

# INSTALACIÓN DE WORDPRESS EN AWS CON SSH



Hecho por Juan Medrano

# INDICE

<b>Preparación del entorno local (WSL / Ubuntu).....</b>	<b>3</b>
Crear directorio ~/.ssh.....	3
Generar clave SSH Ed25519.....	3
Ajustar permisos de la clave privada.....	3
<b>Configuración en AWS.....</b>	<b>4</b>
Crear par de claves en AWS.....	4
Creación del Security Group.....	4
CREAR LA INSTANCIA EC2.....	5
CONEXIÓN SSH A LA INSTANCIA EC2.....	9
<b>INSTALACIÓN DE LAMP.....</b>	<b>10</b>
<b>CREACIÓN DEL SCRIPT DE WORDPRESS.....</b>	<b>10</b>
<b>Ejecutar el script en la instancia EC2.....</b>	<b>11</b>
<b>Ver credenciales generadas por el script.....</b>	<b>11</b>
<b>Probar WordPress desde la IP pública.....</b>	<b>12</b>
<b>Verificación del servidor.....</b>	<b>16</b>
<b>Conclusión.....</b>	<b>17</b>

# Preparación del entorno local (WSL / Ubuntu)

## Crear directorio ~/.ssh

Creación del directorio oculto ~/.ssh y asignación de permisos seguros para almacenar las claves SSH.

```
Microsoft Windows [Versión 10.0.26100.7171]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Windows\System32>wsl
juan@A6Alumno21:/mnt/c/Windows/System32$ mkdir -p ~/.ssh
chmod 700 ~/.ssh
juan@A6Alumno21:/mnt/c/Windows/System32$
```

## Generar clave SSH Ed25519

Generación de la clave SSH tipo Ed25519 para acceder a la instancia de AWS y verificación de los archivos resultantes (wordpress-key y wordpress-key.pub) en el directorio ~/.ssh.

```
Microsoft Windows [Versión 10.0.26100.7171]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Windows\System32>wsl
juan@A6Alumno21:/mnt/c/Windows/System32$ mkdir -p ~/.ssh
chmod 700 ~/.ssh
juan@A6Alumno21:/mnt/c/Windows/System32$ ssh-keygen -t ed25519 -f ~/.ssh/wordpress-key -C "mi-usuario@aws"
Generating public/private ed25519 key pair.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/juan/.ssh/wordpress-key
Your public key has been saved in /home/juan/.ssh/wordpress-key.pub
The key fingerprint is:
SHA256:2CrvjUoU7LPW2Uk1MuqExiFXB3FQIlphU3ZykKrc1MU mi-usuario@aws
The key's randomart image is:
+--[ED25519 256]--+
|
|*=#0*o.
|* B.B. E
|o B o o
|= + + +
| B + S
|o * =
|= = o
|o o o
|.o+ .
+-----[SHA256]-----+
juan@A6Alumno21:/mnt/c/Windows/System32$ ls -la ~/.ssh/wordpress-key*
-rw----- 1 juan juan 411 Dec  9 09:14 /home/juan/.ssh/wordpress-key
-rw-r--r-- 1 juan juan 96 Dec  9 09:14 /home/juan/.ssh/wordpress-key.pub
juan@A6Alumno21:/mnt/c/Windows/System32$
```

## Ajustar permisos de la clave privada

Configuración de permisos restrictivos en la clave privada wordpress-key para cumplir con las buenas prácticas de seguridad en SSH.

```
juan@A6Alumno21:/mnt/c/Windows/System32$ chmod 400 ~/.ssh/wordpress-key
ls -la ~/.ssh/wordpress-key
-r----- 1 juan juan 411 Dec  9 09:14 /home/juan/.ssh/wordpress-key
juan@A6Alumno21:/mnt/c/Windows/System32$
```

# Configuración en AWS

## Crear par de claves en AWS

Se descargó el archivo labsuser.pem desde el entorno de Vocareum, el cual actúa como la clave privada necesaria para las conexiones SSH hacia los recursos del laboratorio en AWS. La clave fue ubicada en el directorio ~/.ssh/ y se configuraron permisos seguros (chmod 400) para garantizar su correcto funcionamiento según las buenas prácticas de SSH.

```
1 juan juan 411 Dec  9 09:14 /home/juan/.ssh/wordpress-key
juan@A6Alumno21:/mnt/c/Windows/System32$ ls -la ~/.ssh/labsuser.pem
-r----- 1 juan juan 1678 Dec  4 14:08 /home/juan/.ssh/labsuser.pem
juan@A6Alumno21:/mnt/c/Windows/System32$
```

## Creación del Security Group

Se creó el Security Group wordpress-aws-sg dentro del entorno AWS proporcionado por Vocareum. Este grupo permite tráfico SSH (puerto 22), tráfico HTTP (80) y tráfico HTTPS (443), necesarios para la conexión remota segura y el despliegue del servidor WordPress en la instancia EC2.

### Crear grupo de seguridad [Información](#)

Un grupo de seguridad actúa como un firewall virtual para que la instancia controle el tráfico de entrada y salida. Para crear un nuevo grupo de seguridad, complete los campos siguientes.

#### Detalles básicos

Nombre del grupo de seguridad [Información](#)

El nombre no se puede editar después de su creación.

Descripción [Información](#)

VPC [Información](#)

### Reglas de entrada [Información](#)

Regla de entrada 1

[Eliminar](#)

Tipo [Información](#)

Protocolo [Información](#)

Intervalo de puertos [Información](#)

Tipo de origen [Información](#)

Origen [Información](#)

Descripción: opcional [Información](#)

Regla de entrada 2

[Eliminar](#)

Tipo [Información](#)

Protocolo [Información](#)

Intervalo de puertos [Información](#)

Tipo de origen [Información](#)

Origen [Información](#)

Descripción: opcional [Información](#)

Regla de entrada 3

[Eliminar](#)

Tipo [Información](#)

Protocolo [Información](#)

Intervalo de puertos [Información](#)

Tipo de origen [Información](#)

Origen [Información](#)

Descripción: opcional [Información](#)

## CREAR LA INSTANCIA EC2

Se creó la instancia EC2 wordpress-server usando la AMI Ubuntu Server 24.04 LTS y el tipo de instancia t3.micro. La instancia se configuró con el par de claves labsuser y el Security Group wordpress-aws-sg, quedando lista para la instalación del entorno WordPress.

### 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

wordpress-server

[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

Recientes

**Inicio rápido**

Amazon Linux  
aws

macOS  
Mac

Ubuntu  
ubuntu®

Windows  
Microsoft

Red Hat  
Red Hat

SUSE Linux  
SUSE

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

Apto para la capa gratuita ▼

#### Descripción

Ubuntu Server 24.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (<http://www.ubuntu.com/cloud/services>).

Canonical, Ubuntu, 24.04, amd64 noble image

Arquitectura

64 bits (x... ▼

ID de AMI

ami-0ecb62995f68bb549

Fecha de

publicación

2025-10-22

Nombre de

usuario ⓘ

ubuntu

Proveedor verificado

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

#### Tipo de instancia

t3.micro

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

Apto para la capa gratuita

☐ Todas las generaciones

[Comparar tipos de instancias](#)

Se aplican costos adicionales a las AMI con software preinstalado

### ▼ 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](#)

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

[Editar](#)

Red [Información](#)

vpc-0ee8ae8cb384cd2d7

Subred [Información](#)

Sin preferencias (subred predeterminada en cualquier zona de disponibilidad)

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

Grupos de seguridad comunes [Información](#)

Seleccionar grupos de seguridad

wordpress-aws-sg sg-0236eac5159e4fee7 ✕  
VPC: vpc-0ee8ae8cb384cd2d7

[Compare reglas de grupo de seguridad](#)

Los grupos de seguridad que agrega o elimine aquí se agregarán a todas las interfaces de red o se eliminarán de ellas.

## ▼ Configurar almacenamiento Información

Avanzado

1x 15 GiB gp3 Volumen raíz, 3000 IOPS, No cifrado

[Agregar un nuevo volumen](#)

La AMI seleccionada contiene volúmenes de almacén de instancias; sin embargo, la instancia no permite dichos volúmenes. Por lo que no se podrá obtener acceso a ninguno de estos volúmenes de la AMI desde la instancia

⌚ 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](#)

✓ **Correcto**  
El lanzamiento de la instancia se inició correctamente ([i-0da004440b9a0445d](#))

► [Registro de lanzamiento](#)

### Pasos siguientes

#### 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](#)

#### Conectarse a la instancia

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

[Conectarse a la instancia](#)

[Más información](#)

#### 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](#)

[Crear una nueva base de datos de RDS](#)

[Más información](#)

[Ver todas las instancias](#)

EC2 > Instancias

Instancias (1/2) Información

Última actualización  
Hace less than a minute

Conectar

Estado de la instancia

Acciones

Lanzar instancias

Buscar Instancia por atributo o etiqueta (case-sensitive)

Todos los estados

< 1 >

	Name	ID de la instancia	Estado de la i...	Tipo de inst...	Comprobación de	Estad
<input checked="" type="checkbox"/>	lor-web	i-0395ccacab2106060	En ejecución	t3.micro	3/3 comprobaci...	Ver al
<input type="checkbox"/>	wordpress-ser...	i-0da004440b9a0445d	En ejecución	t3.micro	Inicializando	Ver al

i-0395ccacab2106060 (mi-servidor-web)

Estado y alarmas

Monitoreo

Seguridad

Redes

Almacenamiento

Et

Resumen de instancia Información

ID de la instancia  
i-0395ccacab2106060

Dirección IPv6  
-

Tipo de nombre de anfitrión  
Nombre de IP: ip-172-31-77-26.ec2.internal

Dirección IPv4 pública  
44.200.23.90 | dirección abierta

Estado de la instancia  
En ejecución

Nombre DNS de IP privada (solo IPv4)  
ip-172-31-77-26.ec2.internal

Direcciones IPv4 privadas  
172.31.77.26

DNS público  
ec2-44-200-23-90.compute-1.amazonaws.com | dirección abierta



## CONEXIÓN SSH A LA INSTANCIA EC2

La captura muestra la conexión SSH realizada desde la máquina local hacia la instancia EC2 usando la clave labsuser.pem, confirmando el acceso remoto exitoso con el usuario ubuntu.

```
ubuntu@ip-172-31-77-26: ~
The key's randomart image is:
+--[ED25519 256]--+
  *#O*O.
  * B.B. E
  o B   o o
  = + + + .
  B + S
  O * = .
  = = O
  O O O
  .O+ .
+-----[SHA256]-----+
juan@A6Alumno21:/mnt/c/windows/System32$ ls -la ~/.ssh/wordpress-key*
-rw----- 1 juan juan 411 Dec  9 09:14 /home/juan/.ssh/wordpress-key
-rw-r--r-- 1 juan juan  96 Dec  9 09:14 /home/juan/.ssh/wordpress-key.pub
juan@A6Alumno21:/mnt/c/windows/System32$ chmod 400 ~/.ssh/wordpress-key
ls -la ~/.ssh/wordpress-key
-r----- 1 juan juan 411 Dec  9 09:14 /home/juan/.ssh/wordpress-key
juan@A6Alumno21:/mnt/c/windows/System32$ ls -la ~/.ssh/labsuser.pem
-r----- 1 juan juan 1678 Dec  4 14:08 /home/juan/.ssh/labsuser.pem
juan@A6Alumno21:/mnt/c/windows/System32$ cp /mnt/c/Users/Alumno.DESKTOP-DI5KTUG/Downloads/labsuser.pem ~/.ssh/
cp: cannot create regular file '/home/juan/.ssh/labsuser.pem': Permission denied
juan@A6Alumno21:/mnt/c/windows/System32$ ssh -i ~/.ssh/labsuser.pem ubuntu@44.200.23.90
The authenticity of host '44.200.23.90 (44.200.23.90)' can't be established.
ED25519 key fingerprint is SHA256:StsVNJqLs3UwM5+qby5XVp6M/SinR60qzhU5eEQFFkU.
This host key is known by the following other names/addresses:
  ~/.ssh/known_hosts:1: [hashed name]
  ~/.ssh/known_hosts:4: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '44.200.23.90' (ED25519) to the list of known hosts.
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-1017-aws x86_64)

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

System information as of Tue Dec  9 09:14:03 UTC 2025

System load:  0.0           Temperature:   -273.1 C
Usage of /:   35.7% of 6.71GB Processes:      122
Memory usage: 26%          Users logged in:  0
Swap usage:   0%           IPv4 address for ens5: 172.31.77.26

 * Ubuntu Pro delivers the most comprehensive open source security and
  compliance features.

  https://ubuntu.com/aws/pro

Expanded Security Maintenance for Applications is not enabled.

9 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

1 additional security update can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm

Last login: Thu Dec  4 13:09:19 2025 from 88.0.31.41
ubuntu@ip-172-31-77-26:~$
```

# INSTALACIÓN DE LAMP

Apache aparece como *active (running)*, verificando el correcto funcionamiento del servidor web.

```
ubuntu@ip-172-31-77-26:~$ 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 Tue 2025-12-09 09:19:26 UTC; 47s ago
     Docs: https://httpd.apache.org/docs/2.4/
  Main PID: 7606 (apache2)
    Tasks: 6 (limit: 1017)
   Memory: 15.2M (peak: 16.9M)
      CPU: 78ms
   CGroup: /system.slice/apache2.service
           └─7606 /usr/sbin/apache2 -k start
             └─7610 /usr/sbin/apache2 -k start
               └─7611 /usr/sbin/apache2 -k start
                 └─7612 /usr/sbin/apache2 -k start
                   └─7613 /usr/sbin/apache2 -k start
                     └─7614 /usr/sbin/apache2 -k start

Dec 09 09:19:26 ip-172-31-77-26 systemd[1]: Starting apache2.service - The Apache HTTP Server...
Dec 09 09:19:26 ip-172-31-77-26 systemd[1]: Started apache2.service - The Apache HTTP Server.
ubuntu@ip-172-31-77-26:~$
```

## CREACIÓN DEL SCRIPT DE WORDPRESS

En esta captura se evidencia la transferencia correcta del archivo `install-wordpress.sh` desde el equipo local hacia la instancia EC2 mediante el comando `scp`. Se observa el uso del par de claves `labsuser.pem` para la autenticación SSH y la confirmación de que el archivo fue copiado al directorio `HOME` del usuario `ubuntu` en la máquina remota, cumpliendo con los requisitos del entregable correspondiente.

```
Trace Maps.d11
juan@A6Alumno21:/mnt/c/Windows/System32$ scp -i ~/.ssh/labsuser.pem install-wordpress.sh ubuntu@44.200.23.98:~/
scp: stat local "install-wordpress.sh": No such file or directory
juan@A6Alumno21:/mnt/c/Windows/System32$ find ~ -type f -name "install-wordpress.sh"
juan@A6Alumno21:/mnt/c/Windows/System32$ cd ~
juan@A6Alumno21:~$ pwd
/home/juan
juan@A6Alumno21:~$ nano install-wordpress.sh
juan@A6Alumno21:~$ ls -la install-wordpress.sh
-rw-r--r-- 1 juan juan 2148 Dec  9 10:29 install-wordpress.sh
juan@A6Alumno21:~$ chmod +x install-wordpress.sh
juan@A6Alumno21:~$ scp -i ~/.ssh/labsuser.pem install-wordpress.sh ubuntu@44.200.23.98:~/
install-wordpress.sh
100% 2148 19.8KB/s 00:00
juan@A6Alumno21:~$
```

## Ejecutar el script en la instancia EC2

En esta captura se observa la ejecución del script automatizado install-wordpress.sh dentro de la instancia EC2. El script realiza la instalación de Apache, PHP, MySQL y WordPress, además de la creación de la base de datos, usuario y credenciales necesarias. La evidencia confirma que el entorno WordPress ha sido desplegado correctamente en el servidor remoto.

```
sent 76,403,928 bytes  received 65,890 bytes  21,848,519.43 bytes/sec
total size is 76,133,582  speedup is 1.00
sed: -e expression #1, char 20: unknown option to `s'
Guardando credenciales...
INSTALACIÓN COMPLETA
ubuntu@ip-172-31-77-26:~$
```

## Ver credenciales generadas por el script

En esta captura se muestran las credenciales generadas automáticamente por el script de instalación de WordPress, incluyendo el nombre de la base de datos, el usuario, la contraseña del usuario y la contraseña del usuario root de MySQL. Este archivo fue generado en el directorio /tmp/ y contiene la información necesaria para la administración de la base de datos del sitio WordPress.

```
/tmp/wordpress-credentials.txt
ubuntu@ip-172-31-77-26:~$ cat /tmp/wordpress-credentials.txt
Base de datos: wordpress
Usuario: wpuser
Password usuario: kr/ut4sGc7XBtM02
Password root: 5ymPRe1h1N+qqQql
ubuntu@ip-172-31-77-26:~$
```

# Probar WordPress desde la IP pública

En esta imagen se muestra la pantalla inicial del instalador de WordPress accesible desde la IP pública del servidor EC2.

Esto confirma que la instancia EC2, el servidor web Apache, la base de datos MySQL y la configuración del archivo wp-config.php se han realizado correctamente.

WordPress ya puede completar su instalación, lo que indica que la conexión a la base de datos es correcta y el despliegue del sitio está listo para continuar.

Hola

¡Este es el famoso proceso de instalación de WordPress en cinco minutos! Simplemente completa la información siguiente y estarás a punto de usar la más enriquecedora y potente plataforma de publicación personal del mundo.

### Información necesaria

Por favor, proporciona la siguiente información. No te preocupes, siempre podrás cambiar estos ajustes más tarde.

**Título del sitio**

**Nombre de usuario**   
Los nombres de usuario pueden tener únicamente caracteres alfanuméricos, espacios, guiones bajos, guiones medios, puntos y el símbolo @.

**Contraseña**  [Ocultar](#)  
**Fuerte**

**Importante:** Necesitas esta contraseña para acceder. Por favor, guárdala en un lugar seguro.

**Tu correo electrónico**   
Comprueba bien tu dirección de correo electrónico antes de continuar.

**Visibilidad en los motores de búsqueda** ☐ Pedir a los motores de búsqueda que no indexen este sitio  
Depende de los motores de búsqueda atender esta petición o no.

[Instalar WordPress](#)

En esta pantalla se introducen los datos necesarios para crear la cuenta de administrador de WordPress, incluyendo el título del sitio, usuario, contraseña y correo electrónico.

Este paso finaliza el proceso de despliegue del CMS en la instancia EC2.

Hola

Debes facilitar una dirección de correo electrónico

**Título del sitio**

**Nombre de usuario**   
Los nombres de usuario pueden tener únicamente caracteres alfanuméricos, espacios, guiones bajos, guiones medios, puntos y el símbolo @.

**Contraseña**  [Ocultar](#)  
Débil

**Importante:** Necesitas esta contraseña para acceder. Por favor, guárdala en un lugar seguro.

**Confirma la contraseña** ☒ Confirma el uso de una contraseña débil.

**Tu correo electrónico**   
Comprueba bien tu dirección de correo electrónico antes de continuar.

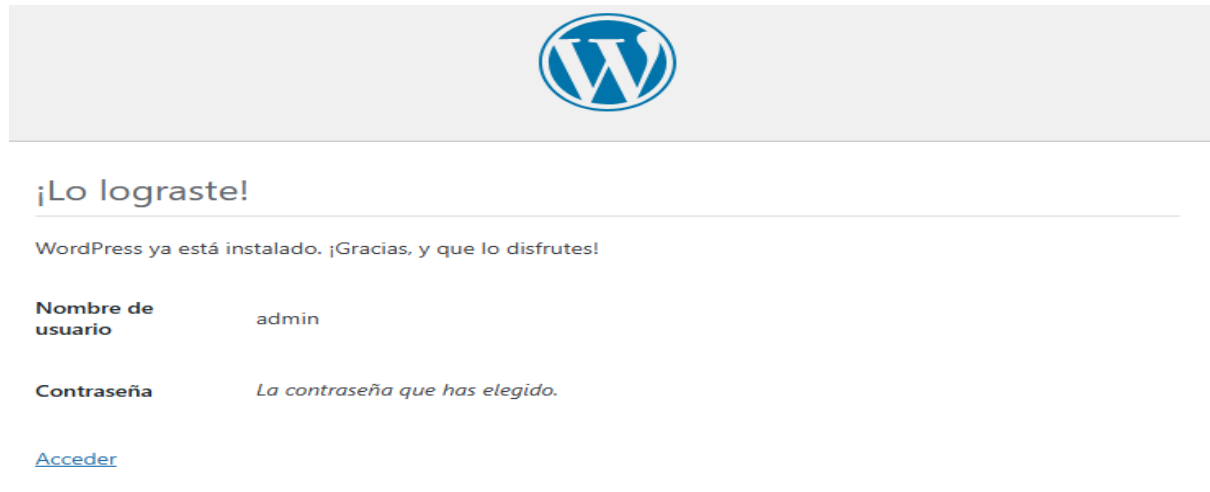
**Visibilidad en los motores de búsqueda** ☐ Pedir a los motores de búsqueda que no indexen este sitio  
Depende de los motores de búsqueda atender esta petición o no.

[Instalar WordPress](#)

En esta pantalla se confirma que WordPress ha sido instalado correctamente en la instancia EC2.

Se muestra el nombre del usuario administrador creado durante el proceso de instalación, lo que indica que la base de datos MySQL está funcionando, Apache sirve correctamente el sitio y la configuración del archivo wp-config.php es válida.

Desde este punto ya se puede acceder al panel de administración de WordPress mediante el enlace *Acceder*.



En esta captura se muestra el escritorio de WordPress ya funcionando sobre la instancia EC2.

Desde aquí es posible administrar entradas, páginas, temas, plugins y realizar toda la gestión del sitio web.

Esto confirma que el despliegue del CMS ha sido exitoso y plenamente operativo en el servidor en la nube.



Esta imagen muestra el front-end del sitio WordPress funcionando sobre la instancia EC2. La página utiliza el tema por defecto e indica que el servidor Apache responde correctamente y que la base de datos está accesible. Esto valida que todo el stack LAMP está operativo en la nube: Linux + Apache + MySQL + PHP.



Para comprobar el correcto funcionamiento del panel de administración, se modificó el título principal del sitio web desde *Ajustes* → *Generales*. Este cambio demuestra que WordPress puede gestionar configuraciones y guardar datos en la base de datos MySQL, validando la instalación.

A screenshot of the WordPress 'Ajustes generales' (General Settings) page. The page title is 'Ajustes generales'. It contains several form fields for site configuration: 'Título del sitio' (Site Title) with the value 'Mi WordPress en AWS', 'Descripción corta' (Short Description) which is empty, 'Icono del sitio' (Site Icon) with a button 'Elige un icono del sitio', 'Dirección de WordPress (URL)' (WordPress Address (URL)) with the value 'http://100.48.47.116', 'Dirección del sitio (URL)' (Site Address (URL)) with the value 'http://100.48.47.116', and 'Dirección de correo electrónico de administración' (Administrator Email Address) with the value 'juan.sebastian.medrano@students.thepower.education'. There are also explanatory text blocks for the description and URL fields.

# Verificación del servidor

Esta captura verifica que tanto Apache2 como MySQL están activos y funcionando en la instancia EC2.

Esto demuestra que el servidor está configurado correctamente para alojar WordPress.

```
ubuntu@ip-172-31-77-26: ~  
Memory: 38.3M (peak: 53.0M)  
CPU: 247ms  
CGroup: /system.slice/apache2.service  
├─582 /usr/sbin/apache2 -k start  
├─990 /usr/sbin/apache2 -k start  
├─991 /usr/sbin/apache2 -k start  
├─992 /usr/sbin/apache2 -k start  
├─993 /usr/sbin/apache2 -k start  
└─994 /usr/sbin/apache2 -k start  
Dec 10 07:53:26 ip-172-31-77-26 systemd[1]: Starting apache2.service - The Apache HTTP Server...  
Dec 10 07:53:26 ip-172-31-77-26 systemd[1]: Started apache2.service - The Apache HTTP Server.  
Dec 10 07:53:28 ip-172-31-77-26 systemd[1]: Reloading apache2.service - The Apache HTTP Server...  
Dec 10 07:53:28 ip-172-31-77-26 systemd[1]: Reloaded apache2.service - The Apache HTTP Server.  
ubuntu@ip-172-31-77-26:~$ sudo systemctl status mysql  
● mysql.service - MySQL Community Server  
   Loaded: loaded (/usr/lib/systemd/system/mysql.service; enabled; preset: enabled)  
   Active: active (running) since Wed 2025-12-10 07:53:29 UTC; 4min 5s ago  
 Process: 550 ExecStartPre=/usr/share/mysql/mysql-systemd-start pre (code=exited, status=0/SUCCESS)  
 Main PID: 811 (mysqld)  
   Status: "Server is operational"  
    Tasks: 37 (limit: 1017)  
  Memory: 398.9M (peak: 427.6M)  
     CPU: 2.606s  
   CGroup: /system.slice/mysql.service  
           └─811 /usr/sbin/mysqld  
Dec 10 07:53:26 ip-172-31-77-26 systemd[1]: Starting mysql.service - MySQL Community Server...  
Dec 10 07:53:29 ip-172-31-77-26 systemd[1]: Started mysql.service - MySQL Community Server.  
ubuntu@ip-172-31-77-26:~$ sudo nano /var/www/html/wp-config.php  
ubuntu@ip-172-31-77-26:~$ sudo nano /var/www/html/wp-config.php  
ubuntu@ip-172-31-77-26:~$ sudo systemctl restart apache2  
Warning: The unit file, source configuration file or drop-ins of apache2.service changed on disk. Run 'systemctl daemon-reload' to reload units.  
ubuntu@ip-172-31-77-26:~$ sudo systemctl status apache2  
Warning: The unit file, source configuration file or drop-ins of apache2.service changed on disk. Run 'systemctl daemon-reload' to reload units.  
● apache2.service - The Apache HTTP Server  
   Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: enabled)  
   Active: active (running) since Wed 2025-12-10 08:00:03 UTC; 13min ago  
    Docs: https://httpd.apache.org/docs/2.4/  
 Process: 1550 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)  
 Main PID: 1554 (apache2)  
    Tasks: 11 (limit: 1017)  
  Memory: 111.2M (peak: 117.4M)  
     CPU: 2.888s  
   CGroup: /system.slice/apache2.service  
           ├─1554 /usr/sbin/apache2 -k start  
           ├─1558 /usr/sbin/apache2 -k start  
           ├─1559 /usr/sbin/apache2 -k start  
           ├─1560 /usr/sbin/apache2 -k start  
           ├─1565 /usr/sbin/apache2 -k start  
           ├─1566 /usr/sbin/apache2 -k start  
           ├─1567 /usr/sbin/apache2 -k start  
           ├─1568 /usr/sbin/apache2 -k start  
           ├─1571 /usr/sbin/apache2 -k start  
           ├─1599 /usr/sbin/apache2 -k start  
           └─1661 /usr/sbin/apache2 -k start  
Dec 10 08:00:02 ip-172-31-77-26 systemd[1]: Starting apache2.service - The Apache HTTP Server...  
Dec 10 08:00:03 ip-172-31-77-26 systemd[1]: Started apache2.service - The Apache HTTP Server.  
lines 1-25/25 (END)
```

Se comprueba el estado del firewall UFW en la instancia.

Aunque esté desactivado por defecto, esta verificación forma parte de la comprobación general de seguridad del sistema.

```
ubuntu@ip-172-31-77-26:~$ sudo ufw status  
Status: inactive  
ubuntu@ip-172-31-77-26:~$
```



# Conclusión

En esta práctica se ha desplegado un servidor web completo en AWS utilizando una instancia EC2 con Ubuntu.

Se instalaron y configuraron los componentes del stack LAMP (Linux, Apache, MySQL y PHP), se creó y configuró la base de datos, y finalmente se instaló el CMS WordPress, verificando su correcto funcionamiento tanto en el panel de administración como en el sitio web público.

El resultado final demuestra la capacidad de desplegar aplicaciones web dinámicas en la nube, entendiendo aspectos esenciales como seguridad, servicios, puertos, credenciales y administración del servidor.