

INSTALACIÓN DE WORDPRESS EN AWS CON SSH



Autor: Diestefano JIampieri

Asignatura: Arquitectura en la Nube

Tutor: Rubén

Fecha de Entrega: 10/12/2025

ÍNDICE

PREPARACIÓN.....	2
CONFIGURACIÓN EN AWS.....	3
CONEXIÓN SSH DESDE WSL A AWS.....	8
INSTALACIÓN BASE DEL SERVIDOR.....	9
SCRIPT DE AUTOMATIZACIÓN DE WORDPRESS.....	10
MIGRACIÓN DE ARCHIVOS CON SCP.....	13
VERIFICACIÓN DE INSTALACIÓN.....	15
WORDPRESS ACCESIBLE DESDE INTERNET CON NGROK.....	16

PREPARACIÓN

1. Empezaremos creando un directorio seguro ya que el sistema necesita una carpeta específica para guardar las llaves, "`mkdir -p ~/.ssh; chmod 700 ~/.ssh`".

```
dies@A6Alumno14:~$ mkdir -p ~/.ssh; chmod 700 ~/.ssh
```

2. Ahora generamos el par de claves SSH con Ed25519, el cual es un algoritmo moderno y rápido, "`ssh-keygen -t ed25519 -f ~/.ssh/wordpress-key -C "dies"`". Nos pedirá una contraseña, podemos darle dos veces al enter para que no nos pida la contraseña siempre que lo usemos.

```
dies@A6Alumno14:~$ ssh-keygen -t ed25519 -f ~/.ssh/wordpress-key -C "dies"
Generating public/private ed25519 key pair.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/dies/.ssh/wordpress-key
Your public key has been saved in /home/dies/.ssh/wordpress-key.pub
The key fingerprint is:
SHA256:6cIRrLOpKUxBKcPyGMJ90g8wHAaz/dOCTkQjWkhYnAQ dies
```

3. Verificamos que se han creado correctamente las claves, "`ls -la ~/.ssh/wordpress-key*`". Nos mostrará wordpress-key que es la clave privada y wordpress-key.pub que es la publica.

```
dies@A6Alumno14:~$ ls -la ~/.ssh/wordpress-key*
-rw----- 1 dies dies 387 Dec  5 12:47 /home/dies/.ssh/wordpress-key
-rw-r--r-- 1 dies dies  86 Dec  5 12:47 /home/dies/.ssh/wordpress-key.pub
```

4. Procederemos ajustar la clave privada, es tan sensible que debemos quitarle permisos de escritura, incluso a nosotros mismos para evitar accidentes, "`chmod 400 ~/.ssh/wordpress-key`".

```
dies@A6Alumno14:~$ chmod 400 ~/.ssh/wordpress-key
```

5. Verificamos que está todo correcto, "`ls -la ~/.ssh/wordpress-key`".

```
dies@A6Alumno14:~$ ls -la ~/.ssh/wordpress-key
-r----- 1 dies dies 387 Dec  5 12:47 /home/dies/.ssh/wordpress-key
```

CONFIGURACIÓN EN AWS

1. Ahora configuraremos la infraestructura en la nube, empezaremos creando la llave en AWS y guardarla en WSL. Iremos a CS2, Red y Seguridad y Pares de Claves.



2. Seguimos dándole a, Crear para de clave, rellenamos el formulario como en la imagen y le damos al botón naranja.

Par de claves

Un par de claves, compuesto por una clave privada y una clave pública, es un conjunto de credenciales 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

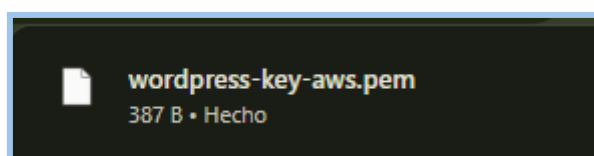
☒ ED25519

Formato de archivo de clave privada

☒ .pem
Para usar con OpenSSH

☐ .ppk
Para usar con PuTTY

3. Se nos descargará un archivo .PEM.



4. Una vez descargado el archivo, crearemos una carpeta segura para guardar las llaves, `mkdir -p ~/.ssh`.

```
dies@A6Alumno14:~$ mkdir -p ~/.ssh
```

5. Continuamos copiando la llave desde Windows, `cp /mnt/c/Users/TuUsuarioDeWindows/Downloads/wordpress-key-aws.pem ~/.ssh/`.

```
dies@A6Alumno14:~$ cp /mnt/c/Users/Alumno.DESKTOP-DISKUG/Downloads/wordpress-key-aws.pem ~/.ssh/
```

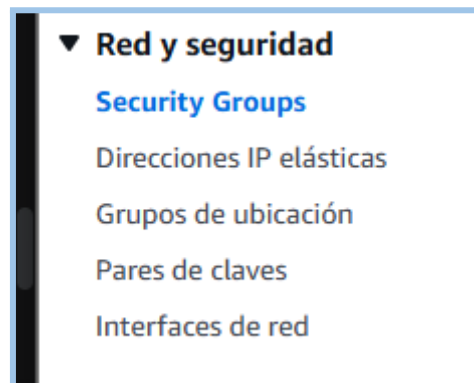
6. Le aplicaremos permisos para que nadie más que solo nosotros podamos leer las llaves, ya que si el archivo es muy abierto, la conexión fallará, `chmod 400 ~/.ssh/wordpress-key-aws.pem`.

```
dies@A6Alumno14:~$ chmod 400 ~/.ssh/wordpress-key-aws.pem
```

7. Revisamos, `ls -l ~/.ssh/wordpress-key-aws.pem`.

```
dies@A6Alumno14:~$ ls -l ~/.ssh/wordpress-key-aws.pem
-r----- 1 dies dies 387 Dec  5 13:29 /home/dies/.ssh/wordpress-key-aws.pem
```

8. Ahora seguiremos creando un grupo de seguridad que permita el tráfico web y la conexión remota. En el apartado de Red y Seguridad haremos click sobre Security Groups y luego a Crear grupo de Seguridad.



9. En los apartados pondremos lo siguiente:

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](#)

Tipo Información	Protocolo Información	Intervalo de puertos Información	Origen Información
SSH	TCP	22	Anywhe... 0.0.0.0/0
HTTP	TCP	80	Anywhe... 0.0.0.0/0
HTTPS	TCP	443	Anywhe... 0.0.0.0/0

10. Cuando tengamos todo le daremos a crear, y se nos creará correctamente.

✓ El grupo de seguridad (sg-053b2827204698af4 | wordpress-sg) se ha creado correctamente

► Detalles

11. Una vez creado el grupo de seguridad, ahora conectaremos todo, crearemos una nueva instancia, en cuanto a la configuración pondremos la siguiente

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

▼ **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

wordpress-key-aws

[Crear un nuevo par de claves](#)

Configuraciones de red Información [Editar](#)

Red Información

vpc-06480ab144a125c7d

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-sg sg-053b2827204698af4 ✕
VPC: vpc-06480ab144a125c7d

🔄

[Compare reglas de grupo de seguridad](#)

12. Cuando tengamos la configuración le daremos a crear y se lanzará la instancia.

✓

wordpress-ser...

i-0e28dccc0b5d44f6e

✓ En ejecución

🔍

t3.micro

CONEXIÓN SSH DESDE WSL A AWS

1. Tras la creación de la instancia, vamos a conectarnos mediante SSH desde la terminal de AWS, "`ssh -i ~/.ssh/wordpress-key-aws.pem ubuntu@13.220.105.216`".

```
dies@A6Alumno14:~$ ssh -i ~/.ssh/wordpress-key-aws.pem ubuntu@13.220.105.216
The authenticity of host '13.220.105.216 (13.220.105.216)' can't be established.
ED25519 key fingerprint is SHA256:VXTIUrr2g90hd+JLFavr+m6hNNpvL5EzLxXTMx0/tm0.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '13.220.105.216' (ED25519) to the list of known hosts.
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 6.11.0-1015-aws x86_64)
```

2. Si se ha conectado correctamente, nos tendría que salir lo siguiente.

```
ubuntu@ip-172-31-71-149:~$
```


INSTALACIÓN BASE DEL SERVIDOR

1. Nos aseguramos de que el servidor esté actualizado, "`sudo apt update && sudo apt upgrade -y`".

```
ubuntu@ip-172-31-71-149:~$ 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 [126 kB]
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 [126 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [126 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [126 kB]
```

2. Instalamos Lamp Stack, con el comando instalaremos, "`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`", el servidor web, el lenguaje PHP con todas sus librerías necesarias para WordPress y la base de datos MySQL.

```
ubuntu@ip-172-31-71-149:~$ sudo apt install apache2 php php-mysql libapache2-mod-php php-curl php-gd p
hp-mbstring php-xml php-xmlrpc php-intl php-zip mysql-server -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
apache2 is already the newest version (2.4.58-1ubuntu8.8).
```

3. Iniciaremos y automatizamos los servicios, si se reinicia el servidor, se prenden solos, "`sudo systemctl start apache2; sudo systemctl start mysql; sudo systemctl enable apache2; sudo systemctl enable mysql`".

```
ubuntu@ip-172-31-71-149:~$ 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-inst
all.
Executing: /usr/lib/systemd/systemd-sysv-install enable apache2
Synchronizing state of mysql.service with SysV service script with /usr/lib/systemd/systemd-sysv-instal
l.
Executing: /usr/lib/systemd/systemd-sysv-install enable mysql
```

4. Revisamos el estado de apache, "`sudo systemctl status apache2`".

```
ubuntu@ip-172-31-71-149:~$ 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 2025-12-05 13:12:14 UTC; 3min 41s ago
     Docs: https://httpd.apache.org/docs/2.4/
```

SCRIPT DE AUTOMATIZACIÓN DE WORDPRESS

1. Automatizamos la instalación con un script, abriremos el editor de texto nano, "[*nano install-wordpress.sh*](#)".

```
ubuntu@ip-172-31-71-149:~$ nano install-wordpress.sh
```

2. Una vez dentro del editor, escribimos lo siguiente, guardamos y nos salimos.

```
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)"

# Paso 1: Configurar MySQL
# Nota: Si ya ejecutaste mysql_secure_installation antes, algunos pasos pueden ser reducidos
echo "Configurando MySQL..."
# Intentamos cambiar la clave root. Si falla es porque ya tiene una, pero seguimos.
sudo mysql -e "ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY '$DB_ROOT_PASSWORD';" || true
sudo mysql -e "DELETE FROM mysql.user WHERE User='';" || true
sudo mysql -e "DELETE FROM mysql.user WHERE User='root' AND Host NOT IN ('localhost', '127.0.0.1');" || true
sudo mysql -e "DROP DATABASE IF EXISTS test;" || true
sudo mysql -e "DELETE FROM mysql.db WHERE Db='test' OR Db='test\\_%';" || true
sudo mysql -e "FLUSH PRIVILEGES;"
```

3. El código entero en la siguiente página.

```

#!/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)"

# Paso 1: Configurar MySQL
# Nota: Si ya ejecutaste mysql_secure_installation antes, algunos pasos pueden ser
# redundantes pero no dañinos.
echo "Configurando MySQL..."
# Intentamos cambiar la clave root. Si falla es porque ya tiene una, pero seguimos.
sudo mysql -e "ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password
BY '${DB_ROOT_PASSWORD}';" || true
sudo mysql -e "DELETE FROM mysql.user WHERE User='';" || true
sudo mysql -e "DELETE FROM mysql.user WHERE User='root' AND Host NOT IN
('localhost', '127.0.0.1', '::1');" || true
sudo mysql -e "DROP DATABASE IF EXISTS test;" || true
sudo mysql -e "DELETE FROM mysql.db WHERE Db='test' OR Db='test\_%';" || true
sudo mysql -e "FLUSH PRIVILEGES;"

# Paso 2: Crear base de datos y usuario de WordPress
echo "Creando base de datos y usuario..."
# Usamos la contraseña root generada para entrar
sudo mysql -u root -p"${DB_ROOT_PASSWORD}" -e "CREATE DATABASE IF NOT
EXISTS ${DB_NAME};"
sudo mysql -u root -p"${DB_ROOT_PASSWORD}" -e "CREATE USER IF NOT EXISTS
'${DB_USER}'@'localhost' IDENTIFIED BY '${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

AQUÍ ESTABA EL ERROR ORIGINAL (Corregido a 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 y reiniciar

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}

EOF

echo "=== Instalación completada ==="

echo "Credenciales guardadas en ~/wordpress-credentials.txt"

MIGRACIÓN DE ARCHIVOS CON SCP

1. Empezaremos moviendo el archivo de tu entorno Local al entorno Remoto, "scp -i ~/.ssh/wordpress-key-aws.pem install-wordpress.sh ubuntu@TU-IP PUBLICA:~/". Este comando lo realizaremos en la máquina local.

```
dies@A6Alumno14:~$ scp -i ~/.ssh/wordpress-key-aws.pem install-wordpress.sh ubuntu@18.206.92.9:~/
install-wordpress.sh 100% 1937 19.3KB/s 00:00
```

2. Ahora volvemos a la sesión de SSH, para que el texto se convierta en un programa que pueda funcionar, "chmod +x ~/install-wordpress.sh".

```
ubuntu@ip-172-31-71-149:~$ chmod +x ~/install-wordpress.sh
```

3. Continuaremos activando la automatización, "./install-wordpress.sh". Aquí me dio un problema por lo que tuve que hacer unos pasos para que me funcionara.

```
ubuntu@ip-172-31-71-149:~$ ./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.
ERROR 1007 (HY000) at line 1: Can't create database 'wordpress'; database exists
```

4. Para que todo funcione correctamente ejecutaré primero, "sudo mysql", en pocas palabras estamos borrando la base de datos por los errores. Y posteriormente escribiré lo de las imágenes.

```
mysql> CREATE DATABASE wordpress;
Query OK, 1 row affected (0.01 sec)
```

```
mysql> DROP USER 'wpuser'@'localhost';
Query OK, 0 rows affected (0.01 sec)
```

```
mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> EXIT
Bye
```

5. Volvemos a ejecutar el comando, “[./install-wordpress.sh](#)” y ahora funcionará.

```
ubuntu@ip-172-31-71-149:~$ ./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...
```

VERIFICACIÓN DE INSTALACIÓN

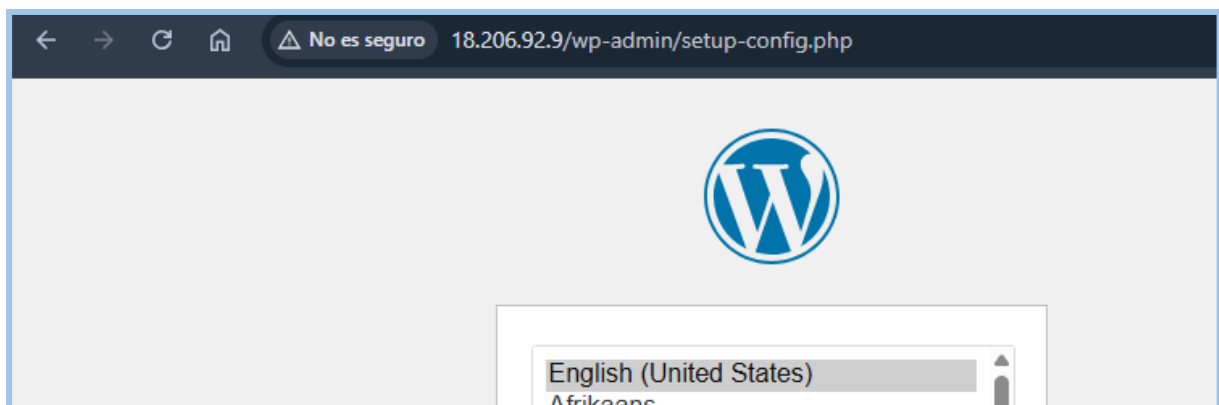
1. Verificamos que todo funciona, empezamos con unas comprobaciones base, "*sudo systemctl status apache2*".

```
ubuntu@ip-172-31-71-149:~$ 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 10:02:36 UTC; 7min ago
     Docs: https://httpd.apache.org/docs/2.4/
   Process: 9065 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
   Main PID: 9068 (apache2)
```

2. Miramos si SQL funciona, "*sudo systemctl status mysql*".

```
ubuntu@ip-172-31-71-149:~$ sudo systemctl status mysql
● mysql.service - MySQL Community Server
   Loaded: loaded (/usr/lib/systemd/system/mysql.service; enabled; preset: enabled)
   Active: active (running) since Tue 2025-12-09 09:42:02 UTC; 28min ago
     Process: 8511 ExecStartPre=/usr/share/mysql/mysql-systemd-start pre (code=exited, status=0/SUCCESS)
    Main PID: 8520 (mysqld)
      Status: "Server is operational"
```

3. Tras las comprobaciones accederemos a la página web, y completamos la configuración, "*http://TU-IP-PUBLICA*".



WORDPRESS ACCESIBLE DESDE INTERNET CON NGROK

1. Para hacer este apartado, empezaremos instalando ngrok en AWS, "`cd ~ wget https://bin.equinox.io/c/bNyj1mQVY4c/ngrok-v3-stable-linux-amd64.tgz`".

```
ubuntu@ip-172-31-71-149:~$ wget https://bin.equinox.io/c/bNyj1mQVY4c/ngrok-v3-stable-linux-amd64.tgz
--2025-12-10 07:59:22-- https://bin.equinox.io/c/bNyj1mQVY4c/ngrok-v3-stable-linux-amd64.tgz
Resolving bin.equinox.io (bin.equinox.io)... 35.71.179.82, 13.248.244.96, 75.2.60.68, ...
Connecting to bin.equinox.io (bin.equinox.io)|35.71.179.82|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 11284206 (11M) [application/octet-stream]
Saving to: 'ngrok-v3-stable-linux-amd64.tgz.1'

ngrok-v3-stable-linux-amd64.t 100%[=====] 10.76M --.-KB/s in 0.04s

2025-12-10 07:59:22 (246 MB/s) - 'ngrok-v3-stable-linux-amd64.tgz.1' saved [11284206/11284206]
```

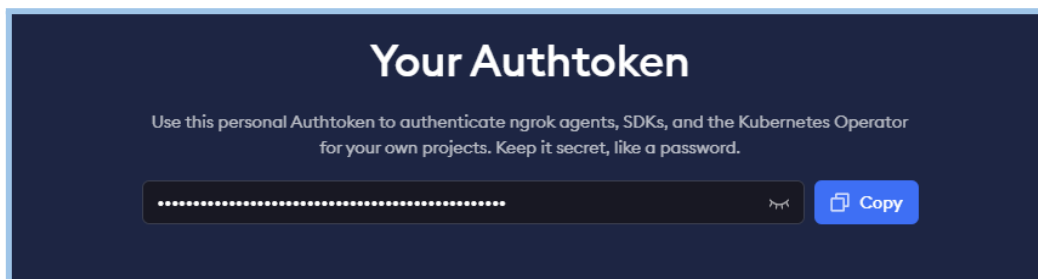
2. Continuamos con, "`tar -xvzf ngrok-v3-stable-linux-amd64.tgz`", que nos ayudará a sacar el programa real.

```
ubuntu@ip-172-31-71-149:~$ tar -xvzf ngrok-v3-stable-linux-amd64.tgz
ngrok
```

3. Por último movemos el programa a la carpeta de herramientas del sistema, "`sudo mv ngrok /usr/local/bin/`".

```
ubuntu@ip-172-31-71-149:~$ sudo mv ngrok /usr/local/bin/
```

4. Para poder hacerlo accesible necesitamos un token ngrok, lo conseguiremos en la web, "[ngrok | API Gateway, Kubernetes Ingress, Webhook Gateway](#)".



5. Una vez tengamos el token lo pegaremos en el siguiente comando de configuración, "`ngrok config add-authtoken TU_TOKEN_AQUI`".

```
ubuntu@ip-172-31-71-149:~$ ngrok config add-authtoken 34JwA4WP4jVFSMEe0rESXBTw024_4AAyh9cD8wtdR2TkWzDxt
Authtoken saved to configuration file: /home/ubuntu/.config/ngrok/ngrok.yml
```


6. Ahora iniciaremos ngrok, "ngrok http 80".

```
ngrok (ct)
▲ Free Users: Agents ≤3.18.x stop connecting 12/17/25. Update or upgrade: https://ngrok.com/pricing

Session Status      online
Account             lucianodiestefano12@gmail.com (Plan: Free)
Version             3.34.0
Region              United States (us)
Latency             13ms
Web Interface       http://127.0.0.1:4040
Forwarding           https://semiarticulately-agley-nicolle.ngrok-free.dev -> http://localhost:80

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

7. Ahora actualizaremos las URLs, primero usaremos, "cat ~/wordpress-credentials.txt" para poder ver las credenciales, posteriormente accederemos a MySQL, "mysql -u wpuser -p -D wordpress".

```
ubuntu@ip-172-31-71-149:~$ cat ~/wordpress-credentials.txt
=== CREDENCIALES DE WORDPRESS ===
Base de datos: wordpress
Usuario BD: wpuser

Contraseña BD: 5hBZqYonIqgfktP0
Usuario root MySQL: root
Contraseña root MySQL: FFyADr/UkPD2iRe0
Acceso local: http://localhost
Acceso remoto: (se configurará con ngrok)
ubuntu@ip-172-31-71-149:~$ mysql -u wp_user -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 15
Server version: 8.0.44-0ubuntu0.24.04.2 (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.
```

8. Insertamos lo siguiente para reemplazar las URLs.

```
mysql> UPDATE wp_options SET option_value='https://abc123def456.ngrok-free.app'  
-> WHERE option_name='siteurl';  
Query OK, 1 row affected (0.01 sec)  
Rows matched: 1 Changed: 1 Warnings: 0  
  
mysql> UPDATE wp_options SET option_value='https://abc123def456.ngrok-free.app'  
-> WHERE option_name='home';  
Query OK, 1 row affected (0.01 sec)  
Rows matched: 1 Changed: 1 Warnings: 0  
  
mysql> EXIT;  
Bye
```

9. Probamos el acceso remoto, "<https://tu-url.ngrok-free.app>".

diesPágina de ejemplo

Blog

¡Hola, mundo!

Te damos la bienvenida a WordPress. Esta es tu primera entrada. Edítala o bórrala, ¡luego empieza a escribir!

9 de diciembre de 2025