ADMINISTRACIÓN REMOTA DE SERVIDORES WEB EN AWS A TRAVÉS DE SSH

Hamza Akdi

Arquitectura en la Nube

09/11/2025

Índice

1. Preparación del entorno local (WSL)

1.1 Creación de directorio para las claves SSH

Necesitaremos crear un directorio con el siguiente comando para guardar la clave privada (.pem).

```
hamza@A6Alumno08:~$ mkdir -p ~/ .ssh
```

A continuación, daremos al directorio **~/ .ssh** permisos para que solo nosotros tengamos acceso a este directorio donde almacenaremos la clave.

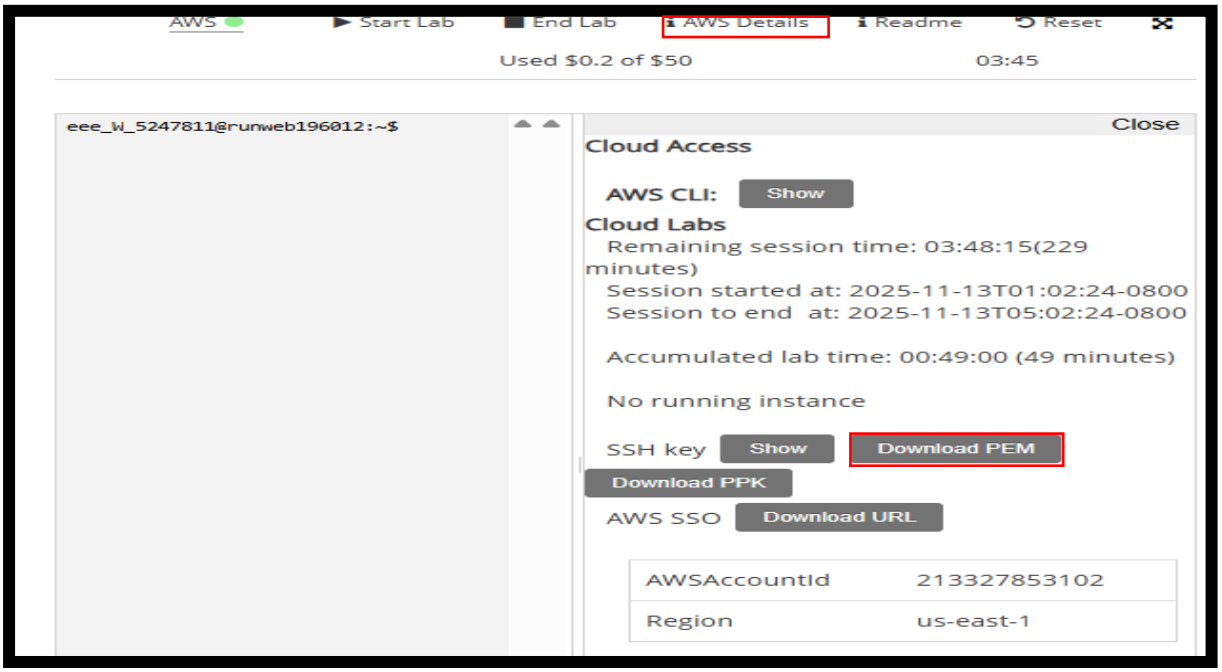
```
hamza@A6Alumno08:~$ chmod 700 ~/ .ssh
```

2. Configuración en AWS (Interfaz Visual)

El archivo **.pem** (clave privada) es como la llave de casa que te permite conectarte de forma segura a tu servidor AWS.

2.1 Descarga y ubicación de clave PEM

Para comenzar con la configuración, debemos descargar la clave PEM. Esta se encuentra en la página del laboratorio.



(Descarga de clave PEM)

Una vez descarga la clave, la guardaremos en una ubicación segura. En la siguiente ruta:

```
root@A6Alumno08:/home/hamza# cp /mnt/c/labsuser.pem ~/.ssh/
```

2.2 Configurar permisos de la clave PEM

El archivo **.pem** debe tener permisos muy restrictivos (400) para que SSH acepte usarlo.

```
root@A6Alumno08:/home/hamza# chmod 400 ~/.ssh/labsuser.pem
```

Verifica los permisos:

A continuación, veremos los permisos que dimos anteriormente.

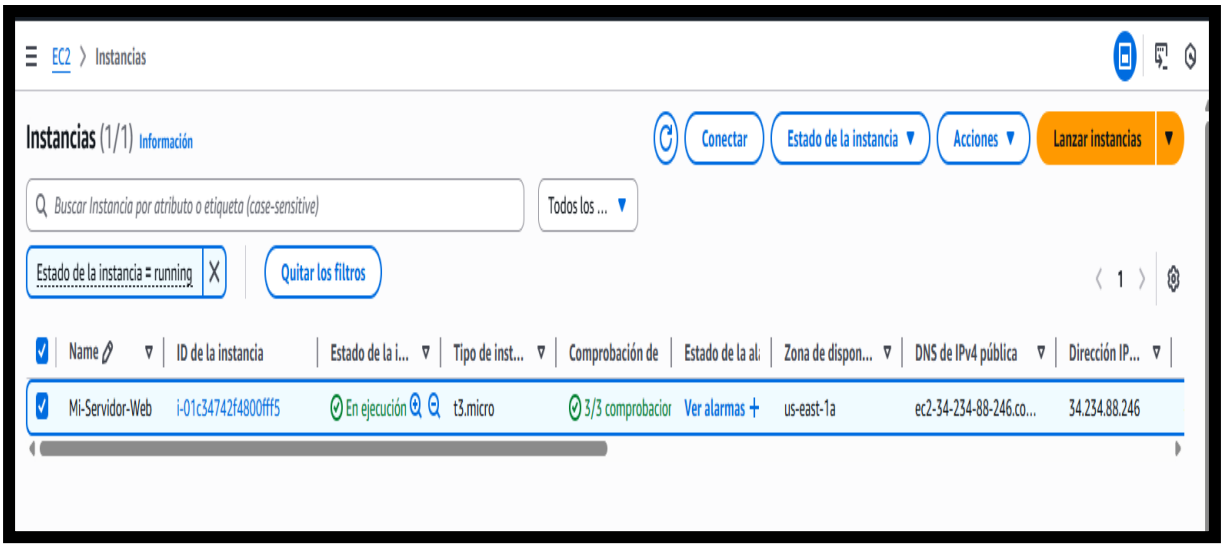
```
root@A6Alumno08:/home/hamza# ls -la ~/.ssh/labsuser.pem
-r-------- 1 root root 1678 Nov 13 10:48 /root/.ssh/labsuser.pem
```

3. Crear instancia EC2

Ahora, procederemos a crear una **Instancia EC2** y su guardaespaldas (Security Group) en la consola de AWS con las siguientes características:

- Nombre: servidor-web-practica

- 2. AMI: Ubuntu 22.04 LTS o Ubuntu 24.04 LTS
- 3.Tipo: t2/3.micro (Free Tier)
- 4. Red: VPC y Subred por defecto
- 5. IP pública: Habilitada
- 6. Almacenamiento: 8 GiB, tipo gp2
- 7.Etiquetas (opcional): Curso → ArquitecturaNube
- 8. Grupo de Seguridad: sg-servidores-web



(Instancia EC2 creada)

4. Configurar Security Group (Reglas de entrada)

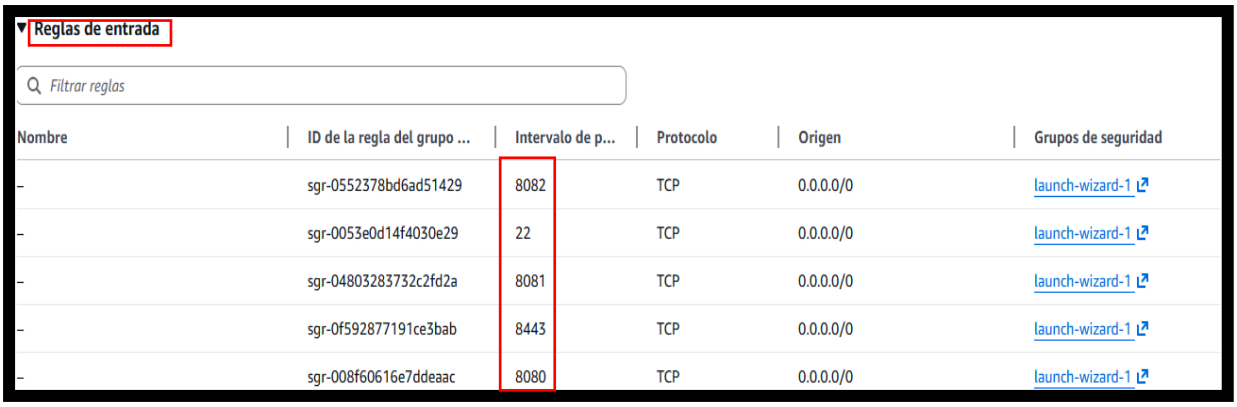
Los Security Groups controlan quién puede hablar con nuestro servidor y por dónde.

Configuraremos las siguientes reglas de entrada como indica en la siguiente tabla:

Puerto	Protocolo	Tipo	Descripción
22	TCP	SSH	Acceso SSH remoto
8080	TCP	HTTP personalizado	Apache HTTP
8081	TCP	HTTP personalizado	Nginx
8082	TCP	HTTP personalizado	Caddy
8443	TCP	HTTPS personalizado	Apache HTTPS (SSL)

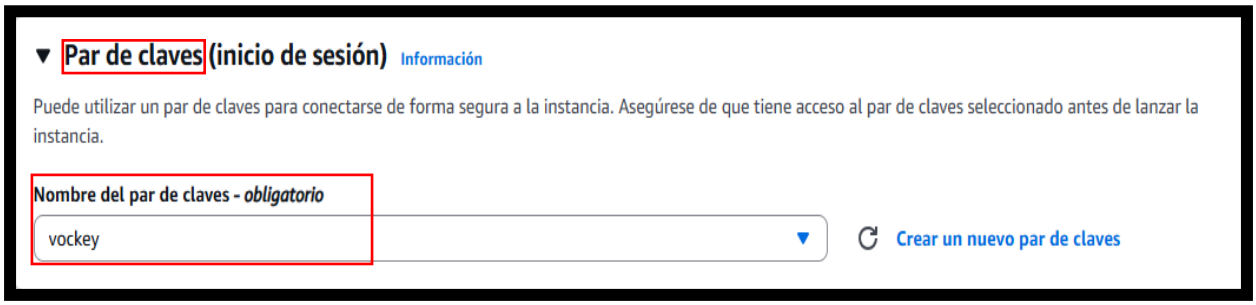
(Tabla de configuración de referencia)

Configuraremos en el servidor la configuración dada en la tabla de arriba.



(Reglas de seguridad configuradas en el Servidor)

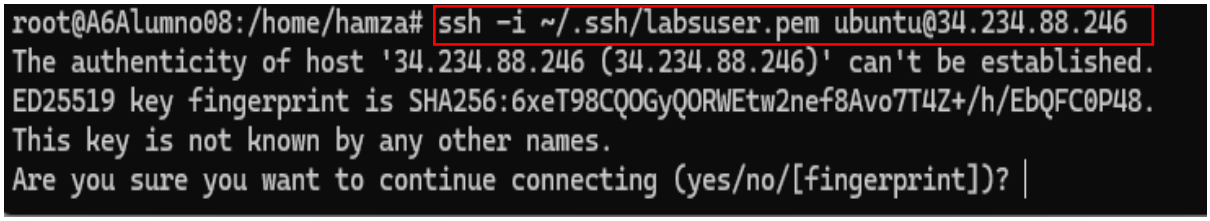
5. Especificar la clave PEM al lanzar la instancia



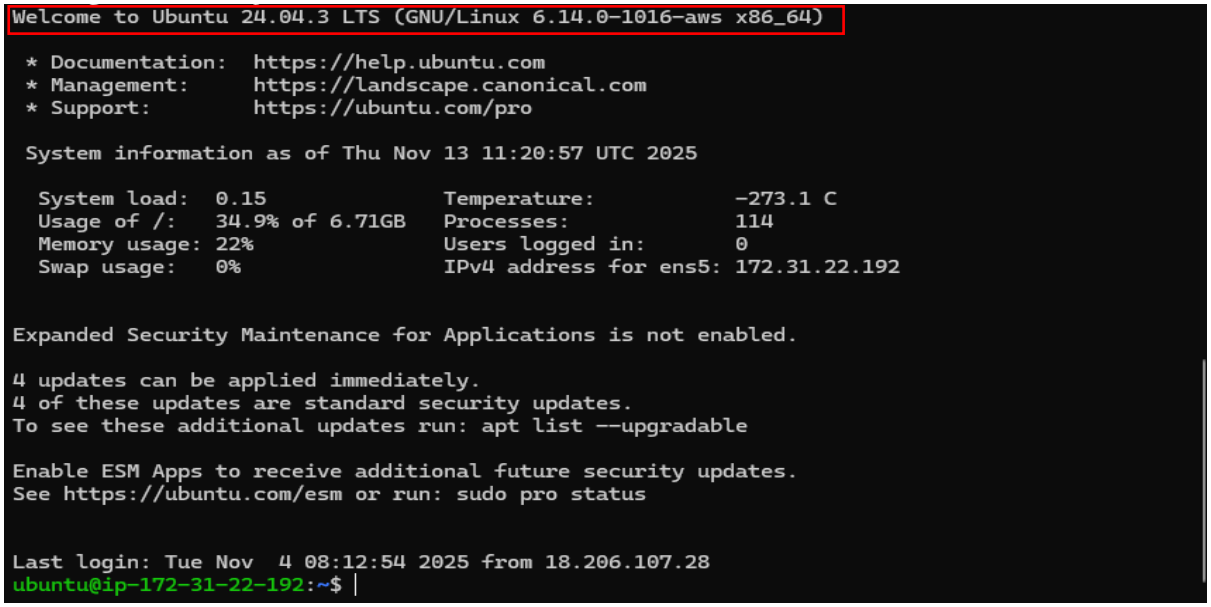
6. Conexión SSH desde WSL a AWS

Ahora nos conectaremos por ssh desde WSL a nuestra instacia en AWS.

En la siguiente imagen podemos ver el comando utilizado para acceder con **ssh,** a través de la clave publica de nuestra instancia en AWS:



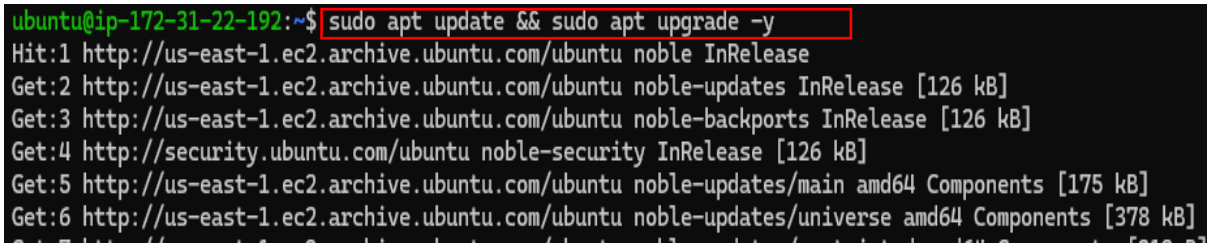
(Acceso a través de ssh y solicitud de fingerprint)



(Acceso a nuestra instancia en AWS)

7. Instalación de Apache2-Gninx-Caddy

Comenzaremos actualizando la lista de paquetes y el sistema.

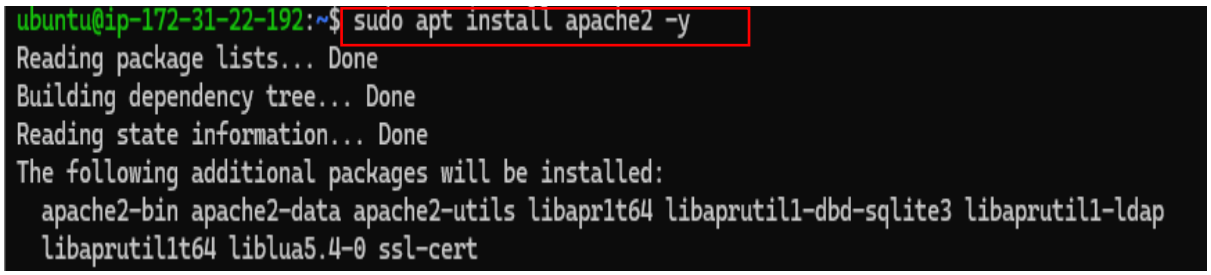


(Ac

Una vez realizadas las actualizaciones correspondientes, podremos comenzar con la instalación de los servicios.

7.1 Instalación Apache2

Instalaremos el servidor web Apache en nuestro sistema.

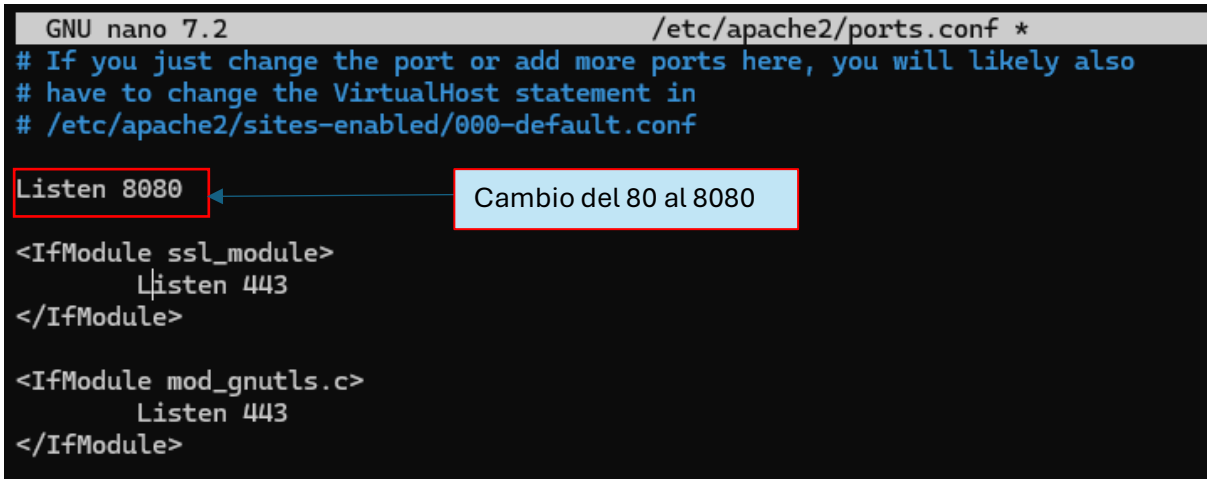


(Instalación del servidor web Apache2)

7.1.1 Configuración fichero de Apache.

Abriremos el archivo de configuración de puertos. Cambia Listen 80 por Listen 8080

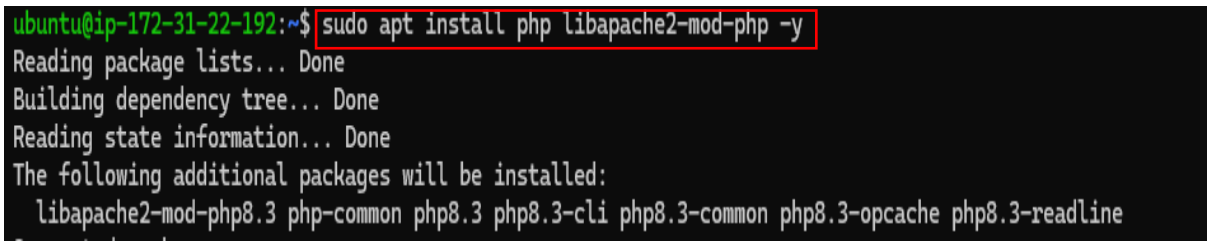
(Comando para acceder al fichero de configuración Apache2)



(Pantallazo del fichero configurado)

7.1.2 Instalación de PHP y reinicio de Apache2

A continuación, haremos la instalación de PHP y su módulo para funcionar con Apache.

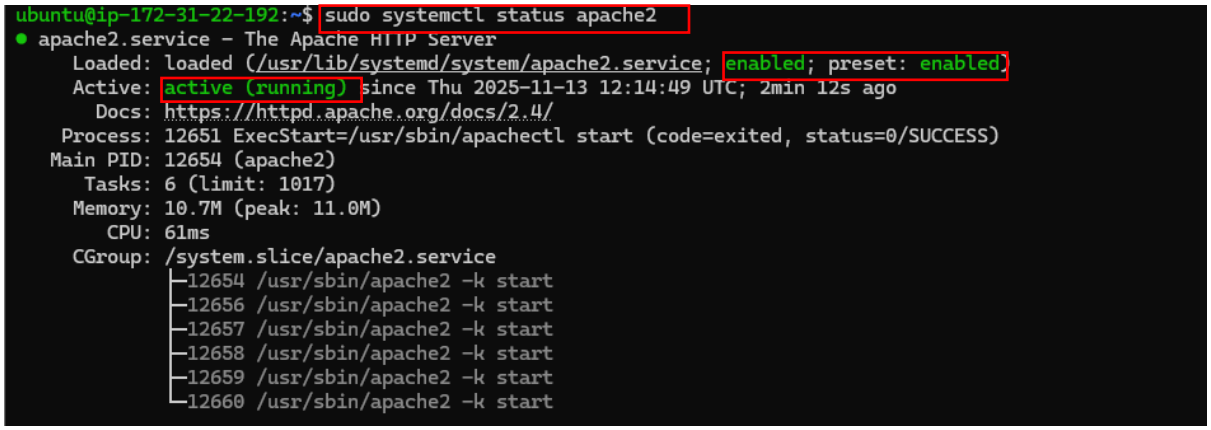


(Comando e instalación de PHP)

Una vez instalado el servicio PHP, reiniciaremos Apache para aplicar los cambios. Para ello, utilizaremos el siguiente comando:

7.1.3 Verificar estado de Apache

Comprobaremos que Apache está funcionando correctamente en el puerto 8080.



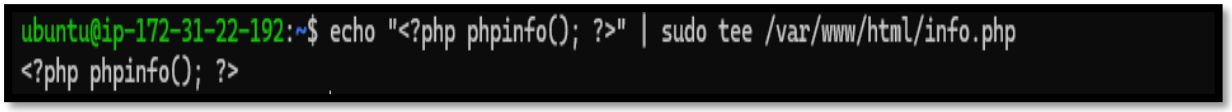
(Estado del servicio de Apache)

Con el siguiente comando, podremos ver y verificar el puerto que está escuchando Apache:

(Verificación puerto de escucha de Apache)

7.1.4 Crear archivo PHP de prueba

Creación de un archivo php de prueba, que muestra información del PHP instalado.



(Creación de archivo PHP)

Para acceder al archivo creado, debemos poner en nuestro buscador local la clave pública de nuestra instancia y el puerto. (8080:23.22.152.252)

(Verificación de archivo de prueba php en buscador local)

También, podemos hacer comprobaciones de Apache desde el terminal y verificar que Apache sirve correctamente el contenido PHP.

```
ubuntu@ip-172-31-22-192:~$ curl http://localhost:8080/info.php
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"><head>
<style type="text/css">
body {background-color: #fff; color: #222; font-family: sans-serif;}
pre {margin: 0; font-family: monospace;}
a:link {color: #009; text-decoration: none; background-color: #fff;}
a:hover {text-decoration: underline;}
table {border-collapse: collapse; border: 0; width: 934px; box-shadow: 1px 2px 3px rgba(0, 0, 0, 0.2);
```

(Comprobación desde terminal)

7.2 Instalación y configuración de Nginx

7.2.1 Instalación Ngnix

Instalaremos Nginx en nuestra instacia.

```
ubuntu@ip-172-31-22-192:~$ sudo apt install nginx -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  nginx-common
Suggested packages:
  fcgiwrap nginx-doc
```

(Instalación de Ngnix)

7.2.2 Configurar Nginx en puerto 8081

Abriremos el fichero de configuración por defecto y la cambiaremos de listen 80 por listen 8081. Para ello iremos a la siguiente ruta:

```
ubuntu@ip-172-31-22-192:~$ sudo nano /etc/nginx/sites-available/default
```

```
  GNU nano 7.2                        /etc/nginx/sites-available/default *
##
# You should look at the following URL's in order to grasp a solid understanding
# of Nginx configuration files in order to fully unleash the power of Nginx.
# https://www.nginx.com/resources/wiki/start/
# https://www.nginx.com/resources/wiki/start/topics/tutorials/config_pitfalls/
# https://wiki.debian.org/Nginx/DirectoryStructure
#
# In most cases, administrators will remove this file from sites-enabled/ and
# leave it as reference inside of sites-available where it will continue to be
# updated by the nginx packaging team.
#
# This file will automatically load configuration files provided by other
# applications, such as Drupal or Wordpress. These applications will be made
# available underneath a path with that package name, such as /drupal8.
#
# Please see /usr/share/doc/nginx-doc/examples/ for more detailed examples.
##

# Default server configuration
#
server {
        listen 8081 default_server;
        listen [::]:8081 default_server;

        # SSL configuration
        #
        # listen 443 ssl default_server;
        # listen [::]:443 ssl default_server;
```

(Configuración de fichero, cambios de puertos de escucha)

7.2.3 Creación de página HTML personalizada.

Con este comando crearemos una página HTML con el siguiente mensaje:

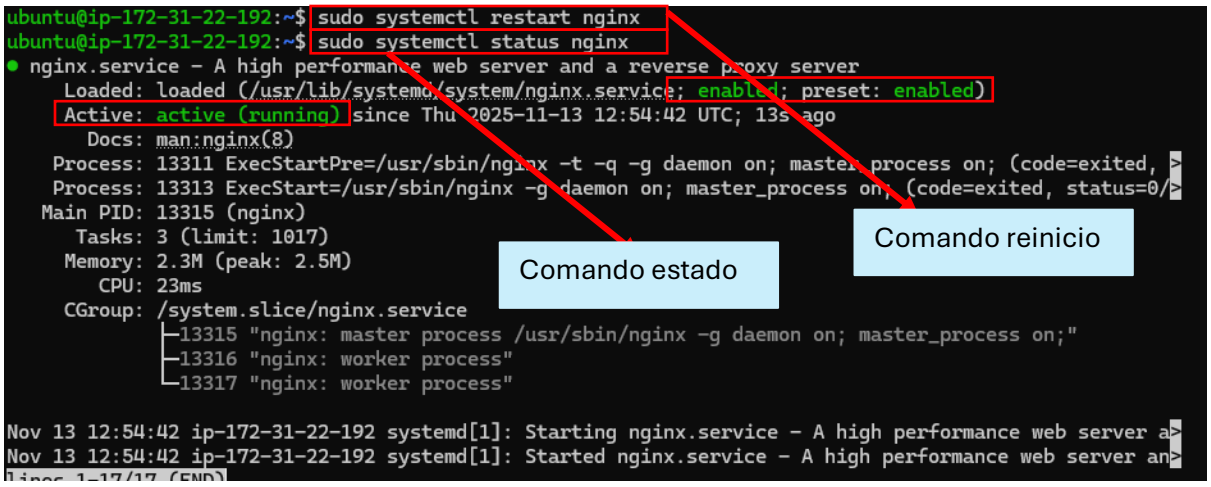
```
ubuntu@ip-172-31-22-192:~$ echo "<h1>Servidor Nginx</h1><p>Funcionando en puerto 8081</p>" | sudo tee
/usr/share/nginx/html/index.html
<h1>Servidor Nginx</h1><p>Funcionando en puerto 8081</p>
```

Resultado

(Creación de html en Nginx)

7.2.4 Reinicio y control de estado de NGINX.

Reiniciaremos Nginx para aplicar los cambios de configuración. Una vez reinciado el servicio, comprobáramos su estado.

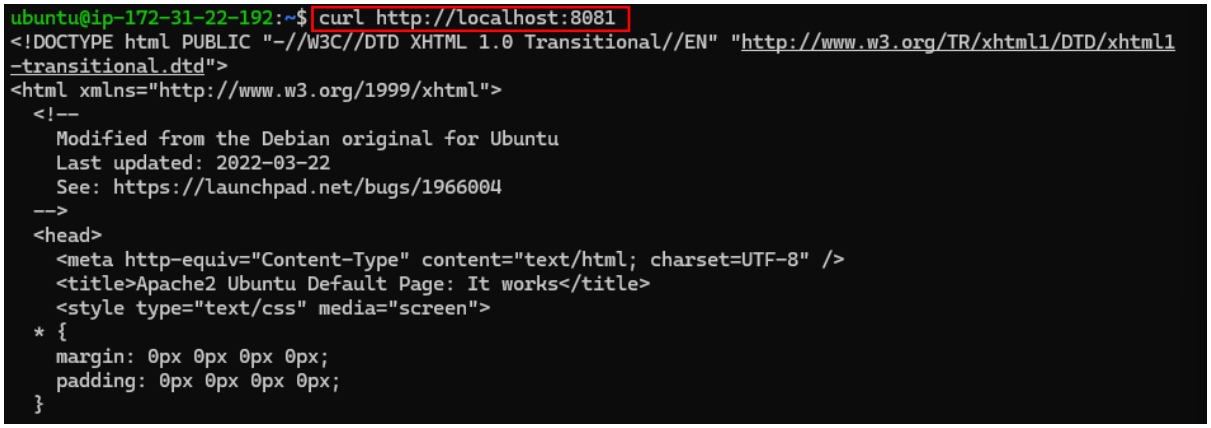


(Reinicio y estado de Ngnix)

Con el siguiente comando, podremos ver y verificar el puerto que está escuchando Nginx:

7.2.5 Prueba Nginx desde la terminal

Verificamos que funciona correctamente el servicio con el siguiente comando en la terminal:



(Prueba funcionamiento de Ngninx en terminal)

7.3 Instalación y configuración de Caddy.

7.3.1 Instalación de dependencias.

Con el siguiente comando instalaremos las herramientas necesarias para añadir los repositorios.

```
ubuntu@ip-172-31-22-192:~/.ssh$ sudo apt install -y debian-keyring debian-archive-keyring apt-transpor
t-https curl
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
curl is already the newest version (8.5.0-2ubuntu10.6).
curl set to manually installed.
The following NEW packages will be installed:
  apt-transport-https debian-archive-keyring debian-keyring
```

(Instalación de Caddy)

7.3.2 Agregar repositorio de Caddy.

Añadiremos el repositorio oficial de Caddy con los siguientes comandos.

```
ubuntu@ip-172-31-22-192:~/.ssh$ curl -1sLf 'https://dl.cloudsmith.io/public/caddy/stable/gpg.key' | su
do gpg --dearmor -o /usr/share/keyrings/caddy-stable-archive-keyring.gpg
```

```
ubuntu@ip-172-31-22-192:~/.ssh$ curl -1sLf 'https://dl.cloudsmith.io/public/caddy/stable/debian.deb.tx
t' | sudo tee /etc/apt/sources.list.d/caddy-stable.list
# Source: Caddy
# Site: https://github.com/caddyserver/caddy
# Repository: Caddy / stable
# Description: Fast, multi-platform web server with automatic HTTPS


deb [signed-by=/usr/share/keyrings/caddy-stable-archive-keyring.gpg] https://dl.cloudsmith.io/public/c
addy/stable/deb/debian any-version main

deb-src [signed-by=/usr/share/keyrings/caddy-stable-archive-keyring.gpg] https://dl.cloudsmith.io/publ
ic/caddy/stable/deb/debian any-version main
```

(Instalación del repositorio oficial de Caddy)

7.3.3 Actualizar e instalar Caddy.

Con el comando a continuación actualizaremos la lista de paquetes e instalaremos Caddy:

```
ubuntu@ip-172-31-22-192:~/.ssh$ sudo apt update && sudo apt install caddy -y
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Get:4 https://dl.cloudsmith.io/public/caddy/stable/deb/debian any-version InRelease [14.8 kB]
Hit:5 http://security.ubuntu.com/ubuntu noble-security InRelease
Get:6 https://dl.cloudsmith.io/public/caddy/stable/deb/debian any-version/main amd64 Packages [4329 B]
Fetched 19.1 kB in 1s (29.2 kB/s)
```

(Actualización e instalación de Caddy)

Después, haremos la creación de un directorio para los archivos de Caddy:

```
ubuntu@ip-172-31-22-192:~/.ssh$ sudo mkdir -p /var/www/caddy
```

7.3.4 Creación de archivo Markdown de prueba (README.md).

Creación de un archivo Markdown de ejemplo con el siguiente contenido:

```
root@ip-172-31-22-192:/home/ubuntu/.ssh# echo "# Bienvenido a Caddy" | sudo tee /var/www/caddy/README.
md
# Bienvenido a Caddy
root@ip-172-31-22-192:/home/ubuntu/.ssh# echo "" | sudo tee -a /var/www/caddy/README.md

root@ip-172-31-22-192:/home/ubuntu/.ssh# echo "Este servidor está funcionando correctamente." | sudo t
ee -a /var/www/caddy/README.md
Este servidor está funcionando correctamente.
root@ip-172-31-22-192:/home/ubuntu/.ssh# echo "" | sudo tee -a /var/www/caddy/README.md

root@ip-172-31-22-192:/home/ubuntu/.ssh# echo "## Características" | sudo tee -a /var/www/caddy/README
.md
## Características
root@ip-172-31-22-192:/home/ubuntu/.ssh# echo "- Servidor moderno" | sudo tee -a /var/www/caddy/README
.md
- Servidor moderno
root@ip-172-31-22-192:/home/ubuntu/.ssh# echo "- HTTPS automático" | sudo tee -a /var/www/caddy/README
.md
- HTTPS automático
root@ip-172-31-22-192:/home/ubuntu/.ssh# echo "- Fácil configuración" | sudo tee -a /var/www/caddy/REA
DME.md
- Fácil configuración
```

(Archivo Markdown creado correctamente)

```
root@ip-172-31-22-192:/home/ubuntu/.ssh# cat /var/www/caddy/README.md
# Bienvenido a Caddy


## Características
- Servidor moderno
- HTTPS automático
- Fácil configuración
```

(Verificación del archivo Markdown creado)

7.3.5 Creación de imagen de prueba

Tendremos en cuenta que si lo hacemos en WSL hay que hacer ajustes previos. Descargaremos una imagen de prueba para verificar que Caddy sirve archivos estáticos.

```
root@ip-172-31-22-192:/home/ubuntu/.ssh# curl -o /tmp/test-image.jpg "https://www.python.org/static/ap
ple-touch-icon-144x144-precomposed.png"
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  7382  100  7382    0     0   162k      0 --:--:-- --:--:-- --:--:--  163k
root@ip-172-31-22-192:/home/ubuntu/.ssh# sudo mv /tmp/test-image.jpg /var/www/caddy/test.jpg
```

(Creación de imagen de prueba)

7.3.6 Creación de Caddyfile personalizado

Con el siguiente comando abriremos el archivo de configuración de Caddy.

```
root@ip-172-31-22-192:/home/ubuntu/.ssh# sudo nano /etc/caddy/Caddyfile
```

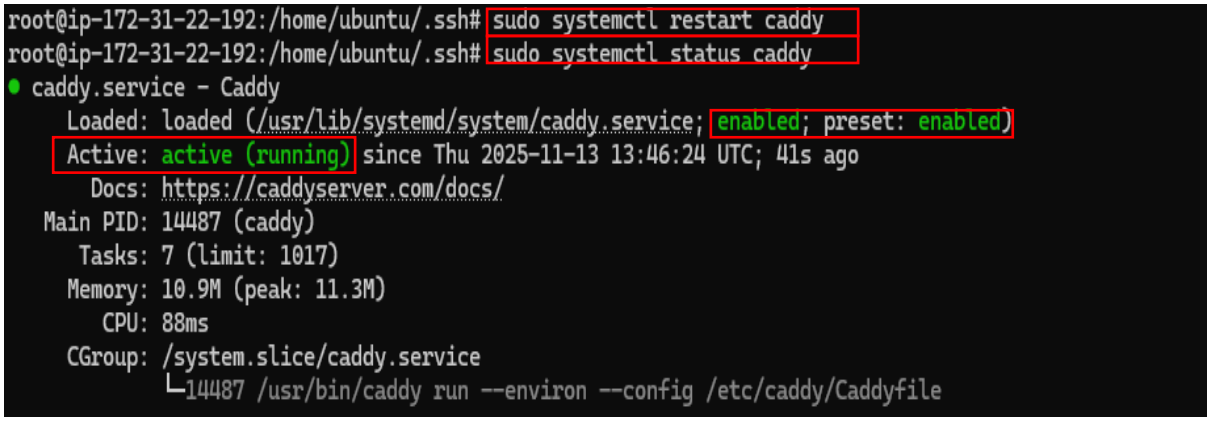
Lo configuraremos de la siguiente manera:

```
  GNU nano 7.2                                  /etc/caddy/Caddyfile *
# The Caddyfile is an easy way to configure your Caddy web server.
#
# Unless the file starts with a global options block, the first
# uncommented line is always the address of your site.
#
# To use your own domain name (with automatic HTTPS), first make
# sure your domain's A/AAAA DNS records are properly pointed to
# this machine's public IP, then replace ":80" below with your
# domain name.

:8082 {
        # Set this path to your site's directory.
        root * /var/www/Caddyfile

        # Enable the static file server.
        file_server browse
        @markdown path *.md
        header @markdown Content-Type text/plain
```

(Configuración de fichero Caddyfile)

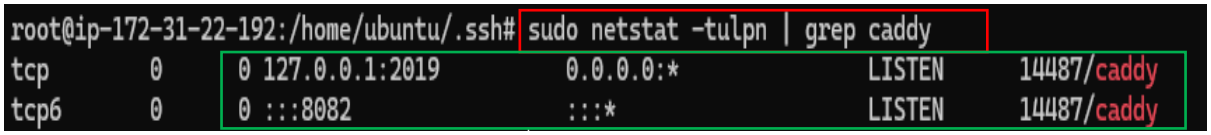
7.3.7 Reinicio de Caddy y verificación de estado.

Reiniciaremos Caddy para aplicar la nueva configuración.



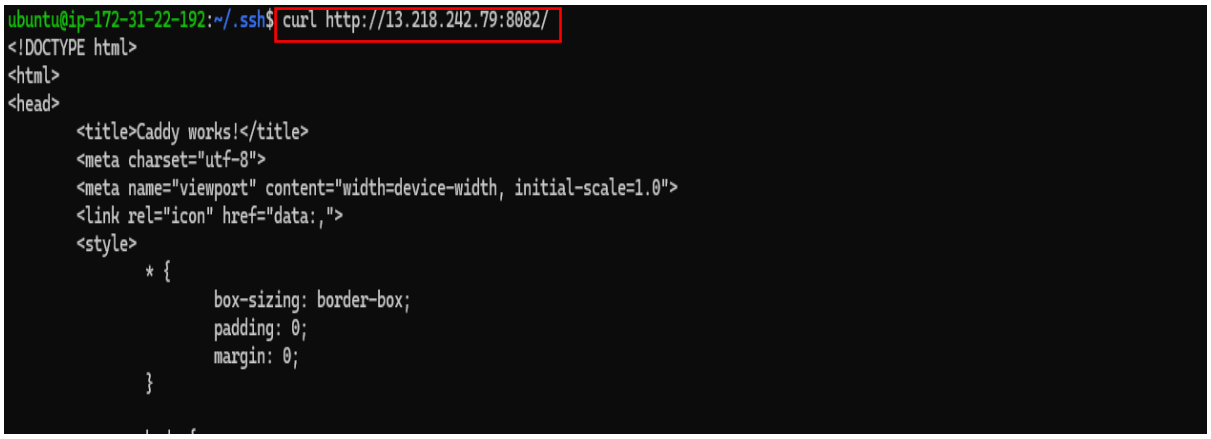
(Reinicio y verificación de estado del servicio)

Podremos ver y verificar el puerto que está escuchando Caddy:



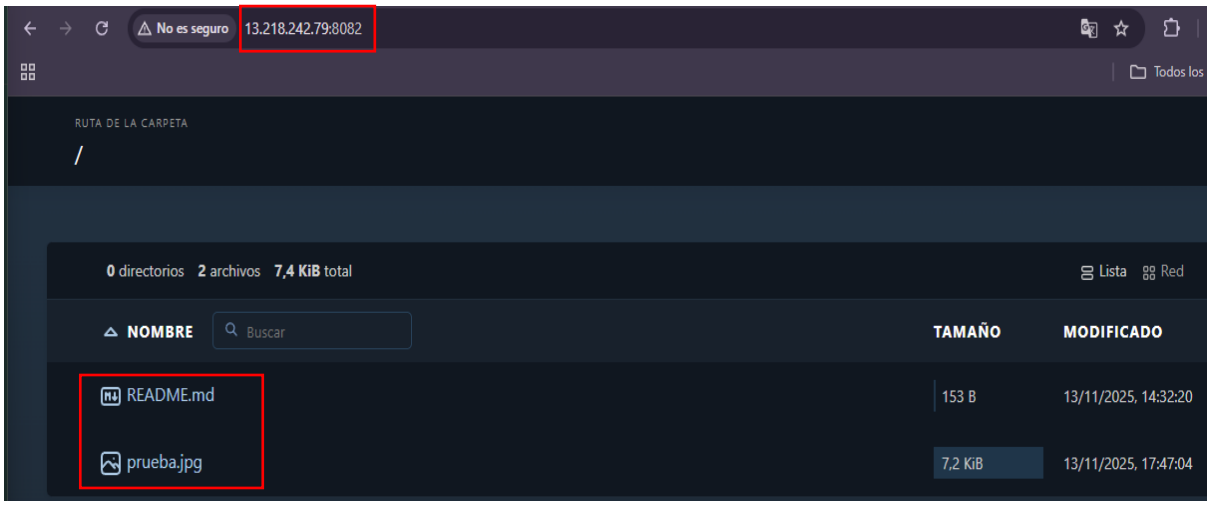
7.3.7 Prueba Caddy desde terminal y navegador web

Probaremos desde la terminal y desde el navegador web como a continuación:



(Pruebas desde el terminal)

Prueba de Caddy desde el navegador web:

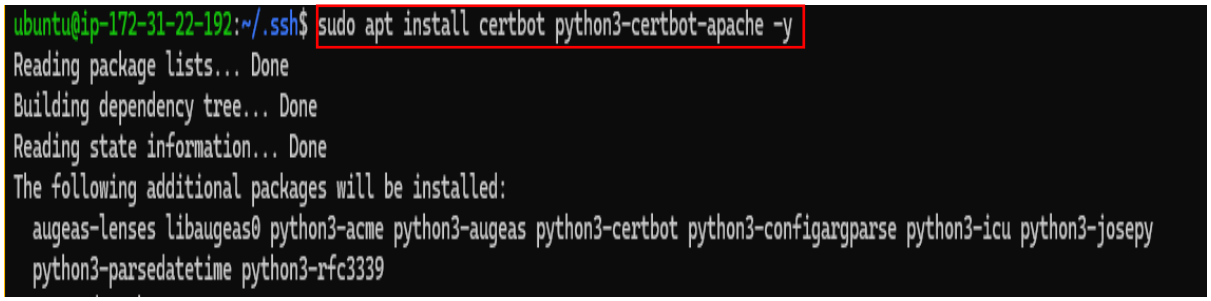


(Prueba Caddy navegador web)

7.4 Configuración de HTTPS con Certbot en Apache

7.4.1 Instalar Certbot y el plugin de Apache.

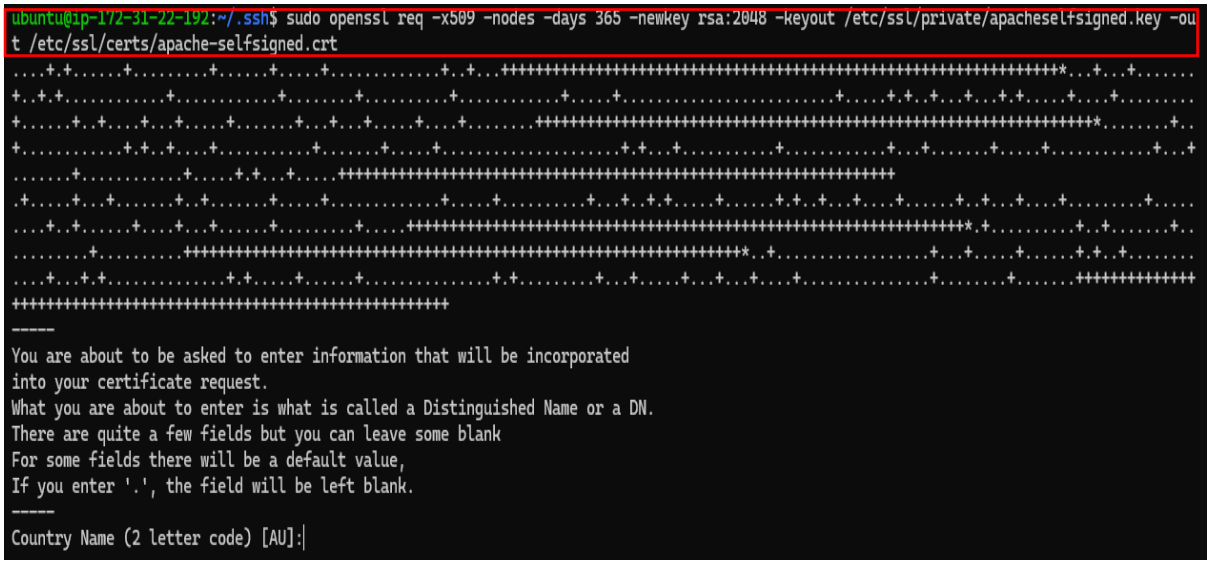
Con el siguiente comando instalaremos Certbot y su integración con Apache para gestionar certificados SSL.



(Instalación de Certbot)

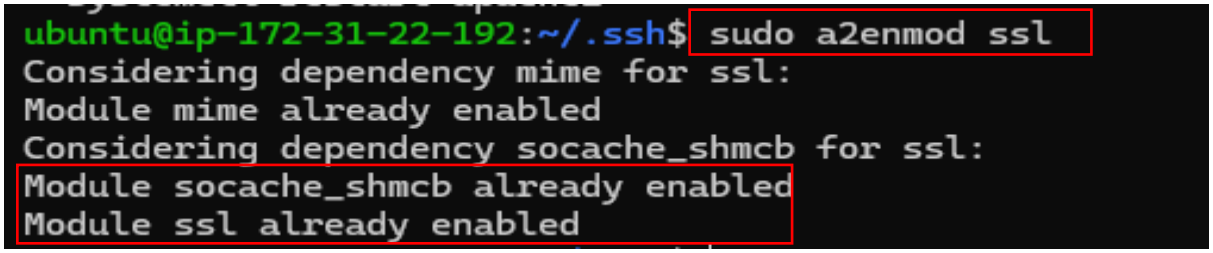
7.4.2 Verificar dominio o usar localhost.

Para obtener certificados reales de Let's Encrypt necesitaremos un dominio público. Para esta práctica usaremos certificados autofirmados. Utilizaremos el comando, el cual creara un certificado autofirmado para practicar HTTPS localmente.



7.4.3 Habilitar módulo SSL en Apache y creación de configuración SSL para Apache.

Activaremos el módulo SSL necesario para HTTPS en Apache.

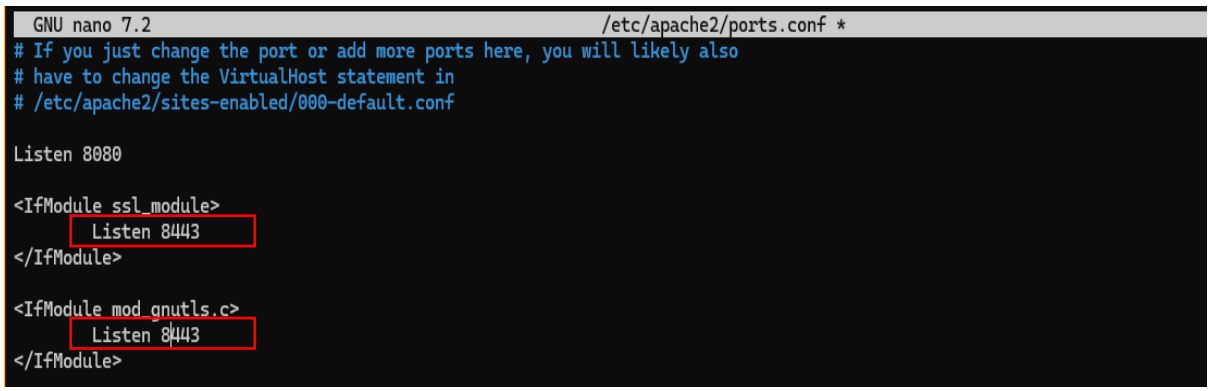


(Activación del módulo SSL)

7.4.4 Cambiar puerto SSL

Iremos a la siguiente ruta para cambiar los puestos SSL

Añadiremos la línea Listen 8443 para que Apache escuche HTTPS en puerto 8443.



(Configuración de puertos de escucha al 8443)

7.4.5 Modificación de VirtualHost SSL

A continuación, entraremos en la ruta con el siguiente comando, donde haremos la configuración SSL. Cambiaremos del primer párrafo en blanco de 433 a 8443.

```
ubuntu@ip-172-31-22-192:~/.ssh$ sudo nano /etc/apache2/sites-available/default-ssl.conf
```

```
 GNU nano 7.2                        /etc/apache2/sites-available/default-ssl.conf *
<VirtualHost *:8443>
        ServerAdmin webmaster@localhost

        DocumentRoot /var/www/html

        # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
        # error, crit, alert, emerg.
        # It is also possible to configure the loglevel for particular
        # modules, e.g.
        #LogLevel info ssl:warn

        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined
```

(Configuración de Virtualhost de 443 a 8443)

7.4.6 Habilitar sitio SSL y verificación HTTPS

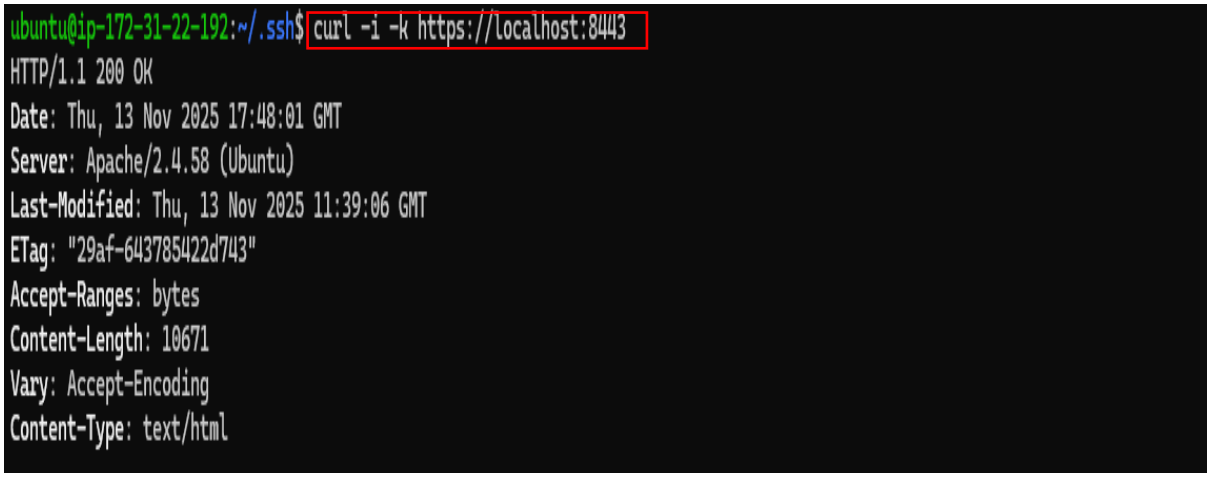
Habilitaremos la configuración SSL en Apache.

```
ubuntu@ip-172-31-22-192:~/.ssh$ sudo a2ensite default-ssl.conf
Site default-ssl already enabled
```

Una vez habilitado, aplicaremos todos los cambios realizados reiniciando el servicio de apache.

```
ubuntu@ip-172-31-22-192:~/.ssh$ sudo systemctl restart apache2
```

Con el siguiente comando verificaremos el funcionamiento en terminal. Prueba la conexión HTTPS (el flag -k ignora el aviso del certificado autofirmado).

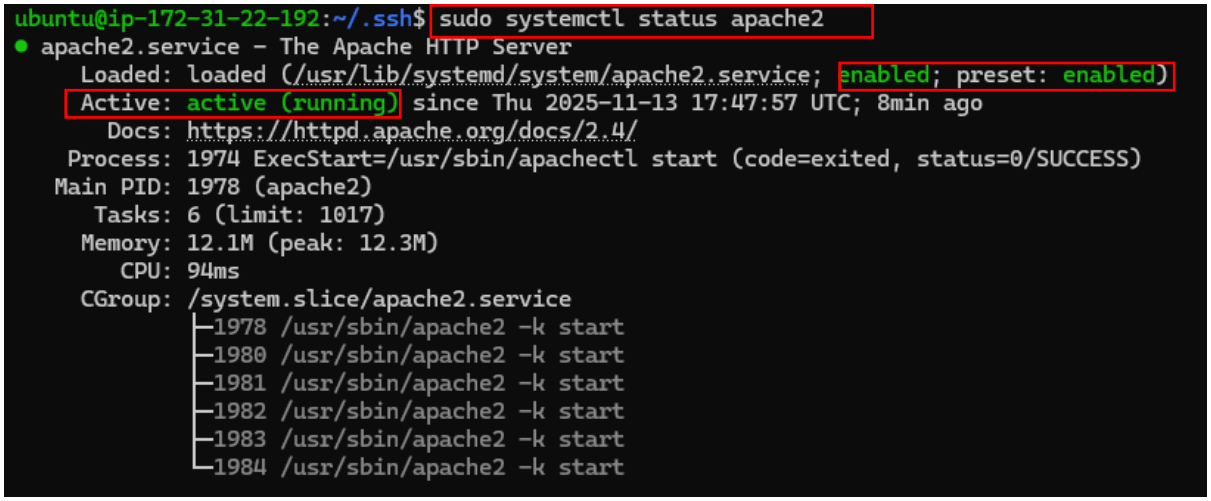


(Prueba de conexión HTTPS desde el terminal)

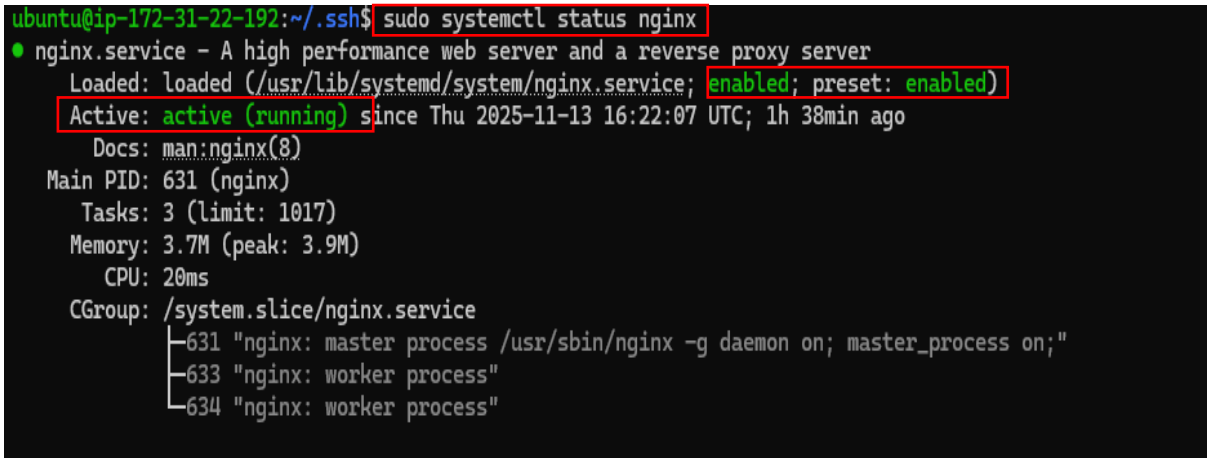
8. Verificación final de los tres servidores

8.1 Verificar que todos los servicios están activos

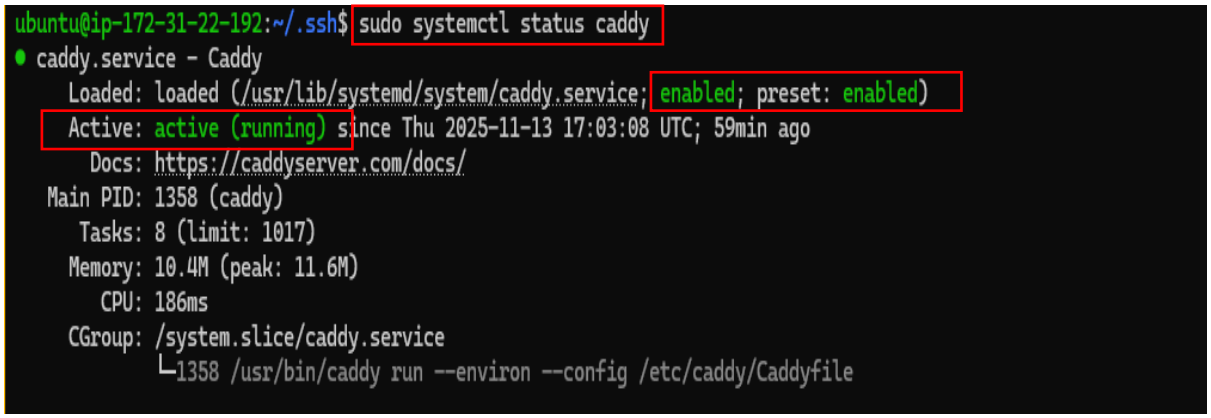
Verificación Apache:



Verificación Nginx:



Verificación Caddy:



8.2 Verificación de puertos en uso

Lista de los puertos donde están escuchando los servicios.