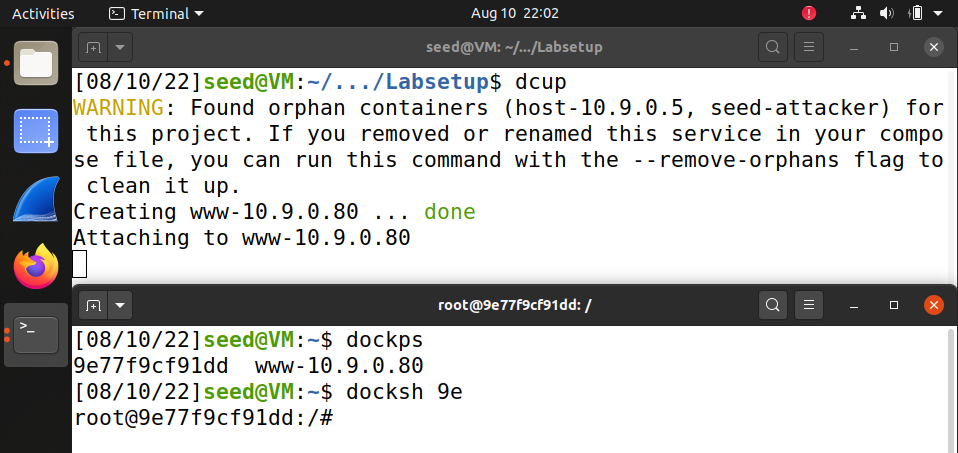
**Task 2 - PKI**

**环境配置**

进入实验目录/home/seed/Desktop/Labs\_20.04/Cryptography/1-Public-Key Infrastructure (PKI) Lab/Labsetup/，使用dcup命令创建容器镜像和启动容器，新开启终端，输入命令dockps查看docker ID，为9e开头，输入docksh 9e连接容器。



1. **开启HTTPS服务**

开启HTTPS服务首先需要openssl签发证书，签发证书需要一个CA，这里我们自己生成CA，然后使用我们自己的CA签发证书。

1. **生成CA的公钥证书和私钥**

OpenSSL是一个[开放源代码](https://baike.baidu.com/item/%E5%BC%80%E6%94%BE%E6%BA%90%E4%BB%A3%E7%A0%81)的[软件](https://baike.baidu.com/item/%E8%BD%AF%E4%BB%B6)[库](https://baike.baidu.com/item/%E5%BA%93)包，应用程序可以使用这个包来进行安全通信，避免窃听，同时确认另一端连接者的身份。这个包广泛被应用在互联网的网页服务器上。

OpenSSL包含如下命令：

enc:对称加密

dgst:单向加密

passwd:生成密码

rand:生成随机数

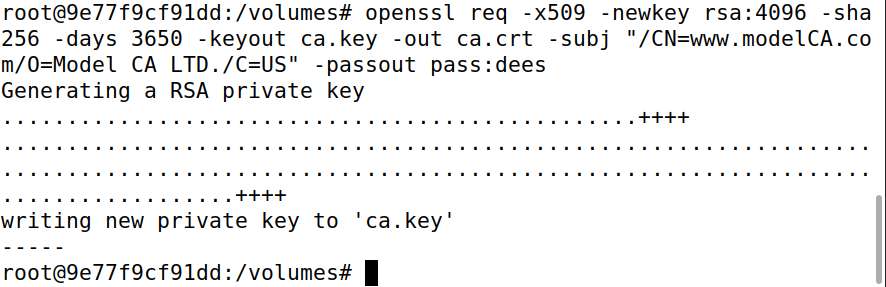
genrsa:生成私钥

rsa:提取公钥（从生成的私钥中提取）

x509:自签证书

req：CSR证书请求文件生成（通过私钥文件生成）

输入命令openssl req -x509 -newkey rsa:4096 -sha256 -days 3650 -keyout ca.key -out ca.crt -subj "/CN=www.modelCA.com/O=Model CA LTD./C=US" -passout pass:dees进行密钥的生成。



（这里第一次错把实验手册里的命令当成多行执行，会被要求自行填写口令和身份信息等，而且是在根目录下执行的，后改为在/volumes文件夹下执行，生成的文件也会在/volumes文件夹下）

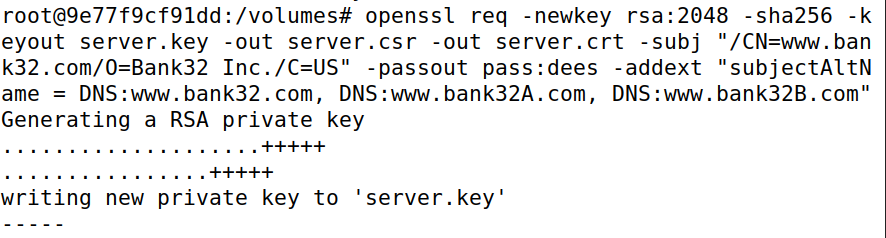
输入ls，可以看到生成的ca.crt和ca.key两个文件。



1. **生成一个CSR**

CSR即证书签名申请（Certificate Signing Request），获取SSL证书，需要先生成 CSR 文件并提交给证书颁发机构（CA）。CSR包含了公钥和标识名称（Distinguished Name），通常从Web服务器生成CSR，同时创建加解密的公钥私钥对。

在容器的/volumes路径下输入openssl req -newkey rsa:2048 -sha256 -keyout server.key -out server.csr ~~-out server.crt~~ -subj "/CN=www.bank32.com/O=Bank32 Inc./C=US" -passout pass:dees -addext "subjectAltName = DNS:www.bank32.com, DNS:www.bank32A.com, DNS:www.bank32B.com"。



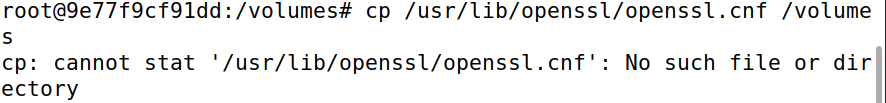
输入ls，可以看到生成的server.crt、server.csr和server.key三个文件。

（这里一开始对步骤理解有误，多添加了一句-out server.crt的命令，多生成了server.crt这个文件，应该在后面再签发证书，在后面的操作中已将其覆盖）

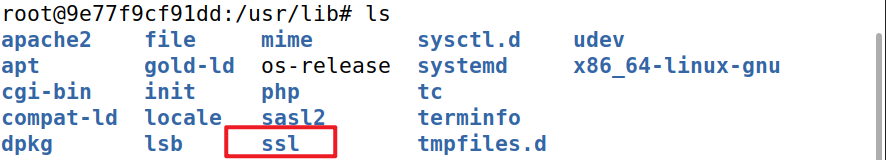


1. **使用前面的CA签发证书**

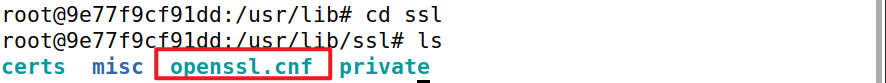
输入cp /usr/lib/openssl/openssl.cnf /volumes将openssl.cnf文件复制到当前文件夹/volumes，但失败，报错为路径不正确。



在翻看文件目录后找到原因：其在目录中的名称为ssl而非openssl。

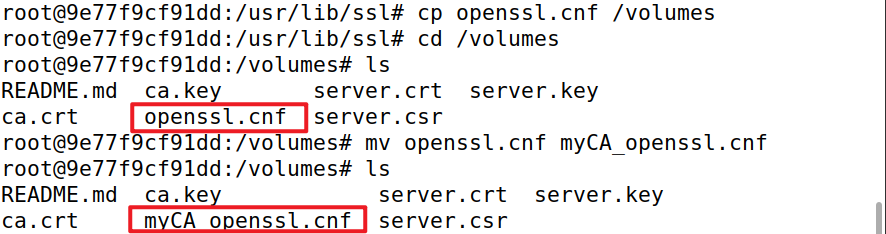


输入cd ssl进入该文件夹，再输入ls查看目录下内容，发现openssl.cnf文件存在。



输入cp openssl.cnf /volumes将该文件复制到/volumes文件夹下。

进入/volumes文件夹输入mv openssl.cnf myCA\_openssl.cnf对文件进行重命名。

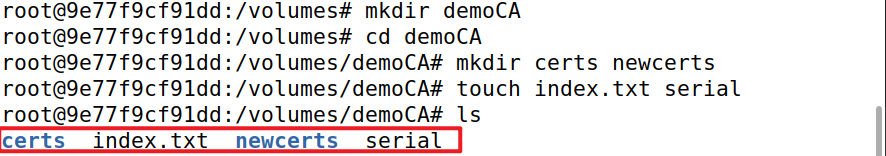


修改文件，删去下图中的两个注释号。

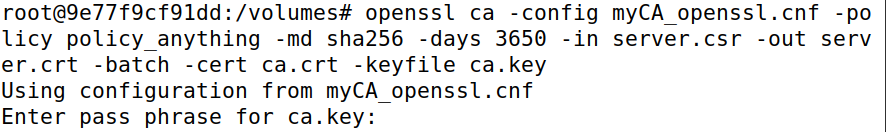




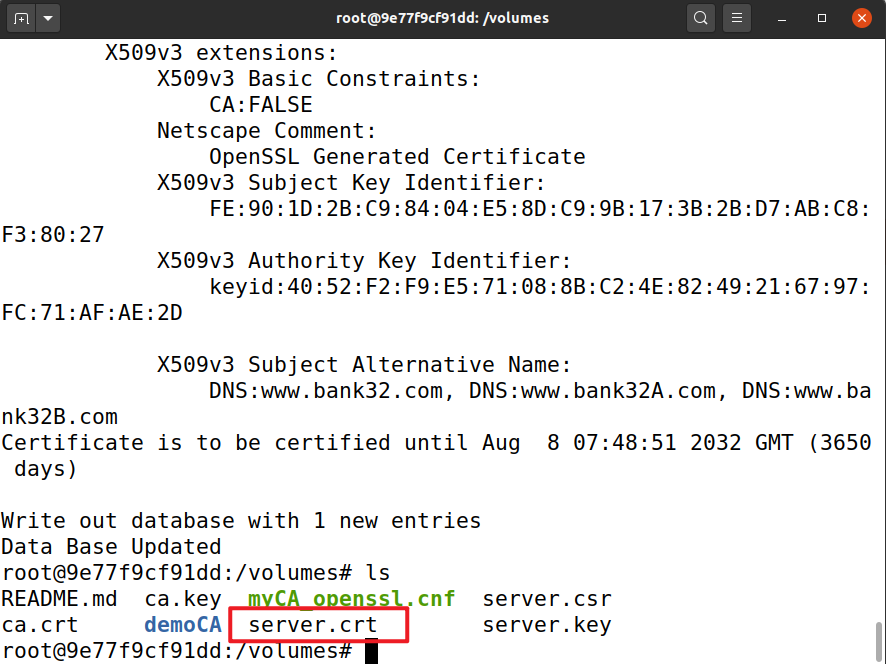
在当前目录下输入mkdir demoCA创建文件夹，进入demoCA文件夹再创建certs和newcerts两个文件夹以及index.txt和serial文件（填入内容为一个序列号）。



输入命令openssl ca -config myCA\_openssl.cnf -policy policy\_anything -md sha256 -days 3650 -in server.csr -out server.crt -batch -cert ca.crt -keyfile ca.key签发证书。



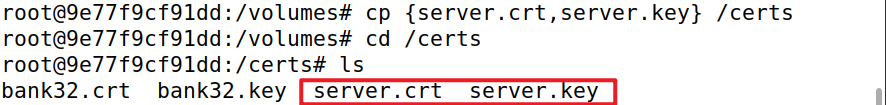
输入前面设置的口令dees。



签发完成，可看到新的名为server.crt的证书。

1. **修改Apach配置文件中的证书与私钥位置**

将私钥和证书server.key和server.crt拷贝到/certs文件夹中。



然后再输入vim /etc/apache2/sites-available/bank32\_apache\_ssl.conf，修改apache的配置文件，修改cert和key的位置。



由



修改为

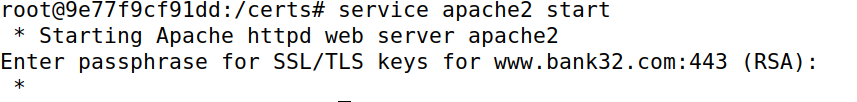


1. **开启网站服务**

输入service apache start开启apache服务，但报错不明服务。



修改为service apache2 start，输入口令dees后成功开启。



1. **修改主机hosts文件**

在虚拟机中输入sudo gedit /etc/hosts来修改hosts文件，添加如下条目。



1. **访问[www.bank32.com](http://www.bank32.com)**

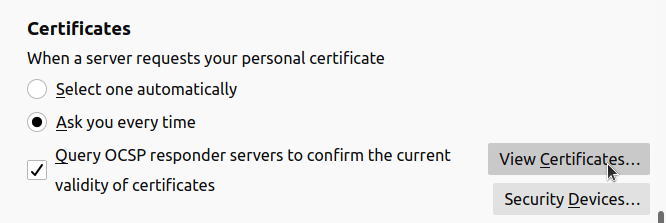
在Firefox浏览器中访问[www.bank32.com](http://www.bank32.com)，并未成功，问题在于Firefox需要CA的公钥证书取验证网站的证书，但是浏览器并没有CA的公钥证书，所以无法验证。因此需要将CA的公钥证书添加到浏览器中。

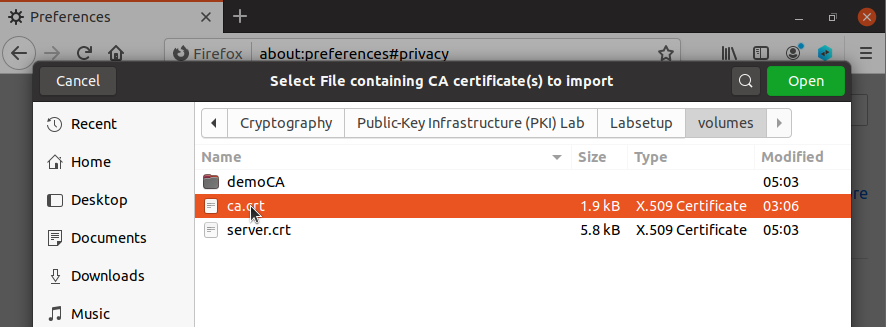
问题在于Firefox需要CA的公钥证书取验证网站的证书，但是浏览器并没有CA的公钥证书，所以无法验证。因此需要将CA的公钥证书添加到浏览器中。

1. **Firefox添加CA证书**

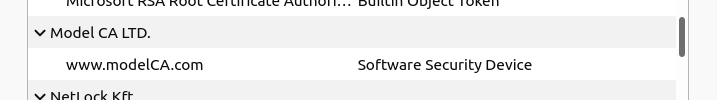
先将容器中的证书ca.crt拷贝到主机（volumes文件夹即可）。

在浏览器地址栏输入about:preferences#privacy，进入设置页面后找到Certificates栏点击View Certificates。



点击Import，选择所在路径，找到ca.crt导入，选择Trust to websites选项。

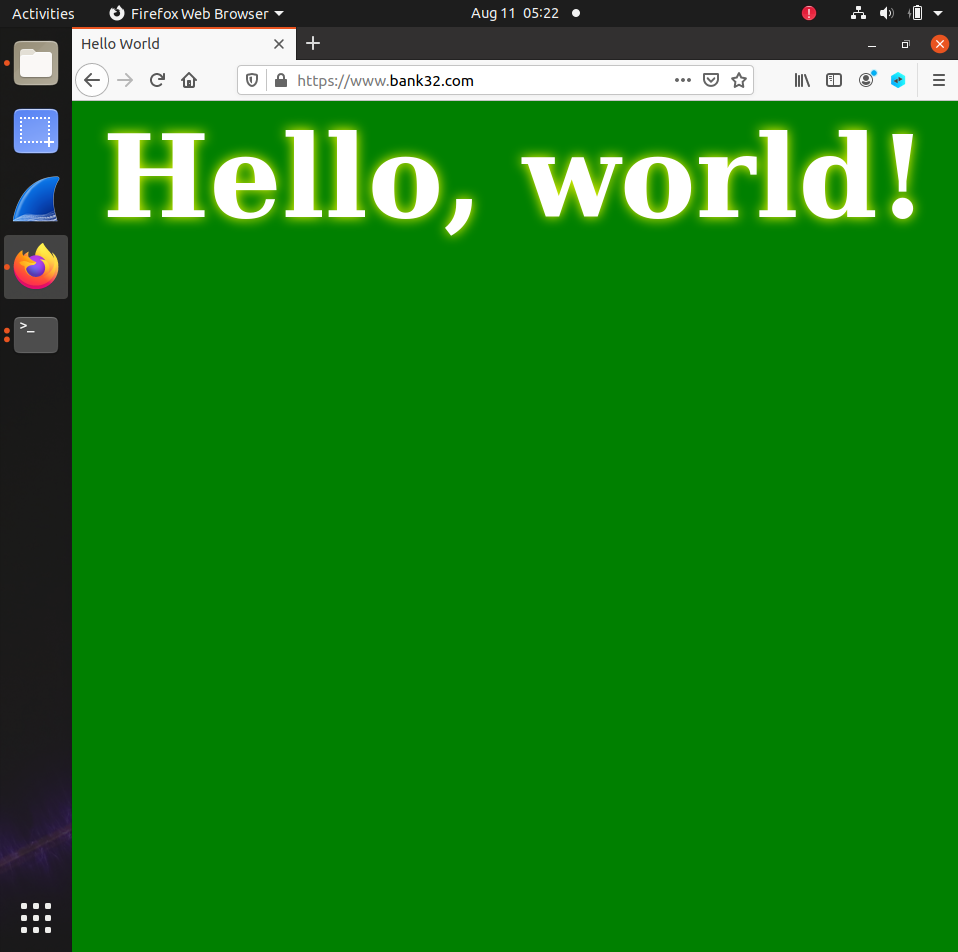
成功导入后便可在列表中找到。



1. **再次访问浏览器**

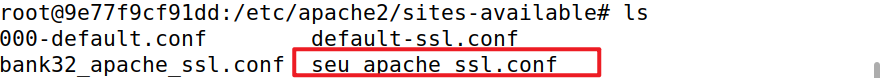
可正常访问。

（https与http网页不相同，http网页是红色）



1. **MITM中间人攻击**

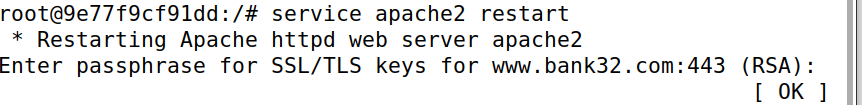
进入/etc/apache2/sites-available文件夹，将bank32\_apache\_ssl.conf复制一份，命名为seu\_apache\_ssl.conf。



修改seu\_apache2\_ssl.conf的内容。



重启apache2服务。



在firefox浏览器中访问[www.seu.edu.cn](http://www.seu.edu.cn)，仍可正常显示。不清楚具体原因，已尝试过清理浏览器数据、编辑hosts文件、重启apache2服务、重启容器、重启虚拟机、更换[www.seu.edu.cn](http://www.seu.edu.cn)为其他网，网页可正常显示。

