Cristóbal Suárez Abad

Seguridad y Alta Disponibilidad  - 2º ASIR

Actividad 1 - VPN WireGuard

Contenido

[1) Instalar WireGuard. 2](#_Toc214646130)

[2) Configurar WireGuard. 3](#_Toc214646131)

[3) Generar la configuración de los clientes de WireGuard. 5](#_Toc214646132)

[4) Administrar los procesos del servidor de VPN WireGuard. 8](#_Toc214646133)

[5) Establecer el Firewall. 10](#_Toc214646134)

[6) Conectar a los clientes al servidor VPN. 13](#_Toc214646135)

# Instalar WireGuard.

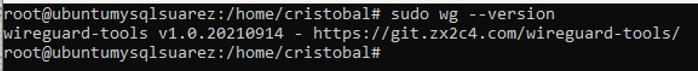
<https://docs.vultr.com/how-to-install-wireguard-vpn-on-ubuntu-24-04>

Hemos configurado una VM Ubuntu 24.04 que usaremos como servidor VPN. Su IP es 10.2.7.45

**sudo apt install wireguard -y**



**sudo wg --version**



# Configurar WireGuard.

Generamos una clave privada para WireGuard y la guardamos en un archivo:

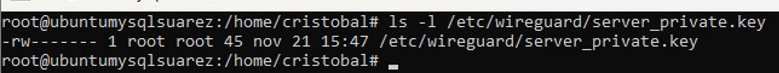
**sudo wg genkey | sudo tee /etc/wireguard/server\_private.key**

Texto

El contenido generado por IA puede ser incorrecto.

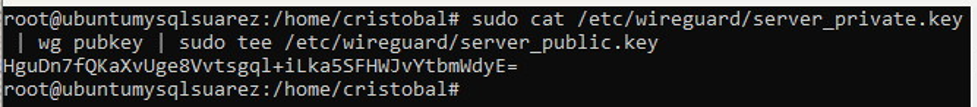
Modificamos los permisos para evitar que ningún otro usuario pueda acceder a su contenido.

**sudo chmod 600 /etc/wireguard/server\_private.key**



Ahora generamos una clave pública mediante la clave privada anterior. La guardamos en un archivo.

**sudo cat /etc/wireguard/server\_private.key | wg pubkey | sudo tee /etc/wireguard/server\_public.key**



**Clave Privada:**

eBuXARj+KzrYmqq9LD1MdbZ6/p/Cyh82YpOJl0tkkUA=

**Clave Pública:**

HguDn7fQKaXvUge8Vvtsgql+iLka5SFHWJvYtbmWdyE=

Témenos que averiguar el nombre la interfaz de red de nuestro equipo, además de nuestra dirección IP. Usamos “**ip a**”:

Texto

El contenido generado por IA puede ser incorrecto.

En nuestro caso el nombre de la interfaz de red es “ens18” y la IP es 10.2.7.45

Nos servirá para crear un nuevo archivo de configuración para WireGuard:

**sudo nano /etc/wireguard/wg0.conf**

*[Interface]*

*Address = 10.8.0.1/24*

*SaveConfig = true*

*PrivateKey = eBuXARj+KzrYmqq9LD1MdbZ6/p/Cyh82YpOJl0tkkUA= # Server-Private-Key*

*PostUp = ufw route allow in on wg0 out on ens18*

*PostUp = iptables -t nat -I POSTROUTING -o ens18 -j MASQUERADE*

*PreDown = ufw route delete allow in on wg0 out on ens18*

*PreDown = iptables -t nat -D POSTROUTING -o ens18 -j MASQUERADE*

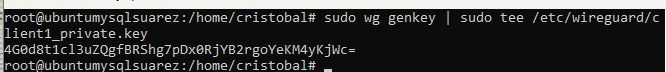
*ListenPort = 51820*

En “**Address**” ponemos la ip privada que se le asignará a la interfaz de WireGuard. En “**PrivateKey**” ponemos la clave privada que hemos generado antes. Allí donde ponga “ens18” debemos cambiarlo por el nombre de la interfaz de nuestro equipo. Para más información sobre el resto de las opciones, consultar la guía.

# Generar la configuración de los clientes de WireGuard.

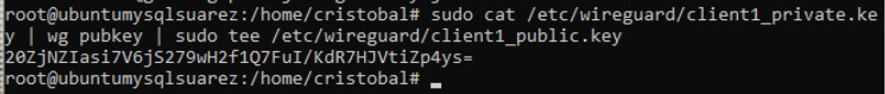
Generamos una clave privada para Clientes y la guardamos en un archivo:

**sudo wg genkey | sudo tee /etc/wireguard/client1\_private.key**



Generamos una clave pública para Clientes y la guardamos en un archivo. Se genera mediante la privada.

**sudo cat /etc/wireguard/client1\_private.key | wg pubkey | sudo tee /etc/wireguard/client1\_public.key**



Ahora creamos el archivo de configuración para clientes.

**sudo nano /etc/wireguard/client1.conf**

*[Interface]*

*PrivateKey = 4G0d8t1cl3uZQgfBRShg7pDx0RjYB2rgoYeKM4yKjWc= # Client-Private-Key*

*Address = 10.8.0.2/24*

*DNS = 8.8.8.8*

*[Peer]*

*PublicKey = HguDn7fQKaXvUge8Vvtsgql+iLka5SFHWJvYtbmWdyE= # Server-Public-Key*

*AllowedIPs = 0.0.0.0/0*

*Endpoint = 10.2.7.45:51820*

*PersistentKeepalive = 15*

Cambiamos las IPs. La primera IP es la que recibirá el equipo cliente cuando active la VPN. La segunda IP es la IP real del servidor VPN.

También debemos incluir las claves. La “PrivateKey” es la clave privada del cliente. La “PublicKey” es la clave pública del servidor VPN.

Texto

El contenido generado por IA puede ser incorrecto.

Copiamos el archivo al directorio de trabajo de nuestro usuario del servidor para poder pasarlo más fácilmente al cliente más adelante.

**sudo cp /etc/wireguard/client1.conf client1.conf**

Texto

El contenido generado por IA puede ser incorrecto.

Ahora debemos hacer un pequeño añadido al archivo de configuración de WireGuard.

**sudo nano /etc/wireguard/wg0.conf**

***[Peer]***

***PublicKey = 20ZjNZIasi7V6jS279wH2f1Q7FuI/KdR7HJVtiZp4ys=***

***AllowedIPs = 10.8.0.2/32***

Debemos añadir la clave pública del cliente y la IP que le vamos a otorgar cuando se conecte a la VPN con una máscara de “**32**”.

Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.

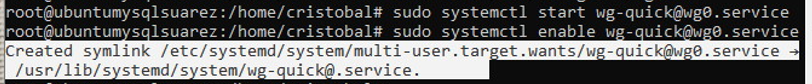
# Administrar los procesos del servidor de VPN WireGuard.

Vamos a levantar la interfaz que hemos creado previamente “**wg0**”.

**sudo systemctl start** [**wg-quick@wg0.service**](mailto:wg-quick@wg0.service)

Y también lo vamos a habilitar para que se inicia cuando se encienda el servidor.

**sudo systemctl enable wg-quick@wg0.service**



Comprobamos su estado.

**sudo systemctl status wg-quick@wg0.service**

Texto

El contenido generado por IA puede ser incorrecto.

Vemos el estado de la interfaz “**wg0**”:

**sudo wg show wg0**

Texto

El contenido generado por IA puede ser incorrecto.

Para comprobar sus registros:

**sudo journalctl -u** [**wg-quick@wg0.service**](mailto:wg-quick@wg0.service)

Texto

El contenido generado por IA puede ser incorrecto.

# Establecer el Firewall.

Comprobamos su estado:

**sudo ufw status**



Lo activamos, pero con la excepción para el puerto 22 que es de SSH. También lo habilitamos para cuando se inicie el servidor.

**sudo ufw allow 22 && sudo ufw enable**

Texto

El contenido generado por IA puede ser incorrecto.

Abrimos el puerto que usará WireGuard.

**sudo ufw allow 51820/udp**

Recargamos:

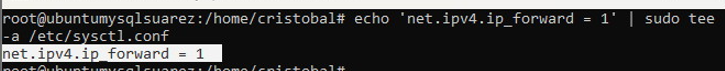
**sudo ufw reload**

Y comprobamos “**status**”:

Texto

El contenido generado por IA puede ser incorrecto.

El siguiente comando es para que Linux se convierta en un “**router**” y mueva paquetes de una interfaz de red a otra:

**echo 'net.ipv4.ip\_forward = 1' | sudo tee -a /etc/sysctl.conf**

Recargamos para que se establezca la configuración:

**sudo sysctl -p**



Vamos a permitir que la interfaz (en nuestro caso la “ens18”) traduzca peticiones de red la subnet de WireGuard VPN “10.8.0.0/24”. **RECUERDA**: Si tu interfaz tiene otro nombre, ponlo en el comando.

**sudo iptables -t nat -A POSTROUTING -s 10.8.0.0/24 -o ens18 -j MASQUERADE**



Vamos a guardar los cambios anteriores en un archivo:

**sudo iptables-save | sudo tee /etc/iptables/rules.v4**



Nos sale un buen tocho.

Texto

El contenido generado por IA puede ser incorrecto.

# Conectar a los clientes al servidor VPN.

Desde el cliente nos descargamos el archivo de configuración de cliente que hemos hecho antes.

**scp linuxuser@wireguard-server-ip:client1.conf .**

En nuestro caso es así: **scp cristobal@10.2.7.45:client1.conf .**

Texto

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Nos descargamos el instalador. En nuestro caso el de Windows.

<https://www.wireguard.com/install/>

Interfaz de usuario gráfica, Aplicación, Sitio web

El contenido generado por IA puede ser incorrecto.

Una vez instalado. Le damos a importar túnel desde archivo.



Seleccionamos el archivo.

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

Le damos a **activar**:

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

Si hacemos “**ipconfig**” en el equipo cliente, veremos que tenemos la IP del equipo y la que le hemos asignado antes en la configuración del cliente VPN.

Texto

El contenido generado por IA puede ser incorrecto.

Para comprobar que funciona perfectamente, debemos hacer ping a la IP del servidor VPN de la subnet configurada: 10.8.0.1

Texto

El contenido generado por IA puede ser incorrecto.

**Funciona**.