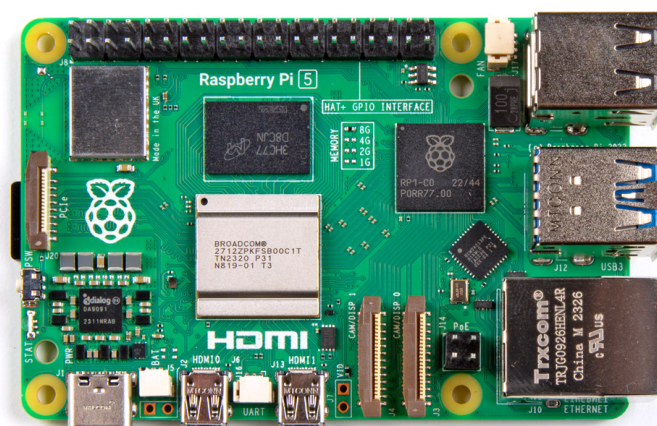


# Raspberry Pi 5 Projects

## A Collection of Practical Applications

Tautminas Cibulskis



April 5, 2025

# Contents

|   |    |
|---|----|
| About . . . . .                                       | 2  |
| Raspberry Pi 5 Preparation . . . . .                  | 3  |
| 1 Hosting a Dark Web Site . . . . .                   | 5  |
| 2 Mining Monero Cryptocurrency . . . . .              | 7  |
| 3 Playing Games with RetroPie . . . . .               | 9  |
| 4 Setting Up a NAS with Samba . . . . .               | 12 |
| 5 Running Large Language Models with Ollama . . . . . | 13 |
| 6 Setting Up Pi-hole for Ad Blocking . . . . .        | 14 |

# About

This document provides information for setting up projects with a Raspberry Pi 5. Each project is inspired by a YouTube video, but this document refines and expands upon the content to offer clear guidance for those who wish to replicate the projects. The purpose of collecting this information in a single document is as follows:

- Some videos lack quality or are no longer relevant. This document selects the useful ones.
- Some videos do not provide commands in written form. By providing them here, readers can avoid needing to retype everything manually.
- Some videos could benefit from additional suggestions or improvements. This document expands on those projects to make them more practical.

The projects in this document were created using the Raspberry Pi 5 Desktop Kit. However, in most cases, a Raspberry Pi 5 Starter Kit provides all the necessary components to follow along. Additionally, these projects are not specific to the Raspberry Pi 5 as most of them can be completed using other Raspberry Pi models as well.

Some instructions require specific values, such as an IP address or cryptocurrency wallet address. These values are represented in the format `<EXAMPLE_VALUE>`, with angle brackets and capitalization to indicate that they must be replaced with actual values. Blindly copying and pasting commands without replacing these placeholders will result in errors. Pay attention to the instructions and substitute the placeholders with the appropriate values.

In the realm of project management, a project is defined as a time-bound endeavour with a unique outcome. While this definition does not fully align with how it is used in the document, it remains the closest approximation to describe the work presented here.

# Raspberry Pi 5 Preparation

Before using your Raspberry Pi, you must first install an operating system onto a microSD card. Follow these steps to set it up:

1. Insert the microSD card into your PC (not the Raspberry Pi) using an SD adapter. Depending on the adapter, plug it into either the SD card slot or a USB port on your computer.
2. Open the [Raspberry Pi Imager](#) application. This tool simplifies the process of installing an operating system onto your Raspberry Pi's storage. It has three main options:
  - Device: Select your Raspberry Pi device (in this case, Raspberry Pi 5).
  - Operating System: Choose the OS based on the project's requirements. The required OS is stated at the start of the project section.
  - Storage: Select the storage device (in this case, a microSD card) that will be used to boot the Raspberry Pi and store data. Be careful, as this process will erase all existing data on the selected storage device.
3. Customize the settings to enable remote access to the Raspberry Pi by pressing **Ctrl + Shift + X**. Then, set the hostname, username, password, configure the wireless LAN, set locale settings and enable SSH.
4. Proceed with the OS installation by selecting **Next**. The Raspberry Pi Imager will then write the OS to the selected storage device.
5. Once the OS is successfully written to the microSD card, insert it into the Raspberry Pi and power it on. Your Raspberry Pi will now be ready to use.

For most projects, you'll need to connect to the Raspberry Pi remotely. To do so, follow these steps:

1. Use a tool like [Fing](#), available for both mobile and desktop, to find the Raspberry Pi's IP address. Fing detects devices on your local network and displays their IP addresses.
2. Once you have the IP address, connect to the Raspberry Pi via SSH. Use a tool like [PuTTY](#), a free SSH client for Windows. Launch PuTTY, enter the Raspberry Pi's IP address, select SSH as the connection type and click **Open**. Log in using your customized credentials (username and password) if you have set them, or use the default credentials (username: pi, password: raspberry) otherwise.

After logging into your Raspberry Pi 5, either remotely or by directly accessing it, you need to run two essential commands that are recommended for every project:

1. **sudo apt update**: This command updates the list of available packages and their versions from the repositories.
2. **sudo apt upgrade**: This command upgrades all the installed packages to their latest available versions.

You can run both commands at once by using:

```
sudo apt update && sudo apt upgrade -y
```

The `-y` flag automatically confirms all prompts during the upgrade process, so you won't need to manually approve each step. This may take some time depending on your internet speed and the number of packages that need to be upgraded.

# 1 Hosting a Dark Web Site

**Video:** [i put a DARK WEB website on a Raspberry Pi!!](#) (Channel: [NetworkChuck](#))

**Operating System:** Raspberry Pi OS Lite (64-bit), remote connection.

## Steps:

1. Update package lists and install Tor:

```
$ sudo apt update
$ sudo apt install tor
```

2. Edit the Tor configuration file:

```
$ sudo nano /etc/tor/torrc
```

Uncomment the following lines:

```
HiddenServiceDir /var/lib/tor/hidden_service/
HiddenServicePort 80 127.0.0.1:80
```

3. Restart the Tor service:

```
$ sudo service tor stop
$ sudo service tor start
$ sudo service tor status
```

4. Obtain your site address:

```
$ sudo cat /var/lib/tor/hidden_service/hostname
```

5. Install and start Nginx:

```
$ sudo apt install nginx
$ sudo service nginx start
$ sudo service nginx status
```

You can now check your website using the .onion address from the previous step.

6. Modify Nginx configuration:

```
$ sudo nano /etc/nginx/nginx.conf
```

Uncomment the following lines:

```
port_in_redirect off;
server_name_in_redirect off;
```

Add the following line below "port\_in\_redirect off;":

```
server_tokens off;
```

7. Restart Nginx to apply changes:

```
$ sudo service nginx restart
```

Visit the site using your .onion address to check if the site works after the configuration changes.

8. Create a simple web page:

```
$ cd /var/www/html  
$ sudo rm index.nginx-debian.html  
$ sudo nano index.html
```

Add your HTML content and save the file.

9. Restart Nginx to apply changes:

```
$ sudo service nginx restart
```

Your site should now be accessible via your .onion address!

## 2 Mining Monero Cryptocurrency

Video: [How to Mine Monero on Raspberry Pi](#) (Channel: [NetworkChuck](#))

Operating System: Raspberry Pi OS Lite (64-bit), remote connection.

### Steps:

1. Update package lists and install dependencies:

```
$ sudo apt update
$ sudo apt install git build-essential cmake libuv1-dev libssl-dev libhwloc-dev -y
```

2. Clone the XMRig repository and build the miner:

```
$ git clone https://github.com/xmrig/xmrig.git
$ cd xmrig
$ mkdir build
$ cd build
$ cmake ..
$ make
```

3. Create a Monero (XMR) wallet: Download and install the Monero GUI Wallet from [getmonero.org](#), choose Simple Mode, create a new wallet, name it, back up the mnemonic seed, set a password, and find your wallet address under the Account tab.

4. Start mining Monero:

```
$ ./xmrig -o gulf.moneroocean.stream:10128 -u
<YOUR_WALLET_ADDRESS> -p <WORKER_NAME>
```

Use the following keys to monitor mining in real time:

- H — Show hashrate
- S — Display the number of accepted shares
- C — Check connection status

To track your mining performance, visit [Monero Ocean](#), enter your wallet address and view your statistics.

### Optional Enhancements:

- Running the miner in a detached session with tmux:

```
$ tmux
$ ./xmrig -o gulf.moneroocean.stream:10128 -u
<YOUR_WALLET_ADDRESS> -p <WORKER_NAME>
```



Press **Ctrl+b**, then **d** to detach the session while keeping it running in the background.

To reattach the session use the following command:

```
$ tmux attach
```

- Automating mining with a startup script:

```
$ nano start_mining.sh
```

Add the following lines:

```
#!/bin/bash
cd <BUILD_DIRECTORY_OF_XMRIG>
tmux new-session -d -s xmrigh_session './xmrigh -o
gulf.moneroocean.stream:10128 -u <YOUR_WALLET_ADDRESS> -p
<WORKER_NAME>'
```

Save and exit, then make the script executable:

```
$ chmod +x start_mining.sh
```

Run the script:

```
$ ./start_mining.sh
```

The script ensures that Monero mining starts in a detached tmux session, allowing it to run persistently in the background. The session can be reattached using **tmux attach**.

### 3 Playing Games with RetroPie

Video: [RetroPie: A Raspberry Pi Gaming Machine](#) (Channel: [NetworkChuck](#))

Operating System: Raspberry Pi OS Lite (64-bit)

#### Steps:

1. Prepare the system and install essential packages for RetroPie setup:

```
$ sudo apt update && sudo apt upgrade -y  
$ sudo apt install git lsb-release
```

2. Clone the RetroPie setup script:

```
$ cd  
$ git clone https://github.com/retropie/retropie-setup.git  
$ cd retropie-setup  
$ chmod +x retropie_setup.sh
```

3. Run the setup script:

```
$ sudo ./retropie_setup.sh
```

Follow the prompts: OK → OK → Basic Install → Yes. The installation will take a while.

Once complete, go to Configuration / tools → autostart → Start EmulationStation at boot → OK.

Then, go to bashwelcometweak → Install Bash Welcome Tweak → OK → Exit.

4. Reboot the system:

```
$ sudo reboot
```

5. Setup the keyboard:

| Gamepad Input      | Keyboard Key        |
|--------------------|---------------------|
| D-PAD UP           | Arrow Up            |
| D-PAD DOWN         | Arrow Down          |
| D-PAD LEFT         | Arrow Left          |
| D-PAD RIGHT        | Arrow Right         |
| START              | Enter               |
| SELECT             | Tab                 |
| BUTTON A / EAST    | Spacebar            |
| BUTTON B / SOUTH   | Left Shift          |
| BUTTON X / NORTH   | Z                   |
| BUTTON Y / WEST    | X                   |
| LEFT SHOULDER      | Q                   |
| RIGHT SHOULDER     | E                   |
| LEFT TRIGGER       | R                   |
| RIGHT TRIGGER      | T                   |
| LEFT THUMB         | C                   |
| RIGHT THUMB        | V                   |
| LEFT ANALOG UP     | W                   |
| LEFT ANALOG DOWN   | S                   |
| LEFT ANALOG LEFT   | A                   |
| LEFT ANALOG RIGHT  | D                   |
| RIGHT ANALOG UP    | I                   |
| RIGHT ANALOG DOWN  | K                   |
| RIGHT ANALOG LEFT  | J                   |
| RIGHT ANALOG RIGHT | L                   |
| HOTKEY ENABLE      | Left Control (Ctrl) |

Table 1: Suggested keyboard mapping for RetroPie

6. Configure Raspberry Pi settings in RetroPie menu in the RASPI-CONFIG option:
  - System Options → Hostname → Enter a hostname
  - System Options → Password → Enter a new password → OK
  - Localisation Options → Timezone → Select location → Select timezone
  - Localisation Options → WLAN Country → Select country
  - Interface Options → SSH → Yes
7. Select the WIFI option in the RetroPie menu → Connect to WiFi network → Pick network and enter the password
8. Download ROMs for your preferred games using another device. ROMs can be downloaded from [emulatorgames.net](http://emulatorgames.net) and [romsgames.net](http://romsgames.net). Check the list of [RetroPie Supported Systems](#) to ensure game compatibility before downloading.
9. In the RetroPie menu select SHOW IP. Transfer ROMs to the Raspberry Pi using SCP from the device where the ROMs are stored.

```
$ scp -r <PATH_TO_ROMS_DIRECTORY>/*
<USERNAME>@<RASPBERRY_PI_IP>:~/RetroPie/roms/
```

10. Organize the ROMs into their appropriate emulator folders on the Raspberry Pi. Pressing **F4** will take you to the command line interface (CLI). For example, PlayStation games should go into `/RetroPie/roms/psx`. Use `cd` command to move ROMs to the needed location. To return to EmulationStation use the `emulationstation` command.
11. Personalize game lists in the MAIN MENU:
  - **UI SETTINGS** → **GAME LIST VIEW STYLE** → **GRID** → set **IGNORE ARTICLES (NAME SORT ONLY)** to OFF
  - **SCRAPER** → **SCRAPE FROM** → **SCREENSCRAPER** → **SCRAPE NOW** → Set **SYSTEMS** to all → set **USER DECIDES ON CONFLICTS** to off → **Start**
12. Now you can play the games. If desired, use these hotkeys while playing:
  - **HOTKEY + Start**: Exit game
  - **HOTKEY + Right Shoulder**: Save game
  - **HOTKEY + Left Shoulder**: Load game
  - **HOTKEY + D-PAD Right**: Change save/load slot
  - **HOTKEY + Button B**: Reset game
  - **HOTKEY + Button X**: Open RetroArch menu

### Optional Enhancements:

- Enable cheats: Open RetroArch menu (**HOTKEY + Button X**) → **Settings** → **User Interface** → set **Show Advanced Settings** to ON → Go back to **User Interface** → **Menu Item Visibility** → set **Show 'Online Updater'** to ON → Go back to **Main menu** of RetroArch → **Online Updater** → **Update Cheats** → Go back to the **Main menu** of RetroArch → **Quick Menu** → **Cheats** → **Load Cheat File (Replace)** → Select console → Turn ON the desired cheats by selecting each one and setting **Enable** to ON → **Apply Changes**.
- Enable achievements:
  1. Sign up at [RetroAchievements](#).
  2. Open the game on the Raspberry Pi in which you would like to turn on the achievements.
  3. Open RetroArch menu (**HOTKEY + Button X**) → **Settings** → **Achievements** → Turn it ON → Select **Username** and put in value → Select **Password** and put in value → set **Unlock Sound** to ON.
  4. Go back to the **Main menu** of RetroArch → **Configuration File** → **Save Current Configuration** → **Quit RetroArch**.

## 4 Setting Up a NAS with Samba

Video: [Pi Network File Share to Windows & More | Pi NAS/SMB | Raspberry Pi Guide](#) (Channel: [TroubleChute](#))

**Operating System:** Raspberry Pi OS Lite (64-bit), remote connection.

### Steps:

1. Prepare the system and install Samba:

```
$ sudo apt update && sudo apt upgrade -y
$ sudo apt install samba samba-common-bin -y
```

2. Create a shared directory:

```
$ mkdir ~/shared
```

3. Edit the Samba configuration file:

```
$ sudo nano /etc/samba/smb.conf
```

Add the following at the end of the file:

```
[shared]
path=/home/<USERNAME>/shared
writable=Yes
create mask=0666
directory mask=0666
public=no
```

4. Set a Samba password for your user:

```
$ sudo smbpasswd -a <USERNAME>
```

5. Restart the Samba service:

```
$ sudo systemctl restart smbd
```

6. Find your Raspberry Pi's IP address:

```
$ hostname -I
```

7. To access the shared folder from Windows, follow these steps:

- (a) Open **This PC**.
- (b) Right-click on an empty area and select **Add a network location**.
- (c) Proceed through the setup by clicking **Next** at each step until you reach the **Specify the location of your website** screen.
- (d) Enter the network path in the format:  
\\<RASPBERRY\_PI\_IP>\shared, then click **Next**.

## 5 Running Large Language Models with Ollama

Video: [Raspberry Pi 5 AI Setup Guide: Run DeepSeek, TinyLlama +more LOCALLY!](#) (Channel: [Wagner's TechTalk](#))

**Operating System:** Raspberry Pi OS Lite (64-bit), remote connection.

### Steps:

1. Prepare the system and install Ollama:

```
$ sudo apt update && sudo apt upgrade -y  
$ curl -fsSL https://ollama.com/install.sh | sh
```

2. Run the `tinylama` large language model:

```
$ ollama run tinylama
```

The first run takes more time as it downloads the model. Later runs start faster.

3. Interact with the model by typing a question and pressing **ENTER**. To exit, type:

```
$ /bye
```

### Optional Enhancements:

- Explore additional models on the [Ollama Model Search](#) page. For example, to launch DeepSeek (a lightweight option), use the following command:

```
$ ollama run deepseek-r1:1.5b
```

- Investigate Ollama commands and model management:

Use the following commands to interact with Ollama:

- `/show` – Display model details
- `/load <model>` – Load a model or session
- `/save <model>` – Save the current session
- `/clear` – Reset the session
- `/bye` – Exit Ollama
- `/help` – Display help

For model management, list installed models:

```
$ ollama list
```

Command for removing a model:

```
$ ollama rm <model>
```

Commands to uninstall Ollama:

```
$ sudo systemctl stop ollama  
$ sudo systemctl disable ollama  
$ sudo rm /etc/systemd/system/ollama.service
```

## 6 Setting Up Pi-hole for Ad Blocking

**Video:** [World's Greatest Pi-hole Tutorial - Easy Raspberry Pi Project!](#)  
(Channel: [Crosstalk Solutions](#))

**Operating System:** Raspberry Pi OS Lite (64-bit), remote connection.

### Steps:

1. Prepare the system and install the DHCP client daemon (needed for a static IP setup):

```
$ sudo apt update && sudo apt upgrade -y  
$ sudo apt install dhcpcd
```

2. Configure a static IP address:

```
$ sudo nano /etc/dhcpcd.conf
```

Add the following lines at the end of the file:

```
interface wlan0  
static ip_address=<RASPBERRY_PI_IP>/24  
static routers=<ROUTER_IP>  
static domain_name_servers=<DNS_1> <DNS_2>
```

If specific DNS servers are not preferred, you can use 8.8.8.8 (Google) for <DNS\_1> and 1.1.1.1 (Cloudflare) for <DNS\_2> as reliable default options.

Save the file and reboot the Raspberry Pi:

```
$ sudo reboot
```

3. Install Pi-hole using the automated script:

```
$ curl -sSL https://install.pi-hole.net | bash
```

Follow the installation prompts:

- **Pi-hole Automated Installer:** Select OK
- **Open Source Software:** Select OK
- **Static IP Needed:** Select Continue
- **Choose An Interface:** Select wlan0
- **Select Upstream DNS Provider:** Choose Google (or another preferred provider)
- **Blocklists:** Select Yes
- **Enable Logging:** Select Yes
- **Select a privacy mode for FTL:** Choose "Show everything"
- **Installation Complete:** Select OK

4. Access the Pi-hole web interface:

`https://<RASPBERRY_PI_IP>/admin`

5. Manually configure Raspberry Pi to use Pi-hole:

```
$ sudo nano /etc/resolv.conf
```

Replace the contents with:

```
nameserver <RASPBERRY_PI_IP>
```

Lock the file to prevent modifications:

```
$ sudo chattr +i /etc/resolv.conf
```

(To unlock the file for changes, use: `$ sudo chattr -i /etc/resolv.conf`)

6. Reboot the Raspberry Pi to apply changes:

```
$ sudo reboot
```

### **Optional Enhancements:**

- Manually configure a Windows device to use Pi-hole:
  - (a) Open **Network Connections**.
  - (b) Right-click on your Wi-Fi connection and select **Properties**.
  - (c) Select **Internet Protocol Version 4 (TCP/IPv4)** and click **Properties**.
  - (d) Choose **Use the following DNS server addresses** and enter:
    - Preferred DNS server: `<RASPBERRY_PI_IP>`
    - Alternate DNS server: Leave blank or use another DNS (e.g., 1.1.1.1)
  - (e) Click OK to apply changes.