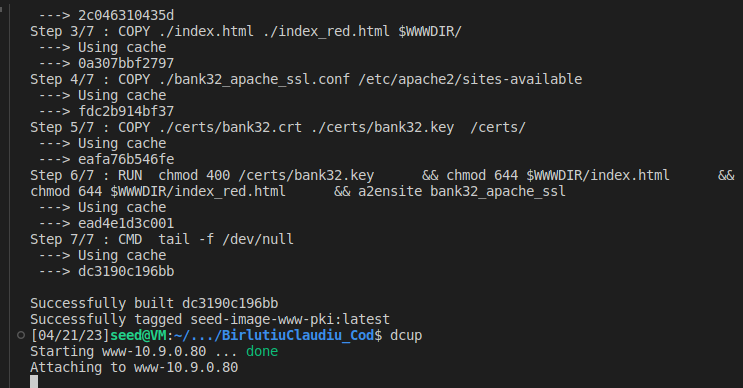
# Raport pentru lucrarea 6: Infrastructura de chei publice (PKI)

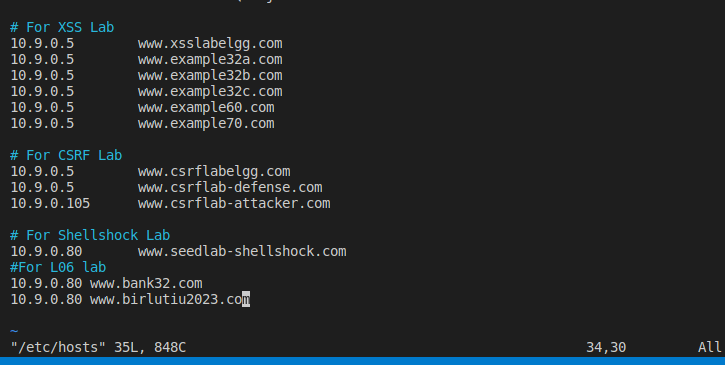
Autor: Birlutiu Claudiu-Andrei

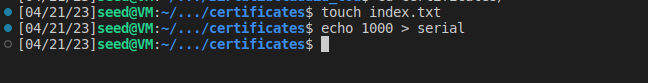
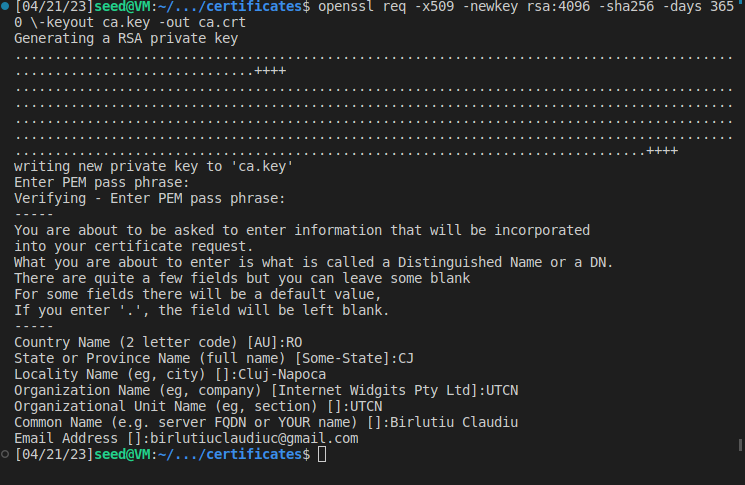
## Sarcina 1: Deveniți o autoritate de certificat (CA)

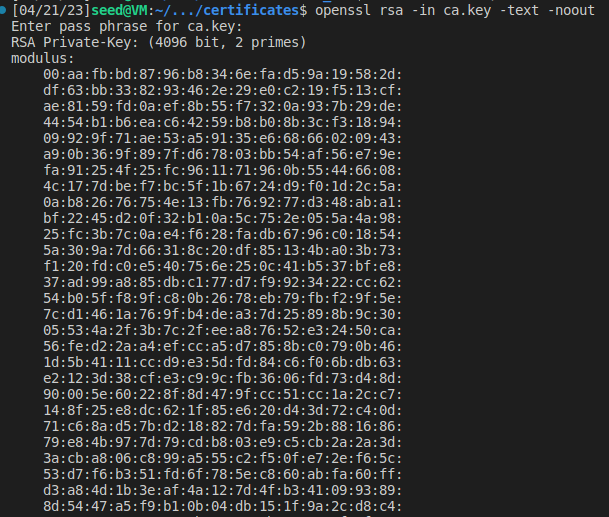
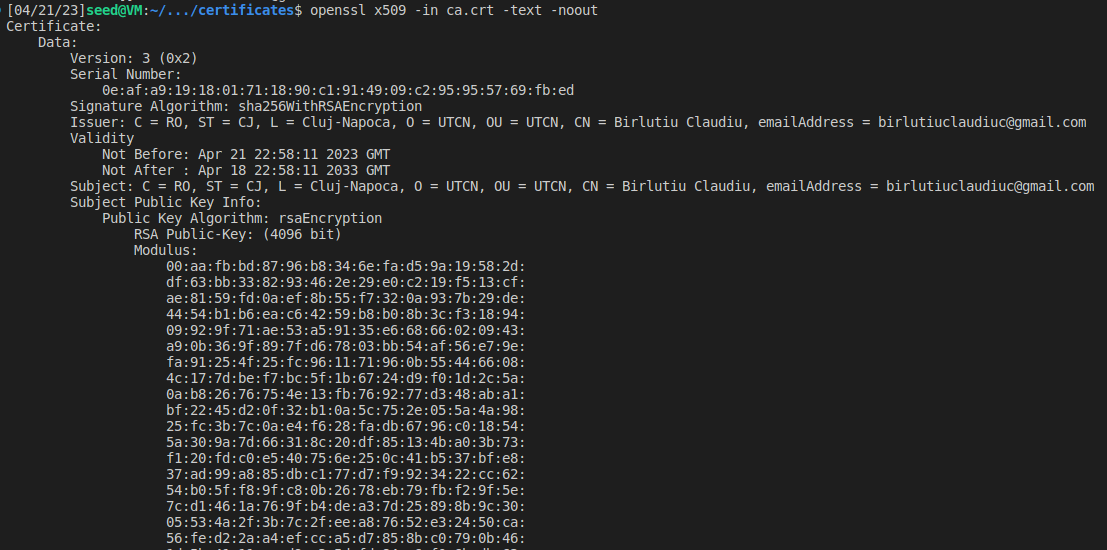
## În prima faza am creat containerul și am pornit acest serviciu



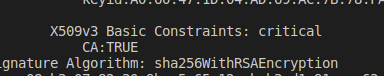
* Am adaugat în /etc/hosts cele 2 intrari



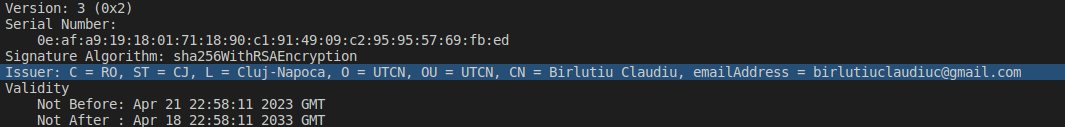
* am creat cele 2 fisiere mentionate
* am generat certificatul X.509 auto-semnat astfel:
* generăm o pereche de chei RSA cu o lungime de 4096 biți prin optiunea -newkey rsa:4096
* -sha256: specifică faptul că dorim să folosim algoritmul de hash SHA-256 pentru semnarea certificatului.
* Certificatul va fi valid 3650 de zile (10 ani aprox)
* am vizualizat continutul decodat al fisierelor



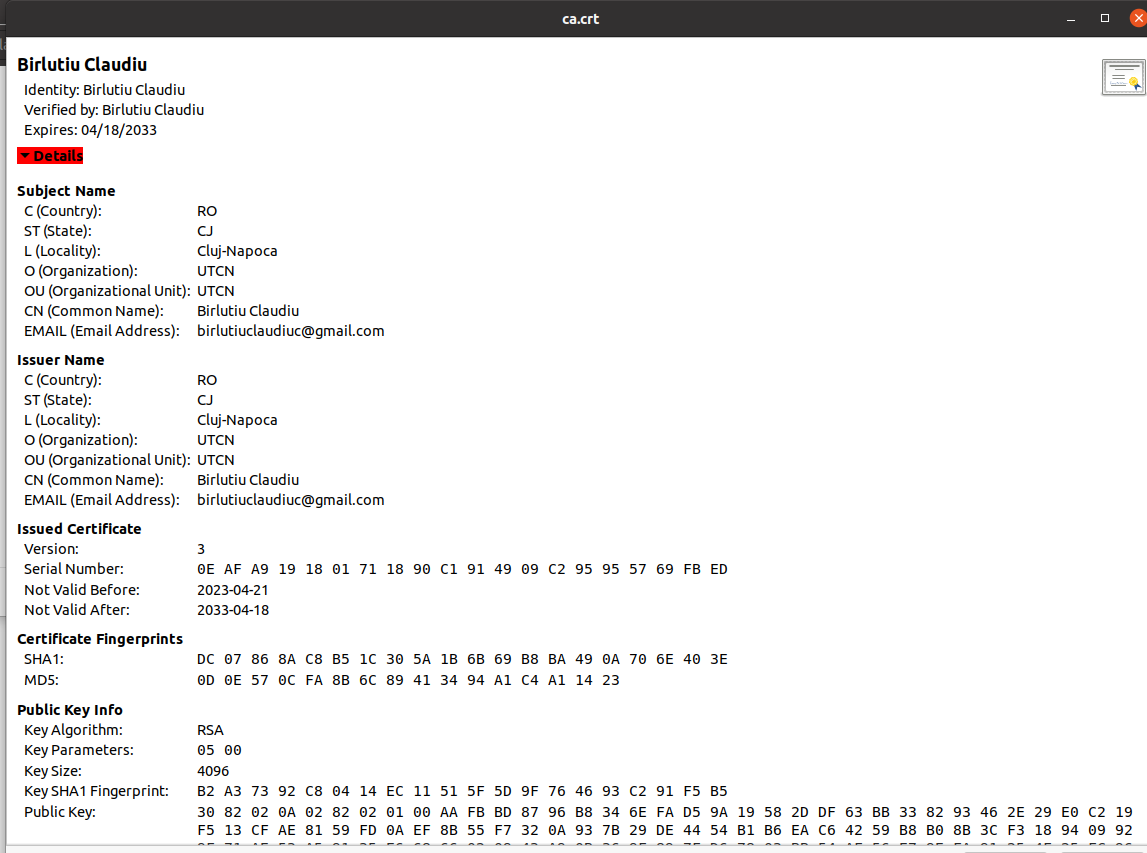
* Ce parte a certificatului indica ca acesta este un certificat CA?
  + există o extensie numită "Basic Constraints" care indică dacă certificatul este un certificat de autoritate de certificare (CA) sau nu.



* Ce parte a certificatului indica ca acesta este un certificat semnat de sine?
  + există un câmp numit "Issuer" care indică entitatea care a emis certificatul.

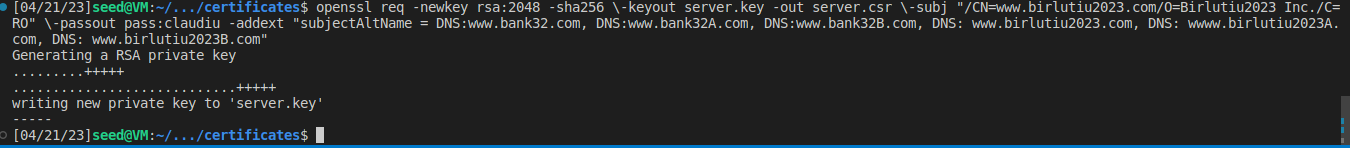


* In algoritmul RSA, avem un exponent public **e,** un exponent privat **d** , un modul **n** si doua secrete, numerele **p** și **q,** astfel ıncat **n=pq.** Va rugam sa identificati valorile pentru aceste elemente in certificatul dvs. si fisierele cheie
  + Exponent: 65537 (0x10001)
  + publicExponent: 65537 (0x10001)
  + celelalte valori se pot observa din : **openssl rsa -in ca.key -text -noout**
* certificatul

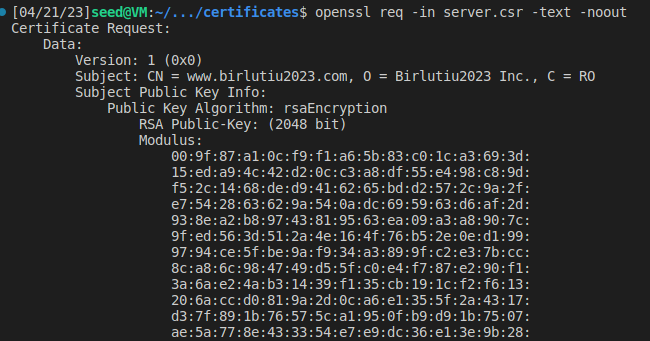


## Sarcina 2: Generarea unei cereri de certificat pentru serverul dvs.

* Am generat o cerere de certificat pentru serverul nostru folsoind comanda:
  + openssl req -newkey rsa:2048 -sha256 \-keyout server.key -out server.csr \-subj "/CN=www.birlutiu2023.com/O=Birlutiu2023 Inc./C=RO" \-passout pass:claudiu -addext "subjectAltName = DNS:www.bank32.com, DNS:www.bank32A.com, DNS:www.bank32B.com, DNS: www.birlutiu2023.com, DNS: wwww.birlutiu2023A.com, DNS: www.birlutiu2023B.com"

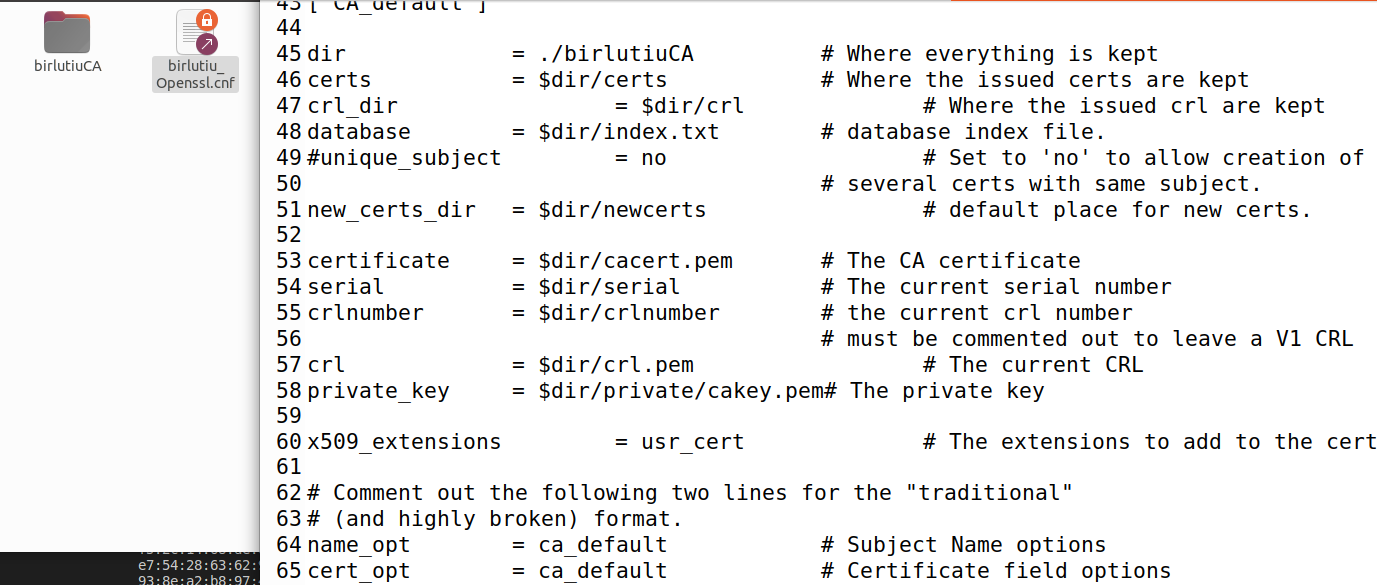


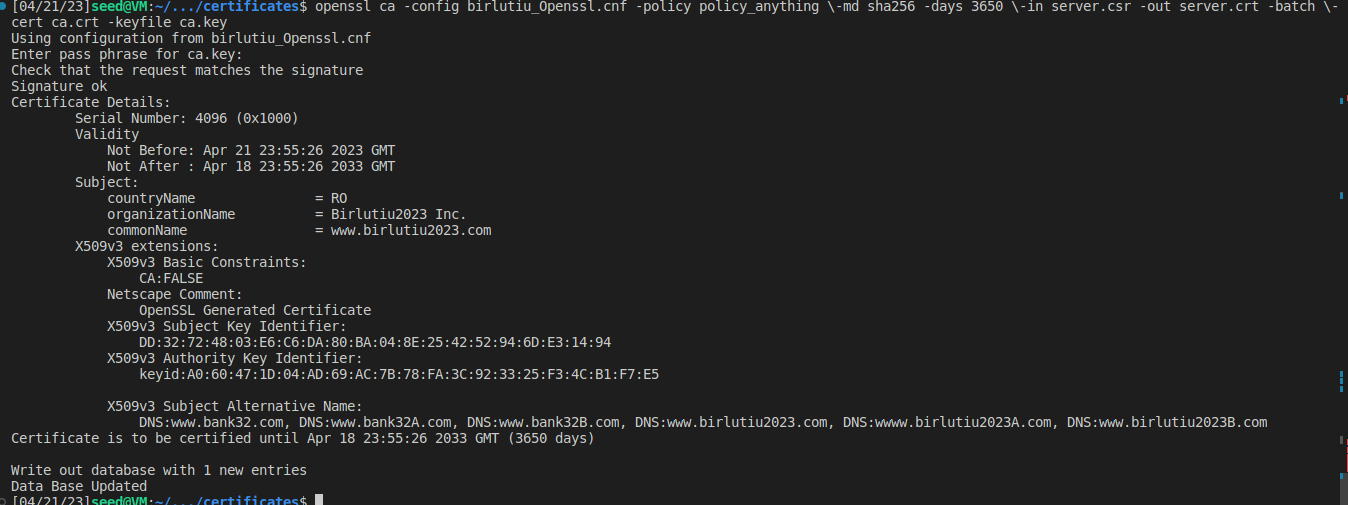
* am urmărit continutul decoda al fisierului .csr

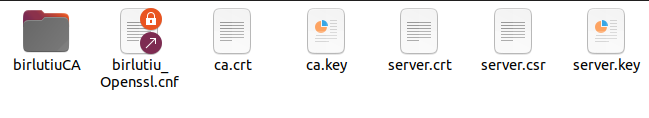


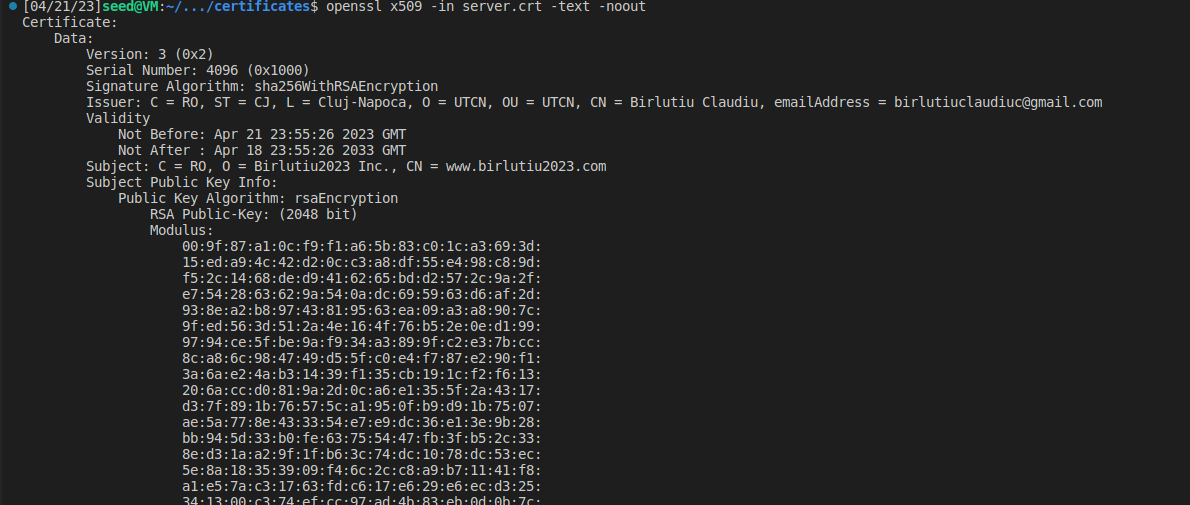
## Sarcina 3: Generarea unui certificat pentru serverul dvs.

* Fișierul **server.csr** trebuie sa aibă **semnatura CA** pentru a forma un certificat.
* Am copiat fișierul openssl.cnf într-un folder din BirlutiuClaudiu\_Cod lângă directorul **birlutiuCA** unde se afla toate certificatele pe care le-am generat

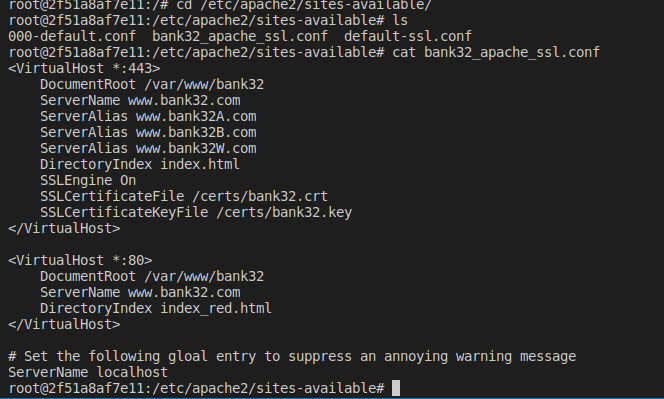


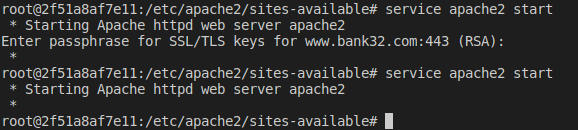
* Astfel, pentru sermnarea certificatului **server.csr** într-un certificat x509 (**server.crt**), folosind **CA.crt** și **ca.key** vom rula următoarea comanda:
  + - openssl ca -config **birlutiu\_Openssl.cnf** -policy **policy\_anything** \-md sha256 -days 3650 \-in **server.csr** -out **server.crt** -batch \-cert **ca.crt** -keyfile **ca.key**
* rezultatul obținut



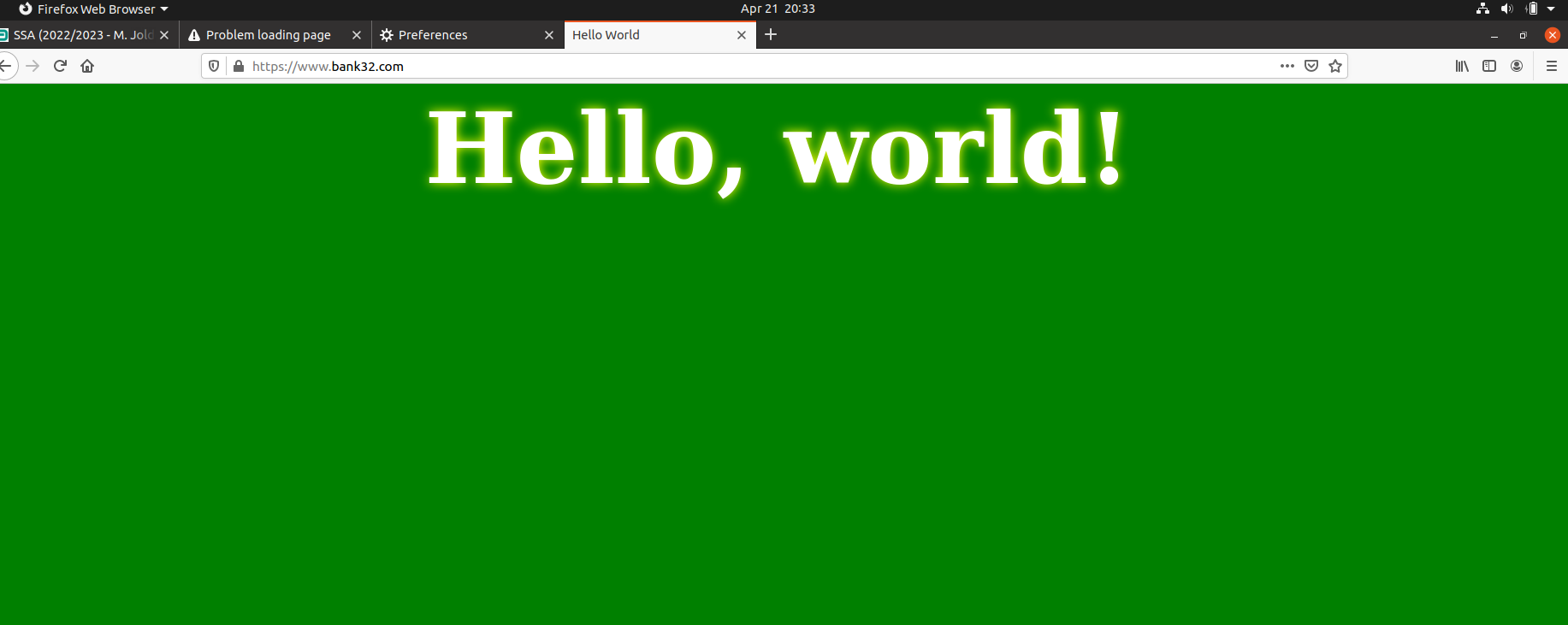
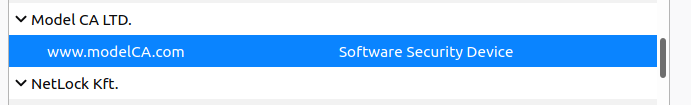
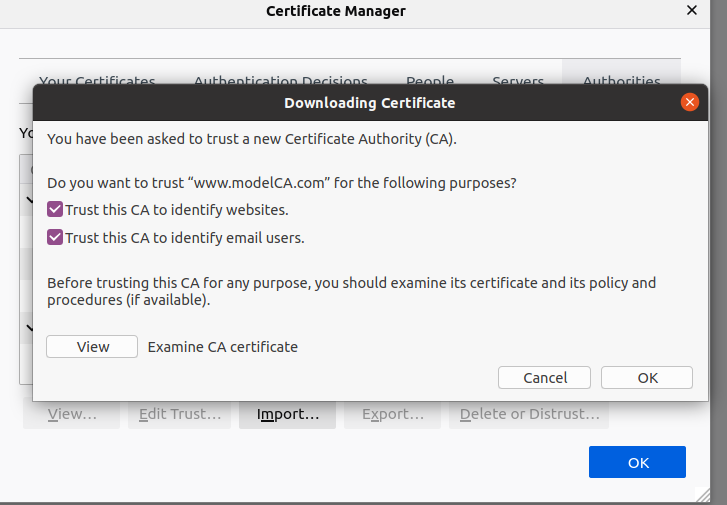


## Sarcina 4: Plasarea unui certificat într-un sit de web HTTPS bazat pe Apache

* Vizualizare în cobatianrul nousru a fisierului **bank32\_apache\_ssl.conf**
* 
* DocumentRoot – unde sunt stocate fisierele pentru site-ul web
* **activam apache2** din container:



* la o prima încercare a accesarii din browser a site-ului [www.bank.com](http://www.bank.com/) observam ca nu se randeaza niciun site deaorece nu avem încărcata autoritatea pentru validarea certif pentru acest site; vom încarca în FireFox certificatul **modelCA.crt** din folderul **certs;** aceasta e autoritatea care va certifica site-ul bank32 cu certificatul bank32.crt



În continuare ne vom ocupa de configurarea propriului nostru site pentru care am obținut certificatul.

* pentru început vom crea în volumes un fisier denumiit certs unde se vor afla fiserele **ca.crt, ca.key, server.crt**  fisiere pe care le-am obținut anterior în

## Sarcina 5: Lansarea unui atac de tipul om-la-mijloc

## Sarcina 7: Lansarea unui atac de tipul om-la-mijloc cu o CA compromisă