

The background is a solid blue-to-purple gradient. It features a repeating pattern of white line-art icons. These icons include clouds, server racks, and network diagrams with nodes and connecting lines, all rendered in a light, semi-transparent style.

Adrián Aceitón Palomo

2º ASIR

Arquitectura
en la Nube

PRÁCTICA: INSTALACIÓN DE WORDPRESS EN AWS CON SSH

Objetivos

Al finalizar esta práctica, serás capaz de:

- 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
-

PARTE 0: PREPARACIÓN DEL ENTORNO LOCAL

0.1- Crear directorio SSH

```
mkdir -p ~/.ssh
```

```
chmod 700 ~/.ssh
```

0.2- Generar clave SSH con Ed2551G

Ejecuta en tu máquina local (WSL/VM):

```
ssh-keygen -t ed25519 -f ~/.ssh/wordpress-key -C "mi-usuario@aws"
```

Cuando te pida contraseña, presiona Enter (*o configura una si lo prefieres*).

Aquí básicamente he creado el directorio ssh y he generado la clave ssh con Ed2551G. En mi caso si que he puesto contraseña: adrian12345.

```

adrian@A6Alumno01: ~
adrian@A6Alumno01:~$ mkdir -p ~/.ssh
chmod 700 ~/.ssh
adrian@A6Alumno01:~$ chmod 700 ~/.ssh
adrian@A6Alumno01:~$ ls -ld ~/.ssh
drwx----- 2 adrian adrian 4096 Nov 24 10:27 /home/adrian/.ssh
adrian@A6Alumno01:~$ ssh-keygen -t ed25519 -f ~/.ssh/wordpress-key -C "adrian@aws"
Generating public/private ed25519 key pair.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/adrian/.ssh/wordpress-key
Your public key has been saved in /home/adrian/.ssh/wordpress-key.pub
The key fingerprint is:
SHA256:58tBwWketu20NqqHxRQDw+PK6DS4DQthk20kTaSJRGk adrian@aws
The key's randomart image is:
+--[ED25519 256]--+
|.o=o  .o.|
|oEoo  ooo.|
|oo=   .Bo|
|. + o  .+.=|
|..o. o .So= o|
|. o + o +oo .|
|. B .  oo =|
| o o   ...= .|
|      .o+|
+-----[SHA256]-----+
adrian@A6Alumno01:~$

```

Verifica que se creó

correctamente: `ls -la`

`~/.ssh/wordpress-key*`

Deberías ver dos archivos:

- `wordpress-key` (clave privada)
- `wordpress-key.pub` (clave pública)

0.3- Ajustar permisos de la clave privada

`chmod 400 ~/.ssh/wordpress-key`

`ls -la ~/.ssh/wordpress-key`

La salida debe mostrar: `-r-----`

Aquí se puede ver que he verificado que se han creado los dos archivos, también se puede ver que he modificado los permisos de la clave privada que acabo de crear

```
adrian@A6Alumno01:~$ ls -la ~/.ssh/wordpress-key*
-rw----- 1 adrian adrian 444 Nov 24 10:31 /home/adrian/.ssh/wordpress-key
-rw-r--r-- 1 adrian adrian 92 Nov 24 10:31 /home/adrian/.ssh/wordpress-key.pub
adrian@A6Alumno01:~$ chmod 400 ~/.ssh/wordpress-key
adrian@A6Alumno01:~$ ls -la ~/.ssh/wordpress-key
-r----- 1 adrian adrian 444 Nov 24 10:31 /home/adrian/.ssh/wordpress-key
adrian@A6Alumno01:~$
```

PARTE 1: CONFIGURACIÓN EN AWS

1.1- Crear par de claves en AWS

1. Accede a AWS Console → EC2 → Key Pairs
2. Haz clic en "Create key pair"
3. Nombre: `wordpress-key-aws`
4. Type: `Ed25519` (o RSA si tu AWS no soporta Ed25519)
5. File format: `.pem`
6. Clic en "Create key pair"

Se descargará un archivo `wordpress-key-aws.pem`

Aquí he iniciado el AWS, una vez iniciado el AWS, pincho en AWS, me voy a EC2 y luego a Key Pairs, aquí selecciono crear un par de claves y le pongo el nombre indicado y selecciono ED25519 en formato .pem

Rate your experience with this EC2 console. ☆ ☆ ☆ ☆ ☆

aws [Alt+S] United States (Northern Virginia) Account ID: 9553-8121-1762 voclabs/user4557760=Adri_n

EC2 > Pares de claves > Crear par de claves

Crear par de claves Información

Par de claves
Un par de claves, compuesto por una clave privada y una clave pública, es un conjunto de credenciales de seguridad que se utilizan para demostrar su identidad cuando se conecta a una instancia.

Nombre

El nombre puede incluir hasta 255 caracteres ASCII. No puede incluir espacios al principio ni al final.

Tipo de par de claves Información
☐ RSA
 ☒ ED25519

Formato de archivo de clave privada
☒ .pem Para usar con OpenSSH
☐ .ppk Para usar con PuTTY

Etiquetas - opcional
No hay etiquetas asociadas a este recurso.
 Puede agregar hasta 50 etiquetas más.

<input type="checkbox"/>	Nombre	Tipo	Se ha crea...	Huella digital	ID
<input type="checkbox"/>	wordpress-ke...	ed25519	2025/11/25 ...	no7QPLvtLN...	key-0b6bc...

1.2- Trasferir la clave descargada a WSL

Si descargaste el archivo en Windows:

```
cp /mnt/c/Users/TU-USUARIO/Downloads/wordpress-key-aws.pem ~/.ssh/
```

```
chmod 400 ~/.ssh/wordpress-key-aws.pem
```

Aquí básicamente he copiado la clave `wordpress-key-aws.pem` desde Windows a

```
adrian@A6Alumno01:/mnt/c/Users/Alumno.DESKTOP-DISKUUG$ cp /mnt/c/Users/Alumno.DESKTOP-DISKUUG/Downloads/wordpress-key-aws.pem ~/.ssh/
adrian@A6Alumno01:/mnt/c/Users/Alumno.DESKTOP-DISKUUG$ chmod 400 ~/.ssh/wordpress-key-aws.pem
adrian@A6Alumno01:/mnt/c/Users/Alumno.DESKTOP-DISKUUG$ ls -la ~/.ssh/wordpress-key-aws.pem
-r----- 1 adrian adrian 387 Nov 25 09:01 /home/adrian/.ssh/wordpress-key-aws.pem
adrian@A6Alumno01:/mnt/c/Users/Alumno.DESKTOP-DISKUUG$
```

WSL y le he puesto permisos 400 para poder usarla con SSH de forma segura.

1.3- Crear Security Group

1. En AWS Console, ve a EC2 → Security Groups
2. Haz clic en "Create security group"
3. Nombre: **wordpress-aws-sg**
4. Descripción: Security group para WordPress en AWS

Create security group Information

A security group acts as a virtual firewall for the instance to control inbound and outbound traffic. T following fields.

Basic details

Security group name Information

The name cannot be edited after it has been created.

Description Information

VPC Information

1.4- Configurar reglas de entrada del Security Group

Añade las siguientes reglas de entrada:

Regla 1: SSH

- Type: **SSH**
- Protocol: **TCP**
- Port: **22**

- Source: 0.0.0.0/0

Entry rules Information

Regla de entrada 1 Eliminar

Tipo <small>Información</small>	Protocolo <small>Información</small>	Intervalo de puertos <small>Información</small>
SSH	TCP	22
Tipo de origen <small>Información</small>	Origen <small>Información</small>	Descripción: opcional <small>Información</small>
Anywhere-IPv4	<input type="text" value="0.0.0.0/0"/>	<input type="text"/>

Add rule

- Regla 2: HTTP

(ngrok)

- Type: Custom TCP
- Port: 80
- Source: 0.0.0.0/0

Regla de entrada 2 Eliminar

Tipo <small>Información</small>	Protocolo <small>Información</small>	Intervalo de puertos <small>Información</small>
TCP personalizado	TCP	80
Tipo de origen <small>Información</small>	Origen <small>Información</small>	Descripción: opcional <small>Información</small>
Anywhere-IPv4	<input type="text" value="0.0.0.0/0"/>	<input type="text"/>

Add rule

⚠ Las reglas cuyo origen es 0.0.0.0/0 o ::/0 permiten a todas las direcciones IP acceder a la instancia. Recomendamos configurar

- **Regla 3: HTTPS**

(ngrok)

- Type: **Custom TCP**
- Port: **443**
- Source: **0.0.0.0/0**

Regla de entrada 3 Eliminar

Tipo [Información](#)
 TCP personalizado ▼

Protocolo [Información](#)
 TCP

Intervalo de puertos [Información](#)
 443

Tipo de origen [Información](#)
 Anywhere-IPv4 ▼

Origen [Información](#)
 0.0.0.0/0 X

Descripción: opcional [Información](#)

Aquí se pueden ver todas las reglas de entradas creadas

Reglas de entrada (3) Administrar e

Q Buscar

<input type="checkbox"/>	Name ▼	ID de la regla de entr... ▼	Versión de IP ▼	Tipo ▼	Protocolo ▼	Intervalo de puertos ▼	Origen
<input type="checkbox"/>	-	sgr-0e5462cdb485bd7fe	IPv4	HTTPS	TCP	443	0.0.0.0/0
<input type="checkbox"/>	-	sgr-07e62b8660169ca63	IPv4	HTTP	TCP	80	0.0.0.0/0
<input type="checkbox"/>	-	sgr-064c3a73af75c0dca	IPv4	SSH	TCP	22	0.0.0.0/0

Guarda el Security Group. (ya lo he guardado arriba).

<input type="checkbox"/>	-	sg-08247a7f9716ea4cb	wordpress-aws-sg	vpc-0c04751fc6024545d	Security group para WordPress en AWS	955381211762
--------------------------	---	--------------------------------------	------------------	---------------------------------------	--------------------------------------	--------------

1.5- Crear 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: Selecciona **wordpress-key-aws**
6. VPC settings: Default

7. Security group: Selecciona **sg-wordpress-aws**

8. Storage: **15 GiB, gp3**

9. Haz clic en "Launch instance"

Espera a que la instancia esté en estado "running".

Aquí he ido a lanzar una instancia, la he nombrado, después he seleccionado ubuntu 24.04 LTS.

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

Launch an instance Information

Amazon EC2 lets you create virtual machines, or instances, that run in the AWS cloud. Get started quickly by following the simple steps below.

Name and labels Information

Name

 [Add additional tags](#)

▼ Application and operating system images (Amazon machine image) Information

An AMI contains the operating system, application server, and applications for the instance. If you don't see a suitable AMI below, use the search field or choose [Find More AMIs](#).

Recientes

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
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

Luego he seleccionado que la instancia sea tipo T3 micro, tambien la key pair he selccionado la que había creado antes.

▼ Instance type Information [Get advice](#)

Instance type

t3.micro

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

Apto para la capa gratuita

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

[Compare instance types](#)

[Se aplican costos adicionales a las AMI con software preinstalado](#)

▼ Key pair (login) Information

You can use a key pair to securely connect to the instance. Ensure you have access to the selected key pair before launching the instance.

Key pair name - *required*

 [Create a new key pair](#)

Ahora he seleccionado el grupo de seguridad y el espacio para la máquina que en este caso son 15gb gp3.

Network configurations

Red: vpc-0c04751fc6024545d

Subnetwork: Sin preferencias (subred predeterminada en cualquier zona de disponibilidad)

Automatically assign the public IP address: Habilitar

Firewall (security groups): 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.

Create security group | Select an existing security group

Grupos de seguridad comunes: Seleccionar grupos de seguridad

wordpress-aws-sg sg-08247a7f9716ea4cb VPC: vpc-0c04751fc6024545d

Compare reglas de grupo de seguridad

Configure storage

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

Agregar un nuevo volumen

Instancia en ejecución ya:

<input type="checkbox"/>	wordpress-...	i-09b42368548496e09	En ejecución	t3.micro
--------------------------	---------------	---------------------	--------------	----------

1.6- Obtener la IP pública

1. Selecciona la instancia
2. Copia la "Public IPv4 address" (ej: 54.123.45.c7)

<input checked="" type="checkbox"/>	wordpress-...	i-09b42368548496e09	En ejecución	t3.micro	3/3 comprobador	Ver alarmas +	us-east-1f	ec2-44-200-171-14.co...	44.200.171.14	-
-------------------------------------	---------------	---------------------	--------------	----------	-----------------	---------------	------------	-------------------------	---------------	---

Aquí he tenido que crear dejar una regla de salida, si no la pongo no va a descargar nada ni salir tráfico a internet.

Reglas de salida (1)							
Name	ID de la regla del gr...	Versión de IP	Tipo	Protocolo	Intervalo de puertos	Destino	
-	sgr-0a5d1e17da439541c	IPv4	Todo el tráfico	Todo	Todo	0.0.0.0/0	

PARTE 2: CONEXIÓN SSH DESDE WSL A AWS

2.1- Conectar a la instancia

Reemplaza TU-IP-PUBLICA con la IP obtenida:

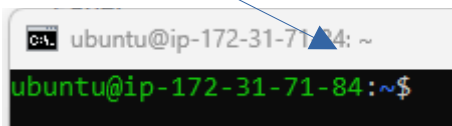
```
ssh -i ~/.ssh/wordpress-key-aws.pem ubuntu@TU-IP-PUBLICA
```

Ejemplo:

```
ssh -i ~/.ssh/wordpress-key-aws.pem ubuntu@54.123.45.67
```

La primera vez, te pedirá confirmar la huella digital del servidor. Escribe **yes**.

Ya estaría conectado y con su IP.



2.2- Verificar conexión

Una vez conectado, deberías ver algo

como: ubuntu@ip-10-0-0-100:~\$

PARTE 3: INSTALACIÓN BASE DEL SERVIDOR (EN AWS)

3.1- Actualizar el sistema

```
sudo apt update
```

```
sudo apt upgrade -y
```

```
ubuntu@ip-172-31-66-76:~$ sudo apt update && sudo apt upgrade -y
```

3.2- Instalar LAMP Stack

```
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
```

Ya estaría instalado

```
ubuntu@ip-172-31-71-84:~$ 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
```

3.3- Iniciar servicios

```
sudo systemctl start apache2
```

```
sudo systemctl start mysql
```

```
sudo systemctl enable apache2
```

```
sudo systemctl enable mysql
```

```
ubuntu@ip-172-31-71-84:~$ sudo systemctl start apache2
ubuntu@ip-172-31-71-84:~$ sudo systemctl start mysql
```

```
ubuntu@ip-172-31-71-84:~$ sudo systemctl enable apache2
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
```

3.4- Verificar servicios

```
sudo systemctl status apache2
```

Aquí se puede ver como esta activo y corriendo.

```
ubuntu@ip-172-31-71-84:~$ sudo systemctl status apache2
Warning: The unit file, source configuration file or drop-ins of apache2.service changed
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: enabled)
   Active: active (running) since Thu 2025-11-27 08:43:06 UTC; 12min ago
```

```
ubuntu@ip-172-31-71-84:~$ sudo systemctl status mysql
● mysql.service - MySQL Community Server
   Loaded: loaded (/usr/lib/systemd/system/mysql.service; enabled; preset: enabled)
   Active: active (running) since Thu 2025-11-27 08:43:08 UTC; 20min ago
```

PARTE 4: SCRIPT DE AUTOMATIZACIÓN DE WORDPRESS

4.1- Crear script de instalación

En tu máquina local, crea un archivo llamado 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"
```

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;"
```

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;"
```

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/*

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/wp-config.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/wp-config.php

Paso 6: Permisos

echo "Configurando permisos..."

sudo chown -R www-data:www-data /var/www/html/

sudo chmod -R 755 /var/www/html/

Paso 7: Habilitar mod_rewrite en Apache

echo "Habilitando mod_rewrite..."

sudo a2enmod rewrite

sudo systemctl restart apache2

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://TU-IP-PUBLICA para finalizar la instalación de WordPress"

Básicamente he creado el archivo install-wordpress.sh y he pegado el código. (antes he salido a el wsl)

```
GNU nano 7.2                                install-wordpress.sh *
echo "Configurando permisos..."

sudo chown -R www-data:www-data /var/www/html/
sudo chmod -R 755 /var/www/html/

# Paso 7: Habilitar mod_rewrite en Apache
echo "Habilitando mod_rewrite..."

sudo a2enmod rewrite
sudo systemctl restart apache2

# 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}
```

PARTE 5: MIGRACIÓN DE ARCHIVOS CON SCP

5.1- Transferir el script a AWS

Desde tu máquina local:

```
scp -i ~/.ssh/wordpress-key-aws.pem install-wordpress.sh ubuntu@TU-IP-PUBLICA:~/
```

Primeramente he tenido que salirme de la maquina de aws, una vez estando en mi wsl ejecuto el comando para transferir el script.

```
adrian@A6Alumno01:~$ scp -i ~/.ssh/wordpress-key-aws.pem install-wordpress.sh ubuntu@98.92.24.65:~/
install-wordpress.sh                                     100% 2673   24.8KB/s   00:00
```

```
ubuntu@ip-172-31-71-84:~$ ls -l install-wordpress.sh
-rwxr-xr-x 1 ubuntu ubuntu 2673 Nov 27 09:26 install-wordpress.sh
ubuntu@ip-172-31-71-84:~$
```

5.2– Dar permisos de ejecución

En AWS (*dentro de la sesión SSH*):

```
chmod +x ~/install-wordpress.sh
```

Aquí he modificado los permisos para poder ejecutar después el script

```
ubuntu@ip-172-31-71-84:~$ chmod +x install-wordpress.sh
```

5.3– Ejecutar el script

```
./install-wordpress.sh
```

Espera a que termine. Al finalizar, verás un archivo con las credenciales:

```
cat ~/wordpress-credentials.txt
```

Aquí básicamente he ejecutado el script, al ejecutarlo me suelta credenciales

```
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://TU-IP-PUBLICA para finalizar la instalación de WordPress
ubuntu@ip-172-31-71-84:~$
```

```
ubuntu@ip-172-31-71-84:~$ cat ~/wordpress-credentials.txt
=== CREDENCIALES DE WORDPRESS ===
Base de datos: wordpress
Usuario BD: wpuser
Contraseña BD: DKpRhjHlyfzduQN6
Usuario root MySQL: root
Contraseña root MySQL: Fr2pjCgkDmKnVOAJ

Acceso local: http://localhost
Acceso remoto: (se configurará con ngrok)
```

PARTE 6: VERIFICACIÓN DE INSTALACIÓN

6.1- Verificar servicios

```
sudo systemctl status apache2
```

```
sudo systemctl status mysql
```

```
ubuntu@ip-172-31-71-84:~$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: enabled)
   Active: active (running) since Thu 2025-11-27 09:39:51 UTC; 4min 1s ago
     Docs: https://httpd.apache.org/docs/2.4/
   Process: 2542 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
  Main PID: 2545 (apache2)
    Tasks: 7 (limit: 1017)
   Memory: 27.3M (peak: 28.4M)
      CPU: 164ms
   CGroup: /system.slice/apache2.service
           └─2545 /usr/sbin/apache2 -k start
             └─2548 /usr/sbin/apache2 -k start
               └─2549 /usr/sbin/apache2 -k start
                 └─2550 /usr/sbin/apache2 -k start
                   └─2551 /usr/sbin/apache2 -k start
                     └─2552 /usr/sbin/apache2 -k start
                       └─2558 /usr/sbin/apache2 -k start

Nov 27 09:39:51 ip-172-31-71-84 systemd[1]: Starting apache2.service - The Apache HTTP Server...
Nov 27 09:39:51 ip-172-31-71-84 systemd[1]: Started apache2.service - The Apache HTTP Server.
ubuntu@ip-172-31-71-84:~$ sudo systemctl status mysql
● mysql.service - MySQL Community Server
   Loaded: loaded (/usr/lib/systemd/system/mysql.service; enabled; preset: enabled)
   Active: active (running) since Thu 2025-11-27 08:43:08 UTC; 1h 0min ago
  Main PID: 862 (mysqld)
    Status: "Server is operational"
```

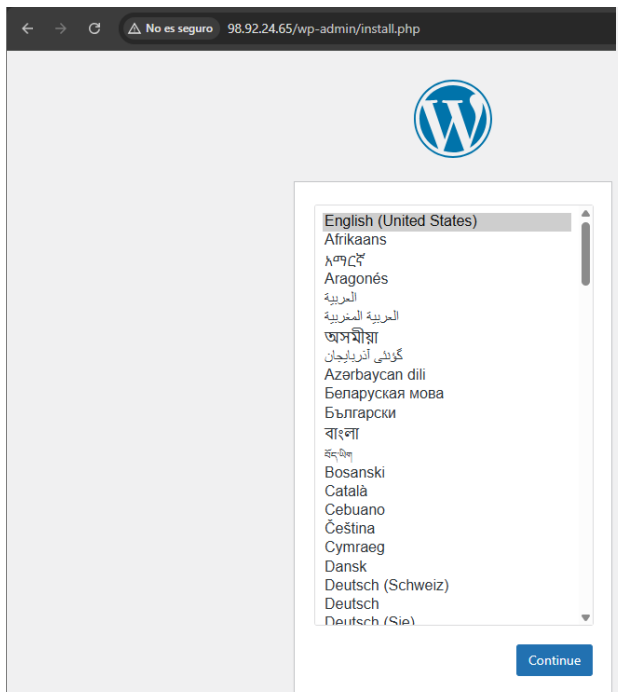
6.2- Acceder desde navegador local

Abre tu navegador y accede a:

<http://TU-IP-PUBLICA>

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

Aquí se puede observar como he accedido a wordpress y a su instalación, seguido a esto he configurado el idioma, usuario y todo eso.... Luego me logueo, una vez logueado se podrá ver el panel de control de wordpress.



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
 Muy débil

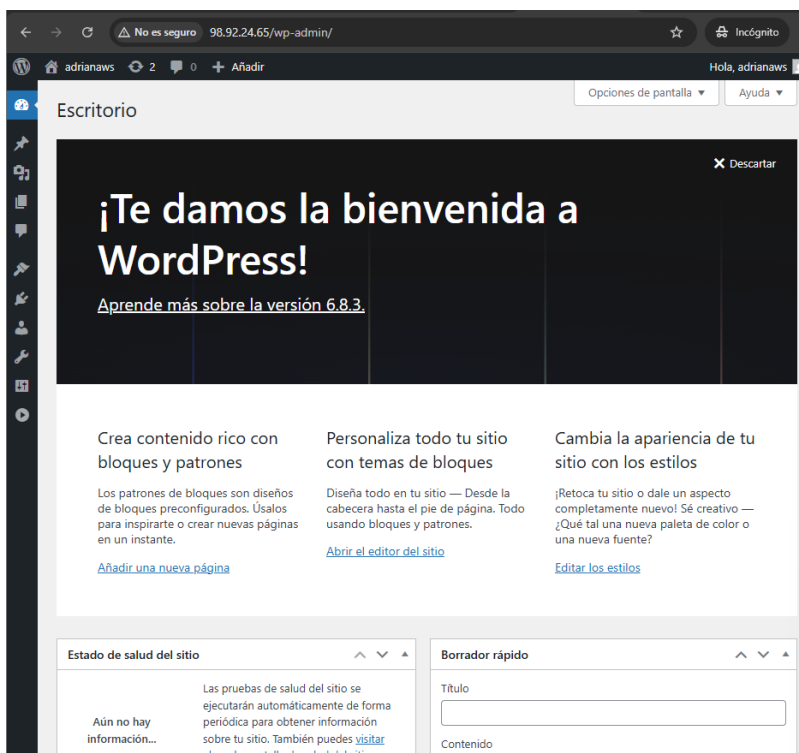
Importante: Necesitas esta contraseña para acceder. Por favor, guárdala en un lugar seguro.

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](#)



Escritorio

¡Te damos la bienvenida a WordPress!

[Aprende más sobre la versión 6.8.3.](#)

Crea contenido rico con bloques y patrones
 Los patrones de bloques son diseños de bloques preconfigurados. Úsalos para inspirarte o crear nuevas páginas en un instante.
[Añadir una nueva página](#)

Personaliza todo tu sitio con temas de bloques
 Diseña todo en tu sitio — Desde la cabecera hasta el pie de página. Todo usando bloques y patrones.
[Abrir el editor del sitio](#)

Cambia la apariencia de tu sitio con los estilos
 ¡Retoca tu sitio o dale un aspecto completamente nuevo! Sé creativo — ¿Qué tal una nueva paleta de color o una nueva fuente?
[Editar los estilos](#)

Estado de salud del sitio
 Las pruebas de salud del sitio se ejecutarán automáticamente de forma periódica para obtener información sobre tu sitio. También puedes [visitar](#)
 Aún no hay información...

Borrador rápido
 Título:
 Contenido:

PARTE 7: HACER WORDPRESS ACCESIBLE DESDE INTERNET CON NGROK

7.1- Instalar ngrok en AWS

En la sesión SSH de AWS:

```
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/
```

Aquí he instalado ngrok en la instancia de AWS. Primero me he ido a mi carpeta home, luego he descargado el binario con wget, lo he descomprimido con tar y por último he movido el ejecutable a /usr/local/bin para poder usar el comando ngrok desde cualquier ruta.a

```
ubuntu@ip-172-31-71-84:~$ wget https://bin.equinox.io/c/bNyj1mQVY4c/ngrok-v3-stable-linux-amd64.tgz
--2025-11-27 10:06:53-- https://bin.equinox.io/c/bNyj1mQVY4c/ngrok-v3-stable-linux-amd64.tgz
Resolving bin.equinox.io (bin.equinox.io)... 35.71.179.82, 99.83.220.108, 13.248.244.96, ...
Connecting to bin.equinox.io (bin.equinox.io)|35.71.179.82|: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-amd64.t 100%[=====] 10.47M --.-KB/s in 0.04s

2025-11-27 10:06:53 (251 MB/s) - 'ngrok-v3-stable-linux-amd64.tgz' saved [10980950/10980950]

ubuntu@ip-172-31-71-84:~$ tar -xvzf ngrok-v3-stable-linux-amd64.tgz
ngrok
ubuntu@ip-172-31-71-84:~$ sudo mv ngrok /usr/local/bin/
ubuntu@ip-172-31-71-84:~$
```

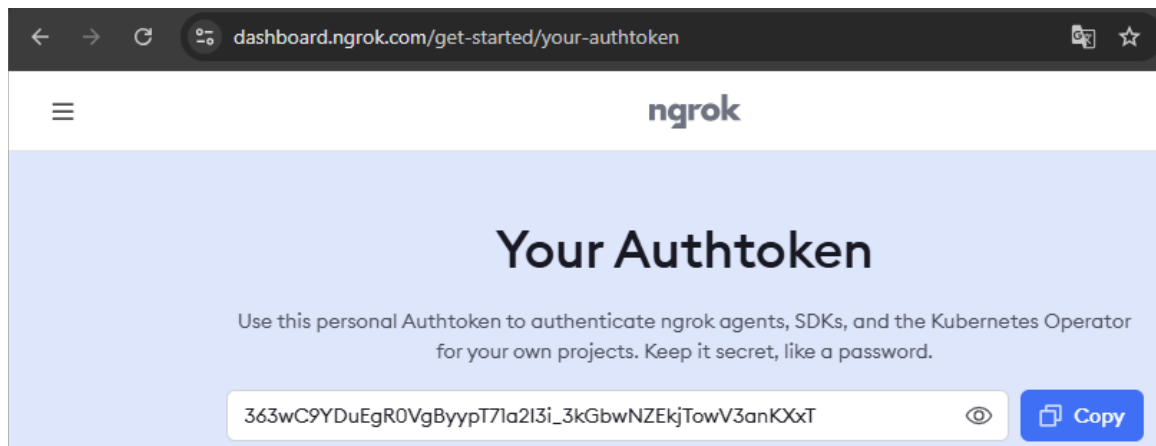
7.2- Autenticar ngrok

Necesitas un token de ngrok. Regístrate en <https://ngrok.com>

Configura tu token (*reemplaza TU_TOKEN*):

```
ngrok config add-authtoken TU_TOKEN_AQUI
```

Aquí básicamente he copiado mi token personal de ngrok desde la web y lo he pegado en el comando ngrok config add-authtoken. Con esto dejo ngrok vinculado a mi cuenta para poder crear túneles desde esta instancia.



```
ubuntu@ip-172-31-71-84: ~  
ubuntu@ip-172-31-71-84:~$ ngrok config add-authtoken 363wC9YDuEgR0VgByypT71a2I3i_3kGbwNZEKjTowV3anKXxT  
Authtoken saved to configuration file: /home/ubuntu/.config/ngrok/ngrok.yml  
ubuntu@ip-172-31-71-84:~$
```

7.3- Iniciar ngrok

ngrok http 80

Verás una salida como:

```
ngrok (Ctrl+C to  
Session Status online  
Account your-email@example.com  
Version 3.x.x  
Region us (United States)  
Forwarding https://abc123def456.ngrok-free.app ->  
http://localhost:80 Copia la URL HTTPS (ej: https://abc123def456.ngrok-  
free.app)
```

Aquí en este paso lanzo ngrok http 80 para exponer el puerto 80 de la máquina. En la salida se ve que la sesión está en estado online y que me genera una URL pública HTTPS que apunta a http://localhost:80. Esa es la dirección que voy a usar desde fuera.

```
ngrok (Ctrl+C to quit)  
Call internal services from your gateway: https://ngrok.com/x/http-request  
Session Status online  
Account adrian (Plan: Free)  
Version 3.33.1  
Region United States (us)  
Web Interface http://127.0.0.1:4040  
Forwarding https://blanche-unscolding-reid.ngrok-free.dev -> http://localhost:80  
Connections
```

	t1	opn	rt1	rt5	p50	p90
Connections	0	0	0.00	0.00	0.00	0.00

7.4- Actualizar WordPress

Necesitas actualizar las URLs en WordPress. En otra terminal SSH (*nueva sesión*):

Aquí he abierto otra sesión (básicamente dos sesiones en una misma instancia).

```
ssh -i ~/.ssh/wordpress-key-aws.pem ubuntu@TU-IP-PUBLICA
```

Accede a MySQL:

```
mysql -u wpuser -p -D wordpress
```

```
ubuntu@ip-172-31-71-84:~$ sudo 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 12
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.

mysql>
```

Ejecuta (*reemplaza la URL*):

```
UPDATE wp_options SET option_value='https://abc123def456.ngrok-free.app'
WHERE option_name='siteurl';
```

```
UPDATE wp_options SET option_value='https://abc123def456.ngrok-free.app'
WHERE option_name='home';
```

```
EXIT;
```

Aquí lógicamente hay que cambiar los enlaces tuyos por los que hemos generado anteriormente con ngrok.

```
mysql> UPDATE wp_options SET option_value='https://blanche-unsolding-reid.ngrok-free.dev' WHERE option_name='siteurl';
Query OK, 1 row affected (0.00 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> UPDATE wp_options SET option_value='https://blanche-unsolding-reid.ngrok-free.dev' WHERE option_name='home';
Query OK, 1 row affected (0.01 sec)
Rows matched: 1 Changed: 1 Warnings: 0

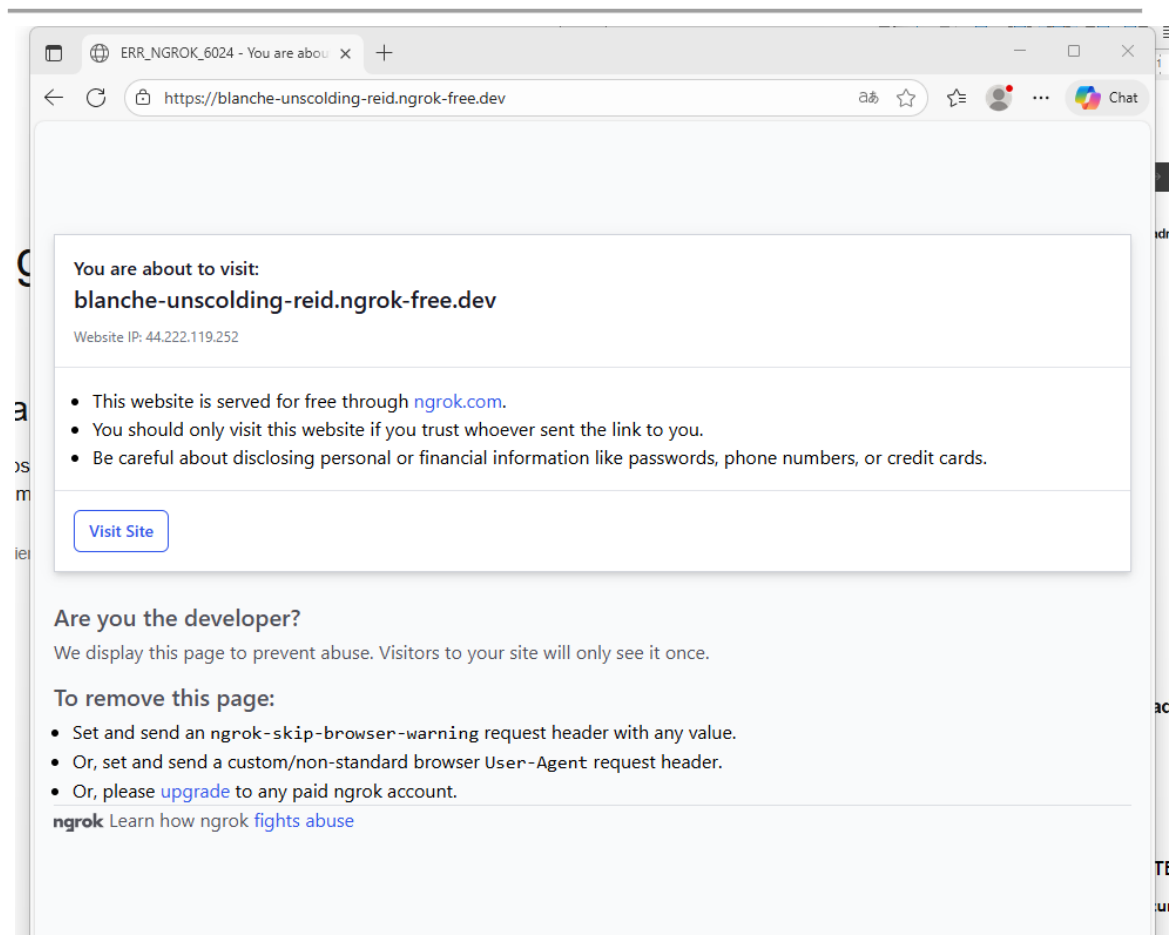
mysql>
```

7.5- Probar acceso remoto

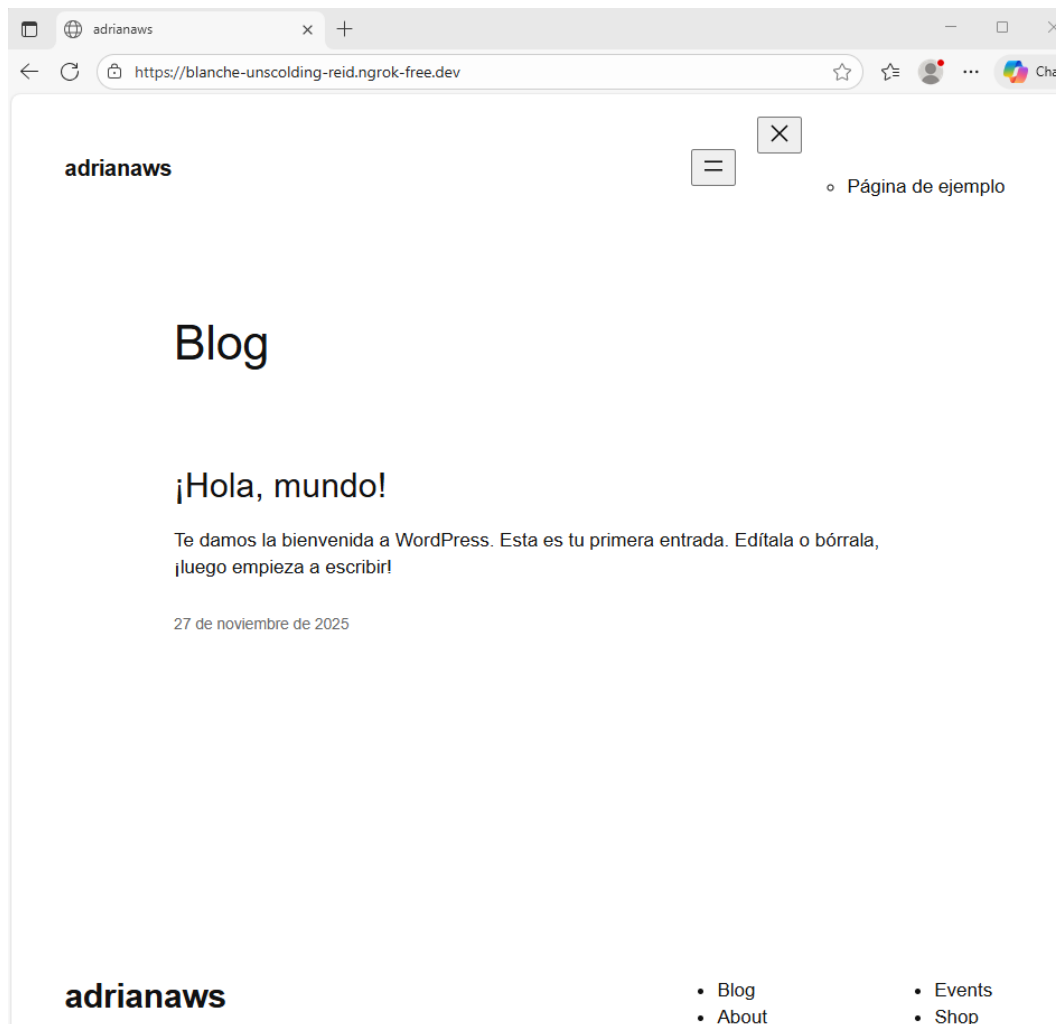
Abre tu navegador y accede a:

<https://tu-url.ngrok-free.app>

Si ngrok muestra una página de advertencia, haz clic en "Visit Site".



Aquí ya se puede ver la página funcionando correctamente



PARTE 8: ENTREGABLES

Capturas Requeridas (*TODOS LOS COMANDOS Y ACCESOS A PÁGINAS WEBS*)

1. Clave SSH generada

- Terminal mostrando la generación con keygen
- Contenido de ~/.ssh mostrando ambas claves (*privada y pública*)

2. AWS - Security Group

- Reglas de entrada configuradas (*SSH, HTTP, HTTPS*)
- Nombre: sg-wordpress-aws

3. AWS - Instancia EC2

- Instancia en estado "running"

- IP pública visible
- Clave asociada

4. Conexión SSH

- Terminal mostrando conexión exitosa con comando ssh
- Prompt del servidor AWS (*ubuntu@ip-...*)

5. Script de automatización

- Archivo install-wordpress.sh
- Ejecución del script mostrando progreso
- Contenido de wordpress-credentials.txt

6. Migraciones SCP

- Comando scp usado para transferir archivos
- Confirmación de transferencia exitosa

7. WordPress funcionando

- Acceso local: <http://TU-IP-PUBLICA>
- Panel de administración: <http://TU-IP-PUBLICA/wp-admin>
- Página con candado HTTPS desde ngrok: <https://tu-url.ngrok-free.app>

8. ngrok en ejecución

- Terminal mostrando ngrok activo con URL pública
- URL tipo <https://xxxxx.ngrok-free.app> visible

9. Base de datos

- Comando mostrando credenciales en wordpress-credentials.txt
- Lista de bases de datos: `mysql -u root -p -e "SHOW DATABASES;"`

```
ubuntu@ip-172-31-71-84:~$ sudo mysql -u root -p -e "SHOW DATABASES;"
Enter password:
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
| wordpress |
+-----+
ubuntu@ip-172-31-71-84:~$
```

NOTAS IMPORTANTES

- **Ed2551G vs RSA:** Ed25519 es más moderno y seguro. Si AWS no lo soporta, usa RSA con `ssh-keygen -t rsa -b 4096`
- **Seguridad:** Nunca compartas tu clave privada (`~/.ssh/wordpress-key-aws.pem`)
- **ngrok:** Solo para desarrollo/pruebas. En producción usa un dominio real
- **URL dinámica:** La URL de ngrok cambia cada reinicio (plan gratuito). Actualiza WordPress en consecuencia
- **Almacenar credenciales:** Guarda `~/wordpress-credentials.txt` en lugar seguro

- **Eliminar instancia:** Cuando termines, detén la instancia para no incurrir en costos innecesarios
-

TROUBLESHOOTING

Error: Permission denied (*publickey*)

`chmod 400 ~/.ssh/wordpress-key-aws.pem`

Error: Connection refused

- Verifica que la instancia esté en estado "running"
- Comprueba que usas la IP correcta
- Usa `ssh -v` para depuración

ngrok no se conecta

- Comprueba que el puerto 80 está abierto en el Security Group
- Verifica que Apache está corriendo: `sudo systemctl status apache2`

WordPress no carga desde ngrok

- Actualiza las URLs en la base de datos (paso 7.4)
- Limpia la caché del navegador
- Usa incógnito/privado para probar