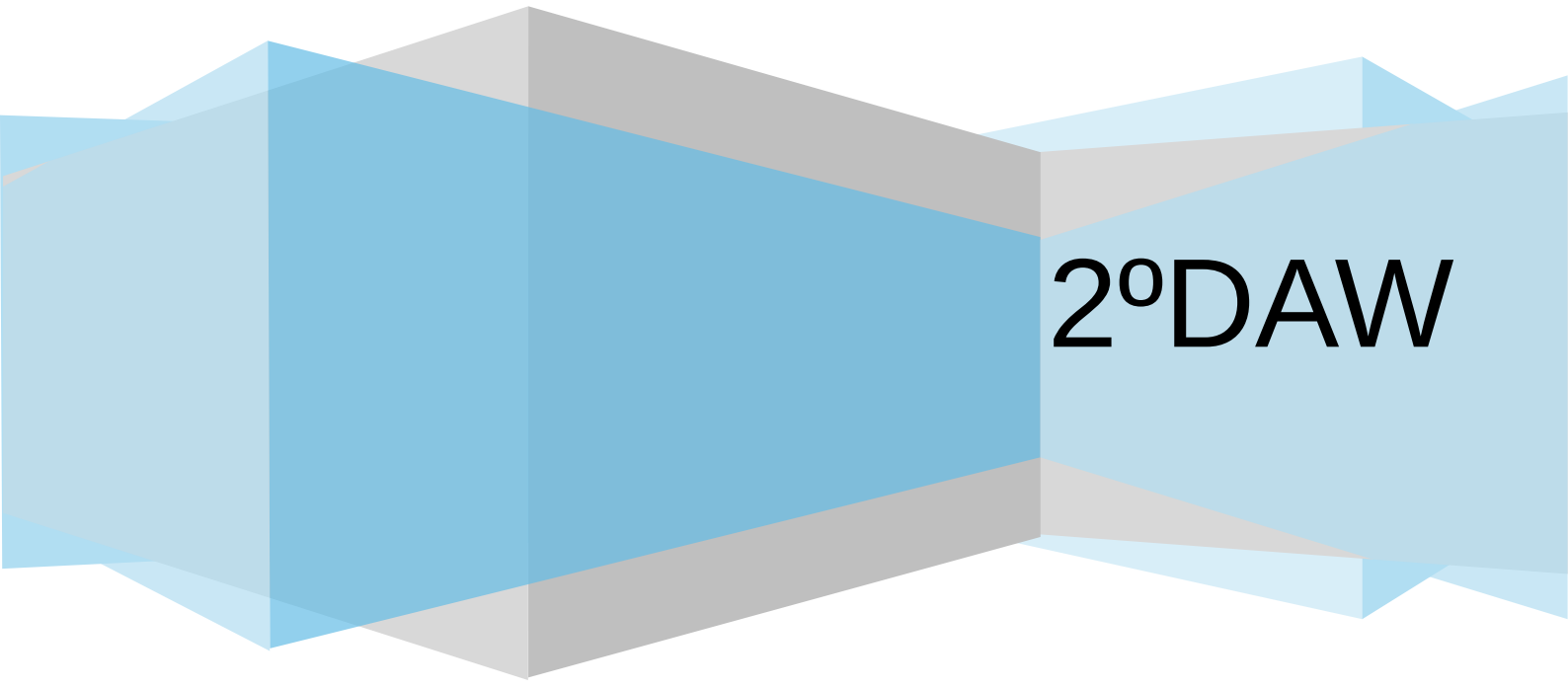
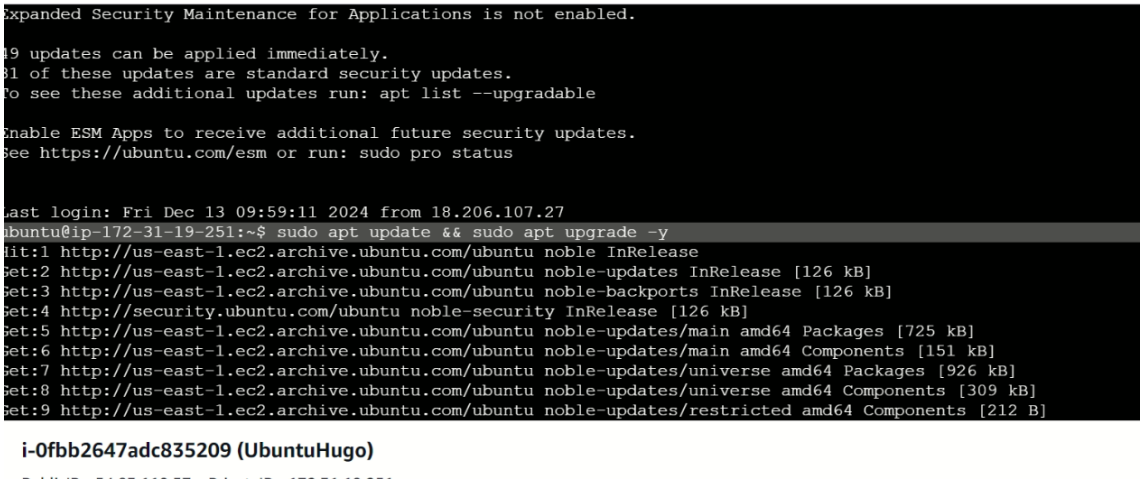
ACTIVIDAD AWS

HUGO MUÑIZ VÁZQUEZ

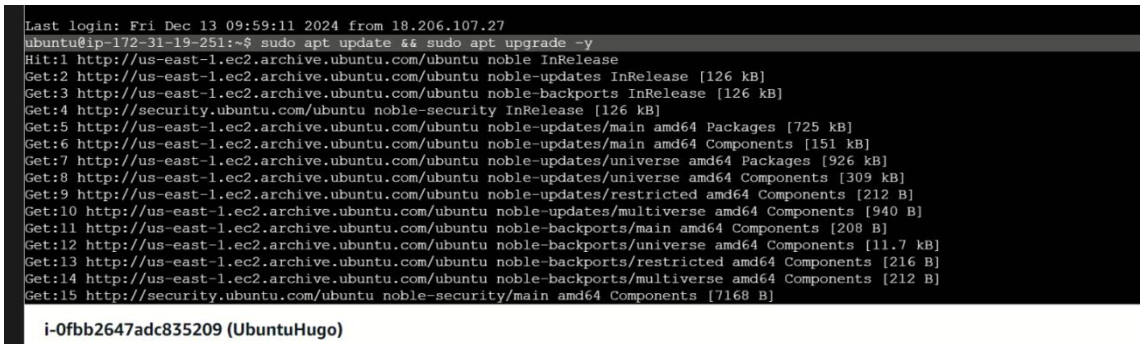


ÍNDICE

En primer lugar, actualizaremos la lista de paquetes para asegurar que todo esté correctamente y al día:

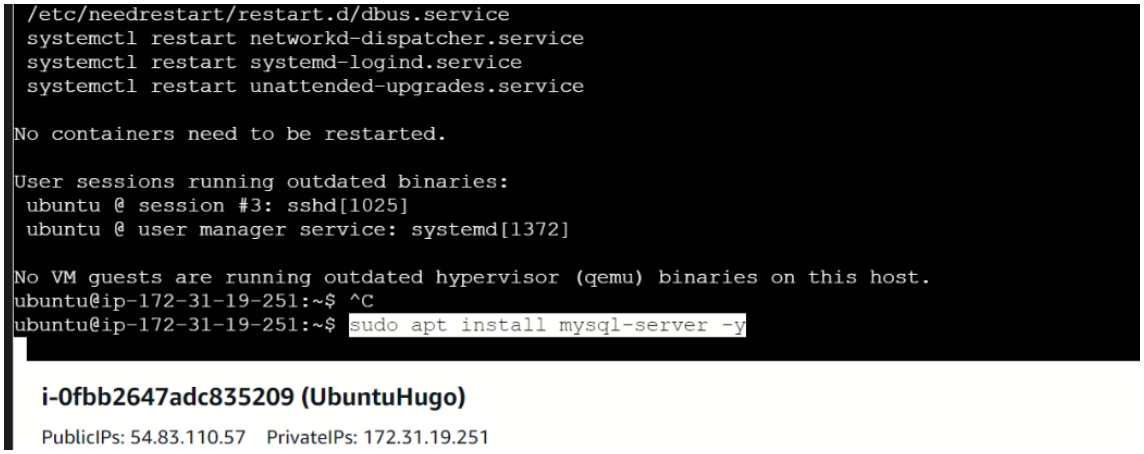


Instalamos el servidor web de Apache:

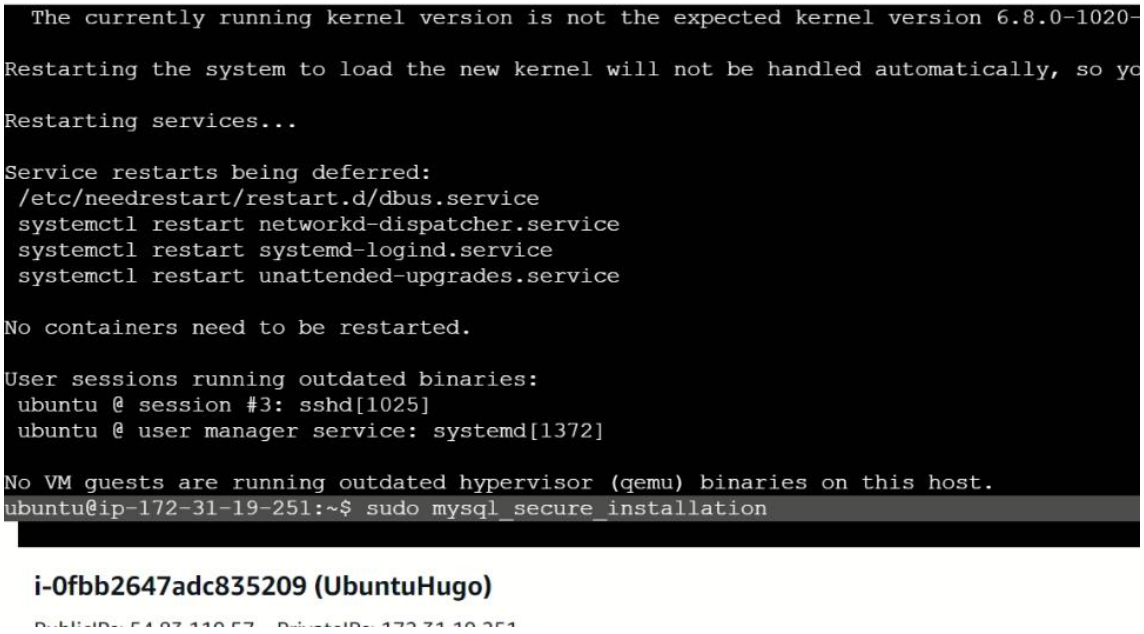


Si accedo a la ip de mi instancia ec2 de aws, podemos comprobar como sale que apache se ha configurado correctamente.

Instalaremos el servidor mysql:



Y aseguramos su instalación:



Crearemos una BD que gestionará los usuarios de apache:

```
yone can access. This is also intended only for testing,
d should be removed before moving into a production
vironment.


move test database and access to it? (Press y|Y for Yes, any other key

.. skipping.
loading the privilege tables will ensure that all changes
de so far will take effect immediately.

load privilege tables now? (Press y|Y for Yes, any other key for No) :
ccess.

l done!
untu@ip-172-31-19-251:~$ sudo mysql -u root -p
```

i-0fbb2647adc835209 (UbuntuHugo)

PublicIPs: 54.83.110.57 PrivateIPs: 172.31.19.251

```
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 11
Server version: 8.0.40-0ubuntu0.24.04.1 (Ubuntu)

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

i-0fbb2647adc835209 (UbuntuHugo)

Y añadimos usuarios a la base de datos que creamos:

```
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> CREATE TABLE users (
    ->     id INT AUTO_INCREMENT PRIMARY KEY,
    ->     username VARCHAR(50) NOT NULL UNIQUE,
    ->     password VARCHAR(255) NOT NULL
    -> );
ERROR 1050 (42S01): Table 'users' already exists
mysql> -- Insertar un usuario con contraseña hasheada
mysql> INSERT INTO users (username, password) VALUES ('usuario1', SHA2('contraseña1', 256));
Query OK, 1 row affected (0.01 sec)

mysql>
mysql> -- Puedes añadir más usuarios de manera similar
mysql> INSERT INTO users (username, password) VALUES ('usuario2', SHA2('contraseña2', 256));
Query OK, 1 row affected (0.00 sec)

mysql>
```

Instalaremos los módulos de autenticación :

```
mysql> INSERT INTO users (username, password) VALUES ('usuario1', S
Query OK, 1 row affected (0.01 sec)

mysql>
mysql> -- Puedes añadir más usuarios de manera similar
mysql> INSERT INTO users (username, password) VALUES ('usuario2', S
Query OK, 1 row affected (0.00 sec)

mysql> exit
Bye
ubuntu@ip-172-31-19-251:~$ sudo a2enmod authn_dbd
```

i-0fbb2647adc835209 (UbuntuHugo)

PublicIPs: 54.83.110.57 PrivateIPs: 172.31.19.251

1

```
mysql> exit
Bye
ubuntu@ip-172-31-19-251:~$ sudo a2enmod authn_dbd
Considering dependency dbd for authn_dbd:
Enabling module dbd.
Enabling module authn_dbd.
To activate the new configuration, you need to run:
  systemctl restart apache2
ubuntu@ip-172-31-19-251:~$ sudo a2enmod dbd
Module dbd already enabled
ubuntu@ip-172-31-19-251:~$ sudo systemctl restart apache2
```

i-0fbb2647adc835209 (UbuntuHugo)

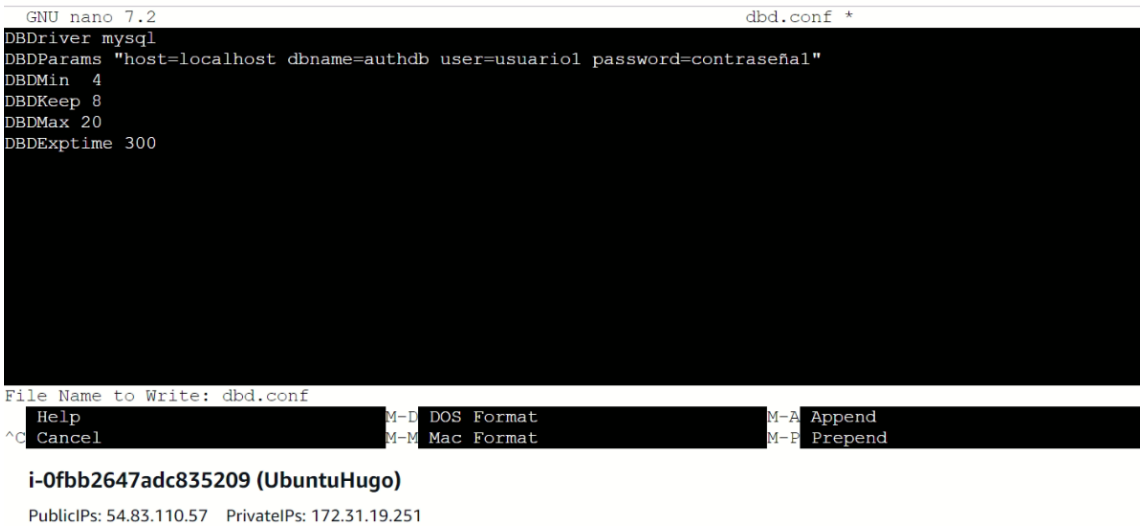
PublicIPs: 54.83.110.57 PrivateIPs: 172.31.19.251

Y reiniciamos apache.

Ahora, configuraremos la conexión con la base de datos:

Para ello en la ruta /etc/apache2/mods-available/

Crearemos el archivo dbd.conf con la siguiente configuración



Crearemos un archivo de configuración para nuestro sitio en la ruta /etc/apache2/sites-available/ llamado secure.conf con el siguiente contenido:

```
  GNU nano 7.2                                        secure.conf *
<VirtualHost *:80>
    ServerName 54.83.110.57

    DocumentRoot /var/www/html

    <Directory /var/www/html>
        Options Indexes FollowSymLinks
        AllowOverride All
        Require all granted
    </Directory>

    <Location /secure>
        AuthType Basic
        AuthName "Contenido Restringido"

        AuthBasicProvider dbd
        AuthDBDUserPWQuery "SELECT password FROM authdb.users WHERE username = %s"

^G Help        ^O Write Out   ^W Where Is    ^K Cut         ^T Execute     ^C Location    M-U Undo       M-A Set Mar
^X Exit        ^R Read File   ^\ Replace     ^U Paste       ^J Justify     ^/ Go To Line  M-E Redo       M-6 Copy
```

E instalaremos un driver necesario para que apache se comunique con mysql.

En servername pondremos la ip publica de nuestra máquina de aws.

```
authn_core.load       cgid.conf            http2.conf                mpm_prefork.conf
authn_dbd.load        cgid.load            http2.load                mpm_prefork.load
authn_dbm.load        charset_lite.load    ident.load                mpm_worker.conf
authn_file.load       data.load            imagemap.load             mpm_worker.load
authn_socache.load    dav.load             include.load              negotiation.conf
authnz_fcgi.load      dav_fs.conf          info.conf                 negotiation.load
authnz_ldap.load      dav_fs.load          info.load                 proxy.conf
authz_core.load       dav_lock.load        lbmethod_bybusyness.load  proxy.load
authz_dbd.load        dbd.load             lbmethod_byrequests.load  proxy_ajp.load
authz_dbm.load        deflate.conf         lbmethod_bytraffic.load   proxy_balancer.conf
authz_groupfile.load  deflate.load         lbmethod_heartbeat.load   proxy_balancer.load
authz_host.load       dialup.load          ldap.conf                 proxy_connect.load
authz_owner.load      dir.conf             ldap.load                 proxy_express.load
ubuntu@ip-172-31-19-251:/etc/apache2/mods-available$ nano dbd.conf
ubuntu@ip-172-31-19-251:/etc/apache2/mods-available$ sudo nano dbd.conf
ubuntu@ip-172-31-19-251:/etc/apache2/mods-available$ cd /etc/apache2/sites-available/
ubuntu@ip-172-31-19-251:/etc/apache2/sites-available$ ls
000-default.conf  default-ssl.conf
ubuntu@ip-172-31-19-251:/etc/apache2/sites-available$ sudo nano secure.conf
ubuntu@ip-172-31-19-251:/etc/apache2/sites-available$
```

i-0fbb2647adc835209 (UbuntuHugo)

PublicIPs: 54.83.110.57 PrivateIPs: 172.31.19.251

Una vez guardado nuestro archivo, habilitaremos el sitio,reiniciaremos apache y comprobaremos.

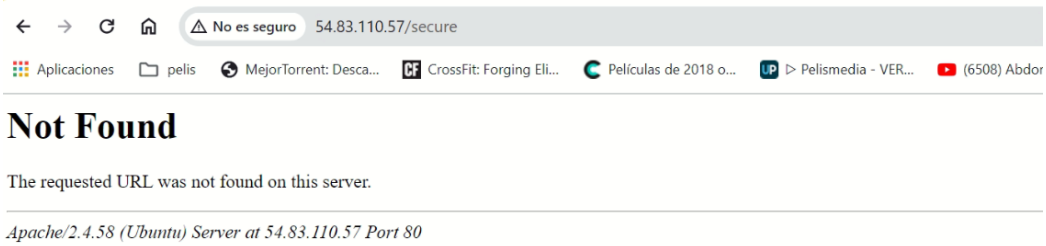
```
buntu@ip-172-31-19-251:/etc/apache2/sites-available$ sudo nano /etc/apache2/sites-
buntu@ip-172-31-19-251:/etc/apache2/sites-available$ sudo a2ensite secure.conf
nabling site secure.
o activate the new configuration, you need to run:
 systemctl reload apache2
buntu@ip-172-31-19-251:/etc/apache2/sites-asudo a2ens^C
buntu@ip-172-31-19-251:/etc/apache2/sites-available$ sudo systemctl reload apache2
buntu@ip-172-31-19-251:/etc/apache2/sites-available$
```

i-0fbb2647adc835209 (UbuntuHugo)

PublicIPs: 54.83.110.57 PrivateIPs: 172.31.19.251

Si intentamos acceder a la dirección ip de nuestra máquina /secure, nos solicita unas credenciales de acceso:

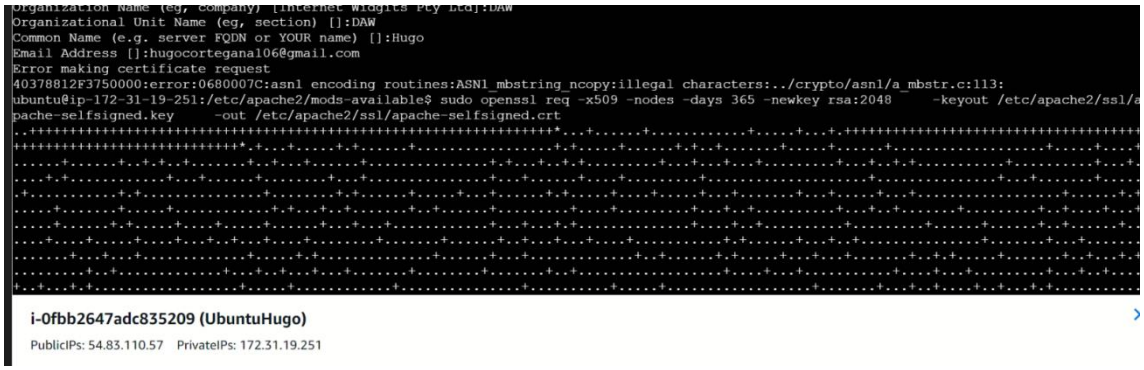
NOTA: Se tuvo que almacenar la contraseña mediante un hash, en la BD ya que apache no permite la autenticación de texto plano de contraseñas.

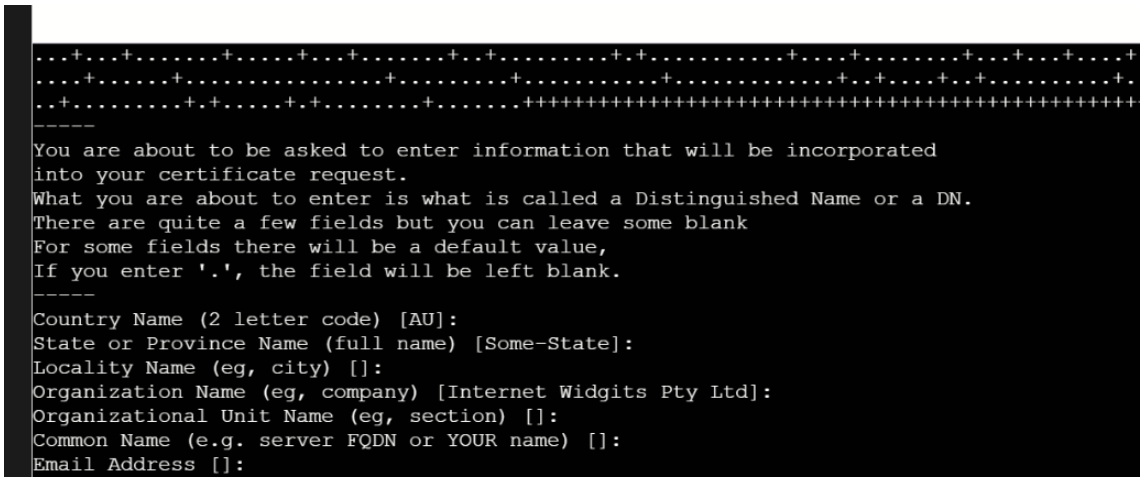


Una vez superada la autenticación se nos muestra un NotFoundya que, en nuestra configuración, no se ha establecido ninguna página para servir.

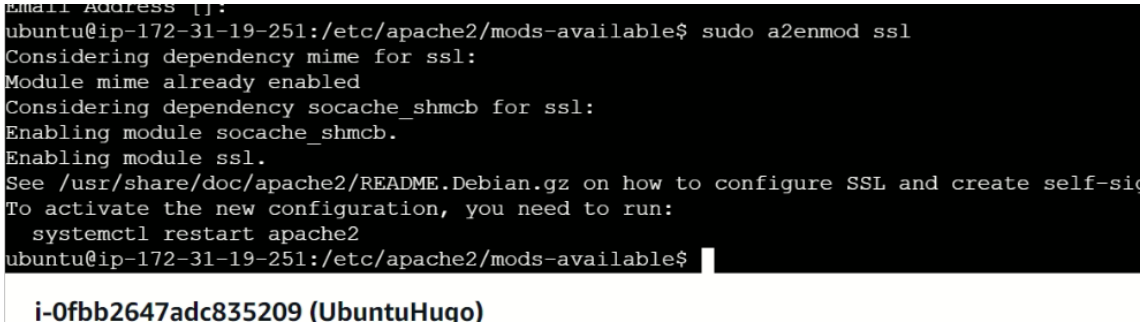
Para crear un certifica autofirmado y activar ssl:

Crearemos un directorio dentro de apache en el que generaremos nuestro certificado:

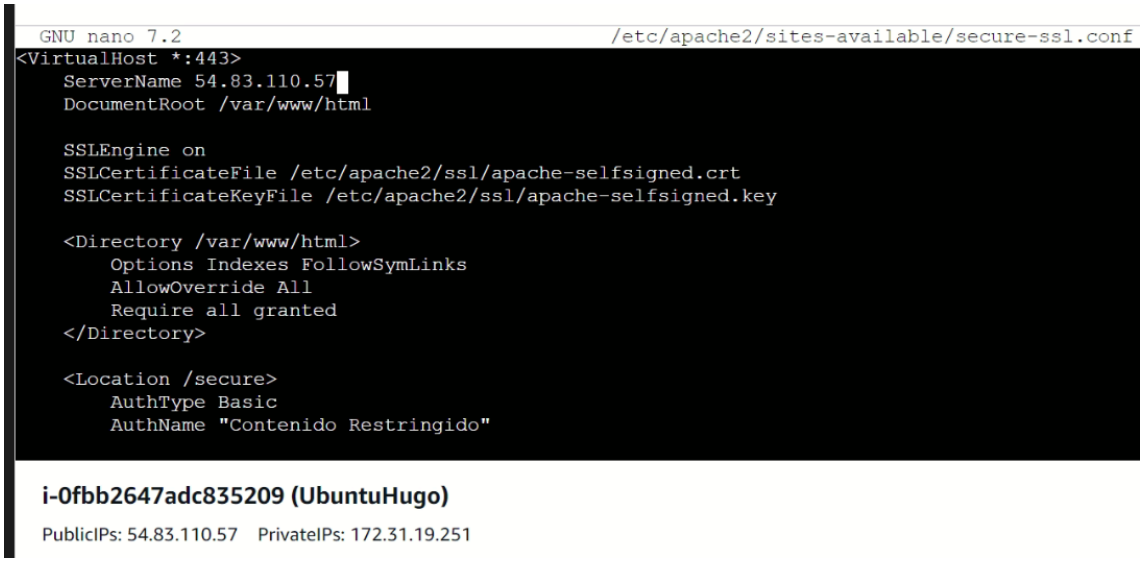




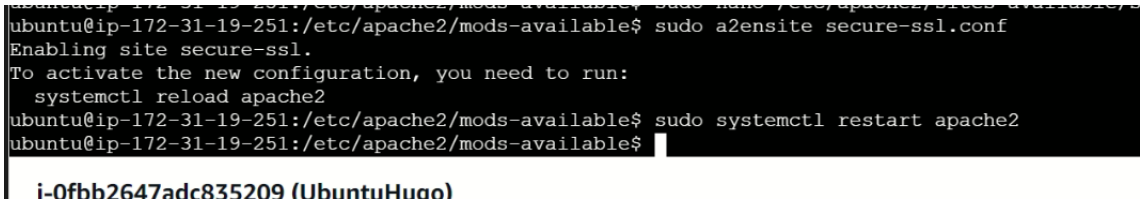
Habilitaremos ssl:



Crearemos un virtualhost para https:



Habilitamos el sitio y reiniciamos apache:



Al acceder nos pide de nuevo las credenciales, pero ahora se emplea el protocolo HTTPS sobre el puerto 443. Si pulsamos en los detalles del certificado, vemos como la fecha de emisión corresponde al día en el que se ha realizado la actividad práctica. Los navegadores modernos, lo catalogan como no seguro ya que al ser un certificado autofirmado, es decir, no firmado por una autoridad certificadora reconocida, lo catalogan como ilegítimo.