

The Setup

Fayza

September 2026

Abstract

Nothing here was bought to be a server. There is a small rented VPS, and everything else is a computer that had already stopped being useful for what it was sold as — a laptop too old for daily work, single-board computers bought for something else, and a drawer of external drives accumulated over years. This describes what each one does, how they fit together, and the reasoning that put each service where it is. Names, addresses and anything else specific to my house are deliberately left out.

Contents

1	The idea	1
2	The machines	2
3	The network	2
4	What runs where	3
4.1	On the rented machine	3
4.2	On the server at home	3
4.3	On the media box	3
4.4	On the backup box	3
5	Storage, and how it got sorted	3
5.1	The pool I did not build	4
6	Media, from download to screen	4
6.1	No media server	4
7	Backups	5
8	Five things worth stealing	5
9	What it costs	6
10	Closing	6

1 The idea

Self-hosting write-ups usually start with a hardware purchase. This one starts with a cupboard.

The premise is that most households already contain enough computing to run the services people pay subscriptions for — photo storage, document archiving, notes, media, backups — and that

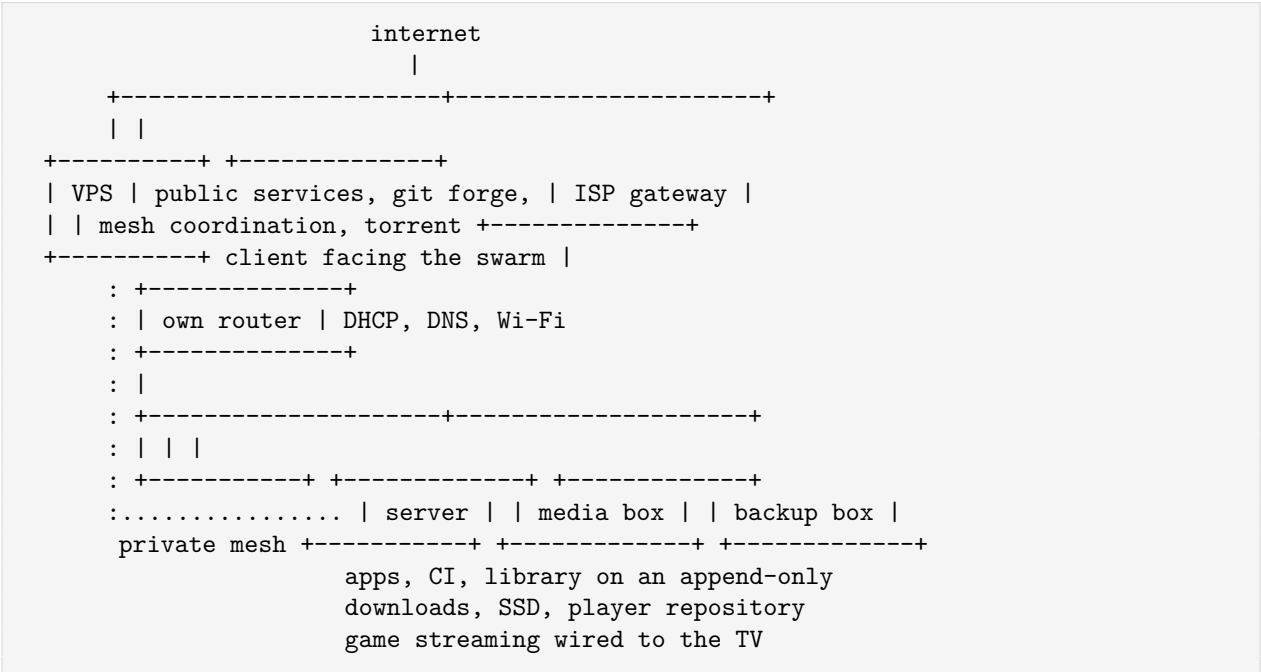
the interesting constraint is not performance but arrangement. Which machine should hold which job, given that none of them were chosen for the job.

Two things follow from that. The first is that every decision is about matching a workload to hardware that already exists rather than buying hardware to match a workload. The second is that reliability has to come from the design, because it is not coming from the components.

2 The machines

Machine	What it was	What it does now
A small VPS	rented, 2 vCPU	everything public-facing
The server	a 2017 mobile workstation	applications, CI, downloads, games
The media box	a small single-board computer	the media library and the player
The backup box	an older single-board computer	an append-only backup store
A workstation	a desktop	where the work happens

Plus five external USB disks of assorted ages, sizes and health, which turn out to matter more to the design than any of the computers do.



The dotted line is a private WireGuard mesh with a self-hosted coordination server. It is the only path between the rented machine and the house, and it means nothing at home needs a port forwarded to the internet.

3 The network

The ISP’s gateway does not expose DNS settings, which is a common enough restriction and an annoying one — it makes network-wide ad filtering impossible without touching every device individually.

The fix is a second router behind it, running DHCP, DNS and the wireless. The gateway keeps doing upstream and nothing else. This is double NAT, which is normally a reason not to do it, but here there is nothing to forward inward: everything that reaches the house arrives through the mesh, and the publicly-reachable services live on the rented machine. The usual objection costs nothing when there is no inbound traffic to break.

Local DNS then does two useful things beyond filtering. It gives the machines names rather than addresses, and it resolves the site's own domain to its local address for devices inside the house, which is otherwise a well-known nuisance when a router will not route a public address back to a machine behind it.

4 What runs where

4.1 On the rented machine

Everything that has to be reachable from the internet, and nothing else:

- A reverse proxy terminating TLS.
- A git forge with a container registry.
- The mesh coordination server.
- This site.
- A torrent client, which is the interesting one — see below.

It has 40 GB of disk, most of it spoken for. That constraint shaped more of the design than anything else.

4.2 On the server at home

The eight-year-old laptop does the work that needs a real CPU:

- Self-hosted photo, document and note services, in containers.
- A CI runner for the forge, building against a Docker daemon deliberately kept separate from the machine's own so that a careless workflow cannot delete the services it is running beside.
- The download and seeding disk.
- Hardware-encoded game streaming, which is what the discrete GPU is for.

4.3 On the media box

An SSD holding the media library, and a player reading it directly. No media server anywhere in the design — more on why below.

4.4 On the backup box

One disk and one service: a restic REST server in append-only mode. It does nothing else, which is the point.

5 Storage, and how it got sorted

Five external disks accumulated with no plan: a portable SSD, two portable hard drives, a laptop drive in an optical-bay enclosure modified years ago, and one more in a generic bridge still formatted NTFS from a previous life. Media was spread across three of them with the same categories in different places.

Consolidating that took a weekend and two rules did most of the work.

Match the access pattern, not the capacity. Streaming a film is sequential reads at a few megabytes a second, which spinning disks have always done well. Torrent traffic is small random reads and writes across many files; game loading is latency-sensitive. So the solid state storage went to the jobs that seek, and the media library went to the disk with the most room — which is the opposite of how I would have allocated by instinct.

Put the most disposable data on the least reliable disk. The oldest drive became the download and seeding lane. In-progress downloads and copies being seeded are the data whose loss costs least, and it is also the most write-intensive job, which is to say the one most likely to kill a drive. If something fails first, it should be that.

5.1 The pool I did not build

The plan was to pool the leftover disks into one large volume. Two things stopped it.

The first was finding a leftover LVM volume group from an old installation: two physical volumes, one of them long since unplugged, and an 800 GB logical volume spanning both. Not part of it was unavailable — all of it was. A linear span is one filesystem laid across several disks, and losing any member loses the filesystem. I had been about to build exactly that across three disks, one of them a drive from 2012.

The second was noticing that I did not need a pool at all. Every media player worth using accepts several folders per library and presents them as one collection. A union filesystem would have been solving a problem the software above it does not have.

So there is no pool. There is one disk holding the library, and the others sit in a drawer still holding the copies they held before the consolidation — a cold backup of most of the media, obtained by not doing work rather than by doing it.

6 Media, from download to screen

```
VPS server media box
-----
torrent client ---> download disk ---> library SSD
faces the swarm keeps seeding copy player reads it
holds no data until ratio met from local disk

                                |
                                HDMI
                                v
                                television
```

The client runs on the rented machine so that its address faces the swarm rather than mine, but the rented machine has no room to store anything. So its save path is a network mount of a disk at home: pieces are written across the mesh as they arrive, and seeding reads them back the same way. What is stored on the VPS is a configuration directory.

The automation stack then copies completed items to the media box, and the client removes its own copy once a share ratio is met. Two copies exist in between, which is the price of the storage living on a different machine from the library.

6.1 No media server

The obvious design has a media server indexing the library and streaming it to clients. I removed it, for two reasons.

If the storage is on one machine and the server on another, every byte crosses the network twice — once to the server, once to the client. Put the player on the same machine as the storage and it is worse still, because the file leaves and comes back.

And the box at the television already runs a player that reads local files. If the library lives on that box, playback touches no network at all, needs no server, and cannot transcode because there

is nothing in the path to transcode with. It is the shortest route from file to screen, and it means the server can reboot in the middle of a film.

A media server earns its place when there are clients that cannot read the files directly. There are not, so there is no media server.

7 Backups

```
live data local backup offsite
-----
server SSD ---> backup box ---> a disk that
photos, docs, nightly append-only monthly leaves the house
notes repository
```

Roughly 60 GB is genuinely irreplaceable: photographs, scanned documents, notes. Media is explicitly excluded — it is re-acquirable, and treating it as precious would make the backup ten times larger for no gain.

Three properties are worth pointing out.

The backup lives on a different machine. A backup on a disk attached to the machine it protects dies with that machine. Separation is most of the value.

Append-only. The server can write new snapshots but cannot delete them; pruning happens on the backup machine, deliberately. An accident or a runaway process on the machine being backed up cannot propagate into the backup history.

The databases are stopped first. A hot copy of a running Postgres directory or an in-use SQLite file looks perfectly fine and will not restore. Two or three minutes of downtime at four in the morning buys a backup that works.

8 Five things worth stealing

An immutable mount point. With the disk unmounted, `chattr +i` the directory it mounts on. It can still be mounted over, but nothing can create files inside it while it is empty. A container whose bind source cannot be created fails to start loudly, instead of starting successfully against an empty directory and quietly rewriting a database. One command, never needs maintaining, and it removes an entire class of silent failures.

Mount by label, never by device letter. Letters move when you plug or unplug anything. A media library whose path changes is a media library that reports its files as deleted, and takes the watch history with them.

The enclosure is a component. An external drive is a disk and a bridge chip and a power path. When one misbehaves — resets, disappearances, poor throughput — swap the enclosure before condemning the disk. In this setup the worst-behaved drive turned out to be fine; its enclosure was a USB 2 bridge that could not reliably spin it up.

Give the most disposable job to the least trustworthy hardware. It is the cheapest form of risk management available, and it costs nothing.

Check that things are still running. The service that started this whole exercise had been failing at every boot for weeks, and a drive had been resetting three times per boot for longer, and nothing said so. Monitoring is worth more than another disk.

9 What it costs

Cost	Detail
Throughput	torrent speed is bounded by the home line, not the VPS
Two copies	downloads exist twice until the seeding copy is pruned
No remote media	no server means no watching from elsewhere
Old hardware	every component is past the age one would choose
Time	this is a hobby that occasionally looks like an outage

None of this is cheaper than a subscription if an hour of your time has value. It is cheaper in money, more interesting, and the data stays in the house. Those are the actual reasons.

10 Closing

The recurring lesson has been subtraction. Almost every part I planned and then removed — the storage pool, the union filesystem, the media server, an extra storage host — existed to manage capacity or complexity I had not accumulated. Sizing the design to the real numbers rather than to the shape of the problem as first imagined left most of the machinery with nothing to do.

What remains is five reused computers, a rented one doing the part that must face outward, and a drawer of old disks that turned out to be a backup. It runs the things a household actually uses, on hardware that would otherwise be waste.