



How to turn a Raspberry Pi into a file server?

Have you always dreamed of having a family NAS with your backups and movies? But you do not necessarily have the budget for that? Raspberry Pi allows you to do that at a lower cost

How to turn your Raspberry Pi into a file server?

You must install and configure Samba on Raspbian or any other Linux distribution. You can then turn it into a media server to play your movies directly from your TV

We will now see in detail a typical installation, i.e., how to install Samba on Raspbian and then configure it.

And then we'll see how to enable DLNA to make your Raspberry Pi accessible from your TV.

I will also give you some tips to improve this installation

[Table of Contents](#)



Why use a Raspberry Pi as a file server?

File server usage

A file server is used to store and **share files on the local network**.

This server allows you, for example, to create automatic backups of your computer on this server

You can also store files that take up space on your computer but are not critical:

- movies
- software
- iso files

Raspberry Pi advantages

A standard NAS server, will cost you more than \$300 for the entry level, make noise and consume a lot of electricity

A Raspberry Pi will cost you less than 50 € and wastes almost no power

You will eventually need to add the price of a hard drive if you need a lot of space, but it is reasonable

This allows you to add a small backup storage server at a lower cost while learning how it works

Add disk space to Raspberry Pi

If you follow this guide only for testing, you do not necessarily need to read this paragraph

It concerns people who do not have enough space on their SD card and who want to have more free disk space

A larger SD card

The first possibility is to buy a larger capacity SD card

You may have an SD card of 8 or 16 GB at the moment, so it's going to be a bit limited
But there are SD cards that go up to 256GB or more and for a reasonable price.
It may be worthwhile to buy a larger SD card before following this tutorial

Feel free to check [my recommended products page](#) if you want to know which SD card I recommend

If you choose this, there is nothing else to do technically than replacing the SD card in the drive. And then resume the rest of this post

USB hard drive

Another possibility is to connect an external USB drive to your Raspberry Pi. With the 2.5" format you should be able to find disks up to at least 2TB, which starts to be a better configuration for file sharing. It also gives you the advantage of being able to bring it with you even if you are not taking your Raspberry Pi. And so to read the content on any computer.

Most of the time your hard drive will be recognized and mounted directly by Raspbian. A mount point will be created in `/media/pi/NAME`. Where NAME depends on the name of your hard drive.

If you have any issues, I advise you to format your disk in a file format readable under Linux (EXT4 for example) and to try again. The `fdisk` and `mount` commands could also serve you in the worst cases.

A SATA hard drive?

If you want something professional, you can add a card to connect SATA hard drives to your Raspberry Pi. This card will allow you to get closer to a classic NAS with faster, better-integrated disks and the ability to manage security with software RAID.

This implementation is another project, so I let you do your research and come back here once your disks are functional 😊

Install Raspbian

So we have our pre-requisites completed:

- an SD card
- possibly additional storage

So let's move on to the installation of Raspbian which will be the basis of our file server.

Download Raspbian

First of all, you will have to **download the latest version of Raspbian** from the official website

If your goal is only to have a file server, the lite version is enough, and you'll save some disk space and CPU/ram

Anyway we will do all the rest of this tutorial with SSH, so it doesn't change

Download the image and continue

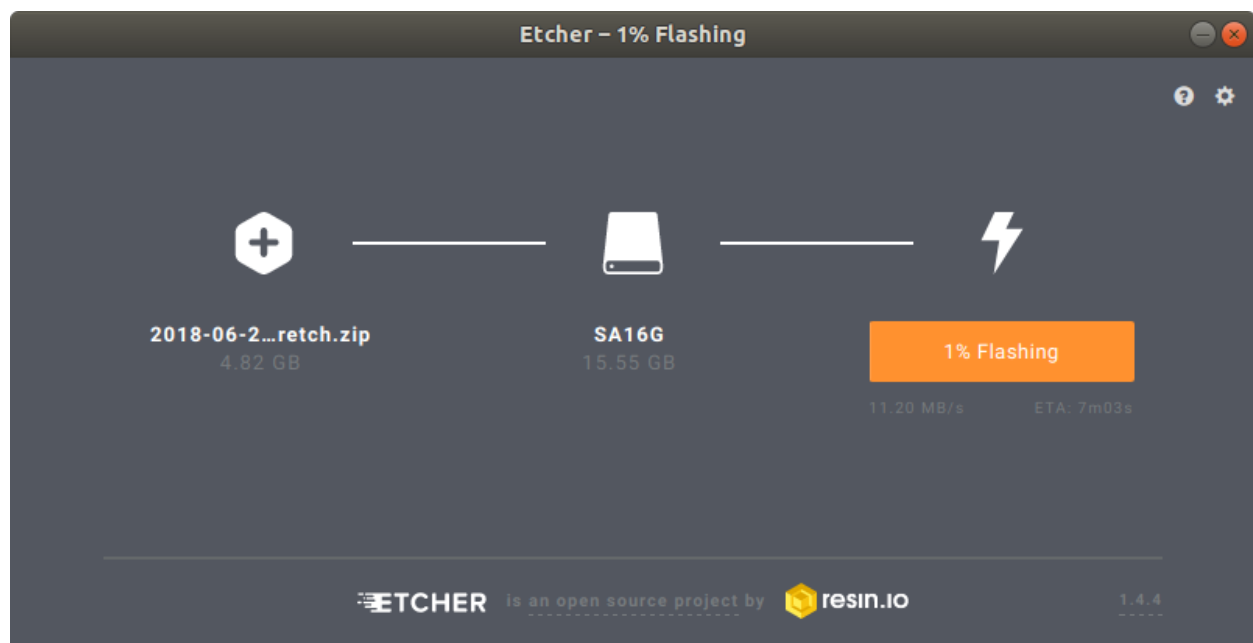
Flash an SD Card with Raspbian

I recommend using Etcher

If you have not installed it yet, get it from **the official website**

It's a tool that allows you to flash an SD card very easily on Linux, Mac, or Windows

- Start Etcher
- Select the location of the Raspbian image
- Choose your SD card
- Click on **Flash**



Once the SD card is ready, eject it and **insert in your Raspberry Pi**

First Boot

The installation is automatic, you have to start your Raspberry Pi, and Raspbian will launch

If you choose the Pi Desktop, a welcome menu will open:

- Choose your language preferences
- Change the default password
- Connect to the wifi if necessary
- Accept system updates

- Reboot the Raspberry Pi

If everything worked fine, move on

Otherwise, it will be necessary at least to succeed to connect the network and to change the password (the following paragraph can help you)

If you have the Lite version, you have to do the same thing within raspi-config

Enable SSH

SSH is a secure remote connection protocol, which allows you to launch commands from another computer on the network

By default, the service is not started on Raspberry Pi

So we will **enable SSH and connect to it** before continuing

To enable SSH, open a terminal and type this command:

```
sudo service ssh start
```

Once SSH is installed connect to it (ssh pi@IP)

If this is not easy for you, check out [this article](#) that will help you

Install the file server

Now that Raspbian is ready let's get to the goal of this tutorial: create a file share

Organization of the file system

Depending on what you initially chose as a storage type, the paths will change a little later

So I'm going to assume that our available storage space is in **/media/share**

Replace this with the right folder you have created

If you have chosen to share your files directly on the SD card, create a new folder of your choice (/media/share for example):

```
sudo mkdir /media/share
```

In other cases, you should create a mount point in /media/pi/share or something like that

Install Samba

Start by checking for updates:

```
sudo apt-get update  
sudo apt-get upgrade
```

Then install samba:

```
sudo apt-get install samba
```

Once samba installed you should be able to access the Raspberry Pi with the file explorer:

- Windows: \\X.X.X.X
- Linux/Mac: smb://X.X.X.X

Replace X.X.X.X with the IP address of the Raspberry Pi

But for now we do not see a shared folder, and we can not create files

Let's see how to fix this

Configure Samba for a guest share

Samba is now available, let's go to the configuration

The configuration will depend on what you want to do, but I'll explain how to create a basic sharing

- Give all permissions to the folder

```
sudo chmod 777 /media/share
```

- Edit the /etc/samba/smb.conf file

```
sudo nano /etc/samba/smb.conf
```

- Add these lines at the end of the file

```
[SharePi]
comment = RaspberryPi
public = yes
writeable = yes
browsable = yes
path = /media/share
create mask = 0777
directory mask = 0777
```

- Save and leave (CTRL+X)
- Restart Samba to apply changes:

```
sudo service smbd restart
```

Test access

We now have a shared folder accessible to all without authentication

To verify that this works, connect to the network drive:

- Windows: Open the file explorer (Win + E) and type the following address:
\\X.X.X.X\SharePi
- Linux/Mac: Open the file explorer and type the following address:
smb://X.X.X.X/SharePi

Make sure the folder opens, and that you can create files or folders, and then delete them without permissions issues

If so, we can move on

Permissions

For now, our share is accessible to anyone, even without a password

If you do not want your little sister to delete your backup history, you will have to forbid guest access.

Configuration

To request a password when connecting, edit the public option in `/etc/samba/smb.conf`

```
[SharePi]
comment = RaspberryPi
public = no
writeable = yes
browsable = yes
path = /media/share
create mask = 0777
directory mask = 0777
```

You must also tell samba that the user is allowed to connect to shares

```
sudo smbpasswd -a pi
```

Then restart Samba:

```
sudo service smbd restart
```

If you try to connect again, it should open an authentication window

If this is not the case check that you are not already connected in guest mode (first you must disconnect, it depends on your OS, so restart your computer if you can't find)

On Ubuntu, I had to click on the “Eject” button in front of the mount in the file explorer

Add another user

If necessary, you can add other user accounts with different passwords

Create a new user:

```
sudo adduser yourname
```

Enable for samba:

```
sudo smbpasswd -a yourname
```

You can now connect with any of the users

But remember that we put everyone's rights on the /media/share folder, so all users you create will have the same permissions

If, for example, you want only the "pi" user to have access to the SharePi share, you will have to change it like this :

```
chown pi /media/share -R  
chmod 700 /media/share -R
```

You can then create multiple shares by changing user permissions for each one

Turn it into a media server

Now that we have a network storage location, why don't we use it to manage our downloads and make it accessible from our TV?

Torrent server

You probably know the Torrent protocol, which is a well-known protocol for downloading. We will see how to install a torrent server on our Raspberry Pi to automate downloads

Create a new folder for torrents to download:

```
sudo mkdir /media/share/torrents  
sudo chmod 777 /media/share/torrents
```

Rtorrent

Start by installing rtorrent and screen:

```
sudo apt-get install rtorrent screen
```

Then create your configuration file:

```
nano ~/.rtorrent.rc
```

Paste these lines:

```
directory = /media/share/torrents
```

You can add all the options that interest you. You will [find a list here](#)

Start rtorrent:


```
screen -s /bin/bash  
rtorrent
```

Leave the screen without closing rtorrent with CTRL+A and CTRL+D

Go back to the screen with screen -r

Screen allows you to let run rtorrent even when you disconnect

Web interface

We will now add the flood interface which will allow us to manage all this from a remote computer

Firstly, you need to install npm :

```
sudo apt-get install npm
```

Then add nod-gyp mod :

```
sudo npm install -g node-gyp
```

Now, clone the git repository of the flood web interface like this:

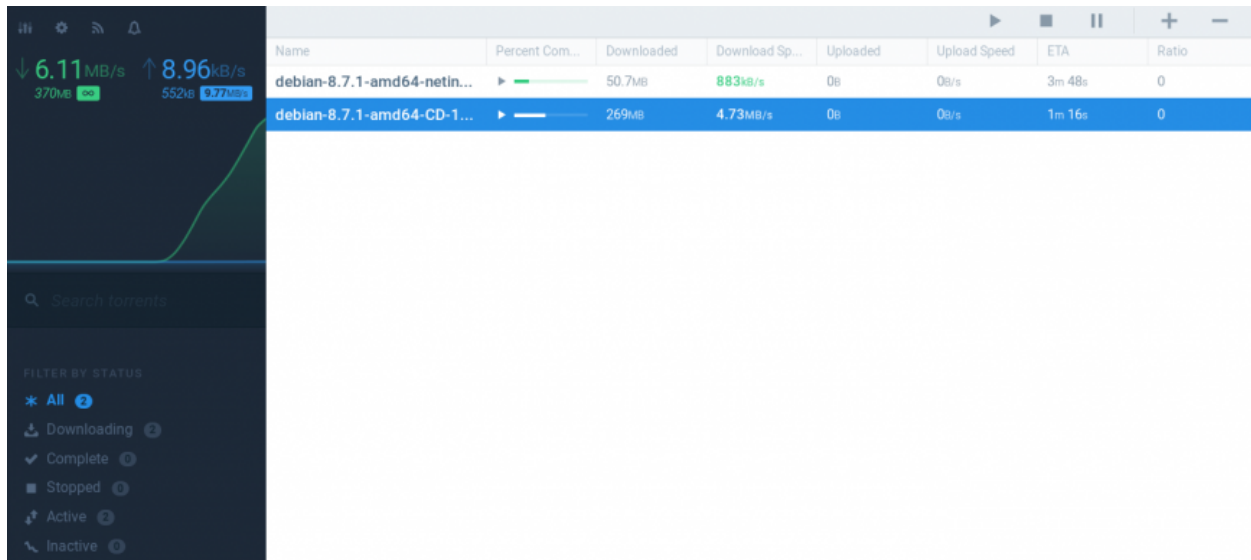
```
cd /opt  
sudo git clone https://github.com/jfurrow/flood.git
```

And run this commands to install and start it:

```
cd /opt/flood  
sudo npm install --unsafe-perm  
sudo npm run build  
sudo npm start
```

This step will take a while, drink a coffee

After the installation is complete and the application started, you can access the interface at <http://X.X.X.X:3000>



If you need more information about configuration options, you can check their [GitHub page](#)

DLNA

DLNA is a technology that allows you to play a video stream on one device from a file stored on another

Typically you can play a movie stored on your Raspberry Pi from your smart TV or PS4

The package that makes it possible on Debian is called minidlna

So we will install and configure it to stream the contents of our shared folder

Install the package:

```
sudo apt-get install minidlna
```

Edit the configuration file:

```
sudo nano /etc/minidlna.conf
```

Set the shared folder in the line that starts with media_dir=

```
media_dir=A,/media/share
```

You can set it as a subfolder like /media/share/movies if you want

Start the service

```
sudo service minidlna start
```

MiniDLNA will then scan your folder and make it available from all your compatible devices

Possible improvements

NAS distribution

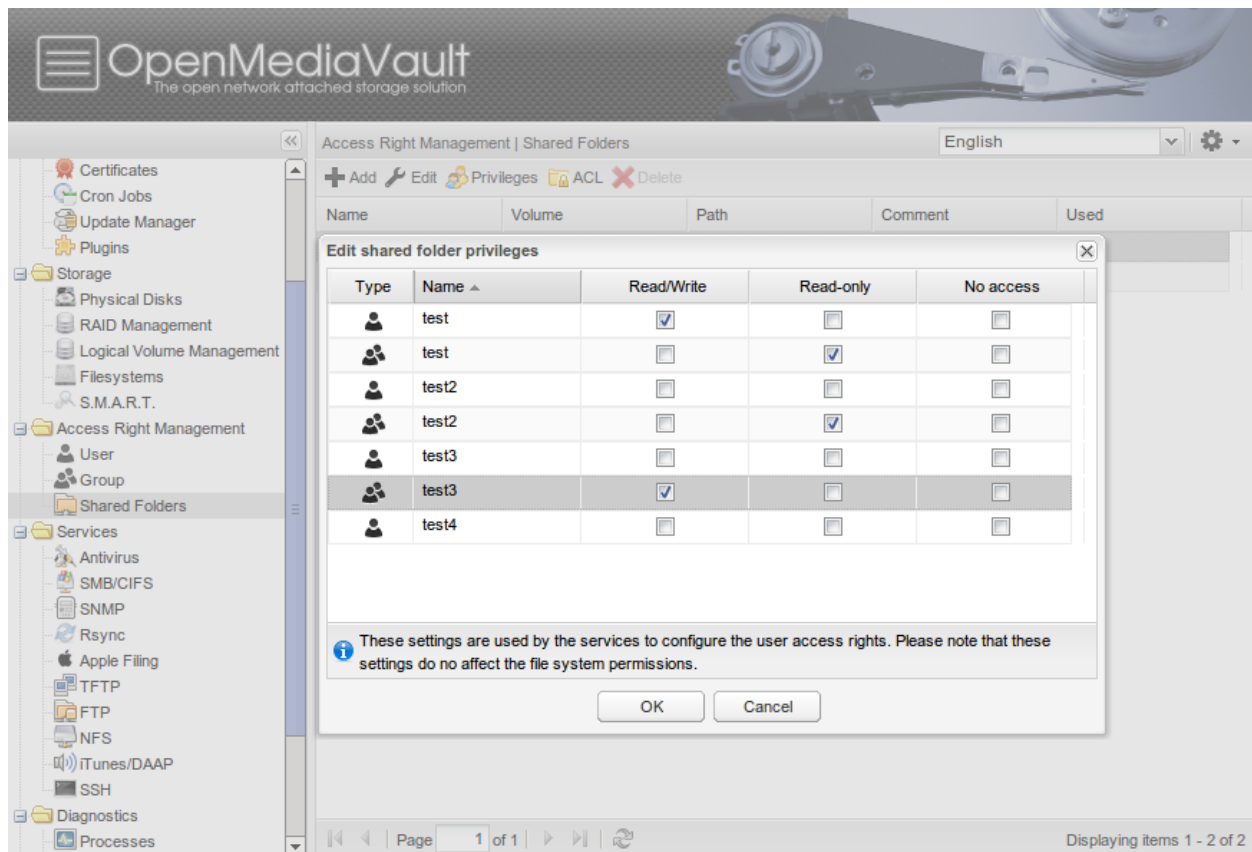
If you do not want to manage all this with complicated commands and configuration files, there is a solution that may interest you.

Indeed, there are dedicated distributions for managing a file server, which let you configure everything via a web interface.

This is the case of OpenMediaVault which is available for Raspberry Pi

You will find the images to [download on SourceForge](#)

Once installed on an SD card, you will have an interface allowing you to manage your shares, your users, etc ...



RAID or backups

We talked about using this file server as a backup in this article

You still have to avoid using it as a primary storage server

Copying all of your photos from the last ten years without storing them elsewhere does not seem like an excellent idea

Whether you have chosen the SD card or USB hard drive option, these are not very reliable storage methods

A hardware failure quickly arrives, and most of the time you will lose all content

If you want to make it a primary storage server, you should not forget about security

It is possible to install multiple disks in a RAID (with data replication) on Raspberry Pi, either in USB or with an additional SATA card

But the simplest is perhaps to have backups of this server elsewhere
One can imagine a second SD card on which is copied regularly the content of the first.
Or a second Raspberry Pi that will serve as a backup

Conclusion

I hope all this will helped you set up your file server at home

To sum up, it's quite simple to create a file sharing accessible to all on Raspberry Pi
However, it is often necessary to spend a little more time adding the required options to implement what you want successfully

In any case, you have here a good information base to create a sharing server, a torrent server or a media server

Newsletter

Signup for our monthly news and special offers!



Patrick Fromaget

I'm the lead author and owner of RaspberryTips.com.
My goal is to help you with your Raspberry Pi problems using detailed guides and tutorials.
In real life, I'm a Linux system administrator with a web developer experience.



LEAVE A REPLY

Comment

Name *

Email *

Website

POST COMMENT

[report this ad](#)

Search ...

SEARCH

Welcome to Raspberry Tips



Hi, I'm Patrick. I am a Linux system administrator, and I am passionate about the Raspberry Pi and all projects on this topic.

I created this site to share with you what I learned about it



Newsletter

Signup for our monthly news and special offers!

Subscribe

[report this ad](#)

[Home](#)

[How-To Tutorials](#)

[Recommended Products](#)

[Privacy Policy](#)

Copyright © 2019 Raspberry tips. All Rights Reserved.

Codilight Theme by [FameThemes](#)

Raspberry Pi is a trademark of The Raspberry Pi Foundation

This site is owned and operated by Patrick Fromaget. RaspberryTips.com is a participant in the Amazon Services LLC Associates Program, an affiliate advertising program designed to provide a means for sites to earn advertising fees by advertising and linking to Amazon.com. This site also participates in other affiliate programs and is compensated for

