



Cloud-Native Gauntlet: Your Two-Week Ordeal

👉 Dear Delusional Adventurers,

So you thought LPIC 1xx was “hard”? 🤖 That was baby Linux with juice boxes 🥤 nap time , and a coloring book 💖 Now staring you down like an angry sysadmin with `root` privileges are **LPIC 2xx** 📖 **CKAD** 🏢 and **AWS Cloud Practitioner** 🇺🇸 These aren't exams—they're life insurance checks waiting to bounce.

But why give you 2 weeks of peace 😊 when I can serve you suffering on a silver platter 😞 Behold, **The Cloud-Native Gauntlet™** 🗡️👤: the challenge nobody asked for but everybody deserves.

Your mission (not optional 🙄):

- Build, from scratch, a **full-stack cloud-native monstrosity** ❤️
- Run it **entirely on your local machine** 💻, because cloud is for amateurs.
- **Offline mode only** 🚰🔌—because if you can't survive without StackOverflow, maybe you shouldn't exist.
- Must be **idempotent** 🔧—aka “run again until it stops crying.”

This thing should:\ 🌟 Make Kubernetes weep 😭\ 🌟 Make Docker question its career 😏\ 🌟 And make your laptop beg for early retirement 💪

Will you suffer? Oh yes 🖤 Will you regret your choices? 100% 😊 But if you somehow crawl out alive , hopped up on caffeine ☹️ and sheer spite 😡 then—and only then—you may challenge the mighty LPIC 2xx, CKAD, and AWS Cloud Practitioner.

Fail, and the GIS gods 🐱 are merciless. No retries. No Ctrl+Z. Just eternal shame 👁️.

So grab your `kubect1` 🖱️ prepare your `docker-compose` 💖, and remember: **in YAML, no one can hear you scream** 🤯👂.



Objective

You, alone (no collaboration, no crying on Discord), must:

- Write a **Rust** web application (todo app, cat meme shrine, or any API that makes sense). Backend must support JWT auth, talk to Postgres, and expose endpoints.
- Use **Postgres** for persistence, deployed via the **CloudNativePG operator**.
- Secure everything with **Keycloak**, enforcing token validation.
- Deploy it all to **K3s**, running on local VMs with **Vagrant** or **Multipass**.

- Provision infra with **Terraform** and configure with **Ansible**. Scripts must be **idempotent**.
- Package the app with **Docker** and load images into an **offline local registry**.
- Use **Kubernetes manifests** (ConfigMaps, Secrets, Deployments, Services, Ingress).
- Implement a **GitOps pipeline** using [**Gitea** or **GitHub**] + Actions or ArgoCD.
- Mesh services with **Linkerd**, gain observability and mTLS.
- Document it all with **Mermaid diagrams** (architecture, pipeline, auth flow).

All of this **offline**. No cloud. You vs. the void.

The Twelve Trials (2–4h per day)

Day 1-2: Summon the Cluster Beasts

- **Spin up local VMs:** Use Vagrant or Multipass to create 1–2 VMs.
- **Provision infra:** Use Terraform + Ansible to install base packages.
- **Install K3s:** Deploy a lightweight Kubernetes cluster.
- **Offline prep:** Configure local DNS/hosts.
- **Pre-pull images:** Keycloak, CNPG, Gitea, Linkerd, app base images.
- **Local registry:** (Optional) set up a registry offline.

Day 3-4: Forge Your Application

- **Build a Rust API:** Choose Actix, Axum, or Rocket.
- **Define models:** At least two entities (e.g., User and Task).
- **Implement endpoints:** Create/list tasks, with authentication guards.
- **Integrate DB layer:** Use Diesel or SQLx.
- **Local test:** Run app with a local Postgres container.

Day 5: Containerize Your Pain

- **Dockerfile magic:** Multi-stage build (builder + runtime).
- **Build & test:** Build image and run locally.
- **Registry push:** Push/import to registry.

Day 6-7: Database & Deployment

- **Deploy CNPG:** Install operator.
- **Create DB cluster:** Write CRD for Postgres.
- **Secrets setup:** Store DB credentials.
- **App Deployment:** Write Deployment + Service.
- **ConfigMaps:** Provide configs.
- **Ingress:** Expose app.
- **Validation:** Confirm persistence.

Day 8: Bow Before Keycloak

- **Deploy Keycloak:** With storage.

- **Expose service:** Ingress at `keycloak.local`.
- **Configure realm:** Create a realm.
- **Add client:** Define credentials + URIs.
- **Add user:** Create test user.
- **Integrate app:** Validate JWTs.
- **Test flow:** Acquire token and call endpoints.

Day 9-10: Embrace the GitOps Curse

- **Install Gitea:** Expose via `gitea.local`.
- **Repos setup:** Create `app-source` + `infra`.
- **Version control:** Commit + push.
- **CI pipeline:** Actions or Drone CI.
- **CD pipeline:** ArgoCD syncs infra repo.
- **End-to-end test:** Push → build → deploy.

Day 11: Enter the Mesh

- **Install Linkerd:** Deploy + check.
- **Namespace injection:** Annotate for sidecars.
- **Redeploy app:** Ensure injection.
- **Observability:** Use `linkerd viz`.
- **mTLS:** Confirm encrypted traffic.

Day 12: Write Your Epic

- **Create README:** Write docs.
- **Mermaid diagrams:** Arch, pipeline, auth.
- **Infra story:** Terraform → Ansible → K3s.
- **Idempotence note:** Proof of reruns.
- **How-to guide:** Repro from scratch.
- **Comic relief:** YAML jokes.



Victory Conditions

- Entire system runs offline.
- Infra + configs are idempotent.
- GitOps works.
- Keycloak protects app.
- Linkerd meshes.
- Docs + diagrams included.

Finish *everything*. No shortcuts. No mercy.



Epilogue

This is not a cozy group project . Each of you must **suffer alone** 📺🔥, staring at logs like hieroglyphics 🙄.

Yes, you can customize 😞—but all requirements must be met. No excuses 🦴.

When (if) you crawl out , you'll have scars 🖐️, PTSD from `docker ps` 💕, and **hatred of indentation errors** 🙄 respected by Python devs 🤖.

That hatred fuels victory. Enough to conquer LPIC 2xx, CKAD, and maybe the mythical **Carrie Anne Certification™** 🎉.

Now go . May your **YAMLs align**, may your **Pods stay Running**, and may you forever remember:

```
kubectl describe 🙏
```

Dismissed. 🚪