## Applied Cryptography Manual

Lab-06

20 October 2022 22:23

Name: Vishwas M

SRN: PES2UG20CS390

SEC: F

DATE: 26/10/2022

LAB: 5

# Prerequisites

Labsetup files - [https://seedsecuritylabs.org/Labs\_](https://seedsecuritylabs.org/Labs_20.04/Crypto/Crypto_PKI/)

[20.04/Crypto/Crypto\_PKI/](https://seedsecuritylabs.org/Labs_20.04/Crypto/Crypto_PKI/)

# Task

**1: Becoming a**

**certificate authority (CA)**

Firstly, copy directory

the /usr/lib/ssl/openssl.cnf

file to your

working

Then create the following files and directories in the working directory:

pki\_lab

* demoCA
  + certs (dir)
  + crl (dir)
  + newcerts (dir)
  + index.txt (blank

text file)

* + Serial

(contains a 4

digit

number, no line ending)

Creating certificate authority

### Command

$ openssl

req

-x509

-newkey rsa:4096

-sha256 -days 3650 \

-keyout ca.key -out ca.crt \

-subj "/CN[=www.modelCA.co](http://www.modelCA.com/O%3DModel)m[/O=Model](http://www.modelCA.com/O%3DModel) CA LTD./C=US" \

-passout pass:dees

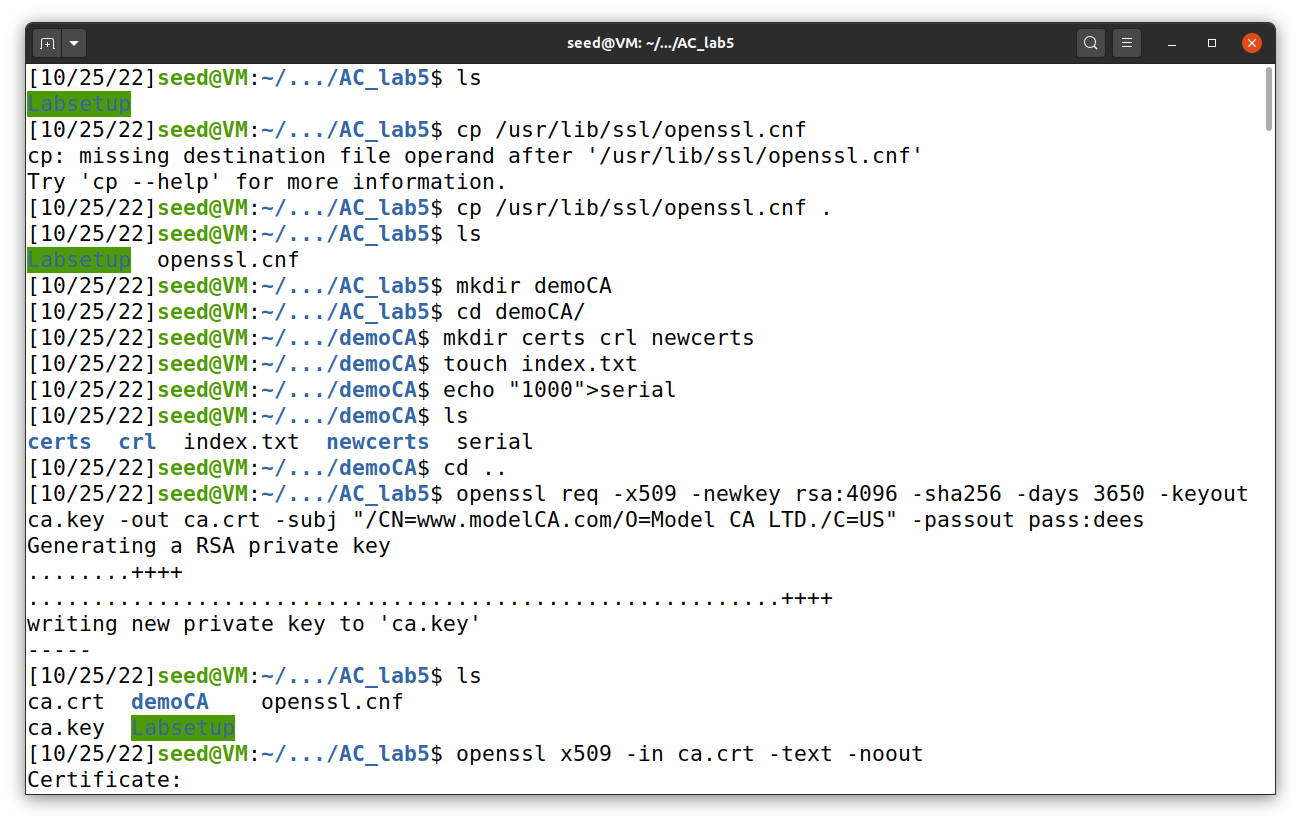
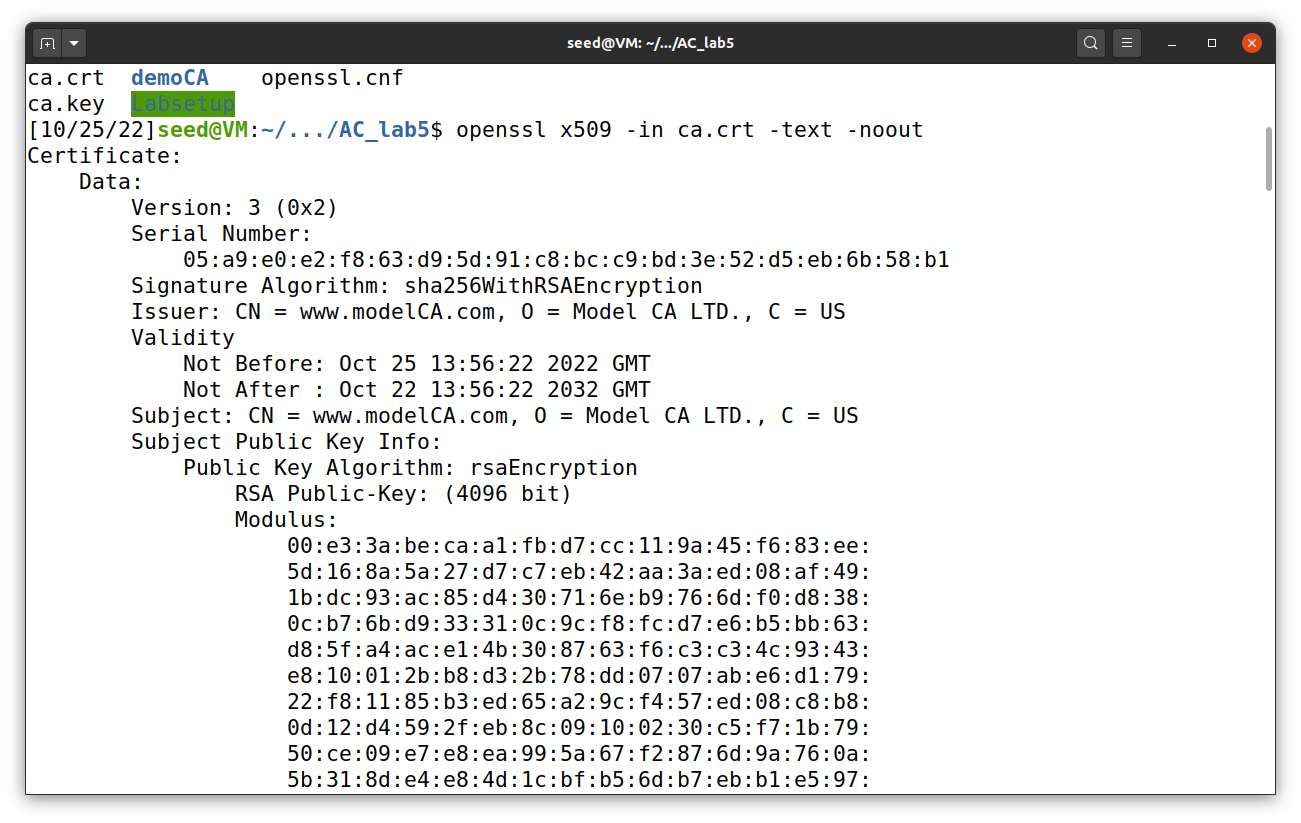
*Remember the passphrase, you'll have to use it in later tasks!*

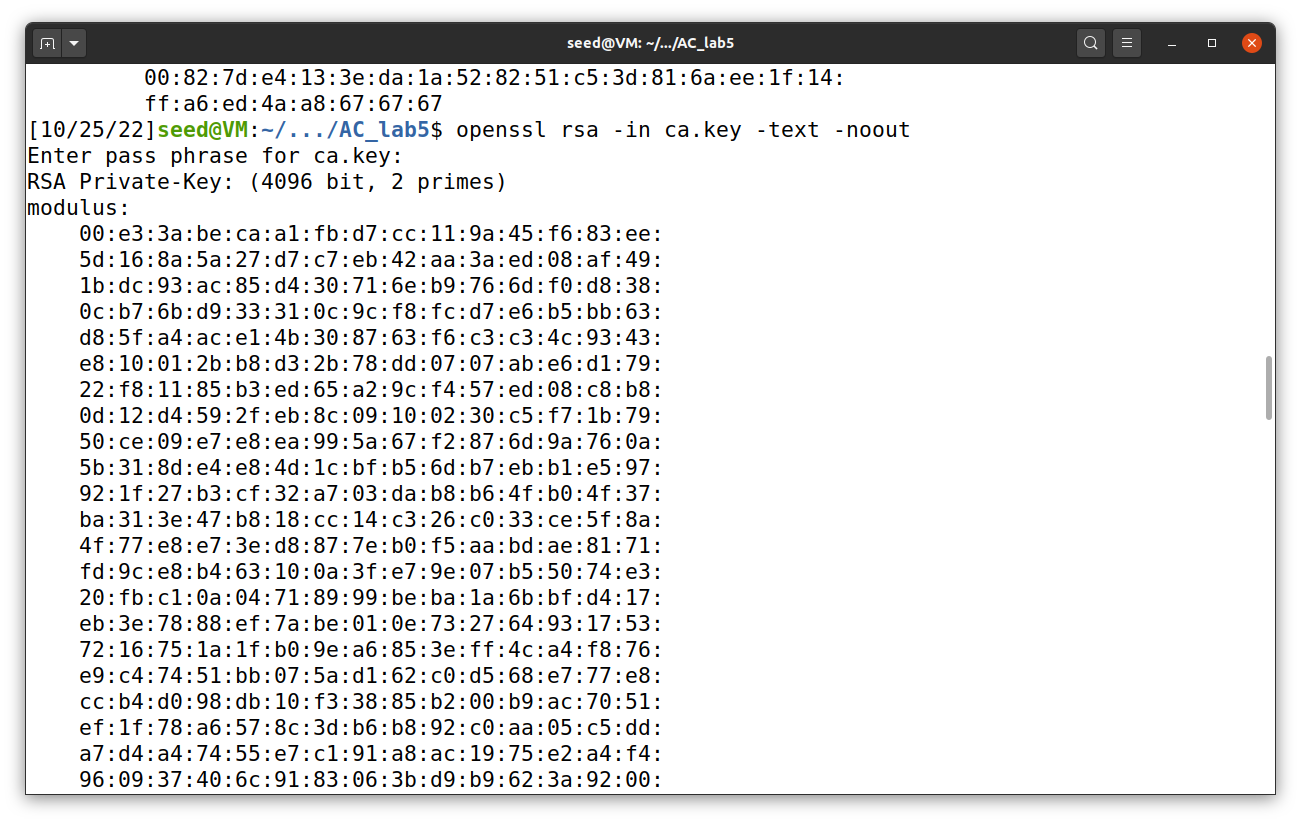
Viewing the contents of files generated

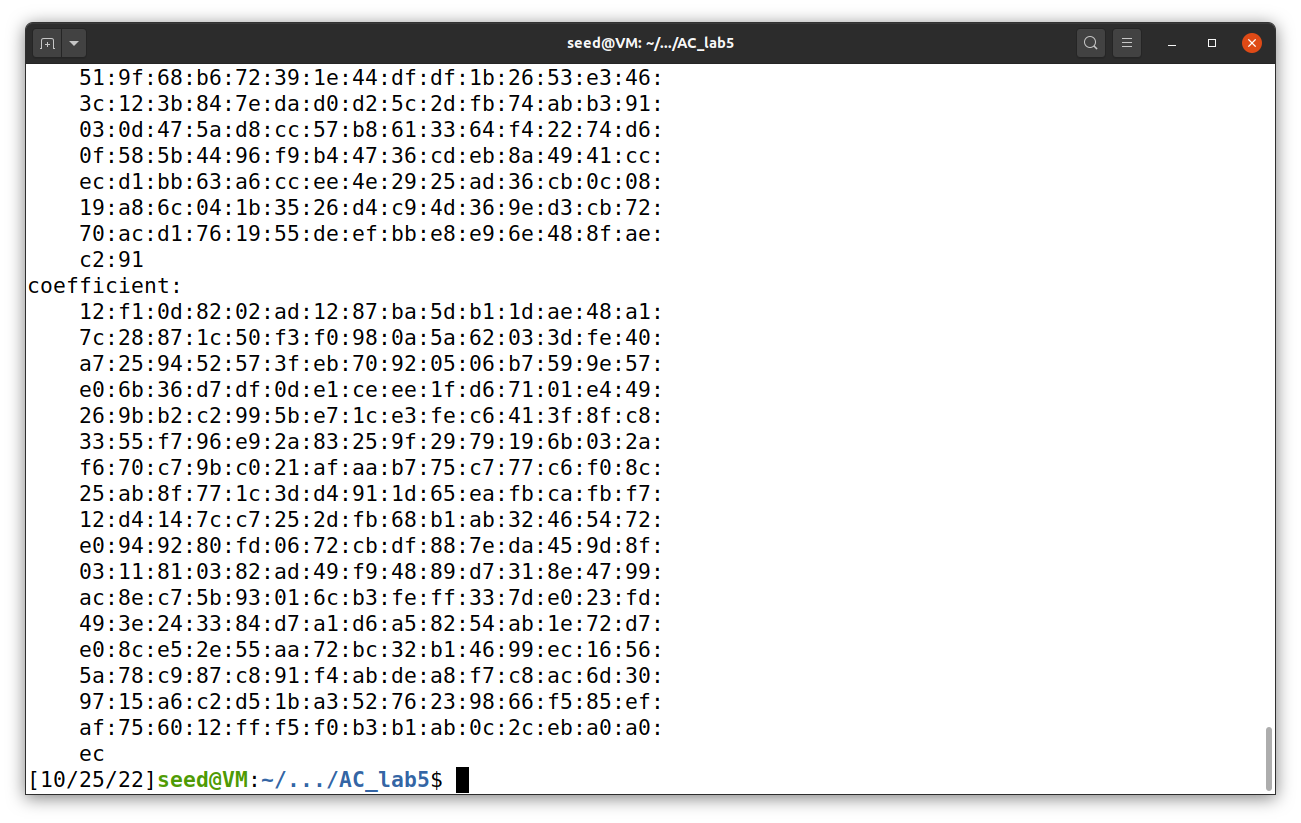
#### Commands

$ openssl x509 -in ca.crt -text -noout

$ openssl rsa -in ca.key -text -noout







**Task**

**2: Generating a**

**Certificate**

**Request for**

**the web server**

**Step 1 -**

#### Command

**Generate**

**a public/private**

**key pair**

$ openssl

req

-newkey

rsa:2048

-sha256 \

-keyout

server.key

-out

server.csr \

-subj

"/CN[=ww](http://www.bank32.com/O%3DBank32)w[.bank32.com/O=Bank32](http://www.bank32.com/O%3DBank32)

Inc./C=US" \

-passout pass:dees \

-addext

"subjectAltName =

DNS[:www.bank32.com,](http://www.bank32.com/) \

DN[S:www.bank32A](http://www.bank32A.com/).[com,](http://www.bank32A.com/) \ DN[S:www.bank32B](http://www.bank32B.com/).[co](http://www.bank32B.com/)m"

The

keys

will be

stored in

server.key

*Again, keep track of the passphrase used.*

View

the

created

file

using the

command:

$ openssl

req

-in

server.csr-text

-noout

$ openssl

rsa

-in

server.key

-text

-noout

*Take a*

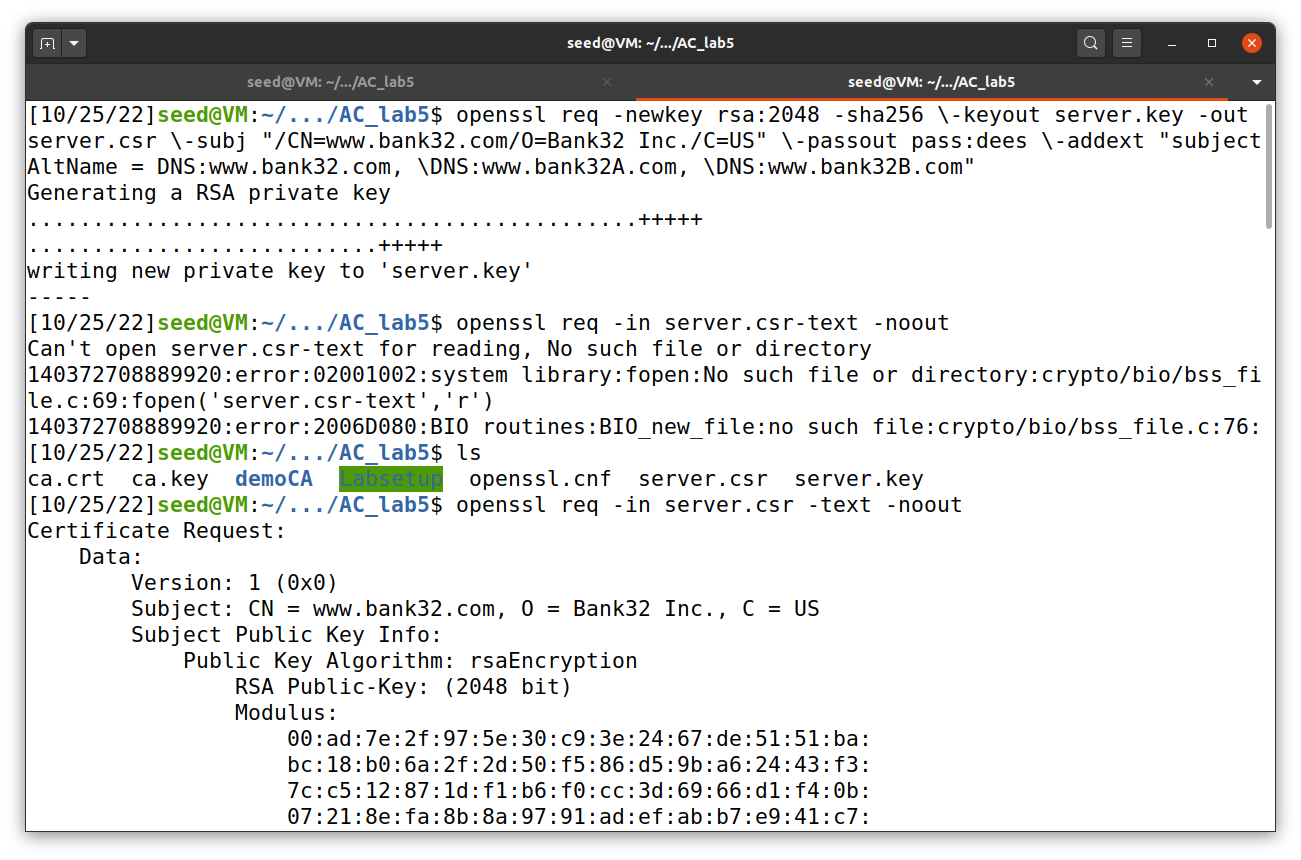
*screenshot*

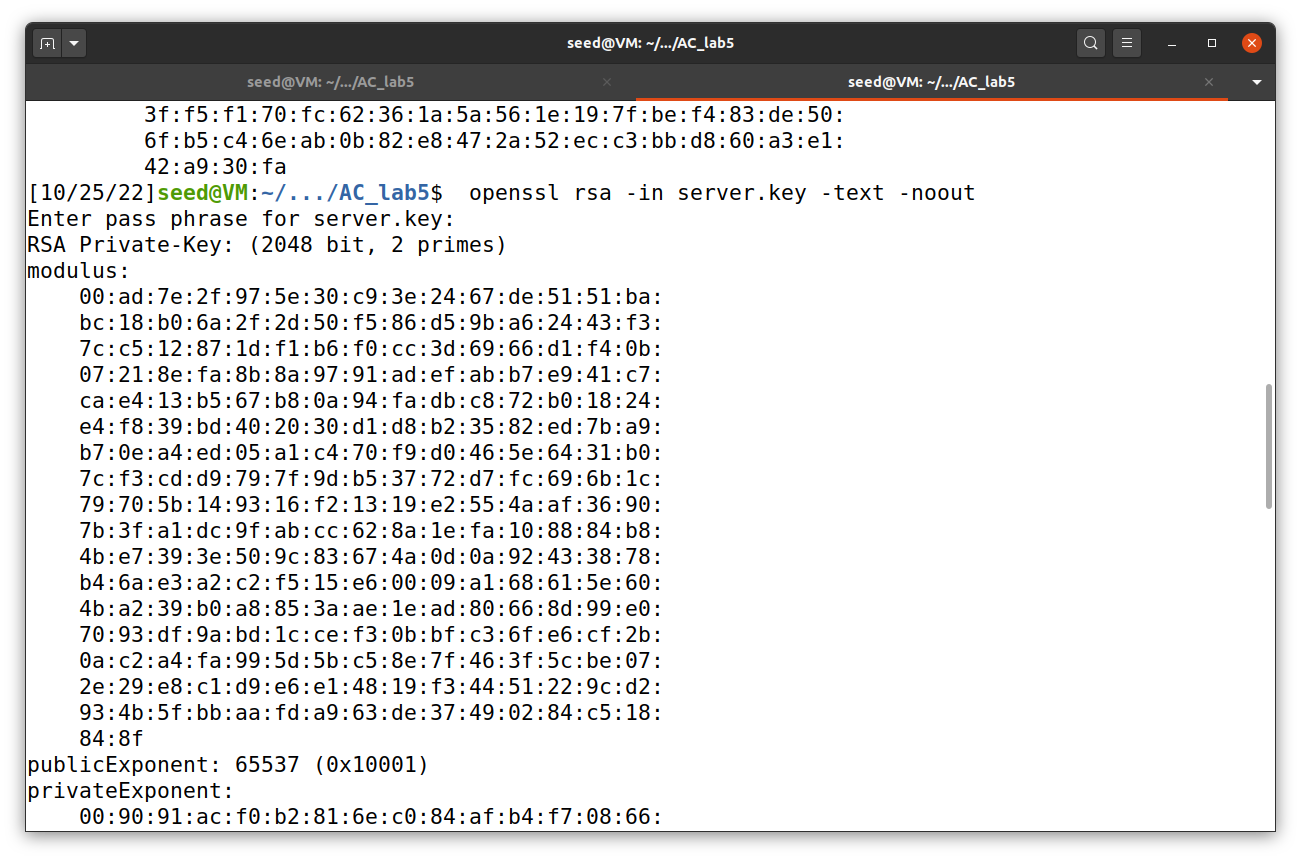
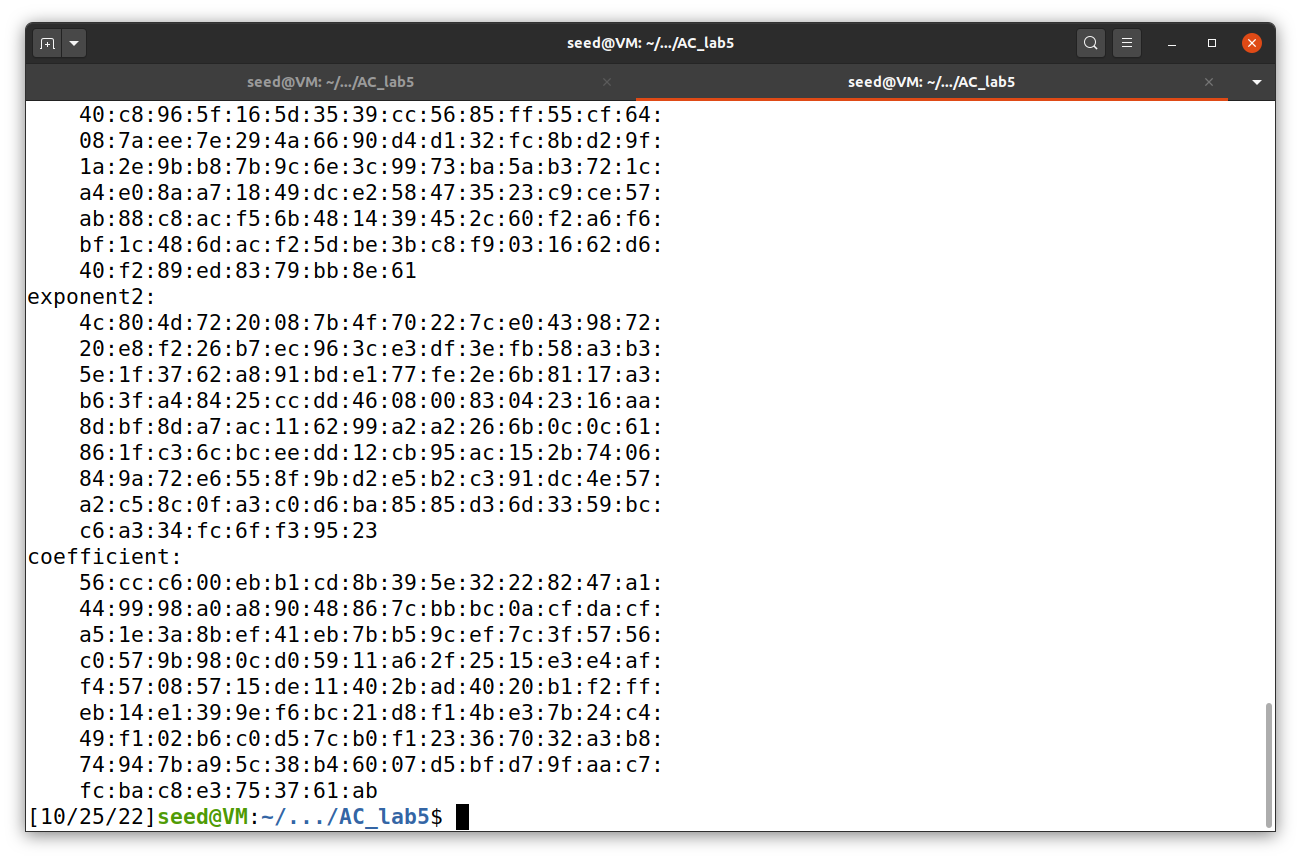
*and*

*note*

*your*

*observations*



**Task**

**3: Generating a**

**Certificate**

**for**

# your

**server**

#### Command

openssl

ca -config

openssl.cnf -policy

policy\_anything \

-md

sha256

-days

3650 \

-in

server.csr -out

server.crt

-batch \

-cert

ca.crt

-keyfile

ca.key

Viewing

#### Command

the

contents

of files

generated

$ openssl

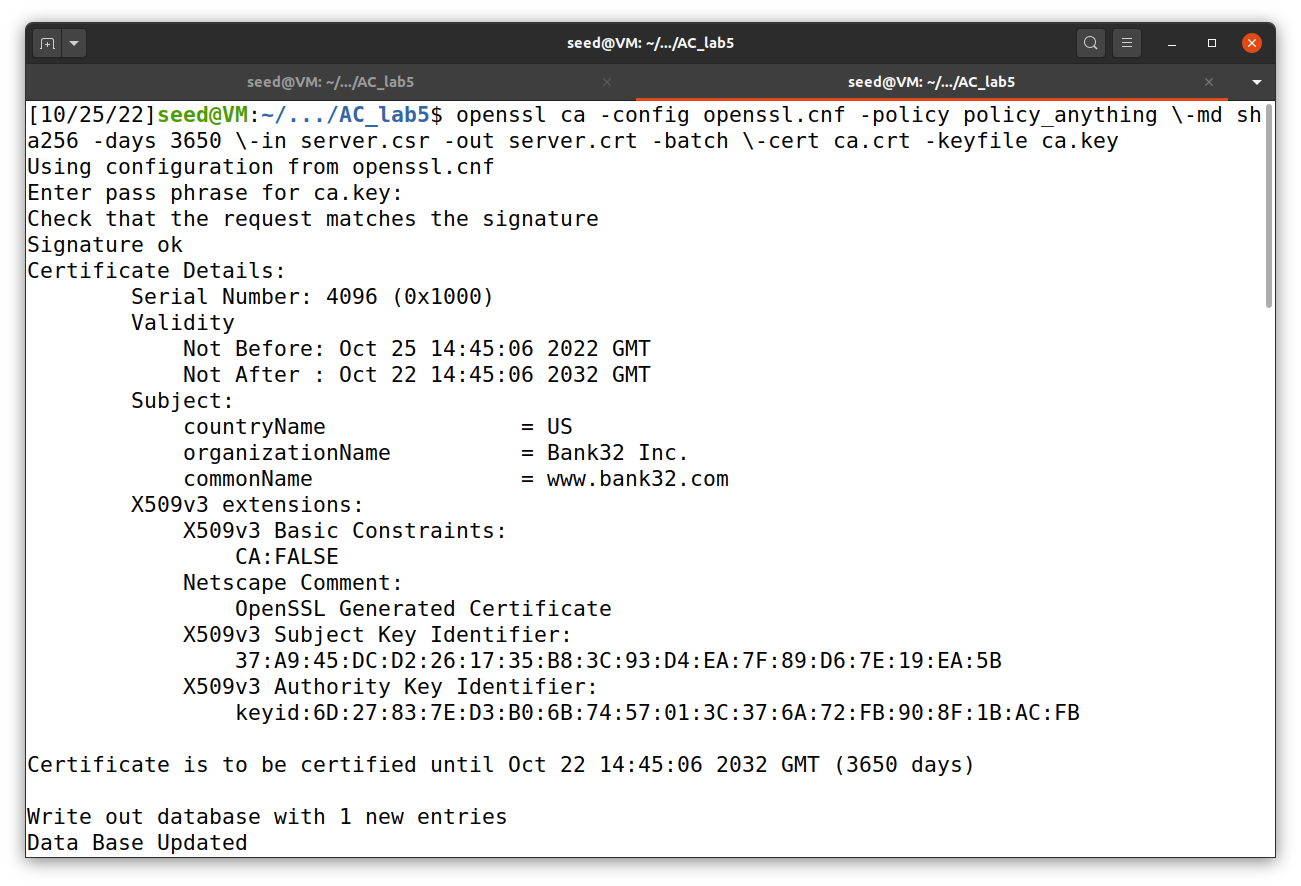
x509

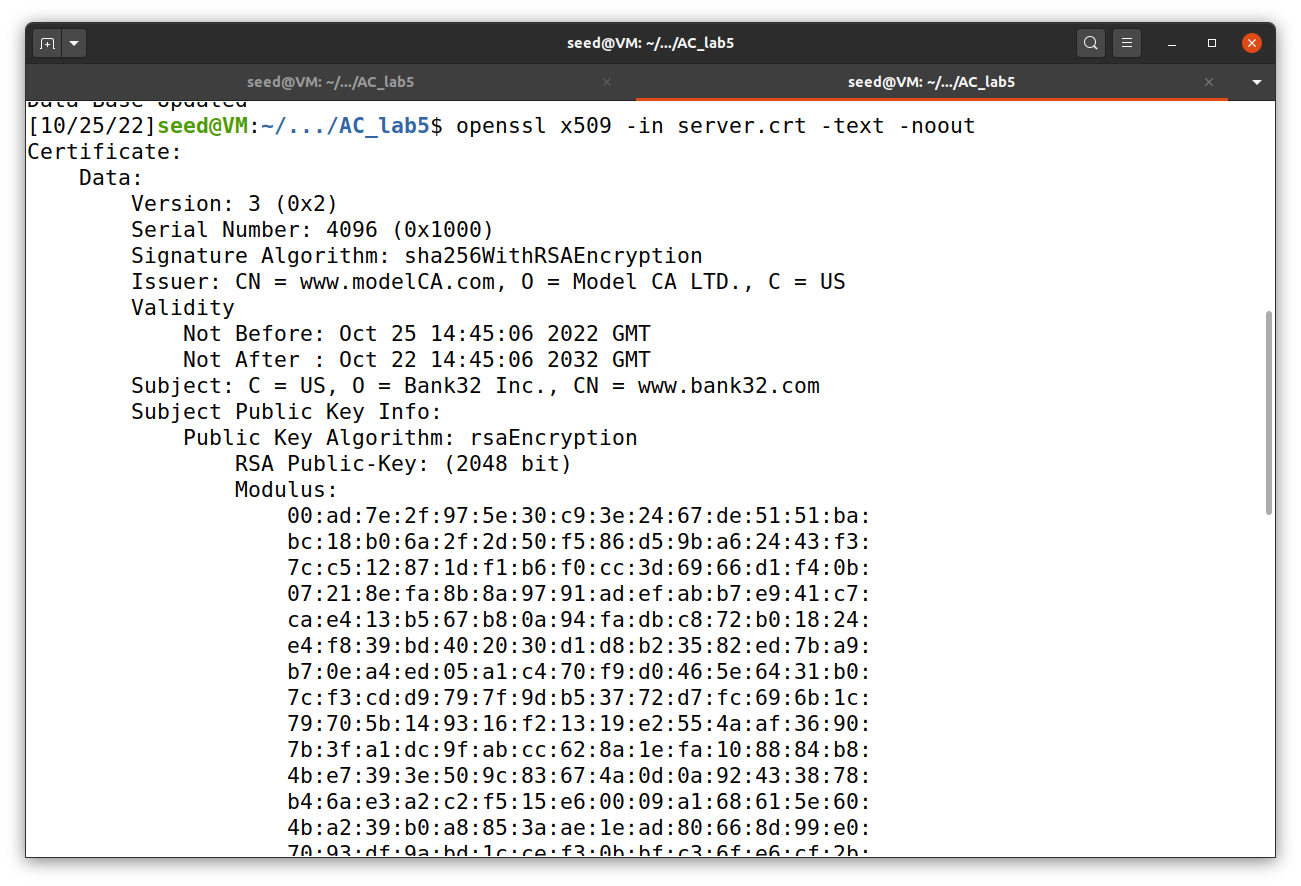
-in

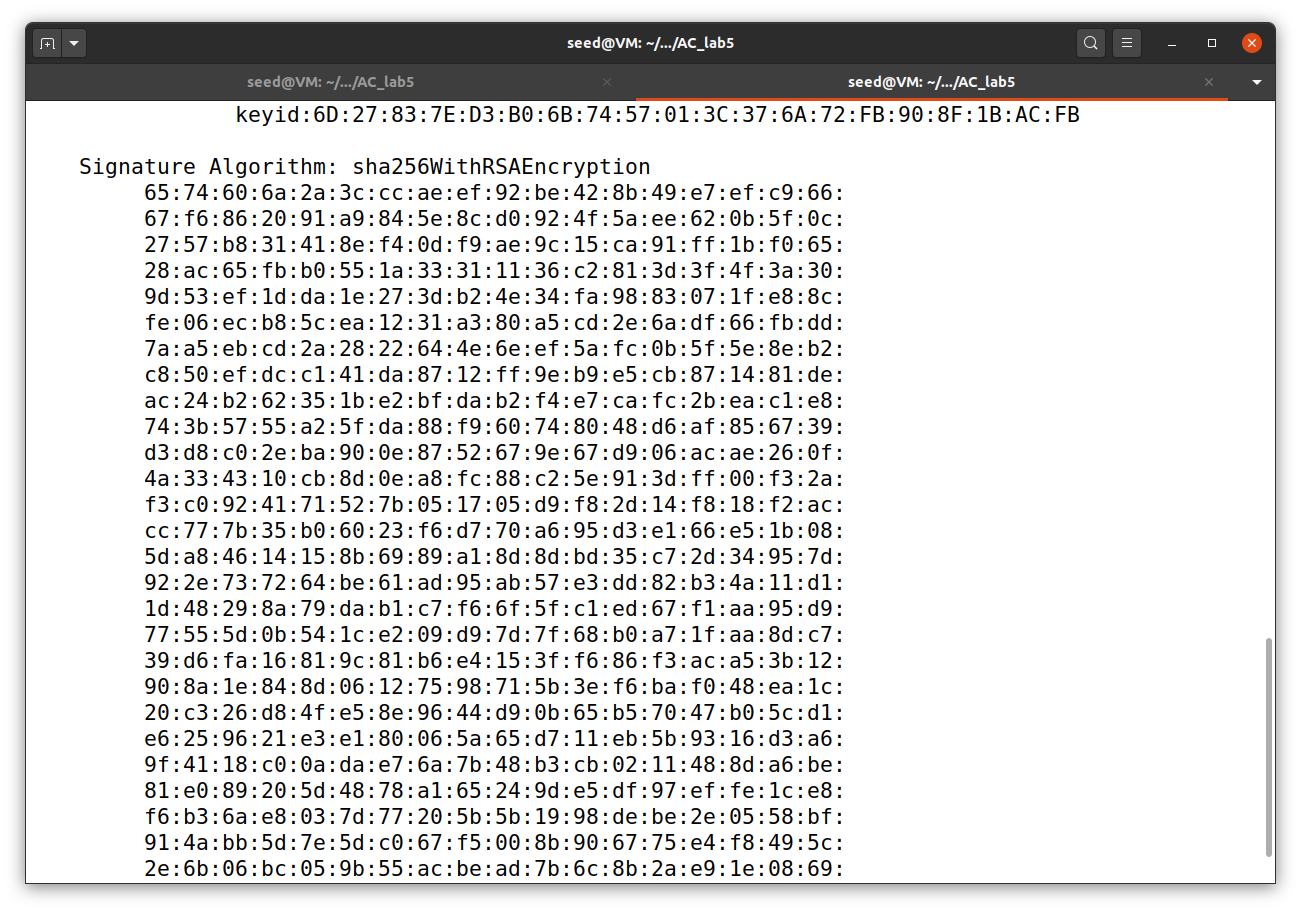
server.crt

-text

-noout







**Task**

**4: Deploying**

**Certificate in**

**an Apache-**

**Based**

**HTTPS**

**Website**

**Step**

**1 - Setting**

**up the**

**required**

**files**

Copy

the

files

server.crt,

server.key

and

ca.crt to

Labsetup/image\_www/certs

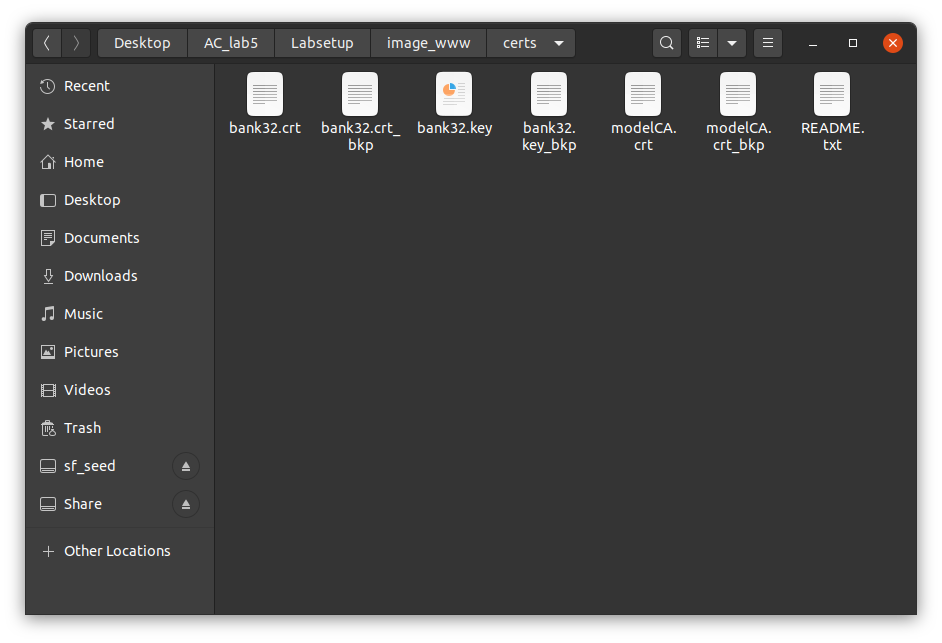
and

rename

them

to bank32.crt,

bank32.key

and

modelCA.crt

respectively.

**Step 2 -**

**Building**

**docker**

Navigate to Labsetup and run the

#### Commands

following

commands

$ docker-compose

$ docker-compose

build up

# in a

different

terminal

$ dockps

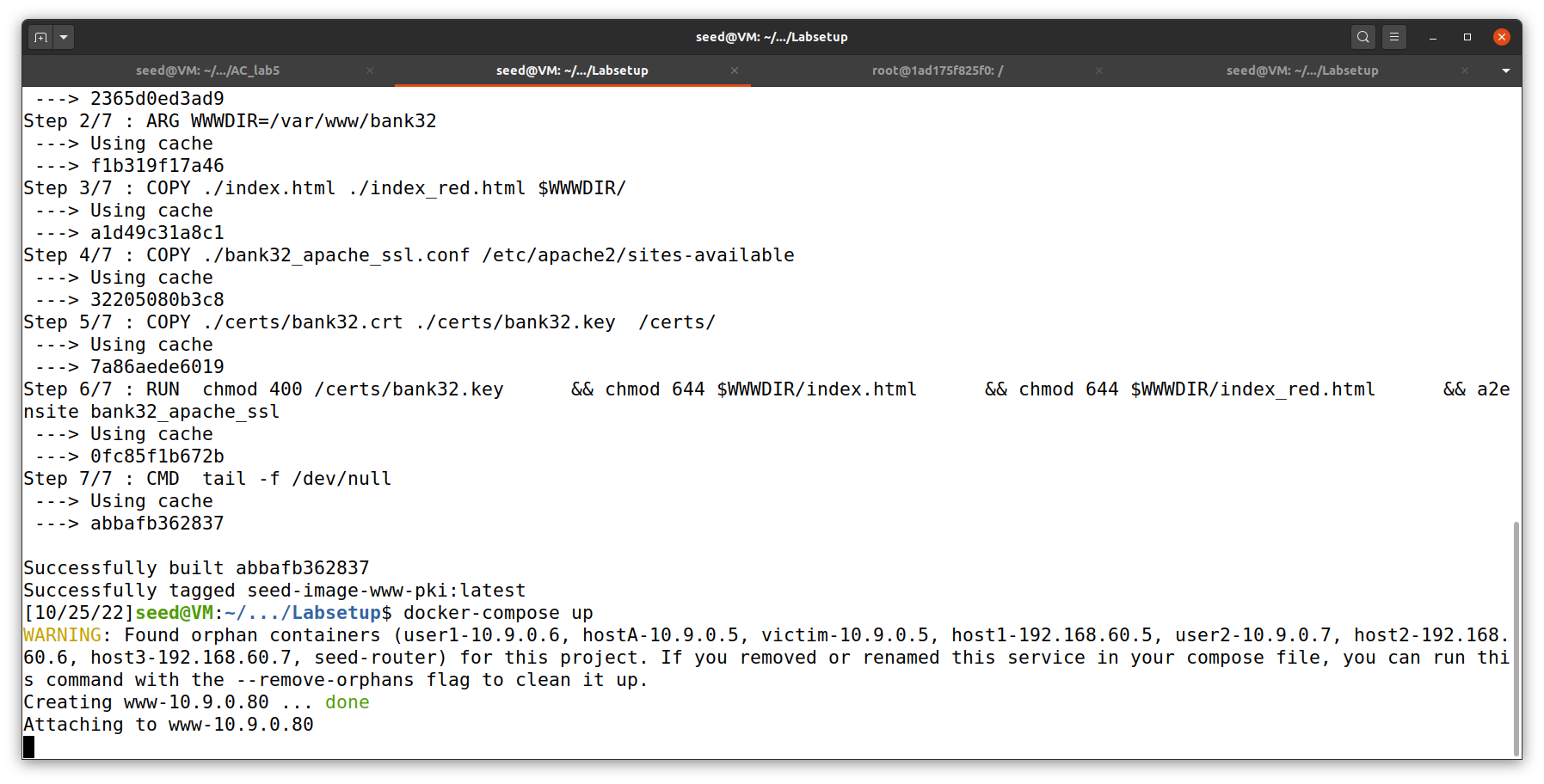
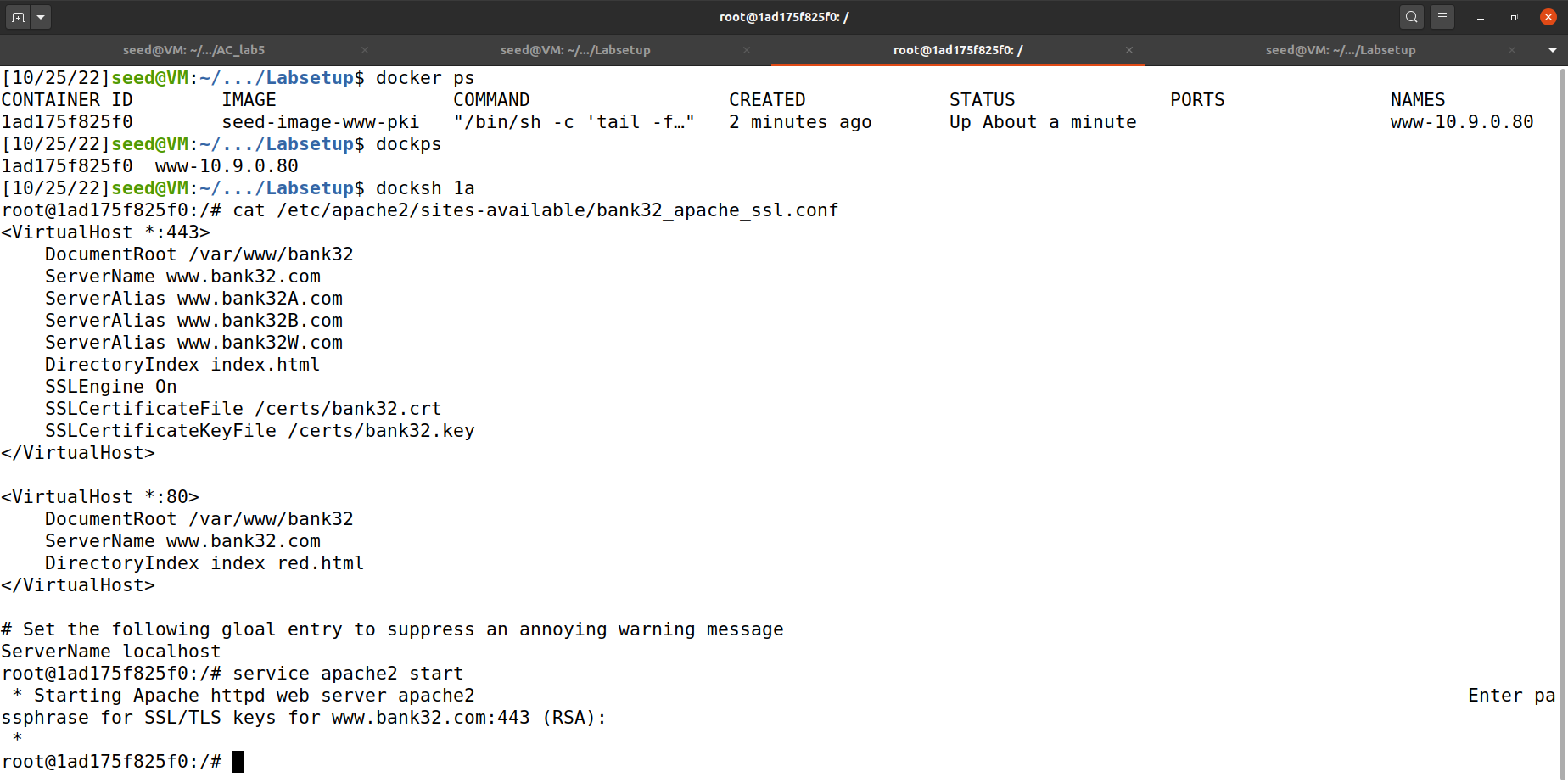
# Note

the id of

the container

$ docksh <id of container>

# Inside the docker shell

% service

### Step 3 -

apache2 start

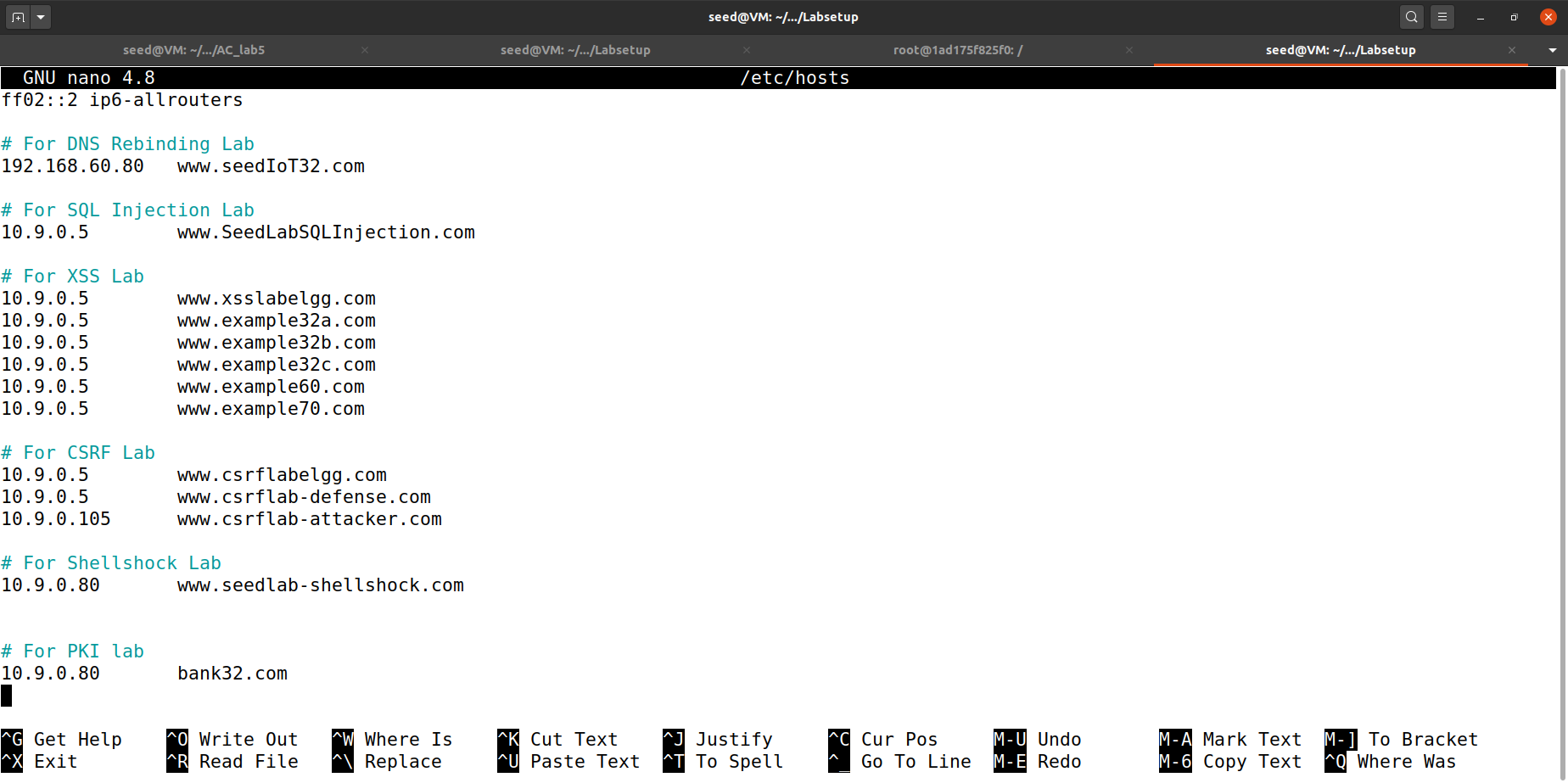
### Setting up DNS

Open /etc/hosts in a text editor as root (in the seed vm)

Add

the following entry at

the end

10.9.0.80

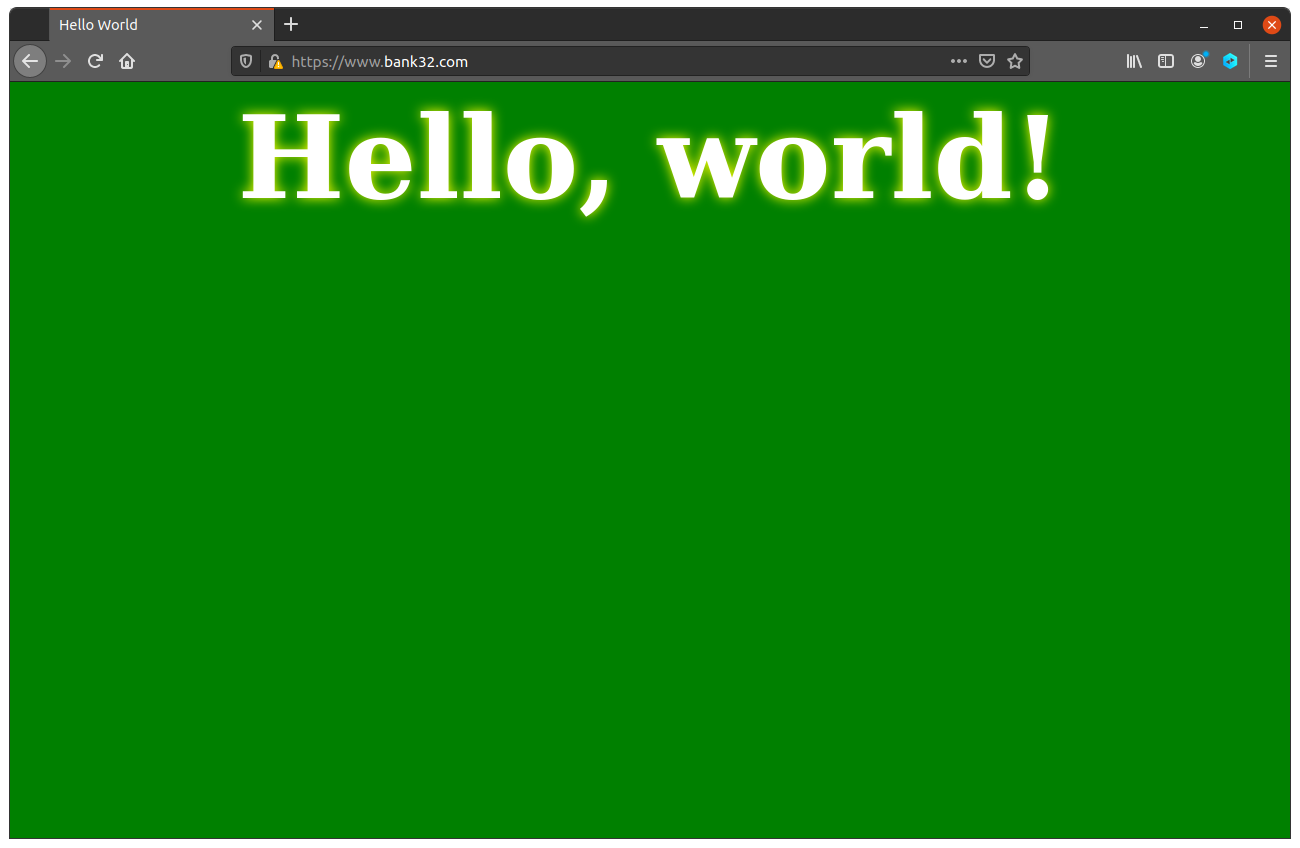
### Step 4

[www.bank32.com](http://www.bank32.com/)

Open firefox and

navigate to

[https://www.bank32.com](https://www.bank32.com/)

*Take a*

**Step 5**

*screenshot and note*

*your observations*

1. Go to about:preferences#privacy
2. At the bottom, under certificates, click on "View

Certificates", then "import"

1. Select

the

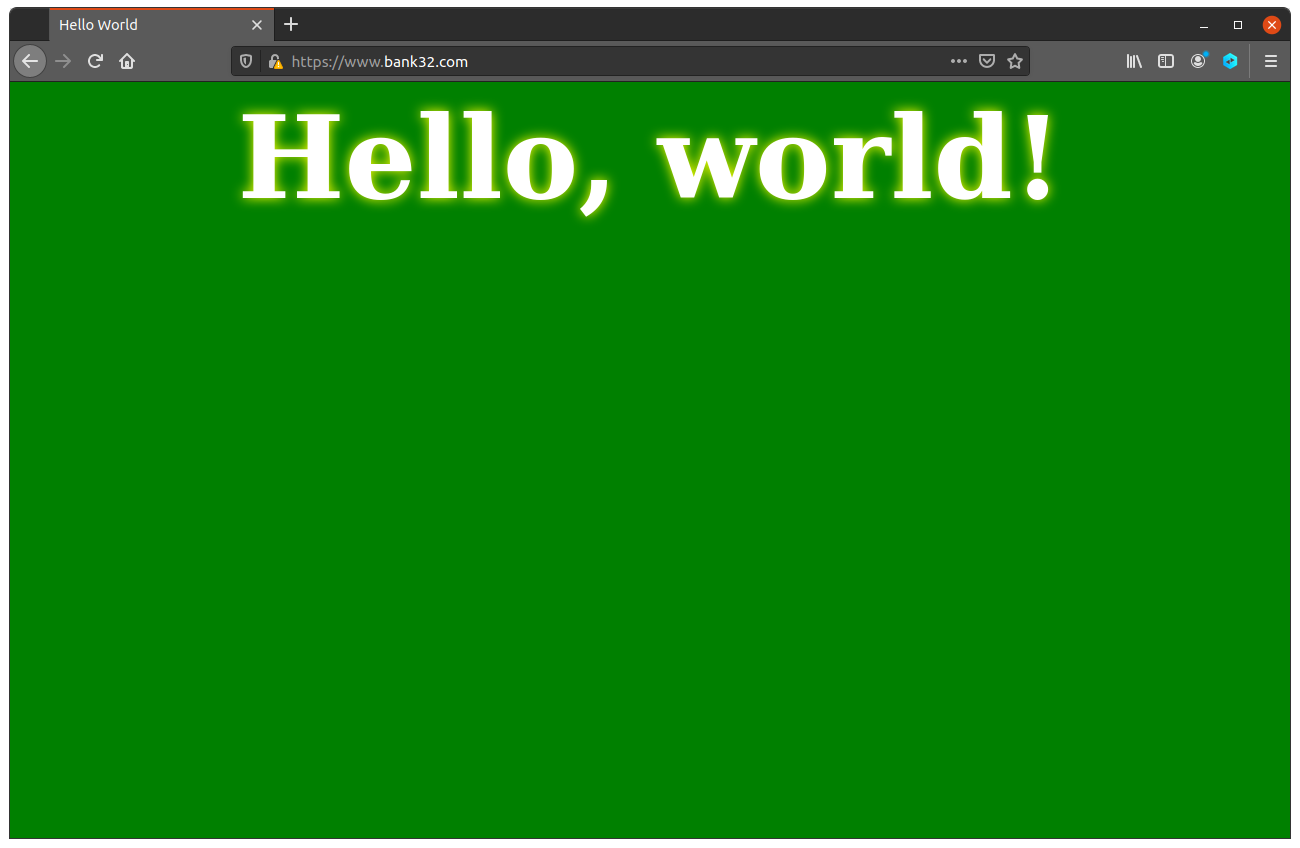
ca.crt that you

generated

and import it

1. Ensure to check the "trust this CA to identify websites"
2. Open [https://www.bank32.com](https://www.bank32.com/) again

*Take a screenshot and note your observations*



**Question**

Since bank32.com points to 10.9.0.80, if we use [https://10.9.0.80](https://10.9.0.80/) instead, we will be connecting to the same

web

server. Please do so, describe and explain

your

observations

Ans: No, it will not be leading to the Hello World page.

# Task 5: Launching a Man-In-The-Middle Attack

### Step 1: Setting up the malicious website.

In Task 4, we

have already

set up an

HTTPS

website. We will

use

the same Apache server to impersonate [www.example.com](http://www.example.com/). To achieve

that, we will

follow the instruction in

Task 4 to

add a

VirtualHost entry to Apache’s SSL configuration file: the

ServerName should be [www.example.com](http://www.example.com/), but the rest of the

configuration

can be the same

as that used

in Task 4.

### Step 2: Becoming the

**man in the**

**middle**

Add

the following entry to

the victim's

/etc/hosts file:

10.9.0.80

### Step 3 -

[www.example.com](http://www.example.com/)

### Browse the target website

Open

[https://www.example.com](https://www.example.com/) in firefox

and note your

observations.

