# 3.7 WireGuard. Servidor de VPN




*Imagen obtenida de https://www.wireguard.com/*

# Esta herramienta sirve para...

crear una VPN de un modo extremadamente sencillo.

> Genial Pablo pero... ¿¡¿¡ para que quiero yo una VPN ?!?!

Hasta el momento hemos ido desplegando diferentes servicios a los que hemos asignado diferentes puertos y cuando nos hemos querido conectar a ellos hemos escrito la IP que tiene la Raspberry Pi **dentro de nuestra red** y el puerto que le hemos asignado en el fichero docker-compose. Ahora bien, es bastante probable que también queramos acceder a estos servicios desde **fuera de nuestra red**. Aquí básicamente se nos abren 2 posibilidades:

1. Acceder al router y "abrir" puertos.
2. Crear una VPN y conectarnos a ella.

Vamos a optar por la segunda opción por seguridad y comodidad. Al conectarnos a la VPN que creemos será como si estuviésemos conectados a la red de casa por lo que para conectarnos a nuestros servicios seguiremos utilizando la misma IP y puerto que en nuestro domicilio. Con ello conseguimos exponer menos puertos de nuestro router al exterior (**seguridad**) y no tener que

Copyright 2026

configurar nada en el router ni aprender nada (**comodidad**).

# Web de proyecto y otros enlaces de interés

Web del proyecto: https://www.wireguard.com/

Repositorio de código que podemos utilizar: https://github.com/linuxserver/docker-wireguard

# Despliegue

Como en ocasiones anteriores vamos a hacer con docker-compose para ello accedemos al terminal y escribimos

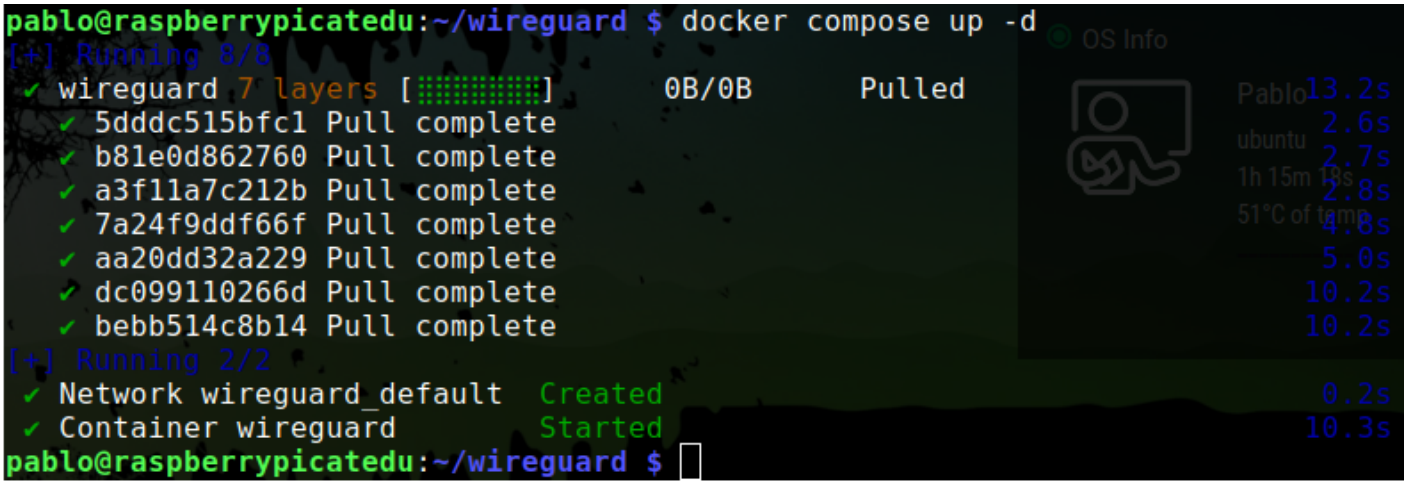
```
cd $HOME
mkdir wireguard
cd wireguard
nano docker-compose.yml
```

y dentro del fichero escribiremos el siguiente contenido

```
version: "2.1"
services:
  wireguard:
    image: lscr.io/linuxserver/wireguard:latest
    container_name: wireguard
    cap_add:
      - NET_ADMIN
      - SYS_MODULE
    environment:
      - PUID=1000
      - PGID=1000
```

```
      - TZ=Europe/Madrid
      - SERVERURL=vuestrodominio.duckdns.org #optional
      - SERVERPORT=51820 #optional
      - PEERS=1 #optional. Numero de personas que se vayan a conectar a la VPN
      - PEERDNS=auto #optional
      - INTERNAL_SUBNET=10.13.13.0 #optional
      - ALLOWEDIPS=0.0.0.0/0 #optional
      - PERSISTENTKEEPALIVE_PEERS= #optional
      - LOG_CONFS=true #optional
    volumes:
      - ./config:/config
      - /lib/modules:/lib/modules #optional
    ports:
      - 51820:51820/udp
    sysctls:
      - net.ipv4.conf.all.src_valid_mark=1
    restart: unless-stopped
```

como en ocasiones anteriores, para guardar los cambios pulsaremos control + x y cuando nos pregunte aceptaremos. Una vez volvamos a estar en el terminal, escribiremos `docker compose up -d` para lanzar los servicios ubicados dentro del fichero docker-compose. El resultado será similar a:




*Elaboración propia*

# Funcionamiento

Si prestamos atención al fichero docker-compose veremos que, en el apartado `volumes`, hemos creado uno volumen llamado `config`. Si desde `$HOME/wireguard` listamos el contenido del directorio con `ls -l` veremos que hay un directorio llamado config. Si accedemos al mismo `cd config` y listamos el contendio veremos que se han creado tantas carpetas `peerX` como PEERS hayamos establecido en el fichero docker-compose. En mi caso tengo 3. Si accedemos a una de esas carpetas dentro hay 2 ficheros relevantes los ficheros peerx.conf y peerx.png. El 1º tiene la configuración del fichero para conectarnos a la VPN con esos datos y el 2º tiene una imagen con un código QR que, una vez escaneado, nos configura directamente la VPN.

```
pi@mediacenter:~/wireguard $ ls -la
total 20
drwxr-xr-x  4 pi   pi   4096 feb  2 10:10 .
drwxr-xr-x 27 pi   pi   4096 feb 27 09:17 ..
drwxr-xr-x  8 pi   pi   4096 feb  2 10:11 config
-rw-r--r--  1 pi   pi    714 feb  2 10:10 docker-compose.yml
drwxr-xr-x  3 root root 4096 dic 10 16:33 lib
pi@mediacenter:~/wireguard $ cd config/
pi@mediacenter:~/wireguard/config $ ls -la
total 40
drwxr-xr-x 8 pi pi 4096 feb  2 10:11 .
drwxr-xr-x 4 pi pi 4096 feb  2 10:10 ..
drwxr-xr-x 2 pi pi 4096 dic 10 16:33 coredns
-rw------- 1 pi pi  163 feb  2 10:11 .donoteditthisfile
drwx------ 2 pi pi 4096 dic 10 16:33 peer1
drwxr-xr-x 2 pi pi 4096 feb  2 10:11 peer2
drwx------ 2 pi pi 4096 feb  2 10:11 peer3
drwxr-xr-x 2 pi pi 4096 dic 10 16:33 server
drwxr-xr-x 2 pi pi 4096 dic 10 16:33 templates
-rw------- 1 pi pi  865 feb  2 10:11 wg0.conf
pi@mediacenter:~/wireguard/config $ cd peer1/
pi@mediacenter:~/wireguard/config/peer1 $ ls -la
total 28
drwx------ 2 pi pi 4096 dic 10 16:33 .
drwxr-xr-x 8 pi pi 4096 feb  2 10:11 ..
-rw------- 1 pi pi  319 feb  2 10:11 peer1.conf
-rw------- 1 pi pi 1134 feb  2 10:11 peer1.png
-rw------- 1 pi pi   45 dic 10 16:33 presharedkey-peer1
-rw------- 1 pi pi   45 dic 10 16:33 privatekey-peer1
-rw------- 1 pi pi   45 dic 10 16:33 publickey-peer1
pi@mediacenter:~/wireguard/config/peer1 $ cat peer1.conf
```

*Elaboración propia*

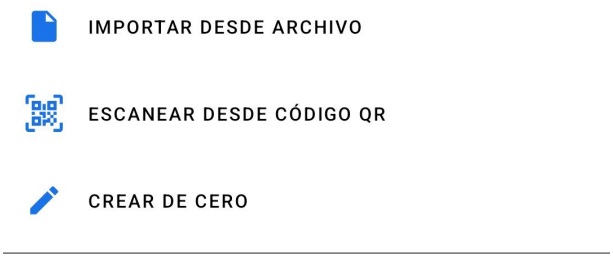
# Configuración desde el teléfono móvil

Desde nuestro teléfono Android accedemos a https://play.google.com/store/apps/details?id=com.wireguard.android e instalamos el cliente de VPN. Una vez instalada la APP pulsamos en el símbolo + y seleccionamos escanear desde código QR. Escaneamos el fichero png comentado en el párrafo anterior y ya está configurada la conexión. Ahora, cada vez que queramos conectarnos a nuestra VPN desde fuera de nuestra red activaremos la VPN y estaremos a todos los efectos conectados a nuestra red. Dejo una serie de capturas de pantalla del proceso.






*Elaboración propia*




*Elaboración propia*




*Elaboración propia*




*Elaboración propia*

Una vez hechos todos los pasos anteriores **y con la VPN activa** únicamente deberemos introducir en el navegador la IP que tiene nuestra Raspberry **en nuestra red local** y el puerto del servicio al que queramos acceder. De este modo nos estaremos conectando a este servicio desde fuera de nuestra red como si estuviéramos en ella.

Revision #8
Created 2023-02-04 10:06:07 CET by Pablo Ruiz
Updated 2023-07-20 17:32:14 CEST by Pablo Ruiz

Copyright 2026