Creating my first home server

Link: https://medium.com/@tararhoseyn/creating-my-first-home-server-2104cc610514

As someone who works in software and product, I've been learning more about the hardware/infrastructure side of computing in recent months. Even if the knowledge doesn't *directly* impact my day-to-day, I think having a decent understanding of general computing is a helpful skill set to have in my line of work.

I decided that a good place to explore would be to try to make a budget-friendly server for my home.

My first thought was to use a Raspberry Pi Zero 2 W as a Pi-hole server. This did technically work, but it was slow, clunky, and couldn't really run any other services. It didn't make a lot of sense to me to have an entire machine (even one as small as a Pi Zero) for just one service.

I then looked at a broken laptop I had laying around, an old HP notebook g5 250 with an i5 inside. I hadn't used it in years for a couple of reasons: the battery died, the second replacement battery died and the hard drive died. It wasn't looking too healthy. But if I could get it up and running, it would be a perfect low-energy server to run multiple services on. So I set about to fix it!

Fixing the laptop

Replacing the hard drive

Luckily for me, I had another dead laptop around, an Acer V5 (I live in a graveyard of tech I can't bring myself to throw away, I'm sorry). I managed to pull the old 2.5" SATA drive out of the Acer and popped it into the HP. This ended up bringing another use case into my server build... I found a bunch of old films on the drive, backed them up, and will use them in my server build (more on this later!)

Replacing Windows with Ubuntu

This is going to be a controversial take... The drive I popped into the HP had Windows 8. Now, I could have updated the Windows operating system (OS) or kept version 8, but I wanted a lightweight OS for a laptop that had aging hardware and only a 750GB hard drive. Especially as one hard drive is used as both storage and as an OS boot. So I decided to go down the Linux route. I chose Ubuntu as it seemed like a good beginner's choice as I've not used Linux before.

It was a simple process. I downloaded the ISO file for Ubuntu, flashed it to a USB drive using Rufus, popped the USB drive into the laptop, booted into BIOS on the laptop, and installed the OS.

The techy folks among you may wonder why I went for Ubuntu Desktop over Ubuntu Server if I was going down the Ubuntu route for a server anyway. The simple answer is that this is my first rodeo, and I wanted a GUI for ease of setup and maintenance. I once learnt a valuable acronym from one of my software engineering lecturers and often repeat it to myself: KISS. Keep it simple, stupid.

Scrapping the battery altogether

This is a bit of a risky move, but replacing the battery with an official HP one (I previously bought a cheap third-party battery, and it went down like a lead balloon) was out of my budget for this build. I wanted to use what I already had, and I plan to keep this server running 24/7 anyway so I just kept using power from the wall.

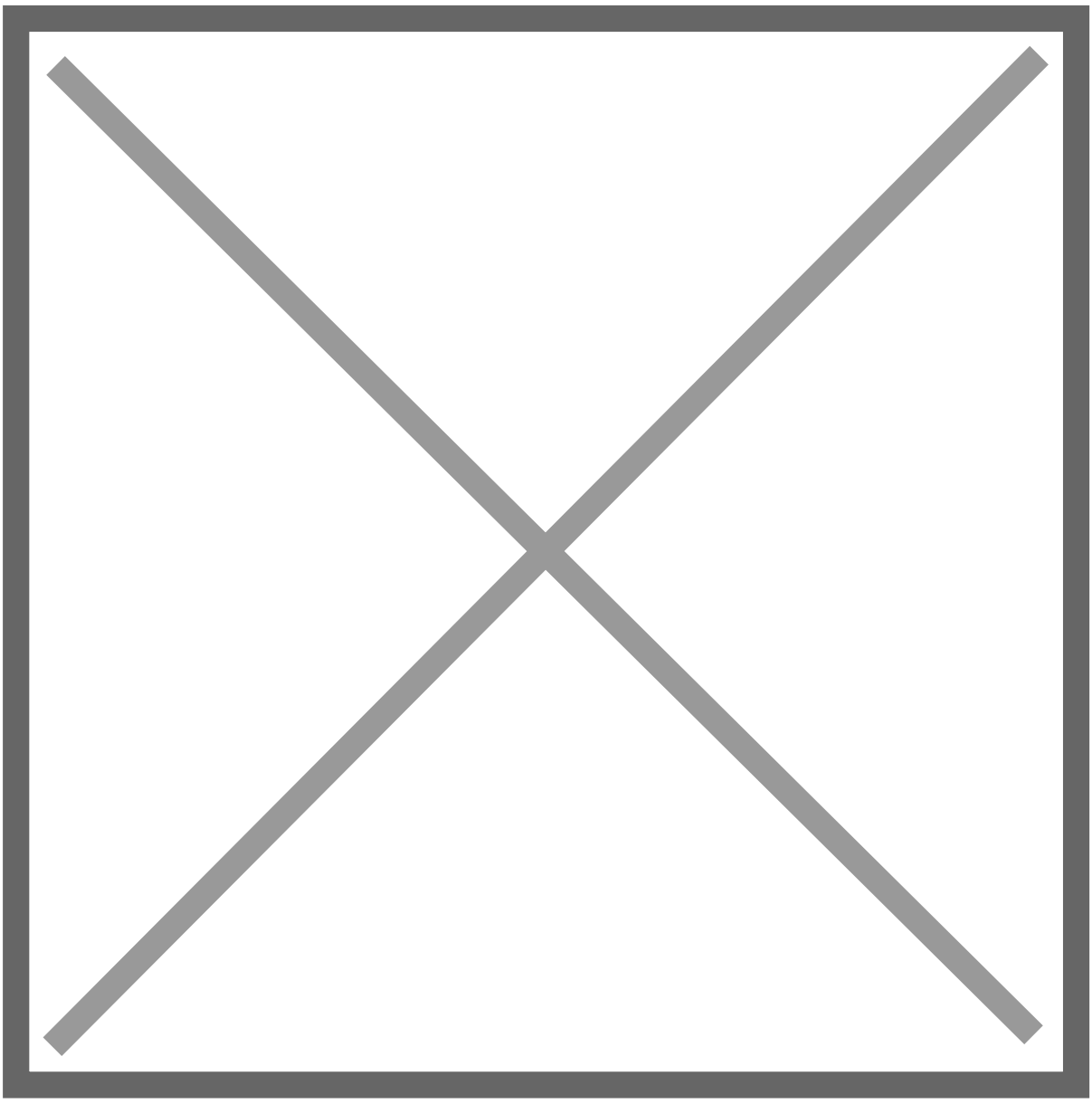
Setting up the server

When I was researching small server setups, I came across a product called a Zima Board and its GUI system, CasaOS, an open-source software that can easily run on Linux.

Casa is the perfect beginner's introduction to server management. After getting my laptop's configuration set up for 24/7 power, I installed Casa in the terminal (Side note: The ease of the terminal and its commands in Linux is super impressive compared to Windows).

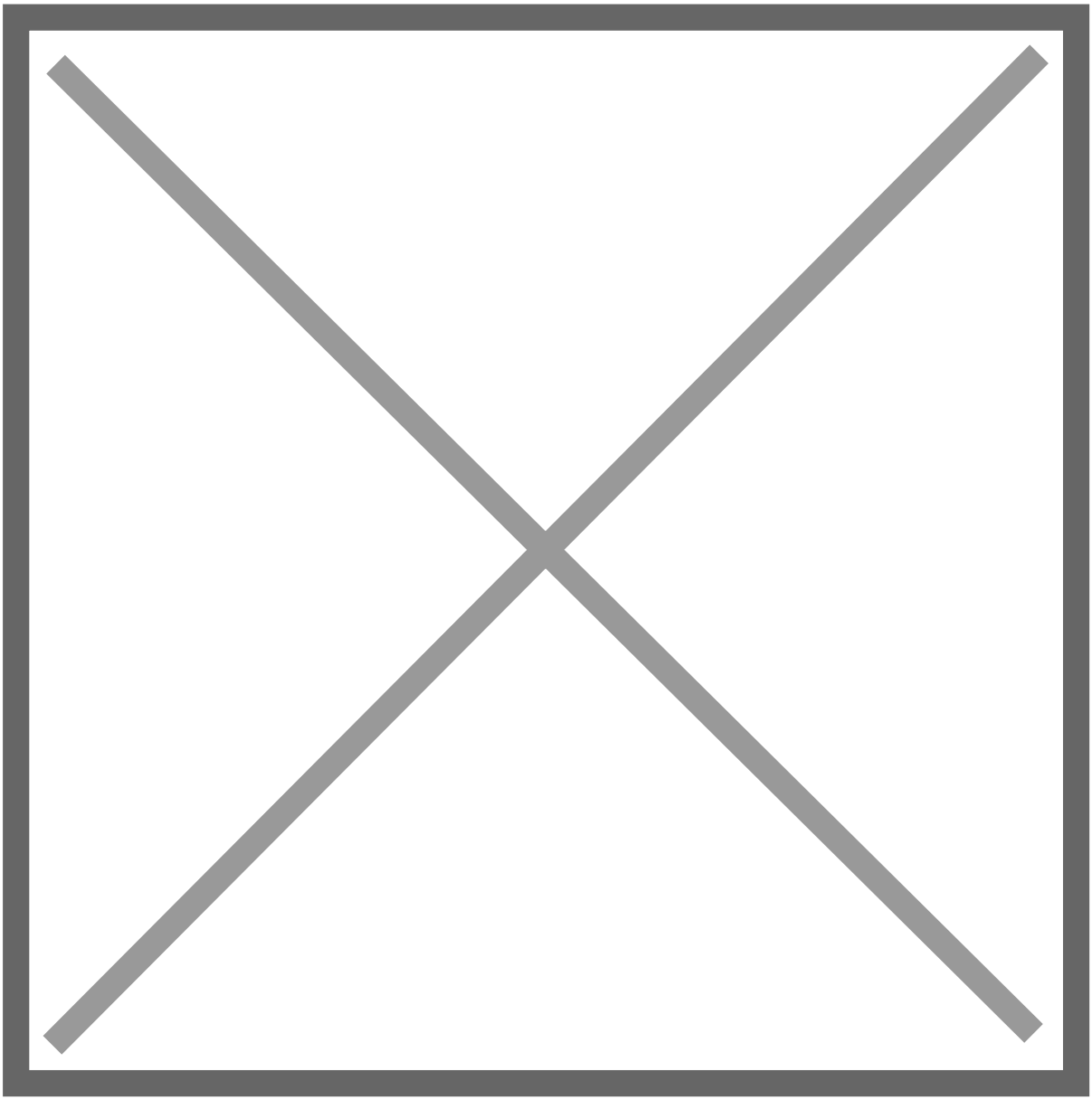
I had a bit of trouble at first as the installation of Casa is supposed to let you know at the end what IP address you can use to access it. This never came up for me in the terminal for some reason, but luckily I ended up finding the IP in my router's event log.

Now Casa was installed and running on my system, all I had to do was pick which apps to download and research others. Casa is basically a visualization of Docker containers. I've had some basic experience with Docker before, and I love how lightweight and simple it is to run services through. Casa makes it even easier.



My server dashboard

I also appreciate how sleek and beginner-friendly the dashboard of Casa is. And because it's hosted on its own IP, I can access the dashboard (and by extension, all the services running on the server) on any device in my home connected to the internet. Once I set up the system, I accessed the IP on my smart TV and did the rest of the setup from the comfort of my sofa!



My dashboard server on my lounge TV

What services I’m running

- **Pi-hole** — a network-wide ad blocker.
- **Jellyfin** — a media player that offers a Netflix-like experience for your own media files. This has been an absolute game-changer for those old film files I had on the hard drive. I’m really pleased I can use this system as a media server through this service.
- **qBittorent** — for torrenting files.
- **Wireguard** — this is a VPN protocol, but I’ve had trouble with its configuration so I need to give this another shot.

Future improvements

- Definitely reconfigure/find an alternative to Wireguard.
- I’m stuck between increasing the storage on this device or getting a separate NAS for storage.
- Exploring the world of Docker more. While Casa comes with a bunch of ‘apps’ that easily set up Docker containers for you, you can do custom installs of any Docker services too.
- I’d love to host a website on the server.

Revision #1
Created 2025-02-23 12:36:04 UTC by Administrador
Updated 2025-02-23 12:38:46 UTC by Administrador