

Lo primero que hacemos es lanzar una instancia desde AWS



Después creamos un directorio ssh y le damos permisos

```
mkdir -p ~/.ssh  
chmod 700 ~/.ssh
```

Ahora creamos una key la cual nos permitirá el uso del ssh

```
Your identification has been saved in /root/.ssh/wordpress-key  
Your public key has been saved in /root/.ssh/wordpress-key.pub  
The key fingerprint is:  
SHA256:hIMY1Wb7Q6SDUxOT7FampxW9zYKoeYft0iay68jY07M Christian@aws  
The key's randomart image is:  
+--[ED25519 256]--+  
|  ...ooo .  |  
|    o .Oo= .  |  
|  . *oX.o +  |  
|    o Oo= o o  |  
|      = XS .  |  
|    o + =  |  
|      . + .  |  
| +o.. o +  |  
| . E*++ +  |
```

Observamos que se haya creado la llave

```
-rW----- 1 root root 399 Nov 28 08:16 /root/.ssh/wordpress-key
-rW-r--r-- 1 root root 95 Nov 28 08:16 /root/.ssh/wordpress-key.pub
```

Le damos solo permisos de lectura a la llave y comprobamos

```
chmod 400 ~/.ssh/wordpress-key
ls -la ~/.ssh/wordpress-key
-r----- 1 root root 399 Nov 28 08:16 /root/.ssh/wordpress-key
```

Ahora creamos un 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

Nombre

wordpress-key-ans

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

Ahora debemos crear reglas de entrada

ID de la regla del grupo de seguridad	Tipo información	Protocolo información	Intervalo de puertos información	Origen información	Descripción opcional información	Eliminar
sg-01b62d6b1d11d11d11	TCP personalizado	TCP	443	Persona... <input type="text" value="0.0.0.0/0"/>		Eliminar
sg-01b62d6b1d11d11d11	TCP personalizado	TCP	80	Persona... <input type="text" value="0.0.0.0/0"/>		Eliminar
sg-01b62d6b1d11d11d11	SSH	TCP	22	Persona... <input type="text" value="0.0.0.0/0"/>		Eliminar

[Agregar regla](#)

Ahora nos conectamos

```
ubuntu@ip-172-31-68-246:~$
```

Le metemos un update y un upgrade
Instalamos el LAMP

```
ubuntu@ip-172-31-68-246:~$ sudo apt install apache2 php php-mysql libapache2-mod-php php-curl php-gd php-mbstring php-xml php-xmlrpc php-intl php-zip mysql-server -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils fontconfig-config fonts-dejavu-core
  fonts-dejavu-nono 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 libnecab2 libprotobuf-lite32t64 libsharpuyuv0 libtiff6
  libtimedate-perl liburi-perl libwebp7 libxmlrpc-epi0t64 libxpm4 libzip4t64
  necab-ipadic necab-ipadic-utf8 necab-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:
```

Ponemos en marcha el apache2 y el mysql

```
ubuntu@ip-172-31-68-246:~$ sudo systemctl start apache2
ubuntu@ip-172-31-68-246:~$ sudo systemctl start mysql
ubuntu@ip-172-31-68-246:~$ 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
ubuntu@ip-172-31-68-246:~$ sudo systemctl enable mysql
Synchronizing state of mysql.service with SysV service script with /usr/lib/systemd/sy
stemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable mysql
ubuntu@ip-172-31-68-246:~$ 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 09:36:11 UTC; 1min 45s ago
     Docs: https://httpd.apache.org/docs/2.4/
```

Creamos un archivo sobre como instalar wordpress

```
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;"
```

transferimos el archivo

```
ubuntu@ip-172-31-68-246:~$ scp -i ~/.ssh/wordpress-key-aws.pem install-wordpress.sh ubuntu@172.31.68.246:
```

Instalamos el wordpress

```
ubuntu@ip-172-31-68-246:~$ chmod +x ~/install-wordpress.sh
ubuntu@ip-172-31-68-246:~$ ./install-wordpress.sh
=== Iniciando instalación automatizada de WordPress ===
./install-wordpress.sh: line 10: Paso: command not found
ubuntu@ip-172-31-68-246:~$ nano install-wordpress.sh
ubuntu@ip-172-31-68-246:~$ ./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...
```

Verificamos el estado de apache2 y mysql

```

● apache2.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: enabled)
   Active: active (running) since Fri 2025-11-28 09:36:11 UTC; 17min ago
     Docs: https://httpd.apache.org/docs/2.4/
    Main PID: 23798 (apache2)
      Tasks: 6 (limit: 1808)
    Memory: 14.8M (peak: 16.6M)
       CPU: 129ms
    CGroup: /system.slice/apache2.service
            └─23798 /usr/sbin/apache2 -k start
              23804 /usr/sbin/apache2 -k start
              23805 /usr/sbin/apache2 -k start
              23806 /usr/sbin/apache2 -k start
              23807 /usr/sbin/apache2 -k start
              23808 /usr/sbin/apache2 -k start

Nov 28 09:36:11 ip-172-31-68-246 systemd[1]: Starting apache2.service - The Apache HTTP Server...
Nov 28 09:36:11 ip-172-31-68-246 systemd[1]: Started apache2.service - The Apache HTTP Server.
● mysql.service - MySQL Community Server
   Loaded: loaded (/usr/lib/systemd/system/mysql.service; enabled; preset: enabled)

```

Instalamos Ngrok

```

ubuntu@ip-172-31-68-246:~$ wget https://bin.equinox.io/c/bNyj1mQVY4c/ngrok-v3-stable-linux-amd64.tgz
--2025-11-28 10:02:56-- https://bin.equinox.io/c/bNyj1mQVY4c/ngrok-v3-stable-linux-amd64.tgz
Resolving bin.equinox.io (bin.equinox.io)... 75.2.60.68, 13.248.244.96, 35.71.179.82, ...
Connecting to bin.equinox.io (bin.equinox.io)|75.2.60.68|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 10980950 (10M) [application/octet-stream]
Saving to: 'ngrok-v3-stable-linux-amd64.tgz'

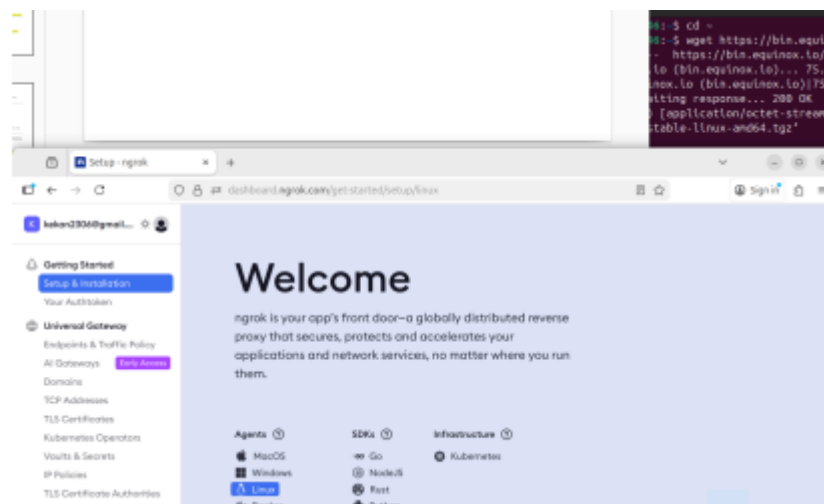
ngrok-v3-stable-linux-amd64.tgz 100%[=====] 10.47M --.-KB/s in 0.05s

2025-11-28 10:02:56 (202 MB/s) - 'ngrok-v3-stable-linux-amd64.tgz' saved [10980950/10980950]

ubuntu@ip-172-31-68-246:~$ tar -xvzf ngrok-v3-stable-linux-amd64.tgz
ngrok
ubuntu@ip-172-31-68-246:~$ sudo mv ngrok /usr/local/bin/

```

entramos en ngrok



Configuramos ngrok con el token

```

ubuntu@ip-172-31-68-246:~$ ngrok config add-authtoken cr_34BXgVtD3lgfpPmgu4gjReiTNEh
Authtoken saved to configuration file: /home/ubuntu/.config/ngrok/ngrok.yml
ubuntu@ip-172-31-68-246:~$

```

Ponemos el puerto 80 en ngrok

```
ngrok (Ctrl+C to quit)
👉 Create instant endpoints for local containers within Docker Desktop → https://ngrok.com/r/docker

Session Status      online
Account             kekan2306@gmail.com (Plan: Free)
Version             3.33.1
Region              United States (us)
Web Interface        http://127.0.0.1:4040
Forwarding           https://chery-righteous-away.ngrok-free.dev -> http://localhost:80

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

Hay fallos con el mysql porque no tengo contraseña, lo único que falta es actualizar wordpress.