阿里云SSL证书（CA证书服务、数据安全）

最佳实践

文档版本：20200610

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. 阿里云文档中所有内容，包括但不限于图片、架构设计、页面布局、文字描述，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。
	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。
	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。	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 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@[redacted]:~# 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：

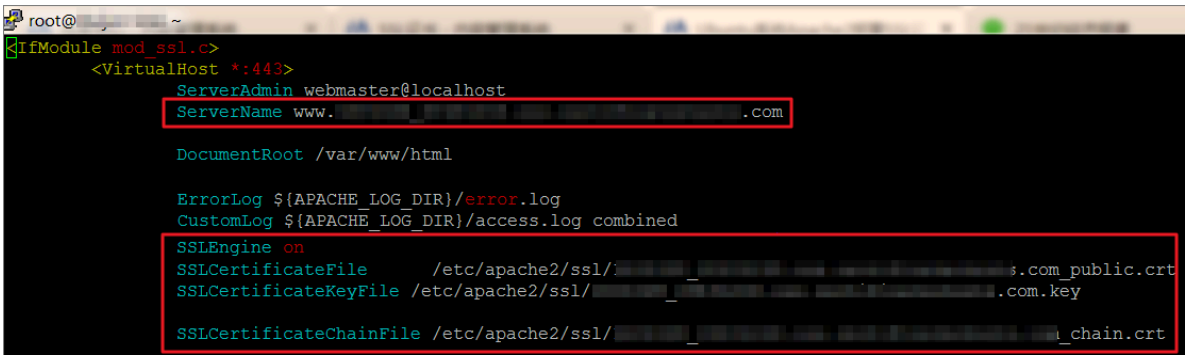
443端口是网络浏览端口，主要用于HTTPS服务。SSL模块启用后会自动放行443端口。若443端口未自动放行，可执行vi /etc/apache2/ports.conf并添加Listen 443手动放行。

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替换为证书密钥文件路径+证书密钥文件名。
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
```

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

7. 运行以下命令重启Apache 2服务。

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

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

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
- #unique_5
- #unique_6
- #unique_7
- #unique_8
- CentOS系统Tomcat 8.5或9部署SSL证书
- #unique_10

2 CentOS系统Tomcat 8.5或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。

```
[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
```

操作步骤

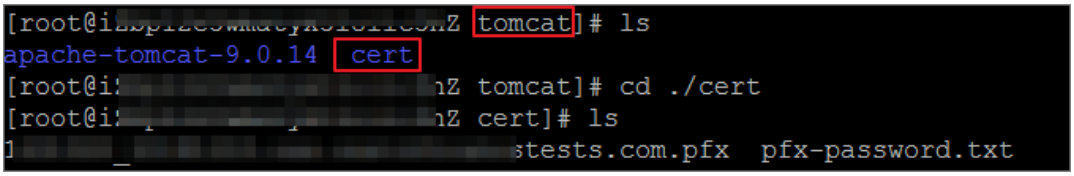
1. 解压Tomcat证书。

   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. 将下载的证书和密码文件拷贝到Tomcate的**cerf**目录下。

说明：



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，certificat
eKeystoreType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. （可选步骤）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. 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
- #unique_5
- Ubuntu系统Apache 2部署SSL证书
- #unique_6
- #unique_7
- #unique_8
- #unique_10

3 为云虚拟主机开启HTTPS加密访问

云虚拟主机提供HTTPS加密访问功能，您可以通过申请免费证书或上传现有的证书开启云虚拟主机的HTTPS加密访问功能。云虚拟主机还支持自动安装您在阿里云SSL证书服务中购买的证书。本文档介绍了如何为云虚拟主机开启HTTPS加密访问。

前提条件

开启HTTPS加密访问功能前，需确认您的云虚拟主机型号是否支持HTTPS加密。

型号	是否支持HTTPS
独享虚拟主机（包含轻云服务器）	支持。 **说明：** Windows操作系统主机仅支持开启1个域名。
共享虚拟主机	• Linux主机：除北京智能多线机房的Linux主机外均支持。 • Windows主机：不支持。

背景信息

云虚拟主机各个版本都支持HTTPS加密访问，详细内容请参见虚拟主机HTTPS加密访问设置。

下表描述了云虚拟主机支持申请和安装的证书类型。

证书类型	云虚拟主机提供的服务
Digicert免费型DV证书	申请新的免费证书并自动安装证书。 **说明：** 通过云虚拟主机控制台**HTTPS加密访问**页面申请的阿里云免费DV证书不支持下载。
云盾证书	自动安装您通过阿里云SSL证书服务购买的证书（包括付费证书和免费证书）。
自定义证书	自动安装您通过第三方平台购买的证书。

操作步骤

1. 登录云虚拟主机控制台。
2. 在**云虚拟主机**的**全部主机**页签中定位到需安装SSL证书的主机，单击该主机操作列**管理** > **域名管理** > **域名绑定**。
3. 在**域名绑定**页面，定位到需要安装SSL证书的域名，单击**开启**设置强制HTTPS加密访问。

4. 在**HTTPS加密访问**对话框中，单击**云盾证书**。

   HTTPS加密访问开启后，使用浏览器访问您的网站时会展示挂锁标志。

4 RDS安装SSL证书

RDS提供SSL加密功能。您可以在RDS控制台开启SSL设置、下载SSL证书并将其安装到您的数据库客户端中。本文档介绍了如何在RDS上安装阿里云SSL证书。

背景信息

RDS支持开启SSL加密功能和将SSL证书安装到所需的应用服务中，以提升通信数据的安全性和完整性。

步骤一：开启SSL加密

1. 登录RDS管理控制台。
2. 在**实例列表**页面单击需要开启SSL加密的实例名称。
3. 在左侧导航栏单击**数据安全性**。
4. 在**数据安全性**页面单击**SSL**。
5. 在**SSL**页面单击**SSL证书信息**的按钮，为该实例开启SSL加密。
   开启SSL加密后，需等待数分钟才能开启成功。
6. 下载证书。

   有关开启SSL加密服务的详细内容，请参见RDS#unique_13。

   如果现有证书无法满足您的需求，可前往阿里云SSL证书控制台购买新的证书。在SSL证书控制台购买的证书会自动同步到您的RDS实例中。

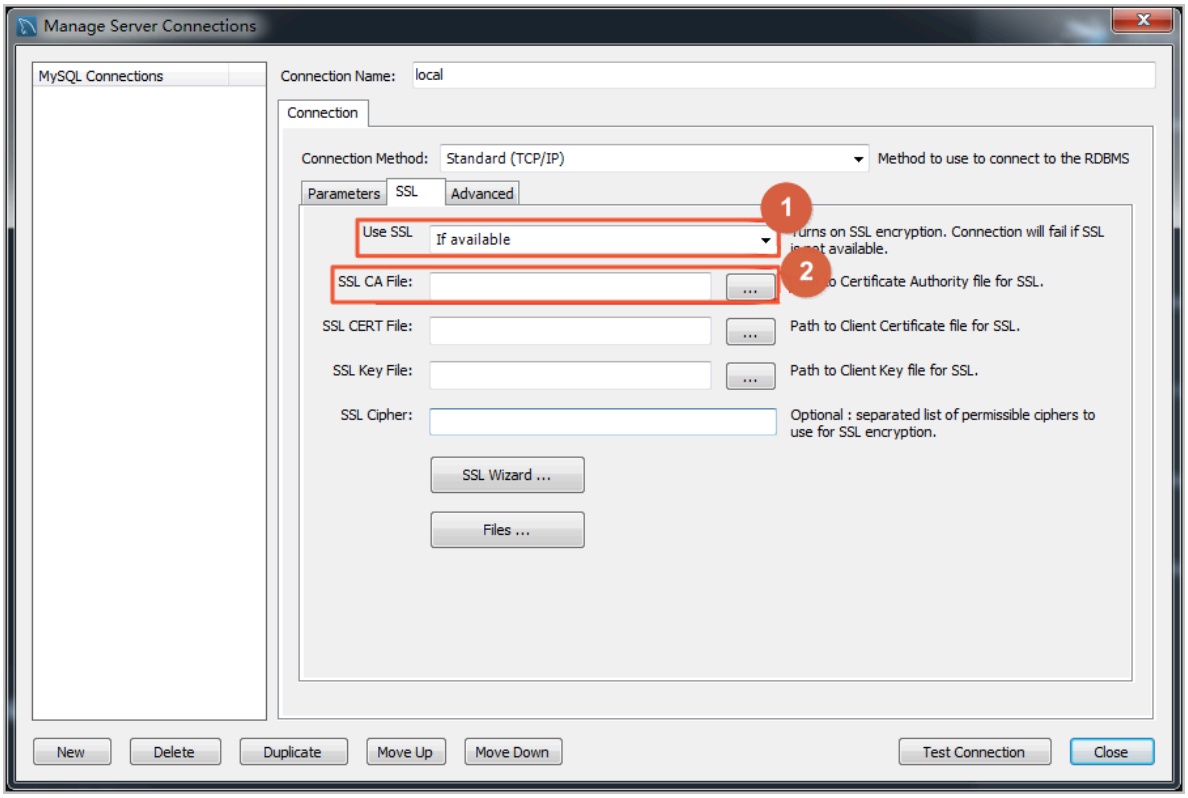
步骤二：配置证书

开通SSL加密服务后，请参考以下步骤在您的数据库应用或客户端中配置SSL证书。

MySQL Workbench配置SSL证书

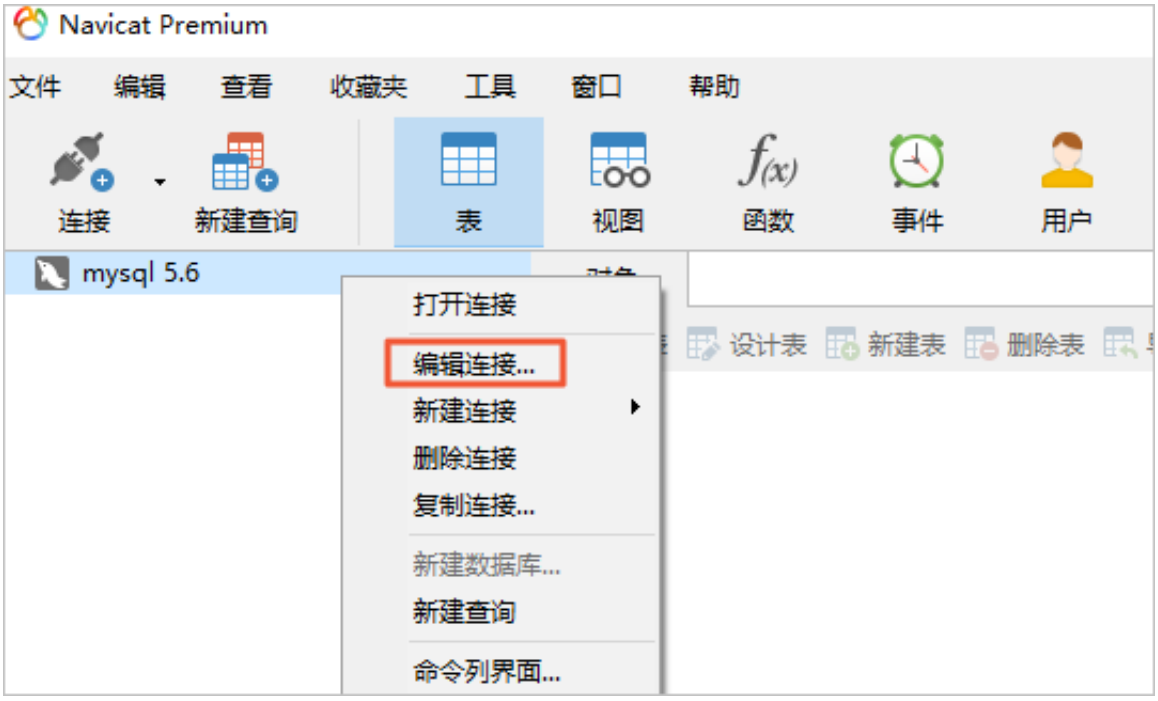
1. 打开MySQL Workbench。
2. 选择**Database** > **Manage Connections** 。

3. 启用**Use SSL**，并导入SSL CA证书，如下图所示。



Navicat配置SSL证书

1. 打开Navicat。
2. 在目标数据库上单击鼠标右键，选择**编辑连接** 。



3. 选择**SSL**页签，选择.pem格式CA证书的路径。参照下图进行设置。

4. 单击**确定**。

说明:

如果报connection is being used错误，是由于之前的会话未断开，请关闭Navicat重新打开。

5. 双击目标数据库测试能否正常连接。

5 DNSPOD域名TXT解析

本文档介绍了如何对DNSPOD域名进行DNS解析。

背景信息

如果您需要绑定阿里云SSL证书的域名是在腾讯云申请的，您需要先在腾讯云DNS解析控制台中添加TXT记录，然后在阿里云SSL证书控制台完成DNS验证。

操作步骤

1. 登录到阿里云SSL证书控制台，定位到需要进行DNS验证的证书，单击**证书申请**获取该证书的主机**记录值**。

2. 登录腾讯云云解析控制台，在**域名解析列表**中添加一条记录并完成相关的设置。

   添加记录设置如下：

   - **主机记录**：输入_dnsauth。
   - **记录类型**：输入TXT。
   - **记录值**：输入该域名在阿里云SSL证书服务台获取的DNS验证值（详见步骤1）。

3. 执行以下命令验证DNS配置是否生效。

   本文档中域名以abc.com为例。

   - Windows系统：

     ```
     nslookup -q=TXT _dnsauth.abc.com
     ```

   - Linux系统：

     ```
     dig TXT _dnsauth.abc.com
     ```

   - MACOS系统：

     ```
     dig TXT _dnsauth.abc.com
     ```

4. 登录到阿里云SSL证书控制台对该证书域名进行验证。验证成功后该证书的状态变为**申请审核中**，请耐心等待。

申请审核通过后，证书状态会变为**已签发**。