WG-easy

1. Introducción

WG-Easy es una solución "todo en uno" que incluye el servidor WireGuard y una interfaz web muy sencilla para crear y gestionar clientes.

El objetivo de este despliegue es conectar la VPN a nuestra red interna (`wg_pihole`) para forzar que todos los clientes utilicen nuestro Pi-hole como servidor DNS.

2. Configuración de Red (Router)

Para que podamos conectarnos a la VPN desde fuera de casa, es obligatorio abrir el puerto en el router (NAT/Port Forwarding) y dirigirlo a la IP fija del servidor (`192.168.1.X`).

Datos a configurar en el Router:

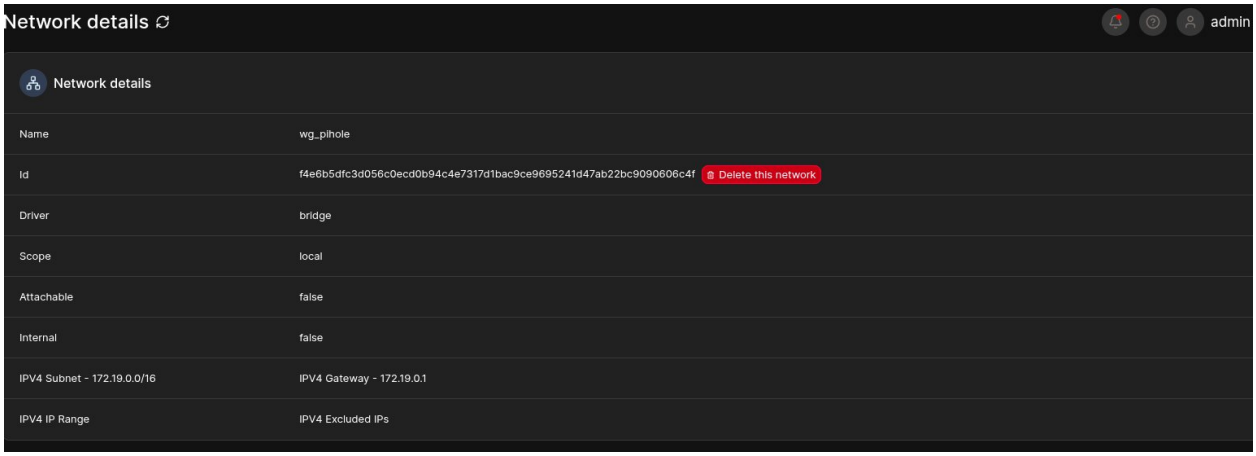
- **Puerto Externo:** 51820
- **Puerto Interno:** 51820
- **Protocolo:** UDP
- **IP Destino:** La IP local de tu servidor

3. Verificación de la Red Interna

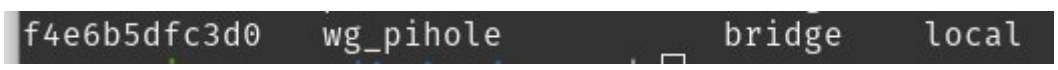
Antes de empezar, debemos asegurarnos de que la red interna creada en el manual de Pi-hole sigue activa, ya que conectaremos la VPN a ella.

Comando a ejecutar: "docker network ls | grep wg_pihole"

Con Portainer:



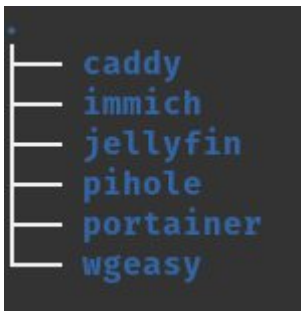
Con terminal:



4. Estructura de Directorios

Creamos la carpeta para la persistencia de la configuración de WireGuard.

Comando a ejecutar: "mkdir -p ~/docker/wgeasy"



5. Configuración de Seguridad (UFW)

Necesitamos abrir dos puertos:

- El puerto UDP para el túnel VPN (Tráfico de datos).
- El puerto TCP para acceder al panel de administración web (Gestión de usuarios).

Comandos a ejecutar: "sudo ufw allow 51820/udp comment 'WireGuard VPN Tunnel'" "sudo ufw allow 51821/tcp comment 'WG-Easy Web UI'"

```
51820/udp                  ALLOW       Anywhere                   # WireGuard servicio

51821/tcp                  ALLOW       Anywhere                   # WireGuard web
```

6. Archivo de Despliegue (docker-compose.yml)

Creamos el archivo `docker-compose.yml`. **Puntos clave a editar:**

- **WG_HOST:** Debes poner tu IP pública o tu dominio (ej. `midominio.duckdns.org`).
- **PASSWORD:** La contraseña para entrar al panel web.
- **networks:** Conectamos a la red `wg_pihole`.

```yaml
services:
  wg-easy:
    environment:
    #  Optional:
    #  - PORT=51821
    #  - HOST=0.0.0.0
      - INSECURE=true
      - WG_HOST=90.71.170.173

    image: ghcr.io/wg-easy/wg-easy:15
    container_name: wg-easy
   #network_mode: "host"
    volumes:
      - etc_wireguard:/etc/wireguard
      - /lib/modules:/lib/modules:ro
    ports:
      - "51820:51820/udp"
      - "51821:51821/tcp"
    restart: unless-stopped
    cap_add:
      - NET_ADMIN
      - SYS_MODULE
      # - NET_RAW # ⚠️Uncomment if using Podman
    networks:
      - wg_pihole

volumes:
  etc_wireguard:

networks:
  wg_pihole:
    external: true
```

7. Despliegue y Verificación

Levantamos el servicio.

Comando a ejecutar: "docker compose up -d"

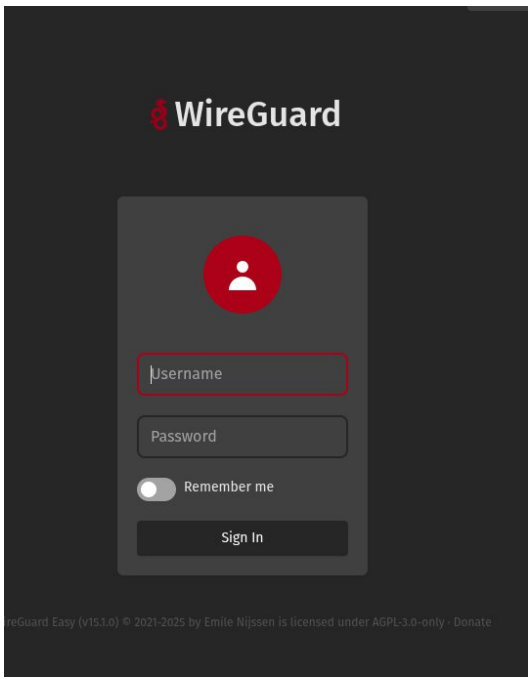
Con Portainer:

Con terminal:

```
eros@saiserver:~/docker/wgeasy$ docker ps | grep 'wg-easy'
a72d2591dd8c   ghcr.io/wg-easy/wg-easy:15                                "docker-entr
ypoint.s…"   20 minutes ago   Up 20 minutes (healthy)   0.0.0.0:51820->51820/udp, [::]:51820
->51820/udp, 0.0.0.0:51821->51821/tcp, [::]:51821->51821/tcp
            wg-easy
```

8. Acceso al Panel

Ahora podemos entrar a la dirección `http://localhost:51821` (o la IP de tu servidor) para crear el primer cliente VPN. Al crearlo, este cliente ya tendrá configurado automáticamente el DNS de Pi-hole.


WireGuard
Username
Password
Remember me
Sign In
reGuard Easy (v15.1.0) © 2021-2025 by Emile Nijssen is licensed under AGPL-3.0-only · Donate