访问您的Web业务，由于客户端没有内置CA根证书，您可能需要在对应客户端手动安装CA根证书，保证客户端能够校验服务端的加密信息。例如，假设服务端安装了DigiCert OV型SSL证书，则客户端需要有DigiCert OV型根证书，才能实现客户端与服务端的HTTPS通信。

> **警告** 在客户端安装根证书以实现客户端校验，可能会因为根证书过期失效或策略变更等导致业务中断，我们不推荐您使用该方式。如果您确认已了解相关风险，且仍需要在App、Java客户端安装根证书，则可以参考下载地址，下载您需要的CA根证书。

相比于在客户端安装根证书以实现客户端校验，我们推荐您使用系统默认信任库进行客户端校验，并开启域名强校验来增加App的安全性。如果需要了解更多信息，请提交工单或在数字证书管理服务控制台左侧导航栏，单击**有问题？找专家！**，然后扫描二维码，进入钉钉服务群联系售后技术支持人员。

下载地址

目前只支持下载部分CA根证书，具体下载地址如下：

- DigiCert EV根证书
- DigiCert OV和DV根证书
- GlobalSign DV和OV根证书
- GlobalSign R1-R3交叉根
- WoSign RSA根证书
- WoSign SM2根证书

中间证书

为了避免客户端与服务端的HTTPS通信失败，您在安装SSL证书时，也需要安装中间证书。通常情况下，您下载的SSL证书已经包含了中间证书，所以您可以直接安装SSL证书。例如，当您下载Apache证书文件后，其中的*domain name_chain.crt*文件已经包含了中间证书。

> **注意** 如果您的SSL证书不包含中间证书或者您的中间证书已过期，请访问证书品牌的官网下载中间证书或者提交工单联系售后技术支持人员来协助您获得中间证书的下载链接。

5.常见问题

5.1. 如何转换证书格式?

不同Web服务器支持的证书格式不同。您需要将已签发的证书转换为适用当前Web服务器的格式，才能正常安装SSL证书。本文介绍如何转换证书格式。

您可参考以下方法实现证书格式之间的转换：

- 将JKS格式证书转换成PFX格式

  您可以使用JDK中自带的Keytool工具，将JKS格式证书文件转换成PFX格式。例如，您可以执行以下命令将*cert_name.jks*证书文件转换成*cert_name.pfx*证书文件：

  ```
  keytool -importkeystore -srckeystore D:\<cert_name>.jks -destkeystore D:\<cert_name>.pfx
          -srcstoretype JKS -deststoretype PKCS12
  ```

  > **说明**
  > - 本文证书名称以*cert_name*为示例，例如：证书文件名称为*cert_name*.pem，证书密钥文件名称为*cert_name*.key。您在实际使用过程中需要将*cert_name*替换成您的证书名称。
  > - Keytool工具是JDK中自带的密钥管理工具，可以制作Keystore（jks）格式的证书文件，您可以从 官方地址 下载JDK工具包来获取Keytool工具。该工具一般在*JDK\jre\bin\security*\目录下。

- 将PFX格式证书转换为JKS格式

  您可以使用JDK中自带的Keytool工具，将PFX格式证书文件转换成JKS格式。例如，您可以执行以下命令将*cert_name.pfx*证书文件转换成*cert_name.jks*证书文件：

  ```
  keytool -importkeystore -srckeystore D:\<cert_name>.pfx -destkeystore D:\<cert_name>.jks
          -srcstoretype PKCS12 -deststoretype JKS
  ```

- 将PEM/KEY/CRT格式证书转换为PFX格式

  您可以使用 OpenSSL工具，将KEY格式密钥文件、PEM或CRT格式公钥文件转换成PFX格式证书文件。例如，将您的KEY格式密钥文件*cert_name.key*和PEM格式公钥文件*cert_name.pem*拷贝至OpenSSL工具安装目录，使用OpenSSL工具执行以下命令将证书转换成*cert_name.jks*证书文件：

  ```
  openssl pkcs12 -export -out <cert_name>.pfx -inkey <cert_name>.key -in <cert_name>.pem
  ```

- 将PFX转换为PEM/KEY/CRT

  您可以使用 OpenSSL工具，将PFX格式证书文件转化为KEY格式密钥文件、PEM或CRT格式公钥文件。例如，将您的PFX格式证书文件*cert_name.jks*拷贝至OpenSSL安装目录，使用OpenSSL工具执行以下命令将证书转换成*cert_name.pem*证书文件和KEY格式密钥文件*cert_name.key*：

  ```
  openssl pkcs12 -in <cert_name>.pfx -nokeys -out <cert_name>.pem
  openssl pkcs12 -in <cert_name>.pfx -nocerts -out <cert_name>.key -nodes
  ```

5.2. 证书安装配置出错或网站无法访问怎么办?

如果您需要证书配置支持，或出现配置问题以及部署证书后无法访问网站的情况，可前往阿里云云市场，购买证书安装和检测技术服务，由第三方售后技术服务人员帮助您快速解决问题。

您也可以参考以下常见问题文档，排查并解决相关问题。

常见的证书安装与配置问题

- 苹果ATS证书的选择及配置
- 使用Google浏览器无法访问安装SSL证书后的IIS服务
- 在IIS部署服务证书后访问资源出现404报错
- 终端的浏览器提示证书不可信的排查方法
- Chrome浏览器出现“ERR_CERTIFICATE_TRANSPARENCY_REQUIRED”报错
- 安装云盾证书后谷歌浏览器无法访问的IIS服务

5.3. 为什么使用火狐浏览器访问已配置证书的网站提示不安全？

问题现象

证书配置完成后，使用Chrome浏览器访问网站显示正常，但是使用火狐浏览器访问网站提示不安全。

可能原因

您的服务器配置的加密算法较弱。

解决方法

建议您的网站使用推荐的加密套件 `ssl_ciphers ECDHE-RSA-AES128-GCM-SHA256:ECDHE:ECDH:AES:HIGH:!NULL:!aNULL:!MD5:!ADH:!RC4` 和协议 `ssl_protocols TLSv1.1 TLSv1.2 TLSv1.3` 。

5.4. 苹果ATS证书的选择及配置

自2017年01月01日起，根据苹果公司要求，所有iOS应用必须使用ATS（App Transport Security），即iOS应用内的连接必须使用安全的HTTPS连接。

> **说明** 阿里云CDN、SLB服务中的HTTPS配置完全符合ATS的要求。

证书配置要求

苹果ATS针对证书相关配置有如下四个方面的要求。

项目	具体要求
证书颁发机构	• 建议您使用DigiCert、GeoTrust品牌的OV或EV型数字证书。 • 对于个人用户，建议您使用DV型数字证书，不推荐使用免费证书。 • CFCA品牌的数字证书只在最新的苹果设备上才支持，因此不推荐您选择CFCA品牌。

项目	具体要求

证书的哈希算法和密钥长度	• 哈希算法：上述推荐的证书品牌使用的哈希算法都是SHA256或者更高强度的算法，符合ATS的要求。
	• 密钥长度：
	  ○ 如果您选择使用系统生成CSR的方式，系统生成的密钥采用的是2,048位的RSA加密算法，完全符合ATS的要求。
	  ○ 如果您选择手动填写CSR文件，请确保使用2,048位或以上的RSA加密算法。

传输协议	您Web服务器上的传输协议必须满足TLSv1.2，需要您在Web服务器上开启TLSv1.2，要求如下：
	• 基于OpenSSL环境的Web服务器，需要您使用OpenSSL 1.0及以上版本，推荐您使用OpenSSL 1.0.1及以上版本。
	• 基于Java环境的Web服务器，需要您使用JDK 1.7及以上版本。
	• 其他Web服务器，除IIS 7.5以及Weblogic 10.3.6比较特殊外，只要Web服务器版本满足要求，默认均开启TLSv1.2。
	Web服务器的详细配置要求如下：
	• Apache、Nginx Web服务器需要您使用OpenSSL 1.0及以上版本来支持TLSv1.2。
	• Tomcat 7及以上版本Web服务器需要您使用JDK 7.0及以上版本来支持TLSv1.2。
	• IIS 7.5 Web服务器默认不开启TLSv1.2，需要您修改注册表来开启TLSv1.2。
	  下载并导入ats.reg注册表脚本后，需要您重启或注销服务器，即可使TLSv1.2生效。
	• IBM Domino Server 9.0.1 FP3 Web服务器支持TLSv1.2。根据ATS要求，建议您使用IBM Domino Server 9.0.1 FP5版本。更多信息，请参见：
	  ○ IBM Notes and Domino wiki
	  ○ IBM HTTP SSL Server Questions and Answers
	• IBM HTTP Server 8.0及以上版本支持TLSv1.2。根据ATS要求，建议您使用IBM HTTP Server 8.5及以上版本。
	• Weblogic 10.3.6及以上版本Web服务器需要您使用Java 7及以上版本来支持TLSv1.2。
	  说明 Weblogic 10.3.6中存在多个SHA256兼容性问题，建议您使用Weblogic 12及以上版本，或者您需要为Weblogic 10.3.6配置前端Apache和Nginx的HTTPS代理或SSL前端负载。
	• Webspere V7.0.0.23及以上版本、Webspere V8.0.0.3及以上版本、Webspere V8.5.0.0及以上版本支持TLSv1.2。关于如何配置Webspere服务器支持TLSv1.2，请参见Configure websphere application server SSL protocol to TLSv1.2。

项目	具体要求
签字算法	签字算法必须满足如下算法要求： • TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 • TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 • TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 • TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA • TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 • TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA • TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 • TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 • TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 • TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 • TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA

配置示例

以下通过举例方式说明不同Web服务器的ATS协议及加密套件的配置方法。

> **注意** 本示例仅列举了与ATS协议有关的属性，请不要完全复制以下配置用于您的实际环境。

Nginx配置文件片段

Nginx配置文件中ssl_ciphers及ssl_protocols属性与ATS协议有关。

```
server {
ssl_ciphers ECDHE-RSA-AES128-GCM-SHA256:ECDHE:ECDH:AES:HIGH:!NULL:!aNULL:!MD5:!ADH:!RC4;
ssl_protocols TLSv1.1 TLSv1.2 TLSv1.3;
}
```

Tomcat配置文件片段

Tomcat配置文件中的SSLProtocol及SSLCipherSuite属性与ATS协议有关。

```
<Connector port="443" protocol="HTTP/1.1" SSLEnabled="true"
scheme="https" secure="true"
ciphers="TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256"
SSLProtocol="TLSv1.1+TLSv1.2+ TLSv1.3"
SSLCipherSuite="ECDHE-RSA-AES128-GCM-SHA256:ECDHE:ECDH:AES:HIGH:!NULL:!aNULL:!MD5:!ADH:!RC4
" />
```

IIS系列Web服务器的配置方法，请参见Enabling TLS 1.2 on IIS 7.5 for 256-bit cipher strength。您也可以使用可视化配置插件进行配置。具体操作，请参见IIS Crypto。

ATS检测工具

您可以在苹果电脑中使用系统自带的工具进行ATS检测，执行以下命令即可：`nscurl --ats-diagnostics --verbose 网址`。

5.5. 证书部署到云产品FAQ

由于控制台功能调整，数字证书管理服务已不支持在数字证书管理服务控制台上将证书部署到阿里云产品中，本文档后期将不再维护并下线，请知悉。

关于安装SSL证书的更多信息，请参见SSL证书安装指南。

5.6. 如何将证书应用到阿里云的产品中？

由于文档优化调整，本页面后期将不再维护并下线。请知悉。

关于证书部署的更多信息，请参见SSL证书安装指南。

5.7. 如何设置证书的TLS协议版本？

TLS协议版本包括TLS v1.0（已禁用）、TLS v1.1、TLS v1.2和TLS v1.3。TLS协议版本越高，HTTPS通信的安全性越高，但是相较于低版本TLS协议，高版本TLS协议对浏览器的兼容性较差。您可以根据业务需要，在安装证书的Web服务器或阿里云产品上设置证书的TLS协议版本。

如果您的证书安装在Web服务器上，请在Web服务器的证书配置文件中找到 `ssl_protocols` ，根据实际需要进行修改。例如，如果您的证书只支持TLS v1.1和TLS v1.2，您需要将 `ssl_protocols` 参数配置为 `TLSv1.1 TLSv1.2` ；如果您的证书需要支持TLS v1.3，您需要在 `ssl_protocols TLSv1.1 TLSv1.2` 中增加 `TLSv1.3` 。

如果您的证书部署在以下阿里云产品，请参考以下链接设置证书的TLS协议版本：

- DDoS高防：自定义TLS安全策略
- Web应用防火墙：自定义TLS配置
- SLB：TLS安全策略说明
- CDN：配置TLS版本控制
- SCDN：配置TLS
- DCDN：配置TLS版本控制