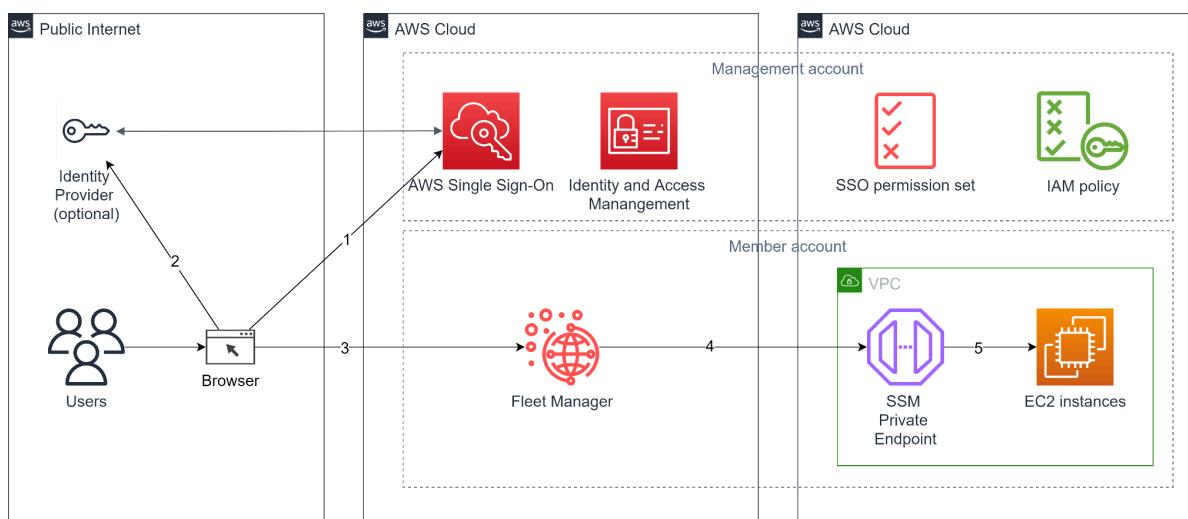
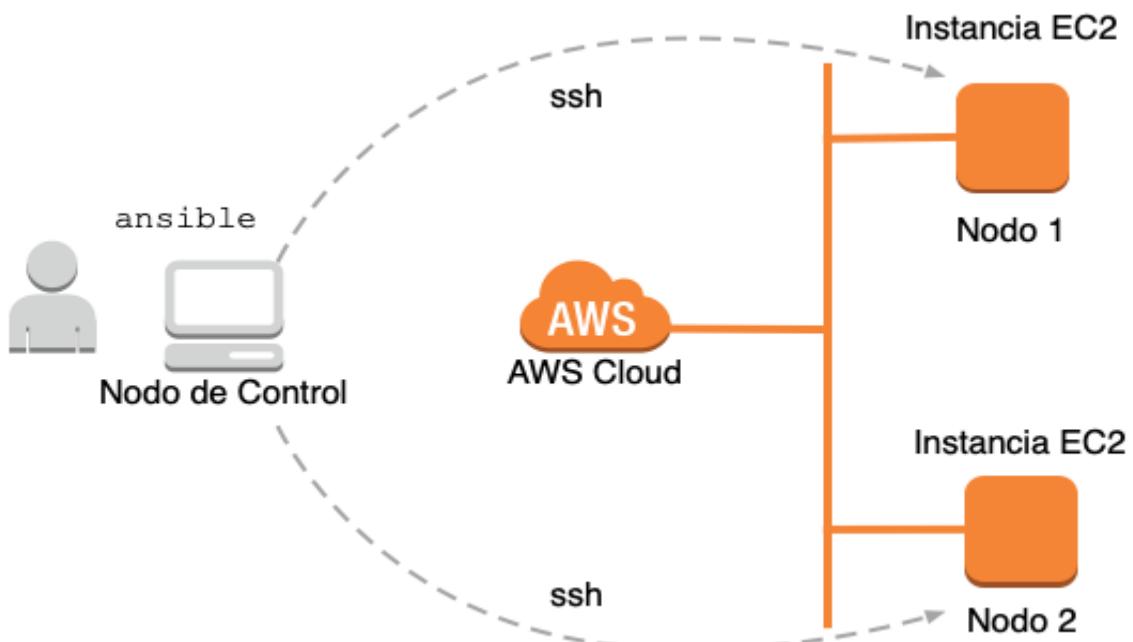


# ADMINISTRACIÓN REMOTA DE SERVIDORES WEB EN AWS A TRAVÉS DE SSH



MARIO CAMACHO PRIETO  
2º ASIR / CAJA MÁGICA  
20 / 11 / 2025

# ÍNDICE

## PARTE 1: PREPARACIÓN DEL ENTORNO LOCAL

- Verificación de versión, creación de carpetas y acceso al laboratorio.

## PARTE 2: GESTIÓN DE LA CLAVE DE SEGURIDAD

- Copia de la clave .pem y asignación de permisos 400.

## PARTE 3: CONFIGURACIÓN E INSTALACIÓN EN AWS

- Proceso completo de creación de la instancia EC2 y configuración del Security Group (Firewall).

## PARTE 4: CONEXIÓN SSH Y DESPLIEGUE DE SERVICIOS

- Conexión al servidor.
- Instalación y configuración de Apache (Puerto 8080) .
- Instalación y configuración de Nginx (Puerto 8081).
- Instalación y configuración de Caddy (Puerto 8082).
- Configuración de HTTPS/SSL (Puerto 8443).
- Comprobaciones finales de todos los servicios simultáneos.

## PARTE 5: COMPROBACIONES VISUALES

- Prueba de acceso externo desde Windows.

## PARTE 6: DOCUMENTACIÓN TEÓRICA

- Explicación de claves PEM y reglas del Security Group.

# PARTE 1: PREPARACIÓN DEL ENTORNO LOCAL

Inicio la práctica verificando que mi entorno de trabajo es correcto. Ejecuto wsl --version para confirmar que tengo el subsistema de Linux instalado y listo en mi Windows, lo que me permitirá usar las herramientas necesarias.

```
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

C:\Users\Alumno.DESKTOP-DI5KTUG>wsl
alumno16@A6Alumno16:/mnt/c/Users/Alumno.DESKTOP-DI5KTUG$ mkdir -p ~/.ssh
alumno16@A6Alumno16:/mnt/c/Users/Alumno.DESKTOP-DI5KTUG$ chmod 700 ~/.ssh
alumno16@A6Alumno16:/mnt/c/Users/Alumno.DESKTOP-DI5KTUG$ |
```

Preparo la seguridad local creando el directorio oculto .ssh con mkdir. Inmediatamente le asigno permisos 700 (chmod 700) para asegurar que sea una 'caja fuerte' privada donde solo mi usuario pueda entrar.

The screenshot shows a web browser window with the following details:

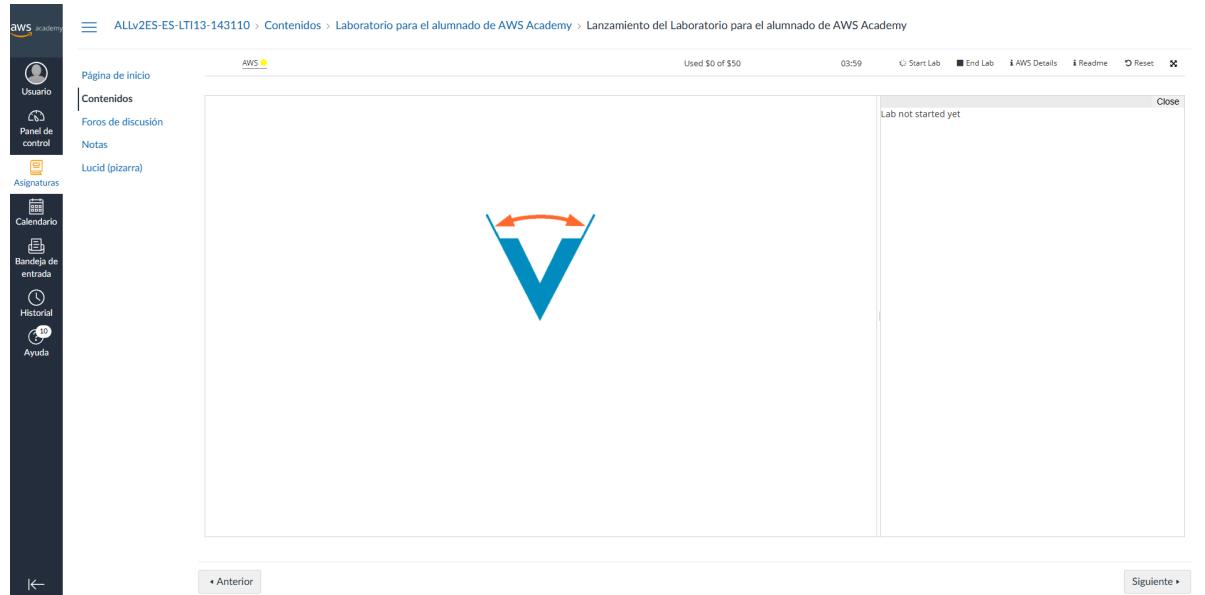
- Header:** The URL is [ALLv2ES-ES-LTI13-143110 > Contenidos > Laboratorio para el alumnado de AWS Academy > Lanzamiento del Laboratorio para el alumnado de AWS Academy](#). The title bar says "Vocareum".
- Left Sidebar (User Navigation):** Includes links for "Panel de control", "Asignaturas", "Calendario", "Bandeja de entrada", "Historial", and "Ayuda".
- Content Area:**
  - A message: "Please read the terms and conditions shown below and click on the "I agree" button at the bottom of this page to continue."
  - Terms Of Use:** Revised: 08.08.2025. It states: "Welcome to the Vocareum, Inc. ("Vocareum") website located at [www.vocareum.com](#) (the "Site"). Please read these Terms of Use (the "Terms") along with our [Acceptable Use Policy](#) and [Privacy Policy](#) carefully because they govern your use of our Site and our web-based education and learning platform, as well as our treatment of information we receive from you. The Site and our platform are collectively called the "Services"."
  - Section 1. Agreement To Terms:** "By using the functionality of our Services, teachers can: (i) create, customize and administer educational courses and (ii) invite students to participate in a class taught and supervised by the teacher using the online tools provided by Vocareum. Subject to your compliance with these Terms, Vocareum will make the Services available to you solely for the purpose of your internal, non-commercial use."
  - Section 2. Changes to Terms or Services:** "We may modify the Terms at any time, in our sole discretion. If we do so, we'll let you know either by posting the modified Terms on the Site or through other communications. The effective date of the modified Terms will be indicated in the modified Terms and any such other communications. It's important that you review the Terms whenever we modify them because if you continue to use the Services after we have posted modified Terms on the Site or otherwise communicated with you about the modified Terms, you are indicating to us that you agree to be bound by the modified Terms."
- Bottom Navigation:** Buttons for "Anterior" and "Siguiente".

Accedo a la plataforma del laboratorio de AWS Academy. Aquí localizo las credenciales y los enlaces necesarios para descargar mi clave privada (labsuser.pem) y el botón para entrar a la consola de gestión de AWS.

The screenshot shows the AWS Academy platform interface. On the left is a dark sidebar with icons for User, Control Panel, Assignments, Calendar, Inbox, History, and Help. The main area has a breadcrumb navigation path: ALLv2ES-ES-LTI13-143110 > Contenidos > Laboratorio para el alumnado de AWS Academy > Lanzamiento del Laboratorio para el alumnado de AWS Academy. A red circle with a dot is positioned above the 'Contenidos' link. The central content area is titled 'AWS' with a red dot. Below it is a terminal window showing the command 'eee\_u\_543035@runweb0198551:~\$'. To the right of the terminal is a 'Learner Lab' section with a title 'Learner Lab' and a 'EN-US' dropdown menu. This section contains links for 'Environment Overview', 'Environment Navigation', 'Access the AWS Management Console', 'Region restriction', 'Service usage and other restrictions', 'Using the terminal in the browser', 'Running AWS CLI commands', 'Using the AWS SDK for Python', 'Preserving your budget', 'Accessing EC2 Instances', 'SSH Access to EC2 Instances', 'SSH Access from Windows', and 'SSH Access from a Mac'. It also includes a note: 'Instructions last updated: 2025-06-24'. Below this is a 'Environment Overview' section stating: 'This Learner Lab provides a sandbox environment for ad-hoc exploration of AWS services.' and 'This environment is long-lived. When the session timer runs to 0:00, the session will end, but any data'. Navigation buttons at the bottom include '< Anterior' and 'Siguiente >'.

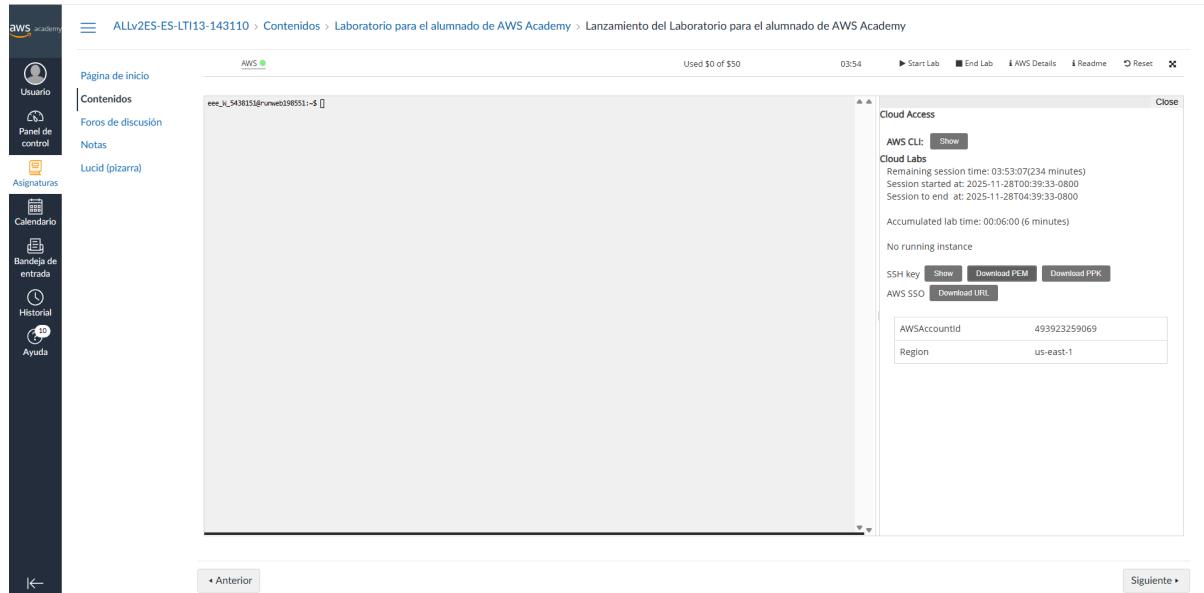
## PARTE 2: GESTIÓN DE LA CLAVE DE SEGURIDAD

Muevo la clave descargada a mi carpeta segura con cp y ejecuto chmod 400. Con esto restrinjo el archivo para que sea de 'solo lectura para mí', cumpliendo el requisito estricto de seguridad de AWS.



## PARTE 3: CONFIGURACIÓN E INSTALACIÓN EN AWS

Dentro de la consola de AWS, accedo al panel EC2 y selecciono la opción 'Lanzar la instancia' para comenzar a configurar mi nuevo servidor virtual en la nube.



Ejecuto el comando cp para copiar el archivo labsuser.pem desde la carpeta de Descargas de mi ordenador al directorio oculto .ssh de mi entorno Linux. Hago esto para tener la llave disponible dentro de WSL y no perderla.

```
alumno16@A6Alumno16:/mnt/c/Users/Alumno.DESKTOP-DISKTUG$ cp Downloads/labsuser.pem ~/.ssh/
alumno16@A6Alumno16:/mnt/c/Users/Alumno.DESKTOP-DISKTUG$ |
```

Aplico el comando chmod 400 sobre mi clave privada. Con esto limito los permisos para que el archivo sea de 'solo lectura' exclusivamente para mi usuario, cumpliendo así con el requisito obligatorio de seguridad que exige AWS para conectarse.

```
alumno16@A6Alumno16:/mnt/c/Users/Alumno.DESKTOP-DISKTUG$ chmod 400 ~/.ssh/labsuser.pem
alumno16@A6Alumno16:/mnt/c/Users/Alumno.DESKTOP-DISKTUG$ |
```

Accedo al panel principal del servicio EC2 en la consola de AWS para tener una visión general de mis recursos en la nube. Desde aquí compruebo la región en la que estoy trabajando (Norte de Virginia) y visualizo el estado de mis instancias y pares de claves antes de comenzar.

The screenshot shows the AWS EC2 dashboard with the following details:

- Panel:** Vista global de EC2, Eventos.
- Instancias:** Instantáneas, Tipos de instancia, Plantillas de lanzamiento, Solicitudes de spot, Savings Plans, Instances reservadas, Almacenamientos dedicados, Reservas de capacidad, Capacity Manager (New).
- Recursos:** Actualmente, utiliza los siguientes recursos de Amazon EC2 en la región Estados Unidos (Norte de Virginia):
 

Instancias (en ejecución)	0	Balanceadores de carga	0	Capacity Reservations	0
Direcciones IP elásticas	0	Grupos de escalamiento automático	0	Grupos de seguridad	1
Grupos de ubicación	0	Hosts dedicados	0	Instancias	0
Instantáneas	0	Pares de claves	1	Volumenes	0
- Lanzar la instancia:** Para comenzar, lance una instancia de Amazon EC2, que es un servidor virtual en la nube. Buttons: Lanzar la instancia (orange), Migrar un servidor.
- Estado del servicio:** Panel de AWS Health, Región: Estados Unidos (Norte de Virginia), Estado: Este servicio funciona con normalidad.
- Zonas:** Nombre de la zona | ID de la zona
 

us-east-1a	use1-az1
us-east-1b	use1-az2
us-east-1c	use1-az4
us-east-1d	use1-az6
us-east-1e	use1-az3
us-east-1f	use1-az5
- Atributos de la cuenta:** VPC predeterminada (vpc-03428f9b6aac6e5af), Configuración (Protección y seguridad de datos, AMI permitidas, Zonas, Consola de serie de EC2, Especificación de crédito predeterminada, Preferencias de la consola de EC2).
- Información adicional:** Instrucciones para comenzar, Guía de introducción, Documentación, Todos los recursos de EC2, Foros, Precios, Póngase en contacto con nosotros.

Localizo y selecciono el botón naranja 'Lanzar la instancia' para iniciar el asistente de creación de mi nuevo servidor virtual. Este es el paso inicial donde comenzaré a definir el sistema operativo y las características técnicas de la máquina que voy a utilizar.

## Lanzar la instancia

Para comenzar, lance una instancia de Amazon EC2, que es un servidor virtual en la nube.

**Lanzar la instancia** ▾ **Migrar un servidor ↗**

Nota: Sus instancias se lanzarán en la región Estados Unidos (Norte de Virginia)

Asigno el nombre servidor-web-practica a mi máquina y selecciono la imagen de sistema operativo Ubuntu Server 24.04 LTS. Verifico también que estoy usando el tipo de instancia gratuito t3.micro.

**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**

servidor-web-practica [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**.

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

**Inicio rápido**

Amazon Linux	macOS	Ubuntu	Windows	Red Hat	SUSE Linux	Debian

[Buscar más AMI](#) Inclusión de AMI de AWS, Marketplace y la comunidad

**Imágenes de máquina de Amazon (AMI)**

**Ubuntu Server 24.04 LTS (HVM), SSD Volume Type** [Apto para la capa gratuita](#)

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

Selecciono el tipo de instancia t3.micro tras verificar que lleva la etiqueta 'Apto para la capa gratuita' para evitar costes. Además, en el apartado de par de claves, elijo obligatoriamente vockey para asegurar que podré conectarme a la máquina usando la clave privada que descargué del laboratorio.

**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

Todas las generaciones

[Comparar tipos de instancias](#)

**Par de claves (inicio de sesión)** Información

Puede utilizar un par de claves para conectarse de forma segura a la instancia. Asegúrese de que tiene acceso al par de claves seleccionado antes de lanzar la instancia.

**Nombre del par de claves - *obligatorio***

vockey [Crear un nuevo par de claves](#)

Configuro la red creando un nuevo grupo de seguridad llamado security-group-servidores-web. Me aseguro de que la opción de asignar IP pública esté habilitada para poder conectarme desde mi casa.

▼ **Configuraciones de red** [Información](#)

**VPC : obligatorio** | [Información](#)

vpc-03428f9b8aac6e5af (predeterminado) ▾ [C](#)

172.31.0.0/16

**Subred** | [Información](#)

Sin preferencias ▾ [C](#) [Crear nueva subred](#) ↗

**Zona de disponibilidad** | [Información](#)

Sin preferencias ▾ [C](#) [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](#)

Configuro la red creando un nuevo grupo de seguridad llamado security-group-servidores-web. Me aseguro de que la opción de asignar IP pública esté habilitada para poder conectarme desde mi casa.

Defino las reglas del firewall manualmente. Abro el puerto 22 para SSH y añado reglas personalizadas para los puertos 8080 (Apache), 8081 (Nginx), 8082 (Caddy) y 8443 (SSL), permitiendo el tráfico desde cualquier lugar.

**Nombre del grupo de seguridad - obligatorio**

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 de grupos de seguridad de entrada**

- ▼ Regla del grupo de seguridad 1 (TCP, 22, 0.0.0.0/0, Acceso SSH) Eliminar

<b>Tipo   Información</b>	<b>Protocolo   Información</b>	<b>Intervalo de puertos   Información</b>
TCP personalizado	TCP	22

<b>Tipo de origen   Información</b>	<b>Origen   Información</b>	<b>Descripción - opcional   Información</b>
Cualquier lugar	<input type="text" value="0.0.0.0/0"/> <span style="float: right;">X</span>	Acceso SSH
- ▼ Regla del grupo de seguridad 2 (TCP, 8080, 0.0.0.0/0, Apache HTTP) Eliminar

<b>Tipo   Información</b>	<b>Protocolo   Información</b>	<b>Intervalo de puertos   Información</b>
TCP personalizado	TCP	8080

<b>Tipo de origen   Información</b>	<b>Origen   Información</b>	<b>Descripción - opcional   Información</b>
Cualquier lugar	<input type="text" value="0.0.0.0/0"/> <span style="float: right;">X</span>	Apache HTTP
- ▼ Regla del grupo de seguridad 3 (TCP, 8081, 0.0.0.0/0, Nginx) Eliminar

<b>Tipo   Información</b>	<b>Protocolo   Información</b>	<b>Intervalo de puertos   Información</b>
TCP personalizado	TCP	8081

<b>Tipo de origen   Información</b>	<b>Origen   Información</b>	<b>Descripción - opcional   Información</b>
Cualquier lugar	<input type="text" value="0.0.0.0/0"/> <span style="float: right;">X</span>	Nginx
- ▼ Regla del grupo de seguridad 4 (TCP, 8082, 0.0.0.0/0, Caddy) Eliminar

<b>Tipo   Información</b>	<b>Protocolo   Información</b>	<b>Intervalo de puertos   Información</b>
TCP personalizado	TCP	8082

<b>Tipo de origen   Información</b>	<b>Origen   Información</b>	<b>Descripción - opcional   Información</b>
Cualquier lugar	<input type="text" value="0.0.0.0/0"/> <span style="float: right;">X</span>	Caddy

▼ Regla del grupo de seguridad 5 (TCP, 8443, 0.0.0.0/0, Apache SSL)

**Tipo** | [Información](#)

TCP personalizado ▾

**Tipo de origen** | [Información](#)

Cualquier lugar ▾

**Protocolo** | [Información](#)

TCP

**Intervalo de puertos** | [Información](#)

8443

**Origen** | [Información](#)

Agregue CIDR, lista de prefijos o grupo de seguridad ▾

0.0.0.0/0 X

**Descripción - opcional** | [Información](#)

Apache SSL

[Eliminar](#)

Reviso el resumen final de la configuración, confirmando el almacenamiento de 8GB y el nuevo grupo de seguridad. Pulso 'Lanzar instancia' para que AWS empiece a crear la máquina.

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

①
Fullscreen
↶

ⓘ Haga clic en actualizar para ver la información de la copia de seguridad  
Las etiquetas que asigne determinan si alguna política de Data Lifecycle Manager realizará una copia de seguridad de la instancia.

0 x sistemas de archivos Editar

**► Detalles avanzados** [Información](#)

**▼ Resumen**

Número de instancias | [Información](#)

1

**Imagen de software (AMI)**  
Canonical, Ubuntu, 24.04, amd64...[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](#)

[Código de versión preliminar](#)

AWS me confirma que el lanzamiento se ha iniciado correctamente. Desde aquí accedo al enlace del ID de la instancia para ver sus detalles y estado.

The screenshot shows the AWS EC2 console with the following details:

- Header:** AWS logo, search bar (Buscar), keyboard shortcut [Alt+S], notifications, account information (ID de cuenta: 4939-2325-9069, Estados Unidos (Norte de), vclabs/user4626052=Mario).
- Breadcrumbs:** EC2 > Instancias > Lanzar una instancia
- Success Message:** A green box indicates "Correcto" (Correct) and states "El lanzamiento de la instancia se inició correctamente (i-051c5a896243cbc8b)".
- Section:** ▶ Registro de lanzamiento
- Section:** Pasos siguientes
- Card 1: Crea alertas de uso de facturación**

Para controlar los costos y evitar cargos inesperados, configure notificaciones por correo electrónico que avisen cuando se alcancen ciertos umbrales de uso.

[Crear alertas de facturación ↗](#)
- Card 2: Conectarse a la instancia**

Una vez que la instancia esté en ejecución, inicie sesión en ella desde el equipo local.

[Conectarse a la instancia ↗](#)
- Card 3: Conectar una base de datos de RDS**

Configure la conexión entre una instancia de EC2 y una base de datos para permitir el flujo de tráfico entre ellas.

[Conectar una base de datos de RDS ↗](#)
- Bottom Right:** [Ver todas las instancias](#)

Verifico que mi instancia está en estado 'En ejecución'. Aquí copio mi dirección IP Pública (44.192.69.137), que es el dato fundamental que necesito para la conexión remota.

The screenshot shows the AWS EC2 Instance Details page for instance **i-051c5a896243cbc8b**. The instance is currently running. Key details displayed include:

- ID de la instancia:** i-051c5a896243cbc8b
- Dirección IPv4 pública:** 44.192.69.137 (with a link to copy)
- Direcciones IPv4 privadas:** 172.31.68.60
- Estado de la instancia:** En ejecución (Running)
- Tipo de nombre de anfitrión:** Nombre de IP: ip-172-31-68-60.ec2.internal
- Tipo de instancia:** t3.micro
- Dirección IP asignada automáticamente:** 44.192.69.137 [IP pública]
- Hallazgo de AWS Compute Optimizer:** Suscribirse a AWS Compute Optimizer para recibir recomendaciones. (with a link to more information)
- ID de subred:** subnet-0c2bfff4e78ebd14e1
- Dirección IPv6:** -
- DNS público:** ec2-44-192-69-137.compute-1.amazonaws.com (with a link to copy)
- Nombre DNS de IP privada (solo IPv4):** ip-172-31-68-60.ec2.internal
- Responder al nombre DNS de recurso privado:** IPv4 (A)
- Direcciones IP elásticas:** -
- ID de VPC:** vpc-03428f9b8aac6e5af
- Rol de IAM:** -
- Nombre del grupo de Auto Scaling:** -

The left sidebar shows the navigation menu for EC2, including sections for Panel, Instances, Images, Elastic Block Store, and Red y seguridad.

Realizo la conexión remota ejecutando ssh -i ... ubuntu@IP. Al aceptar la huella digital, el prompt cambia a verde, confirmando que ya estoy dentro del servidor de Amazon y listo para trabajar.

```
alumno16@A6Alumno16:/mnt/c/Users/Alumno.DESKTOP-DI5KTUG$ ssh -i ~/.ssh/labsuser.pem ubuntu@44.192.69.1
37
The authenticity of host '44.192.69.137 (44.192.69.137)' can't be established.
ED25519 key fingerprint is SHA256:QL0/GdRu3FmSoKENwPTMMV3La7sC22bMi6Fob665SFc.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '44.192.69.137' (ED25519) to the list of known hosts.
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-1015-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

System information as of Fri Nov 28 10:11:02 UTC 2025

System load:  0.0          Temperature:      -273.1 C
Usage of /:   25.8% of 6.71GB  Processes:       116
Memory usage: 23%           Users logged in:  0
Swap usage:   0%            IPv4 address for ens5: 172.31.68.60

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-68-60:~$ |
```

## PARTE 4: CONEXIÓN SSH Y DESPLIEGUE DE SERVICIOS

Comienzo actualizando los repositorios con apt update e instalo el servidor Apache2. Esto prepara la base para configurar mi primer servicio web.

```
ubuntu@ip-172-31-68-60:~$ 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://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
Get:5 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Get:8 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [1340 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [301 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [269 kB]
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse Translation-en [118 kB]
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Components [35.0 kB]
Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 c-n-f Metadata [8328 B]
Get:14 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [1620 kB]
Get:15 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main Translation-en [303 kB]
Get:16 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Components [175 kB]
Get:17 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 c-n-f Metadata [15.7 kB]
]
```

Ejecuto el comando sudo apt install apache2 -y para descargar e instalar el servidor web Apache junto con todas sus dependencias necesarias. El sistema conecta con los repositorios de Ubuntu, baja los paquetes y los instala automáticamente, dejando el servicio listo para ser configurado posteriormente.

```
ubuntu@ip-172-31-68-60:~$ sudo apt install apache2 -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils libapr1t64 libaprutil1-dbd-sqlite3 libaprutil1-ldap
  libaprutil1t64 liblua5.4-0 ssl-cert
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom www-browser
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils libapr1t64 libaprutil1-dbd-sqlite3 libaprutil1-ldap
  libaprutil1t64 liblua5.4-0 ssl-cert
0 upgraded, 10 newly installed, 0 to remove and 0 not upgraded.
Need to get 2086 kB of archives.
After this operation, 8090 kB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 libapr1t64 amd64 1.7.2-3
  .lubuntu0.1 [108 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libaprutil1t64 amd64 1.6.3-1.1ub
  untu7 [91.9 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libaprutil1-dbd-sqlite3 amd64 1.
  6.3-1.lubuntu7 [11.2 kB]
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libaprutil1-ldap amd64 1.6.3-1.1
  ubuntu7 [9116 B]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 liblua5.4-0 amd64 5.4.6-3build2
  [166 kB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 apache2-bin amd64 2.4.58
  -lubuntu8.8 [1331 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 apache2-data all 2.4.58-
```

Edito el archivo ports.conf cambiando 'Listen 80' por 'Listen 8080'. También modiflico el archivo del VirtualHost para que Apache sepa que debe servir la web por ese puerto nuevo.

```
GNU nano 7.2                               /etc/apache2/ports.conf
# If you just change the port or add more ports here, you will likely also
# 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>
```

Utilizo el editor nano para modificar el archivo de configuración del sitio por defecto (000-default.conf). Cambio la primera línea a <VirtualHost \*:8080> para instruir a Apache que debe escuchar y servir las páginas web específicamente a través del puerto 8080 en lugar del 80.

```
GNU nano 7.2                               /etc/apache2/sites-available/000-default.conf
<VirtualHost *:8080>
    # The ServerName directive sets the request scheme, hostname and port that
    # the server uses to identify itself. This is used when creating
    # redirection URLs. In the context of virtual hosts, the ServerName
    # specifies what hostname must appear in the request's Host: header to
    # match this virtual host. For the default virtual host (this file) this
    # value is not decisive as it is used as a last resort host regardless.
    # However, you must set it for any further virtual host explicitly.
    #ServerName www.example.com

    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

    # For most configuration files from conf-available/, which are
    # enabled or disabled at a global level, it is possible to
    # include a line for only one particular virtual host. For example the
    # following line enables the CGI configuration for this host only
    # after it has been globally disabled with "a2disconf".
    #Include conf-available/serve-cgi-bin.conf
</VirtualHost>
```

Instalo PHP y sus módulos para Apache. Luego reinicio el servicio con systemctl restart para aplicar los cambios de puerto y verifco que no hay errores.

```
ubuntu@ip-172-31-68-60:~$ 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:
  libapache2-mod-php8.3 php-common php8.3 php8.3-cli php8.3-common php8.3-opcache php8.3-readline
Suggested packages:
  php-pear
The following NEW packages will be installed:
  libapache2-mod-php libapache2-mod-php8.3 php php-common php8.3 php8.3-cli php8.3-common
  php8.3-opcache php8.3-readline
0 upgraded, 9 newly installed, 0 to remove and 0 not upgraded.
Need to get 4922 kB of archives.
After this operation, 22.4 MB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 php-common all 2:93ubuntu2 [13.9 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 php8.3-common amd64 8.3.6-0ubuntu0.24.04.5 [740 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 php8.3-opcache amd64 8.3.6-0ubuntu0.24.04.5 [371 kB]
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 php8.3-readline amd64 8.3.6-0ubuntu0.24.04.5 [13.5 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 php8.3-cli amd64 8.3.6-0ubuntu0.24.04.5 [1915 kB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 libapache2-mod-php8.3 amd64 8.3.6-0ubuntu0.24.04.5 [1851 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libapache2-mod-php all 2:8.3+93ubuntu2 [4224 B]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 php8.3 all 8.3.6-0ubuntu0.24.04.5 [9174 B]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 php all 2:8.3+93ubuntu2 [4076 B]
Fetched 4922 kB in 0s (67.8 MB/s)
Selecting previously unselected package php-common.
(Reading database ... 104110 files and directories currently installed.)
Preparing to unpack .../0-php-common_2%3a93ubuntu2_all.deb ...
```

```
ubuntu@ip-172-31-68-60:~$ sudo systemctl restart apache2
ubuntu@ip-172-31-68-60:~$ 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 Fri 2025-11-28 11:33:47 UTC; 20s ago
     Docs: https://httpd.apache.org/docs/2.4/
 Process: 17913 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
 Main PID: 17916 (apache2)
    Tasks: 6 (limit: 1008)
   Memory: 10.6M (peak: 11.0M)
      CPU: 56ms
     CGroup: /system.slice/apache2.service
             ├─17916 /usr/sbin/apache2 -k start
             ├─17918 /usr/sbin/apache2 -k start
             ├─17919 /usr/sbin/apache2 -k start
             ├─17920 /usr/sbin/apache2 -k start
             ├─17921 /usr/sbin/apache2 -k start
             └─17922 /usr/sbin/apache2 -k start

Nov 28 11:33:47 ip-172-31-68-60 systemd[1]: Starting apache2.service - The Apache HTTP Server...
Nov 28 11:33:47 ip-172-31-68-60 systemd[1]: Started apache2.service - The Apache HTTP Server.
ubuntu@ip-172-31-68-60:~$ |
```

Instalo net-tools y uso netstat para confirmar que Apache escucha en el 8080. Creo un archivo info.php y lo pruebo con curl, verificando que el servidor responde correctamente código HTML/PHP.

```
ubuntu@ip-172-31-68-60:~$ sudo apt update
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
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.
ubuntu@ip-172-31-68-60:~$ sudo apt install net-tools -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  net-tools
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 204 kB of archives.
After this operation, 811 kB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 net-tools amd64 2.10-0.1
ubuntu4.4 [204 kB]
Fetched 204 kB in 0s (11.7 MB/s)
Selecting previously unselected package net-tools.
(Reading database ... 104242 files and directories currently installed.)
Preparing to unpack .../net-tools_2.10-0.1ubuntu4.4_amd64.deb ...
Unpacking net-tools (2.10-0.1ubuntu4.4) ...
Setting up net-tools (2.10-0.1ubuntu4.4) ...
Processing triggers for man-db (2.12.0-4build2) ...
Scanning processes... [=====] ]
```

```
ubuntu@ip-172-31-68-60:~$ sudo netstat -tulpn | grep apache2
tcp6      0      0 :::8080          ::*:*              LISTEN      17916/apache2
ubuntu@ip-172-31-68-60:~$
```

```
ubuntu@ip-172-31-68-60:~$ 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;}
.center table {margin: 1em auto; text-align: left;}
.center th {text-align: center !important;}
td, th {border: 1px solid #666; font-size: 75%; vertical-align: baseline; padding: 4px 5px;}
th {position: sticky; top: 0; background: inherit;}
h1 {font-size: 150%;}
h2 {font-size: 125%;}
h2 a:link, h2 a:visited{color: inherit; background: inherit;}
.p {text-align: left;}
.e {background-color: #ccf; width: 300px; font-weight: bold;}
.h {background-color: #99c; font-weight: bold;}
.v {background-color: #ddd; max-width: 300px; overflow-x: auto; word-wrap: break-word;}
.v i {color: #999;}
img {float: right; border: 0;}
hr {width: 934px; background-color: #ccc; border: 0; height: 1px;}
:root {--php-dark-grey: #333; --php-dark-blue: #4F5B93; --php-medium-blue: #8892BF; --php-light-blue: #E2E4EF; --php-accent-purple: #793862}@media (prefers-color-scheme: dark) {
  body {background: var(--php-dark-grey); color: var(--php-light-blue)}
```

Instalo el segundo servidor web, Nginx. Al instalarse, intento configurarlo para que no entre en conflicto con los otros servicios.

```
</div></body></html>ubuntu@ip-172-31-68-69:~$ 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
The following NEW packages will be installed:
  nginx nginx-common
0 upgraded, 2 newly installed, 0 to remove and 0 not upgraded.
Need to get 564 kB of archives.
After this operation, 1596 kB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 nginx-common all 1.24.0-2ubuntu7.5 [43.4 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 nginx amd64 1.24.0-2ubuntu7.5 [520 kB]
Fetched 564 kB in 0s (25.0 MB/s)
Preconfiguring packages ...
Selecting previously unselected package nginx-common.
(Reading database ... 104290 files and directories currently installed.)
Preparing to unpack .../nginx-common_1.24.0-2ubuntu7.5_all.deb ...
Unpacking nginx-common (1.24.0-2ubuntu7.5) ...
Selecting previously unselected package nginx.
Preparing to unpack .../nginx_1.24.0-2ubuntu7.5_amd64.deb ...
Unpacking nginx (1.24.0-2ubuntu7.5) ...
Setting up nginx-common (1.24.0-2ubuntu7.5) ...
Created symlink /etc/systemd/system/multi-user.target.wants/nginx.service → /usr/lib/systemd/system/nginx.service.
Setting up nginx (1.24.0-2ubuntu7.5) ...
[Progress: [ 78%] [########################################.....]]
```

Edito el archivo default de Nginx cambiando listen al puerto 8081. Además, creo un archivo index.html personalizado usando echo con el texto 'Servidor Nginx', para poder distinguir visualmente este servidor de los demás cuando acceda desde el navegador.

```
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;
    #
    # Note: You should disable gzip for SSL traffic.
    # See: https://bugs.debian.org/773332
    #
    # Read up on ssl_ciphers to ensure a secure configuration.
    # See: https://bugs.debian.org/765782
    #
    # Self signed certs generated by the ssl-cert package
    # Don't use them in a production server!
    #
    # include snippets/snakeoil.conf;

    root /var/www/html;

    # Add index.php to the list if you are using PHP
    index index.html index.htm index.nginx-debian.html;

    server_name _;
```

```
ubuntu@ip-172-31-68-60:~$ 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>
ubuntu@ip-172-31-68-60:~$ |
```

Ejecuto dos acciones consecutivas para aplicar los cambios en Nginx: primero utilizo sudo systemctl restart nginx para reiniciar el servicio y que cargue la nueva configuración del puerto 8081, e inmediatamente después lanzo sudo systemctl status nginx para confirmar visualmente que el servidor se ha levantado correctamente y se encuentra en estado 'active (running)'.

```
ubuntu@ip-172-31-68-60:~$ sudo systemctl restart nginx
ubuntu@ip-172-31-68-60:~$ 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 Fri 2025-11-28 11:41:33 UTC; 6s ago
    Docs: man:nginx(8)
 Process: 18699 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=exited, >
 Process: 18702 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=0/>
 Main PID: 18703 (nginx)
   Tasks: 3 (limit: 1008)
     Memory: 2.4M (peak: 2.8M)
       CPU: 16ms
      CGroup: /system.slice/nginx.service
              ├─18703 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"
              ├─18704 "nginx: worker process"
              ├─18705 "nginx: worker process"

Nov 28 11:41:33 ip-172-31-68-60 systemd[1]: Starting nginx.service - A high performance web server and a reverse proxy server...
Nov 28 11:41:33 ip-172-31-68-60 systemd[1]: Started nginx.service - A high performance web server and a reverse proxy server.
lines 1-17/17 (END)
```

Verifico con netstat que el puerto 8081 está ocupado por Nginx y hago una petición curl local para confirmar que me devuelve mi página personalizada.

```
ubuntu@ip-172-31-68-60:~$ sudo netstat -tulpn | grep nginx
tcp        0      0 0.0.0.0:8081          0.0.0.0:*          LISTEN      18703/nginx: master
tcp6       0      0 :::8081           ::::*           LISTEN      18703/nginx: master
ubuntu@ip-172-31-68-60:~$ |

ubuntu@ip-172-31-68-60:~$ curl http://localhost:8081
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<!--
Modified from the Debian original for Ubuntu
Last updated: 2022-03-22
See: https://launchpad.net/bugs/1966004
-->
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>Apache2 Ubuntu Default Page: It works</title>
<style type="text/css" media="screen">
* {
  margin: 0px 0px 0px 0px;
  padding: 0px 0px 0px 0px;
}

body, html {
  padding: 3px 3px 3px 3px;
  background-color: #D8DBE2;

  font-family: Ubuntu, Verdana, sans-serif;
  font-size: 11pt;
  text-align: center;
}
```

Preparo la instalación de Caddy descargando las llaves GPG y añadiendo los repositorios oficiales con curl y tee. Esto es necesario porque Caddy es un software moderno que requiere repositorios externos para instalar su última versión estable.

```
ubuntu@ip-172-31-68-60:~$ sudo apt install -y debian-keyring debian-archive-keyring apt-transport-https
s 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 0 not upgraded.
Need to get 31.5 MB of archives.
After this operation, 33.4 MB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 apt-transport-https
all 2.8.3 [3970 B]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 debian-archive-keyring all 2
023.4ubuntu1 [168 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 debian-keyring all 2023.12.2
4 [31.3 MB]
55% [3 debian-keyring 16.1 MB/31.3 MB 51%]
```

Ejecuto este primer comando para descargar la clave de seguridad GPG de Caddy y guardarla en el anillo de claves de mi sistema (/usr/share/keyrings). Esto sirve para autenticar el software: le estoy diciendo a mi Ubuntu que confie en los paquetes firmados digitalmente por los desarrolladores oficiales de Caddy, evitando así instalar versiones falsas o modificadas.

Con este segundo comando, añado la dirección web del repositorio oficial a la lista de fuentes de mi sistema dentro de la carpeta /etc/apt/sources.list.d/. Utilizo tee con permisos de administrador para crear el archivo de configuración necesario, permitiendo que a partir de ahora el gestor de paquetes apt sepa dónde buscar para encontrar y descargar Caddy.

```
ubuntu@ip-172-31-68-60:~$ curl -lsLF 'https://dl.cloudsmith.io/public/caddy/stable/gpg.key' | sudo gpg
--dearmor -o /usr/share/keyrings/caddy-stable-archive-keyring.gpg
ubuntu@ip-172-31-68-60:~$ curl -lsLF 'https://dl.cloudsmith.io/public/caddy/stable/debian.deb.txt' | s
udo 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
ubuntu@ip-172-31-68-60:~$ |
```

Instalo Caddy y preparo su contenido: creo una carpeta, un archivo Markdown de bienvenida y descargo una imagen de prueba para verificar que puede servir diferentes tipos de archivos.

```
ubuntu@ip-172-31-68-60:~$ 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
Hit:4 http://security.ubuntu.com/ubuntu noble-security InRelease
Get:5 https://dl.cloudsmith.io/public/caddy/stable/deb/debian any-version InRelease [14.8 kB]
Get:6 https://dl.cloudsmith.io/public/caddy/stable/deb/debian any-version/main amd64 Packages [4329 B]
Fetched 19.1 kB in 1s (28.6 kB/s)
```

Una vez instalado el servidor, preparo el entorno creando un directorio específico con mkdir -p /var/www/caddy. A continuación, genero un archivo README.md utilizando echo y tee para escribir un mensaje de bienvenida en formato Markdown. Por último, descargo una imagen de prueba desde internet con curl y la muevo a este directorio; estos archivos me servirán para comprobar más adelante que Caddy es capaz de listar directorios y servir diferentes tipos de archivos estáticos correctamente.

```
ubuntu@ip-172-31-68-60:~$ sudo mkdir -p /var/www/caddy
ubuntu@ip-172-31-68-60:~$ echo "# Bienvenido a Caddy" | sudo tee /var/www/caddy/README.md
# Bienvenido a Caddy
ubuntu@ip-172-31-68-60:~$ echo "" | sudo tee -a /var/www/caddy/README.md

ubuntu@ip-172-31-68-60:~$ echo "Este servidor está funcionando correctamente." | sudo tee -a /var/www/caddy/README.md
Este servidor está funcionando correctamente.
ubuntu@ip-172-31-68-60:~$ echo "" | sudo tee -a /var/www/caddy/README.md

ubuntu@ip-172-31-68-60:~$ echo "## Características" | sudo tee -a /var/www/caddy/README.md
## Características
ubuntu@ip-172-31-68-60:~$ echo "- Servidor moderno" | sudo tee -a /var/www/caddy/README.md
- Servidor moderno
ubuntu@ip-172-31-68-60:~$ echo "- HTTPS automático" | sudo tee -a /var/www/caddy/README.md
- HTTPS automático
ubuntu@ip-172-31-68-60:~$ echo "- Fácil configuración" | sudo tee -a /var/www/caddy/README.md
- Fácil configuración
ubuntu@ip-172-31-68-60:~$ 

ubuntu@ip-172-31-68-60:~$ curl -o /tmp/test-image.jpg "https://www.python.org/static/apple-touch-icon-144x144-precomposed.png"
% Total    % Received % Xferd  Average Speed   Time     Time      Current
          Dload  Upload   Total Spent  Left Speed
100  7382  100  7382    0     0  173k      0 --:--:-- --:--:-- 175k
ubuntu@ip-172-31-68-60:~$ sudo mv /tmp/test-image.jpg /var/www/caddy/test.jpg
ubuntu@ip-172-31-68-60:~$ |
```

Edité el archivo de configuración Caddyfile definiendo el puerto 8082 y estableciendo la raíz del servidor en /var/www/caddy. Habilite la directiva file\_server browse para permitir la navegación de archivos y reinicié el servicio con systemctl restart caddy para aplicar los cambios y verificar que el estado es "active (running)".

```
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 {
    root * /var/www/caddy
    file_server browse

    @markdown path *.md
    header @markdown Content-Type text/plain
}
:8082{|

# Refer to the Caddy docs for more information:
# https://caddyserver.com/docs/caddyfile
```

```
ubuntu@ip-172-31-68-60:~$ sudo systemctl restart caddy
```

```
ubuntu@ip-172-31-68-60:~$ sudo systemctl status caddy
● caddy.service - Caddy
   Loaded: loaded (/usr/lib/systemd/system/caddy.service; enabled; preset: enabled)
   Active: active (running) since Fri 2025-11-28 11:59:24 UTC; 33s ago
     Docs: https://caddyserver.com/docs/
   Main PID: 19659 (caddy)
      Tasks: 8 (limit: 1008)
     Memory: 10.4M (peak: 10.9M)
        CPU: 87ms
       CGroup: /system.slice/caddy.service
               └─19659 /usr/bin/caddy run --environ --config /etc/caddy/Caddyfile

Nov 28 11:59:24 ip-172-31-68-60 caddy[19659]: {"level":"info","ts":1764331164.05806,"logger":"tls.obe>
Nov 28 11:59:24 ip-172-31-68-60 caddy[19659]: {"level":"info","ts":1764331164.0585442,"logger":"http">
Nov 28 11:59:24 ip-172-31-68-60 caddy[19659]: {"level":"info","ts":1764331164.0585935,"logger":"http">
Nov 28 11:59:24 ip-172-31-68-60 caddy[19659]: {"level":"info","ts":1764331164.061676,"logger":"http",>
Nov 28 11:59:24 ip-172-31-68-60 caddy[19659]: {"level":"info","ts":1764331164.3545523,"logger":"http">
Nov 28 11:59:24 ip-172-31-68-60 caddy[19659]: {"level":"info","ts":1764331164.3613696,"logger":"http">
Nov 28 11:59:24 ip-172-31-68-60 caddy[19659]: {"level":"info","ts":1764331164.3614109,"logger":"http">
Nov 28 11:59:24 ip-172-31-68-60 caddy[19659]: {"level":"info","ts":1764331164.361439,"logger":"http",>
Nov 28 11:59:24 ip-172-31-68-60 caddy[19659]: {"level":"error","ts":1764331164.4114769,"logger":"tls.>
Nov 28 11:59:24 ip-172-31-68-60 caddy[19659]: {"level":"error","ts":1764331164.4115312,"logger":"tls.>
[lines 1-21/21 (END)]
```

Utilicé netstat para confirmar que Caddy está escuchando correctamente en el puerto 8082.

Posteriormente, realicé peticiones locales con curl tanto a la raíz como al archivo README.md que creé anteriormente, verificando que el servidor me devuelva el contenido esperado.

```
ubuntu@ip-172-31-68-60:~$ sudo netstat -tulpn | grep caddy
tcp        0      0 127.0.0.1:2019          0.0.0.0:*                  LISTEN      19659/caddy
tcp6       0      0 :::80                   ::::*                   LISTEN      19659/caddy
tcp6       0      0 :::443                 ::::*                   LISTEN      19659/caddy
udp6       0      0 :::443                 ::::*                   LISTEN      19659/caddy
ubuntu@ip-172-31-68-60:~$ |
```

```
ubuntu@ip-172-31-68-60:~$ curl http://localhost:8082
Hola Caddyubuntu@ip-172-31-68-60:~$ curl http://localhost:8082
Hola Caddyubuntu@ip-172-31-68-60:~$ |
```

```
ubuntu@ip-172-31curl http://localhost:8082/README.md/README.md
ubuntu@ip-172-31-68-60:~$ |
```

Instalé el paquete certbot y las dependencias de Python para Apache. A continuación, generé un certificado SSL autofirmado válido por 365 días utilizando openssl, completando los datos de ubicación (España, Madrid) y configurando el "Common Name" como localhost para las pruebas seguras.

```
Hola Caddyubuntu@ip-172-31sudo apt install certbot python3-certbot-apache -y-apache -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  augeas-lenses libaugeas0 python3-acme python3-augeas python3-certbot python3-configargparse
  python3-icu python3-josepy python3-parsedatetime python3-rfc3339
Suggested packages:
  augeas-doc python-certbot-doc python3-certbot-nginx augeas-tools python-acme-doc
  python-certbot-apache-doc
The following NEW packages will be installed:
  augeas-lenses certbot libaugeas0 python3-acme python3-augeas python3-certbot
  python3-certbot-apache python3-configargparse python3-icu python3-josepy python3-parsedatetime
  python3-rfc3339
0 upgraded, 12 newly installed, 0 to remove and 0 not upgraded.
Need to get 1657 kB of archives.
After this operation, 8599 kB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 augeas-lenses all 1.14.1-1ubuntu2 [323 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 libaugeas0 amd64 1.14.1-1ubuntu2 [166 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 python3-josepy all 1.14.0-1 [22.1 kB]
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 python3-rfc3339 all 1.1-4 [6744 B]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 python3-acme all 2.9.0-1 [48.5 kB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 python3-augeas all 0.5.0-1.1 [9124 B]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 python3-configargparse all 1.7-1 [31.7 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 python3-parsedatetime all 2.6-3 [32.8 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 python3-certbot all 2.9.0-1 [267 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 certbot all 2.9.0-1 [89.2 kB]
```

Habilité el módulo SSL en Apache con a2enmod y edité ports.conf para añadir la escucha en el puerto 8443. Luego, configuré el VirtualHost en el archivo default-ssl.conf para usar ese puerto y activé el motor SSL (SSLEngine on), reiniciando finalmente el servicio para aplicar todo

```
ubuntu@ip-172-31-68-60:~$ 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
ubuntu@ip-172-31-68-60:~$ |
```

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

Listen 8443

<IfModule ssl_module>
    Listen 443
</IfModule>

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

```

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

    # For most configuration files from conf-available/, which are
    # enabled or disabled at a global level, it is possible to
    # include a line for only one particular virtual host. For example the
    # following line enables the CGI configuration for this host only
    # after it has been globally disabled with "a2disconf".
    #Include conf-available/serve-cgi-bin.conf

    #   SSL Engine Switch:
    #     Enable/Disable SSL for this virtual host.
    SSLEngine on

```

```
ubuntu@ip-172-31-68-60:~$ sudo systemctl restart apache2
```

Realicé una petición curl al puerto 8443 usando el parámetro -k para ignorar la advertencia de seguridad del certificado autofirmado. La respuesta "HTTP/1.1 200 OK" y la visualización del código HTML confirman que el servidor Apache está sirviendo contenido correctamente a través de HTTPS.

```

ubuntu@ip-172-31-68-60:~$ curl -i -k https://localhost:8443
HTTP/1.1 200 OK
Date: Fri, 28 Nov 2025 12:26:32 GMT
Server: Apache/2.4.58 (Ubuntu)
Last-Modified: Fri, 28 Nov 2025 11:31:55 GMT
ETag: "29af-644a5fa0d4537"
Accept-Ranges: bytes
Content-Length: 10671
Vary: Accept-Encoding
Content-Type: text/html

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<!--
    Modified from the Debian original for Ubuntu
    Last updated: 2022-03-22
    See: https://launchpad.net/bugs/1966004
-->
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <title>Apache2 Ubuntu Default Page: It works</title>
    <style type="text/css" media="screen">
        *
        margin: 0px 0px 0px 0px;
        padding: 0px 0px 0px 0px;
    }

    body, html {
        padding: 3px 3px 3px 3px;
        background-color: #D8DBE2;

        font-family: Ubuntu, Verdana, sans-serif;
        font-size: 11pt;
        text-align: center;
    }

    div.main_page {

```

Verifiqué el estado de todos los servicios simultáneamente: comprobé que Apache corre en los puertos 8080 y 8443, Nginx está activo en el puerto 8081 y Caddy funciona en el 8082.

Utilicé systemctl status y netstat para asegurar que no hay conflictos y que cada servidor ocupa su puerto asignado.

```
ubuntu@ip-172-31-68-60:~$ sudo netstat -tulpn | grep apache2
tcp6      0      0 :::8443          :::*          LISTEN      20174/apache2
tcp6      0      0 :::443           :::*          LISTEN      20174/apache2

ubuntu@ip-172-31-68-60:~$ 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 Fri 2025-11-28 12:23:30 UTC; 5min ago
     Docs: https://httpd.apache.org/docs/2.4/
 Process: 20170 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
 Main PID: 20174 (apache2)
    Tasks: 6 (limit: 1008)
   Memory: 11.8M (peak: 12.3M)
      CPU: 82ms
     CGroup: /system.slice/apache2.service
             ├─20174 /usr/sbin/apache2 -k start
             ├─20176 /usr/sbin/apache2 -k start
             ├─20177 /usr/sbin/apache2 -k start
             ├─20178 /usr/sbin/apache2 -k start
             ├─20179 /usr/sbin/apache2 -k start
             └─20180 /usr/sbin/apache2 -k start

Nov 28 12:23:30 ip-172-31-68-60 systemd[1]: Starting apache2.service - The Apache HTTP Server...
Nov 28 12:23:30 ip-172-31-68-60 systemd[1]: Started apache2.service - The Apache HTTP Server.

ubuntu@ip-172-31-68-60:~$ sudo netstat -tulpn | grep nginx
tcp        0      0 0.0.0.0:8081          0.0.0.0:*          LISTEN      18703/nginx: master
tcp6      0      0 :::8081           :::*          LISTEN      18703/nginx: master

ubuntu@ip-172-31-68-60:~$ 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 Fri 2025-11-28 11:41:33 UTC; 47min ago
     Docs: man:nginx(8)
 Main PID: 18703 (nginx)
    Tasks: 3 (limit: 1008)
   Memory: 2.4M (peak: 2.9M)
      CPU: 17ms
     CGroup: /system.slice/nginx.service
             ├─18703 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"
             ├─18704 "nginx: worker process"
             └─18705 "nginx: worker process"

Nov 28 11:41:33 ip-172-31-68-60 systemd[1]: Starting nginx.service - A high performance web server and>
Nov 28 11:41:33 ip-172-31-68-60 systemd[1]: Started nginx.service - A high performance web server and >
lines 1-15/15 (END)
```

```

ubuntu@ip-172-31-68-60:~$ sudo netstat -tulpn | grep caddy
tcp      0      0 127.0.0.1:2019          0.0.0.0:*
                                         ::::*
LISTEN      19753/caddy
tcp6     0      0 :::8082              ::::*
                                         ::::*
LISTEN      19753/caddy
ubuntu@ip-172-31-68-60:~$ 

ubuntu@ip-172-31-68-60:~$ sudo systemctl status caddy
● caddy.service - Caddy
   Loaded: loaded (/usr/lib/systemd/system/caddy.service; enabled; preset: enabled)
   Active: active (running) since Fri 2025-11-28 12:12:23 UTC; 17min ago
     Docs: https://caddyserver.com/docs/
     Main PID: 19753 (caddy)
        Tasks: 7 (limit: 1008)
       Memory: 9.4M (peak: 11.0M)
          CPU: 99ms
        CGroup: /system.slice/caddy.service
                  └─19753 /usr/bin/caddy run --environ --config /etc/caddy/Caddyfile

Nov 28 12:12:23 ip-172-31-68-60 caddy[19753]: {"level":"info","ts":1764331943.5666575,"logger":"admin">>
Nov 28 12:12:23 ip-172-31-68-60 caddy[19753]: {"level":"warn","ts":1764331943.5669413,"logger":"http",>
Nov 28 12:12:23 ip-172-31-68-60 caddy[19753]: {"level":"warn","ts":1764331943.566955,"logger":"http",>
Nov 28 12:12:23 ip-172-31-68-60 caddy[19753]: {"level":"info","ts":1764331943.5669603,"logger":"http.l">>
Nov 28 12:12:23 ip-172-31-68-60 caddy[19753]: {"level":"info","ts":1764331943.5671406,"msg":"autosaved">>
Nov 28 12:12:23 ip-172-31-68-60 caddy[19753]: {"level":"info","ts":1764331943.5671935,"msg":"serving i">>
Nov 28 12:12:23 ip-172-31-68-60 caddy[19753]: {"level":"info","ts":1764331943.5674686,"logger":"tls.ca">>
Nov 28 12:12:23 ip-172-31-68-60 systemd[1]: Started caddy.service - Caddy.
Nov 28 12:12:23 ip-172-31-68-60 caddy[19753]: {"level":"info","ts":1764331943.5719557,"logger":"tls", ">
Nov 28 12:12:23 ip-172-31-68-60 caddy[19753]: {"level":"info","ts":1764331943.5720556,"logger":"tls", ">

```

Ejecuté el comando netstat filtrando específicamente los puertos relevantes para comprobar el estado simultáneo de todos los servicios.

```

ubuntu@ip-172-31-68-60:~$ sudo netstat -tulpn | grep -E '8080|8081|8082|8443'
tcp      0      0 0.0.0.0:8081          0.0.0.0:*
                                         ::::*
LISTEN      18703/nginx: master
tcp6     0      0 :::8443              ::::*
                                         ::::*
LISTEN      20174/apache2
tcp6     0      0 :::8082              ::::*
                                         ::::*
LISTEN      19753/caddy
tcp6     0      0 :::8081              ::::*
                                         ::::*
LISTEN      18703/nginx: master
ubuntu@ip-172-31-68-60:~$ 

```

Realicé una comprobación rápida del funcionamiento de Apache ejecutando curl contra los puertos 8080/8081/8082/8443.

```
ubuntu@ip-172-31-68-60:~$ curl http://localhost:8080
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <!--
    Modified from the Debian original for Ubuntu
    Last updated: 2022-03-22
    See: https://launchpad.net/bugs/1966004
  -->
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <title>Apache2 Ubuntu Default Page: It works</title>
    <style type="text/css" media="screen">
      * {
        margin: 0px 0px 0px 0px;
        padding: 0px 0px 0px 0px;
      }

      body, html {
        padding: 3px 3px 3px 3px;
        background-color: #D8DBE2;
        font-family: Ubuntu, Verdana, sans-serif;
      }
    </style>
  </head>
  <body>
    <h1>It works</h1>
    <p>This is the default web page for this server.<br/>
    The web server software is running normally.<br/>
    The page you requested (<a href="http://localhost:8080">http://localhost:8080</a>) is available at<br/>
    port 8080 on this server.</p>
  </body>
</html>
```

```
ubuntu@ip-172-31-68-60:~$ curl http://localhost:8081
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <!--
    Modified from the Debian original for Ubuntu
    Last updated: 2022-03-22
    See: https://launchpad.net/bugs/1966004
  -->
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <title>Apache2 Ubuntu Default Page: It works</title>
    <style type="text/css" media="screen">
      * {
        margin: 0px 0px 0px 0px;
        padding: 0px 0px 0px 0px;
      }

      body, html {
        padding: 3px 3px 3px 3px;
        background-color: #D8DBE2;
        font-family: Ubuntu, Verdana, sans-serif;
        font-size: 11pt;
        text-align: center;
      }

      div.main_page {
        position: relative;
        display: table;
        width: 800px;
      }
    </style>
  </head>
  <body>
    <h1>It works</h1>
    <p>This is the default web page for this server.<br/>
    The web server software is running normally.<br/>
    The page you requested (<a href="http://localhost:8081">http://localhost:8081</a>) is available at<br/>
    port 8081 on this server.</p>
  </body>
</html>
```

```

ubuntu@ip-172-31-65-230:~$ curl http://localhost:8082

<!DOCTYPE html>
<html>
  <head>
    <title></title>
    <link rel="canonical" href="//" />
    <meta charset="utf-8">
    <meta name="color-scheme" content="light dark">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
<style nonce="b3b93fb8-e274-4caa-aab2-40e4b86de348">
* { padding: 0; margin: 0; box-sizing: border-box; }

body {
    font-family: Inter, system-ui, sans-serif;
    font-size: 16px;
    text-rendering: optimizespeed;
    background-color: #f3f6f7;
    min-height: 100vh;
}

img,
svg {
    vertical-align: middle;
    z-index: 1;
}

img {

```

Repetí la prueba de verificación para el servicio seguro de Apache lanzando un curl -i -k contra el puerto 8443. El sistema devolvió las cabeceras correctas y el cuerpo de la página por defecto de Ubuntu, confirmando que la configuración SSL sigue estable tras todas las instalaciones.

```

Hola Caddyubuntu@ip-172-31curl -i -k https://localhost:8443
HTTP/1.1 200 OK
Date: Fri, 28 Nov 2025 12:33:36 GMT
Server: Apache/2.4.58 (Ubuntu)
Last-Modified: Fri, 28 Nov 2025 11:31:55 GMT
ETag: "29af-644a5fa0d4537"
Accept-Ranges: bytes
Content-Length: 10671
Vary: Accept-Encoding
Content-Type: text/html

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<!--
    Modified from the Debian original for Ubuntu
    Last updated: 2022-03-22
    See: https://launchpad.net/bugs/1966004
-->
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <title>Apache2 Ubuntu Default Page: It works</title>
    <style type="text/css" media="screen">
    * {
        margin: 0px 0px 0px 0px;
        padding: 0px 0px 0px 0px;
    }

    body, html {
        padding: 3px 3px 3px 3px;
        background-color: #D8DBE2;

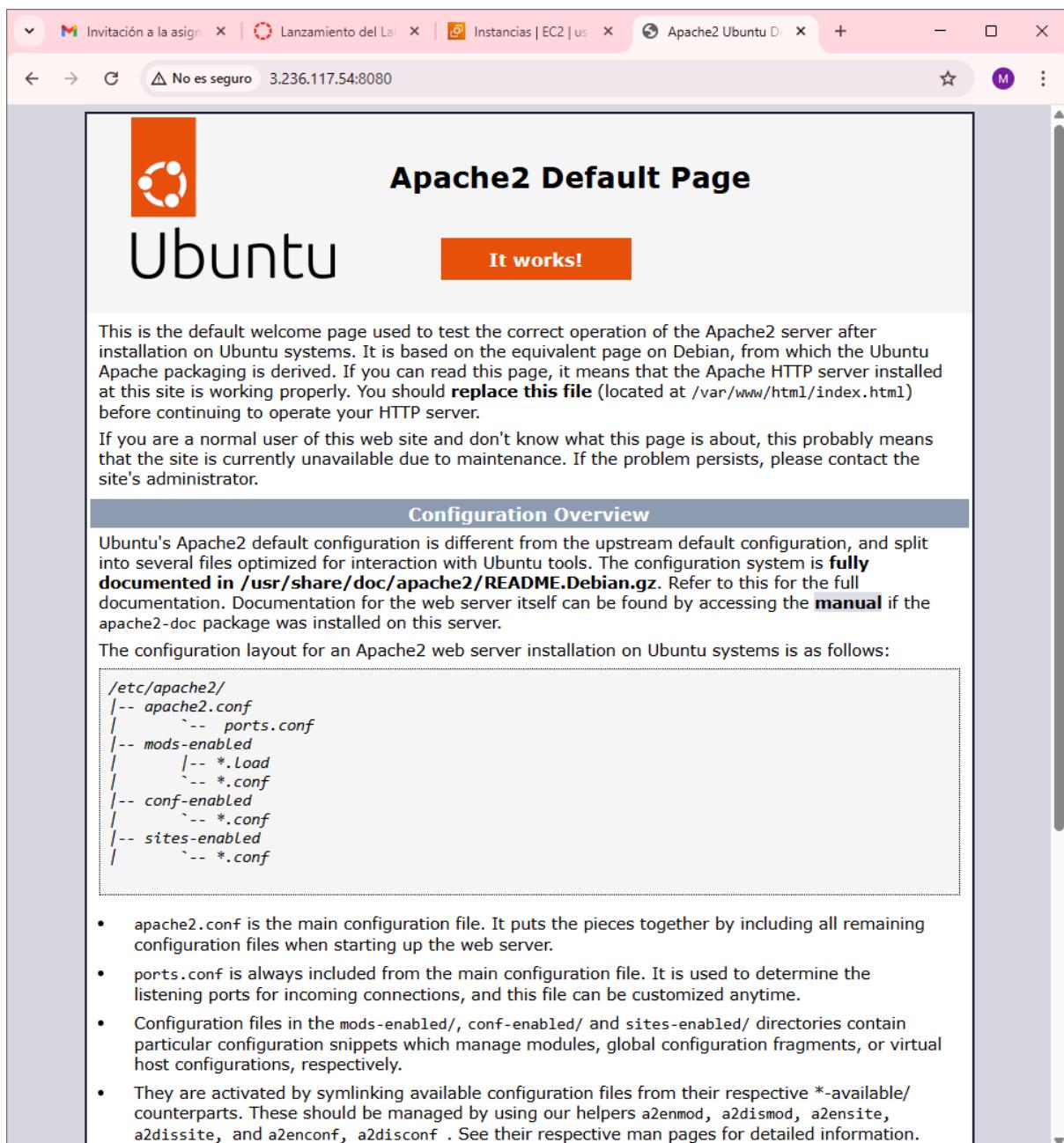
        font-family: Ubuntu, Verdana, sans-serif;
        font-size: 11pt;
        text-align: center;
    }

```

## PARTE 5: COMPROBACIONES VISUALES

Accedí a la IP pública de mi instancia a través del navegador en el puerto 8080. Esto verifica definitivamente que las reglas del Security Group están bien configuradas en AWS, permitiendo el tráfico externo y mostrando la página por defecto de Apache.

No puse las otras comprobaciones de las otras ip ya que me iba a ocupar mucho más y la práctica ya me ha quedado muy larga, las he comprobado igualmente en el google chrome y en firefox ya que en chrome no me salía alguna comprobación no se porque.



The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/  
|-- apache2.conf  
|   '-- ports.conf  
|-- mods-enabled  
|   '-- *.Load  
|   '-- *.conf  
|-- conf-enabled  
|   '-- *.conf  
|-- sites-enabled  
|   '-- *.conf
```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.
- They are activated by symlinking available configuration files from their respective `*-available/` counterparts. These should be managed by using our helpers `a2enmod`, `a2dismod`, `a2ensite`, `a2dissite`, and `a2enconf`, `a2disconf`. See their respective man pages for detailed information.

## PARTE 6: DOCUMENTACIÓN TEÓRICA

### 1. Explicación de clave PEM y su seguridad

Un archivo .pem actúa como una clave privada en un sistema de criptografía asimétrica. Funciona bajo el concepto de llave-cerradura: yo poseo la clave privada (archivo .pem) en mi máquina local, la cual nunca debe compartirse, mientras que AWS almacena la clave pública en el servidor. Es segura porque la autenticación se realiza mediante un desafío criptográfico matemático sin necesidad de enviar contraseñas a través de la red. Además, los permisos 400 en el sistema de archivos aseguran que solo el usuario propietario pueda leer la clave, evitando robos locales.

### 2. Descripción de las reglas del Security Group:

Se ha configurado un Security Group que actúa como firewall virtual controlando el tráfico de entrada:

- **Puerto 22 (SSH):** Permite la administración remota segura del servidor mediante línea de comandos.
- **Puerto 8080:** Habilitado para recibir tráfico HTTP servido por Apache.
- **Puerto 8081:** Habilitado para recibir tráfico HTTP servido por Nginx.
- **Puerto 8082:** Habilitado para recibir tráfico HTTP servido por Caddy.
- **Puerto 8443:** Habilitado para tráfico seguro HTTPS gestionado por Apache con SSL.