

Gluetun Docker Stack

Gluetun is a docker image for several VPN providers. It can be used to route specific other docker services through the configured VPN while the main system remains unaffected.

The following is a docker compose file using gluetun as VPN and http proxy and several additional services with auto heal capability to restart the services if they stop responding.

This stack includes:

- autoheal
- qbittorrent
- prowlarr
- transmission
- flaresolvert
- stash
- radarr
- sonarr
- jdownloader2
- socks5

The whole stack uses the network of the gluetun service and all other services can interact with each other through localhost:port. The external ports are all configured in the gluetun service.

sonarr and radarr are dependent on prowlarr, all services (except for autoheal) are dependent on gluetun to ensure the VPN is started first.

Note that volume paths need to be adjusted and services can be removed as required.

The http or socks5 proxies can be used by other devices in the network (browser settings / command line settings etc) as necessary. If only specific domains should be routed through the VPN but all others should bypass it a custom proxy auto configuration could be used like [Firefox/Chrome Proxy](#)

[docker-gluetun.yml](#)

```
name: gluetun
services:
  gluetun:
    container_name: gluetun
    hostname: gluetun
    image: qmcgaw/gluetun
    # container_name: gluetun
    # line above must be uncommented to allow external containers to
    connect.
    # See
    https://github.com/qdm12/gluetun-wiki/blob/main/setup/connect-a-contain
    er-to-gluetun.md#external-container-to-gluetun
    cap_add:
      - NET_ADMIN
    devices:
```

```
- /dev/net/tun:/dev/net/tun
ports:
- 8888:8888/tcp # HTTP proxy
  #- 8388:8388/tcp # Shadowsocks
  #- 8388:8388/udp # Shadowsocks
- 9092:9092 # qBittorrent web
- 6881:6881 # qBittorrent bt data
- 6881:6881/udp # qBittorrent bt data
- 9696:9696 # Prowlarr
- 9091:9091 # Transmission web
- 51413:51413 # Transmission bt data
- 51413:51413/udp # Transmission bt data
- 8191:8191 # Flaresolverr
- 8488:1080 # socks5 proxy
- 9998:9998 # Stash3D
- 8310:8310 # Radarr
- 8989:8989 # Sonarr
- 5800:5800 # JDownloader2 web interface port
- 5900:5900 # JDownloader2 vnc port
- 3129:3129 # JDownloader2 myjdownloader port see
https://github.com/jlesage/docker-jdownloader-2#direct-connection

volumes:
- /opt/gluetun/config:/gluetun
  #- /opt/gluetun/wg0.conf:/gluetun/wireguard/wg0.conf
environment:
- FIREWALL_OUTBOUND_SUBNETS=192.168.1.0/24
- VPN_SERVICE_PROVIDER=mullvad
- VPN_TYPE=wireguard
- WIREGUARD_PRIVATE_KEY=xxxxxxxxPrivateKeyxxxxxxxx
- WIREGUARD_ADDRESSES=10.70.149.4/32
- SERVER_COUNTRIES=Sweden,USA,Germany,Canada
  #- SERVER_CITIES=got,lax,atl,sto,fra,dus,ber,tor,mtr
# See https://github.com/qdm12/gluetun-wiki/tree/main/setup#setup
# Timezone for accurate log times
- TZ=Europe/London
# Server list updater
# See
https://github.com/qdm12/gluetun-wiki/blob/main/setup/servers.md#update-the-vpn-servers-list
- UPDATER_PERIOD=24h
- HTTPPROXY=on
- HTTPPROXY_LISTENING_ADDRESS=:8888
- HTTPPROXY_STEALTH=on
  #- HTTPPROXY_USER=proxy
  #- HTTPPROXY_PASSWORD=server123
  #- SHADOWSOCKS=on
  #- SHADOWSOCKS_LISTENING_ADDRESS=:8388
restart: unless-stopped
labels:
  autoheal-app: true
```

```
autoheal:
  container_name: autoheal
  deploy:
    replicas: 1
  environment:
    AUTOHEAL_CONTAINER_LABEL: autoheal-app
  image: willfarrell/autoheal:latest
  network_mode: none
  restart: always
  volumes:
    - /etc/localtime:/etc/localtime:ro
    - /var/run/docker.sock:/var/run/docker.sock

socks5:
  image: serjs/go-socks5-proxy
  container_name: socks5
  depends_on:
    gluetun:
      condition: service_healthy
  network_mode: "service:gluetun"
  restart: always
  labels:
    autoheal-app: true

qbittorrent:
  image: lscr.io/linuxserver/qbittorrent:latest
  container_name: qbittorrent
  environment:
    - PUID=1004
    - PGID=1001
    - TZ=Europe/London
    - WEBUI_PORT=9092
    - TORRENTING_PORT=6881
  volumes:
    - /opt/qbittorrent:/config
    - /mnt/download:/downloads
    - /media:/media
  #ports:
  #- 9092:9092
  #- 6881:6881
  #- 6881:6881/udp
  depends_on:
    gluetun:
      condition: service_healthy
  network_mode: service:gluetun
  restart: unless-stopped
  labels:
    autoheal-app: true
  healthcheck:
    interval: 5m
    retries: 5
```

```
start_period: 2m
test: curl -f http://localhost:9092 || exit 1
timeout: 20s

flaresolVERR:
image: ghcr.io/flaresolVERR/flaresolVERR:latest
# DockerHub mirror flaresolVERR/flaresolVERR:latest
#image: 21hsmw/flaresolVERR:fixlooping
#image: docker.io/library/oui:oui
#image: ghcr.io/flaresolVERR/flaresolVERR:latest
#image: ghcr.io/flaresolVERR/flaresolVERR:pr-1282
#jcolfej/flaresolVERR:pr-1272
#image: alexfozor/flaresolVERR:pr-1300
#image: alexfozor/flaresolVERR:pr-1300-experimental
container_name: flaresolVERR
environment:
- LOG_LEVEL=debug
- LOG_HTML=false
- CAPTCHA_SOLVER=none
- TZ=Europe/London
- LANG=en_GB
#ports:
# - "8191:8191"
network_mode: service:gluetun
depends_on:
  gluetun:
    condition: service_healthy
restart: unless-stopped
labels:
  autoheal-app: true
healthcheck:
  interval: 5m
  retries: 5
  start_period: 2m
  test: curl -f http://localhost:8191/health || exit 1
  timeout: 20s
transmission:
image: lscr.io/linuxserver/transmission:latest
container_name: transmission
environment:
- PUID=1004
- PGID=1001
- TZ=Europe/London
# - TRANSMISSION_WEB_HOME= #optional
# - USER= #optional
# - PASS= #optional
# - WHITELIST= #optional
# - PEERPORT= #optional
# - HOST_WHITELIST= #optional
volumes:
- /opt/transmission:/config
```

```
- /mnt/download:/downloads
- /mnt/download/watch:/watch
- /media:/media
#ports:
# - 9091:9091
# - 51413:51413
# - 51413:51413/udp
network_mode: service:gluetun
depends_on:
  gluetun:
    condition: service_healthy
restart: unless-stopped
labels:
  autoheal-app: true
healthcheck:
  interval: 5m
  retries: 5
  start_period: 2m
  test: curl -f http://localhost:9091 || exit 1
  timeout: 20s

prowlarr:
  image: lscr.io/linuxserver/prowlarr:develop #or latest
  container_name: prowlarr
  environment:
    - PUID=1001
    - PGID=1001
    - TZ=Europe/London
  volumes:
    - /opt/prowlarr:/config
  #ports:
  # - 9696:9696
  network_mode: service:gluetun
  depends_on:
    gluetun:
      condition: service_healthy
  restart: unless-stopped
  labels:
    autoheal-app: true
  healthcheck:
    interval: 5m
    retries: 5
    start_period: 2m
    test: wget --no-verbose --tries=1 --spider http://localhost:9696
  || exit 1
  timeout: 20s

stash3d:
  image: stashapp/stash:development #alt: latest
  #image: cr.hotio.dev/hotio/stash:nightly #alt release
  container_name: stash3d
```

```
## the container's port must be the same with the STASH_PORT in the
environment section
#ports:
# - "9998:9998"
## If you intend to use stash's DLNA functionality uncomment the
below network mode and comment out the above ports section
# network_mode: host
logging:
  driver: "json-file"
  options:
    max-file: "10"
    max-size: "2m"
environment:
  - STASH_STASH=/data/
  - STASH_GENERATED=/generated/
  - STASH_METADATA=/metadata/
  - STASH_CACHE=/cache/
## Adjust below to change default port (9999)
  - STASH_PORT=9998
  - PUID=1006
  - PGID=1001
  - UMASK=002
  - TZ=Europe/London
volumes:
  - /etc/localtime:/etc/localtime:ro
## Adjust below paths (the left part) to your liking.
## E.g. you can change ./config:/root/.stash to
./stash:/root/.stash

## Keep configs, scrapers, and plugins here.
  - /opt/stash3d-storage/config:/root/.stash
## Point this at your collection.
  - /media/stash3d:/data
## This is where your stash's metadata lives
  - /opt/stash3d-storage/metadata:/metadata
## Any other cache content.
  - /opt/stash3d-storage/cache:/cache
## Where to store binary blob data (scene covers, images)
  - /opt/stash3d-storage/blobs:/blobs
## Where to store generated content
(screenshots, previews, transcodes, sprites)
  - /opt/stash3d-storage/generated:/generated
devices:
  - /dev/dri:/dev/dri
# use if not using the update bash script intel-media-driver for
newer, libva-intel-driver for older
#command: /bin/sh -c 'apk --no-cache add libva-intel-driver &&
stash'
post_start:
  - command: /usr/bin/pip install pystashlib stashapp-tools --
break-system-packages
```

```
    user: root
    # for 13th gen Iris Xe GPUs use intel-media-driver for older gens
libva-intel-driver
  - command: apk add libva-intel-driver intel-media-driver
  user: root
restart: unless-stopped
network_mode: service:gluetun
depends_on:
  gluetun:
    condition: service_healthy
labels:
  autoheal-app: true
healthcheck:
  interval: 5m
  retries: 5
  start_period: 2m
  test: wget --no-verbose --tries=1 --spider http://localhost:9998
|| exit 1
  timeout: 20s

radarr:
  image: lscr.io/linuxserver/radarr:develop
  container_name: radarr
  environment:
    - UMASK_SET=022
    - PUID=1003
    - PGID=1001
    - TZ=Europe/London
  volumes:
    - /media/movies:/movies
    - /mnt/download/complete:/downloads
    - /opt/radarr:/config
  #ports:
  #- 8310:8310
  network_mode: service:gluetun
  depends_on:
    gluetun:
      condition: service_healthy
  prowlarr:
    condition: service_started
  restart: unless-stopped
  labels:
    autoheal-app: true
  healthcheck:
    interval: 5m
    retries: 5
    start_period: 2m
    test: curl -f http://localhost:8310/ping || exit 1
    timeout: 20s

sonarr:
```

```
image: lscr.io/linuxserver/sonarr:latest
container_name: sonarr
environment:
  - PUID=1002
  - PGID=1001
  - TZ=Europe/London
volumes:
  - /media/tv:/data
  - /opt/sonarr:/config
  - /mnt/download/complete:/downloads
#ports:
#- 8989:8989
network_mode: service:gluetun
depends_on:
  gluetun:
    condition: service_healthy
  prowlarr:
    condition: service_started
restart: unless-stopped
labels:
  autoheal-app: true
healthcheck:
  interval: 5m
  retries: 5
  start_period: 2m
  test: curl -f http://localhost:8989/ping || exit 1
  timeout: 20s
```

jdownloader2:

```
image: jlesage/jdownloader-2
container_name: jdownloader2
#ports:
#- "5800:5800" #web interface port
#- "5900:5900" #vnc port
#- "3129:3129" #myjdownloader port see
https://github.com/jlesage/docker-jdownloader-2#direct-connection
volumes:
  - "/opt/jdownloader2:/config:rw"
  - "/media/downloads:/output:rw"
network_mode: service:gluetun
depends_on:
  gluetun:
    condition: service_healthy
  socks5:
    condition: service_started
restart: unless-stopped
labels:
  autoheal-app: true
healthcheck:
  interval: 5m
  retries: 5
```

```
start_period: 2m
test: curl -f http://localhost:5800 || exit 1
timeout: 20s
```

Starting the whole stack:

```
docker compose -f docker-gluetun.yml up -d
```

Updating individual services (e.g. autohealth):

```
docker compose -f docker-gluetun.yml pull autohealth
docker compose -f docker-gluetun.yml up autohealth -d
```

As a restart/rebuild of the gluetun container prevents the other services from connecting, forcing gluetun to only connect to a different VPN endpoint can be done using the following 'hack':

```
docker exec -ti gluetun 'wget' '-q0-' '--method=PUT' '--body-data={"status":"stopped"}' 'http://127.0.0.1:8000/v1/openvpn/status'
```

From:

<http://wuff.dyndns.org/> - **Wulf's Various Things**

Permanent link:

<http://wuff.dyndns.org/doku.php?id=linux:gluetun-docker-stack>

Last update: **2025/08/27 12:28**

