HOW TO INSTALL A FULL ARR-STACK ON SYNOLOGY NAS RUNNING DSM 7.2

Mathias Furenes

Table of Contents

Before we begin

If you find any mistakes, incorrect facts, or would like to give your feedback, you can go so either by contacting me on Reddit, Discord or Github.

Reddit: https://www.reddit.com/user/MattiTheGamer/

Discord: @matti1003

Github: https://github.com/MathiasFurenes/synology-arr-guide/tree/main

If you want to you can also support me with a PayPal donation here:

https://www.paypal.com/donate/?hosted_button_id=DK7VP9RD2LEQ2

Info about TL;DR

The TL;DR comes before the main guide. For the best experience, skip this section and go to the full guide. The TL;DR only covers the installation, and NOT the configs as they are too complex to put in a TL;DR. It also requires some knowledge on docker beforehand so you can edit the docker-compose to match your system. I have also not added anything UseNet related to the TL;DR.

Legal Disclaimer

For legal reasons I must state that I do not condone illegal pirating of copyrighted material. This is made for educational purposes only. I expect that everyone who follows this guide will only use this for legal purposes, like downloading free to use Linux ISOs. Please never ever use this to illegally download copyrighted material such as, but not limited to, movies and TV Shows.

Introduction

Before we start let's figure out your needs. Do you want to download movies, tv shows, music, e-books, comics or adult videos? Most likely you want a combination of a lot of

them. We also need to figure out whether you want it to be connected to a VPN. Here is a break-down of all the apps and their use-cases:

What are the *arrs?

From the wiki itself:

“Lidarr, Prowlarr, Radarr, Readarr, Sonarr, and Whisparr are collectively referred to as "*arr" or "*arrs". They are designed to automatically grab, sort, organize, and monitor your Music, Movie, E-Book, or TV Show collections for Lidarr, Radarr, Readarr, Sonarr, and Whisparr; and to manage your indexers and keep them in sync with the aforementioned apps for Prowlarr.”

In Simple terms:

Prowlarr:

An index manager. This just means it searches for files to download on websites you assign it.

Radarr:

When Prowlarr find a movie file, it gives it to Radarr. Radarr then send it over to a download client, like QBitTorrent. After it’s done downloading, Radarr takes it away from QBItTorrent again and rename the file appropriately before it puts it inside your Media library.

Sonarr:

Same as Radarr, but for TV shows.

Lidarr:

Same as Radarr, but for music

Readarr:

Same as Radarr, but for e-books

Whisparr:

Same as Radarr, but for adult videos

Bazarr:

Connects with Radarr and Sonarr to download subtitles for your movies and TV shows.

Flaresolverr:

Bypasses cloudfare solver

Overseerr:

A requesting application where you browse or search for movies and TV shows, kinda like Netflix, and with a click of a button they start downloading to your own media collection!

Requestrr:

Goes together with Overseerr or Radarr and Sonarr to allow for requesting through discord chat.

Arr-stack:

The arr-stack just refers to a collection of these applications bundled and working together.

All the different *arr apps (and related ones that I know of)

Prowlarr – Index manager. It searches the torrent sites for downloads (Recommended for everyone)

Flaresolverr – Some indexers require you to solve a Cloudflare captcha. Flaresolverr can do this.

Jackett – Alternative to Prowlarr. Most find Prowlarr to be both easier to set up and better to use.

Sonarr – TV Shows/Anime Shows downloader.

Radarr – Movie downloader.

Lidarr – Music downloader.

Readarr – E-book downloader

Mylar – Comics downloader.

Bazarr – Subtitle downloader for your Movies and TV Shows.

Lazy Librarian – A program to follow authors and grab metadata for digital reading.

Whisparr – Adult videos downloader.

GlueTUN – Required for use of VPN

Plex – A frontend to your media server. It's where you access all your media, in the style of something like Netflix. It also has a very good Spotify-like app called plexamp.

Jellyfin – Also a frontend to your media server. Jellyfin is, in contrast to Plex, open-source. This means that all of it's features is and always will be free, but it also means that it doesn't have the same funding and therefore might not have all of the features Plex has.

Overseerr – Allows for easy requesting of movies and TV shows to add to Plex.

Jellyseerr – Same as Overseer but for Jellyfin.

Ombi – App to request Movies and TV Shows for plex or Emby.

Requestrr – Allows for requests for Sonarr and Radarr via chat, like Discord. It can also be integrated with Ombi and Overseer.

qBitTorrent – Torrent download client.

NZBGet – NZB download client.

Tautulli – Plex media server statistics

There are even more, but I have not gotten into these myself. These are the ones I have atleast some knowledge about.

The apps I will cover in this guide

I would like to cover as many as possible, but I have not used or tried some of them myself. I host on Plex, so I use Overseer, and have not tried neither Jellyseer or Ombi. Even though I'm pretty sure Jellyseer is the exact same just for Jellyfin. But you should always read the official docs yourself. Anyway, the apps I will go over is:

- GlueTUN
- Prowlarr
- Flaresolverr
- Sonarr
- Radarr
- Lidarr
- Overseer
- Requestrr
- qBitTorrent
- Tautulli
- Sabznbd (UseNet)
- NZBHydra2 (UseNet)

Hardware specs

I am running all this on a DS423+, with an extra 16GB memory stick and 512GB SSD cache. I have not tested it myself on any other devices, but I have made a few google searches and asked ChatGPT for some help to determine the systems requirements. Therefore, I ask you to take these number with a grain of salt.

Minimum requirements

CPU

A quad-core 64bit CPU with x86 architecture. (Docker can only run on x86, and not any ARM CPUs natively. You might be able to still try this out, but you will have to do some workarounds.) An Intel Celeron J4105 or Intel Celeron J4125 should be sufficient for basic use.

RAM

4GB RAM (reported by ChatGPT). I think it might be able to run on 2GB for low, basic use. But don't expect the best performance

Storage

These apps don't take up more than 2GB-5GB for database and configs.

A 2-hour movie in 480p will take 700MB-2GB space, and 3GB-6GB in 720p. A 12-episode TV show with 45 minutes long episodes will take 4GB-10GB in 480p and 12GB-27GB in 720p

Recommended requirements

CPU

Intel Celeron J4125 or higher. For best performance, Intel Core i3/i5 or AMD equivalent. But since most will probably run this on a Synology, the J4125 or better is sufficient.

RAM

8GB RAM. For best performance, an upgrade to 16GB will make it a lot smoother.

Storage

A 2-hour long movie in 1080p will take 8GB-15GB storage space, and in 4k it will be about 20GB-50GB. A 12-episode show with 45 minutes long episodes will take up 35GB-60GB in 1080p and 90GB-225GB in 4k.

Recommended NAS models for this setup

Since I assume most people will use a Synology NAS, as this is what this guide was meant for, I will list some recommendations.

Minimum

- **Synology DS220+**
  - CPU: Intel Celeron J4025 (dual-core)
  - RAM: 2GB (expandable to 6GB)
  - Suitable for running a few applications simultaneously with small to medium libraries.

- **Synology DS720+**
  - CPU: Intel Celeron J4125 (quad-core)
  - RAM: 2GB (expandable to 6GB)
  - Great for small to mid-size media libraries, running Docker containers, and handling multiple tasks at once.

Recommended

- **Synology DS920+**
  - CPU: Intel Celeron J4125 (quad-core)
  - RAM: 4GB (expandable to 8GB)
  - Supports SSD caching, making it a solid choice for heavier workloads like streaming, transcoding, and multiple apps running concurrently.

- **Synology DS423+**
  - CPU: Intel Celeron J4125 (quad-core)
  - 2GB RAM (Recommended to upgrade to at least 4GB or 6GB)
  - 4-bay NAS with support for 2 NVME drives

- **Synology DS1821+**
  - CPU: AMD Ryzen V1500B (quad-core)
  - RAM: 4GB (expandable to 32GB)
  - 8-bay NAS with strong processing power for large libraries and heavy multitasking.

Models to avoid

Low-End ARM-Based NAS Models

- **Synology DS216j** (Marvell Armada 385, dual-core 1.0GHz, 512MB)
- **Synology DS218j** (Marvell Armada 385, Dual-Core 1.3 GHz, 512 MB RAM)
- **Synology DS219j** (Marvell Armada 3720, Dual-core 800 MHz, 256 MB RAM)
- **Synology DS220j** (Realtek RTD1296, Quad-Core 1.4 GHz, 512 MB RAM)

Why not:

1. **CPU**: These models come with weak, low-power ARM processors that are not suited for running multiple Docker containers or handling tasks like torrenting and media management.
1. **RAM**: Most of these devices have **512 MB RAM or less**, which is far too little for running multiple services in Docker.
1. **Docker Support**: ARM-based models may not fully support Docker, especially for complex workloads, and will struggle with performance under even light to moderate use.

Models with Less Than 2 GB RAM

1. **Synology DS220j** (512 MB RAM)
1. **Synology DS218play** (1 GB RAM)
1. **Synology DS218** (2 GB RAM)
1. **Synology DS118** (1 GB RAM)

Why not:

- **Insufficient RAM**: Apps like **Overseerr** and **qBittorrent** require more memory, and these models would quickly run out of resources. With less than 2 GB, you'll experience poor performance, constant swapping to disk, or the inability to run all your containers simultaneously.
- **No Upgrade Path**: Many of these models do not allow you to upgrade the RAM, so you're stuck with what they offer.

Older/Legacy Models

- **Synology DS214+** (Dual-core 1.33 GHz, 1 GB RAM)
- **Synology DS415play** (Intel Atom CE5335, Dual-core 1.6 GHz, 1 GB RAM)
- **Synology DS216play** (ARM Cortex-A9, Dual-core 1.5 GHz, 1 GB RAM)

Why not:

- **Outdated CPU architecture**: These older CPUs lack the power and modern architecture needed for virtualization and handling Docker workloads.
- **RAM limitations**: Even if some of these have x86 architecture, 1 GB or even 2 GB of RAM is not sufficient for your use case.

Single-Bay NAS Models

- **Synology DS118** (Realtek RTD1296, Quad-core 1.4 GHz, 1 GB RAM)
- **Synology DS119j** (Marvell Armada 3700, Dual-core 800 MHz, 256 MB RAM)

Why not:

- **Low performance**: These single-bay models come with very basic hardware, meaning you'll struggle to run Docker and multiple applications.
- **No redundancy**: With only one drive, there's no data redundancy (no RAID), which is a concern when managing large amounts of media files.

Models without integrated graphics

- Synology DS923+ (AMD Ryzen R1600, Quad-core 2.6GHz, 4GB RAM)
- Synology DS 1522+ (AMD Ryzen R1600, Quad-core 2.6GHz, 8GB RAM)

Why not:

- **No integrated graphics**: Although these can work just fine, they are not ideal as you won't be able to do hardware transcoding. For native playback of h264 files these can be perfect.

What you need instead

To run this full arr-stack smoothly on Docker, aim for:

- **x86-64 architecture** (with embedded graphics)
- **Minimum 4 GB RAM**, ideally **8 GB or more**.
- **Expandable RAM** for future growth.
- **At least dual-bay** for RAID redundancy.

TL;DR

If you don't want to read all that, just do this:

1. Create these folder inside /volume1/docker/arr-stack:

- gluetun
- lidarr > config
- overseer > config
- prowlarr > config
- qbittorrent > config, downloads
- radarr >config
- requestrr > config
- sonar > config

2. SSH into your NAS and type these commands

   - `sudo chmod -R 777 /volume1/Media`
   - `id.` Take note of the GID and UID output
   - `sudo chown -R < UID>:< GID> \volume1\Media`

3. Go to Task Scheduler and create a trigger task on start-up to run this script:

```
#!/bin/sh -e

insmod /lib/modules/tun.ko
```

   I. Create a firewall rule to allow port 1194 and 1195
   II. Go to think link: https://www.blackvoid.club/wireguard-spk-for-your-synology-nas/. Find the correct version for your system and manually install the .spk file in package cener. Don't run when finished. Reboot afterwards
   III. SSH into your device and type this command:
   IV. `/var/packages/WireGuard/scripts/start`

4. Find out how to configure GlueTUN for your VPN provider: https://github.com/qdm12/gluetun-wiki/tree/main/setup/providers

5. Create a new network inside container manager. Configure the network like this:
   - Subnet: 172.20.0.0/16
   - IP range: 172.20.0.2/25
   - Gateway: 172.20.0.1
   - IPv6: Disabled
   - IP Masquerade: enabled (Leave the “disable” option unticked)

6. Create a new docker project. Path should be /volume1/docker/arr-stack. Paste this docker-compose.yml wand make any necessary changes:

```
version: "3"
services:
  gluetun:
    image: qmcgaw/gluetun
    container_name: gluetun
    hostname: gluetun
    cap_add:
      - NET_ADMIN
    devices:
      - /dev/net/tun:/dev/net/tun
    ports:
      - 6881:6881
      - 6881:6881/udp
      - 8085:8085 # qbittorrent
      - 8989:8989 # Sonarr
      - 9696:9696 # Prowlarr
      - 7878:7878 # Radarr
      - 8686:8686 #Lidarr
      - 8191:8191 #FlareSolverr
      - 5055:5055 #Overseerr
      - 4545:4545 #Requestrr

    volumes:
      - /volume1/docker/arr-stack/gluetun:/gluetun
    environment:
      - VPN_SERVICE_PROVIDER=<your-provider>
      - VPN_TYPE=wireguard
      - VPN_DISABLE_IPV6=true
      # OpenVPN:
      # - OPENVPN_USER=
      # - OPENVPN_PASSWORD=
      # Wireguard:
      - WIREGUARD_PRIVATE_KEY= <your-private-key>
      - WIREGUARD_ADDRESSES=<your-address>
```

```
      - DNS=<your-dns>
      - SERVER_HOSTNAMES=<your-hostnames>
      - SERVER_CITIES=<your-city>
      # Timezone for accurate log times
      - TZ=Europe/Oslo
      # Server list updater
      # See https://github.com/qdm12/gluetun-
wiki/blob/main/setup/servers.md#update-the-vpn-servers-list
      - UPDATER_PERIOD=24h
      - FIREWALL_OUTBOUND_SUBNETS=172.20.0.0/192.168.0.0/24 #change this in
line with your subnet see note on guide
    network_mode: synobridge
    labels:
      - com.centurylinklabs.watchtower.enable=false #Disables Watchtower
    security_opt:
      - no-new-privileges:true #For security
  qbittorrent:
    image: lscr.io/linuxserver/qbittorrent
    container_name: qbittorrent
    network_mode: "service:gluetun"
    environment:
      - PUID=<your-UID>
      - PGID=<your-GID>
      - TZ=Europe/Oslo
      - WEBUI_PORT=8085
      - UMASK=022
    volumes:
      - /volume1/docker/arr-stack/qbittorrent/config:/config
      - /volume1/Media/Torrents:/Media/Torrents
    depends_on:
      gluetun:
        condition: service_healthy
    security_opt:
      - no-new-privileges:true
    restart: unless-stopped

  sonarr:
    image: lscr.io/linuxserver/sonarr:latest
    container_name: sonarr
    network_mode: "service:gluetun"
    environment:
      - PUID=<your-UID>
      - PGID=<your-GID>
      - TZ=Europe/Oslo
    volumes:
      - /volume1/docker/arr-stack/sonarr/config:\config
```

```
    - /volume1/Media:/Media
  depends_on:
    gluetun:
      condition: service_healthy
  restart: unless-stopped

prowlarr:
  image: lscr.io/linuxserver/prowlarr:latest
  container_name: prowlarr
  network_mode: "service:gluetun"
  environment:
    - PUID=<your-UID>
    - PGID=<your-GID>
    - TZ=Europe/Oslo
  volumes:
    - /volume1/docker/arr-stack/prowlarr/config:/config
  depends_on:
    gluetun:
      condition: service_healthy
  restart: unless-stopped

radarr:
  image: lscr.io/linuxserver/radarr:latest
  container_name: radarr
  network_mode: "service:gluetun"
  environment:
    - PUID=<your-UID>
    - PGID=<your-GID>
    - TZ=Europe/Oslo
  volumes:
    - /volume1/docker/arr-stack/radarr/config:/config
    - /volume1/Media:/Media
  depends_on:
    gluetun:
      condition: service_healthy
  restart: unless-stopped

lidarr:
  image: lscr.io/linuxserver/lidarr:latest
  container_name: lidarr
  network_mode: "service:gluetun"
  environment:
    - PUID=<your-UID>
    - PGID=<your-GID>
    - TZ=Europe/Oslo
  volumes:
```

```
    - /volume1/docker/arr-stack/lidarr/config:/config
    - /volume1/Media:/Media
  depends_on:
    gluetun:
      condition: service_healthy
  restart: unless-stopped

flaresolverr:
  image: ghcr.io/flaresolverr/flaresolverr:latest
  container_name: flaresolverr
  network_mode: "service:gluetun"
  environment:
    - TZ=Europe/Oslo
  depends_on:
    gluetun:
      condition: service_healthy
  security_opt:
    - no-new-privileges:true
  restart: unless-stopped

overseerr:
  image: sctx/overseerr:latest
  container_name: overseerr
  network_mode: "service:gluetun"
  environment:
    - LOG_LEVEL=debug
    - TZ=Europe/Oslo
  volumes:
    - /volume1/docker/arr-stack/overseer/config:/app/config
  depends_on:
    gluetun:
      condition: service_healthy
  restart: unless-stopped

requestrr:
  image: darkalfx/requestrr
  container_name: requestrr
  network_mode: "service:gluetun"
  volumes:
  - /volume1/docker/arr-stack/requestrr/config:/root/config
  depends_on:
    gluetun:
      condition: service_healthy
  restart: unless-stopped
```

11. For config of apps, I can't really put it in a TL;DR, as it's way too long and complicated. Chek out the full guide, or find out yourself on the official docs. The TL;DR is also not updated for UseNet support. Check out UseNet here.

Preparations

Folder Structure

I assume you have Installed container manager (docker) on your Synology system already, but if not do that now. Then we need to go into our “docker” shared folder and make some new folder.

The arr stack

Inside the “docker” shared folder, create a folder and call it something like “arr-stack”

Inside “arr-stack”, create these folders:

- gluetun
- lidarr
- overseer
- prowlarr
- qbittorrent
- radarr
- requestrr
- sonar
- tautulli
- sabnzbd (Optional, for UseNet only. For more info on UseNet, go here)
- nzbhydra2 (Optional, mainly for UseNet)

If you have any other apps beside these, create a folder for them too as described in the official docs.

Create a folder called “config” inside lidarr, overseer, prowlarr, qbittorrent, radarr, requestrr, sonar, sabnzbd and tautulli.

If you don’t have a Media folder

Create a new shared folder for housing your media. You can call this Data or Media. Within this shared library, you need another folder where you put the actual media. You can call this Media. In “/volume1/Media/Media” you create a folder for movies, tv shows, anime, music and whatever else you would like. Go back to “/volume1/Media” and create a folder called “torrents.” This is where we download our torrent files with qBitTorrent. Inside it create “Incomplete” and “Complete”. If you want to use Usenet, you can create folder called “UseNet” with “Complete” and “Incomplete” inside it.

The full folder structure should look like this:

Media

```
├── Torrents
│   ├── Incomplete
│   └── Complete
├── Usenet
│   ├── Incomplete
│   └── Complete
│       ├── Movies
│       ├── Music
|       ├── Anime TV Shows
│       └── TV Shows
└── Media
    ├── Movies
    ├── Music
    ├── Anime TV Shows
    └── TV Shows
```

docker

```
└──arr-stack
   ├──bazarr
   │  └── config
   ├── gluetun
   ├── lidarr
   │  └── config
   ├── sabnzbd (Optional)
   │  └── config
   ├── overseer
   │  └── config
   ├── prowlarr
   │  └── config
   ├── qbittorrent
   │  └── config
   ├── radarr
   │  └── config
   ├── requestrr
   │  └── config
   ├── sonarr
   │  └── config
   ├──nzbhydra2
   │
   └── tautulli
      └── config
```

Setting permissions

To make sure that all the apps have the right permissions to read and write the relevant folder, we need to ssh into the NAS. I prefer PuTTY, but you can use powershell or anything else that you like.

1. SSH into your NAS
2. Type the following 2 commands:
   - `sudo chmod -R 777 /volume1/Media/Media`
   - `sudo chmod -R 777 /volume1/Media/Torrents`
3. Find your UID and GID
   - Type: `id`
4. You should get an output like this:

As you can see, we get an output with uid=XXXX and gid=YYY.

5. Type the following 2 commands:
   - `sudo chown -R <UID>:<GID> /volume1/Media/Media`
   - `sudo chown -R <UID>:<GID> /volume1/Media/Torrents`

   NOTE: If you have your Media library at another location than "/volume1/Media", then put your path in the 2 relevant commands.

These commands ensure that we have the correct permissions to read and write the files of the directories and their subdirectories.

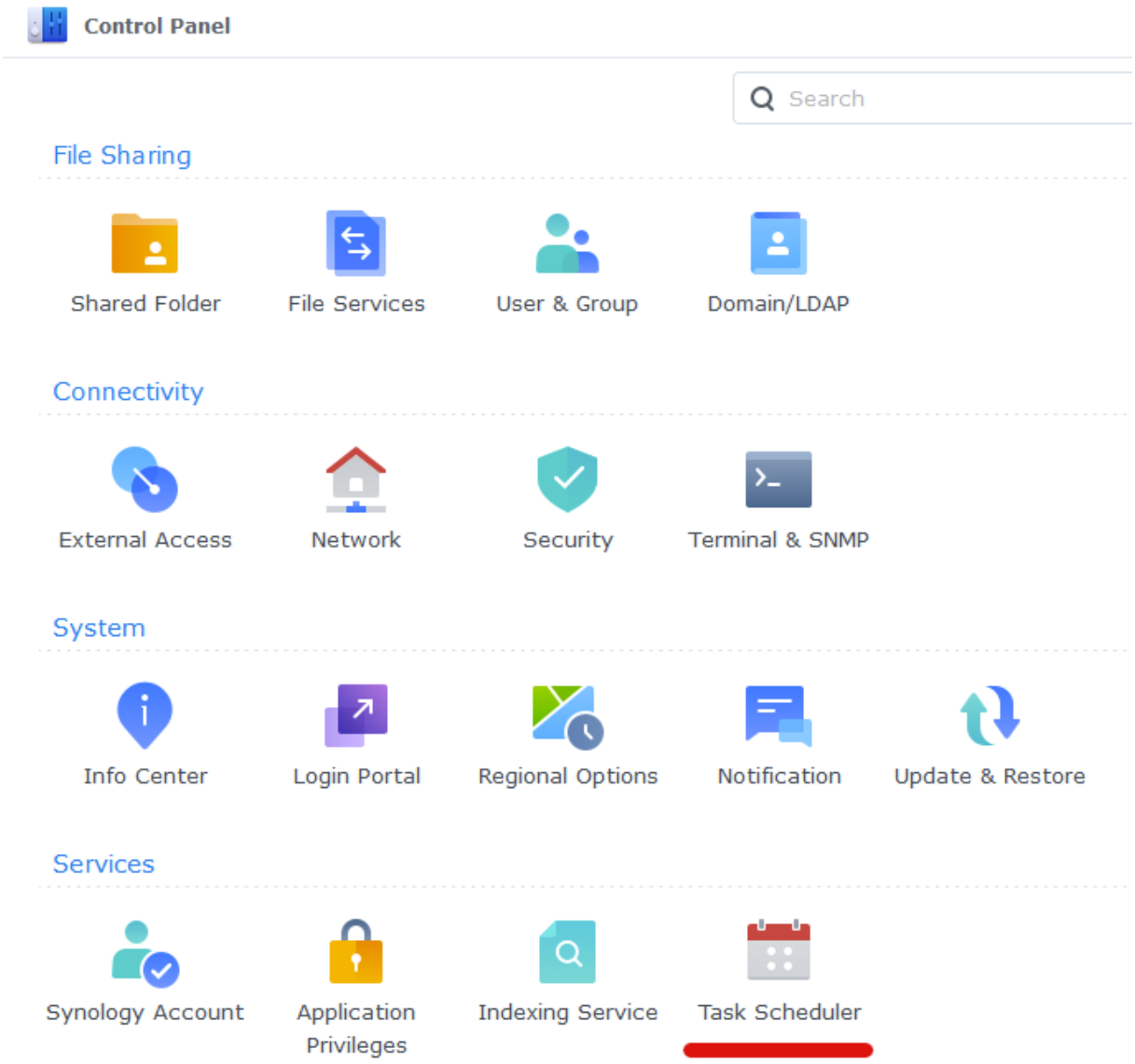
The installation

Now that we have our folder structure ready, let's begin with the actual installation.

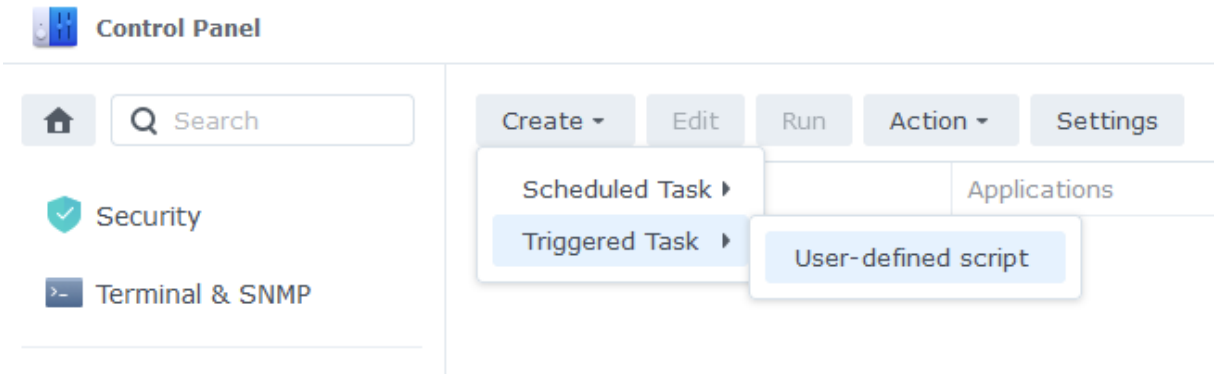
Setting up the start-up script for GlueTUN

For GlueTUN to start up automatically, we need to create a task on a schedule.

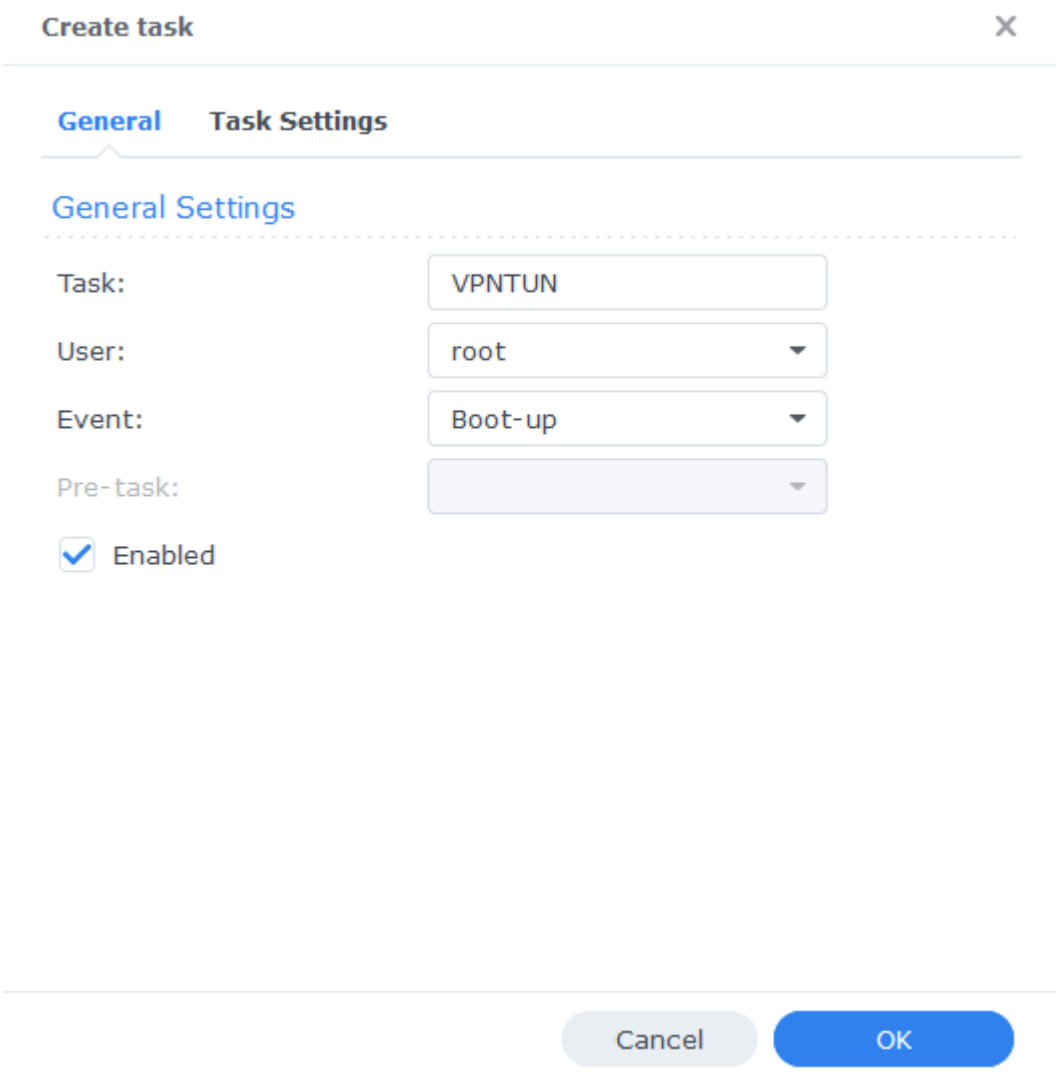
1. Open up Control Panel, then click on Task Scheduler.



2. Next click on Create, Triggered Task then User Defined Script.



3. Now enter a name for the script. It doesn't matter what you choose The user **must** be 'root' and 'Boot-up' for the Event. Don't click OK yet.

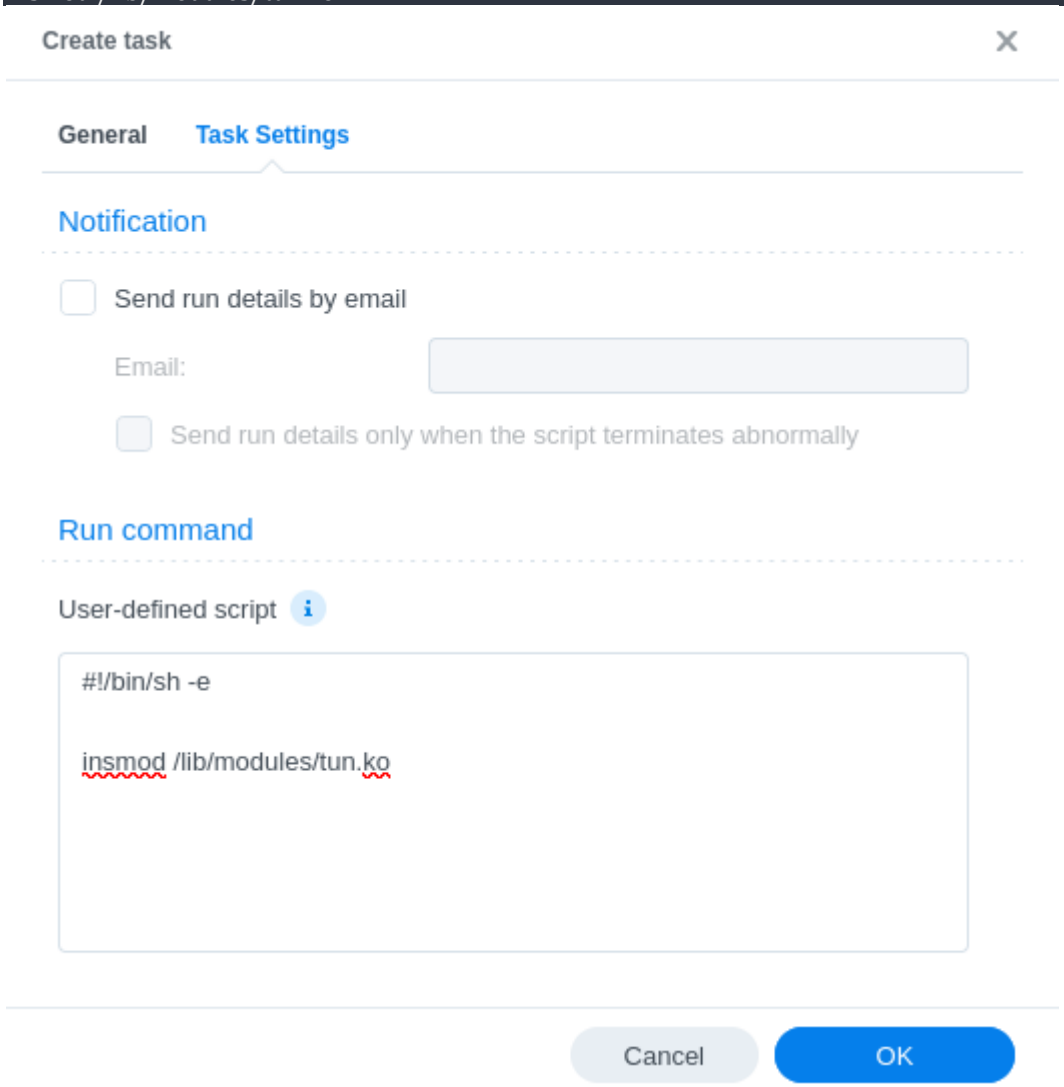


4. On the Task Settings tab copy and paste the code below in the 'User-Defined script' section. It will look like screenshot

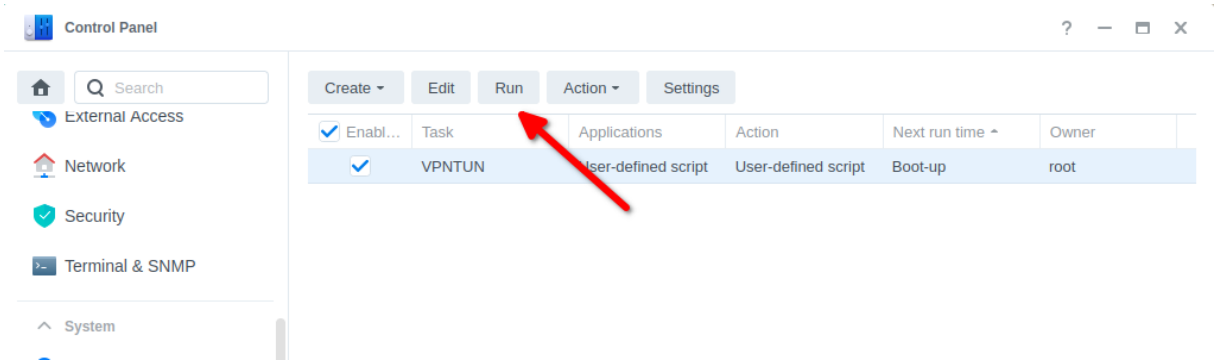
On the Task Settings tab copy and paste the code below in the ‘User-Defined script’ section:

```
#!/bin/sh -e

insmod /lib/modules/tun.ko
```



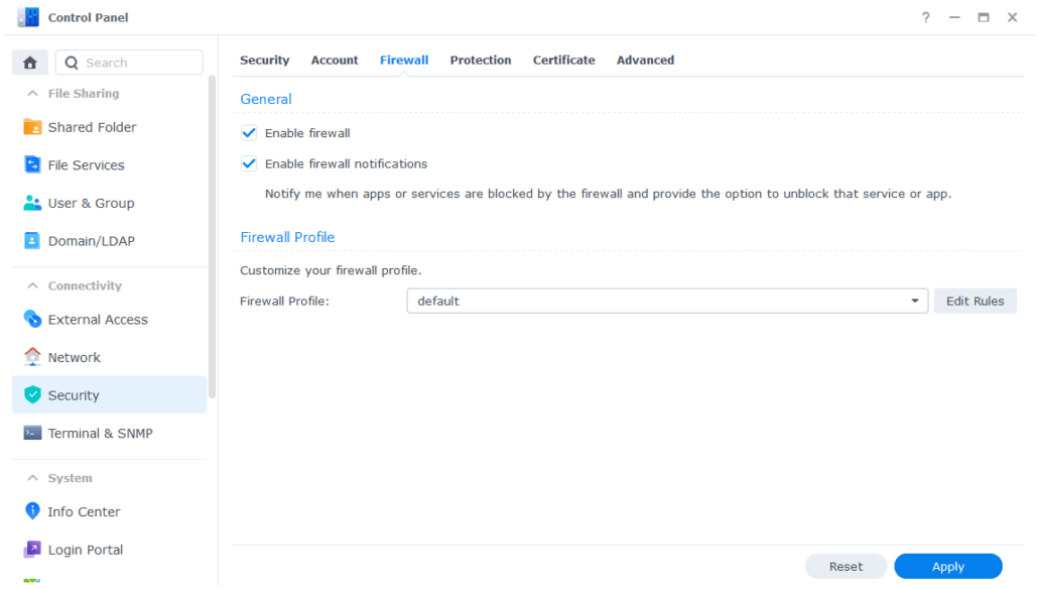
5. You can now press OK and agree to the warning message. Next run the script which will enable the TUN device.



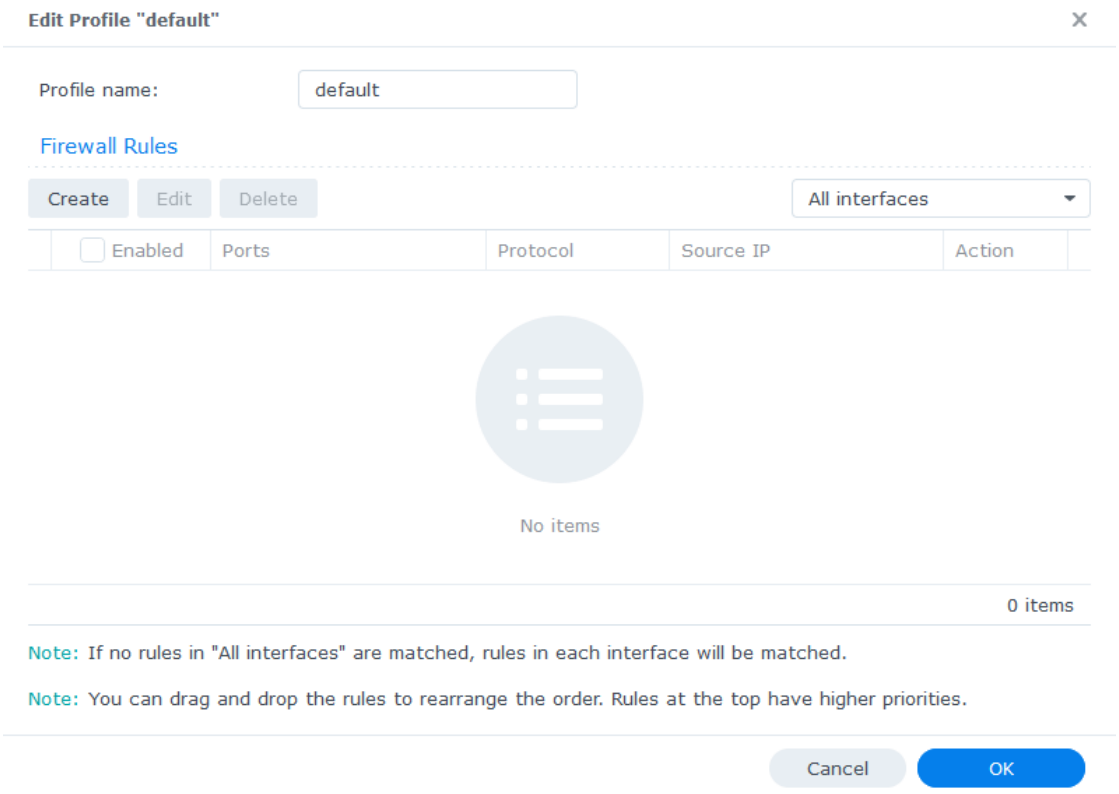
Firewall rules (if you have firewall set up)

If you have firewall rules set up on your synology to block all outgoing connections, we need to make some exception rules.

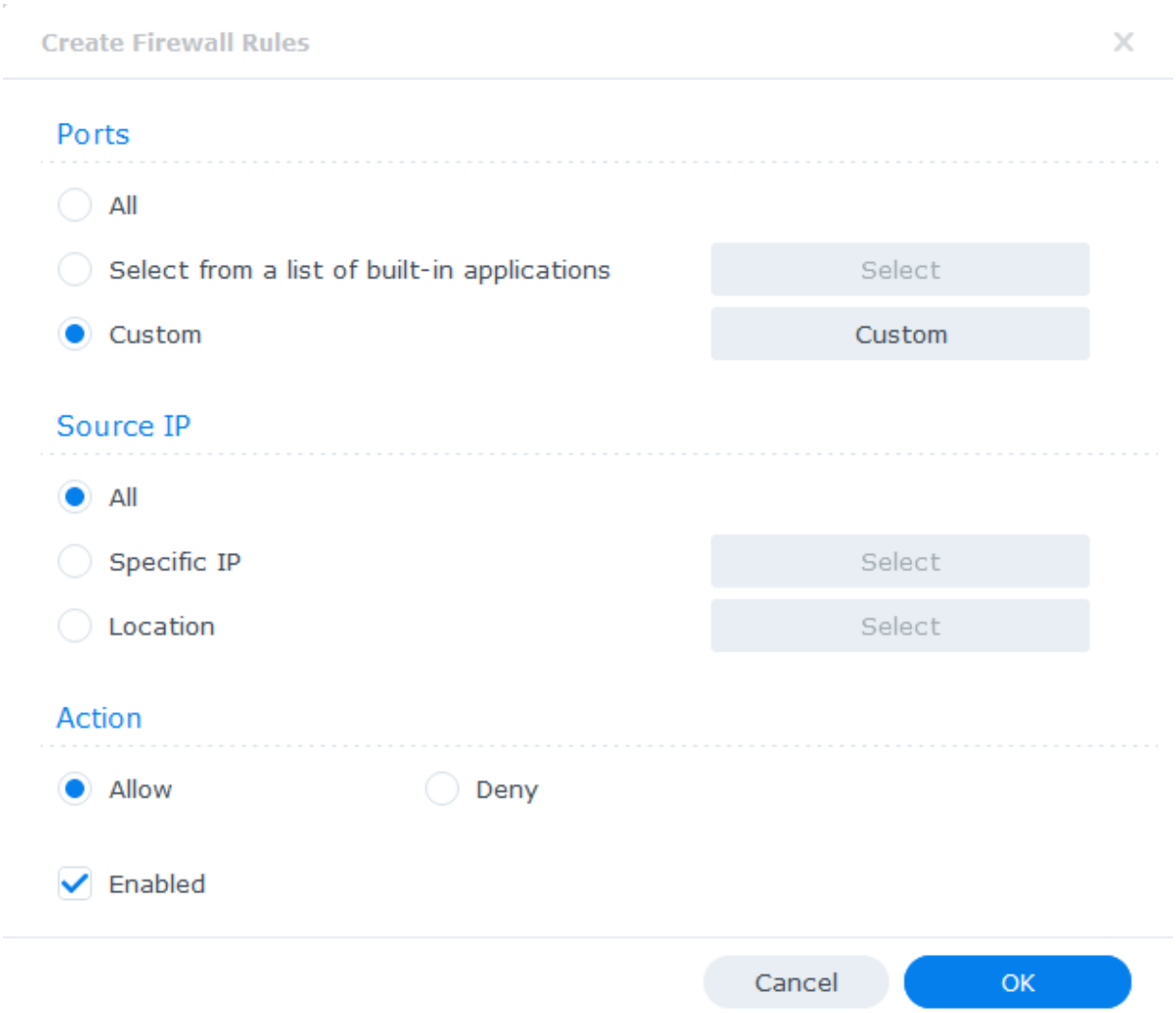
Go into Control Panel > Security > Firewall



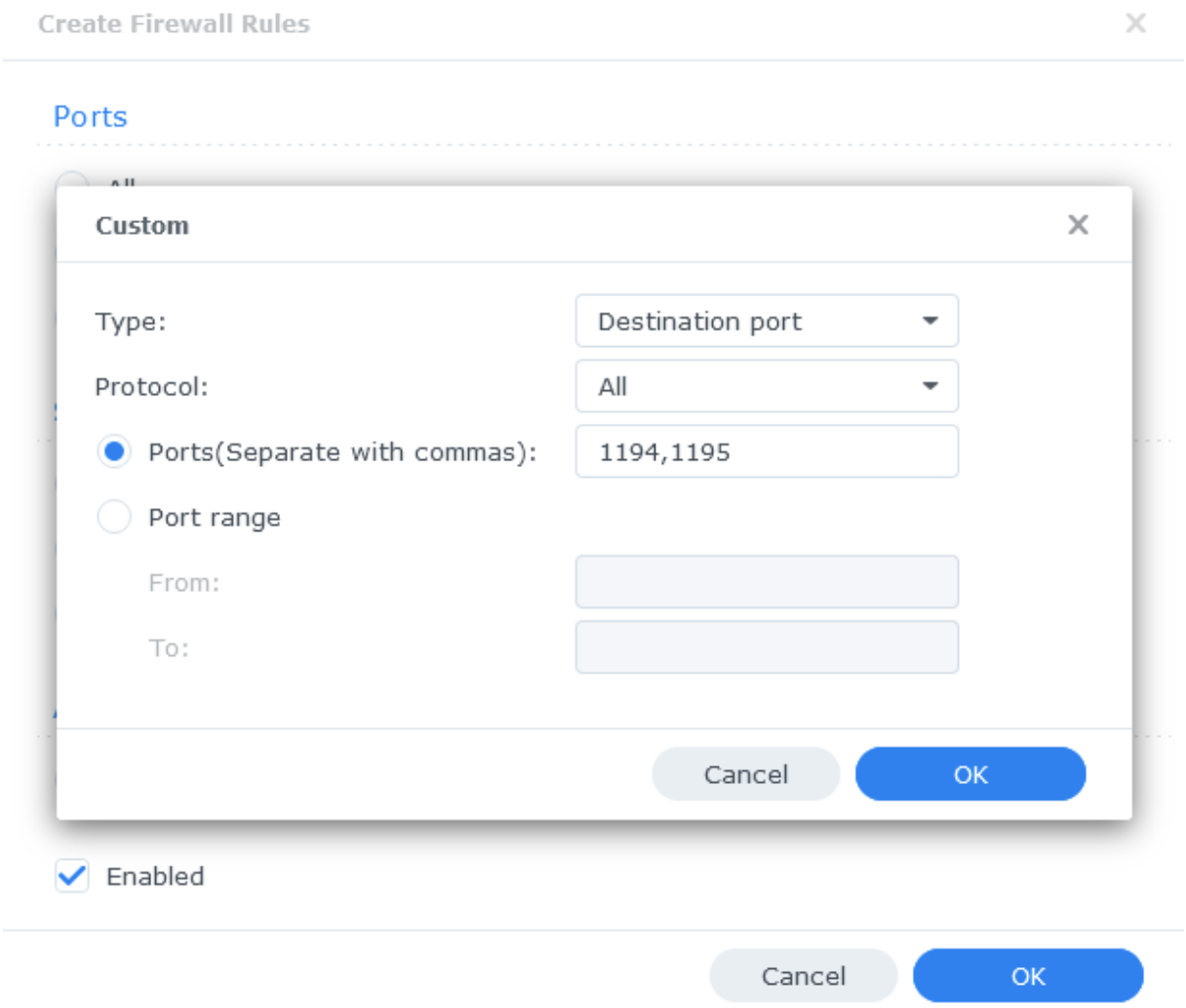
Click on Edit Rules and then click on “Create”



On the “Ports” section, select “Custom”



On the screen that appears select the Type as “Destination Port” and Protocol as “All”. In this example I am going to open up both 1194 and 1195 as some providers use UDP and some TCP and these are the most commonly used ports.



Click on OK. Leave the “Source IP” as “All” and “Action” as “Allow”, then “OK” again to apply.

WireGuard Kernel Module

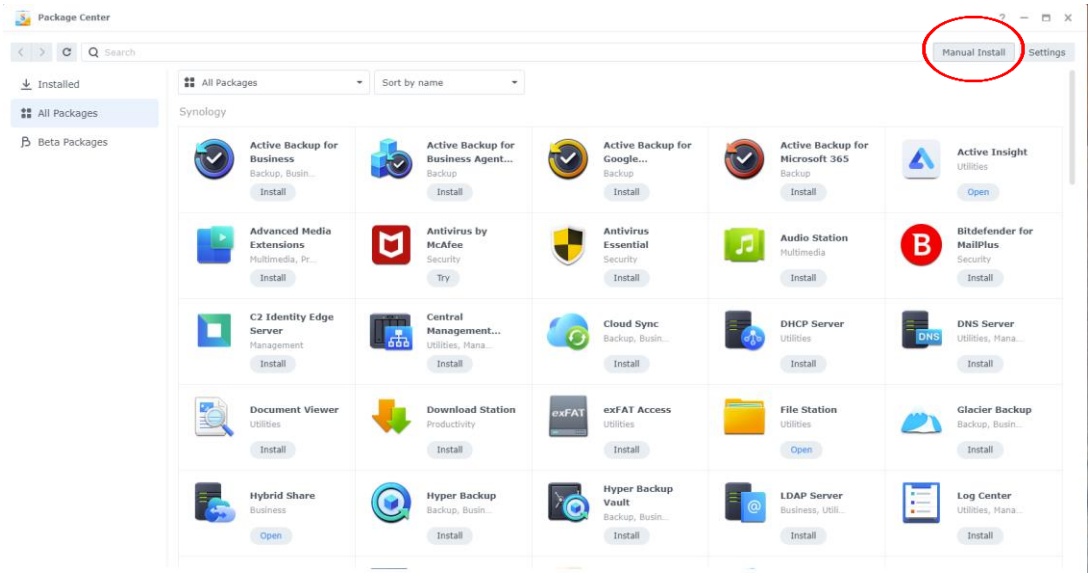
The WireGuard kernel module is not necessary, but it does lower the CPU usage a little bit. This in turn allows for better performance and better power efficiency.

"The default Gluetun Wireguard setup uses a 'Userspace' implementation of Wireguard which normally should not use much from a CPU resource perspective. However, on Synology it tends to require high CPU utilisation.

(DrFrankenstein's Tech Stuff)

BlackVoid.club have put together a Kernel Module for Synology which allows Gluetun to use the lower level Kernel to perform Wireguard duties.

1. Open this link: https://www.blackvoid.club/wireguard-spk-for-your-synology-nas/
2. Find your model of NAS under the correct DSM version section (If you are following this guide it will be 7.2) and download the pre compiled .spk file
3. Head into Package Center and click 'Manual Install' on the top right and install the .spk file and **untick** the box to run after install



4. Reboot
5. SSH Into your NAS using PuTTY, powershell or any other SSH client and elevate yourself to root by typing "sudo -I" and entering your password
6. Enter this command and press enter to start up the module `/var/packages/WireGuard/scripts/start`

Creating a Synology bridge network

As default, there is already a bridge network in container manager for Synology. The problem with the default one is that the IPs it assigns are not static, and therefore may change. This is fine for single containers that don't communicate with each other, but when connecting multiple containers with IPs the IP address needs to always stay the same. To remedy this, we will create our own bridge network.

1. Open container manager and click on the network tab to the left.
2. Click "Add" at the top
3. Configure the network like this:
    - Subnet: 172.20.0.0/16
    - IP range: 172.20.0.2/25
    - Gateway: 172.20.0.1
    - IPv6: Disabled
    - IP Masquerade: enabled (Leave the "disable" option unticked)

Docker project

Now we are ready to create the project. A docker project is just a collection of multiple docker containers.

1. Open “container manager” and head to the “Project” tab.
2. Click create
3. Give the project a name. I chose “arr-stack”
4. Set the path to be “docker > arr-stack” (should say “\docker\arr-stack”)
5. As source, select “Create docker-compose.yml”

Now this next step will be just a little bit different for everyone. What you will put in the docker-compose, will depend on what apps you plan to use and what VPN provider you have. But don’t worry, as you can always come back to the project and edit the docker-compose to add more apps, or to fix any potential problems that may occur. I will try my best to explain what each section does, and if it is relevant to you or not.

Required

Everyone will need to start the docker-compose out like this:

```
version: "3"
services:
```

Then under services we will add all of our apps and configs to each app.

GlueTUN:

```
  gluetun:
    image: qmcgaw/gluetun
    container_name: gluetun
    hostname: gluetun
    cap_add:
      - NET_ADMIN
    devices:
      - /dev/net/tun:/dev/net/tun
    ports:
      - 6881:6881
      - 6881:6881/udp
#     - Add all other ports required by the different apps here
    volumes:
      - </path/to/your/gluetun>:/gluetun
    environment:
      - VPN_SERVICE_PROVIDER=<your provider>
      - VPN_TYPE=<wireguard or openvpn>
      # OpenVPN:
      # - OPENVPN_USER=
      # - OPENVPN_PASSWORD=
      # Wireguard:
      - WIREGUARD_PRIVATE_KEY=<your-private-key>
      - WIREGUARD_ADDRESSES=<your-wireguard-adress>
      - DNS=<your-wireguard-dns>
      - SERVER_HOSTNAMES=<your-hostnames>
      - SERVER_CITIES=<your-cities>
      - HTTPPROXY=off #change to on if you wish to enable
      - SHADOWSOCKS=off #change to on if you wish to enable
      # Timezone for accurate log times
      - TZ=<your-timezone>
      # Server list updater
      # See https://github.com/qdm12/gluetun-
wiki/blob/main/setup/servers.md#update-the-vpn-servers-list
      - UPDATER_PERIOD=24h
      - FIREWALL_OUTBOUND_SUBNETS=<synobridge-subnet>/16,<host-machine-subnet>/24
#change this in line with your subnet see note on guide
#      - FIREWALL_VPN_INPUT_PORTS=12345 #uncomment or remove this line based on
the notes below
    network_mode: synobridge
    labels:
      - com.centurylinklabs.watchtower.enable=false
    security_opt:
      - no-new-privileges:true
    restart: always
```

We have a lot to unpack here. So let's just begin:

Ports

It is in the "ports:" section we put in all the network ports we are going to use. When we look at each of the official docker-compose.yml to all the different apps, they have their ports listed under their own service. However, since we are going to use a VPN, we need it to be in the GlueTUN network.

Volumes

Here we put the path to the folder we made earlier. If you only have one volume on your Synology and followed the same naming scheme as me, it should be "/volume1/docker/arr-stack/gluetun". Then we mount it as "/gluetun" by adding a ":". So the full volume mapping should be "/volume1/docker/arr-stack/gluetun:/gluetun".

Note: Linux systems expect the use of a forward slash (/), and not backwards slash like in Windows (\).

Environment

It's in the environment we put in all the configuration settings.

VPN Service provider

In the "VPN_SERVICE_PROVIDER" you fill in your VPN provider.

Here you can see a list of all the supported providers, as well as how to configure them: https://github.com/qdm12/gluetun-wiki/tree/main/setup/providers

As I have only used Mullvad, AirVPN, and Privado, those are what I will show in detail how to setup.

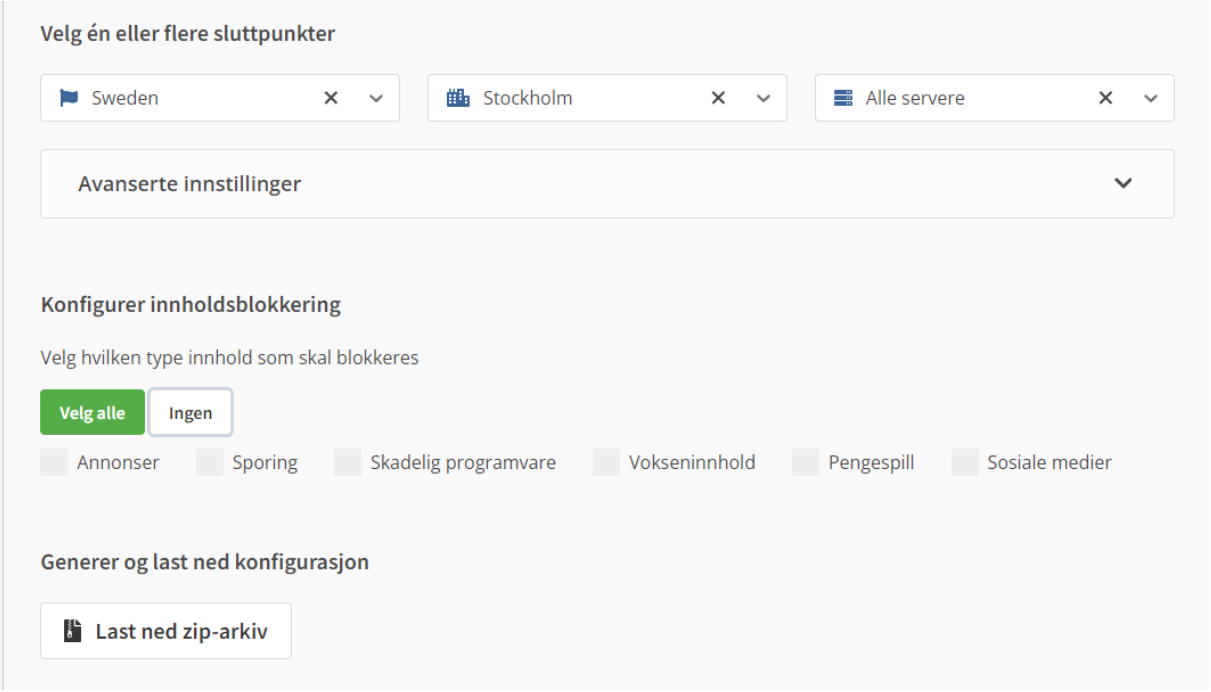
You can also choose whether to use the WireGuard or OpenVPN protocol. In this example, we are using WireGuard as that is what I have found to work best. Except for Privado, which only offers OpenVPN for GlueTUN at the time of writing. But beware, the OpenVPN protocol is slow and resource demanding.

If you don’t use Mullvad, AirVPN or Privado, you will have to find out yourself how to configure GlueTUN for your provider. The official docs and ChatGPT will be your friend here. Skip to Page 42 to continue. But if you are using either Mullvad, AirVPN or Privado, you can follow my guide:

Mullvad

1. Go to this link and put in your account number: https://mullvad.net/no/account/wireguard-config
2. Click on “Generate Key” **Note:** The displayed key is NOT the private key.
3. Choose a country, city and then the servers you wish to use. I have selected Sweden, Stockholm and all servers.
4. Click “Download zip-archive”

0



5. Inside the zip file, you will find a bunch on .conf files. (Might be .json files) Open these in notepad.
6. You only need one of them, as the relevant information is the same in every single one. You need to look at your [Interface] section and copy your “PrivateKey” as well as “Address” and “DNS”.

```
[Interface]
# Device: Hip Prawn
PrivateKey = iExD5V5kkXnh+40dyo/PmCL1aus8eNBdHQMWergYFWo=
Address = 10.72.171.113/32,fc00:bbbb:bbbb:bb01::9:ab70/128
DNS = 10.64.0.1

[Peer]
PublicKey = MkP/Jytkg51/Y/EostONjIN6YaFRpsAYiNKMX27/CAY=
AllowedIPs = 0.0.0.0/0,::0/0
Endpoint = 185.195.233.76:51820
```

So for this example I would have to use:

"iExD5V5kkXnh+40dyo/PmCL1aus8eNBdHQMWergYFWo=" as my private key,

"10.72.171.113/32" as my address and

"10.64.0.1" as my DNS.

7. Also take note of the filenames. They should be something like "se-sto-wg-001.conf", where "se" is the country, "sto" the city and "wg" the protocol. So, for me it's Sweden, Stockholm, WireGuard.

AirVPN

1. Login to AirVPN and head over to "Client Area". Click on "Manage" under "VPN Devices". Or you can click this link: https://airvpn.org/devices/
2. Click on "Add new device". Call it something you will remember. For me, I went with GlueTUN. The name doesn't really matter.
3. Head back to "Client Area", and this time select "Config Generator". Or follow this link: https://airvpn.org/generator/
4. Select Linux as OS and WireGuard as Protocol. Select the server you want to use. For simplicity, I will just go for "Cepheus" in Norway. Scroll all the way down and click on "Generate." A download of a text file should start. If you have selected multiple server, you might need to scroll all the way up where you will find a download for each one.
5. The .conf text file should look kind of like this (Not my actual details, I changed them)

```
[Interface]
Address = 10.184.127.25/32,fd7d:76ee:e68f:a993:a677:d8c0:2660:a8f3/128
PrivateKey = iEIR+vLDwV9KSqC+j1VxolEZ4WhlHvDMXIv20AwKXFw=
MTU = 1320
DNS = 10.128.0.1, fd7d:76ee:e68f:a993::1

[Peer]
PublicKey = PyLCXAQT8KkM4T+dUsOQfn+Ub3pGxfGlxkIApuig+hk=
PresharedKey = //QhNVyHOGTXOB0by0sZ/OZj9PSTwFZ8yI6OiDg+6OY=
Endpoint = 82.102.27.173:1637
AllowedIPs = 0.0.0.0/0,::/0
PersistentKeepalive = 15
```

Now we can start to fill in the information in our docker-compose variables:

```
- VPN_SERVICE_PROVIDER=airvpn
      - VPN_TYPE=wireguard
      - WIREGUARD_PRIVATE_KEY=iEIR+vLDwV9KSqC+j1VxolEZ4WhlHvDMXIv20AwKXFw=
      - WIREGUARD_PRESHARED_KEY=//QhNVyHOGTXOB0by0sZ/OZj9PSTwFZ8yI6OiDg+6OY=
      - WIREGUARD_ADDRESSES=10.184.194.25/32
      - DNS=10.128.0.1,1.1.1.1,fd7d:76ee:e68f:a993::1
      - SERVER_HOSTNAMES=
      - SERVER_REGIONS=Europe
      - SERVER_COUNTRIES=Norway
      - SERVER_CITIES=Oslo
```

If you want to, you can do like me and add more DNS servers. I also added CloudFares 1.1.1.1. Just remember to separate it with a “,”(comma). Same goes for the server regions, countries and cites. If you have multiple, list all of them with a “,”(comma) separating each one. Kind of like this:

```
      - SERVER_COUNTRIES=Norway,Sweden,Denmark
```

Now we just need to find out our server hostname(s). It does not say in the config file, so to find the server hostname we will have to go to github and look at the source code:

https://raw.githubusercontent.com/qdm12/gluetun/refs/heads/master/internal/storage/servers.json.

Use ctrl+f (Search) and search for your server name. For me it was “Cepheus”. For “Cepheus” we have 6 results. You can find the hostname under: "hostname": "no3.ipv6.vpn.airdns.org". So for this server it is “no3.ipv6.vpn.airdns.org”. Select as many as you want, I would say atleast 2-3 so if one goes offline you won’t get disconnected, then put them in the docker-compose with comma separating each one.


```
},
{
  "vpn": "wireguard",
  "country": "Norway",
  "region": "Europe",
  "city": "Oslo",
  "server_name": "Cepheus",
  "hostname": "no.ipv6.vpn.airdns.org",
  "wgpubkey": "PyLCXAQT8KkM4T+dUsOQfn+Ub3pGxfGlxkIApuig+hk=",
  "ips": [
    "2001:ac8:38:22:9f73:11d5:96d8:f0bc"
  ]
},
{
  "vpn": "wireguard",
  "country": "Norway",
  "region": "Europe",
  "city": "Oslo",
  "server_name": "Cepheus",
  "hostname": "no.vpn.airdns.org",
  "wgpubkey": "PyLCXAQT8KkM4T+dUsOQfn+Ub3pGxfGlxkIApuig+hk=",
  "ips": [
    "82.102.27.170"
  ]
},
{
  "vpn": "openvpn",
  "country": "Norway",
  "region": "Europe",
  "city": "Oslo",
  "server_name": "Cepheus",
  "hostname": "no3.ipv6.vpn.airdns.org",
  "tcp": true,
  "udp": true,
  "ips": [
    "2001:ac8:38:22:4340:77ed:b03a:e954"
  ]
},
{
  "vpn": "openvpn",
  "country": "Norway",
  "region": "Europe",
  "city": "Oslo",
  "server_name": "Cepheus",
  "hostname": "no3.vpn.airdns.org",
  "tcp": true,
  "udp": true,
  "ips": [
    "82.102.27.173"
  ]
},
{
  "vpn": "openvpn",
  "country": "Norway",
  "region": "Europe",
  "city": "Oslo",
  "server_name": "Cepheus",
  "hostname": "no4.ipv6.vpn.airdns.org",
  "tcp": true,
  "udp": true,
  "ips": [
    "2001:ac8:38:22:3cfe:32ad:92e0:4ceb"
  ]
},
{
  "vpn": "openvpn",
  "country": "Norway",
  "region": "Europe",
  "city": "Oslo",
  "server_name": "Cepheus",
  "hostname": "no4.vpn.airdns.org",
  "tcp": true,
  "udp": true,
  "ips": [
    "82.102.27.174"
  ]
```


PrivadoVPN

For PrivadoVPN we will have to use OpenVPN. Even though PrivadoVPN supports WireGuard now, GlueTUN does not support Privado with WireGuard. The country

selection pool is also not that when it come sto GlueTUN support. But the good thing about OpenVPN is that it's a whole lot easier to setup. You won't need to download anything, nor find hostnames anywhere. This is how you do it:

1. Copy-Paste this into your docker-compose:

```
      - VPN_SERVICE_PROVIDER=privado
      - VPN_TYPE=openvpn
      - VPN_DISABLE_IPV6=true
      - OPENVPN_USER=
      - OPENVPN_PASSWORD=
      - SERVER_COUNTRIES=
      - SERVER_CITIES=
```

2. Fill in your PrivadoVPN username and password.
3. Fill in your desired Country and City. I would recommend Netherlands, Germany or Hungary Amsterdam as I know this is supported aswell as it's in the EU. The server being in the EU means that the government can't access any logs or anything like that by EU law (GDPR), which means that it's much more secure and private.

The only available servers for PrivadoVPN with GlueTUN are:

1. Germany, Frankfurt.
2. Hungary, Budapest,
3. Netherlands, Amsterdam
4. United States, Los Angeles
5. United States, Washington
6. United States, North Potomac
7. United States, Ashburn

Proxy

If you are using a proxy, you can enable httpproxy and shadowsocks. I have not used it and will therefore not go over that now.

Timezone

In the timezone you just put in your own timezone. To find your TZ format, find your region in this list:
https://en.wikipedia.org/wiki/List_of_tz_database_time_zones

Firewall Outbound Subnets

In this line:

“FIREWALL_OUTBOUND_SUBNETS=<your-bridge-subnet>/,<your-host-subnet>”

We need to change the first IP to the one we just made for our synobridge. If you followed me, it will be 172.20.0.0/16. Then we will need to fill in our host machines subnet,

We have a few choices on how to identify it. If you are connected to the same network on your PC or phone, you can just figure out your IP address on your device of choice. So, for PC, open “cmd” and type “ipconfig”. Look for IPv4 section.

If you have an iPhone, you can open settings >Wi-Fi > i (next to the Wi-Fi named) then scroll down to find the “IP address” field under “IPv4 address”

The IPv4 address will probably be something like this “192.168.X.X”. Take the first 3 digits, e.g. “192.168.0” or whatever you have, then replace the last digit with a 0. So if your IP address on your phone is “192.168.0.53” the subnet would be “192.168.0.0”. Then we just add the network mask at the end. If you don’t know what that means, it’s probably /24 at the end, so the full subnet is “192.168.0.0/24”.

So, my line would look like this:

“FIREWALL_OUTBOUND_SUBNETS=172.20.0.0/16,192.168.0.0/24”

Network_mode

We need to specify the network we made. If you gave it the same name as me, it should be “synobridge”.

labels: - com.centurylinklabs.watchtower.enable=false

These lines just makes it so Watchtower doesn’t update it automatically. The reason we don’t want it to update is because it will break the whole arr-stack each time. So the best practice would be to just leave it in. You can always manually update it when you want.

security_opt: - no-new-privileges:true:

Makes it so the container can’t gain any more privileges than they are assigned, and makes it more secure. Probably best to include on every container.

Restart: always

This line tells the container to restart if it shuts down unexpectedly.

Full GlueTUN docker-compose.yml

With all of the info above, I now need you to make the appropriate edits for your docker-compose. This will be mine for this example:

```
gluetun:
    image: qmcgaw/gluetun
    container_name: gluetun
    hostname: gluetun
    cap_add:
      - NET_ADMIN
    devices:
      - /dev/net/tun:/dev/net/tun
    ports:
      - 6881:6881
      - 6881:6881/udp
      - 8085:8085 # qbittorrent
      - 8989:8989 # Sonarr
      - 9696:9696 # Prowlarr
      - 7878:7878 # Radarr
      - 8686:8686 #Lidarr
      - 8191:8191 #FlareSolverr
      - 5055:5055 #Overseer
      - 4545:4545 #Requestrr

    volumes:
      - \volume1\docker\arr-stack\gluetun:/gluetun
    environment:
      - VPN_SERVICE_PROVIDER=mullvad
      - VPN_TYPE=wireguard
      - VPN_DISABLE_IPV6=true
      # OpenVPN:
      # - OPENVPN_USER=
      # - OPENVPN_PASSWORD=
      # Wireguard:
      - WIREGUARD_PRIVATE_KEY= iExD5V5kkXnh+40dyo/PmCL1aus8eNBdHQMWergYFWo=
      - WIREGUARD_ADDRESSES=10.72.171.113/32
      - DNS=10.64.0.
      - SERVER_HOSTNAMES=se-sto-wg-001,se-sto-wg-002,se-sto-wg-003,se-sto-wg-004
      - SERVER_CITIES=stockholm
      - HTTPPROXY=off #change to on if you wish to enable
      - SHADOWSOCKS=off #change to on if you wish to enable
      # Timezone for accurate log times
      - TZ=Europe/Oslo
      - UPDATER_PERIOD=24h
      - FIREWALL_OUTBOUND_SUBNETS=172.20.0.0/192.168.0.0/24
    network_mode: synobridge
    labels:
      - com.centurylinklabs.watchtower.enable=false
    security_opt:
```

Download client

The best torrent download client in my opinion is qBitTorrent and is therefore what I will use today. But you can use rTorrent or any other preferred torrent client. Here is the docker-compose for qBitTorrent:

```
qbittorrent:
    image: lscr.io/linuxserver/qbittorrent
    container_name: qbittorrent
    network_mode: "service:gluetun"
    environment:
      - PUID=<your-UID>
      - PGID=<your-GID>
      - TZ=<your-timezone>
      - WEBUI_PORT=8085
      - UMASK=022
    volumes:
      - /path/to/your/config:/config
      - /path/to/your/media/torrents:/Media/Torrents
    depends_on:
      gluetun:
        condition: service_healthy
    security_opt:
      - no-new-privileges:true
    restart: always
```

network_mode

If you also are using a VPN, and therefore by extension gluetun, you will need to set network_mode to “service:gluetun”. If you don’t use VPN, you can just do “network_mode: bridge” or any other network you have set up.

Environment

PUID and PGID

For PUID and PGID you need to find the ID for your user. To do this, SSH into your Synology, log in to your own user, then type “id”. The output should include UID (user ID) and GID (Group ID).

UMASK

I simply don't have enough information on this to tell you what it is or how it works. If you are interested, you can read here: https://en.wikipedia.org/wiki/Umask

Volumes

If you followed me in the creation of the directories, it should be:

/volume1/docker/arr-stack/qbittorrent/config:/config

/volume1/Media//orrents:/Media/Torrents

NOTE: It is very important that we have torrents in the same shared folder as our media library. This is because of how hardlinks work. If we do not have it in the same folder, we will then copy the files. So, for every 10GB file, you would have to use 20GB of storage. This adds up really quickly to many TBs of wasted storage.

Full docker-compose.yml for qbittorrent

The full docker-compose could look something like this:

```
qbittorrent:
    image: lscr.io/linuxserver/qbittorrent
    container_name: qbittorrent
    network_mode: "service:gluetun"
    environment:
      - PUID=1026
      - PGID=100
      - TZ=Europe/Oslo
      - WEBUI_PORT=8085
      - UMASK=022
    volumes:
      - /volume1/docker/arr-stack/qbittorrent/config:/config
      - /volume1/Media/Torrents:/media/Torrents
    depends_on:
      gluetun:
        condition: service_healthy
    security_opt:
      - no-new-privileges:true
    restart: always
```

Sonarr

This is the docker-compose.yml

```
sonarr:
    image: lscr.io/linuxserver/sonarr:latest
    container_name: sonarr
    network_mode: "service:gluetun"
    environment:
      - PUID=1026
      - PGID=100
      - TZ=Europe/Oslo
    volumes:
      - /path/to/your/config:/config
      - /path/to/your/media:/Media
    depends_on:
      gluetun:
        condition: service_healthy
    restart: unless-stopped
```

We have already covered most of the components in the docker-compose file, so I won't repeat myself. The only thing I want to mention is the fact that if you separate your TV shows and Anime shows in your Media folder, like me, you need to mount both of them here.

Something important:

The path to your media **NEED** to be in the same shared folder as the torrents, for the reason we talked about earlier.

Full docker-compose.yml file

```
sonarr:
    image: lscr.io/linuxserver/sonarr:latest
    container_name: sonarr
    network_mode: "service:gluetun"
    environment:
      - PUID=<your-UID>
      - PGID=<your-GID>
      - TZ=Europe/Oslo
    volumes:
      - /volume1/docker/arr-stack/sonarr/config:/config
      - /volume1/Media:/Media
    depends_on:
      gluetun:
        condition: service_healthy
    restart: unless-stopped
```

Radarr

Here we have Radarr's docker-compose:

```
radarr:
    image: lscr.io/linuxserver/radarr:latest
    container_name: radarr
    network_mode: "service:gluetun"
    environment:
      - PUID=<your-UID>
      - PGID=<your-GID>
      - TZ=Europe/Oslo
    volumes:
      - /volume1/docker/arr-stack/radarr/config:/config
      - /volume1/Media:/Media
    depends_on:
      gluetun:
        condition: service_healthy
    restart: unless-stopped
```

So, as you can see, it's identical to Sonarr's, except for the image and name.

Lidarr

Docker-compose:

```
lidarr:
    image: lscr.io/linuxserver/lidarr:latest
    container_name: lidarr
    network_mode: "service:gluetun"
    environment:
      - PUID=<your-UID>
      - PGID=<your-GID>
      - TZ=Europe/Oslo
    volumes:
      - /volume1/docker/arr-stack/lidarr/config:/config
      - /volume1/Media:/Media
    depends_on:
      gluetun:
        condition: service_healthy
    restart: unless-stopped
```

Again, we have a pretty familiar looking docker-compose file. Edit the appropriate settings just like you did for Radarr.

Prowlarr

Prowlarr is what we use connect our indexers to our apps. Here is the docker-compose:

```
  prowlarr:
    image: lscr.io/linuxserver/prowlarr:latest
    container_name: prowlarr
    network_mode: "service:gluetun"
    environment:
      - PUID=<your-UID>
      - PGID=<your-GID>
      - TZ=Europe/Oslo
    volumes:
      - /path/to/your/config:/config
    depends_on:
      gluetun:
        condition: service_healthy
    restart: unless-stopped
```

Obviously change the PUID and PGID. Other than that, we only have to change the config path. It should be “/volume1/docker/arr-stack/prowlarr/config:/config”.

Flaresolverr

NOTE: This does not work anymore.

https://trash-guides.info/Prowlarr/prowlarr-setup-flaresolverr/

Docker-compose:

```
flaresolverr:
    image: ghcr.io/flaresolverr/flaresolverr:latest
    container_name: flaresolverr
    network_mode: "service:gluetun"
    environment:
      - TZ=Europe/Oslo
    depends_on:
      gluetun:
        condition: service_healthy
    security_opt:
      - no-new-privileges:true
    restart: unless-stopped
```

You don’t need to change anything here. Just paste it in. The only exception is if you don’t use VPN, then change “network_mode” and delete “depends_on”.

Overseerr

Docker-compose:

```
overseerr:
    image: sctx/overseerr:latest
    container_name: overseerr
    network_mode: "service:gluetun"
    environment:
      - LOG_LEVEL=debug
      - TZ=Europe/Oslo
    volumes:
      - /path/to/your/config:/app/config
    depends_on:
      gluetun:
        condition: service_healthy
    restart: unless-stopped
```

The only thing you need to change here is the path to your config. It should be “/volume1/docker/arr-stack/overseer/config:/app/config”

Requestrr

Docker-compose:

```
requestrr:
   image: darkalfx/requestrr
   container_name: requestrr
   network_mode: "service:gluetun"
   volumes:
   - /path/to/your/config:/root/config
   depends_on:
     gluetun:
       condition: service_healthy
   restart: unless-stopped
```

Again, only change the config path. Should be

“\volume1\docker\arr-stack\requestrr\config”

Tautulli

Docker-compose:

```
tautulli:
  image: ghcr.io/tautulli/tautulli
  container_name: tautulli
  network_mode: service:gluetun #Replace this bridge if you don't use GlueTUN
  restart: unless-stopped
  volumes:
    - /path/to/your/config:/config
  environment:
    - PUID=<your UID>
    - PGID=<your GID>
    - TZ=Europe/Oslo
  depends_on: #use this if you use GlueTUN for VPN
    gluetun:
      condition: service_healthy
```

For me it would look like this:

```
tautulli:
  image: ghcr.io/tautulli/tautulli
  container_name: tautulli
  network_mode: service:gluetun
  restart: unless-stopped
  volumes:
    - /volume1/docker/arr-stack/tautulli/config:/config
  environment:
    - PUID=1026
    - PGID=100
    - TZ=Europe/Oslo
  depends_on:
    gluetun:
      condition: service_healthy
```

Putting it all together

Now we can put all our relevant docker-compose files together inside one big file. Remember, if you want to add more *arr apps or other download-clients, you can just add them in this docker-compose just like we did for all the other apps. Just remember to open up the ports in GlueTUN as well. For me the docker-compose.yml is like this:

```
version: "3"
services:
  gluetun:
    image: qmcgaw/gluetun
    container_name: gluetun
    hostname: gluetun
    cap_add:
      - NET_ADMIN
    devices:
      - /dev/net/tun:/dev/net/tun
    ports:
      - 6881:6881
      - 6881:6881/udp
      - 8085:8085 # qbittorrent
      - 8989:8989 # Sonarr
      - 9696:9696 # Prowlarr
      - 7878:7878 # Radarr
      - 8686:8686 #Lidarr
      - 8191:8191 #FlareSolverr
      - 5055:5055 #Overseerr
      - 4545:4545 #Requestrr

    volumes:
      - \volume1\docker\arr-stack\gluetun:/gluetun
    environment:
      - VPN_SERVICE_PROVIDER=mullvad
      - VPN_TYPE=wireguard
      - VPN_DISABLE_IPV6=true
      # OpenVPN:
      # - OPENVPN_USER=
      # - OPENVPN_PASSWORD=
      # Wireguard:
      - WIREGUARD_PRIVATE_KEY= iExD5V5kkXnh+40dyo/PmCL1aus8eNBdHQMWergYFWo=
      - WIREGUARD_ADDRESSES=10.72.171.113/32
      - DNS=10.64.0.
      - SERVER_HOSTNAMES=se-sto-wg-001,se-sto-wg-002,se-sto-wg-003,se-sto-wg-004
```

```
      - SERVER_CITIES=stockholm
      - HTTPPROXY=off #change to on if you wish to enable
      - SHADOWSOCKS=off #change to on if you wish to enable
      # Timezone for accurate log times
      - TZ=Europe/Oslo
      # Server list updater
      # See https://github.com/qdm12/gluetun-
wiki/blob/main/setup/servers.md#update-the-vpn-servers-list
      - UPDATER_PERIOD=24h
      - FIREWALL_OUTBOUND_SUBNETS=172.20.0.0/192.168.0.0/24 #change this in line
with your subnet see note on guide
#      - FIREWALL_VPN_INPUT_PORTS=12345 #uncomment or remove this line based on
the notes below
    network_mode: synobridge
    labels:
      - com.centurylinklabs.watchtower.enable=false
    security_opt:

  qbittorrent:
    image: lscr.io/linuxserver/qbittorrent
    container_name: qbittorrent
    network_mode: "service:gluetun"
    environment:
      - PUID=1026
      - PGID=100
      - TZ=Europe/Oslo
      - WEBUI_PORT=8085
      - UMASK=022
    volumes:
      - /volume1/docker/arr-stack/qbittorrent/config:/config
      - /volume1/Media/Torrents:/Media/Torrents
    depends_on:
      gluetun:
        condition: service_healthy
    security_opt:
      - no-new-privileges:true
    restart: always

  sonarr:
    image: lscr.io/linuxserver/sonarr:latest
    container_name: sonarr
    network_mode: "service:gluetun"
    environment:
      - PUID=1026
      - PGID=100
      - TZ=Europe/Oslo
```

```
    volumes:
      - /volume1/docker/arr-stack/sonarr/config:/config
      - /volume1/Media:/Media
    depends_on:
      gluetun:
        condition: service_healthy
    restart: unless-stopped

  prowlarr:
    image: lscr.io/linuxserver/prowlarr:latest
    container_name: prowlarr
    network_mode: "service:gluetun"
    environment:
      - PUID=1026
      - PGID=100
      - TZ=Europe/Oslo
    volumes:
      - /volume1/docker/arr-stack/prowlarr/config:/config
    depends_on:
      gluetun:
        condition: service_healthy
    restart: unless-stopped

  radarr:
    image: lscr.io/linuxserver/radarr:latest
    container_name: radarr
    network_mode: "service:gluetun"
    environment:
      - PUID=1026
      - PGID=100
      - TZ=Europe/Oslo
    volumes:
      - /volume1/docker/arr-stack/radarr/config:/config
      - /volume1/Media:/Media
    depends_on:
      gluetun:
        condition: service_healthy
    restart: unless-stopped

  lidarr:
    image: lscr.io/linuxserver/lidarr:latest
    container_name: lidarr
    network_mode: "service:gluetun"
    environment:
      - PUID=1026
      - PGID=100
```

```
    - TZ=Europe/Oslo
  volumes:
    - /volume1/docker/arr-stack/lidarr/config:/config
    - /volume1/Media:/Media
  depends_on:
    gluetun:
      condition: service_healthy
  restart: unless-stopped

flaresolverr:
  image: ghcr.io/flaresolverr/flaresolverr:latest
  container_name: flaresolverr
  network_mode: "service:gluetun"
  environment:
    - TZ=Europe/Oslo
  depends_on:
    gluetun:
      condition: service_healthy
  security_opt:
    - no-new-privileges:true
  restart: unless-stopped

overseerr:
  image: sctx/overseerr:latest
  container_name: overseerr
  network_mode: "service:gluetun"
  environment:
    - LOG_LEVEL=debug
    - TZ=Europe/Oslo
  volumes:
    - /volume1/docker/arr-stack/overseer/config:/app/config
  depends_on:
    gluetun:
      condition: service_healthy
  restart: unless-stopped

requestrr:
  image: darkalfx/requestrr
  container_name: requestrr
  network_mode: "service:gluetun"
  volumes:
  - /volume1/docker/arr-stack/requestrr/config:/root/config
  depends_on:
    gluetun:
      condition: service_healthy
  restart: unless-stopped
```

```yaml
tautulli:
  image: ghcr.io/tautulli/tautulli
  container_name: tautulli
  network_mode: service:gluetun
  restart: unless-stopped
  volumes:
    - /volume1/docker/arr-stack/tautulli/config:/config
  environment:
    - PUID=1026
    - PGID=100
    - TZ=Europe/Oslo
  depends_on:
    gluetun:
      condition: service_healthy
```

Common Errors

The most common error to get now is “gluetun is unhealthy” If you get this, it is likely an error in the config file. Usually, it relates to the provider specific elements. If you check the logs for the GlueTUN container it will tell you why it couldn’t connect. My best guess would be incorrect private key, incorrect hostnames or something similar. If you can’t figure it out, please drop a comment or DM me with your logs, and I’ll take a look.

Also please, please, please! Double check the volume mounts. If they are not correctly set up, you will lose **HALF** your storage space to waste. It should be as follows:

```
Media
├── Torrents
│   ├── Incomplete
│   └── Complete
│       ├── Movies
│       ├── Music
|       ├── Anime TV Shows
│       └── TV Shows
├── Usenet
│   ├── Incomplete
│   └── Complete
│       ├── Movies
│       ├── Music
|       ├── Anime TV Shows
│       └── TV Shows
└── Media
    ├── Movies
    ├── Music
    ├── Anime TV Shows
    └── TV Shows
```

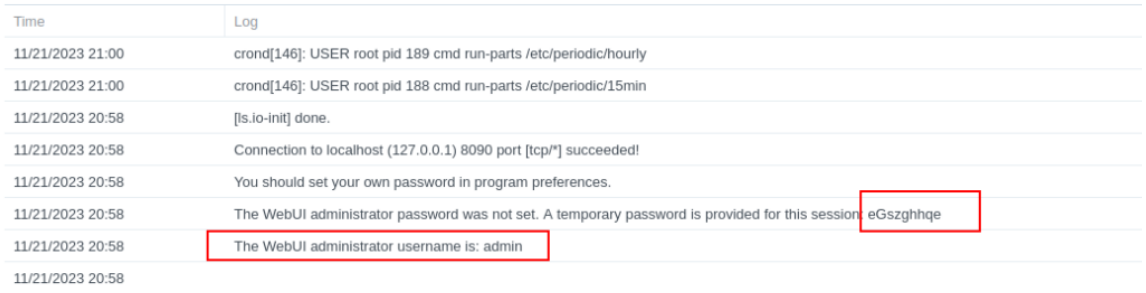
Configuration of the apps

qBitTorrent

Login

The first thing we should do is configure qBitTorrent. Open a web browser (on your computer) then type in your NAS IP address followed by port 8085. For this example, it would look like this: 192.168.0.2:8085

You will then get to the login page for the qBitTorrent WebUI. The username is always admin. The password could be adminadmin, as this is the default. But most likely you will find a temporary password in the logs for the qBitTorrent container inside container manager.

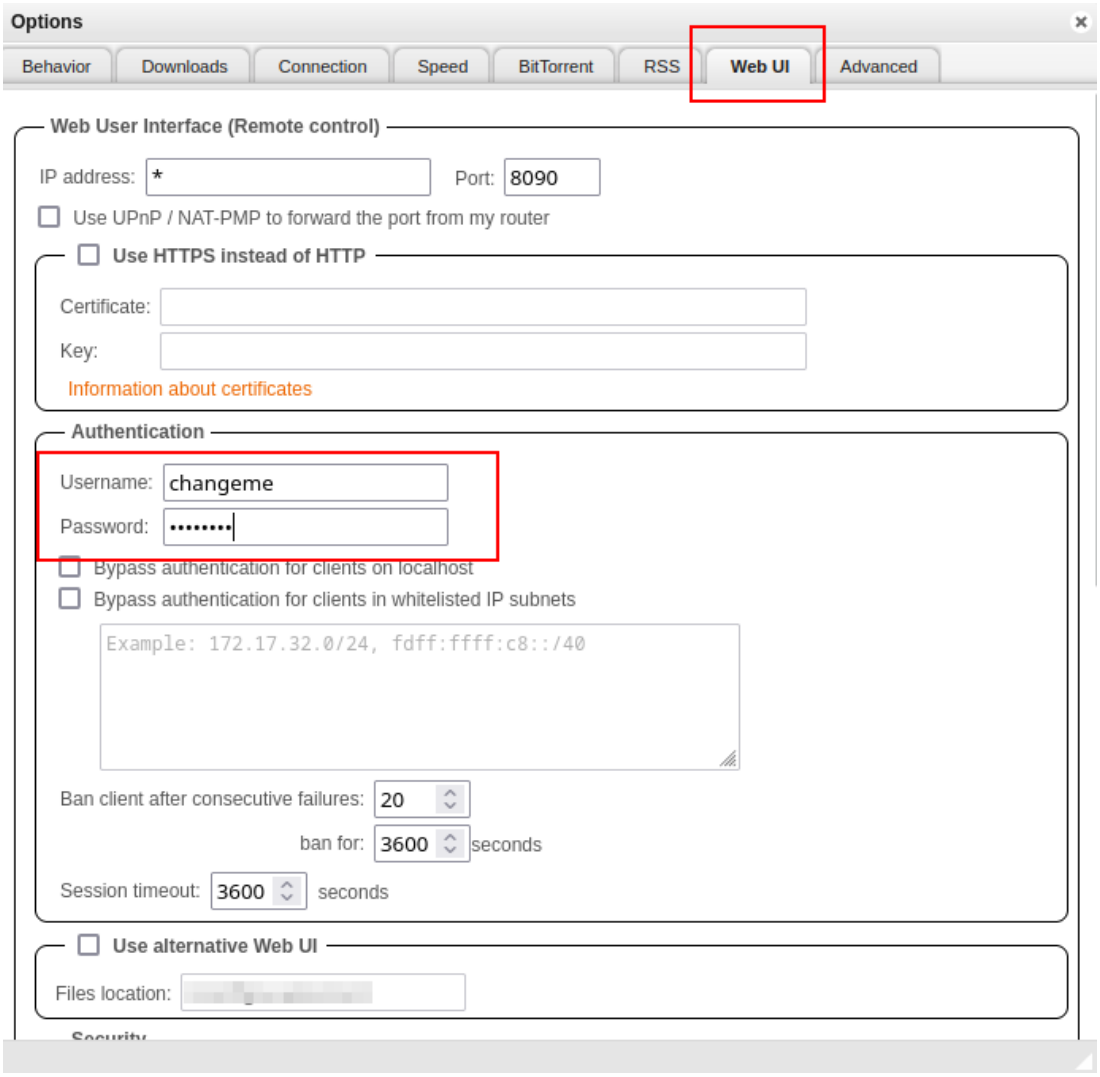


Time	Log
11/21/2023 21:00	crond[146]: USER root pid 189 cmd run-parts /etc/periodic/hourly
11/21/2023 21:00	crond[146]: USER root pid 188 cmd run-parts /etc/periodic/15min
11/21/2023 20:58	[ls.io-init] done.
11/21/2023 20:58	Connection to localhost (127.0.0.1) 8090 port [tcp/*] succeeded!
11/21/2023 20:58	You should set your own password in program preferences.
11/21/2023 20:58	The WebUI administrator password was not set. A temporary password is provided for this session: eGszghhqe
11/21/2023 20:58	The WebUI administrator username is: admin
11/21/2023 20:58	

Change username and password

The first thing to do now that you are logged in is to change your username and password. Click the cog icon at the top of the page, then go to the Web UI tab. Put in your new details then click save at the bottom of the page.

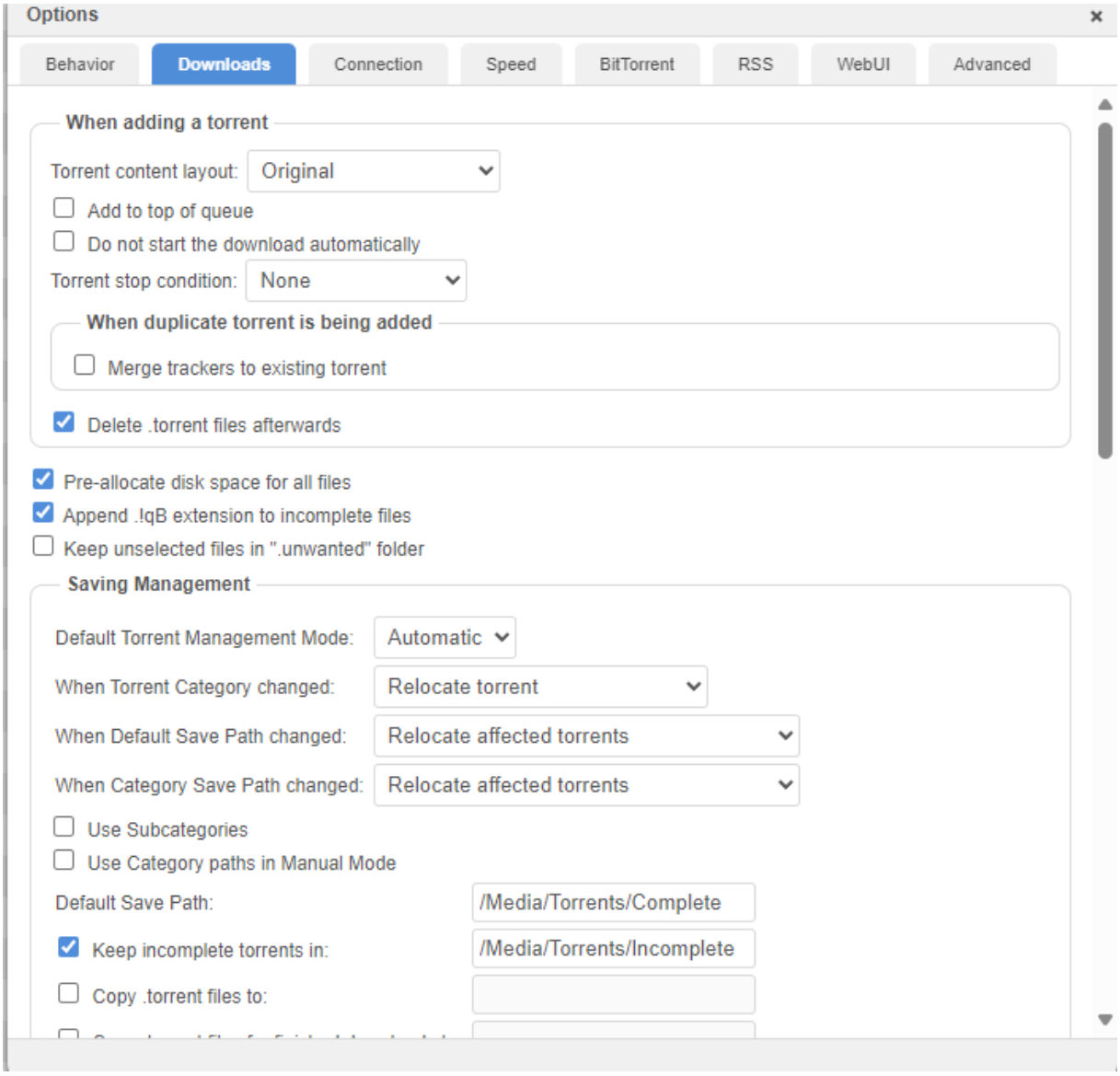
NOTE: Please change both your username and password for maximum security.



Change downloads path

Go back to settings, then go to the “Downloads” tab.

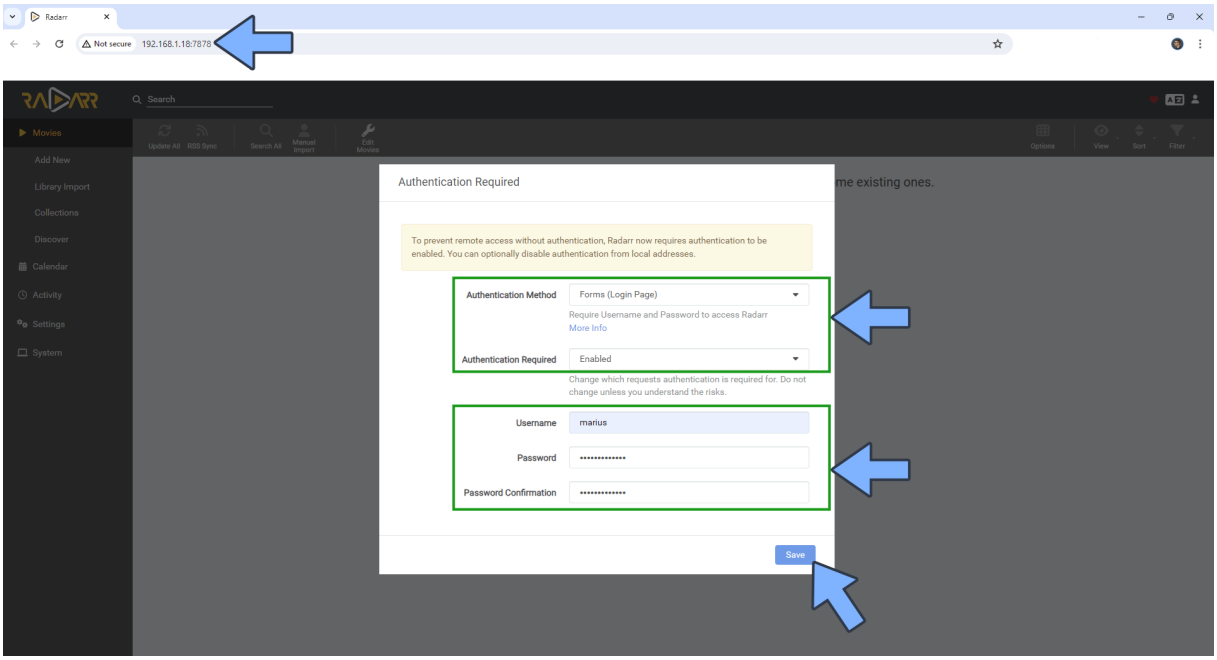
Change the default save path to “/Media/Torrents/Complete” and tick the box for “Keep incomplete torrents in:”. Select “/Media/Torrents/Incomplete”



Radarr

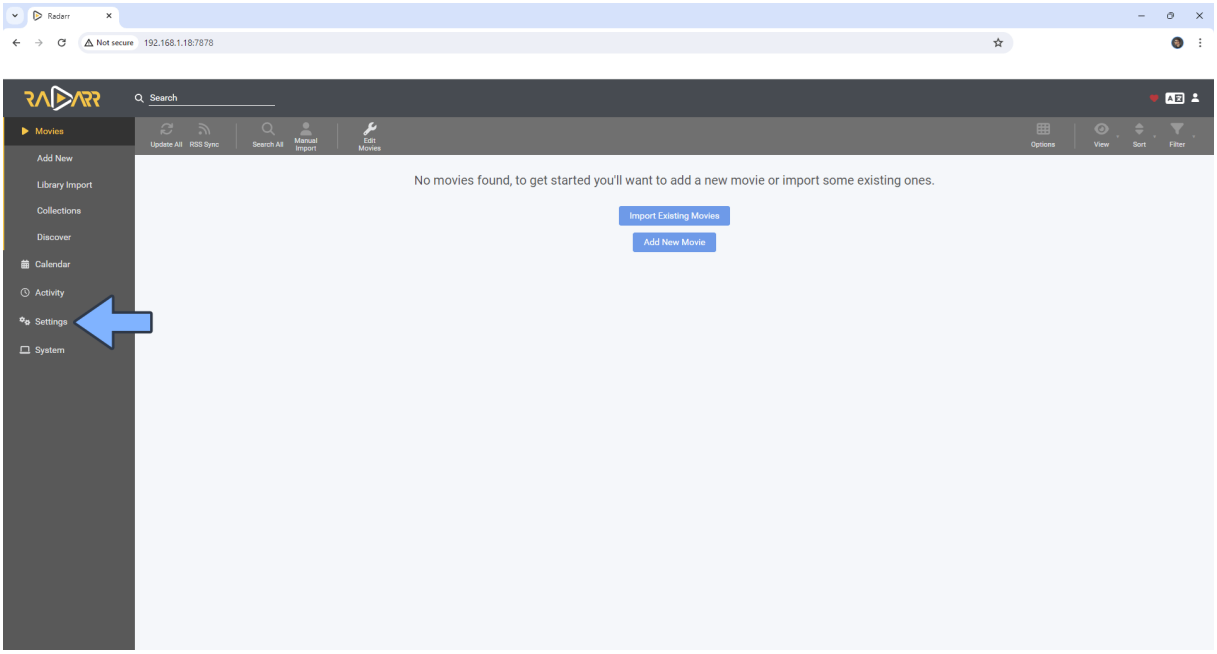
Adding Authentication method

Open a web browser on your PC, then type in the IP of your NAS followed by port 7878. So e.g. 192.168.0.2:7878. The first time you open this, you will be prompted to add an authentication method. I recommend using forms and having it enabled. Set username to something else than admin or administrator as this is easy to guess.

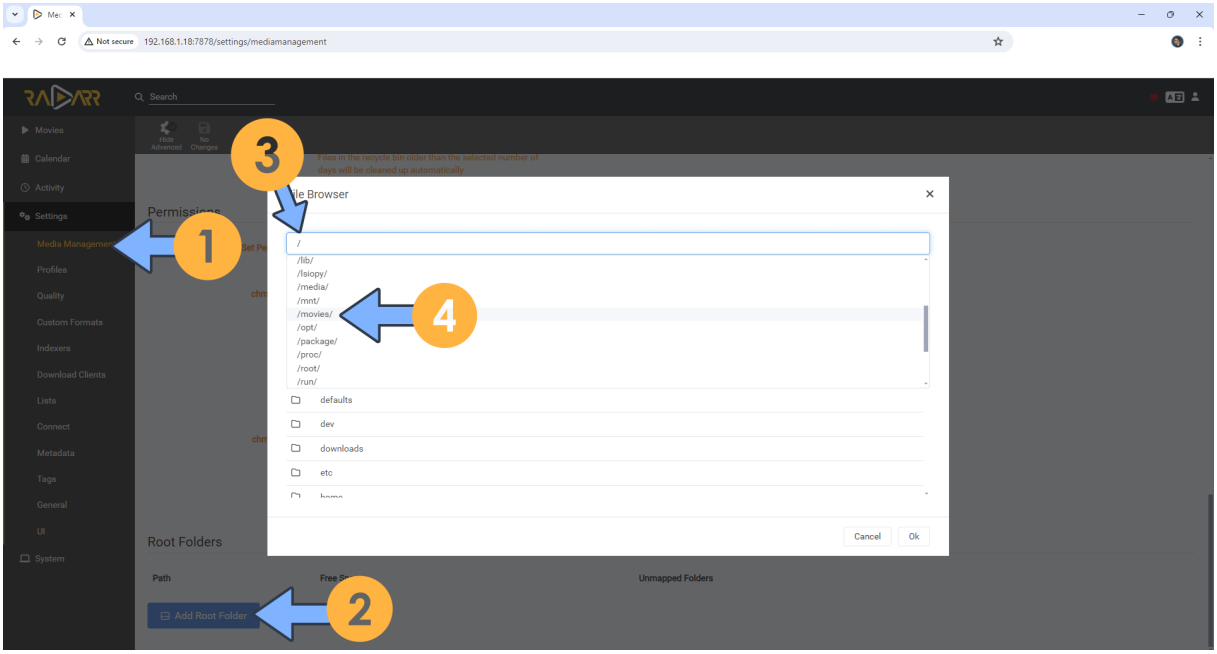


Adding Root Folder

Now go to settings on the left-hand side menu.

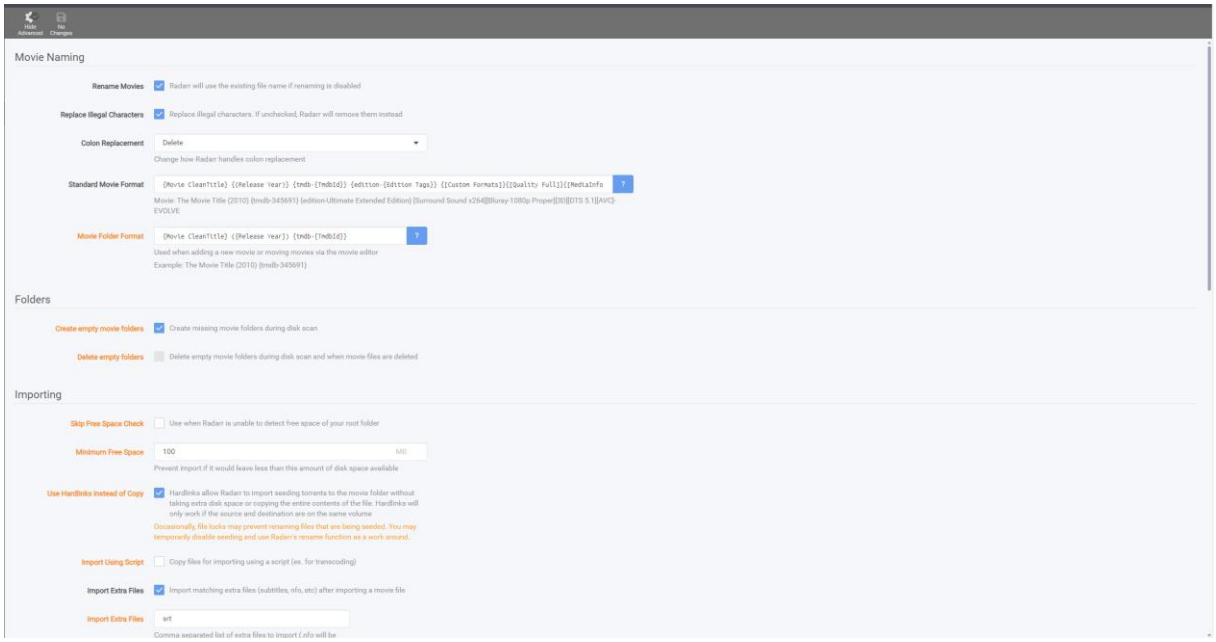


Now go to media management, scroll down and click on “Add Root Folder”. Type /Media/Media/Movies and select it. Click OK.

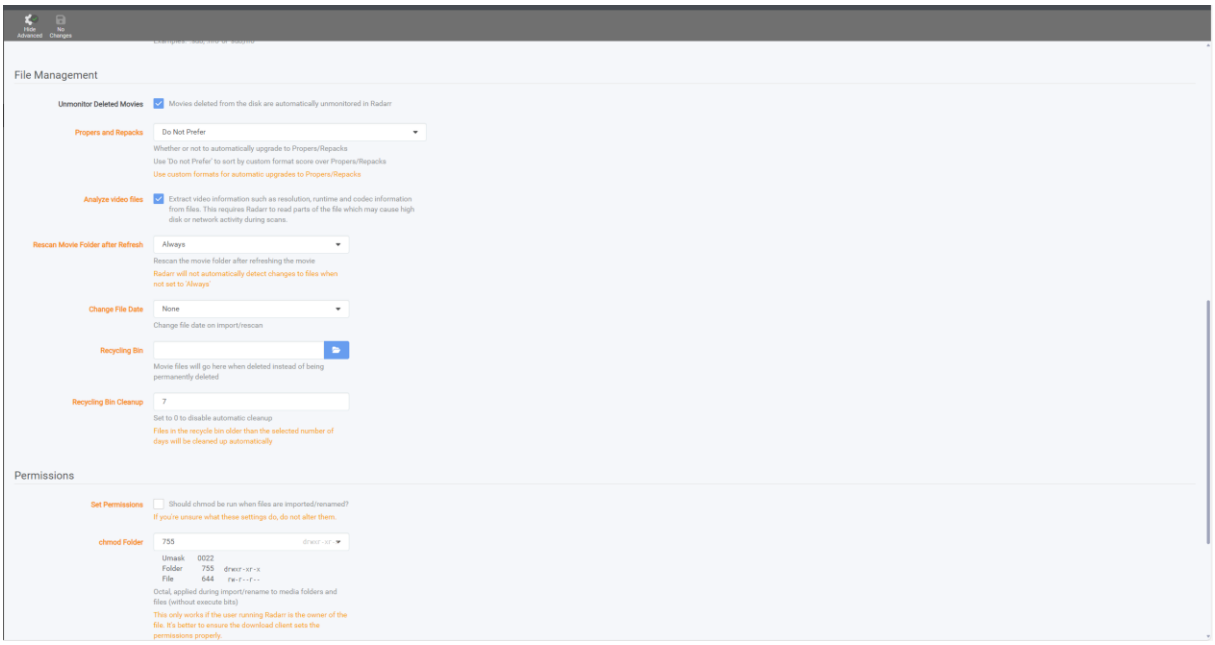


Changing Movie Naming Scheme

1. Go to settings, then choose media management. Turn on advanced settings (right under the search bar).
2. Tick the box next to “Rename Movies” and “Replace illegal characters.”
3. Change the “Standard Movie Format” to the one that fits you the best on TRaSH’s website. For Plex, I recommend to stick with Plex (TVDB). You can read more about recommended naming schemes here: https://trash-guides.info/Radarr/Radarr-recommended-naming-scheme/
4. For Movie Folder Format I would go for “{Movie CleanTitle} ({Release Year}) {tmdb-{TmdbId}}”, but again you can read more on TRaSH Guides.



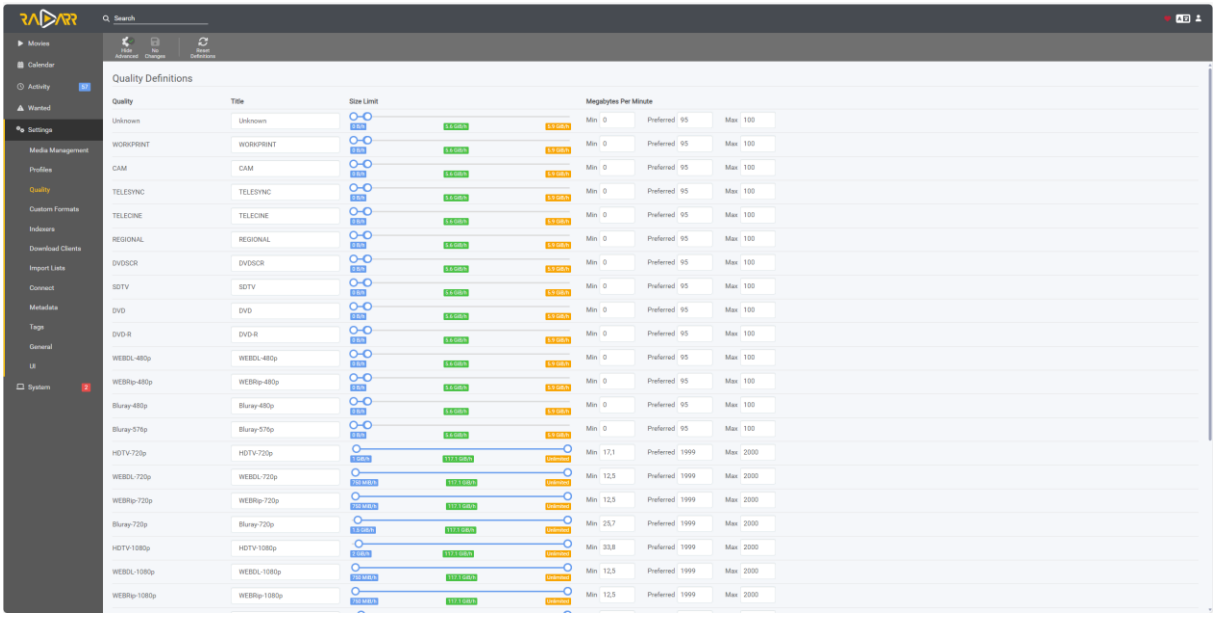
5. Make sure “Use Hardlinks instead of Copy” under “Importing” is enabled.
6. Scroll down until you get to “File Management” and select “Do Not Prefer for” “Propers and Repacks”. This is important for our quality profiles later.



Now be sure to click save, next to the “Show advanced” switch under the search bar.

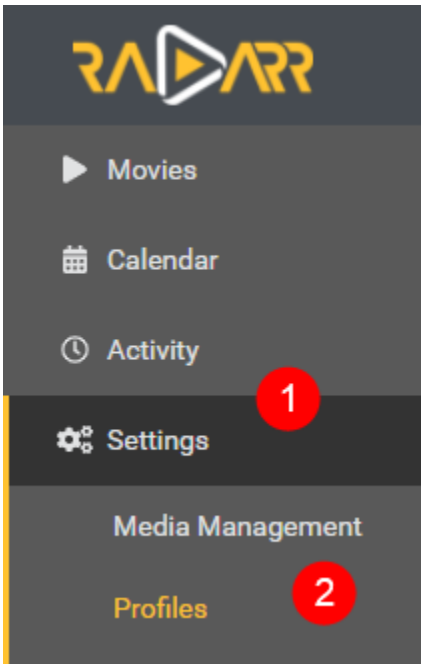
Quality Settings (File Size)

1. Go to settings, then select “Quality” from the menu. Make sure to enable “Show Advanced” by clicking the cog under the search bar.
2. Go to TRaSH Guids and change the appropriate settings as described for the best quality settings. You can edit these at will to save storage space. https://trash-guides.info/Radarr/Radarr-Quality-Settings-File-Size/



Quality profiles

1. Go to settings, Profiles.

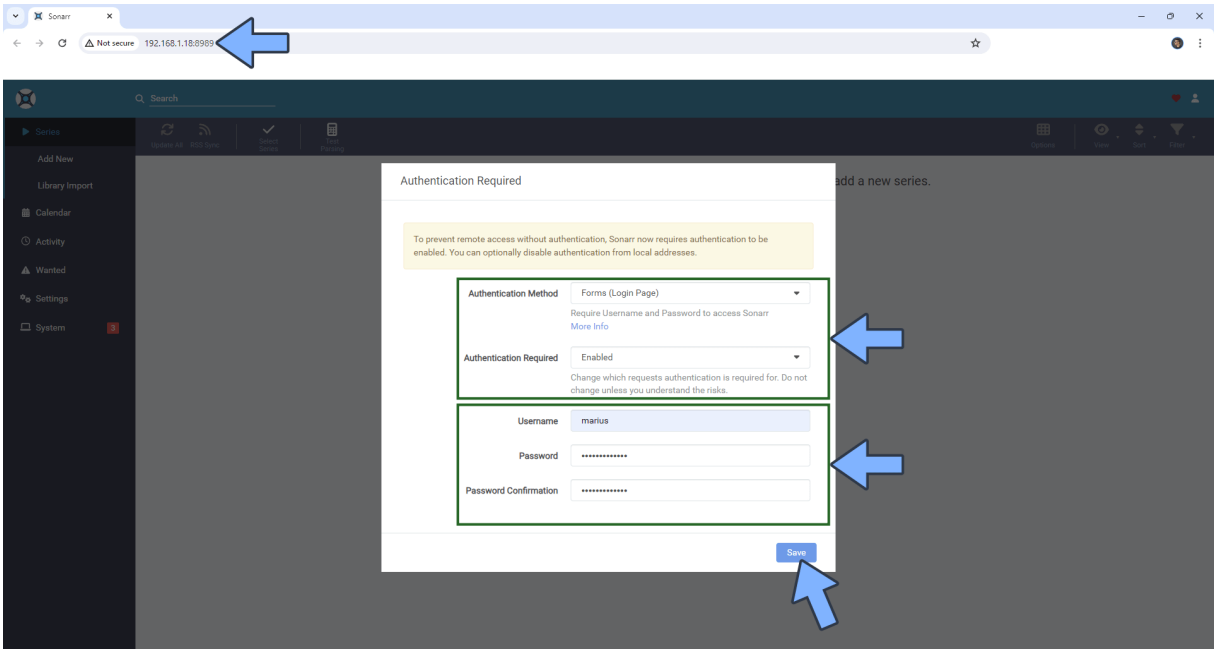


2. Delete all the default ones.
3. Create a new one
4. Go to TRaSH Guide's and follow their instructions on how to setup quality profiles, aswell as to set custom formats. Add all the profiles you need for your setup's needs and wants.
   https://trash-guides.info/Radarr/radarr-setup-quality-profiles/

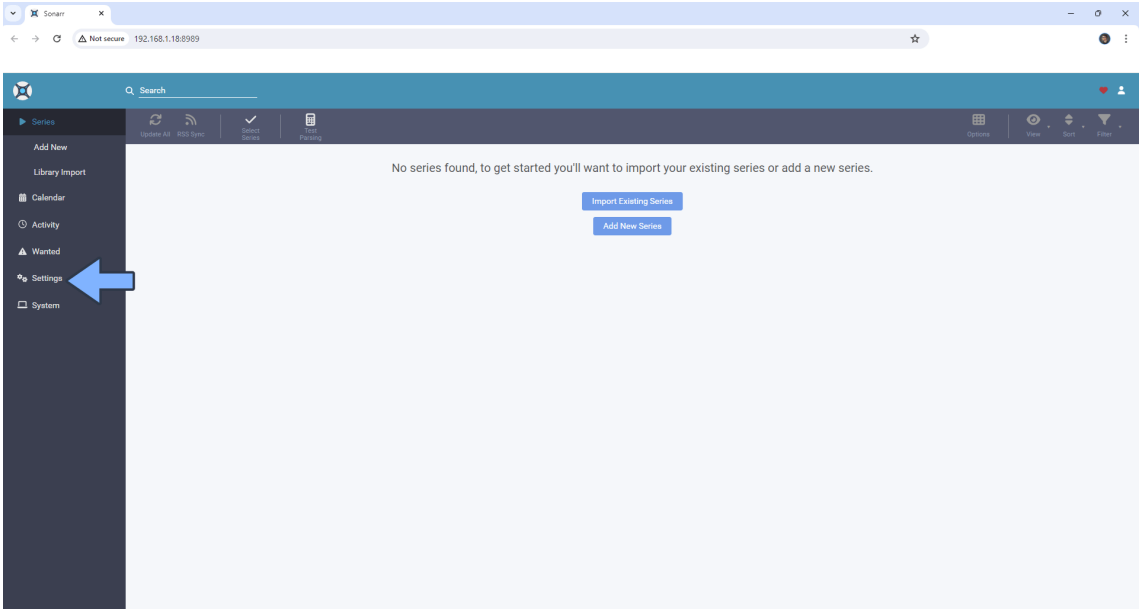
Sonarr

Adding Root Folder(s)

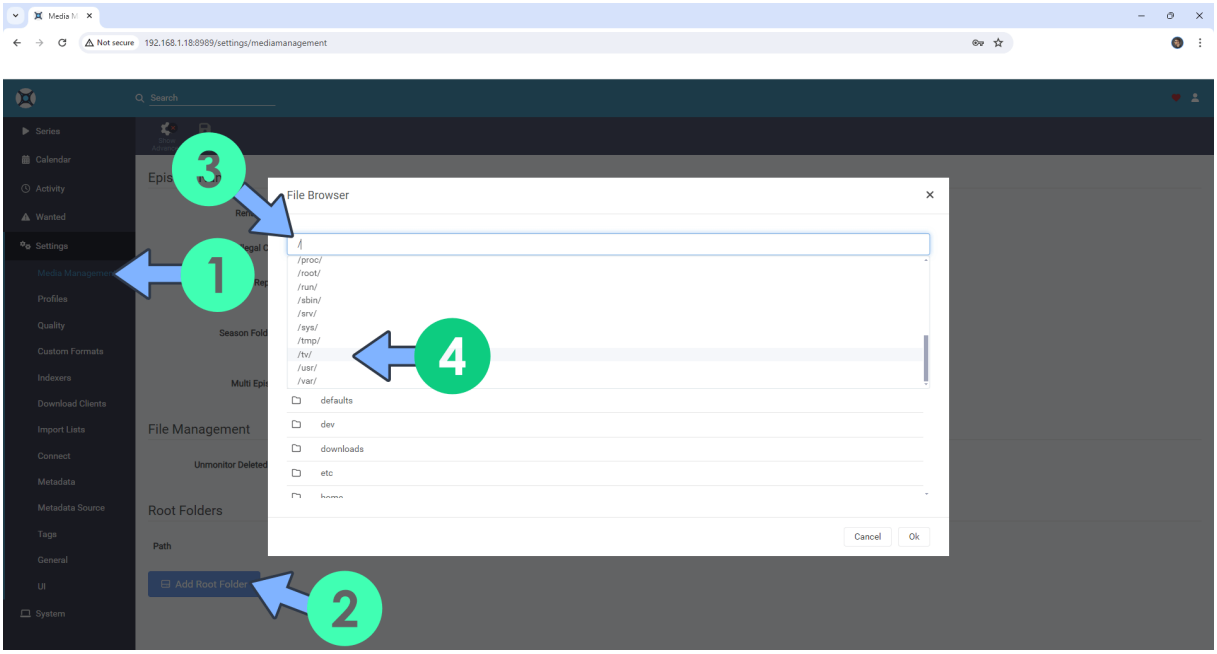
1. Open a web browser on your PC and type in the IP of your NAS followed by port 8989. So e.g. 192.168.0.2:8989.
2. Here you will be prompted to set up an authentication method. I recommend selecting “Forms” and having it enabled. Choose a username that is not admin or administrator, then select a password.



3. Go to settings

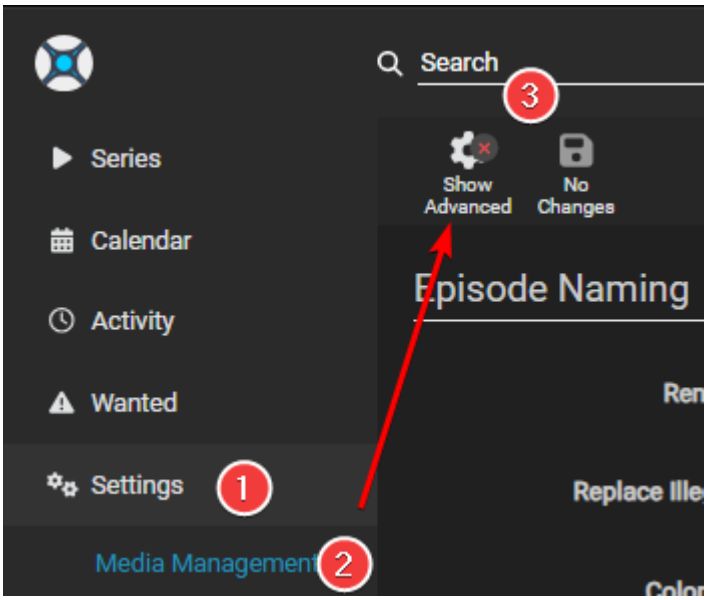


4. Select “Media Management”, scroll down and click on “Add Root Folder”. Type in “/Media/Media/TV Shows/” and select it before you click OK. If you have a separate folder for anime, add another root folder and search for “/Media/Media/Anime TV Shows” and select it. Click OK.

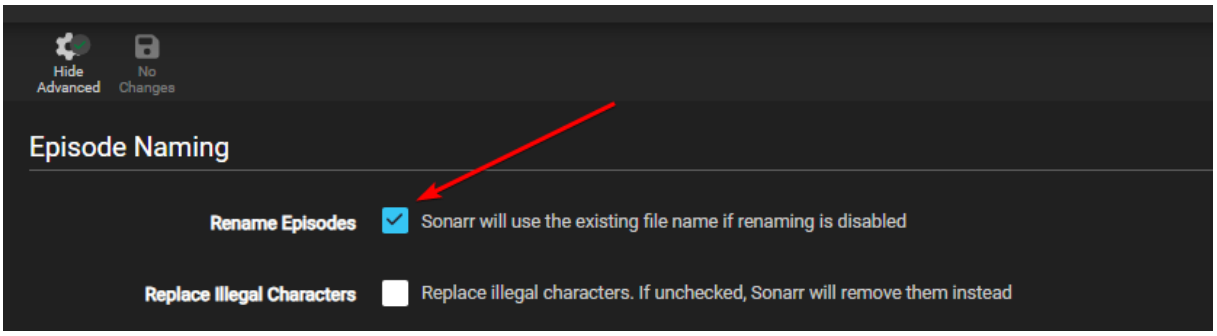


Changing Naming Scheme(s)

1. Go to settings > Media Management and click on Show Advanced.



2. Enable “Rename Episodes”. I recommend to also enable “Replace illegal characters”



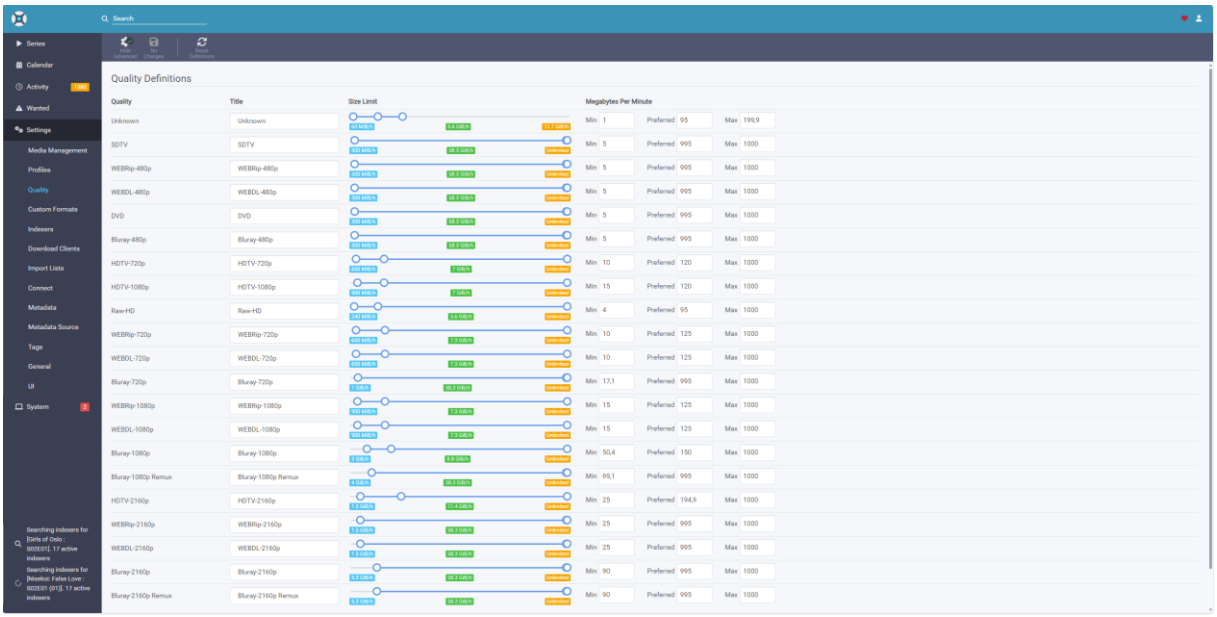
3. Go to TRaSH Guide’s and select the settings they recommend.

https://trash-guides.info/Sonarr/Sonarr-recommended-naming-scheme/#standard

4. Scroll down to “File Management” and select “Do not Prefer” for “Propers and Repacks”. This is important for our quality profiles later.
5. Under “Importing” make sure “Use Hardlinks instead of Copy” is enabled.

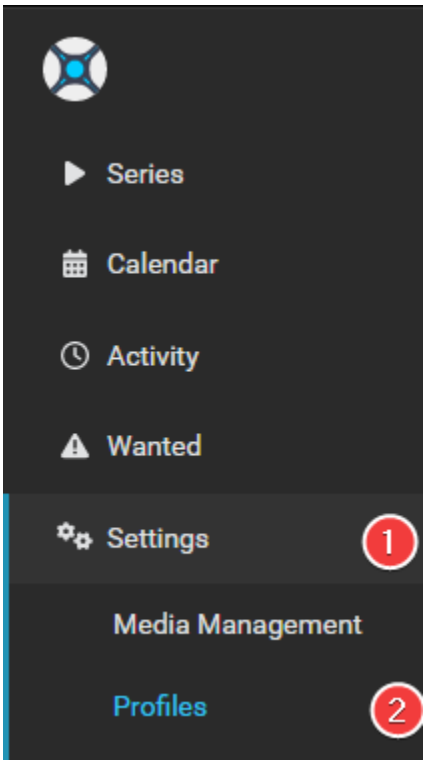
Quality Settings (File Size)

1. Go to settings and select “Quality”
2. Enabled “Show Advanced” by clicking the cog at the top.
3. Go to TRaSH Guides and edit the appropriate settings for the best quality. https://trash-guides.info/Sonarr/Sonarr-Quality-Settings-File-Size/



Quality Profiles

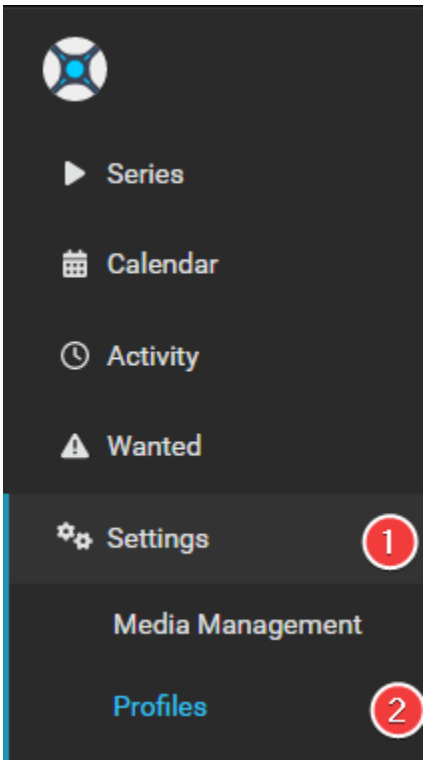
1. Go to settings and select “Profiles”



2. Delete all the default ones.
3. Go to TRaSH Guides and set up profiles as described there. Import the relevant custom formats that fit your needs and wants.
   https://trash-guides.info/Sonarr/sonarr-setup-quality-profiles/

Quality Profiles (Anime)

1. Go to settings and select “Profiles”

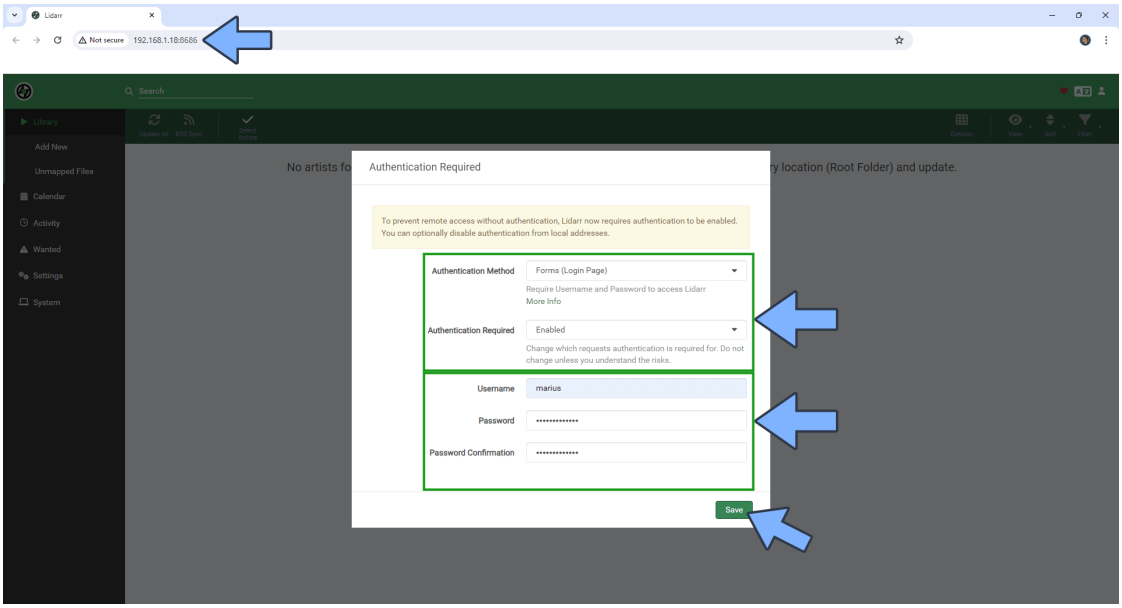


2. Delete all the default ones.
3. Go to TRaSH Guides and set up profiles as described there. Import the relevant custom formats that fits your needs and wants.
   https://trash-guides.info/Sonarr/sonarr-setup-quality-profiles-anime/

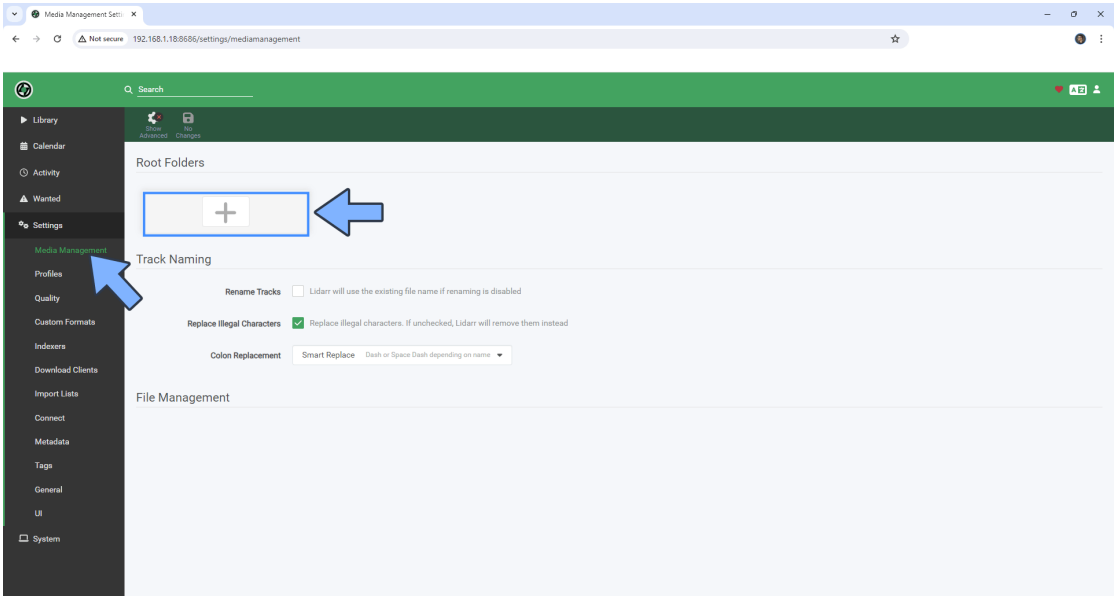
Lidarr

Adding Root Folder(s)

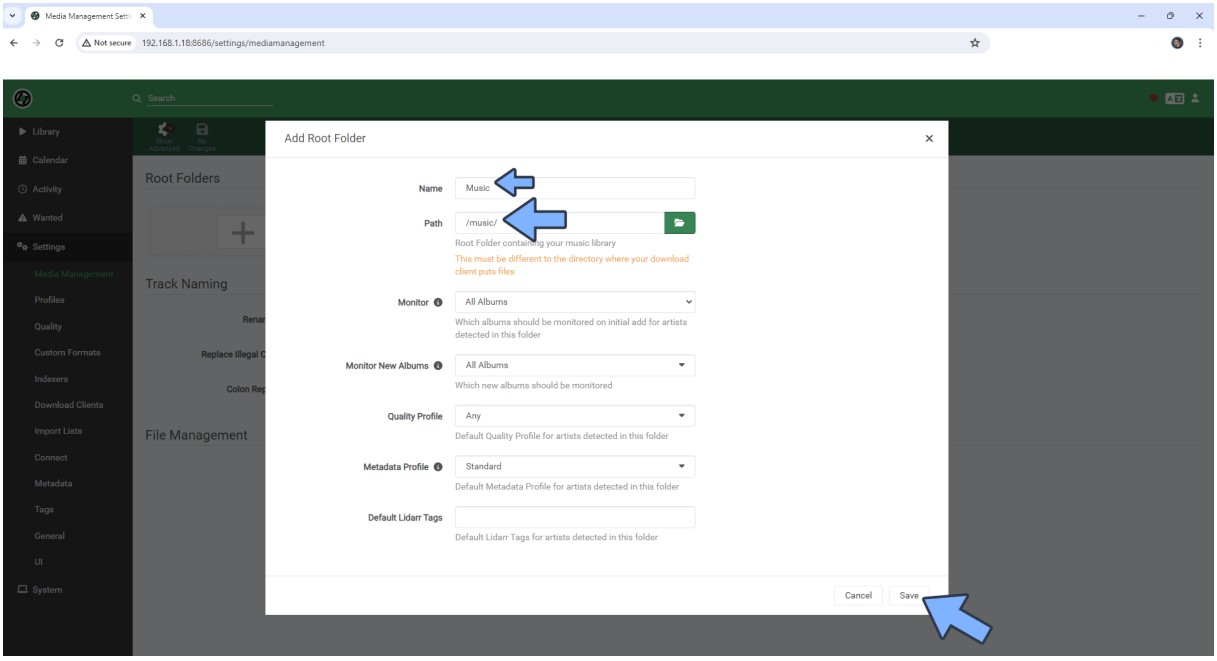
1. Open a web browser on your PC and type in the IP of your NAS followed by port 8686. So e.g. 192.168.0.2:8686.
2. Here you will be prompted to set up an authentication method. I recommend selecting “Forms” and having it enabled. Choose a username that is not admin or administrator, then select a password.



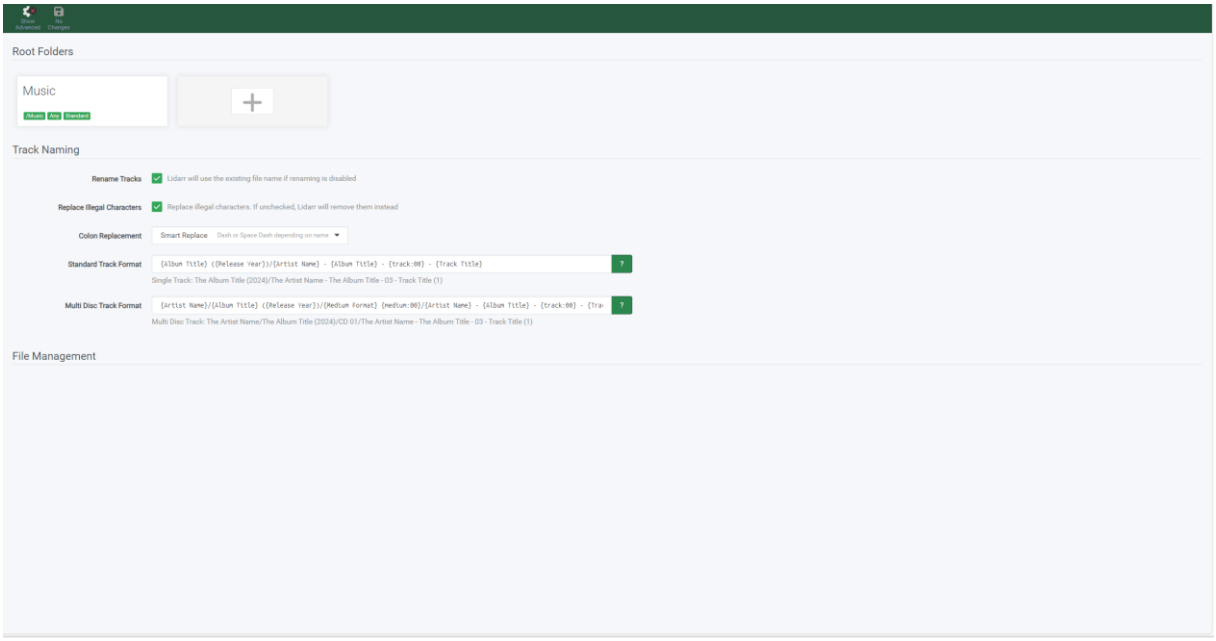
3. Go to settings, then select “Media Management”.
4. Click on the big “+” sign under “Add Root Folder



5. Search for “/Media/Media/Music” select it then click on OK



6. Make sure “Rename Tracks” and “Replace illegal characters” are enabled.



“Standard Track Format” should be:

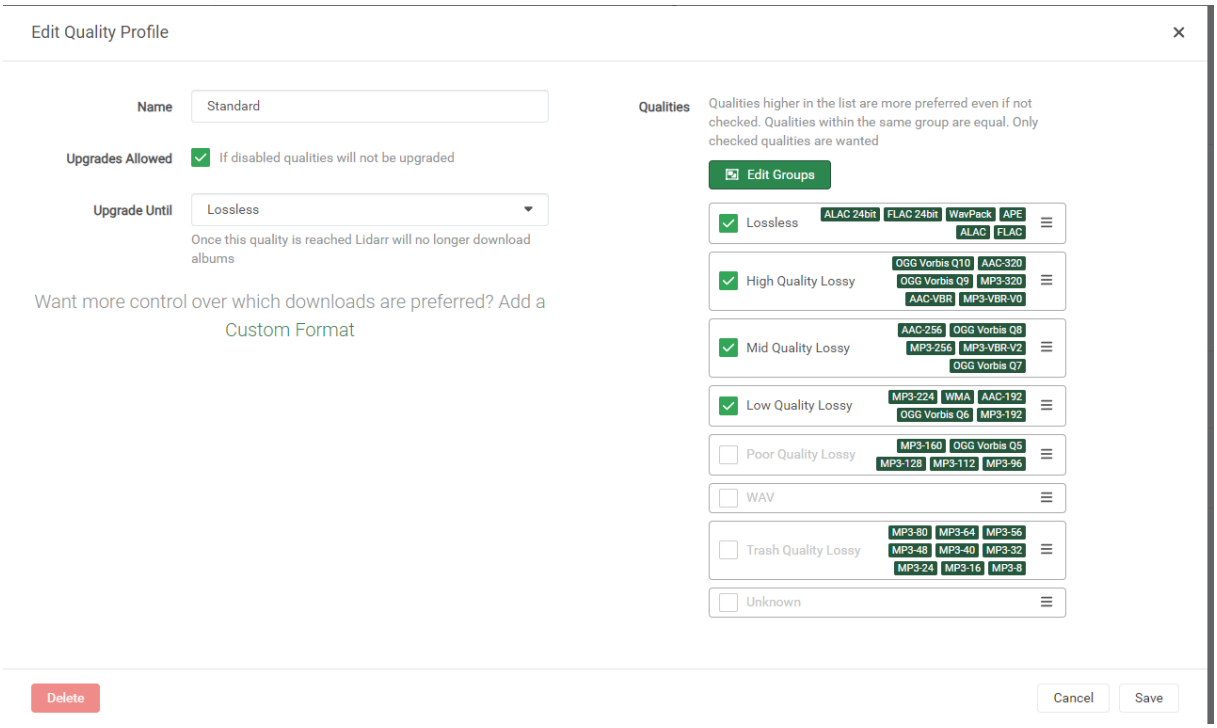
{Album Title} ({Release Year})/{Artist Name} - {Album Title} - {track:00} - {Track Title}

And “Multi Track Format” should be:

{Artist Name}/{Album Title} ({Release Year})/{Medium Format} {medium:00}/{Artist Name} - {Album Title} - {track:00} - {Track Title}

Quality Settings

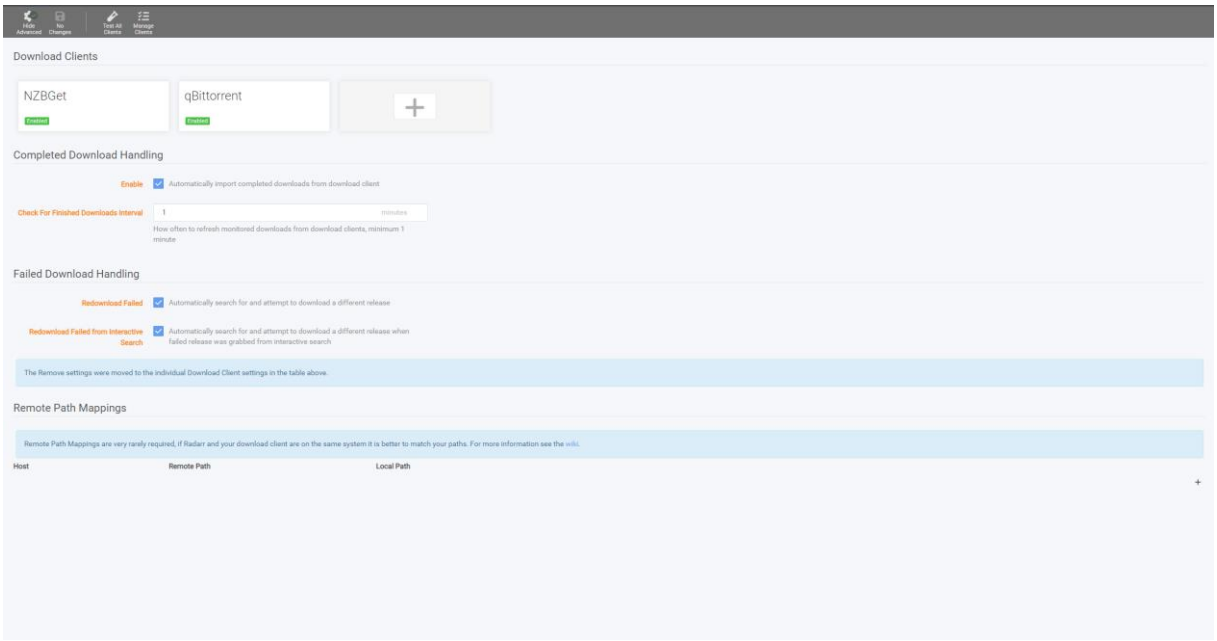
TRaSH Guides have no guides on lidarr. I have tried to search the internet but have not found any guides on quality settings or profiles for lidarr. I have only set up one profile myself on Lidarr, which is a pretty basic one without any scoring systemor custom formats. I have also not changed the size for the different Quality Settings.



Connecting download client to Radarr, Sonarr and Lidarr

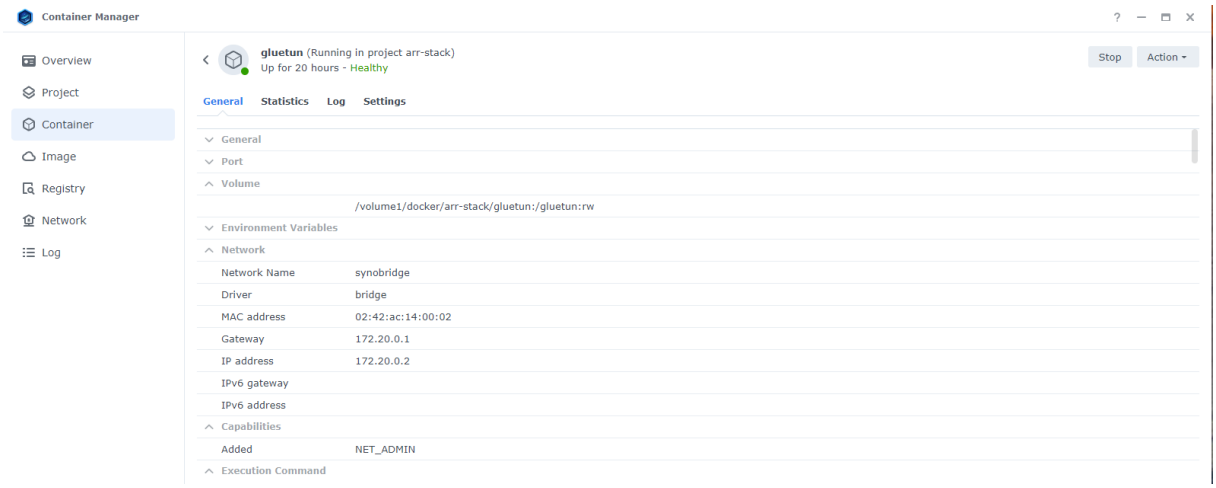
Now that we have configured all our main arr-apps, we need to actually give them something to send the downloads to. For this guide we are using qBitTorrent. The setup will be identical for all of the apps, so just repeat the steps for all of them.

1. Go to settings, then select “Download Clients”
2. Make sure “Automatically import completed downloads from download client” is enabled.
3. Click on the big “+” icon.



4. Select qBitTorrent
5. Fill out the details
    i. Name: qBitTorrent
    ii. Enable
    iii. Host: 170.20.0.2

NOTE: This must be the IP of the **GlueTUN** container, and **NOT** the NAS itself. So instead of 192.168.0.2, we put 172.20.0.2. To confirm this is the correct IP, we can go back to Synology container manager > container and click on gluetun. Scroll down until you find “Network Settings” and look for where is says “IP Address”



iv. Port: 8085

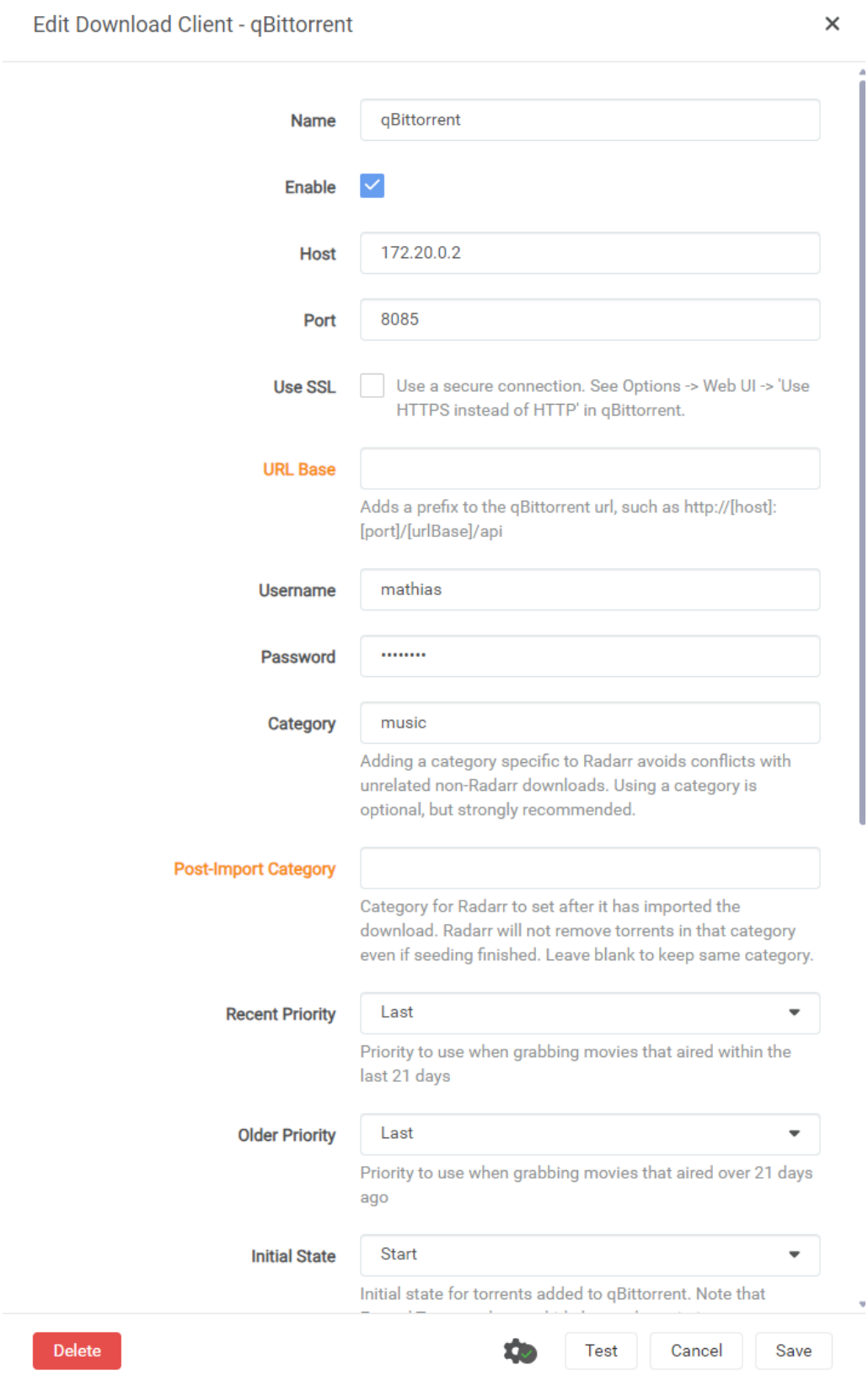
v. Username: <username on qBitTorrent>

vi. Password: <password on qBitTorrent>

vii. Category: <the appropriate category. music/movies/tv/anime>

NOTE: For Sonarr, you will need to setup 2 download clients if you want to separate anime to a separate folder than tv. Both can be the same client, with identical setup. Only thing that must change is the "Category" has to be "anime" for anime and "tv" or "series" for normal TV Series.

6. Click "Test". If it becomes green for a second, then press "Add" If it becomes red, review and double check your settings.

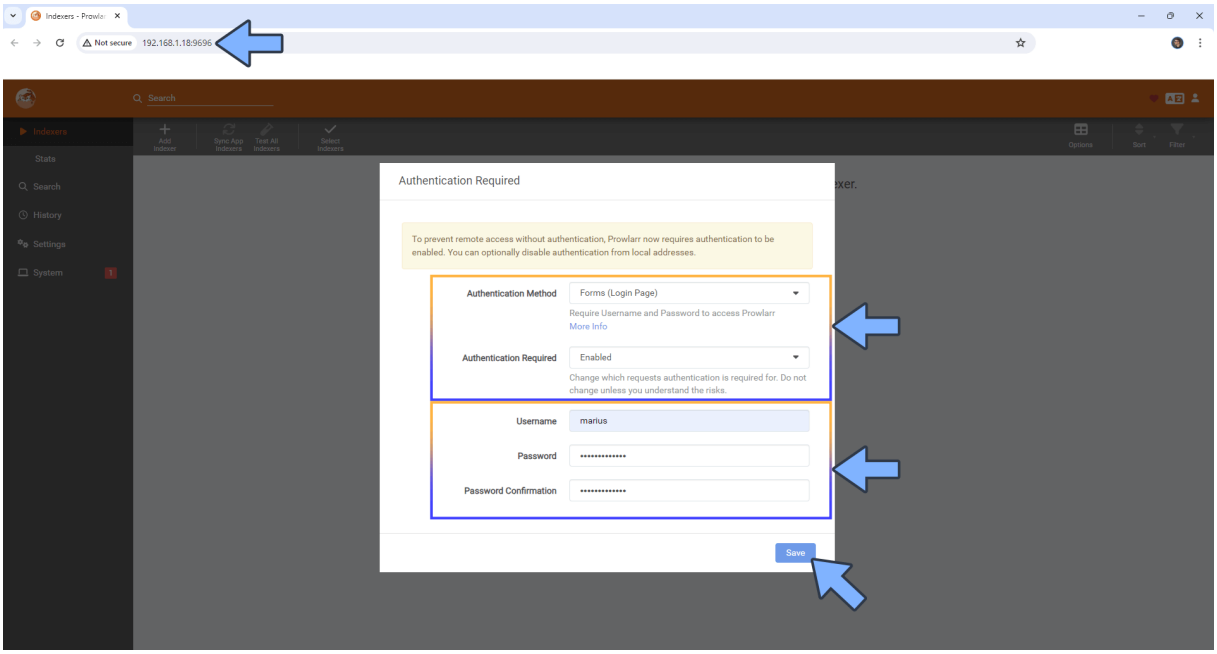

Edit Download Client - qBittorrent
Name
qBittorrent
Enable
Host
172.20.0.2
Port
8085
Use SSL
Use a secure connection. See Options -> Web UI -> 'Use HTTPS instead of HTTP' in qBittorrent.
URL Base
Adds a prefix to the qBittorrent url, such as http://[host]:[port]/[urlBase]/api
Username
mathias
Password
Category
music
Adding a category specific to Radarr avoids conflicts with unrelated non-Radarr downloads. Using a category is optional, but strongly recommended.
Post-Import Category
Category for Radarr to set after it has imported the download. Radarr will not remove torrents in that category even if seeding finished. Leave blank to keep same category.
Recent Priority
Last
Priority to use when grabbing movies that aired within the last 21 days
Older Priority
Last
Priority to use when grabbing movies that aired over 21 days ago
Initial State
Start
Initial state for torrents added to qBittorrent. Note that
Delete
Test
Cancel
Save

Prowlarr

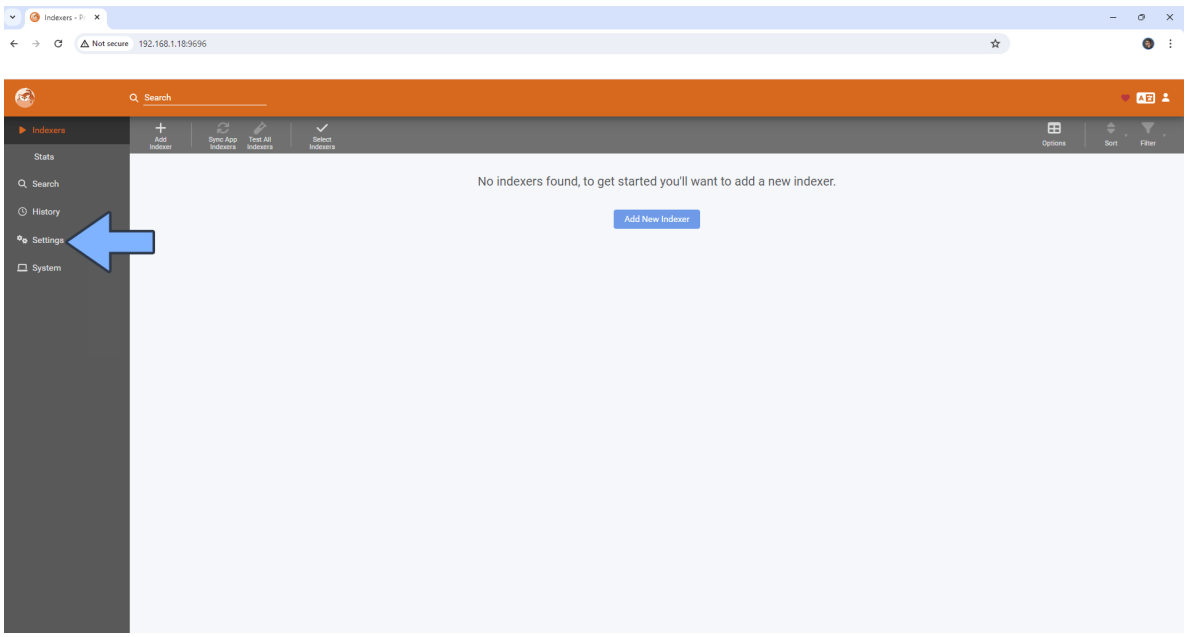
Adding the apps to Prowlarr

Now that we have set up all our downloaders, we need something to search for the files we are going to download. For that, we will use Prowlarr for best control.

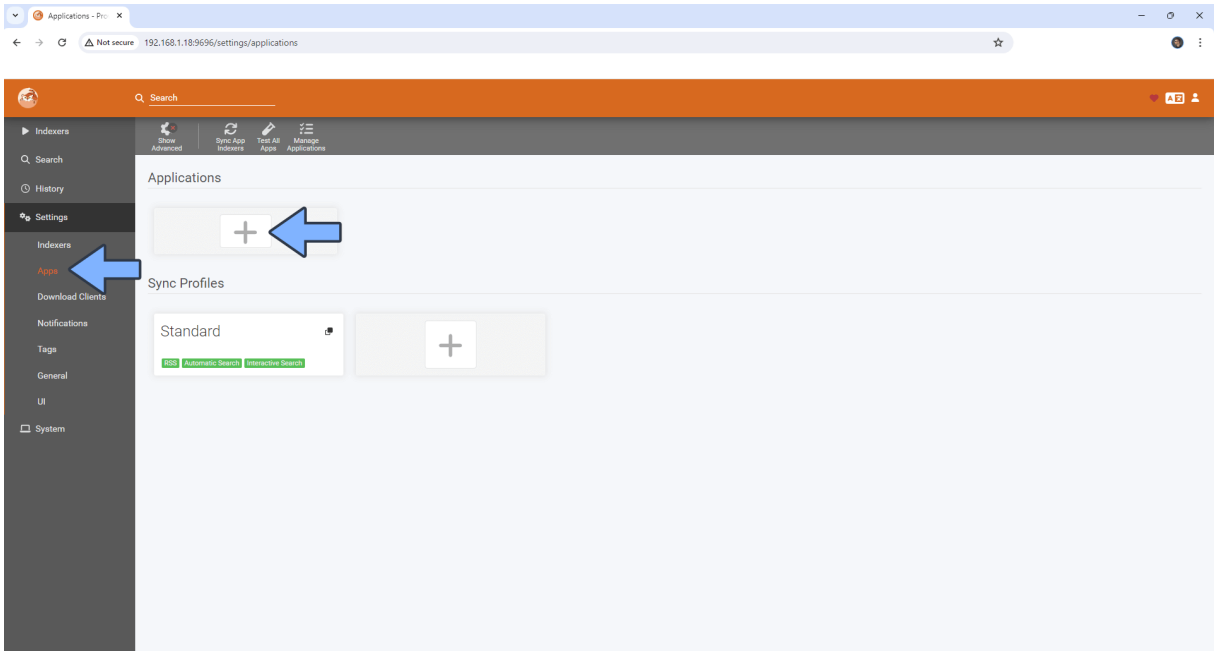
1. Go to <NAS-IP>:9696 e.g. 192.168.0.2:9696
2. Fill out the form just like earlier.



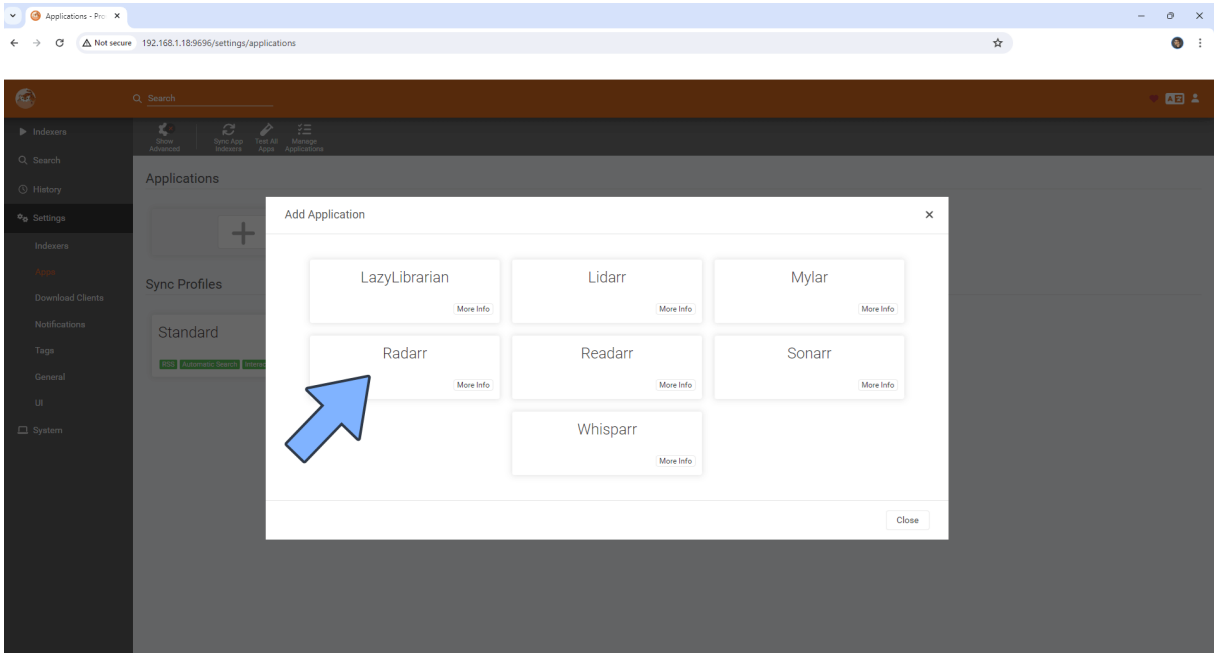
3. Go to settings on the left-hand side menu.



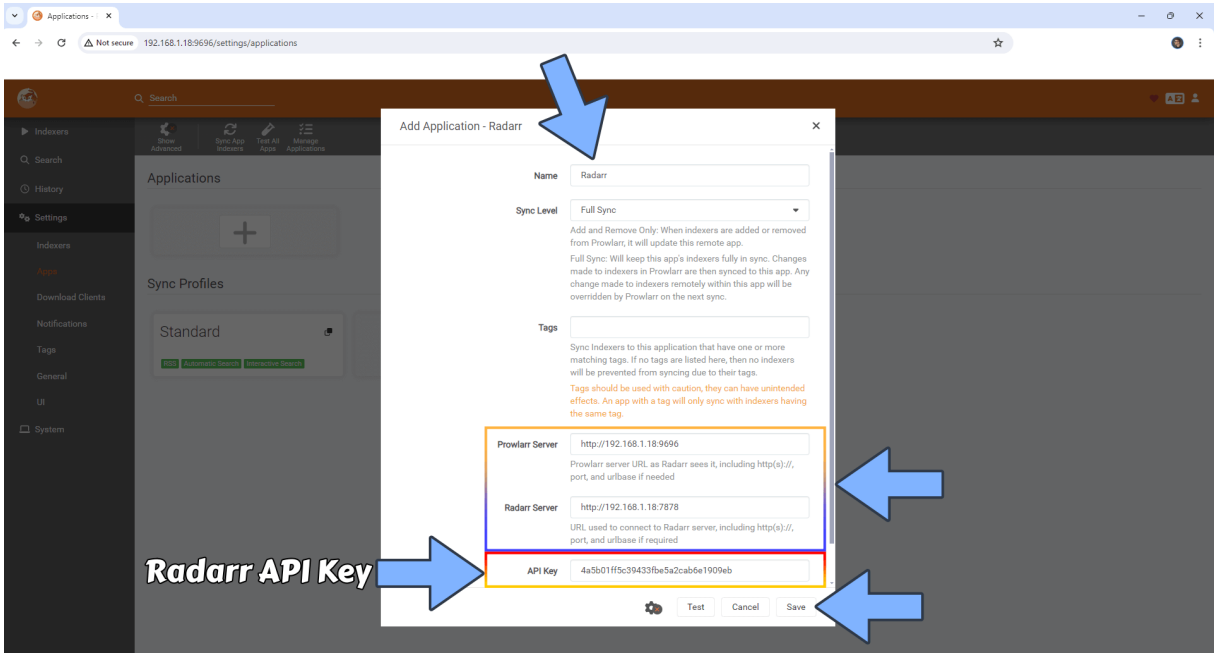
4. Click on apps, then the big “+” under applications



5. Click on the app you want. We can start with Radarr.

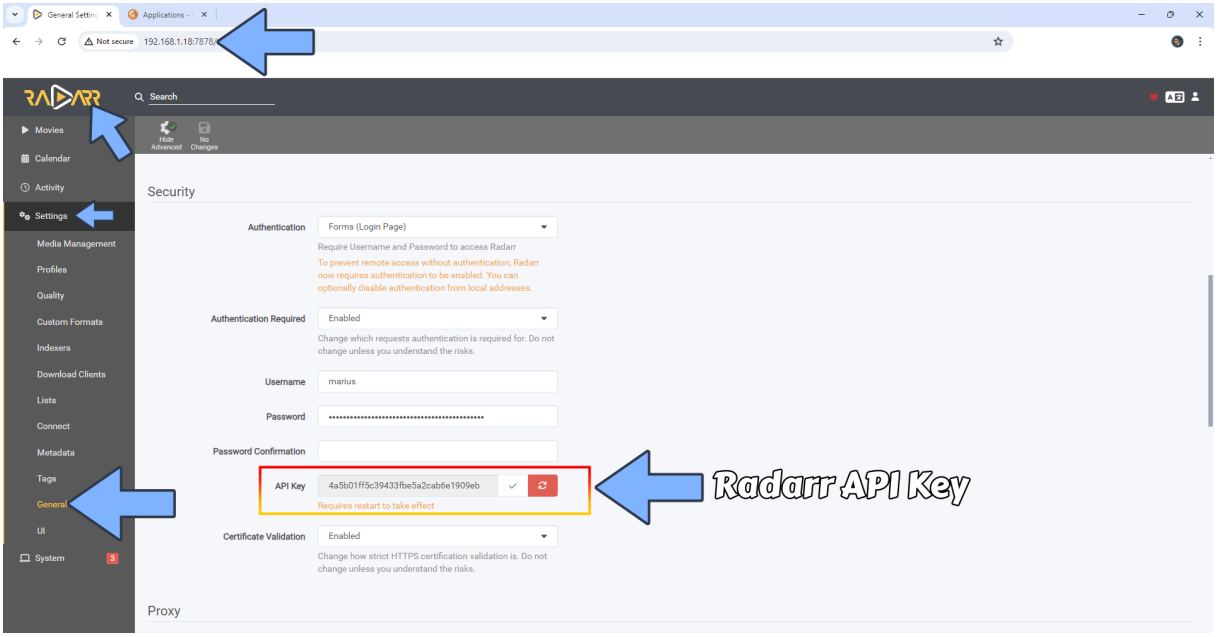


6. Fill out the form:
   - Name can be default.
   - Sync Level: Full Sync
   - Tags: Leave blank
   - Prowlarr server: http://172.20.0.2:9696
   - Radarr server: http://172.20.0.2:7878
     NOTE: If you chose another IP for your “synobridge” network, then put that in instead.
   - API Key: Get this from the relevant app. In this case, Radarr. Scroll down to find out how.
7. Repeat this step for all the apps you want; LazyLibrarian, Lidarr, Mylar, Radarr, Readarr, Sonarr and Whisparr.



How do I get the API key?

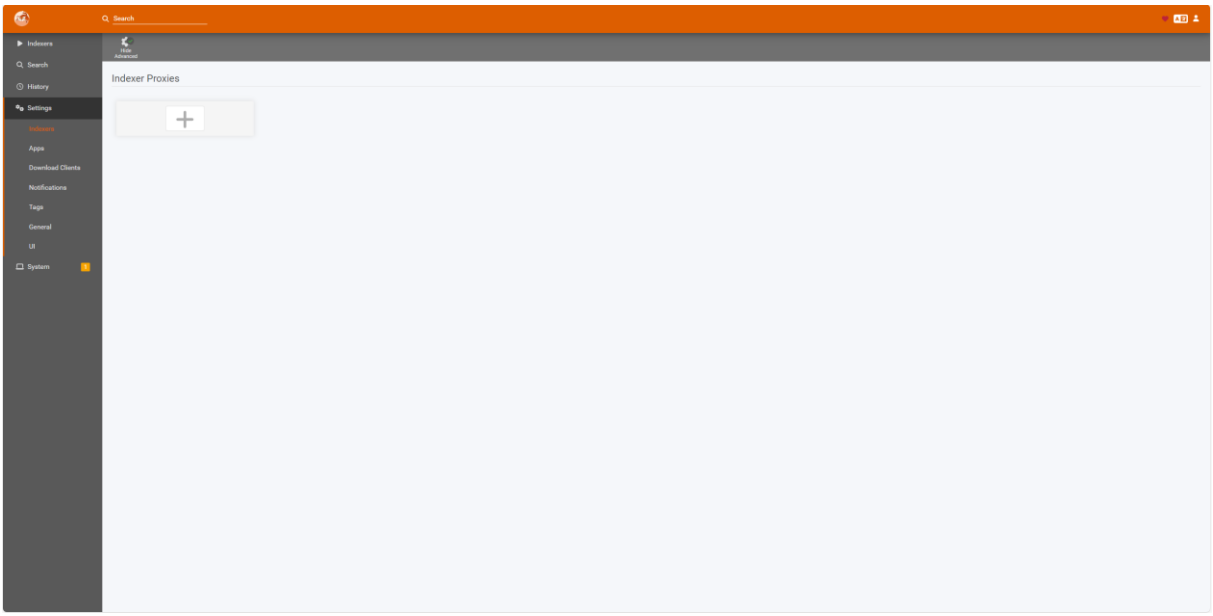
1. Open your relevant application in the browser.
2. Go to settings, then click on “General”
3. The API Key should be right there. Just click the copy button on the right, and paste that into Prowlarr.



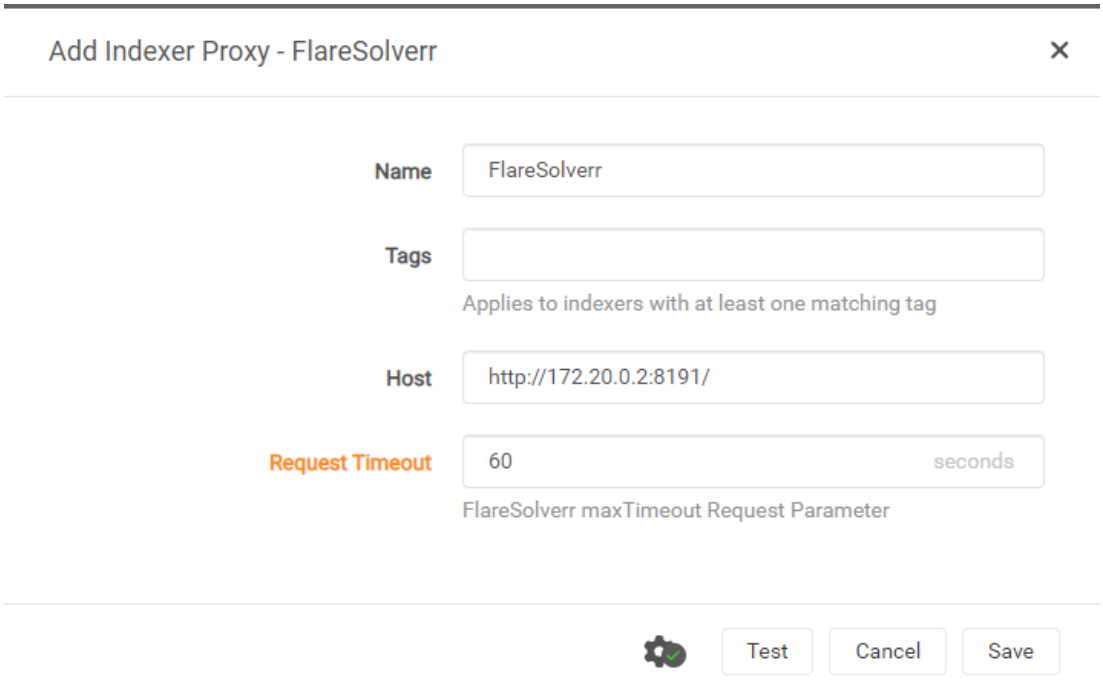
Connecting Flaresolverr

Some indexers, like 1337x which is one of the best free ones, require you to solve a Cloudflare verification before you get access. To make prowlarr do this, you must use flaresolverr. To do this:

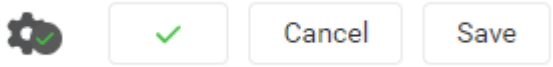
1. Go to settings, then click on indexers
2. Click the big + icon.



3. Edit the “Host” field to be “http://172.20.0.2:8191”



4. Click on “Test” and if it becomes green for a second it works, and you can click “Save”. If it becomes red, then double check your Host IP. It should be the same as your “synobridge” network in docker.

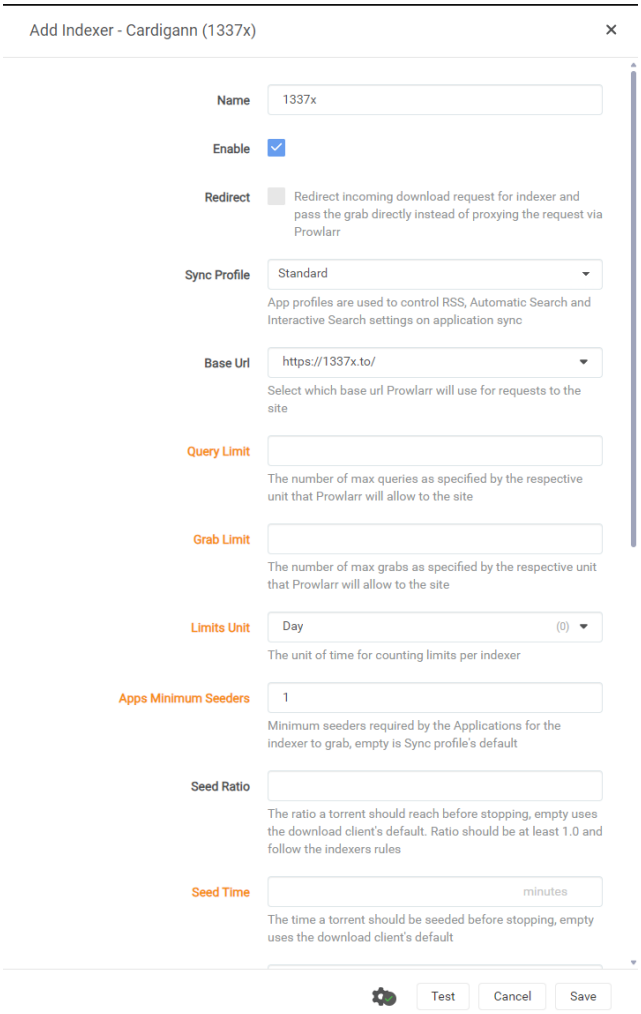


Adding indexers

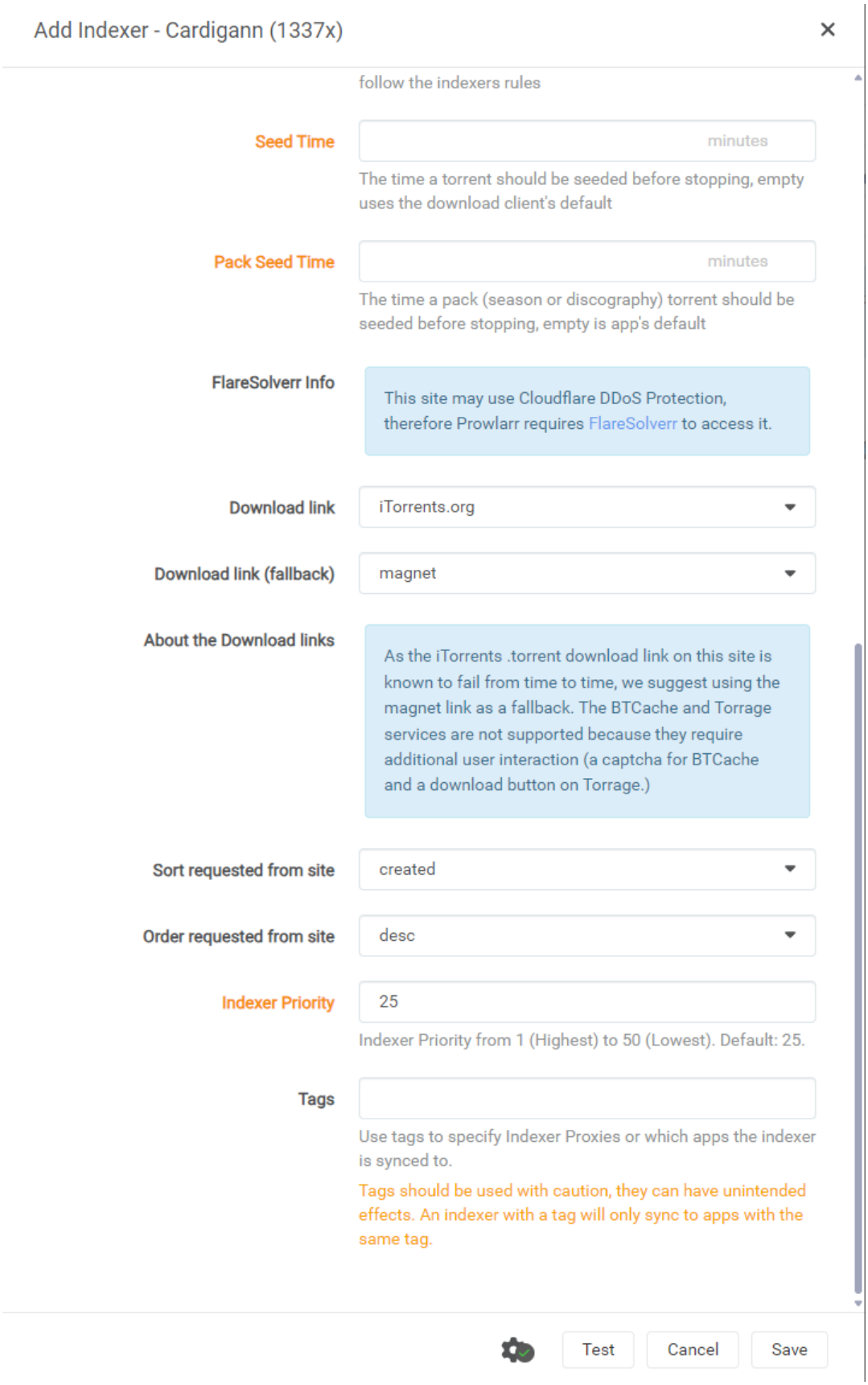
Indexers are the websites prowlarr search for files. Sadly, a lot of the good ones are private. This means that to use them you will most likely need to pay to get access or get invited. Therefore, I will focus on the free ones.

1. Click on Indexers, then “Add Indexer”

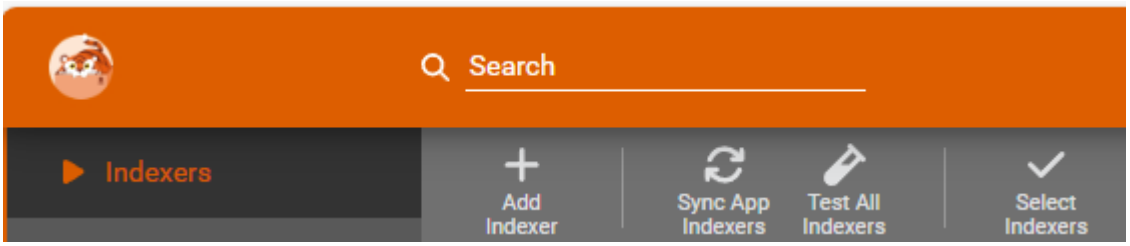
2. Here can filter by protocol, Language, Privacy and Category. All indexers that use the “nzb” protocol are for UseNet and most of these are private. But this is for torrents, and we will therefore ignore them for now. You could just set the privacy to public to get a list of all the free ones. The ones I would recommend getting at least are 1337x and TheRARBG for general TV and movies, and Nyaa for anime.
3. We can now search for our desired indexers, then click on the result
4. Fill out the settings. Base URL should be 1337x.to and I would also recommend to tur non advanced settings, then choose “minimum seeders: 1”.



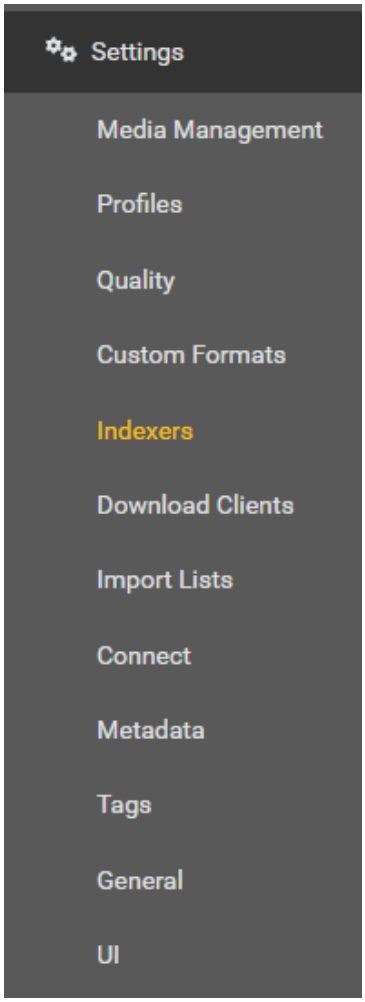
If we scroll down, we can see that it says it requires flaresolverr. Now we can click “Test” to see if it works, and if it does, we can click “Save”. Do this for all your wanted indexers.

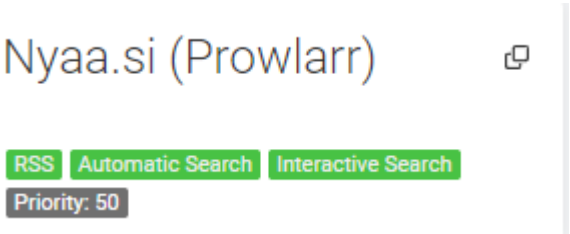


5. After you have added all your desired indexers, click on “Sync App Indexers” to push them to all your apps.



6. Let’s wait 2-5 minutes for it to finish up, then we can head into our apps, like Radarr or Sonarr, and click on settings > indexers to see our selected indexers with a (prowlarr) at the end to indicate where it comes from.



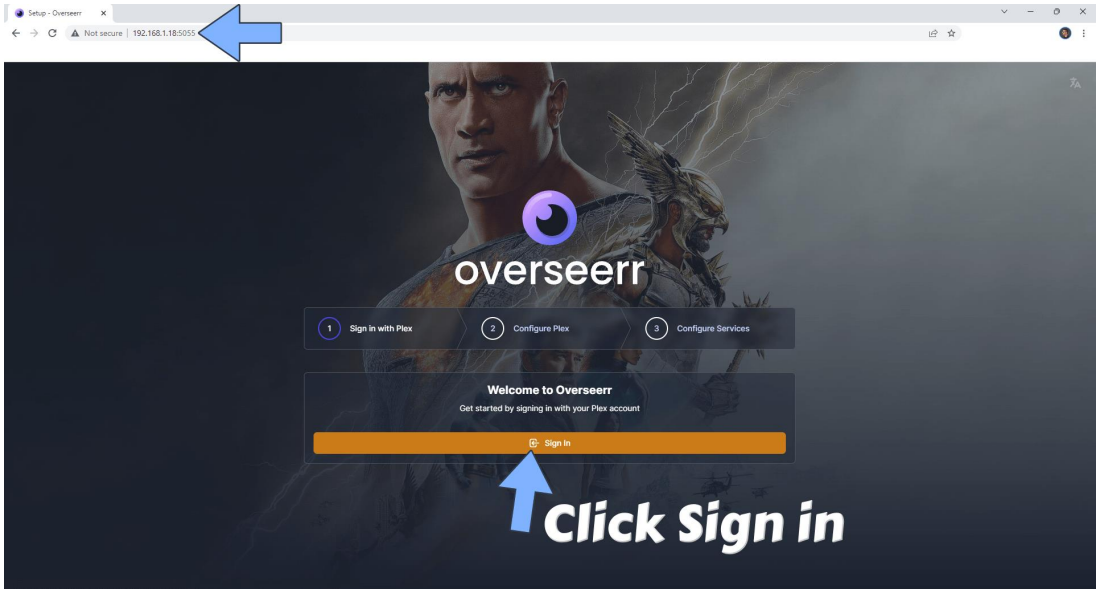


Overseerr

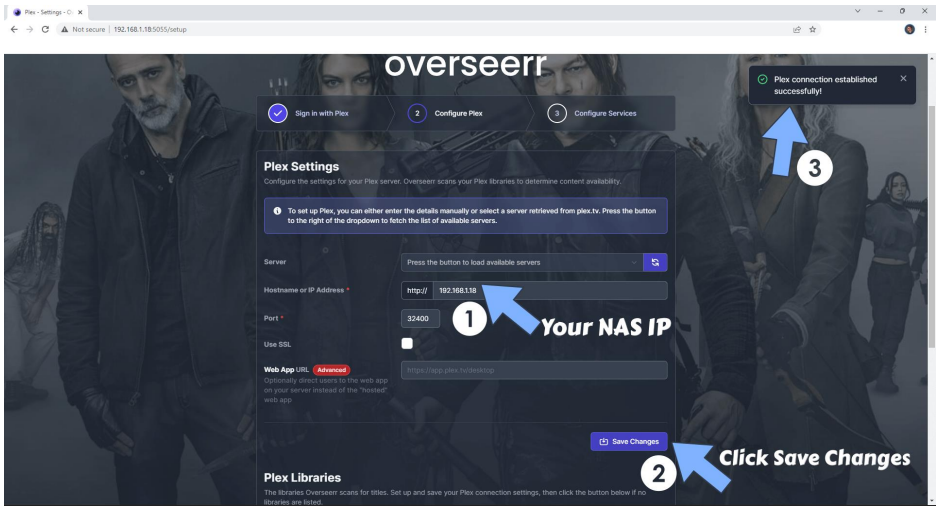
Now it's time to take the automation to the next level. With overseer, we get a huge catalogue of movies and shows at our fingertips and can download them with just one click.

Configuring Overseerr

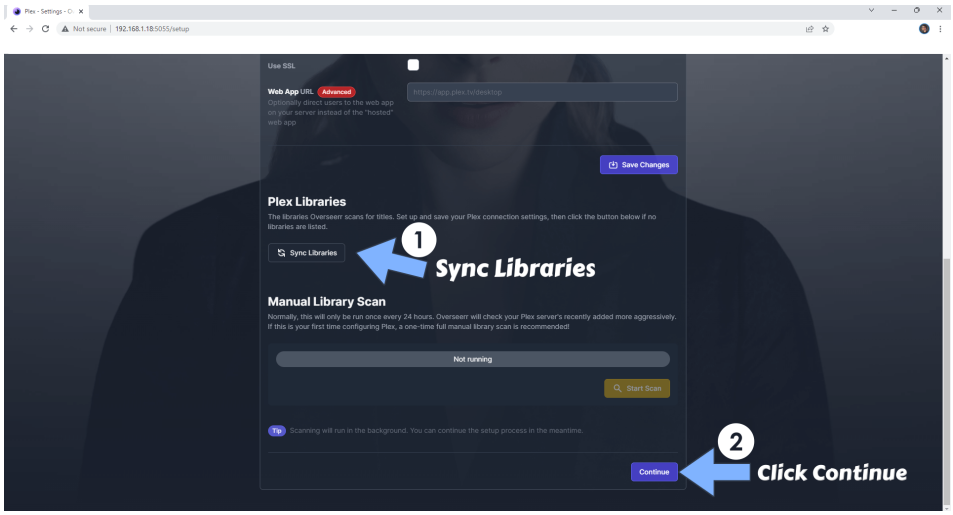
1. Go to <synolog-ip>:5055.
2. Click "Sign In", and follow the sign in instructions



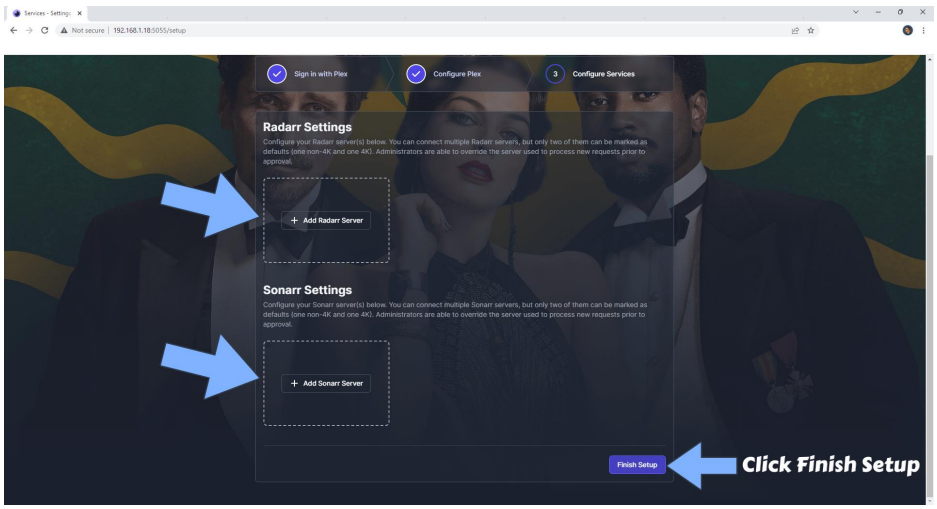
3. On the setup screen, **DON'T** use your synobridge but your **NAS** IP (same one yo use to access this website). So, for me it's 192.168.0.2. If you haven't changed the plex port, it can stay as default. Click "Save Changes". You should now see a green checkmark at the top.



4. Scroll down a bit and click “Sync Libraries”. Afterwards you can click continue. If it’s not done syncing, it will continue in the background.

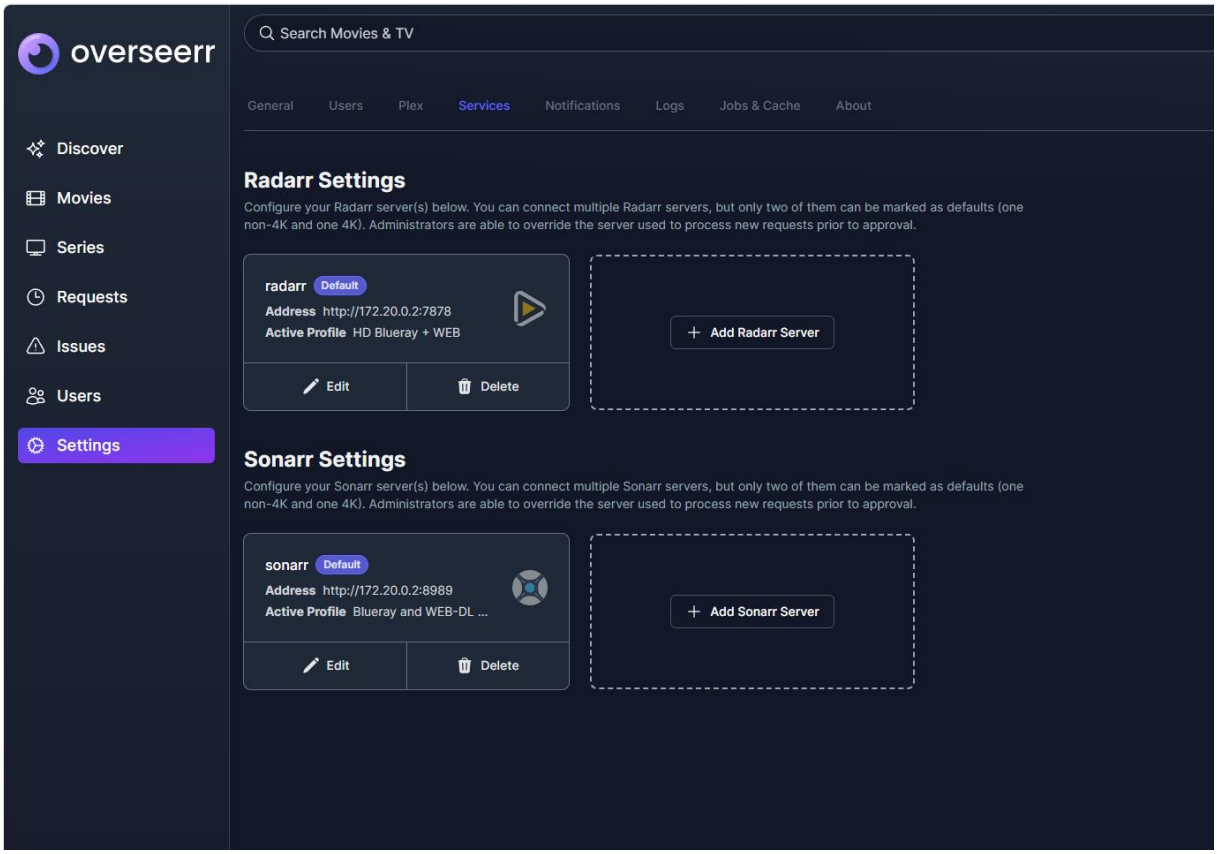


5. Add your Radarr and Sonarr by clicking the buttons as shown in the image below. Scroll down to continue.



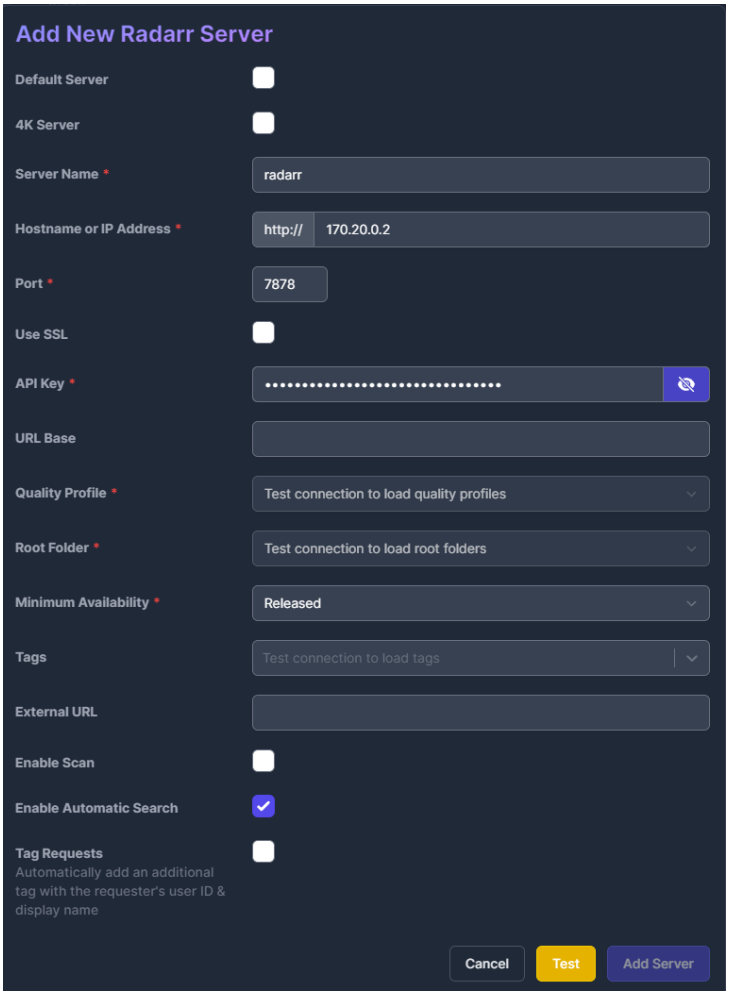
Adding Radarr and Sonarr

On the first-time setup, you will get a prompt to add Radarr and Sonarr. If you manage to miss it, freight not as you can do it in settings later. Just head to settings > services then click on add.

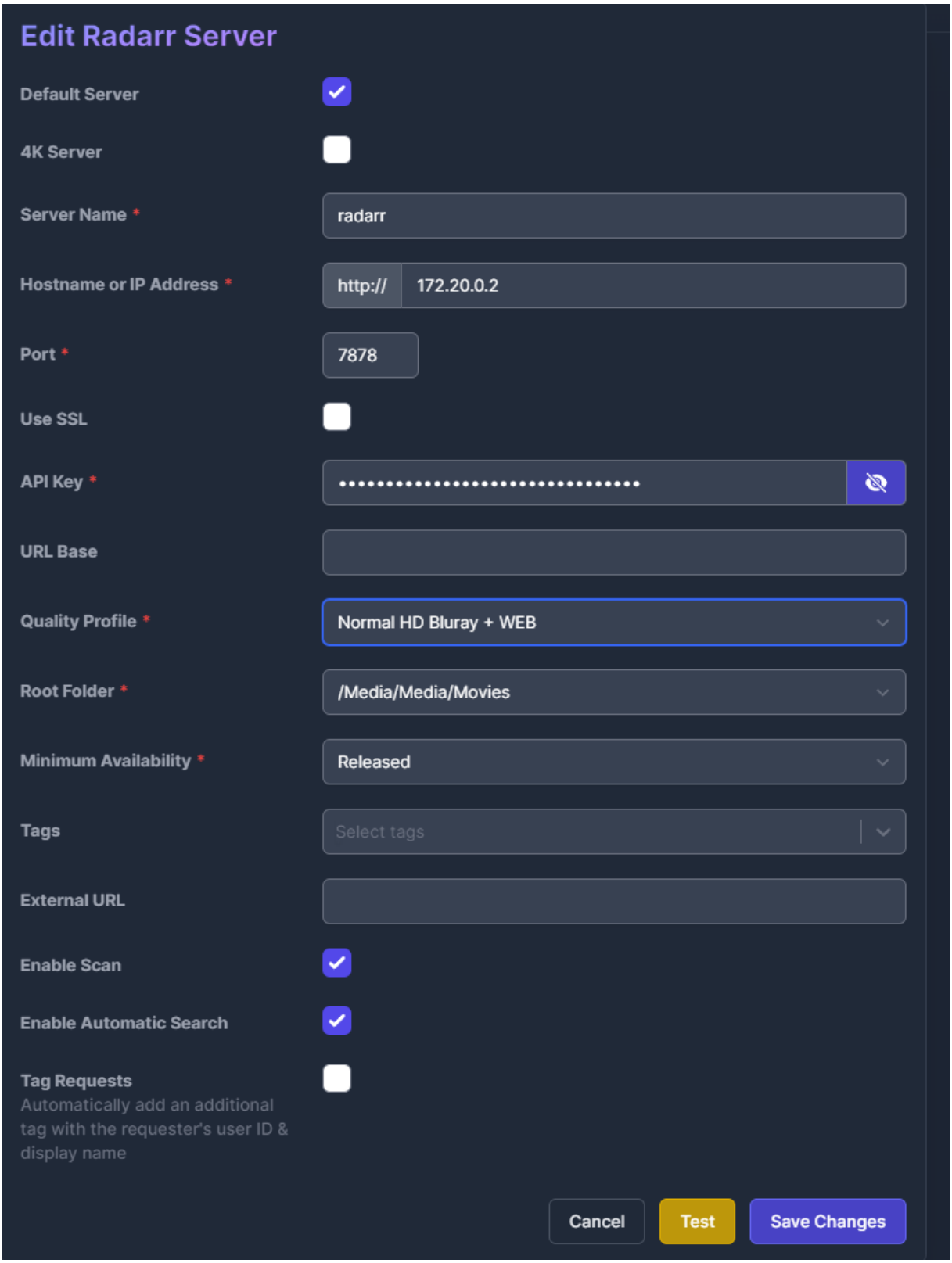


Radarr

1. When you click on “Add Radarr Server” you should see something like this:



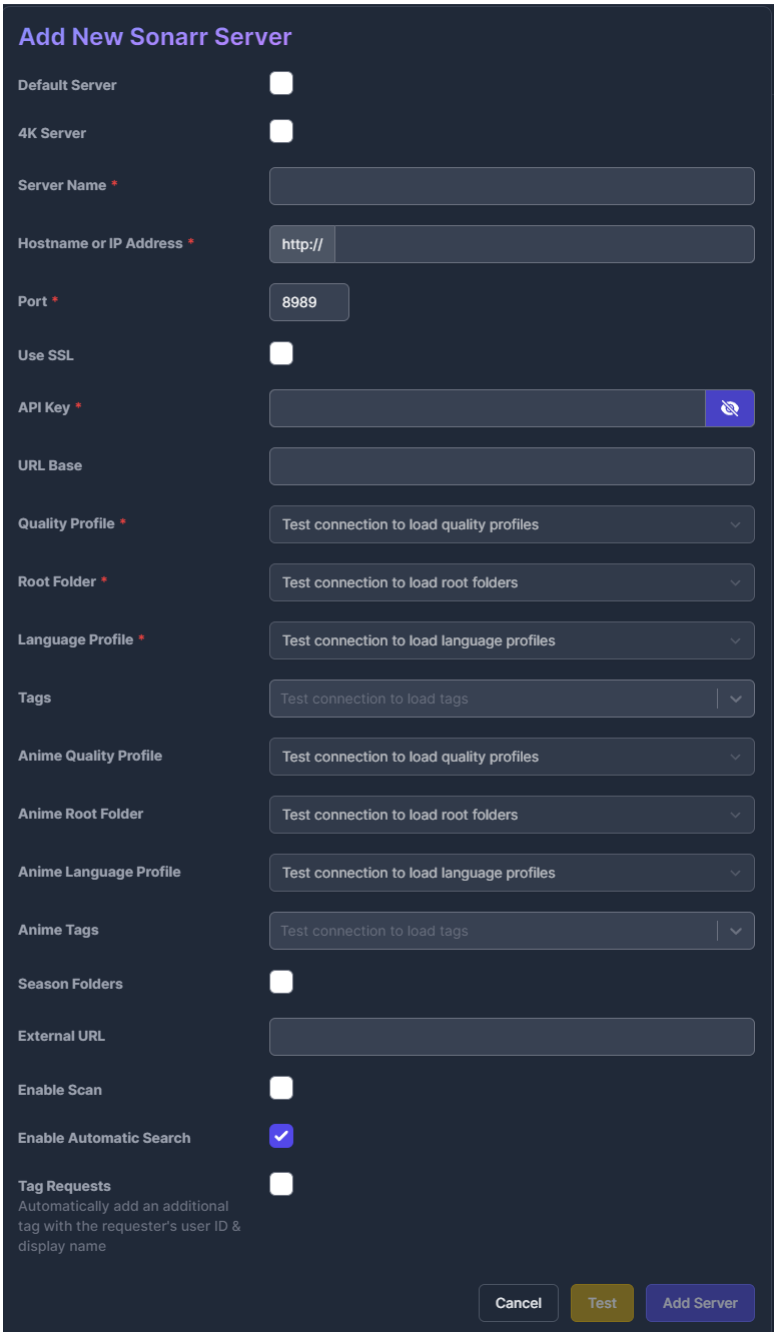
2. Fill out server name, hostname, port and API Key as follows:
   - Server Name radar (or anything you like)
   - Hostname: 170.20.0.2 (your synobridge IP)
   - Port: Default. 7878 for Radar
   - API Key: Your relevant API Key. Don’t know how to get it? Check the table of contents.
3. Now click on “Test”. If it’s successful, you will now get to select the rest of the settings.
4. Now configure the rest of the settings:
   - Quality Profile: Your desired default quality profile. This can be changed every time you request in Overseer, but not in Requestrr
   - Root folder: “/Media/Media/Movies”
   - Minimum availability: Whatever you prefer. I want it to be released.
   - Tick the options for “Enable scan” and “Enable Automatic Search”



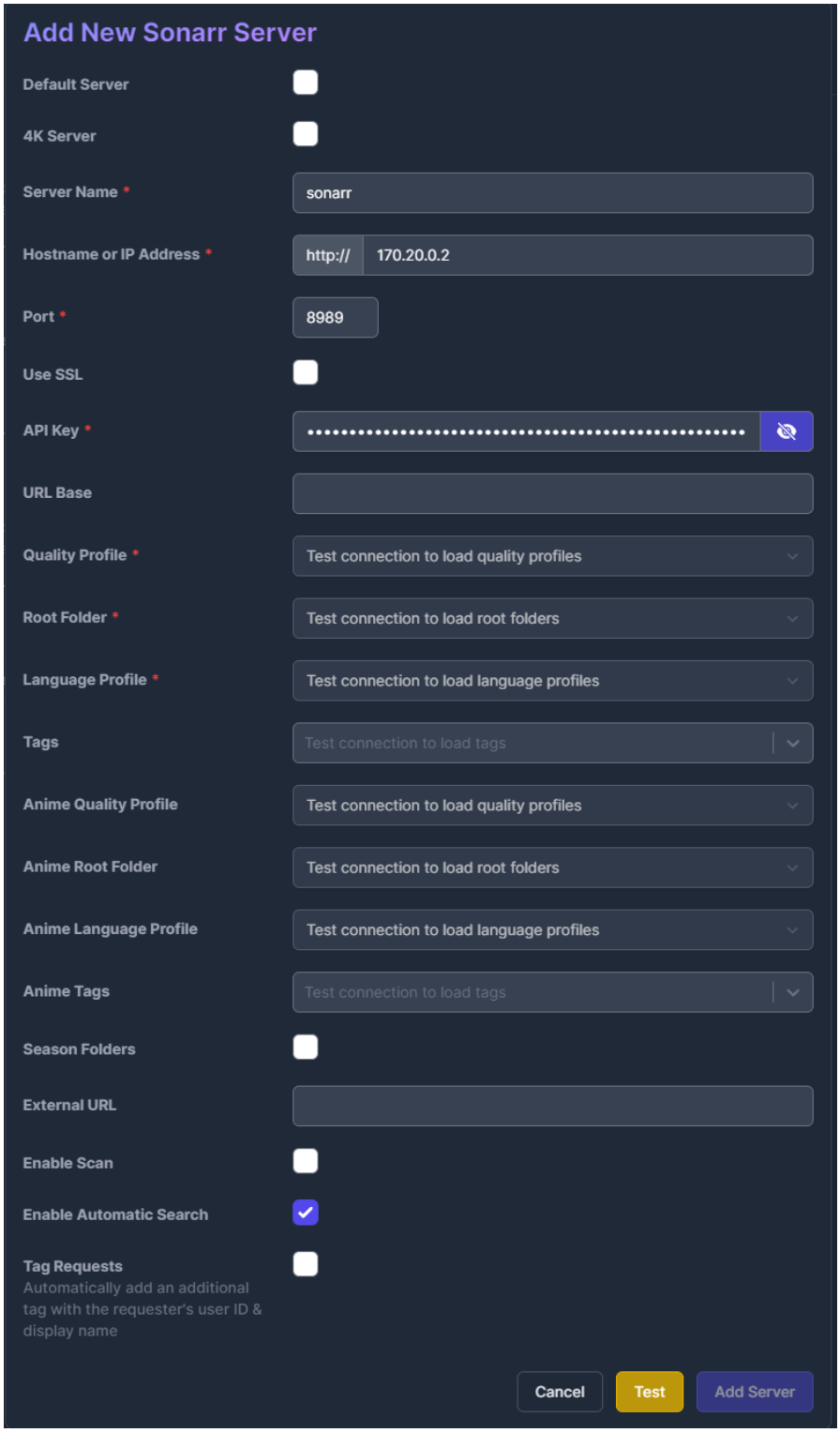
5. Now we can save our changes.

Sonarr

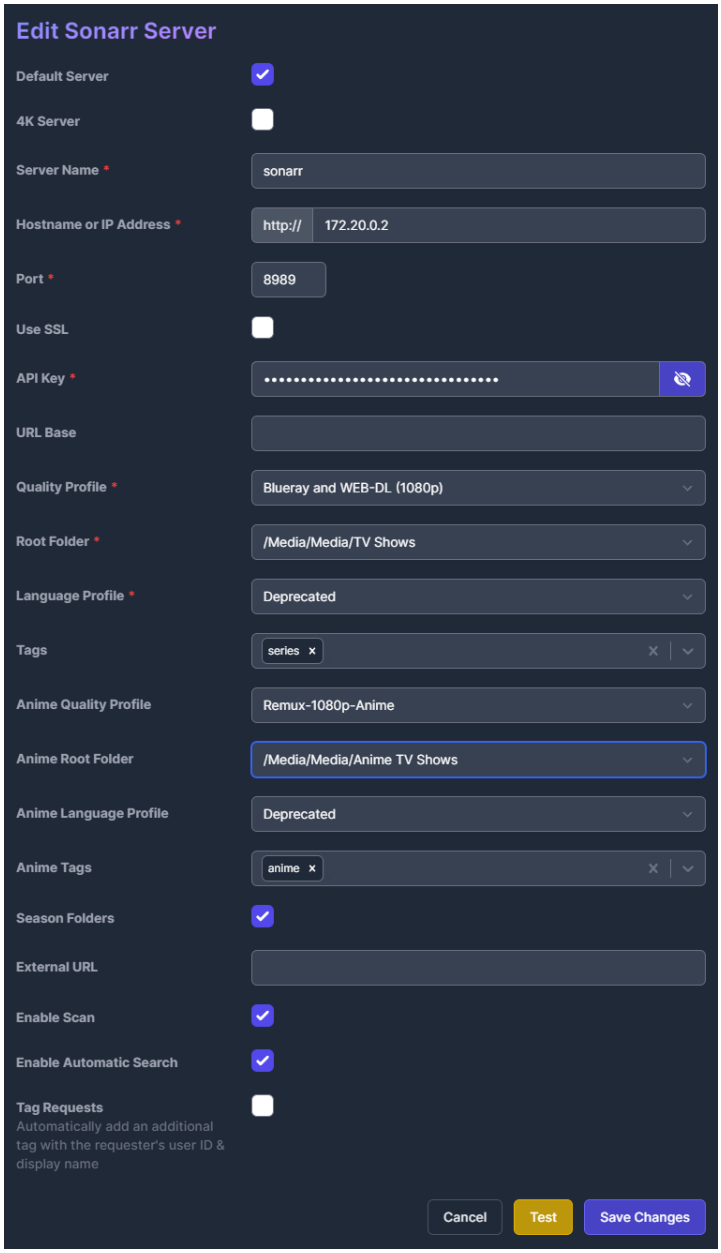
1. After you click “Add Sonarr”, you should see something like this:



2. Fill out the settings like this:
   - Servername: sonarr (or anything you like)
   - Hostname: 170.20.0.2 (Your synobridge IP)
   - Port: Default. 8989 for Sonarr
   - API Key: Your API Key. Don't know where to get it? Check the table of contents.
3. Now we can click "Test" to test out connection. If it's successful, we can now move on to edit some more settings.



4. Now make these changes to the settings:
    - Quality Profile: Your desired default quality profile. This can be changed every time you request in Overseer, but not in Requestrr
    - Root Folder: “/Media/Media/TV Shows”
    - Language Profile: Depricated
    - Tags: series
    - Anime Quality profile: Your desired default quality profile for anime shows. This can be changed every time you request in Overseer, but not in Requestrr
    - Anime Root Folder: “/Media/Media/Anime TV Shows”
    - Language Profile: Depricated
    - Anime Tags: anime
    - Be sure to tick the boxes for “Enable Scan” and “Enable Automatic Search”
5. Click “Save”

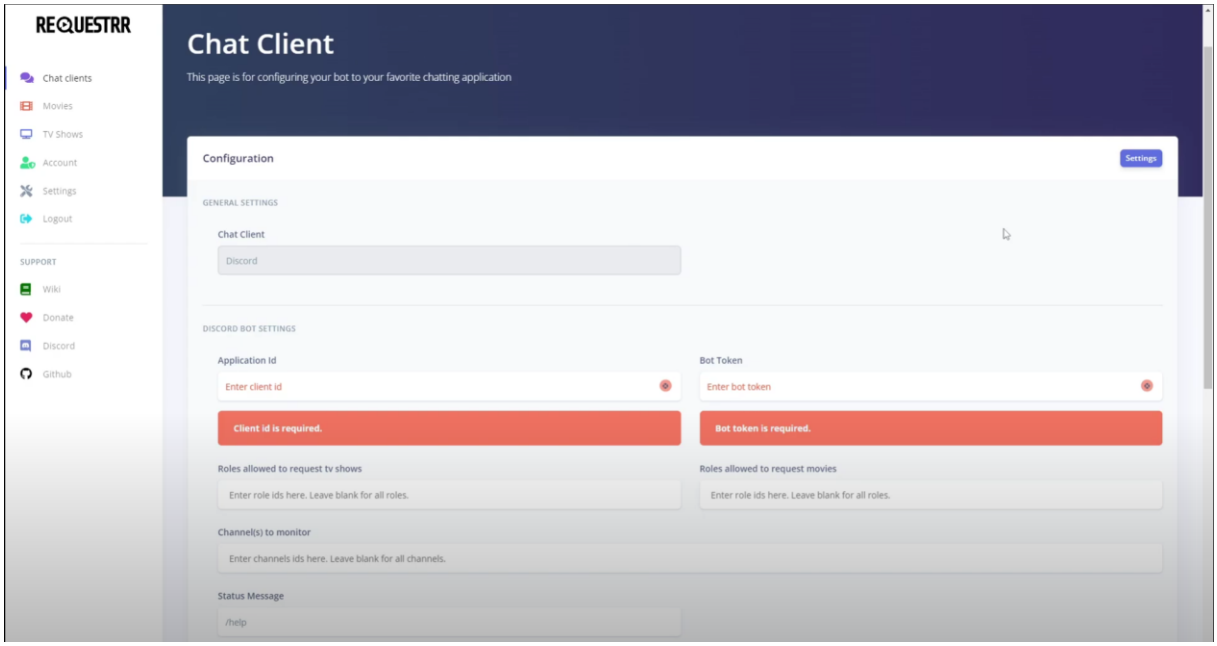


Requestrr

Now we can configure Requestrr. It allows us, and our users, to request movies and TV shows through chat in discord. This is a good alternative if you don't know how or don't want to port forward your Overseer. Or if you plan to share your Plex with strangers and don't want them to know your public IP-address

Configuring Requestrr

1. Open Requestrr in a web browser by typing <NAS-IP>:4045
2. The first time you log in, you might get a pop-up to create an authentication method just like all the other apps. Fill out the form like earlier then.
3. When we open up Requestrr, we should see our Chat clients. If not, navigate to it on the left-hand side menu.

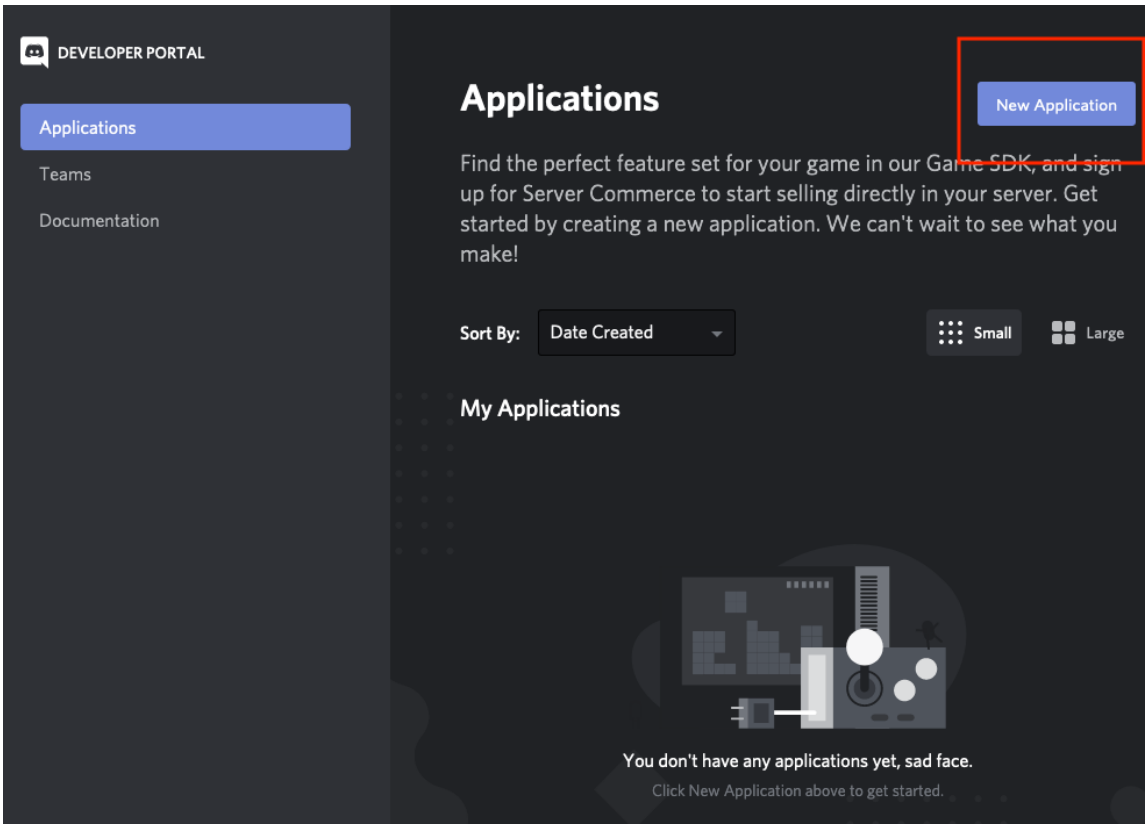


4. As you can see, we need an Application Id and Bot Token for Discord. Here is how you can get it:

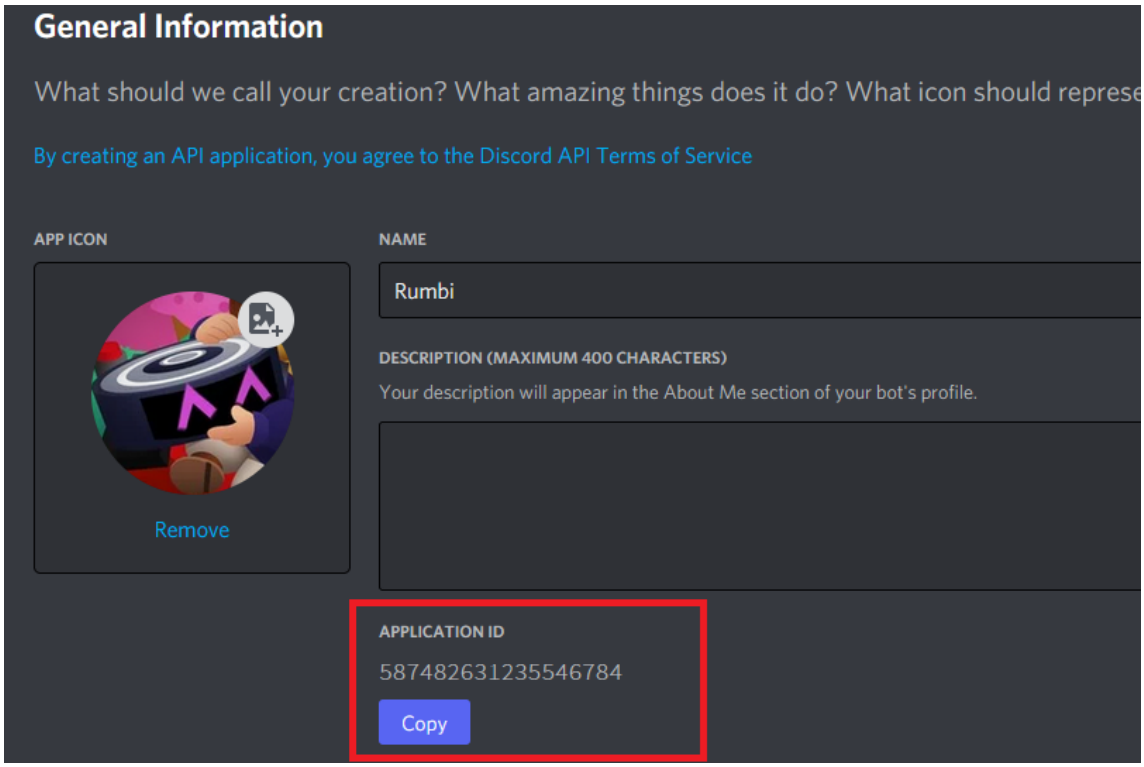
How to get Discord Application Id and Bot Token

1. Go to Discord Developer Portal by clicking this link: https://discordapp.com/developers/applications

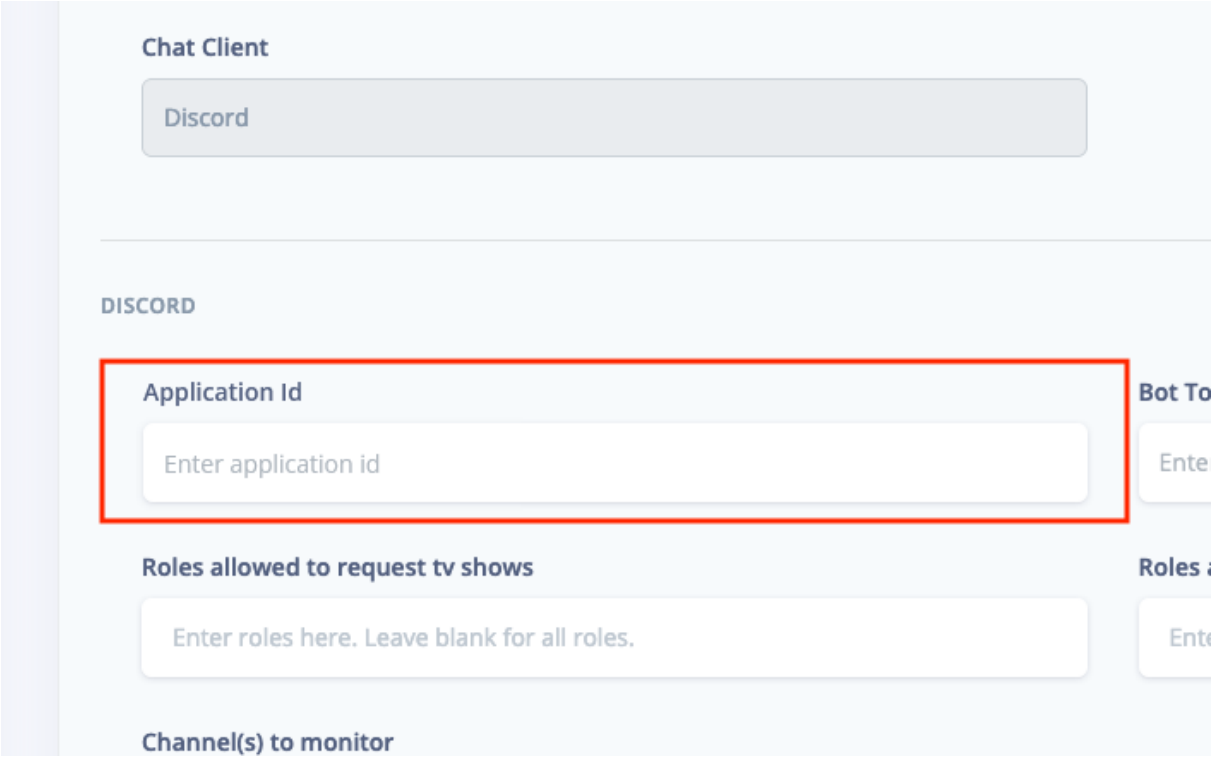
2. Click on New Application



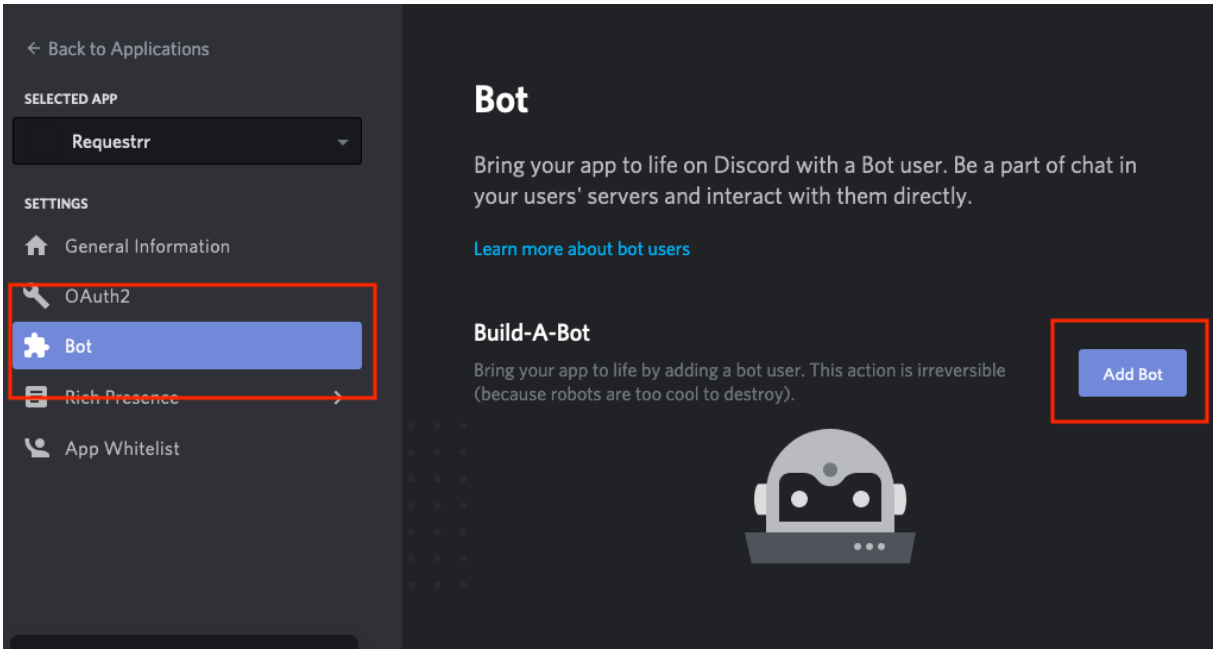
3. Give it a name and a profile picture if you so wish. The important thing is to copy the Application Id.



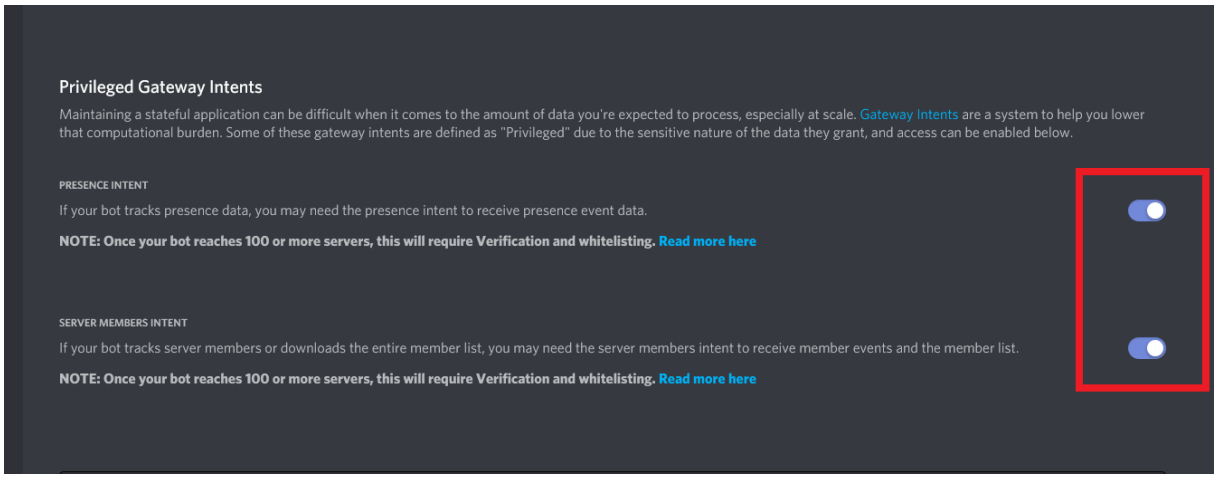
4. Paste it into Requestrr



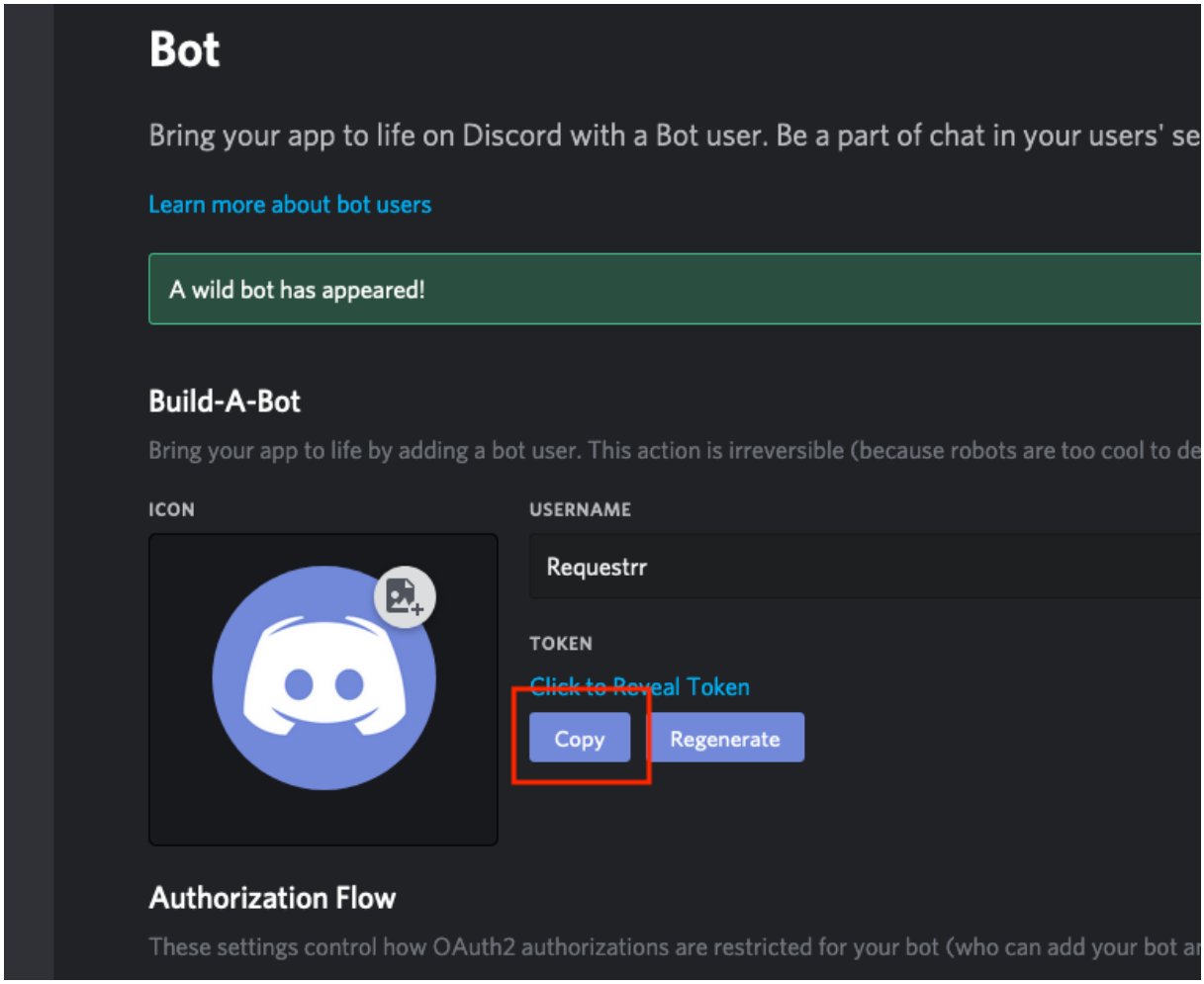
5. Go back into Discord Developer Portal and create a bot for your newly created application.



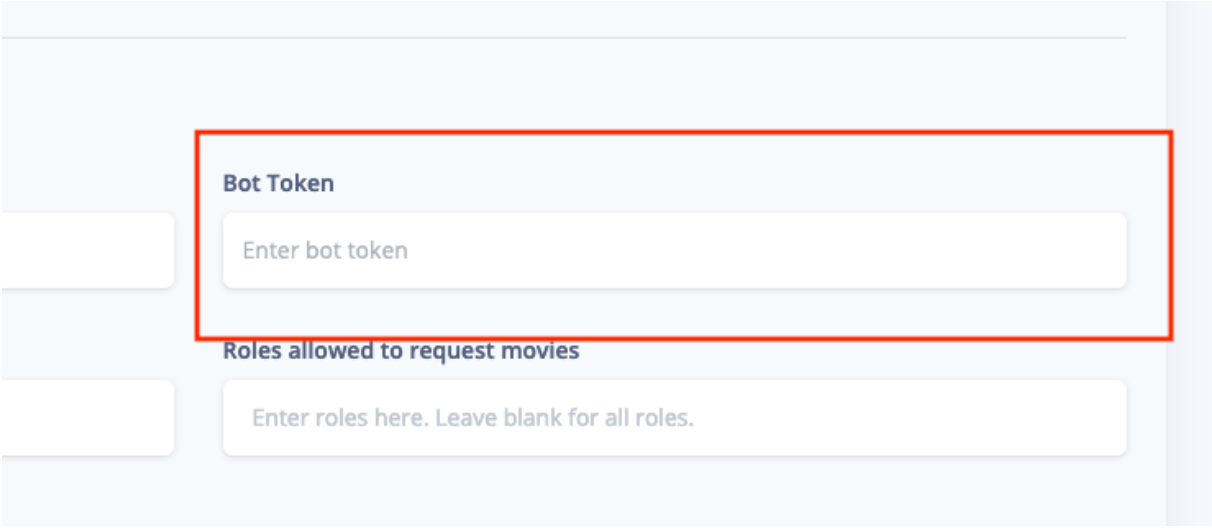
6. Make sure these 2 settings are enabled:



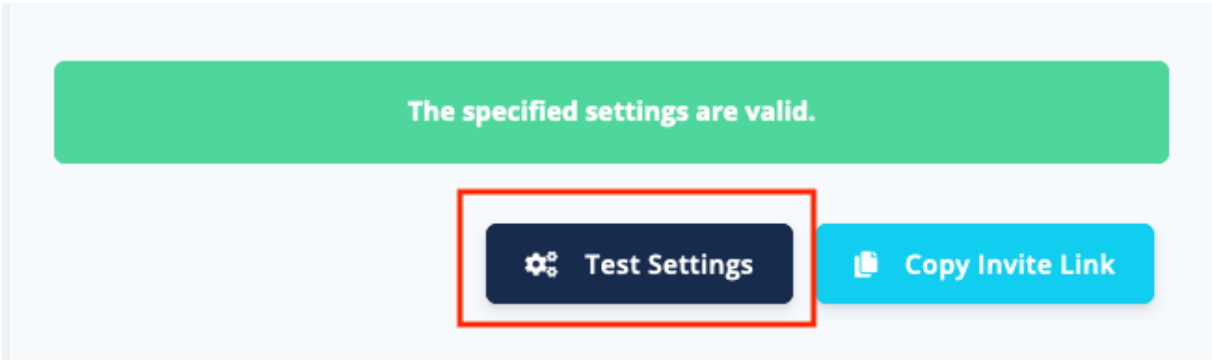
7. Now copy the Bot Token



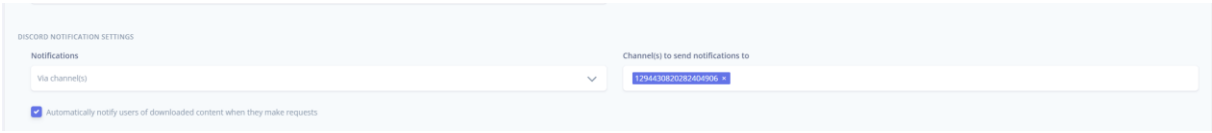
8. Paste the Bot Token into Requestrr



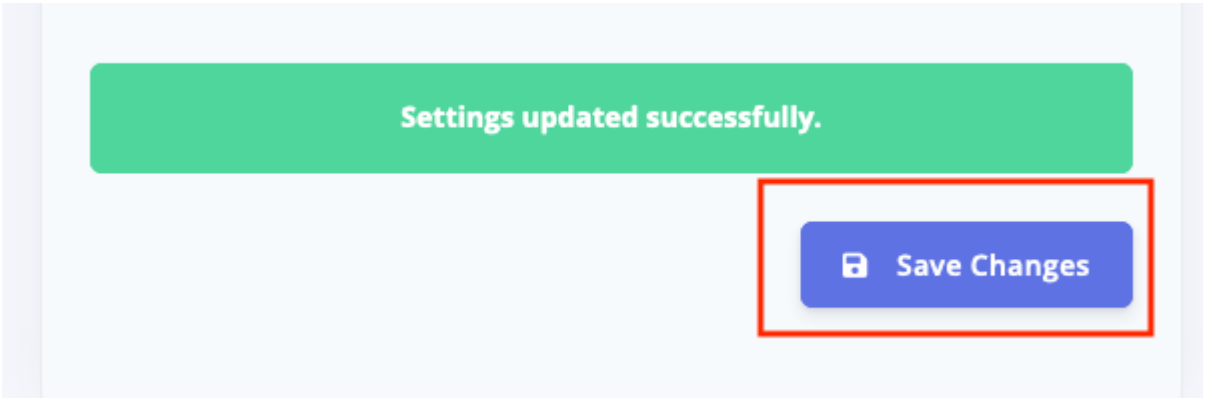
9. Click on “Test Settings” to check if the connection is good



10. Go to your discord server, right click on the channel you want your bot to be in and copy the channel ID. Paste it into “Channel(s) to send notifications to”

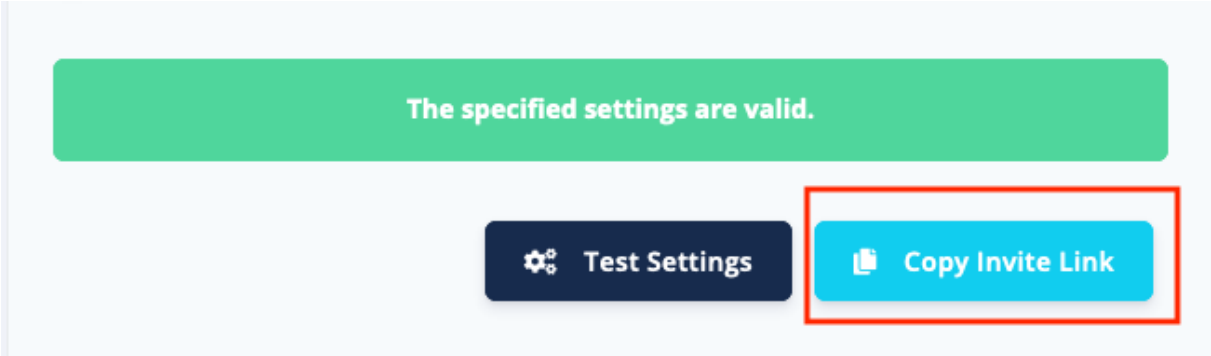


11. Finally, we can save our changes

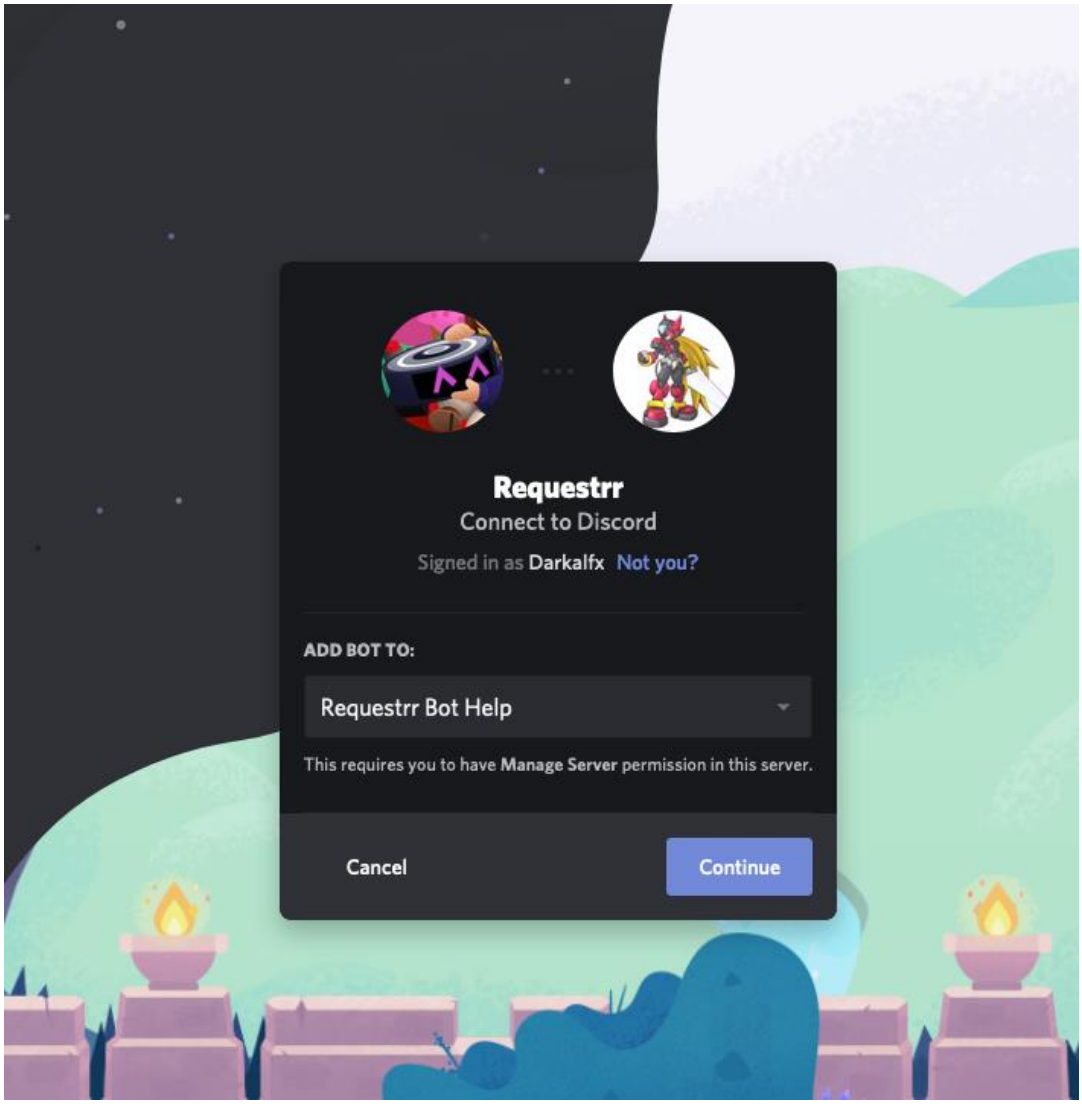


How to invite the bot to our Discord server

1. Be sure you have created a Discord server, or are admin in an existing one
1. Click “Copy Invite Link”



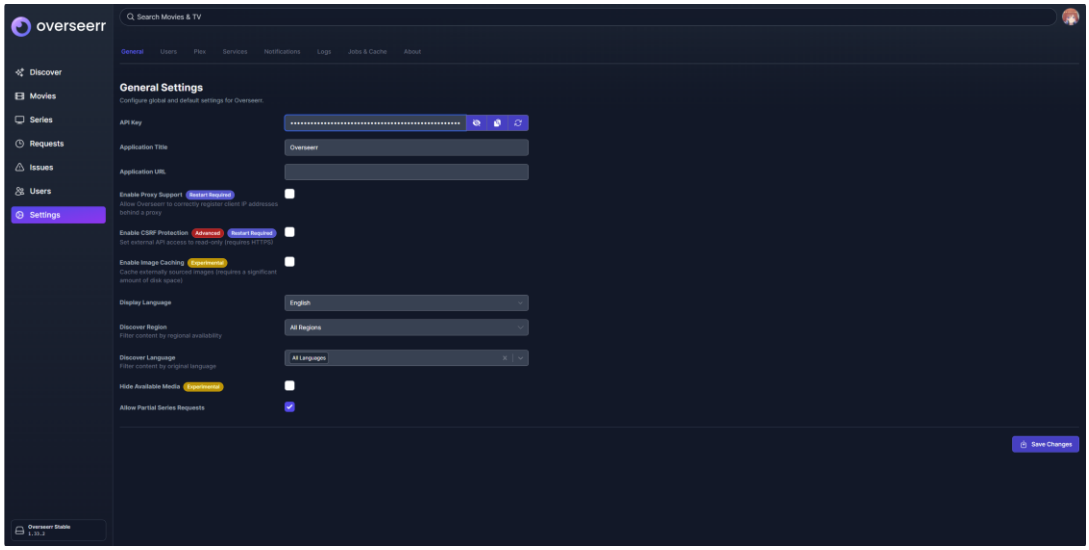
1. Choose the Server you whish to add the Bot to



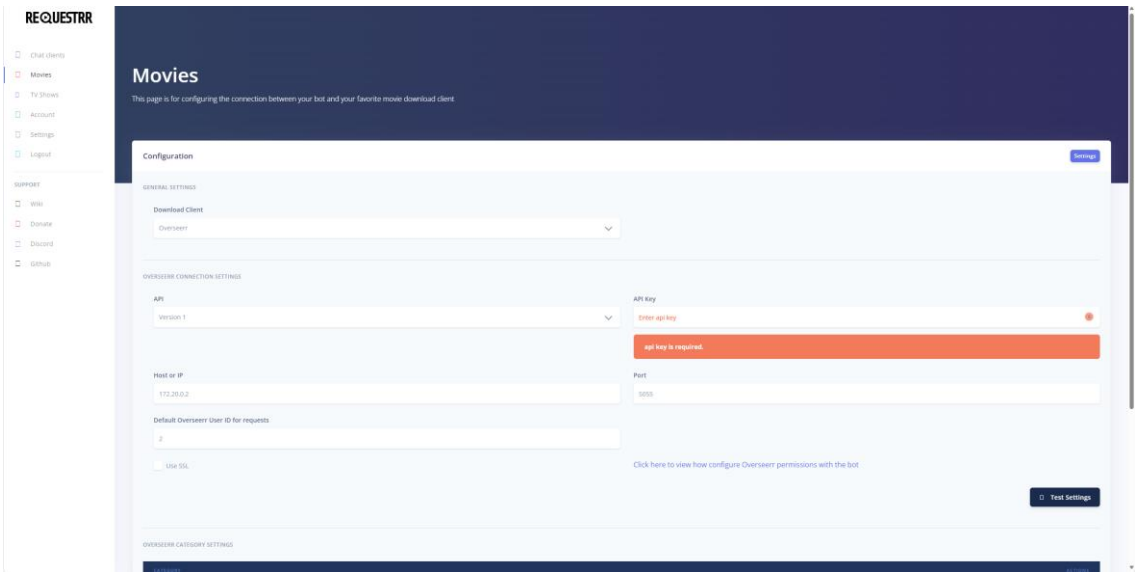
Connecting Requestrr to Overseerr (Or Radarr, Sonarr, Ombi)

Now we have to set up Requestrr to actually, well, request. The best way would be to use Overseer or Ombi, but it also works directly with Radarr and Sonarr.

1. Go to Overseer on your web-browser. <NAS-IP>:5055
2. Go to settings (On the left-hand side menu) and make sure “Enable CSRF Protection” is disabled (unticked)
3. While you’re here, also copy your Overseer API key.



4. Go to Requestrr on your web-browser. <NAS-IP>:4045
5. Go to Movies (On the left-hand side menu)

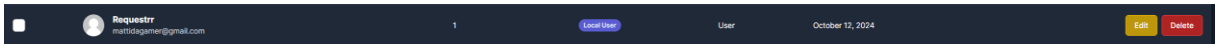


6. Choose Overseer as Download Client. Paste in your API Key. Host should be 170.20.0.2 (your synobridge) and port 5055.Default Overseer user is which user the requests should come from. To find out your user ID to as follows:

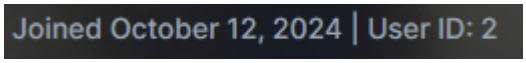
   i Go back to overseer

   ii Select “Users” from the menu

   iii Then you should see a list of all your users. I made a custom Requestrr user.



   Click on the desired user. Now you should see your user id



7. Click “Test Settings”, then scroll down and click on “Save Changes”
8. Now go to “TV Shows” from the menu and do the exact same as you did for movies. Most of it should already be automatically filled out by your movies section when you select “Overseerr” as client.
9. Now it works! You can go to your server and type /help in the channel you copied the ID for earlier, and it will list all available commands.

Modifications to set it up for UseNet

FAQ

What is Usenet?

Use(er) Net(work) is a worldwide distributed discussion system available on computers. It was developed from the general-purpose Unix-to-Unix Copy (UUCP) dial-up network architecture.

A major difference between a BBS or web message board and Usenet is the absence of a central server and dedicated administrator or hosting provider. Usenet is distributed among a large, constantly changing set of news servers that store and forward messages to one another via "news feeds".

-Wikipedia. https://en.wikipedia.org/wiki/Usenet ’

What is UseNet used for

In simple terms, when UseNet was created, it was meant to share news and messages, kind of like what reddit is today. But today it is (almost) only used to share media content.

Why are the files split into parts?

Usenet is a text forum. Message size is limited to 1 million bytes. A binary file must be split into pieces smaller than 1 million bytes. Each of these segments is a Usenet message.

Do I need to manually combine all the parts of the files afterwards?

You download an NZB file from your indexer. The NZB file contains a list of files, and a list of Message-IDs for each file. Your download client (newsreader) uses the NZB to download all the messages and reassemble the original file(s).

UseNet vs Torrenting for Media Downloads

Some will argue UseNet is better in every way, and in a lot of cases it is. But it isn't as black and white as this. UseNet will offer much greater speeds, often maxing out your ISP speed of 500Mbps/1Gbps/2.5Gbps. You will also not be dependent on seeders, or to be seeding yourself.

One thing to point out is the aspect of retention. Files are only stored so long on the file server. This is often about 10-15 years depending on provider but reposts happen all the time so it's not such a big issue. Another thing is that not all backbones (fileservers) have all the files. Therefore, even though an indexer could grab it you won't have all the files to complete a download. Another downside is that it costs money. All providers cost money, and most indexers do aswell. Some indexers have free trials and/or free plans though but keep a lookout for sales and deals.

To find some good deals, check out the wiki on reddit:

Providers: https://www.reddit.com/r/usenet/wiki/providerdeals/

Indexers: https://www.reddit.com/r/usenet/wiki/indexers/

Pros and Cons list for Torrent vs UseNet

Torrent

Pros	Cons
Torrenting is free besides the VPN (optional but highly recommended depending on region)	torrents are tracked by your ISP and illegal in certain countries
May be able to find stuff you can't find on usenet	Slow Downloads
Huge variety of content	Requires you to share or reseed
Easy to learn and setup	Completion is iffy with older and less sought after files
No retention limit, as longs as people are seeding the files	Quality of files is typically not as good as usenet
	Better chance of Internet provider banning/blocking you or busting you for downloads (if ou don’t use VPN)

UseNet

Pros	Cons
Always the fastest speed the server can provide (usually enough to max out your ISP)	Cost money (but you can find some good deals)
No VPN needed as you get SSL encryption with most servers included	A little tricky to setup but using this guide will hopefully help
No seeding, so you can turn your computer off & save electricity and not waste bandwidth	Difficult to understand at first, but hopefully this guide will help!
High quality files	Files can be removed for copyright
You can find almost anything popular, even if old. 10-15 years is how long data is kept on the servers typically (growing daily) but anybody can upload older files.	Rare/obscure stuff can be hard to find

Why are people talking about Linux ISOs?

To bypass the rule of not talking about piracy in r/usenet, people started calling media (movies, TV shows etc) for linux ISOs (since linux is free and open source). It's really just an ongoing joke aswell as a method of bypassing this rule.

What is the difference between providers and indexers, and what are their roles?

ELI5 (Explain Like I'm 5):

Let's say a provider owns a forest of LAND, a forest in which many people have buried their favorite.. Linux ISO's ... for all to share. These buried items are magical in that when you unbury it, you get a copy of the item and the item stays buried in the same spot for others to find. There's probably all kind of things buried all over the place, some are things you'd probably never want or need. You dig into one random spot and you find something! It's an empty Coca Cola bottle... well that's not great. I'm sure there are people out there who would want it, but you have zero interest int that bottle.

This makes you realize that you now a huge problem, you paid the provider to have access to the forest but you have no clue where anything you want is buried.

That's where the indexer comes in. Indexers give you access to their treasure maps that show where stuff is buried. Now you can use your treasure map to find the things you want!

As you go along your path of unburying the stuff you want, you realize that their map isn't perfect. Sometimes the thing you want is nowhere on their map. It is possible that the thing you want isn't buried anywhere in the forest, but it is just as possible that this particular map doesn't know where everything is buried, they just do the best they can to keep track of as many buried things as they can. That's when you decide to get other indexers, I mean treasure maps. With multiple treasure maps of the same forest, you have a good chance of finding items that you want, because one map may have a location that the others are missing! So having multiple maps is great because you are likelier to know the location of anything you are looking for in the forest.

Bonus eli5: you notice indexer treasure maps are way bigger than the forest, they go out into the mountains and the desert and the plains, all just as good to bury stuff. Of course you only paid one provider access to the forest. If you want to have access to other lands you realize you have to find and pay the provider that owns the deserts, or the one that owns the mountains. Having multiple providers (with different backbones) gives you access to more land that has potential to have buried items in it. Some providers sell the

access to the same forest(aka same backbone) because they pay the real forest owner for rights to also sell access to that forest. Buying access to that forest twice(two providers with same backbone) is a waste of money because you already have access from one, so you want to make sure that if you get multiple providers, they give you access to different backbones, err, I mean lands.

-Ericstern, rediit.
https://www.reddit.com/r/usenet/comments/1641vak/comment/jy7gsj7/?utm_source=share&utm_medium=web3x&utm_name=web3xcss&utm_term=1&utm_content=share_button

A lot of the provider deals, especially on black Friday, are for 15 months instead of 12. Is this a trick to make you renew when it's not a sale?

Well, not really. Almost all the popular providers and indexers let you stack time, so either you can buy in multiple bulks now to get 2x15, 3x15 or even 4x15 months of access. Or you can wait until next black Friday (or another deal opportunity) and stack more time.

But beware the process can be a little bit different for some providers. Some might for example only stack on the same plan, so you can't upgrade your account. It is always a good idea to research a little bit beforehand and maybe even contact either support on their website or a representative on reddit. They are usually really helpful with this kind of things.

Okay, I'm convinced. But what provider should I go for, and should I do an unlimited monthly/yearly plan or a block account?

Check out this reddit post:
https://www.reddit.com/r/usenet/comments/f215ko/a_new_users_guide_to_picking_providers_questions/

My recommendation would probably to go for Eweka has your main provider with unlimited access if you are in the EU, or Newshosting if you are in the US. Try to get a deal for $2.50 or $3.00 a month. Then if you wish you can also do block accounts on different backbones like Usenet Express or Abavia. Depending on usage you could also do something as crazy as myself and go for 2 unlimited accounts. I have 1 on Eweka(Omicron) and 1 on Frugal Usenet (Usenet.Farm and NetNews multi-backbone).

To check which providers use which backbone, take a look at this map (updated as of November 2024) https://cdn.rexum.space/usenet/14t2.svg.

The important thing is the big main block at the top. The branches from that don't really matter. E.g Eweka, Base IP, and HW Media are all on Omicron and will therefore yield the same results.

To find some good deals on providers, check out the reddit wiki here:
https://www.reddit.com/r/usenet/wiki/providerdeals/

Okay cool, what about indexer?

For indexers, you should have multiple for the best possible results and experience. The most popular (and best ones) are NZBGeek, DrunkenSlug, NZB.su and NinjaCentral. Note that both DrunkenSlug and NinjaCentral are semi-private indexers which either requires you to register when they have open registration a few times a year or be invited by another user.

Some indexers also provide lifetime access, while others are a yearly sum you pay. Some offer both as different plans. Note that lifetime refers to the lifetime of the service, and it is not guaranteed to last your lifetime. It might shut down tomorrow, or in 50 years. Who really knows?

Here are a few good indexers I would recommend, in no particular order:

Lifetime Membership:

Indexer	Registrations	Membership fee	Crypto	Account Limits
NZBGeek	Open	$80	Yes	Unlimited
Usenet-Crawler	Open	$20	Yes	1000 NZBs, 10,000 API hits/day
AltHUB	Open	$55	Yes	Unlimited
NzbPlanet	Open	£40	Yes	Unlimited NZBs, 20,000 API hits/day

Yearly Membership:

Indexer	Registrations	Membership fee	Crypto	Account Limits
abNZB	Open	$15 / $25	Yes	750 NZBs, 2,500 API hits/day / 2,500 NZBs, 10,000 Api hits/day
AltHub	Open	$10/yr / $25/3yrs / $40/5yrs	Yes	Unlimited
DrunkenSlug	Invite Only	€15 / €25	Yes	100 NZBs, 100 API hits/day, Unlimited
NinjaCentral	Closed	$15 / $25 / $150	Yes	Dan 0: 350 NZBs , 2000 API hits/day / Purplex: 900 NZBs , 3000 API hits/day / UNLIM: 10000 NZBs , 20000 API hits/day
NzbPlanet	Open	$12 / $24	Yes	VIP: Unlim NZBs, 2000 API hits/day / Platinum: Unlim NZBs, 20000 API hits/day
NZBGeek	Open	6U$/6Months / $12/yr /	Yes	Unlimited

		$18/2yrs / $40/5yrs		
nzb.su	Open	$15/yr / $28/2yrs / $42/3yrs	Yes	VIP: 600 NZBs, 5000 API hits/day

Free Membership

Indexer	Registration	Free Account Limits	Crypto	Upgradable through donations?
AnimeTosho	Open	Not that I'm aware of	No	No
DrunkenSlug	Invite-only	5 NZBs, 25 API hits/day	Yes	Yes
BinSearch	Open	Not that I'm aware	No	No
DigitalCarnage	Open	10 NZBs, 10 API hits/day	Yes	Yes
AltHub	Open	25 NZBs, 100 API hits/day	Yes	Yes

You can check out the whole matrix/list of indexers on the reddit wiki here: https://www.reddit.com/r/usenet/wiki/indexers/

Wow, that's a lot of choices. What do you recommend for a beginner?

If you feel overwhelmed and don't know where to start, I would go for a membership on eweka as your provider (on sale of course, as there are always sales. Check the wiki). For indexer I would recommend NZBGeek. Check also if DrunkenSlug and NinjaCentral have open registration, as they are 2 of the most powerful indexers. If you just want something cheap, usenet-crawler wouldn't be bad either.

If you want to really do something crazy, you could even have multiple indexers to find even more stuff. And add block accounts on UsenetExpress or Abavia to fill in Ewekas gaps. But this can also come at a later point in time.

What is a newsreader?

The newsreader is the term used for the downloader. You need it to read the xml files (news), download them, and then to reconstruct the original file. Before, in the old days, a lot of the good ones cost money. Today, the 2 most popular ones are free; Sabnzbd and NZBGet. NZBGet is very on and off with support, and at the time of writing it haven't had any updates in over 3 years. Meanwhile Sabnzbd got an update yesterday. So, I would personally choose Sabnzbd, but I guess that's personal preference.

Now I have a provider and an indexer, how to set it up with the *arrs?

It is actually very simple to set it up with the *arrs. If you wish to keep your VPN (good idea if you still have time on it for more privacy) you don't have to do anything on the yaml file except add a newsreader of choice. Then you just add this portion to your yaml:

```
  sabnzbd:
    image: lscr.io/linuxserver/sabnzbd:latest
    container_name: sabnzbd
    network_mode: service:gluetun
    environment:
      - PUID=<your UID>
      - PGID=<your GID>
      - TZ=Europe/Oslo
    volumes:
      - /volume1/docker/arr-stack/sabnzbd/config:/config
      - /volume1/Media:/Media
    restart: unless-stopped
    security_opt:
      - no-new-privileges:true
    depends_on:
      gluetun:
        condition: service_healthy
```

Also be sure to add port 8080:8080 in GlueTun.

If you don’t want to use a VPN, either because you want to use that money on a better provider, or an additional indexer or something else, then you will have to remove the whole GlueTun container. In addition, you would need to change the network mode on all containers from service:gluetun to host, or another bridge you might have. Then you would need to add all the ports required by each service. This is an example:

```
  sabnzbd:
    image: lscr.io/linuxserver/sabnzbd:latest
    container_name: sabnzbd
    network_mode: host
    environment:
      - PUID=<your UID>
      - PGID=<your UID>
      - TZ=Europe/Oslo
    volumes:
      - /volume1/docker/arr-stack/sabnzbd/config:/config
      - /volume1/Media:/Media
    ports:
      - 8080:8080/tcp
    restart: unless-stopped
    security_opt:
      - no-new-privileges:true
    depends_on:
      gluetun:
        condition: service_healthy
```

Personally, I also prefer to replace prowlarr with nzbhydra2. This is 100% optional and personal preference. But I do recommend using either one of these 2 instead of adding the indexers directly to Sonarr and Radarr, especially with UseNet. That is because they provide good stats that can help you with your financial decisions. If you see an indexer is nearly never used, why bother renewing it next year?

For NZBHydra 2, you can add this to your yaml, and make the appropriate modifications:

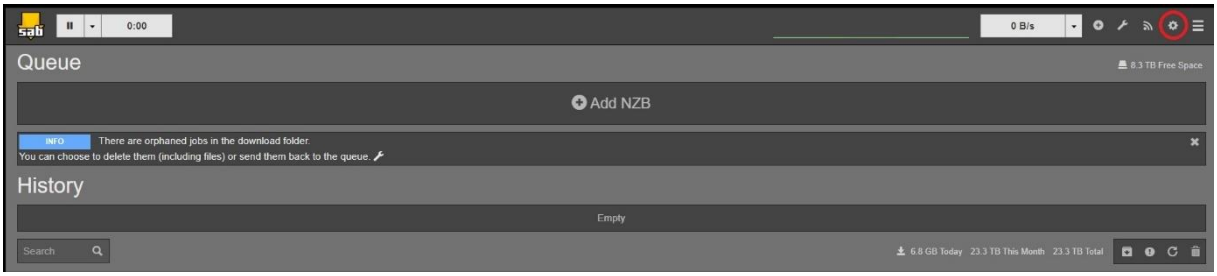
```
nzbhydra2:
    image: lscr.io/linuxserver/nzbhydra2:latest
    container_name: nzbhydra2
    network_mode: host
    environment:
      - PUID=<your UID>
      - PGID=<your GID>
      - TZ=Europe/Oslo
    ports:
      - 5076:5076/tcp
    volumes:
      - /volume1/docker/arr-stack/nzbhydra2:/config
      - /volume1/Media/Usenet
    restart: unless-stopped
```

Then you will also have to create a nzbhydra2 folder inside your arr-stack folder (in docker shared volume). Now we can start to configure our apps.

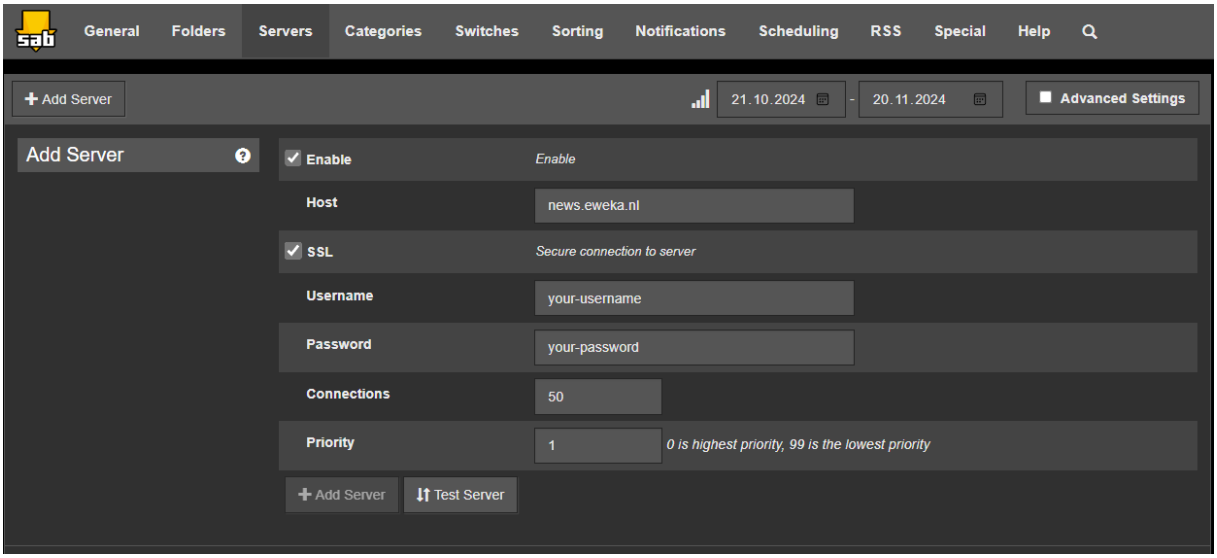
Configuring the apps

Sabnzbd

Open your web browser and navigate to <nas-ip>:8080. You will have to create an account for login credentials. When you have signed in, you will need to click the gear icon at the top right.



Then navigate to the “Servers” tab. Click add server. Then fill in your information, you can find everything on your provider’s website. Also select the priority you want, you should always have your main unlimited account set to 0 or 1 as the highest priority, then block with a lower priority (higher number). That way it only use the blocks if the main provider does not have the complete file. And always, always, always have SSL enable, if not your ISP can and will send you warning for copyright infringement. If you click on “Advanced Settings”, you can even double check the SSL port is correct.

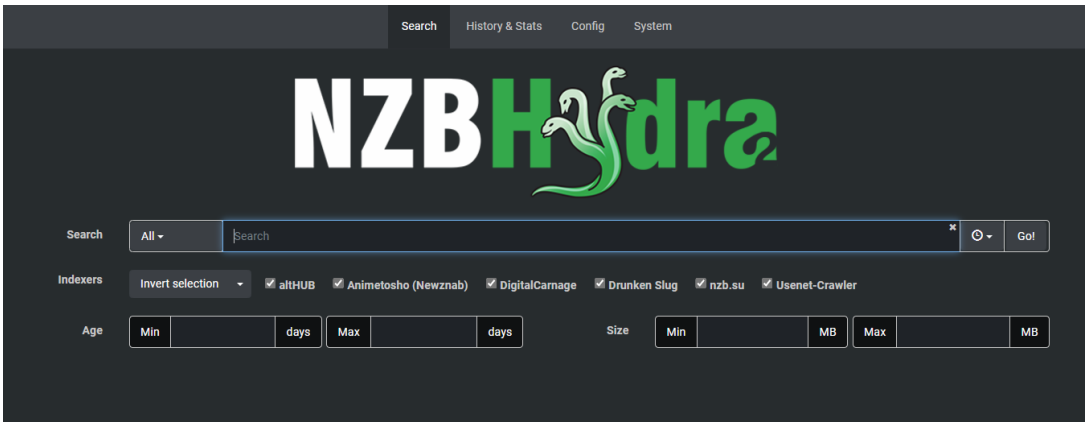


When that is done, you can click “Test Server” and if it works great, if not double check your information is correct.

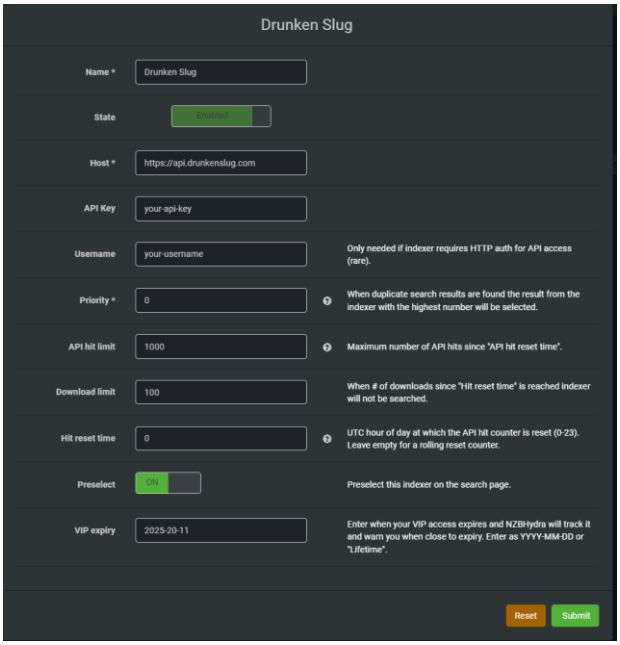
NZBHydra2

I am using NZBHydra 2, but Prowlarr will be almost identical. You have already set up torrent trackers on Prowlarr, so this will be no problem for you. But this is how to setup NZBHydra2:

1. Navigate to <nas-ip>:5076 on your browser.
2. Select config



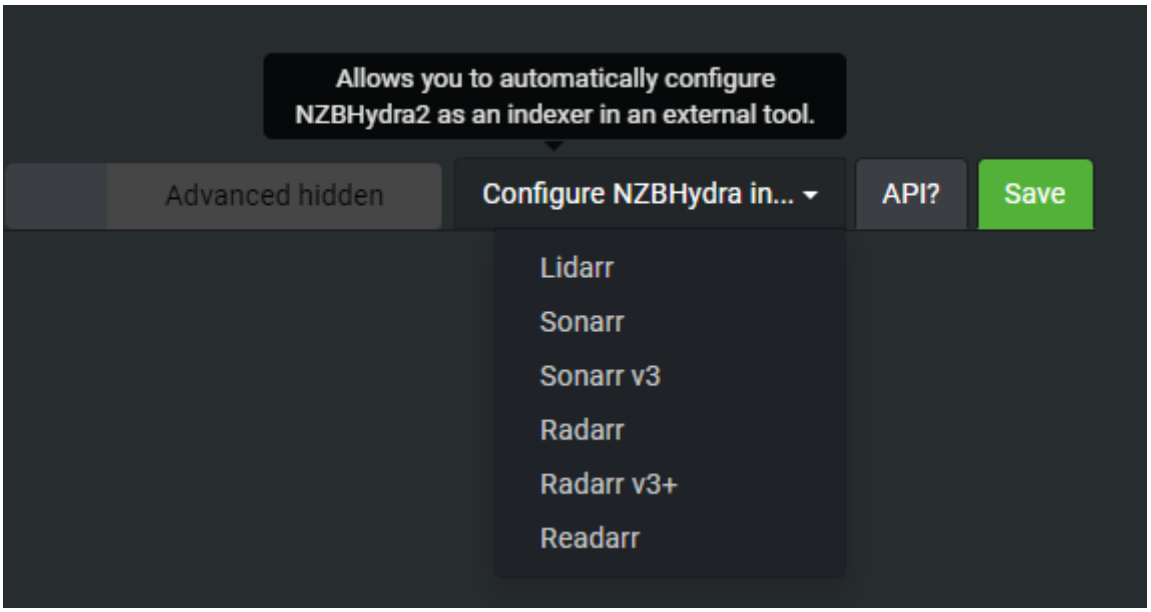
3. Go to the indexers tab and click on “Add Indexer”
4. Select “Choose from preset”. If you can’t find your indexer, then select “Add custom newznab indexer”
5. For this example, I will add DrunkenSlug. Fill in your information. All should be on your provider website:



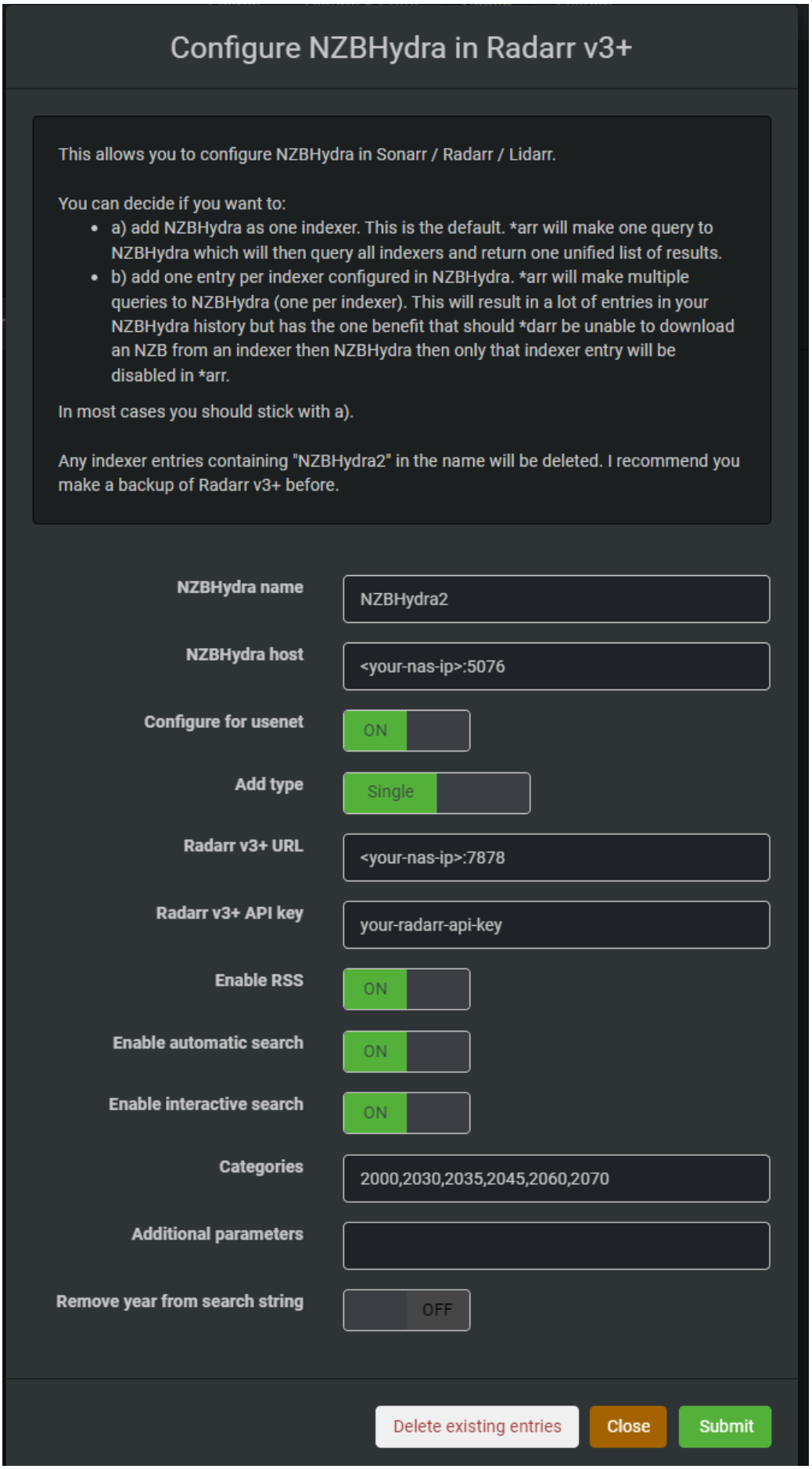
If you don’t have an API or download limit, and your VIP never expires you can leave these fields blank.

6. Click “Submit”
7. **REMEMBER TO CLICK “SAVE”**

NZBHydra makes it very easy to connect it to the *arrs. You can do it manually, like with Prowlarr, but you can also just click on “Configure NZBhydra in…” next to the save button



Select Lidarr, Sonarr v3, Radarr v3 or Readarr. Normal Sonarr and Radarr are outdated and not used anymore. Then you just fill in your information.


Configure NZBHydra in Radarr v3+
This allows you to configure NZBHydra in Sonarr / Radarr / Lidarr.
You can decide if you want to:
a) add NZBHydra as one indexer. This is the default. *arr will make one query to NZBHydra which will then query all indexers and return one unified list of results.
b) add one entry per indexer configured in NZBHydra. *arr will make multiple queries to NZBHydra (one per indexer). This will result in a lot of entries in your NZBHydra history but has the one benefit that should *darr be unable to download an NZB from an indexer then NZBHydra then only that indexer entry will be disabled in *arr.
In most cases you should stick with a).
Any indexer entries containing "NZBHydra2" in the name will be deleted. I recommend you make a backup of Radarr v3+ before.
NZBHydra name
NZBHydra2
NZBHydra host
<your-nas-ip>:5076
Configure for usenet
ON
Add type
Single
Radarr v3+ URL
<your-nas-ip>:7878
Radarr v3+ API key
your-radarr-api-key
Enable RSS
ON
Enable automatic search
ON
Enable interactive search
ON
Categories
2000,2030,2035,2045,2060,2070
Additional parameters
Remove year from search string
OFF
Delete existing entries
Close
Submit

That’s it! Now you can download with Radarr and Sonarr using UseNet! You can also do it manually. Sometimes Radarr and Sonarr can’t find everything on the indexer. Then it can be good to browse yourself.

You could do this by going to each indexer and searching, but with NZBHydra you can search ALL your indexers at the same time. Then just select the one(s) you wish to download and select “Results as zip”. You can also download each file individually by clicking the download button to the right of each file. Then just go to Sabnzbd, click the + button at the top right and upload either your zip file or nzb file. If you select category movies or series (or whatever you have configured with Sonarr and Radarr) they will automatically also get picked up and imported by the respective apps.

Now, just lean back and enjoy your experience with UseNet!

Thanks for reading my guide. I hope this was helpful to you. If it was, be sure to let me know with a comment, or upvote the reddit post! You can also star the Github project or send me a DM letting me know how awesome this guide was 😊

Sources

Marius Hosting have a lot of great guides on how to setup different things on Synology:

https://mariushosting.com/

DrFrankenstein's Tech Stuff also has a lot of useful information on a bunch of subjects within the synology ecosystem:

https://drfrankenstein.co.uk/

TRaSH Guides. The BEST source for configuring Radarr and Sonarr:

https://trash-guides.info/

Radarrs official docs:

https://github.com/Radarr/Radarr

Sonarrs official docs:

https://github.com/sonarr/sonarr

Lidarrs official docs:

https://github.com/lidarr/lidarr

Overseerrs official docs:

https://github.com/sct/overseerr

Requestrrs official docs:

https://github.com/darkalfx/requestrr

GlueTUNs official docs:

https://github.com/qdm12/gluetun-wiki

Awesome -arrs wiki:

https://github.com/Ravencentric/awesome-arr

Wikipedia, Usenet:

https://en.wikipedia.org/wiki/Usenet

NZBGet GitHub Project:

https://github.com/nzbget/nzbget

Sabnzbd GitHub Project:

https://github.com/sabnzbd/sabnzbd

The amazing UseNet subreddit. Lots of helpful people and helpful articles:

https://www.reddit.com/r/usenet