1 签名证书与自签名证书

签名证书：由权威颁发机构颁发给服务器或者个人用于证明自己身份的东西。
自签名证书：由服务器自己颁发给自己，用于证明自己身份的东西，非权威颁发机构发布。

2 openssl简介

openssl 是一个强大的安全套接字层密码库，囊括主要的密码算法、常用的密钥和证书封装管理功能及SSL协议，并提供丰富的应用程序供测试或其它目的使用。

3 KEY与CSR的区别

Key通常用来存放一个公钥或者私钥,并非X.509证书,编码同样的,可能是PEM,也可能是DER。证书自身拥有一个密钥对（即一个公钥和一个私钥），由公钥（Public Key）与私钥（Private Key）是通过一种算法得到，公钥是密钥对中公开的部分，私钥则是非公开的部分。一般公钥和密钥的关系为：1，公钥和私钥成对出现、2，公开的密钥叫公钥，只有自己知道的叫私钥、3，用公钥加密的数据只有对应的私钥可以解密、4，用私钥加密的数据只有对应的公钥可以解密、5，如果可以用公钥解密，则必然是对应的私钥加的密、6，如果可以用私钥解密，则必然是对应的公钥加的密。
CSR文件必须在申请和购买SSL证书之前创建。也就是证书申请者在申请数字证书时由CSP(加密服务提供者)在生成私钥的同时也生成证书请求文件，证书申请 者只要把CSR文件提交给证书颁发机构后，证书颁发机构使用其根证书私钥签名就生成了证书公钥文件，也就是颁发给用户的证书。

4 证书配置之生成根证书

创建证书存放目录：

```
mkdir -p /data/cert && cd /data/cert
```

```
[root@bogon ~]# mkdir -p /data/cert && cd /data/cert
[root@bogon cert]#
```

4.1创建自己的CA证书（不使用第三方权威机构的CA来认证，自己充当CA的角色）

```
1    openssl genrsa -out ca.key 2048  #生成根证书私钥（无加密）
```

```
[root@bogon cert]# openssl genrsa -out ca.key 2048
Generating RSA private key, 2048 bit long modulus
..............................+++
..............................+++
e is 65537 (0x10001)
[root@bogon cert]#
```

4.2生成自签名证书（使用已有私钥ca.key自行签发根证书）

```
openssl req -x509 -new -nodes -key ca.key -days 10000 -out ca.crt -subj "/CN=Harbor-ca"
req  产生证书签发申请命令
-x509     签发X.509格式证书命令。X.509是最通用的一种签名证书格式。
-new 生成证书请求
-key      指定私钥文件
```

-nodes 表示私钥不加密

-out 输出

-subj 指定用户信息

-days 有效期

```
[root@bogon cert]# openssl req -x509 -new -nodes -key ca.key -days 10000 -out ca.crt -subj "/CN=Harbor-
ca"
[root@bogon cert]#
```

5 证书配置之生成服务器端证书

5.1生成服务器端私钥和CSR签名请求

```
openssl req -newkey rsa:4096 -nodes -sha256 -keyout server.key -out server.csr
```

```
[root@bogon cert]# openssl req -newkey rsa:4096 -nodes -sha256 -keyout server.key -out server.csr
Generating a 4096 bit RSA private key
..............++
.....................................................................................................
.....................................................................................................
.............................++
writing new private key to 'server.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [XX]:
```

一路回车。。。

5.2签发服务器证书

```
echo subjectAltName = IP:192.168.88.128 > extfile.cnf
openssl x509 -req -in server.csr -CA ca.crt -CAkey ca.key -CAcreateserial -days 365 -extfile
extfile.cnf -out server.crt
```

x509 签发X.509格式证书命令。

-req 表示证书输入请求。

-days 表示有效天数

-extensions 表示按OpenSSL配置文件v3_req项添加扩展。

-CA 表示CA证书,这里为ca.crt

-CAkey 表示CA证书密钥,这里为ca.key

-CAcreateserial表示创建CA证书序列号

-extfile 指定文件

```
[root@bogon cert]# echo subjectAltName = IP:192.168.88.128 > extfile.cnf
[root@bogon cert]# openssl x509 -req -in server.csr -CA ca.crt -CAkey ca.key -CAcreateserial -days 365
-extfile extfile.cnf -out server.crt
Signature ok
subject=/C=XX/L=Default City/O=Default Company Ltd
Getting CA Private Key
```

6 修改hardor的配置文件hardor.cfg

更新主机名和协议

#set主机名

```
hostname = 192.168.88.128
#set ui_url_protocol
ui_url_protocol = https
```

```
[root@bogon cert]# cd
[root@bogon ~]# cd harbor
[root@bogon harbor]# vi harbor.cfg
[root@bogon harbor]#
```

```
#The IP address or hostname to access admin UI and registry service.
#DO NOT use localhost or 127.0.0.1, because Harbor needs to be accessed by external clients.
hostname = 192.168.88.128

#The protocol for accessing the UI and token/notification service, by default it is http.
#It can be set to https if ssl is enabled on nginx.
ui_url_protocol = https
```

7 设置docker证书

```
# 如果如下目录不存在，请创建，如果有域名请按此格式依次创建
mkdir -p /etc/docker/certs.d/192.168.88.128
# mkdir -p /etc/docker/certs.d/[IP2]
# mkdir -p /etc/docker/certs.d/[example1.com]
# 如果端口为443，则不需要指定。如果为自定义端口，请指定端口
# /etc/docker/certs.d/yourdomain.com:port
# 将ca根证书依次复制到上述创建的目录中
cp ca.crt /etc/docker/certs.d/192.168.88.128/
```

8 为Harbor生成配置文件

```
#首先重启下docker
service docker restart
#为Harbor生成配置文件
./prepare
```

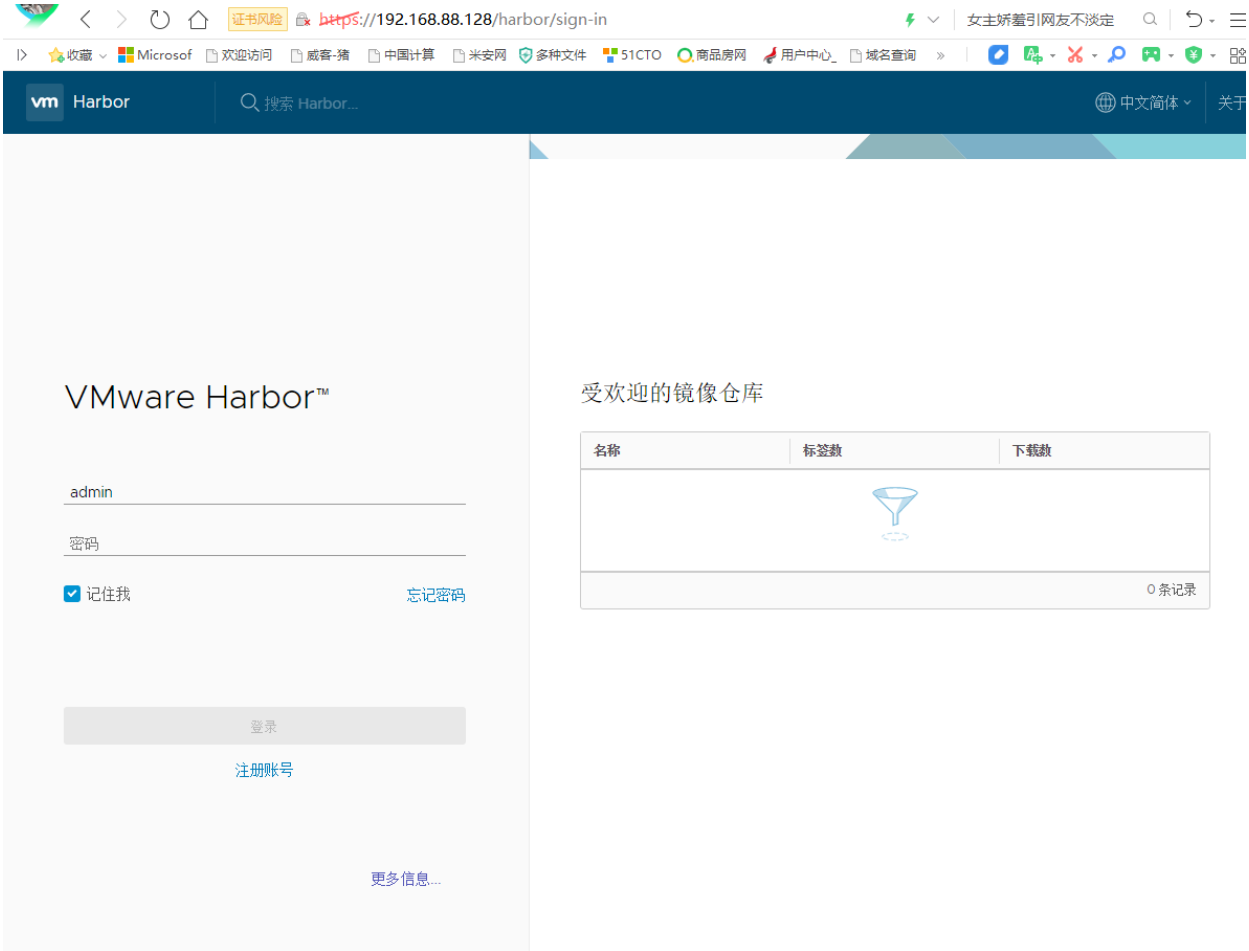
```
[root@bogon harbor]# ./prepare
Clearing the configuration file: ./common/config/adminserver/env
Clearing the configuration file: ./common/config/ui/env
Clearing the configuration file: ./common/config/ui/app.conf
Clearing the configuration file: ./common/config/ui/private_key.pem
Clearing the configuration file: ./common/config/db/env
Clearing the configuration file: ./common/config/jobservice/env
Clearing the configuration file: ./common/config/jobservice/app.conf
Clearing the configuration file: ./common/config/registry/config.yml
Clearing the configuration file: ./common/config/registry/root.crt
Clearing the configuration file: ./common/config/nginx/nginx.conf
Clearing the configuration file: ./common/config/log/logrotate.conf
loaded secret from file: /data/secretkey
Generated configuration file: ./common/config/nginx/nginx.conf
Generated configuration file: ./common/config/adminserver/env
Generated configuration file: ./common/config/ui/env
Generated configuration file: ./common/config/registry/config.yml
Generated configuration file: ./common/config/db/env
Generated configuration file: ./common/config/jobservice/env
Generated configuration file: ./common/config/log/logrotate.conf
Generated configuration file: ./common/config/jobservice/app.conf
Generated configuration file: ./common/config/ui/app.conf
Generated certificate, key file: ./common/config/ui/private_key.pem, cert file: ./common/config/registr
y/root.crt
The configuration files are ready, please use docker-compose to start the service.
[root@bogon harbor]#
```

9 完成启动Harbor

```
docker-compose up -d
```

```
[root@bogon harbor]# docker-compose up -d
harbor-log is up-to-date
registry is up-to-date
Recreating harbor-adminserver ...
Recreating harbor-adminserver ... done
Recreating harbor-ui ... done
Recreating nginx ...
Recreating harbor-jobservice ... done
[root@bogon harbor]#
```

访问测试：




```
[root@bogon harbor]# docker login 192.168.88.128
Username: admin
Password:
Login Succeeded
[root@bogon harbor]#
```

docker-compose down -v