

Laboratorio VPN Site-to-Site con WireGuard

Autor: Mauricio Teliz Duche

Objetivo principal del laboratorio: Configuración de una VPN Site-to-Site usando WireGuard para conectar un laboratorio local con un entorno remoto (por ejemplo, AWS). El laboratorio incluye generación de claves públicas/privadas, configuración de AllowedIPs, rutas específicas y verificación de conectividad, brindando experiencia práctica en VPNs, segmentación de tráfico y conexión segura entre redes.

Preparación del entorno

1)

Objetivo: Preparamos los entornos local y remoto, instalaremos WireGuard y verificamos conectividad de red.

Entornos:

Local: máquina virtual Ubuntu 22.04.

Remoto: instancia AWS Ubuntu 22.04.

2)

Instalación de WireGuard - (En ambos extremos, local y aws)

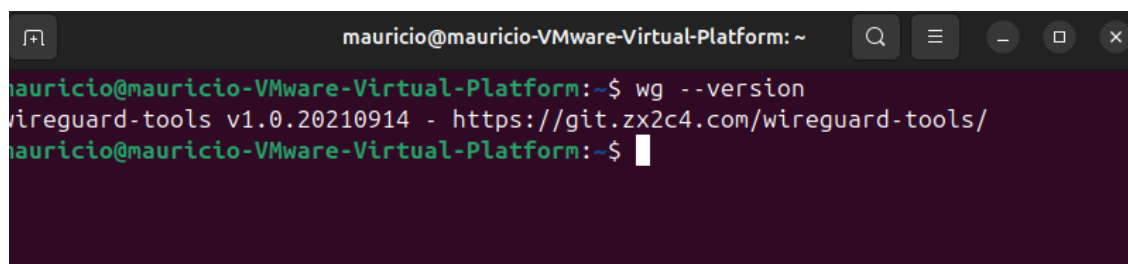
Comandos:

- sudo apt update
- sudo apt upgrade
- sudo apt install wireguard

comprobamos que wireguard se instaló correctamente:

- wg --version

Máquina Ubuntu local:



```
mauricio@mauricio-VMware-Virtual-Platform: ~  
mauricio@mauricio-VMware-Virtual-Platform:~$ wg --version  
wireguard-tools v1.0.20210914 - https://git.zx2c4.com/wireguard-tools/  
mauricio@mauricio-VMware-Virtual-Platform:~$
```

Máquina Ubuntu AWS remoto:

```
ubuntu@ip-10-0-1-157: ~  
Session Acciones Editar Vista Ayuda  
ubuntu@ip-10-0-1-157:~$ wg --version  
wireguard-tools v1.0.20210914 - https://git.zx2c4.com/wireguard-tools/  
ubuntu@ip-10-0-1-157:~$
```

WireGuard instalado correctamente.

3)

Generación de claves y asignación de IPs para WireGuard

Objetivo: Establecer un túnel VPN site-to-site entre la red local y AWS.

- Generaremos un par de claves en cada extremo.

Comandos:

- wg genkey | tee privatekey | wg pubkey > publickey

Máquina Ubuntu local:

privatekey → Clave privada.

Publickey → Clave pública.

```
mauricio@mauricio-VMware-Virtual-Platform:~$ wg --version  
mauricio@mauricio-VMware-Virtual-Platform:~$ wg genkey | tee privatekey | wg pubkey > publickey  
mauricio@mauricio-VMware-Virtual-Platform:~$ ls  
Descargas Documentos Escritorio Imágenes Música Plantillas privatekey publickey Público snap Videos  
mauricio@mauricio-VMware-Virtual-Platform:~$ cat privatekey  
sGd/1pj9BkPVziPHcX1CtJ0rAQIX6ooMzhDh+P6D6XU=  
mauricio@mauricio-VMware-Virtual-Platform:~$ cat publickey  
IdPWAXi45q8xjfrGFcS0JIubX2ztAYciH9vODJAXc2s=  
mauricio@mauricio-VMware-Virtual-Platform:~$
```

Máquina Ubuntu AWS remoto:

```
ubuntu@ip-10-0-1-157: ~  
Session Acciones Editar Vista Ayuda  
ubuntu@ip-10-0-1-157:~$ wg genkey | tee privatekey | wg pubkey > publickey  
ubuntu@ip-10-0-1-157:~$ ls  
privatekey publickey  
ubuntu@ip-10-0-1-157:~$ cat privatekey  
qPwxcX4nATT+NK2MJ0kaNULrJ6ktKM2zK5L+9l3wGEQ=  
ubuntu@ip-10-0-1-157:~$ cat publickey  
fDGPiGT/yaI1jgCKsSfcJGuGQt1nYGboyRQklxvKdh4=  
ubuntu@ip-10-0-1-157:~$
```

4)

Configuración de la interfaz WireGuard (wg0.conf)

- Máquina Ubuntu local:

```
GNU nano 7.2 /etc/wireguard/wg0.conf *
[Interface]
PrivateKey = sGd/1pj9BkPVziPHcX1CtJ0rAQIX6ooMzhDh+P6D6XU=
Address = 10.8.0.1/24
ListenPort = 51820

[Peer]
PublicKey = fdGPiGT/yaI1jgCKsSfcJGuGQt1nYGboyRQklxvKdh4=
AllowedIPs = 10.8.0.2/32
Endpoint = 54.94.34.123:51820
PersistentKeepalive = 25
```

Explicación:

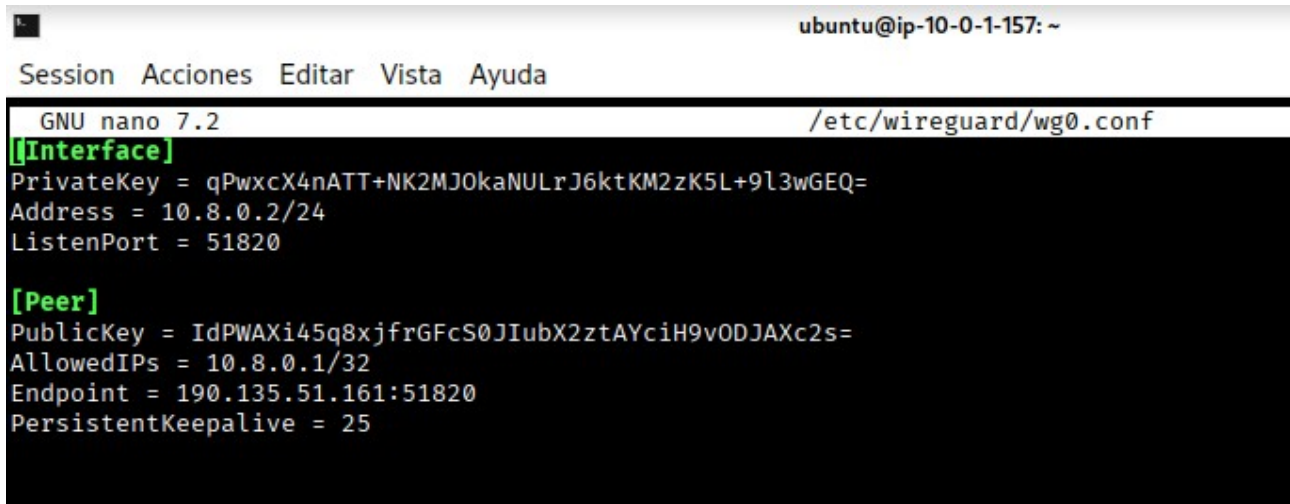
[Interface] → Interfaz de configuración de la maquina Ubuntu local.

- * PrivateKey: Clave privada [Maquina Ubuntu local]
- * Address: IP de [Maquina Ubuntu Local] dentro del túnel VPN.
- * ListenPort: Puerto UDP donde WireGuard escuchará el tráfico.

[Peer] → Información del peer (Máquina Ubuntu AWS):

- * PublicKey: Clave pública de la máquina Ubuntu AWS.
- * AllowedIPs: Todos los paquetes cuyo destino sea la IP interna del peer (10.8.0.2) se dirigirán a través del túnel cifrado de la VPN.
- * Endpoint: IP pública del peer remoto (AWS) a la que la VPN enviará los paquetes cifrados.
- * PersistentKeepalive: Mantiene la conexión VPN activa aunque haya NAT, enviando paquetes periódicos para que el túnel no se cierre por inactividad.

- Máquina Ubuntu AWS:



```
ubuntu@ip-10-0-1-157: ~  
Session Acciones Editar Vista Ayuda  
GNU nano 7.2 /etc/wireguard/wg0.conf  
[Interface]  
PrivateKey = qPwxcX4nATT+NK2MJ0kaNULrJ6ktKM2zK5L+9l3wGEQ=  
Address = 10.8.0.2/24  
ListenPort = 51820  
  
[Peer]  
PublicKey = IdPWAXi45q8xjfrGFcS0JIubX2ztAYciH9v0DJAXc2s=  
AllowedIPs = 10.8.0.1/32  
Endpoint = 190.135.51.161:51820  
PersistentKeepalive = 25
```

Explicación:

[Interface] → Interfaz de configuración de la maquina Ubuntu AWS.

- * PrivateKey: Clave privada [Maquina Ubuntu AWS]
- * Address: IP de [Maquina Ubuntu AWS] dentro del túnel VPN.
- * ListenPort: Puerto UDP donde WireGuard escuchará el tráfico.

[Peer] → Información del peer (Máquina Ubuntu LOCAL):

- * PublicKey: Clave pública de la máquina Ubuntu LOCAL.
- * AllowedIPs: Todos los paquetes cuyo destino sea la IP interna del peer (10.8.0.1) se dirigirán a través del túnel cifrado de la VPN.
- * Endpoint: IP pública de la máquina Ubuntu local a la que la VPN enviará los paquetes cifrados.
- * PersistentKeepalive: Mantiene la conexión VPN activa aunque haya NAT, enviando paquetes periódicos para que el túnel no se cierre por inactividad.

5)

Verificación del túnel VPN

Objetivo: Confirmar que la VPN site-to-site funciona correctamente y que ambas máquinas pueden comunicarse a través del túnel.

Desde la máquina Ubuntu Local:

* Levantamos la interfaz y cargamos la configuración de la `wg0.conf`

* Comandos a utilizar:

- `sudo wg-quick up wg0`

```
mauricio@mauricio-VMware-Virtual-Platform:~$ sudo wg-quick up wg0
[#] ip link add wg0 type wireguard
[#] wg setconf wg0 /dev/fd/63
[#] ip -4 address add 10.8.0.1/24 dev wg0
[#] ip link set mtu 1420 up dev wg0
mauricio@mauricio-VMware-Virtual-Platform:~$
```

Desde la máquina Ubuntu AWS:

```
ubuntu@ip-10-0-1-157:~$ sudo wg-quick up wg0
[#] ip link add wg0 type wireguard
[#] wg setconf wg0 /dev/fd/63
[#] ip -4 address add 10.8.0.2/24 dev wg0
[#] ip link set mtu 8921 up dev wg0
ubuntu@ip-10-0-1-157:~$
```

Verificamos el estado del túnel:

- `sudo wg show`

Desde la máquina Ubuntu Local:

```
mauricio@mauricio-VMware-Virtual-Platform:~$ sudo wg show
[sudo] contraseña para mauricio:
Interface: wg0
  public key: IdPWAXi45q8xjfrGFcS0JIubX2ztAYciH9vODJAXc2s=
  private key: (hidden)
  listening port: 51820

peer: fDGPiGT/yaI1jgCKsSfcJGuGQt1nYGboyRQklxvKdh4=
  endpoint: 54.94.34.123:51820
  allowed ips: 10.8.0.2/32
  latest handshake: 1 minute, 56 seconds ago
  transfer: 1.45 KiB received, 16.91 KiB sent
  persistent keepalive: every 25 seconds
mauricio@mauricio-VMware-Virtual-Platform:~$
```

Desde la máquina Ubuntu AWS:

```
ubuntu@ip-10-0-1-157:~$ sudo wg show
interface: wg0
  public key: fDGPiGT/yaI1jgCKsSfcJGuGQt1nYGboyRQklxvKdh4=
  private key: (hidden)
  listening port: 51820

peer: IdPWAXi45q8xjfrGFcS0JIubX2ztAYciH9vODJAXc2s=
  endpoint: 190.135.51.161:64505
  allowed ips: 10.8.0.1/32
  latest handshake: 25 seconds ago
  transfer: 4.76 KiB received, 1.87 KiB sent
  persistent keepalive: every 25 seconds
ubuntu@ip-10-0-1-157:~$
```

Esto confirma que el túnel VPN está funcionando desde ambos extremos.

Ahora probaremos conectividad desde ambos extremos:

Desde Máquina Ubuntu local:

Comandos:

- ping 10.8.0.2

```
mauricio@mauricio-VMware-Virtual-Platform:~$ ping 10.8.0.2
PING 10.8.0.2 (10.8.0.2) 56(84) bytes of data.
64 bytes from 10.8.0.2: icmp_seq=1 ttl=64 time=38.2 ms
64 bytes from 10.8.0.2: icmp_seq=2 ttl=64 time=38.5 ms
64 bytes from 10.8.0.2: icmp_seq=3 ttl=64 time=37.9 ms
64 bytes from 10.8.0.2: icmp_seq=4 ttl=64 time=38.1 ms
^C
--- 10.8.0.2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3007ms
rtt min/avg/max/mdev = 37.857/38.159/38.479/0.222 ms
mauricio@mauricio-VMware-Virtual-Platform:~$
```

Desde Máquina Ubuntu AWS:

Comandos:

- ping 10.8.0.1

```
ubuntu@ip-10-0-1-157:~$ ping 10.8.0.1
PING 10.8.0.1 (10.8.0.1) 56(84) bytes of data.
64 bytes from 10.8.0.1: icmp_seq=1 ttl=64 time=37.4 ms
64 bytes from 10.8.0.1: icmp_seq=2 ttl=64 time=37.7 ms
64 bytes from 10.8.0.1: icmp_seq=3 ttl=64 time=147 ms
^C
--- 10.8.0.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2004ms
rtt min/avg/max/mdev = 37.419/74.020/146.898/51.532 ms
ubuntu@ip-10-0-1-157:~$
```

Las pruebas realizadas confirman que la conectividad es bidireccional. Por lo tanto, la conexión de extremo a extremo se realizó con éxito.

Conclusión:

Se realizó un ping desde ambas máquinas Ubuntu hacia sus respectivas IP internas. Todos los paquetes fueron recibidos correctamente, confirmando la conectividad bidireccional a través del túnel VPN site-to-site. El tráfico cifrado fluye correctamente entre ambos extremos.

Puertos y seguridad:

- La VPN WireGuard utiliza UDP 51820 como puerto por defecto en ambos extremos.
- * En la máquina local este puerto debe estar habilitado por cualquier firewall.
- * En AWS, se debe abrir **UDP 51820** en el security group de la instancia para permitir tráfico entrante desde la IP pública del local.
- * El uso de UDP es necesario porque WireGuard no utiliza TCP. Necesita comunicación rápida y confiable para el cifrado y handshake como la que le puede brindar UDP.

Diagrama conceptual del túnel VPN

