

# New ARR Stack !!!



The graphic features the title "ARR Stack" in large, bold, gold letters. To the right is a red and gold "NEW!" starburst. Below the title are eight logos arranged in two columns: RADARR, Sonarr, Lidarr, Prowlarr on the left, and qBittorrent, Jellyfin, and Azarr on the right. A man in a dark shirt is in the bottom right corner, pointing towards the logos.

**ARR Stack**

**NEW!**

**RADARR**

**Sonarr**

**Lidarr**

**Prowlarr**

**qb qBittorrent**

**Jellyfin**

**azarr**

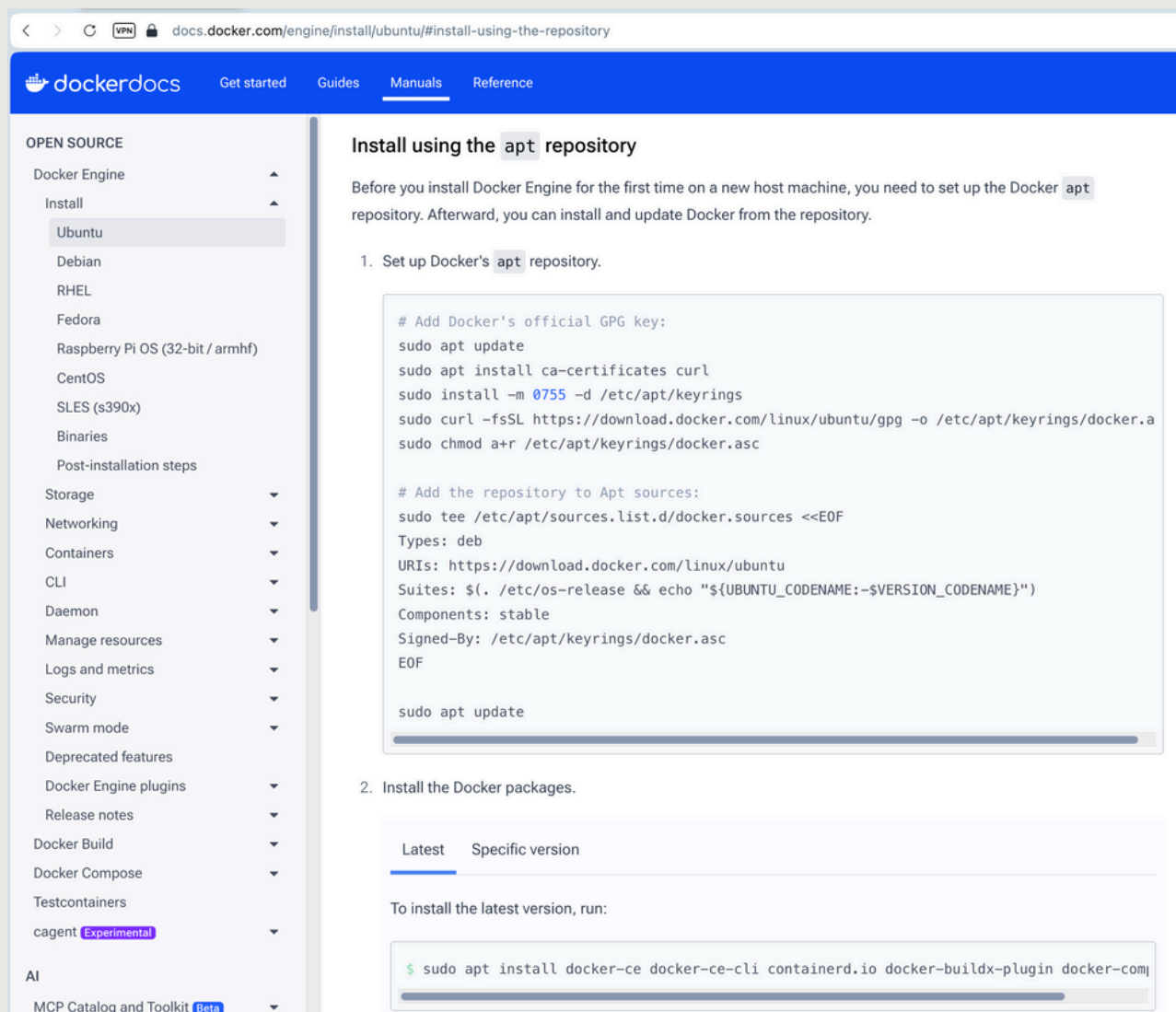
[Link to Automation Avenue platform](#)

New instructions to run ARR stack as a single docker compose file.  
ARR stack includes Jellyfin, qBittorrent, Radarr, Sonarr, Lidarr and Bazarr.  
You will find README.md file and docker-compose.yml file in [THIS](#) Github repo

To run this ARR stack, you need an operating system able to run docker compose. Every Linux OS can run it natively, and Windows / Mac can run docker and docker compose if you install Docker Desktop ( this runs tiny Linux VM in the background and is capable of running docker and docker compose).

Go to [THIS](#) Docker Docs website and use 'Install using repository' section to find your operating system.

For example - for Ubuntu its this section:



The screenshot shows the Docker Docs website at [docs.docker.com/engine/install/ubuntu/#install-using-the-repository](https://docs.docker.com/engine/install/ubuntu/#install-using-the-repository). The left sidebar shows the navigation menu with 'Ubuntu' selected under the 'Install' section. The main content area is titled 'Install using the apt repository' and includes the following text: 'Before you install Docker Engine for the first time on a new host machine, you need to set up the Docker apt repository. Afterward, you can install and update Docker from the repository.'

1. Set up Docker's apt repository.

```
# Add Docker's official GPG key:
sudo apt update
sudo apt install ca-certificates curl
sudo install -m 0755 -d /etc/apt/keyrings
sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o /etc/apt/keyrings/docker.asc
sudo chmod a+r /etc/apt/keyrings/docker.asc

# Add the repository to Apt sources:
sudo tee /etc/apt/sources.list.d/docker.sources <<EOF
Types: deb
URIs: https://download.docker.com/linux/ubuntu
Suites: $(. /etc/os-release && echo "${UBUNTU_CODENAME:-$VERSION_CODENAME}")
Components: stable
Signed-By: /etc/apt/keyrings/docker.asc
EOF

sudo apt update
```

2. Install the Docker packages.

Latest Specific version

To install the latest version, run:

```
$ sudo apt install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-com
```

Follow that guide to install docker and docker compose components:

*# Add Docker's official GPG key:*

```
sudo apt update
sudo apt install ca-certificates curl
sudo install -m 0755 -d /etc/apt/keyrings
sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o /etc/apt/keyrings/
docker.asc
sudo chmod a+r /etc/apt/keyrings/docker.asc
```

*# Add the repository to Apt sources:*

```
sudo tee /etc/apt/sources.list.d/docker.sources <<EOF
Types: deb
URIs: https://download.docker.com/linux/ubuntu
Suites: $(. /etc/os-release && echo "${UBUNTU_CODENAME:-
$VERSION_CODENAME}")
Components: stable
Signed-By: /etc/apt/keyrings/docker.asc
EOF
```

```
sudo apt update
```

```
sudo apt install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-
compose-plugin
sudo systemctl status docker
sudo docker run hello-world
docker compose version
```

If the last 2 commands are succesful, that means docker and docker compose are installed correctly.

Now we will follow [TRASH GUIDES](#) and [SERVARR WIKI](#) to build our docker compose file.

To create folder structure we need to run below:

```
sudo mkdir -p /data/{torrents/{tv,movies,music},media/{tv,movies,music}}
sudo apt install tree
tree /data
sudo chown -R 1000:1000 /data
sudo chmod -R a=,a+rX,u+w,g+w /data
ls -ln /data
```

It will create a folder structure as shown below:

```
marek@new-arr:~$ tree /data
/data
├── media
│   ├── movies
│   ├── music
│   └── tv
└── torrents
    ├── movies
    ├── music
    └── tv
```

Now we need to copy the docker-compose.yml file from [THIS](#) location to our server.  
We can use git clone command like:

```
git clone https://github.com/automation-avenue/arr-new.git
```

or simply create new file with:

```
sudo nano docker-compose.yml
```

and in Github click 'copy raw file' and paste to our docker-compose.yml file.

Now Ctrl-O , Enter, Ctrl-X to save and we should be ok to run all services with:

```
sudo docker compose up -d
```

command.

This should pull and run all services:

```

marek@new-arr:/data$ sudo docker compose up -d
[+] up 55/57
✓ Image ghcr.io/hotio/lidarr:latest Pulled
✓ Image ghcr.io/hotio/bazarr:latest Pulled
✓ Image ghcr.io/hotio/prowlarr:latest Pulled
✓ Image ghcr.io/hotio/qbittorrent:latest Pulled
✓ Image ghcr.io/hotio/sonarr:latest Pulled
✓ Image ghcr.io/hotio/jellyfin:latest Pulled
✓ Image ghcr.io/hotio/radarr:latest Pulled
✓ Network arr_network Created
✓ Container radarr Created
✓ Container qbittorrent Created
✓ Container sonarr Created
✓ Container lidarr Created
✓ Container jellyfin Created
✓ Container bazarr Created
✓ Container prowlarr Created
marek@new-arr:/data$

```

Next step is to configure all services.

Configure qBittorrent:

First run

sudo docker logs qbittorrent, it will show us the temporary password for it:

```

***** Information *****
To control qBittorrent, access the WebUI at: http://localhost:8080
The WebUI administrator username is: admin
The WebUI administrator password was not set. A temporary password is provided for this session: 6aR2XvnRL
You should set your own password in program preferences.
marek@new-arr:/data$

```

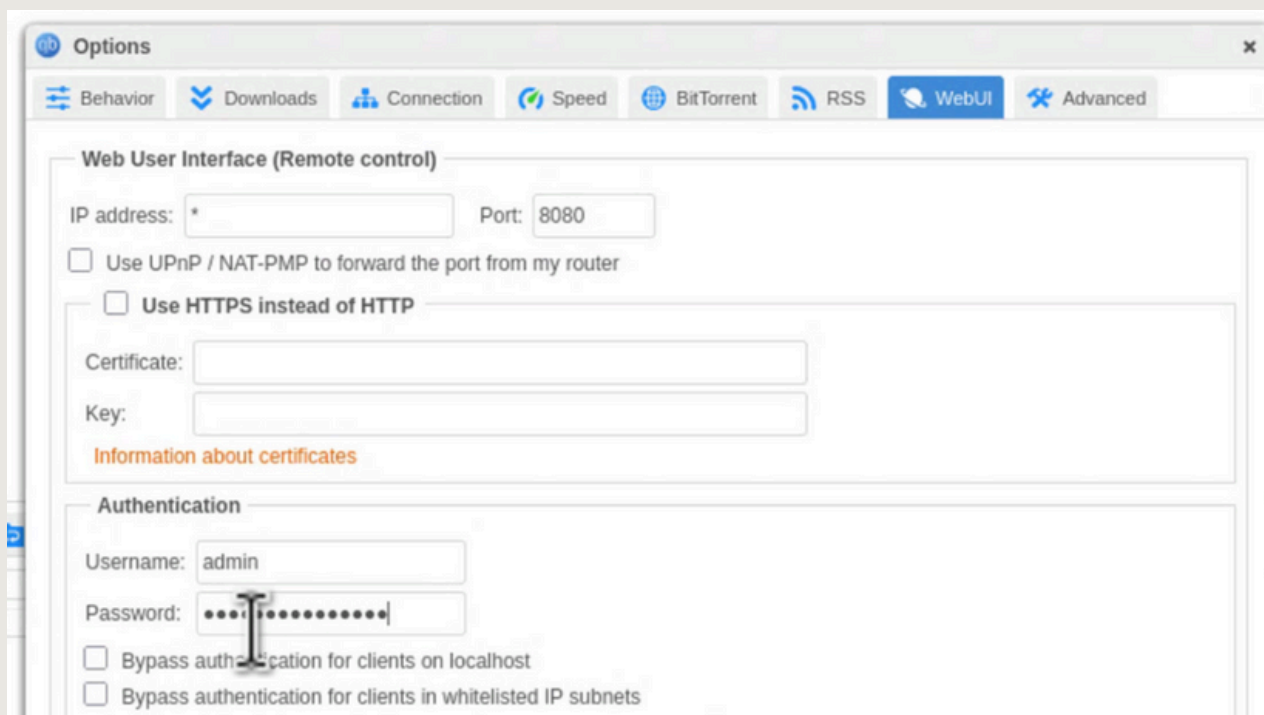
Now you can go to URL:

If you are on the host: <http://localhost:8080>

From other device on your network: <http://<host ip address>:8080>

and log on using details provided in container logs.

Go to Tools - Options - WebUI - you can change the user and password here but remember to scroll down and save it.



In left panel go to Categories - All - right click and 'add category':

For Radarr: Category: movies

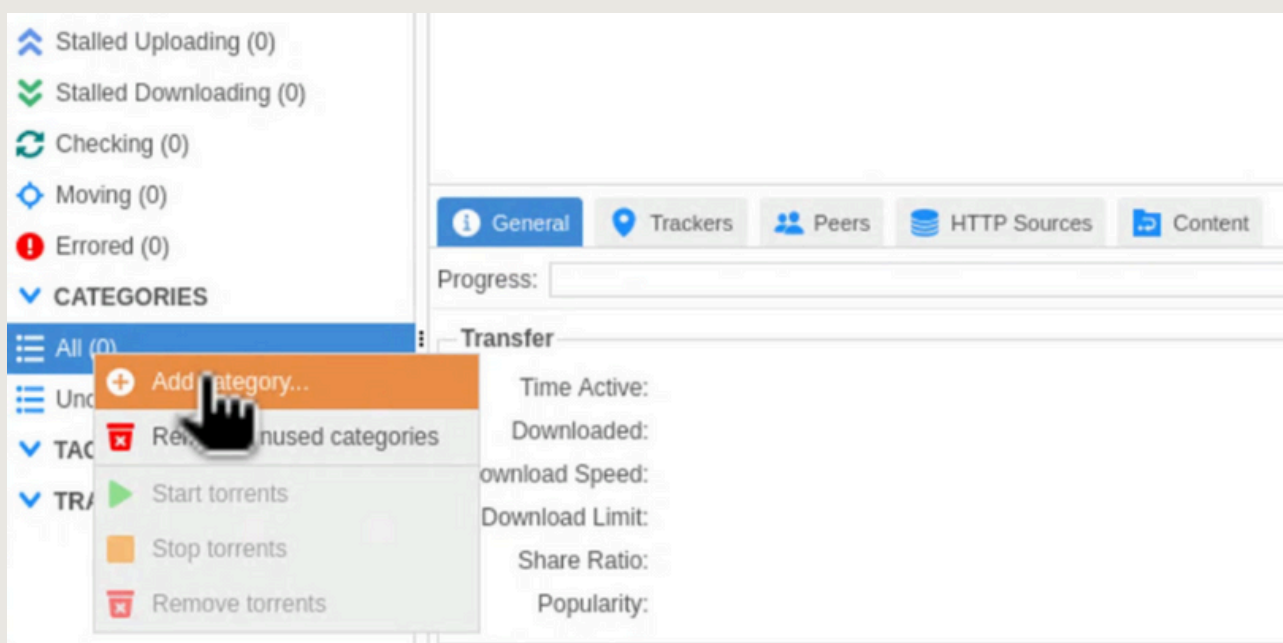
Save Path: movies

For Sonarr: Category: tv

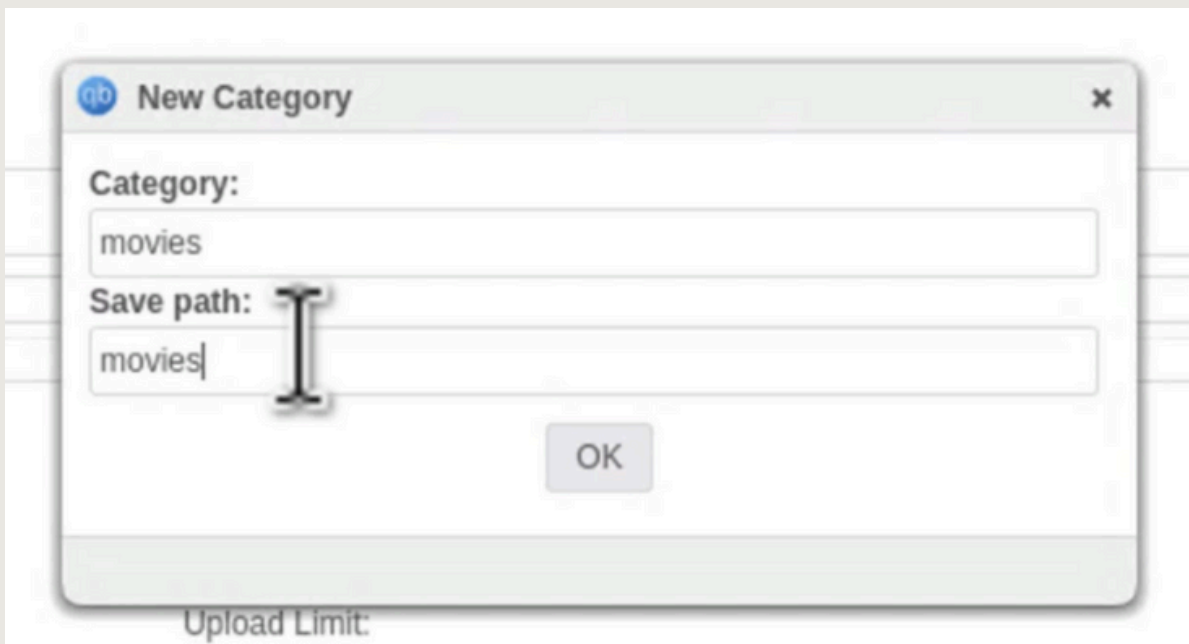
Save Path: tv

For Lidarr: Category: music

Save Path: music







Create categories first and only then configure the steps below, as doing it opposite way round caused the Categories to disappear :)

With categories created - go to - Tools - Options - Downloads and in Saving Management make sure your settings match **THIS**

So Default Torrent Management Mode - Automatic

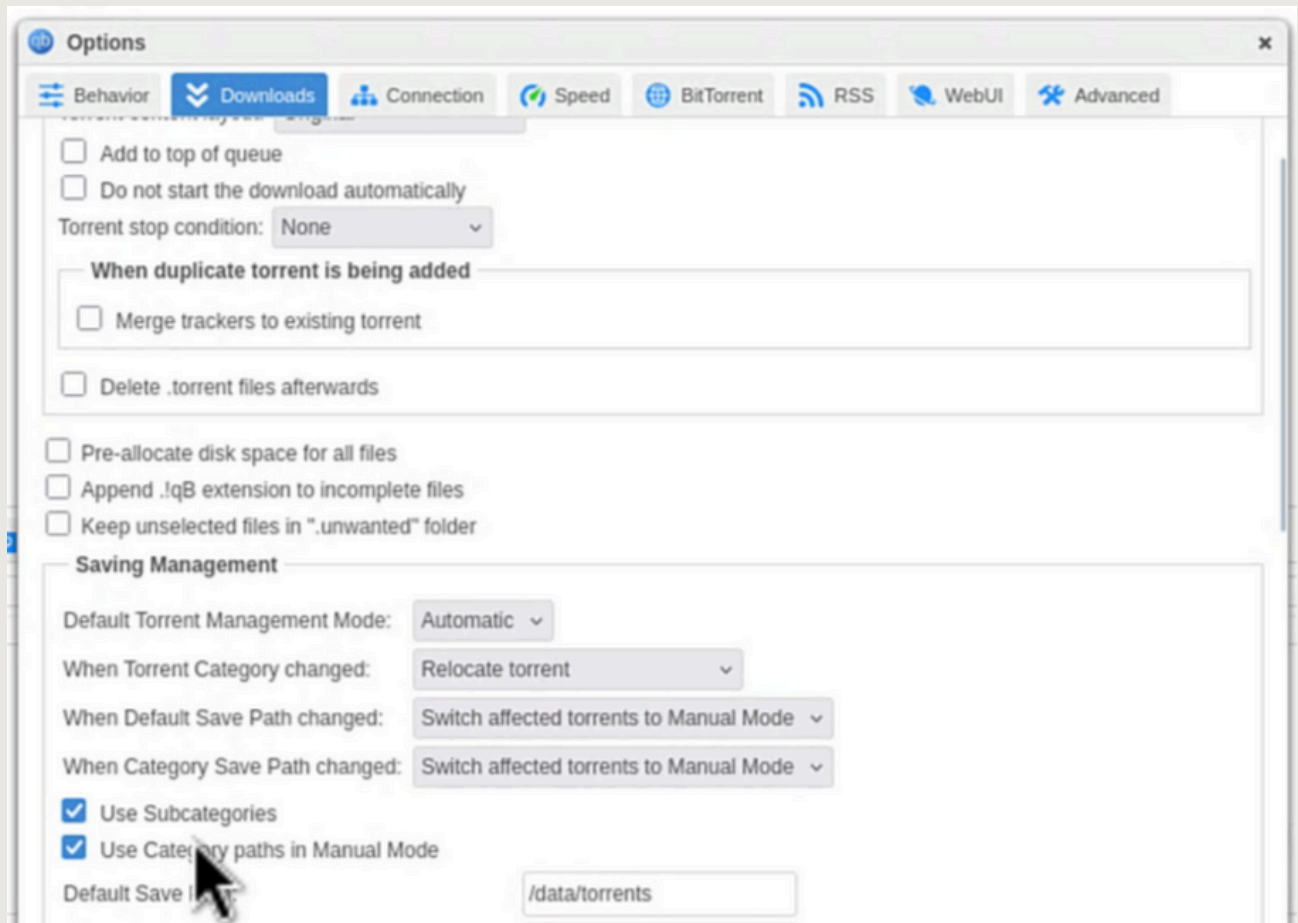
When Torrent Category changed - Relocate torrent

When Default Save Path Changed - Switch affected torrents to Manual Mode

When Category Save Path Changed - Switch affected torrents to Manual Mode

Tick BOTH BOXES for Use Subcategories and Use Category paths in Manual Mode  
(NOT shown on Trash Guides)

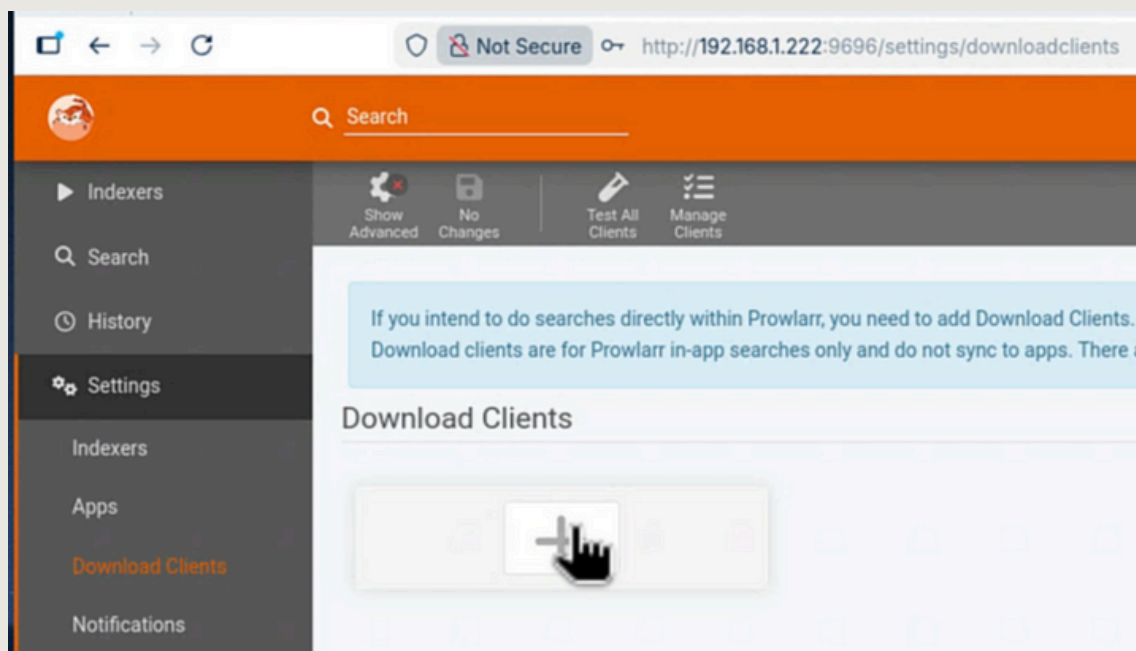
Default Save Path: - set to /data/torrents (so it matches your folder structure) - then scroll down and Save



Now configure Prowlarr service:

<http://<host ip>:9696>

Go to Settings - Download Clients - + symbol - Add download client - choose qBittorrent (unless you decided to use different download client)





## Add Download Client - qBittorrent



Name

Enable ☒

Host

Port

Use SSL ☐ Use a secure connection. See Options -> Web UI -> 'Use HTTPS instead of HTTP' in qBittorrent.

Username

Password

Default Category

Default fallback category if no mapped category exists for a release. Adding a category specific to Prowlarr avoids conflicts with unrelated non-Prowlarr downloads. Using a category is optional, but strongly recommended.

Priority  (0) ▼

Priority to use when grabbing items

Initial State  (0) ▼

Initial state for torrents added to qBittorrent. Note that Forced Torrents do not abide by seed restrictions



Test

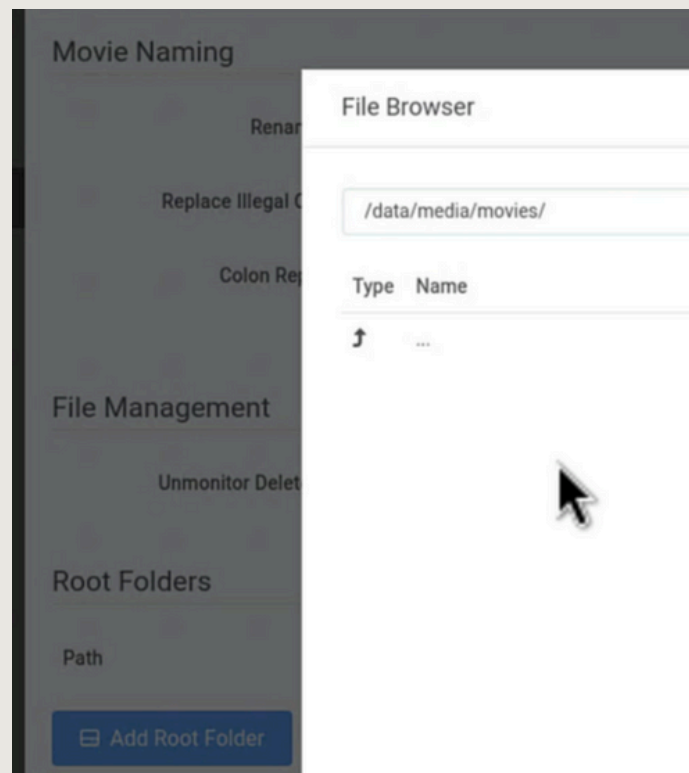
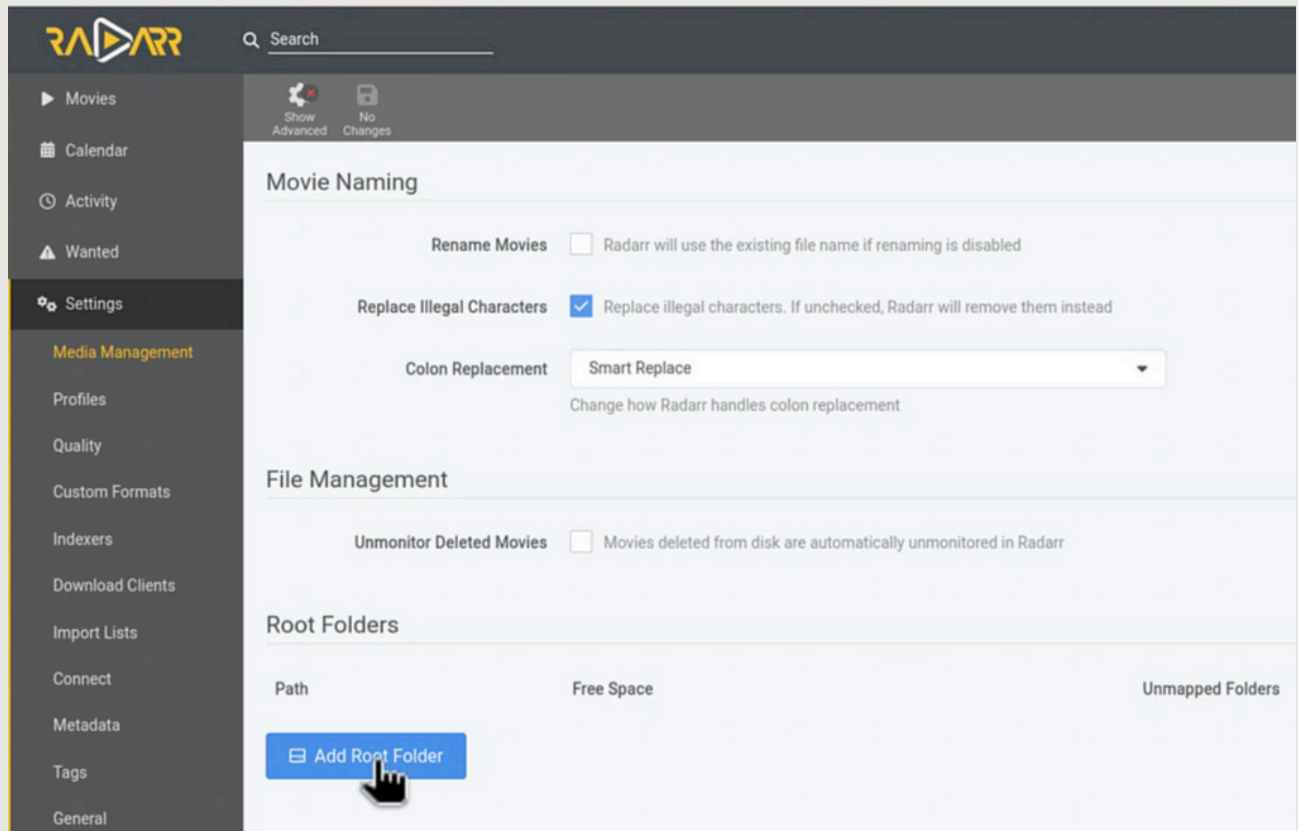
Cancel

Save

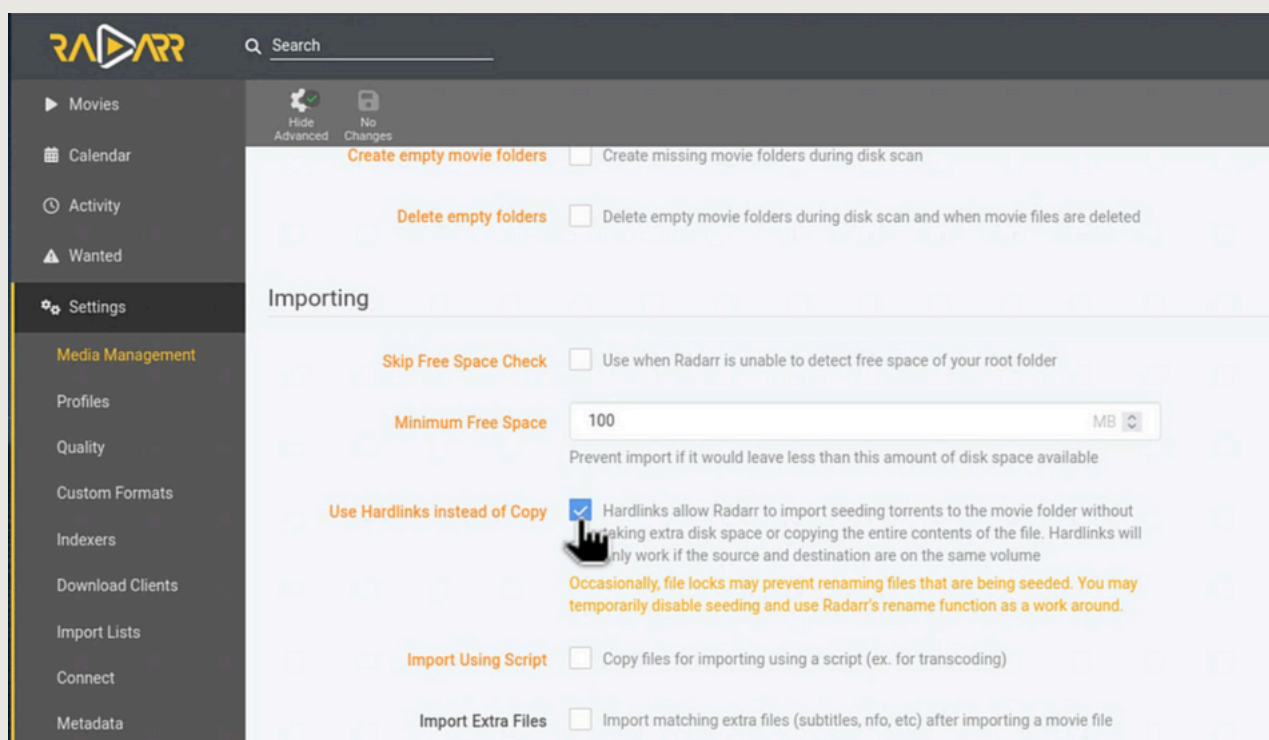
Now configure Radarr:

http://<host ip address>:7878

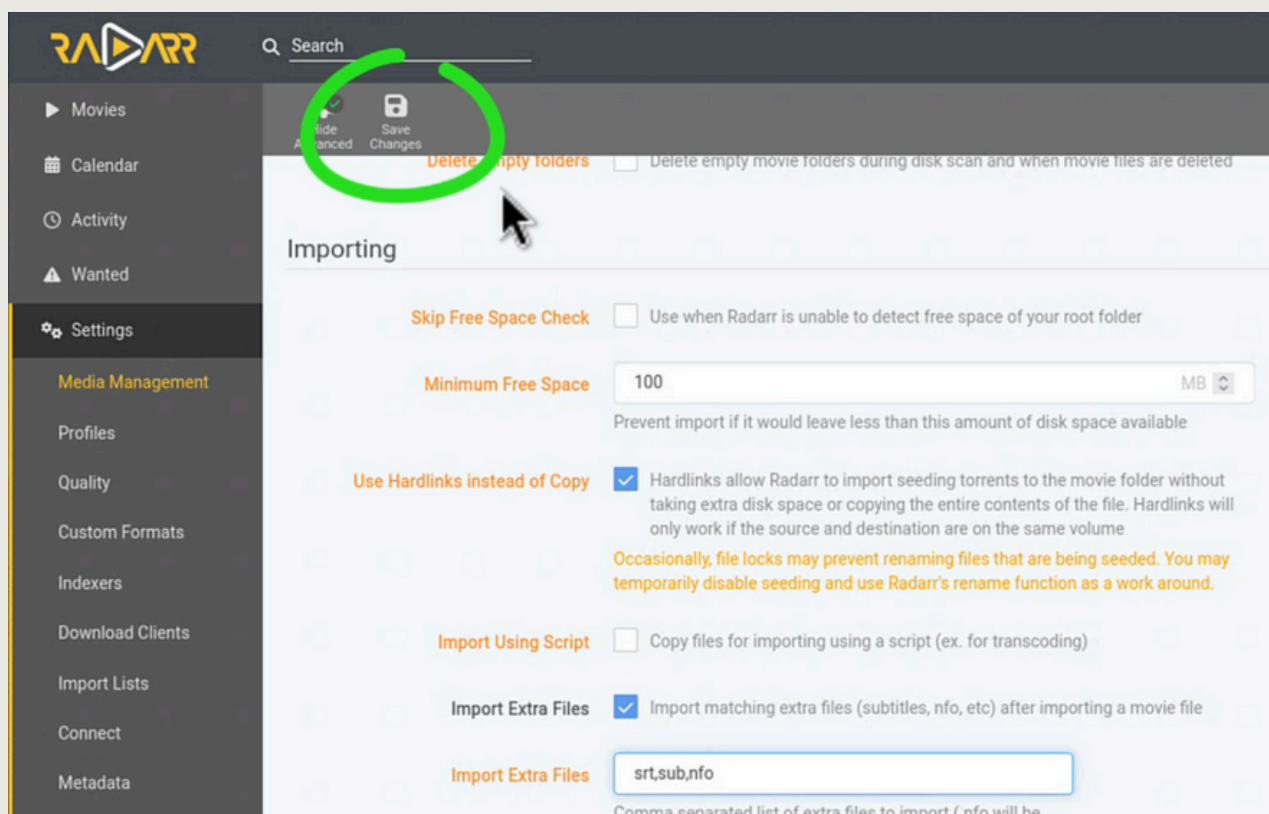
Go to Settings - Media Management - Add Root Folder (scroll down to the bottom).  
Set `/data/media/movies` as your root folder



Still in Settings - Media Management - click Show Advanced - Importing - Use Hardlinks instead of Copy - make sure its 'ticked'



Optional - you can also tick Rename Movies and Delete empty movie folders during disk scan , and in Import Extra Files - make sure that box is ticked and in Import Extra files field type srt,sub,nfo (those 3 changes are all optional)



Then Settings- Download clients - click plus symbol, choose qBittorrent etc - basically same steps as for Prowlarr but change the Category to movies (needs to match Category that you created in qBittorrent)

Add Download Client - qBittorrent

Name

qBittorrent

Enable

☒

Host

qbittorrent

Port

8080

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

admin

Password

••••••••••••••••

Category

movies

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

Cancel

Save

Now go to Settings - General - scroll down to API key - Copy API key

Not Secure http://192.168.1.222:7878/settings/general

**RADARR** Search

Hide Advanced No Changes

This application's external URL including http(s)://, port and URL base

Enable SSL ☐ Requires restart running as administrator to take effect

### Security

Authentication Forms (Login Page) ▼

Require Username and Password to access Radarr  
To prevent remote access without authentication, Radarr now requires authentication to be enabled. You can optionally disable authentication from local addresses.



Authentication Required Enabled ▼

Change which requests authentication is required for. Do not change unless you understand the risks.

Username marek

Password .....

Password Confirmation

API Key 5d5c4a313a64407888918edc3a646907  

Requires restart to take effect

Certificate Validation Enabled ▼

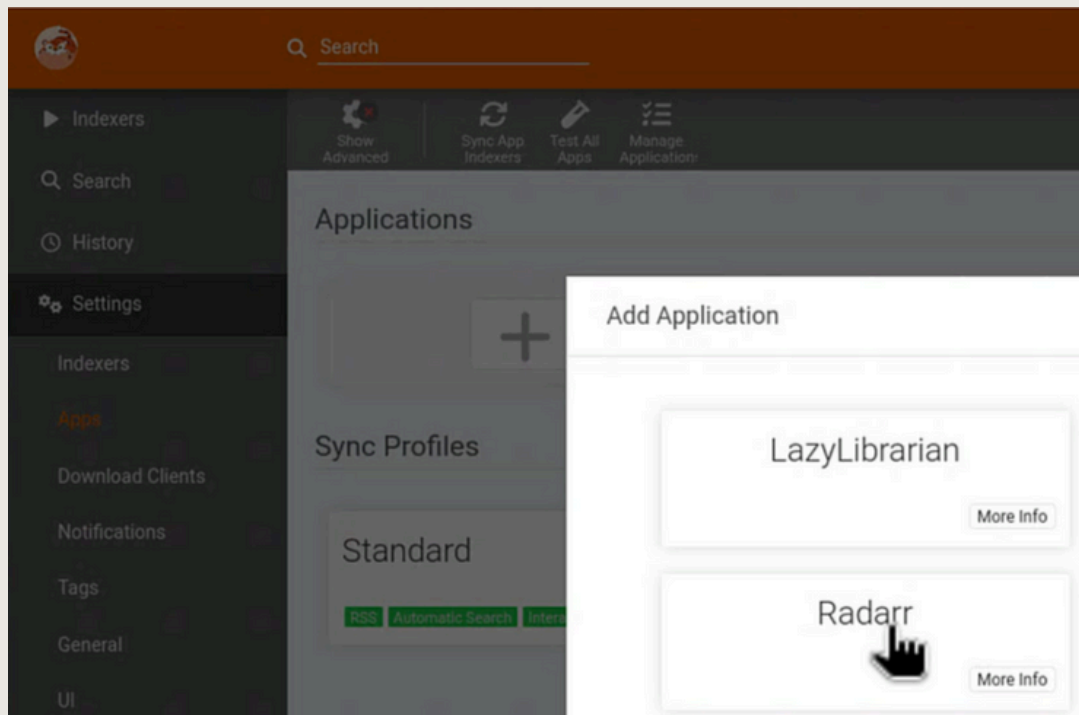
Copy API key and go back to Prowlarr.

In Settings - Apps -click little + and add Radarr as your App and paste API key.

Then change Prowlarr Server to `http://prowlarr:9696`

and Radarr Server to `http://radarr:7878`

Click Test and if Green - Save

A screenshot of the configuration form for the Radarr application. It includes the following fields and instructions:

- Tags:** A text input field. Below it, text explains that syncers are filtered by tags and provides a warning: "Tags should be used with caution, they can have unintended effects. An app with a tag will only sync with indexers having the same tag."
- Prowlarr Server:** A text input field containing `http://prowlarr:9696`. Below it, text explains: "Prowlarr server URL as Radarr sees it, including http(s)://, port, and urlbase if needed".
- Radarr Server:** A text input field containing `http://radarr:7878`. Below it, text explains: "URL used to connect to Radarr server, including http(s)://, port, and urlbase if required".
- API Key:** A text input field containing `5d5c4a313a64407888918edc3a646907`. Below it, text explains: "The ApiKey generated by Radarr in Settings/General".

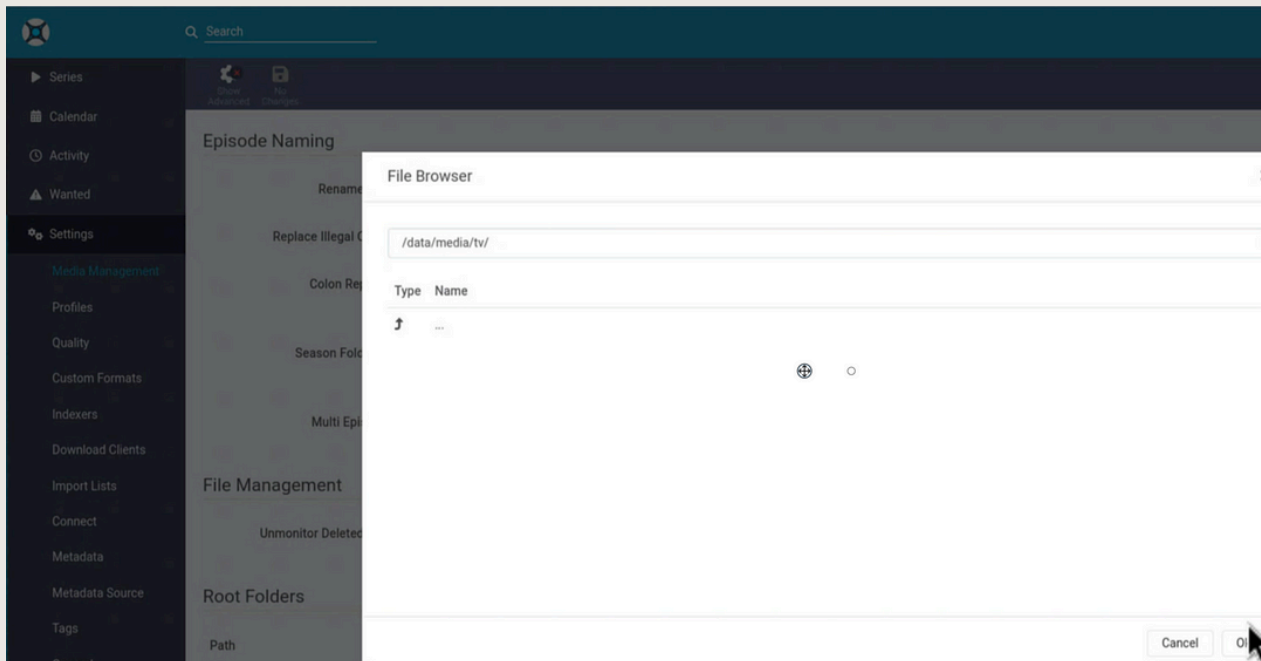
At the bottom, there is a 'Test' button with a green checkmark icon, a 'Cancel' button, and a 'Save' button. A mouse cursor is clicking on the 'Test' button.



Now configure Sonarr, steps are very similar:

Go to `http://<host ip>:8989`

Go to Settings - Media Management - Add Root Folder - set `/data/media/tv` as Root folder



Still in Settings - Media Management - Show Advanced - Importing - Use Hardlinks instead of Copy - make sure its 'ticked'

Optional - you can also tick Rename Episodes and Delete empty Folders - delete empty series and season folders during disk scan

Then in Import Extra Files - make sure that box is ticked and in Import Extra files field type srt,sub,nfo (those 3 changes are all optional)

Then Settings- Download clients - click plus symbol, choose qBittorrent etc - basically same steps as for previous services with main difference being Category you choose. Host qbittorrent, port 8080, ,make sure SSL is unticked, username admin and password - one you configured for qBittorrent and change the Category to 'tv' (by default its 'tv-sonarr', but you need to match qbittorrent Category)

Add Download Client - qBittorrent

Name

qBittorrent

Enable

☒

Host

qbittorrent

Port

8080

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

admin

Password

••••••••••••••••


Category


tv

Adding a category specific to Sonarr avoids conflicts with unrelated non-Sonarr downloads. Using a category is optional, but strongly recommended.

Post-Import Category

Category for Sonarr to set after it has imported the download. Sonarr will not remove torrents in that category





Cancel

Save

Now go to Settings - General - scroll down to API key - Copy API key - go back to Prowlarr - Settings - Apps -click '+' - Sonarr - paste API key.

Then change Prowlarr Server to `http://prowlarr:9696`

and Sonarr Server to `http://sonarr:8989`

Click Test and if Green - Save - basically you need to repeat the exact steps as we did for Radarr.

Configure Lidarr the exact same way you did with Radarr and Sonarr.

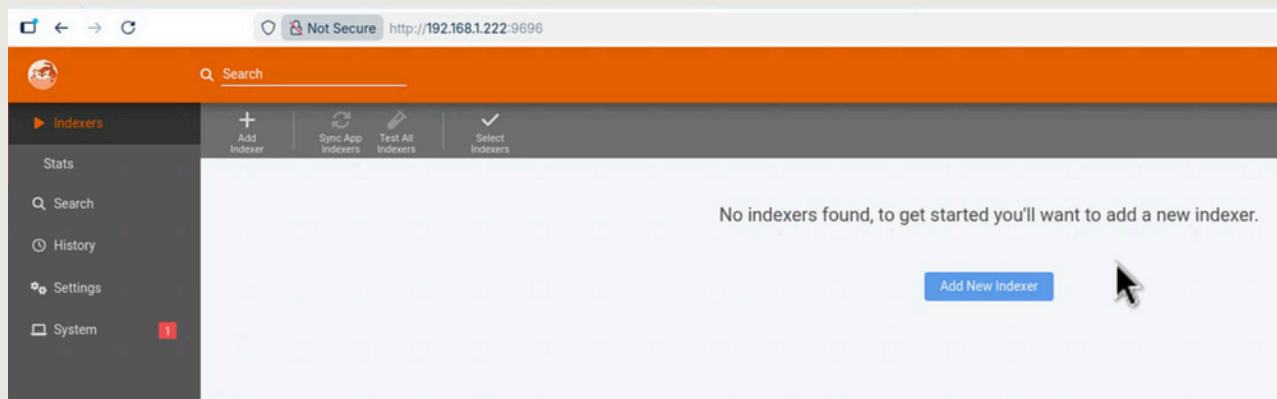
Now its a good time to restart the services and see if they come back up as expected, so run :

`sudo docker compose down`

`sudo docker compose up -d`

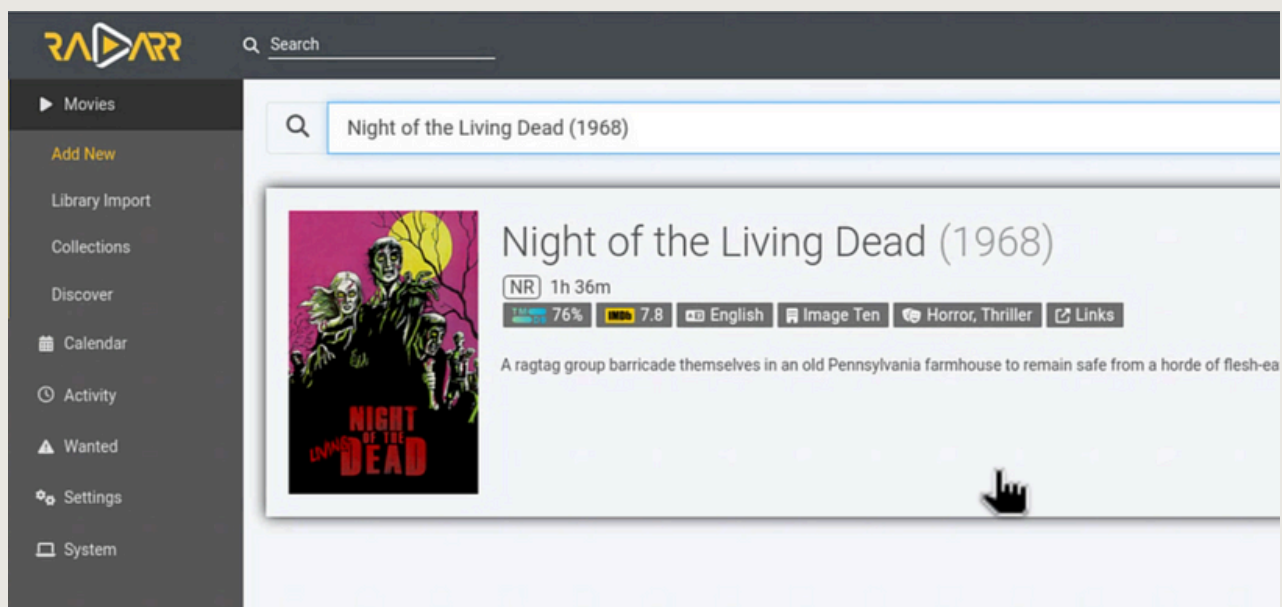
```
marek@new-arr:/data$ sudo docker compose down
[+] down 8/8
✓ Container radarr      Removed
✓ Container bazarr      Removed
✓ Container prowlarr    Removed
✓ Container sonarr      Removed
✓ Container qbittorrent Removed
✓ Container lidarr      Removed
✓ Container jellyfin    Removed
✓ Network arr_network   Removed
marek@new-arr:/data$
marek@new-arr:/data$
marek@new-arr:/data$ sudo docker compose up -d
[+] up 8/8
✓ Network arr_network   Created
✓ Container bazarr      Created
✓ Container prowlarr    Created
✓ Container qbittorrent Created
✓ Container jellyfin    Created
✓ Container radarr      Created
✓ Container sonarr      Created
✓ Container lidarr      Created
```

Next step is to add some indexers to Prowlarr:  
Go to Prowlarr - Indexer, Add new indexer:

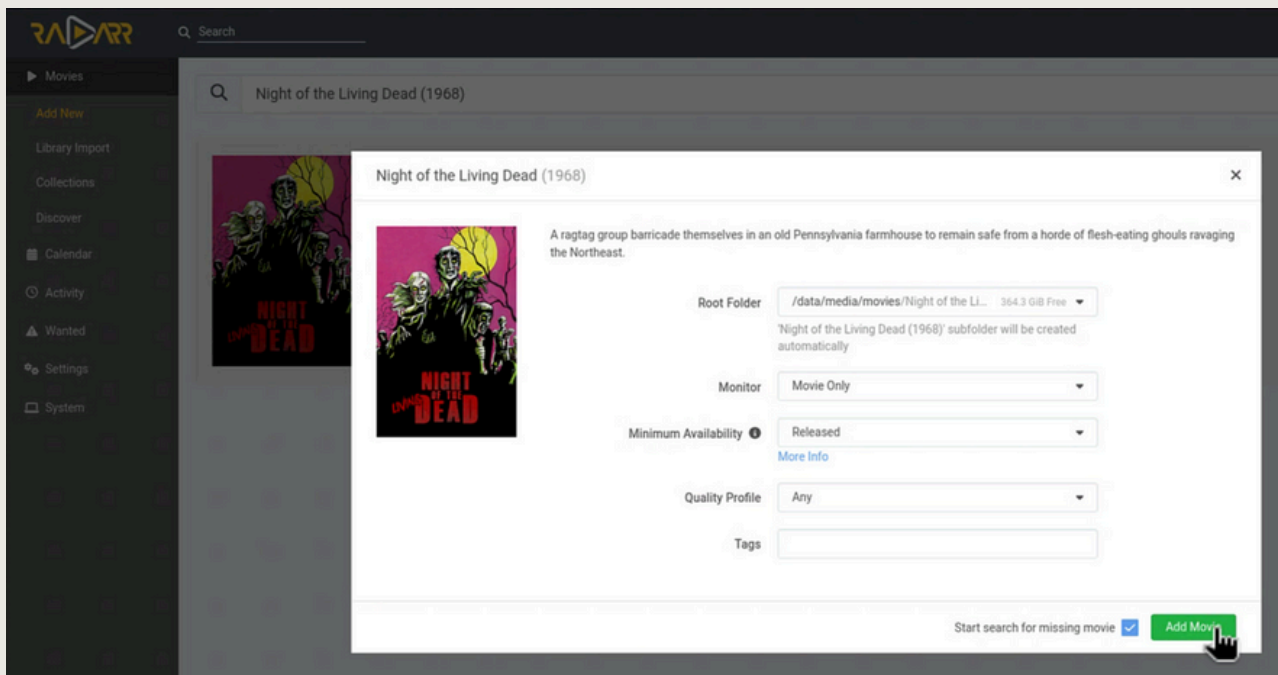


You might want to google for good prowlar indexers.  
Once you have them configured your ARR stack should be fully up and running so you can test by downloading a video in Radarr.  
Please remember to only use ARR stack for legal copyright free / Public Domain videos like for example Night of the Living Dead from 1968:

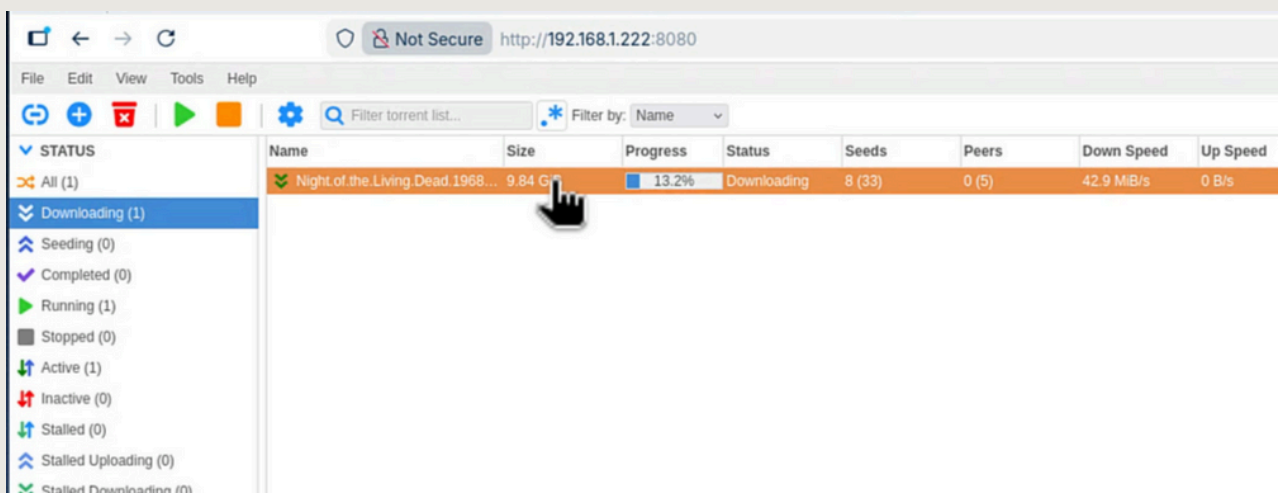
Search for it in Radarr:



Click on it and then 'Add movie'



Your download should start automatically:



Once downloaded, you should see it in your folder structure like that:

```
marek@new-arr:/data$ tree
.
├── docker-compose.yml
├── media
│   ├── movies
│   │   └── Night of the Living Dead (1968)
│   │       ├── Night of the Living Dead (1968) Bluray-1080p.mkv
│   │       ├── Night of the Living Dead (1968) Bluray-1080p.nfo
│   │       └── Night of the Living Dead (1968) Bluray-1080p.sub
│   ├── music
│   └── tv
├── torrents
│   ├── movies
│   │   └── Night.of.the.Living.Dead.1968.REMASTERED.1080p.BluRay.X264-AMIABLE[rarbg]
│   │       ├── Night.of.the.Living.Dead.1968.REMASTERED.1080p.BluRay.X264-AMIABLE.mkv
│   │       ├── night.of.the.living.dead.1968.remastered.1080p.bluray.x264-amiable.nfo
│   │       ├── RARBG.txt
│   │       └── Subs
│   │           ├── Night.of.the.Living.Dead.1968.REMASTERED.1080p.BluRay.X264-AMIABLE.idx
│   │           └── Night.of.the.Living.Dead.1968.REMASTERED.1080p.BluRay.X264-AMIABLE.sub
│   ├── music
│   └── tv
└── 12 directories, 9 files

marek@new-arr:/data$ du -sch *
4.0K   docker-compose.yml
9.9G   media
76K    torrents
9.9G   total
```

While the size already suggests the hardlinking works as expected, you can further check it with `ls -li` command to see if they point to the same inode.

You can see below that both source and destination files point to 3145740 and 3145745 inodes, which proves they are hard linked

```
marek@new-arr:/data$ ls -li /data/media/movies/Night\ of\ the\ Living\ Dead\ \ (1968\)/
3145740 'Night of the Living Dead (1968) Bluray-1080p.mkv' 3145742 'Night of the Living Dead (1968)
3145745 'Night of the Living Dead (1968) Bluray-1080p.nfo'
marek@new-arr:/data$ ls -li /data/torrents/movies/Night.of.the.Living.Dead.1968.REMASTERED.1080p.BluF
3145740 Night.of.the.Living.Dead.1968.REMASTERED.1080p.BluRay.X264-AMIABLE.mkv 3145743 RARBG.txt
3145745 night.of.the.living.dead.1968.remastered.1080p.bluray.x264-amiable.nfo 3145741 Subs
```

That should be it - enjoy!