

Aim : Implement Storage as a Service using Own Cloud

Theory :

1. Prepare a detailed study of OwnCloud

ownCloud is a suite of client-server software for creating and using file hosting services. ownCloud functionally has similarities to the widely used Dropbox. The primary functional difference between ownCloud and Dropbox is that ownCloud does not offer data centre capacity to host stored files. The Server Edition of ownCloud is free and open-source, thereby allowing anyone to install and operate it without charge on their own private server.

The architecture models of ownCloud will help us understand it's working.



Figure 1: OwnCloud Solution Architecture

The core of the ownCloud solution is the ownCloud server. Unlike consumer-grade file sharing services, ownCloud's server enables IT to protect and manage files within the ownCloud environment – from file storage to user provisioning and data processing. ownCloud monitors and logs all data access events for downstream auditing and analysis using popular tools like Splunk.

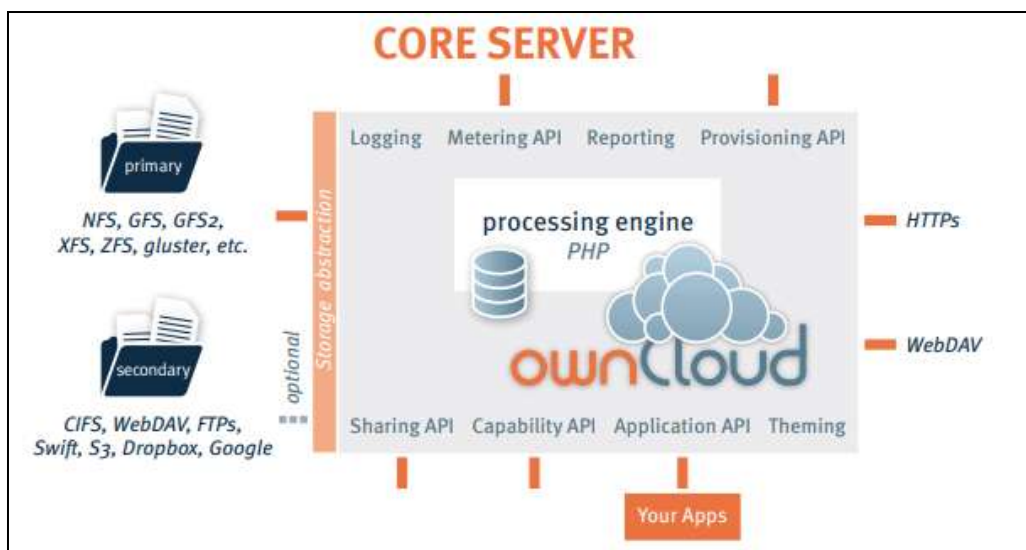


Figure 2: OwnCloud Core Server Architecture

At its core, ownCloud is a PHP web application running on top of IIS or Apache on Windows or Linux. This PHP application manages every other aspect of ownCloud, from user management to plug-ins, file sharing and storage. Attached to the PHP application is a database where ownCloud stores users, user-shared file details, plug-in application states, and the ownCloud file cache (a performance accelerator). ownCloud accesses the database through an abstraction layer, enabling support for Oracle, MySQL, SQL Server, and PostgreSQL. Complete webserver logging is provided via webserver logs, and user and system logs are provided in a separate ownCloud log, or can be directed to a syslog file

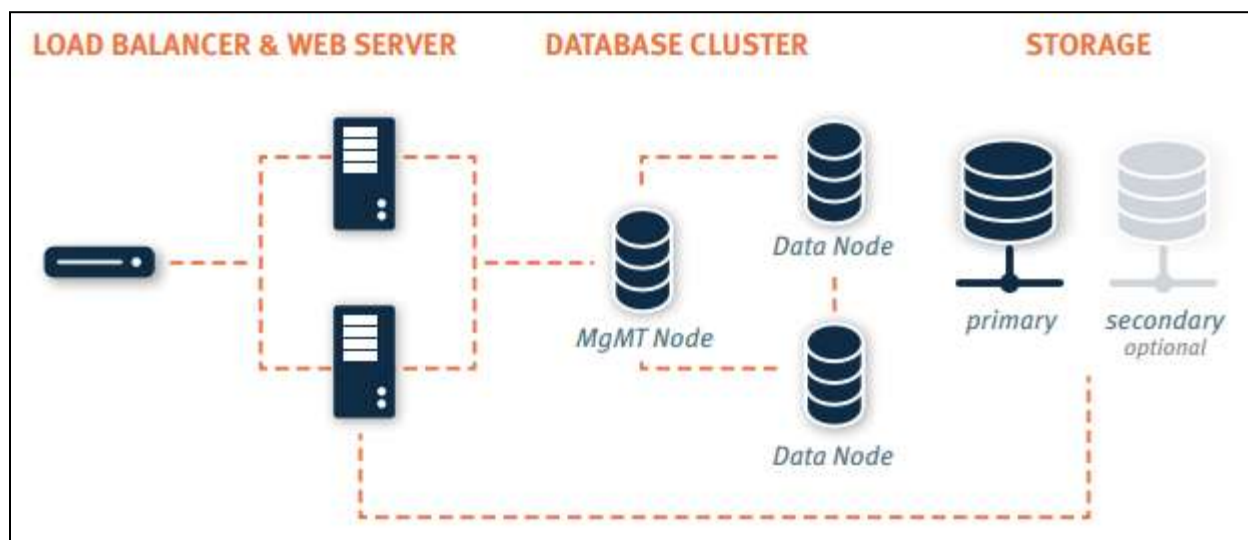


Figure 3: OwnCloud Deployment Architecture

In production, ownCloud is most often deployed as an n-tier load balanced web application running in a data center or managed cloud infrastructure. ownCloud can be deployed to physical, virtual, or private cloud servers using native binaries or a virtual appliance footprint.

Features:

ownCloud includes a variety of open APIs for integrating with other systems. These include:

1. Activity – provides an RSS feed or API call to deliver all activities associated with a user's files, such as sharing activity, updated, renamed, deleted and removed files
2. Applications – the most powerful API, enabling customers to expand ownCloud out of the box, to integrate with existing infrastructure and systems, and to create new plug-in applications. Examples of this API in use include the custom authentication back ends, music and video streaming applications, a bit.ly-inspired app called shorty, and an image preview application
3. Capability – offers information about the installed ownCloud capabilities, so that ownCloud and third party applications can query for the enabled features and plug-in applications.
4. External provisioning – provides the ability to add and remove users remotely, and enables admins to query metering information about ownCloud storage usage and quota.
5. Sharing – provides the ability for external apps, such as the ownCloud mobile app, to share files from remote devices.
6. Themeing – a simplified mechanism for branding the ownCloud server to match your corporate look and feel, enabling colors and logos to be updated with style sheets

2. Advantages and disadvantages of Owncloud

Advantages :

- Desktop clients and mobile apps are available for all major platforms
- Minimal system requirements (hardware and software)
- Open platform, one can federate with others' OwnCloud and share files.
- Scalable and lightweight solution for cloud storing and easy to use

Disadvantages:

- Some components are subject to a commercial license
- Encryption is server-side only, by default
- Occasional performance issues due to the many small files
- No online video viewer by default.
- No online document editor

Activity : Install and configure OwnCloud

Explain each steps of installation and configuration using supporting Screenshots

1. Install Linux in VM (If Required)

Opened Pre-installed Kali Linux



2. Install Own Cloud Server

Installing Pre-requisites

- Lamp Stack installation that is Linux Apache Mysql PHP

First we will update our system and install Apache server

```
kali@kali:~$ sudo apt-get update
Get:1 http://kali.download/kali kali-rolling InRelease [30.5 kB]
Get:2 http://kali.download/kali kali-rolling/main amd64 Packages [17.7 MB]
Get:3 http://kali.download/kali kali-rolling/non-free amd64 Packages [205 kB]
Get:4 http://kali.download/kali kali-rolling/contrib amd64 Packages [109 kB]
Fetched 18.1 MB in 5s (3,291 kB/s)
Reading package lists... Done
```

```
kali@kali:~$ sudo apt-get 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
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom
The following packages will be upgraded:
  apache2 apache2-bin apache2-data apache2-utils libapr1
5 upgraded, 0 newly installed, 0 to remove and 1630 not upgraded.
Need to get 2,141 kB of archives.
After this operation, 124 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://kali.download/kali kali-rolling/main amd64 libapr1 amd64 1.7.0-6 [105 kB]
Get:2 http://kali.download/kali kali-rolling/main amd64 apache2 amd64 2.4.46-4 [261 kB]
Get:3 http://kali.download/kali kali-rolling/main amd64 apache2-bin amd64 2.4.46-4 [1,369 kB]
Get:4 http://kali.download/kali kali-rolling/main amd64 apache2-data all 2.4.46-4 [160 kB]
Get:5 http://kali.download/kali kali-rolling/main amd64 apache2-utils amd64 2.4.46-4 [246 kB]
Fetched 2,141 kB in 1s (1,885 kB/s)
Reading changelogs ... Done
(Reading database ... 255564 files and directories currently installed.)
Preparing to unpack .../libapr1_1.7.0-6_amd64.deb ...
Unpacking libapr1:amd64 (1.7.0-6) over (1.6.5-1+b1) ...
Preparing to unpack .../apache2_2.4.46-4_amd64.deb ...
Unpacking apache2 (2.4.46-4) over (2.4.41-2) ...
Preparing to unpack .../apache2-bin_2.4.46-4_amd64.deb ...
Unpacking apache2-bin (2.4.46-4) over (2.4.41-2) ...
Preparing to unpack .../apache2-data_2.4.46-4_all.deb ...
Unpacking apache2-data (2.4.46-4) over (2.4.41-2) ...
Preparing to unpack .../apache2-utils_2.4.46-4_amd64.deb ...
Unpacking apache2-utils (2.4.46-4) over (2.4.41-2) ...
Setting up libapr1:amd64 (1.7.0-6) ...
Setting up apache2-bin (2.4.46-4) ...
Setting up apache2-data (2.4.46-4) ...
Setting up apache2-utils (2.4.46-4) ...
Setting up apache2 (2.4.46-4) ...
Installing new version of config file /etc/apache2/mods-available/dav.load ...
Installing new version of config file /etc/apache2/mods-available/deflate.conf ...
Installing new version of config file /etc/apache2/mods-available/mime.conf ...
apache2.service is a disabled or a static unit not running, not starting it.
apache-htcacheclean.service is a disabled or a static unit not running, not starting it.
Processing triggers for systemd (244-3) ...
Processing triggers for man-db (2.9.0-2) ...
Processing triggers for kali-menu (2020.1.7) ...
Processing triggers for libc-bin (2.29-9) ...
kali@kali:~$
```

Now we Set Global ServerName to Suppress Syntax Warnings

```
kali@kali:~$ sudo apache2ctl configtest
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.1.1. Set the 'ServerName' directive globally to suppress this message
Syntax OK
kali@kali:~$
```


Using "sudo nano /etc/apache2/apache2.conf" we will edit this configuration and add ServerName (Your_server_domain_or_IP)

```
kali@kali:~$ sudo nano /etc/apache2/apache2.conf
kali@kali:~$ sudo apache2ctl configtest
Syntax OK
kali@kali:~$
```

Hence the message is gone. Now we restart apache to save.

```
kali@kali:~$ sudo systemctl restart apache2
kali@kali:~$
```


- Going to the public IP of the machine, we can see Apache is running



Apache2 Debian Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Debian systems. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Debian's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Debian tools. The configuration system is **fully documented in `/usr/share/doc/apache2/README.Debian.gz`**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Debian systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|   |-- ports.conf
|-- mods-enabled
|   |-- *.load
|   |-- *.conf
|-- conf-enabled
|   |-- *.conf
|-- sites-enabled
|   |-- *.conf
```

- Now we install Mysql for the lamp stack

```
kali@kali:~/var/lib$ sudo apt install mariadb-server
```

start the mysql service

```
kali@kali:~$ service mysql start
*** AUTHENTICATION FOR org.freedesktop.systemd1.Manager ***
Authentication is required to start 'mariadb.service'.
Authenticating as: kali... (kali)
Password:
*** AUTHENTICATION COMPLETE ***
kali@kali:~$ systemctl status mariadb.service
● mariadb.service - MariaDB 10.5.0 database server
   Loaded: loaded (/lib/systemd/system/mariadb.service; disabled; vendor preset: disabled)
   Active: active (running) since Thu 2021-02-18 01:09:32 EST; 28s ago
     Docs: man:mariadb(8)
           https://mariadb.com/kb/en/library/systemd/
   Process: 2234 ExecStartPre=/usr/bin/install -s 755 -o mysql -g root -d /usr/var/mysql (code=exited, status=0/SUCCESS)
   Process: 2236 ExecStartPre=/bin/sh -c systemctl unset-environment _RESPEC_START_POSITION (code=exited, status=0/SUCCESS)
   Process: 2238 ExecStartPre=/bin/sh -c [ ! -e /usr/bin/galera_recovery ] && VAR=11 VAR=/cd /usr/bin/.. /usr/bin/galera_recovery; [ 57 -eq 0 ] && syst
   Process: 2301 ExecStartPost=/bin/sh -c systemctl unset-environment _RESPEC_START_POSITION (code=exited, status=0/SUCCESS)
   Process: 2303 ExecStartPost=/etc/mysql/debian-start (code=exited, status=0/SUCCESS)
   Main PID: 2236 (mariadbd)
   Status: "Taking your SQL requests now..."
   Tasks: 14 (limit: 2333)
   Memory: 81.5M
   CGroup: /system.slice/mariadb.service
           └─2236 /usr/sbin/mariadbd

(11)~ Stopped                                systemctl status mariadb.service
kali@kali:~$
```

```
kali@kali:~$ sudo mysql
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 31
Server version: 10.5.8-MariaDB-3 Debian builddd-unstable

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)]> exit
Bye
kali@kali:~$
```

Hence mariadb (fork of mysql) is working)

- Now we install PHP to complete lamp stack

```
kali@kali:~$ sudo apt-get install php libapache2-mod-php php-mysql
Reading package lists... Done
Building dependency tree
Reading state information... Done
libapache2-mod-php is already the newest version (2:7.4+76).
php is already the newest version (2:7.4+76).
php-mysql is already the newest version (2:7.4+76).
```

Now we want apache to look for php files so we use

```
kali@kali:~$ sudo nano /etc/apache2/mods-enabled/dir.conf
```

and keep index.php before all extensions and save

```
GNU nano 4.5 /etc/apache2/mods-enabled
<IfModule mod_dir.c>
    DirectoryIndex index.php index.html index.cgi index.pl index.xhtml index.htm
</IfModule>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

Now we restart apache and check if it is running

```
kali@kali:~$ sudo systemctl restart apache2
kali@kali:~$ sudo systemctl status apache2
■ apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; disabled; vendor preset: disabled)
   Active: active (running) since Thu 2021-02-18 00:35:11 EST; 12s ago
     Docs: https://httpd.apache.org/docs/2.4/
   Process: 11805 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
   Main PID: 11810 (apache2)
     Tasks: 6 (limit: 2333)
    Memory: 12.8M
   CGroup: /system.slice/apache2.service
           └─11810 /usr/sbin/apache2 -k start
             11811 /usr/sbin/apache2 -k start
             11812 /usr/sbin/apache2 -k start
             11813 /usr/sbin/apache2 -k start
             11814 /usr/sbin/apache2 -k start
             11815 /usr/sbin/apache2 -k start

Feb 18 00:35:11 kali systemd[1]: Starting The Apache HTTP Server...
Feb 18 00:35:11 kali systemd[1]: Started The Apache HTTP Server.
kali@kali:~$
```

Installing php-cli

```
kali@kali:~$ sudo apt install php-cli
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  libcroc3
Use 'sudo apt autoremove' to remove it.
The following NEW packages will be installed:
  php-cli
0 upgraded, 1 newly installed, 0 to remove and 1181 not upgraded.
Need to get 6,796 B of archives.
After this operation, 24.6 kB of additional disk space will be used.
Get:1 http://kali.download/kali kali-rolling/main amd64 php-cli all 2:7.4+76 [6,796 B]
Fetched 6,796 B in 1s (4,671 B/s)
Selecting previously unselected package php-cli.
(Reading database ... 265857 files and directories currently installed.)
Preparing to unpack .../php-cli_2%3a7.4+76_all.deb ...
Unpacking php-cli (2:7.4+76) ...
Setting up php-cli (2:7.4+76) ...
update-alternatives: using /usr/bin/php.default to provide /usr/bin/php (php) in auto mode
update-alternatives: using /usr/bin/phar.default to provide /usr/bin/phar (phar) in auto mode
update-alternatives: using /usr/bin/phar.phar.default to provide /usr/bin/phar.phar (phar.phar) in auto mode
Processing triggers for man-db (2.9.0-2) ...
Processing triggers for kali-menu (2020.1.7) ...
kali@kali:~$
```


- LAMP STACK is not installed. To check it,

```
kali@kali:~$ cd ../../var/www
kali@kali:/var/www$ cd html
kali@kali:/var/www/html$ sudo touch info.php
kali@kali:/var/www/html$ sudo nano info.php
kali@kali:/var/www/html$
```

where in info.php we write

```
GNU nano 4.5
?php
phpinfo();
?>
```

- Now on public_ip/info.php we should get information, which proves LAMP works

PHP Version 7.3.12-1	
	
System	Linux kali 5.4.0-kali3-amd64 #1 SMP Debian 5.4.13-1kali1 (2020-01-20) x86_64
Build Date	Nov 28 2019 07:34:08
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.3/apache2
Loaded Configuration File	/etc/php/7.3/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/7.3/apache2/conf.d
Additional .ini files parsed	/etc/php/7.3/apache2/conf.d/10-opcache.ini, /etc/php/7.3/apache2/conf.d/10-pdo.ini, /etc/php/7.3/apache2/conf.d/20-calendar.ini, /etc/php/7.3/apache2/conf.d/20-ctype.ini, /etc/php/7.3/apache2/conf.d/20-exif.ini, /etc/php/7.3/apache2/conf.d/20-fileinfo.ini, /etc/php/7.3/apache2/conf.d/20-ftp.ini, /etc/php/7.3/apache2/conf.d/20-gettext.ini, /etc/php/7.3/apache2/conf.d/20-iconv.ini, /etc/php/7.3/apache2/conf.d/20-json.ini, /etc/php/7.3/apache2/conf.d/20-phar.ini, /etc/php/7.3/apache2/conf.d/20-posix.ini, /etc/php/7.3/apache2/conf.d/20-readline.ini, /etc/php/7.3/apache2/conf.d/20-shmop.ini, /etc/php/7.3/apache2/conf.d/20-sockets.ini, /etc/php/7.3/apache2/conf.d/20-sysvmsg.ini, /etc/php/7.3/apache2/conf.d/20-sysvsem.ini, /etc/php/7.3/apache2/conf.d/20-sysvshm.ini, /etc/php/7.3/apache2/conf.d/20-tokenizer.ini
PHP API	20180731
PHP Extension	20180731
Zend Extension	320180731
Zend Extension Build	API320180731.NTS
PHP Extension Build	API20180731.NTS
Debug Build	no

Installing Owncloud

Firstly we install all dependencies related to owncloud

```
root@kali:/home/kali# apt install -y php-imagick php-common php-curl php-gd php-imap php-intl php-json php-mbstring php-mysql php-ssh2
php-xml php-zip php-apcu php-redis redis-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
php-curl is already the newest version (2:7.4+76).
php-gd is already the newest version (2:7.4+76).
php-imagick is already the newest version (3.4.4-4.2).
php-intl is already the newest version (2:7.4+76).
php-mbstring is already the newest version (2:7.4+76).
php-mysql is already the newest version (2:7.4+76).
php-xml is already the newest version (2:7.4+76).
```

Now we create an OCC helper script

```
kali@kali:~$ sudo su
[sudo] password for kali:
root@kali:/home/kali# FILE="/usr/local/bin/occ"
root@kali:/home/kali# /bin/cat <<EOM >$FILE
> #!/bin/bash
>
> cd /var/www/owncloud
> sudo -u www-data /usr/bin/php /var/www/owncloud/occ "$@"
> EOM
root@kali:/home/kali#
```

Now we use chmod and make script executable

and after that we will use sed and change document root

Finally we create a Virtual Host Configuration

```
root@kali:/home/kali# chmod +x /usr/local/bin/occ
root@kali:/home/kali# sed -i "s#html#owncloud#" /etc/apache2/sites-available/000-default.conf
root@kali:/home/kali# FILE="/etc/apache2/sites-available/owncloud.conf"
root@kali:/home/kali# /bin/cat <<EOM >$FILE
> Alias /owncloud "/var/www/owncloud/"
>
> <Directory /var/www/owncloud/>
>     Options +FollowSymLinks
>     AllowOverride All
>
> <IfModule mod_dav.c>
>     Dav off
> </IfModule>
>
> SetEnv HOME /var/www/owncloud
> SetEnv HTTP_HOME /var/www/owncloud
> </Directory>
> EOM
root@kali:/home/kali# service apache2 start
```

We enable the Virtual Host Configuration and Enable Apache Modules with it

```
root@kali:/home/kali# a2ensite owncloud.conf
Enabling site owncloud.
To activate the new configuration, you need to run:
  systemctl reload apache2
root@kali:/home/kali# systemctl reload apache2
root@kali:/home/kali# echo "Enabling Apache Modules"
Enabling Apache Modules
root@kali:/home/kali#
root@kali:/home/kali# a2enmod dir env headers mime rewrite setenvif
Module dir already enabled
Module env already enabled
Enabling module headers.
Module mime already enabled
Enabling module rewrite.
Module setenvif already enabled
To activate the new configuration, you need to run:
  systemctl restart apache2
root@kali:/home/kali# service apache2 reload
root@kali:/home/kali#
```

Downloading and installing OwnCloud

```
root@kali:/# cd var/www/
root@kali:/var/www# wget https://download.owncloud.org/community/owncloud-10.6.0.tar.bz2 && \
> tar -xjf owncloud-10.6.0.tar.bz2 && \
> chown -R www-data: owncloud
--2021-02-19 03:13:02-- https://download.owncloud.org/community/owncloud-10.6.0.tar.bz2
Resolving download.owncloud.org (download.owncloud.org)... 167.233.14.167, 2a01:4f8:1c1d:3d1::1
Connecting to download.owncloud.org (download.owncloud.org)|167.233.14.167|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 24569416 (23M) [application/x-bzip2]
Saving to: 'owncloud-10.6.0.tar.bz2'

owncloud-10.6.0.tar.bz2      100%[=====>] 23.43M  2.16MB/s   in 24s

2021-02-19 03:13:27 (1008 KB/s) - 'owncloud-10.6.0.tar.bz2' saved [24569416/24569416]

root@kali:/var/www#
```

Configuring the database

```
root@kali:/var/www# mysql -u root -e "CREATE DATABASE IF NOT EXISTS owncloud; \
> GRANT ALL PRIVILEGES ON owncloud.* \
> TO owncloud@localhost \
> IDENTIFIED BY 'Your_db_pass'";
```

Using occ to successfully install owncloud

```
root@kali:/var/www# occ maintenance:install --database "mysql" --database-name "owncloud" --database-user "owncloud" --
database-pass "your_db_pass" --admin-user "admin" --admin-pass "your_admin_pass"
ownCloud was successfully installed
root@kali:/var/www#
```

3. Configure

After Owncloud is installed we need to set it up

First we configure ownclouds trusted domain list

```
root@kali:/var/www# myip=$(hostname -I|cut -f1 -d ' ')
root@kali:/var/www# occ config:system:set trusted_domains 1 --value="$myip"
System config value trusted_domains => 1 set to string your_public_ip_will be shown here
```

Now we setup a cron job

```
root@kali:/var/www# occ background:cron
Set mode for background jobs to 'cron'
root@kali:/var/www#
root@kali:/var/www# echo "*/15 * * * * /var/www/owncloud/occ system:cron" \
> > /var/spool/cron/crontabs/www-data
root@kali:/var/www# chown www-data.crontab /var/spool/cron/crontabs/www-data
root@kali:/var/www# chmod 0600 /var/spool/cron/crontabs/www-data
root@kali:/var/www#
root@kali:/var/www# echo "*/15 * * * * /var/www/owncloud/occ user:sync 'OCA\User_LDAP\User_Proxy' -n disable -vvv >> /var/log/ldap-sync
/user-sync.log 2>&1" > /var/spool/cron/crontabs/www-data
root@kali:/var/www# chown www-data.crontab /var/spool/cron/crontabs/www-data
root@kali:/var/www# chmod 0600 /var/spool/cron/crontabs/www-data
root@kali:/var/www# mkdir -p /var/log/ldap-sync
root@kali:/var/www# touch /var/log/ldap-sync/user-sync.log
root@kali:/var/www# chown www-data, /var/log/ldap-sync/user-sync.log
```

Now we configure caching and file locking

```
root@kali:/var/www# occ config:system:set \
> memcache.local \
> --value '\OC\Memcache\APCu'
System config value memcache.local set to string \OC\Memcache\APCu
root@kali:/var/www#
root@kali:/var/www#
root@kali:/var/www# occ config:system:set \
> memcache.locking \
> --value '\OC\Memcache\Redis'
System config value memcache.locking set to string \OC\Memcache\Redis
root@kali:/var/www#
root@kali:/var/www#
root@kali:/var/www# service redis-server start
root@kali:/var/www#
root@kali:/var/www# occ config:system:set \
> redis \
> --value '{"host": "127.0.0.1", "port": "6379"}' \
> --type json
System config value redis set to json {"host": "127.0.0.1", "port": "6379"}
root@kali:/var/www#
```

Now we configure Log Rotation

```
root@kali:/var/www# FILE="/etc/logrotate.d/owncloud"
root@kali:/var/www# sudo /bin/cat <<EOM >$FILE
> /var/www/owncloud/data/owncloud.log {
>   size 10M
>   rotate 12
>   copytruncate
>   missingok
>   compress
>   compresscmd /bin/gzip
> }
> EOM
root@kali:/var/www#
root@kali:/var/www# chown -R www-data: owncloud
root@kali:/var/www# systemctl restart apache2
root@kali:/var/www#
```

At this point, Php code wont run when we start the server so , we go to apache config and do the changes below

```
root@kali:/etc/apache2# sudo nano apache2.conf
```

```
File Edit View Terminal Tabs Help
GNU nano 4.5
</Directory>

<Directory /usr/share>
    AllowOverride None
    Require all granted
</Directory>

<Directory /var/www/>
    Options Indexes FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>

<IfModule php7.4_module>
    AddType application/x-httpd-php .php
    AddType application/x-httpd-php-source .phps
    <IfModule dir_module>
        DirectoryIndex index.html index.php
    </IfModule>
</IfModule>
#<Directory /srv/>
```

Executing the changes

```
root@kali:/etc/apache2# a2enmod php7.4
Considering dependency mpm_prefork for php7.4:
Considering conflict mpm_event for mpm_prefork:
Considering conflict mpm_worker for mpm_prefork:
Module mpm_prefork already enabled
Considering conflict php5 for php7.4:
Enabling module php7.4.
To activate the new configuration, you need to run:
  systemctl restart apache2
root@kali:/etc/apache2# systemctl restart apache2
```

Finally we have OwnCloud configured

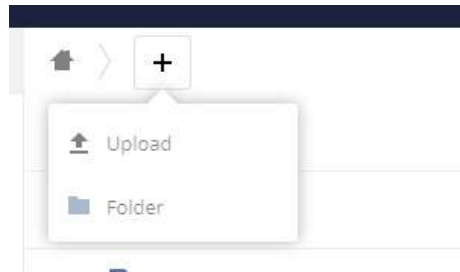


Now we use the admin credentials and log in and we get the dashboard as shown below

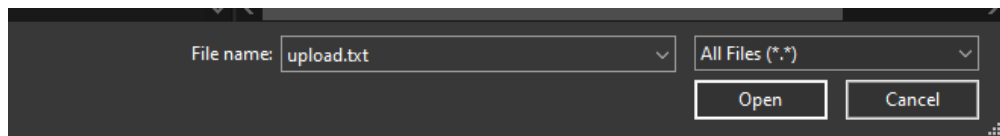


4. Upload documents

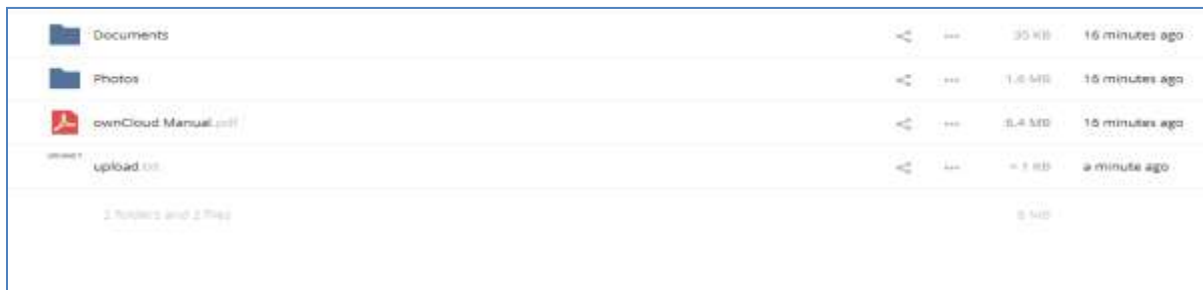
To upload documents we click on plus Icon and it gives us two options




we can upload a folder or file, Lets upload a file

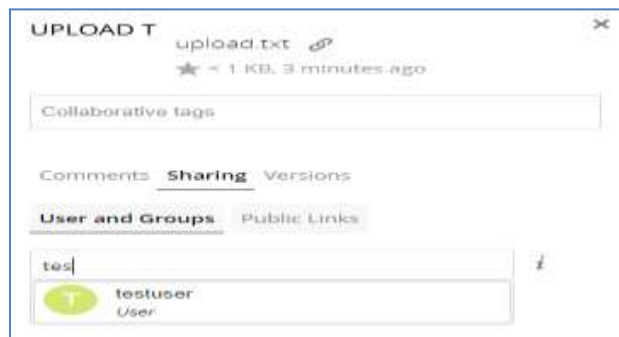


Now on our dashboard we can see that the file is uploaded

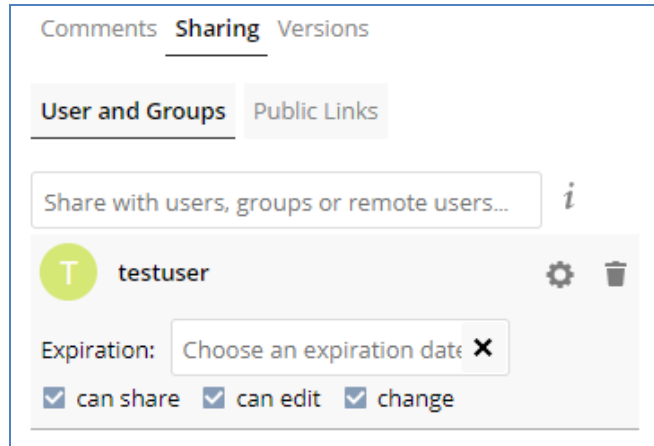


5. Share with users

To share a file we will click on the share button () besides the file. Lets share it with a user called "testuser"



Now that user is added, we can select what access the user can have on the file(s).



6. Access file from user account

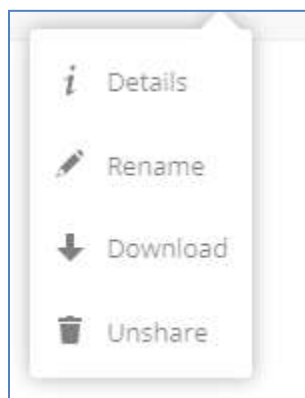
Lets login to testuser account and see if we have upload.txt



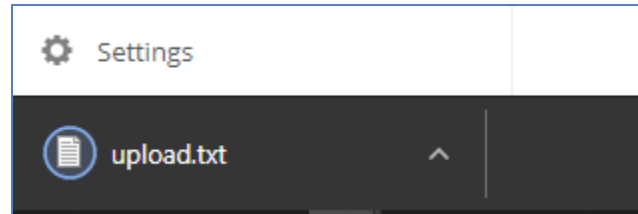
As we can see on the top corner, we are logged in through testuser account and we have upload.txt shared by "admin"

7. Download it

To download the file(s) we click on the three dots icon (**...**) near share button and we get



Now we click on download and it gets downloaded



CONCLUSION :

From this experiment we studied about OwnCloud which is a Storage as a Service(SaaS). OwnCloud is a suite of client–server software for creating and using file hosting services. It is similar to Google Drive or Dropbox as per functionality. We prepared a case study on ownCloud and understood its architecture and features. We also understood its advantages and disadvantages. Finally we installed OwnCloud in Kali Linux virtual machine and then tested the features on the dashboard like adding users, upload, sharing and downloading files.

REFERENCES :

1. <https://www.digitalocean.com/community/tutorials/how-to-install-and-configure-owncloud-on-ubuntu-18-04>
2. https://doc.owncloud.org/server/10.5/admin_manual/installation/quick_guides/ubuntu_18_04.html
3. <https://central.owncloud.org/t/owncloud-do-not-show-start-page/11732/9>
4. https://owncloud.com/wp-content/uploads/2014/03/oc_architecture_overview.pdf