

# Next.js Meets Raspberry Pi 5: A Cloud-Connected Camera Solution

A Practical Guide to Setting Up a Remote Camera with Raspberry Pi  
and Next.js

Umut SERIFLER - Senior Lead Engineer

# Presentation Contents

## Overview of Topics Covered

- Biography
- Raspberry Pi 5 Setup
- Packages & Tools Setup (VS Code | NodeJS | Python)
- Cloudflare Tunnel
- Python Application
- Web Application

# Biography

Umut SERIFLER - Senior Lead Engineer

- Education:

- **BSc in Electronic Engineering** - Istanbul Technical University (2009 - 2013)

- **MSc in Electronic Engineering** - Yildiz Technical University (2014 - 2017)

- Work Experience

- **Vestel** (2014 - 2018) | Türkiye - [vestelinternational.com](http://vestelinternational.com)

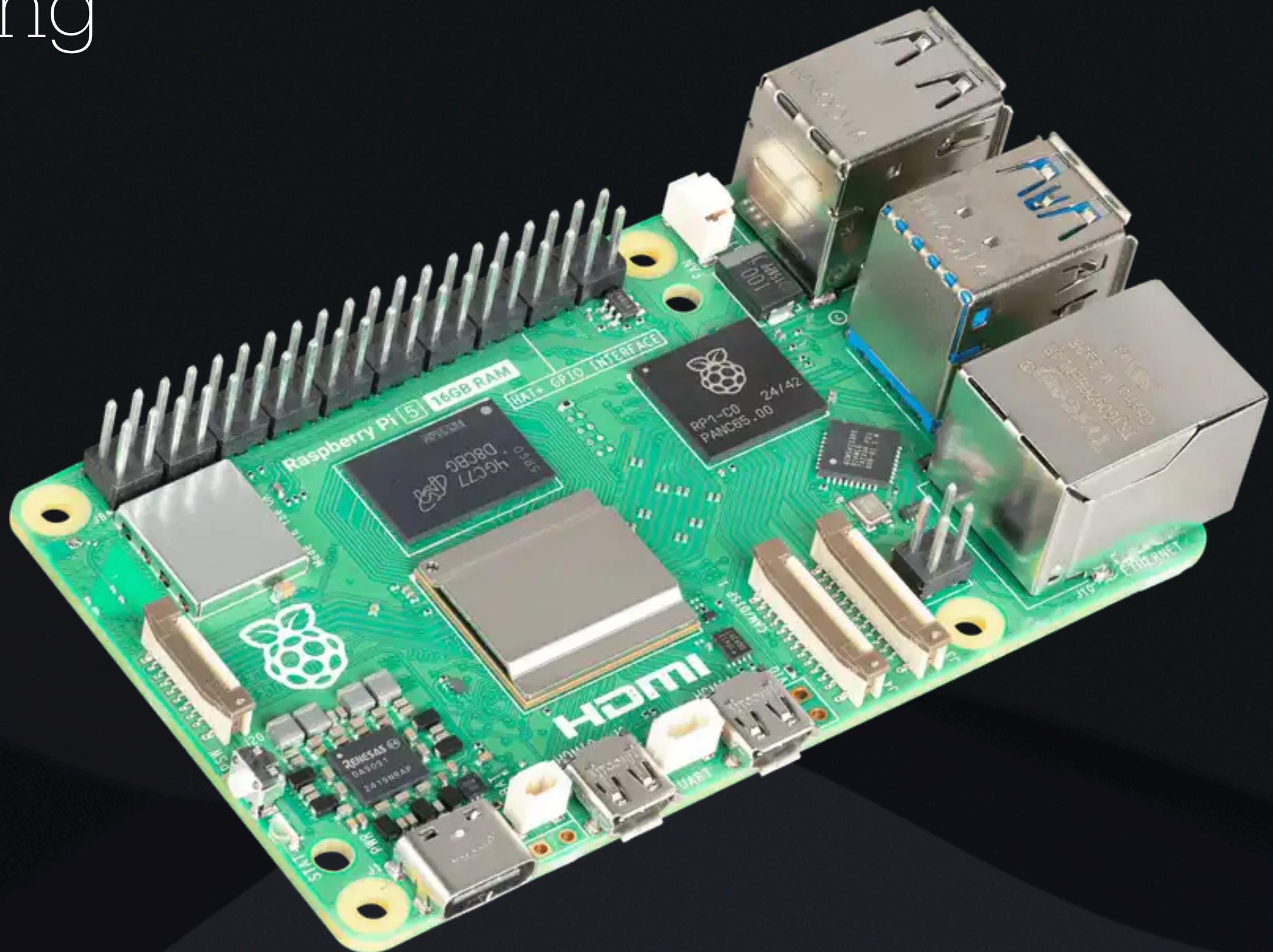
- **Salling Group** (2018 - 2021) | Denmark - [sallinggroup.com](http://sallinggroup.com)

- **On** (2021 - Current) | Germany - [on.com](http://on.com)

# Raspberry Pi 5 Build & Cost Breakdown

## Essential Components and Pricing

- Raspberry Pi 5 (8GB) ≈ 95 Euro
- Case + Active Cooler ≈ 23 Euro
- Camera Module V3 (12MP) ≈ 35 Euro
- SD Card (128GB) ≈ 25 Euro



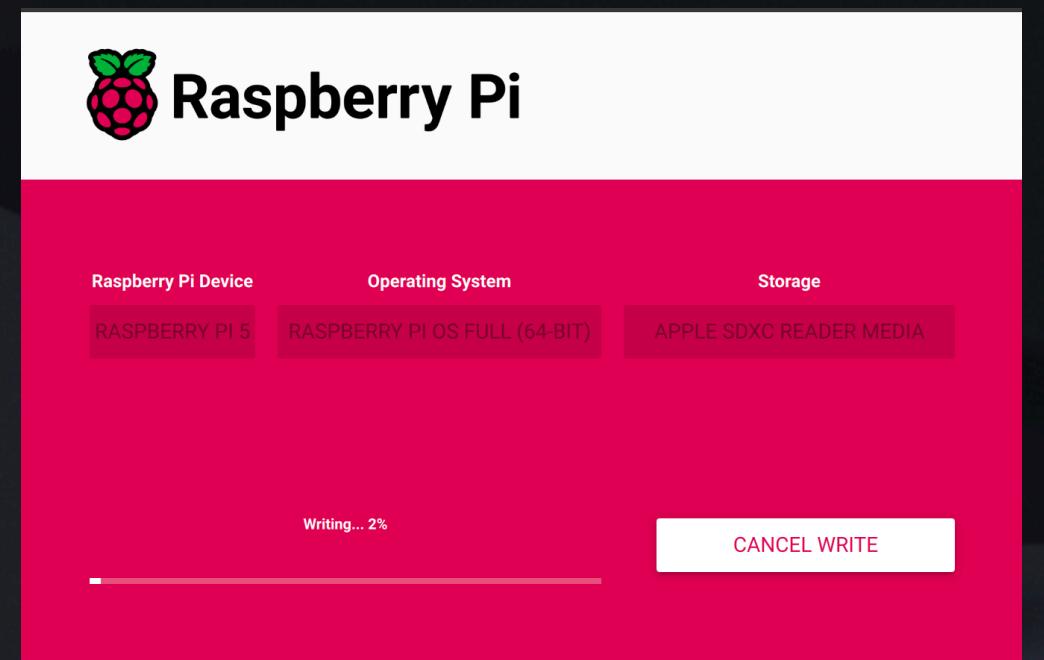
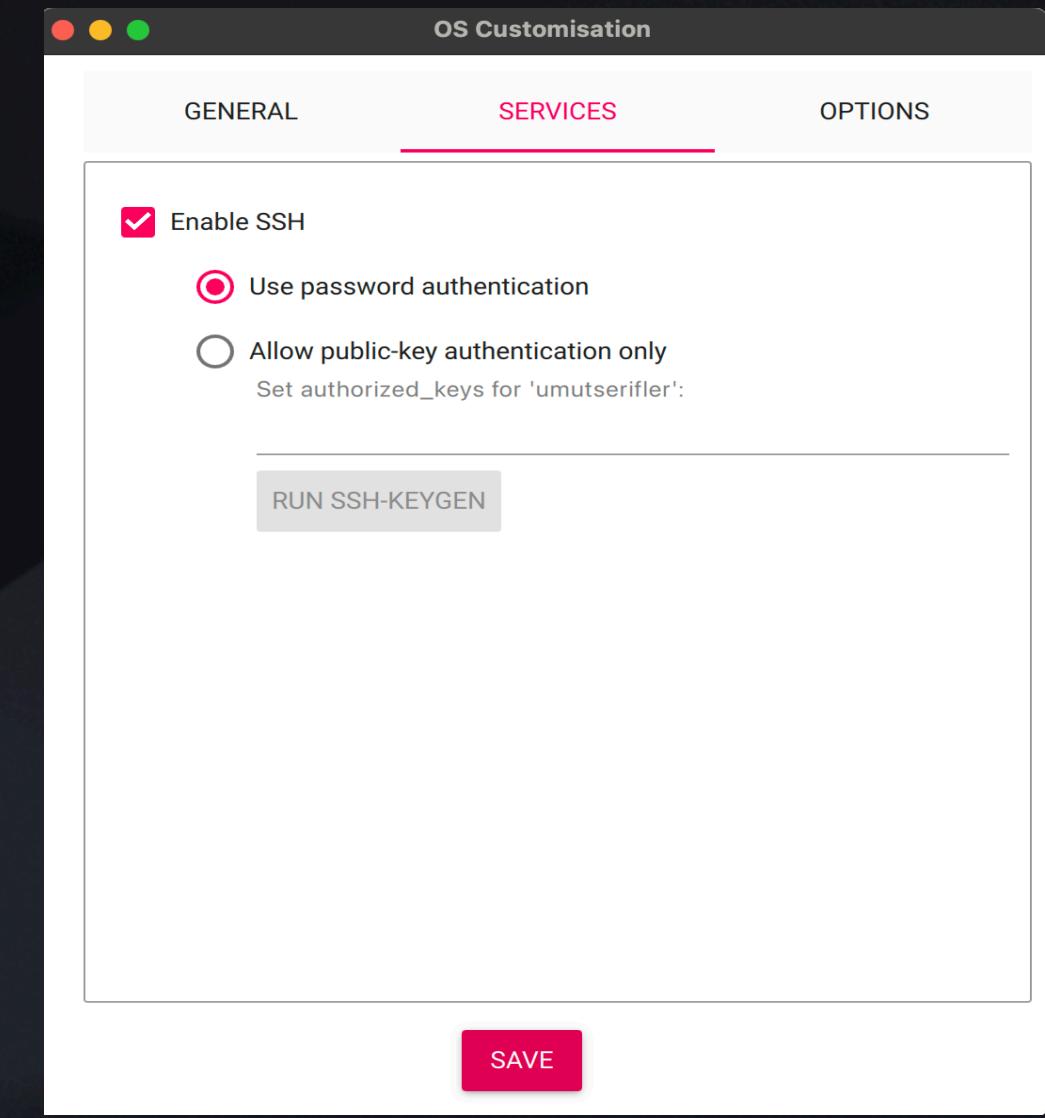
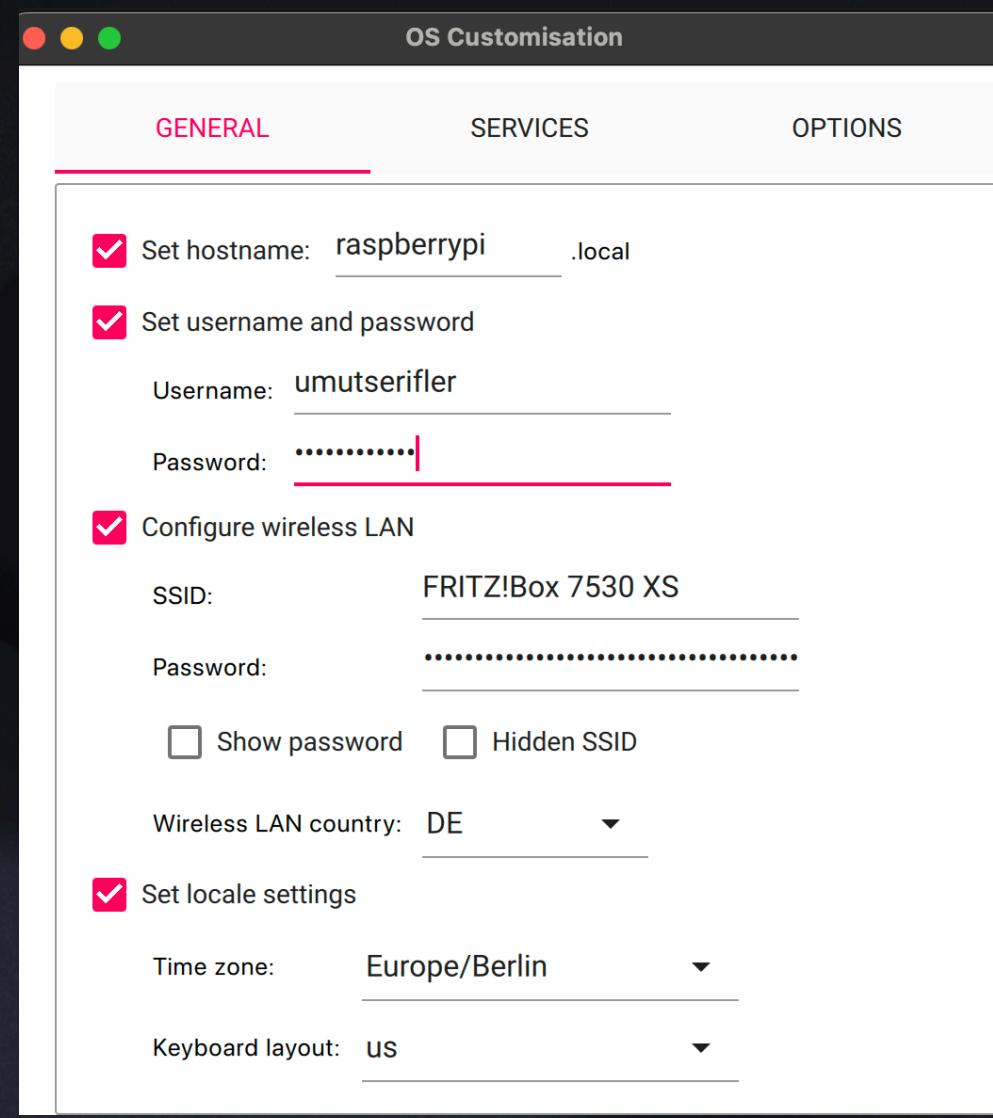
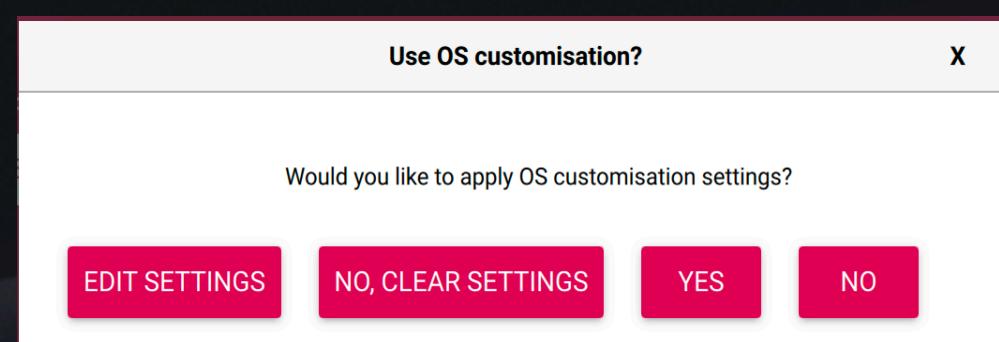
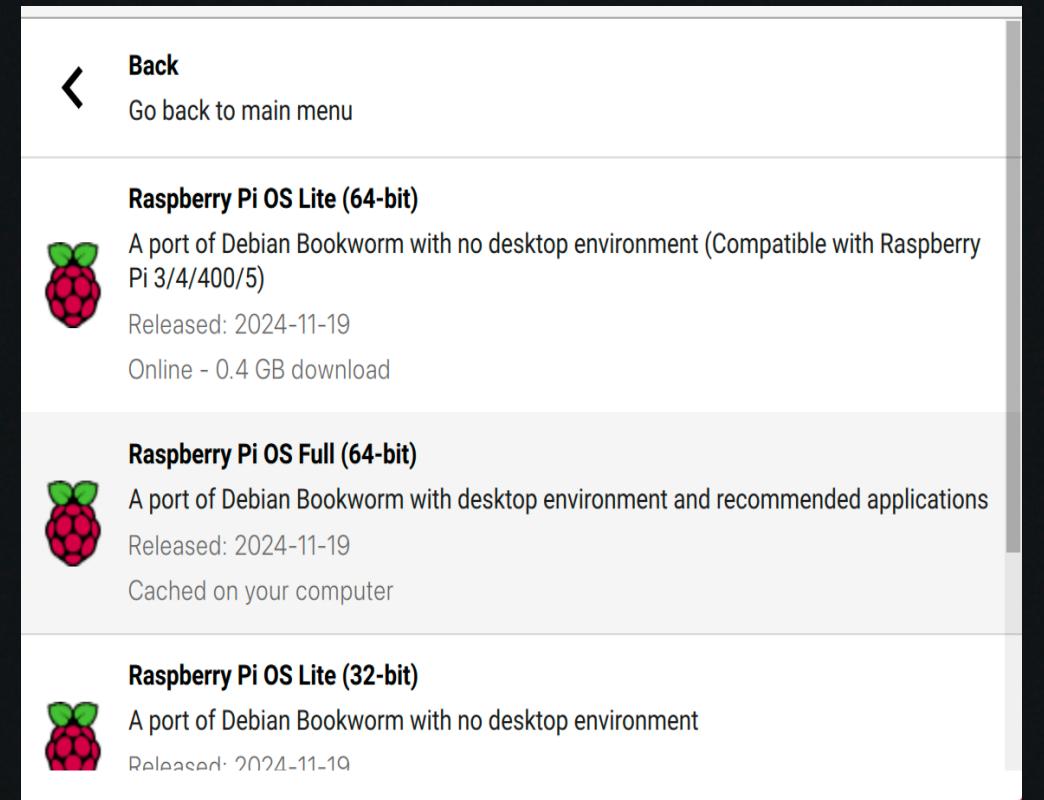
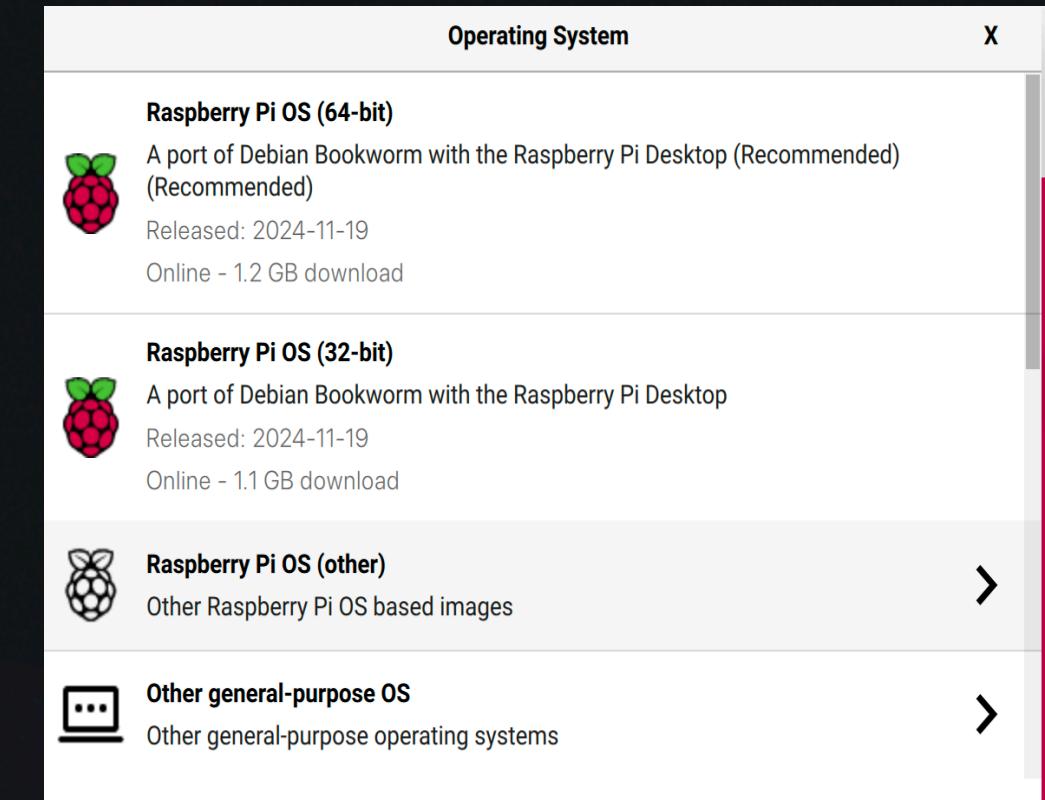
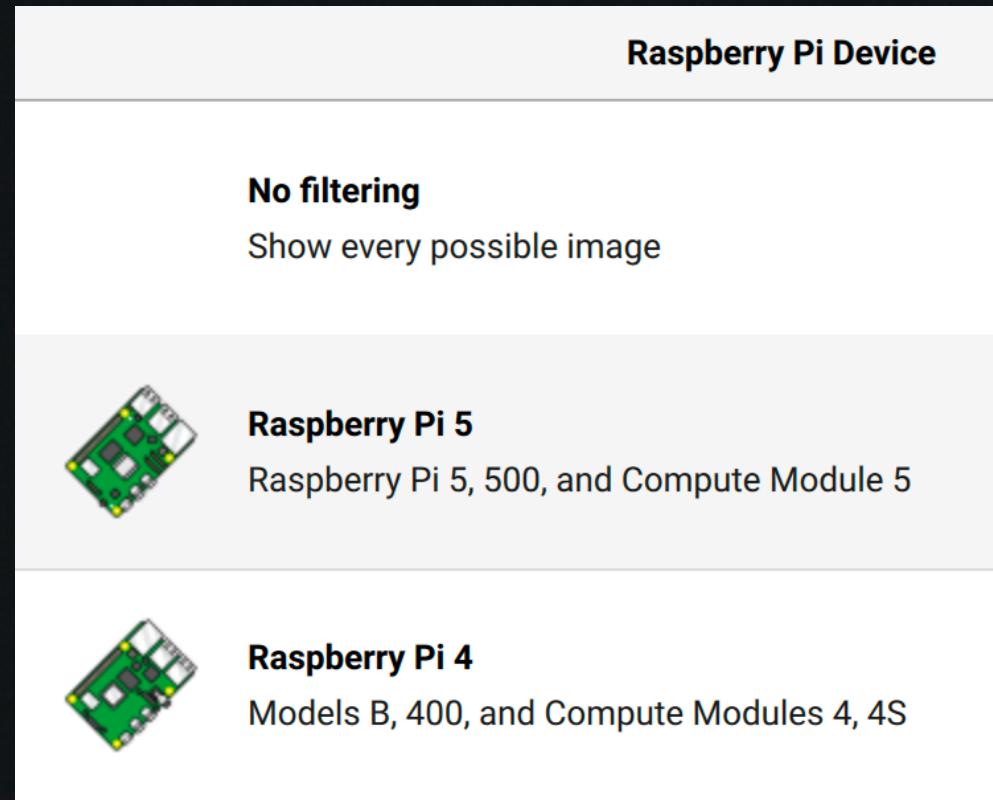
# Raspberry Pi 5 Specification

## Raspberry Pi

- Broadcom BCM2712 2.4GHz quad-core 64-bit Arm Cortex-A76 CPU, with cryptography extensions, 512KB per-core L2 caches and a 2MB shared L3 cache
- Dual 4Kp60 HDMI® display output with HDR support
- 4Kp60 HEVC decoder
- LPDDR4X-4267 SDRAM (2GB, 4GB, 8GB, and 16GB)
- Dual-band 802.11ac Wi-Fi®
- Bluetooth 5.0 / Bluetooth Low Energy (BLE)
- microSD card slot, with support for high-speed SDR104 mode
- 2 × USB 3.0 ports, supporting simultaneous 5Gbps operation
- 2 × USB 2.0 ports
- Gigabit Ethernet, with PoE+ support (requires separate PoE+ HAT)
- 2 × 4-lane MIPI camera/display transceivers
- PCIe 2.0 x1 interface for fast peripherals (requires separate M.2 HAT or other adapter)
- 5V/5A DC power via USB-C, with Power Delivery support
- Raspberry Pi standard 40-pin header
- Real-time clock (RTC), powered from external battery

# Raspberry Pi Setup

## Installation - SSH Access



# Setting Up SSH Key Authentication

## Connecting to Your Device via SSH in VS Code

- Once it is connected to wifi, it is ready to connect through SSH
- Install [Remote - SSH \(Microsoft\)](#)
- VS Code Ctrl+Shift+P -> Remote-SSH: Connect Current Window to Host
- ssh <username>@<hostname>.local => ssh [umutdemo@on.local](#)

# Update Packages

- sudo apt update;
- sudo apt -y upgrade;

# Installing Node.js Dependencies

<https://github.com/nodesource/distributions>

- sudo apt-get install -y curl
- curl -fsSL https://deb.nodesource.com/setup\_23.x -o nodesource\_setup.sh
- sudo -E bash nodesource\_setup.sh
- sudo apt-get install -y nodejs
- node -v (Verify)
- To update run: npm install -g npm@11.0.0

# Next.js Application

Create | Run Web Application

- `npx create-next-app@latest`
- Run Application - `npm run dev`
- Local Test - `http://localhost:3000/`

# Create a Tunnel

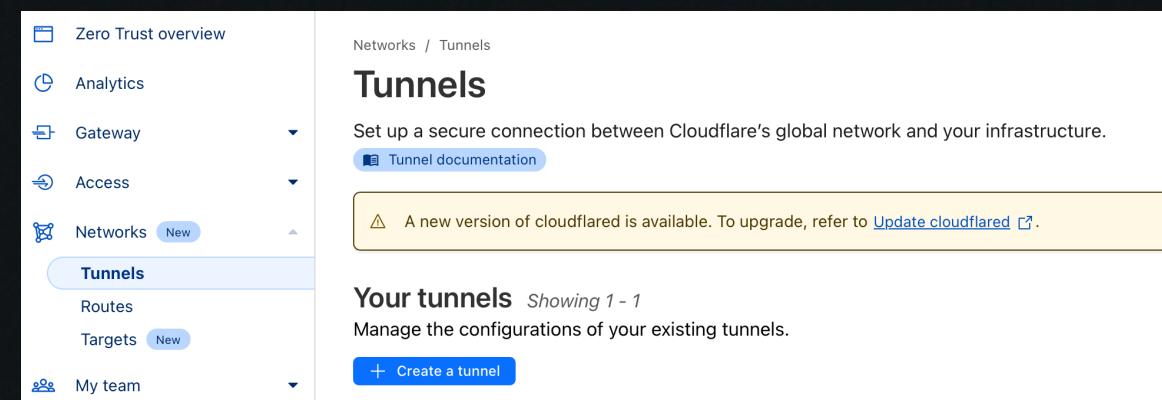
## Cloudflare Tunnel

- Multiple approaches to create a tunnel: ngrok, localhost.run, telebit, page kit, cloud flare..
- Why cloudflare:
  - Cost-Effective (Free Tier)
  - Flexible: Support using your own custom domains, set up DNS records
  - Easy of use: Straightforward installation
  - Scalable: Design to scale horizontally

# Make It Public

## Cloudflare Tunnel

- Zero Trust -> Networks -> Tunnels
- Create a Tunnel
- Install and Run Connector
- Test Connection



### Create a tunnel

Create a tunnel to connect HTTP web servers, SSH servers, remote desktops, and other protocols safely to Cloudflare.

[Learn more](#)

#### Select your tunnel type

Choose the method used to connect your resources to Cloudflare's global network.

 **Cloudflared**  
Recommended

Establishes a secure, outbound-only connection to Cloudflare for user-to-network connectivity.

[Select Cloudflared](#)

 **WARP Connector** Beta  
Linux distros only

Supports on-ramping and off-ramping traffic for site-to-site, bidirectional, and mesh networking connectivity.

[Select WARP Connector](#)

**Note:** WARP tunnel connections will require global configurations, which may affect other services currently in use.

# Make It Public

The image displays two side-by-side screenshots illustrating the process of making a local application public.

**Left Screenshot (Terminal View):**

- A dark-themed terminal window titled "bash - next\_application".
- The command `umutdemo@on:~/Documents/backend/next_application $` is visible at the bottom.
- The sidebar on the left lists various files and folders, including `.next`, `node_modules`, `public`, `src`, `.gitignore`, `eslint.config.mjs`, `next-env.d.ts`, `next.config.ts`, `package-lock.json`, `package.json`, `postcss.config.mjs`, `README.md`, `tailwind.config.ts`, `tsconfig.json`, `Downloads`, `Music`, `Pictures`, `Public`, `Templates`, `Videos`, `.bash_history`, `.bash_logout`, `.bashrc`, `.profile`, `.sudo_as_admin_successful`, `.Xauthority`, `.xsession-errors`, `.xsession-errors.old`, and `nodeSource_setup.sh`.

**Right Screenshot (Cloudflare Tunnels Configuration):**

- The URL is `one.dashboard.cloudflare.com`.
- The page title is "Configure".
- The sidebar on the left includes links for Zero Trust overview, Analytics, Gateway, Access, Networks (New), Tunnels (selected), Routes, Targets (New), My team, Logs, CASB, DLP, DEX, Email Security (New), and Settings.
- The main content area has a heading "Choose your environment" with operating system options: Windows (selected), Mac, Debian, Red Hat, and Docker.
- A section titled "Install and run a connector" provides instructions to copy and paste commands into a terminal window:

  - If you don't have cloudfaired installed: `brew install cloudfaired && sudo cloudfaired service install eyJhIjoiNT...`
  - If you already have cloudfaired installed: `$ sudo cloudfaired service install eyJhIjoiNT...`

- A note states: "Store your token carefully. This command includes a sensitive token that allows the connector to run. Anyone with access to this token will be able to run the tunnel."
- A link "View Frequently Asked Questions" is available.
- A "Connectors" section indicates "No connectors installed".
- Navigation buttons "Back" and "Next" are at the bottom right.

# Using WebSockets in React

## Installing and Setting Up react-use-websocket

- npm install react-use-websocket
- import useWebSocket from 'react-use-websocket';
- const { sendMessage, lastMessage } ...

# ShadCN UI Setup & Components

Button, Progress, Alert, Card Usages

- `npx shadcn@latest init -d`
- `npx shadcn@latest add button alert card progress`

Source: <https://ui.shadcn.com/docs/installation/next>

# Python Setup

## Python Application & Camera Module v3 Installation

- Install Camera Dependency:

```
sudo apt-get install python3-picamera2
```

- Create a python app:

```
mkdir camera-app
```

```
cd camera-app/
```

```
python3 -m venv --system-site-packages venv
```

```
source venv/bin/activate
```

```
touch main.py
```

# Python Setup

## Python Application & Camera Module v3 Installation

```
from picamera2 import Picamera2, Preview  
picam2 = Picamera2()  
picam2.start_preview(Preview.NULL)  
picam2.start_and_capture_file("test.jpg")
```

Samples:

[C++ Samples](#), [Python Samples](#)

Documentation: [https://www.raspberrypi.com/documentation/computers/camera\\_software.html](https://www.raspberrypi.com/documentation/computers/camera_software.html)

# Using Websocket with Python

## FastApi - Websocket

- Create Camera App Folder
- Install FastApi: pip install "fastapi[standard]"
- Activate venv (Virtual Environment): source venv/bin/activate
- FastApi Websocket
- Run App: fastapi dev --host 0.0.0 main.py

# Live Demo

# Q&A



# Troubleshooting

## Common Errors and Fixes for Installation & Execution

- When to install Cloudflare tunnel gives an error

1. Check for Running Instances

```
sudo systemctl stop cloudflared
```

```
sudo systemctl disable cloudflared
```

2. Delete / Install Cloudflare

```
sudo cloudflared service uninstall
```

installation command can be seen in page 13

- Running Python Application error:

Check if venv is activated or not  
source venv/bin/activate

Re-try running application: fastapi dev --host 0.0.0.0 main\_websocket.py