

ARQUITECTURA EN LA NUBE - ADMINISTRACIÓN REMOTA DE SERVIDORES WEB EN AWS A TRAVÉS DE SSH



Héctor Ávila Manjón
2ºASIR

PARTE 0: PREPARACIÓN DEL ENTORNO LOCAL (WSL/VM)

1. Verificar WSL2

wsl --version

```
C:\Users\Alumno.DESKTOP-DI5KTUG>wsl --version
Versión de WSL: 2.6.1.0
Versión de kernel: 6.6.87.2-1
Versión de WSLg: 1.0.66
Versión de MSRDC: 1.2.6353
Versión de Direct3D: 1.611.1-81528511
Versión de DXCore: 10.0.26100.1-240331-1435.ge-release
Versión de Windows: 10.0.26100.5074
```

2. Crear directorio para las claves SSH

mkdir -p ~/.ssh

chmod 700 ~/.ssh

```
hect@A6Alumno14:/mnt/c/Users/Alumno.DESKTOP-DI5KTUG$ mkdir -p ~/.ssh
```

```
hect@A6Alumno14:/mnt/c/Users/Alumno.DESKTOP-DI5KTUG$ chmod 700 ~/.ssh
```

PARTE 1: CONFIGURACIÓN EN AWS (Interfaz Visual)

1. Descargar la clave PEM desde la página del laboratorio

The screenshot shows the AWS Labs interface. At the top, there are navigation links: AWS, Start Lab, End Lab, AWS Details, Readme, Reset, and a close icon. Below these, it shows 'Used \$0 of \$50' and a timer '03:53'. The main area is a terminal window with the prompt 'eee_wl_5246324@runweb199094:~\$'. On the right side, there is a 'Cloud Access' panel with a 'Close' button. It contains an 'AWS CLI' section with a 'Show' button. Below that is a 'Cloud Labs' section with session details: 'Remaining session time: 03:58:08(239 minutes)', 'Session started at: 2025-12-04T01:26:16-0800', and 'Session to end at: 2025-12-04T05:26:16-0800'. It also shows 'Accumulated lab time: 04:01:00 (241 minutes)' and 'No running instance'. At the bottom of the panel, there is an 'SSH key' section with a 'Show' button and a 'Download PEM' button.

Guardamos este archivo en una ubicación segura, preferiblemente en WSL

```
hect@A6Alumno14:/mnt/c/Users/Alumno.DESKTOP-DI5KTUG$ cp /mnt/c/Users/Alumno.DESKTOP-DI5KTUG/D/h/
```

2. Configurar permisos de la clave PEM

chmod 400 ~/.ssh/labsuser.pem

```
hect@A6Alumno14:/mnt/c/Users/Alumno.DESKTOP-DI5KTUG$ chmod 400 ~/.ssh/labsuser.pem
```

Verifica los permisos:

ls -la ~/.ssh/labsuser.pem

```
hect@A6Alumno14:/mnt/c/Users/Alumno.DESKTOP-DI5KTUG$ ls -la ~/.ssh/labsuser.pem
-r----- 1 hect hect 1674 Dec  4 10:48 /home/hect/.ssh/labsuser.pem
```

3. Crear instancia EC2

[EC2](#) > [Instancias](#) > Lanzar una instancia

Lanzar una instancia Información

Amazon EC2 le permite crear máquinas virtuales, o instancias, que se ejecutan en la nube de AWS. Comience rápidamente siguiendo los sencillos pasos que se indican a continuación.

Nombre y etiquetas Información

Nombre

por ejemplo, Mi servidor web

Agregar etiquetas adicionales

▼ Imágenes de aplicaciones y sistemas operativos (Imagen de máquina de Amazon) Información

Una AMI posee el sistema operativo, el servidor de aplicaciones y las aplicaciones de la instancia. Si a continuación no ve una AMI adecuada, utilice el campo de búsqueda o elija **Buscar más AMI**.

Q Busque en nuestro catálogo completo que incluye miles de imágenes de sistemas operativos y aplicaciones

Inicio rápido

Amazon Linux
aws

macOS
Mac

Ubuntu
ubuntu

Windows
Microsoft

Red Hat
Red Hat

SUSE Linux
SUSE

Debian
debian

Q

Buscar más AMI

Inclusión de AMI de AWS, Marketplace y la comunidad

Una vez con la instancia creada le ponemos estas dos etiquetas.

▼ Nombre y etiquetas [Información](#)

Clave [Información](#)
Q Name X

Valor [Información](#)
Q servidor-web-practi X

Tipos de recurso [Información](#)
Seleccionar tipos de r... ▼
Instancias X

Eliminar

Clave [Información](#)
Q Curso X

Valor [Información](#)
Q ArquitecturaNube X

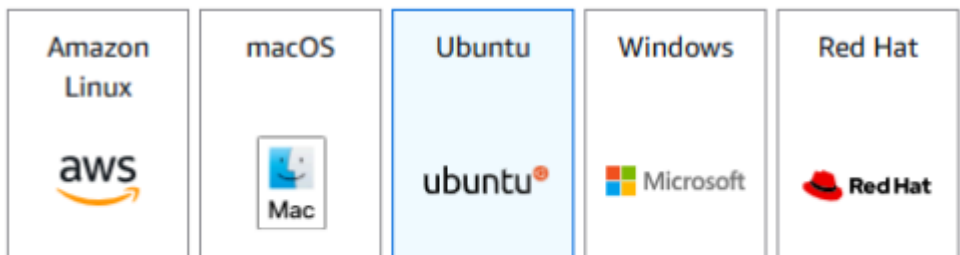
Tipos de recurso [Información](#)
Seleccionar tipos de r... ▼
Instancias X

Eliminar

Agregar nueva etiqueta

Puede agregar hasta 48 etiquetas más.

Luego en cuanto al sistema operativo seleccionaremos Ubuntu Server 24.04 LTS.



Imágenes de máquina de Amazon (AMI)

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type

ami-0ecb62995f68bb549 (64 bits (x86)) / ami-01b9f1e7dc427266e (64 bits (Arm))
Virtualización: hvm Activado para ENA: true Tipo de dispositivo raíz: ebs

Tipo: t2/3.micro (Free Tier)

▼ Tipo de instancia [Información](#) | [Obtener asesoramiento](#)

Tipo de instancia

t3.micro

Apto para la capa gratuita

Familia: t3 2 vCPU 1 GiB Memoria Generación actual: true

Bajo demanda Ubuntu Pro base precios: 0.0139 USD por hora

Bajo demanda SUSE base precios: 0.0104 USD por hora

Bajo demanda Linux base precios: 0.0104 USD por hora

Bajo demanda RHEL base precios: 0.0392 USD por hora

Bajo demanda Windows base precios: 0.0196 USD por hora

Se aplican costos adicionales a las AMI con software preinstalado

En configuración de red lo dejamos como aquí:

▼ Configuraciones de red

Información

VPC : obligatorio

Información

vpc-06480ab144a125c7d

(predeterminado)

▼

172.31.0.0/16

↻

Subred

Información

Sin preferencias

▼

↻

Crear nueva subred

↗

Zona de disponibilidad

Información

Sin preferencias

▼

↻

Habilitar zonas adicionales

↗

Asignar automáticamente la IP pública

Información

Habilitar

▼

Firewall (grupos de seguridad)

Información

Un grupo de seguridad es un conjunto de reglas de firewall que controlan el tráfico de la instancia. Agregue reglas para permitir que un tráfico específico llegue a la instancia.

☒ Crear grupo de seguridad

☐ Seleccionar un grupo de seguridad existente

sg-servidores-web

Este grupo de seguridad se agregará a todas las interfaces de red. El nombre no se puede editar después de crear el grupo de seguridad. La longitud máxima es de 255 caracteres. Caracteres válidos: a-z, A-Z, 0-9, espacios y _-./()!@,[]+~&,:{}!\$*

Descripción - obligatorio

Información

Reglas practica

4. Configurar Security Group (Reglas de entrada)

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)

Pondremos 8 GiB y gp3

▼ Configurar almacenamiento [Información](#)

[Ayar](#)

1x GiB ▼ Volumen raíz, 3000 IOPS,

No cifrado

Y lanzamos la instancia

▼ Resumen

Número de instancias | [Información](#)

Imagen de software (AMI)

Canonical, Ubuntu, 24.04, amd6...[más información](#)
ami-0ecb62995f68bb549

Tipo de servidor virtual (tipo de instancia)

t3.micro

Firewall (grupo de seguridad)

Nuevo grupo de seguridad

Almacenamiento (volúmenes)

Volúmenes: 1 (8 GiB)

[Cancelar](#)

[Lanzar instancia](#)

PARTE 2: CONEXIÓN SSH DESDE WSL A AWS

1. Obtener la IP pública de la instancia

En AWS Console: EC2 → Instances → Public IPv4 address

2. Conectar por SSH

ssh -i ~/.ssh/labsuser.pem ubuntu@34.228.52.111

```
hect@A6Alumno14:/mnt/c/Users/Alumno.DESKTOP-DISKU6$ ssh -i ~/.ssh/labsuser.pem ubuntu@34.228.52.111
The authenticity of host '34.228.52.111 (34.228.52.111)' can't be established.
ED25519 key fingerprint is SHA256:MP1PyC40w+/EiHR8S/Q/0CpUgMb78FVDkF+kaykcSXw.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '34.228.52.111' (ED25519) to the list of known hosts.
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-1015-aws x86_64)
```

```
ubuntu@ip-172-31-66-175:~$
```

PARTE 3: EJECUTAR LA SEGUNDA PRÁCTICA ORIGINAL

SERVIDOR APACHE CON PHP

sudo apt update && sudo apt upgrade -y

```
ubuntu@ip-172-31-66-175:~$ sudo apt update && sudo apt upgrade -y
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
```

sudo apt install apache2 -y

```
ubuntu@ip-172-31-66-175:~$ sudo apt install apache2 -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
```

Configuraremos Apache en el puerto 8080, sudo nano /etc/apache2/ports.conf En la configuración cambiaremos listen 80 por listen 8080.

```
GNU nano 7.2 /etc/apache2/ports.conf
# If you just change the port or add more ports here, you will li
# have to change the VirtualHost statement in
# /etc/apache2/sites-enabled/000-default.conf

Listen 8080

<IfModule ssl_module>
    Listen 443
</IfModule>

<IfModule mod_gnutls.c>
    Listen 443
</IfModule>
```

sudo nano /etc/apache2/sites-available/000-default.conf Una vez dentro cambiamos VirtualHost *:80 por VirtualHost *:8080.

```
GNU nano 7.2 /etc/apache2/sites-available/000-default.conf *
<VirtualHost *:8080>
```

sudo apt install php libapache2-mod-php -y

```
ubuntu@ip-172-31-66-175:~$ sudo apt install php libapache2-mod-php -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
```

sudo systemctl restart apache2

```
ubuntu@ip-172-31-66-175:~$ sudo systemctl restart apache2
```

sudo systemctl status apache2

sudo netstat -tulpn | grep 8080

```
ubuntu@ip-172-31-66-175:~$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: enabled)
   Active: active (running) since Thu 2025-12-04 13:12:32 UTC; 24s ago
     Docs: https://httpd.apache.org/docs/2.4/

ubuntu@ip-172-31-66-175:~$ sudo netstat -tulpn | grep 8080
tcp6      0      0 :::8080          :::*
```

echo "" | sudo tee /var/www/html/info.php

```
ubuntu@ip-172-31-66-175:~$ echo "<?php phpinfo(); ?>" | sudo tee /var/www/html/info.php
<?php phpinfo(); ?>
```



```
ubuntu@ip-172-31-86-175:~$ 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);
}.center {text-align: center;}

```

[illegible]

```
ubuntu@ip-172-31-66-175:~$ 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 confi
s.
To activate the new configuration, you need to run:
    systemctl restart apache2
```

```
GNU nano 2.2 /etc/apache2/sites-available/default-ssl.conf
VirtualHost *:80 {
    ServerAdmin webmaster@localhost

    SSLCertificateFile      /etc/ssl/certs/apache-selfsigned.crt
    SSLCertificateKeyFile   /etc/ssl/private/apache-selfsigned.key
```

`sudo nano /etc/apache2/ports.conf`

pondremos debajo de listen 8080, listen 8443.

```
GNU nano 7.2 /etc/apache2/ports.conf
# If you just change the port or add more ports here, you will li
# have to change the VirtualHost statement in
# /etc/apache2/sites-enabled/000-default.conf

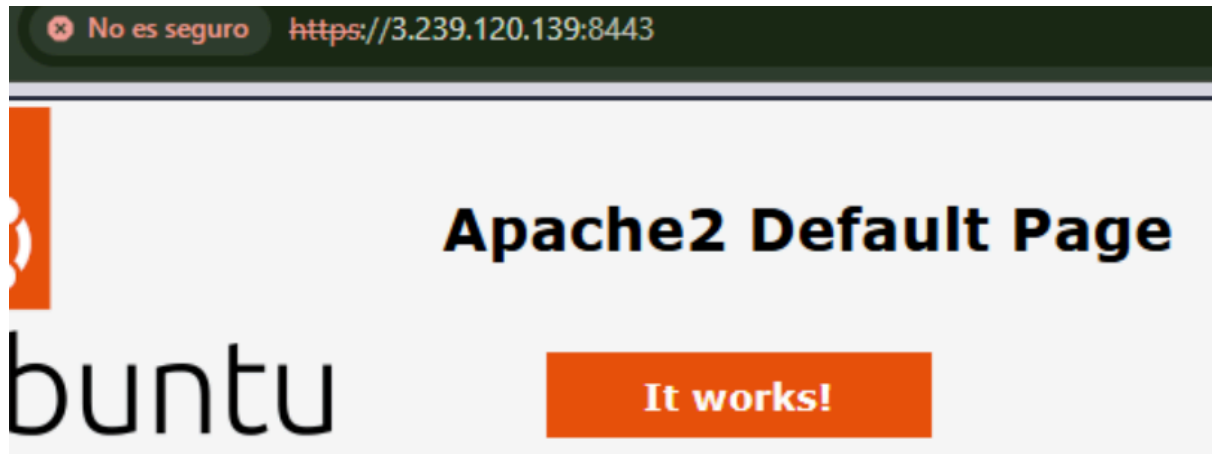
Listen 8080
Listen 8443
```

`sudo a2ensite default-ssl.conf`

```
ubuntu@ip-172-31-66-175:~$ sudo a2ensite default-ssl.conf
Enabling site default-ssl.
To activate the new configuration, you need to run:
  systemctl reload apache2
```

```
ubuntu@ip-172-31-66-175:~$ sudo systemctl restart apache2
```

Para terminar esta parte, accederemos a “https://3.239.120.139:8443”, y veremos la pantalla con el candado



SERVIDOR NGINX CON HTML

`sudo apt install nginx -y`

```
ubuntu@ip-172-31-66-175:~$ 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
```

`sudo nano /etc/nginx/sites-available/default`

Cambiaremos listen 80 por listen 8081

```
ubuntu@ip-172-31-66-175:~$ sudo nano /etc/nginx/sites-available/default

server {
    listen 8081 default_server;
    listen [::]:8081 default_server;
```

`echo "<h1>Servidor`

`Nginx</h1><p> Funcionando en puerto 8081</p>" | sudo tee`

`/var/www/html/index.html".`

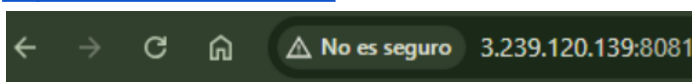
```
ubuntu@ip-172-31-66-175:~$ echo "<h1>Servidor Nginx</h1><p> Funcionando en puerto 8081</p>" | sudo tee
/usr/share/nginx/html/index.html
<h1>Servidor Nginx</h1><p> Funcionando en puerto 8081</p>
```

`sudo systemctl restart nginx`

`sudo systemctl status nginx`

```
ubuntu@ip-172-31-66-175:~$ sudo systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
   Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; preset: enabled)
   Active: active (running) since Thu 2025-12-04 13:36:41 UTC; 15s ago
     Docs: man:nginx(8)
```

<http://3.239.120.139:8081>



Servidor Nginx

Funcionando en puerto 8081

SERVIDOR CADDY CON ARCHIVOS ESPECIALES

sudo apt install -y debian-keyring debian-archive-keyring apt-transport-https curl

```
ubuntu@ip-172-31-66-175:~$ sudo apt install -y debian-keyring debian-archive-keyring apt-transport-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
0 upgraded, 3 newly installed, 0 to remove and 5 not upgraded.
```

curl -1sLf 'https://dl.cloudsmith.io/public/caddy/stable/gpg.key' | sudo gpg --dearmor -o /usr/share/keyrings/caddy-stable-archive-keyring.gpg

```
ubuntu@ip-172-31-66-175:~$ curl -1sLf 'https://dl.cloudsmith.io/public/caddy/stable/gpg.key' | sudo gpg --dearmor -o /usr/share/keyrings/caddy-stable-archive-keyring.gpg
```

curl -1sLf 'https://dl.cloudsmith.io/public/caddy/stable/debian.deb.txt' | sudo tee /etc/apt/sources.list.d/caddy-stable.list

```
ubuntu@ip-172-31-66-175:~$ curl -1sLf 'https://dl.cloudsmith.io/public/caddy/stable/debian.deb.txt' | 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
```

sudo apt update && sudo apt install caddy -y

```
ubuntu@ip-172-31-66-175:~$ sudo apt update && sudo apt install caddy -y
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:5 https://dl.cloudsmith.io/public/caddy/stable/deb/debian any-version InRelease [14.8 kB]
```

sudo mkdir -p /var/www/caddy

```
ubuntu@ip-172-31-66-175:~$ sudo mkdir -p /var/www/caddy
```

echo "# Bienvenido a Caddy" | sudo tee /var/www/caddy/README.md

```
ubuntu@ip-172-31-66-175:~$ echo "# Bienvenido a Caddy" | sudo tee /var/www/caddy/README.md
# Bienvenido a Caddy
```

echo "" | sudo tee -a /var/www/caddy/README.md

```
ubuntu@ip-172-31-66-175:~$ echo "" | sudo tee -a /var/www/caddy/README.md
```

`echo "Este servidor está funcionando correctamente." | sudo tee -a /var/www/caddy/README.md`

```
ubuntu@ip-172-31-66-175:~$ echo "Este servidor está funcionando correctamente." | sudo tee -a /var/www/caddy/README.md
Este servidor está funcionando correctamente.
```

`echo "" | sudo tee -a /var/www/caddy/README.md`

```
ubuntu@ip-172-31-66-175:~$ echo "" | sudo tee -a /var/www/caddy/README.md
```

`echo "## Características" | sudo tee -a /var/www/caddy/README.md`

```
ubuntu@ip-172-31-66-175:~$ echo "## Características" | sudo tee -a /var/www/caddy/README.md
## Características
```

`curl -o /tmp/test-image.jpg`

`"https://www.python.org/static/apple-touch-icon-144x144-precomposed.png" y "sudo mv /tmp/test-image.jpg /var/www/caddy/test.jpg"`

```
ubuntu@ip-172-31-66-175:~$ curl -o /tmp/test-image.jpg "https://www.python.org/static/apple-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  178k      0  --:--:-- --:--:-- --:--:--  180k
ubuntu@ip-172-31-66-175:~$ sudo mv /tmp/test-image.jpg /var/www/caddy/test.jpg
```

`sudo nano /etc/caddy/Caddyfile`

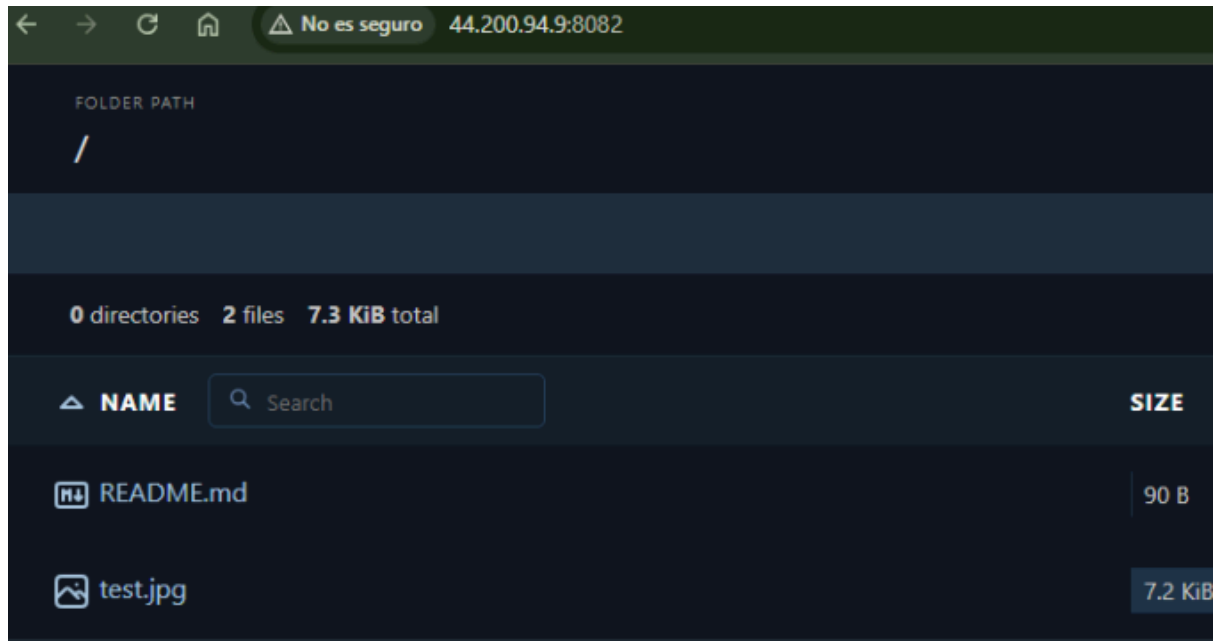
escribiremos lo siguiente `:8082 { root * /var/www/caddy file_server browse @markdown path *.md header @markdown Content-Type text/plain }`

```
GNU nano 7.2 /etc/caddy/Caddyfile
:8082 {
    root * /var/www/caddy
    file_server browse
    @markdown path *.md
    header @markdown Content-Type text/plain
}
```

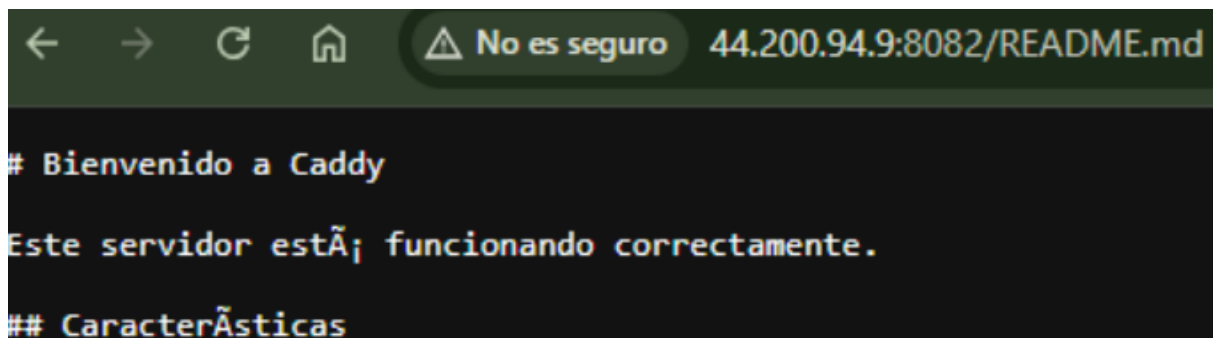
`sudo systemctl restart caddy`

`sudo systemctl status caddy`

<http://44.200.94.9:8082>



<http://44.200.94.9:8082/README.md>



<http://44.200.94.9:8082/test.jpg>



VERIFICACIÓN SIMULTÁNEA DE LOS TRES SERVIDORES

`sudo systemctl status apache2 nginx caddy`

```
ubuntu@ip-172-31-66-175:~$ sudo syste
● apache2.service - The Apache HTTP S
   Loaded: loaded (/usr/lib/systemd/sy
   Active: active (running) since Fri
   Docs: https://httpd.apache.org
   Main PID: 559 (apache2)

● nginx.service - A high performance we
   Loaded: loaded (/usr/lib/systemd/sy
   Active: active (running) since Fri
   Docs: https://nginx.org/
   Main PID: 623 (nginx)

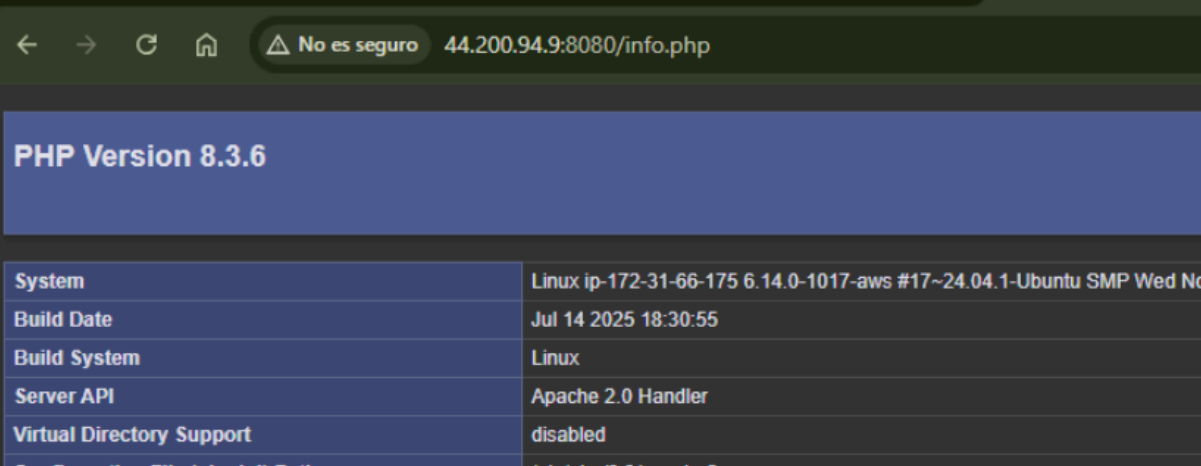
● caddy.service - Caddy
   Loaded: loaded (/usr/lib/systemd/syste
   Active: active (running) since Fri 202
   Docs: https://caddyserver.com/docs/
   Main PID: 2642 (caddy)
```

`sudo netstat -tulpn | grep -E '8080|8081|8082|8443'`

```
ubuntu@ip-172-31-66-175:~$ sudo netstat -tulpn | grep -E '8080|8081|8082|8443'
tcp        0      0 0.0.0.0:8081          0.0.0.0:*            LISTEN     623/nginx: master p
tcp6       0      0 :::8443              :::*                  LISTEN     559/apache2
tcp6       0      0 :::8081              :::*                  LISTEN     623/nginx: master p
tcp6       0      0 :::8080              :::*                  LISTEN     559/apache2
tcp6       0      0 :::8082              :::*                  LISTEN     2642/caddy
```

Abriremos las cuatro pestañas del navegador simultáneamente, Apache HTTP, Apache HTTPS, Nginx y Caddy.

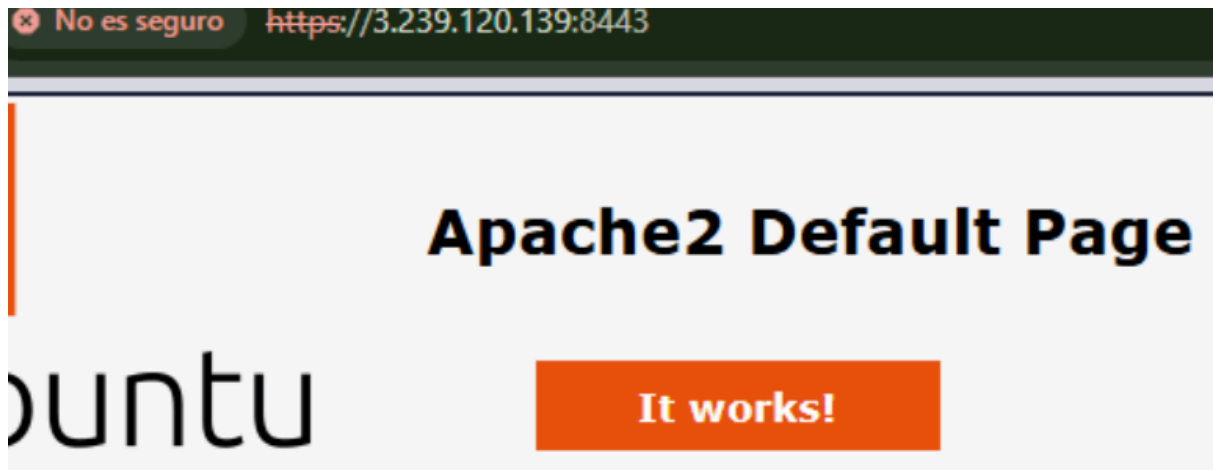
Apache HTTP



The screenshot shows a web browser window with the address bar displaying "No es seguro" (Not secure) and the URL "44.200.94.9:8080/info.php". The page content includes the title "PHP Version 8.3.6" and a table with system information.

System	Linux ip-172-31-66-175 6.14.0-1017-aws #17~24.04.1-Ubuntu SMP Wed No
Build Date	Jul 14 2025 18:30:55
Build System	Linux
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (path)	/etc/apache2/2/apache2

Apache HTTPS



Nginx



Caddy

