

Creating a Linux based home cloud & media service by using Raspberry Pi 3

Bence Vadnai
Faculty of informatics
University of Debrecen, Faculty of
Informatics
Debrecen, Hungary
vadnaibence@gmail.com

Edwin Shaw
London South Bank
University
School of Law and Social
Sciences,
London, United Kingdom
shawe5@lsbu.ac.uk

Timotei István Erdei
Department of Mechatronics
University of Debrecen, Department
of Mechatronics
Debrecen, Hungary
timoteierdei@eng.unideb.hu

Abstract — Nowadays the microcontrollers are getting cheaper and easier to configure. In my project I am going to show you the best ways to share the data at home between your devices. My goal was to create a home cloud service with some extras only using some old laptop HDDs and a Raspberry Pi.

Keywords—network; home; cloud service; Raspberry Pi; Linux; Samba; HDD

I. INTRODUCTION

Everyone has data what has to be stored, like pictures, videos, music files, texts for the university etc. The best way to share files with others is by using the internet. Internet is for communication, for sharing information with each other. There are a lot of services where you can easily store your irreplaceable data, but in my case, at home I have slow internet connection which can increase the file sync processing time and reduce the efficiency, therefore to solve the file transferring issue and get rid of using pendrives to copy my files I wanted to create a home network file sharing [19].

II. PLANNING VIEWPOINTS

At planning the most important viewpoint was the price. I didn't want to spend too much money for it, so I had to write down a price limit, what was 100\$. (100\$ is about 25000 Hungarian Forints). This is the reason why I decided on Raspberry Pi.

On the other hand; the Linux operation systems are well optimized for server and industrial simulation works. Moreover the Android Linux os became the most popular mobile operation system in the last 5 years. The desktop

distributions of Linux are also well known in research aspects.

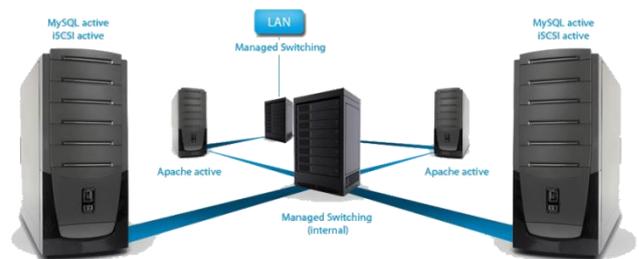


Figure 1.: Linux based servers-clusters [11]

My planning viewpoints contained the low energy consumption and the Raspberry was a good choice in this case as well, because it only needs 3,7W at the CPU stress test, which is an overestimated situation [1]. Most of the times the device will be in Stand-by mode, which only needs about 1,4W, which is great.

One of the most indispensable part of the planning was to reach the highest level of security. I prefer the UNIX base operating systems for file sharing. Firstly, they are using the ext4 file system which has much better and clear file topography. On second hand I got some experience of Debian, when I was working at the University of Debrecen's Informatics building as a system administrator. I feel secure after seeing how easy it is to get in a Windows based system, even in 2017. It is enough to write only one filename and you are already get an administrator command line. Overall the Raspberry Pi was the best choice for this project, doing well at all of the points above.

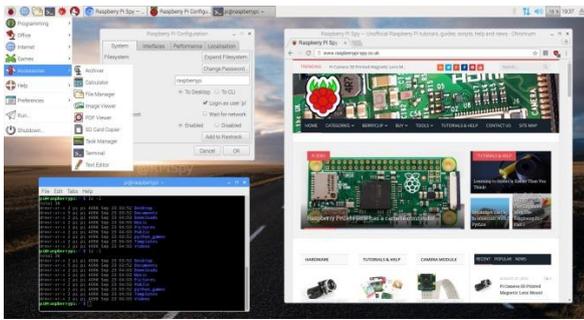


Figure 2.: Image Raspberry using Debian based system, Raspbian [12]

III. THE HARDWARE STRUCTURE OF A HOME FILE SHARING SYSTEM

The root of the network system is a router, which can be found in every single home. Each Wi-Fi routers have LAN ports, which is the best way to connect to the raspberry. We can use Wi-Fi as well, because the latest version of raspberry pi; the Wi-Fi transmitter is already included, inbuilt, but the performance won't be the best. So to avoid the speed issues I connected my Raspberry to my Wi-Fi router using the LAN port. It makes it possible to reach the Raspberry (for instance SSH) by only using the Wi-Fi connection.

The Raspberry Pi has a built in a compact case to avoid the dust and other damages. The case has a small fan to keep the device at good temperature; as you can see in the second image below. The Raspberry is using a 16 GB SD card for the operating system, which is enough for this task.

For the storage I use an old HDD laptop. To connect the HDD to the Raspberry I needed to buy a SATA to USB adapter. (I am going to explain the purchase in session IV.)

I had some problems with the SATA to USB case because of the following reason: there are two types of the laptop 2.5 inch HDDs. One is 9.5 mm thick while the other is 12.5 mm. You can check the difference on the third image at the right side of the page. Most of the 2.5 inch USB case is for the thinner one, therefore I wasn't able to close the case. In the future I am going to get a wider one to make it possible to fit in. Otherwise the case is working just fine, I got no other issues with that. It will just do the job for a while, until the new one arrives.



Figure 3.: Raspberry Pi 3 case with fan [13].

Let me to write about the speeds, and the bottlenecks of the structure. The bottleneck is that fragment of the system that reduces the performance between two parts. Almost every network system has a bottleneck. In my system the biggest performance issue is caused by the USB data transferring. USB 3.0 can handle 640 MB/sec but the write speed can't even be close to this number. The best USB write speed is about 40 MB/sec. This value depends of the quantity and size of the files, what needs to be written on read in.

IV. PURCHASING & PRICES

Nowadays purchasing from the internet is getting common. You can find a ton of things which are computer science related. You can buy the devices at an international price, which is great.

In this session; I am going to write about the purchasing of the devices. The main device is the raspberry, which was about **16000 HUF** from eBay. This was a starter kit, that contains the PI, a 16GB SD-card, the 5V 2A adapter, with switch, a poor quality HDMI cable, but enough for the configurations and the case with the fan. The package also contains some self-sticker heat sink for better cooling.

The second part was the HDD. I already had this, but you can easily buy one (or an external HDD) for a good price on the second-hand market. I don't calculate this item in my budget. But I needed the HDD case which was **1600 HUF** from the internet as well, but from another site called AliExpress [2].

V. THE PROCESS OF THE OS CONFIGURATION

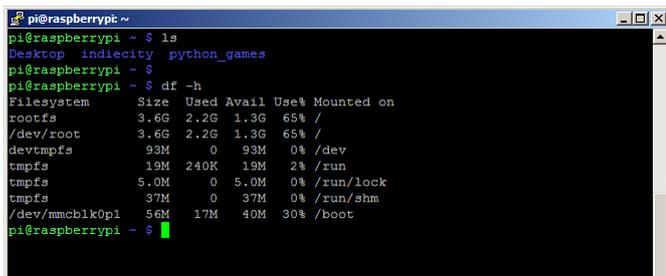
The recovery partition has been preconfigured on the SD card, what I bought. So the installation was a piece of cake. The NOOBs is a graphical installer interface to the raspberry, which makes the first steps effortless. There were two options in my head: I was sure, if I want to use the Raspbian, there are two version of it. The Raspbian with desktop and Raspbian lite; I decided with the latter, for the following reasons.

The raspberry is not the strongest hardware in the world, therefore to get the best experience, I need to save as much

power as possible and turn off those services that are using too much power even in standby mode. The lite version of Rasbian does not contain the graphical interface, it has no desktop but terminal. I thought this will be a good way to learn and to get comfortable in using only terminal. I think I was right, because I learned a lot and it got easy to navigate, installing packages. The pure terminal is like every Linux based operating system's command line.

After the operating system installation it is recommended to make some updates and upgrades with the following command:

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



```
pi@raspberrypi ~ $ ls
Desktop  indiecity  python_games
pi@raspberrypi ~ $
pi@raspberrypi ~ $ df -h
Filesystem      Size  Used Avail Use% Mounted on
rootfs           3.6G  2.2G  1.3G   65% /
/dev/root        3.6G  2.2G  1.3G   65% /
devtmpfs         93M   0     93M   0% /dev
tmpfs            19M   19M   0     100% /run
tmpfs            5.0M   0     5.0M   0% /run/lock
tmpfs            37M   0     37M   0% /run/shm
/dev/mmcblk0p1  56M   17M   40M   30% /boot
pi@raspberrypi ~ $
```

Figure 4.: SSH connection with Putty [14]

After a quick reboot (sudo reboot) it is the time to change to default passwords.

The default username and password combo is *pi* and '*raspberry*' which would be too weak to keep it, considering the raspberry should be available on the internet. The rookie 'hackers' best practice is to scan the available ports on a site and try the default username password combos. Use the following commands to change the password of user *pi* and create a new user with root rights:

```
sudo passwd pi
sudo adduser "username"
sudo usermod -a -G root "username"
```

We will need to prepare the storage [4]. It is recommended to use ext4 file system, because you can avoid some issues with this step. I tried to connect the usage to an existing and working NTFS file system. It could work as well, but you will need to install some extra packages, for instance *ntfs-3g* (sudo apt-get install ntfs-3g). I do not indorse this way, because you can get some permission issues with it later.

You can see the further steps on image 5.

Before you format the drive you will need to unmount it:

```
Sudo umount /dev/ yourpartitionhere
```

After this you format the drive using the following command:

```
sudo mkfs.ext4 /dev/yourpartitionhere
```

Let's create a folder to mount the HDD and mount it, then create a link for it to our home folder:

```
Su
Mkdir /mnt/hdd
Mount /dev/yourpartition /mnt/hdd
Cd; ln -s /mnt/hdd hdd
```

After all let's check what network interface is in use (ifconfig -a) and disable the wlan0 (sudo ifconfig wlan0 down). To make sure if we have internet connection just try to ping www.google.com [15].

At the end of this session I am going to give some indication to you about the other packages what we will need later:

Apache2: "The Apache HTTP Server Project is an effort to develop and maintain an open-source HTTP server for modern operating systems including UNIX and Windows"[5].

TightVNCServer: "TightVNC can be used to perform remote control and administration tasks in Windows, UNIX and mixed network environments. It can be very helpful in distance learning and remote customer support. Finally, you can find a number of additional VNC-compatible utilities and packages that can extend the areas where TightVNC can be helpful [6].

TightVNC is a project maintained by Constantin Kaplinsky. Many other individuals and companies participate in development, testing and support."

BMON: This is a useful network monitoring tool. The network traffic can be easily monitored with this tool. What is a big advantage is that it can be used in terminal. No graphical interface needed [16].

Samba: "Samba is the standard Windows interoperability suite of programs for Linux and UNIX. Samba is an important component to seamlessly integrate Linux/Unix Servers and Desktops into Active Directory environments. Samba is a software package that gives network administrators flexibility and freedom in terms of setup, configuration, and choice of systems and equipment" [7].

Owncloud: I am going to explain this later, there will be a full session of this [18].

Kodi: Kodi is an open source software media centre, allowing you to access all kinds of content, including videos and music. You can run that content on any device of your choosing, including your TV, laptop, phone, tablet and others.

Kodi comes with its own user interface and works with both local and network storage services you're already using [8].

Deluged: Deluge is a fully-featured cross-platform BitTorrent client. It is Free Software, licensed under the GNU GPLv3+ and adheres to freedesktop standards enabling it to work across many desktop environments [17].

VI. REMOTE CONTROLLING

To remote the raspberry as I mention before, I only use the SSH connection. On Windows the most common SSH client is the Putty. In the future the Windows will be capable to connect to other devices using SSH from PowerShell. Next to the SSH I installed as you seen in the previous session the TightVNC. If I really need graphical interface there is the possibility to turn it on or off. So this is the second way to

remote the device. In my plans I would like to upgrade the raspberry with an IR receiver to turn on and off Kodi in the future.

```
Disklabel type: dos
Disk identifier: 0x000903ee

Device      Boot  Start      End  Sectors  Size Id Type
/dev/mmcblk0p1      8192  3140625  3132434   1.5G e W95 FAT16 (LBA)
/dev/mmcblk0p2    3140626  30924799  27784174  13.3G 5 Extended
/dev/mmcblk0p5    3145728  3211261    65534   32M 83 Linux
/dev/mmcblk0p6    3211264  3346431   135168   66M c W95 FAT32 (LBA)
/dev/mmcblk0p7    3350528  30924799  27574272  13.2G 83 Linux

Disk /dev/sda: 1.8 TiB, 2000398933504 bytes, 3907029167 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0xe9740205

Device      Boot  Start      End  Sectors  Size Id Type
/dev/sdal      *    64 3907024128 3907024065   1.8T 7 HPFS/NTFS/exFAT
vadnay@raspberrypi:~/hdd $ sudo mount /dev/sdal /mnt/hdd/
vadnay@raspberrypi:~/hdd $ ls
owncloud.log  proba
vadnay@raspberrypi:~/hdd $ cd ..
vadnay@raspberrypi:~ $ ls
hdd
vadnay@raspberrypi:~ $ cd hdd
vadnay@raspberrypi:~/hdd $ ls
lost+found
vadnay@raspberrypi:~/hdd $ cd hdd
```

Figure 5.: Image: Mounting hdd using Putty

VII. CONFIGURING OWNCLLOUD SERVICE

We reach the main point of the project. After browsing the internet the best self-configurable cloud service is the OwnCloud [10]. OwnCloud is a suite of client-server software for creating file hosting services and using them. OwnCloud is functionally very similar to the widely used Dropbox, with the primary functional difference being that the Server Edition of ownCloud is free and open-source, and thereby allowing anyone to install and operate it without charge on a private server. It also supports extensions that allow it to work like Google Drive [3], with online document editing, calendar and contact synchronization, and more. Its openness eschews enforced quotas on storage space or the number of connected clients, instead having hard limits (like on storage space or number of users) defined only by the physical capabilities of the server [9].

Warning for the following part of this session you might be administrator, so it is recommended to do it as *su*.

Before the installation of the ownCloud we need a webserver what is going to be the Apache2 (installed in session V). We need to start Apache2 and enable to start on boot:

```
systemctl start apache2
systemctl enable apache2
```

We need to install additional packages required by Owncloud Server:

```
apt install -y apache2 mariadb-server libapache2-mod-
php7.0 \
php7.0-gd php7.0-json php7.0-mysql php7.0-curl \
php7.0-intl php7.0-mcrypt php-imagick \
php7.0-zip php7.0-xml php7.0-mbstring
```

Now, after some installation, we are ready to download and install ownCloud - 10.0.3.tar.bz2 [18]. Go to folder */tmp* with *cd /tmp*.

As you can see, the downloaded file is a .tar file, what needs to be unzipped. There is a good inbuilt tool in UNIX based system called tar. Use the following command to extract Owncloud and change the permission: *tar -xvf owncloud-10.0.3.tar.bz2*

```
chown -R www-data:www-data ownCloud
```

This will create a folder called ownCloud. We need to move this directory to */var/www/html* (*mv directory_to _move target_location*).

The Apache2 webserver needs some configuration as well before start ownCloud. Create a new configuration file with your favourite text editor called *owncloud.conf* to here:

```
/etc/apache2/sites-available/
```

Write the following lines to the the configuration file:

```
Alias /owncloud "/var/www/html/owncloud/"
```

```
<Directory /var/www/html/owncloud/>
Options +FollowSymlinks
AllowOverride All
```

```
<IfModule mod_dav.c>
Dav off
</IfModule>
```

```
SetEnv HOME /var/www/html/owncloud
SetEnv HTTP_HOME /var/www/html/owncloud
</Directory>
```

These lines for the Apache2. We set an alias name, the directory files of the webpage and we set the home page of the website. Now create a symbolic link of this file to the enabled sites folder in Apache2 and enable some modules:

```
ln -s /etc/apache2/sites-available/owncloud.conf
/etc/apache2/sites-enabled/owncloud.conf
a2enmod headers
systemctl restart apache2
a2enmod env
a2enmod dir
a2enmod mime
```

We need to create a database for ownCloud what was to be expected. For this we are going to use mysql. Let's open mysql as root: *mysql -u root -p*

Create a database and a user what gets every rights for the new database. Here are the required commands:

```
> create database owncloud;
> create user *yourusername*@localhost identified by
*password*;
>grant all privileges on owncloud.* to *yourusername*@
localhost identified by *password*;
>flush privileges;
>exit;
```

Now everything is ready to try. Get your ip address by ifconfig. As you can see the configurations above, the address will be **your_ip_adress/owncloud**

If you write it to your browser you will get into the owncloud start page as you can see on image six.

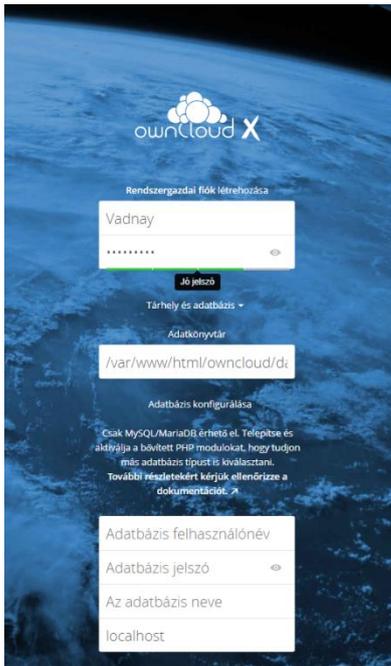


Figure 6.: The start configuration page of ownCloud

Now we can create the admin user. Write a username and password to the first two field. Set the data directory. This will be my mount point of the external HDD. And the last four field have to be filled with the mysql database username and password and with the database name.

I got the 0770 permission error after the configuration. The permission wasn't set right. To fix it use the next commands:

```
>sudo chmod 0770 /mnt/hdd  
>sudo chown www-data:www-data /mnt/hdd
```

We are ready to log in our new cloud service. Just use the admin username and password, what was set before.

To use the user interface is pretty easy. You can manage the users of the system. Give them privileges, reset their password, and set upload limit.

As I mentioned before this is a free-to-use software until you don't use it for enterprise tasks. In case of that, you should buy the enterprise version.

The ownCloud works with Android and iOS devices as well. It has application for it which is not free of charge, but you can use the browser version instead.

You can see the default look of ownCloud at picture 7th on the next page.

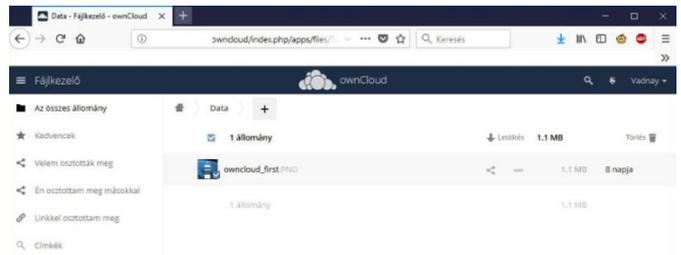


Figure 7.: ownCloud clear storage

VIII. EXTRAS

In this session I would like to talk about the features above the cloud service. As I mentioned before I installed the open sourced Kodi. I am in a fortunate situation that my TV is close to the router. So I decided to use my raspberry pi as a media player as well. Using the samba share, I can copy my records, pictures or music to the raspberry without cables.

One of the most annoying issue with Kodi was the controlling. I have an LG smart TV, and wasn't too comfortable to use a wireless keyboard.

There is two type of HDMI 2.0. One is the standard and one is the ARC HDMI, which is capable to send signal to the device, what gives the picture. With this technology we are able to send "commands" to the raspberry pi and to navigate in Kodi's menu using the TV-s remote. It can actually turn the TV on or off.

Let's write about the Kodi's features [8]. On the music front, Kodi can play a slew of formats, including AAC, MP3, and OGG. It also comes with smart playlists and the ability to tag different tracks. With video, Kodi works with ISO, 3D, and H.264, among other formats, and can also stream content over the Internet. Once you import the films, Kodi can add posters, fan art, trailers, video extras, and more to each item. Next to all of these it has fancy applications of news weather forecast, Youtube, Spotify and ton of more.



Figure 8.: Kodi's user interface

IX. SUMMARY AND "FUTURE FEATURES"

It is just a cheap and useful device what can make the everyday file sharing between my devices.

The overall price was about 20 000 HUF what is below then except and it is cheaper than one year of 1tb storage of Google Drive. The project is done, I am using it almost every day.

In the future I will contact to my internet suppliers, to help me to open a port to make my device available on the internet. The reason is I need help is that I have a dynamic IP at home, so it is not enough to forward a port on the router.

I will extend the storage with a 2tb of external hard drive. This will be enough for my data.

X. ACKNOWLEDGEMENT

The work is supported by the EFOP-3.6.1-16-2016-00022 project. The project is co-financed by the European Union and the European Social Fund.

XI. REFERENCES

- [1] Google, (2017.12.01.) [Online]. Available: <https://www.pidramble.com/wiki/benchmarks/power-consumption>
- [2] Google, (2017.12.01.) [Online]. Available: <http://www.tomshardware.com/reviews/2.5-inch-12.5-mm-9.5-mm,2623.html>.
- [3] Google, (2017, May 16). [Online]. Available: <https://www.google.com/drive/>
- [4] How to format a drive under linux, (2017.12.15). [Online]. Available: <https://askubuntu.com/questions/549078/format-external-hard-drive-to-ext4-in-command-line>
- [5] Apache2, (2017, Dec. 15.). [Online]. Available: <https://httpd.apache.org/>
- [6] TightVNC, (2017.12.13). [Online]. Available: <http://www.tightvnc.com/intro.php>
- [7] Samba Server, (2017.12.12.). [Online]. Available: https://www.samba.org/samba/what_is_samba.html
- [8] Kodi (2017.12.12) [Online]. Available: <https://www.tomsguide.com/us/what-is-kodi,review-4160.html>
- [9] ownCloud 2017.12.12 [Online]. Available: <http://www.zdnet.com/article/owncloud-build-your-own-or-manage-your-public-cloud-storage-services/>
- [10] ownCloud configuration 2017.12.12 [Online]. Available: <https://www.avoiderrors.net/owncloud-10-raspberry-pi-3-raspbian-stretch/>
- [11] Linux Clusters – Servers configuration 2017.12.12 [Online]. Available: <http://www.cloud4hosting.eu/cheap-servers/linux-servers-cluster-network/>
- [12] Linux Pixel Os – Servers configuration 2017.12.12 [Online]. Available: <https://www.raspberrypi-spy.co.uk/2016/12/running-pixel-desktop-on-a-pc-or-mac/>
- [13] Raspberry Pi 3 Case – Servers configuration 2017.12.12 [Online]. Available: <https://www.amazon.fr/Eleduino-Raspberr-Mod%C3%A8le-acrylique-ventilateur/dp/B01CVFEP6G>
- [14] Putty – SSH configuration 2017.12.12 [Online]. Available: <https://www.raspberrypi.org/documentation/remote-access/ssh/windows.md>
- [15] Google 2017.12.12 [Online]. Available: <https://www.google.com/>
- [16] Bmon 2017.12.12 [Online]. Available: <https://github.com/tgraf/bmon>
- [17] Deluge 2017.12.12 [Online]. Available: <https://deluge-torrent.org/>
- [18] Deluge 2017.12.12 [Online]. Available: <https://owncloud.org>
- [19] T. I. Erdei, Zs. Molnár, N. C. Obinna, G. Husi, „Cyber physical systems in mechatronic research centre,” MATEC Web Conf. Volume 126, 2017./