# 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

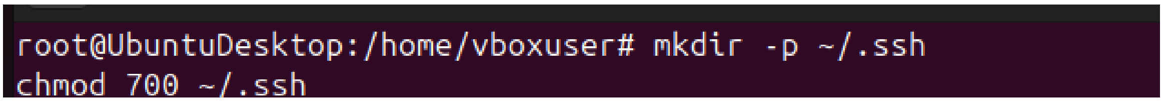
---

## PARTE 0: PREPARACIÓN DEL ENTORNO LOCAL

### 0.1 – Crear directorio SSH

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

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

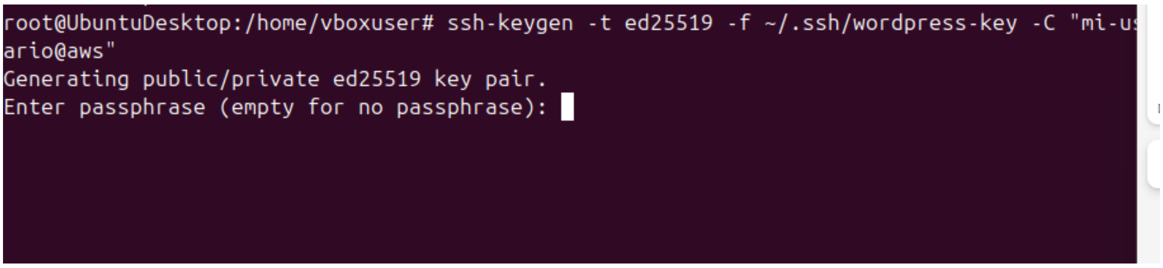


```
root@UbuntuDesktop:/home/vboxuser# mkdir -p ~/.ssh
chmod 700 ~/.ssh
```

### 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"
```

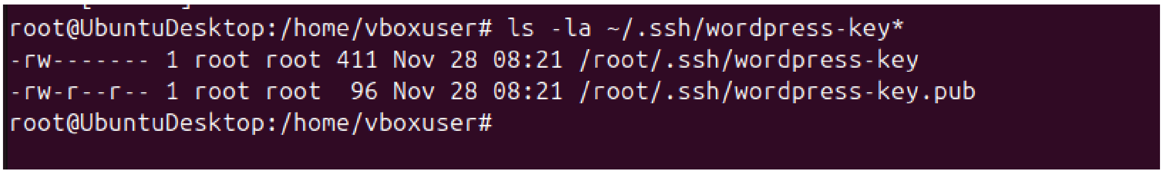


```
root@UbuntuDesktop:/home/vboxuser# ssh-keygen -t ed25519 -f ~/.ssh/wordpress-key -C "mi-us
ario@aws"
Generating public/private ed25519 key pair.
Enter passphrase (empty for no passphrase):
```

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

Verifica que se creó correctamente:

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

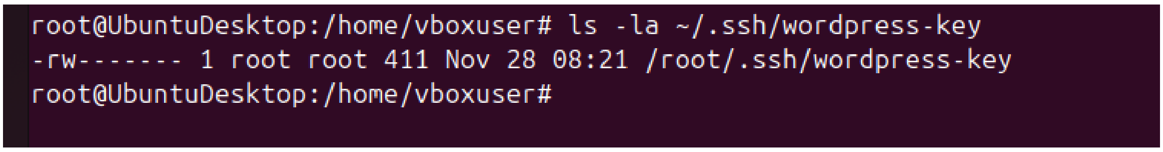


```
root@UbuntuDesktop:/home/vboxuser# ls -la ~/.ssh/wordpress-key*
-rw------- 1 root root 411 Nov 28 08:21 /root/.ssh/wordpress-key
-rw-r--r-- 1 root root  96 Nov 28 08:21 /root/.ssh/wordpress-key.pub
root@UbuntuDesktop:/home/vboxuser#
```

Deberías ver dos archivos:

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

### 0.3 – Ajustar permisos de la clave privada



```
root@UbuntuDesktop:/home/vboxuser# ls -la ~/.ssh/wordpress-key
-rw------- 1 root root 411 Nov 28 08:21 /root/.ssh/wordpress-key
root@UbuntuDesktop:/home/vboxuser#
```

chmod 400 ~/.ssh/wordpress-key

ls -la ~/.ssh/wordpress-key

La salida debe mostrar: **-r--------**

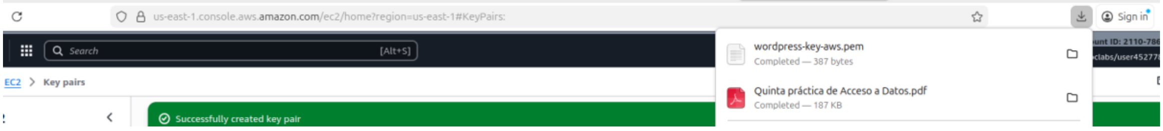
---

## 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: **Ed2551G** (*o RSA si tu AWS no soporta Ed2551S*)
5. File format: **.pem**
6. Clic en "Create key pair"

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



us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#KeyPairs:

wordpress-key-aws.pem
Completed — 387 bytes

Quinta práctica de Acceso a Datos.pdf
Completed — 187 KB

Successfully created key pair

### 1.2 – Trasferir la clave descargada a WSL :

**No hay que transferir nada simplemente me meti a aws desde la mv y descargué lla pem desde ahi**

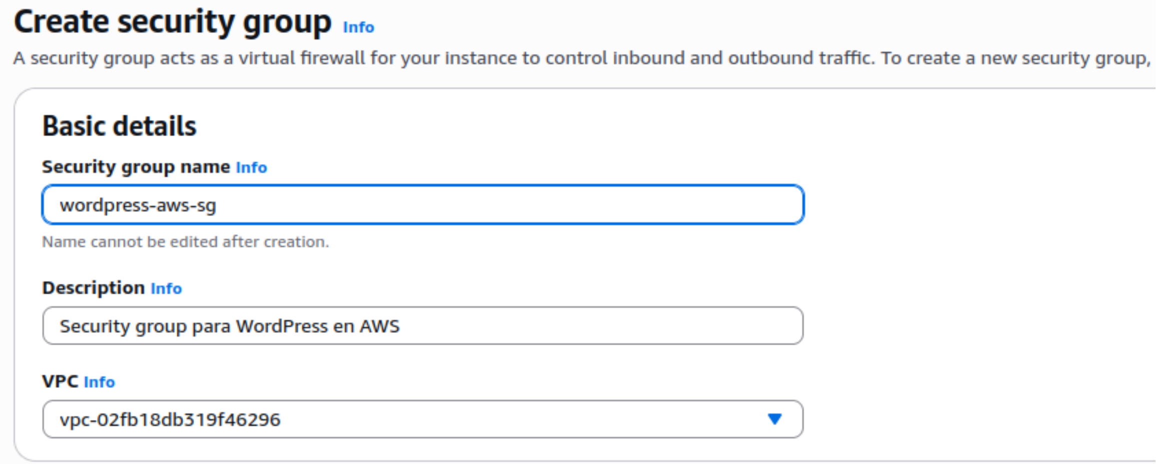
Si descargaste el archivo en Windows:

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

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

### 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 Info

A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. To create a new security group,

Basic details

Security group name Info

wordpress-aws-sg

Name cannot be edited after creation.

Description Info

Security group para WordPress en AWS

VPC Info

vpc-02fb18db319f46296

### 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**

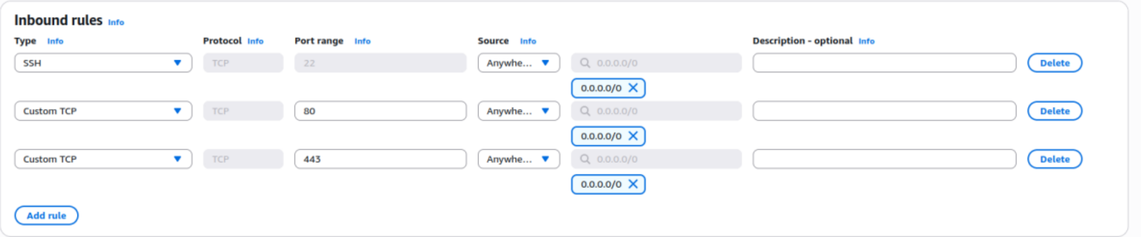
**Regla 2: HTTP (*ngrok*)**

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

**Regla 3: HTTPS (*ngrok*)**

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

Guarda el Security Group.



Inbound rules

| Type | Protocol | Port range | Source | | Description - optional | |
|---|---|---|---|---|---|---|
| SSH | TCP | 22 | Anywhe... | 0.0.0.0/0 | | Delete |
| | | | | 0.0.0.0/0 ✕ | | |
| Custom TCP | TCP | 80 | Anywhe... | 0.0.0.0/0 | | Delete |
| | | | | 0.0.0.0/0 ✕ | | |
| Custom TCP | TCP | 443 | Anywhe... | 0.0.0.0/0 | | Delete |
| | | | | 0.0.0.0/0 ✕ | | |

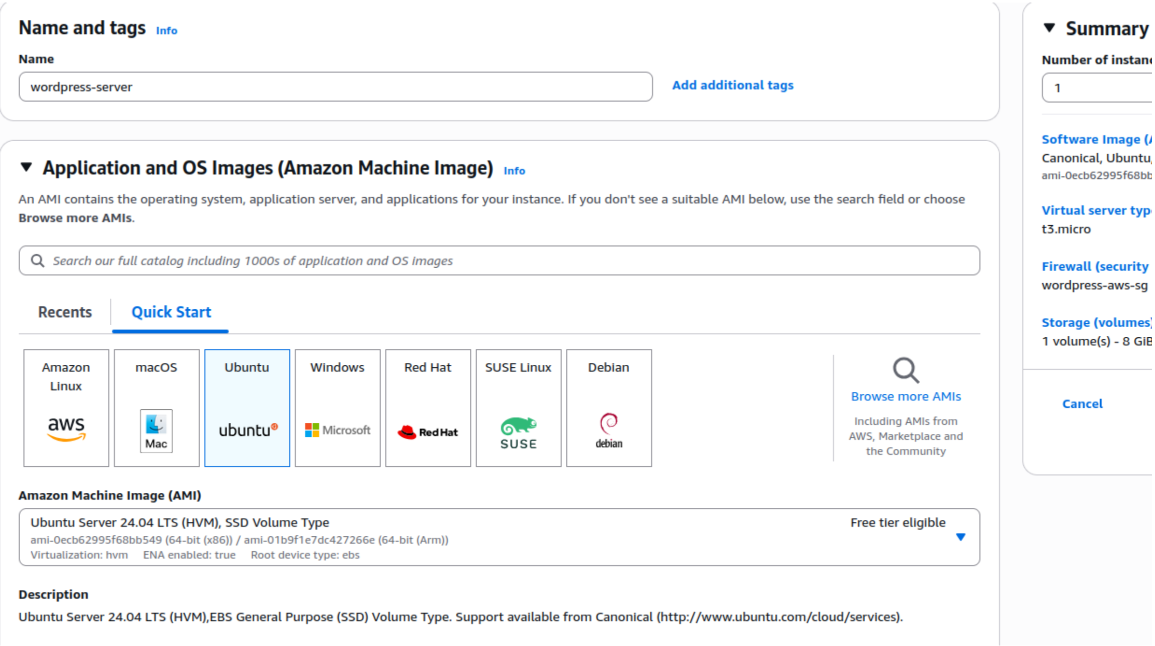
## 1.5 – Crear instancia EC2

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"



Name and tags

Name: wordpress-server

Application and OS Images (Amazon Machine Image)

An AMI contains the operating system, application server, and applications for your instance. If you don't see a suitable AMI below, use the search field or choose **Browse more AMIs**.

Amazon Machine Image (AMI)

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type — Free tier eligible
ami-0ecb62995f68bb549 (64-bit (x86)) / ami-01b9f1e7dc427266e (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs

Description

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

Summary

Number of instanc: 1

Software Image (/: Canonical, Ubuntu, ami-0ecb62995f68bb

Virtual server typ: t3.micro

Firewall (security: wordpress-aws-sg

Storage (volumes): 1 volume(s) - 8 GiE

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

### 1.6 – Obtener la IP pública

1. Selecciona la instancia

2. Copia la "Public IPv4 address" (*ej: 54.123.45.c7*)

---

## PARTE 2: CONEXIÓN SSH DESDE WSL A AWS

### 2.1 – Conectar a la instancia

Reemplaza TU-IP-PUBLICA con la IP obtenida:

En este paso me dio un fallo porque no me coipaba el .pem en el directorio ssh y he usado este comando :

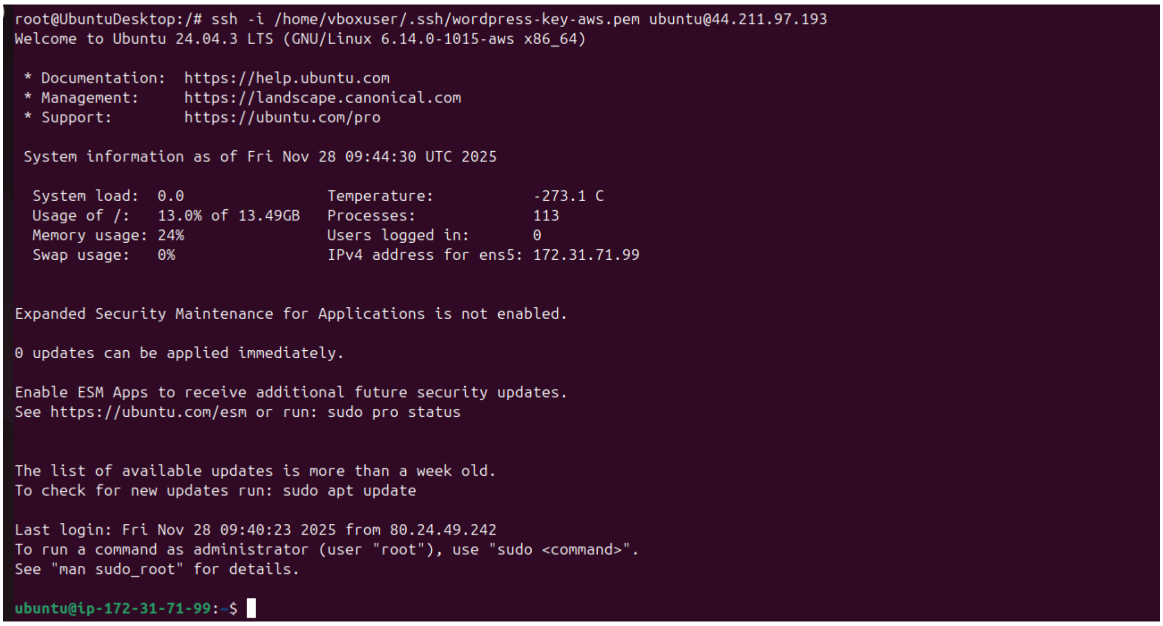
```
cp /home/vboxuser/Downloads/wordpress-key-aws.pem /home/vboxuser/.ssh/
```

```
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**.



```
root@UbuntuDesktop:/# ssh -i /home/vboxuser/.ssh/wordpress-key-aws.pem ubuntu@44.211.97.193
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 09:44:30 UTC 2025

  System load:  0.0                Temperature:           -273.1 C
  Usage of /:   13.0% of 13.49GB   Processes:             113
  Memory usage: 24%                Users logged in:       0
  Swap usage:   0%                 IPv4 address for ens5: 172.31.71.99


Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

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


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

Last login: Fri Nov 28 09:40:23 2025 from 80.24.49.242
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-71-99:~$
```

### 2.2 – Verificar conexión

Una vez conectado, deberías ver algo como:

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

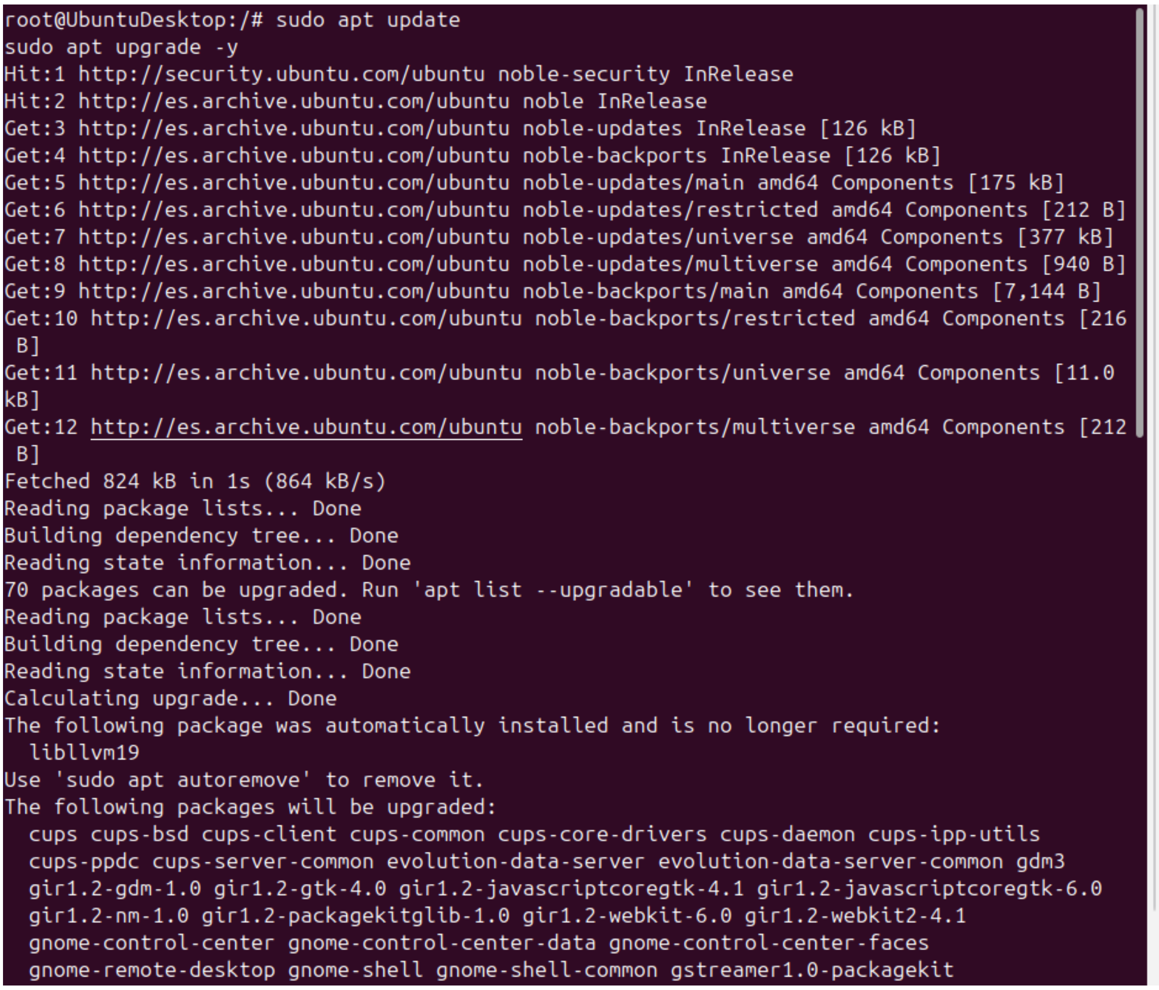
---

## PARTE 3: INSTALACIÓN BASE DEL SERVIDOR (EN AWS)

### 3.1 – Actualizar el sistema

sudo apt update

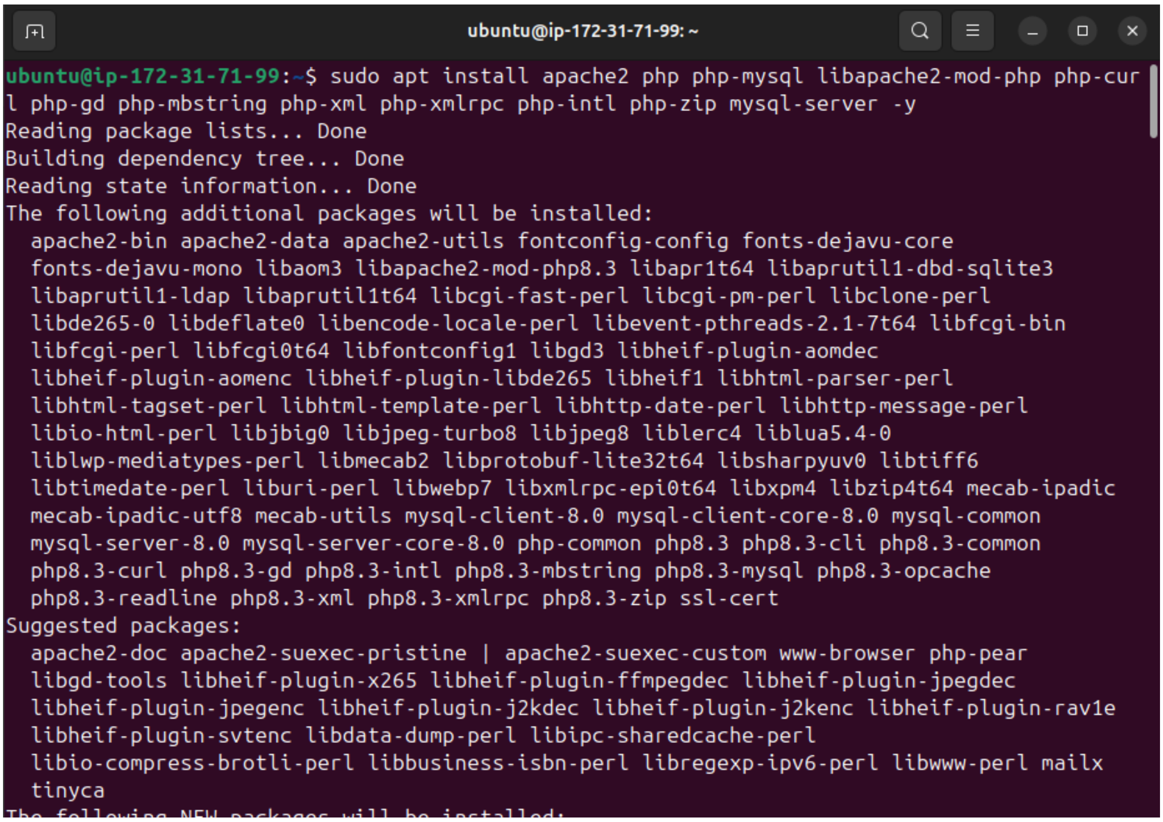
sudo apt upgrade –y



```
root@UbuntuDesktop:/# sudo apt update
sudo apt upgrade -y
Hit:1 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:2 http://es.archive.ubuntu.com/ubuntu noble InRelease
Get:3 http://es.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:4 http://es.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:5 http://es.archive.ubuntu.com/ubuntu noble-updates/main amd64 Components [175 kB]
Get:6 http://es.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Components [212 B]
Get:7 http://es.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Components [377 kB]
Get:8 http://es.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Components [940 B]
Get:9 http://es.archive.ubuntu.com/ubuntu noble-backports/main amd64 Components [7,144 B]
Get:10 http://es.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 Components [216
 B]
Get:11 http://es.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Components [11.0
kB]
Get:12 http://es.archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 Components [212
 B]
Fetched 824 kB in 1s (864 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
70 packages can be upgraded. Run 'apt list --upgradable' to see them.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following package was automatically installed and is no longer required:
  libllvm19
Use 'sudo apt autoremove' to remove it.
The following packages will be upgraded:
  cups cups-bsd cups-client cups-common cups-core-drivers cups-daemon cups-ipp-utils
  cups-ppdc cups-server-common evolution-data-server evolution-data-server-common gdm3
  gir1.2-gdm-1.0 gir1.2-gtk-4.0 gir1.2-javascriptcoregtk-4.1 gir1.2-javascriptcoregtk-6.0
  gir1.2-nm-1.0 gir1.2-packagekitglib-1.0 gir1.2-webkit-6.0 gir1.2-webkit2-4.1
  gnome-control-center gnome-control-center-data gnome-control-center-faces
  gnome-remote-desktop gnome-shell gnome-shell-common gstreamer1.0-packagekit
```

## 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



```
ubuntu@ip-172-31-71-99:~$ sudo apt install apache2 php php-mysql libapache2-mod-php php-cur
l 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 libjbig0 libjpeg-turbo8 libjpeg8 liblerc4 liblua5.4-0
  liblwp-mediatypes-perl libmecab2 libprotobuf-lite32t64 libsharpyuv0 libtiff6
  libtimedate-perl liburi-perl libwebp7 libxmlrpc-epi0t64 libxpm4 libzip4t64 mecab-ipadic
  mecab-ipadic-utf8 mecab-utils mysql-client-8.0 mysql-client-core-8.0 mysql-common
  mysql-server-8.0 mysql-server-core-8.0 php-common php8.3 php8.3-cli php8.3-common
  php8.3-curl php8.3-gd php8.3-intl php8.3-mbstring php8.3-mysql php8.3-opcache
  php8.3-readline php8.3-xml php8.3-xmlrpc php8.3-zip ssl-cert
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom www-browser php-pear
  libgd-tools libheif-plugin-x265 libheif-plugin-ffmpegdec libheif-plugin-jpegdec
  libheif-plugin-jpegenc libheif-plugin-j2kdec libheif-plugin-j2kenc libheif-plugin-rav1e
  libheif-plugin-svtenc libdata-dump-perl libipc-sharedcache-perl
  libio-compress-brotli-perl libbusiness-isbn-perl libregexp-ipv6-perl libwww-perl mailx
  tinyca
The following NEW packages will be installed:
```

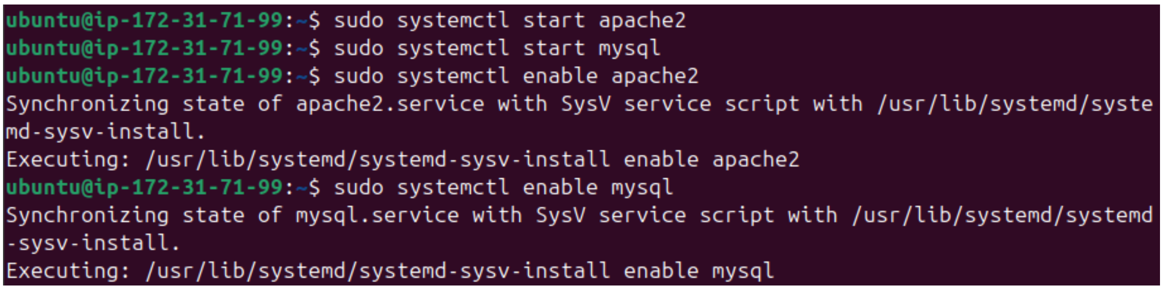
### 3.3 – Iniciar servicios

sudo systemctl start apache2

sudo systemctl start mysql

sudo systemctl enable apache2

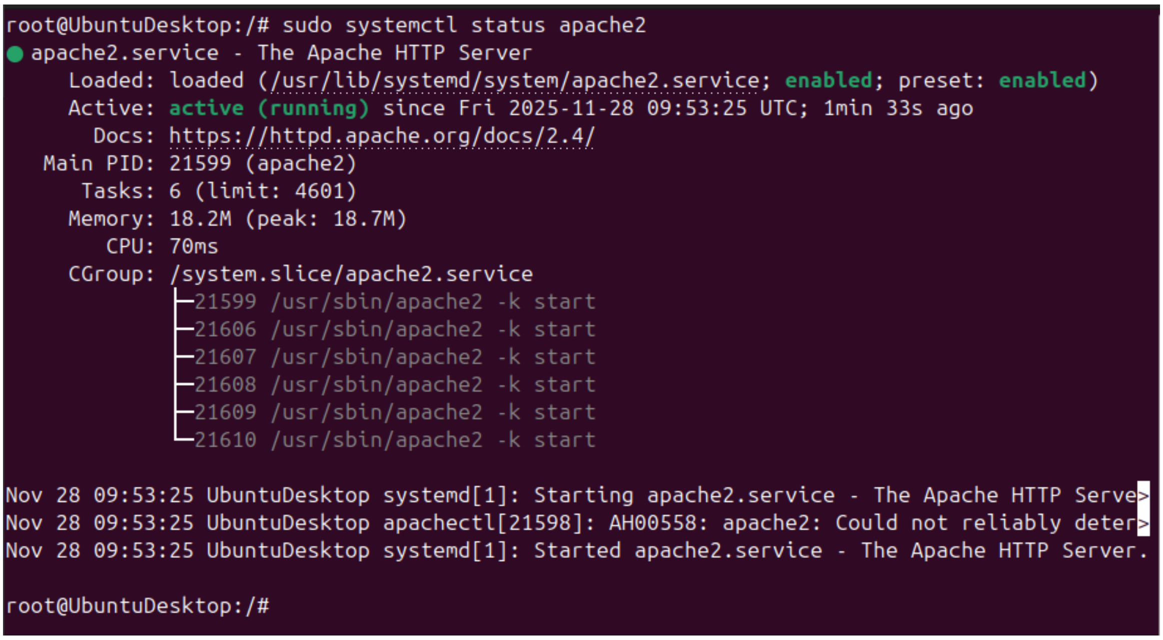
sudo systemctl enable mysql



```
ubuntu@ip-172-31-71-99:~$ sudo systemctl start apache2
ubuntu@ip-172-31-71-99:~$ sudo systemctl start mysql
ubuntu@ip-172-31-71-99:~$ sudo systemctl enable apache2
Synchronizing state of apache2.service with SysV service script with /usr/lib/systemd/syste
md-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable apache2
ubuntu@ip-172-31-71-99:~$ sudo systemctl enable mysql
Synchronizing state of mysql.service with SysV service script with /usr/lib/systemd/systemd
-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable mysql
```

### 3.4 – Verificar servicios

sudo systemctl status apache2

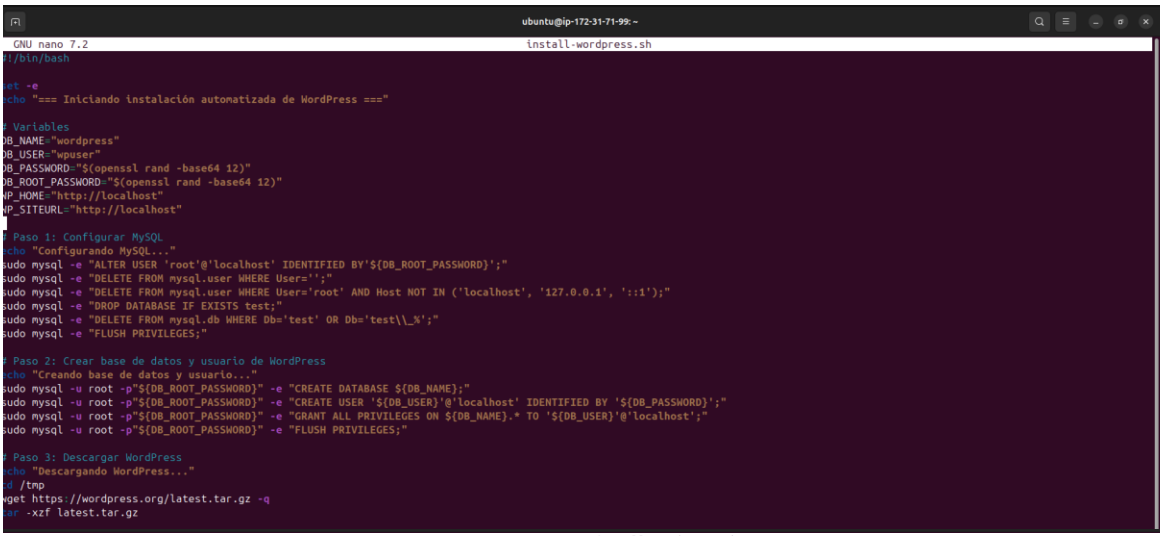


```
root@UbuntuDesktop:/# sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: enabled)
     Active: active (running) since Fri 2025-11-28 09:53:25 UTC; 1min 33s ago
       Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 21599 (apache2)
      Tasks: 6 (limit: 4601)
     Memory: 18.2M (peak: 18.7M)
        CPU: 70ms
     CGroup: /system.slice/apache2.service
             ├─21599 /usr/sbin/apache2 -k start
             ├─21606 /usr/sbin/apache2 -k start
             ├─21607 /usr/sbin/apache2 -k start
             ├─21608 /usr/sbin/apache2 -k start
             ├─21609 /usr/sbin/apache2 -k start
             └─21610 /usr/sbin/apache2 -k start

Nov 28 09:53:25 UbuntuDesktop systemd[1]: Starting apache2.service - The Apache HTTP Serve>
Nov 28 09:53:25 UbuntuDesktop apachectl[21598]: AH00558: apache2: Could not reliably deter>
Nov 28 09:53:25 UbuntuDesktop systemd[1]: Started apache2.service - The Apache HTTP Server.

root@UbuntuDesktop:/#
```

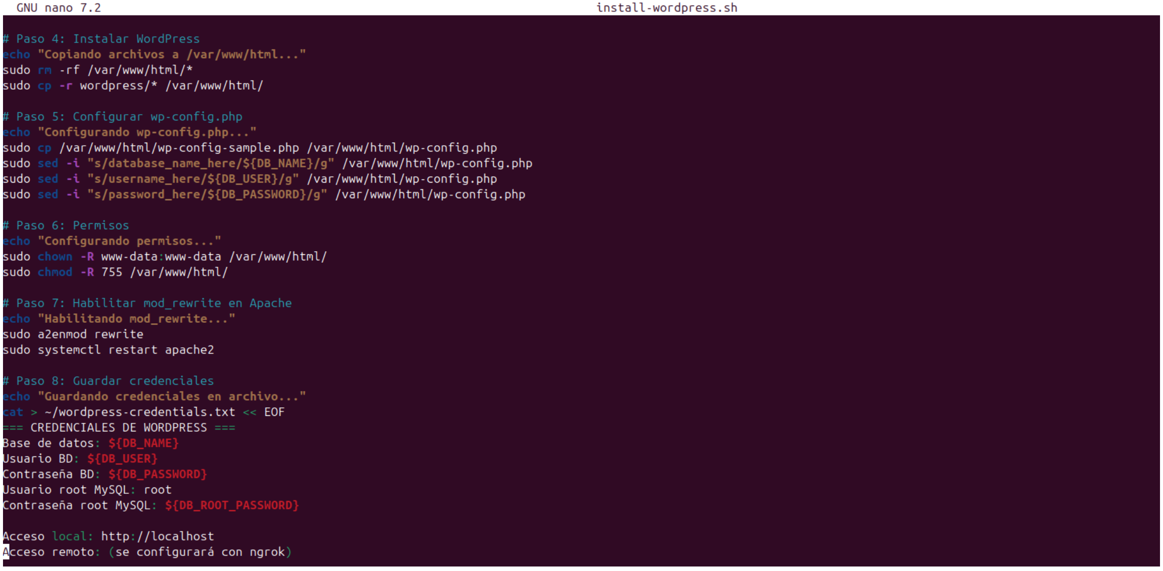
## PARTE 4: SCRIPT DE AUTOMATIZACIÓN DE WORDPRESS

### 4.1 – Crear script de instalación

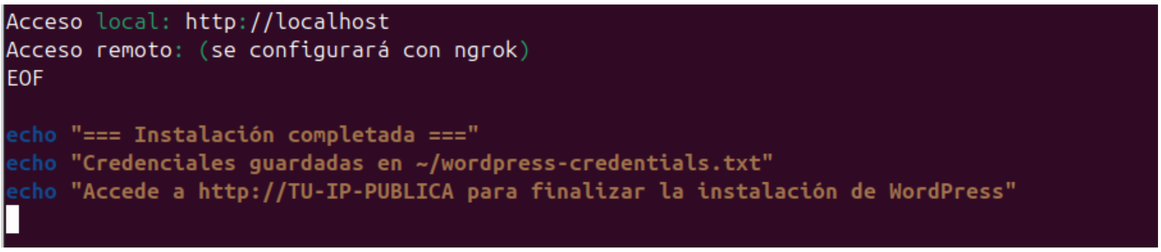
En tu máquina local, crea un archivo llamado install-wordpress.sh:



```
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"

# 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
```



```
GNU nano 7.2                                   install-wordpress.sh

# 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)
```



```
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"
```

*#!/bin/bash*

*set -e*

*echo "=== Iniciando instalación automatizada de WordPress ==="*

*# Variables*

```
DB_NAME="wordpress"

DB_USER="wpuser"

DB_PASSWORD=":(openssl rand -basec4 12)"

DB_ROOT_PASSWORD=":(openssl rand -basec4 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 c: 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"
```

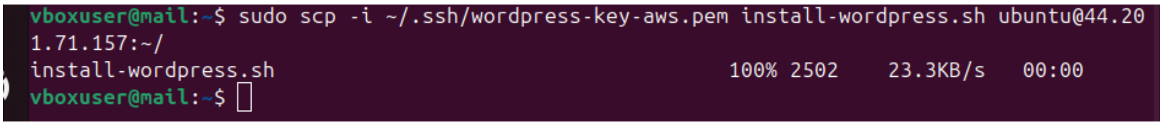
---

## 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:~/
```

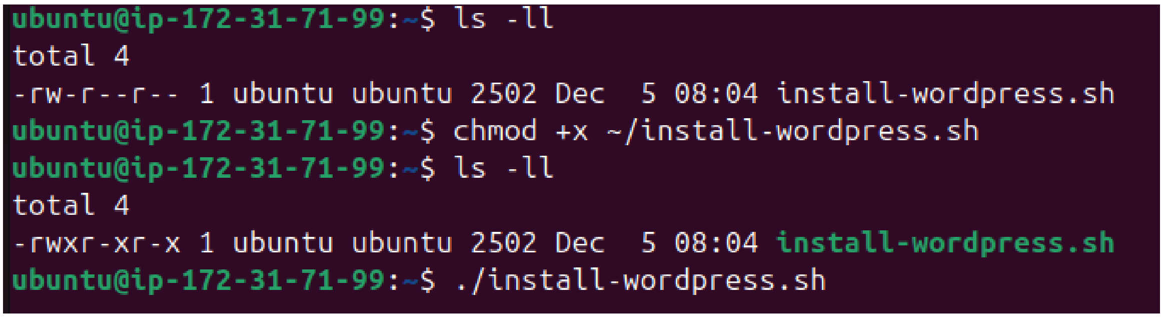


```
vboxuser@mail:~$ sudo scp -i ~/.ssh/wordpress-key-aws.pem install-wordpress.sh ubuntu@44.20
1.71.157:~/
install-wordpress.sh                                          100% 2502     23.3KB/s   00:00
vboxuser@mail:~$ 
```

### 5.2 – Dar permisos de ejecución

En AWS (*dentro de la sesión SSH*):

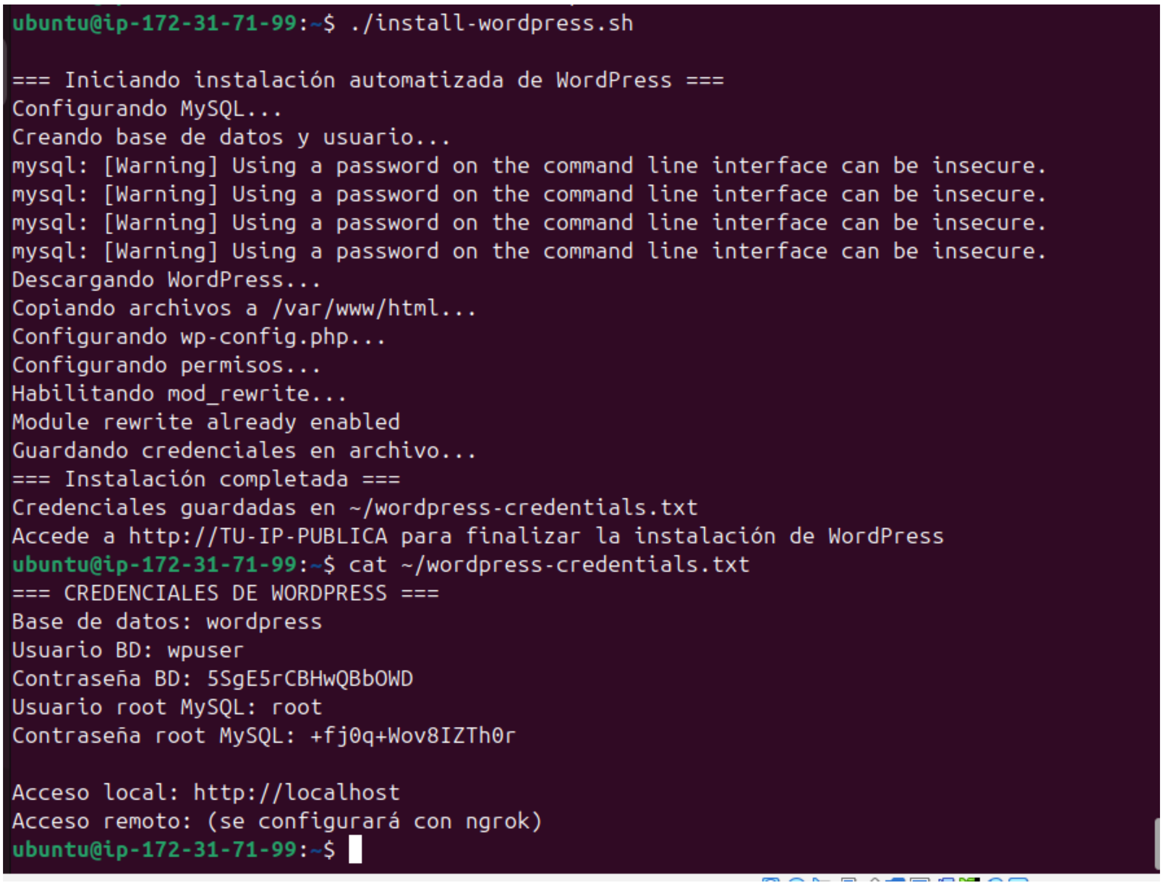
```
chmod +x ~/install-wordpress.sh
```



```
ubuntu@ip-172-31-71-99:~$ ls -ll
total 4
-rw-r--r-- 1 ubuntu ubuntu 2502 Dec  5 08:04 install-wordpress.sh
ubuntu@ip-172-31-71-99:~$ chmod +x ~/install-wordpress.sh
ubuntu@ip-172-31-71-99:~$ ls -ll
total 4
-rwxr-xr-x 1 ubuntu ubuntu 2502 Dec  5 08:04 install-wordpress.sh
ubuntu@ip-172-31-71-99:~$ ./install-wordpress.sh
```

### 5.3 – Ejecutar el script

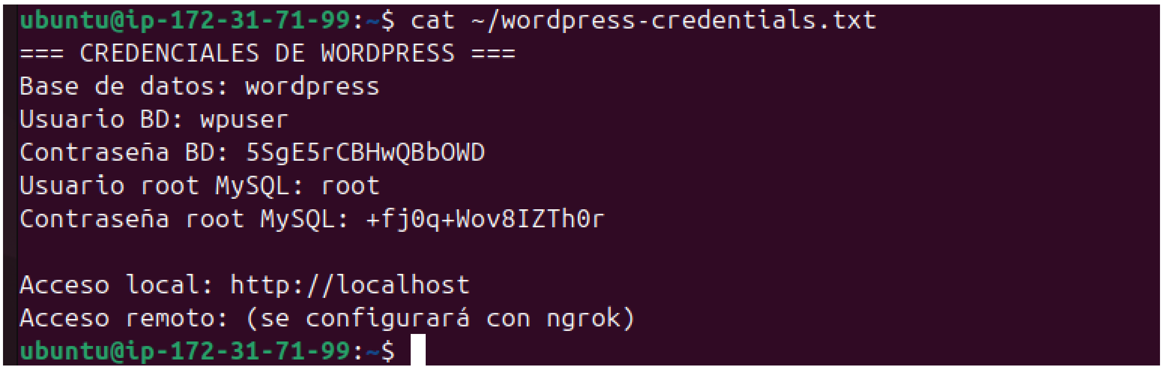
./install-wordpress.sh



```
ubuntu@ip-172-31-71-99:~$ ./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...
Module rewrite already enabled
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-172-31-71-99:~$ cat ~/wordpress-credentials.txt
=== CREDENCIALES DE WORDPRESS ===
Base de datos: wordpress
Usuario BD: wpuser
Contraseña BD: 5SgE5rCBHwQBbOWD
Usuario root MySQL: root
Contraseña root MySQL: +fj0q+Wov8IZTh0r

Acceso local: http://localhost
Acceso remoto: (se configurará con ngrok)
ubuntu@ip-172-31-71-99:~$
```

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

cat ~/wordpress-credentials.txt

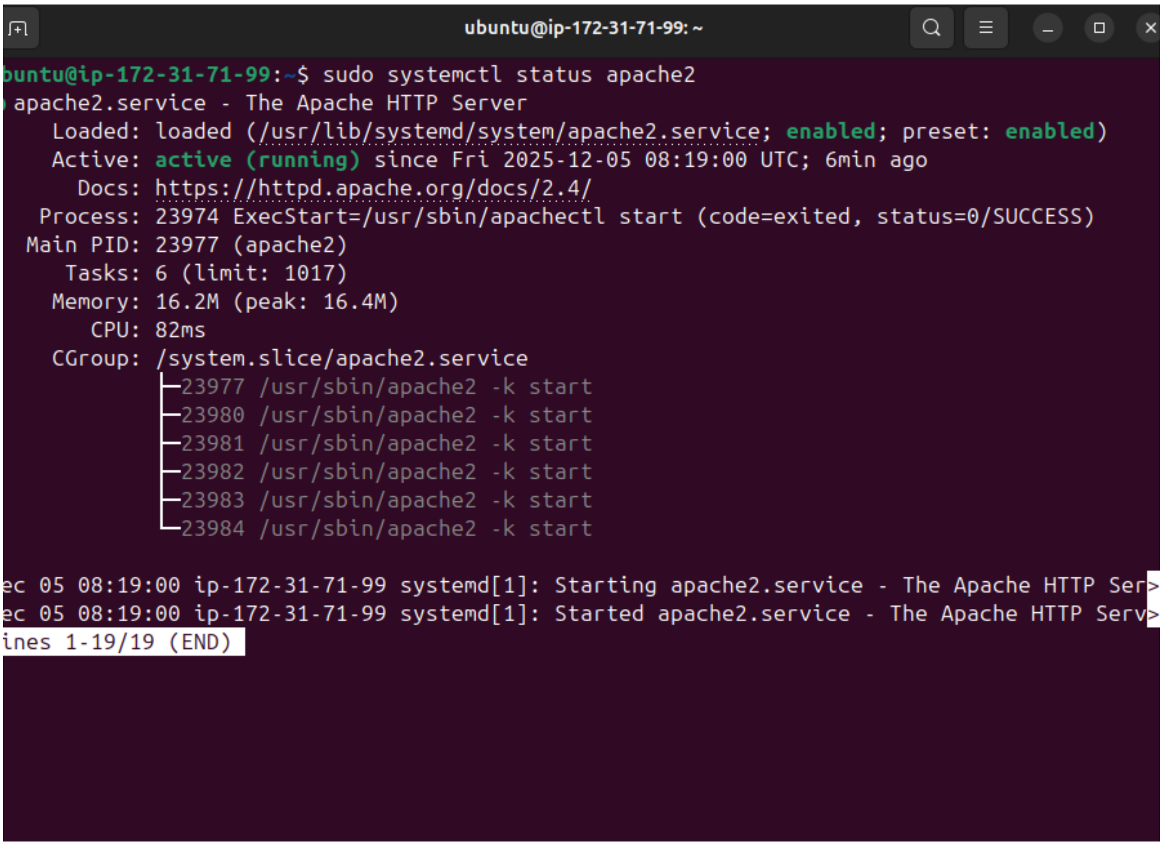


```
ubuntu@ip-172-31-71-99:~$ cat ~/wordpress-credentials.txt
=== CREDENCIALES DE WORDPRESS ===
Base de datos: wordpress
Usuario BD: wpuser
Contraseña BD: 5SgE5rCBHwQBbOWD
Usuario root MySQL: root
Contraseña root MySQL: +fj0q+Wov8IZTh0r

Acceso local: http://localhost
Acceso remoto: (se configurará con ngrok)
ubuntu@ip-172-31-71-99:~$
```

## PARTE 6: VERIFICACIÓN DE INSTALACIÓN

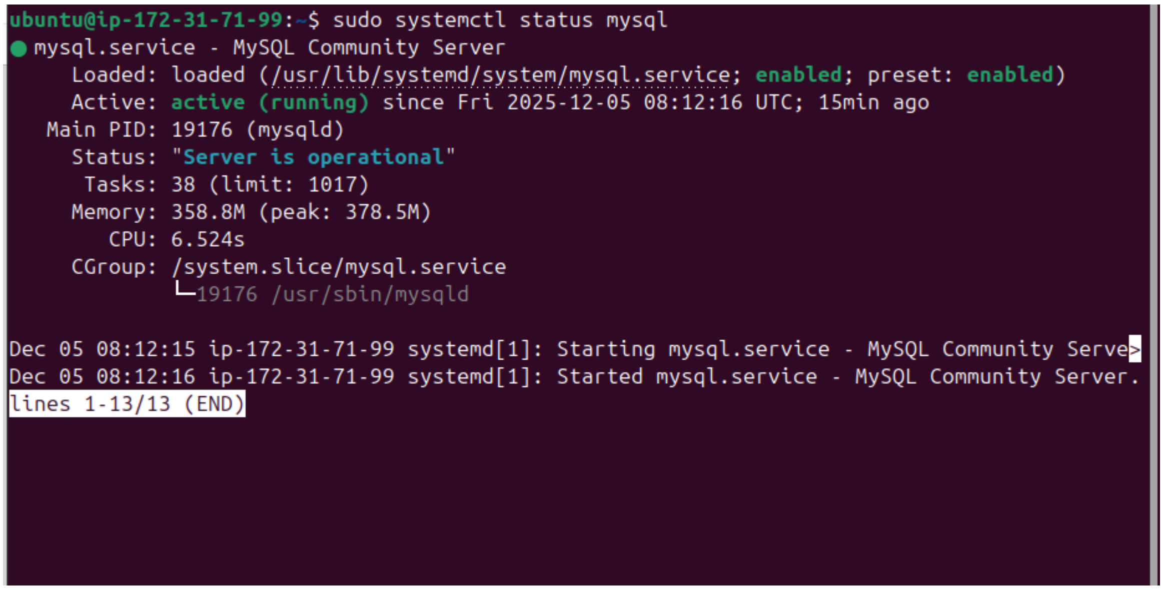
### 6.1 – Verificar servicios

sudo systemctl status apache2



```
buntu@ip-172-31-71-99:~$ 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 08:19:00 UTC; 6min ago
       Docs: https://httpd.apache.org/docs/2.4/
    Process: 23974 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
   Main PID: 23977 (apache2)
      Tasks: 6 (limit: 1017)
     Memory: 16.2M (peak: 16.4M)
        CPU: 82ms
     CGroup: /system.slice/apache2.service
             ├─23977 /usr/sbin/apache2 -k start
             ├─23980 /usr/sbin/apache2 -k start
             ├─23981 /usr/sbin/apache2 -k start
             ├─23982 /usr/sbin/apache2 -k start
             ├─23983 /usr/sbin/apache2 -k start
             └─23984 /usr/sbin/apache2 -k start

ec 05 08:19:00 ip-172-31-71-99 systemd[1]: Starting apache2.service - The Apache HTTP Ser>
ec 05 08:19:00 ip-172-31-71-99 systemd[1]: Started apache2.service - The Apache HTTP Serv>
ines 1-19/19 (END)
```

sudo systemctl status mysql



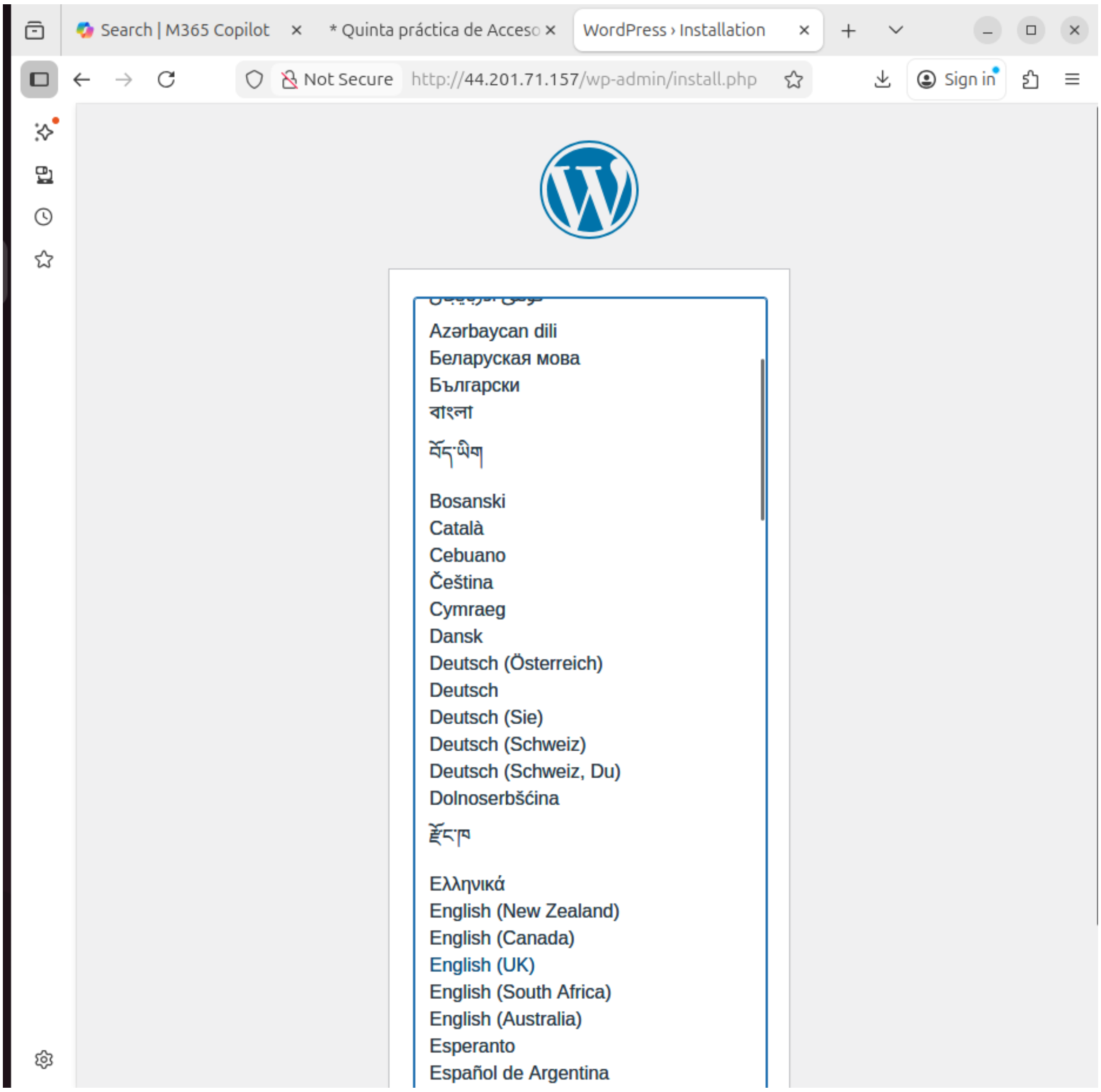
```
ubuntu@ip-172-31-71-99:~$ sudo systemctl status mysql
● mysql.service - MySQL Community Server
     Loaded: loaded (/usr/lib/systemd/system/mysql.service; enabled; preset: enabled)
     Active: active (running) since Fri 2025-12-05 08:12:16 UTC; 15min ago
   Main PID: 19176 (mysqld)
     Status: "Server is operational"
      Tasks: 38 (limit: 1017)
     Memory: 358.8M (peak: 378.5M)
        CPU: 6.524s
     CGroup: /system.slice/mysql.service
             └─19176 /usr/sbin/mysqld

Dec 05 08:12:15 ip-172-31-71-99 systemd[1]: Starting mysql.service - MySQL Community Serve>
Dec 05 08:12:16 ip-172-31-71-99 systemd[1]: Started mysql.service - MySQL Community Server.
lines 1-13/13 (END)
```

### 6.2 – Acceder desde navegador local

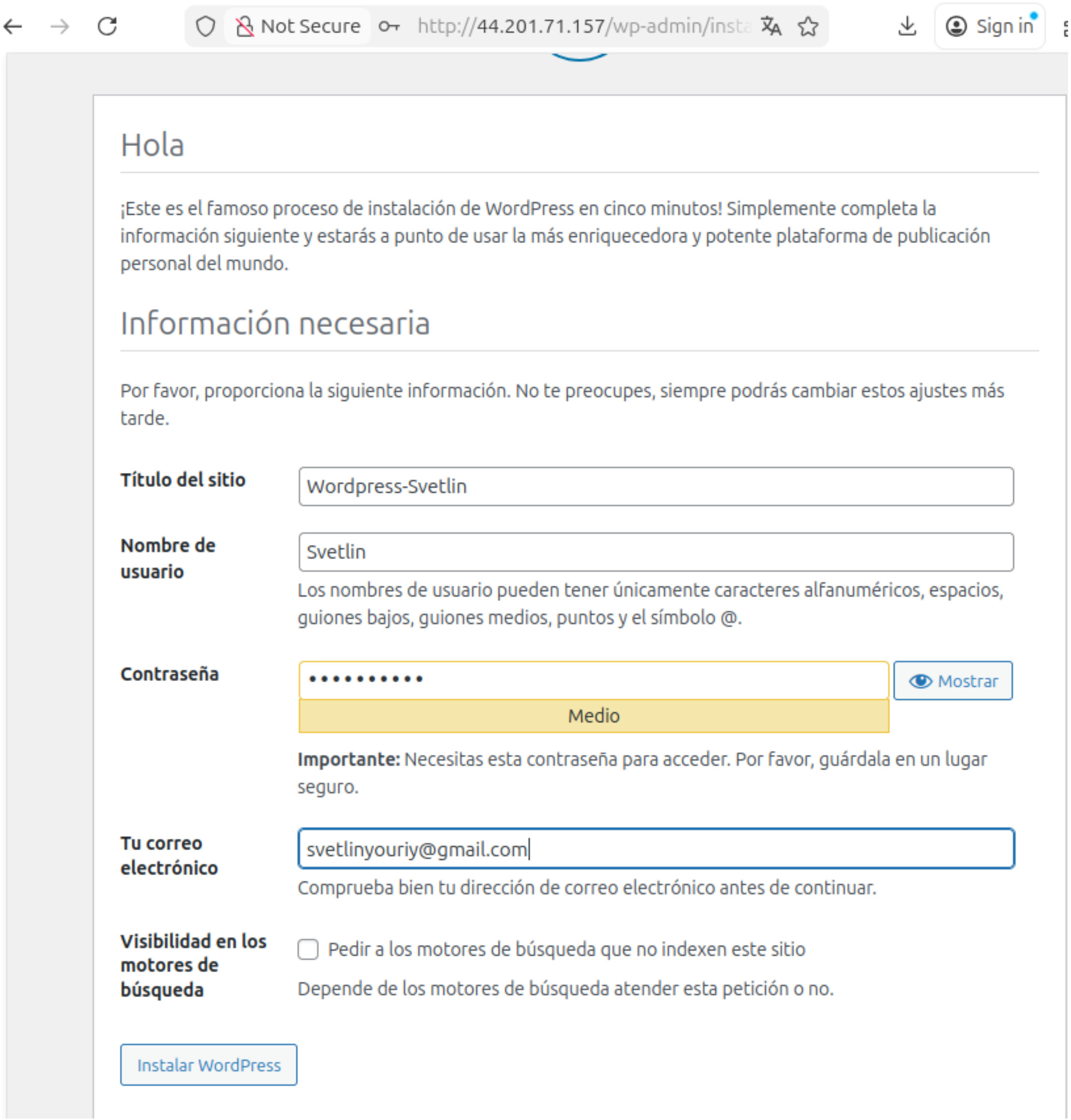
Abre tu navegador y accede a:

http://TU-IP-PUBLICA

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


Search | M365 Copilot
* Quinta práctica de Acceso
WordPress › Installation
Not Secure
http://44.201.71.157/wp-admin/install.php
Sign in

Azərbaycan dili
Беларуская мова
Български
বাংলা
བོད་ཡིག
Bosanski
Català
Cebuano
Čeština
Cymraeg
Dansk
Deutsch (Österreich)
Deutsch
Deutsch (Sie)
Deutsch (Schweiz)
Deutsch (Schweiz, Du)
Dolnoserbšćina
རྫོང་ཁ
Ελληνικά
English (New Zealand)
English (Canada)
English (UK)
English (South Africa)
English (Australia)
Esperanto
Español de Argentina



## 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** Wordpress-Svetlin

**Nombre de usuario** Svetlin

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

**Contraseña** •••••••••• Mostrar

Medio

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

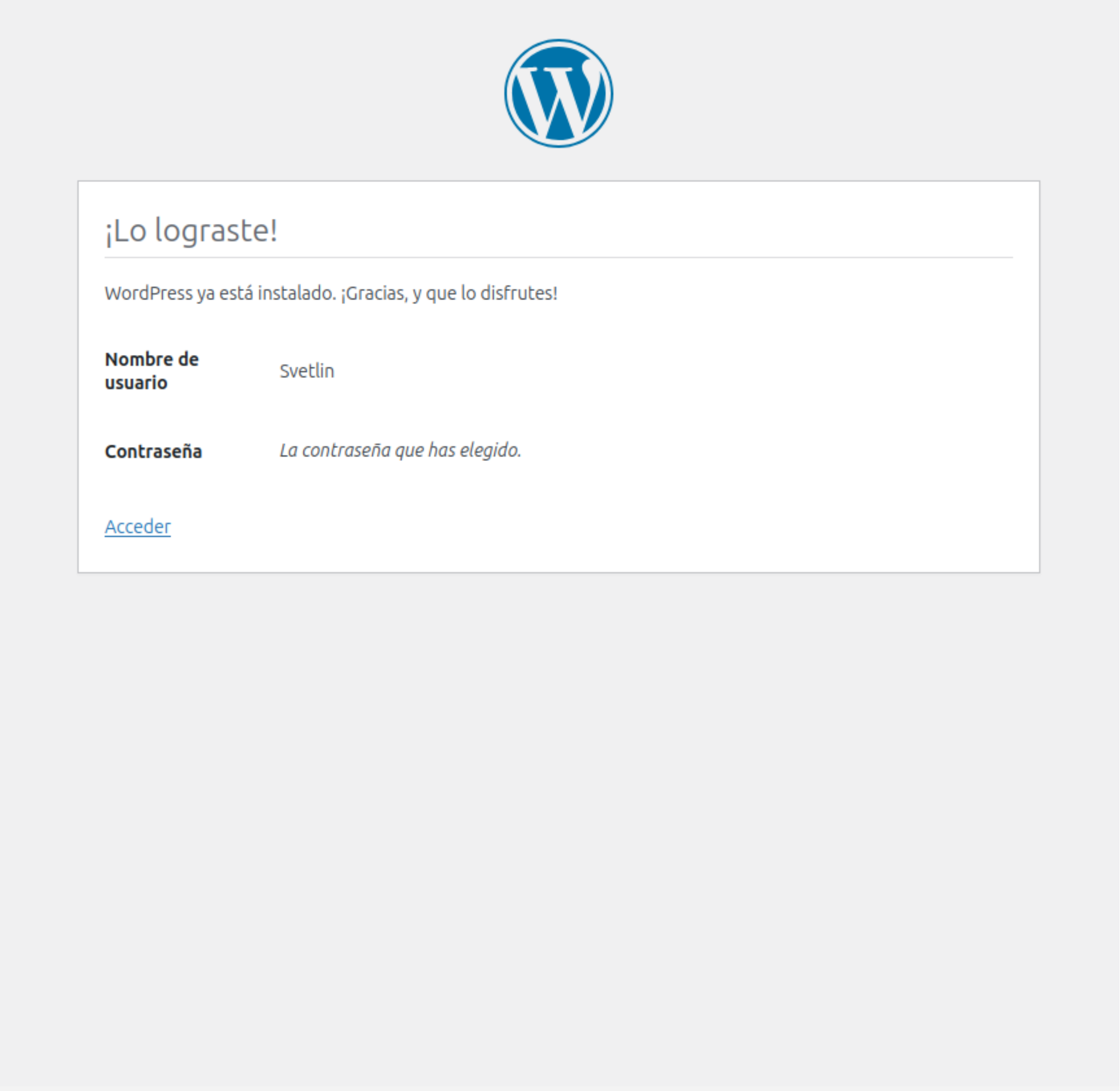
**Tu correo electrónico** svetlinyouriy@gmail.com

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



¡Lo lograste!

WordPress ya está instalado. ¡Gracias, y que lo disfrutes!

| | |
|---|---|
| **Nombre de usuario** | Svetlin |
| **Contraseña** | *La contraseña que has elegido.* |

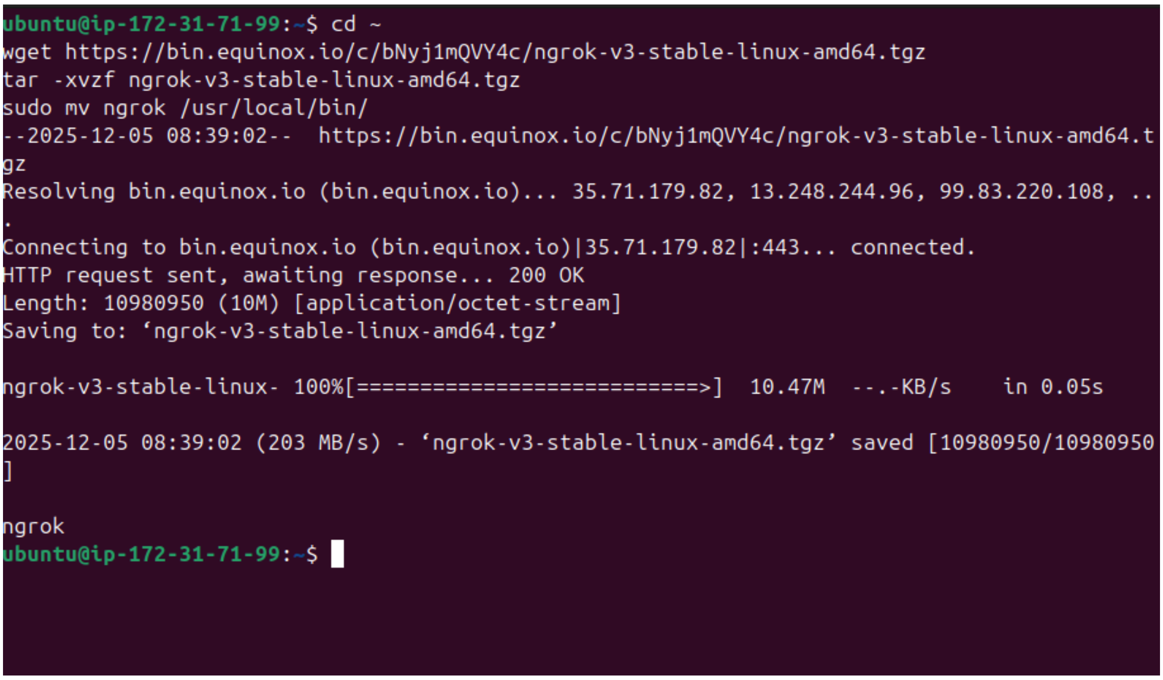
Acceder

# 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/
```



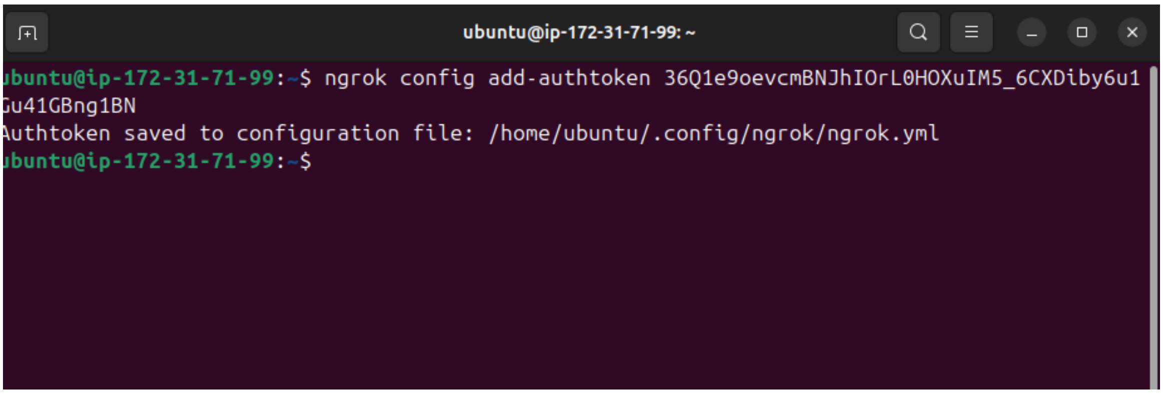
```
ubuntu@ip-172-31-71-99:~$ cd ~
wget https://bin.equinox.io/c/bNyj1mQVY4c/ngrok-v3-stable-linux-amd64.tgz
tar -xvzf ngrok-v3-stable-linux-amd64.tgz
sudo mv ngrok /usr/local/bin/
--2025-12-05 08:39:02--  https://bin.equinox.io/c/bNyj1mQVY4c/ngrok-v3-stable-linux-amd64.t
gz
Resolving bin.equinox.io (bin.equinox.io)... 35.71.179.82, 13.248.244.96, 99.83.220.108, ..
.
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- 100%[===========================>]  10.47M  --.-KB/s    in 0.05s

2025-12-05 08:39:02 (203 MB/s) - ‘ngrok-v3-stable-linux-amd64.tgz’ saved [10980950/10980950
]

ngrok
ubuntu@ip-172-31-71-99:~$ 
```

### 7.2 – Autenticar ngrok

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

Configura tu token (*reemplaza TU_TOKEN*):

W3



```
ubuntu@ip-172-31-71-99:~$ ngrok config add-authtoken 36Q1e9oevcmBNJhIOrL0HOXuIM5_6CXDiby6u1
Gu41GBng1BN
Authtoken saved to configuration file: /home/ubuntu/.config/ngrok/ngrok.yml
ubuntu@ip-172-31-71-99:~$
```

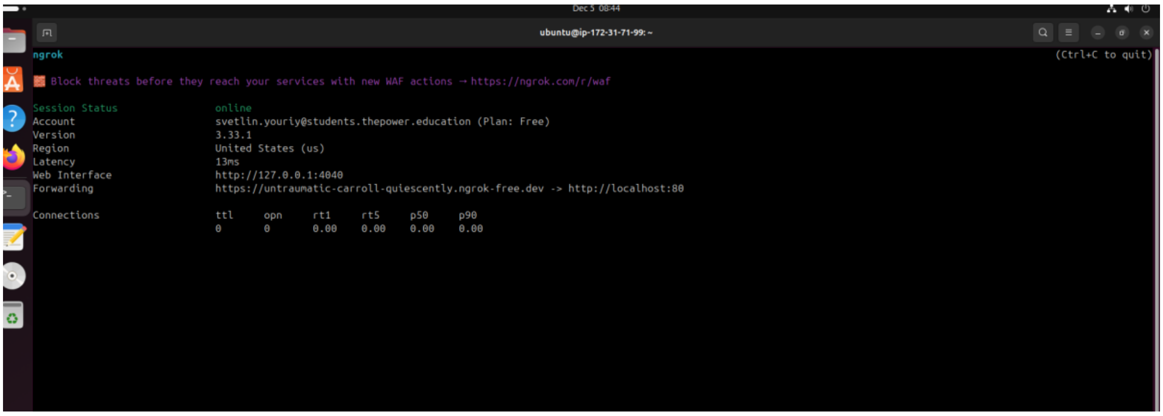
ngrok config add-authtoken TU_TOKEN_AQUI

### 7.3 – Iniciar ngrok

ngrok http 80

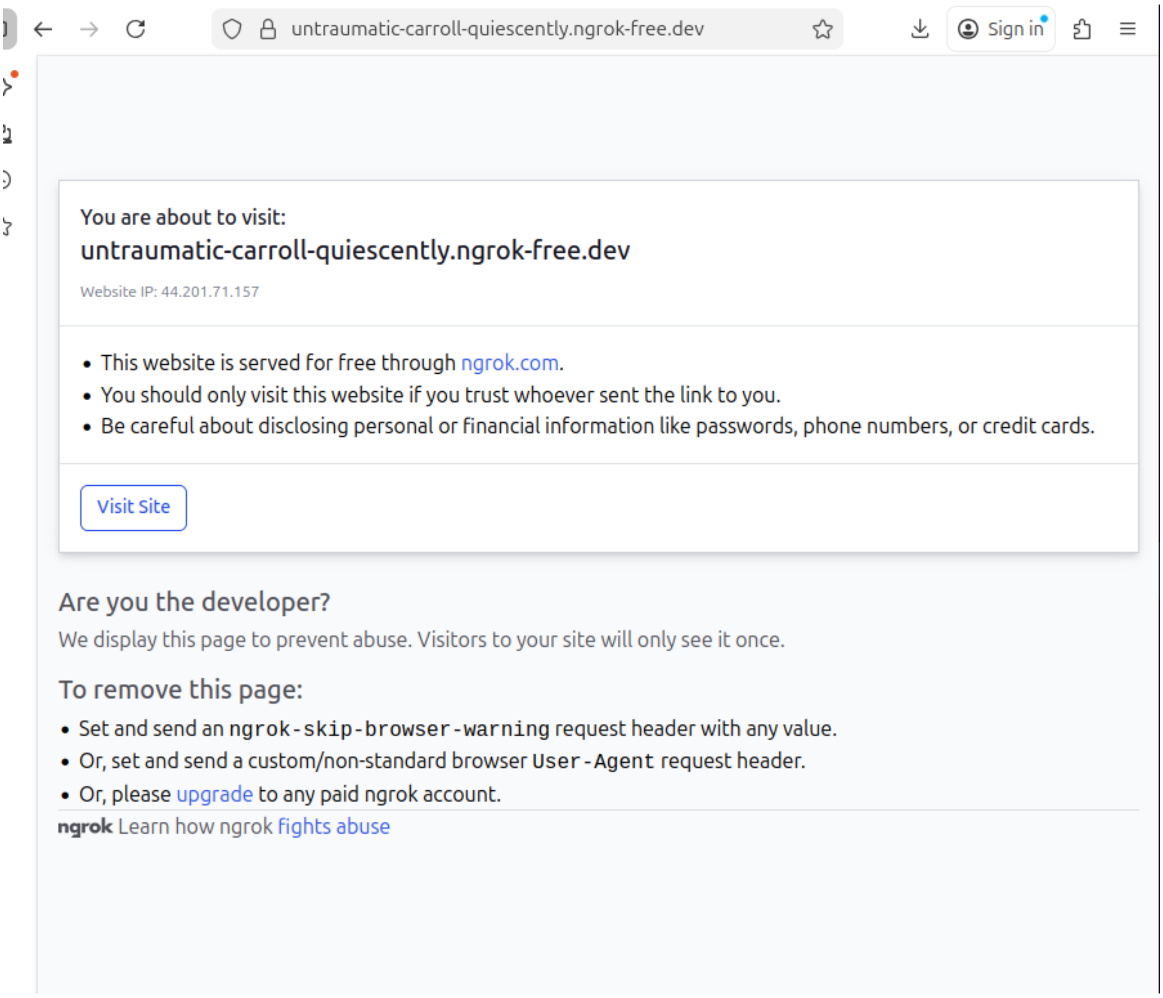
Verás una salida como:

```
ngrok                                    (Ctrl+C to quit)


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
```



```
ngrok                                                                                          (Ctrl+C to quit)

🧱 Block threats before they reach your services with new WAF actions → https://ngrok.com/r/waf

Session Status                online
Account                       svetlin.youriy@students.thepower.education (Plan: Free)
Version                       3.33.1
Region                        United States (us)
Latency                       13ms
Web Interface                 http://127.0.0.1:4040
Forwarding                    https://untraumatic-carroll-quiescently.ngrok-free.dev -> http://localhost:80

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

Copia la URL HTTPS (ej: https://abc123def456.ngrok-free.app)



You are about to visit:
untraumatic-carroll-quiescently.ngrok-free.dev

Website IP: 44.201.71.157

- This website is served for free through ngrok.com.
- You should only visit this website if you trust whoever sent the link to you.
- Be careful about disclosing personal or financial information like passwords, phone numbers, or credit cards.

Visit Site

Are you the developer?

We display this page to prevent abuse. Visitors to your site will only see it once.

To remove this page:

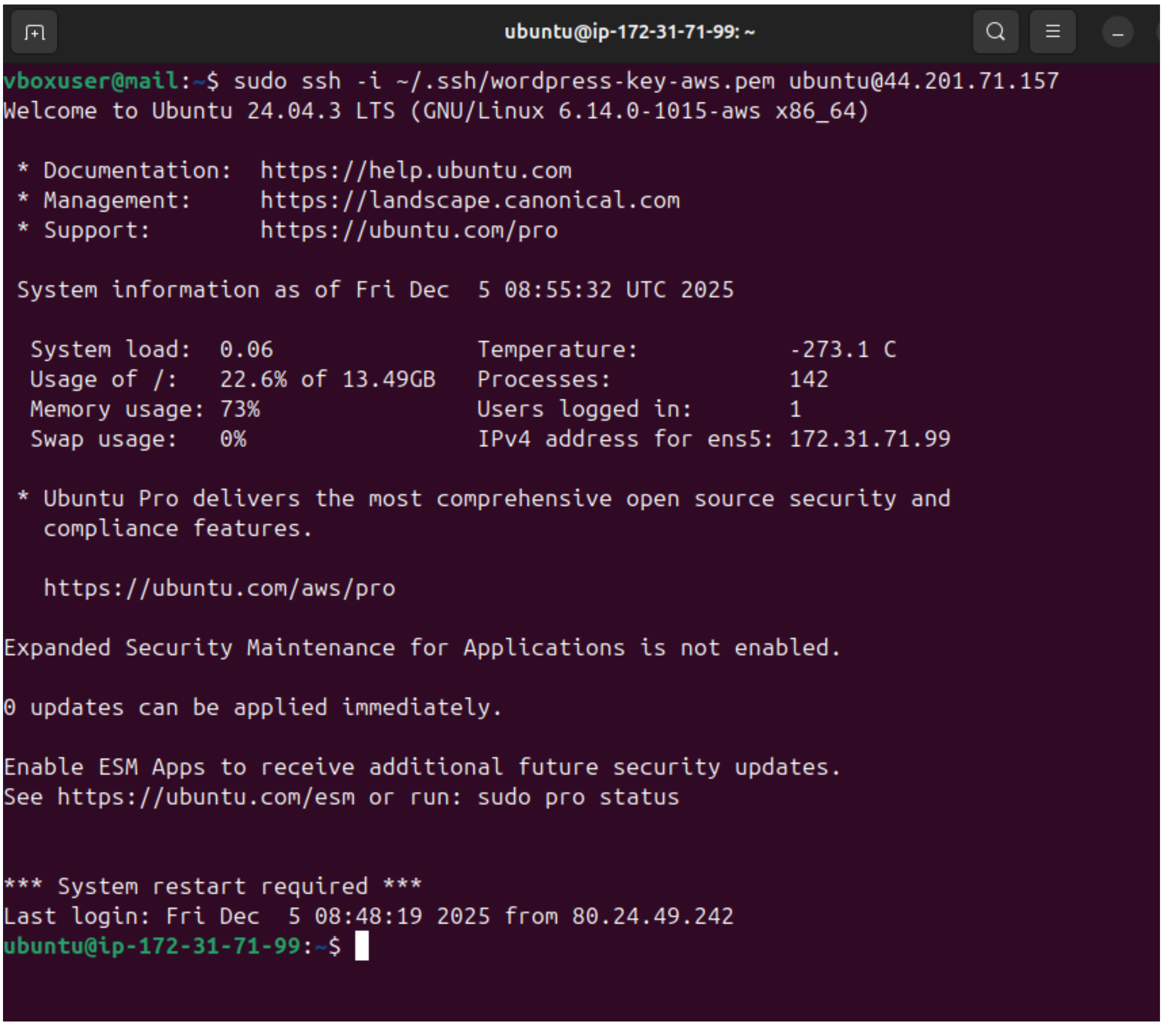
- Set and send an `ngrok-skip-browser-warning` request header with any value.
- Or, set and send a custom/non-standard browser `User-Agent` request header.
- Or, please upgrade to any paid ngrok account.

ngrok Learn how ngrok fights abuse

### 7.4 – Actualizar WordPress

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

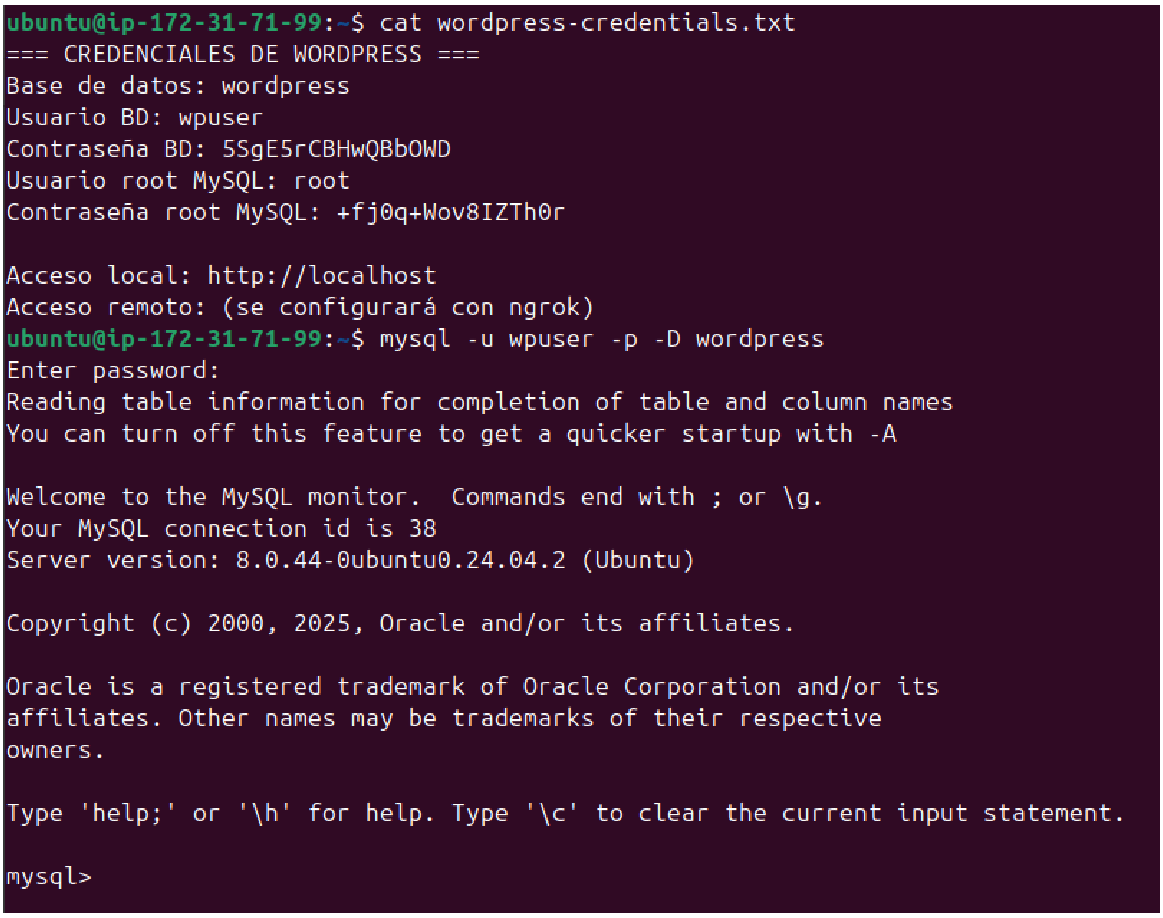
```
ssh -i ~/.ssh/wordpress-key-aws.pem ubuntu@TU-IP-PUBLICA
```



```
vboxuser@mail:~$ sudo ssh -i ~/.ssh/wordpress-key-aws.pem ubuntu@44.201.71.157
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 Dec  5 08:55:32 UTC 2025

  System load:  0.06              Temperature:           -273.1 C
  Usage of /:   22.6% of 13.49GB   Processes:             142
  Memory usage: 73%               Users logged in:       1
  Swap usage:   0%                IPv4 address for ens5: 172.31.71.99

 * 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.

0 updates can be applied immediately.

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


*** System restart required ***
Last login: Fri Dec  5 08:48:19 2025 from 80.24.49.242
ubuntu@ip-172-31-71-99:~$ 
```

Accede a MySQL:

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



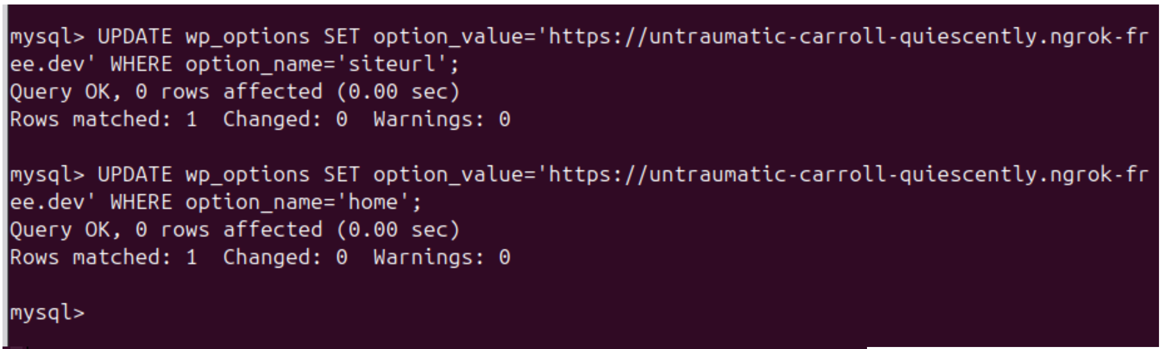
```
ubuntu@ip-172-31-71-99:~$ cat wordpress-credentials.txt
=== CREDENCIALES DE WORDPRESS ===
Base de datos: wordpress
Usuario BD: wpuser
Contraseña BD: 5SgE5rCBHwQBbOWD
Usuario root MySQL: root
Contraseña root MySQL: +fj0q+Wov8IZTh0r

Acceso local: http://localhost
Acceso remoto: (se configurará con ngrok)
ubuntu@ip-172-31-71-99:~$ 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 38
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.

mysql>
```

Ejecuta (*reemplaza la URL*):

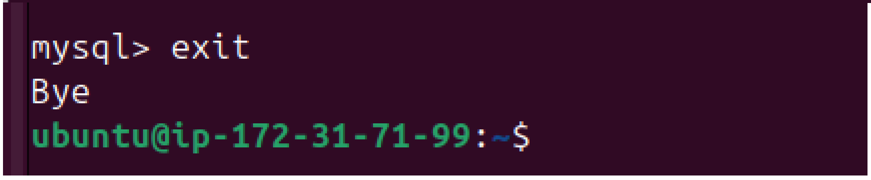
UPDATE wp_options SET option_value='https://abc123def456.ngrok-free.app' WHERE option_name='siteurl';

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

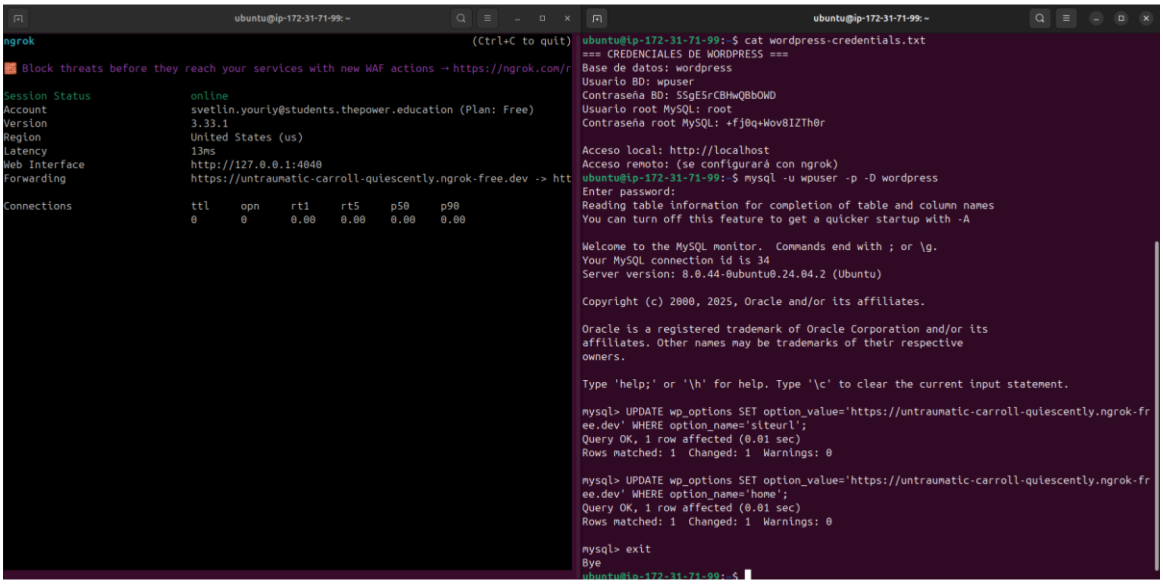
EXIT;



```
mysql> UPDATE wp_options SET option_value='https://untraumatic-carroll-quiescently.ngrok-fr
ee.dev' WHERE option_name='siteurl';
Query OK, 0 rows affected (0.00 sec)
Rows matched: 1  Changed: 0  Warnings: 0

mysql> UPDATE wp_options SET option_value='https://untraumatic-carroll-quiescently.ngrok-fr
ee.dev' WHERE option_name='home';
Query OK, 0 rows affected (0.00 sec)
Rows matched: 1  Changed: 0  Warnings: 0

mysql>
```



```
mysql> exit
Bye
ubuntu@ip-172-31-71-99:~$
```



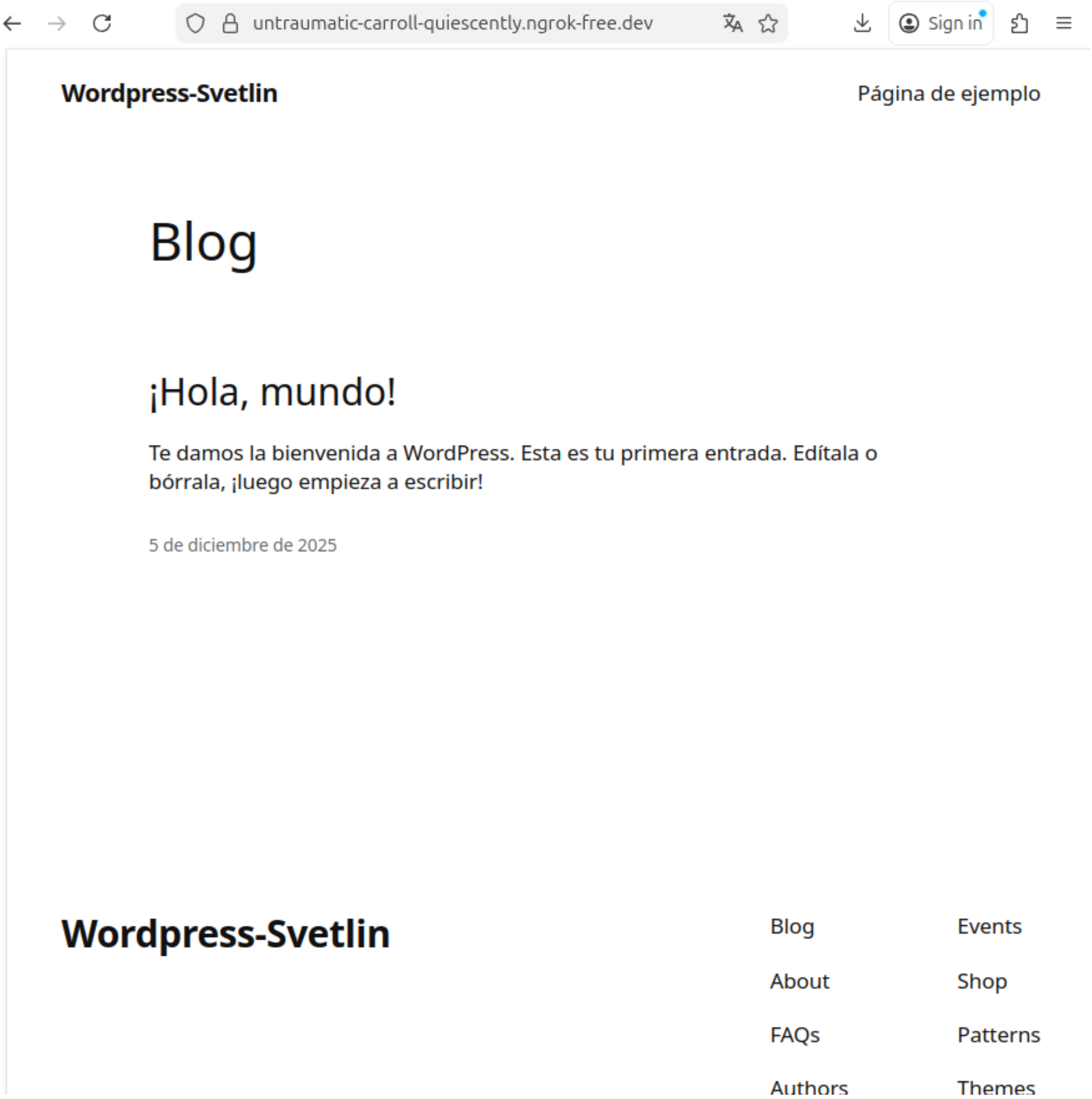
```
ngrok                                                          (Ctrl+C to quit)

🧱 Block threats before they reach your services with new WAF actions → https://ngrok.com/r

Session Status                online
Account                       svetlin.youriy@students.thepower.education (Plan: Free)
Version                       3.33.1
Region                        United States (us)
Latency                       13ms
Web Interface                 http://127.0.0.1:4040
Forwarding                    https://untraumatic-carroll-quiescently.ngrok-free.dev -> htt

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

```
ubuntu@ip-172-31-71-99:~$ cat wordpress-credentials.txt
=== CREDENCIALES DE WORDPRESS ===
Base de datos: wordpress
Usuario BD: wpuser
Contraseña BD: 5SgE5rCBHwQBbOWD
Usuario root MySQL: root
Contraseña root MySQL: +fj0q+Wov8IZTh0r

Acceso local: http://localhost
Acceso remoto: (se configurará con ngrok)
ubuntu@ip-172-31-71-99:~$ 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 34
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.

mysql> UPDATE wp_options SET option_value='https://untraumatic-carroll-quiescently.ngrok-fr
ee.dev' 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://untraumatic-carroll-quiescently.ngrok-fr
ee.dev' WHERE option_name='home';
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> exit
Bye
ubuntu@ip-172-31-71-99:~$
```

### 7.5 – Probar acceso remoto

Abre tu navegador y accede a:

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



untraumatic-carroll-quiescently.ngrok-free.dev

Wordpress-Svetlin

Página de ejemplo

Blog

¡Hola, mundo!

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

5 de diciembre de 2025

Wordpress-Svetlin

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

---

## 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;"

---

## 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