

## VPN WireGuard amb Docker

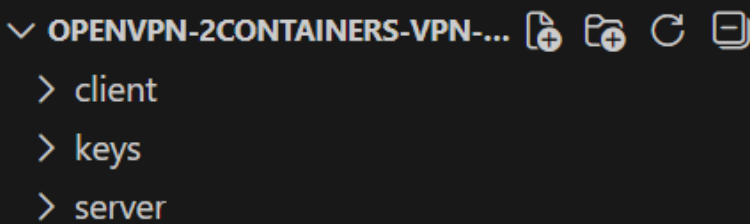
<b>Curs</b>	25/26	<b>Grup</b>	IFC31C	<b>Data lliurament</b>	04/Febrer
<b>Mòdul</b>	SISAD				
<b>Títol</b>	VPN WireGuard amb Docker				

<b>Tipus de treball</b>	Individual
<b>Pautes de realització</b>	<b>Toni Crespí Mulet</b>
<ol style="list-style-type: none"> <li>1. Crear dues imatges Docker pròpies (server i client) amb Ubuntu base</li> <li>2. Instal·lar WireGuard manualment dins dels contenidors</li> <li>3. Configurar una VPN punt-a-punt segura entre els contenidors</li> <li>4. Practicar resolució d'errors i troubleshooting</li> <li>5. Verificar la comunicació i fer diagnòstic de la xarxa</li> </ol> <p>Contenidors construïts correctament WireGuard instal·lat manualment Configuració wg0.conf correcta (server i client) Connexió VPN funcionant (ping, accés a servei)</p>	

## 1. Preparació de l'entorn

Primer de tot, treballarem dins l'entorn del Visual Studio Code, on primerament crearem les carpetes → **mkdir -p server client keys**

```
toni_crespi@DESKTOP-1PIFMMB:/mnt/c/Users/Toni Crespi Mulet/Desktop/openvpn-2containers-vpn-individual$ mkdir -p server client keys
toni_crespi@DESKTOP-1PIFMMB:/mnt/c/Users/Toni Crespi Mulet/Desktop/openvpn-2containers-vpn-individual$
```



## 2. Generació de les claus (Hosts)

Ens ficam a la carpeta que acabam de crear → **cd keys**

```
toni_crespi@DESKTOP-1PIFMMB:/mnt/c/Users/Toni Crespi Mulet/Desktop/openvpn-2containers-vpn-individual$ cd keys
toni_crespi@DESKTOP-1PIFMMB:/mnt/c/Users/Toni Crespi Mulet/Desktop/openvpn-2containers-vpn-individual/keys$
```

Seguidament, generarem les claus del Servidor → **wg genkey | tee server.key | wg pubkey > server.pub**

```
toni_crespi@DESKTOP-1PIFMMB:/mnt/c/Users/Toni Crespi Mulet/Desktop/openvpn-2containers-vpn-individual/keys$ wg genkey | tee server.key | wg pubkey > server.pub
toni_crespi@DESKTOP-1PIFMMB:/mnt/c/Users/Toni Crespi Mulet/Desktop/openvpn-2containers-vpn-individual/keys$
```

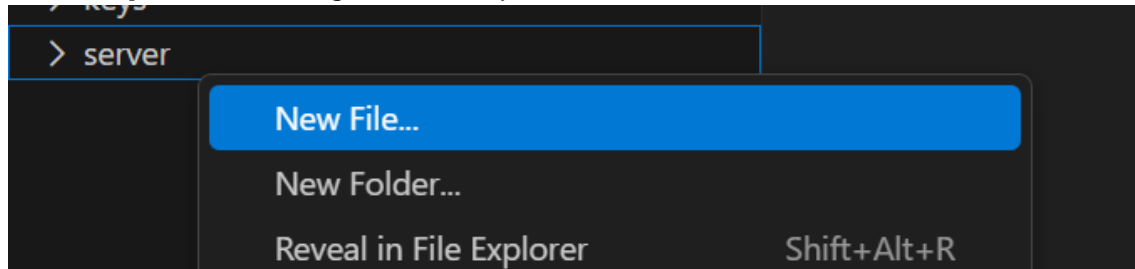
Generam les claus dels client → **wg genkey | tee client.key | wg pubkey > client.pub**

```
toni_crespi@DESKTOP-1PIFMMB:/mnt/c/Users/Toni Crespi Mulet/Desktop/openvpn-2containers-vpn-individual/keys$ wg genkey | tee client.key | wg pubkey > client.pub
toni_crespi@DESKTOP-1PIFMMB:/mnt/c/Users/Toni Crespi Mulet/Desktop/openvpn-2containers-vpn-individual/keys$
```

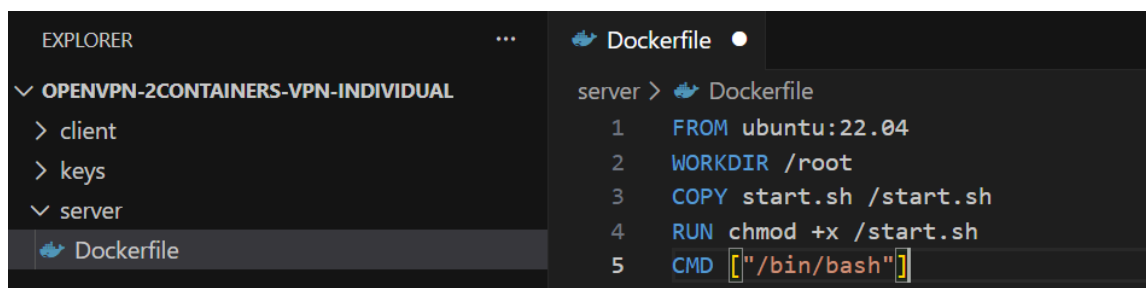
Ara torarem a l'arrel del projecte → **cd ..**

### 3. Creació dels fitxers de configuració

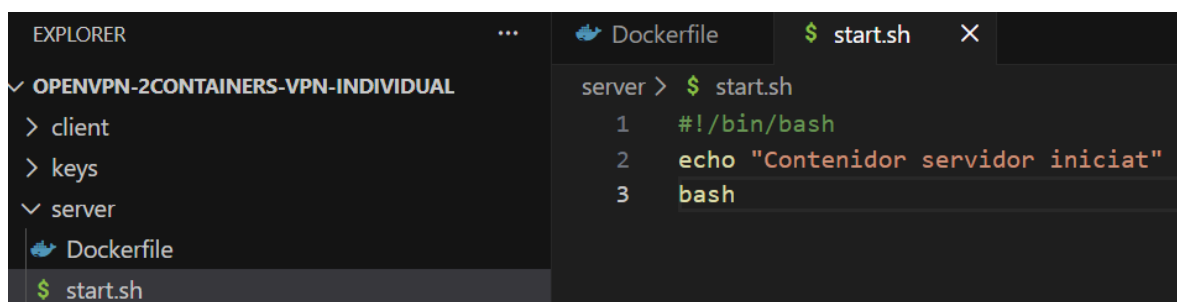
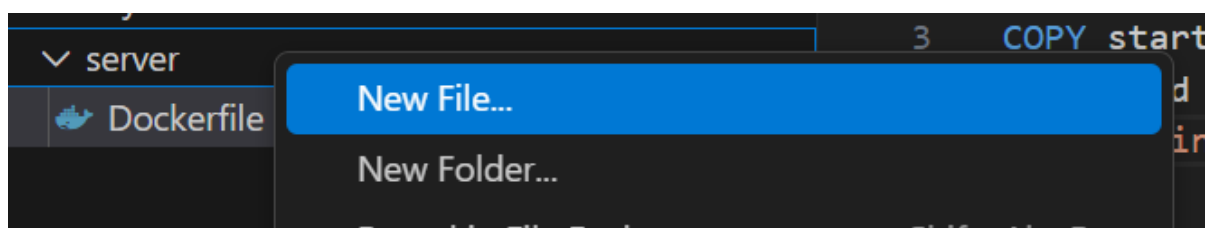
I començarem a configurar la carpeta de **server** →



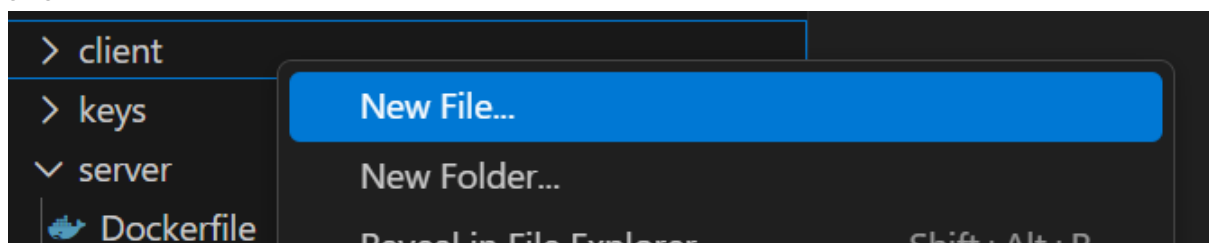
Començam amb l'arxiu **Dockerfile**



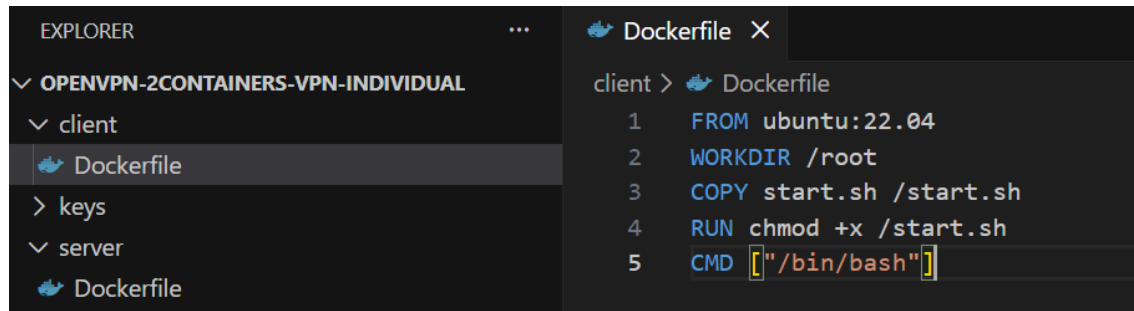
I ara amb l'**start.sh**



Una vegada configurada la carpeta de server, passarem amb la configuració del **client** →



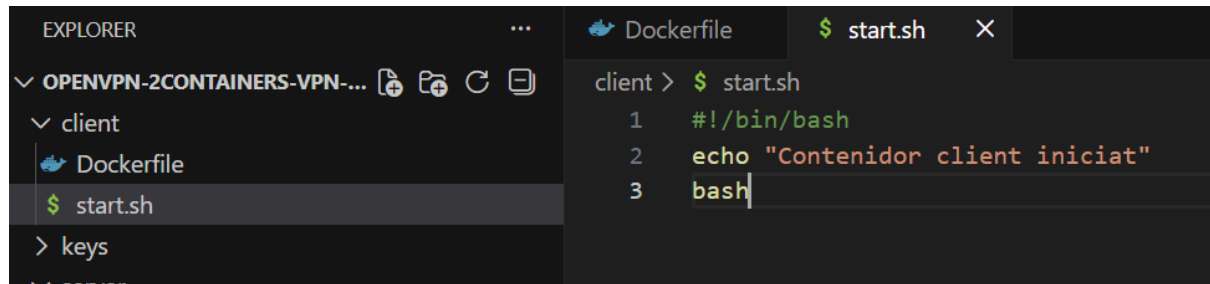
## Dockerfile



The screenshot shows the VS Code interface. On the left, the Explorer sidebar is open, showing a project structure with folders 'client', 'keys', and 'server'. The 'client' folder is expanded, and the 'Dockerfile' file is selected. On the right, the Dockerfile content is displayed in the editor:

```
client > Dockerfile
1 FROM ubuntu:22.04
2 WORKDIR /root
3 COPY start.sh /start.sh
4 RUN chmod +x /start.sh
5 CMD ["/bin/bash"]
```

## Start.sh

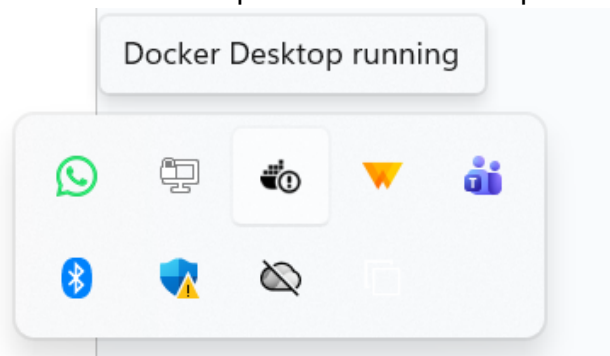


The screenshot shows the VS Code interface. On the left, the Explorer sidebar is open, showing the same project structure. The 'start.sh' file is selected. On the right, the start.sh content is displayed in the editor:

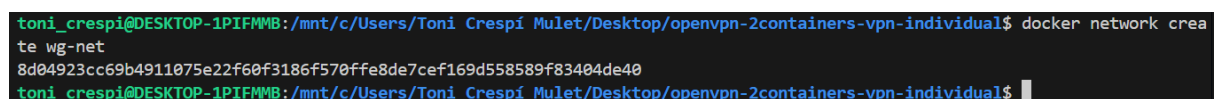
```
client > $ start.sh
1 #!/bin/bash
2 echo "Contenidor client iniciat"
3 bash
```

## 4. Construcció i Execució dels Contenedors

Utilitzarem Docker per aixecar les màquines. És important tenir el docker encès:



Creem la xarxa → **docker network create wg-net**



The screenshot shows a terminal window with the following text:

```
toni_crespi@DESKTOP-1PIFMMB:/mnt/c/Users/Toni Crespi Mulet/Desktop/openvpn-2containers-vpn-individual$ docker network crea
te wg-net
8d04923cc69b4911075e22f60f3186f570ffe8de7cef169d558589f83404de40
toni_crespi@DESKTOP-1PIFMMB:/mnt/c/Users/Toni Crespi Mulet/Desktop/openvpn-2containers-vpn-individual$
```

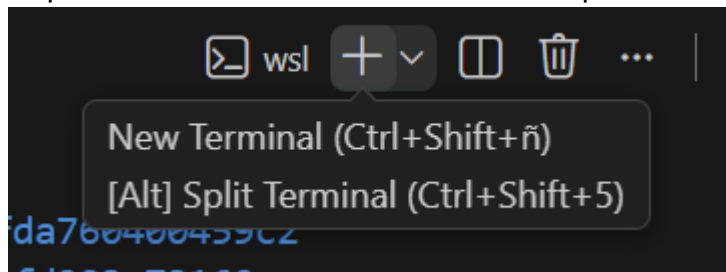
Construïm les imatges, tant del servidor com del client → **docker build -t wg-server ./server**  
**docker build -t wg-client ./client**

```
toni_crespi@DESKTOP-1PIFMMB:/mnt/c/Users/Toni Crespi Mulet/Desktop/openvpn-2containers-vpn-individual$ docker build -t wg-server ./server
docker build -t wg-client ./client
[+] Building 2.5s (9/9) FINISHED
=> [internal] load build definition from Dockerfile 0.0s
=> => transferring dockerfile: 137B 0.0s
=> [internal] load metadata for docker.io/library/ubuntu:22.04 1.6s
```

Arrencam el servidor → **docker run -it --name wireguard-server --network wg-net --cap-add=NET\_ADMIN --cap-add=SYS\_MODULE -p 51820:51820/udp wg-server**

```
toni_crespi@DESKTOP-1PIFMMB:/mnt/c/Users/Toni Crespi Mulet/Desktop/openvpn-2containers-vpn-individual$ docker run -it --name wireguard-server --network wg-net --cap-add=NET_ADMIN --cap-add=SYS_MODULE -p 51820:51820/udp wg-server
root@2630181bbd7a:~#
```

El que farem ara serà arrencar el client però a una terminal diferent →



I executam → **docker run -it --name wireguard-client --network wg-net --cap-add=NET\_ADMIN wg-client**

```
toni_crespi@DESKTOP-1PIFMMB:/mnt/c/Users/Toni Crespi Mulet/Desktop/openvpn-2containers-vpn-individual$ docker run -it --name wireguard-client --network wg-net --cap-add=NET_ADMIN wg-client
root@6be9c4d43ac5:~#
```

## 5. Instal·lació Manual i Configuració

Instal·lam els paquets a la terminal del servidor → **apt update && apt install wireguard iproute2 iptables nano -y**

```
root@2630181bbd7a:~# apt update && apt install wireguard iproute2 iptables nano -y
Get:1 http://archive.ubuntu.com/ubuntu jammy InRelease [270 kB]
Get:2 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Get:3 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 Packages [6396 kB]
Get:4 http://archive.ubuntu.com/ubuntu jammy-updates InRelease [128 kB]
Get:5 http://archive.ubuntu.com/ubuntu jammy-backports InRelease [127 kB]
Get:6 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 Packages [62.6 kB]
Get:7 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [1297 kB]
Get:8 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [3683 kB]
Get:9 http://archive.ubuntu.com/ubuntu jammy/universe amd64 Packages [17.5 MB]
```

I activam IP Forwarding → **sysctl -w net.ipv4.ip\_forward=1**

```
root@75f9513550cc:~# sysctl -w net.ipv4.ip_forward=1
net.ipv4.ip_forward = 1
root@75f9513550cc:~#
```

Creem la carpeta i editam la configuració → **mkdir -p /etc/wireguard**  
**nano /etc/wireguard/wg0.conf**

El que farem serà anar a la clau privada del Server i la clau pública del client

```
GNU nano 6.2 /etc/wireguard/wg0.conf *
ListenPort = 51820
PrivateKey = KltvMI8WFUA/h2879KHnxwffto7xwldwewOx6ORYj0Vw=

[Peer]
PublicKey = 4KOYHOyUluCxxhn5fZeGdFUl66y//v8m1kohH97i43Q=
AllowedIPs = 10.9.0.2/32

^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location  M-U Undo
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^/ Go To Line M-E Redo
Ln 1, Col 1 (44 selected) Spaces: 4 UTF-8 LF {} Plain Text
```

Aixecam VPN al Server → **wg-quick up wg0**

```
root@75f9513550cc:~# wg-quick up wg0
[#] ip link add wg0 type wireguard
[#] wg setconf wg0 /dev/fd/63
[#] ip -4 address add 10.9.0.1/24 dev wg0
[#] ip link set mtu 1420 up dev wg0
root@75f9513550cc:~#
```

És el torn de configurar el Client, ens movem a la terminal que hem afegit abans i començam executant aquesta comanda → **apt update && apt install wireguard iproute2 iputils-ping nano -y**

```
root@6be9c4d43ac5:~# apt update && apt install wireguard iproute2 iputils-ping nano -y
Get:1 http://archive.ubuntu.com/ubuntu jammy InRelease [270 kB]
Get:2 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Get:3 http://archive.ubuntu.com/ubuntu jammy-updates InRelease [128 kB]
Get:4 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [1297 kB]
Get:5 http://archive.ubuntu.com/ubuntu jammy-backports InRelease [127 kB]
0% [4 Packages 1069 kB/1297 kB 82%]
```

Creem carpeta i editam la configuració → **mkdir -p /etc/wireguard**  
**nano /etc/wireguard/wg0.conf**

I en aquesta configuració el que hem fer és, primer aferrar la clau privada del **Client** i després la clau pública del **Servidor**

```
GNU nano 6.2 /etc/wireguard/wg0.conf
[Interface]
Address = 10.9.0.2/24
PrivateKey = 4KOYHOyUluCxxhn5fZeGdFU166y//v8m1kohH97i43Q=

[Peer]
PublicKey = 5jrXEs8EXH51VCpVmwdnQ4xmezrSw4fb00Xg0zAJ9hU=
[ Read 9 lines ]
^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^_/ Go To Line  M-E Redo
```

Una vegada configurat el fitxer, aixecam VPN al Client→ **wg-quick up wg0**

```
root@6be9c4d43ac5:~# wg-quick up wg0
[#] ip link add wg0 type wireguard
[#] wg setconf wg0 /dev/fd/63
[#] ip -4 address add 10.9.0.2/24 dev wg0
[#] ip link set mtu 1420 up dev wg0
root@6be9c4d43ac5:~#
```

## 6. Verificació Final

Des del terminal del **Client**, farem un ping a la IP del servidor VPN→ **ping 10.9.0.1**

```
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=1.03 ms
64 bytes from 10.8.0.1: icmp_seq=2 ttl=64 time=0.904 ms
64 bytes from 10.8.0.1: icmp_seq=3 ttl=64 time=1.08 ms
^C
--- 10.8.0.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2000ms
rtt min/avg/max/mdev = 0.904/1.004/1.082/0.074 ms
```

Enllaç GitHub →

<https://github.com/toni-crespi/practica-openvpn-docker-individual.git>

