Desarrollo de Aplicaciones Web — **Francisco Javier Muriel Orta**

Despliegue de Aplicaciones Web — **Actividad 10 Tema 1**

****IMPORTANTE LEER POR FAVOR****

Para hacer este ejercicio había que crear una cuenta en AWS y para poder crearla tienes que poner tu tarjeta bancaria, entonces no lo he podido hacer así. Sin embargo, he realizado el ejercicio localmente desde mi máquina de Ubuntu para cumplir con el objetivo de este ejercicio que es configurar Apache con SSL, crear certificados SSL y aprender el uso de dominios y DNS dinámico.

1- **Sigue las instrucciones: https://josejuansanchez.org/iaw/practica-01-04/index.html**
Create un certificado ssl. Activa el módulo ssl, configura el virtual host ssl y comprueba que todo funciona correctamente.

Actualizamos servidor:

```
fran@FranDesplie:~$ sudo apt update && sudo apt upgrade -y
[sudo] password for fran:
Hit:1 http://es.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://es.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://es.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu noble-security InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
fran@FranDesplie:~$
```

Activamos el modulo SSL:

```
fran@FranDesplie:~$ sudo a2enmod ssl
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Enabling module socache_shmcb.
Enabling module ssl.
See /usr/share/doc/apache2/README.Debian.gz on how to configure SSL and create self-signed certificates.
To activate the new configuration, you need to run:
  systemctl restart apache2
fran@FranDesplie:~$
```

Luego, reiniciamos:

```
fran@FranDesplie:~$ sudo systemctl restart apache2
fran@FranDesplie:~$ _
```

Creamos una carpeta para los certificados:

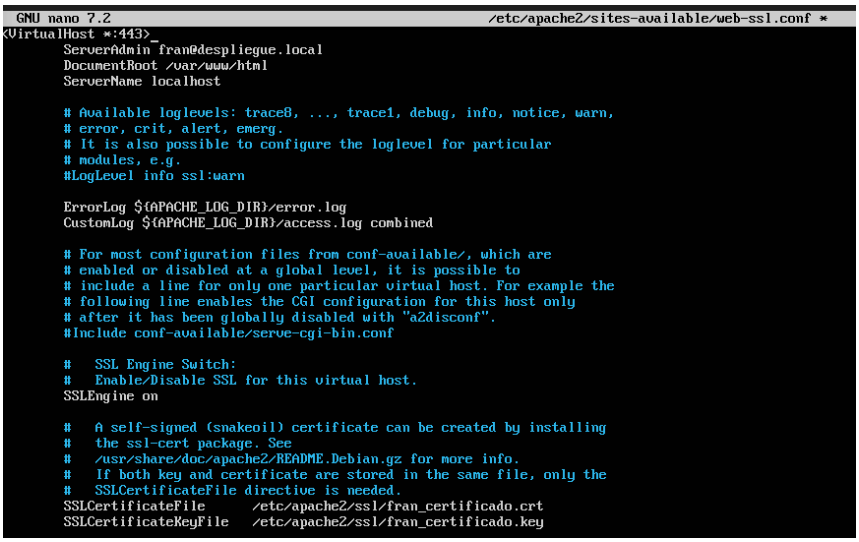
```
fran@FranDesplie:~$ sudo mkdir /etc/apache2/ssl
fran@FranDesplie:~$
```

Ahora, generamos la clave privada y el certificado autofirmado completando la información que nos va pidiendo:

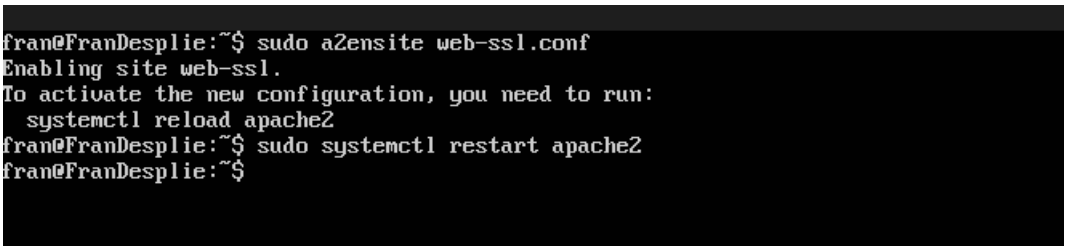
```
fran@FranDesplie:~$ sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/apache2/ssl/fran_certificado.k
o.crt
.....+....+.....+....+.....+.....+.+...+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++*.
++++++++++++++++++++++++++++++++++++++++++*.....+.....+..+.....+.....+....+...+.+...+...+.....+...+
....+.+.....+.....+...+..+...+.....+.....+.....+...+...+.....+..+.....+.....+.+.....+.....
.....+.....+.+.....+.....+...+.....+.....+.....+.+.+.....+..+...+...+.+...
.....+...+.....+.....+....+.+.....+..+...+...+...+.....+.....+....+.+.....+.....+....+..
+..+....+.....+....+..+...+.....+....+.....+.+..+.+.....+.....++++++++++++++++++++++++++++++++++++++++++++++++++++++++
....+..+...+....+..+...+...+....+.....+...+.+...+...+.+.....+.+.....+.....+.....+...+.....+.+..+...
+...+.....+.+.....++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++*.++++++++++++++++++++++++++++++++
..+.+...+......+.....................+...+.....+.+.....+...+...+.+.....+.....+.....+..+.+.....+.
.+.....+.....+..+.+.....+.....+.....+.+..+.+.+..+.....+.....+....+.+...+.....+.....+.+.
..+.....+...+.....+.+.....+....+.....+.....+.+.....+..+.....+....+.+.+....+.....+.....+..
.....+.....++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
-----
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) [AU]:ES
State or Province Name (full name) [Some-State]:Huelva
Locality Name (eg, city) []:Huelva
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Despliegue
Organizational Unit Name (eg, section) []:IT
Common Name (e.g. server FQDN or YOUR name) []:localhost
Email Address []:francisco.javier.muriel.orta@ieslaarboleda.es_
```

Configuramos el virtual host:

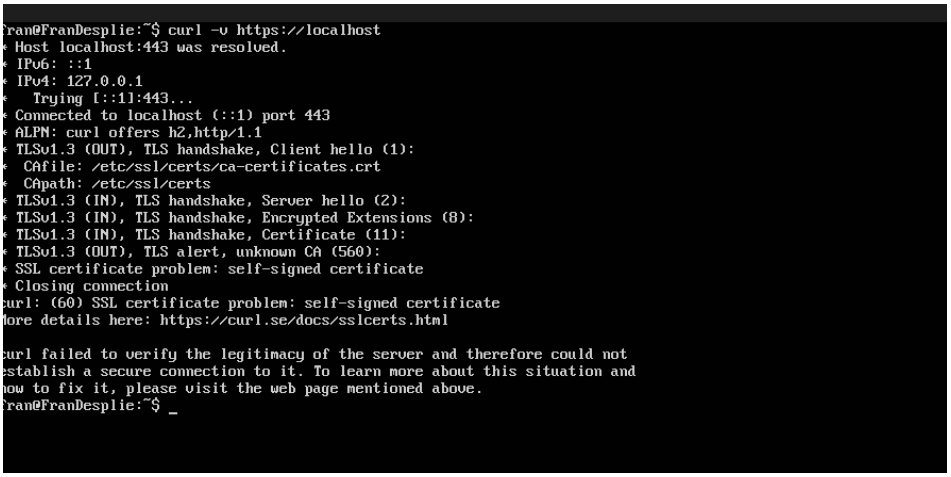
```
fran@FranDesplie:~$ sudo cp /etc/apache2/sites-available/default-ssl.conf /etc/apache2/sites-available/web-ssl.conf
fran@FranDesplie:~$ _
```



Habilitamos la web y reiniciamos:



Ahora, comprobamos con curl -v localhost que nos muestra que el procedimiento del SSL:



2- Sigue las instrucciones de: https://josejuansanchez.org/iaw/practica-https/index.html
Registrate en no-ip y configura los DNS para que apunten a la dirección pública de la instancia. A continuación instala certbot y comprueba que todo funciona correctamente. Documenta la práctica en Github.

Para esta parte he seguido todos los pasos para poder realizarla correctamente como:

Crear un hostname en No-IP (fran16121993.ddns.net).
Instalar Certbot y configurar Apache.
Intentar generar un certificado Let's Encrypt con sudo certbot --apache -d fran16121993.ddns.net (mi no-ip).

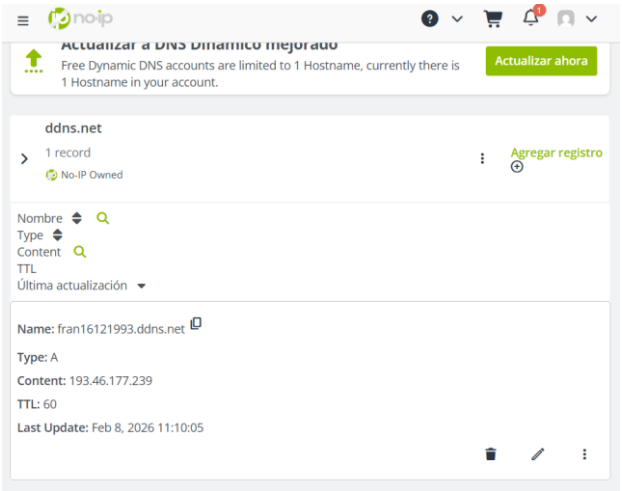
La validación no funcionó porque mi servidor está dentro de VirtualBox usando NAT, lo que significa que tiene una IP que solo funciona dentro de mi máquina. Por eso, Let's Encrypt no pudo conectarse desde internet para comprobar el dominio y emitir un certificado real.

En un entorno con servidor accesible públicamente el procedimiento funcionaría correctamente, y el certificado perfectamente.

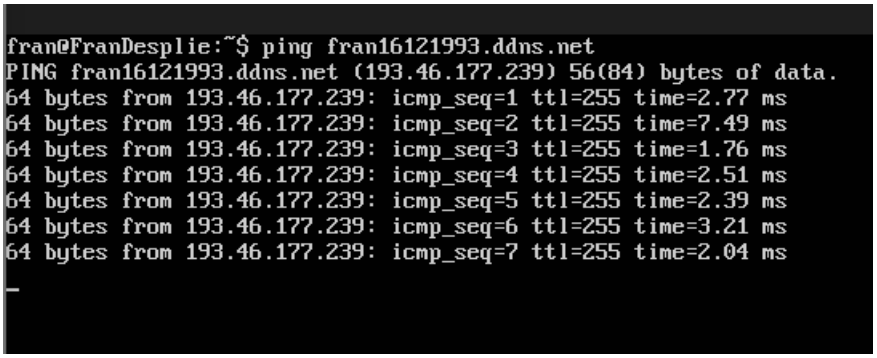
Como comenté al principio de esta tarea, no pude hacer el ejercicio con AWS porque el crear la cuenta cloud te requiere una tarjeta bancaria. Entonces, me ha servido hacerlo así porque para ambas partes he aprendido los pasos necesarios para poder realizar correctamente.

Dejo las capturas de esta parte en local:

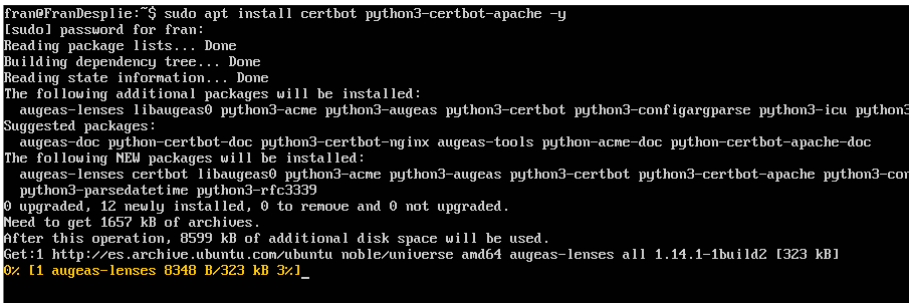
Me creé la cuenta en no-ip:



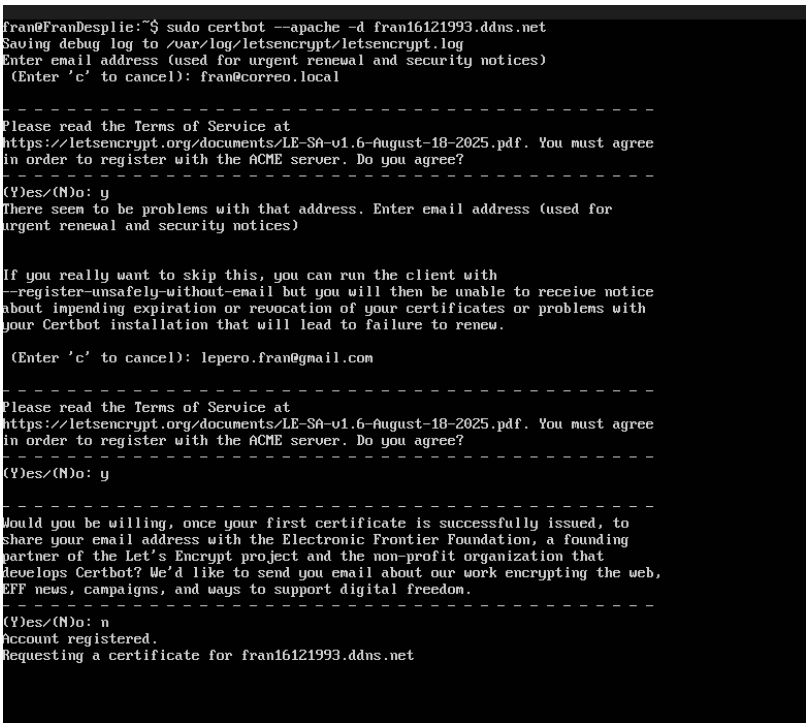
Verifico que el DNS apunte correctamente haciéndole un ping:



Instalo ahora el certbot y el plugin de apache:



Y como paso final, ejecutamos el certbot para generar el certificado:



Y después de esto, no puedo comprobar el certificado porque el servidor no es accesible desde Internet. Entonces, con AWS, funcionaría fácil al sí ser accesible

desde internet, por lo demás, la idea y los pasos que he realizado reflejan como se haría por lo que he podido aprender como es el funcionamiento.