

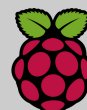
Raspberry Pi Media Server

A home media server project with live monitoring,
remote access and file sharing.





Pre-requisites	2
<input type="checkbox"/> Hardware	2
<input type="checkbox"/> Software	2
<input type="checkbox"/> Installing the OS	2
<input type="checkbox"/> Connecting to the Raspberry Pi	3
<input type="checkbox"/> Raspbian setup	3
Docker engine	4
<input type="checkbox"/> Install using the convenience script	4
<input type="checkbox"/> Install Compose	5
<input type="checkbox"/> Manage Docker as a non-root user	5
Getting started	5
• Tools	6
<input type="checkbox"/> Portainer	6
<input type="checkbox"/> Samba	6
• Monitoring	7
<input type="checkbox"/> Prometheus and Node Exporter	7
<input type="checkbox"/> Grafana	7
<input type="checkbox"/> Internet Monitoring	8
• Seedbox	9
<input type="checkbox"/> Deluge	10
<input type="checkbox"/> Jackett	10
<input type="checkbox"/> Radarr and Sonarr	12
<input type="checkbox"/> Jellyfin	12
• Web	13
<input type="checkbox"/> Nginx Proxy Manager	13
Conclusion	15



Pre-requisites

The main idea of this project is to create a Raspberry Pi home media server, with the least setup possible. Everything is containerized using Docker and written into multiple docker-compose files for easy building. The project is separated into three stages. Raspberry setup, Docker installation and services deployment.

Hardware

For this project I used a [Raspberry Pi 4](#) with 4Gb of memory and a 64Gb MicroSD card for external storage. I also bought an [aluminium case](#) with dedicated passive heatsinks for preventing overheats. In terms of connectivity I used a wired connection (Ethernet) to the Pi's, but you can use a USB Wi-Fi adapter if you don't have access to one.

Software

The main purpose of this project was to create a home server. That is why I used the Raspberry Pi OS Lite image, which is lightweight and has no GUI. I used [SSH](#) to access my Pi through my desktop computer.

Installing the OS

This step consists of getting the OS image flashed into your SD card. I personally like to use a tool called balenaEtcher, which is basically used for writing image files such as .iso and .img, onto storage media to create live SD cards and USB flash drives.

1. Download [balenaEtcher](#).
2. Run the program.
3. Go to [Raspberry's software](#) section and search for the Raspberry Pi OS Lite download button.
4. Copy link address by right clicking over it.
5. Go back to the balenaEtcher app and click on **Flash from URL**.
6. Paste the OS link address.
7. Select your SD card as target.
8. Click on **Flash**.
9. Wait till it finishes writing the card.

Before inserting the card into the Pi, go to your file explorer and navigate to the SD card boot partition. Once there, create a file named **ssh**, without any extension. This will enable [SSH](#) when the computer is first booted.

Now you can insert the card into the Raspberry Pi, connect the ethernet cable and the power supply.



Connecting to the Raspberry Pi

Wait a couple of minutes till the Pi finishes booting. Once ready, go to your main computer and open your terminal (Powershell for Windows users) and run

```
ssh pi@<RASPBERRY_IP>
```

You can find its IP in your Router's DHCP lease allocation table

Type **yes** to establish connection and enter the default Raspberry's password which is **raspberry**. You are now connected to the Pi.

Raspbian setup

The first thing that I recommend doing is changing the default password that we previously entered. Simply run

```
sudo raspi-config
```

to enter the configuration menu. Once there navigate to **System Options > Password** and press Enter. Type your new password when asked.

You can now update and upgrade your packages. To do that, run

```
sudo apt update && sudo apt full-upgrade
```

Another convenient thing to do is configure your Raspberry Pi to allow access from another computer without needing to provide a password each time you connect. To do this, you need to use an SSH key instead of a password. To generate an SSH key:

1. Log out from the current SSH session by running

```
logout
```

2. On your local machine run

```
ssh-keygen
```

Upon entering this command, you will be asked where to save the key. Just press Enter to save it on its default location.

3. Append the public key to your **authorized_keys** file on the Raspberry Pi by sending it over SSH:

```
ssh-copy-id <USERNAME>@<RASPBERRY_IP>
```

Done. Try reconnecting to your Pi and see if it works. If you are having trouble make sure to follow the [original documentation](#).



One last thing to do, which is optional, is to update the ssh configuration to increase linux security. On your Raspberry open the **/etc/ssh/sshd_config** file with your favorite text editor.

```
sudo nano /etc/ssh/sshd_config
```

Once in, update the following configuration:

```
# SSH protocol uses port 22 by default, so it will be convenient to
change it to something else. If you do so, don't forget to add the -p
<NEW-PORT> flag when trying to ssh after configuration changes.
Port <NEW-PORT>

#IPV4 only
AddressFamily inet

PermitRootLogin no
PasswordAuthentication no
PermitEmptyPasswords no
```

Save the file and run the following command to restart the sshd server.

```
sudo systemctl restart sshd
```

Before logging out from the current session, try reconnecting to your Pi using another terminal tab on your local machine. If you get an error make sure you don't misspelled anything when changing the config file. Otherwise you are good to go!

Docker engine

Docker is a set of platform as a service (PaaS) products that use OS-level virtualization to deliver software in packages called containers. Containers are isolated from one another and bundle their own software, libraries and configuration files; they can communicate with each other through well-defined channels.

If you are having trouble running the following commands, go and check the original docker documentation.

1. [Install using the convenience script](#)
2. [Install Compose](#)
3. [Manage Docker as a non-root user](#)

Just copied and pasted the necessary parts for easy access.

Install using the convenience script

Docker provides a convenience script at get.docker.com to install Docker into development environments quickly and non-interactively.



This example downloads the script from get.docker.com and runs it to install the latest stable release of Docker on Linux:

```
curl -fsSL https://get.docker.com -o get-docker.sh
sudo sh get-docker.sh
```

Docker is installed. The docker service starts automatically on Debian based distributions.

Install Compose

On Linux, you can download the Docker Compose binary from the Compose repository release page on GitHub. Follow the instructions from the link, which involve running the curl command in your terminal to download the binaries. These step-by-step instructions are also included below.

1. Run this command to download the current stable release of Docker Compose:

```
sudo curl -L
"https://github.com/docker/compose/releases/download/1.29.2/docker-co
mpose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
```

2. Apply executable permissions to the binary:

```
sudo chmod +x /usr/local/bin/docker-compose
```

Manage Docker as a non-root user

The Docker daemon binds to a Unix socket instead of a TCP port. By default that Unix socket is owned by the user root and other users can only access it using sudo. The Docker daemon always runs as the root user.

If you don't want to preface the docker command with sudo, create a Unix group called docker and add users to it. When the Docker daemon starts, it creates a Unix socket accessible by members of the docker group.

To create the docker group and add your user:

1. Create the docker group.

```
sudo groupadd docker
```

2. Add your user to the docker group.

```
sudo usermod -aG docker pi
```



Getting started

If you successfully config and installed the pre-requisites and your docker engine is running as expected you are now ready to clone the repo. Just SSH into your raspberry, make sure you are in the **/home/pi** directory, and run the following command.

```
git clone https://github.com/LucasACH/raspberry-pi-media-server
```

Perfect! Before running the docker containers you will need to make some configuration. I created an **.env** file inside each folder that you will need to fill up with your custom data.

Tools

In this stack I included two services. The first one is called Portainer, which is an universal container management tool that helps users deploy and manage container-based applications without needing to know how to write any platform-specific code, and the second one is known as Samba, which allows file and print sharing between computers running Microsoft Windows and computers running Unix.

To create the Tools stack:

1. Run the following commands.

```
cd tools/  
sudo nano .env
```

2. Fill the variables with your custom data.

```
USER=pi #your user name  
TZ=America/Argentina/Buenos_Aires #Search for your time zone:  
https://en.wikipedia.org/wiki/List\_of\_tz\_database\_time\_zones
```

3. Create the stack by running the following command.

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

Portainer

To access Portainer, open your web browser and go to **http://<RASPBERRY_IP>:9000/**. You can now follow the [initial setup](#) instructions.

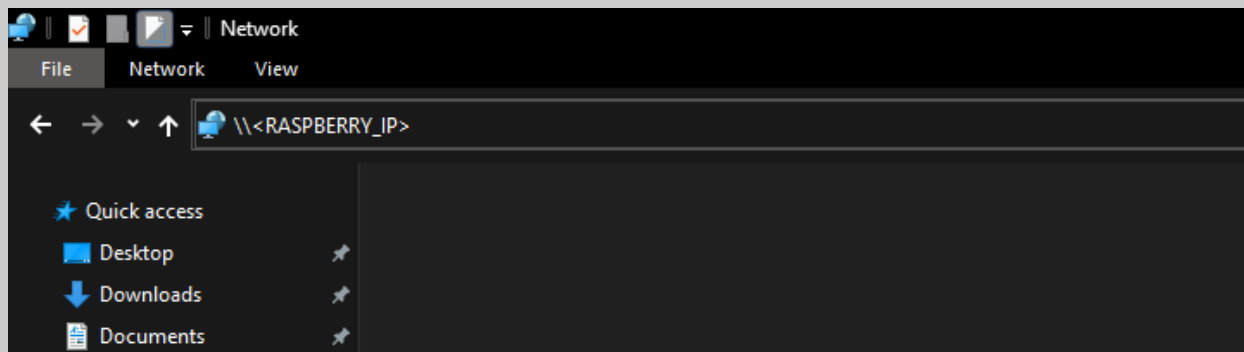
Samba

This way of connecting to the samba server running on your Pi is meant to work on Windows machines, although is quite similar when attempting from another OS.

1. Open Windows Explorer.



2. In the path bar, write `\\<RASPBERRY_IP>` (pay attention to the backward slashes).



3. That should create a new entry in **Network** and show its content.

Samba is going to be useful to access downloaded media.

Monitoring

In this stack I included three services. The first one is called Prometheus, which is a free software application used for event monitoring and alerting. It records real-time metrics in a time series database (allowing for high dimensionality) built using a HTTP pull model, with flexible queries and real-time alerting. The second one is known as Grafana, a multi-platform open source analytics and interactive visualization web application. It provides charts, graphs, and alerts for the web when connected to supported data sources. Finally, a Prometheus Node Exporter container will run in the background and expose a wide variety of hardware- and kernel-related metrics.

To create the Monitoring stack:

1. Run the following commands.

```
cd monitoring/  
sudo nano .env
```

2. Fill the variables with your custom data.

```
USER=pi # your user name
```

3. Create the stack by running the following command.

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

Prometheus and Node Exporter

You don't need to do extra configuration. It runs out of the box!

Grafana

To access Grafana, open your web browser and go to **`http://<RASPBERRY_IP>:3030/`**. A sign in form should appear. The default credentials are **admin admin**. Once in, make



sure to change your username and password. To see your dashboards navigate to **http://<RASPBERRY_IP>:3030/dashboards**. For now, the only one that will display data is the *System Monitoring* dashboard.



Internet Monitoring

This step is optional. I found this cool [repo](#) for monitoring your internet speed. Check it out for more details.

To create the Internet monitoring stack:

1. Run the following commands.

```
cd internet/  
sudo nano .env
```

2. Fill the variables with your custom data.

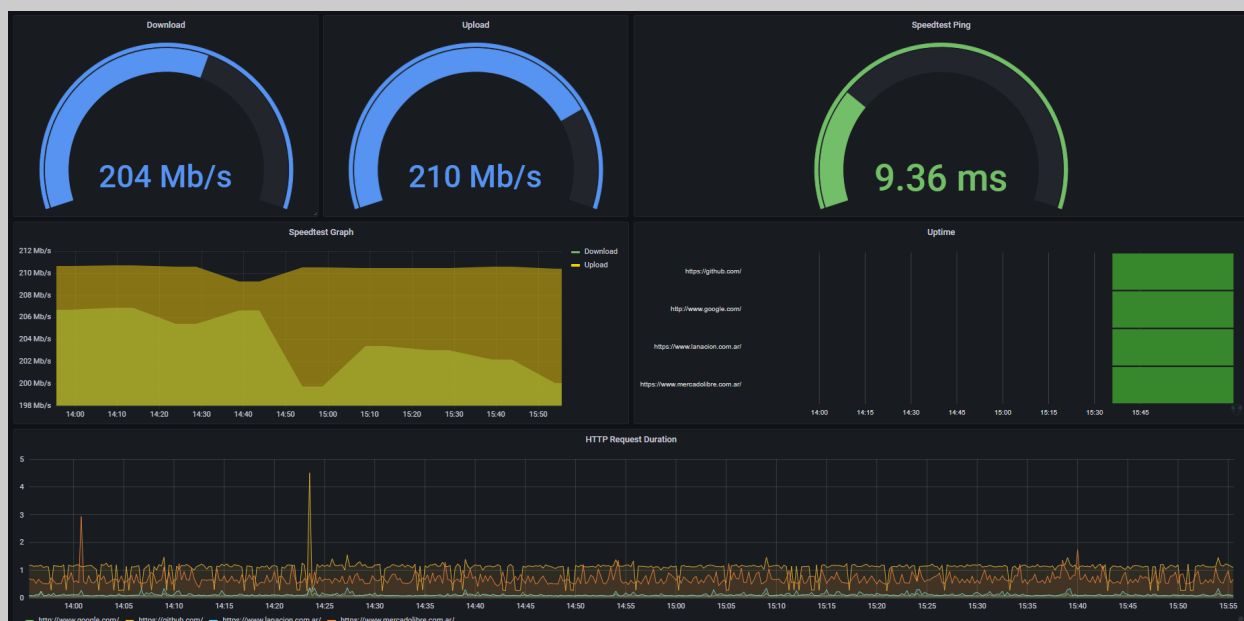
```
USER=pi # your user name
```

3. Create the stack by running the following command.



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

Now go to your Grafana dashboards tab and open **Internet connection**. After some time, the dashboard should look something like this.



You can activate or deactivate the internet monitoring tool just by running or stopping the two docker containers (**monitoring-internet-ping** and **monitoring-internet-speedtest**).

Seedbox

In this stack I included five services. The first one is called Deluge, which is a free and open-source, cross-platform BitTorrent client written in Python. The second one is known as Jackett. It works as a proxy server: it translates queries from apps (Sonarr, Radarr, SickRage, CouchPotato, Mylar, Lidarr, DuckieTV, qBittorrent, Nefarious etc.) into tracker-site-specific http queries, parses the html response, then sends results back to the requesting software. Radarr and Sonarr are media collection managers for Usenet and BitTorrent users. They can monitor multiple RSS feeds for new movies (Radarr) or shows (Sonarr) and will interface with clients and indexers to grab, sort, and rename them. The last service is called Jellyfin, a volunteer-built media solution that puts you in control of your media. Stream to any device from your own server, with no strings attached.

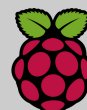
To create the Seedbox stack:

1. Run the following commands.

```
cd seedbox/  
sudo nano .env
```

2. Fill the variables with your custom data.

```
USER=pi # your user name  
PUID=1000 # your PUID
```



```
PGID=1000 # your PGID
TZ= # Search for your time zone:
https://en.wikipedia.org/wiki/List_of_tz_database_time_zones
```

3. Make the data directory for all the downloaded media and assign them to the pi user to avoid permission issues in the future.

```
sudo mkdir -p /data/torrents /data/movies /data/tv
sudo chown pi:pi /data/torrents /data/movies /data/tv
```

4. Create the stack by running the following command.

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

Deluge

To access Deluge, open your web browser and go to ***http://<RASPBERRY_IP>:8112/***. The default user and password are **admin deluge**. To change the password (recommended) log in to the web interface and go to **Preferences > Interface > Password**. Then change the downloads location to **/downloads/torrents**. You can find this setting on **Preferences > Downloads**.

You will need to enable Labeling for Radarr and Sonarr to work. Go to **Preferences > Plugins** and check the **Label** checkbox. Once ready, apply changes and click OK.

You're all set! Try adding manually a torrent and see if everything works fine.

Jackett

To access Jackett, open your web browser and go to ***http://<RASPBERRY_IP>:9117/***. As always, creating a password is recommended. You can do it on the Jackett Configuration section. Start adding your indexers by clicking on the **Add indexer** button.

Radarr and Sonarr

You can access Radarr on port **7878** and Sonarr on port **8989**. The configuration is similar for both services. Before adding your selected indexers and download client (Deluge), specify the root folder for the downloaded media. Simply go to **Settings > Media Management** and click on **Add Root Folder** button. The path for Radarr is **/downloads/movies** and the one for Sonarr is **/downloads/tv**.

To add an indexer, go to **Settings > Indexers** and click on the **+** sign. Then click on Custom Torznab and fill up the fields. To get indexers URL go to Jackett, locate the indexer you want to add and click on Copy Torznab Feed. It should look something like this ***http://<RASPBERRY_IP>:9117/api/v2.0/indexers/<INDEXER>/results/torznab/***. You can also replace your raspberry ip with your jackett container name (jackett).



Add Indexer - Torznab ✕

Name

YTS

Enable RSS

☒

Enable Automatic Search

☒ Will be used when automatic searches are performed via the UI or by Radarr

Enable Interactive Search

☒ Will be used when interactive search is used

URL

http://jackett:9117/api/v2.0/indexers/yts/results/torzna

API Key

YOUR JACKETT API KEY

Categories

2000 2010 2020 2030 2040 2045 2050 2060

▼

Drop down list, leave blank to disable all categories

Test

Cancel

Save

Now for the download client go to **Settings > Download Clients**, click the **+** sign and select Deluge. If you get an error when setting up the host as **deluge**, try replacing it with your Pi's ip address. The password should be the one you used for Deluge. If everything went well, you should get a green thick when clicking on Test.



Add Download Client - Deluge

Name

Deluge

Enable

☒

Host

deluge

Port

8112

Use SSL

☐ Use secure connection when connecting to Deluge

Password

.....

Category

movies

Recent Priority

Last (0) ▾

Older Priority

Last (0) ▾

Add Paused

☐

Adding a category specific to Radarr avoids conflicts with unrelated downloads, but it's optional

Priority to use when grabbing movies that aired within the last 14 days

Priority to use when grabbing movies that aired over 14 days ago

Test

Cancel

Save

Great! You can now start downloading movies by clicking on **Movies > Add New**. Follow the same steps for setting up Sonarr. Make sure to add **tv** category instead of **movies**, when adding Deluge downloading client.

Jellyfin

To access Jellyfin, open your web browser and go to **http://<RASPBERRY_IP>:8096/**. The setup is pretty straight forward.



Web

In this stack I included two services. The first one is called DuckDNS, which is a free dynamic DNS hosted on Amazon VPC and the second one is known as Nginx Proxy Manager, a software designed to manage a computer network's proxy servers, write and implement their policies and filter resource requests.

To create the Web stack:

1. Run the following commands.

```
cd web/  
sudo nano .env
```

2. Create a [DuckDNS](#) account and add your subdomains.
3. Fill the variables with your custom data.

```
USER=pi #your user name  
PUID=1000 #your PUID  
PGID=1000 #your PGID  
TZ=America/Argentina/Buenos_Aires #  
https://en.wikipedia.org/wiki/List_of_tz_database_time_zones  
SUBDOMAINS=subdomain1,subdomain2 #your DuckDNS subdomain/s  
DUCKDNS_TOKEN=token #your DuckDNS token  
DB_MYSQL_USER=npn #database user  
DB_MYSQL_PASSWORD=npn #database password
```

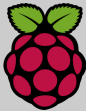
4. Create the stack by running the following command.

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

Nginx Proxy Manager

To access Nginx Proxy Manager, open your web browser and go to **http://<RASPBERRY_IP>:81/**. You should be able to see a sign in form. The default credentials are **admin@example.com changeme**. To access your Raspberry from outside your local network you will need to create two Port Forwarding rules into your router's configuration. Port 80 and 443 should point to your Pi's ip address.

Perfect! You can now start adding your proxy hosts. Go to the Nginx Proxy Manager running on the Pi and navigate to the **Proxy Hosts** section. Once in, click on **Add Proxy Host**. You will need to specify the domain name created with DuckDNS, your Raspberry's IP address and the port you want to forward to. For example, if you want to show the Grafana dashboard add port 3030.



New Proxy Host

Details

Custom locations

SSL

Advanced

Domain Names *

subdomain.duckdns.org

Scheme *

http

Forward Hostname / IP *

YOUR PI'S IP ADDRESS

Forward Port *

3030

Cache Assets

Block Common Exploits

Websockets Support

Access List

Publicly Accessible

Cancel

Save

You can also create an SSL Certificate for encrypting data transfers between client and host. Simply go to the SSL tab and request a new SSL Certificate.

New Proxy Host

Details

Custom locations

SSL

Advanced

SSL Certificate

Request a new SSL Certificate

Force SSL

HTTP/2 Support

HSTS Enabled

HSTS Subdomains

Use a DNS Challenge

Email Address for Let's Encrypt *

YOUR EMAIL

I Agree to the Let's Encrypt Terms of Service *

Cancel

Save

If everything went well, you should be able to access your desired services from outside your local network, by browsing your DuckDNS domain name.



Conclusion

If you successfully complete every step listed in this documentation you should have running:

- Poratiner for controlling your docker container with a friendly GUI.
- Samba for accessing Raspberry Pi's share folder from another computer.
- Grafana and Prometheus for viewing live metrics of your hardware.
- Radarr and Sonarr for downloading media using torrents technology.
- Nginx Proxy Manager for accessing your self hosted services from outside your local network through the internet.