Contents

1. Project Introduction

Project Title: NiCloud (Local NAS) – Phase 2
Nirmal Thapa, Minnesota State University Mankato
Date: September 5, 2024

2. Executive Summary

Brief Overview:
At present cloud services are widely used by individuals and organizations for effective, secure, and easy data management solutions. This research proposal aims to develop and implement a local NAS (Network Attached Storage) for individuals looking for solutions to store their data without having to pay other external vendors like Google Cloud, Apple Cloud, or OneDrive. This research addresses data storage, performance, power consumption, and data access reliability. NiCloud is a personal storage server that helps to store files, videos, music etc.

3. Problem Statement

Each day a large amount of digital data is collected (Pictures, videos and in other forms). This data needs designated solutions for data retention. Devices, for example, iPhones, run out of storage capacity eventually. Users are recommended to back up the data either by purchasing more storage in the cloud or transferring it to different traditional storage devices like pen drives and hard drives. The process of collecting data never stops but instead increases significantly. Cloud subscribers have no option but to buy more storage by paying more money each month. of traditional devices like pen drives poses a huge risk to data integrity and security as those devices can easily be lost or accessed by other people. This project will provide data storage solution by developing a local NAS that stores digital data at an individual level.

The research aims to provide storage capacity up to 1 TB in the initial phase and continue to make the NAS expandable in the later phases. After the project's completion, individuals will no longer have to keep paying for cloud services or carry traditional storage devices.

4. Project Objectives

- Development and implementation of a local NAS system: A functioning NAS will be developed that has storage capacity of 1 TB; download and upload speed of 100mbps.
- Increase reliability and performance: Make sure NAS will be accessible for authorized users for data storage whenever needed.
- Evaluate cost effectiveness: Comparative Analysis on initial cost to set up, maintenance and power consumption will be carried out.
- Storage expansion capability: It will also allow us to expand storage capacity by adding more HDD or SSD.

5. Requirements:

1.1. Raspberry pi
1.2. Power Adapter 5V/5A
1.3. SD card (Preferred 64 GB)
1.4. SD card reader
1.4. Penta SATA HAT (4 SATA + 1 eSATA interface)
1.5. Power adapter for SATA HAT
1.6. Hard Drive (1-5)
1.7. FPC

6. Installation of Nextcloud in Ubuntu Server

Phase 2 is continuation of phase 1. Click here to access Phase 1.

If you run into any problems, visit nextcloud.com. They have blogs, community support, documentation, FAQs, and user manuals that may help to solve the issue.

Disclaimer: Procedures listed below have been tested with Raspberry Pi 3 B+ and Raspberry Pi 5.

Recommended Environments: Use of newly flashed SD card in Raspberry Pi to avoid process and port conflicts (preferred) with existing applications. OS: Ubuntu Server 22.04.5 LTS/Ubuntu Server 24.04.1 LTS. Nextcloud version 29.0.7. This paper explains procedures to mount External Storage Drives in Nextcloud, before that the drives need to be mounted and configured in the Ubuntu server. Please refer to Paper 1 for that.

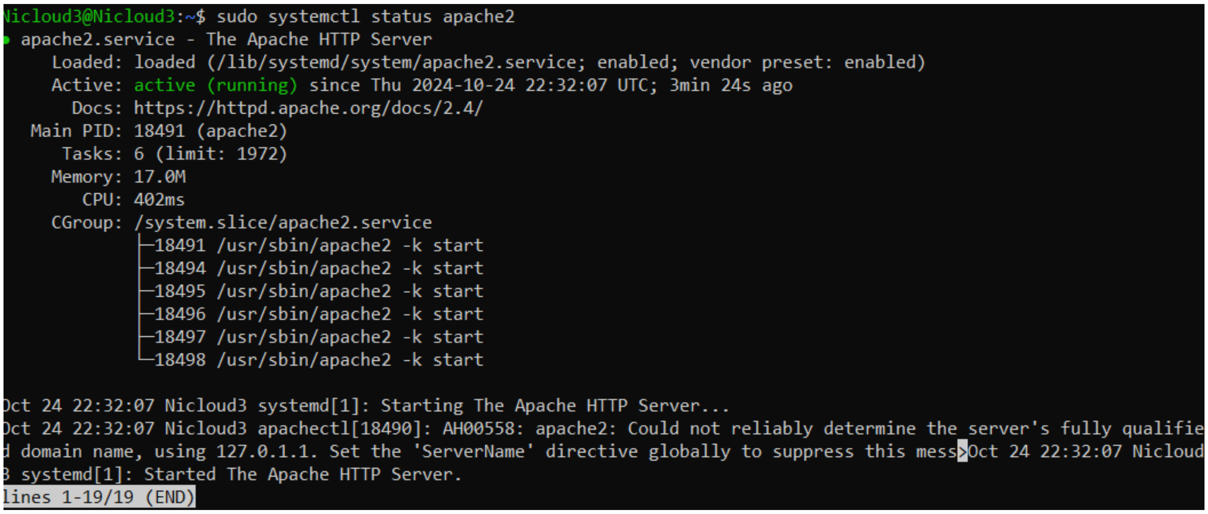
6.1. Installation of Apache2 and MariaDB

1. Use the command given in the box below to install Apache2 server, MariaDB and other required packages.

```
sudo apt install apache2 mariadb-server libapache2-mod-php php php-mysql php-cli php-zip php-curl php-gd php-mbstring php-intl php-xml php-json php-ldap php-imagick php-gmp unzip -y
```

Note: It will take few minutes to unzip and install all the packages (2-5 minutes)

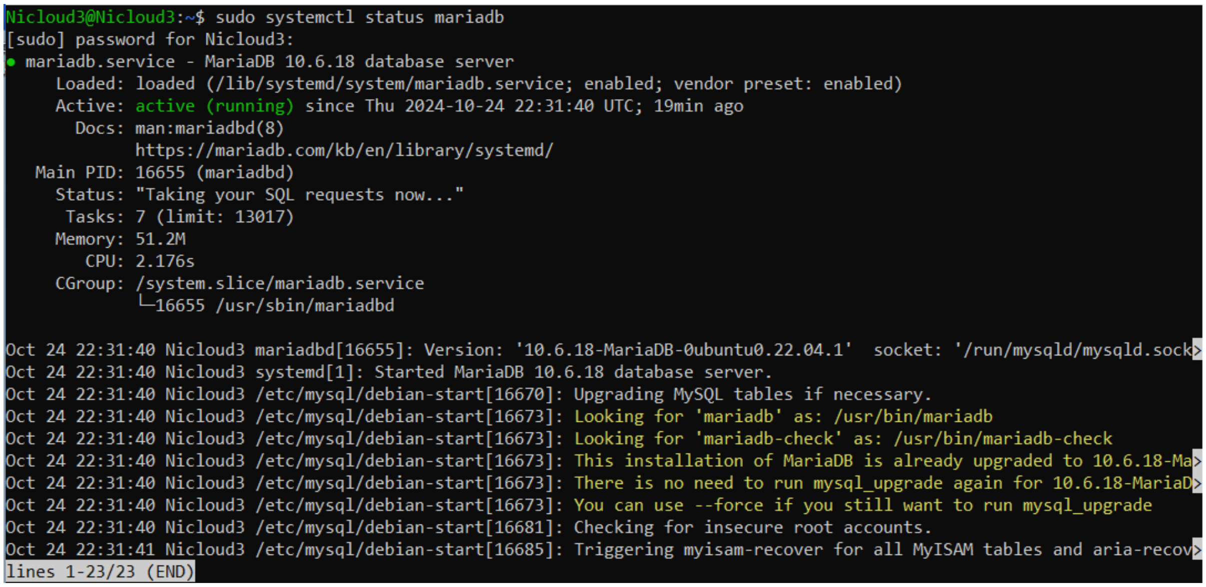
2. Check the status of apche2 server using command: *sudo systemctl status apache2*



```
Nicloud3@Nicloud3:~$ sudo systemctl status apache2
• apache2.service - The Apache HTTP Server
     Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
     Active: active (running) since Thu 2024-10-24 22:32:07 UTC; 3min 24s ago
       Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 18491 (apache2)
      Tasks: 6 (limit: 1972)
     Memory: 17.0M
        CPU: 402ms
     CGroup: /system.slice/apache2.service
             ├─18491 /usr/sbin/apache2 -k start
             ├─18494 /usr/sbin/apache2 -k start
             ├─18495 /usr/sbin/apache2 -k start
             ├─18496 /usr/sbin/apache2 -k start
             ├─18497 /usr/sbin/apache2 -k start
             └─18498 /usr/sbin/apache2 -k start

Oct 24 22:32:07 Nicloud3 systemd[1]: Starting The Apache HTTP Server...
Oct 24 22:32:07 Nicloud3 apachectl[18490]: AH00558: apache2: Could not reliably determine the server's fully qualifie
d domain name, using 127.0.1.1. Set the 'ServerName' directive globally to suppress this mess>Oct 24 22:32:07 Nicloud
3 systemd[1]: Started The Apache HTTP Server.
lines 1-19/19 (END)
```

Fig 1 Apache2 status

3. Make sure that Apache2 server *is enabled, active* and *running*. If you do not get similar result as Fig 1, reinstall apache2 by using command *sudo apt install apache2*. Use commands *sudo systemctl enable apache2* to enable and *sudo systemctl start apache2*, to start apache2. Check the status again with command - *sudo systemctl status apache2*.
4. Check the status of MariaDB, use command *sudo systemctl status mariadb*. Make sure it is *active* and *running*.



```
Nicloud3@Nicloud3:~$ sudo systemctl status mariadb
[sudo] password for Nicloud3:
● mariadb.service - MariaDB 10.6.18 database server
     Loaded: loaded (/lib/systemd/system/mariadb.service; enabled; vendor preset: enabled)
     Active: active (running) since Thu 2024-10-24 22:31:40 UTC; 19min ago
       Docs: man:mariadbd(8)
             https://mariadb.com/kb/en/library/systemd/
   Main PID: 16655 (mariadbd)
     Status: "Taking your SQL requests now..."
      Tasks: 7 (limit: 13017)
     Memory: 51.2M
        CPU: 2.176s
     CGroup: /system.slice/mariadb.service
             └─16655 /usr/sbin/mariadbd

Oct 24 22:31:40 Nicloud3 mariadbd[16655]: Version: '10.6.18-MariaDB-0ubuntu0.22.04.1'  socket: '/run/mysqld/mysqld.sock>
Oct 24 22:31:40 Nicloud3 systemd[1]: Started MariaDB 10.6.18 database server.
Oct 24 22:31:40 Nicloud3 /etc/mysql/debian-start[16670]: Upgrading MySQL tables if necessary.
Oct 24 22:31:40 Nicloud3 /etc/mysql/debian-start[16673]: Looking for 'mariadb' as: /usr/bin/mariadb
Oct 24 22:31:40 Nicloud3 /etc/mysql/debian-start[16673]: Looking for 'mariadb-check' as: /usr/bin/mariadb-check
Oct 24 22:31:40 Nicloud3 /etc/mysql/debian-start[16673]: This installation of MariaDB is already upgraded to 10.6.18-Ma>
Oct 24 22:31:40 Nicloud3 /etc/mysql/debian-start[16673]: There is no need to run mysql_upgrade again for 10.6.18-MariaD>
Oct 24 22:31:40 Nicloud3 /etc/mysql/debian-start[16673]: You can use --force if you still want to run mysql_upgrade
Oct 24 22:31:40 Nicloud3 /etc/mysql/debian-start[16681]: Checking for insecure root accounts.
Oct 24 22:31:41 Nicloud3 /etc/mysql/debian-start[16685]: Triggering myisam-recover for all MyISAM tables and aria-recov>
lines 1-23/23 (END)
```

Fig 2 MariaDB status

5. Secure MariaDB, use command *sudo mysql_secure_installation*. This command opens MariaDB. Multiple Questions will be asked to configure and clean database. All the questions and answers are listed below.
   i. Enter current password for root (enter for none): There is no root password, just hit *Enter*.
   ii. Switch to unix_socket authentication [Y/n]: Just hit *Enter.*
   iii. Change the root password? [Y/n]: *Y*
      - New password: *dnc662* (Note: Create your own password and note it down somewhere, we will need it in later steps)
      - Re-enter new password: *dnc662*
   iv. Remove anonymous users? [Y/n]: Just hit *Enter.*
   v. Disallow root login remotely? [Y/n] : Just hit *Enter.*
   vi. Remove test database and access to it? [Y/n]: Just hit *Enter.*
   vii. Reload privilege tables now? [Y/n] : Just hit *Enter.*
6. Login to the MariaDB using command: *sudo mysql -u root -p*
   i. Enter password: *dnc662* (Enter password created in step 5(iii))

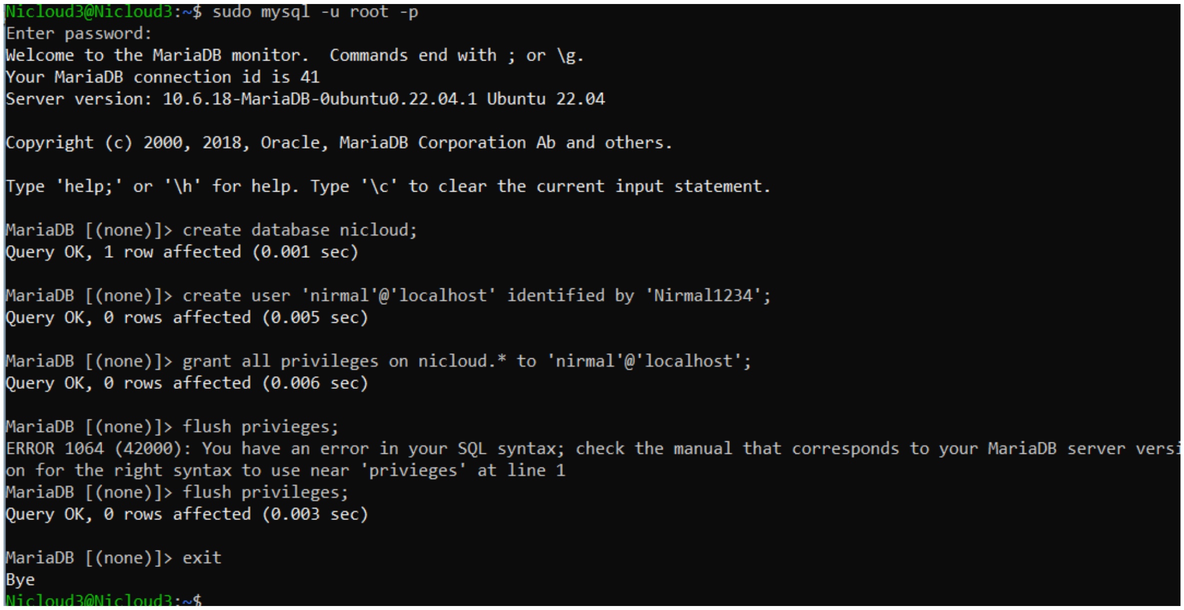
ii. Create name of the database: MariaDB [(none)]> *create database nicloud;*
(You can give your own database name)
Note: Make sure to note down this information, it will be required in later steps.

iii. Create database user: MariaDB [(none)]> *create user 'nirmal'@'localhost' identified by 'Nirmal1234';*
(You can create your own user and password)
Note: Make sure to note down this information; it will be required in later steps.

iv. Grant all privileges to the user: MariaDB [(none)]> *grant all privileges on nicloud.* to 'nirmal'@'localhost';*
Note: Instead of 'nicloud' and 'nirmal' enter your own database name and user respectively.

v. Apply granted privileges: *MariaDB [(none)]> flush privileges;*

vi. Exit from database: *MariaDB [(none)]> exit*

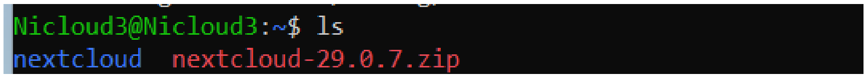


```
Nicloud3@Nicloud3:~$ sudo mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 41
Server version: 10.6.18-MariaDB-0ubuntu0.22.04.1 Ubuntu 22.04

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 nicloud;
Query OK, 1 row affected (0.001 sec)

MariaDB [(none)]> create user 'nirmal'@'localhost' identified by 'Nirmal1234';
Query OK, 0 rows affected (0.005 sec)

MariaDB [(none)]> grant all privileges on nicloud.* to 'nirmal'@'localhost';
Query OK, 0 rows affected (0.006 sec)

MariaDB [(none)]> flush privieges;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server versi
on for the right syntax to use near 'privieges' at line 1
MariaDB [(none)]> flush privileges;
Query OK, 0 rows affected (0.003 sec)

MariaDB [(none)]> exit
Bye
Nicloud3@Nicloud3:~$
```

Fig 3 Database Configuration

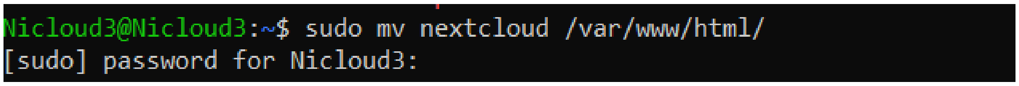
6.2. Nextcloud Installation

1. Download NextCloud zip file into the server using wget:
   wget https://download.nextcloud.com/server/releases/nextcloud-29.0.7.zip
   Note: It will take few minutes (1-2 minutes)
2. Once downloaded, unzip the file: *unzip nextcloud-29.0.7.zip* (Note: It will take 1 or 2 minutes).
3. Once unzipped, type *ls* command, there should be a folder named 'nextcloud' in the current directory.



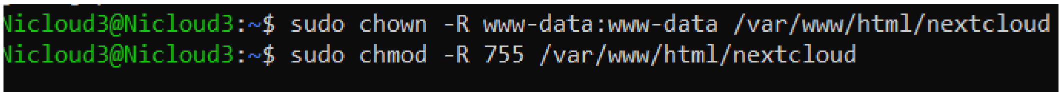
```
Nicloud3@Nicloud3:~$ ls
nextcloud  nextcloud-29.0.7.zip
```

4. Move ‘nextcloud’ folder to the apache2 directory where web files are stored so when IP address of the server is typed in the browser, NextCloud site is displayed. Use command *sudo mv nextcloud /var/www/html/*



```
Nicloud3@Nicloud3:~$ sudo mv nextcloud /var/www/html/
[sudo] password for Nicloud3:
```

5. Make changes to the ownership and permission to that directory
   i. Change ownership of ‘nextcloud’ folder and other files or folders inside ‘nextcloud’, use command: *sudo chown -R www-data:www-data /var/www/html/nextcloud*
   ii. Set permissions to the folder, use command: *sudo chmod -R 755 /var/www/html/nextcloud*



```
Nicloud3@Nicloud3:~$ sudo chown -R www-data:www-data /var/www/html/nextcloud
Nicloud3@Nicloud3:~$ sudo chmod -R 755 /var/www/html/nextcloud
```

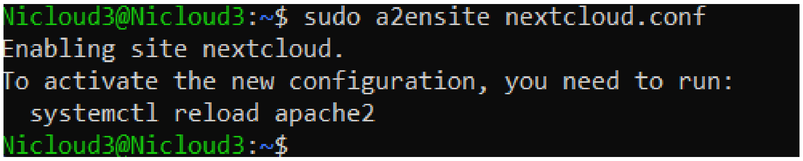
6. Create a configuration file, it can be found in documentation on Nextcloud’s website or use the one in the textbox below. Use command: *sudo nano /etc/apache2/sites-available/nextcloud.conf*. This command opens a text editor.
7. Once the Text Editor opens. Paste commands in the Text Editor from the text box below.
   i. To save configuration file*: ctrl+o* and Hit *Enter*

ii. To close Text Editor: *ctrl+x*

```
<VirtualHost *:80>
 DocumentRoot /var/www/html/nextcloud
 ServerName  192.168.1.75  (Note: Enter IP address of your server here)

 <Directory /var/www/html/nextcloud/>
  Options +FollowSymLinks
  AllowOverride All

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

  SetEnv HOME /var/www/html/nextcloud
  SetEnv HTTP_HOME /var/www/html/nextcloud

 </Directory>
 ErrorLog ${APACHE_LOG_DIR}/nextcloud_error.log
 CustomLog ${APACHE_LOG_DIR}/nextcloud_access.log combined
</VirtualHost>
```

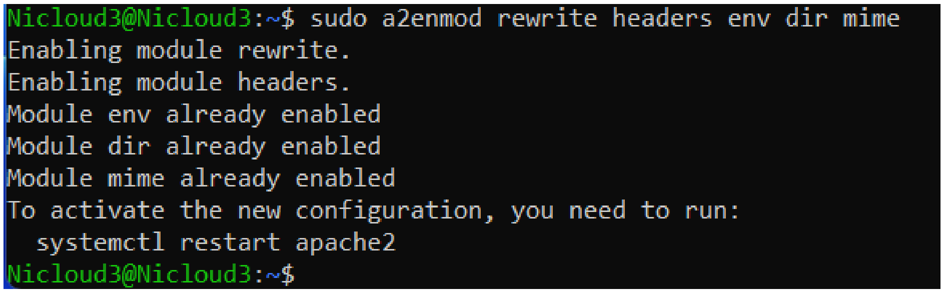
File: nextcloud.conf

8. After the configuration file is saved, enable the Nextcloud site using command*: sudo a2ensite nextcloud.conf*



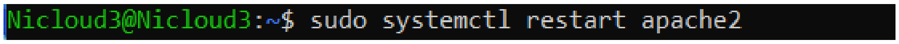
```
Nicloud3@Nicloud3:~$ sudo a2ensite nextcloud.conf
Enabling site nextcloud.
To activate the new configuration, you need to run:
  systemctl reload apache2
Nicloud3@Nicloud3:~$
```

9. Enable other mods using command: *sudo a2enmod rewrite headers env dir mime*



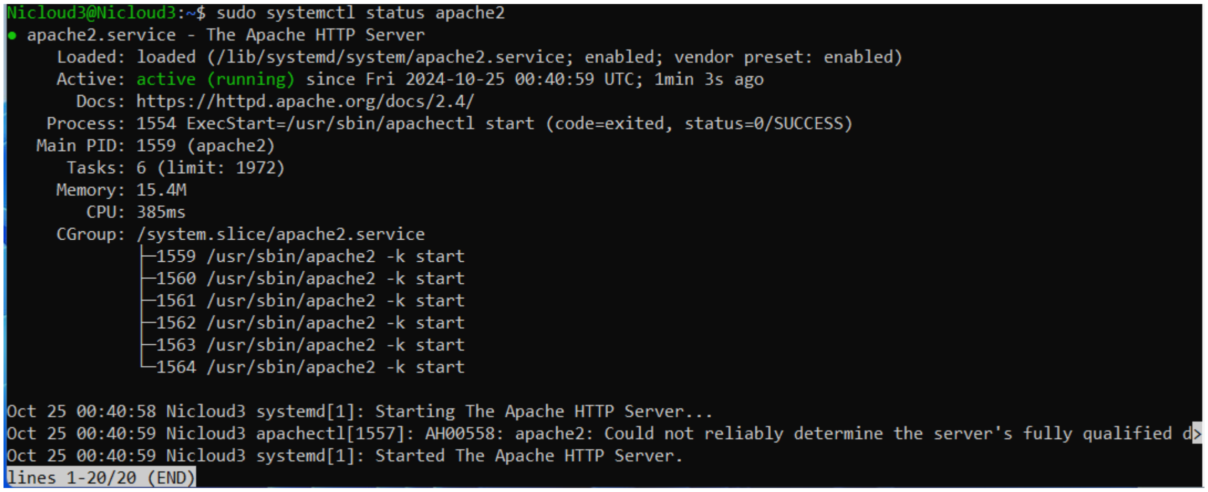
```
Nicloud3@Nicloud3:~$ sudo a2enmod rewrite headers env dir mime
Enabling module rewrite.
Enabling module headers.
Module env already enabled
Module dir already enabled
Module mime already enabled
To activate the new configuration, you need to run:
  systemctl restart apache2
Nicloud3@Nicloud3:~$
```

10. Restart the server. Use command: *sudo systemctl restart apache2*



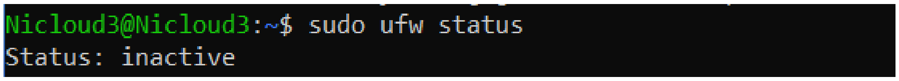
```
Nicloud3@Nicloud3:~$ sudo systemctl restart apache2
```

11. Check the status of the server: *sudo systemctl status apache2*



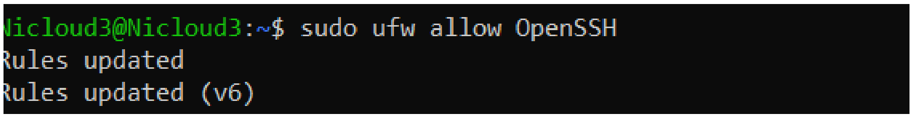
```
Nicloud3@Nicloud3:~$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
     Active: active (running) since Fri 2024-10-25 00:40:59 UTC; 1min 3s ago
       Docs: https://httpd.apache.org/docs/2.4/
    Process: 1554 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
   Main PID: 1559 (apache2)
      Tasks: 6 (limit: 1972)
     Memory: 15.4M
        CPU: 385ms
     CGroup: /system.slice/apache2.service
             ├─1559 /usr/sbin/apache2 -k start
             ├─1560 /usr/sbin/apache2 -k start
             ├─1561 /usr/sbin/apache2 -k start
             ├─1562 /usr/sbin/apache2 -k start
             ├─1563 /usr/sbin/apache2 -k start
             └─1564 /usr/sbin/apache2 -k start

Oct 25 00:40:58 Nicloud3 systemd[1]: Starting The Apache HTTP Server...
Oct 25 00:40:59 Nicloud3 apachectl[1557]: AH00558: apache2: Could not reliably determine the server's fully qualified d>
Oct 25 00:40:59 Nicloud3 systemd[1]: Started The Apache HTTP Server.
lines 1-20/20 (END)
```

12. Check the status of firewall: *sudo ufw status*



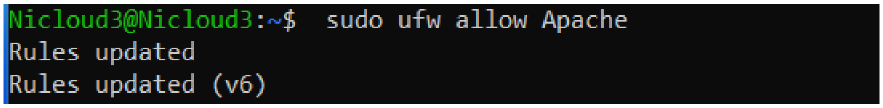
```
Nicloud3@Nicloud3:~$ sudo ufw status
Status: inactive
```

13. Before turning on the Firewall, allow ssh connection by using command: *sudo ufw allow OpenSSH*



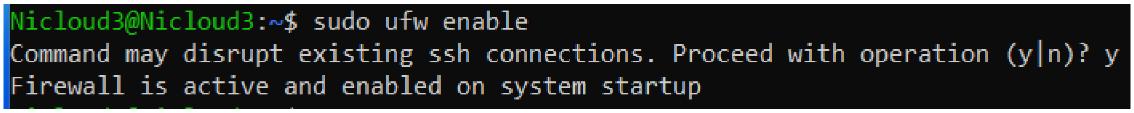
```
Nicloud3@Nicloud3:~$ sudo ufw allow OpenSSH
Rules updated
Rules updated (v6)
```

14. Allow Apache server across firewall (It should open up port 80): *sudo ufw allow Apache*



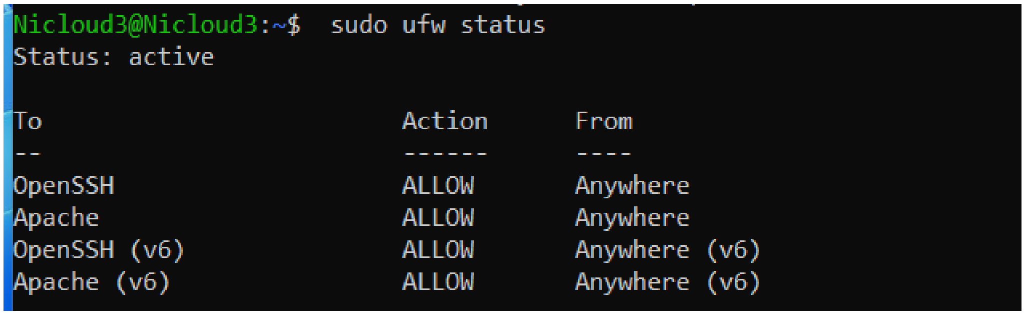
```
Nicloud3@Nicloud3:~$  sudo ufw allow Apache
Rules updated
Rules updated (v6)
```

15. Enable Firewall: *sudo ufw enable*



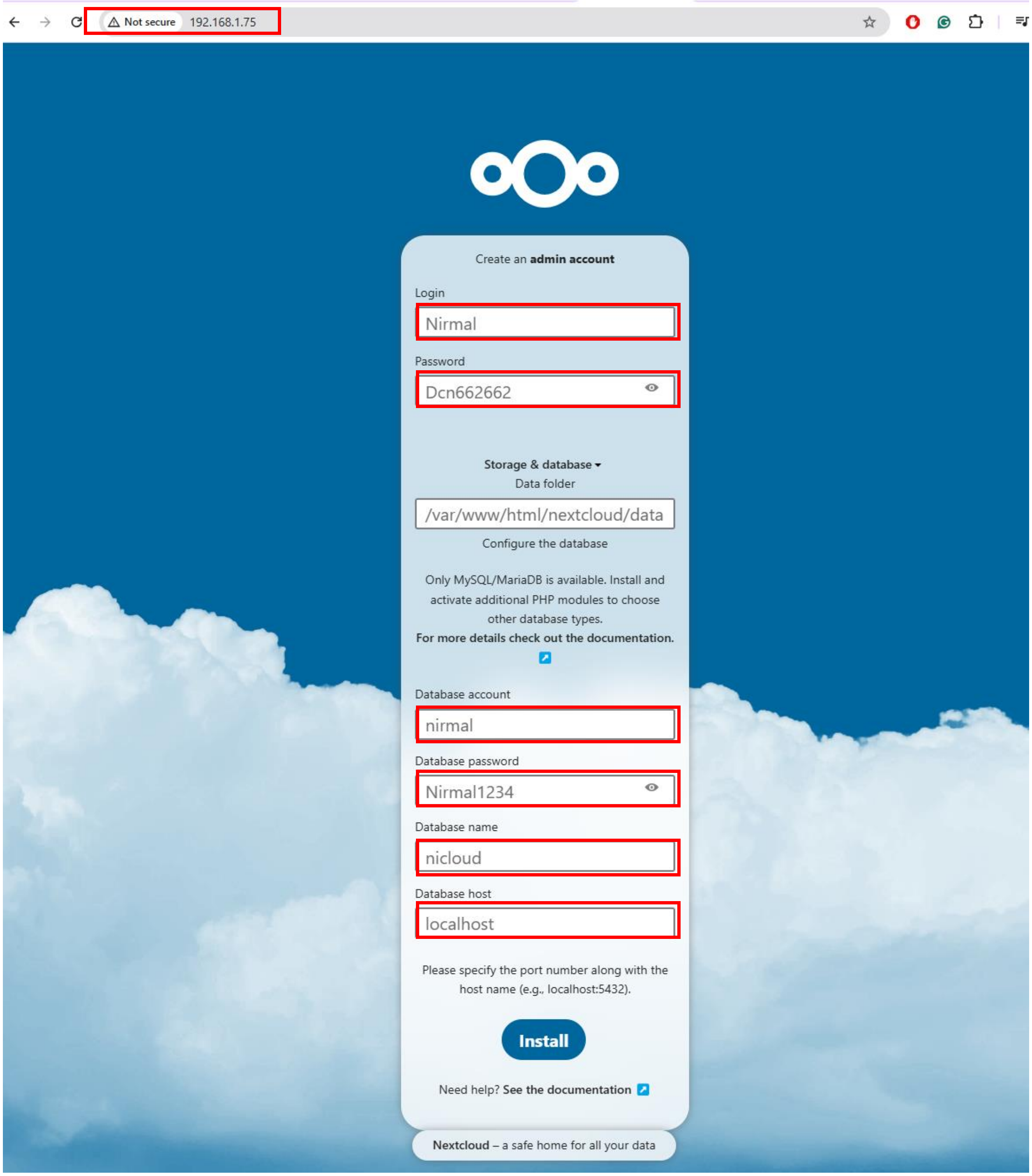
```
Nicloud3@Nicloud3:~$ sudo ufw enable
Command may disrupt existing ssh connections. Proceed with operation (y|n)? y
Firewall is active and enabled on system startup
```

16. Check the status on Firewall*: sudo ufw status*



```
Nicloud3@Nicloud3:~$  sudo ufw status
Status: active

To                         Action      From
--                         ------      ----
OpenSSH                    ALLOW       Anywhere
Apache                     ALLOW       Anywhere
OpenSSH (v6)               ALLOW       Anywhere (v6)
Apache (v6)                ALLOW       Anywhere (v6)
```

17. Go to the browser and type in the IP address of the server. NextCloud login page should be displayed. (Note: To know the IP address, type command *ifconfig)*
    i. Create username of your choice, for me I will keep '*Nirmal*'.
    ii. Create your password, for me I will use '*Dcn662662*'
    iii. In Database account field, use database user name created in step 6(iii), I had created user '*nirmal'*, I will use that, use what you had created.
    iv. In Database password field, use database password created in step 6(iii), I had created password '*Nirmal1234*', I will use that, use what you had created.
    v. In Database name field, use database name created in step 6(ii), I had created database name '*nicloud*', I will use that, use what you had created.
    vi. In Database host field, use 'localhost'. It should be auto populated.
    vii. Click on Install.



Not secure 192.168.1.75

Create an **admin account**

Login: Nirmal

Password: Dcn662662

Storage & database ▾

Data folder: /var/www/html/nextcloud/data

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.

Database account: nirmal

Database password: Nirmal1234

Database name: nicloud

Database host: localhost

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

Install

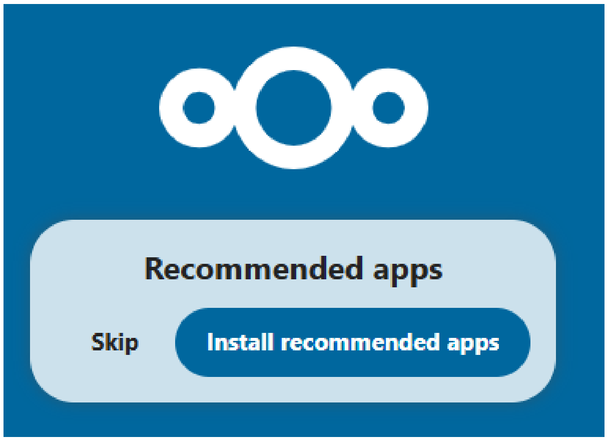
Need help? **See the documentation**

Nextcloud – a safe home for all your data

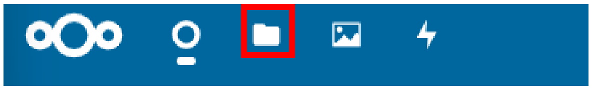
Note: It takes a few minutes to configure the site.

6.3. Start using NextCloud

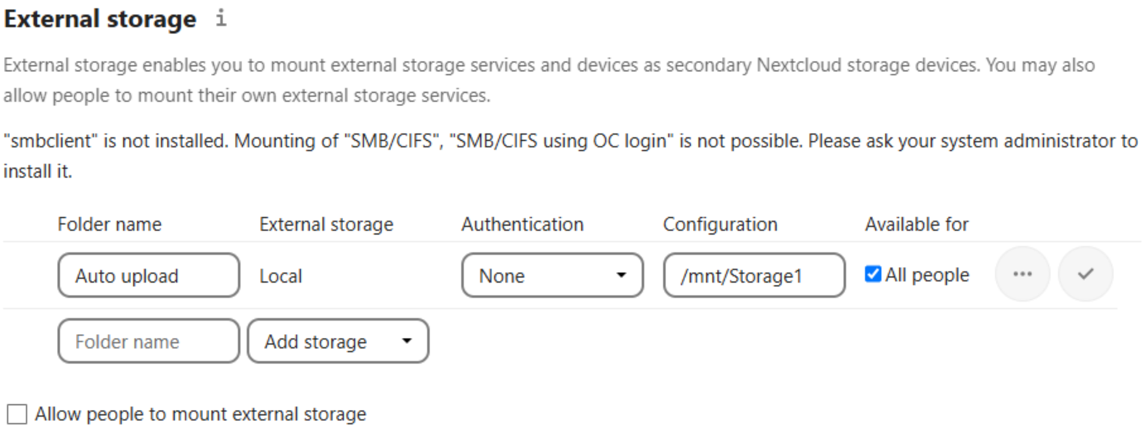
1. Start using NextCloud. It may ask to install recommended apps. You can skip it.



2. To upload files into the server, click on the files tab in the navigation bar



3. To mount external hard drives connected to the server:
   i. Click on *profile picture*→ from drop-down *menu*→ click on *Apps*→ Click on *Disabled apps*, from the list of apps, find *External storage Support*→ Click on *Enable button*.
   ii. Click on profile picture again→ from the menu→ click on Administrative settings→ on left hand side navigation panel find and click on External Storage under Administration (Note: Do not go to External storage under Personal, it is to view only, it will not allow to mount drives.)



External storage i

External storage enables you to mount external storage services and devices as secondary Nextcloud storage devices. You may also allow people to mount their own external storage services.

"smbclient" is not installed. Mounting of "SMB/CIFS", "SMB/CIFS using OC login" is not possible. Please ask your system administrator to install it.

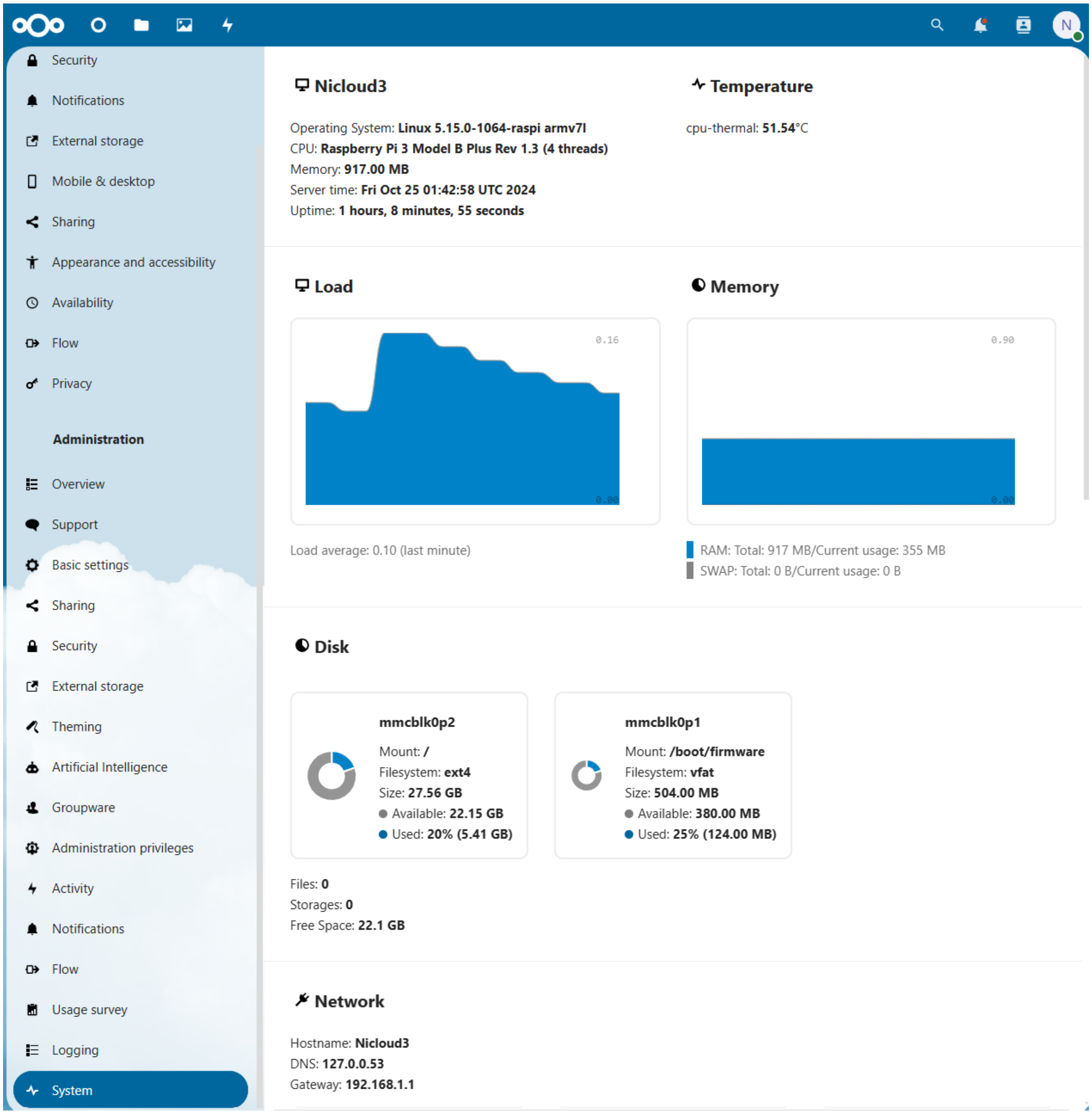
Folder name	External storage	Authentication	Configuration	Available for		
Auto upload	Local	None ▾	/mnt/Storage1	☑ All people	···	✓
Folder name	Add storage ▾					

☐ Allow people to mount external storage

- Folder name can be of your choice, you can name anything.
- Under External Storage, choose '*Local*' from the drop-down options.
- Authentication: none

- Configuration: Find in which folder the drive is mounted, type the exact same path directory. For me, I have mounted the drive in */mnt/Storage1* directory.
- Manage access, who do you want to give access to?
- Click on the check mark
- The drive is mounted and ready to use.
- To start uploading data, click on the files tab, find and go into the folder created above.

4. To View system information, click on the System option under Administration.



Nicloud3

Operating System: **Linux 5.15.0-1064-raspi armv7l**
CPU: **Raspberry Pi 3 Model B Plus Rev 1.3 (4 threads)**
Memory: **917.00 MB**
Server time: **Fri Oct 25 01:42:58 UTC 2024**
Uptime: **1 hours, 8 minutes, 55 seconds**

Temperature

cpu-thermal: **51.54**°C

Load

Load average: 0.10 (last minute)

Memory

RAM: Total: 917 MB/Current usage: 355 MB
SWAP: Total: 0 B/Current usage: 0 B

Disk

mmcblk0p2
Mount: **/**
Filesystem: **ext4**
Size: **27.56 GB**
● Available: **22.15 GB**
● Used: **20% (5.41 GB)**

mmcblk0p1
Mount: **/boot/firmware**
Filesystem: **vfat**
Size: **504.00 MB**
● Available: **380.00 MB**
● Used: **25% (124.00 MB)**

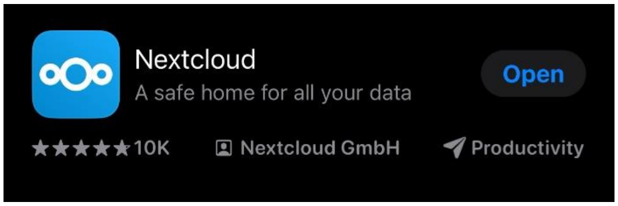
Files: **0**
Storages: **0**
Free Space: **22.1 GB**

Network

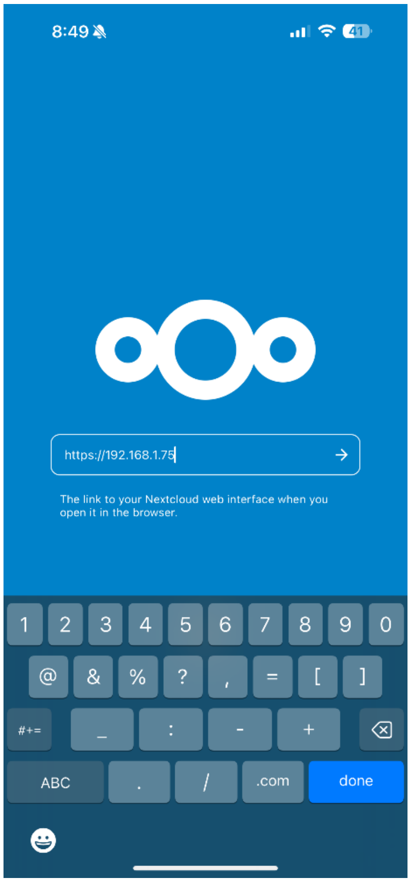
Hostname: **Nicloud3**
DNS: **127.0.0.53**
Gateway: **192.168.1.1**

6.4. Configuration of Auto-upload feature in mobile Apps

1. Go to App Store (iOS) or Play Store (Android). Search and install Nextcloud App.



2. Once installation process is complete, open the app and click on Log in. Type https://<your ip address>

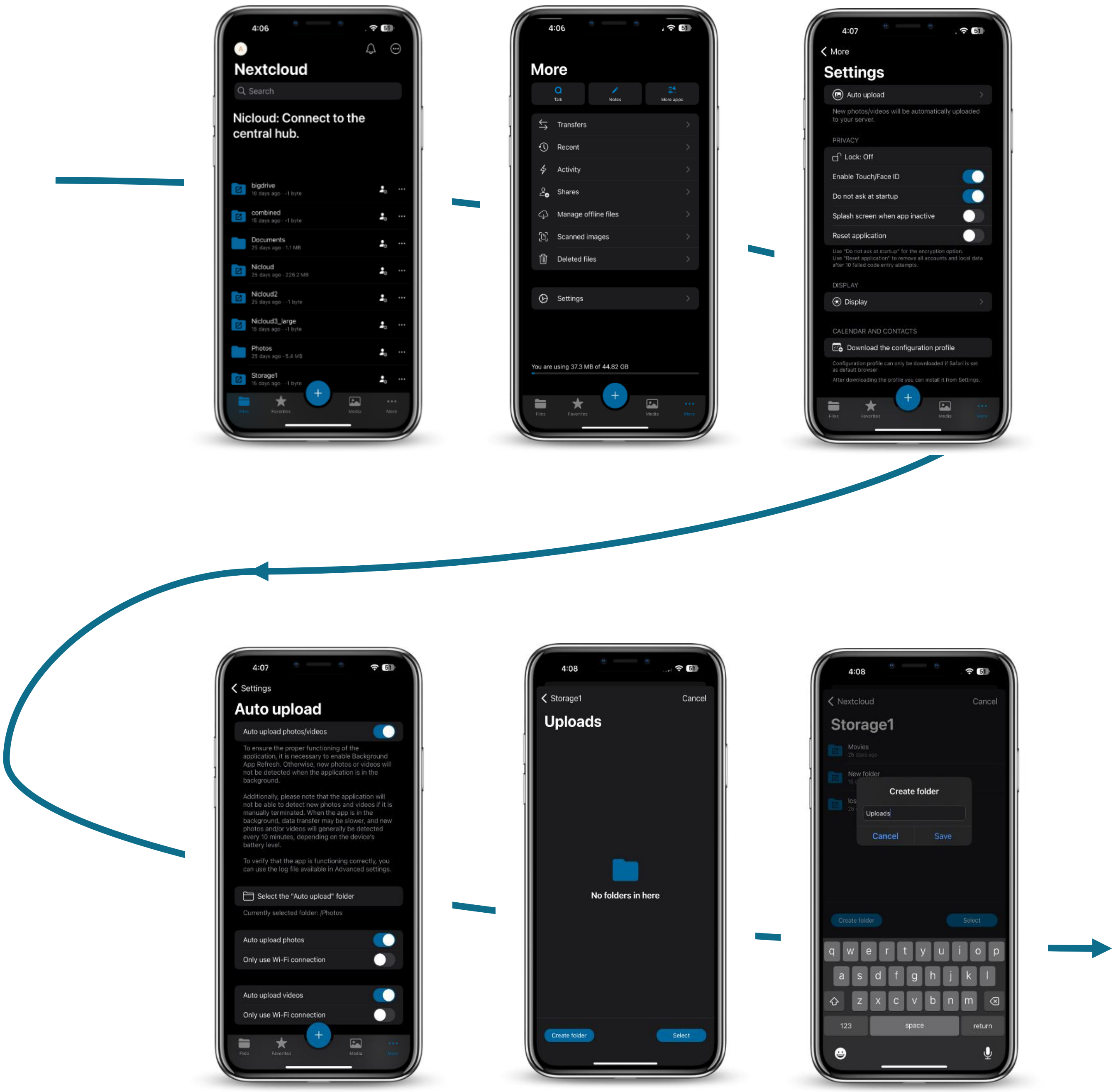


https://192.168.1.75

The link to your Nextcloud web interface when you open it in the browser.

3. User the username and password created in 18(i) and 18(ii) respectively to log into the Nextcloud server.
4. Once log in completed, Click on More (bottom nav bar)→Settings→Auto upload→toggle on the Auto upload photos/videos. Select the folder or create a new one where you want to save your auto uploaded files.

5. Toggle on Auto upload photos and Auto upload videos options to continue to automatically upload files.



New photos/videos will be automatically uploaded to your server.

Lock: Off

Enable Touch/Face ID: on

Do not ask at startup: on

Splash screen when app inactive: off

Reset application: off

Use "Do not ask at startup" for the encryption option. Use "Reset application" to remove all accounts and local data after 10 failed code entry attempts.

Configuration profile can only be downloaded if Safari is set as default browser

After downloading the profile you can install it from Settings.

Auto upload photos/videos: on

To ensure the proper functioning of the application, it is necessary to enable Background App Refresh. Otherwise, new photos or videos will not be detected when the application is in the background.

Additionally, please note that the application will not be able to detect new photos and videos if it is manually terminated. When the app is in the background, data transfer may be slower, and new photos and/or videos will generally be detected every 10 minutes, depending on the device's battery level.

To verify that the app is functioning correctly, you can use the log file available in Advanced settings.

Currently selected folder: /Photos

Auto upload photos: on

Only use Wi-Fi connection: off

Auto upload videos: on

Only use Wi-Fi connection: off