

Build an Arch Linux file server with the Raspberry Pi

The Raspberry Pi is small, silent and energy efficient, so let's turn it into the ultimate file server!

We'll be using Arch Linux as the operating system for our file server, because it is small, and has only the minimum packages required for a working system. This means that we can set up the file server without wasting resources on a graphical user interface and other unnecessary packages. Arch Linux comes with hardly any RAM allocated to the GPU by default, which is exactly how we want it for use as a headless server.

Our file server will be made up of the following software components:

- Base Arch Linux system
- SSH – will provide secure remote access to the Raspberry Pi and the files on it
- Samba – provides access to files on the server to a Windows, Mac or Linux client
- A dynamic DNS daemon (noip) – software that runs in the background and points a domain name to your router's IP address, meaning that you can access your Raspberry Pi from anywhere using an easy-to-remember web address
- Transmission daemon – a torrent client that runs in the background and can be accessed through a web browser

This tutorial assumes that you have flashed the latest Arch Linux ARM image to an SD card. If you haven't, the instructions for flashing an image can be found on our tutorial on how to set up a [Raspberry Pi](#).

You'll only need to go up to the step where you write the image to the SD card – or if you plan to use the SD card to store files, then you should follow the GParted steps also. You'll have to adapt the instructions slightly for using the Arch Linux image rather than the Debian one.

Resources

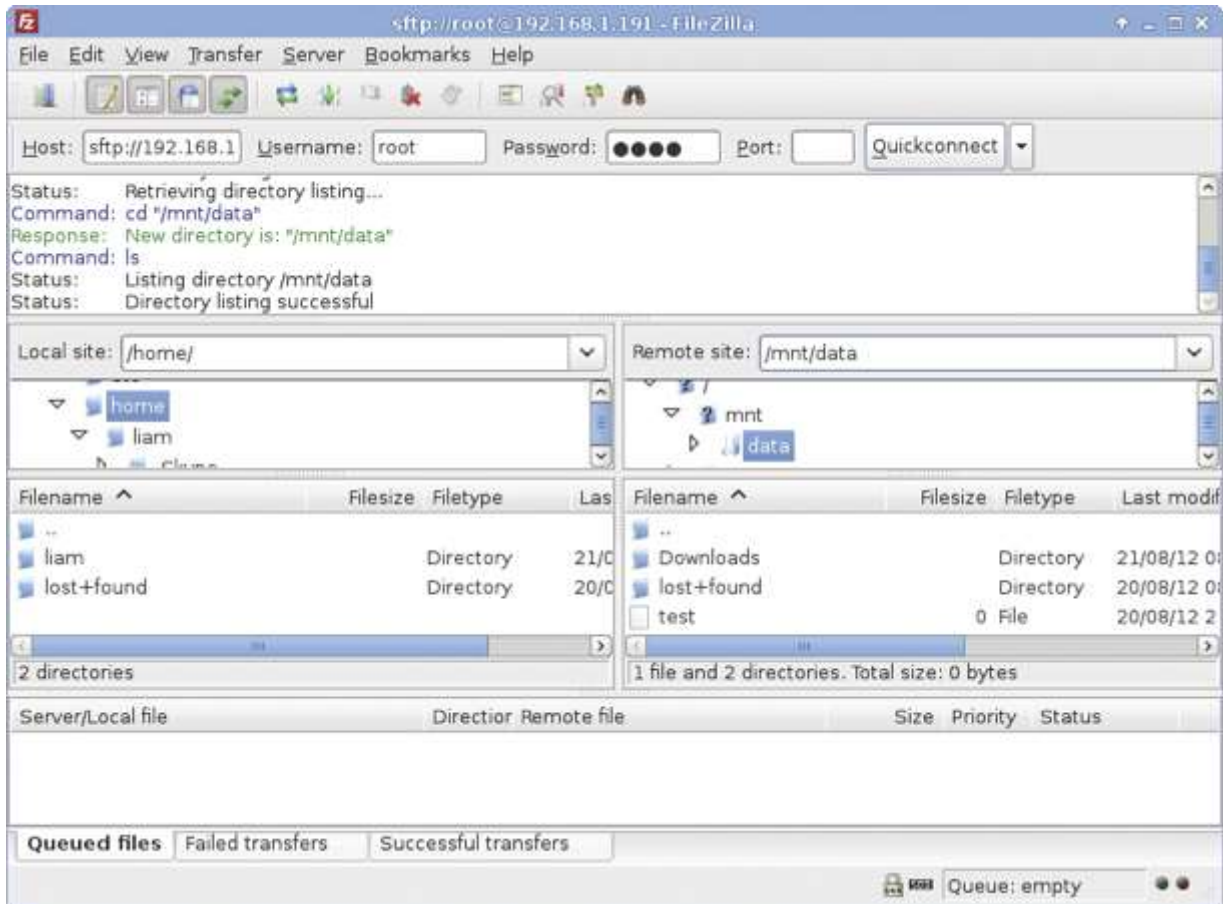
Raspberry Pi with all necessary peripherals

SD card containing the [latest Arch Linux](#) image for RasPi

Powered USB hub

External hard drive or pen drive

A computer running Linux – it can be another Raspberry Pi



Step by Step

Step 01 Log into Arch Linux

Connect the necessary cables to the Pi and wait for the Arch Linux login prompt. The login is root, and the password is also 'root'. We'll change the root password from the default later on.

Step 02 Run a full system update

Arch Linux runs on a rolling release schedule, meaning that there are no version numbers and software is continually updated. Use the command 'pacman -Syu' to start a full system update. You'll be shown a prompt explaining that pacman needs to be updated first. Answer 'Y' to the prompt. You can now run 'pacman -Syu' again to start a full system update. There will be a lot of packages to update so it will probably take a while, especially because the Pi runs from an SD card.

Step 03 Investigate your network

We highly recommend assigning a static IP to your Raspberry Pi rather than being handed one by your router because you'll always know where to find it on the network, which will be useful for accessing it remotely. You'll also need a static IP if you want to access the Raspberry Pi from the internet. We'll need to find out a couple of things about your current

network setup before setting a static IP. You can use the commands ‘ifconfig eth0’ and ‘ip route show’ to do this.

Step 04 Set up a static IP address

Now that we have found out things about your network, such as your current IP address, the network mask and so on, we can set up a static IP address. The majority of the Arch Linux configuration is done from a file called rc.conf, which is located in the /etc directory. You can open this file in the nano editor with the command ‘nano /etc/rc.conf’. Scroll down using the arrow keys until you come to the networking section.

Begin to fill in the blank eth0 section. Our IP address was 192.168.1.215. When assigning an IP address, you’ll only want to change the last number. We’re going to change our address to 192.168.1.191. You’ll probably want to use a number between 100 and 200. Set the netmask and broadcast to the values from ‘ifconfig eth0’ so that the interface is configured correctly for your home network. Set the gateway to the address in the line similar to default via ‘192.168.1.254 dev eth0 metric 202’ from the output of ‘ip route show’. Once you have done this, use the key combination Ctrl+O followed by the Enter key to save the changes. You can then use Ctrl+X to exit nano.

Step 05 Log in with SSH

We’ll now log into the Pi using SSH – but before that, we’ll restart so that the static IP and any updates take place. Use the command ‘init 6’ to do this. Once the Pi has booted back up, open a terminal on your Linux computer and type ‘ssh root@[ip of your pi]’. Answer yes, to say that you want to connect, and type in the root password, which will still be root. You are now logged in over SSH.

Step 06 Set the default gateway

This step is really part of step 4 (set up a static IP address). The reason for doing this step after restarting, is that we found the changes werelostwhenwediditbeforearestart. This is probably due to DHCP cleaning up any settings it changes during shutdown. We will now set the IP of the DNS server, which will be used to resolve addresses such as google.co.uk to an IP address that your computer can connect to. Your DNS server will be provided by the same IP as your default gateway. You can set the DNS server using the following command:

```
echo ‘nameserver [ip of default gateway]’ > /etc/resolv.conf.
```

Step 07 Change the root password

Since we might be exposing an SSH login to the internet, we need to change the password to a more secure one. Type ‘passwd’, then follow the on-screen instructions to change your password. Your SSH session will stay logged in, but you’ll need to use the new password next time you log in.

Step 08 Install the required packages

Use the command ‘pacman -S samba noip transmission-cli’ to install the required packages mentioned at the start of the article. Answer ‘Y’ to any prompts you may encounter.

Step 09 Formatting your hard drive

Before continuing, note that formatting a disk will lose all the data on it. It's now time to connect your hard drive to the Pi. (We're using a pen drive, but the steps are exactly the same.)

Use the command

```
fdisk -l | grep Disk
```

to get a list of disks attached to the system. You should be able to tell which disk is the one you intend to use to store files on by the size of the disk. For example, our pen drive is 16GB, so we know that the 16.1GB disk is ours. Run the following command:

```
(echo o; echo n; echo p; echo 1; echo ; echo; echo w) | fdisk [path to your disk]
```

to create a partition that will be whole size of the disk. Fdisk is an interactive program, so this command simply inputs the keys that you would press in a specific order to create a partition that will take up the whole size of the disk.

Step 10 Create a file system & mount point

Once we have a partition on the disk, we need to create a file system and a place to mount it. You can create an ext4 filesystem on the partition using the command

```
mkfs.ext4 [path to disk]1
```

You can make a directory to mount the drive in using the command

```
mkdir /mnt/data
```

Step 11 Make the drive accessible to Linux

The fstab file on Linux system contains a list of storage devices on the system, and where to mount them. The file is read on boot, so any devices in the fstab file will be mounted automatically as long as it is connected during boot.

We'll edit the fstab in nano using the command 'nano /etc/fstab'. You want to add a line like the following:

```
[path to disk]1 /mnt/data ext4 defaults 0 1
```

Save the changes in the same way we did earlier. Once you have done this, you can use the command 'mount -a' to reread the fstab file and mount the new entry we just added.

Step 12 Share the drive with network clients

We need to open an empty Samba configuration file in nano using the command

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

Input the following information:

```
[global]
# workgroup = NT-Domain-Name or Workgroup-Name, eg: MIDEARTH
workgroup = WORKGROUP
security = user
load printers = no

# Some performance tuning
socket_options = TCP_NODELAY SO_
RCVBUF=65536 SO_SNDBUF=65536
```

```
[data]
path = /mnt/data
public = no
writable = yes
```

Save the changes in the same way we did earlier.

Step 13 Permit access to the shared drive

Set a password for your Samba user using the command ‘smbpasswd -a’ and then enter the password you want to use twice. Now that we’ve done that, we can start the Samba daemon using the command

```
/etc/rc.d/samba start
```

Note that the method of accessing a Samba share from each operating system will be different, so you may have to look it up. It should show up as ‘ALARMPI’ in the Network browser on most operating systems and user- friendly Linux distributions. Use the username ‘root’ and the password that you set with the smbpasswd command.

Step 14 Configure the Transmission torrent daemon

We’ll need to start the Transmission daemon because we’ll need to edit some configuration files that it will only make on first execution. Start it using the command

```
/etc/rc.d/transmissiond start
```

You’ll then want to stop it using the command

```
/etc/rc.d/transmissiond stop
```

so that we can edit the configuration file without Transmission interfering. Open the configuration file using the command

```
nano /root/.config/transmission- daemon/settings.json
```

You need to set `rpc-whitelist-enabled` from true to false, and also change the `download-dir` to `/mnt/data/Downloads`. You can then save the changes and make that directory using the command

```
mkdir /mnt/data/Downloads
```

Step 15 The Transmission web interface

We can now start the Transmission daemon using the same start command from the previous step. Transmission can sometimes be a bit picky with the syntax of its configuration files, so now that you've started it, you'll want to check that your configuration changes are still there. If not, stop the Transmission daemon and try again.

You can go to the Transmission web interface by opening up a web browser and going to the IP address of your Pi, followed by `:9091`. This is needed because the web interface is served from port 9091. In our case, we used `192.168.1.191:9091`. Click on the icon in the top-left corner of the interface to add a torrent file. You can either upload a file from your computer, or copy and paste a URL.

The Transmission web interface is really straightforward and easy to use, so you should be able to figure out anything else you want to do. There is also more documentation on the `settings.json` configuration file on the [Transmission website](#).

Step 16 Set up dynamic DNS

Head over to [No-IP](#) and sign up for the No-IP Free option. Once you have done that, don't bother downloading No-IP's client because we've already installed it. Go to your email Inbox and follow the activation link that was just sent to you by No-IP. You can now sign in to your account. Once you have logged in, select the 'Add a host' option. Choose a hostname and a domain to be part of from the drop-down list. Leave the Host Type as 'DNS Host' and then click the 'Create Host' button.

Your advisor used the hostname `liam-ludtest` with the domain `no-ip.org`, so would access that using `liam-ludtest.no-ip.org`.

Step 17 Configure No-IP

Run the command

```
noip2 -C -Y
```

to be taken through interactive configuration of the No-IP client. We left the update interval to the default of 30 minutes, meaning the client will check every 30 minutes for an IP address change. Once you've finished, start the daemon with the command `/etc/rc.d/noip start`.

After a minute or two, your IP address will be accessible via your No-IP hostname. However, it's likely that trying it from inside your house will simply take you to your router's homepage.

Step 18 NAT port forwarding

It is likely that there are multiple devices behind your router that all use the same external IP address. This is because of the shortage of IPv4 addresses, and also because it is more secure to segregate the internet from your internal home network. NAT (network address translation) forwards a port from the router's external IP address to a computer on the LAN (local area network). In this case, we'll want to forward any traffic for port 22 that comes to your router's external IP address to the IP address of your Raspberry Pi. Port 22 is the port used for SSH, which is the only port we'd recommend that you forward. SSH will provide access to your files, and also port forwarding, so you can access the Transmission web interface should you want to.

The configuration of port forwarding really depends on the router that you are using, so you may have to look it up. The chances are that it will be hidden away in the 'advanced' section of your wireless router. You should be able to access your router by typing your No-IP hostname into your web browser. If not, it should be at the address of your default gateway that we used earlier on.

On his router, your advisor had to go to Advanced>NAT>Port Mapping and add a mapping.

Step 19 Use FileZilla for accessing files

We recommend that you use FileZilla, which you should be able to install with your package manager, to access files via SSH. Set the host to the IP address of your Pi (or your No-IP hostname if outside your LAN), the username to 'root', and the password to your password. Set the port to 22 and click 'Quickconnect'. Click OK to trust the host and connect to it. To access files on your external drive, change remote site from /root to /mnt/data.

Step 20 Use SSH to port forward

Use the SSH command with the following option to forward any traffic from 127.0.0.1:9091 to 126.0.0.1:9091 on the Pi:

```
ssh -L 9091:127.0.0.1:9091 root@192.168.1.191
```

(You can replace the IP address of the Pi with your No-IP hostname if you are outside of your LAN.) 127.0.0.1 is a loopback address, which points back to 'this computer'. Once you have run that command, you can go to the web browser on your computer, type 127.0.0.1:9091 and access the Transmission web interface as if it was on the LAN! Enjoy your new file server!

Related Raspberry Pi articles

- [How to set up the Raspberry Pi](#)
- [Build a Raspberry Pi Media Centre](#)
- [Raspberry Pi review – the price is right, but the software is not](#)
- [Raspberry Pi Interview: Eben Upton Reveals All](#)