HOW TO INSTALL APACHE, MYSQL, PHP7.2, PYTHON-DJANGO, WEBMIN, LET'S ENCRYPT AND PHPMYADMIN ON UBUNTU 18.04

root@snet[~]# sudo -s

Then enter your password

root@snet[~]# sudo add-apt-repository universe

root@snet[~]# sudo add-apt-repository multiverse

INSTALLATION OF LAMP (LINUX APACHE MYSQL PHP)

A "LAMP" stack is a group of open source software that is typically installed together to enable a server to host dynamic websites and web apps. This term is actually an acronym which represents the Linux operating system, with the Apache web server. The site data is stored in a MySQL database, and dynamic content is processed by PHP.

INSTALLATION OF APACHE2 SERVER

root@snet[~]# sudo apt-get -y install apache2

INSTALLATION OF MYSQL SERVER

root@snet[~]# sudo apt-get -y install mysql-server mysql-client

root@snet[~]# mysql_secure_installation

Enter password for user root: <-- Enter the MySQL root password

Change the password for root? (Press y|Y for Yes, any other key for No): <-- n

Disallow root login remotely? (Press y|Y for Yes, any other key for No) : <-- y

Remove test database and access to it? (Press y|Y for Yes, any other key for No) : <-- y
Reload privilege tables now? (Press y|Y for Yes, any other key for No) : <-- y

INSTALLATION OF PHP7.2 AND ITS PACKAGES

root@snet[~]# sudo apt-get update -y

root@snet[~]# apt-get install php7.2 libapache2-mod-php7.2 php7.2-cgi php7.2-cli php7.2-common php7.2-curl php7.2-gd php7.2-imap php7.2-intl php7.2-json php7.2-ldap php7.2-mbstring php7.2-mysql php7.2-opcache php7.2-pspell php7.2-readline php7.2-soap php7.2-xml php7.2-zip php7.2-xmlrpc php-imagick php7.2-bcmath php-dompdf php-fpm php-mysql php7.2-memcache php7.2-memcached php7.2-stomp php7.2-xdebug php7.2-gearman php7.2-odbc php7.2-recode php7.2-tidy php7.2-xsl php7.2-enchant php7.2-geoip php7.2-gnupg php7.2-interbase php7.2-mongo php7.2-pinba php7.2-ps php7.2-sybase php7.2-embed php7.2-dba php7.2-phpdbg php7.2-fpm

root@snet[~]# apt-get -y -f install libdbd-mysql-perl libdbi-perl

root@snet[~]# apt-get install build-essential apache2 php7.2 openssl perl make php7.2-gd libgd-dev libapache2-mod-php7.2 libperl-dev libssl-dev daemon wget apache2-utils unzip

root@snet[~]# sudo apt-get install unzip

root@snet[~]# apt-get install php-apcu php-apcu-bc

root@snet[~]# sudo apt install php-apcu redis-server php-redis

root@snet[~]# sudo a2enmod rewrite headers env dir mime

root@snet[~]# sudo service apache2 restart

root@snet[~]# sudo service php7.2-fpm restart

root@snet[~]# sudo apt-get update -y

root@snet[~]# sudo nano /etc/php/7.2/apache2/php.ini

After making the change below, save the file and close out.

post_max_size = 1000M

max_input_time = 5000

memory_limit = 1000M

upload_max_file_size = 1000M

max_execution_time = 5000

CONFIGURE APACHE2 AND PHP-FPM

root@snet[~]# sudo a2dismod php7.2 mpm_prefork

root@snet[~]# sudo a2enmod proxy_fcgi setenvif mpm_event rewrite headers env dir mime ssl http2

root@snet[~]# sudo a2enconf php7.2-fpm

root@snet[~]# sudo nano /etc/php/7.2/fpm/php.ini

After making the change below, save the file and close out.

opcache.enable=1

opcache.enable_cli=1

opcache.interned_strings_buffer=8

opcache.max_accelerated_files=10000

opcache.memory_consumption=128

opcache.save_comments=1

opcache.revalidate_freq=1

root@snet[~]# sudo systemctl restart apache2

root@snet[~]# sudo systemctl restart php7.2-fpm

CONFIGURATION OF REDIS SERVER TO ENABLE CACHING IN UBUNTU SERVER 16.04

root@snet[~]# sudo nano /etc/redis/redis.conf

Now, find and change: ***port 6379 to port 0***

Then uncomment: ***unixsocket /var/run/redis/redis.sock***

unixsocketperm 700 changing permissions to 770 at the same time:
unixsocketperm 770

root@snet[~]# sudo usermod -a -G redis www-data

root@snet[~]# sudo service apache2 restart

root@snet[~]# sudo service redis-server start

root@snet[~]# sudo systemctl enable redis-server

INSTALLATION OF WEBMAIN

Webmin is a web-based interface for system administration for Unix. Using any modern web browser, you can setup user accounts, Apache, DNS, file sharing and much more. Webmin removes the need to manually edit Unix configuration files like /etc/passwd, and lets you manage a system from the console or remotely.

Credit by www.webmin.com/

INSTALLATION COMMANDS

root@snet[~]# sudo nano /etc/apt/sources.list

Add the following line at the bottom of the file:

root@snet[~]# deb http://download.webmin.com/download/repository sarge contrib

root@snet[~]# wget http://www.webmin.com/jcameron-key.asc

root@snet[~]# sudo apt-key add jcameron-key.asc

root@snet[~]# sudo apt-get update

root@snet[~]# sudo apt-get install webmin

INSTALLATION OF PHPMYADMIN

phpMyAdmin is one of the most popular and widely used web-based database management tools. It a free and open source PHP application that allows the users to manage single or multiple SQL database servers locally or on a remote server using a web browser (GUI).

INSTALLATION COMMANDS

root@snet[~]# apt-get -y install phpmyadmin php-mbstring php-gettext

root@snet[~]# sudo systemctl restart apache2

SECURE YOUR PHPMYADMIN

root@snet[~]# sudo nano /etc/apache2/conf-available/phpmyadmin.conf

To Configure Apache‘s .htaccess files

We need to add an AllowOverride All **Then find the line where there is**

<Directory /usr/share/phpmyadmin>

Options FollowSymLinks

DirectoryIndex index.php

AllowOverride All

root@snet[~]# sudo systemctl restart apache2

root@snet[~]# sudo nano /usr/share/phpmyadmin/.htaccess

Add the following information

AuthType Basic

AuthName "Restricted Files"

AuthUserFile /etc/phpmyadmin/.htpasswd

Require valid-user

root@snet[~]# sudo apt-get install apache2-utils

ADDING A USER TO THE PHPMYADMIN AUTHENTICATION

root@snet[~]# sudo htpasswd -c /etc/phpmyadmin/.htpasswd snet

Adding Another User

root@snet[~]# sudo htpasswd /etc/phpmyadmin/.htpasswd stepehen

HOW TO FIX PHPMYADMIN ERRORS WITH PHP7.2

Firstly, backup sql.lib.php before editing.

root@snet[~]# sudo cp /usr/share/phpmyadmin/libraries/sql.lib.php /usr/share/phpmyadmin/libraries/sql.lib.php.bak

root@snet[~]# sudo nano /usr/share/phpmyadmin/libraries/sql.lib.php

Press `CTRL` + `W` and search for `(count($analyzed_sql_results['select_expr'] == 1)`

Replace it with `((count($analyzed_sql_results['select_expr']) == 1)`

Save file and exit. (Press `CTRL` + `X`, press `Y` and then press `ENTER`)

Backup plugin_interface.lib.php

root@snet[~]# sudo cp /usr/share/phpmyadmin/libraries/plugin_interface.lib.php /usr/share/phpmyadmin/libraries/plugin_interface.lib.php.bak

root@snet[~]# sudo nano /usr/share/phpmyadmin/libraries/plugin_interface.lib.php

Press `CTRL` + `W` and search for

```
if ($options != null && count($options) > 0) {
```

Replace with

```
if ($options != null) {
```

Save file and exit. (Press `CTRL` + `X`, press `Y` and then press `ENTER`)

CREATING OF DATABASE FOR OWNCLOUD SERVER

root@snet[~]# mysql -u root -p

root@snet[~]# CREATE DATABASE owncloud;

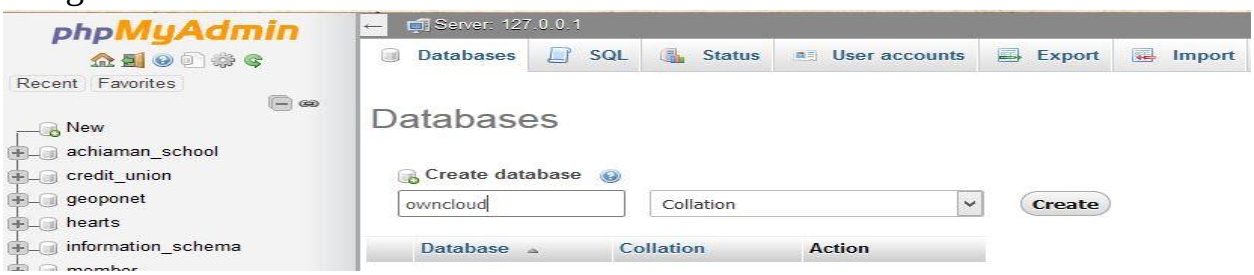
root@snet[~]# GRANT ALL ON owncloud.* to 'owncloud'@'localhost' IDENTIFIED BY 'stephen';

root@snet[~]# FLUSH PRIVILEGES;

root@snet[~]# exit

OR

Navigate to phpmyadmin by http://server's name or IP/phpmyadmin and create database name called "**owncloud**" and click on create that's all. Very easier than using the console or above.

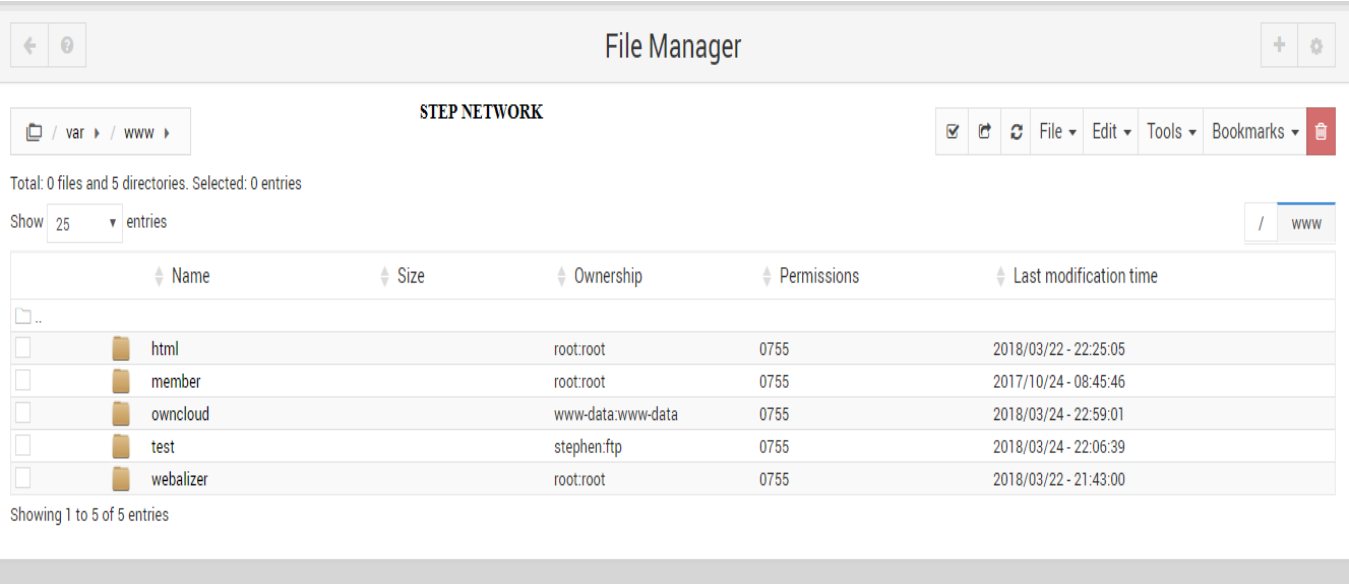


CONFIGURING VIRTUAL HOST USING WEBMIN CONTROL PANEL

Let's login to webmin.

Access the webmin control panel using http://IP:10000

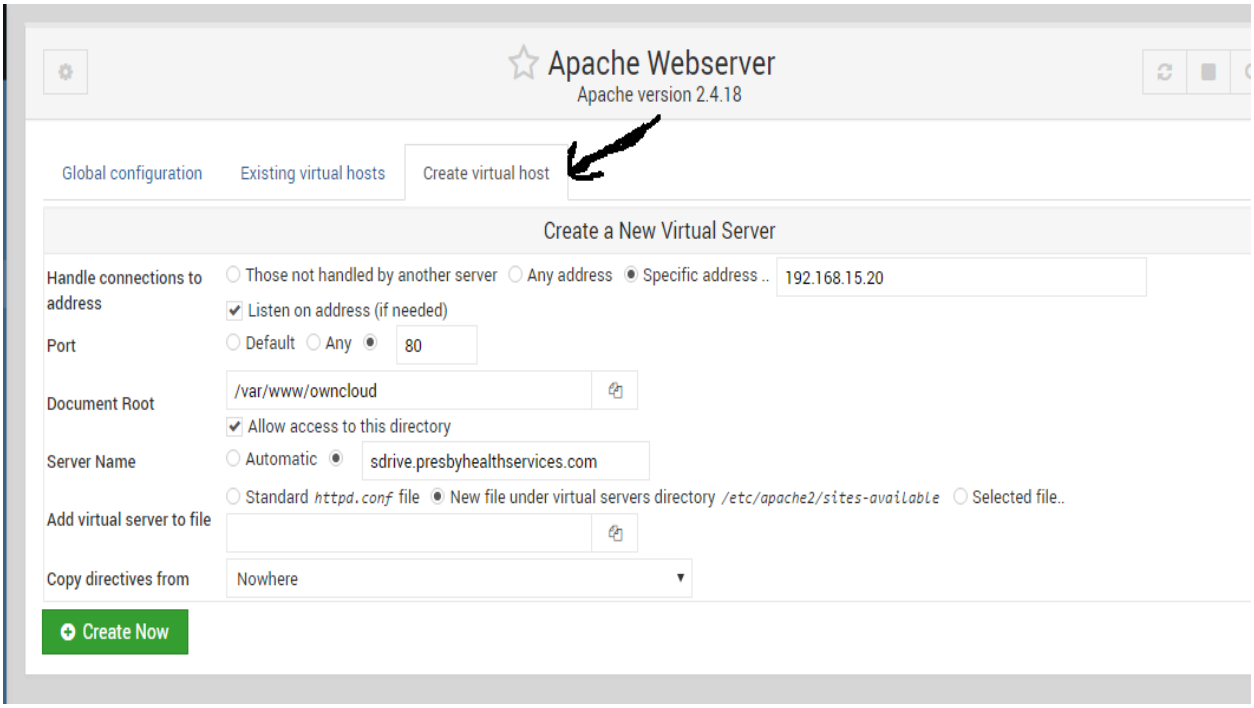
Create the folder where the site files need to be stored. You can create a folder clicking **Others** >**File Manager** section in webmin. Example **/var/www/owncloud**



SETTING UP VIRTUAL HOST

Click on **Servers** => **Apache webserver**

Click on '**create virtual host'** tab



Enter the following fields:

Handle connections to: select: Specific Address and enter the server's IP

Port: Enter port 80

Document Root: This is where site files are stored. Use for eg: /var/www/owncloud

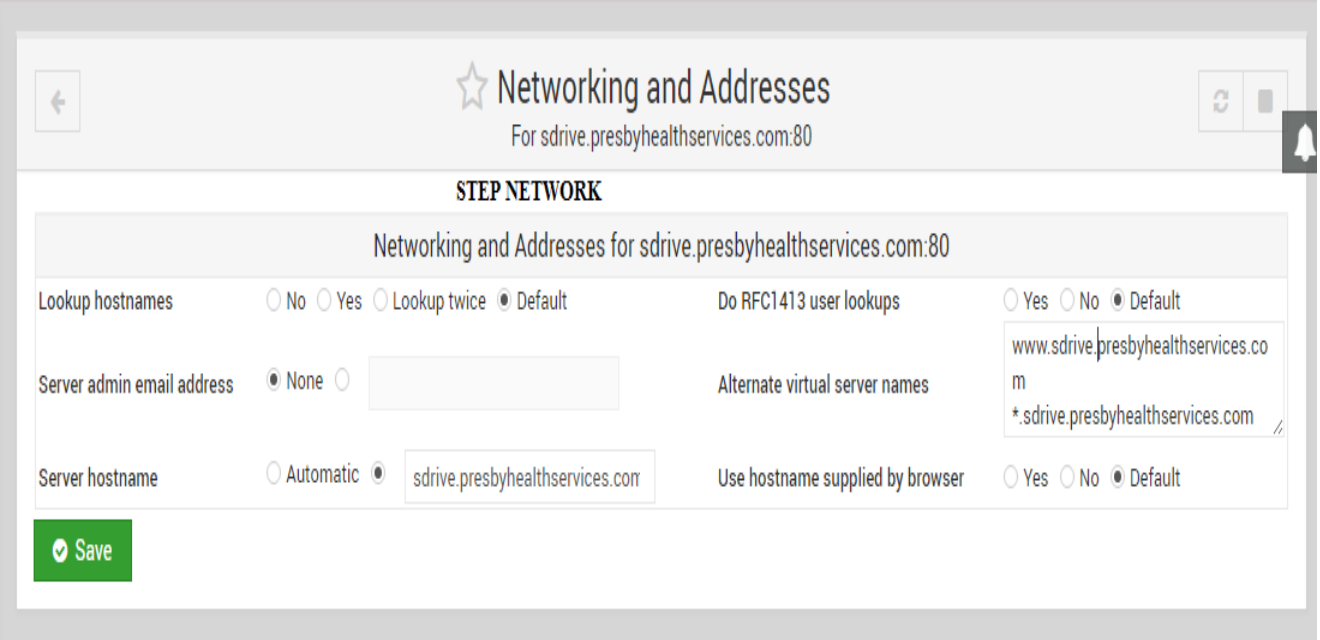
Server Name: your domain. Example. sdrive.snetgh.net

Add virtual server to file: (leave default as Standard httpd.conf file)

Copy directives from: (leave default: Nowhere)

Click create now.

Add a Server alias for domain www.sdrive.snetgh.net



Click on Servers => **Apache Web Server** => **Existing Virtual Hosts**

Click on the virtual server you just created and Click on "**Networking and Addresses**"

Add any other server aliases to use in '**Alternate Virtual Server Names'** box

Type in the space: **www.sdrive.snetgh.net** and ***. sdrive.snetgh.net** and Click Save.

Click on '**Apply changes'**. This will save the configuration and restart Apache.

HOW TO INSTALL LET'S ENCRYPT WITH APACHE

Let's Encrypt is an open SSL Certificate Authority (CA) that offers free domain-validated (DV) certificates for your websites. SSL Certificates are used to establish a secure encrypted connection between a web server and a user's web browser. The SSL certificates that have been issued by Let's Encrypt are valid for 90 days and are trusted by most web browsers today.

You need a registered domain name with its A record pointing to your server's IP address.

Install CertBot

root@snet[~]# sudo add-apt-repository ppa:certbot/certbot

root@snet[~]# sudo apt-get update

root@snet[~]# sudo apt-get install python-certbot-apache

Install Let's Encrypt SSL

root@snet[~]# sudo certbot --apache -d sdrive.snetgh.net -d www.sdrive.snetgh.net

Renew the SSL certificate with a cron job

root@snet[~]# crontab -e

root@snet[~]# 0 0 1 * * /usr/bin/letsencrypt renew >> /var/log/letsencrypt-renew.log

root@snet[~]# service cron restart

INSTALLATION OF OWNCLOUD SERVER

root@snet[~]# cd /var/www

root@snet[~]# wget https://download.owncloud.org/community/owncloud-10.0.10.zip

root@snet[~]# unzip owncloud-10.0.10.zip

root@snet[~]# sudo chown -R www-data:www-data /var/www/owncloud

HOW TO ADD NEW DRIVE FOR OWNCLOUD DATA STORAGE

root@snet[~]# sudo fdisk -l

root@snet[~]# sudo fdisk /dev/sdb

Press m

Press n

Press p

Press 1

Choose default

Choose default

Press p

Press w

root@snet[~]# df -h

root@snet[~]# mkfs.ext4 -b 4096 /dev/sdb

root@snet[~]# sudo nano /etc/fstab

Add below to fstab

/dev/sdb /sdrive ext4 defaults 0 0

root@snet[~]# cat /etc/fstab

root@snet[~]# mkdir -p /sdrive

root@snet[~]# mount /dev/sdb /sdrive

root@snet[~]# df -h

Grant www-data permissions to the sdrive so that owncloud can read the files

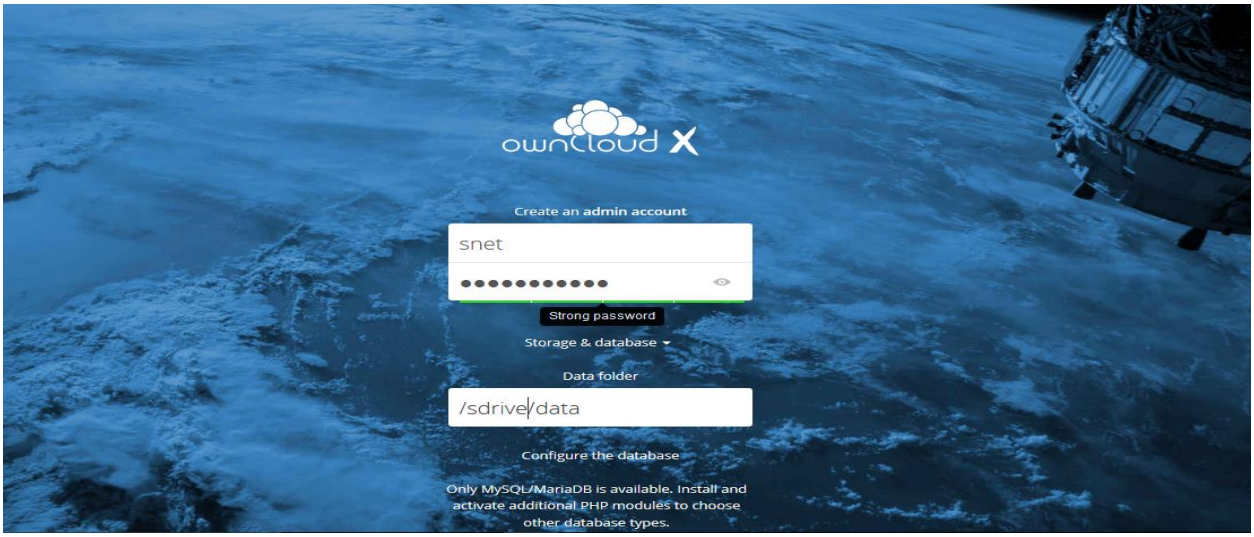
root@snet[~]# sudo chown -R www-data:www-data /sdrive

FINAL CONFIGURATION

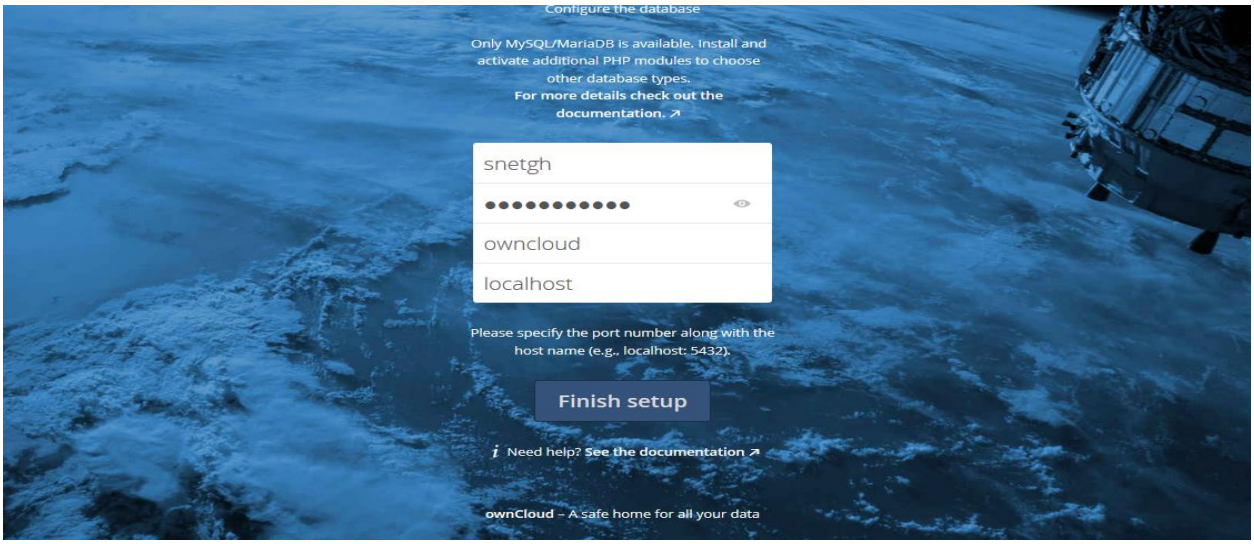
Now, if you go to your domain name **sdrive.snetgh.net** in your browser, you will see a page that looks like this:

Create an admin account by choosing a username and a password. For security purposes it is not recommended to use something like "admin" for the username.

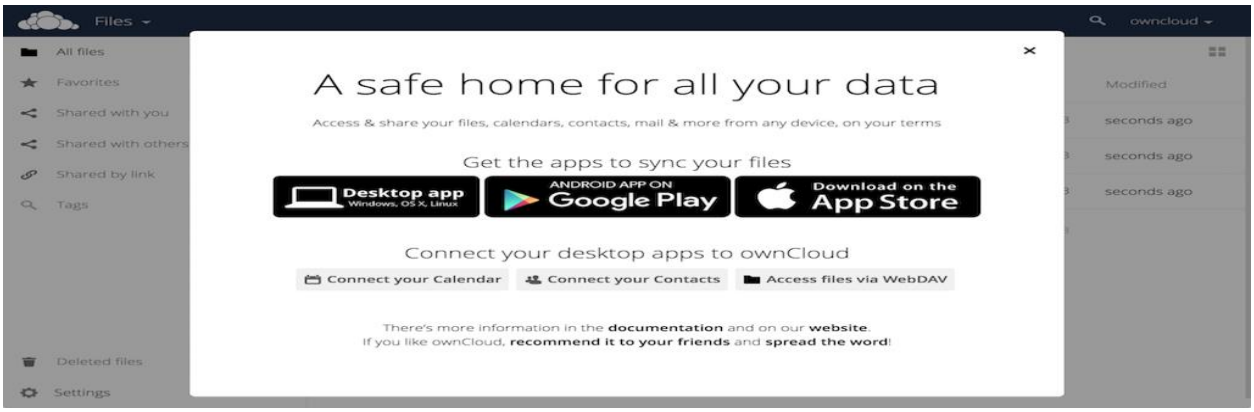
Before clicking the Finish setup button, change the data Storage to the external drive configured above.



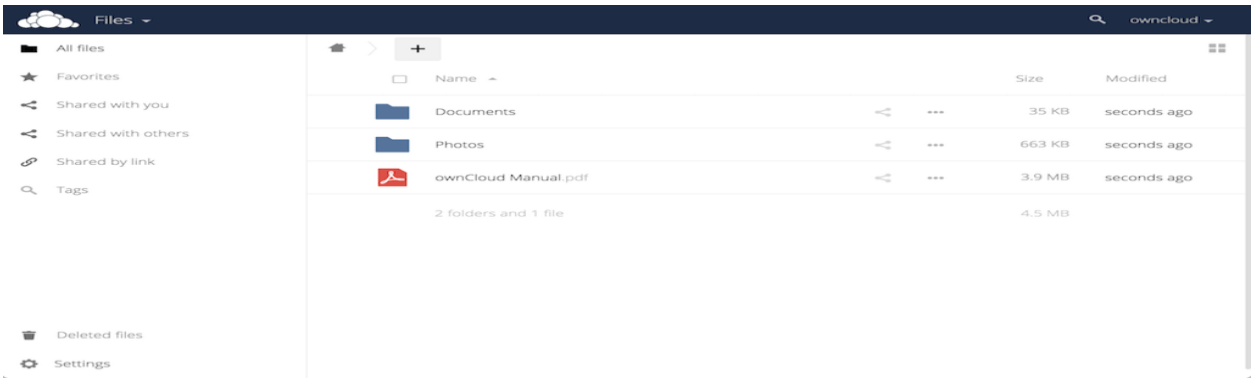
Enter the database information that you configured in the previous step. Below is an example, which matches the database credentials that we used in this guide:



Click the Finish setup button to sign into OwnCloud. A safe home for all your data splash screen should appear:



Click the x in the top-right corner of the splash screen to access the main interface:



Here, you can create or upload files to your personal cloud.

MEMORY CACHE SETUP IN OWNCLOUD

root@snet[~]# nano /var/www/owncloud/config/config.php

Type the following line of text to the config.php at the top

```
'memcache.local' => '\OC\Memcache\Redis',
'memcache.locking' => '\\OC\\Memcache\\Redis',
'redis' => array(
'host' => 'localhost',
'port' => 6379,
),
```

HOW TO DISABLE CODE INTEGRITY CHECK AND HELP IN OWNCLOUD

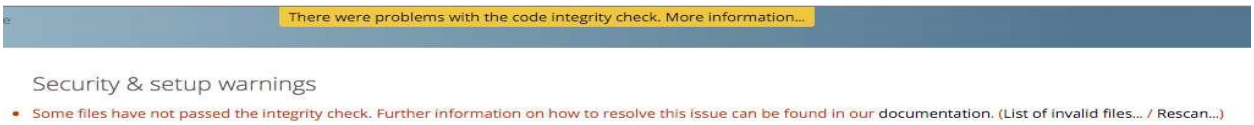
root@snet[~]# nano /var/www/html/owncloud/config/config.php

Go to the end of file and add the following line before the line with**);** and insert

'integrity.check.disabled' => true,

'knowledgebaseenabled' => false,

Now you can login as an admin into your OwnCloud and if you still see the message "There are problems with the code integrity check. More information ..." go to the Admin Page.
There you can click the link Rescan.



HOW TO CONFIGURE STRICT TRANSPORT SECURITY (HSTS)

STRICT TRANSPORT SECURITY

root@snet[~]# nano /etc/apache2/sites-available/owncloud-ssl.conf

Add the following snippet of code to the SSL.conf file

Header always add Strict-Transport-Security "max-age=15768000; includeSubDomains; preload"

root@snet[~]# a2enmod headers

root@snet[~]# sudo service apache2 restart

DEPLOYING PYTHON-DJANGO APPLICATION

root@snet[~]# sudo apt-get install python3-pip

root@snet[~]# sudo apt-get install python3-venv

(venv) root@snet[~]# sudo apt-get install python3-dev libmysqlclient-dev

CREATE DJANGO VIRTUAL ENVIRONMENT

root@snet[~]# python3 -m venv /var/www/member/venv

ACTIVATING OF DJANGO VIRTUAL ENVIRONMENT

Make sure you are in member folder

root@snet[~]# cd /var/www/member

root@snet[~]# source venv/bin/activate

Install django using below command if there is no requirements.txt is not created

(venv) root@snet[~]# pip install django

If requirements.txt is created then run

(venv) root@snet[~]# pip install -r requirements.txt

(venv) root@snet[~]# pip install mysqlclient

(venv) root@snet[~]# pip install django mysqlclient

(venv) root@snet[~]# sudo apt-get install libapache2-mod-wsgi-py3

ADDING HOST NAME/ IP IN DJANGO PROJECT

(venv) root@snet[~]# sudo nano setting.py

Look for Allowed_Host=[‘hostname’,’IP’] and the Hostname/IP

Look for STATIC_URL and add below on top of it.

STATIC_ROOT = ‘project_name/staticfolder

Save ctrl+o enter and ctrl+x to exit

RUN STATIC ON THE SERVER

(venv) root@snet[~]# python manage.py collectstatic

RUN PYTHON DEVELOPMENT SERVER FOR PUBLIC ACCESS

(venv) root@snet[~]# python manage.py runserver 0.0.0.0:8000

CONNECT MYSQL AND DJANGO

Create database using console command line or phpmyadmin shown above

In member folder create a file call mysql.cnf and add the following

```
 [client]
database = DB_NAME
host = localhost  # Or an IP Address that your DB is hosted on
user = DB_USER
password = DB_PASSWORD
default-character-set = utf8
```

RESTART MYSQL

(venv) root@snet[~]# sudo systemctl restart mysql

SETTING DJANGO UP TO USE MYSQL

(venv) root@snet[~]# sudo nano settings.py

Look for Database and comment the default database connection

Add the following to settings.py

```
DATABASES = {
   'default': {
      'ENGINE': 'django.db.backends.mysql',
      'NAME': 'DB_NAME',
      'USER': 'DB_USER',
      'PASSWORD': 'DB_PASSWORD',
      'HOST': 'localhost',   # Or an IP Address that your DB is hosted on
      'PORT': '3306',
   }
}
```

CHECK DJANGO, CREATE SUPERUSER, MAKE MIGRATIONS, RUNSERVER

(venv) root@snet[~]# python manage.py check

(venv) root@snet[~]# python manage.py migrate

(venv) root@snet[~]# python manage.py makemigrations

(venv) root@snet[~]# python manage.py migrate

(venv) root@snet[~]# python manage.py createsuperuser

(venv) root@snet[~]# python manage.py runserver 0.0.0.0:8000

USE DJANGO WITH APACHE AND MOD_WSGI

Create a vhost in apache2 and edit using webmin as shown above

```
<VirtualHost 192.168.15.18:80>
DocumentRoot /var/www/member
ServerName rasp.presbyhealthservices.com
<Directory "/var/www/member">
Allowoverride All
</Directory>
ServerAlias www.rasp.presbyhealthservices.com .*rasp.presbyhealthservices.com

 Alias /static /var/www/member/static
 <Directory /var/www/member/static>
   Require all granted
 </Directory>

 #Alias /media /var/www/django/media
 #<Directory /var/www/django/media>
 #  Require all granted
 #</Directory>

 <Directory /var/www/member/member>
   <Files wsgi.py>
     Require all granted
   </Files>
```

</Directory>

WSGIScriptAlias / /var/www/member/member/wsgi.py

WSGIDaemonProcess member python-path=/var/www/member python-home=/var/www/member/venv

WSGIProcessGroup member

</VirtualHost>

Restart Apache2 for changes to take effect

(venv) root@snet[~]# sudo service apache2 restart

BY STEPHEN FOSU

21ST FEBRUARY, 2019