

INSTALACIÓN DE WORDPRESS EN AWS CON SSH

Hamza Akdi

23/11/2025

Índice:

Objetivos:.....	3
Requisitos Previos.....	3
1. Preparación del entorno local.....	4
1.1 Crear directorio SSH y permisos	5
1.2 Generar clave SSH con Ed25519.....	5
1.3 Ajuste de permisos de la clave privada	5
2. Configuración en AWS.....	6
2.1 Creación de par de claves en AWS	6
2.2 Trasferencia de la clave descargada a WSL.....	6
2.3 – Creación de Security Group	7
2.4 Creación de instancia EC2	7
3. Conexión SSH desde WSL a AWS.....	9
3.1 Conectar a la instancia	9
4. Instalación base de un servidor en AWS	9
4.1 Actualización del sistema	9
4.2 Instalación de LAMP Stack	9
4.3 Inicio de servicios y verificación de estado	10
5. Script de automatización de Wordpress	10
5.1 Creación de script de instalación.....	10
6. Migración de archivos con SCP.....	13
6.1 Transferencia de Script a AWS	13
6.2 Permisos y ejecución del script	13
7. Verificación de instalación	14
7.1 Verificación de servicios	14
8. WordPress accesible desde internet con Ngrok	15
8.1 Instalación de Ngrok en AWS.....	15
8.2 Autentificación de Ngrok.....	16
8.3 Iniciación de Ngrok	16
8.4 – Actualizar WordPress	17
8.5-Probar acceso remoto	18

Objetivos:

- Generar claves SSH con Ed25519/RSA usando keygen
- Configurar Security Groups en AWS
- Conectarte a una instancia EC2 mediante SSH
- Migrar archivos de WordPress usando SCP
- Automatizar la instalación y configuración de WordPress con un script
- Acceder a WordPress desde Internet usando ngrok

Requisitos Previos

- Cuenta en AWS (Free Tier)
- WSL2 o máquina virtual con Ubuntu 22.04 o superior
- Acceso a Internet
- Terminal/PowerShell en Windows o Terminal en Linux/Mac
- Cuenta en ngrok.com

1. Preparación del entorno local

1.1 Crear directorio SSH y permisos

Se creará un directorio oculto llamado **.ssh** en nuestro directorio de usuario si no existe, y se le asignan permisos restrictivos (solo el propietario puede leer, escribir y ejecutar).

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

1.2 Generar clave SSH con Ed25519

Se genera un par de claves SSH (pública y privada) que se utilizará en AWS. **Ed25519** es el tipo de clave recomendado por ser más moderno y seguro

```
hamza@A6Alumno08:~$ ssh-keygen -t ed25519 -f ~/.ssh/wordpress-key -C "hamza@18.212.206.22"  
Generating public/private ed25519 key pair.  
Enter passphrase (empty for no passphrase):  
Enter same passphrase again:  
Your identification has been saved in /home/hamza/.ssh/wordpress-key  
Your public key has been saved in /home/hamza/.ssh/wordpress-key.pub  
The key fingerprint is:  
SHA256:Le6WufGqKD/hpivJI941VnxL+AmRNJ860mHiWSkg4Jc hamza@18.212.206.22  
The key's randomart image is:  
+--[ED25519 256]--  
|o. . o |  
|. . o . = . |  
. E o B o |  
. . O =. |  
+ OSo. |  
. o.*.o |  
. . = o* |  
+oo *...+o |  
|oo+Bo..o+o. |  
+---[SHA256]---
```

Verificación de creación correcta:

```
hamza@A6Alumno08:~$ ls -la ~/.ssh/wordpress-key*  
-rw----- 1 hamza hamza 464 Nov 26 08:59 /home/hamza/.ssh/wordpress-key  
-rw-r--r-- 1 hamza hamza 101 Nov 26 08:59 /home/hamza/.ssh/wordpress-key.pub
```

1.3 Ajuste de permisos de la clave privada

La clave privada debe tener permisos muy restringidos (solo lectura para el propietario) para que SSH la acepte por seguridad.

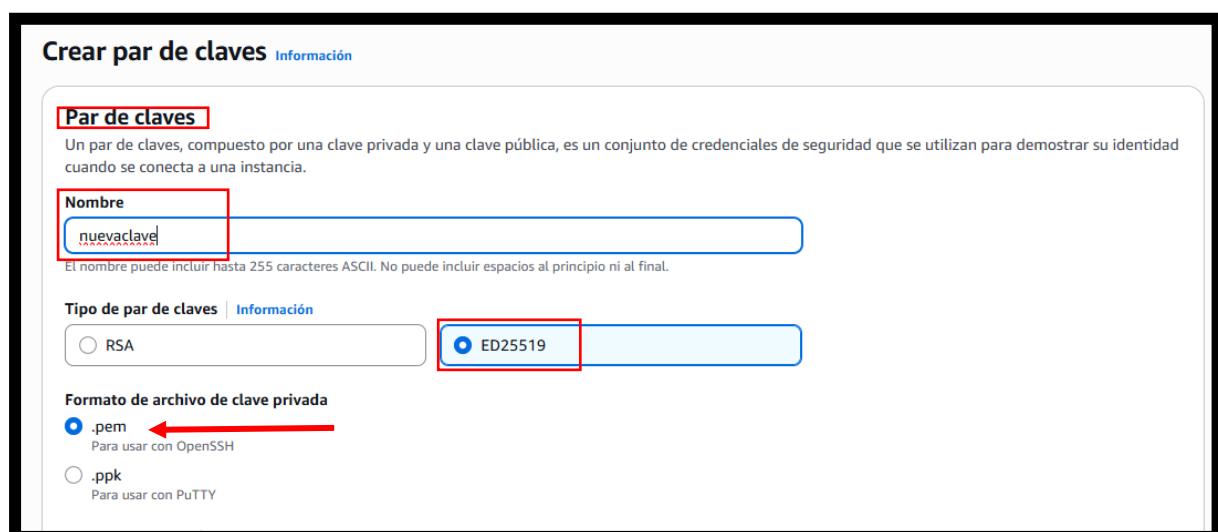
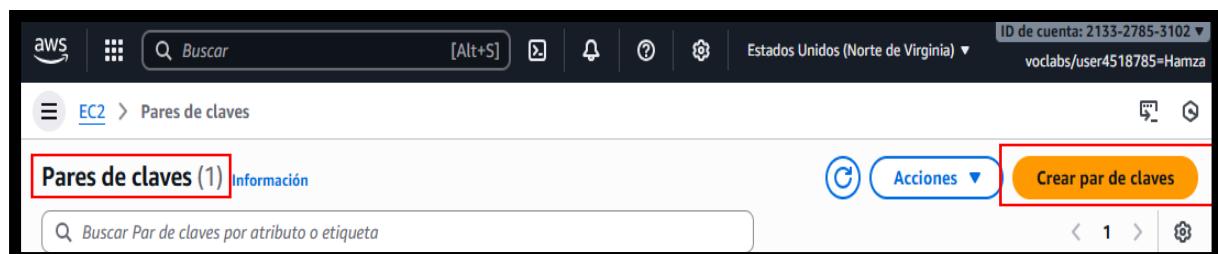
```
hamza@A6Alumno08:~$ chmod 400 ~/.ssh/wordpress-key
hamza@A6Alumno08:~$ ls -la ~/.ssh/wordpress-key
-r----- 1 hamza hamza 464 Nov 26 08:59 /home/hamza/.ssh/wordpress-key
```

2. Configuración en AWS

Esta parte se centra en la consola de Amazon Web Services (AWS).

2.1 Creación de par de claves en AWS

Crearemos un par de claves en la consola de AWS que se asociará a la instancia EC2.



2.2 Trasferencia de la clave descargada a WSL

Se descargó desde nuestro equipo (Windows), hay que moverla al directorio .ssh de nuestro entorno local (WSL) y ajustar sus permisos.

```
hamza@A6Alumno08:~$ cp /mnt/c/nuevaclave.pem ~/.ssh/
hamza@A6Alumno08:~$ chmod 400 ~/.ssh/nuevaclave.pem
```

Verificación de clave y permisos:

```
hamza@A6Alumno08:~$ ls -la ~/.ssh/nuevaclave.pem
-r----- 1 hamza hamza 387 Nov 26 10:03 /home/hamza/.ssh/nuevaclave.pem
```

2.3 – Creación de Security Group

Añadiremos las siguientes reglas:

- Regla1: SSH (Puerto 22)
- Regla2: HTTP (Puerto 80)
- Regla3: HTTPS (Puerto43)

The screenshot shows the AWS Security Groups console for the security group `sg-0d6b7821e2eb0ca29 - launch-wizard-1`. The 'Reglas de entrada' (Ingress Rules) tab is selected, displaying three rules:

Name	ID de la regla del grupo	Versión de IP	Tipo	Protocolo	Intervalo de puertos
-	sgr-0053e0d14f4030e29	IPv4	SSH	TCP	22
-	sgr-008f60616e7ddeaac	IPv4	HTTP	TCP	80
-	sgr-0b144d9fe2c42323f	IPv4	HTTPS	TCP	443

Each row in the table has a red box around it, highlighting the rule types and port numbers.

2.4 Creación de instancia EC2

1. Accede a EC2 → Instances → Launch instances
2. Nombre de instancia: wordpress-server
3. AMI: Ubuntu 24.04 LTS
4. Instance type: t3.micro (Free Tier)
5. Key pair: nuevaclave
6. VPC settings: Default
7. Security group: Selecciona sg-wordpress-aws
8. Storage: 15 GiB, gp3

Lanzar una instancia Información

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

Nombre y etiquetas Información

Nombre

wordpress-server

Agregar

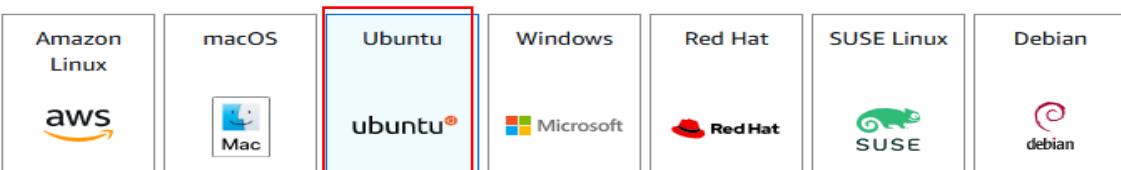
▼ Imágenes de aplicaciones y sistemas operativos (Imagen de máquina de Amazon)

Una AMI posee el sistema operativo, el servidor de aplicaciones y las aplicaciones de la instancia. Si a continuación 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



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

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*

nuevaclave

[Crear un nuevo par de claves](#)

▼ Configuraciones de red Información

[Editar](#)

Red | [Información](#)

vpc-006a759b8d44f0a53

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](#)

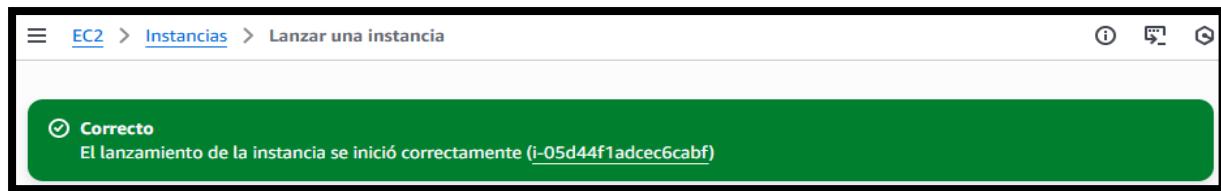
Seleccionar grupos de seguridad

launch-wizard-1 sg-0d6b7821e2eb0ca29

VPC: vpc-006a759b8d44f0a53

[Compare reglas de grupo de seguridad](#)

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



3. Conexión SSH desde WSL a AWS

3.1 Conectar a la instancia

Para conectarnos a nuestra instancia creada, usaremos el comando ssh con la clave privada .pem y la IP pública obtenida.

```
hamza@A6Alumno08:~$ ssh -i ~/ssh/nuevaClave.pem ubuntu@3.238.124.253
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 08:11:27 UTC 2025

 System load: 0.01          Temperature:      -273.1 °C
 Usage of /:   26.5% of 6.71GB Processes:        115
 Memory usage: 24%          Users logged in:    0
 Swap usage:   0%           IPv4 address for ens5: 172.31.64.241
```

4. Instalación base de un servidor en AWS

Todos los comandos de esta parte se ejecutan dentro de la sesión SSH en la instancia AWS.

4.1 Actualización del sistema

Se actualizan las listas de paquetes y se actualizan los paquetes instalados.

```
ubuntu@ip-172-31-64-241:~$ 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://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [301 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [269 kB]
```

4.2 Instalación de LAMP Stack

Instalaremos el conjunto de software necesario para WordPress: Apache (servidor web), PHP y sus extensiones necesarias, y MySQL (servidor de base de datos).

```
ubuntu@ip-172-31-64-241:~$ sudo apt install apache2 php php-mysql libapache2-mod-php php-curl php-gd php-mbstring php-xml php-xmlrpc php-intl php-zip mysql-server -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 fontconfig-config fonts-dejavu-core fonts-dejavu-mono
  libaom3 libapache2-mod-php8.3 libapr1t64 libaprutil1-dbd-sqlite3 libaprutil1-ldap libaprutil1t64
  libcgi-fast-perl libcgi-pm-perl libclone-perl libde265-0 libdeflate0 libencode-locale-perl
  libevent-pthreads-2.1-7t64 libfcgi-bin libfcgi-perl libfcgi0t64 libfontconfig1 libgd3
  libheif-plugin-aomdec libheif-plugin-aomenc libheif-plugin-libde265 libheif1 libhtml-parser-perl
  libhtml-tagset-perl libhtml-template-perl libhttp-date-perl libhttp-message-perl libio-html-perl
```

4.3 Inicio de servicios y verificación de estado

Iniciaremos los servicios de Apache y MySQL, y se configuran para que arranquen automáticamente al iniciar el sistema.

```
ubuntu@ip-172-31-64-241:~$ sudo systemctl start apache2
sudo systemctl start mysql
sudo systemctl enable apache2
sudo systemctl enable mysql
Synchronizing state of apache2.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable apache2
Synchronizing state of mysql.service with SysV service script with /usr/lib/systemd/systemd-sysv-inst
```

Después, comprobaremos el estado del servidor web Apache. Verificando que está activo y disponible.

```
ubuntu@ip-172-31-64-241:~$ 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 08:21:54 UTC; 2min 12s ago
    Docs: https://httpd.apache.org/docs/2.4/
 Main PID: 23433 (Apache2)
   Tasks: 6 (limit: 1017)
  Memory: 14.7M (peak: 16.6M)
    CPU: 73ms
   CGroup: /system.slice/apache2.service
           ├─23433 /usr/sbin/apache2 -k start
           ├─23438 /usr/sbin/apache2 -k start
           ├─23439 /usr/sbin/apache2 -k start
           ├─23440 /usr/sbin/apache2 -k start
           ├─23441 /usr/sbin/apache2 -k start
           └─23442 /usr/sbin/apache2 -k start
```

5. Script de automatización de Wordpress

Esta parte se realizará en nuestra **máquina local (WSL)**, donde se creará un script para automatizar la configuración de la base de datos e instalación de WordPress.

5.1 Creación de script de instalación

En nuestra máquina local, crearemos un archivo llamado install-wordpress.sh con el siguiente contenido:

Variables

```
GNU nano 7.2                               install-wordpress.sh *
#!/bin/bash
set -e
echo "==== Iniciando instalación automatizada de WordPress ==="

# Variables
DB_NAME="wordpress"
DB_USER="wpuser"
DB_PASSWORD=$(openssl rand -base64 12)
DB_ROOT_PASSWORD=$(openssl rand -base64 12)
WP_HOME="http://localhost"
WP_SITEURL="http://localhost"
```

Configuración para MySQL

```
# Paso 1: Configurar MySQL
echo "Configurando MySQL..."
sudo mysql -e "ALTER USER 'root'@'localhost' IDENTIFIED BY
'${DB_ROOT_PASSWORD}';"
sudo mysql -e "DELETE FROM mysql.user WHERE User='';"
sudo mysql -e "DELETE FROM mysql.user WHERE User='root' AND Host NOT IN
('localhost', '127.0.0.1', '::1');"
sudo mysql -e "DROP DATABASE IF EXISTS test;"
sudo mysql -e "DELETE FROM mysql.db WHERE Db='test' OR Db='test\\_%';"
sudo mysql -e "FLUSH PRIVILEGES;"
```

Creación de base de datos y usuario de Wordpress

```
# Paso 2: Crear base de datos y usuario de WordPress
echo "Creando base de datos y usuario..."
sudo mysql -u root -p"${DB_ROOT_PASSWORD}" -e "CREATE DATABASE
${DB_NAME};"
sudo mysql -u root -p"${DB_ROOT_PASSWORD}" -e "CREATE USER
'${DB_USER}'@'localhost' IDENTIFIED BY '${DB_PASSWORD}';"
sudo mysql -u root -p"${DB_ROOT_PASSWORD}" -e "GRANT ALL PRIVILEGES ON
${DB_NAME}.* TO '${DB_USER}'@'localhost';"
sudo mysql -u root -p"${DB_ROOT_PASSWORD}" -e "FLUSH PRIVILEGES;"
```

Descarga e instalación de Wordpress

```
# Paso 3: Descargar WordPress
echo "Descargando WordPress..."
cd /tmp
wget https://wordpress.org/latest.tar.gz -q
tar -xzf latest.tar.gz

# Paso 4: Instalar WordPress
echo "Copiando archivos a /var/www/html..."
sudo rm -rf /var/www/html/*
sudo cp -r wordpress/* /var/www/html/
```

Configuración wp-config.php

```
# Paso 5: Configurar wp-config.php
echo "Configurando wp-config.php..."
sudo cp /var/www/html/wp-config-sample.php /var/www/html/wp-config.php
sudo sed -i "s/database_name_here/${DB_NAME}/g" /var/www/html/wpconfig.php
sudo sed -i "s/username_here/${DB_USER}/g" /var/www/html/wp-config.php
sudo sed -i "s/password_here/${DB_PASSWORD}/g" /var/www/html/wpconfig.php
```

Permisos

```
# Paso 6: Permisos
echo "Configurando permisos..."
sudo chown -R www-data:www-data /var/www/html/
sudo chmod -R 755 /var/www/html/
```

Habilitación mod_rewrite en Apache

```
# Paso 7: Habilitar mod_rewrite en Apache
echo "Habilitando mod_rewrite..."
sudo a2enmod rewrite
sudo systemctl restart apache2
```

Guardado de credenciales y final del script de instalación

```
#Paso 8: Guardar credenciales
echo "Guardando credenciales en archivo..."
cat > ~/wordpress-credentials.txt << EOF
==== CREDENCIALES DE WORDPRESS ====
Base de datos: ${DB_NAME}
Usuario BD: ${DB_USER}
Contraseña BD: ${DB_PASSWORD}
Usuario root MySQL: root
Contraseña root MySQL: ${DB_ROOT_PASSWORD}
Acceso local: http://localhost
Acceso remoto: (se configurará con ngrok)
EOF
echo "==== Instalación completada ==="
echo "Credenciales guardadas en ~/wordpress-credentials.txt"
echo "Accede a http://3.238.124.253 para finalizar la instalación de WordPress"
```

6. Migración de archivos con SCP

Ahora transferiremos el script de la máquina local al servidor AWS usando SCP (Secure Copy Protocol).

6.1 Transferencia de Script a AWS

Desde nuestra máquina local ejecutaremos el siguiente comando para iniciar la transferencia:

```
hamza@A6Alumno08:~$ scp -i ~/.ssh/nuevaclave.pem install-wordpress.sh ubuntu@3.238.124.253:~
/
install-wordpress.sh
656   23.5KB/s  00:00                                         100% 2
```

6.2 Permisos y ejecución del script

Volveremos a la **sesión SSH de AWS** y haremos que el script sea ejecutable. Una vez dados los permisos, ejecutaremos el script.

```
ubuntu@ip-172-31-64-241:~$ chmod +x ~/install-wordpress.sh
ubuntu@ip-172-31-64-241:~$ ./install-wordpress.sh
== Iniciando instalación automatizada de WordPress ==
Configurando MySQL...
Creando base de datos y usuario...
mysql: [Warning] Using a password on the command line interface can be insecure.
mysql: [Warning] Using a password on the command line interface can be insecure.
mysql: [Warning] Using a password on the command line interface can be insecure.
mysql: [Warning] Using a password on the command line interface can be insecure.
Descargando WordPress...
Copiando archivos a /var/www/html...
Configurando wp-config.php...
Configurando permisos...
Habilitando mod_rewrite...
Enabling module rewrite.
To activate the new configuration, you need to run:
  systemctl restart apache2
Guardando credenciales en archivo...
== Instalación completada ==
Credenciales guardadas en ~/wordpress-credentials.txt
Accede a http://3.238.124.253 para finalizar la instalación de WordPress
```

Una vez finalizado el script, verificaremos las credenciales. Guardaremos el contenido de este archivo en un lugar seguro, ya que contiene las contraseñas generadas.

```
ubuntu@ip-172-31-64-241:~$ cat ~/wordpress-credentials.txt
== CREDENCIALES DE WORDPRESS ==
Base de datos: wordpress
Usuario BD: wpuser
Contraseña BD: YmM79NMuErKA+GQ0
Usuario root MySQL: root
Contraseña root MySQL: Sr5TatoGA/dqXM0R
Acceso local: http://localhost
Acceso remoto: (se configurará con ngrok)
```

7. Verificación de instalación

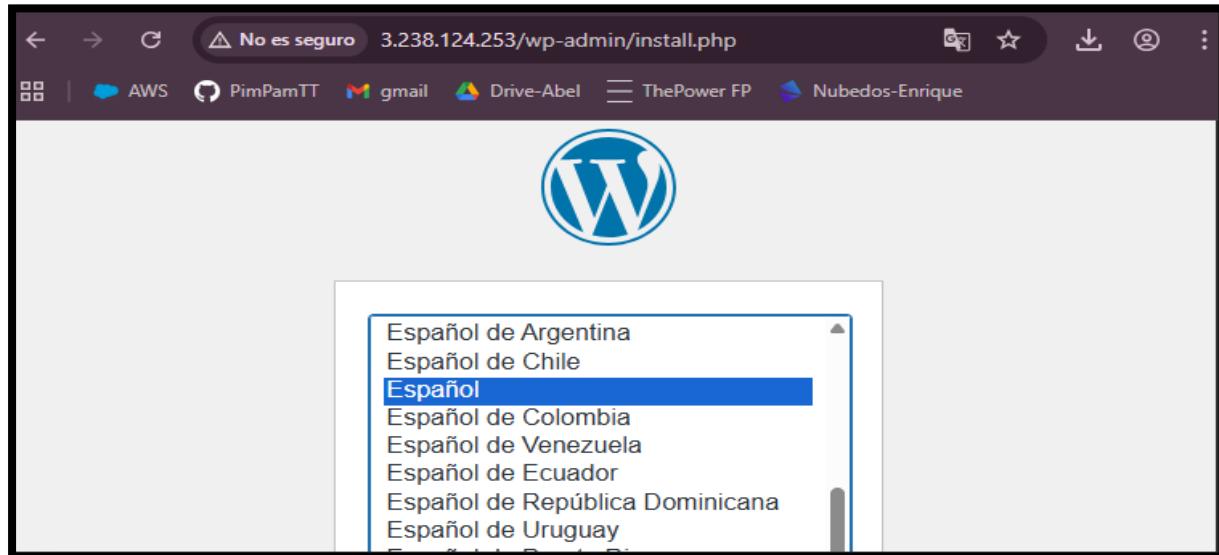
7.1 Verificación de servicios

Verificaremos que Apache y MySQL sigan activos después de la ejecución del script.

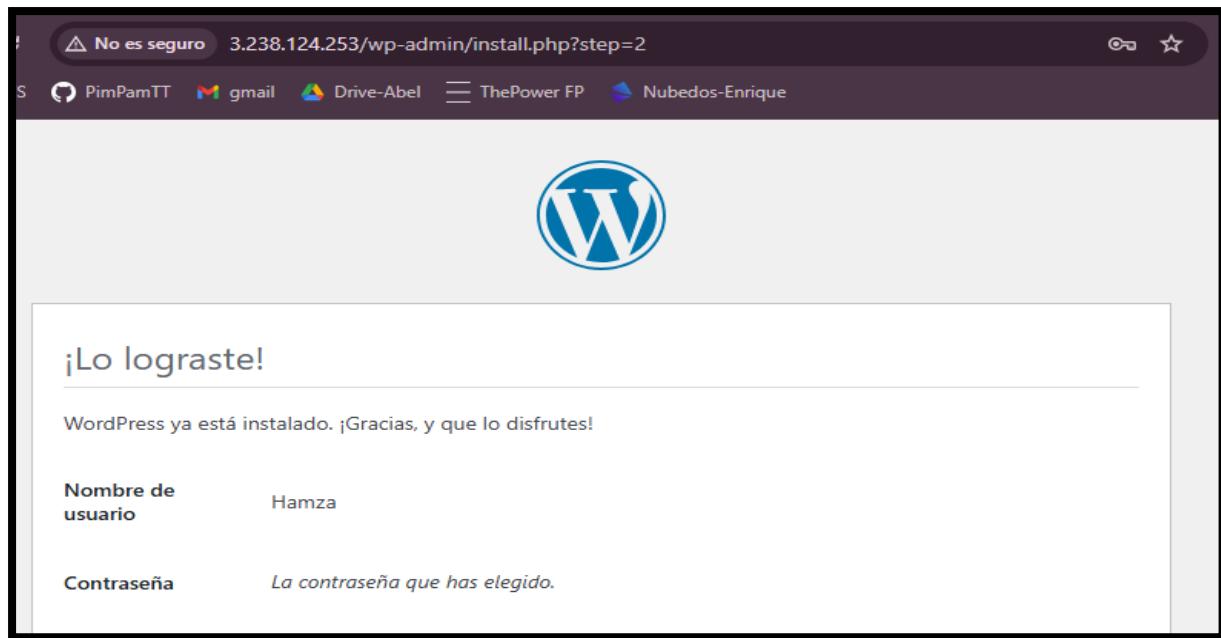
```
ubuntu@ip-172-31-64-241:~$ sudo systemctl status apache2
sudo systemctl status mysql
● apache2.service - The Apache HTTP Server
  Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: enabled)
  Active: active (running) since Fri 2025-11-28 08:57:45 UTC; 40min ago
    Docs: https://httpd.apache.org/docs/2.4/
 Process: 24160 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
 Main PID: 24165 (apache2)
   Tasks: 11 (limit: 1017)
  Memory: 63.2M (peak: 63.5M)
     CPU: 894ms
```

```
● mysql.service - MySQL Community Server
  Loaded: loaded (/usr/lib/systemd/system/mysql.service; enabled; preset: enabled)
  Active: active (running) since Fri 2025-11-28 08:21:38 UTC; 1h 16min ago
 Main PID: 19208 (mysqld)
   Status: "Server is operational"
    Tasks: 38 (limit: 1017)
   Memory: 358.2M (peak: 378.9M)
      CPU: 32.568s
     CGroup: /system.slice/mysql.service
             └─19208 /usr/sbin/mysqld
```

Una vez finalice el script, podremos finalizar la instalación de WordPress accediendo a la IP publica de nuestra instancia en AWS.



Completaremos la instalación de WordPress (idioma, usuario, contraseña, título del sitio).



8. WordPress accesible desde internet con Ngrok

Ngrok es una herramienta que crea un túnel seguro desde un puerto local (puerto 80 de nuestro servidor) hacia una URL pública de Internet, útil para pruebas o desarrollo.

8.1 Instalación de Ngrok en AWS

Descargaremos, extraeremos y moveros el ejecutable de Ngrok.

```
ubuntu@ip-172-31-64-241:~$ cd ~
wget https://bin.equinox.io/c/bNyj1mQVY4c/ngrok-v3-stable-linux-amd64.tgz
tar -xvzf ngrok-v3-stable-linux-amd64.tgz
sudo mv ngrok /usr/local/bin/
--2025-11-28 09:46:20-- https://bin.equinox.io/c/bNyj1mQVY4c/ngrok-v3-stable-linux-amd64.tgz
Resolving bin.equinox.io (bin.equinox.io)... 75.2.60.68, 35.71.179.82, 99.83.220.108, ...
Connecting to bin.equinox.io (bin.equinox.io)|75.2.60.68|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 10980950 (10M) [application/octet-stream]
Saving to: ‘ngrok-v3-stable-linux-amd64.tgz’

ngrok-v3-stable-linux- 100%[=====] 10.47M --.-KB/s   in 0.04s
2025-11-28 09:46:20 (234 MB/s) - ‘ngrok-v3-stable-linux-amd64.tgz’ saved [10980950/10980950]
ngrok
```

8.2 Autentificación de Ngrok

Necesitaremos una cuenta en ngrok.com para obtener un token de autenticación. Para ello iremos a registrarnos a <https://ngrok.com> y obtendremos el token necesario.

Una vez obtenido el token, usaremos el siguiente comando con el token obtenido:

```
ubuntu@ip-172-31-64-241:~$ ngrok config add-authtoken 366064jyzdlnBVIb1BZCr0hkhVa_2cPw2oDRJr  
b7cCA1mmKa6  
Authtoken saved to configuration file: /home/ubuntu/.config/ngrok/ngrok.yml
```

8.3 Iniciación de Ngrok

A continuación, iniciaremos el túnel que redirige el tráfico de ngrok a Apache (que escucha en el puerto 80). Ngrok comenzará a ejecutarse y mostrará el estado.

Coparemos la URL HTTPS que aparece en la línea Forwarding (<https://unnauseating-clemencia-overt.ngrok-free.dev>).

Para acceder, utilizaremos el siguiente comando:

```
ubuntu@ip-172-31-64-241:~$ ngrok http 80
```

```
ngrok                                     (Ctrl+C to quit)

Call internal services from your gateway: https://ngrok.com/r/http-request

Session Status: online
Account: hamza (Plan: Free)
Version: 3.33.1
Region: United States (us)
Web Interface: http://127.0.0.1:4040
Forwarding: https://unnauseating-clemencia-overt.ngrok-free.dev -> http://

Connections: ttl     opn      rt1     rt5     p50     p90
              0        0       0.00    0.00    0.00    0.00
```

The screenshot shows a browser window with the URL `unnauseating-clemencia-overt.ngrok-free.dev`. The page title is "Prueba WordPress". Below the title, there is a terminal window titled "ubuntu@ip-172-31-64-241: ~". The terminal displays the output of the `ngrok` command. It shows session details like account (hamza), version (3.33.1), region (United States (us)), latency (11ms), and web interface (http://127.0.0.1:4040). It also lists connections with their respective TTL, open connections, and performance metrics (rt1, rt5, p50, p90). The "HTTP Requests" section is highlighted with a red box and shows two entries: a 404 Not Found for /favicon.ico at 10:02:27.316 UTC and a 200 OK for / at 10:02:26.995 UTC.

```
Session Status      online
Account            hamza (Plan: Free)
Version            3.33.1
Region             United States (us)
Latency            11ms
Web Interface     http://127.0.0.1:4040
Forwarding         https://unnauseating-clemencia-overt.ngrok-free.dev -> http://127.0.0.1:4040

Connections        ttl     opn     rt1     rt5     p50     p90
                   1       0      0.01   0.00   5.33   5.33

HTTP Requests
-----
10:02:27.316 UTC GET /favicon.ico          404 Not Found
10:02:26.995 UTC GET /                      200 OK
```

8.4 Actualización de WordPress

WordPress almacena las URLs de su sitio en la base de datos. Para que funcione correctamente con la URL de ngrok, debemos actualizar estas URLs. **Esto se hace en otra terminal SSH, dejando la sesión de ngrok corriendo.**

Para ello iremos a otra terminal y nos conectaremos a nuestra instancia en AWS.

Accederemos a la base de datos de WordPress (usa el usuario y contraseña del archivo `wordpress-credentials.txt`):

The screenshot shows a terminal window on an Ubuntu instance. The user has run the command `mysql -u wpuser -p -D wordpress`. They are prompted to enter the password. After entering it, they see the MySQL monitor welcome message, which includes the connection ID (70), server version (8.0.44-0ubuntu0.24.04.1 (Ubuntu)), and copyright information from Oracle. At the bottom, there is a note about Oracle trademarks and a prompt for help or clearing the input statement.

```
ubuntu@ip-172-31-64-241:~$ mysql -u wpuser -p -D wordpress
Enter password:
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 70
Server version: 8.0.44-0ubuntu0.24.04.1 (Ubuntu)

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

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

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

Ejecutaremos los comandos SQL:

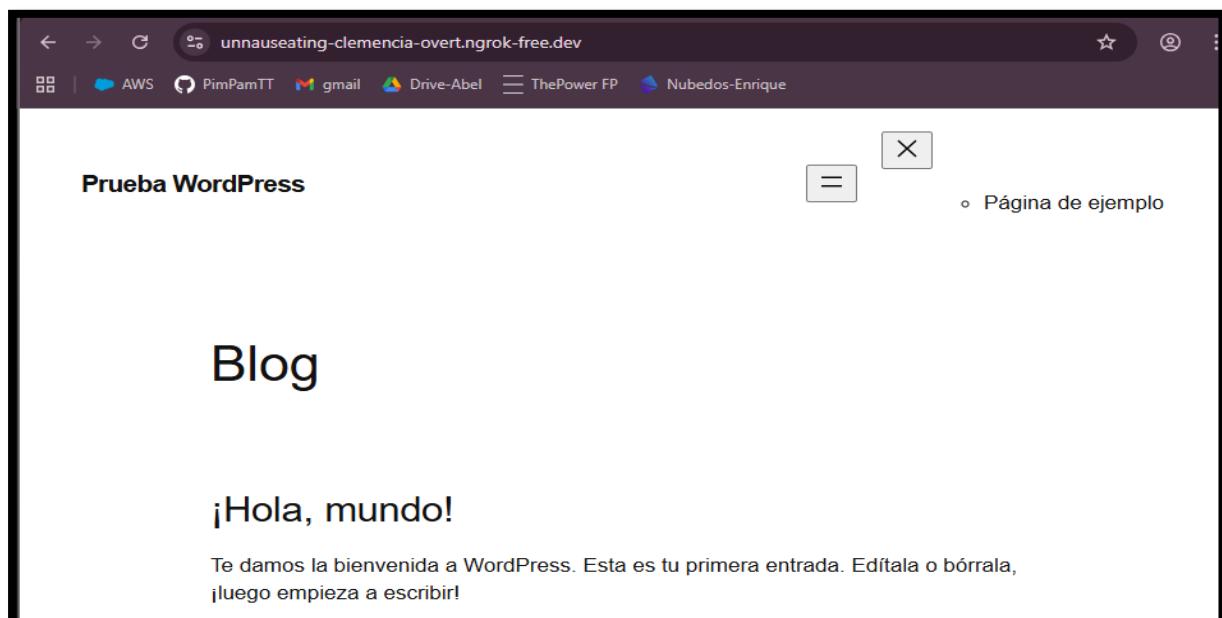
```
UPDATE wp_options SET option_value='https://unnauseating-clemencia-overt.ngrok-free.dev/' WHERE option_name='siteurl';
```

```
UPDATE wp_options SET option_value='https://unnauseating-clemencia-overt.ngrok-free.dev/' WHERE option_name='home';
```

```
mysql> UPDATE wp_options SET option_value='https://unnauseating-clemencia-overt.ngrok-free.dev/' WHERE option_name='home';
Query OK, 1 row affected (0.08 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

8.5-Probar acceso remoto

Abriremos el navegador y accederemos a la URL de ngrok para verificar que el sitio es accesible desde cualquier lugar.



```
ngrok                                     (Ctrl+C to quit)

• Call internal services from your gateway: https://ngrok.com/r/http-request

Session Status      online
Account            hamza (Plan: Free)
Version             3.33.1
Region              United States (us)
Latency             12ms
Web Interface       http://127.0.0.1:4040
Forwarding          https://unnauseating-clemencia-overt.ngrok-free.dev -> http://

Connections        ttl     opn         rt1        rt5        p50        p90
                   1       0       0.01      0.00      5.18      5.18

HTTP Requests
-----
11:33:35.977 UTC GET /
200 OK
```