Implementation done / code snippets

24 September 2021 23:51

VirtualBox Network Setup

SSH rule

```
127.0.0.1 1313 10.0.2.15 22
```

Apache2 Access

```
127.0.0.1 5555 10.0.2.15 80
```

`sudo apt-get update`
- *Before installing and setting up SSH on the OwnCloud server you need to run this*

`sudo apt install ssh`
- *This will run the install for SSH*

`sudo apt-get install apache2`
- *This runs the install for Apache2*

`sudo sed -i "s/Options Indexes FollowSymLinks/Options FollowSymLinks/" /etc/apache2/apache2.conf`
- *This is a part of the install process that is recommended to be done in the real world to handle links better*

`sudo systemctl start apache2.service`
- *This command starts the service manually*

`sudo systemctl enable apache2.service`
- *This command lets the Apache2 service start automatically whenever we start our server/service*

`sudo apt-get install mariadb-server mariadb-client -y`
- *This command installs the Maria Database service into our own service. MariaDB is one of the best DBs for Cloud Services so that is why we are using it*

`sudo systemctl stop mariadb.service`
- *Stops MariaDB from running*

`sudo systemctl start mairadb.service`
- *Stars MariaDB*

`sudo systemctl enable mairadb.service`
- *Enables MariaDB to run whenever the server/service is turned on*

`sudo mysql -u root -p`
- *This command allows us to use MariaDB without really setting it up. By default the base username is root and there is no password. But obviously using the root without a password is bad practice. To exit this command line type **exit***

`sudo mysql_secure_installation`
- *For the safety of our services we can run this script which will run through the majority of risks that a new installation of MySQL can bring. I changed the username to stay as root and the password for this example I used the same one as I used for the server (DO NOT DO THIS IN THE REAL WORLD!!!!) **3008**. I removed the anonymous users, blocked remote access and removed the test database*

`sudo systemctl restart mairadb.service`
- *This restarts our MariaDB service (it would be the same as if we stopped and started it again)*

`sudo apt-get install software-properties-common -y`
- *This will run the installation for general software we might need I think?*

`sudo add-apt-reposatory ppa:ondrej/php`
- *The installation command for a repository we needed for our service. After this I ran the **sudo apt-get update** command as well.*

`sudo apt-get install php7.1 libapache2-mod-php7.1 php7.1-common php7.1-mbstring php7.1-xmlrpc`

- *One of a few lines for the installation of various things our service will need.*

```
sudo apt-get install php7.1-soap php7.1-apcu php7.1-smbclient php7.1-ldap php7.1-redis php7.1-gd
php7.1-xml php7.1-intl php7.1-json php7.1-imagick php7.1-mysql php7.1-cli php7.1-mcrypt php7.1-zip
php7.1-curl -y
```
- *Another set of packages we need*

```
sudo nano /etc/php/7.1/apache2/php.ini
```
- *Using this we could search the .ini files for anything we want to change or alter that we want. Also to check things like file opening and how much memory is allowed and so on. Speaking of I raised the **memory_limit = 1024M.***

```
sudo systemctl restart apache2.service
```
- *After making a few alterations in the .ini file I restarted the server to make sure that everything was ok.*

```
CREATE DATABASE owncloud;
```
- *While inside MariaDB, using this command will allow us to insert sql commands, such as creating a new database.*

```
CREATE USER 'owncloduser'@'localhost' IDENTIFIED BY 'singidunum';
```
- *With this line of code we created a user called owncloduser whose password is singidunum.*

```
GRANT ALL ON owncloud.* TO 'owncloduser'@'localhost' IDENTIFIED BY 'singidunum' WITH GRANT OPTION;
```
- *This line of code grants all privileges to our newly created user 'owncloduser'.*

```
FLUSH PRIVILEGES;
```
- *This allows any and all changes that the user wishes to make to be written directly into the database, instead of the commands and changes waiting in the RAM for a while before taking place.*

```
SHOW DATABASE;
```
- *Lists any and all databases located on our server*

```
USE owncloud;
```
- *This allows to now use our newly created database and begin making any changes we wish onto it.*

```
SHOW TABLES;
```
- *Right now our database is empty, but this would allow us to see our tables located in our owncloud database.*

```
exit
```
- *We are now leaving the MySQL enviroment as we need to locate our tmp folder so we can continue setting up service.*

```
cd /tmp
```
- *Heading into our tmp folder*

```
/tmp$ ls
```
-

```
/tmp$ wget https://download.owncloud.org/community/owncloud-10.3.1.zip
```
- *With this we can download our owncloud zip*

sudo apt install unzip
- *Before unzipping our owncloud we first need to install unzip*

```
unzip owncloud-10.3.1.zip
```
- *This actually now unzips the files we downloaded, after this we run the **ls** command to make sure that it is there.*

Before we can run our newly downloaded and unzipped owncloud, we need to move it where our apache2 server is.
```
/tmp$ ls /var/www/html
```
- *Just making sure that we have our index.html, we can now move our owncloud folder here*

```
/tmp$ sudo mv owncloud /var/www/html/owncloud/
```
- *This is the command to move owncloud into the folder where our apache2 server is located.*

```
cd /var/www/html/
```
- *Changing the directory we are interacting with, we run the **ls** command here as well just to check that it was moved correctly (it was)*

```
cd owncloud/
```
- *Entering the new folder, where if we run **ls** we can see a bunch of files of various types, aka everything is there*

```
sudo chown -R www-data:www-data /var/www/html/owncloud
```
- *We need to change the ownership of the users before pushing forward, as our apache2 server will treat anonymous users as data if I understood correctly. Also this will allow public users from the internet to access our server.*

```
sudo chmod -R 755 /var/www/html/owncloud/
```
- *Changing the permission: 7 refers to all users, 5 means the ability to read and execute and the last 5 refers to other. This will cover pretty much everything*

```
sudo nano /etc/apache2/sites-available/owncloud.conf
```
- *We need to create a configuration folder for out service*

The owncloud.conf file:

```
<VirtualHost *:80>
ServerAdmin admin@example.ac.rs
DocumentRoot /var/www/html/owncloud/
ServerName owncloud
ServerAlias owncloud

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

<Directory /var/www/html/owncloud>
Options +FollowSymlinks
AllowOverride All
Require granted all

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

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

ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined

</VirtualHost>
```

```
sudo a2ensite owncloud.conf
```
- *This enables our newly created configuration file to all the sites hosted on our web server*

```
sudo a2enmod rewrite
```
- *This module enables the end users to override their own files*

```
sudo a2enmod headers
```
- *This enables headers for our sites*

```
sudo a2enmod env
```
- *This enables our environments*

```
sudo a2enmod dir
```
- *This allows our end users to browse their directories as well as alter them*

```
sudo a2enmod mime
```
- *Allows more multimedia on our servers*

```
sudo systemctl restart apache2.service
```
- *After setting everything up and hoping/troubleshooting a lot, we run this to make sure our service is configured correctly*

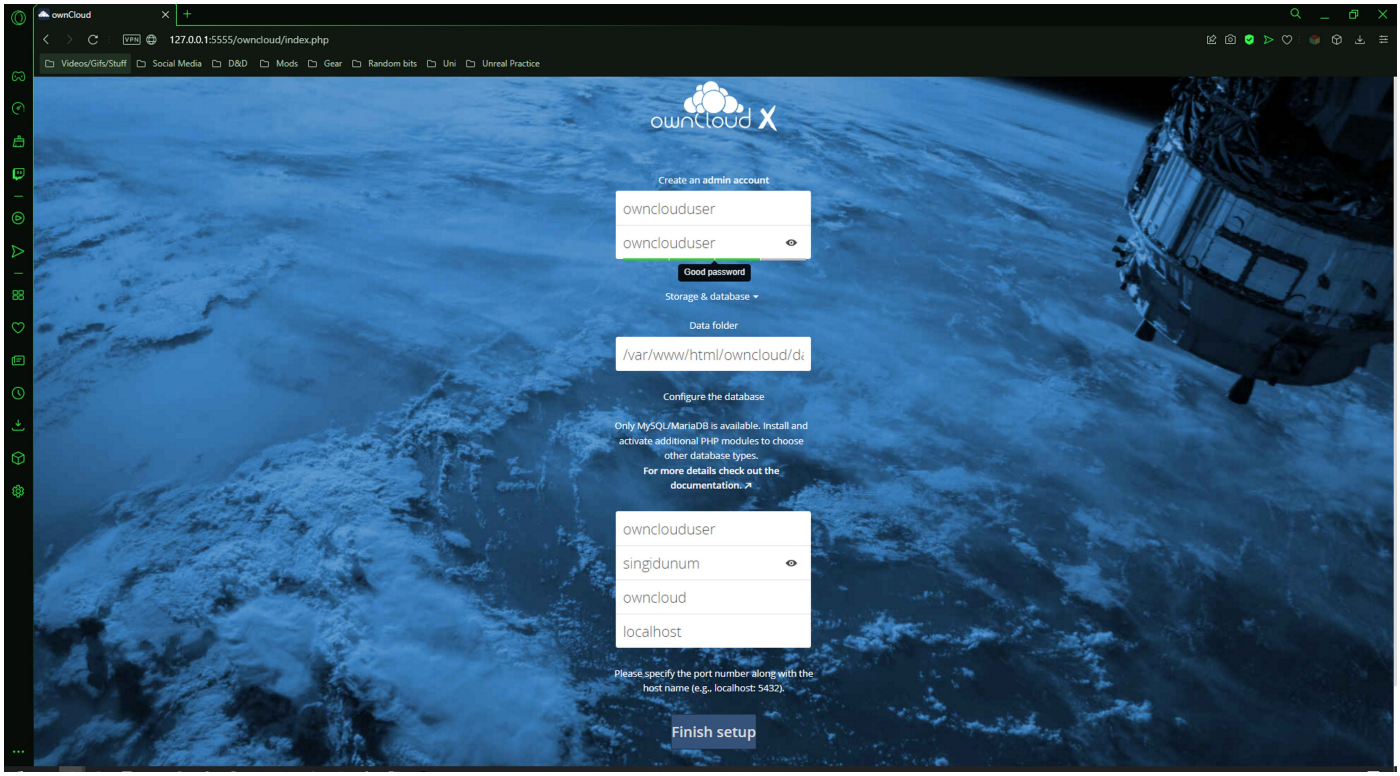
Now in our browser under 127.0.0.1:5555/owncloud we can see that the service is open to be used. On the new page, we create an admin account.

Admin

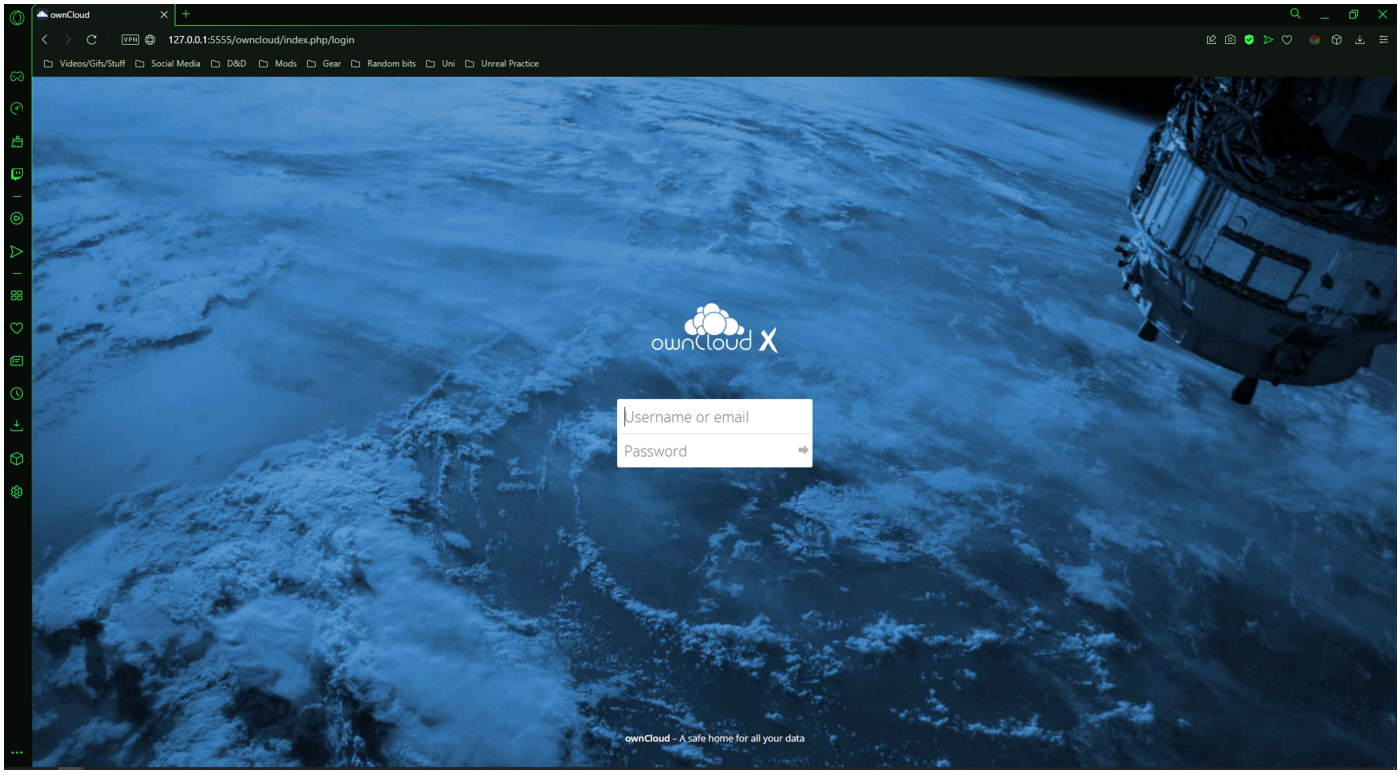
Username: owncloduser Password: owncloduser

New User *(we defined this before when we originally made our database)*

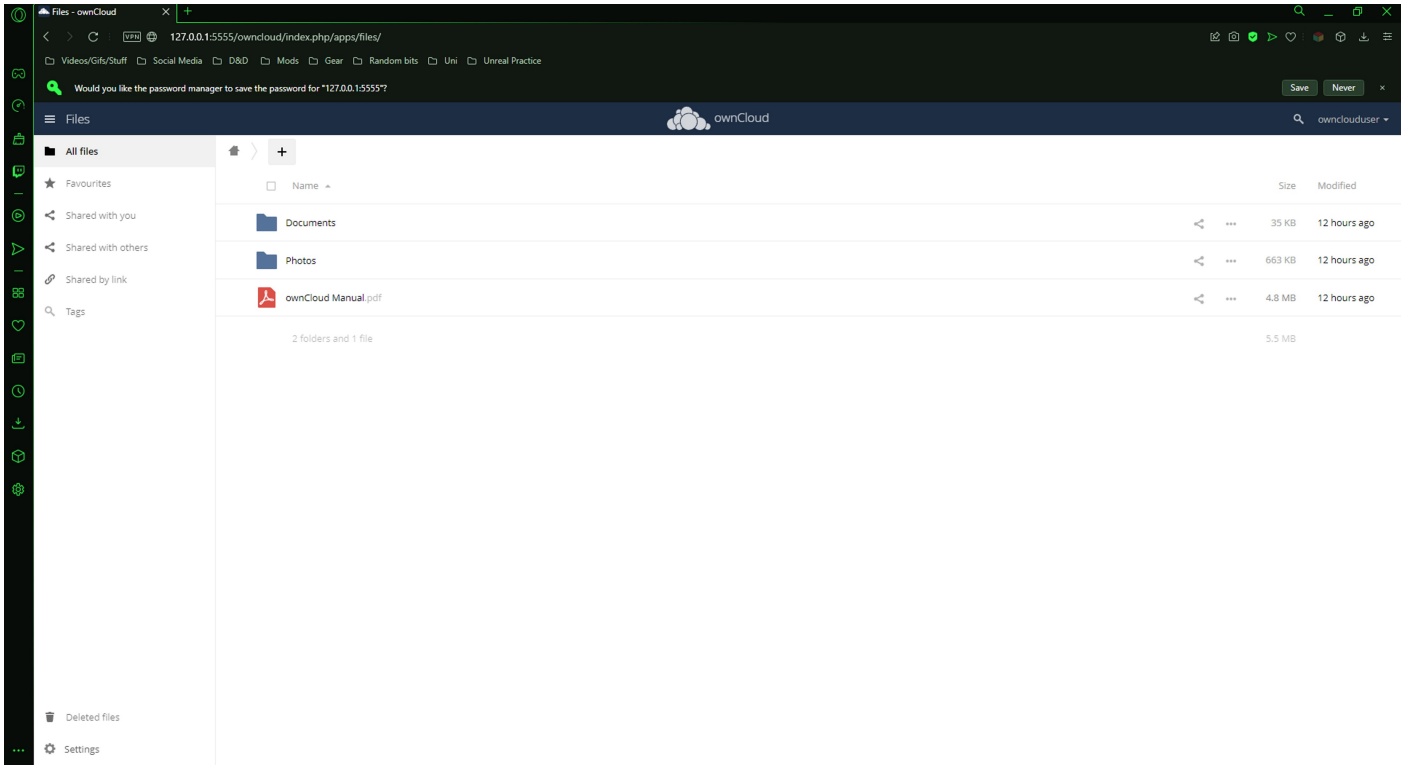
Username: owncloduser Password: singidunum



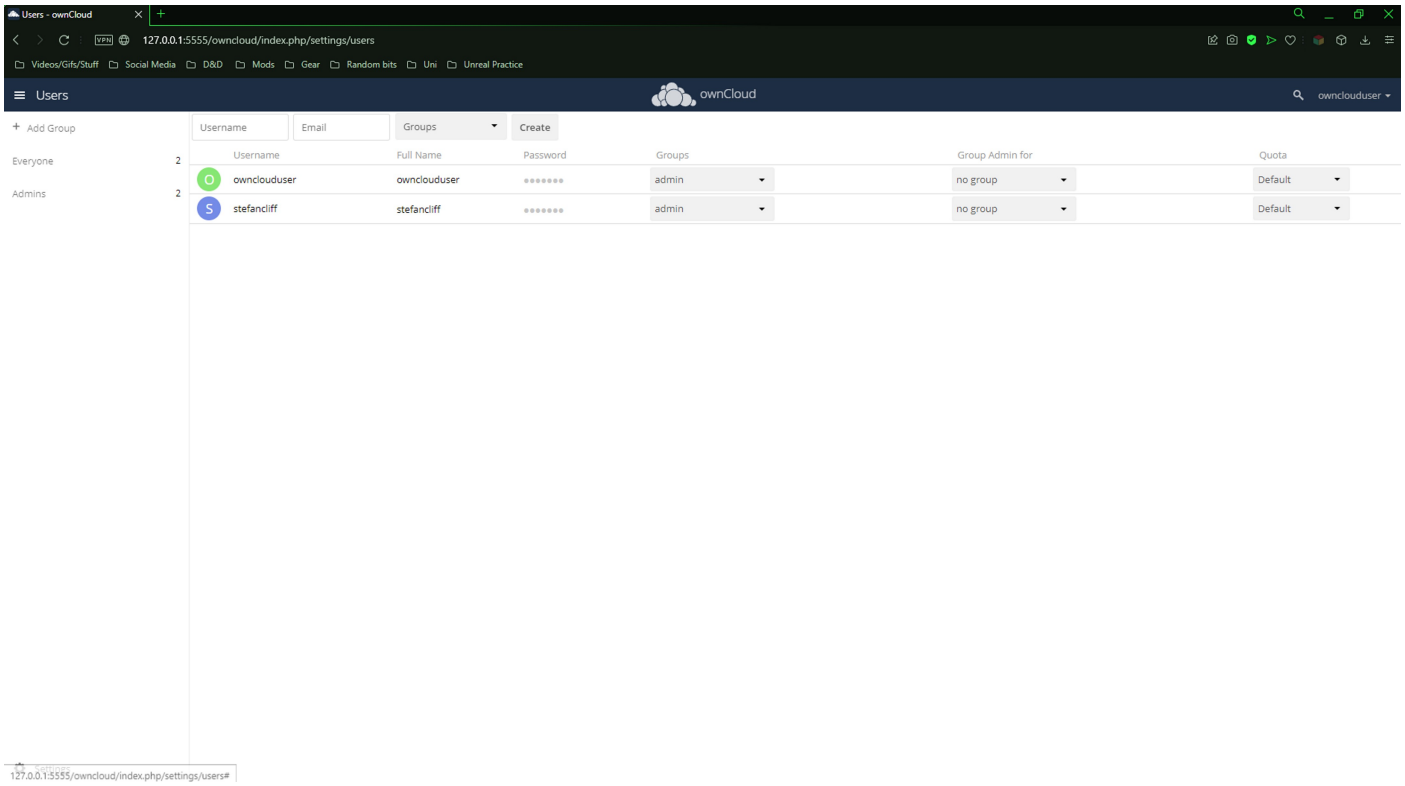
After finishing the setup you are greeted with this screen, using the admin password and username



After logging in you are greeted with this screen



Using the top right we can then go into the users tab where we can quickly add, remove and edit our users



Since everything is finished by running the following command I closed and exited the virtual machine enviroment

```
sudo init 0
```