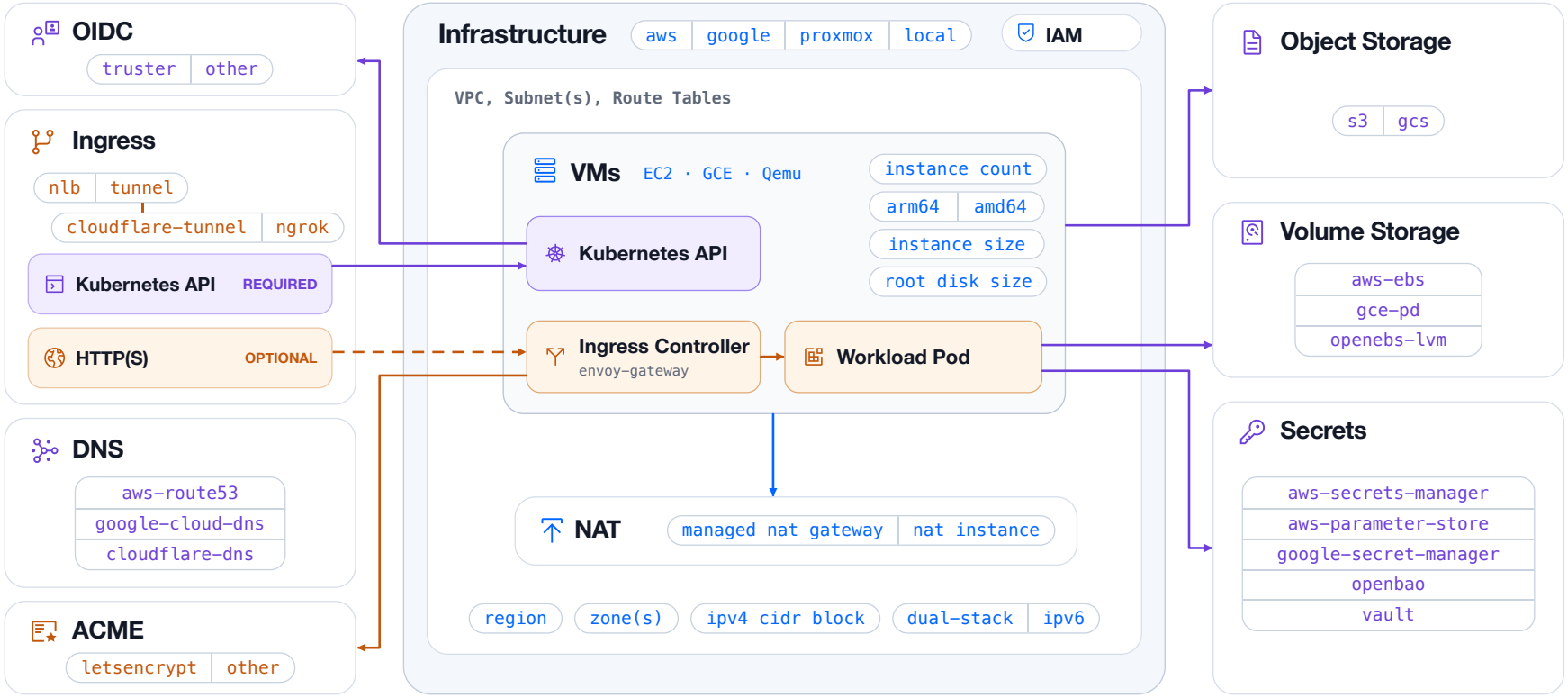
Building My Own IaC Engine Was a Mistake: Why I Now Use OpenTofu

The goal: a Kubernetes PaaS in 6-18 decisions + 1 click



Why not OpenTofu? Serverless

The hypothesis: a serverless control plane and provisioner would be reliable, cheap, and easy to operate.

OPTION A

Compile OpenTofu to WebAssembly

- Large binaries, slow cold starts, and more memory than TypeScript would need.
- Providers run as separate plugin processes, which an isolate can't spawn.

OPTION B

Call the Cloud SDKs from TypeScript

- A fixed scope: ~40 resources on 3 clouds, not a general-purpose IaC engine.
- Every cloud provider had a mature and lightweight TypeScript SDK available.

> I chose the Cloud SDKs. **It worked... until it didn't.**

Version 1: Stateless Reconciliation

01

Expected state

Determine the infra we want

→

02

Query the cloud

Discover what exists now

→

03

Reconcile

Create, update, or delete

Benefits: No database · Cheap · Fast happy path · Operationally simple

Avoiding state introduces challenges

Audit history

It created resources ($$) for customers, so what did the engine attempt?

Resource inventory

It manages resources ($$) for customers, so what is this cluster responsible for?

Failures & retries

Each run starts fresh, with no record of past attempts to back off on or give up after.

> If you need a durable audit log and resource list, **you might as well have a database.**

By Version 4: Starting To Look Like OpenTofu?

V2 — V3 — FEEDBACK — V4 — PRODUCTION

Serverless + SQLite

A resource inventory and audit log.

Migrations and table locks, write concurrency, other limitations.

Serverless + PostgreSQL

Faster schema iteration and scalability.

Limited by which providers Serverless could connect to.

> *Can we get a Terraform-like view?*
>
> Pre-beta tester

Requires plans, ability to preview changes and show progress.

Dependency Graph

Efficiently retry or reschedule a failed plan.

Super buggy and hard to get right.

PostgreSQL via Tunnels

Our preferred PostgreSQL setup needed a tunnel.

Using tunnels required running containers or VMs.

Market Feedback: IaC Is The Expectation

WHAT THE PROTOTYPE SHOWED

Stateless didn't work → a database is required.

Tunnels are required → VMs or containers are required.

The dependency graph was the hardest part.

A custom IaC engine wouldn't differentiate the product.

WHAT POTENTIAL CUSTOMERS EXPECTED

> *Can we take it and run it ourselves if we need to?*

> *Can we see plan/apply output like Terraform?*

62–73% **of infrastructure engineers use Terraform or OpenTofu**

The market expects IaC. We needed a DB + servers. **What if we removed Serverless, and used OpenTofu?**

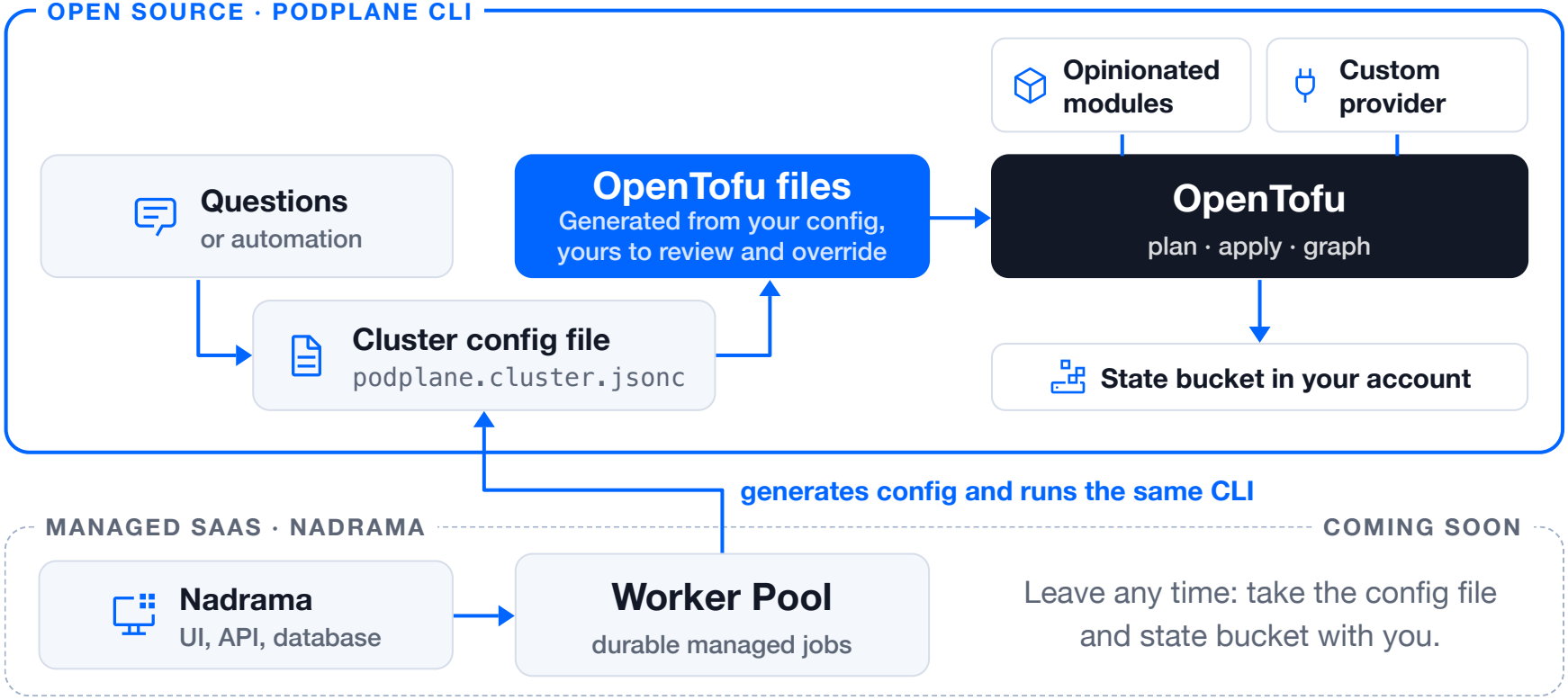
Starting Again: Open Source, Serverless-less, OpenTofu

Without Serverless, OpenTofu is viable and already does state, plans, dependencies, and safe retries.

Rethinking requirements to meet market needs

Same Product Experience	6-18 decisions → cluster config file. One command to deploy.
Escape Hatches	OpenTofu files are generated from the cluster config file. You must be able to override, not restricted by the abstraction.
CLI-first	CLI orchestrates OpenTofu. No SaaS or server(less) required.
State in your bucket	OpenTofu state in an object storage bucket in your cloud account.

The Plan For SaaS? It Just Runs the CLI



Open Source Demo: 6 Questions → Running Cluster

1. **Answer a few questions**
   Writes podplane.cluster.jsonc
2. **Generate files & review**
   OpenTofu files, ready to commit
3. **Plan & apply**
   OpenTofu JSONL output → progress UX
4. **Inspect the result**
   Cluster ready, state in your account

```
brew install podplane/tap/podplane
# OR:
# go install github.com/podplane/podplane@latest

podplane cluster create --no-apply
# questions → podplane.cluster.jsonc → *.tf

podplane cluster create -f podplane.cluster.jsonc
# OpenTofu plan & apply

kubectl get nodes
```

> **Podplane 0.x developer preview**: open source today.

OpenTofu: A Great Platform To Build On



podplane

Podplane is open source, Apache 2.0

podplane/podplane

Key Learnings

Just use containers

Most tools, OpenTofu included, want a normal machine. Save serverless for the jobs that clearly fit it.

Build on OpenTofu

It's an excellent base for your product. If you think you need your own engine, question that assumption first.

Ryan Djurovich: @ryan0x44 ryan0x44 ryan0x44.com