Our goal is to set up a Docker orchestration environment by installing Coolify on Proxmox. A report needs to be prepared and presented to determine whether Coolify is sufficient for this purpose or not.

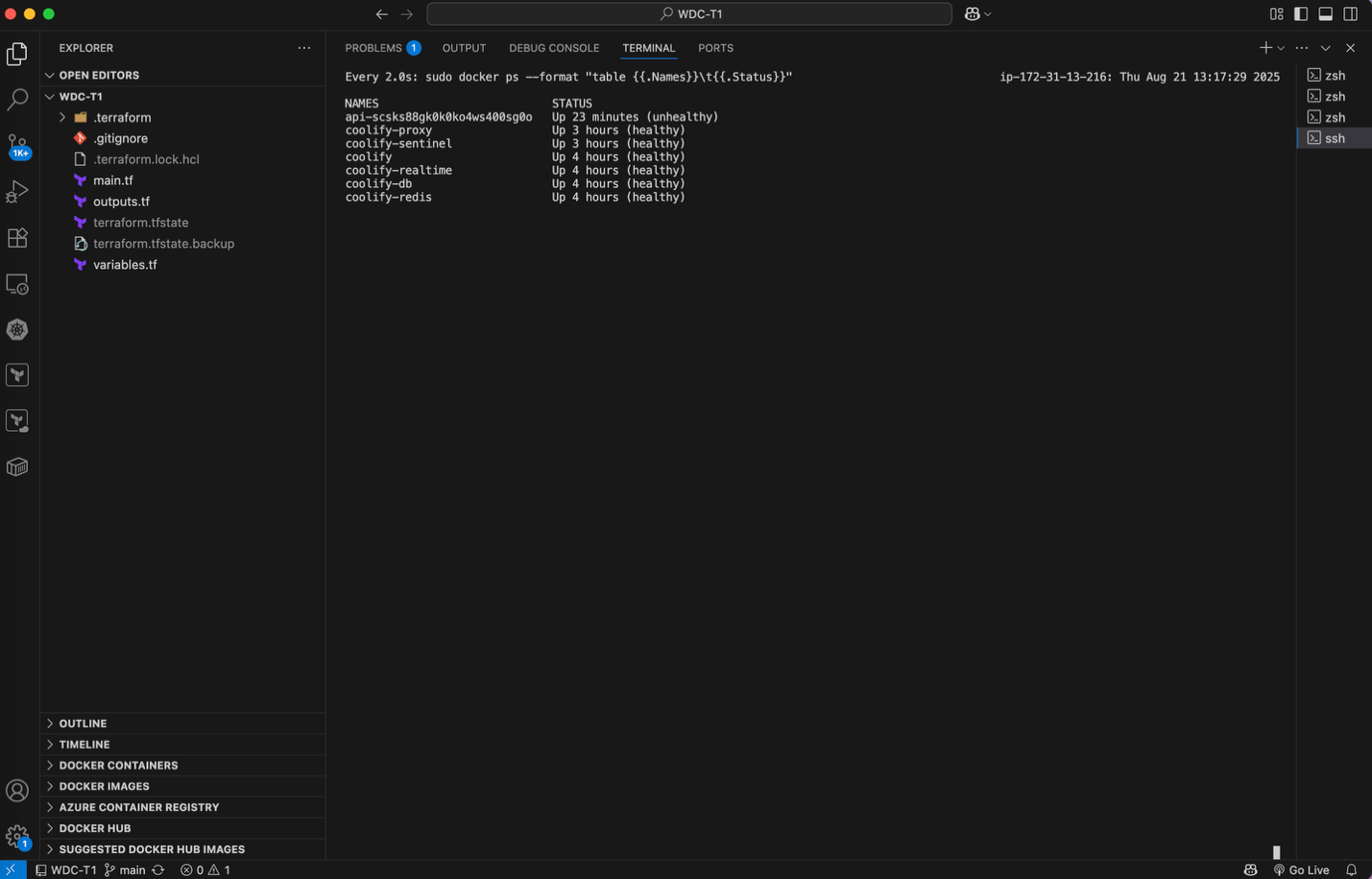
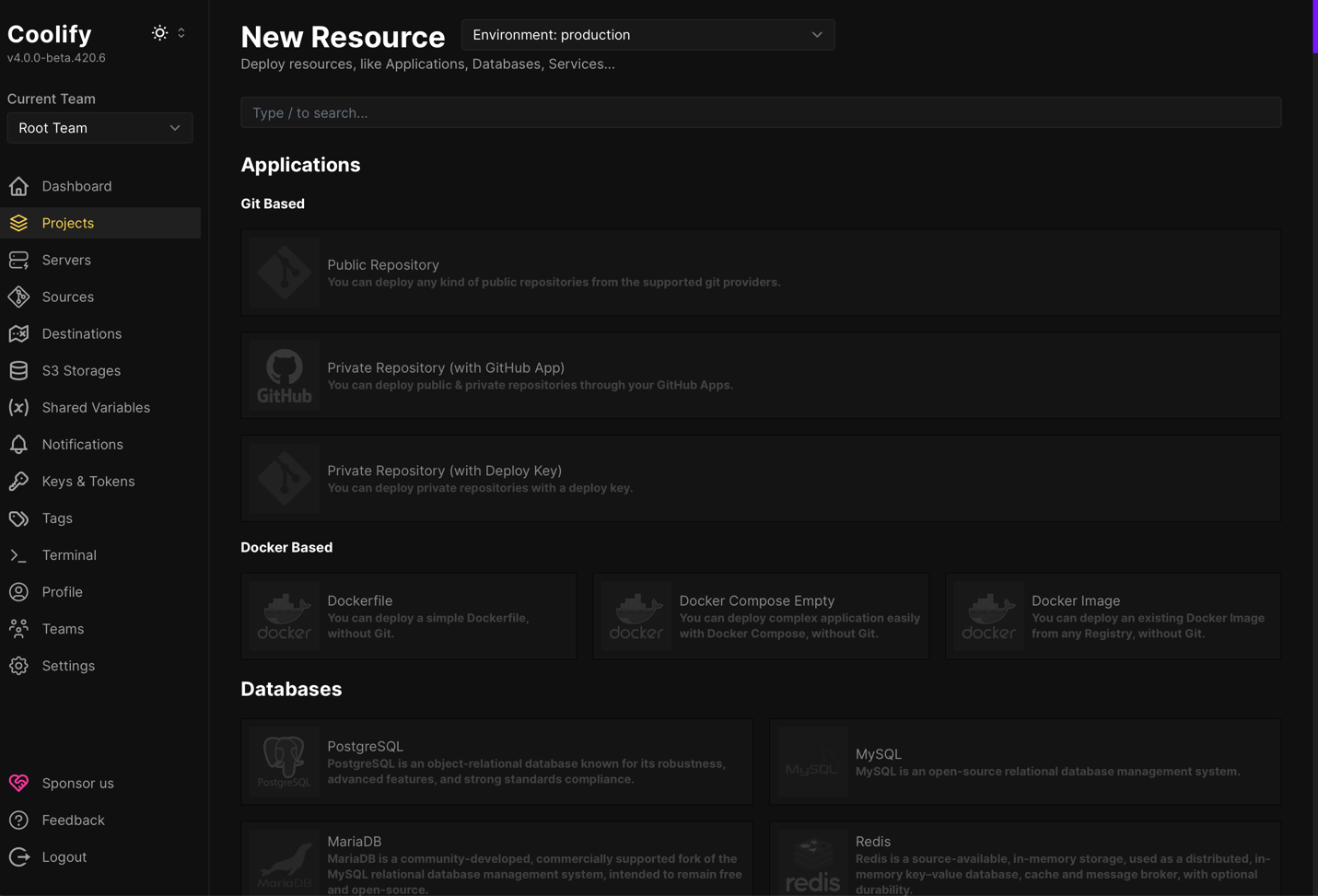
## **Coolify Evaluation**

Coolify is an open-source, self-hosted platform designed to simplify application deployments like Heroku/Vercel. In my testing, I found it very easy to deploy and manage, with a clean interface and strong GitHub integration that automates SSL certificates and reverse proxy configuration.

However, there are some important limitations. Coolify installs itself directly on the host machine, which makes setup less isolated and can raise security and maintenance concerns. I also found that its Cloudflare tunnel integration is unreliable, might create a problem for certain applications. Finally, Coolify’s version updates can be unstable, meaning new releases may occasionally break functionality.

Compared to Vercel, Coolify provides the advantage of being selfhosted with no vendor lock in or recurring usage based fees, but this comes at the cost of greater operational responsibility (managing updates, security hardening). Vercel, while more expensive, offers high reliability, scaling, and managed support, which Coolify currently cannot do.

Overall, I see Coolify as well suited for small projects, startups, or internal tools where cost control and open-source flexibility are important. For production critical or large scale systems, especially where uptime and stability are essential, Vercel or other mature platforms may be safer choices.



I evaluated **Dokploy** as a Coolify alternative. I installed Dokploy v0.24.12 on an EC2 host, deployed a simple service, then confirmed its **multi container** capability by running a Compose stack (web+API) with inter-service networking and external reachability. Sanity load tests with  (60s) and k6 (2–3 min at 200–400 VUs) completed with **0% HTTP errors**; wrk sustained low-thousands req/s on the hello endpoint and k6 showed p95 roughly **150–220 ms** on this single host (useful for validation, not capacity planning). The UI includes Projects, Monitoring, Schedules, Docker/Swarm, and settings for Web Server, Git/Registry/S3, SSH Keys, and Users; I didn’t see a dedicated Secrets view in this build.

**Where Dokploy stands out:** it does branch/PR previews with their own URLs and **automatically cleans them up after merge**, which keeps maintenance low; it also makes TLS/custom domains straightforward, supports single host or Swarm (including remote servers), and includes handy bits like Schedules and Registry/S3. It gives more built in deploy help than Dockge without the heavier feel of Coolify.

**Important caveat:** I don’t yet know the final production goal. These results validate end to end function and basic stability only. If the target is a multi service stack with a database, or a high traffic app (e.g., ~100k daily active users), we should repeat tests with that topology and workload.

**Bottom line:** For our PR-preview + auto-cleanup requirement, **Dokploy** best matches what we need, **Dockge** needs CI glue, **Coolify** is heavier.

**Alternatives**

* **Dokploy**: Lightweight, clean UI with Traefik backed routing. Good fit for single host or small Swarm; Git/Registry/S3 built in. More “DIY PaaS” than fully Heroku like (secrets UX is basic).
* **Coolify**: Most PaaS like: Git deploys, SSL automation, reverse proxy out of the box; polished GUI but can be update sensitive.
* **Dockge**: Purpose built for **Docker Compose**. Great if you only need to run/manage Compose stacks. Not a full PaaS (no built-in SSL/proxy automation or Git pipelines).

