Build Your Own Cloud Storage File Server With A Raspberry Pi

As time progresses, removable storage is rapidly becoming a thing of the past. USB drives, CDs and SD cards get lost, forgotten and damaged and this is invoking a mass exodus towards the so-called ‘cloud storage’. Most services out there offer a hearty free plan with plenty of disk space for day-to-day use, but the more intense user may find themselves evaluating a paid plan. If you’ve got a Raspberry Pi and your internet’s zippy enough, why not save yourself another bill and build your own cloud storage server? You’ll be sure to always have your files with you, and you'll be able to maintain control over your own data, so no vendor lock-in! So get ready and begin learning how to create your own cloud storage file server with a Raspberry Pi.

What You'll Need

- Raspberry Pi
- Compatible power supply
- Ethernet cable
- SD card

1.) Flash the latest version of Raspbian to your SD card with Win32DiskImager. Pop it into the Raspberry Pi once complete and boot up.
2.) Connect to your Raspberry Pi using your favourite SSH client. Log in with the default credentials - the username is ***pi*** and the password is ***raspberry***.
3.) Type sudo raspi-config into the terminal and, using the arrow keys, select 'expand filesystem'. Hit return/enter. This is a fairly quick process that ensures that all of the SD card space is available to be used by Raspbian.

4.) Back at the command line, run a quick update and upgrade. This updates your software and ensures you have access to the latest list of available packages.

```
sudo apt-get update; sudo apt-get upgrade -y
```

5.) It's a very good idea to change the default password at this stage.

To do this, use the sudo passwd command.

6.) Next, we need to install Apache. This is a web server package that will serve up the ownCloud interface. In the same command, we will also install some helper libraries.

```
sudo apt-get install apache2 php5 php5-fpm php5-gd php5-curl php5-mysql mysql-server -y
```

7.) This may take a while. At some point during the installation process, you will be presented with a setup wizard for mysql-server:

Set root password in mysql-server

It's important that you set a root password at this stage for security reasons. Hit enter/return once complete.

8.) Restart Apache to apply changes.

```
sudo service apache2 restart
```

9.) Head over to your Pi's internal IP address (you'll find this by typing ipconfig eth0). You should see the default Apache page:

Apache Welcome Page

If you do, congratulations! You've just installed a web server on your Raspberry Pi.

What you can see here is simply a static HTML file stored in the /var/www/html directory. Let's take a look.

10.) Change directory, and take a look what's inside.

```
cd /var/www/html; ls
```

We don't need the Apache welcome page, so let's delete it.

```
sudo rm index.html
```

11.) Give Apache permissions for the /var/www/ directory.

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

12.) Now, let's grab the ownCloud installer file. This is just a single PHP file which will check the system dependencies over, download ownCloud and install it.

```
sudo wget https://download.owncloud.com/download/community/setup-owncloud.php
```

13.) Head over to your web browser, and type in the hostname or IP address that we found earlier. To recap, you can find this by using the command ipconfig eth0. You should see a file called setup-owncloud.php

Directory listing on Apache

Click this file. You'll be taken to the setup wizard.

ownCloud Setup Wizard

Work your way through until you reach the database configuration section.

14.) Head back to the Terminal. We need to create a new MySQL user. To do this, open the MySQL shell using this command:

```
mysql -u root -p
```

To create the new user, we need to enter this command:

```
CREATE USER 'newuser'@'localhost' IDENTIFIED BY 'password';
```

Of course, substitute *newuser* for a username, and *password* for a good, strong password.

15.) We need to create a new MySQL database for ownCloud.

```
CREATE DATABASE `database_name`;
```

16.) We also need to assign the necessary privileges so that the new user can use this database.

```
GRANT ALL PRIVILEGES ON database_name.* TO 'newuser'@'localhost';
```

It's probably not a good idea to grant all privileges - we should only grant the necessary permissions here. However, for the sake of maintaining the simplicity of the tutorial, we'll proceed by granting all. To make sure these privileges come into effect, we need to flush them with this command:

```
FLUSH PRIVILEGES;
```

We're all done with MySQL for now. You can quit the shell by simply typing quit and hitting return.

Head back to your web browser and create an ownCloud account for yourself, then type in the database details.

If all went well, you should now see something like this:

In which case, **congratulations!** You now have a fully-functioning file server.

What's next?

For security reasons, I'd strongly recommend that you read up on hardening SSH, Apache, and ownCloud. Also consider installing Fail2Ban and ModSecurity. This tutorial is intended as a straightforward guide to get you started with ownCloud, and is by no means exhaustive from a security perspective.