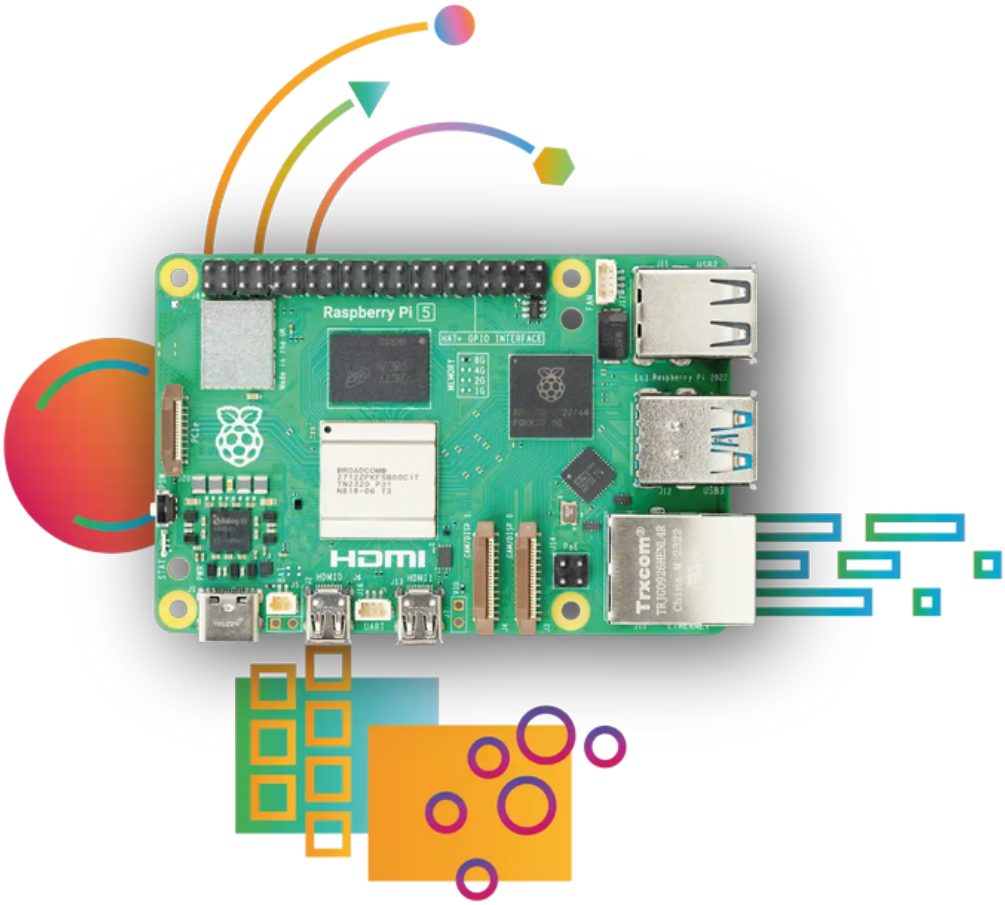
Mahatma Education Society's
Pillai HOC College of Arts, Science and Commerce
Pillai HOCL Educational Campus, Rasayani
Re-accredited 'A+' Grade by NAAC (CGPA 3.26, 2nd Cycle)
(ISO 9001: 2015 Certified)

Home Labs



Self-Hosted Cloud Storage, Torrenting, Media Streaming, Ad Blocking, and Code Development

Submitted By: Shriyash Balasaheb Patil

On 25th November 2023

Home Labs

Self-Hosted Cloud Storage, Torrenting, Media Streaming, Ad Blocking, and Code Development

Mr. Shriyash Balasaheb Patil **PHCASC, Rasayani**
Second Year BSc Computer Science

ABSTRACT

In the era where our lives are incomplete without digital devices, all our data is present on the cloud, where data privacy, customization, and control over the ecosystem is concerned, the concept of self-hosted home labs has gained significant momentum. The user can customize the site design and functionality according to their preferences and needs.This project presents you a way to self-hosting the ecosystem which includes Cloud Storage, Torrent Client, Media Server, Network Wide Ad-Blocker, Code Server.

Introduction

In an increasingly digital world, there is a need for data privacy, personalization and control over the online experiences has become more important. It is a necessity. As the prevalence of online services and cloud services continues to grow, it concerns data security, tracking and often invasive advertising online. In response to these challenges, the home lab concept has emerged as an effective solution that offers people the opportunity to create and maintain digital ecosystems in their homes.

Self Hosting is a form of running your own website or application by setting up a server and network yourself. Instead of using a Platform as a Service or a Public Cloud Provider, those who choose to self-host will run their own networks and be responsible for the maintenance and uptime in addition to building their website or application.

This self-hosted home lab project is a perfect example of that idea, and includes a rich set of essential services that together allow people to manage their data, entertainment, networking and even their coding work. Within the confines of their own hosted home lab, users can achieve a level of Security and customization that exceeds traditional cloud services and commercial software. They can set their own privacy standards, stream their media collections securely, block unwanted online ads and code in their browser while maintaining strong security measures.

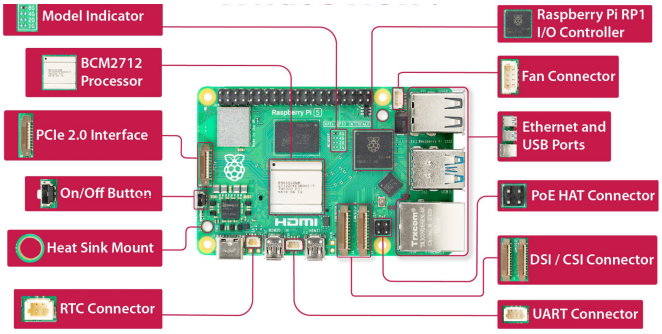
In this project we are integrating Cloud Storage, Torrent Client, Jellyfin Media Server, Pi-hole Ad Blocker, and Code Development Server into one unified ecosystem. Each of these components plays an important role in a

digital network that prioritizes user needs, provides control and preferences. Integration of these services, the self-hosted home laboratory is a testament to the power of technology, innovation and the individual operator in the modern digital age.

A homelab can be used to provide a variety of services to your home network, such as file sharing, media streaming, and web hosting. This can save you money on subscription fees and give you more control over your data. Having fun! Building and maintaining a homelab can be a lot of fun.

Materials:

1) Raspberry Pi 3/4/5: We are using Raspberry Pi 5



- Raspberry Pi is a Single Board Computer or SBC, Raspberry Pi is a small, low-cost, single-board computer the size of a credit card.
- We are using Raspberry Pi 5:
- CPU: Broadcom BCM2712 2.4GHz quad-core 64-bit Arm Cortex-A76
- RAM: 4GB LPDDR4X-4267 SDRAM
- Networking: Gigabit Ethernet

2) Storage Devices:
- We are using a 16GB SD Card as a boot device.
- 500GB HDD as media storage device.

3) Network Switch/Router:
- We are using a TP-Link home router which supports 1 Gig Ethernet.

4) Operating System:
- Raspberry Pi OS Lite (64-bit) Linux ARM-64 Based
  A port Debian Bookworm with no desktop environment.

5) Application Software:
- Raspberry Pi Imager, Docker, Portainer, Apache2, PHP, Nextcloud, Database Server, QbitTorrent, Pi-hole

Portainer Containers: Twingate

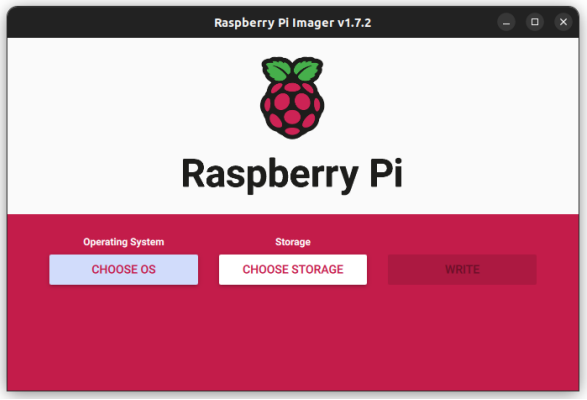
Methodology:

1) We will use the **Raspberry Pi Imager** tool to flash our SD Card with our Operating system.

- Here we will configure our WIFI(if required), User, and SSH/ Secure Shell.
- Now we will flash the SD Card with the OS including our configurations.

- We will now insert our SD Card in our Raspberry Pi and Power it on.

- Connect the Pi5 to the Network switch or Router using Wifi or Ethernet.



2) After we have successfully booted the Raspberry Pi we can connect to the Pi5 using SSH by using this command in the terminal.
   Command: **ssh username@IPaddress/hostname**
   →**Username:** Username which we have given in the imager tool
   →**IPaddress:** IP address which is assigned by the Network Switch.
   →**Hostname:** We can also use the hostname by default for Pi is for Rpi
   For example: **ssh shriyash@192.168.0.106** or
   - **ssh shriyash@raspberrypi**

   As I have chosen a OS with no desktop environment in order to save resources.
   If you are connecting to the Pi for the first time it will ask for fingerprint confirmation,Authenticate using yes keyword.
   Now, it will prompt you to enter the password you have given during the initial configuration.
   Now, you have connected to the Pi5 remotely using Secure Shell

- It is a good practice to Update and Upgrade our repositories.
- We can do that using following command: **sudo apt update && sudo apt upgrade -y**

3) Installing Docker:
With our Raspberry Pi entirely up to date, we can now go ahead and install Docker to the Raspberry Pi.
Luckily for us, Docker has made this process incredibly quick and straightforward by providing a bash script that installs everything for you.
You can download and run the official Docker setup script by running the following command.
curl -sSL https://get.docker.com | sh

Note: This process takes time to complete, it may look like the Pi has frozen.

Once Docker has finished installing your Raspberry Pi,
There are a couple more things we need to Do.
For another user to be able to interact with Docker, it needs to be added to the docker group.
So, our next step is to add our current user to the docker group by using the usermod

command as shown below. By using "$USER" we are inserting the environment variable that stores the current user's name.
sudo usermod -aG docker $USER

To test if Docker is working, we are going to go ahead and run the following command on our Pi.
This command will tell Docker to download, setup and run a docker container called "hello-world".
docker run hello-world

4) Now, we are going to install Portainer:

- Portainer is a lightweight management UI which allows you to easily manage your different Docker environments.
- With Docker set up and configured, we can use it
  to install Portainer to our Raspberry Pi.
- As Portainer is available as a Docker container on the official Docker hub, we can pull the latest version using the following command.

  sudo docker pull portainer/portainer-ce:latest

  Once Docker finishes downloading the Portainer image to your Raspberry Pi, we can now run it.
  Telling Docker to run this container requires us to pass in a few extra parameters.
  In the terminal on your Pi, run the following command to start up Portainer.
- Command for Starting Portainer:
  sudo docker run -d -p 9000:9000 --name=portainer --restart=always -v/var/run/docker.sock:/var/run/docker.sock -v portainer_data:/data portainer/portainer-ce:latest

- Now, we can access the WEB GUI of portainer using this URL:
  IPaddress:[PORT] or **Hostname:[PORT]**

- Here we have set our port to 9000 for Portainer.
  For example: **192.168.0.106:9000** or **raspberrypi.local:9000.**

  Visit the URL:
  →Portainer will prompt you to create a USER.
  Now, our Portainer setup is now completed, We can start containerizing and installing applications.

  Note: Using Docker, you can quickly deploy and scale applications into any environment and know your code will run. Running Docker on raspberry provides users and admins a highly reliable, low-cost way to build, ship, and run distributed applications at any scale. We can containerize or isolate each application, it will prevent conflicts between them.

5) As we don't want your services to only work on our home network, before proceeding further we are now making your Pi accessible through the internet.
   Method 1: Setting up port forwarding:

- Log in to the router as admin. You need to know the router's IP address, username, and password
- Locate the port forwarding options.
- Type the port number or port range that you want to forward.
- Type the static IP address you chose.
- Enable the port forwarding rule with an Enable or On option.

Note:If you port forward incorrectly, it can lead to security risks within your network. More specifically, it can open a network to security vulnerabilities and cyber attacks. Here we can use twingate docker image, as our project is more focused on docker containers we are going to use Twingate.

Method 2: Twingate enables secure, remote access to your home network and services like Home Assistant, Plex, security cameras, and other self-hosted apps.

- Go to the twingate official website and create an account and create a network.
- Navigate to the Remote Networks and Add a remote network and configure the Location to On Premise .
- Now create a new resource, and provide your Raspberry Pi IP address.
- Now setup a connector and generate the access token.
- Now run the following command to execute twingate docker image. Command:(Execute in the RPI5 SSH Terminal)

docker run -d --sysctl net.ipv4.ping_group_range="0 2147483647" --env TWINGATE_NETWORK="{network_name}" --env TWINGATE_ACCESS_TOKEN="{Your_access_token}" --env TWINGATE_REFRESH_TOKEN="{Your_refresh_token}" --env TWINGATE_LABEL_HOSTNAME="`hostname`" --name "twingate-green-kangaroo" --restart=unless-stopped --pull=always twingate/connector:1

- Now, if we can see a container in our portainer admin panel, we have successfully created a twingate docker image and we can access our raspberry pi through the internet.
- Now, setup the twingate client on android, ios, windows, mac etc to access it through the internet.
- Now, we are able to access our portainer dashboard through internet:

- We can now access our Portainer Dashboard through the internet, using this same method we are going to access our other services.
- To make them easy to access we are going to use a web based dashboard to navigate through apps.
- After setting up all your services and applications we are going to make the dashboard.

6) Installing Apache web server:To install

apache2 on your Raspberry Pi, enter the following command into the terminal.
Command: **sudo apt install apache2 -y**
You can check by visiting IP address for your Pi or visiting raspberrypi.local

7) Now we are going to install PHP:

- With Apache2 now installed onto the Raspberry Pi, we just need to install PHP and several of its packages.
run the following command to install the “lsb-release” package.
Command: **sudo apt install lsb-release**
To use any third-party repository within Raspberry Pi OS / Debian, you need to provide its GPG key.
Command: **curl https://packages.sury.org/php/apt.gpg | sudo tee /usr/share/keyrings/suryphp-archive-keyring.gpg >/dev/null**
Once the key has been saved to your Raspberry Pi, we can create a new source file that points to the repository. Use the following one-liner to create this source file with the link to the repository.
Command:**echo "deb [signed-by=/usr/share/keyrings/suryphp-archive-keyring.gpg] https://packages.sury.org/php/ $(lsb_release -cs) main" | sudo tee /etc/apt/sources.list.d/sury-php.list**
By running an update, we are requesting new package lists from all of the sources. This will make APT aware of the packages being provided by our new PHP repository.
Command:**sudo apt update**
With Apache2 now installed onto the Raspberry Pi, we just need to install PHP and several of its packages.
To install PHP and the packages we need, run the following command.
Command: **sudo apt install php8.2 php8.2-gd php8.2-sqlite3 php8.2-curl php8.2-zip php8.2-xml php8.2-mbstring php8.2-mysql php8.2-bz2 php8.2-intl php8.2-smbclient php8.2-imap php8.2-gmp php8.2-bcmath libapache2-mod-php8.2**
With Apache and PHP now installed there is one final thing we need to do, and that is to restart Apache.
Command:**sudo service apache2 restart**

Now we are ready to install our services and applications on our Home Lab Server.

8) Now we are going to Nextcloud:
Nextcloud is a suite of client-server software for creating and using file hosting services. Nextcloud provides functionality similar to Dropbox, Office 365 or Google Drive when used with integrated office suites Collabora Online or OnlyOffice.

First we need to setup MySQL Database and User for Nextcloud:

→Installing MySQL to the Raspberry Pi is a simple process and can be done with the following command.

Command: **sudo apt install mariadb-server**
→With the MySQL server software installed to the Raspberry Pi, we will now need to secure it by setting a password for the “root” user.
By default, MySQL is installed without any password set up meaning you can access the MySQL server without any authentication.
Command:**sudo mysql_secure_installation**
→Now, The first thing we need to do is open the MySQL command line tool by running the following command.
Command: **sudo mysql -u root -p**
→Once you have logged in to the tool, we can start by creating a database.
We will be creating this database called nextclouddb by running the following command.
SQL Command: **CREATE DATABASE nextclouddb;**
→Our next step is to create a user that we will be using to interact with our new database.

We will be creating a user called nextclouduser by running the command below. Make sure that you replace [PASSWORD] with a secure password and make note of it for later.
SQL Command: **CREATE USER 'nextclouduser'@'localhost' IDENTIFIED BY '[PASSWORD]';**
→With our user created we need to now give it permissions to interact with our database.
We can do that by running the following command.
SQL Command: **GRANT ALL PRIVILEGES ON nextclouddb.* TO 'nextclouduser'@'localhost';**
→Our final task is to flush the privilege table.
SQL Command: **FLUSH PRIVILEGES;**
Downloading Nextcloud on your Raspberry Pi:
→To get started let’s first move to our html directory with the following change directory command.
Command: **cd /var/www/**
→Now we can download the latest version of Nextcloud to our device.
To do this we will use wget to download the latest release to the current folder.
Command: **sudo wget https://download.nextcloud.com/server/releases/latest.tar.bz2**
→With Nextcloud now downloaded to our Raspberry Pi, let us extract the archive.
To extract the archive using tar we need to use the command below.
Command: **sudo tar -xvf latest.tar.bz2**
→We now need to create a data directory for Nextcloud to operate in, for the initial setup of Nextcloud we must make this folder in our html/nextcloud directory.
Create the directory by using the mkdir command as shown below:
Command:**sudo mkdir -p /var/www/nextcloud/data**

→Now let's give the correct user and group control over the entire Nextcloud folder and everything inside it by running the following chown command.
Command: **sudo chown -R www-data:www-data /var/www/nextcloud/**
→Finally we need to give it the right permissions, again run the following chmod command.
Command: **sudo chmod 750 /var/www/nextcloud/data**

9) Configuring Apache for Nextcloud:
→let's create a file which will store our configuration changes for Nextcloud:
Command: **sudo nano /etc/apache2/sites-available/nextcloud.conf**
→This configuration is simple and means whenever someone goes to your Pi's IP address, followed by "/nextcloud" they will be greeted with its interface.
Config file:

```
Alias /nextcloud "/var/www/nextcloud/"

<Directory /var/www/nextcloud/>
  Require all granted
  AllowOverride All
  Options FollowSymLinks MultiViews

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

</Directory>
```

→These lines basically tell Apache2 how to handle itself within the /var/www/nextcloud/ folder.
These changes will allow Apache2 to read and utilize the .htaccess files within the Nextcloud directory.
Now we can save and quit out of the file by pressing CTRL + X then pressing Y and then ENTER.
→With the file created we now need to tell Apache to make use of it.
We can do this by utilizing the a2ensite command followed by nextcloud.conf.
Command: **sudo a2ensite nextcloud.conf**
→Now we need to restart Apache2 to force it to read in the updated configuration file. We can do that easily with the following
Command: **sudo systemctl reload apache2**
→ So now our nextcloud service is ready to use, you can access it by visiting
URL: **ipaddress/nextcloud** or **raspberrypi.local/nextcloud**
10) NextCloud Setup:
→Visit: URL: **ipaddress/nextcloud** or **raspberrypi.local/nextcloud.**
Here we have successfully installed Nextcloud and now we can access remotely from anywhere in the world.We can store our data in our own house in our own environment.Its a great replace from Gdrive, Gcalender, Docs, sheets etc as all these features are present in nextcloud.

11) Now, we are going to install a torrenting service:

Qbittorrent: To install qbittorrent we will simply run a docker run command and our torrent client will be successfully installed.

Command: **docker run -d --name=qbittorrent -e PUID=1000 -e PGID=1000 -e TZ=Etc/UTC -e WEBUI_PORT=8080 -p 8080:8080 -p 6881:6881 -p 6881:6881/udp -v /path/to/appdata/config:/config -v /path/to/downloads:/downloads --restart unless-stopped lscr.io/linuxserver/qbittorrent:latest**

12) Installing Network-Wide Ad-Blocker:

Here, I have chosen pi-hole as it provides a web-based and user-friendly interface to manage our DNS Server and block list. Pi-Hole will create and operate as a Domain Name System (DNS) pathway for your internet system. With a completed system your Raspberry Pi Single Board Computer will take all the internet data coming in, filter it meticulously, and only display what you actually want to see.

Installation:

run the below commands for Docker to create two volumes

Command:

docker volume create pihole_app
docker volume create dns_config

run the command below to pull the pihole/pihole base image from Docker hub.

Command:

docker run --name=pihole -e TZ=Asia/Manila -e WEBPASSWORD=password -e SERVERIP=YourIPAddressHere -v pihole_app:/etc/pihole -v dns_config:/etc/dnsmasq.d -p 81:80 -p 53:53/tcp -p 53:53/udp --restart=unless-stopped pihole/pihole

To visit Pi-Hole Dashboard Navigate to:

[IP-Address:PORT]/admin

13) Installing Media Server:

Here, I have chosen Jellyfin Media Server. If you're looking for a free, customizable platform that offers complete control over your media collection, Jellyfin may be the best option for you as well.

Installation:

- Download the latest container image.

docker pull jellyfin/jellyfin

- create two directories on the host and use bind mounts:

mkdir /path/to/config
mkdir /path/to/cache

Or create two persistent volumes:

docker volume create jellyfin-config
docker volume create jellyfin-cache

- Execute the container:

```
docker run -d \
  --name=jellyfin \
  -e PUID=1000 \
  -e PGID=1000 \
  -e TZ=Etc/UTC \
```

```
-e
  JELLYFIN_PublishedServerUrl=192.1
  68.0.5 `#optional` \
-p 8096:8096 \
-p 8920:8920 `#optional` \
-p 7359:7359/udp `#optional` \
-p 1900:1900/udp `#optional` \
-v /path/to/library:/config \
-v /path/to/tvseries:/data/tvshows \
-v /path/to/movies:/data/movies \
--restart unless-stopped \
lscr.io/linuxserver/jellyfin:latest
```

Here We have Installed Jellyfin server.

14) Installing Code Server:
Code Server Provides a local networder wide environment to write and execute different code in our local environment.
Installation:

```
docker run -d \
  --name=code-server \
  -e PUID=1000 \
  -e PGID=1000 \
  -e TZ=Etc/UTC \
  -e PASSWORD=password `#optional`
\
  -e HASHED_PASSWORD= `#optional`
\
  -e SUDO_PASSWORD=password
`#optional` \
  -e SUDO_PASSWORD_HASH=
`#optional` \
  -e
PROXY_DOMAIN=code-server.my.dom
ain `#optional` \
  -e
DEFAULT_WORKSPACE=/config/work
space `#optional` \
  -p 8443:8443 \
  -v /path/to/appdata/config:/config \
  --restart unless-stopped \
  lscr.io/linuxserver/code-server:latest
```

15) Here we are going to now create a dashboard to easily access all our services and servers.

You can download the dashboard html file from my github:
Link:
https://github.com/ShriyashBPatil/HomeLab-Dashboard

And save the files in following location
/var/www/html

Now visit [IP-Address] to access the dashboard.

Home Labs:
Self-Hosted Cloud Storage, Torrenting, Media Streaming, Ad Blocking, and Code Development

You first need to know your needs before building your home lab server, you can choose any hardware which is based on x86, x64, arm platform. These are widely used platforms and have a wide range of application support as well as cross platform support.
In this project, I have chosen Raspberry Pi 5 as the hardware which is based on arm-64 architecture.

Despite its small size and low cost, a Raspberry Pi single-board computer can be used to run servers. In fact, server hosting is one of the most popular uses for a Raspberry Pi, and for good reason. They are cheap, power-efficient, and very powerful for their size. However, there are several factors that need to be taken into consideration when choosing a Raspberry Pi to run your server. But at the end of the day RPI is a great choice for home labs.

I want as much raw performance as I can get from my Pi 5. If I have chosen an operating system with a desktop environment, it will unnecessarily consume our resources. So, I have opted for an operating system with no desktop environment. Here I can access the server headless using SSH/Secure Shell.

Raspberry Pi Imager is the quick and easy way to install Raspberry Pi OS and other operating systems to a microSD card, ready to use with your Raspberry Pi. Download and install Raspberry Pi
Image to a computer with an SD card reader.

Docker is a set of platform as a service products that use OS-level virtualization to deliver software in packages called containers.
Portainer is a lightweight management UI which allows you to easily manage your different Docker environments.

If you port forward incorrectly, it can lead to security risks within your network. More specifically, it can open a network to security vulnerabilities and cyber attacks. Here we can use twingate docker image, as our project is more focused on docker containers we are going to use Twingate. It creates a private tunnel to our home network that only you can access.

Apache is a web server software that is responsible for accepting HTTP requests from visitors and sending them back the requested information in the form of web pages.
Or in simpler terms, it allows visitors to view content on your website.

Maria DB instead of mysql or mysql lite because of its advantage over both of them that Maria DB is more faster and efficient as compared to both of them when it comes to replication and

performing mariadb is significantly faster than MySql and my SQL lite.
As we are building a home lab which concerns more about the Performance, In case of performance in efficiency here the Marya DB is a better alternative as it is a lightweight version of my SQL and also offers better performance and more features .

Next cloud is a cloud storage service which allows us to store and access data over the internet. There are various features in the next cloud which includes office suite, calendar to organize events, file sharing remote access manager and activity manager. Next cloud is a free and open source cloud storing service which enables us to access files remotely through the internet. It allows us to integrate cloud services with various databases such as Maria DB ,MySQL Lite, MySQL.

Qbittorrent Parameters→

Here is simple explanation to the parameters we have used:

Parameter	Function
`-p 8080`	WebUI
`-p 6881`	tcp connection port
`-p 6881/udp`	udp connection port
`-e PUID=1000`	for UserID - see below for explanation
`-e PGID=1000`	for GroupID - see below for explanation
`-e TZ=Etc/UTC`	specify a timezone to use, see this list.
`-e WEBUI_PORT=8080`	for changing the port of the webui, see below for explanation
`-v /config`	Contains all relevant configuration files.
`-v /downloads`	Location of downloads on disk.

2) Raspberry Pi Deluge
   Deluge is a popular torrent client that you can install for the Raspberry Pi. Much like Transmission, it has a web interface that you can access remotely.

Pi-Hole:

Pi-hole can reduce your bandwidth usage and data consumption, as it prevents unwanted traffic from reaching your devices. This can save you money and improve your network performance, especially if you have a limited or slow internet connection.

Jellyfin:

Jellyfin is a Free Software Media System that puts you in control of managing and streaming your media. It is an alternative to the proprietary Emby and Plex, to provide media from a dedicated server to end-user devices via multiple apps. Jellyfin is descended from Emby's 3.5.2 release and ported to the .NET Core framework to enable full cross-platform support. There are no strings attached, no premium licenses or features, and no hidden agendas: just a team who want to build something better and work together to achieve it.

Code Server:

Code-server is an open-source project that allows you to run Visual Studio Code (VSCode) on a remote server through the browser. VSCode is an open-source code editor by Microsoft

that offers features like IntelliSense, Git integration, and plugins, while staying relatively lightweight.

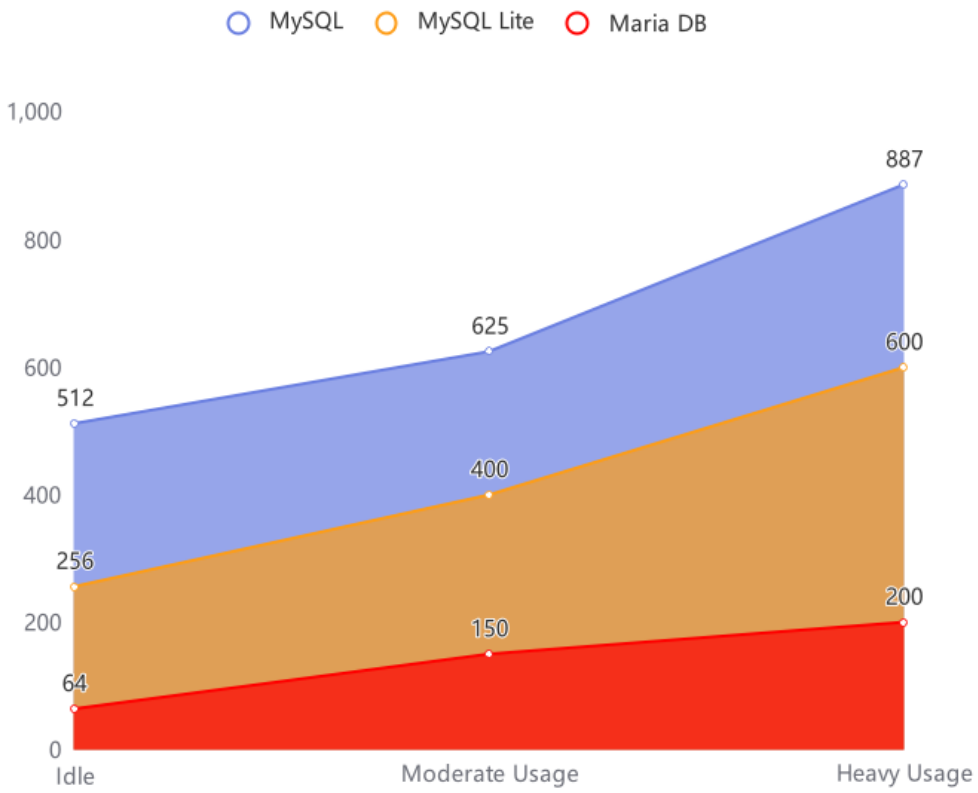
Result

- Cloud Server requires database services.
- Here, I have tested different databases for our Cloud Storage Service.
- We have tested three different use cases

Case 1: Idle → When the server is doing nothing.

Case 2: Moderate Usage → Here we are normally using our cloud services accessing and modifying files.

Case 3: Heavy Usage → Here we are stressing our server by fetching login details and accessing-modifying files.

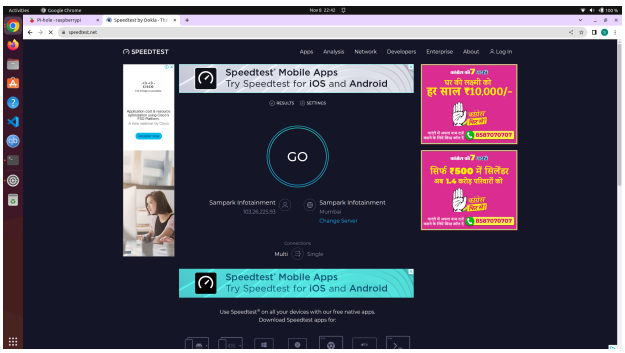


- As we have already mentioned, MariaDB shows faster speed and better performance than MySQL. If you're looking for a high-performance RDBMS, MariaDB is a good choice. The biggest advantage of MariaDB is that it is open-source and free.
- Maria DB consumed less amount of memory as compared to MySQL and MySQL Lite.Here our main concern is performance and efficiency, home labs are not meant to be upgraded frequently. So, I am focusing on lightweight applications to make the HomeLab ecosystem.

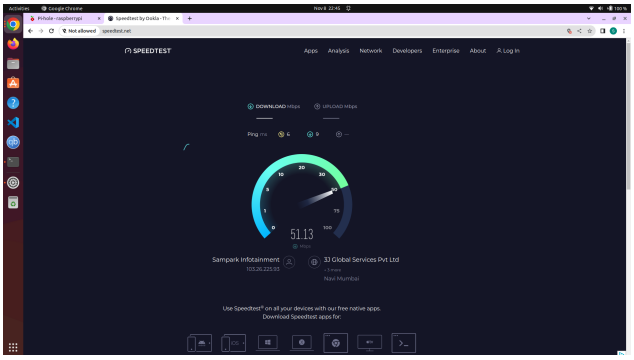
Home Labs:
Self-Hosted Cloud Storage, Torrenting, Media Streaming, Ad Blocking, and Code Development

Here we have tested the result for the Pi-hole ad-blocking feature, I have visited speedtest.net site and we can see the results.

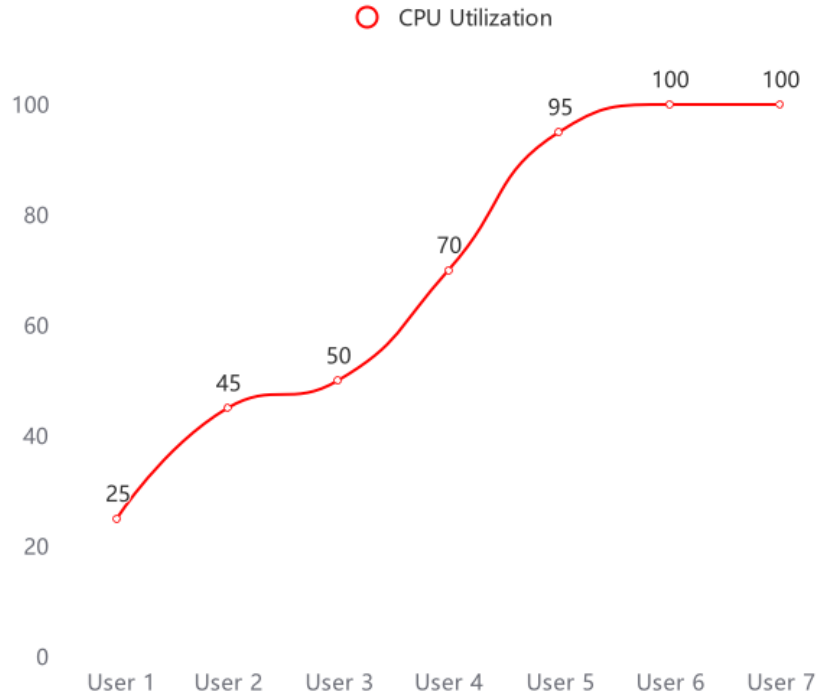
Pi-Hole:(Pi-hole disabled)



(Pi-hole enabled)



Jellyfin Streams VS System Usage



Here we can see that our server can handle 7 users at the same time.
After the 8th user tries to access the server the stream lags across the board.

Code Server:
Allows you to run Visual Studio Code (VSCode) on a remote server through the browser. VSCode is an open-source code editor by Microsoft that offers features like IntelliSense, Git integration, and plugins, while staying relatively lightweight.

Conclusion

We have managed to create a Home Server that can run Personalized Services depending on the user.To summarize, the rise of self-hosted home labs is a compelling response to the growing concerns about data privacy, personalization, and control in the digital age. People are looking for options that give them more control over their digital experiences as our reliance on cloud computing and internet services grows.

The integrated home lab project, which is one example of the self-hosting trend, represents a move toward increased customization and autonomy. When people choose self-hosting, they are able to take back control of their data and customize their digital ecosystems to fit their own tastes and security requirements.

References

Websites:

Raspberry Pi : https://www.raspberrypi.org/
Docker: https://docs.docker.com/
Portainer: https://docs.portainer.io/
NextCloud: https://nextcloud.com/
Twingate: https://www.twingate.com/docs/how-twingate-works
Maria DB: https://mariadb.org/documentation/
Apache: https://cwiki.apache.org/confluence/display/httpd/FAQ
Qbit: https://hub.docker.com/r/linuxserver/qbittorrent
Pi-hole: https://adamtheautomator.com/pi-hole-in-docker/
Ad Blocker list: https://firebog.net/
Jellyfin: https://hub.docker.com/r/linuxserver/jellyfin
CodeServer: https://hub.docker.com/r/linuxserver/code-server