A PRACTICAL GUIDE

Generative AI for K8s Platform Engineering

TALOS LINUX · GITOPS · AGENT SKILLS

Muthukumaran Navaneethakrishnan

Hari Balaji Murugaiyan Karunanidhi

Generative AI for K8s Platform Engineering

Talos Linux, GitOps & Agent Skills

Muthukumaran Navaneethakrishnan and Hari Balaji M K

This book is available at
https://leanpub.com/aipoweredkubernetesplatformengineering

This version was published on 2026-06-07

Contents

Acknowledgments

Thank you to the open-source communities whose work this book stands on — **Sidero Labs** and the Talos Linux project, the **CNCF** ecosystem (Kubernetes, ArgoCD, CloudNativePG, KubeVirt, Cilium, SPIFFE/SPIRE), and the **DuckDB**, **Apache Arrow**, and **Apache Iceberg** communities. None of these experiments would exist without the tools they give away for free.

Thank you to the early readers and reviewers who ran the labs, found the rough edges, and told me honestly when a number didn't add up — that honesty is the whole point of this book.

And, as always, heartfelt thanks to my family and friends for their patience and unwavering support throughout this project.

Preface

This book started from a simple, slightly uncomfortable question: if I am going to let an AI assistant anywhere near a production Kubernetes cluster, how do I make absolutely sure it cannot do something stupid? Not "probably won't" — *cannot*. That question turned into a concrete project: a safe, governed AI SRE agent skill that reviews a platform running on Talos Linux, read-only by default, that asks which cluster it is touching, shows every command before it runs, and is grounded in evidence rather than vibes.

This is not an "I am an expert, look at me" book. It is a practical, dev-to-dev guide. We build the agent one capability per chapter against a local, throwaway Talos lab — cluster health, reliability, security drift, certificates, a scored maturity report, GitOps remediation — and then we push past the toy: vulnerabilities and fearless upgrades, the honest economics of bare metal and on-prem, running stateful databases and a data lakehouse, and finally an air-gapped AI operator that runs the model itself on your own hardware.

Two commitments run through every page. The first is **safety as a property of the system, not a promise** — the guardrails are enforced in code, not in good intentions. The second is **honesty about numbers** — where this book quotes a result, it was measured, on a real cluster, and where something was not measured, it says so. The companion repository holds every experiment so you can run it, break it, and measure it yourself.

If you operate Kubernetes platforms and you are curious whether generative AI can be a careful teammate rather than a liability, this book is for you. Let's build it together — carefully.

Trademark Notice

This book and its companion project reference technologies developed by various organizations and use their names solely for educational and instructional purposes. Trademark acknowledgments are as follows:

- Talos Linux, Sidero Metal, and Omni are products of Sidero Labs, Inc.
- Kubernetes, ArgoCD, CloudNativePG, KubeVirt, Cilium, SPIFFE, and SPIRE are projects of the Cloud Native Computing Foundation (CNCF) and The Linux Foundation.
- Linux is a registered trademark of Linus Torvalds.
- PostgreSQL and the Slonik logo are trademarks or registered trademarks of the PostgreSQL Community Association of Canada.
- DuckDB and DuckLake are projects of DuckDB Labs; Apache Iceberg and Apache Arrow are projects of the Apache Software Foundation.
- Amazon Web Services, EKS, and Fargate are trademarks of Amazon.com, Inc.; GKE and Google Cloud are trademarks of Google LLC; Azure and AKS are trademarks of Microsoft Corporation.
- Cherry Servers is a trademark of Cherry Servers, UAB.
- Any other trademarks mentioned are the property of their respective owners.

This book and its documentation are independent of the companies and entities listed and do not imply any endorsement. The use of these trademarks is strictly for identification and does not signify association with any of the mentioned parties.

Chapter 1: GenAI for Platform Engineering & the Talos Lab

It is 3 a.m. A certificate expired four hours ago, the renewal job failed silently three weeks before that, and the first anyone hears of it is a pager going off because half the ingress traffic is now failing TLS. The dashboards were green the whole time. They were green because a dashboard answers the question *"what is the value of this metric right now?"* — and nobody had a panel for *"what is quietly about to take us down?"*

This book is about closing that gap. Not with another dashboard, and not with an AI that simply runs `kubectl` for you when you ask nicely. It is about building an agent that *reasons over* a Kubernetes platform — one that can look at a cluster running on Talos Linux, gather the evidence a senior engineer would gather, and tell you what is wrong, what is about to be wrong, and what to do about it, with the evidence attached.

> The thesis of this book in one line: dashboards report state; the agent we build reasons about consequences. State is "cert expires in 8 days." Consequence is "cert expires in 8 days, the renewal job last succeeded 40 days ago, and these three Services depend on it — here is the outage timeline and the fix."

1.1 In This Chapter

We will keep the shape of every chapter the same, so you always know the terrain before you walk it. In this one you will:

- See why dashboards, alerts, and visualization tools leave a real gap — and what kind of work actually fills it.
- Understand what "an AI that reasons over your platform" means in practice, and how it differs from wrapping a chat model around the command line.
- Learn why **Talos Linux** — immutable, API-driven, and without SSH — turns out to be an unusually good substrate for an AI operator.
- See the four administrative pains this book ultimately pays off — **upgrades**, **vulnerabilities**, **sizing**, and **lock-in** — named here and tackled later in the book.
- Understand the larger bet: that **bare metal** plus a disciplined AI operator can be both cheaper and more operable than a managed control plane — with the AI itself runnable **on-prem or air-gapped**.
- Stand up a local lab of Talos clusters on Docker, at zero cloud cost, so every later chapter has something real to run against.
- Make your first read-only call against a live cluster — the seed of the agent skill we grow over the rest of the book.

This is an enterprise book, which to me means one thing above all: nothing here is allowed to do something to a production cluster that you would be afraid to let a new hire do unsupervised. We will earn that trust deliberately.

1.2 The Problem With Dashboards

Let me be fair to dashboards first. Grafana, the Talos dashboard, Lens, `k9s` — these are excellent. If you operate Kubernetes without them you are working too hard. They are not the problem.

The problem is that they answer *point-in-time, single-signal* questions, and they leave the hard part — *correlation across signals, over time* — to the human staring at them. Consider three facts, each of which has a perfectly good dashboard panel:

1. A Deployment was rolled out at 14:02.
2. A downstream Service started returning 5xx at 14:05.
3. A NetworkPolicy was last modified two days ago.

Every one of those is visible. The *conclusion* — "the 14:02 rollout changed a label selector, which the two-day-old NetworkPolicy no longer matches, which is why the Service is failing" — is not on any panel. It lives in the head of whoever is on call, and only if they happen to be experienced enough and awake enough to assemble it.

> **This is the work we are automating.** Not the *display* of metrics — that is solved. The *reasoning across* metrics, configuration, and history to reach a conclusion a tired human might miss at 3 a.m. That reasoning is exactly what large language models are good at, *provided* they are given grounded evidence to reason over and not asked to hallucinate it.

That last clause is the whole game, and we will come back to it constantly.

1.3 What "Reasoning Over Your Platform" Actually Means

There is a version of "AI for Kubernetes" that I want to steer you away from immediately, because it is the easy version and it is not impressive: take a chat model, give it the ability to run `kubectl`, and let it improvise. It demos well for ninety seconds and then it does something like `kubectl delete` against the wrong namespace, or it confidently invents a reason for an outage that is simply wrong.

The version we are building is different in three specific ways.

- **It is grounded.** Deterministic scripts gather the facts — node status, probe configuration, certificate dates, the actual YAML. The model never *fetches* anything by guessing; it reasons over a packet of evidence that was collected for it.
- **It cites.** Every finding points at the thing that produced it: the manifest line, the expiry date, the failing readiness check. A finding you cannot trace back to evidence is treated as a bug, not a feature.
- **It is read-only until it has earned more.** For most of this book the agent looks and reasons but does not touch. When it finally proposes changes, those changes go through a pull request and your existing GitOps review — never a direct mutation.

> A good way to hold the distinction in your head: we are not building an AI that *operates* your cluster. We are building an AI that *reviews* it the way a careful principal engineer would — and only later, and only through the front door, helps fix what it finds.

1.4 Why Talos Linux

You can do everything in this book against any conformant Kubernetes cluster. So why build the lab on Talos Linux specifically? Because Talos's design philosophy and the needs of a trustworthy AI operator line up almost perfectly.

Talos is a Linux distribution built for exactly one job: running Kubernetes. It makes three choices that matter to us:

- **It is immutable.** The OS is read-only. There is no package manager, no `apt install`, no configuration drift accumulating on individual nodes over months. What you declared is what is running.
- **It is API-driven.** You do not configure a Talos node by editing files on it. You apply a machine configuration through an API. The whole system is meant to be operated programmatically.
- **There is no SSH and no shell.** You cannot log into a Talos node and poke around. Every interaction goes through the typed, authenticated `talosctl` API.

  Sit with that last point for a moment, because it is the one that changes how you should feel about an AI operator. The usual fear with "let an agent touch my servers" is the unbounded blast radius of a shell — an agent with SSH can do *anything*. Talos has no shell to give it. The agent's entire surface area is a well-defined, auditable API. The platform itself enforces a boundary we would otherwise have to build by hand.

In other words, the property that makes Talos pleasant for humans — no fragile pets, no snowflake nodes, everything through a clean API — is the

same property that makes it *safe* to put an agent in front of. We get a meaningful chunk of our guardrails for free, from the operating system.

There is a second, more pragmatic argument, and it is the one the back half of the book cashes in. Ask any platform team what actually hurts and you get the same short list — four pains, each named here and paid off in a later chapter:

- **Upgrades** top it. Major-version Kubernetes upgrades fail in ways you usually cannot rehearse, so they are done holding your breath — whereas Talos lets you *dry-run* an upgrade and keep an A/B image to roll back to (Chapter 6).
- **Vulnerabilities** arrive faster every year — the public CVE feed ran around 50 disclosures a day in 2020 and is past 130 a day in 2025 — and Talos answers with a tiny attack surface: no shell, no SSH, no package manager, roughly a dozen binaries, so there is simply far less to be vulnerable *in* (Chapter 6).
- **Sizing** is the quiet waste. Industry utilization surveys keep finding Kubernetes clusters sitting around 8% CPU and 20% memory utilization, which means most teams are paying for orchestration overhead they only occasionally need (Chapter 7).
- **Lock-in** is the tax nobody budgets for. Most enterprises run a vendor flavor (OpenShift, Rancher, EKS, GKE) with its own ecosystem and its own exit cost, rather than the vanilla Kubernetes underneath (Chapter 7).

Put those four together and you get the wager of this book: **bare metal plus a safe AI operator can be cheaper *and* more operable than a managed offering.** Cheaper because you stop paying the overhead and the vendor tax; more operable because the agent does the patient, evidence-gathering reasoning that upgrades and CVE

triage demand. And because the operator is just an orchestrator of read-only calls over a typed API, the AI itself can run **on-prem or fully air-gapped** (Chapter 9) — no cluster secret ever has to leave the building. Chapter 1 only *names* these terms; the chapters they point to are where we earn them.

This is not a thought experiment. The reliability, elasticity, and storage numbers later in the book were *measured* — on a real Cherry Servers bare-metal box (a 4-core/32 GB Xeon E3 with a genuine `/dev/kvm`), running **Talos v1.13.3 / Kubernetes v1.36.1** as KVM VMs on it. Those two version numbers are not decoration: they are what makes the measurements reproducible and what tells you the box was small — a single 4-core host, so treat the figures as lab-scale evidence of a *direction*, not datacenter throughput. Appendix A is the exact, hour-billed recipe for that box, so you can reproduce the measurements rather than take them on faith. Where a figure is lab-scale or nested virtualization, the book says so plainly; where an experiment is authored but not yet run, it is flagged as a runnable spike, never dressed up with invented numbers.

1.5 The Shape of What We'll Build

Over the rest of the book we build a single artifact — an **SRE agent skill** for Talos and Kubernetes — and then push it all the way onto your own hardware. It is a skill in the agent-tooling sense (a packaged capability your AI assistant can invoke), and it grows one ability per chapter:

- **Chapter 2** builds the scaffold and the guardrails: read-only by default, ask which cluster first, show every command before running it, refuse to run against an unrecognized context.

- **Chapter 3** teaches it to assess cluster health and reliability — nodes, etcd, the control plane, readiness and liveness probes, replica counts, disruption budgets.
- **Chapter 4** adds security drift and certificate analysis, and pulls everything together into a *platform maturity report*: a scored, ranked, evidence-cited summary of how healthy the platform actually is.
- **Chapter 5** closes the loop with GitOps — the agent proposes fixes as pull requests that flow through ArgoCD.
- **Chapter 6** turns to vulnerabilities and the pain that drives every upgrade, and shows how Talos makes upgrades *fearless* — dry-run preflight and atomic A/B rollback.
- **Chapter 7** leaves the cloud: running this on bare metal in your own datacenter, the ecosystem tax you avoid, and the honest economics.
- **Chapter 8** answers the "don't run databases on Kubernetes" fear with evidence — stateful workloads, storage, and a data lakehouse.
- **Chapter 9** runs the AI operator itself on your hardware: a sovereign, air-gapped plugin with a local model and your own tools.

The full, tested code lives in the companion repository (https://github.com/muthuishere/powered-platform-engineering) so you can run it, not just read it. In the book itself I will show the parts that teach something and explain the *why* behind them, rather than reprinting every line.

1.6 Setting Up the Lab

Everything in this book runs locally, on Docker, for free. Talos can create throwaway clusters as Docker containers in under a minute, which means you can break things, reset, and try again without touching a cloud bill or an ISO.

> The Docker lab is enough for almost every chapter. Two things want *real* hardware: KubeVirt (running VMs in Kubernetes needs a genuine `/dev/kvm`, Chapter 8) and any benchmark you want undistorted by a laptop. When you reach those, **Appendix A** stands up a real Talos cluster on a bare-metal box billed by the hour — a few cents for an afternoon of experiments — and that is the box the book's measured numbers came from.

You need two command-line tools: `talosctl` (to create and talk to Talos clusters) and `kubectl` (to talk to Kubernetes). On macOS with Homebrew:

```
brew install siderolabs/tap/talosctl
brew install kubectl
```

With those in place, create your first cluster:

```
talosctl cluster create --name dev
```

That single command provisions a small Talos-based Kubernetes cluster as a set of Docker containers, writes a `talosconfig` so `talosctl` can reach it, and merges a context into your `kubeconfig` so `kubectl` can too. Confirm both halves are reachable:

```
talosctl --context dev version
kubectl config get-contexts
```

> **If the cluster hangs on "waiting for etcd to be healthy," it is DNS.** On some Docker runtimes — notably OrbStack on macOS — a fresh Talos node comes up with an empty nameserver list, so it cannot resolve `registry.k8s.io` to pull the etcd image, and the cluster then waits forever for an etcd that never starts. The fix is to pin

a nameserver in the Talos machine config. The companion repo's `spikes/talos-gitops/scripts/01-create-clusters.sh` does this for you via a `--config-patch` (and it also names the lab's clusters and wires the multi-cluster topology). If your bare `talosctl cluster create` hangs on etcd, use that script.

Name your clusters deliberately from day one. We are going to build an agent that asks "which cluster am I operating on?" before it does anything, and that question is only as safe as your naming. Create `dev` now; when we get to multi-cluster work we will add `staging` the same way. Treat the names as a contract, not a convenience.

If you want to see the multi-cluster future early, you can create a second cluster now and watch both appear as separate contexts:

```
talosctl cluster create --name staging
kubectl config get-contexts
```

The companion repo's scripts go further and build the full lab this book uses — a dedicated `ops` cluster (running ArgoCD and Gitea for Chapter 5) alongside `dev`, `staging`, and `prod`.

When you are done experimenting, tearing a cluster down is just as quick:

```
talosctl cluster destroy --name dev
```

Because Talos clusters here are disposable Docker containers, the safest way to learn is to be reckless *on purpose*. Break a cluster, observe what the symptoms look like, then destroy and recreate it. The agent we build is only as good as our understanding of what "unhealthy" actually looks like, and the cheapest place to build that intuition is a cluster you can throw away.

1.7 Your First Read-Only Call

The agent we build is, at its core, an orchestrator of read-only calls plus a reasoning layer on top. So let us make the very first read-only call by hand — the kind of thing the agent will later do for us, but worth doing once with our own fingers.

Ask Talos about the health of the control plane:

```
talosctl --context dev health
```

Then ask Kubernetes what its nodes look like:

```
kubectl --context dev get nodes -o wide
```

Neither command changes anything. Both return evidence. That is the entire spirit of the first half of this book: gather evidence with commands that cannot do harm, then reason about what the evidence means. In the next chapter we stop typing these by hand and start wrapping them in a skill — with the safety rules built in from the first line, not bolted on later.

1.8 Who Should Read This Book

This book is written for platform engineers, SREs, and Kubernetes practitioners who want to put generative AI to real, production-minded use. I assume you are comfortable with Kubernetes basics — Pods, Deployments, Services, namespaces — and that you have operated, or want to operate, real clusters. I do *not* assume any machine-learning background; you will not be training models, you will be putting a capable one to careful work.

By the end you will have built an AI operator that takes on the four pains we just named — rehearsable **upgrades**, a shrunken **vulnerability** surface, honest **sizing**, and freedom from vendor **lock-in** — and you will have made the case, in working code, that **bare metal with a safe agent on top** is a defensible alternative to a managed control plane, runnable **on-prem or air-gapped** when your data cannot leave the building.

If your reaction to "let an AI near my cluster" is a healthy flinch, you are exactly the reader I am writing for. The whole book is an argument, made in code, that the flinch is right *and* that there is a disciplined way to do this anyway. Let's build it.

Chapter 2: Anatomy of a Safe SRE Agent Skill

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipoweredkubernetesplatformengineering.

2.1 In This Chapter

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipoweredkubernetesplatformengineering.

2.2 The Structure Is the Point

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipoweredkubernetesplatformengineering.

2.3 SKILL.md: Telling the Assistant When to Wake Up

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipoweredkubernetesplatformengineering.

2.4 The Five Guardrails

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipoweredkubernetesplatformengineering.

2.5 Read-Only, By Construction

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipoweredkubernetesplatformengineering.

2.6 Ask Which Cluster First

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipoweredkubernetesplatformengineering.

2.7 Show Every Command Before Running It

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipoweredkubernetesplatformengineering.

2.8 Preflight: The Cluster Guard

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipoweredkubernetesplatformengineering.

2.9 Why This Chapter Is the Enterprise Chapter

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipoweredkubernetesplatformengineering.

Chapter 3: Cluster Health & Reliability Review

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipoweredkubernetesplatformengineering.

3.1 In This Chapter

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipoweredkubernetesplatformengineering.

3.2 Step 01: Cluster Health

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipoweredkubernetesplatformengineering.

3.3 From Evidence to Finding

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipoweredkubernetesplatformengineering.

3.4 Step 02: Reliability

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipoweredkubernetesplatformengineering.

3.5 Detecting a Missing Probe — and Proposing the Fix

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipoweredkubernetesplatformengineering.

3.6 Honesty About False Positives

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipoweredkubernetesplatformengineering.

3.7 What's Next

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipoweredkubernetesplatformengineering.

Chapter 4: Security, Certs & the Platform Maturity Report

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipoweredkubernetesplatformengineering.

4.1 In This Chapter

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipoweredkubernetesplatformengineering.

4.2 Step 03: Security Drift

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipoweredkubernetesplatformengineering.

4.3 From a List to a Ranking

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipoweredkubernetesplatformengineering.

4.4 Step 04: Certificates, and the Art of Not Being Surprised

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipoweredkubernetesplatformengineering.

4.5 Prediction, Not Alerting

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipoweredkubernetesplatformengineering.

4.6 The Platform Maturity Report

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipoweredkubernetesplatformengineering.

4.7 Honest About the Score

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipoweredkubernetesplatformengineering.

4.8 What's Next

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipoweredkubernetesplatformengineering.

Chapter 5: GitOps & Autonomous Remediation

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipoweredkubernetesplatformengineering.

5.1 In This Chapter

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipoweredkubernetesplatformengineering.

5.2 Why GitOps Is the Only Door

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipoweredkubernetesplatformengineering.

5.3 The Remediation Loop

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipoweredkubernetesplatformengineering.

5.4 Anatomy of an Agent-Authored Pull Request

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipoweredkubernetesplatformengineering.

5.5 ArgoCD Closes the Loop

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipoweredkubernetesplatformengineering.

5.6 How Much Autonomy?

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipoweredkubernetesplatformengineering.

5.7 From One Cluster to a Fleet

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipoweredkubernetesplatformengineering.

5.8 What We Built, and Where It Goes

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipoweredkubernetesplatformengineering.

Chapter 6: Vulnerability, Patching & Fearless Upgrades

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipoweredkubernetesplatformengineering.

6.1 In This Chapter

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipoweredkubernetesplatformengineering.

6.2 The CVE Firehose, and the CVE You Inherited Without Asking

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipoweredkubernetesplatformengineering.

6.3 “Rust Won’t Save Us” — and the Limits of Smallness

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipoweredkubernetesplatformengineering.

6.4 Fearless Upgrades

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipoweredkubernetesplatformengineering.

6.5 Teaching the Agent: `vuln.py` and `upgrade.py`

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipoweredkubernetesplatformengineering.

6.6 The Vulnerability Dimension in the Maturity Report

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipoweredkubernetesplatformengineering.

6.7 What This Saves — and What It Won't Do

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipoweredkubernetesplatformengineering.

6.8 What's Next

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipoweredkubernetesplatformengineering.

Chapter 7: Bare Metal & the On-Prem Cloud-Native Datacenter

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipoweredkubernetesplatformengineering.

7.1 In This Chapter

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipoweredkubernetesplatformengineering.

7.2 Nothing Here Assumed the Cloud

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipoweredkubernetesplatformengineering.

7.3 The Integration Tax, and What Lock-In Actually Is

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipoweredkubernetesplatformengineering.

7.4 Is Kubernetes Even the Right Tool?

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipoweredkubernetesplatformengineering.

7.5 The Honest Economics of Leaving the Cloud

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipoweredkubernetesplatformengineering.

7.6 Why Talos Makes On-Prem Palatable

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipoweredkubernetesplatformengineering.

7.7 How a Bare-Metal Node Actually Joins

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipoweredkubernetesplatformengineering.

7.8 From One Rack to a Fleet of Datacenters

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipoweredkubernetesplatformengineering.

7.9 “But Who Operates It at 3 a.m.?”

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipoweredkubernetesplatformengineering.

7.10 Further Reading: Two Building Blocks Worth Knowing

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipoweredkubernetesplatformengineering.

7.11 Where This Leaves You

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipoweredkubernetesplatformengineering.

Chapter 8: Stateful, Storage & Data on Talos

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipoweredkubernetesplatformengineering.

8.1 In This Chapter

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipoweredkubernetesplatformengineering.

8.2 The StatefulSet Fear, Answered With Evidence

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipoweredkubernetesplatformengineering.

8.3 Measuring It Honestly

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipoweredkubernetesplatformengineering.

8.4 When Each One Wins

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipoweredkubernetesplatformengineering.

8.5 Object Storage Is Not Database I/O

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipoweredkubernetesplatformengineering.

8.6 DuckDB: A Query Runner, Not a Database Server

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipoweredkubernetesplatformengineering.

8.7 DuckLake versus Apache Iceberg

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipoweredkubernetesplatformengineering.

8.8 What the Lab Actually Measured

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipoweredkubernetesplatformengineering.

8.9 The Agent Reviews the Stateful Layer

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipoweredkubernetesplatformengineering.

8.10 What's Next

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipoweredkubernetesplatformengineering.

Chapter 9: A Sovereign On-Prem AI Operator: Custom Tools + a Local Model

9.1 In This Chapter

9.2 The Last Thing Leaving the Building

9.3 One Image, Two Jobs

9.4 How to Build a Custom Tool

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipoweredkubernetesplatformengineering.

9.5 How to Wire a Local Tool-Calling Model

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipoweredkubernetesplatformengineering.

9.6 Making It Air-Tight

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipoweredkubernetesplatformengineering.

9.7 Honest Caveats

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipoweredkubernetesplatformengineering.

9.8 Platform, Data, and the Brain — All on Your Hardware

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipoweredkubernetesplatformengineering.

Appendix A: Standing Up a Bare-Metal Talos Lab on Cherry Servers

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipoweredkubernetesplatformengineering.

A.1 What You'll Learn

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipoweredkubernetesplatformengineering.

A.2 Order the Box

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipoweredkubernetesplatformengineering.

A.3 The Networking Reality: Run Talos as QEMU VMs

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipoweredkubernetesplatformengineering.

A.4 Install the Toolchain

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipoweredkubernetesplatformengineering.

A.5 Create the Cluster

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipoweredkubernetesplatformengineering.

A.6 Two Talos-Specific Fixes Before Workloads Run

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipoweredkubernetesplatformengineering.

A.7 Tear It Down (Stop the Meter)

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipoweredkubernetesplatformengineering.