ARR stack - single command deployment !

Link to GitHub repo

Link to Automation Avenue platform

This PDF was created for **THIS YOUTUBE VIDEO**.
(you might also want to see new arr stack video with Gluetun VPN container added - click **HERE** to watch)

To configure services - first download files from **THIS LOCATION**.

Then run commands :
cd /home/<your user>/Downloads
unzip youtube-39-arr-apps-1-click

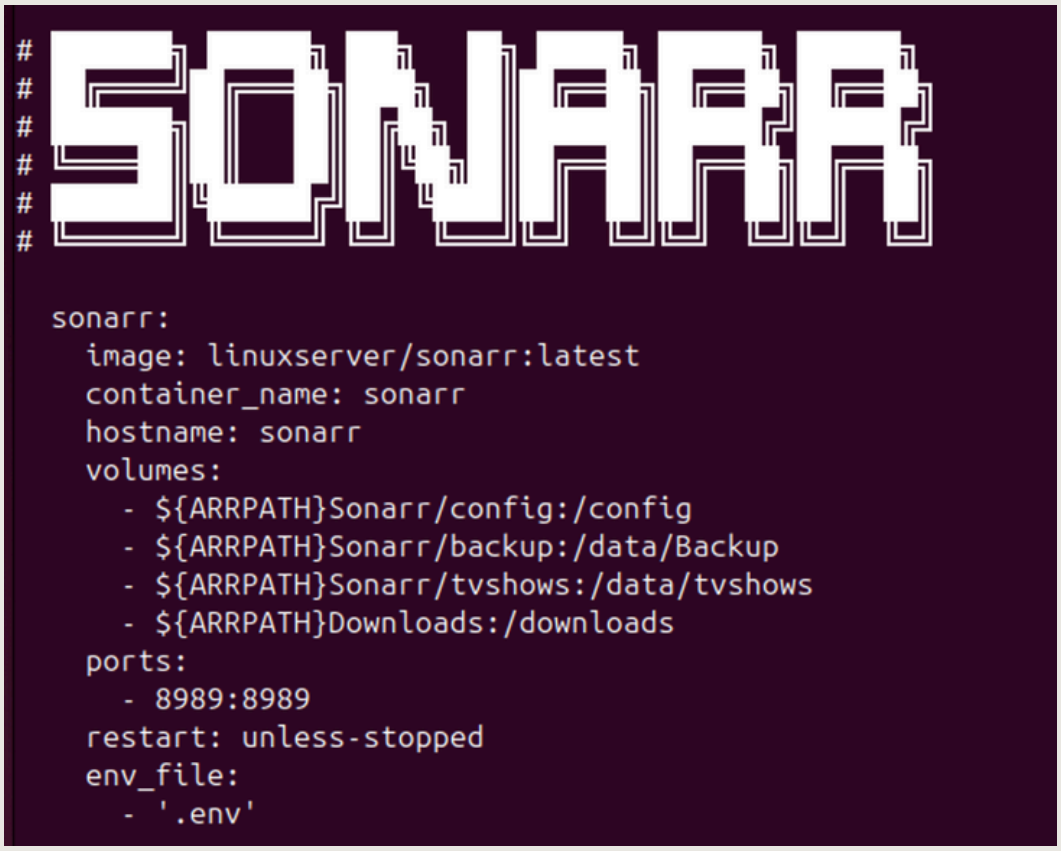
You should see following services configured:
Prowlarr:

```
services:

#  ____                 _               _
# |  _ \ _ __ _____   _| | __ _ _ __ _ __
# | |_) | '__/ _ \ \ / / |/ _` | '__| '__|
# |  __/| | | (_) \ V /| | (_| | |  | |
# |_|   |_|  \___/ \_/ |_|\__,_|_|  |_|

  prowlarr:
    image: linuxserver/prowlarr:latest
    container_name: prowlarr
    hostname: prowlarr
    volumes:
      - ${ARRPATH}Prowlarr/config:/config
      - ${ARRPATH}Prowlarr/backup:/data/Backup
      - ${ARRPATH}Downloads:/downloads
    ports:
      - 9696:9696
    restart: unless-stopped
    env_file:
      - '.env'
```

Sonarr:



Radarr:

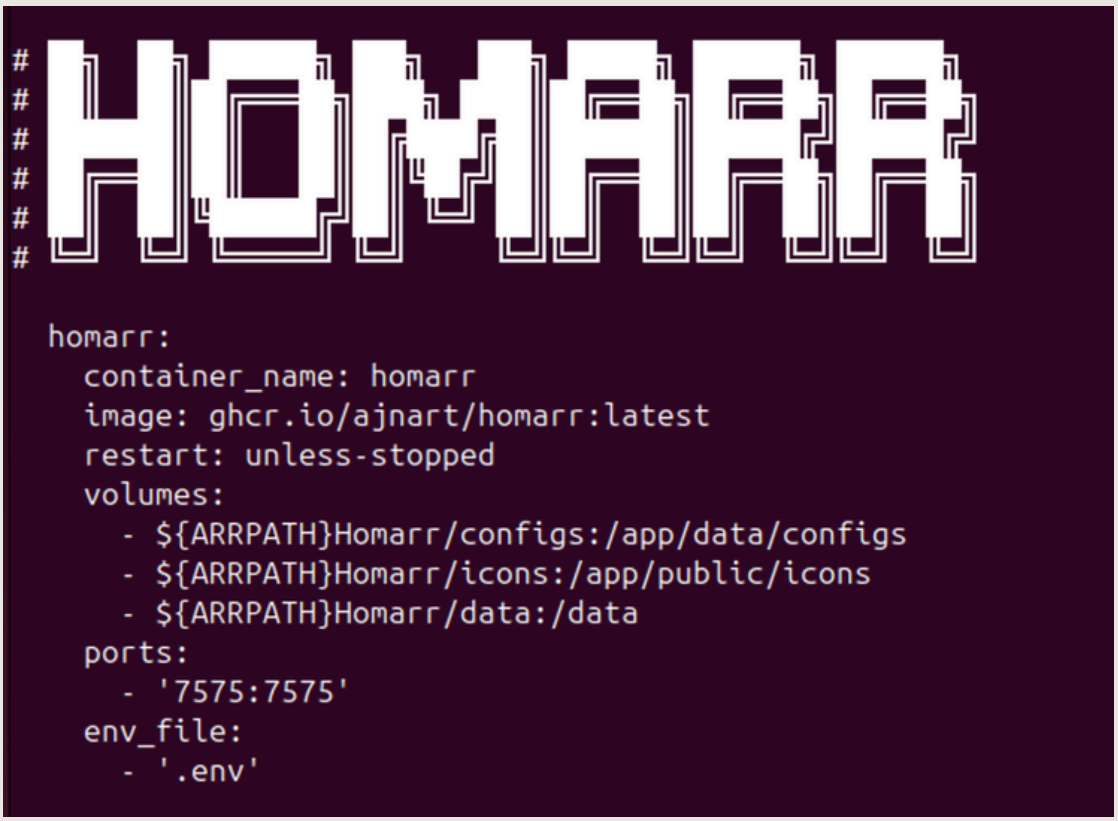
Lidarr

Readarr

qBittorrent

Jellyfin

Homarr:



When you have all the services you need in your docker-compose file, you can start installing and configuring them as per instructions 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)

Chage ownership of the folder specified in .env file (by default its /media/Arr) and run 'chown' command with the user id and group id also configured in that .env file:

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:

sudo 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 - you might want to change from 'localhost'to ip address of the host machine (run 'ip address' command on your host system) or to 'qbittorrent' - click little 'Test' button at the bottom before saving to make sure you get a green 'tick'.

Sonarr:

http://localhost:8989

Go to Settings - Media Management - Add Root Folder - set /data/tvshows as your root folder

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. You might also have to change 'localhost' to ip address of the Ubuntu/Host - use 'Test' button below to see if you get green 'tick'.

Then Settings - General - switch to 'show advanced' in top left corner - scroll down to 'Backups' and choose /data/Backup (or whatever location you have in your docker compose file for Sonarr backups)

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

Homarr:
http://localhost:7575

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

That might depend on the image though, you basically match the right side of the config in Jellyfin's 'volume' configuration.
If the volume configuration looks like that:

```
    volumes:
      - ${ARRPATH}Radarr/movies:/data/Movies
      - ${ARRPATH}Sonarr/tvshows:/data/TVShows
```

then in Jellyfin you match that /data/Movies or /data/TVShows etc.
Thank you !

EMAIL

info@automation-avenue.com

WEB

https://www.automation-avenue.com