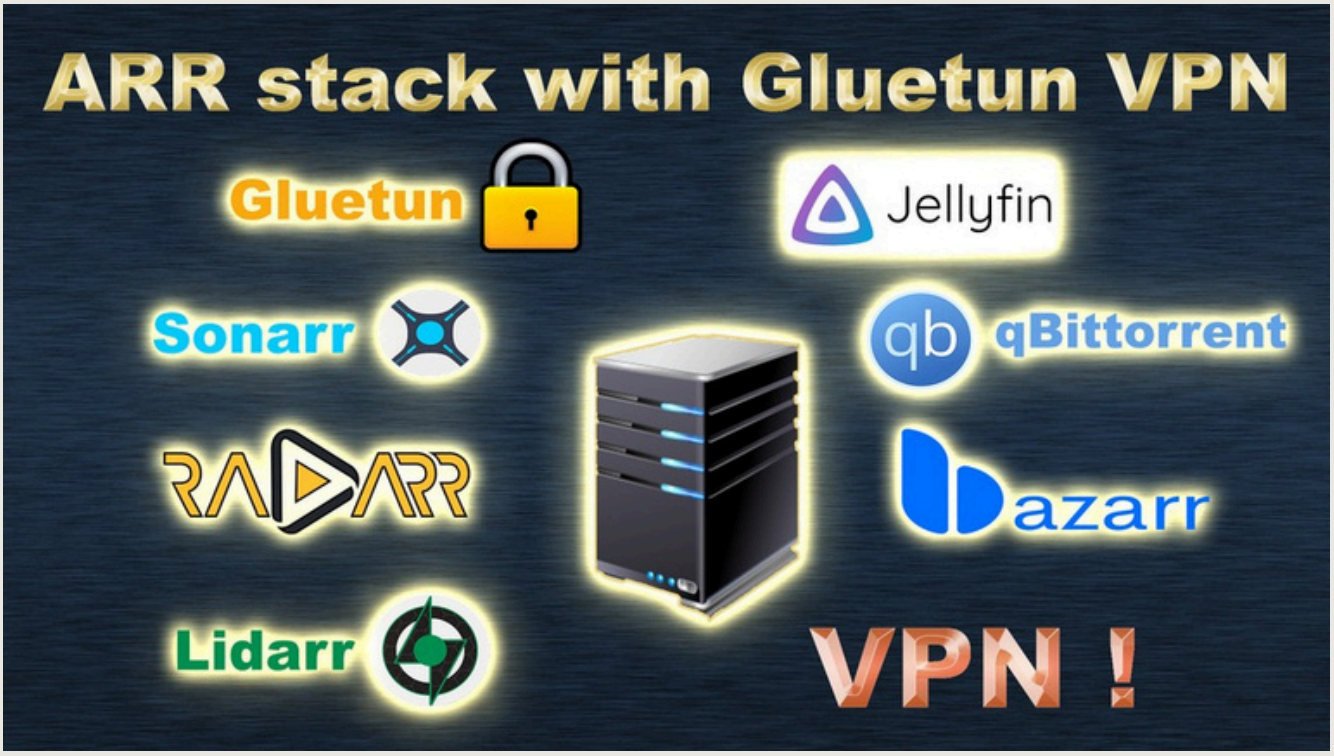
ARR stack with Gluetun VPN



Link to GitHub repo

Link to Automation Avenue platform

This PDF was created for **THIS YOUTUBE VIDEO**.
Click **THIS LINK** to see source docker-compose file.
You can either just copy entire file or start building your own version of that file by copying a single service, like Sonarr or Radarr.
Remeber to add ‘services’ line at the very top as below:

```
---
services:

##############################
# RADARR
##############################

  radarr:
    image: lscr.io/linuxserver/radarr:latest
    container_name: radarr
    network_mode: "service:gluetun"
    environment:
      - PUID=1000
      - PGID=1000
      - TZ=Etc/UTC
    volumes:
      - /media/arr/radarr/config:/config
      - /media/arr/radarr/movies:/movies #optional
      - /media/arr/qbittorrent/downloads:/downloads #optional
    restart: unless-stopped
```

Now keep adding new services, this time omitting that top ‘services’ line. To add Sonarr - copy from the source file that portion :

```
############################
# SONARR
############################

  sonarr:
    image: lscr.io/linuxserver/sonarr:latest
    container_name: sonarr
    network_mode: "service:gluetun"
    environment:
      - PUID=1000
      - PGID=1000
      - TZ=Etc/UTC
    volumes:
      - /media/arr/sonarr/config:/config
      - /media/arr/sonarr/tvseries:/tv #optional
      - /media/arr/qbittorrent/downloads:/downloads #optional
    restart: unless-stopped
```

For Prowlarr, copy that portion:

```
############################
# PROWLARR
############################

  prowlarr:
    image: lscr.io/linuxserver/prowlarr:latest
    container_name: prowlarr
    network_mode: "service:gluetun"
    environment:
      - PUID=1000
      - PGID=1000
      - TZ=Etc/UTC
    volumes:
      - /media/arr/prowlarr/config:/config
    restart: unless-stopped
```

Lidarr

```
############################
# LIDARR
############################

  lidarr:
    image: lscr.io/linuxserver/lidarr:latest
    container_name: lidarr
    network_mode: "service:gluetun"
    environment:
      - PUID=1000
      - PGID=1000
      - TZ=Etc/UTC
    volumes:
      - /media/arr/lidarr/config:/config
      - /media/arr/lidarr/music:/music #optional
      - /media/arr/qbittorrent/downloads:/downloads #optional
    restart: unless-stopped
```

Readarr

```
############################
# READARR
############################

  readarr:
    image: lscr.io/linuxserver/readarr:develop
    container_name: readarr
    network_mode: "service:gluetun"
    environment:
      - PUID=1000
      - PGID=1000
      - TZ=Etc/UTC
    volumes:
      - /media/arr/readarr/config:/config
      - /media/arr/readarr/books:/books #optional
      - /media/arr/qbittorrent/downloads:/downloads #optional
    restart: unless-stopped
```

qBittorrent

```
############################
# QBITTORRENT
############################

  qbittorrent:
    image: lscr.io/linuxserver/qbittorrent:latest
    container_name: qbittorrent
    network_mode: "service:gluetun"
    environment:
      - PUID=1000
      - PGID=1000
      - TZ=Etc/UTC
      - WEBUI_PORT=8080
      - TORRENTING_PORT=6881
    volumes:
      - /media/arr/qbittorrent/config:/config
      - /media/arr/qbittorrent/downloads:/downloads #optional
    restart: unless-stopped
```

Jellyfin

```
############################
# JELLYFIN
############################

  jellyfin:
    image: lscr.io/linuxserver/jellyfin:latest
    container_name: jellyfin
    environment:
      - PUID=1000
      - PGID=1000
      - TZ=Etc/UTC
    volumes:
      - /media/arr/jellyfin/config:/config
      - /media/arr/sonarr/tvseries:/data/tvshows
      - /media/arr/radarr/movies:/data/movies
    ports:
      - 8096:8096
      - 8920:8920 #optional
      - 7359:7359/udp #optional
      - 1900:1900/udp #optional
    restart: unless-stopped
```

Last but not least - Gluetun VPN:

```
############################
# GLUETUN
############################

  gluetun:
    image: qmcgaw/gluetun
    container_name: gluetun
    ports:
      - 9696:9696 #prowlarr
      - 7878:7878 #radarr
      - 8989:8989 #sonarr
      - 6767:6767 #bazarr
      - 8686:8686 #lidarr
      - 8787:8787 #readarr
      - 8080:8080 #qbittorrent
      - 6881:6881 #qbittorrent
      - 6881:6881/udp #qbittorrent
    cap_add:
      - NET_ADMIN
    devices:
      - /dev/net/tun:/dev/net/tun
    environment:
      - VPN_SERVICE_PROVIDER=nordvpn
      - VPN_TYPE=openvpn # or wireguard
      - OPENVPN_USER=<your vpn user>
      - OPENVPN_PASSWORD=<your vpn password>
      - SERVER_COUNTRIES=Netherlands

############################
```

Now that you have all services you need, you can start configuring them in user interface.
You can see the configuration process in **THIS VIDEO**, the only difference will be in HOST portion because you have to leave it as ‘localhost’ (which is default setting).
Please see the process below:

Installation process:

Make sure you are in the same folder as docker-compose.yml file. Run docker-compose - 'up -d' to deploy, 'stop' and 'rm' to stop and remove the stack:

sudo docker-compose up -d
sudo docker-compose stop
sudo docker-compose rm
sudo docker-compose down (will do both - stop and remove)

Go to the folder specified in .yml file (if its /media/arr then go to /media as root) and run ‘chown’ command with the user id and group id configured in that .yml file, for example:

sudo chown -R 1000:1000 /media/arr

Now you can log on and work with all services:

qBittorrent:

Check what qbittorrent temporary password is to be able to log o by running:

docker logs qbittorrent
You will see in the logs something like:

The WebUI administrator username is: admin
The WebUI administrator password was not set. A temporary password is provided for this session: <temp-password>

Now you can go to URL:
http://localhost:8080

and log on using details provided in container logs.
Go to Tools - Options - WebUI - change the user and password and tick 'bypass authentication for clients on localhost' .

Then configure Prowlarr service (each of these services will first ask you to set up user and password):

Prowlarr:
http://localhost:9696
Go to Settings - Download Clients - `+` symbol - Add download client - choose qBittorrent (unless you decided touse different download client)
Put the port id matching the WebUI in docker-compose for qBittorrent (default is 8080) and username and password that you configured for qBittorrent in previous step.
Host - leave 'localhost' setting as it is.

Sonarr:
http://localhost:8989

Go to Settings - Media Management - Add Root Folder - set your root folder.
What is your root folder depends on what you have in ‘volume’ configuration for your Sonarr.
If your config for Sonarr looks like that:

volumes:
- /media/arr/sonarr/config:/config
*- /media/arr/sonarr/tvseries:**/tv***

then you set root to ‘**/tv**’.
You simply set the root to whatever is on the right side of the colon for the line with tv series.
Depending on the image though, you might have something like that:

volumes:
- /media/arr/sonarr/config:/config
*- /media/arr/sonarr/tvseries:**/data/TVSeries***

then you set your root to **/data/TVSeries** - you simply match whatever is on the right side of the colon in that line.

Go to Settings - Download Clients - click `+` symbol - choose qBittorrent and repeat the steps from Prowlarr.

Go to Settings - General - scroll down to API key - copy - go to Prowlarr - Settings - Apps -click '+' - Sonarr - paste API key.

Radarr:
http://localhost:7878
Go to Settings - Media Management - Add Root Folder - set /data/movies as your root folder

Then Settings- Download clients - click 'plus' symbol, choose qBittorrent etc - basically same steps as for Sonarr

Settings - General - scroll down to API key - copy - go to Prowlarr - add same way as in Sonarr

Settings - General - switch to 'show advanced'- Backups - choose /data/Backup folder

Lidarr:
http://localhost:8686
Follow the same steps for Lidarr and Readarr as for above applications.

Readarr:
http://localhost:8787

Now go back to Prowlarr and click 'Indexers at the top right, click 'Add indexer' - search for sth like 'rarbg' or 'yts' etc then test - save

Then click 'Sync App Indexers icon (next to 'Add indexer').
If you go to Settings - Apps - you should see green 'Full sync' next to each application.

Arr stack completed - you can now 'add movie' in Radarr or 'add series' in Sonarr etc and click 'search all' or 'search monitored' - that will trigger the download process.

Jellyfin:
http://localhost:8096
Jellyfin uses port 1900 as well. If you run:
docker-compose up -d
and have something running on port 1900 - its most possibly rygel service, run:
sudo apt-get remove rygel
and run the
sudo docker-compose up -d again

Then add media library in Jellyfin matching folders configured in docker-compose.yml file, so in Jellyfin you should see them as:
/data/movies
/data/tvshows

Note that this also might depend on the image, you basically match the right side of the config in Jellyfin's 'volume' configuration:

volumes:
*- /media/arr/sonarr/tvseries:**/data/tvshows***
*- /media/arr/radarr/movies:**/data/movies***

Thank you !

EMAIL

info@automation-avenue.com

WEB

https://www.automation-avenue.com