Zero downtime deploys with Caddy

Isaac Bythewood · 2026-10-03 · 2 min read

sysadmin webdev



One line in Caddy holds requests while a Docker container is replaced, so a deploy looks like a slow page instead of a 502.

If you run your sites with Docker Compose behind Caddy then a deploy with `docker compose up -d --build` stops the old container before it starts the new one. For a moment there's nothing listening and Caddy answers anyone who shows up with a 502. For a small app that's maybe a second but anything that takes a while to shut down, like an app holding open server-sent event streams, can leave you with 502s for 10 or 15 seconds.

Simon Willison wrote a TIL on this back in 2021 after Matt Holt pointed him at `lb_try_duration`. It's meant for load balancing across several upstreams but it works fine with just one. When Caddy can't reach the upstream it keeps retrying for that long instead of giving up, so the request waits on the new container and then gets its page:

```
example.com {
  reverse_proxy app:8000 {
    lb_try_duration 40s
  }
}
```

You might worry about a POST being sent twice but Caddy's defaults already handle it. A failed dial is retried for any method since nothing reached the app, and a request that fails after it reached the app is only retried if it's a GET.

I went with 40 seconds since Docker's `stop_grace_period` is 30 seconds on my containers, so a replace can't take longer than that. The downside is that if a site really does crash a visitor waits 40 seconds before they see the 502, which I'm fine with for a handful of personal sites.

That’s it, one line of config and deploys don’t drop requests anymore.