ACTIVIDAD 1 - VPN WIREGUARD

Cristóbal Suárez Abad

SEGURIDAD Y ALTA DISPONIBILIDAD - 2º ASIR

Contenido

1) 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

```
cristobal@ubuntumysqlsuarez:~$ sudo apt install wireguard -y
```

sudo wg --version

```
root@ubuntumysqlsuarez:/home/cristobal# sudo wg --version
wireguard-tools v1.0.20210914 - https://git.zx2c4.com/wireguard-tools/
root@ubuntumysqlsuarez:/home/cristobal#
```

2) Configurar WireGuard.

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

sudo wg genkey | sudo tee /etc/wireguard/server_private.key

```
root@ubuntumysqlsuarez:/home/cristobal# sudo wg genkey | sudo tee /etc/wireguard/s
erver_private.key
eBuXARj+KzrYmqq9LD1MdbZ6/p/Cyh82YpOJl0tkkUA=
root@ubuntumysqlsuarez:/home/cristobal# cat /etc/wireguard/server_private.key
eBuXARj+KzrYmqq9LD1MdbZ6/p/Cyh82YpOJl0tkkUA=
```

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

sudo chmod 600 /etc/wireguard/server_private.key

```
root@ubuntumysqlsuarez:/home/cristobal# ls -l /etc/wireguard/server_private.key
-rw------- 1 root root 45 nov 21 15:47 /etc/wireguard/server_private.key
root@ubuntumysqlsuarez:/home/cristobal#
```

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

```
root@ubuntumysqlsuarez:/home/cristobal# sudo cat /etc/wireguard/server_private.key
 | wg pubkey | sudo tee /etc/wireguard/server_public.key
HguDn7fQKaXvUge8Vvtsgql+iLka5SFHWJvYtbmWdyE=
root@ubuntumysqlsuarez:/home/cristobal#
```

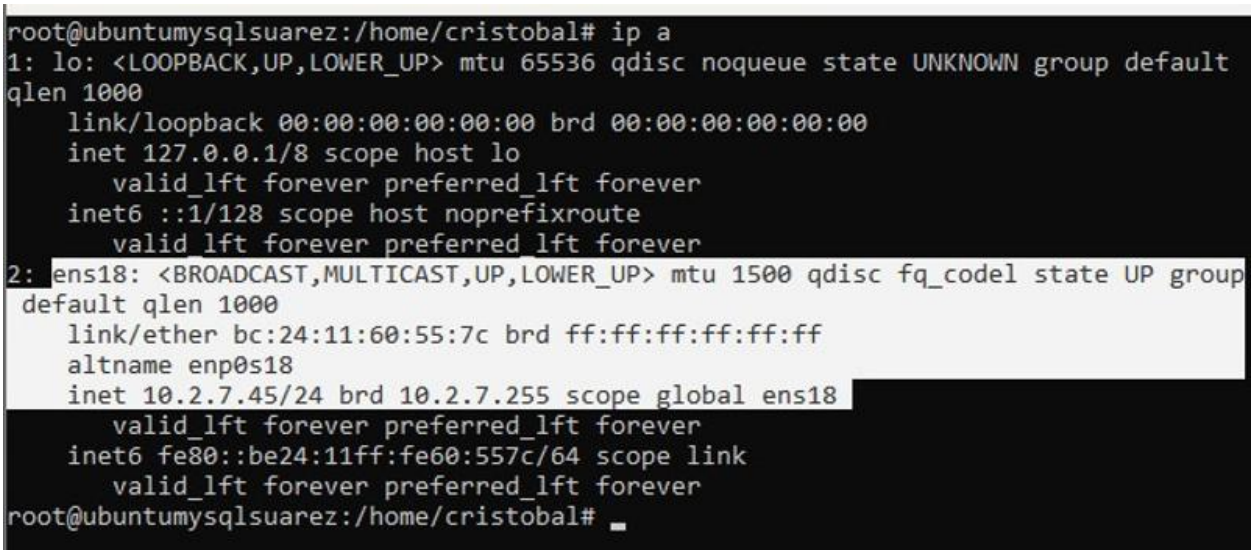
Clave Privada:

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**":



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.

3) 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

```
root@ubuntumysqlsuarez:/home/cristobal# sudo wg genkey | sudo tee /etc/wireguard/c
lient1_private.key
4G0d8t1cl3uZQgfBRShg7pDx0RjYB2rgoYeKM4yKjWc=
root@ubuntumysqlsuarez:/home/cristobal#
```

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

```
root@ubuntumysqlsuarez:/home/cristobal# sudo cat /etc/wireguard/client1_private.ke
y | wg pubkey | sudo tee /etc/wireguard/client1_public.key
20ZjNZIasi7V6jS279wH2f1Q7FuI/KdR7HJVtiZp4ys=
root@ubuntumysqlsuarez:/home/cristobal#
```

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.

```
cristobal@ubuntumysqlsuarez:~$ cat 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
```

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

```
root@ubuntumysqlsuarez:/home/cristobal# sudo cp /etc/wireguard/client1.conf client
1.conf
root@ubuntumysqlsuarez:/home/cristobal# ls -l
total 793228
-rw-r--r-- 1 root      root            293 nov 21 16:02 client1.conf
drwxrwxr-x 2 cristobal cristobal      4096 nov 11 11:51 comprimido
-rw-rw-r-- 1 cristobal cristobal 811275696 sep  6 12:34 debian-13.1.0-amd64-netins
t.iso.gz
-rw-r--r-- 1 root      root            196 nov  5 16:28 politicacontrasena.sh
-rw-r--r-- 1 root      root         128736 nov 12 08:07 rkhunter-12112025.log
-rw-r--r-- 1 root      root         817896 nov 12 08:21 scanner
-rw-rw-r-- 1 cristobal cristobal     17604 nov 11 17:41 webmin-setup-repo.sh
root@ubuntumysqlsuarez:/home/cristobal#
```

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**".

```
  GNU nano 7.2                  /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

[Peer]
PublicKey = 20ZjNZIasi7V6jS279wH2f1Q7FuI/KdR7HJVtiZp4ys=
AllowedIPs = 10.8.0.2/32
```

4) 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

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

sudo systemctl enable wg-quick@wg0.service

```
root@ubuntumysqlsuarez:/home/cristobal# sudo systemctl start wg-quick@wg0.service
root@ubuntumysqlsuarez:/home/cristobal# sudo systemctl enable wg-quick@wg0.service
Created symlink /etc/systemd/system/multi-user.target.wants/wg-quick@wg0.service →
 /usr/lib/systemd/system/wg-quick@.service.
```

Comprobamos su estado.

sudo systemctl status wg-quick@wg0.service

```
● wg-quick@wg0.service - WireGuard via wg-quick(8) for wg0
     Loaded: loaded (/usr/lib/systemd/system/wg-quick@.service; enabled; preset: enabled)
     Active: active (exited) since Fri 2025-11-21 16:03:45 UTC; 21s ago
       Docs: man:wg-quick(8)
             man:wg(8)
             https://www.wireguard.com/
             https://www.wireguard.com/quickstart/
             https://git.zx2c4.com/wireguard-tools/about/src/man/wg-quick.8
             https://git.zx2c4.com/wireguard-tools/about/src/man/wg.8
    Main PID: 3062 (code=exited, status=0/SUCCESS)
         CPU: 415ms

nov 21 16:03:44 ubuntumysqlsuarez systemd[1]: Starting wg-quick@wg0.service - WireGuard via wg-quick(8) for wg0...
nov 21 16:03:44 ubuntumysqlsuarez wg-quick[3062]: [#] ip link add wg0 type wireguard
nov 21 16:03:44 ubuntumysqlsuarez wg-quick[3062]: [#] wg setconf wg0 /dev/fd/63
nov 21 16:03:44 ubuntumysqlsuarez wg-quick[3062]: [#] ip -4 address add 10.8.0.1/24 dev wg0
nov 21 16:03:44 ubuntumysqlsuarez wg-quick[3062]: [#] ip link set mtu 1420 up dev wg0
nov 21 16:03:44 ubuntumysqlsuarez wg-quick[3062]: [#] ufw route allow in on wg0 out on ens18
nov 21 16:03:45 ubuntumysqlsuarez wg-quick[3109]: Rules updated
nov 21 16:03:45 ubuntumysqlsuarez wg-quick[3109]: Rules updated (v6)
nov 21 16:03:45 ubuntumysqlsuarez wg-quick[3062]: [#] iptables -t nat -I POSTROUTING -o ens18 -j MASQUERADE
nov 21 16:03:45 ubuntumysqlsuarez systemd[1]: Finished wg-quick@wg0.service - WireGuard via wg-quick(8) for wg0.
```

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

sudo wg show wg0

```
cristobal@ubuntumysqlsuarez:~$ sudo wg show wg0
[sudo] password for cristobal:
interface: wg0
  public key: HguDn7fQKaXvUge8Vvtsgql+iLka5SFHWJvYtbmWdyE=
  private key: (hidden)
  listening port: 51820

peer: 20ZjNZIasi7V6jS279wH2f1Q7FuI/KdR7HJVtiZp4ys=
  endpoint: 10.2.17.111:60518
  allowed ips: 10.8.0.2/32
  latest handshake: 1 minute, 32 seconds ago
  transfer: 469.99 KiB received, 7.49 MiB sent
cristobal@ubuntumysqlsuarez:~$
```

Para comprobar sus registros:

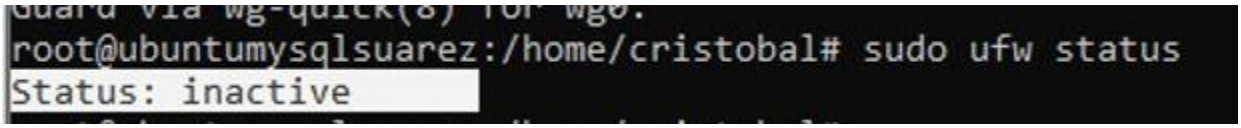
sudo journalctl -u wg-quick@wg0.service

```
root@ubuntumysqlsuarez:/home/cristobal# sudo journalctl -u wg-quick@wg0.service
nov 21 16:03:44 ubuntumysqlsuarez systemd[1]: Starting wg-quick@wg0.service - WireGuard via wg-quick(8) for wg0...
nov 21 16:03:44 ubuntumysqlsuarez wg-quick[3062]: [#] ip link add wg0 type wireguard
nov 21 16:03:44 ubuntumysqlsuarez wg-quick[3062]: [#] wg setconf wg0 /dev/fd/63
nov 21 16:03:44 ubuntumysqlsuarez wg-quick[3062]: [#] ip -4 address add 10.8.0.1/24 dev wg0
nov 21 16:03:44 ubuntumysqlsuarez wg-quick[3062]: [#] ip link set mtu 1420 up dev wg0
nov 21 16:03:44 ubuntumysqlsuarez wg-quick[3062]: [#] ufw route allow in on wg0 out on ens18
nov 21 16:03:45 ubuntumysqlsuarez wg-quick[3109]: Rules updated
nov 21 16:03:45 ubuntumysqlsuarez wg-quick[3109]: Rules updated (v6)
nov 21 16:03:45 ubuntumysqlsuarez wg-quick[3062]: [#] iptables -t nat -I POSTROUTING -o ens18 -j MASQUERADE
nov 21 16:03:45 ubuntumysqlsuarez systemd[1]: Finished wg-quick@wg0.service - WireGuard via wg-quick(8) for wg0.
root@ubuntumysqlsuarez:/home/cristobal#
```

5) Establecer el Firewall.

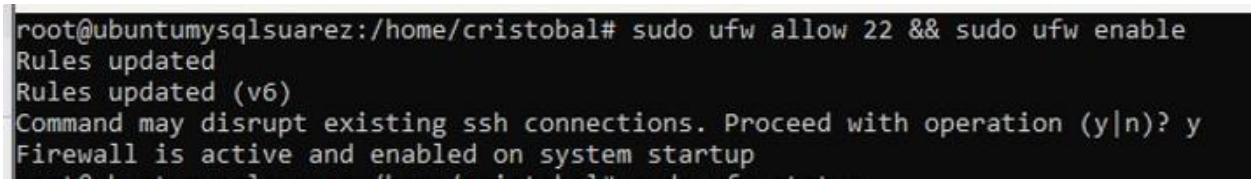
Comprobamos su estado:

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



Abrimos el puerto que usará WireGuard.

sudo ufw allow 51820/udp

Recargamos:

sudo ufw reload

Y comprobamos "**status**":

```
root@ubuntumysqlsuarez:/home/cristobal# sudo ufw allow 51820/udp
Rule added
Rule added (v6)
root@ubuntumysqlsuarez:/home/cristobal# sudo ufw reload
Firewall reloaded
root@ubuntumysqlsuarez:/home/cristobal# sudo ufw status
Status: active

To                         Action      From
--                         ------      ----
3306                       ALLOW       Anywhere
22                         ALLOW       Anywhere
51820/udp                  ALLOW       Anywhere
3306 (v6)                  ALLOW       Anywhere (v6)
22 (v6)                    ALLOW       Anywhere (v6)
51820/udp (v6)             ALLOW       Anywhere (v6)

Anywhere on ens18          ALLOW FWD   Anywhere on wg0
Anywhere (v6) on ens18     ALLOW FWD   Anywhere (v6) on wg0

root@ubuntumysqlsuarez:/home/cristobal#
```

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

```
root@ubuntumysqlsuarez:/home/cristobal# echo 'net.ipv4.ip_forward = 1' | sudo tee
-a /etc/sysctl.conf
net.ipv4.ip_forward = 1
```

Recargamos para que se establezca la configuración:

sudo sysctl -p

```
root@ubuntumysqlsuarez:/home/cristobal# sudo sysctl -p
net.ipv4.ip_forward = 1
```

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

```
root@ubuntumysqlsuarez:/home/cristobal# 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

```
root@ubuntumysqlsuarez:/home/cristobal# sudo iptables-save | sudo tee /etc/iptable
s/rules.v4
```

Nos sale un buen tocho.

```
:PREROUTING ACCEPT [0:0]
:INPUT ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
:POSTROUTING ACCEPT [1:78]
-A POSTROUTING -o ens18 -j MASQUERADE
-A POSTROUTING -s 10.8.0.0/24 -o ens18 -j MASQUERADE
COMMIT
# Completed on Fri Nov 21 16:09:06 2025
root@ubuntumysqlsuarez:/home/cristobal#
```

6) Conectar a los clientes al servidor VPN.

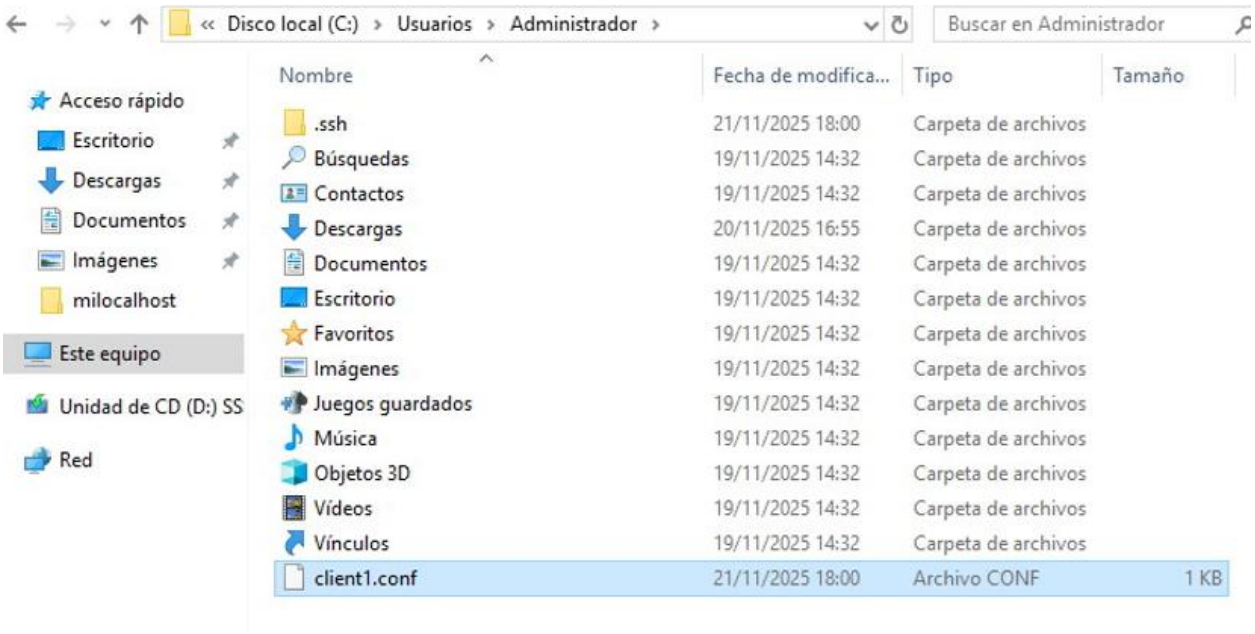
Desde el cliente nos descargamos el archivo de configuración de cliente que hemos hecho antes.

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

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

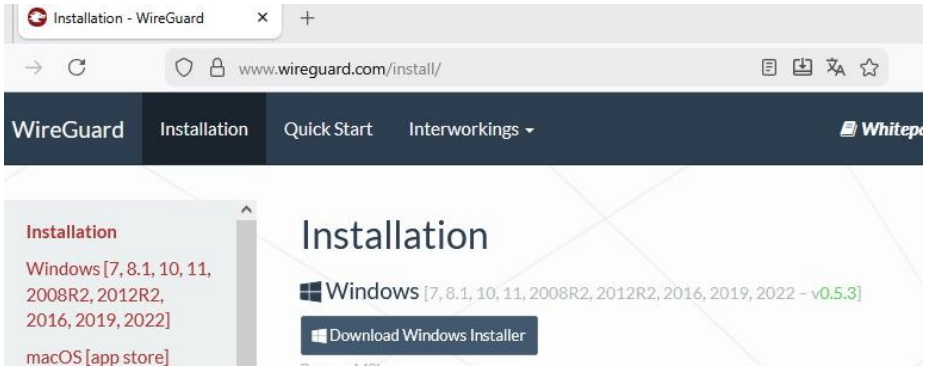
```
C:\Users\Administrador>scp cristobal@10.2.7.45:/home/cristobal/client1.conf .
The authenticity of host '10.2.7.45 (10.2.7.45)' can't be established.
ED25519 key fingerprint is SHA256:yj9bzz2ott5peUK8FhotSi+xIY44rycB2Ho6PtjuKXY.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])?
Warning: Permanently added '10.2.7.45' (ED25519) to the list of known hosts.
cristobal@10.2.7.45's password:
client1.conf                                  100%  293     0.3KB/s   00:00

C:\Users\Administrador>
```

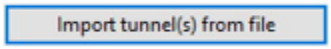


Nos descargamos el instalador. En nuestro caso el de Windows.

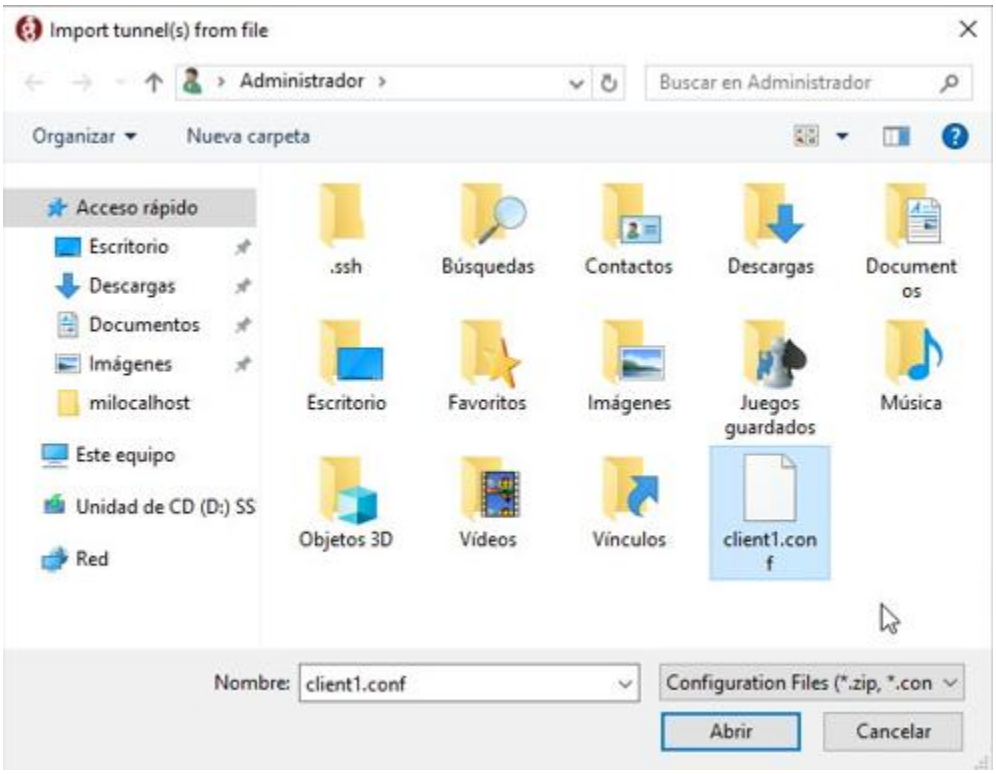
https://www.wireguard.com/install/



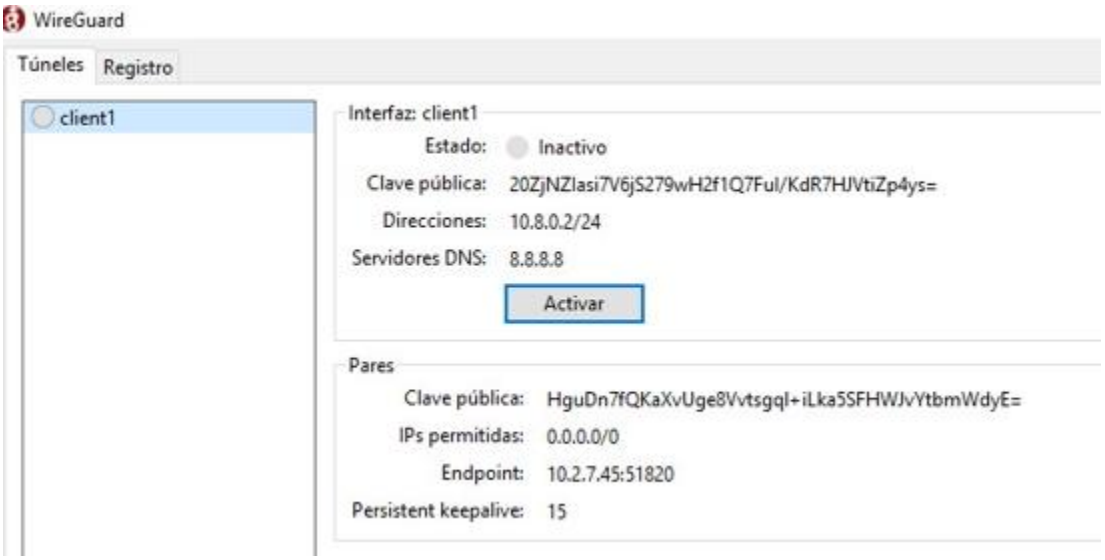
Una vez instalado. Le damos a importar túnel desde archivo.

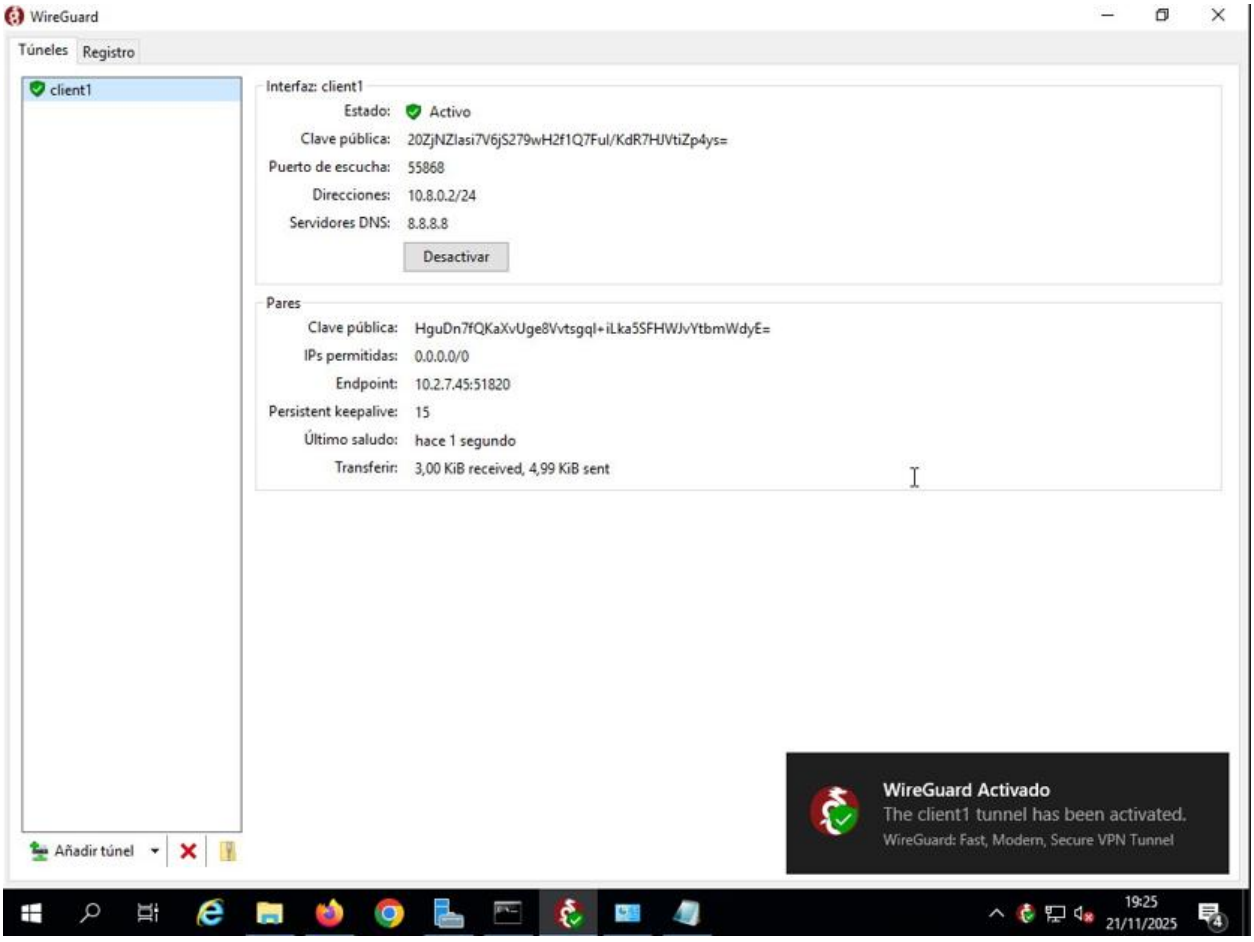


Seleccionamos el archivo.



Le damos a **activar**:





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.

```
C:\Users\Administrador>ipconfig

Configuración IP de Windows


Adaptador desconocido client1:

   Sufijo DNS específico para la conexión. . :
   Dirección IPv4. . . . . . . . . . . . . . : 10.8.0.2
   Máscara de subred . . . . . . . . . . . . : 255.255.255.0
   Puerta de enlace predeterminada . . . . . : 0.0.0.0

Adaptador de Ethernet Ethernet:

   Sufijo DNS específico para la conexión. . : institutodh.net
   Vínculo: dirección IPv6 local. . . . : fe80::5982:1509:5cda:3303%2
   Dirección IPv4. . . . . . . . . . . . . . : 10.2.17.111
   Máscara de subred . . . . . . . . . . . . : 255.255.255.0
   Puerta de enlace predeterminada . . . . . : 10.2.17.1
```

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

```
C:\Users\Administrador>ping 10.8.0.1

Haciendo ping a 10.8.0.1 con 32 bytes de datos:
Respuesta desde 10.8.0.1: bytes=32 tiempo=1ms TTL=64
Respuesta desde 10.8.0.1: bytes=32 tiempo=1ms TTL=64
Respuesta desde 10.8.0.1: bytes=32 tiempo=1ms TTL=64

Estadísticas de ping para 10.8.0.1:
    Paquetes: enviados = 3, recibidos = 3, perdidos = 0
    (0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 1ms, Máximo = 1ms, Media = 1ms
Control-C
^C
C:\Users\Administrador>_
```

Funciona.