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 Ed25519

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*).

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--------**

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**

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

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

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.

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".

1.6 – Obtener la IP pública

1. Selecciona la instancia
2. Copia la "Public IPv4 address" (*ej: 54.123.45.67*)

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**.

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

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

3.3 – Iniciar servicios

sudo systemctl start apache2

sudo systemctl start mysql

sudo systemctl enable apache2

sudo systemctl enable mysql

3.4 – Verificar servicios

sudo systemctl status apache2

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"
```

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:~/
```

5.2 – Dar permisos de ejecución

En AWS (*dentro de la sesión SSH*):

```
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

PARTE 6: VERIFICACIÓN DE INSTALACIÓN

6.1 – Verificar servicios

sudo systemctl status apache2

sudo systemctl status mysql

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*).

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/

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
```

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
```

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

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
```

Accede a MySQL:

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

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;
```

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".

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

- **Ed25519 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