

1. Iniciamos lanzando una nueva instancia en AWS

▼ 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.

The screenshot shows the AWS CloudWatch Metrics interface. At the top, there's a search bar and a navigation bar with tabs for 'Recent' and 'Inicio rápido'. Below this, there's a grid of icons representing different operating systems: Amazon Linux, macOS, Ubuntu, Windows, Red Hat, SUSE Linux, and Debian. To the right of the grid is a search icon and a link to 'Buscar más AMI'. A note below the grid states 'Inclusión de AMI de AWS, Marketplace y la comunidad'. Below the grid, there's a section titled 'Imagenes de máquina de Amazon (AMI)' which lists 'Ubuntu Server 24.04 LTS [HVM], SSD Volume Type' with an ID of 'ami-0ecb62995f68bb549'. It also shows 'Apto para la capa gratuita'. Further down, there's a detailed description of the AMI, its architecture (64 bits x86), ID (ami-0ecb62995f68bb549), publication date (2025-10-22), and owner (Canonical). A green button indicates 'Proveedor verificado'. On the far right, the number '2.' is displayed.

2. Creamos el directorio ssh y asignamos los permisos correspondientes

```
root@UbuntuRuben:/home/Silver# mkdir -p ~/.ssh
root@UbuntuRuben:/home/Silver# chmod 700 ~/.ssh
```

3. Generamos una clave que permitirá el acceso mediante 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]--+
| ...000 . |
| o .0o= . |
| . .*oX.o + |
| o 0o= o o |
| = XS . |
| 0 + = |
| . + . |
| +o... o + |
| . E*++ + |
+----[SHA256]-----+
```

4. Verificamos la correcta creación de la clave.

```
root@UbuntuRuben:/home/Silver# ls -la ~/ssh/wordpress-key
-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
```

5. Ajustamos los permisos de lectura sobre la llave y comprobamos su estado.

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

6. Creamos un nuevo par de claves para la conexión segura.

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 un sistema.

Nombre:

El nombre puede incluir hasta 256 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

7. Configuramos las reglas de entrada necesarias en el sistema.

ID de la regla del grupo de seguridad	Tipo	Protocolo	Intervalo de puertos	Origen	Descripción opcional
reg-0910a2e0-0a1d-4a1d-ba0f-000000000000	TCP personalizado	TCP	443	Personas...	<input type="text"/> 8000/8
reg-0910a2e1-0a1d-4a1d-ba0f-000000000000	TCP personalizado	TCP	80	Personas...	<input type="text"/> 8000/8
reg-0910a2e2-0a1d-4a1d-ba0f-000000000000	HTTP	TCP	22	Personas...	<input type="text"/> 8000/8

[Añadir regla](#)

8. Establecemos la conexión a la instancia.

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

9. Ejecutamos update y upgrade para mantener el sistema actualizado.

10. Instalamos el entorno 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-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 liblrc4 liblua5.4-0
  liblwp-mediatypes-perl libmecab2 libprotobuf-lite32t64 libsharpuyuv0 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:
```

11. Activamos los servicios apache2 y 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/
```

12. Creamos un archivo con las instrucciones para 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;"
```

13. Transferimos el archivo a la instancia.

```
ubuntu@ip-172-31-68-246:~$ scp -t ~/ssh/wordpress-key-aws.pem install-wordpress.sh ubuntu@172.31.68.2
46
```

14. Procedemos con la instalación de 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...
```

15. Verificamos nuevamente 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: 1000)
       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
```

16. Instalamos Ngrok y accedemos a su interfaz.

```
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/
```

17. Configuramos Ngrok introduciendo el token correspondiente.

18. Asignamos el puerto 80 a Ngrok.