Docker-Compose with Mullvad Wireguard & arbitrary service

I was able to make this work really easily in native Kubernetes pods, but lots of folks had been asking questions about getting Wireguard connected to an arbitrary service properly and safely that may not have the means to use that infrastructure. Below are my notes on making that dream a reality with only compose and a few minutes of trial and error.

This compose shows wireguard + qbittorrent with some useful notes in-line. The crux of it though is as follows:

1. Move the exposed ports off the qbittorrent service definition, and into the wireguard definition
2. Add network_mode: "service:wireguard" to force the containers to use the same interfaces.

```
version: "3.7"
services:
  wireguard:
    image: linuxserver/wireguard
    container_name: wireguard
    cap_add:
      - NET_ADMIN
      - SYS_MODULE
    environment:
      - PUID=1000
      - PGID=1000
      - TZ=Europe/London
    volumes:
      - /appdata/config/wireguard-test/wg:/config
      - /lib/modules:/lib/modules
    ports:
```

```
      - 6881:6881
      - 6881:6881/udp
      - 8088:8088
    sysctls:
      - net.ipv4.conf.all.src_valid_mark=1
    restart: unless-stopped
qbittorrent:
image: linuxserver/qbittorrent
container_name: qbittorrent
environment:
- PUID=1000
- PGID=1000
- TZ=Europe/London
- UMASK_SET=022
#Remember to make this the same port as the exposed port
- WEBUI_PORT=8088
volumes:
- /appdata/config/wireguard-test/qbt:/config
- /appdata/downloads:/downloads
#"ports" moved to wireguard config
restart: unless-stopped
#use the wireguard interfaces instead
network_mode: "service:wireguard"
```

In the wireguard `wg0.conf` configuration, you must add a route back to your host network **only if you want to access things** like webUIs from your host. If everything's in the same network, you can just leave this headless, too.

`PostUp = ip route add 192.168.0.0/16 via $(ip route | grep default | awk '{print $3}')`

```
[Interface]
PrivateKey = <MULLVAD KEY>
Address = <MULLVAD ADDRESS>
DNS = <MULLVAD DNS>
PostUp = DROUTE=$(ip route | grep default | awk '{print $3}'); HOMENET=192.168.0.0/16;
HOMENET2=10.0.0.0/8; HOMENET3=172.16.0.0/12; ip route add $HOMENET3 via $DROUTE;ip route add
$HOMENET2 via $DROUTE; ip route add $HOMENET via $DROUTE;iptables -I OUTPUT -d $HOMENET -j
ACCEPT;iptables -A OUTPUT -d $HOMENET2 -j ACCEPT; iptables -A OUTPUT -d $HOMENET3 -j ACCEPT;  iptables -
A OUTPUT ! -o %i -m mark ! --mark $(wg show %i fwmark) -m addrtype ! --dst-type LOCAL -j REJECT
PreDown = HOMENET=192.168.0.0/16; HOMENET2=10.0.0.0/8; HOMENET3=172.16.0.0/12; ip route del
```

```
$HOMENET3 via $DROUTE;ip route del $HOMENET2 via $DROUTE; ip route del $HOMENET via $DROUTE;
iptables -D OUTPUT ! -o %i -m mark ! --mark $(wg show %i fwmark) -m addrtype ! --dst-type LOCAL -j REJECT;
iptables -D OUTPUT -d $HOMENET -j ACCEPT; iptables -D OUTPUT -d $HOMENET2 -j ACCEPT; iptables -D OUTPUT
-d $HOMENET3 -j ACCEPT
[Peer]
PublicKey = jHxY2OKpxjqAwWH4r1Pb2K6xDUDt087ivxpM1KpE0Ec=
AllowedIPs = 0.0.0.0/0
Endpoint = <MULLVAD SERVER>:51820
```

Pretty simple, right? Here's the results of what you came here to see.

```
root@f316f4f274fb:/# curl https://am.i.mullvad.net/connected
You are connected to Mullvad (server us32-wireguard). Your IP address is 206.217.xxx.xxx
```

If you're curious about the nitty gritty, here's the output from each containers interfaces & routes to give an illustration on how this works as if it were on the same host instead of dedicated network stacks:

From Wireguard

```
root@347666d9f127:/# ps -ef
UID        PID  PPID  C STIME TTY          TIME CMD
root         1     0  0 16:52 ?        00:00:00 s6-svscan -t0 /var/run/s6/services
root        32     1  0 16:52 ?        00:00:00 s6-supervise s6-fdholderd
root       265     1  0 16:52 ?        00:00:00 s6-supervise coredns
root       266     1  0 16:52 ?        00:00:00 s6-supervise wireguard
root       268   266  0 16:52 ?        00:00:00 bash ./run
root       270   265  0 16:52 ?        00:00:00 /app/coredns -dns.port=53
root       357   268  0 16:52 ?        00:00:00 sleep infinity
root       358     0  0 16:59 pts/0    00:00:00 bash
root       378   358  0 17:00 pts/0    00:00:00 ps -ef
root@347666d9f127:/# ip route
default via 172.24.0.1 dev eth0
172.24.0.0/16 dev eth0 proto kernel scope link src 172.24.0.2
192.168.0.0/16 via 172.24.0.1 dev eth0
```

root@347666d9f127:/# ip address

1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000

link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00

inet 127.0.0.1/8 scope host lo

valid_lft forever preferred_lft forever

3: wg0: <POINTOPOINT,NOARP,UP,LOWER_UP> mtu 1420 qdisc noqueue state UNKNOWN group default qlen 1000

link/none

inet 10.67.xxx.xx/32 scope global wg0

valid_lft forever preferred_lft forever

151: eth0@if152: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default

link/ether 02:42:ac:18:00:02 brd ff:ff:ff:ff:ff:ff link-netnsid 0

inet 172.24.0.2/16 brd 172.24.255.255 scope global eth0

valid_lft forever preferred_lft forever

root@347666d9f127:/# iptables-save

Generated by iptables-save v1.6.1 on Sat Aug 8 14:48:00 2020

*filter

:INPUT ACCEPT [16:2307]

:FORWARD ACCEPT [0:0]

:OUTPUT ACCEPT [17:1615]

-A OUTPUT -d 192.168.0.0/16 -j ACCEPT

-A OUTPUT -d 10.0.0.0/8 -j ACCEPT

-A OUTPUT -d 172.16.0.0/12 -j ACCEPT

-A OUTPUT ! -o wg0 -m mark ! --mark 0xca6c -m addrtype ! --dst-type LOCAL -j REJECT --reject-with icmp-port-unreachable

COMMIT

Completed on Sat Aug 8 14:48:00 2020

Generated by iptables-save v1.6.1 on Sat Aug 8 14:48:00 2020

*mangle
:PREROUTING ACCEPT [16:2307]
:INPUT ACCEPT [16:2307]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [19:1729]
:POSTROUTING ACCEPT [19:1729]
-A PREROUTING -p udp -m comment --comment "wg-quick(8) rule for wg0" -j CONNMARK --restore-mark --nfmask 0xffffffff --ctmask 0xffffffff
-A POSTROUTING -p udp -m mark --mark 0xca6c -m comment --comment "wg-quick(8) rule for wg0" -j CONNMARK --save-mark --nfmask 0xffffffff --ctmask 0xffffffff
COMMIT

Completed on Sat Aug 8 14:48:00 2020

Generated by iptables-save v1.6.1 on Sat Aug 8 14:48:00 2020

*raw
:PREROUTING ACCEPT [16:2307]
:OUTPUT ACCEPT [19:1729]
-A PREROUTING -d 10.67.xxx.xxx/32 ! -i wg0 -m addrtype ! --src-type LOCAL -m comment --comment "wg-quick(8) rule for wg0" -j DROP
COMMIT

Completed on Sat Aug 8 14:48:00 2020

From qbittorrent

```
root@347666d9f127:/# ps -ef
UID        PID  PPID  C STIME TTY          TIME CMD
root         1     0  0 16:52 ?        00:00:00 s6-svscan -t0 /var/run/s6/services
root        32     1  0 16:52 ?        00:00:00 s6-supervise s6-fdholderd
root       250     1  0 16:52 ?        00:00:00 s6-supervise qbittorrent
abc        252   250  0 16:52 ?        00:00:02 /usr/bin/qbittorrent-nox --webui-port=8088
root       276     0  0 17:00 pts/0    00:00:00 bash
root       669   276  0 17:02 pts/0    00:00:00 ps -ef
root@347666d9f127:/# ip route
default via 172.24.0.1 dev eth0
172.24.0.0/16 dev eth0 proto kernel scope link src 172.24.0.2
192.168.0.0/16 via 172.24.0.1 dev eth0
```

```
root@347666d9f127:/# ip address
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
inet 127.0.0.1/8 scope host lo
valid_lft forever preferred_lft forever
3: wg0: <POINTOPOINT,NOARP,UP,LOWER_UP> mtu 1420 qdisc noqueue state UNKNOWN group default qlen 1000
link/none
inet 10.67.xx.xx scope global wg0
valid_lft forever preferred_lft forever
151: eth0@if152: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
link/ether 02:42:ac:18:00:02 brd ff:ff:ff:ff:ff:ff link-netnsid 0
inet 172.24.0.2/16 brd 172.24.255.255 scope global eth0
valid_lft forever preferred_lft forever
```

Sources:

https://nbsoftsolutions.com/blog/routing-select-docker-containers-through-wireguard-vpn

---

Revision #9
Created 2 August 2020 15:48:02 by Tokugero
Updated 17 January 2021 18:34:05 by Tokugero