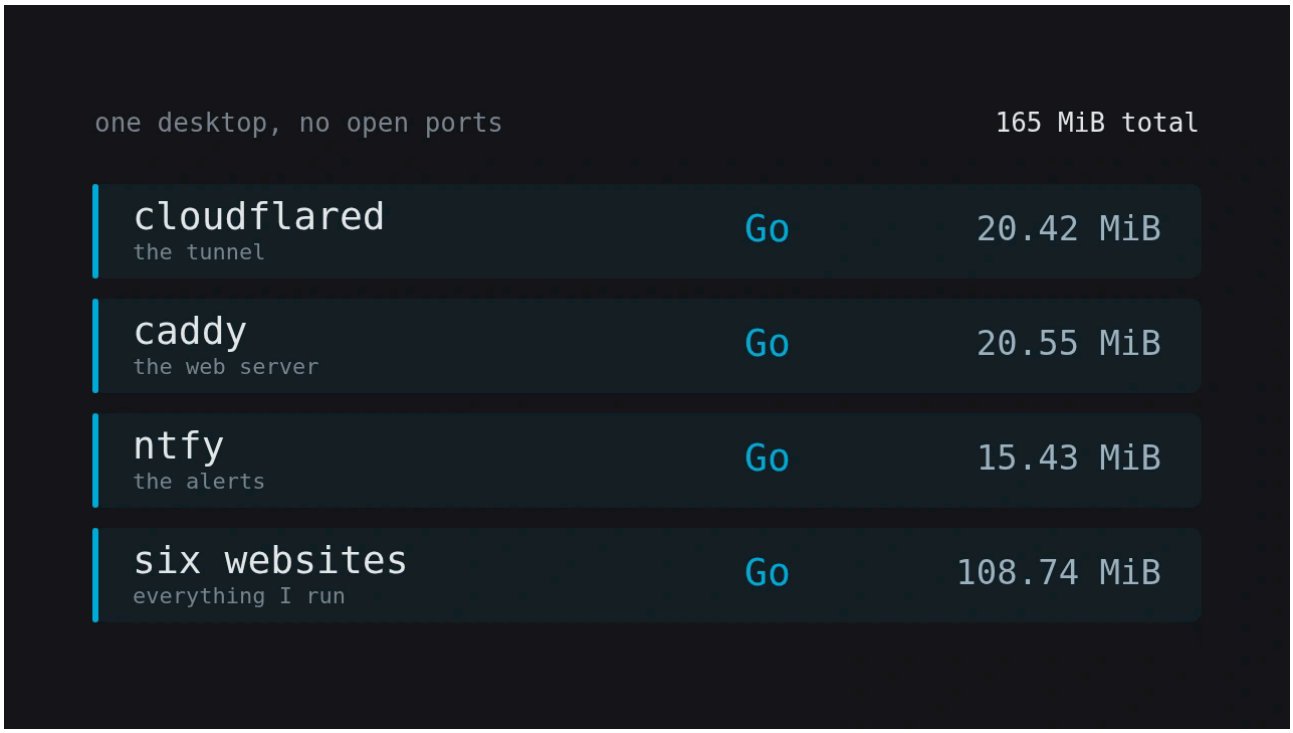
It's Go all the way down

Isaac Bythewood · 2026-08-30 · 4 min read

go sysadmin webdev



Every process in front of my websites is a Go binary now, from the tunnel to the web server to the sites themselves, and the whole stack runs on a desktop I already owned for a few dollars a month in power.

I cancelled my Linode this week and moved every site I still run onto the desktop sitting next to me, behind a Cloudflare Tunnel. I did it for the bill and to get my home IP out of DNS, but once everything was up and I ran `docker stats` for the first time I noticed something I hadn't planned on. Every single process in the request path is a Go binary now.

Here's the whole stack, from the outside in:

- `cloudflared` holds the tunnel open to Cloudflare, and it's written in Go.
- `caddy` takes the plain HTTP the tunnel hands it and reverse proxies to each app, and it's written in Go.
- `ntfy` pushes alerts to my phone when something falls over, and it's written in Go.
- The six sites are all Go, since I spent the last week rewriting them off Rust and Next.js.

I didn't plan any of that. cloudflared came with Cloudflare, I picked Caddy back in 2022 because its config file is short, and I found ntfy while looking for something that could push to my phone without an account. They all just happen to be Go, which I think says something about what Go turned out to be good at.

Here's `docker stats` on the running stack, all nine containers:

```
NAME                    MEM USAGE   CPU %
orchard-cloudflared     20.42MiB    0.24%
orchard-caddy           20.55MiB    0.00%
orchard-ntfy            15.43MiB    0.00%
orchard-analytics       33.35MiB    0.01%
```

```
orchard-repos            21.53MiB    0.00%
orchard-status           20.00MiB    0.00%
orchard-logging          13.47MiB    0.07%
orchard-blog             13.15MiB    0.00%
orchard-isaacbythewood    7.24MiB    0.00%
```

That's about 165 MB for six websites, a tunnel, a web server, and an alerting service. The whole thing fits in a third of the 512MB Linode I was paying for, and CPU sits at zero almost all the time since I get roughly a thousand page views a month and a Go binary rendering a page is a fraction of a millisecond of work. My portfolio is the smallest at 7 MB, which is about what the process costs just to exist.

The images came out small too. Two of them build `FROM scratch`, so the container holds one static binary and nothing else at all, and `isaacbythewood.com` is 24 MB with every asset embedded in it. The rest sit on Alpine because they shell out to typst to render PDFs, and status is 1.5 GB because it bundles Chromium for Lighthouse audits, so not everything here is small.

The part I keep being surprised by is that none of this has an open port. Nothing is published to the host, so `docker ps` shows an empty ports column straight down. cloudflared makes an outbound connection to Cloudflare and holds it open, and requests come back down that same connection. I don't have a port forward on my router or a dynamic DNS client running, there's no inbound firewall rule for me to get wrong, and nothing on my home network is reachable from the internet. Every hostname resolves to Cloudflare anycast, so my residential IP isn't sitting in a DNS record either. If my ISP moved me behind CGNAT tomorrow this would carry on working.

The bill is what I actually did this for. The Linode plus its backups came to something like $25 a month and that's gone now. What replaced it is a machine that was already running 12 to 16 hours a day for everything else I do, and the containers add close to nothing on top. If I charge the entire idle desktop to it at North Carolina power prices it comes out around $3 a month, and realistically it's less than that since I'd have the machine on anyway.

I'm old enough to remember when self hosting meant a static IP or a dynamic DNS client, a port forward, a certificate you renewed by hand, and a decent chance of putting your home network on the internet by accident. Cloudflare gives away the tunnel, the certificates, the anycast, and a free plan with enough cache rules to be useful, and setting it up was one `cloudflared tunnel login` in a browser plus a CNAME for each hostname. I'm not sure how I feel about routing this much of my stuff through one company, but what they're handing out for free used to be most of the work.

I also wrote about rewriting this blog in Rust a few months back, and it's Go now, so take my language opinions with the appropriate amount of salt. Six sites on 165 MB with nothing listening, and I'm paying a few dollars a month in power for it.