Raspberry Pi-Based Private Cloud Storage with Client-Side Encryption

By

NUR AIMAN ISKANDAR BIN NUR ISKANDAR

TP060914

APD2F2211CS(CYB)

A report submitted in partial fulfillment of the requirements for the degree of

B.Sc. (Hons) Computer Science (Cyber Security)

at Asia Pacific University of Technology and Innovation.

Supervised by Dr. Julia Juremi

2nd Marker: Ts. Umapathy Eaganathan

2024

Table of Contents

1. Install Docker on Raspberry Pi OS

Step 1:

Command: sudo apt-get update && sudo apt-get upgrade -y

Step 2:

curl -fsSL https://get.docker.com -o get-docker.sh

Command: sudo sh get-docker.sh

Step 3:

Add user to the Docker group, to avoid needing to use sudo with Docker for every command.

Command: sudo usermod -aG docker $USER

Step 4:

Ensure Docker starts automatically after a reboot.

Command: sudo systemctl enable docker

Step 5:

Reboot Raspberry Pi for the group changes to take effect.

Command: Sudo reboot

Step 6:

Verify Docker installation.

Command: docker --version

2. Install Docker Compose

Docker Compose is used to easily manage multi-container Docker applications (like Nextcloud, which may require multiple containers such as a database and the web app).

Step 1: Install docker compose.

Command: sudo apt install -y docker-compose

Step 2: Verify if Docker Compose is installed.

Command: docker-compose --version

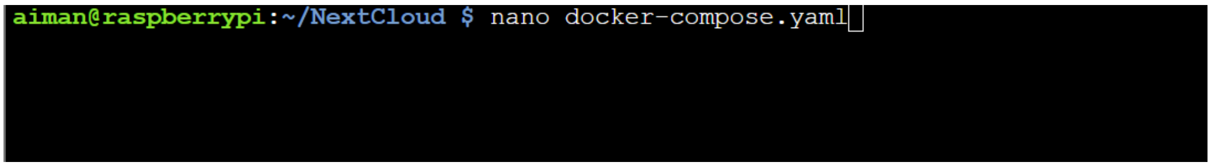
3. Set up Nextcloud with Docker Compose

Step 1: Create a new directory for Nextcloud.

Command: Mkdir NextCloud

Step 2: Create a docker-compose.yaml

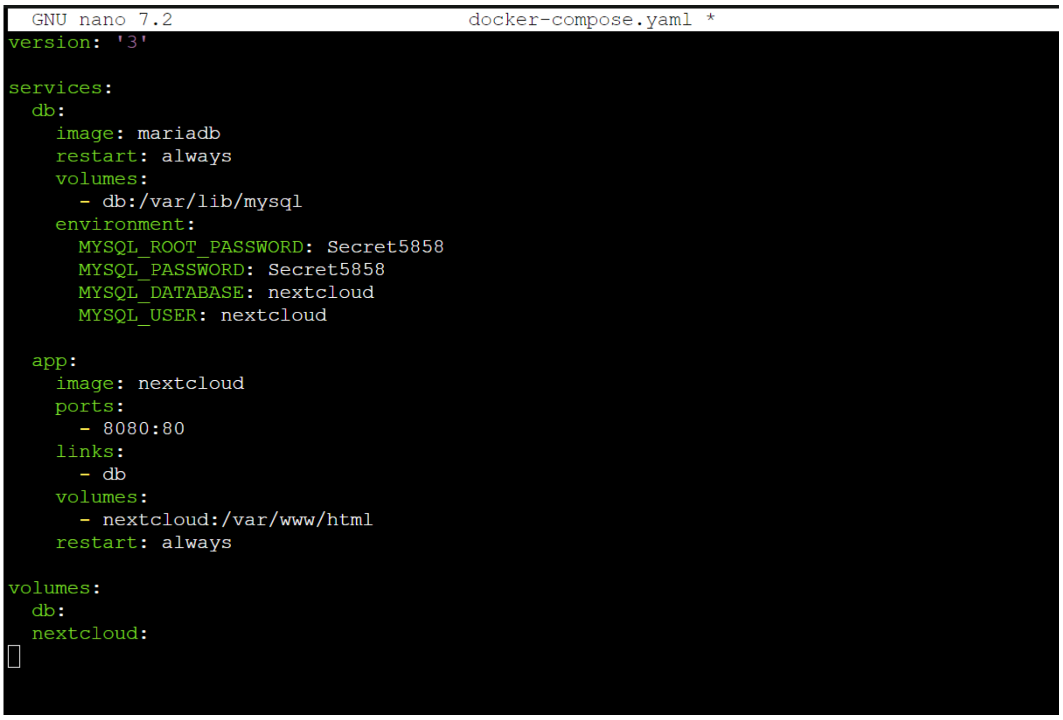
Create a new directory called NextCloud and create a docker-compose.yaml for multi container environment that run Nextcloud and MariaDB. Instead of running multiple docker run command for each service, docker-compose.yml run both services at the same time.



```
aiman@raspberrypi:~/NextCloud $ nano docker-compose.yaml
```

Step 3: Modify the file:

1. Change the environment MYSQL_ROOT_PASSWORD, MYSQL_PASSWORD, MYSQL_DATABASE, and MYSQL_USER, to secure credentials and your desired database name.
2. Adjust the volumes path under the Nextcloud service to point to the correct location where you want to store Nextcloud data.
3. Ensure the port mapping (- 8080:80) is correct for your setup and modify it if needed.

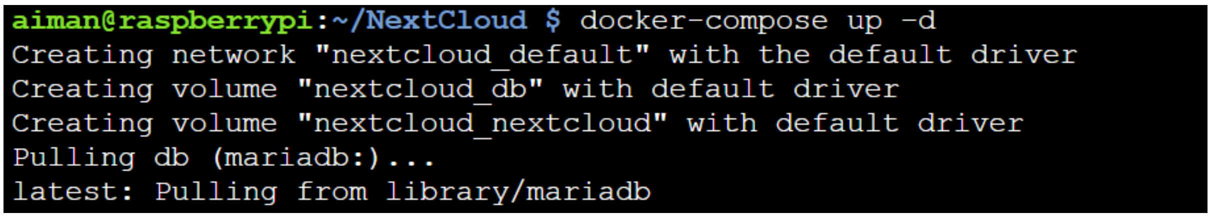


```
  GNU nano 7.2                                  docker-compose.yaml *
version: '3'

services:
  db:
    image: mariadb
    restart: always
    volumes:
      - db:/var/lib/mysql
    environment:
      MYSQL_ROOT_PASSWORD: Secret5858
      MYSQL_PASSWORD: Secret5858
      MYSQL_DATABASE: nextcloud
      MYSQL_USER: nextcloud

  app:
    image: nextcloud
    ports:
      - 8080:80
    links:
      - db
    volumes:
      - nextcloud:/var/www/html
    restart: always

volumes:
  db:
  nextcloud:
```

Step 4: Run Docker Compose.

Command: docker-compose up -d

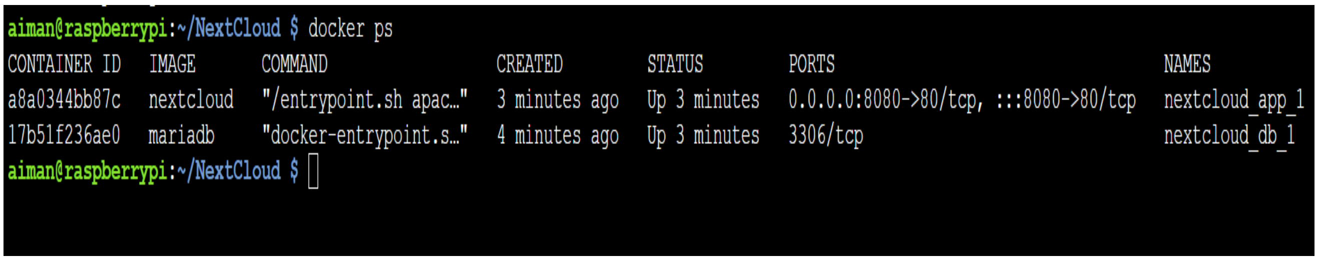
Run Docker Compose after configuring docker-compose.yaml, starting Nextcloud.



```
aiman@raspberrypi:~/NextCloud $ docker-compose up -d
Creating network "nextcloud_default" with the default driver
Creating volume "nextcloud_db" with default driver
Creating volume "nextcloud_nextcloud" with default driver
Pulling db (mariadb:)...
latest: Pulling from library/mariadb
```

Step 5: Verify if the containers are running.

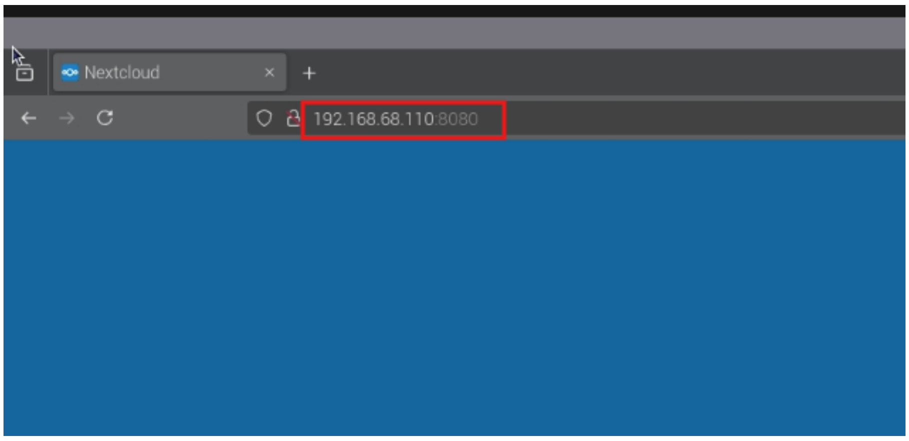
Command: docker ps



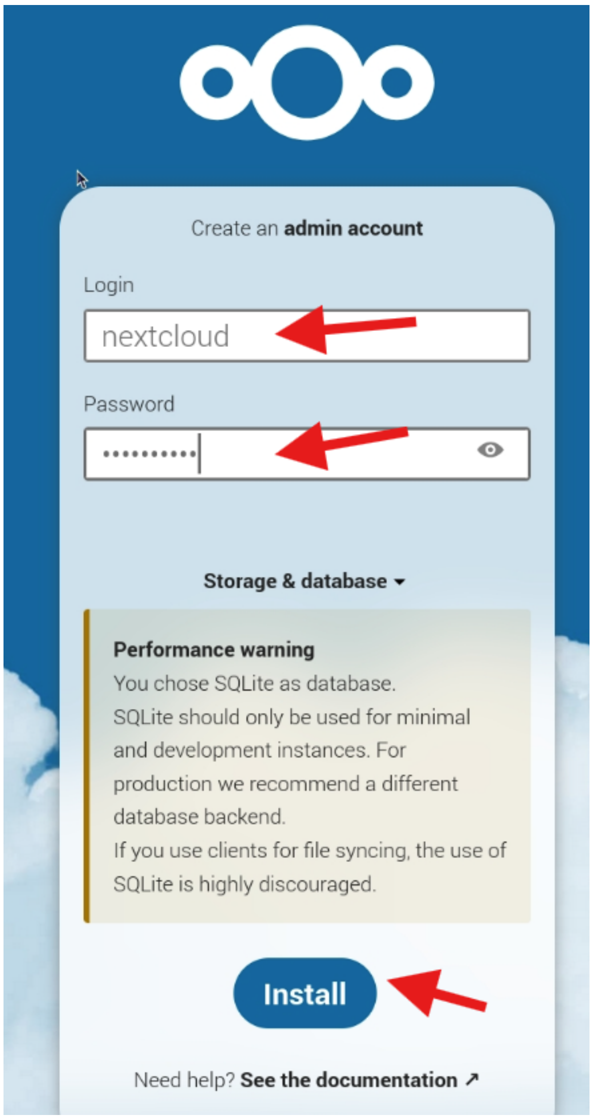
```
aiman@raspberrypi:~/NextCloud $ docker ps
CONTAINER ID   IMAGE       COMMAND                  CREATED         STATUS         PORTS                                   NAMES
a8a0344bb87c   nextcloud   "/entrypoint.sh apac…"   3 minutes ago   Up 3 minutes   0.0.0.0:8080->80/tcp, :::8080->80/tcp   nextcloud_app_1
17b51f236ae0   mariadb     "docker-entrypoint.s…"   4 minutes ago   Up 3 minutes   3306/tcp                                nextcloud_db_1
aiman@raspberrypi:~/NextCloud $ 
```

4. Access and install Nextcloud

Step 1: Open browser > enter local IP address followed by port number. In this case, it is 192.68.110:8080.



Step 2: Enter username and password created and click on install.


Create an **admin account**

Login

nextcloud

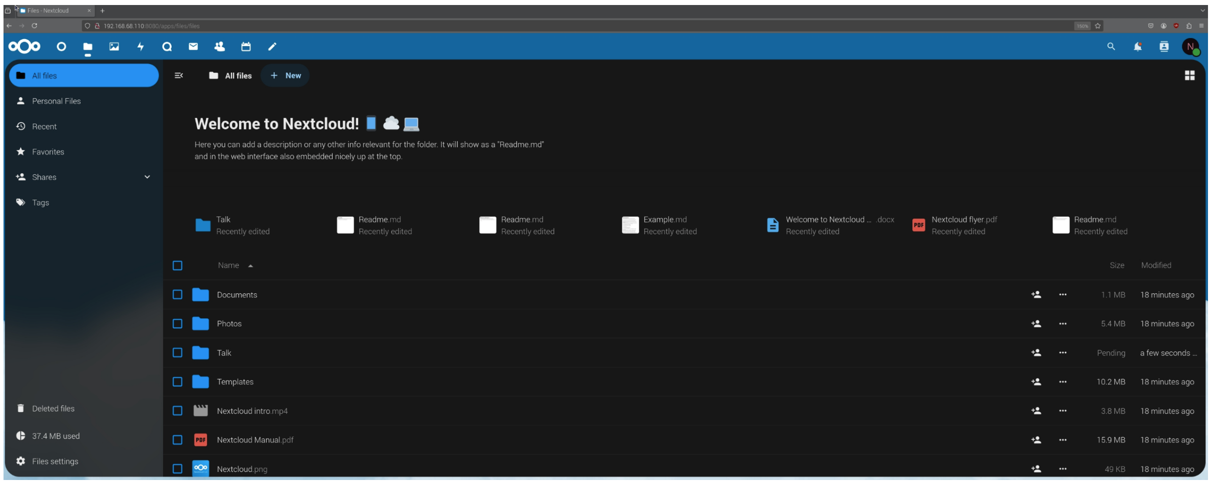
Password

••••••••••

Storage & database

Performance warning
You chose SQLite as database.
SQLite should only be used for minimal and development instances. For production we recommend a different database backend.
If you use clients for file syncing, the use of SQLite is highly discouraged.

Step 3: Once Nextcloud is installed, a welcome page will be displayed.



Welcome to Nextcloud! 📱 ☁️ 💻

Here you can add a description or any other info relevant for the folder. It will show as a "Readme.md" and in the web interface also embedded nicely up at the top.

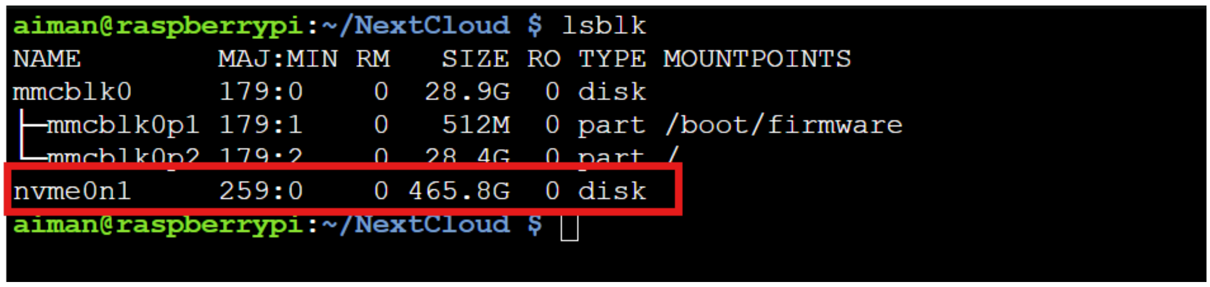
Talk	Recently edited
Readme.md	Recently edited
Readme.md	Recently edited
Example.md	Recently edited
Welcome to Nextcloud ... .docx	Recently edited
Nextcloud flyer.pdf	Recently edited
Readme.md	Recently edited

Name	Size	Modified
Documents	1.1 MB	18 minutes ago
Photos	5.4 MB	18 minutes ago
Talk	Pending	a few seconds ...
Templates	10.2 MB	18 minutes ago
Nextcloud intro.mp4	3.8 MB	18 minutes ago
Nextcloud Manual.pdf	15.9 MB	18 minutes ago
Nextcloud.png	49 KB	18 minutes ago

5. Set up NVMe Drive for NextCloud File Storage

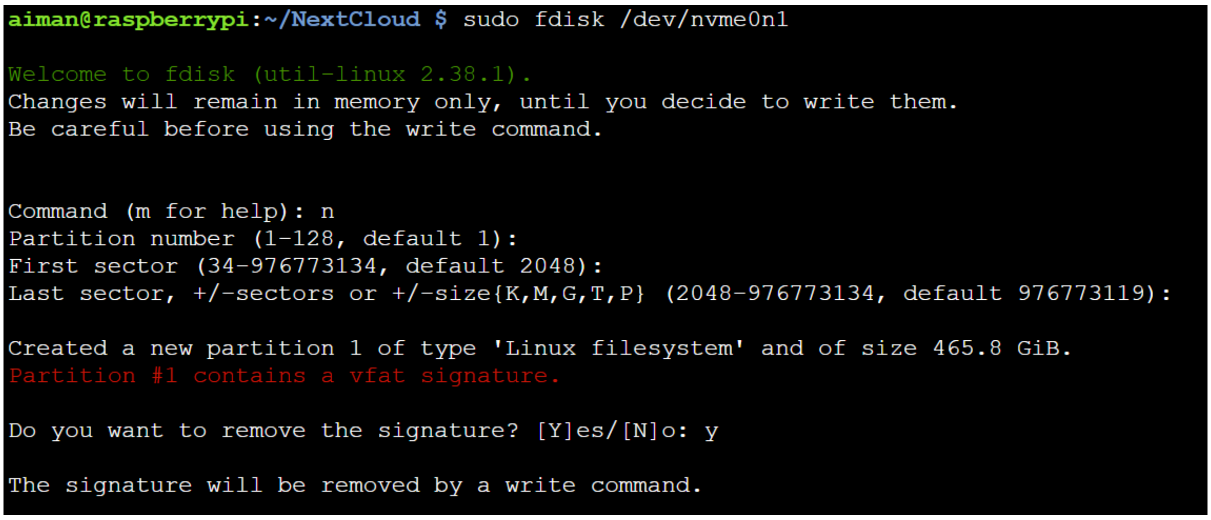
Step 1: Identify the NVMe Drive.

Command: lsblk



```
aiman@raspberrypi:~/NextCloud $ lsblk
NAME        MAJ:MIN RM   SIZE RO TYPE MOUNTPOINTS
mmcblk0     179:0    0  28.9G  0 disk
├─mmcblk0p1 179:1    0   512M  0 part /boot/firmware
└─mmcblk0p2 179:2    0  28.4G  0 part /
nvme0n1     259:0    0 465.8G  0 disk
aiman@raspberrypi:~/NextCloud $ 
```

Step 2: Enter "n" to create a new partition and click enter for the prompts.

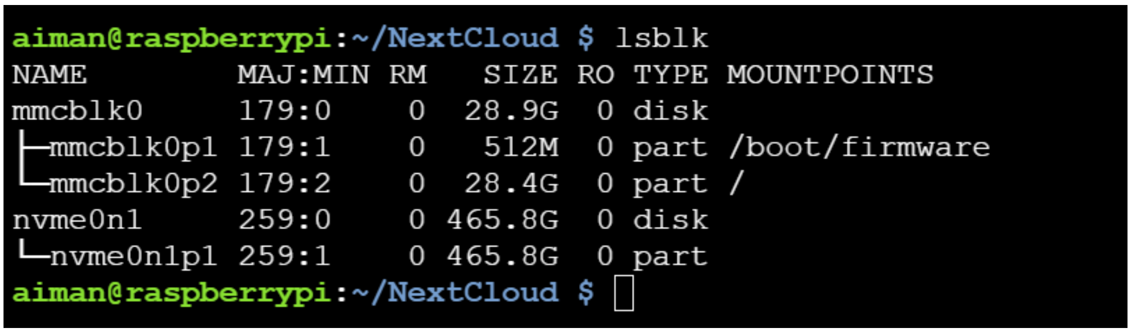


```
aiman@raspberrypi:~/NextCloud $ sudo fdisk /dev/nvme0n1

Welcome to fdisk (util-linux 2.38.1).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.


Command (m for help): n
Partition number (1-128, default 1):
First sector (34-976773134, default 2048):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (2048-976773134, default 976773119):

Created a new partition 1 of type 'Linux filesystem' and of size 465.8 GiB.
Partition #1 contains a vfat signature.

Do you want to remove the signature? [Y]es/[N]o: y

The signature will be removed by a write command.
```

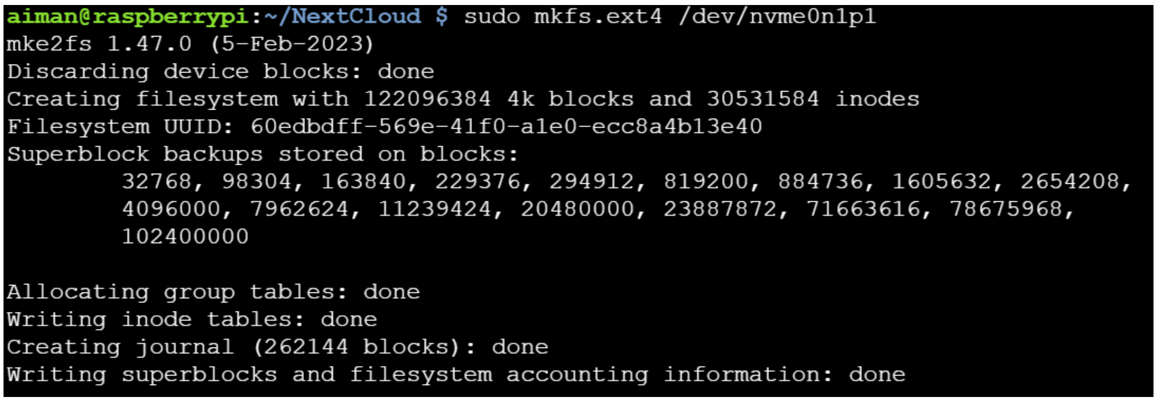
Step 3: Verify the partition and format.

Command: lsblk

Command: sudo mkfs.ext4 /dev/nvme0n1p1



```
aiman@raspberrypi:~/NextCloud $ lsblk
NAME          MAJ:MIN RM   SIZE RO TYPE MOUNTPOINTS
mmcblk0       179:0    0  28.9G  0 disk
├─mmcblk0p1   179:1    0   512M  0 part /boot/firmware
└─mmcblk0p2   179:2    0  28.4G  0 part /
nvme0n1       259:0    0 465.8G  0 disk
└─nvme0n1p1   259:1    0 465.8G  0 part
aiman@raspberrypi:~/NextCloud $
```



```
aiman@raspberrypi:~/NextCloud $ sudo mkfs.ext4 /dev/nvme0n1p1
mke2fs 1.47.0 (5-Feb-2023)
Discarding device blocks: done
Creating filesystem with 122096384 4k blocks and 30531584 inodes
Filesystem UUID: 60edbdff-569e-41f0-a1e0-ecc8a4b13e40
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
        4096000, 7962624, 11239424, 20480000, 23887872, 71663616, 78675968,
        102400000

Allocating group tables: done
Writing inode tables: done
Creating journal (262144 blocks): done
Writing superblocks and filesystem accounting information: done
```

Step 4: Create a mount point and mount the partition.

Create a directory to mount the NVMe Drive where NextCloud can store its data

Command: sudo mkdir /mnt/nvme

Command: sudo mount /dev/nvme0n1p1 /mnt/nvme

Step 5: Make the mount permanent.

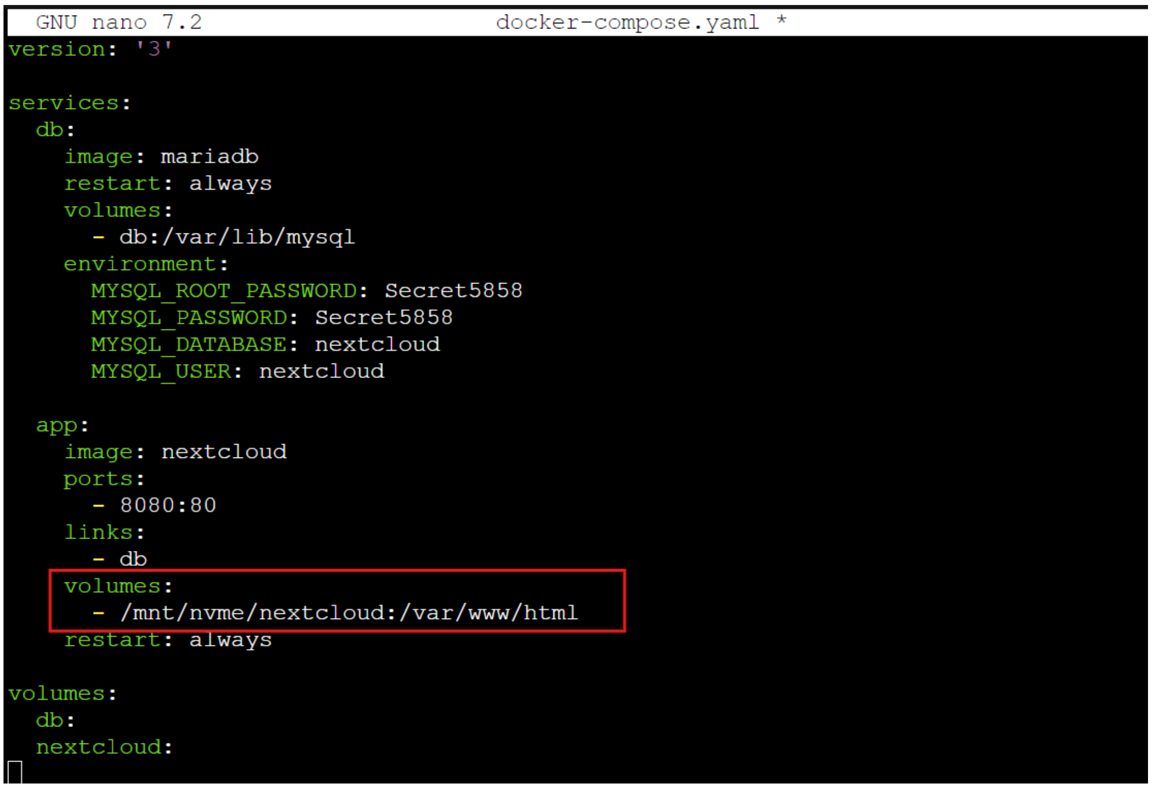
Ensure that the NVMe drive is automatically mounted on boot and add it to the fstab file

Command: sudo nano /etc/fstab

Command: /dev/nvme0n1p1 /mnt/nvme ext4 defaults 0 2

Step 6: Update docker-compose.yml

Update the volume section for NextCloud container to point the mounted NVMe drive



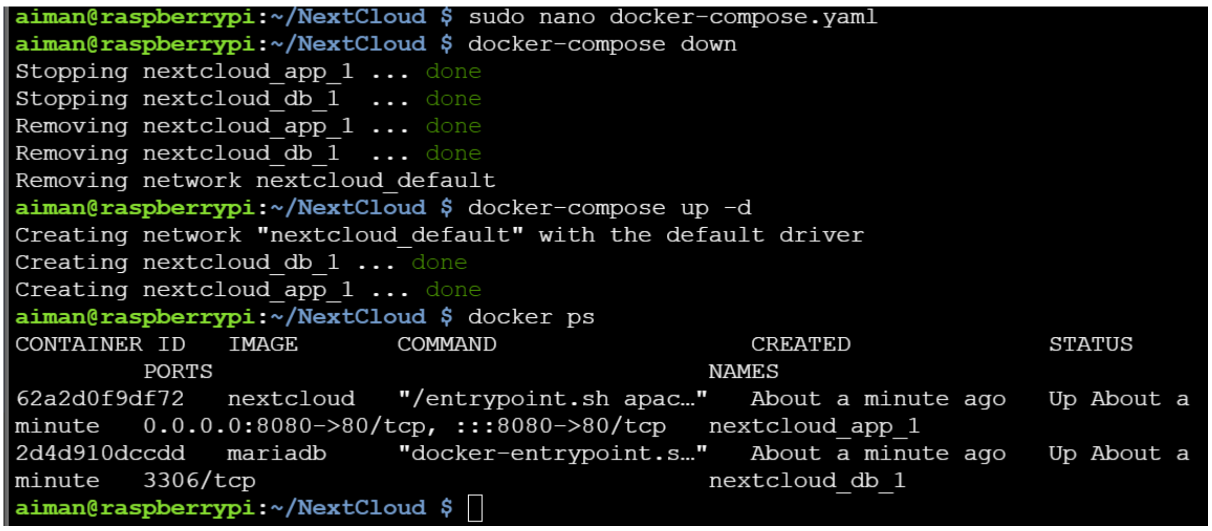
```
  GNU nano 7.2                         docker-compose.yaml *
version: '3'

services:
  db:
    image: mariadb
    restart: always
    volumes:
      - db:/var/lib/mysql
    environment:
      MYSQL_ROOT_PASSWORD: Secret5858
      MYSQL_PASSWORD: Secret5858
      MYSQL_DATABASE: nextcloud
      MYSQL_USER: nextcloud

  app:
    image: nextcloud
    ports:
      - 8080:80
    links:
      - db
    volumes:
      - /mnt/nvme/nextcloud:/var/www/html
    restart: always

volumes:
  db:
  nextcloud:
```

Step 7: Restart NextCloud and MariaDB containers.

Command: docker-compose down

Command: docker-compose up -d

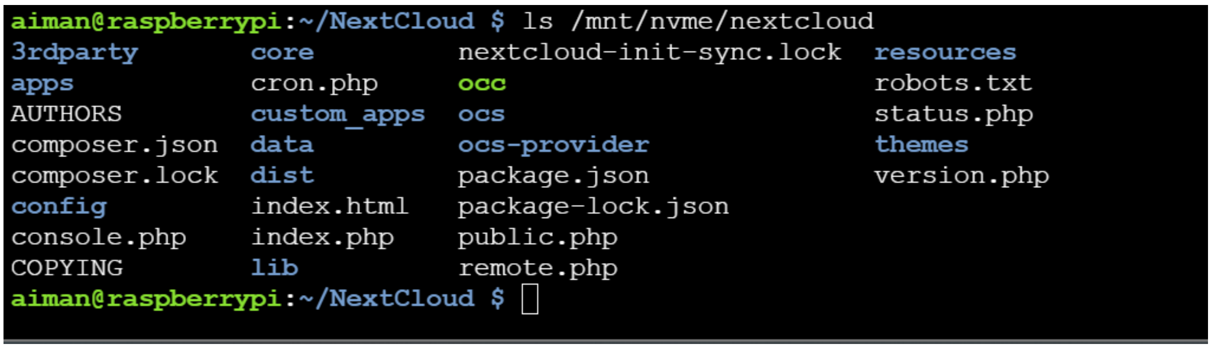
Command: docker ps



```
aiman@raspberrypi:~/NextCloud $ sudo nano docker-compose.yaml
aiman@raspberrypi:~/NextCloud $ docker-compose down
Stopping nextcloud_app_1 ... done
Stopping nextcloud_db_1  ... done
Removing nextcloud_app_1 ... done
Removing nextcloud_db_1  ... done
Removing network nextcloud_default
aiman@raspberrypi:~/NextCloud $ docker-compose up -d
Creating network "nextcloud_default" with the default driver
Creating nextcloud_db_1 ... done
Creating nextcloud_app_1 ... done
aiman@raspberrypi:~/NextCloud $ docker ps
CONTAINER ID   IMAGE       COMMAND                  CREATED              STATUS
         PORTS                                NAMES
62a2d0f9df72   nextcloud   "/entrypoint.sh apac…"   About a minute ago   Up About a
minute   0.0.0.0:8080->80/tcp, :::8080->80/tcp   nextcloud_app_1
2d4d910dccdd   mariadb     "docker-entrypoint.s…"   About a minute ago   Up About a
minute   3306/tcp                             nextcloud_db_1
aiman@raspberrypi:~/NextCloud $ 
```

Step 8: Verify that NextCloud is using the NVMe drive.

Command: ls /mnt/nvme/nextcloud

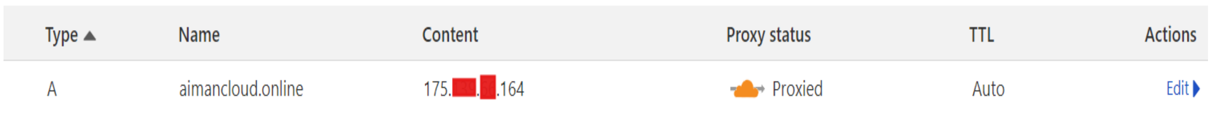


```
aiman@raspberrypi:~/NextCloud $ ls /mnt/nvme/nextcloud
3rdparty       core          nextcloud-init-sync.lock  resources
apps           cron.php      occ                       robots.txt
AUTHORS        custom_apps   ocs                       status.php
composer.json  data          ocs-provider              themes
composer.lock  dist          package.json              version.php
config         index.html    package-lock.json
console.php    index.php     public.php
COPYING        lib           remote.php
aiman@raspberrypi:~/NextCloud $ 
```

6. Configuring DNS Records

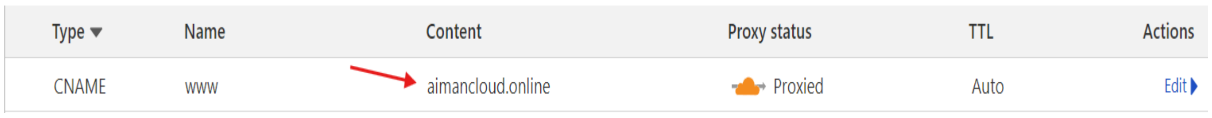
Before setting up DNS redirection, a domain is bought. In this project it's called aimancloud.online. The nameservers were then redirected to take advantage of the security and performance features such as DDoS protection, SSL certificate management, and global CDN caching.

Step 1: Fill up the A information such as domain name, Public Ip address, and leave TTL to default setting.



Type ▲	Name	Content	Proxy status	TTL	Actions
A	aimancloud.online	175.[redacted].[redacted].164	Proxied	Auto	Edit ▸

Step 2: (Optional) Set Up a **www** Subdomain (if you want users to access your Nextcloud via www.aimancloud.online.



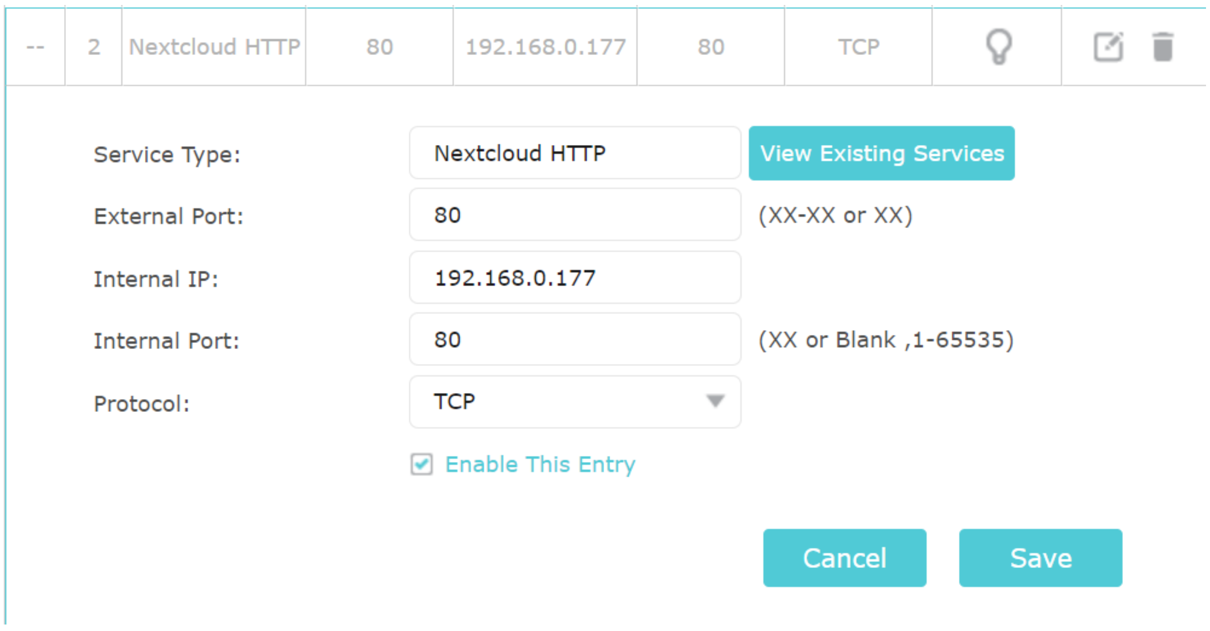
Type ▼	Name	Content	Proxy status	TTL	Actions
CNAME	www	aimancloud.online	Proxied	Auto	Edit ▸

7. Configure and set up port forwarding

Step 1: Log into home router's admin panel.

Step 2: Navigate to the Port Forwarding section > Virtual Servers.

Step 3: Set up port forwarding for HTTP (Port 80). In this case, the IP address will be configured to Raspberry Pi IP address.

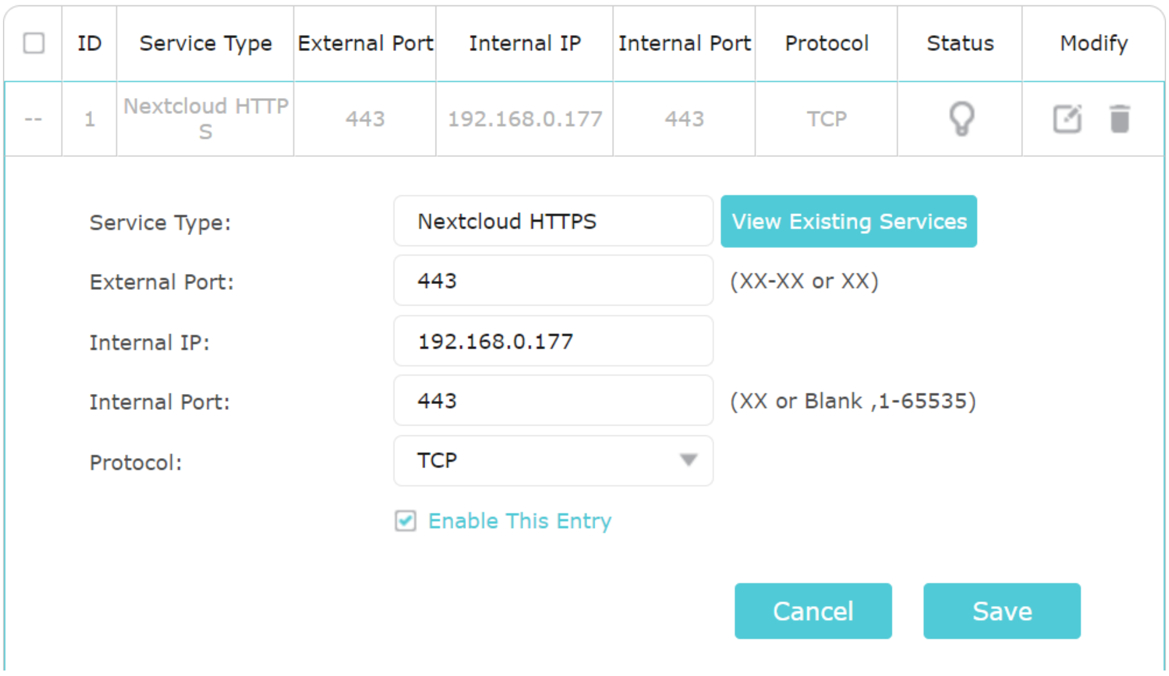


--	2	Nextcloud HTTP	80	192.168.0.177	80	TCP

Service Type:	Nextcloud HTTP
External Port:	80 (XX-XX or XX)
Internal IP:	192.168.0.177
Internal Port:	80 (XX or Blank ,1-65535)
Protocol:	TCP

☑ Enable This Entry

Step 4: Set up port forwarding for HTTPS (Port 443), In this case, the IP address will be configured to Raspberry Pi IP address.

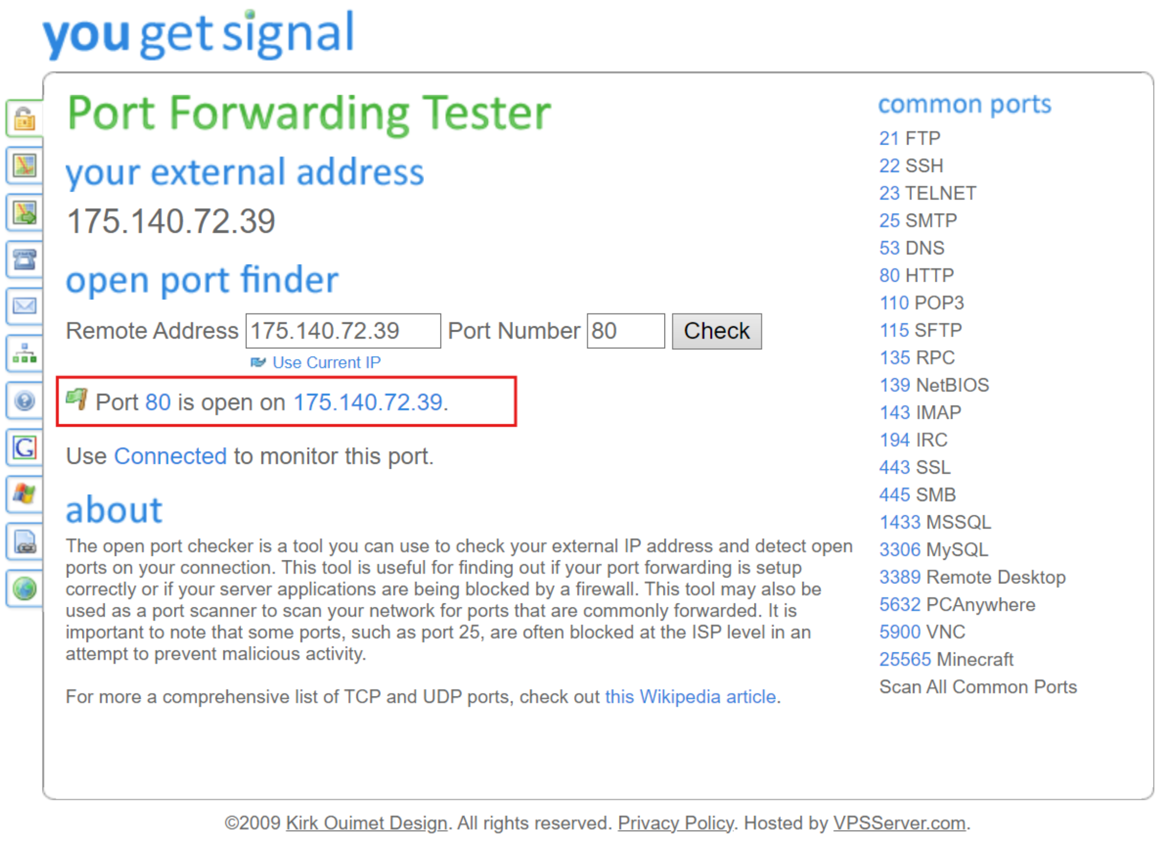


☐	ID	Service Type	External Port	Internal IP	Internal Port	Protocol	Status	Modify
--	1	Nextcloud HTTPS	443	192.168.0.177	443	TCP		

Service Type:	Nextcloud HTTPS	
External Port:	443	(XX-XX or XX)
Internal IP:	192.168.0.177	
Internal Port:	443	(XX or Blank ,1-65535)
Protocol:	TCP	

☑ Enable This Entry

Step 5: Verify the port is 80 (HTTP) is open using port forwarding website such as yougetsignal.


you get signal

Port Forwarding Tester

your external address

175.140.72.39

open port finder

Remote Address 175.140.72.39 Port Number 80 Check

Use Current IP

Port 80 is open on 175.140.72.39.

Use Connected to monitor this port.

about

The open port checker is a tool you can use to check your external IP address and detect open ports on your connection. This tool is useful for finding out if your port forwarding is setup correctly or if your server applications are being blocked by a firewall. This tool may also be used as a port scanner to scan your network for ports that are commonly forwarded. It is important to note that some ports, such as port 25, are often blocked at the ISP level in an attempt to prevent malicious activity.

For more a comprehensive list of TCP and UDP ports, check out this Wikipedia article.

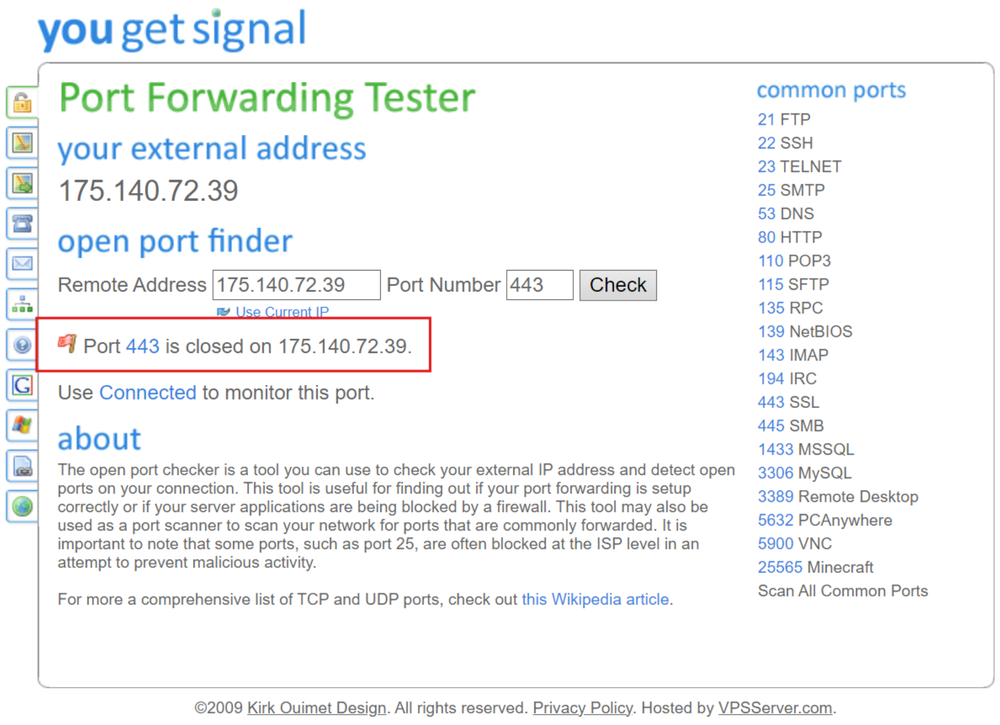
common ports

21	FTP
22	SSH
23	TELNET
25	SMTP
53	DNS
80	HTTP
110	POP3
115	SFTP
135	RPC
139	NetBIOS
143	IMAP
194	IRC
443	SSL
445	SMB
1433	MSSQL
3306	MySQL
3389	Remote Desktop
5632	PCAnywhere
5900	VNC
25565	Minecraft

Scan All Common Ports

©2009 Kirk Ouimet Design. All rights reserved. Privacy Policy. Hosted by VPSServer.com.

Step 6: Verify that port 443 (HTTPS) is closed as the SSL certificate hasn't been configured to the domain.



you get signal

Port Forwarding Tester

your external address

175.140.72.39

open port finder

Remote Address 175.140.72.39 Port Number 443 Check

Use Current IP

Port 443 is closed on 175.140.72.39.

Use Connected to monitor this port.

about

The open port checker is a tool you can use to check your external IP address and detect open ports on your connection. This tool is useful for finding out if your port forwarding is setup correctly or if your server applications are being blocked by a firewall. This tool may also be used as a port scanner to scan your network for ports that are commonly forwarded. It is important to note that some ports, such as port 25, are often blocked at the ISP level in an attempt to prevent malicious activity.

For more a comprehensive list of TCP and UDP ports, check out this Wikipedia article.

common ports

21 FTP
22 SSH
23 TELNET
25 SMTP
53 DNS
80 HTTP
110 POP3
115 SFTP
135 RPC
139 NetBIOS
143 IMAP
194 IRC
443 SSL
445 SMB
1433 MSSQL
3306 MySQL
3389 Remote Desktop
5632 PCAnywhere
5900 VNC
25565 Minecraft
Scan All Common Ports

©2009 Kirk Ouimet Design. All rights reserved. Privacy Policy. Hosted by VPSServer.com.

8. Configure Nginx for Nextcloud

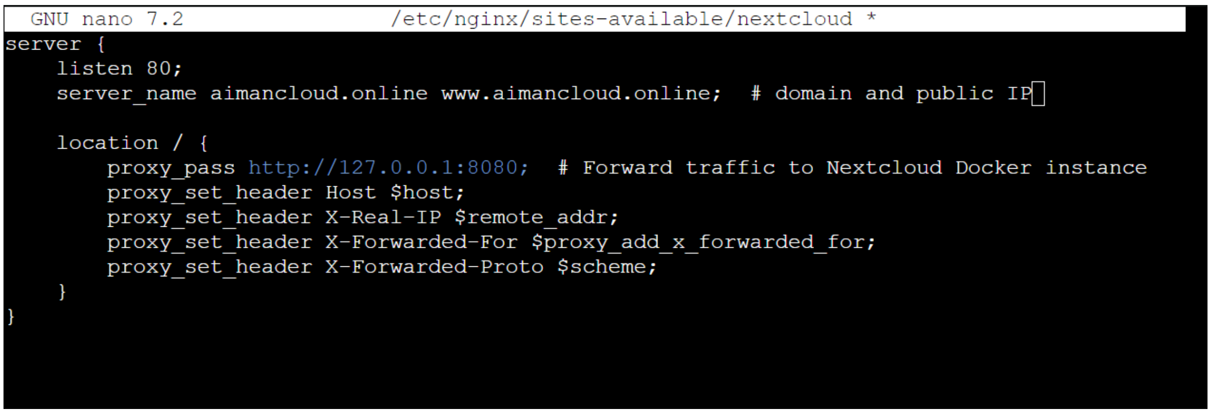
Step 1: Install Nginx.

Command: sudo apt install nginx -y

Step 2: Create a new Nginx configuration for Nextcloud and configure.

This configuration file is for Nginx to handle web traffic directed to the domain (or public IP) and pass that traffic to the Nextcloud instances running inside the Docker container

Command: sudo nano /etc/nginx/sites-available/nextcloud

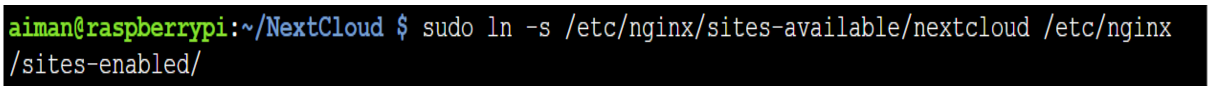


```
  GNU nano 7.2                    /etc/nginx/sites-available/nextcloud *
server {
    listen 80;
    server_name aimancloud.online www.aimancloud.online;  # domain and public IP

    location / {
        proxy_pass http://127.0.0.1:8080;  # Forward traffic to Nextcloud Docker instance
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
    }
}
```

Step 3: Create a symbolic link for configuration

Nginx requires enabled site configurations to be stored in /etc/nginx/sites-enabled/. Instead of copying the file, a symbolic link (shortcut) is created from the file in sites-available to sites-enabled.

Command: sudo ln -s /etc/nginx/sites-available/nextcloud /etc/nginx/sites-enabled/

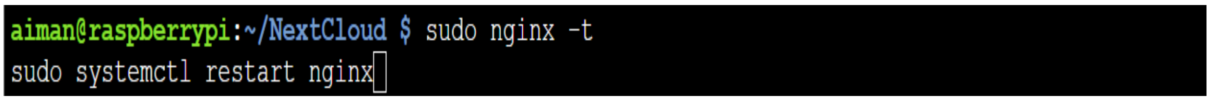


```
aiman@raspberrypi:~/NextCloud $ sudo ln -s /etc/nginx/sites-available/nextcloud /etc/nginx
/sites-enabled/
```

Step 4: Test nginx configuration and restart Nginx

Command: sudo nginx -t

Command: sudo systemctl restart nginx

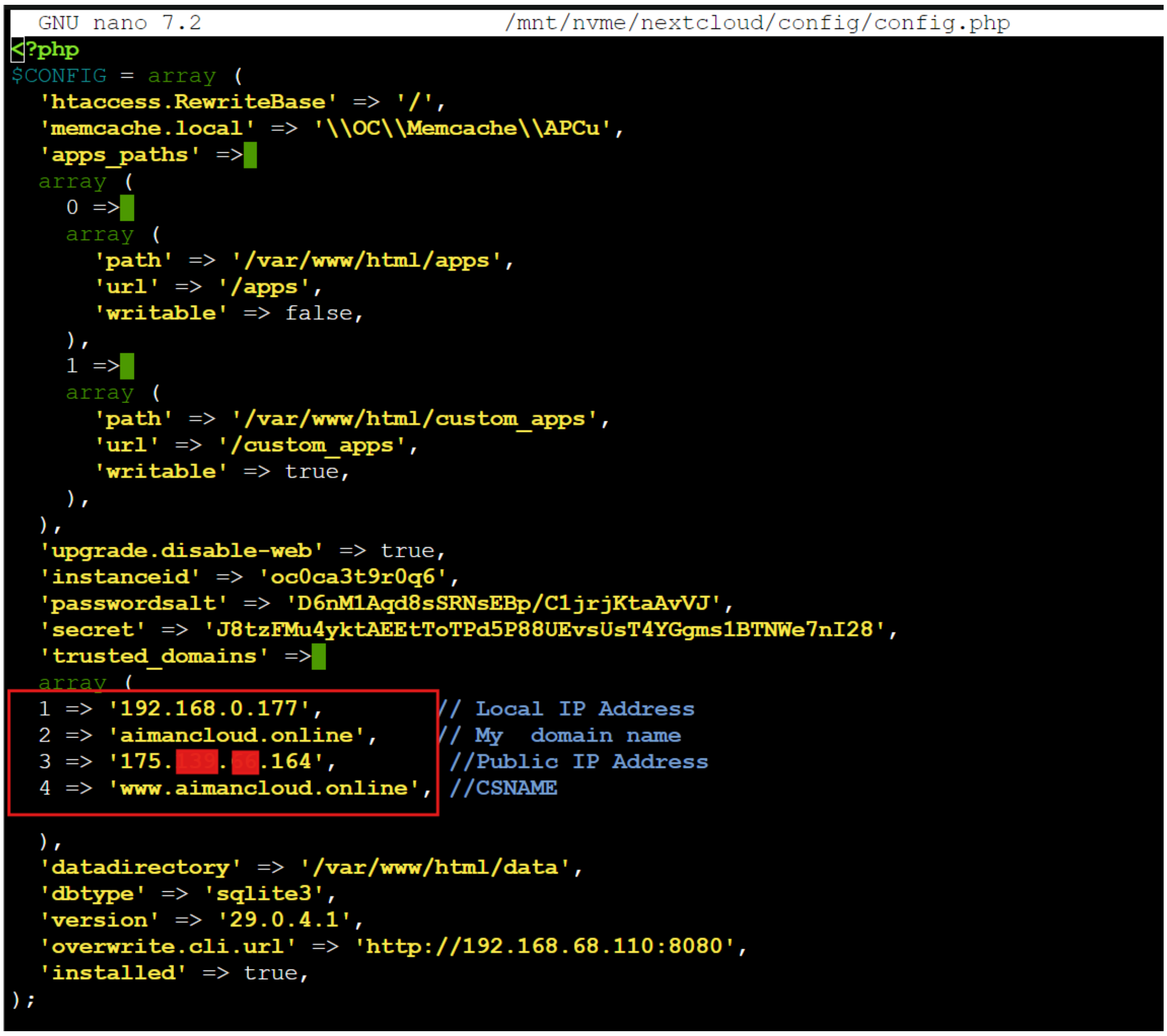


```
aiman@raspberrypi:~/NextCloud $ sudo nginx -t
sudo systemctl restart nginx
```

9. Set up Nextcloud Trusted Domains

Add the domains that will access your Nextcloud instance to the config.php file under the trusted domains section. This ensures that Nextcloud allows access only from these specified domains.

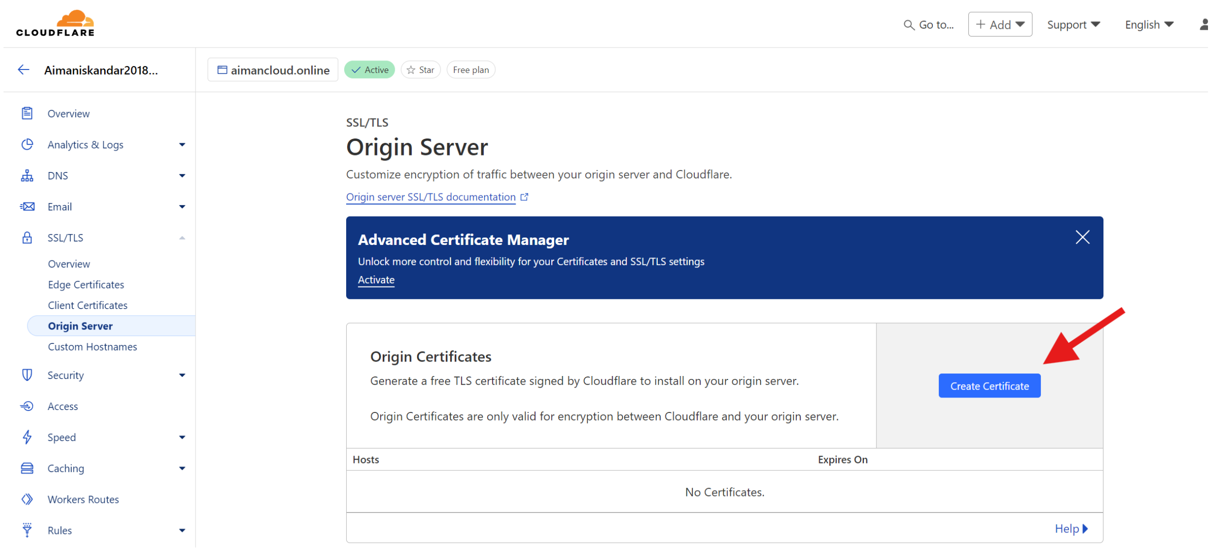
Command: sudo nano /mnt/nvme/nextcloud/config/config.php



```
  GNU nano 7.2                          /mnt/nvme/nextcloud/config/config.php
<?php
$CONFIG = array (
  'htaccess.RewriteBase' => '/',
  'memcache.local' => '\\OC\\Memcache\\APCu',
  'apps_paths' =>
  array (
    0 =>
    array (
      'path' => '/var/www/html/apps',
      'url' => '/apps',
      'writable' => false,
    ),
    1 =>
    array (
      'path' => '/var/www/html/custom_apps',
      'url' => '/custom_apps',
      'writable' => true,
    ),
  ),
  'upgrade.disable-web' => true,
  'instanceid' => 'oc0ca3t9r0q6',
  'passwordsalt' => 'D6nM1Aqd8sSRNsEBp/C1jrjKtaAvVJ',
  'secret' => 'J8tzFMu4yktAEEtToTPd5P88UEvsUsT4YGgms1BTNWe7nI28',
  'trusted_domains' =>
  array (
  1 => '192.168.0.177',                // Local IP Address
  2 => 'aimancloud.online',            // My  domain name
  3 => '175.[redacted].[redacted].164',                 //Public IP Address
  4 => 'www.aimancloud.online',         //CSNAME

  ),
  'datadirectory' => '/var/www/html/data',
  'dbtype' => 'sqlite3',
  'version' => '29.0.4.1',
  'overwrite.cli.url' => 'http://192.168.68.110:8080',
  'installed' => true,
);
```

10. Obtain SSL certificate and install SSL Certificate

Step 1: Login into Cloudflare > Select domain name > Origin Server



SSL/TLS

Origin Server

Customize encryption of traffic between your origin server and Cloudflare.

Origin server SSL/TLS documentation

Advanced Certificate Manager

Unlock more control and flexibility for your Certificates and SSL/TLS settings

Origin Certificates

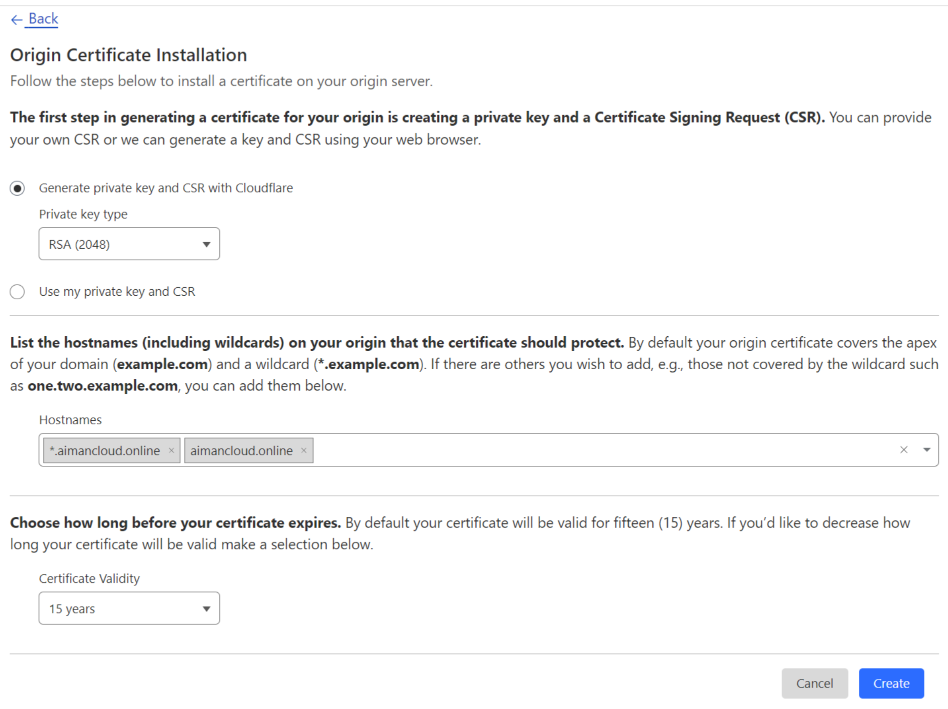
Generate a free TLS certificate signed by Cloudflare to install on your origin server.

Origin Certificates are only valid for encryption between Cloudflare and your origin server.

Hosts	Expires On
No Certificates.	

Step 2: Create certificate and generate certificate

1. Select RSA (2048)
2. Choose the certificate validity, in this case 15 years.
3. Click next to generate the certificate



← Back

Origin Certificate Installation

Follow the steps below to install a certificate on your origin server.

The first step in generating a certificate for your origin is creating a private key and a Certificate Signing Request (CSR). You can provide your own CSR or we can generate a key and CSR using your web browser.

[✓ Generate private key and CSR with Cloudflare]

Private key type: RSA (2048)

Use my private key and CSR

List the hostnames (including wildcards) on your origin that the certificate should protect. By default your origin certificate covers the apex of your domain (**example.com**) and a wildcard (***.example.com**). If there are others you wish to add, e.g., those not covered by the wildcard such as **one.two.example.com**, you can add them below.

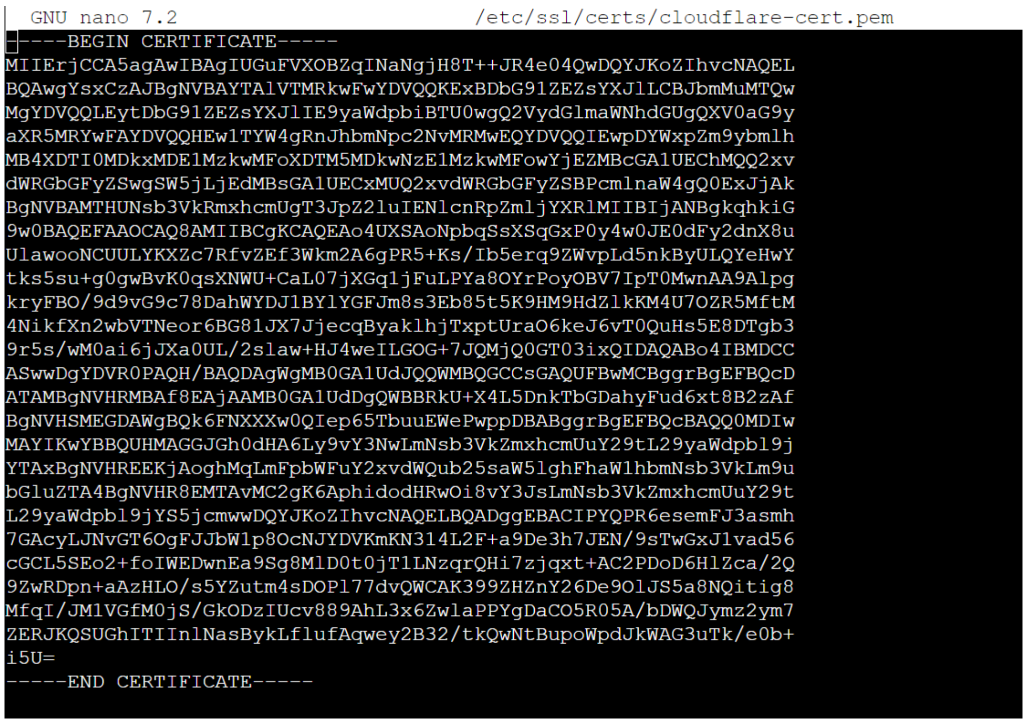
Hostnames: *.aimancloud.online, aimancloud.online

Choose how long before your certificate expires. By default your certificate will be valid for fifteen (15) years. If you'd like to decrease how long your certificate will be valid make a selection below.

Certificate Validity: 15 years

Step 3: Create a folder and file to store the origins certificate.

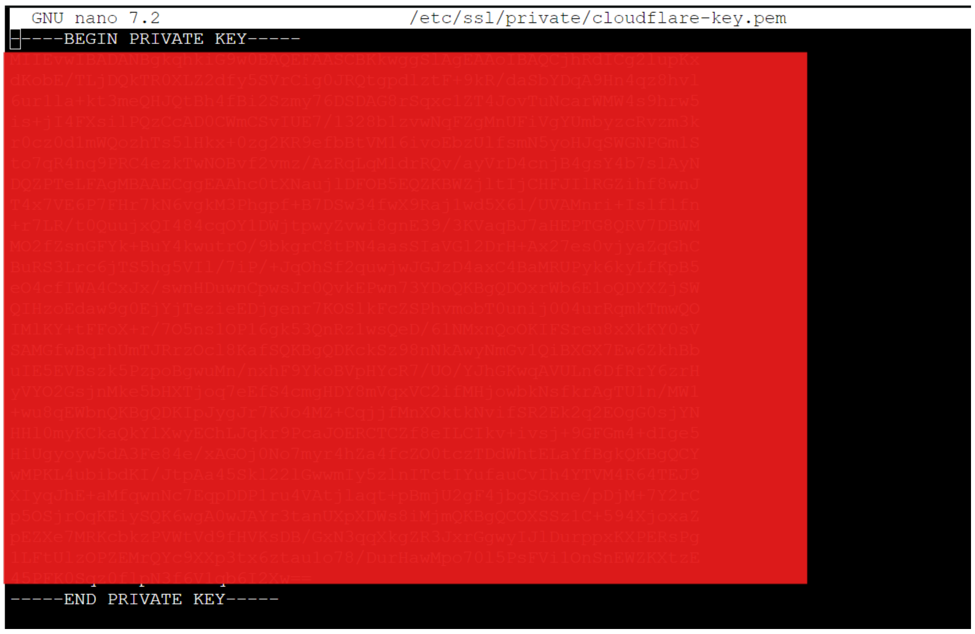
Command: sudo nano /etc/ssl/certs/cloudflare-cert.pem



```
  GNU nano 7.2                    /etc/ssl/certs/cloudflare-cert.pem
-----BEGIN CERTIFICATE-----
MIIErjCCA5agAwIBAgIUGuFVXOBZqINaNgjH8T++JR4e04QwDQYJKoZIhvcNAQEL
BQAwgYsxCzAJBgNVBAYTAlVTMRkwFwYDVQQKExBDbG91ZEZsYXJlLCBJbmMuMTQw
MgYDVQQLEytDbG91ZEZsYXJlIE9yaWdpbiBTU0wgQ2VydGlmaWNhdGUgQXV0aG9y
aXR5MRYwFAYDVQQHEw1TYW4gRnJhbmNpc2NvMRMwEQYDVQQIEwpDYWxpZm9ybmlh
MB4XDTI0MDkxMDE1MzkwMFoXDTM5MDkwNzE1MzkwMFowYjEZMBcGA1UEChMQQ2xv
dWRGbGFyZSwgSW5jLjEdMBsGA1UECxMUQ2xvdWRGbGFyZSBPcmlnaW4gQ0ExJjAk
BgNVBAMTHUNsb3VkRmxhcmUgT3JpZ2luIENlcnRpZmljYXRlMIIBIjANBgkqhkiG
9w0BAQEFAAOCAQ8AMIIBCgKCAQEAo4UXSAoNpbqSsXSqGxP0y4w0JE0dFy2dnX8u
UlawooNCUULYKXZc7RfvZEf3Wkm2A6gPR5+Ks/Ib5erq9ZWvpLd5nkByULQYeHwY
tks5su+g0gwBvK0qsXNWU+CaL07jXGq1jFuLPYa8OYrPoyOBV7IpT0MwnAA9Alpg
kryFBO/9d9vG9c78DahWYDJ1BYlYGFJm8s3Eb85t5K9HM9HdZlkKM4U7OZR5MftM
4NikfXn2wbVTNeor6BG81JX7JjecqByaklhjTxptUraO6keJ6vT0QuHs5E8DTgb3
9r5s/wM0ai6jJXa0UL/2slaw+HJ4weILGOG+7JQMjQ0GT03ixQIDAQABo4IBMDCC
ASwwDgYDVR0PAQH/BAQDAgWgMB0GA1UdJQQWMBQGCCsGAQUFBwMCBggrBgEFBQcD
ATAMBgNVHRMBAf8EAjAAMB0GA1UdDgQWBBRkU+X4L5DnkTbGDahyFud6xt8B2zAf
BgNVHSMEGDAWgBQk6FNXXXw0QIep65TbuuEWePwppDBABggrBgEFBQcBAQQ0MDIw
MAYIKwYBBQUHMAGGJGh0dHA6Ly9vY3NwLmNsb3VkZmxhcmUuY29tL29yaWdpbl9j
YTAxBgNVHREEKjAoghMqLmFpbWFuY2xvdWQub25saW5lghFhaW1hbmNsb3VkLm9u
bGluZTA4BgNVHR8EMTAvMC2gK6AphidodHRwOi8vY3JsLmNsb3VkZmxhcmUuY29t
L29yaWdpbl9jYS5jcmwwDQYJKoZIhvcNAQELBQADggEBACIPYQPR6esemFJ3asmh
7GAcyLJNvGT6OgFJJbW1p8OcNJYDVKmKN314L2F+a9De3h7JEN/9sTwGxJ1vad56
cGCL5SEo2+foIWEDwnEa9Sg8MlD0t0jT1LNzqrQHi7zjqxt+AC2PDoD6HlZca/2Q
9ZwRDpn+aAzHLO/s5YZutm4sDOPl77dvQWCAK399ZHZnY26De9OlJS5a8NQitig8
MfqI/JM1VGfM0jS/GkODzIUcv889AhL3x6ZwlaPPYgDaCO5R05A/bDWQJymz2ym7
ZERJKQSUGhITIInlNasBykLflufAqwey2B32/tkQwNtBupoWpdJkWAG3uTk/e0b+
i5U=
-----END CERTIFICATE-----
```

Step 4: Create a folder and file to store private key.

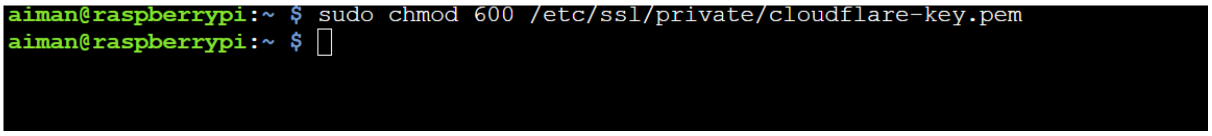
Command: sudo nano /etc/ssl/private/cloudflare-key.pem



```
  GNU nano 7.2                        /etc/ssl/private/cloudflare-key.pem
-----BEGIN PRIVATE KEY-----
[redacted]
-----END PRIVATE KEY-----
```

Step 5: Secure the Private Key to make sure only root user can read the content.

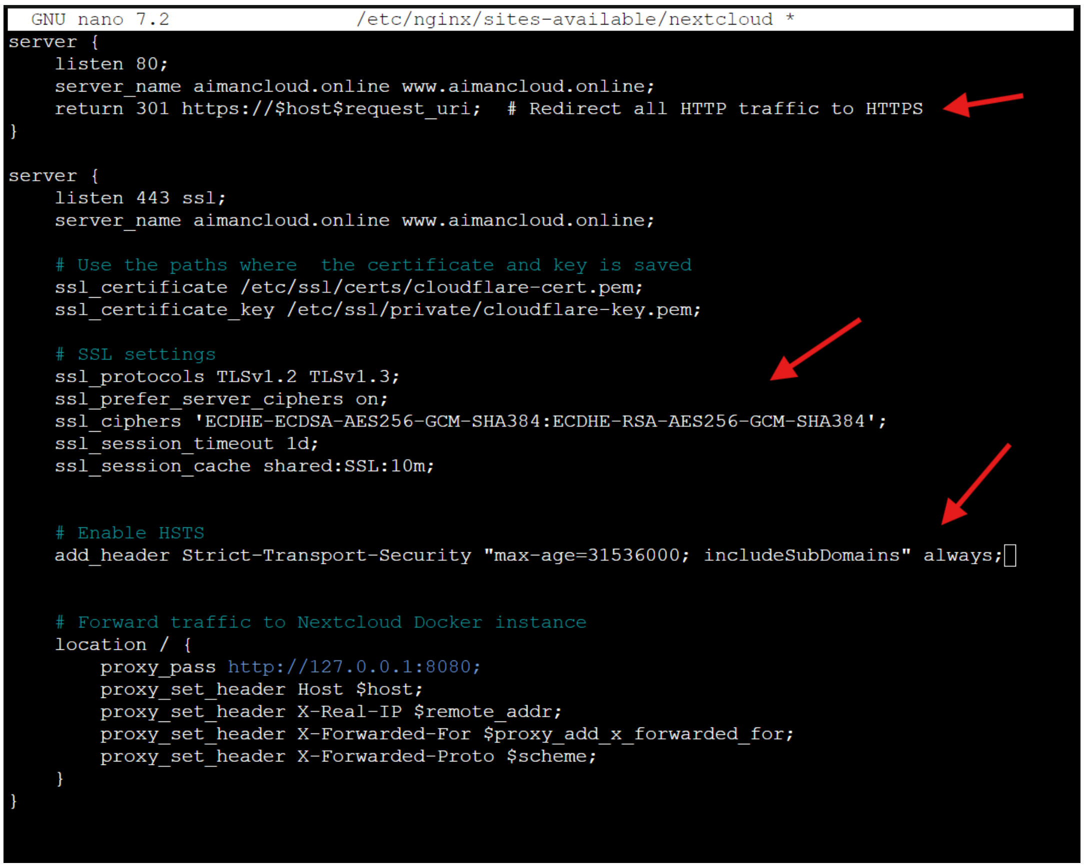
Command: sudo chmod 600 /etc/ssl/private/cloudflare-key.pem



```
aiman@raspberrypi:~ $ sudo chmod 600 /etc/ssl/private/cloudflare-key.pem
aiman@raspberrypi:~ $
```

Step 6: Modify Nginx configuration to utilize the SSL certificate.

1. Add redirection from HTTP traffic to HTTPS
2. Add SSL settings
3. Add HSTS to prevents attack such as protocol downgrade attacks, cookie hijacking and MITM

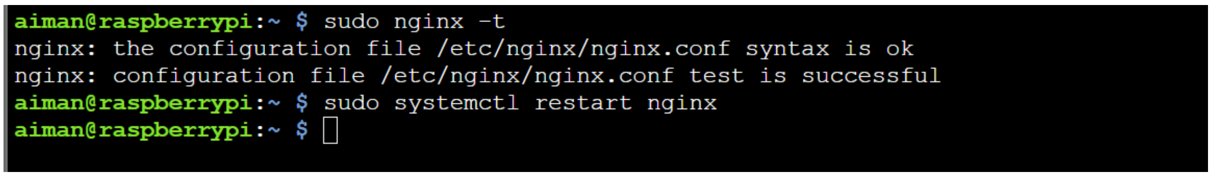


```
  GNU nano 7.2                    /etc/nginx/sites-available/nextcloud *
server {
    listen 80;
    server_name aimancloud.online www.aimancloud.online;
    return 301 https://$host$request_uri;  # Redirect all HTTP traffic to HTTPS
}

server {
    listen 443 ssl;
    server_name aimancloud.online www.aimancloud.online;

    # Use the paths where  the certificate and key is saved
    ssl_certificate /etc/ssl/certs/cloudflare-cert.pem;
    ssl_certificate_key /etc/ssl/private/cloudflare-key.pem;

    # SSL settings
    ssl_protocols TLSv1.2 TLSv1.3;
    ssl_prefer_server_ciphers on;
    ssl_ciphers 'ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384';
    ssl_session_timeout 1d;
    ssl_session_cache shared:SSL:10m;


    # Enable HSTS
    add_header Strict-Transport-Security "max-age=31536000; includeSubDomains" always;


    # Forward traffic to Nextcloud Docker instance
    location / {
        proxy_pass http://127.0.0.1:8080;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
    }
}
```

Step 7: Test and restart Nginx to ensure there are no syntax errors.

Command: sudo nginx -t

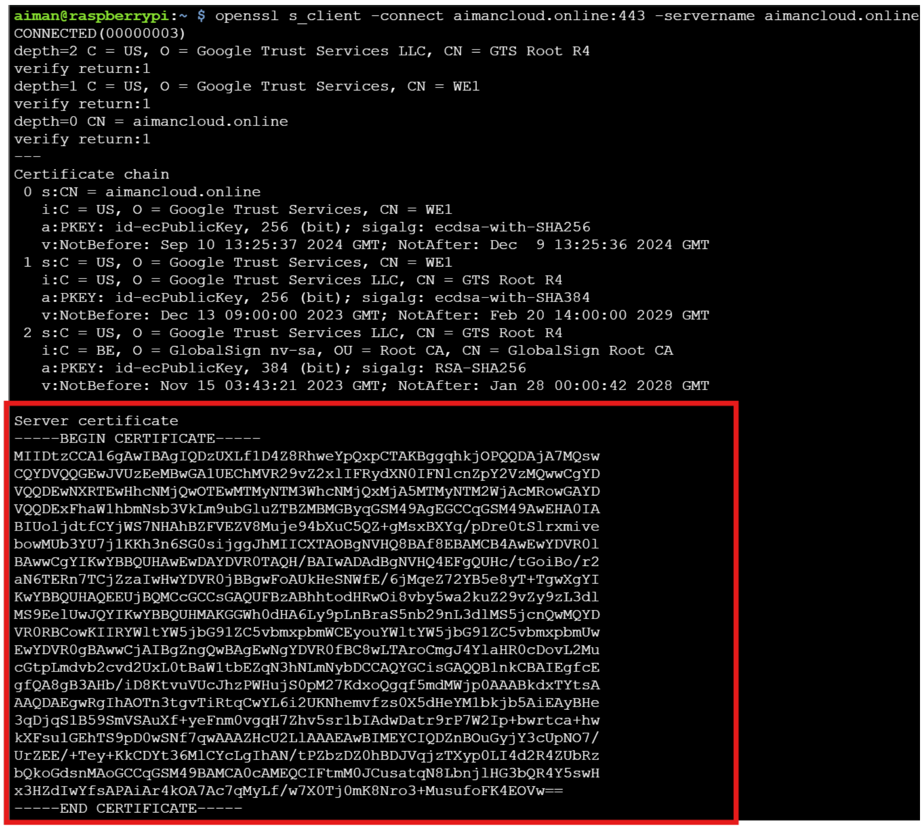
Command: sudo systemctl restart nginx



```
aiman@raspberrypi:~ $ sudo nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
aiman@raspberrypi:~ $ sudo systemctl restart nginx
aiman@raspberrypi:~ $ 
```

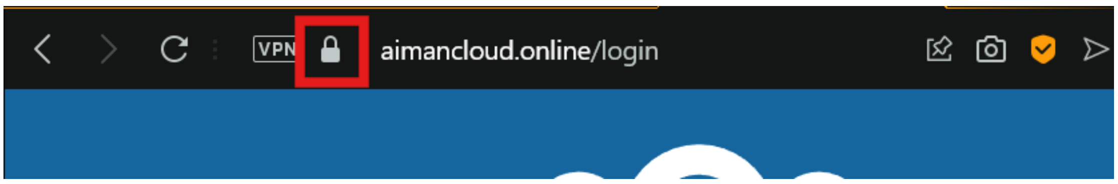
Step 8: Use OpenSSL to verify SSL certificate of the domain.

Command: openssl s_client -connect aimancloud.online:443 -servername aimancloud.online



```
aiman@raspberrypi:~ $ openssl s_client -connect aimancloud.online:443 -servername aimancloud.online
CONNECTED(00000003)
depth=2 C = US, O = Google Trust Services LLC, CN = GTS Root R4
verify return:1
depth=1 C = US, O = Google Trust Services, CN = WE1
verify return:1
depth=0 CN = aimancloud.online
verify return:1
---
Certificate chain
 0 s:CN = aimancloud.online
   i:C = US, O = Google Trust Services, CN = WE1
   a:PKEY: id-ecPublicKey, 256 (bit); sigalg: ecdsa-with-SHA256
   v:NotBefore: Sep 10 13:25:37 2024 GMT; NotAfter: Dec  9 13:25:36 2024 GMT
 1 s:C = US, O = Google Trust Services, CN = WE1
   i:C = US, O = Google Trust Services LLC, CN = GTS Root R4
   a:PKEY: id-ecPublicKey, 256 (bit); sigalg: ecdsa-with-SHA384
   v:NotBefore: Dec 13 09:00:00 2023 GMT; NotAfter: Feb 20 14:00:00 2029 GMT
 2 s:C = US, O = Google Trust Services LLC, CN = GTS Root R4
   i:C = BE, O = GlobalSign nv-sa, OU = Root CA, CN = GlobalSign Root CA
   a:PKEY: id-ecPublicKey, 384 (bit); sigalg: RSA-SHA256
   v:NotBefore: Nov 15 03:43:21 2023 GMT; NotAfter: Jan 28 00:00:42 2028 GMT
Server certificate
-----BEGIN CERTIFICATE-----
MIIDtzCCA16gAwIBAgIQDzUXLf1D4Z8RhweYpQxpCTAKBggqhkjOPQQDAjA7MQsw
CQYDVQQGEwJVUzEeMBwGA1UEChMVR29vZ2xlIFRydXN0IFNlcnZpY2VzMQwwCgYD
VQQDEwNXRTEwHhcNMjQwOTEwMTMyNTM3WhcNMjQxMjA5MTMyNTM2WjAcMRowGAYD
VQQDExFhaW1hbmNsb3VkLm9ubGluZTBZMBMGByqGSM49AgEGCCqGSM49AwEHA0IA
BIUo1jdtfCYjWS7NHAhBZFVEZV8Muje94bXuC5QZ+gMsxBXYq/pDre0tSlrxmive
bowMUb3YU7j1KKh3n6SG0sijggJhMIICXTAOBgNVHQ8BAf8EBAMCB4AwEwYDVR0l
BAwwCgYIKwYBBQUHAwEwDAYDVR0TAQH/BAIwADAdBgNVHQ4EFgQUHc/tGoiBo/r2
aN6TERn7TCjZzaIwHwYDVR0jBBgwFoAUkHeSNWfE/6jMqeZ72YB5e8yT+TgwXgYI
KwYBBQUHAQEEUjBQMCcGCCsGAQUFBzABhhtodHRwOi8vby5wa2kuZ29vZy9zL3dl
MS9EelUwJQYIKwYBBQUHMAKGGWh0dHA6Ly9pLnBraS5nb29nL3dlMS5jcnQwMQYD
VR0RBCowKIIRYWltYW5jbG91ZC5vbmxpbmWCEyouYWltYW5jbG91ZC5vbmxpbmUw
EwYDVR0gBAwwCjAIBgZngQwBAgEwNgYDVR0fBC8wLTAroCmgJ4YlaHR0cDovL2Mu
cGtpLmdvb2cvd2UxL0tBaW1tbEZqN3hNLmNybDCCAQYGCisGAQQB1nkCBAIEgfcE
gfQA8gB3AHb/iD8KtvuVUcJhzPWHujS0pM27KdxoQgqf5mdMWjp0AAABkdxTYtsA
AAQDAEgwRgIhAOTn3tgvTiRtqCwYL6i2UKNhemvfzs0X5dHeYM1bkjb5AiEAyBHe
3qDjqS1B59SmVSAuXf+yeFnm0vgqH7Zhv5sr1bIAdwDatr9rP7W2Ip+bwrtca+hw
kXFsu1GEhTS9pD0wSNf7qwAAAZHcU2LlAAAEAwBIMEYCIQDZnBOuGyjY3cUpNO7/
UrZEE/+Tey+KkCDYt36MlCYcLgIhAN/tPZbzDZ0hBDJVqjzTXyp0LI4d2R4ZUbRz
bQkoGdsnMAoGCCqGSM49BAMCA0cAMEQCIFtmM0JCusatqN8LbnjlHG3bQR4Y5swH
x3HZdIwYfsAPAiAr4kOA7Ac7qMyLf/w7X0Tj0mK8Nro3+MusufoFK4EOVw==
-----END CERTIFICATE-----
```

Step 9: Browse to domain name and https is enabled. Users can now log into the system securely via https.


VPN
aimancloud.online/login