Penetration Testing Lab Setup:Cloud Computing

November 19, 2019   By Raj Chandel

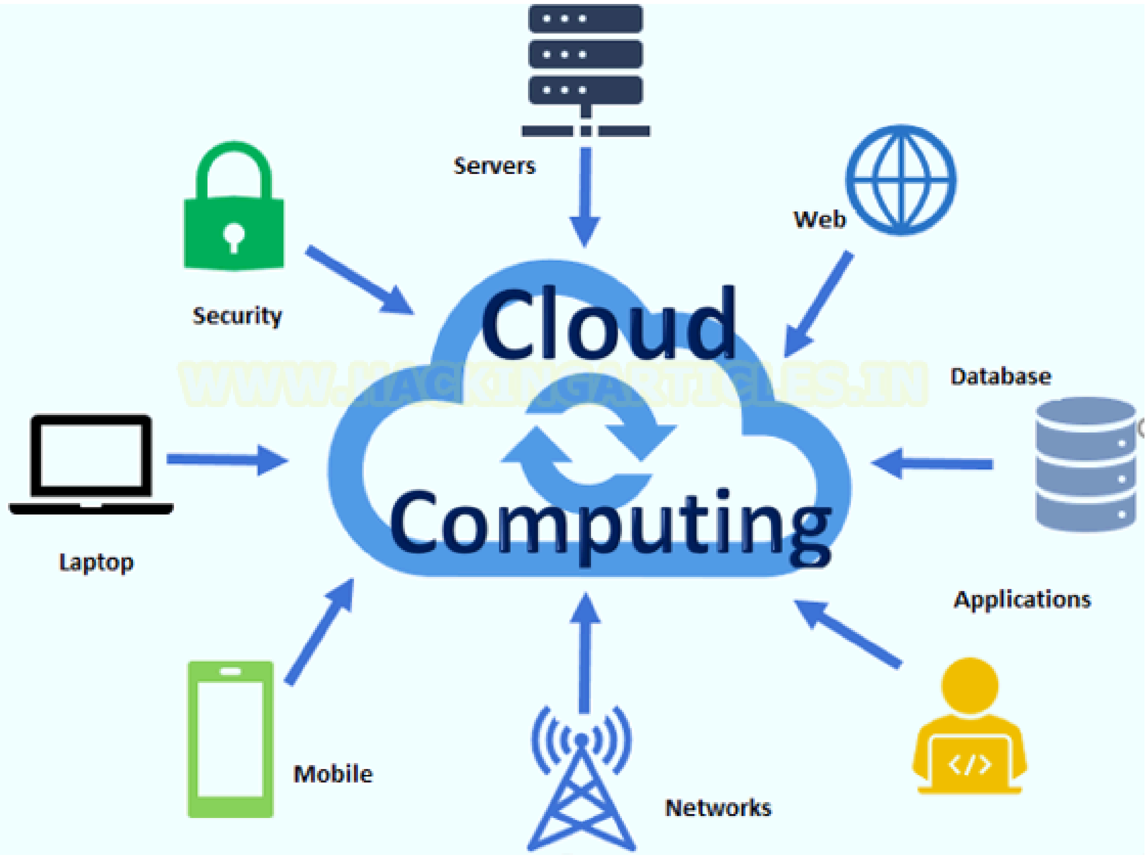
This article is all about setting up a Private Cloud on your local machine on ubuntu, docker and VM. But before it is installed and configured, you should know what the cloud is and why it is a very important part of IT organizations.

Table of Content

Cloud Computing

Cloud computing is the on-demand delivery of compute power, database, storage, applications, and other IT resources via the internet with pay-as-you-go pricing. Whether you are using it to run applications that share photos to millions of mobile users or to support business-critical operations, a cloud services platform provides rapid access to flexible and low-cost IT resources.

In other words, cloud computing means, storing and accessing information and programs over the internet instead of the hard drive of your computer. You can access as many resources as you need, almost instantly, and only pay for what you use.



References: https://aws.amazon.com/what-is-cloud-computing/

Benefits of Cloud Computing

- **Cost Saving** – Pay for what you use.
- **Agile deployment** – Easy and fast access a broad range technology (database, storage, compute etc.) on as per the requirement.
- **Location Independent** –Deploy your application in multiple physical locations around the world with just a click.
- **Disaster Recovery** – No environmental disruption, no natural calamity effect.
- **Elasticity**– Instantly scale up or down the amount of resources that actually need.

Types of Cloud Computing

There are three main types of models of cloud computing. Each type of cloud service and deployment method provides you with different levels of control, flexibility, and management.

- **Infrastructure as a Service (IaaS)** -It is a cloud computing offering in which a vendor provides users access to computing resources such as servers, storage and networking. Organizations use their own platforms and applications within a service provider's infrastructure.

Example: *Linode, Rackspace, Amazon Web Services (AWS), Cisco Metapod, Microsoft Azure, Google Compute Engine (GCE)*

- **Platform as a service (PaaS)**– It is a cloud computing offering that provides users with a cloud environment in which they can develop manage and deliver applications. In addition to storage and other computing resources, users are able to use a suite of prebuilt tools to develop, customize and test their own application also can providers manage security, operating systems, server software and backups.

Example: *AWS Elastic Beanstalk, Windows Azure, Force.com, Google App Engine, Apache Stratos.*

- **Software as a service (SaaS)**-It is a cloud computing offering that provides users with access to a vendor's cloud-based software. Users do not install applications on their local devices. Instead, the applications reside on a remote cloud network accessed through the web or an API. Through the application, users can store and analyse data and collaborate on projects.

Example: *Google Apps, Dropbox, Salesforce, Cisco WebEx,*

Cloud Computing Deployment Models

- **Cloud (Public) –** A cloud-based application is fully deployed in the cloud and all parts of the application run in the cloud. Applications in the cloud have either been created in the cloud or have been migrated from an existing infrastructure to take advantage of the benefits of cloud computing.
- **Hybrid-** A hybrid deployment is a way to connect infrastructure and applications between cloud-based resources and existing resources that are not located in the cloud. The most common method of hybrid deployment is between the cloud and existing on-premises infrastructure to extend, and grow, an organization's infrastructure into the cloud while connecting cloud resources to the internal system.
- **On-premises (Private)** – Private Cloud refers to the cloud solution dedicated for use by a single organization. The data centre resources may be located on-premise or operated by a third-party vendor off-site. The computing resources are isolated and delivered via a secure private network, and not shared with other customers.

How cloud computing works

Cloud computing gives you access to servers, storage, databases, and a broad set of application services over the Internet. A cloud services provider owns and maintains the network-connected hardware required for these application services, while you provision and use what you need via a web application.

Hope, now you have a basic understanding of cloud computing. Let's start the installation of Owncloud in multiple ways.

Installation of Own cloud in Ubuntu

OwnCloud is the market-leading open-source software for cloud-based collaboration platforms. As an alternative to Dropbox, OneDrive and Google Drive, ownCloud offers real data security and privacy for you and your data. Store your files in one central location – protected from unauthorized access. Many features designed for absolute data security help you to work productively and securely.

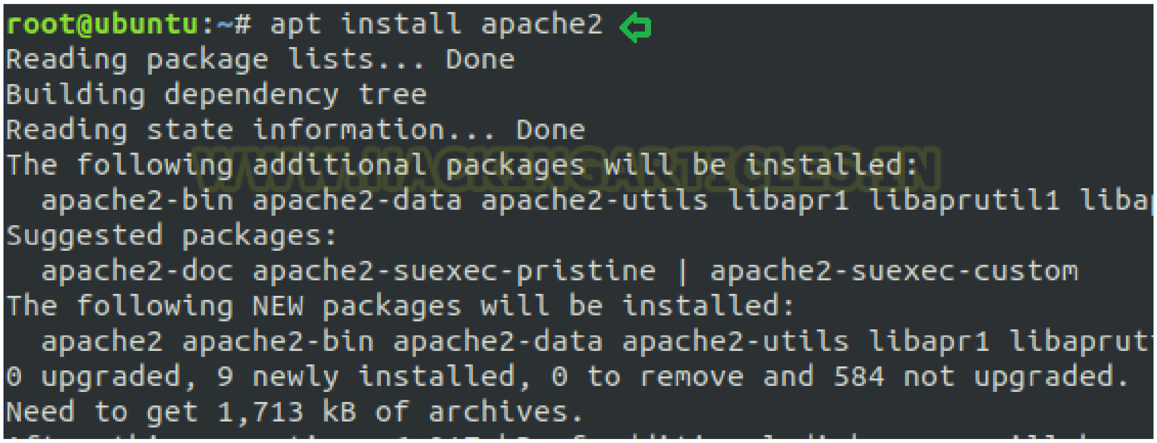
Before starting the installation, I want to confirm that you should already have Ubuntu in PC or you can install ubuntu. As I already have Ubuntu 18.04 LTS.

Let’s start the journey together with below steps:

Install Apache2

OwnCloud requires a webserver to function. So, we install Apache2 on Ubuntu.

```
sudo apt install apache2
```



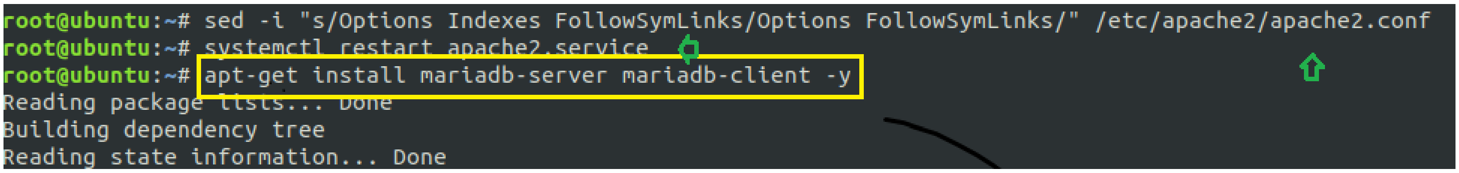
```
root@ubuntu:~# apt install apache2
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils libapr1 libaprutil1 liba
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils libapr1 libaprut
0 upgraded, 9 newly installed, 0 to remove and 584 not upgraded.
Need to get 1,713 kB of archives.
```

Install the MariaDB Server

After apache2 installation, run the commands to disable the directory listing and also to Restart the Apache2 services.

MariaDB is the database server. It is an enhanced, drop-in replacement for MySQL. MariaDB is used because it is fast, scalable and robust, with a rich ecosystem of storage engines, plugins and many other tools make it very versatile for a wide variety of use cases.

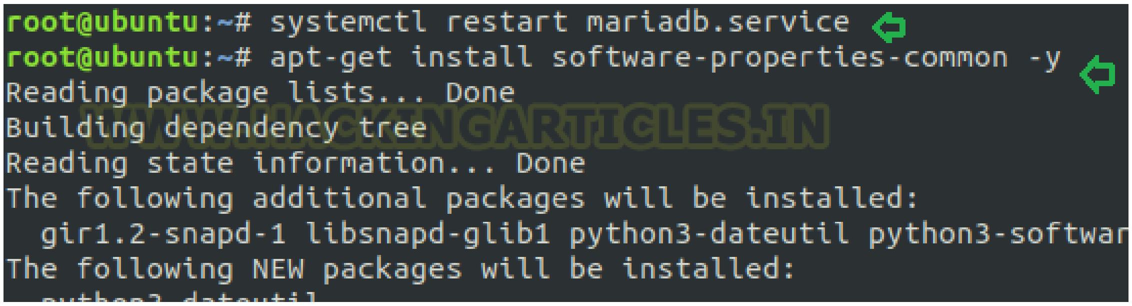
```
sudo sed -i "s/Options Indexes FollowSymLinks/Options FollowSymLinks/" /etc/apache
sudo systemctl restart apache2.service
sudo apt-get install mariadb-server mariadb-client -y
```



```
root@ubuntu:~# sed -i "s/Options Indexes FollowSymLinks/Options FollowSymLinks/" /etc/apache2/apache2.conf
root@ubuntu:~# systemctl restart apache2.service
root@ubuntu:~# apt-get install mariadb-server mariadb-client -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
```

After installation of MariaDB , restart the service and enable MariaDB service to always start up when the server boots.

```
sudo systemctl restart mariadb.service
sudo apt-get install software-properties-common -y
```

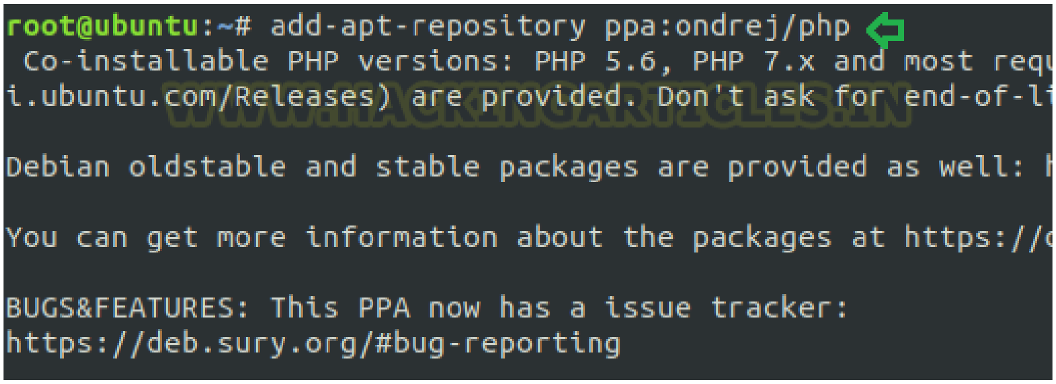


```
root@ubuntu:~# systemctl restart mariadb.service
root@ubuntu:~# apt-get install software-properties-common -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  gir1.2-snapd-1 libsnapd-glib1 python3-dateutil python3-softwar
The following NEW packages will be installed:
  python3-dateutil
```

Install PHP and its modules

Run the commands to add a third-party repository and upgrade to PHP 7.1

```
sudo add-apt-repository ppa:ondrej/php
```



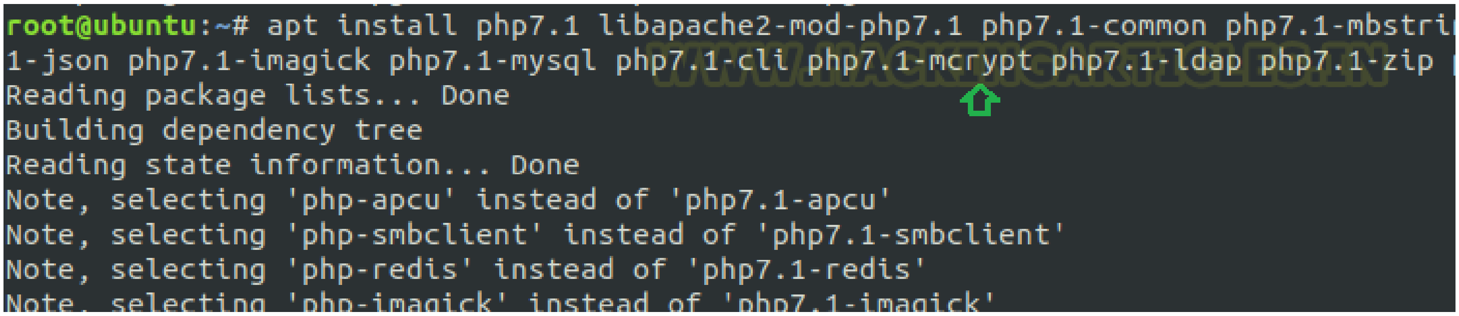
```
root@ubuntu:~# add-apt-repository ppa:ondrej/php
 Co-installable PHP versions: PHP 5.6, PHP 7.x and most requ
i.ubuntu.com/Releases) are provided. Don't ask for end-of-li

Debian oldstable and stable packages are provided as well: h

You can get more information about the packages at https://d

BUGS&FEATURES: This PPA now has a issue tracker:
https://deb.sury.org/#bug-reporting
```

Now, update and upgrade

```
sudo apt update
```

Next, we install PHP7.1 and related modules

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



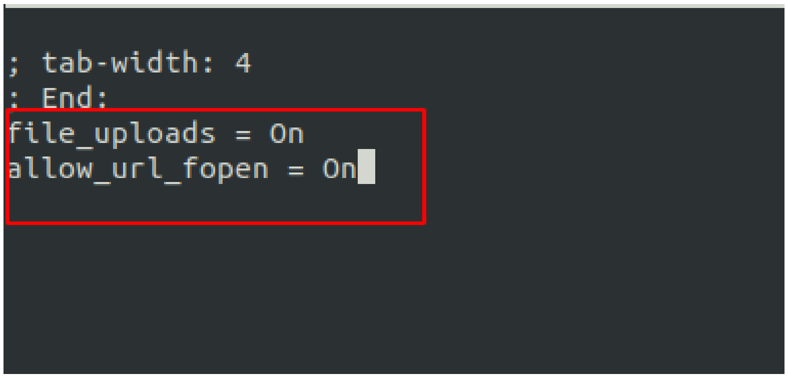
```
root@ubuntu:~# apt install php7.1 libapache2-mod-php7.1 php7.1-common php7.1-mbstri
1-json php7.1-imagick php7.1-mysql php7.1-cli php7.1-mcrypt php7.1-ldap php7.1-zip
Reading package lists... Done
Building dependency tree
Reading state information... Done
Note, selecting 'php-apcu' instead of 'php7.1-apcu'
Note, selecting 'php-smbclient' instead of 'php7.1-smbclient'
Note, selecting 'php-redis' instead of 'php7.1-redis'
Note, selecting 'php-imagick' instead of 'php7.1-imagick'
```

After installation of PHP 7.1, open FPM PHP default file.

```
sudo nano /etc/php/7.1/apache2/php.ini
```

```
root@ubuntu:~# sudo nano /etc/php/7.1/apache2/php.ini
```

```
file_uploads = On
allow_url_fopen = On
```



```
; tab-width: 4
; End:
file_uploads = On
allow_url_fopen = On
```

Create OwnCloud Database

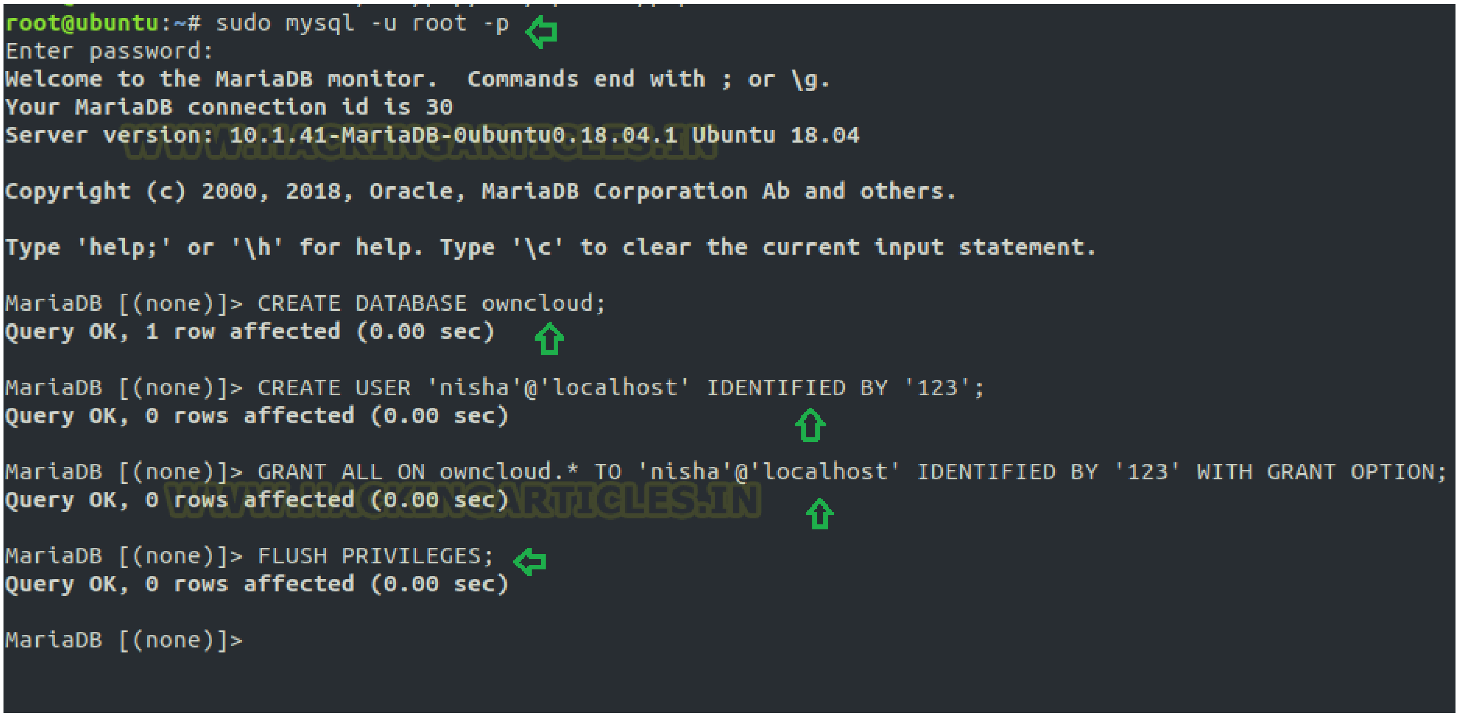
After the installation of all the necessary LAMP packages, we will continue to configure the servers. First, we create the OwnCloud Database. Below are the steps:

1. Run the MySQL command to logon to the database server. In the next prompt, type the root password.
2. Create a database called Owncloud.
3. Create a database user g nisha with the new password e.g 123(you Should put the strong password for security purpose).

4. Then, we grant the user full access to the database.
5. Finally, save your changes and exit.

Note: In the database, the command should be ended by a sign ; otherwise you will get an error.

```
sudo mysql -u root -p
CREATE DATABASE owncloud;
CREATE USER 'nisha'@'localhost' IDENTIFIED BY '123';
GRANT ALL ON owncloud.* TO 'nisha'@'localhost' IDENTIFIED BY '123' WITH GRANT OPTI
FLUSH PRIVILEGES;
```

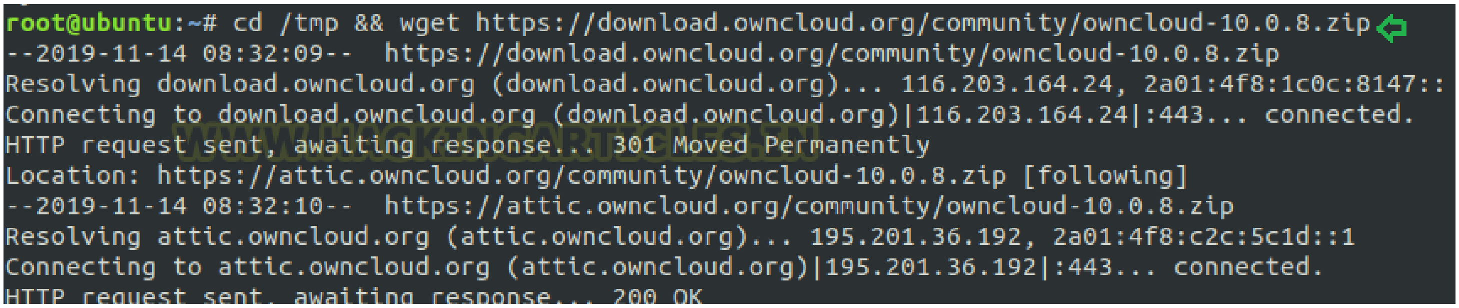


```
root@ubuntu:~# sudo mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 30
Server version: 10.1.41-MariaDB-0ubuntu0.18.04.1 Ubuntu 18.04

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> CREATE DATABASE owncloud;
Query OK, 1 row affected (0.00 sec)

MariaDB [(none)]> CREATE USER 'nisha'@'localhost' IDENTIFIED BY '123';
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> GRANT ALL ON owncloud.* TO 'nisha'@'localhost' IDENTIFIED BY '123' WITH GRANT OPTION;
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]>
```

Download Latest Owncloud Release

Visit **https://owncloud.com/download**/ for download and extract OwnCloud Files into the /var/www/html directory.

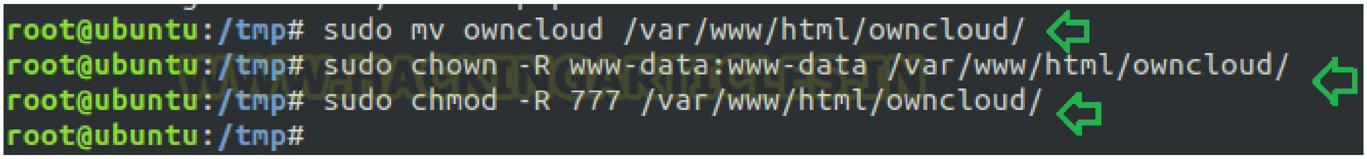
```
cd /tmp && wget https://download.owncloud.org/community/owncloud-10.0.8.zip
unzip owncloud-10.0.8.zip
```



```
root@ubuntu:~# cd /tmp && wget https://download.owncloud.org/community/owncloud-10.0.8.zip
--2019-11-14 08:32:09--  https://download.owncloud.org/community/owncloud-10.0.8.zip
Resolving download.owncloud.org (download.owncloud.org)... 116.203.164.24, 2a01:4f8:1c0c:8147::
Connecting to download.owncloud.org (download.owncloud.org)|116.203.164.24|:443... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: https://attic.owncloud.org/community/owncloud-10.0.8.zip [following]
--2019-11-14 08:32:10--  https://attic.owncloud.org/community/owncloud-10.0.8.zip
Resolving attic.owncloud.org (attic.owncloud.org)... 195.201.36.192, 2a01:4f8:c2c:5c1d::1
Connecting to attic.owncloud.org (attic.owncloud.org)|195.201.36.192|:443... connected.
HTTP request sent, awaiting response... 200 OK
```

```
sudo mv owncloud /var/www/html/owncloud/
sudo chown -R www-data:www-data /var/www/html/owncloud/
sudo chmod -R 777 /var/www/html/owncloud/
```

Then set the correct permissions for OwnCloud to function, change the ownership and mod e.g as we grant (Read Write Execute i.e 777) permission.



```
root@ubuntu:/tmp# sudo mv owncloud /var/www/html/owncloud/
root@ubuntu:/tmp# sudo chown -R www-data:www-data /var/www/html/owncloud/
root@ubuntu:/tmp# sudo chmod -R 777 /var/www/html/owncloud/
root@ubuntu:/tmp#
```

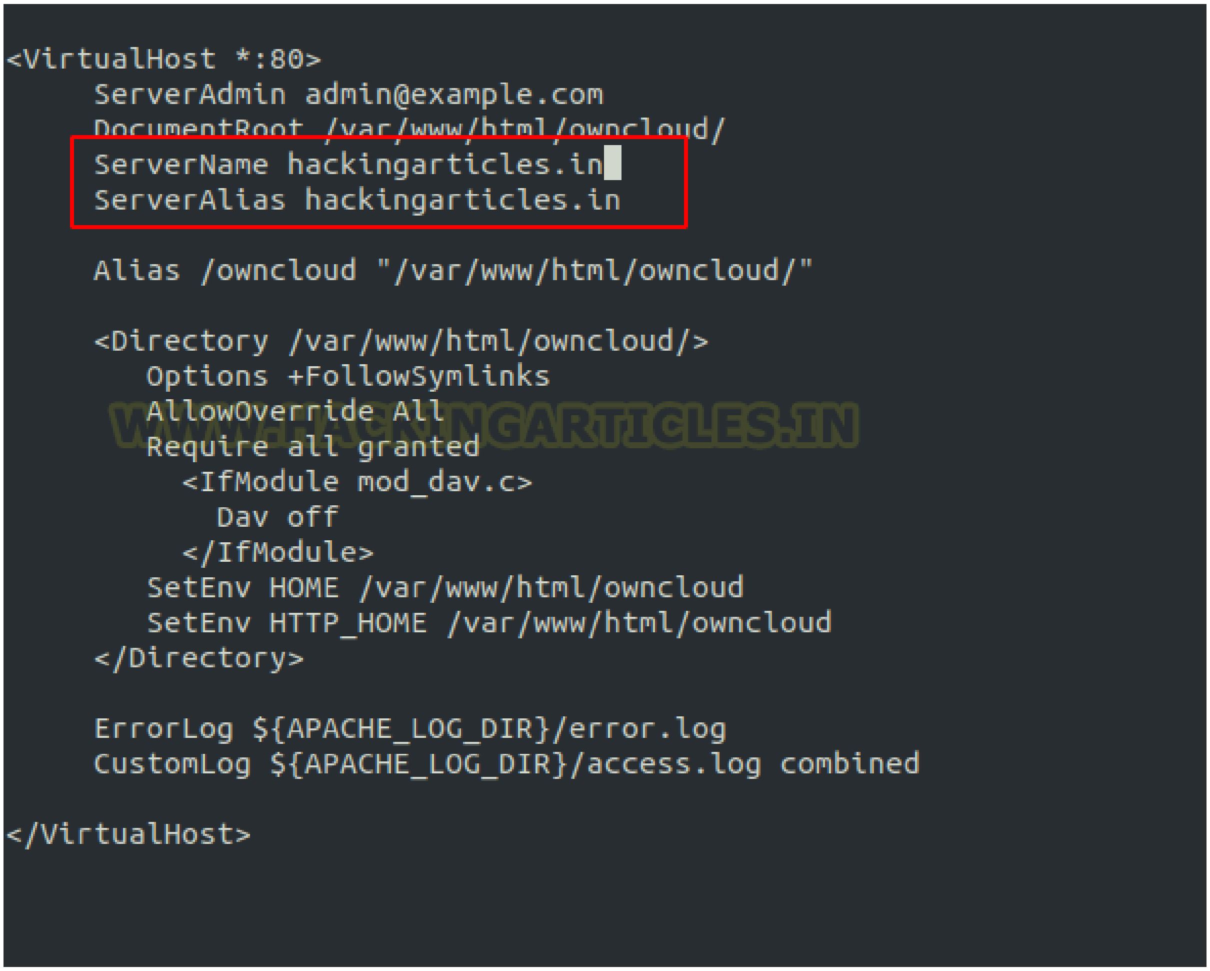
Configure Apache2

Configure Apahce2 site configuration file for OwnCloud. This file will control how users access OwnCloud content. Create a new configuration file called owncloud.conf as shown.

```
sudo nano /etc/apache2/sites-available/owncloud.conf
```

```
<VirtualHost *:80>
     ServerAdmin admin@example.com
     DocumentRoot /var/www/html/owncloud/
     ServerName hackingarticles.in
     ServerAlias hackingarticles.in
     Alias /owncloud "/var/www/html/owncloud/"
     <Directory /var/www/html/owncloud/>
        Options +FollowSymlinks
        AllowOverride All
        Require all granted
          <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>
```

Then copy and paste the content below into the file and save it. Replace the highlighted in yellow lines with your own domain name and directory root location and then save the file.

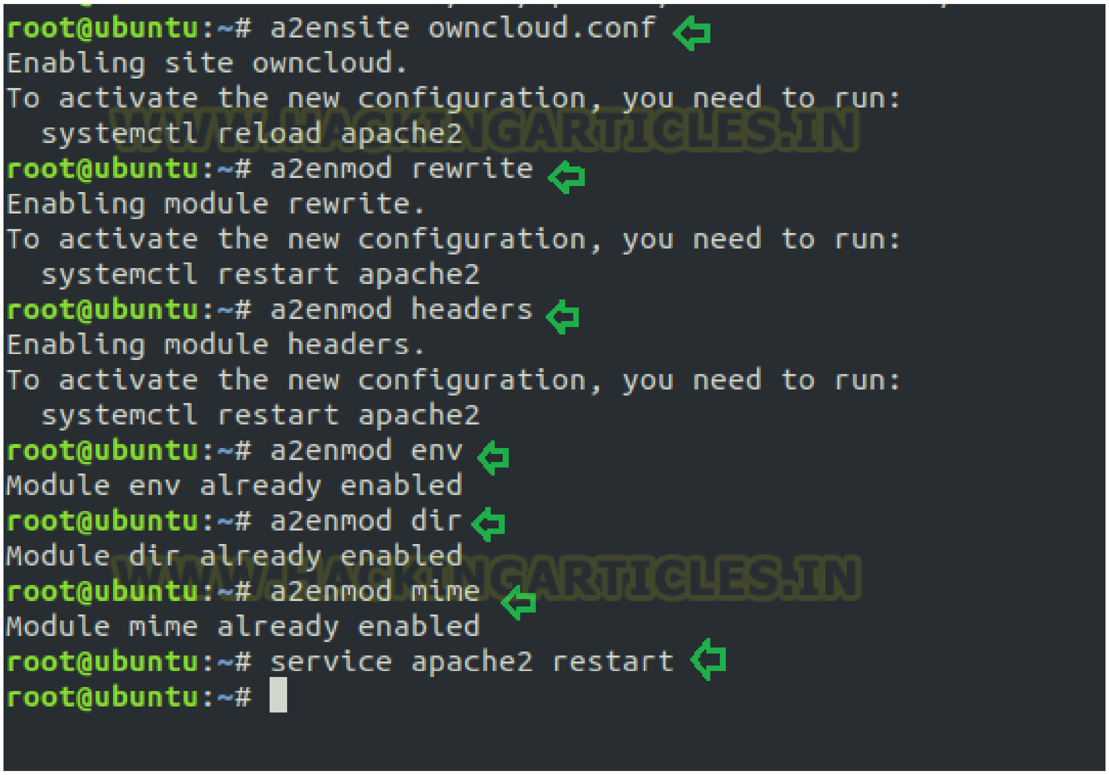


```
<VirtualHost *:80>
     ServerAdmin admin@example.com
     DocumentRoot /var/www/html/owncloud/
     ServerName hackingarticles.in
     ServerAlias hackingarticles.in

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

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

Enable the OwnCloud and Rewrite Module

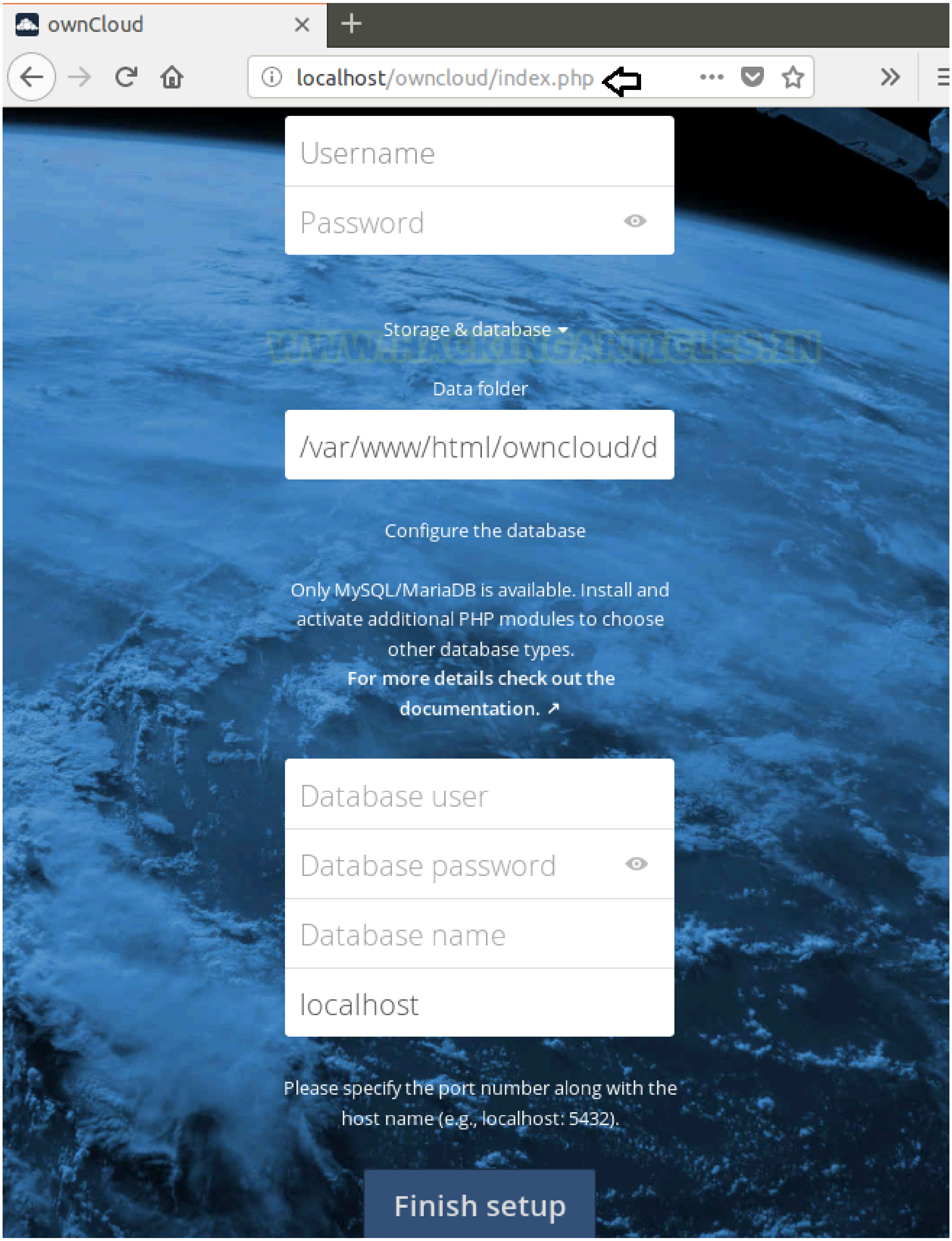
After configuring the VirtualHost above, enable it by running the commands below and at last restart the Apache2 service.

```
sudo a2ensite owncloud.conf
sudo a2enmod rewrite
sudo a2enmod headers
sudo a2enmod env
sudo a2enmod dir
sudo a2enmod mime
sudo service apache2 restart
```



```
root@ubuntu:~# a2ensite owncloud.conf
Enabling site owncloud.
To activate the new configuration, you need to run:
  systemctl reload apache2
root@ubuntu:~# a2enmod rewrite
Enabling module rewrite.
To activate the new configuration, you need to run:
  systemctl restart apache2
root@ubuntu:~# a2enmod headers
Enabling module headers.
To activate the new configuration, you need to run:
  systemctl restart apache2
root@ubuntu:~# a2enmod env
Module env already enabled
root@ubuntu:~# a2enmod dir
Module dir already enabled
root@ubuntu:~# a2enmod mime
Module mime already enabled
root@ubuntu:~# service apache2 restart
root@ubuntu:~#
```

Open the browser and put localhost(local IP) e.g **http://localhost/owncloud**



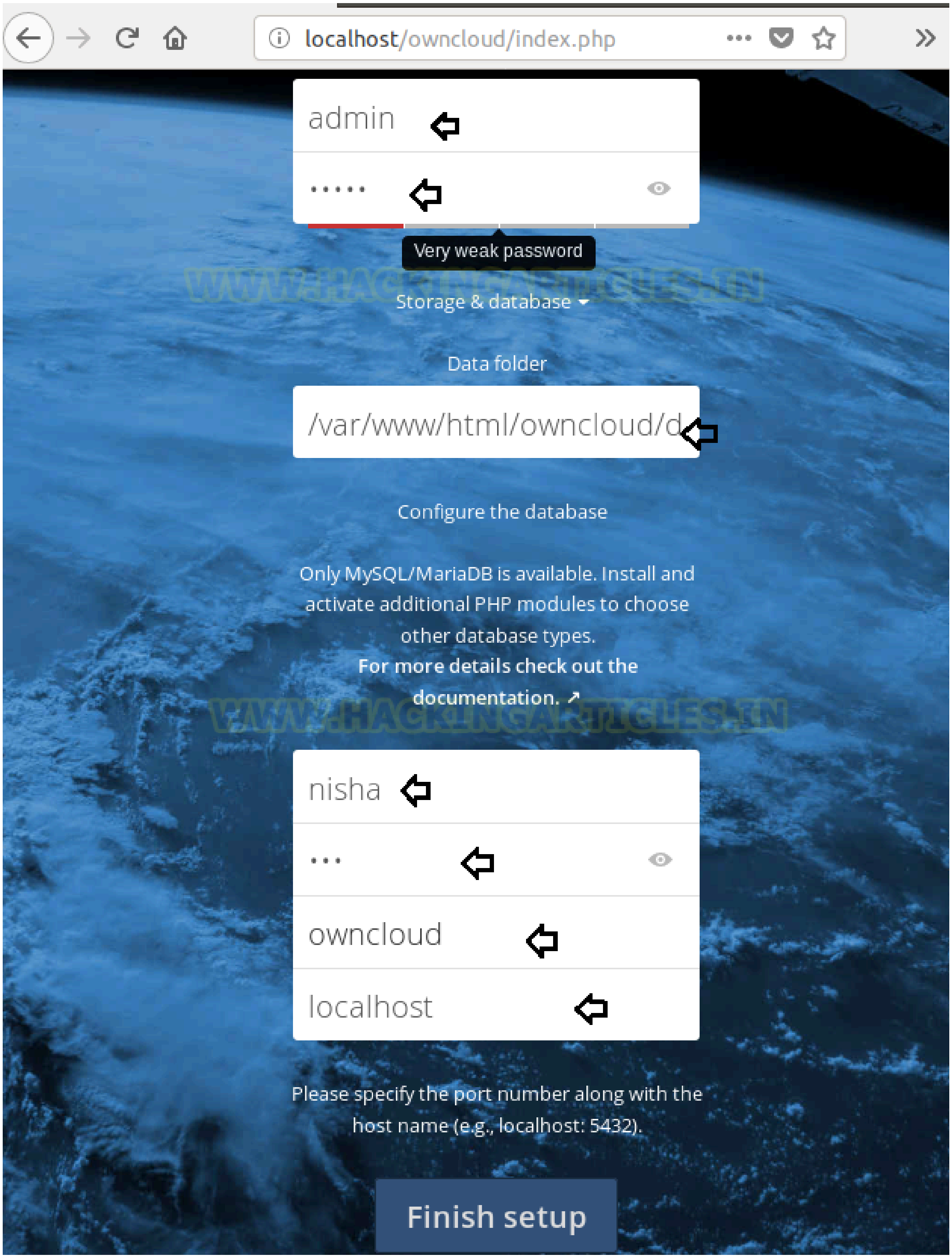
Data folder: /var/www/html/owncloud/d

Only MySQL/MariaDB is available. Install and activate additional PHP modules to choose other database types.
For more details check out the documentation. ↗

localhost

Please specify the port number along with the host name (e.g., localhost: 5432).

You'll be prompted to create an admin account and password. Connect to the database using the information you created and then click on finish setup.



admin

••••••

Very weak password

Storage & database ▾

Data folder

/var/www/html/owncloud/d

Configure the database

Only MySQL/MariaDB is available. Install and activate additional PHP modules to choose other database types.
For more details check out the documentation. ↗

nisha

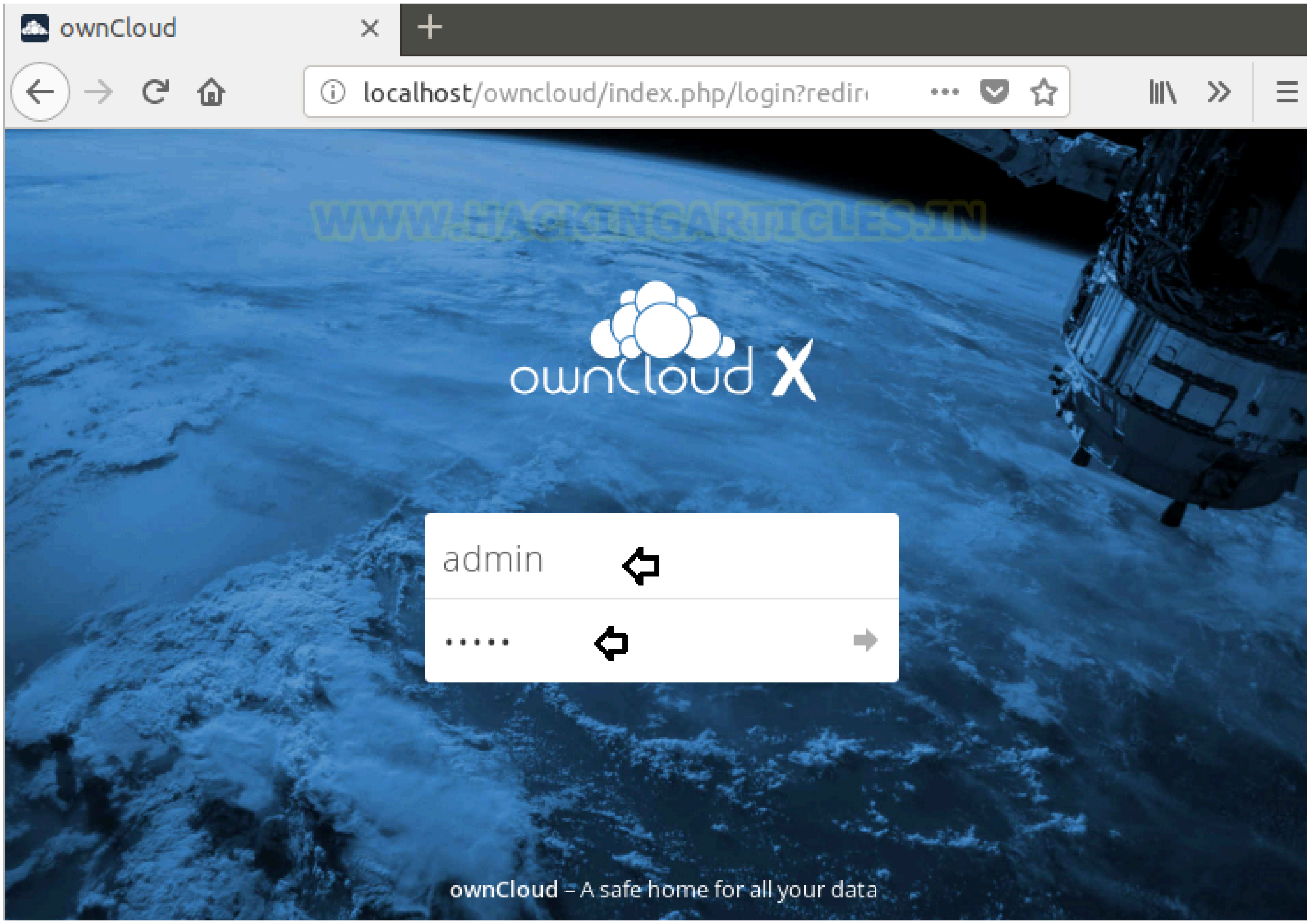
•••

owncloud

localhost

Please specify the port number along with the host name (e.g., localhost: 5432).

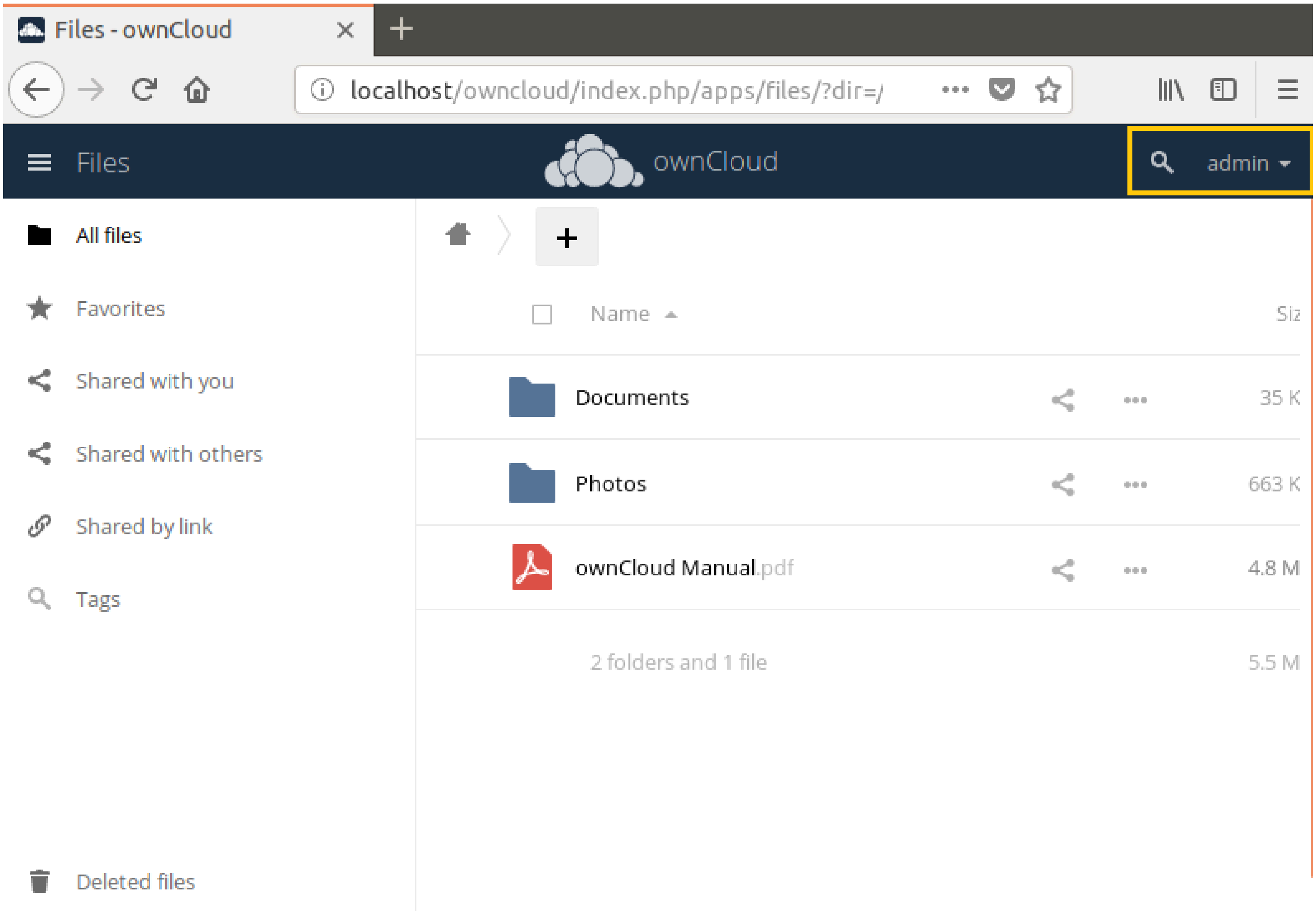
Put the admin Credentials and continue.


ownCloud
localhost/owncloud/index.php/login?redir
WWW.HACKINGARTICLES.IN
ownCloud X
ownCloud – A safe home for all your data

admin

••••••

Happy to see the final Picture of OwnCloud, now you can upload and store your data safely on Owncloud.



Installation of OwnCloud using Docker

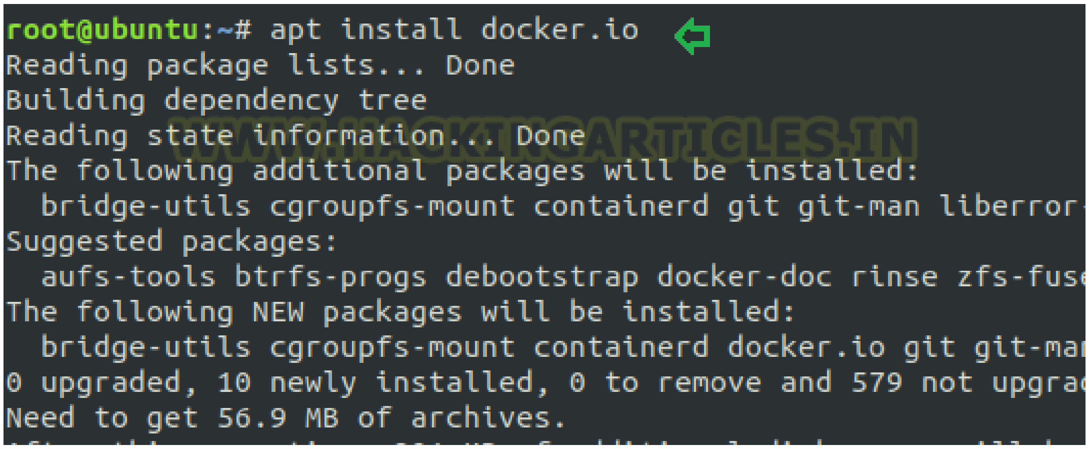
Docker is a tool designed to make it easier to create, deploy, and run applications by using containers. Containers allow a developer to package up an application with all of the parts it needs, such as libraries and other dependencies, and ship it all out as one package.

Let’s start the installation with the below steps :

Install Docker

To install docker, simply open the terminal of Linux and type the following command:

```
apt install docker.io
```



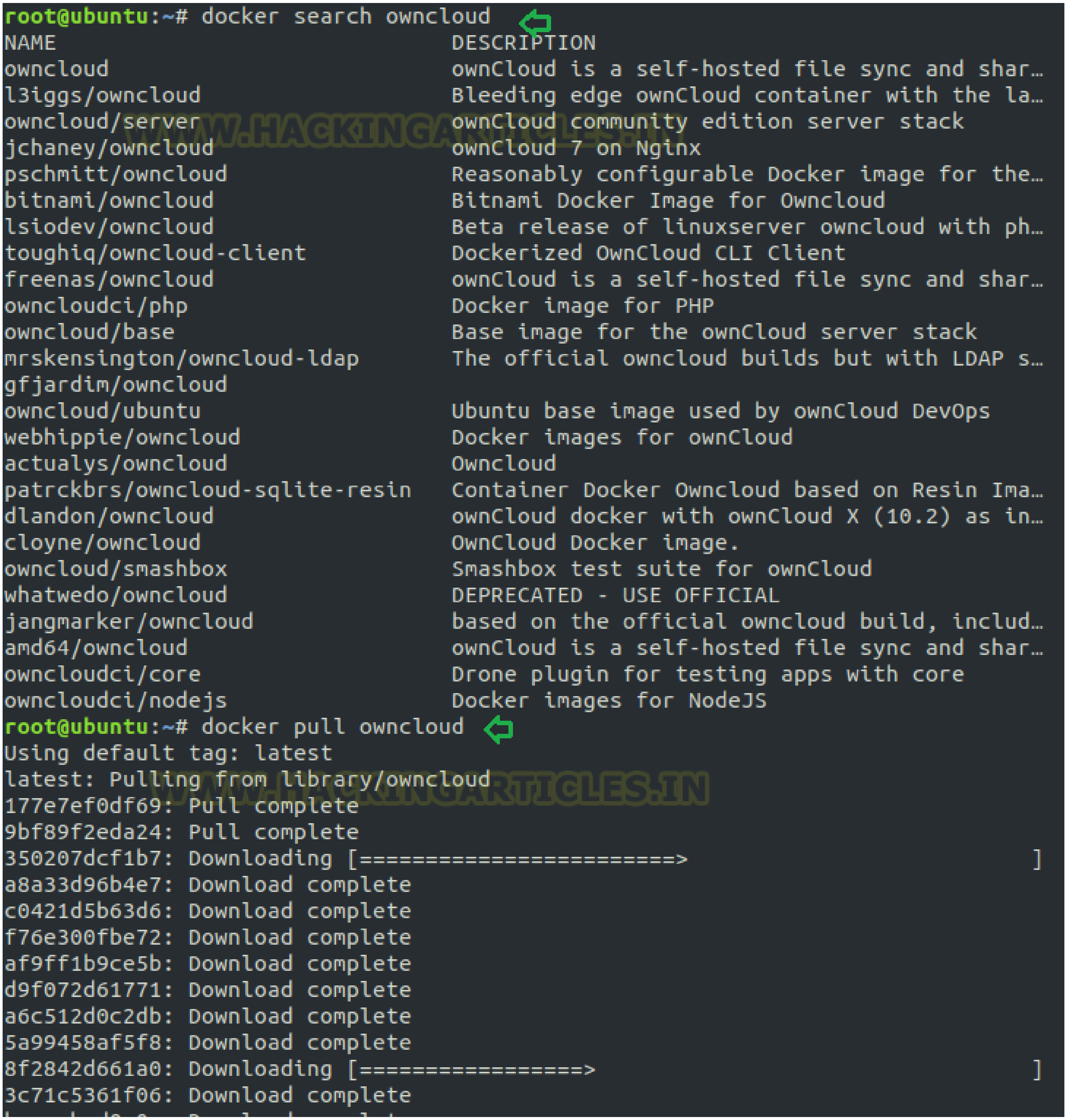
```
root@ubuntu:~# apt install docker.io
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  bridge-utils cgroupfs-mount containerd git git-man liberror-
Suggested packages:
  aufs-tools btrfs-progs debootstrap docker-doc rinse zfs-fuse
The following NEW packages will be installed:
  bridge-utils cgroupfs-mount containerd docker.io git git-man
0 upgraded, 10 newly installed, 0 to remove and 579 not upgrad
Need to get 56.9 MB of archives.
```

Once the docker is up and running, you can run or pull any image in your docker. As per the requirement, we are going to search owncloud image. When you run the following command, it will first check your local repository; if the image is not available there then it will pull it from docker hub.

```
docker search owncloud
```

Once you find your image, you can pull it into your container and download the Owncloud image.

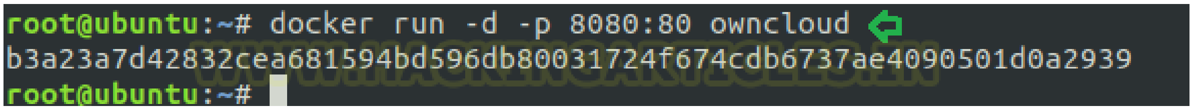
```
docker pull owncloud
```



```
root@ubuntu:~# docker search owncloud
NAME                                DESCRIPTION
owncloud                            ownCloud is a self-hosted file sync and shar…
l3iggs/owncloud                     Bleeding edge ownCloud container with the la…
owncloud/server                     ownCloud community edition server stack
jchaney/owncloud                    ownCloud 7 on Nginx
pschmitt/owncloud                   Reasonably configurable Docker image for the…
bitnami/owncloud                    Bitnami Docker Image for Owncloud
lsiodev/owncloud                    Beta release of linuxserver owncloud with ph…
toughiq/owncloud-client             Dockerized OwnCloud CLI Client
freenas/owncloud                    ownCloud is a self-hosted file sync and shar…
owncloudci/php                      Docker image for PHP
owncloud/base                       Base image for the ownCloud server stack
mrskensington/owncloud-ldap         The official owncloud builds but with LDAP s…
gfjardim/owncloud
owncloud/ubuntu                     Ubuntu base image used by ownCloud DevOps
webhippie/owncloud                  Docker images for ownCloud
actualys/owncloud                   Owncloud
patrckbrs/owncloud-sqlite-resin     Container Docker Owncloud based on Resin Ima…
dlandon/owncloud                    ownCloud docker with ownCloud X (10.2) as in…
cloyne/owncloud                     OwnCloud Docker image.
owncloud/smashbox                   Smashbox test suite for ownCloud
whatwedo/owncloud                   DEPRECATED - USE OFFICIAL
jangmarker/owncloud                 based on the official owncloud build, includ…
amd64/owncloud                      ownCloud is a self-hosted file sync and shar…
owncloudci/core                     Drone plugin for testing apps with core
owncloudci/nodejs                   Docker images for NodeJS
root@ubuntu:~# docker pull owncloud
Using default tag: latest
latest: Pulling from library/owncloud
177e7ef0df69: Pull complete
9bf89f2eda24: Pull complete
350207dcf1b7: Downloading [==========================>                        ]
a8a33d96b4e7: Download complete
c0421d5b63d6: Download complete
f76e300fbe72: Download complete
af9ff1b9ce5b: Download complete
d9f072d61771: Download complete
a6c512d0c2db: Download complete
5a99458af5f8: Download complete
8f2842d661a0: Downloading [===================>                               ]
3c71c5361f06: Download complete
```

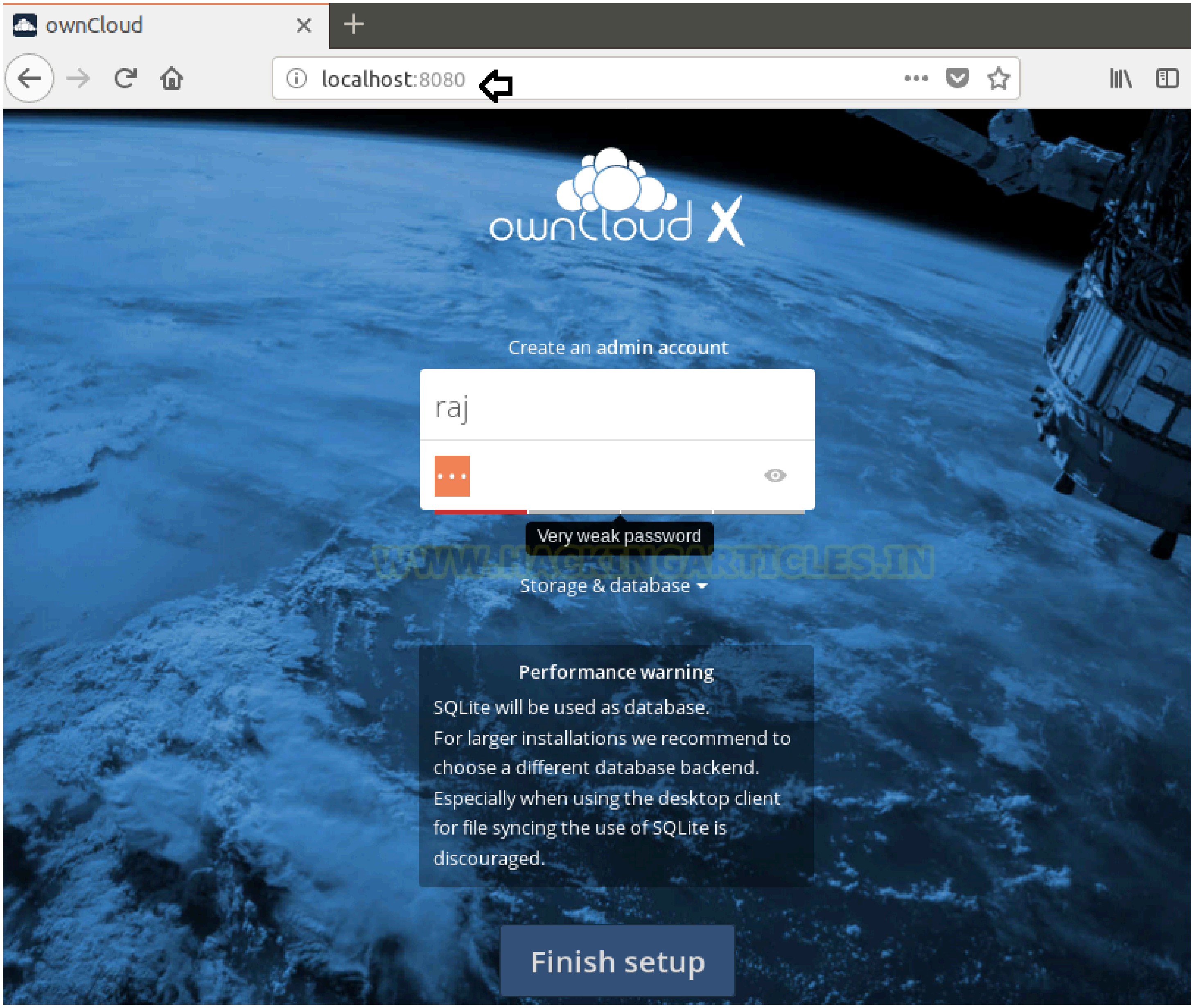
The docker attaches command permits you to attach to a running container using the container ID or name you can use one instance of shell only though attach command or you can directly run the container with container id. ownCloud is accessible via port 8080 on the host machine. But if you crave to open a new terminal with a new instance of container’s shell, we just need run docker exec.

```
docker run -d -p 8080:80 owncloud
```



```
root@ubuntu:~# docker run -d -p 8080:80 owncloud
b3a23a7d42832cea681594bd596db80031724f674cdb6737ae4090501d0a2939
root@ubuntu:~#
```

To log in to the ownCloud UI, open http://localhost:8080 in your browser of choice, where you see the standard ownCloud login screen, as in the image below.



Create an admin account

raj

Performance warning

SQLite will be used as database.
For larger installations we recommend to choose a different database backend.
Especially when using the desktop client for file syncing the use of SQLite is discouraged.

Finally welcome to your owncloud platform to perform your services (upload, safety storage of data etc.)

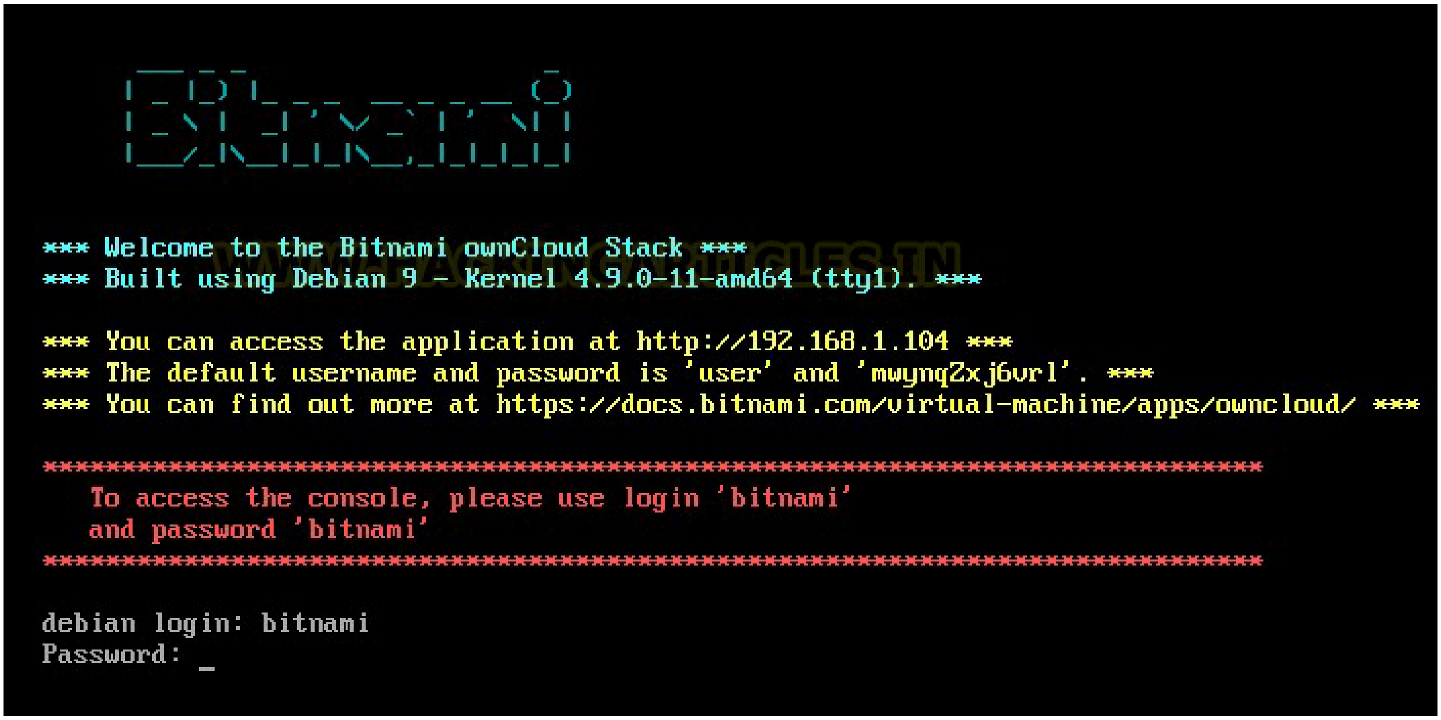
Bitnami Owncloud Stack Virtual Machines

Bitnami Virtual Machines contain a minimal Linux operating system with ownCloud installed and configured. Using the Bitnami Virtual Machine image requires hypervisor software such as VMware Player or VirtualBox. Both of these hypervisors are available free of charge.

You can download from **here**

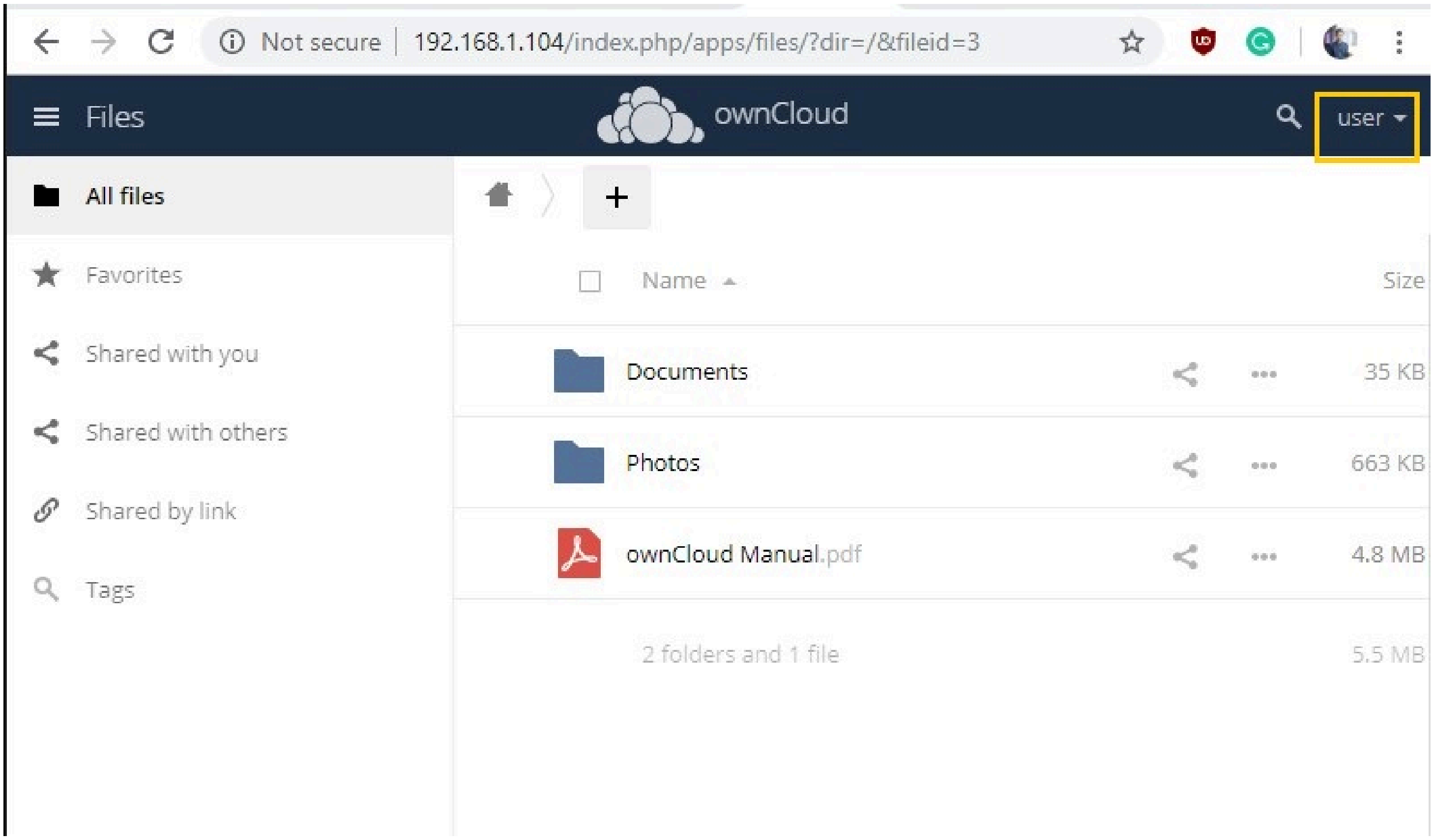
Login with default credential

```
Username: bitnami
Password: bitnami
```



```
 ___ _ _                   _
| _ |_) |_ _ _  __ _ _ __ (_)
| _ \ |  _| ' \/ _` | '  \| |
|___/_|\__|_|_|\__,_|_|_|_|_|

*** Welcome to the Bitnami ownCloud Stack ***
*** Built using Debian 9 - Kernel 4.9.0-11-amd64 (tty1). ***

*** You can access the application at http://192.168.1.104 ***
*** The default username and password is 'user' and 'mwynq2xj6vrl'. ***
*** You can find out more at https://docs.bitnami.com/virtual-machine/apps/owncloud/ ***

******************************************************************************
   To access the console, please use login 'bitnami'
   and password 'bitnami'
******************************************************************************

debian login: bitnami
Password: _
```

It very simple, only just navigate to the web browser and explore VM IP as shown below.



Not secure | 192.168.1.104/index.php/apps/files/?dir=/&fileid=3

Name ▲	Size
Documents	35 KB
Photos	663 KB
ownCloud Manual.pdf	4.8 MB
2 folders and 1 file	5.5 MB