



(Established under Galgotias University Uttar Pradesh Act No. 14 of 2011)

PERSONAL CLOUD USING RASPBERRY PI

A Report for the Evaluation 3 of Project 2

Submitted by

DHRUV SAINI

(1613105038)

in partial fulfilment for the award of the degree

of

BACHELOR OF TECHNOLOGY

IN

**COMPUTER SCIENCE AND ENGINEERING WITH SPECIALIZATION OF
CLOUD COMPUTING AND VIRTUALIZATION**

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING

**Under the Supervision of
Mr. SOUMYA RANJAN JENA, M.Tech.
Assistant Professor**

APRIL / MAY- 2020



**SCHOOL OF COMPUTING AND SCIENCE AND
ENGINEERING**

BONAFIDE CERTIFICATE

Certified that this project report **“PERSONAL CLOUD USING RASPBERRY PI”** is the bonafide work of **“DHRUV SAINI (1613105038)”** who carried out the project work under my supervision.

SIGNATURE OF HEAD

Dr. MUNISH SHABARWAL,
PhD (Management), PhD (CS)
Professor & Dean,
**School of Computing Science &
Engineering**

SIGNATURE OF SUPERVISOR

Mr. SOUMYA RANJAN JENA, M.Tech.
Assistant Professor
**School of Computing Science &
Engineering**

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	iii
	LIST OF FIGURES	iv
1.	Introduction	1
2.	Existing System	4
3.	Proposed system	6
4.	Implementation or architecture diagrams	9
5.	Output / Result / Screenshot	22
6.	Conclusion/Future Enhancement	24
7.	References	26

Abstract

The research work explains about the personal edge cloud storage using Raspberry Pi where we create a Cloud storage for our personal use rather than relying on other Cloud storage services like Dropbox, Google Drive, iCloud etc. In this dynamic environment and ever changing technologies, security of our data is utmost important as well as the storage we need to store data and have control over our data. We focuses on the above mentioned problems. As third party Cloud services are open to other users too, so this creates an issue of security. As well as these Cloud service providers provide a limited amount of storage and they also have some control over our data too and we have to pay some hefty amount of money to use these Cloud services. Using Raspberry Pi we can use our external hard drive as Cloud storage for our personal use only. We can decide the amount of memory by ourselves, and can have full control over our data. Using ownCloud we can access our data in our external hard drive through any device which has internet, treating our external hard drive as a Cloud storage device.

List of Figures

The figures used in this project are as follows-

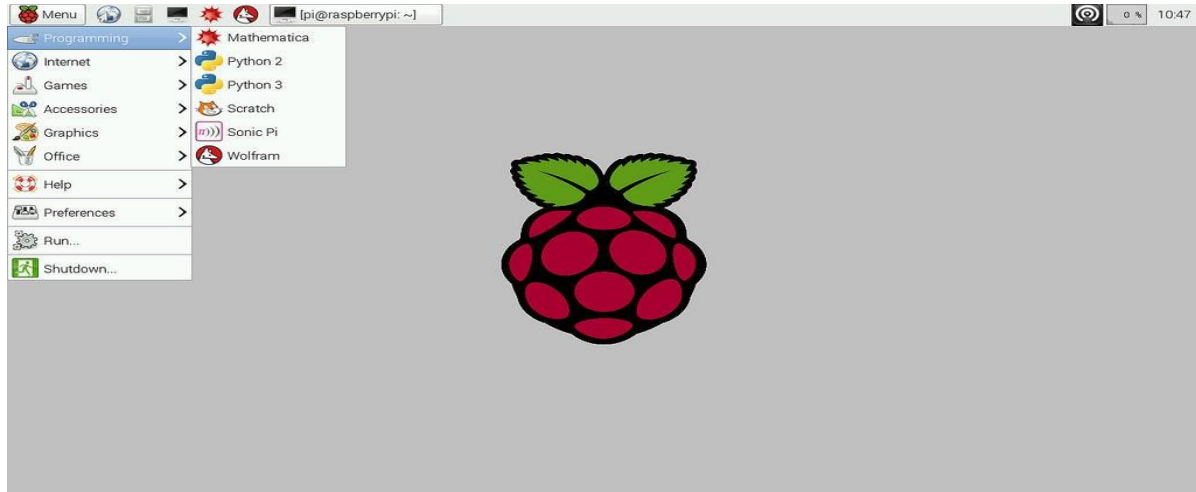


Fig.1 User interface of Raspbian OS

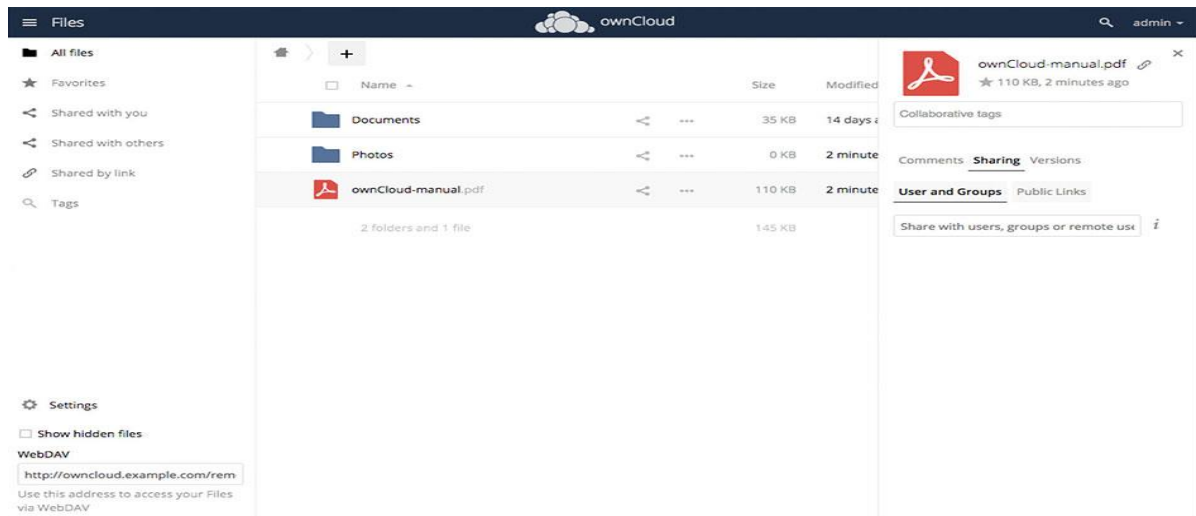


Fig.2 Owncloud User Interface

```
pi@raspberrypi: /var/www/owncloud
GNU nano 2.2.6 File: .user.ini Modified
upload_max_filesize=2000M
post_max_size=2000M
memory_limit=2000M
mbstring.func_overload=0
always_populate_raw_post_data=-1
default_charset='UTF-8'
output_buffering=off
File Name to Write: .user.ini
^G Get Help      M-D DOS Format  M-A Append     M-B Backup File
^C Cancel        M-M Mac Format  M-E Prepend
```

Fig.3 After saving and then exiting from this screen the setup of owncloud server will be completed.

```
pi@raspberrypi: /var/www/owncloud
pi@raspberrypi /var/www/owncloud $ id -g www-data
33
pi@raspberrypi /var/www/owncloud $ id -u www-data
33
pi@raspberrypi /var/www/owncloud $ ls -l /dev/disk/by-uuid
total 0
lrwxrwxrwx 1 root root 15 Aug 29 06:25 13d368bf-6dbf-4751-8ba1-88bed06bef77 -> .
../mmcblk0p2
lrwxrwxrwx 1 root root 15 Aug 29 06:25 15CD-3B79 -> ../mmcblk0p1
lrwxrwxrwx 1 root root 10 Aug 29 06:51 DC72-0315 -> ../sda1
pi@raspberrypi /var/www/owncloud $ sudo nano /etc/fstab
pi@raspberrypi /var/www/owncloud $ sudo reboot

Broadcast message from root@raspberrypi (pts/0) (Sat Aug 29 06:57:02 2015):
The system is going down for reboot NOW!
pi@raspberrypi /var/www/owncloud $
```

Fig.4 The external hard drive has been mounted as well as rebooted.

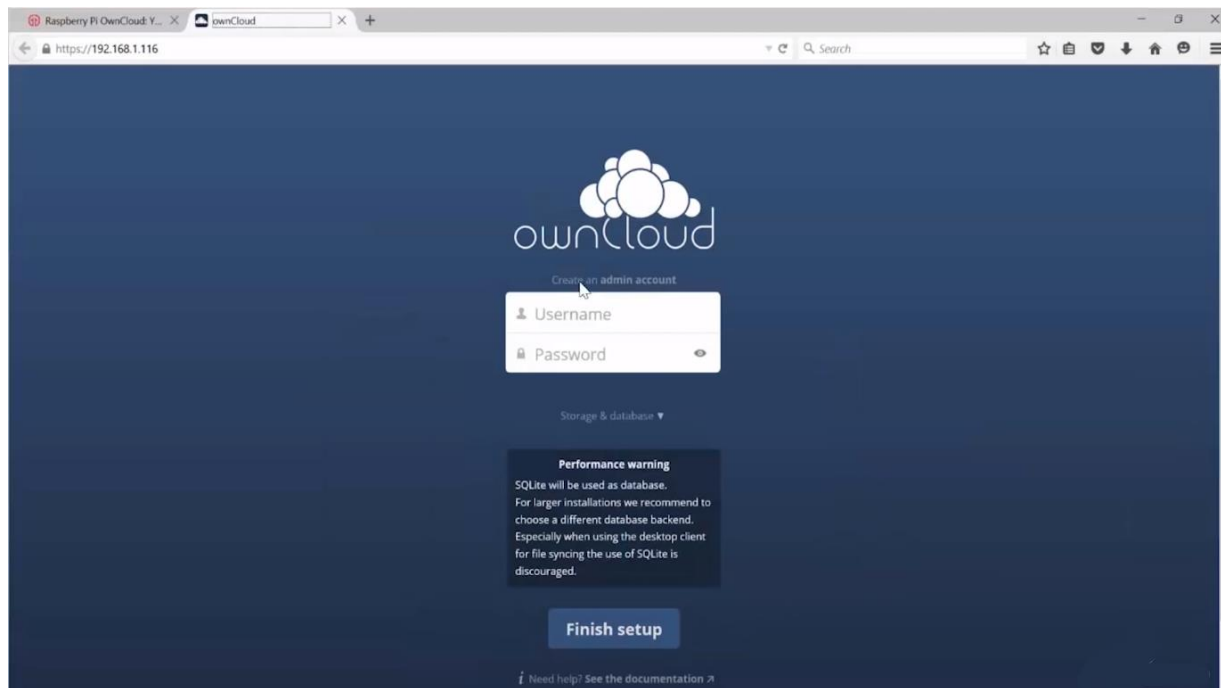


Fig.5 After setting the owncloud server and mounting the hard drive, enter the Pi's IP address onto the browser. The above-mentioned screen will appear. Enter the desired username and password to create an admin account and also mention the external hard drive information by clicking on storage and database.

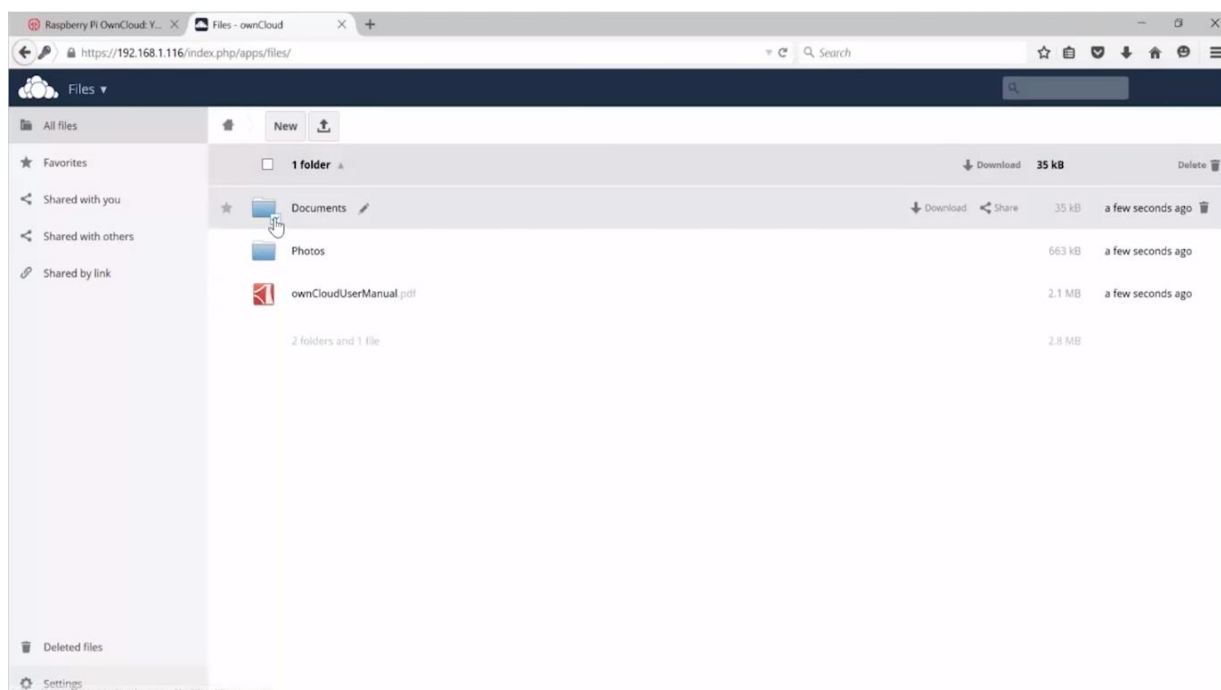


Fig.6 After creating the admin account and logging in this screen will appear which will show to data that is stored in the external hard drive. The data stored can be accessed here.

Introduction

When we are out of storage on our device then cloud storage is an option to store data over the internet and access it whenever we want. Some cloud storage service providers are Dropbox, Google Drive, iCloud. Cloud storage helps in storing data over the internet. This prevents users from carrying unnecessary storage devices, they just have to create an account on one of these above-mentioned cloud services and pay on a monthly or yearly basis to use cloud storage services. With an increase in data over time, storing it becomes important as many important things rely on the data. Buying expensive storage devices and carrying them around becomes difficult for the user. Here cloud storage comes in the limelight. With cloud storage, the user does not have to buy any expensive storage device and neither bears the pain of carrying them. Cloud service providers provide the cloud storage to the users and users can access their data using any device which has an internet-enabled in it.

1. Description

To create your cloud storage you will need Raspberry Pi and an external hard disk drive to convert it into cloud storage, and a Micro SD card, which will act as ROM for the Raspberry Pi. You can use ownCloud as a platform to access your cloud storage, manipulate data in it, and upload or delete data in your cloud storage. Although there are many other files hosting services like Dropbox, Google Drive, iCloud, you can create your cloud storage and use ownCloud as secured access to your cloud storage drive.

This Raspberry Pi will act as a server, but first you have configured it to save data that is stored from the ownCloud. To do this you have to install ownCloud on your Raspberry Pi.

You can use 500GB- 1TB external hard drive as your cloud storage. To access drive from any part of the world, you have to have an internet connection.

The different requirements of this research work are:

Hardware Requirements:

1. Raspberry Pi
2. Raspberry Pi power supply 5v/10v
3. Micro-SD card 4/8 GB class 10
4. Internet connection (LAN/Wifi)
5. External hard drive

Software Requirements:

1. Raspbian OS
2. Linux terminal/Putty
3. Constant internet connection
4. Application (Owncloud)

2. Purpose

The purpose of this project is to create cloud storage for our personal use rather than relying on third-party cloud storage services like Dropbox, Google Drive, iCloud, etc. Using Raspberry Pi, we can use our external hard drive as cloud storage for our personal use only. We can decide the amount of memory by ourselves and can have full control over our data. Using ownCloud we can access our data in our external hard drive through any device which has internet, treating our external hard drive as a cloud storage device.

3. Motivations and Scope

In this dynamic environment and ever-changing technologies, the security of our data is of utmost importance as well as the amount of storage we need to store the data and control over our data. This project primarily focuses on the above-mentioned problems. As third party cloud services are open to other users too, so this creates an issue of privacy. As well as these cloud service providers provide a limited amount of storage and they have some control over our data too. And we have to pay some hefty amount of money to use these cloud services. Using Raspberry Pi we can use our external hard drive as cloud storage for our personal use only. We can decide the amount of memory by ourselves and can have full control over our data. Using ownCloud we can access our data in our external hard drive through any device which has internet, treating our external hard drive as a cloud storage device.

Existing System

With so many new technologies emerging, Cloud computing is one of the most used technology in the current time. Cloud computing has solved many major problems like Mobility, Security and disaster recovery, Scalability and Flexibility, Cost control. Cloud computing has a major part in it which is cloud storage. When we are out of storage on our device then cloud storage is an option to store data over the internet and access it whenever we want. Some cloud storage service providers are Dropbox, Google Drive, and iCloud. Cloud storage helps in storing data over the internet. This prevents users from carrying unnecessary storage devices, they just have to create an account on one of these above-mentioned cloud services and pay on a monthly or yearly basis to use cloud storage services. With an increase in data over time, storing it becomes important as many important things rely on the data. Buying expensive storage devices and carrying them around becomes difficult for the user. Here cloud storage comes in the limelight. With cloud storage, the user does not have to buy any expensive storage device and neither bears the pain of carrying them. Cloud service providers provide the cloud storage to the users and users can access their data using any device that has internet-enabled in it.

Problem Statement:

Right now and consistently evolving innovations, security of our information is most extreme significant just as the measure of capacity we have to store the information and power over our information. Below mentioned are challenges that users are facing with the third party cloud storage services:

1. As third party cloud services are open to other users too, so this creates an issue of security. As well as these cloud service providers provide a limited amount of storage and they have some control over our data too.
2. We have to pay some hefty amount of money to use these cloud services.
3. Buying expensive storage devices and carrying them around becomes difficult for the users.

Proposed System

In this dynamic environment and ever-changing technologies, the security of our data is of utmost importance as well as the amount of storage we need to store the data and control over our data. This project primarily focuses on the above-mentioned problems. As third party cloud services are open to other users too, so this creates an issue of privacy. As well as these cloud service providers provide a limited amount of storage and they have some control over our data too. And we have to pay some hefty amount of money to use these cloud services. Using Raspberry Pi we can use our external hard drive as cloud storage for our personal use only. We can decide the amount of memory by ourselves and can have full control over our data. Using ownCloud we can access our data in our external hard drive through any device which has internet, treating our external hard drive as a cloud storage device.

To create your cloud storage you will need Raspberry Pi and an external hard disk drive to convert it into cloud storage, and a micro SD card of minimum 8 GB of class 10 that will act as ROM for the Raspberry Pi. You can use ownCloud as a platform to access your cloud storage, manipulate data in it, and upload or delete data in your cloud storage. Although there are many other files hosting services like Dropbox, Google Drive, iCloud, you can create your cloud storage and use ownCloud as secured access to your cloud storage drive.

This Raspberry Pi will act as a service, but first you have configured it to save data that is stored from the ownCloud. To do this you have to install ownCloud on your Raspberry Pi. You can use 500GB- 1TB external hard drive as your cloud storage. To access drive from any part of the world, you have to have an internet connection.

The different requirements of this research work are:

Hardware Requirements:

1. **Raspberry Pi:** A device which is used for low computing operations. It serves the purpose of a mini computer. It has Double band LAN, USB ports, HDMI port. It can be used as a server also.
2. **Raspberry Pi power supply 5v/10v:** To power up this device at least 5v-10v power supply is needed. A power supply adapter with appropriate pin type should be chosen.
3. **Micro-SD card 4/8 GB class 10:** MicroSD card serves the purpose of rom of the raspberry pi. Entire raspbian os is installed on this MicroSD card. For proper functioning of this device at least class 10 or U1 MicroSD cards should be used.
4. **Internet connection (LAN/Wifi):** Internet connection is a must in setting up the entire set-up as well as further using it. As all the commands to be executed need internet and further tasks to be performed too. A LAN or Wifi is recommended.
5. **External hard drive:** An external hard drive is needed to make it a centralised cloud storage. It needs to be mounted before using it. Then it needs to be linked with Owncloud so that the files stored in it can be accessed.

Software Requirements:

1. **Raspbian OS:** It is an operating system which has been designed specifically for the Raspberry Pi device. It is based on Unix/Linux. It consists of Python, Scratch, Sonic Pi, Java and that's just beginning.
2. **Linux terminal/Putty:** Linux can be used to configure Owncloud server with the Raspberry Pi. For Windows users Putty can be used which is an open source terminal emulator. Otherwise Raspbian OS is based on Linux so it has a built-in terminal.
3. **Constant internet connection:** Internet connection is a must for configurations and further using it. Owncloud would also need the internet connection for its work.
4. **Application (Owncloud):** Owncloud is a suite for file hosting services. It is the same as Dropbox but the distinction is that Owncloud does not offer data centre capacity. It is open source. Users don't have to pay any amount for their private servers.

Implementation

Raspberry Pi Set-up –

1. **Download Raspbian:** We can download Raspbian from the internet.
2. **Unzip the file:** We have to unzip the Raspbian disk image file.
3. **Load Raspbian OS into your microSD:** This is done by connecting the microSD card to the desktop and using Win32 disk imager.
4. **Insert microSD card in the Raspberry Pi device:** After the raspbian has been installed onto the microSD card then put it into the device. Put username as **pi** and password **raspberry**, as they are default.

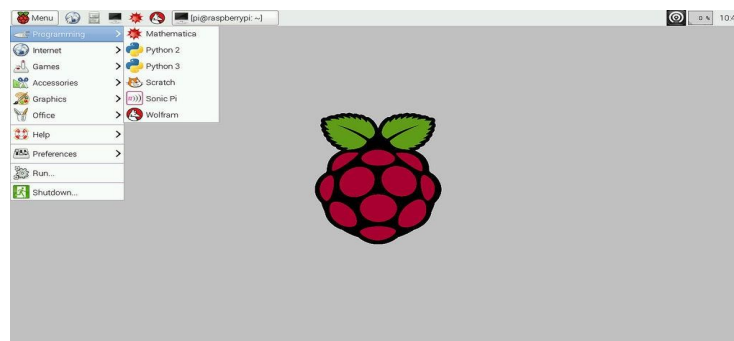


Fig.1 User interface of Raspbian OS

Setting Owncloud Server on the Raspberry Pi –

1. Firstly, in either The Pi's command line or [via SSH](#) we will need to update the Raspberry Pi and its packages, do this by entering:


```
sudo apt-get update
```

```
sudo apt-get upgrade
```

2. Now we need to open up the Raspi Config Tool to change a few settings.

```
sudo raspi-config
```

3. In here we will need to change a few settings.

- Change Locale to en_US.UTF8 in internationalization options -> change local.
- Change memory split to 16m in Advanced options -> Memory split.
- Change overclock to medium.

4. Add the www-data user to the www-data group.

```
sudo usermod -a -G www-data www-data
```

These instructions have been updated to work with Raspbian Buster, if you're on an earlier version then I highly recommend you upgrade to Raspbian Buster before continuing.

5. Once you are running Raspbian Buster, you can safely continue with this tutorial.

In this step, we will be installing all the packages that we require to run Owncloud. This includes php7.2 and its numerous modules that OwnCloud relies upon.

Run the following command to install everything we need.

```
sudo apt-get install nginx openssl ssl-cert php7.2-cli php7.2-sqlite3 php7.2-gd php7.2-
```

```
common php7.2-cgi sqlite3 php-pear curl libapr1 libtool curl libcurl4-openssl-dev
```

```
php7.2-xml php7.2 php7.2-dev php7.2-curl php7.2-gd php7.2-fpm memcached php-
```

```
memcache php7.2-zip php7.2-intl php7.2-mbstring varnish
```

6. Now we need to create an SSL certificate you can do this by running the following command:

```
sudo openssl req @$ -new -x509 -days 730 -nodes -out /etc/nginx/cert.pem -keyout
```

```
/etc/nginx/cert.key
```

Just enter the relevant data for each of the questions it asks you.

7. In addition to the SSL certificate we also need to generate ourselves a custom dhparam file. This file helps ensure that our SSL connections are kept secure, by default this would use a default one that isn't nearly as secure.

To generate a 4096 byte long DH Param run the following command on your Raspberry Pi. This process will take quite a long time, up to 16 hours. Adding the **-dsaparam** flag to the command will help speed up the process, but arguably is less secure.

```
sudo openssl dhparam -out /etc/nginx/dh4096.pem 4096
```

8. Now we need to chmod the three cert files we just generated.

```
sudo chmod 600 /etc/nginx/cert.pem
```

```
sudo chmod 600 /etc/nginx/cert.key
```

```
sudo chmod 600 /etc/nginx/dh4096.pem
```

9. Let's clear the server config file since we will be copying and pasting our own version in it.

```
sudo sh -c "echo " > /etc/nginx/sites-available/default"
```

10. Now let's configure the web server configuration so that it runs Owncloud correctly.

```
sudo nano /etc/nginx/sites-available/default
```

11. Now simply copy and paste the following code into the file.

```
upstream php-handler {
```

```
server unix:/var/run/php/php7.2-fpm.sock;
```

```
}
```

```
server {
```

```
listen 80;
```

```
server_name _;
```

```
#Allow letsencrypt through
```

```
location /.well-known/acme-challenge/ {
```

```
root /var/www/owncloud;
```

```
}
```

```
# enforce https
```

```
location / {
```

```
return 301 https://$host$request_uri;
```

```
}
```

```
}
```

```
server {
```

```
listen 443 ssl http2;
```

```
server_name _;
```

```
ssl_certificate /etc/nginx/cert.pem;
```

```
ssl_certificate_key /etc/nginx/cert.key;
```

```
ssl_session_timeout 5m;
```

```
ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
```

```
ssl_ciphers 'ECDHE-RSA-AES128-GCM-  
SHA256:AES256+EECDH:AES256+EDH';
```

```
ssl_dhparam /etc/nginx/dh4096.pem;
```

```
ssl_prefer_server_ciphers on;
```

```
keepalive_timeout 70;
```

```
ssl_stapling on;
```

```
ssl_stapling_verify on;
```

```
add_header X-Content-Type-Options nosniff;
```

```
add_header X-Frame-Options "SAMEORIGIN";
```

```
add_header X-XSS-Protection "1; mode=block";
```

```
add_header X-Robots-Tag none;
```

```
add_header X-Download-Options noopen;
```

```
add_header X-Permitted-Cross-Domain-Policies none;
```

```
root /var/www/owncloud/;
```

```
location = /robots.txt {
```

```
allow all;
```

```
log_not_found off;
```

```
access_log off;
```

```
}
```

```
# The following 2 rules are only needed for the user_webfinger app.
```

```
# Uncomment it if you're planning to use this app.
```

```
#rewrite ^/.well-known/host-meta /public.php?service=host-meta last;
```

```
#rewrite ^/.well-known/host-meta.json /public.php?service=host-meta-json last;
```

```
location = /.well-known/carddav {
```

```
return 301 $scheme://$host/remote.php/dav;
```

```
}
```

```
location = /.well-known/caldav {
```

```
return 301 $scheme://$host/remote.php/dav;
```

```
}
```

```
# set max upload size
```

```
client_max_body_size 512M;
```

```
fastcgi_buffers 8 4K;
```

```
fastcgi_ignore_headers X-Accel-Buffering;
```

```
gzip off;
```

```
error_page 403 /core/templates/403.php;
```

```
error_page 404 /core/templates/404.php;
```

```
location / {
```

```
rewrite ^ /index.php$uri;
```

```
}
```

```
location ~ ^/(?:\.|\.|autotest|occ|issue|indie|db_|console)/ {
```

```
return 404;
```

```
}
```

```
location ~ ^/(?:\.|\.|autotest|occ|issue|indie|db_|console) {
```

```
return 404;
```

```
}
```

```
location ~
```

```
^(?:index|remote|public|cron|core/ajax/update|status|ocs/v[12]|updater/.+|ocs-
```

```
provider/.+|core/templates/40[34])\.php(?:$|/) {
```

```
fastcgi_split_path_info ^(.+\.(php|\.*)$);
```

```
include fastcgi_params;
```

```
fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
```

```
fastcgi_param SCRIPT_NAME $fastcgi_script_name;
```

```
fastcgi_param PATH_INFO $fastcgi_path_info;
```

```
fastcgi_param HTTPS on;
```

```
fastcgi_param modHeadersAvailable true;
```

```
fastcgi_param front_controller_active true;
```

```
fastcgi_read_timeout 180;
```

```
fastcgi_pass php-handler;
```

```
fastcgi_intercept_errors on;
```

```
fastcgi_request_buffering off; #Available since NGINX 1.7.11
```



```
}
```

```
location ~ ^/(?:(?:updater|ocs-provider)(?:$|/)) {
```

```
try_files $uri $uri/ =404;
```

```
index index.php;
```

```
}
```

```
location ~ \.(?:css|js)$ {
```

```
try_files $uri /index.php$uri$is_args$args;
```

```
add_header Cache-Control "max-age=15778463";
```

```
# Before enabling Strict-Transport-Security headers please read into this topic first.
```

```
#add_header Strict-Transport-Security "max-age=15552000; includeSubDomains";
```

```
add_header X-Content-Type-Options nosniff;
```

```
add_header X-Frame-Options "SAMEORIGIN";
```

```
add_header X-XSS-Protection "1; mode=block";
```

```
add_header X-Robots-Tag none;
```

```
add_header X-Download-Options noopen;
```

```
add_header X-Permitted-Cross-Domain-Policies none;
```

```
access_log off;
```

```
}  
  
location ~ \.(?:svg|gif|png|html|ttf|woff|ico|jpg|jpeg|map)$ {  
  
    add_header Cache-Control "public, max-age=7200";  
  
    try_files $uri /index.php$uri$is_args$args;  
  
    access_log off;  
  
}  
  
}
```

12. Now simply save and exit.

13. Now that is done there is a few more configurations we will need to update, first open up the PHP config file by entering.

```
sudo nano /etc/php/7.2/fpm/php.ini
```

14. In this file, we want to find and update the following lines. (*Ctrl + w* allows you to search)

```
upload_max_filesize = 2000M
```

```
post_max_size = 2000M
```

15. Once done, save and then exit. Now we also need to edit the `dphys-swapfile`. To do this open up the file by entering:

```
sudo nano /etc/dphys-swapfile
```

16. Now update the `conf_swapsz` line to the following:

```
CONF_SWAPSIZE = 512
```

17. Restart the Pi by entering:

```
sudo reboot
```

18. Once the Pi has restarted, you will need to install Owncloud onto the Raspberry Pi.

Do this by entering the following commands:

```
sudo mkdir -p /var/www/owncloud
```

```
cd /var/www/
```

```
curl https://download.owncloud.org/community/owncloud-10.2.1.tar.bz2 | sudo tar -jxv
```

```
sudo chown -R www-data:www-data /var/www
```

19. Now we need to open up the `.user.ini` file to enforce some of the changes we made earlier in the tutorial

```
sudo nano /var/www/owncloud/.user.ini
```

20. In here update the following values so they are 2000M:

```
upload_max_filesize=2000M
```

```
post_max_size=2000M
```

```
memory_limit=2000M
```

21. Now that is done we should be able to connect to Owncloud at your PI's IP address.

Before you set up the admin account, you might want to mount an external drive, so you have lots of disk space for your Raspberry Pi Owncloud Server. Just follow the instructions in the next section.

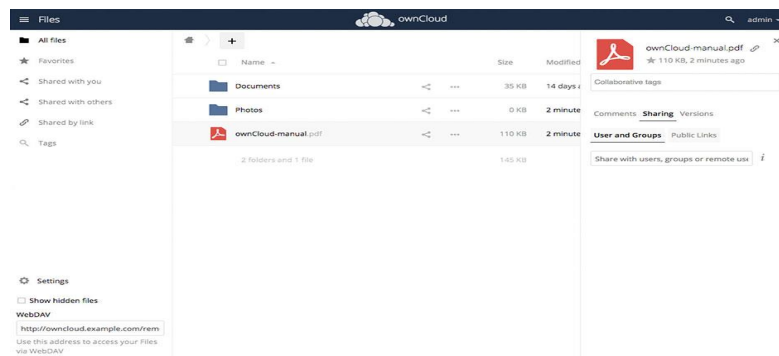


Fig.2 Owncloud User Interface

Mounting and Setting Hard Drive –

These instructions are for mounting and allowing Owncloud to store files onto an external hard drive.

1. Firstly if you have an NTFS drive we will need to install an NTFS package by entering the following:

```
sudo apt-get install ntfs-3g
```

2. Now let's make a directory we can mount.

```
sudo mkdir /media/owncloudrive
```

3. Now we need to get the *GID*, *UID*, and the *UUID* as we will need to use these soon. Enter the following command for the *GID*:

```
id -g www-data
```

4. Now for the *UID* enter the following command:

```
id -u www-data
```

5. Also if we get the *UUID* of the hard drive, the Pi will remember this drive even if you plug it into a different USB port.

```
ls -l /dev/disk/by-uuid
```

```
pi@raspberrypi ~ $ ls -l /dev/disk/by-uuid
total 0
lrwxrwxrwx 1 root root 15 Aug 28 08:08 13d368bf-6dbf-4751-8ba1-88bed06bef77 -> .
./../mmcblk0p2
lrwxrwxrwx 1 root root 15 Aug 28 08:08 15CD-3B79 -> ./../mmcblk0p1
lrwxrwxrwx 1 root root 10 Aug 28 08:08 DC72-0315 -> ./../sda1
```

Copy the light blue letters and numbers of the last entry (Should have something like -> ./../sda1 at the end of it).

6. Now let's add your drive into the fstab file so that it will boot with the correct permissions.

```
sudo nano /etc/fstab
```

7. Now add the following line to the bottom of the file, updating *UID*, *GUID* and the *UUID* with the values we got above. (The following should all be on a single line)

```
UUID=DC72-0315 /media/owncloudrive auto
```

```
nofail,uid=33,gid=33,umask=0027,dmask=0027,noatime 0 0
```

8. Reboot the Raspberry Pi, and the drives should automatically be mounted. If they are mounted, we're all good to go.

Note: If you get an error stating the Pi is in emergency mode at boot up then this likely means a problem with the fstab entry. Just edit the fstab file (`sudo nano /etc/fstab`) and remove the added line or look for a mistake and fix it.

Setting up Owncloud Account –

1. In your browser enter your Pi's IP address.
2. Once you go to the IP you're like to get a certificate error, simply add this to your exception list as it will be safe to proceed.
3. When you first open up ownCloud you should be presented with a simple setup screen and no errors.
4. Enter your desired username and password.
5. Click on storage & database and enter your external drive /media/ownclouddrive (Skip this step if you didn't setup an external drive).
6. Click finish setup.

Result

The screenshots are shown below display the final output of each step mentioned above.

```
pi@raspberrypi: /var/www/owncloud
GNU nano 2.2.6 File: .user.ini Modified
upload_max_filesize=2000M
post_max_size=2000M
memory_limit=2000M
mbstring.func_overload=0
always_populate_raw_post_data=-1
default_charset='UTF-8'
output_buffering=off
File Name to Write: .user.ini
^G Get Help      M-D DOS Format  M-A Append     M-B Backup File
^C Cancel        M-M Mac Format  M-P Prepend
```

Fig.3 After saving and then exiting from this screen the setup of owncloud server will be completed.

```
pi@raspberrypi /var/www/owncloud
pi@raspberrypi /var/www/owncloud $ id -g www-data
33
pi@raspberrypi /var/www/owncloud $ id -u www-data
33
pi@raspberrypi /var/www/owncloud $ ls -l /dev/disk/by-uuid
total 0
lrwxrwxrwx 1 root root 15 Aug 29 06:25 13d368bf-6dbf-4751-8ba1-88bed06bef77 -> .
../mmcblk0p2
lrwxrwxrwx 1 root root 15 Aug 29 06:25 15CD-3B79 -> ../mmcblk0p1
lrwxrwxrwx 1 root root 10 Aug 29 06:51 DC72-0315 -> ../sda1
pi@raspberrypi /var/www/owncloud $ sudo nano /etc/fstab
pi@raspberrypi /var/www/owncloud $ sudo reboot

Broadcast message from root@raspberrypi (pts/0) (Sat Aug 29 06:57:02 2015):
The system is going down for reboot NOW!
pi@raspberrypi /var/www/owncloud $
```

Fig.4 The external hard drive has been mounted as well as rebooted.

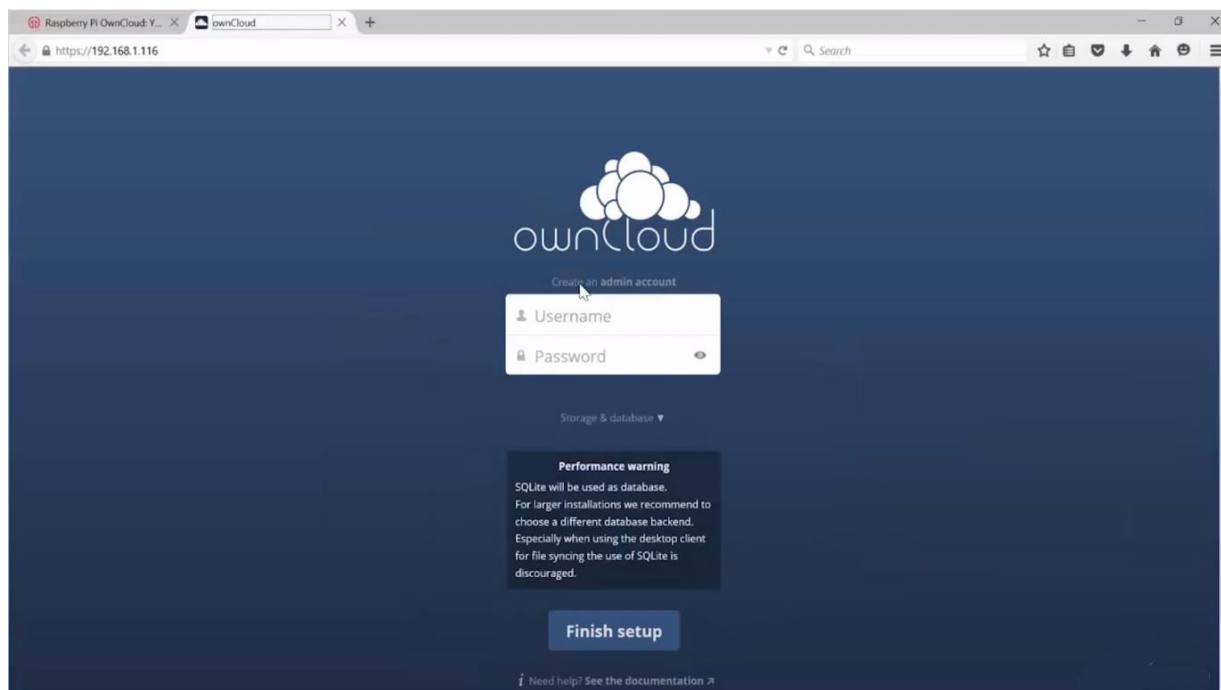


Fig.5 After setting the owncloud server and mounting the hard drive, enter the Pi's IP address onto the browser. The above-mentioned screen will appear. Enter the desired username and password to create an admin account and also mention the external hard drive information by clicking on storage and database.

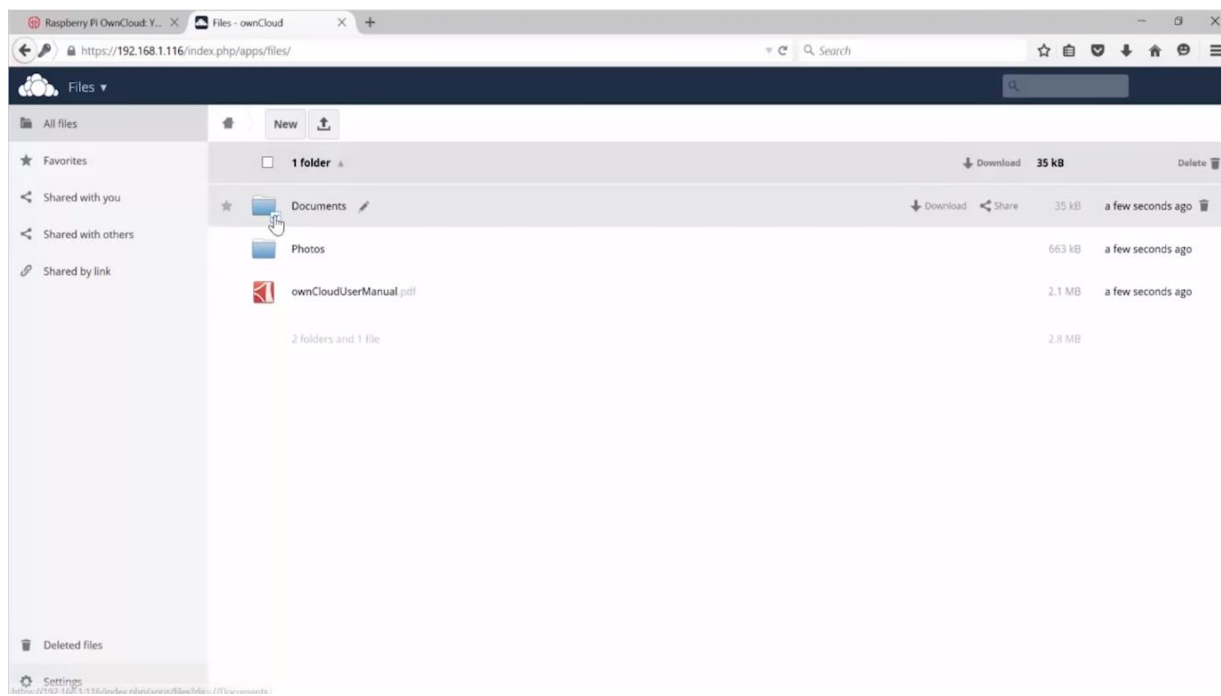


Fig.6 After creating the admin account and logging in this screen will appear which will show to data that is stored in the external hard drive. The data stored can be accessed here.

Conclusion and Future Enhancement

The individual distributed storage with Raspberry pi gave a great deal advantages, for example, Cloud administrations with no charges, likewise client can decide their extra room utilizing their own hard plate what's more, included security highlights including encryption. With this security highlights will guarantee all information secure. Client can have utilized hard plate with enormous space consequently client can store the huge of information on it. Moreover, client can access to their own information from any anyplace with this distributed storage administrations if get to web. This sort of Cloud administrations with utilizing Raspberry Pi is only a model and it actualized with sensible cost, if there are a few highlights need to add on, for occurrence upgrade ease of use, security or equipment get harmed. For improve security, engineer need to arrange and discover reasonable like make secret key on hard circle and Raspberry pi. Subsequently, it becomes increasingly secure.

At that point, Raspberry may get harm due to lifetime. To tackle this issue, engineer give a cooling framework to the model to broaden the life time subsequently it would be some cost troubles for this task. Next, for upgrade ease of use is likewise remembering for highlights for let numerous clients can access into framework. As an end, by and large venture documentation has been continuing with appropriate directions effectively. The basic role of utilizing distributed storage is to store significant document, access the data from anyplace and we can back up records in the event that work area or PC has issue that may all records get harmed. Thus, individual distributed storage utilizing Raspberry Pi have been looked to constrain or repress the hindrances and shortcomings of current business Cloud administration that utilized by client.

Some authors have narrated about the features of the Raspberry Pi, its prototype, Pi in Cloud and edge, and its future. In this authors also have given the direction regarding how the industry is moving from Cloud to edge. Authors have developed renewable energy monitoring system which is used to reduced cost with commercial hardware and software allowing online distribution and free uses. Apart from that it is also helpful for a remotely located plant. Authors have developed a motion detection system using Raspberry Pi using Android application. Authors have proposed a cost benefited low-cost computing Raspberry Pi (minicomputer) which can be used for University classroom purposes. Authors have introduced a thought regarding a computerization framework for the users to control home electronic machines at high versatility and securely. Last but not the least authors have narrated a remote sensor arrange framework planned with Arduino, Raspberry Pi, XBee, and various open-source programming bundles that give minimal effort, smaller, versatile, simple to alter, simple to send, and simple to look after highlights.

References

- Gus, (2015). Raspberry Pi OwnCloud: Your Own Personal Cloud Storage. Available:
<https://pimylifeup.com/raspberry-pi-owncloud/>
- Fairuz Rauf, Maalik Ithing, Zuraidy Adnan (2018) Available:
<https://www.ijcaonline.org/archives/volume180/number22/rauf-2018-ijca-916331.pdf>
- Naveen Lakkundi (2018). Personal Cloud using Raspberry Pi Available:
<https://www.skyfilabs.com/project-ideas/personal-cloud-using-raspberry-pi>
- https://en.wikipedia.org/wiki/Raspberry_Pi (Accesed on 24th March 2020).
- <https://www.infoworld.com/article/3532310/raspberry-pi-is-your-new-private-cloud.html> (Accesed on 24th March 2020).