Why OpenShift?

The Hidden Cost of Choice in Kubernetes

Dan Mikita, Solution Architect
DevsOperative Inc.

March 2026

Quick Intro

- Founder & Solution Architect @ DevsOperative
- Platform & Kubernetes consulting
- Spend most of my time in day-2 operations
- I like reference architectures because snowflakes are expensive

Find me on LinkedIn here!



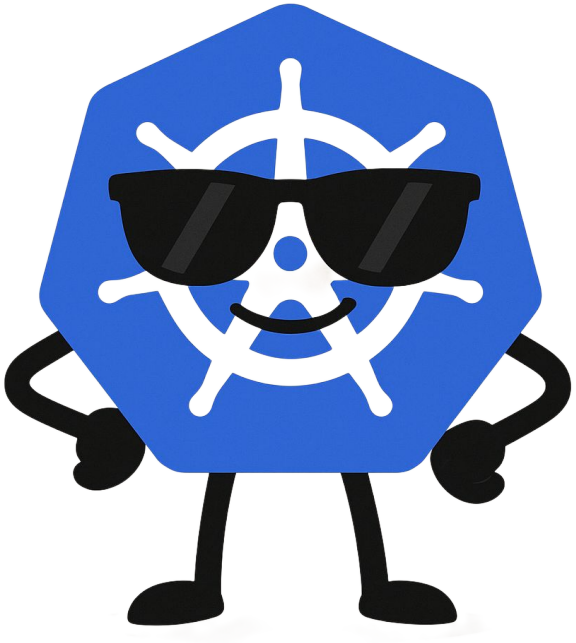
Dan Mikita

Two Questions

1. Who runs OpenShift today?
2. Who started Kubernetes on OpenShift?

Kubernetes Is Not A Platform

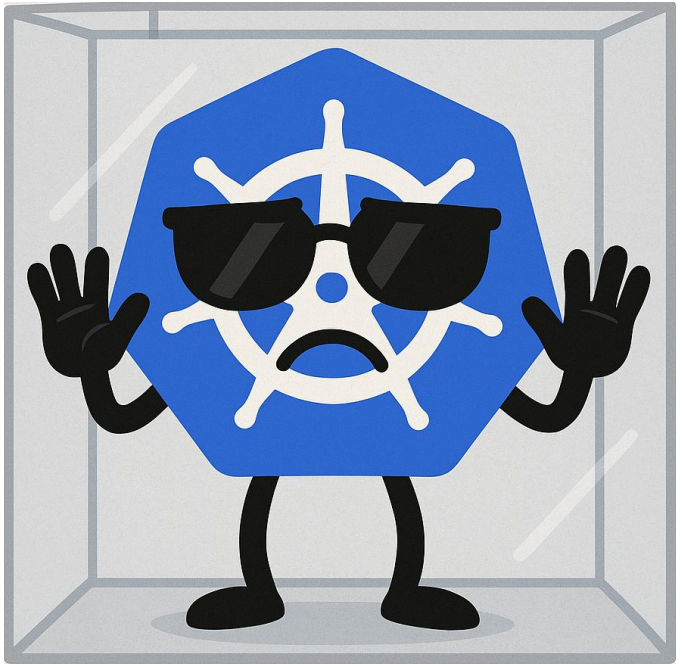
It’s an API and a scheduler.



You Ran: `kubeadm init`

It bootstraps:

- kube-apiserver
- kube-controller-manager
- kube-scheduler
- etcd
- CoreDNS
- kube-proxy



And that’s it.

No CNI installed.

The Real Work Starts Now.

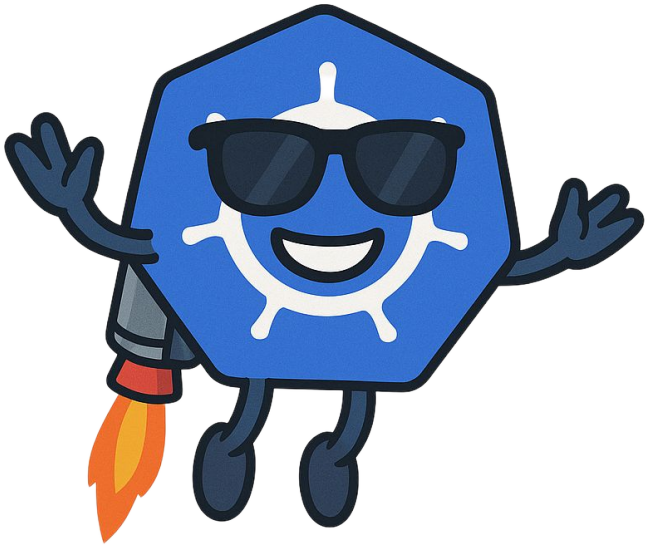
Layer 1 - Foundational Decisions

You must choose:

- CNI
- Ingress controller
- Load balancing strategy
- Storage + CSI
- Control plane topology
- Upgrade approach



Layer 2 – Operability Decisions



Day 2 Reality

You must choose:

- Monitoring & alerting
- Logging stack
- Certificate lifecycle
- Secrets management
- Backup & DR
- Policy enforcement
- Identity integration
- Artifact / image management

Layer 3 - Platform Expectations

What Your Developers Expect:

- GitOps / delivery workflows
- Self-service onboarding
- Templates / golden paths
- Secure image supply chain
- Multi-cluster governance
- Consistent upgrades



Kubernetes Gives You Choices

Every choice creates a lifecycle you now own.

The Hidden Cost

Not installation.

Ownership.

- Lifecycle burden
- Compatibility burden
- Security burden
- Support burden
- Staffing burden

At Enterprise Scale

You’re not building one cluster.

You’re governing a fleet.

Variance becomes the enemy.

Reframe

OpenShift isn’t special because it adds features.

It’s Special Because It Constrains Variance

Opinionated by Design

- Curated integrations
- Validated combinations
- Integrated lifecycle
- Unified release train
- Supported upgrade graph

OpenShift Is a Reference Architecture Disguised as a Distribution.

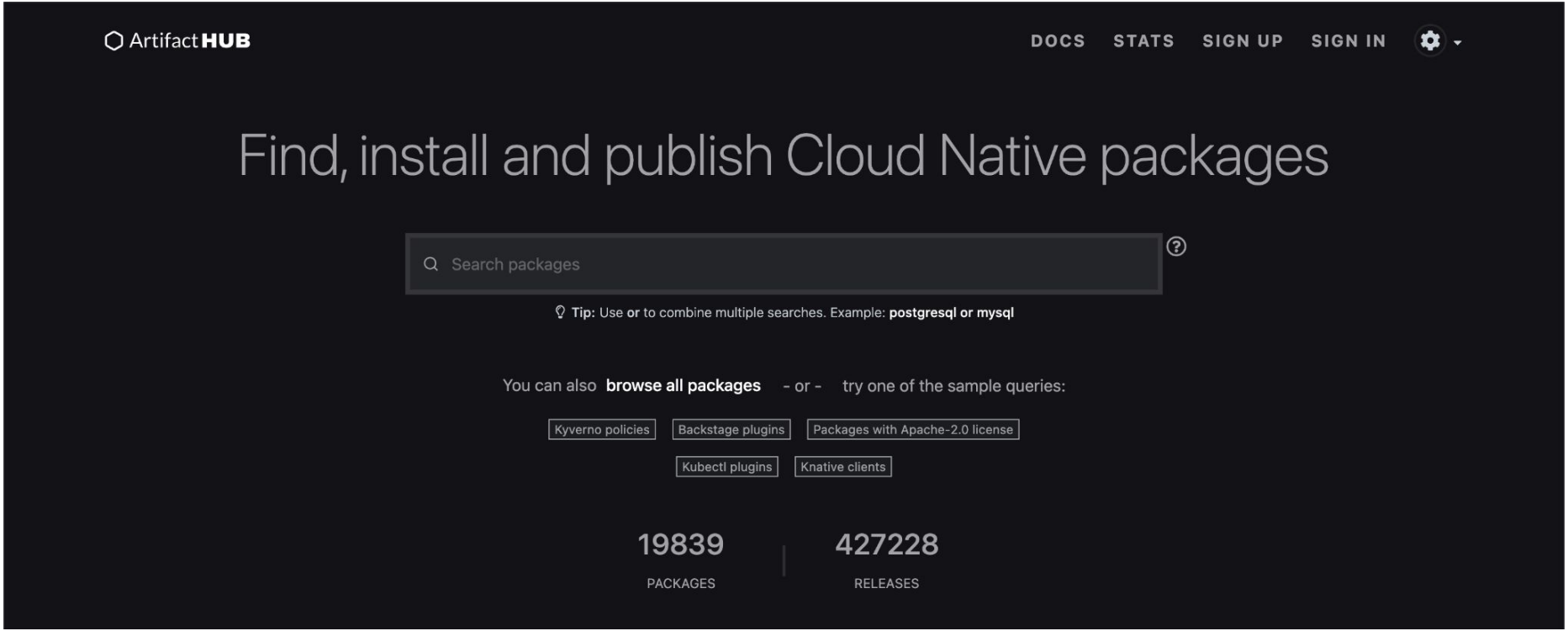
What Is a Reference Architecture?

- Known-good patterns
- Reduced decision surface
- Reduced compatibility risk
- Operational consistency
- Predictable upgrades

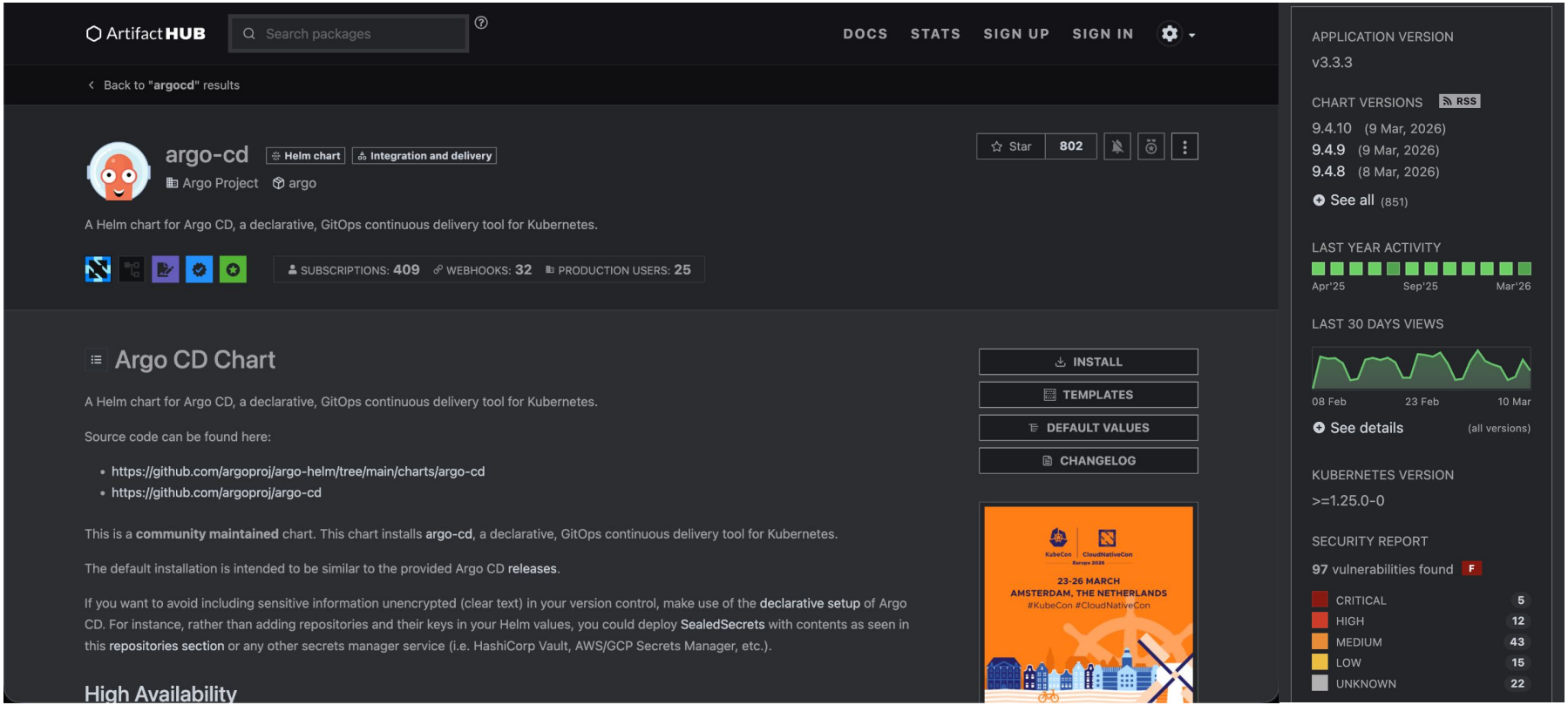
Vanilla Add-on Lifecycle

- Find chart on ArtifactHub
- Check Kubernetes compatibility
- Read upgrade notes
- Validate dependencies
- Tune values.yaml
- Repeat for every environment

Vanilla Add-on Lifecycle



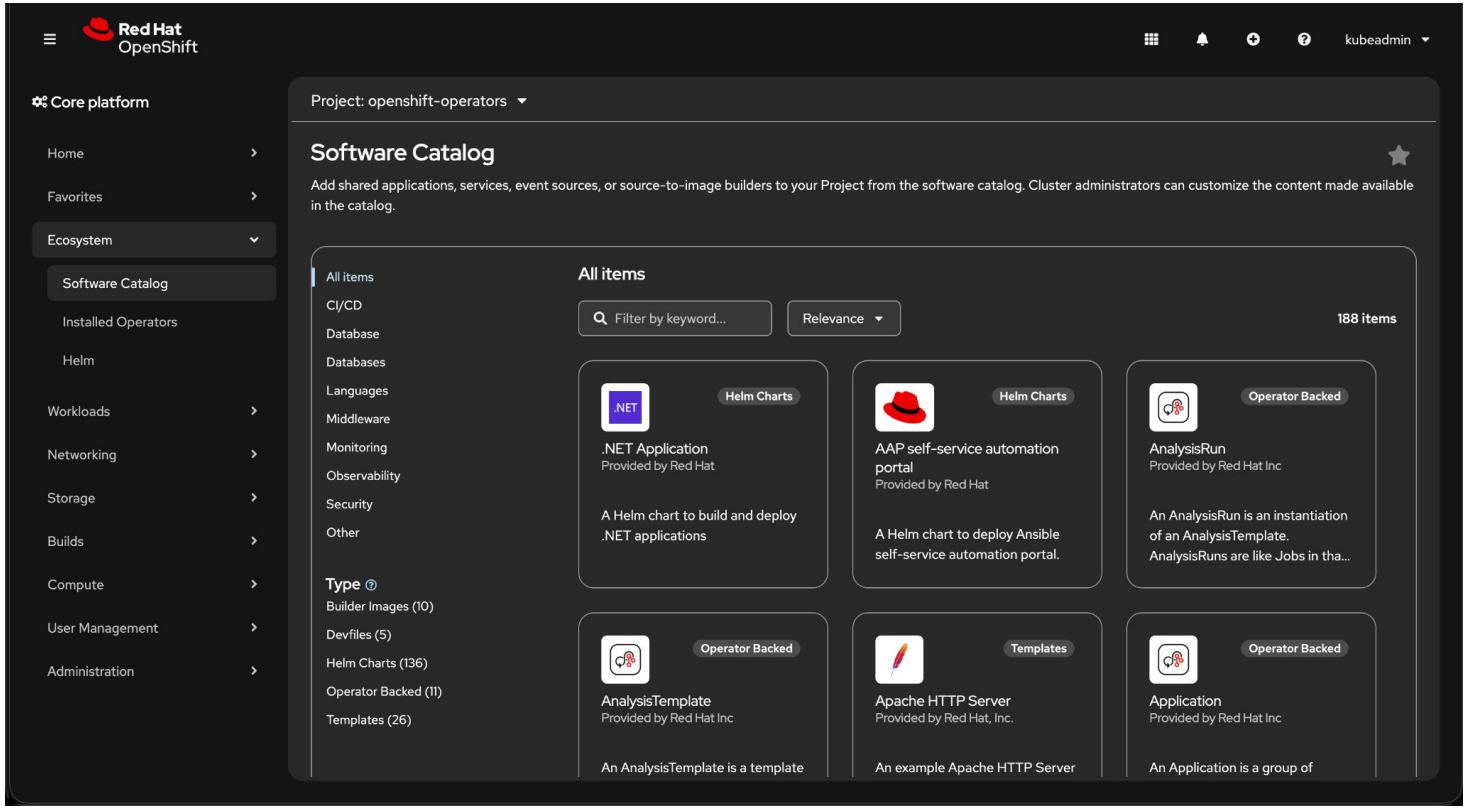
Vanilla Add-on Lifecycle



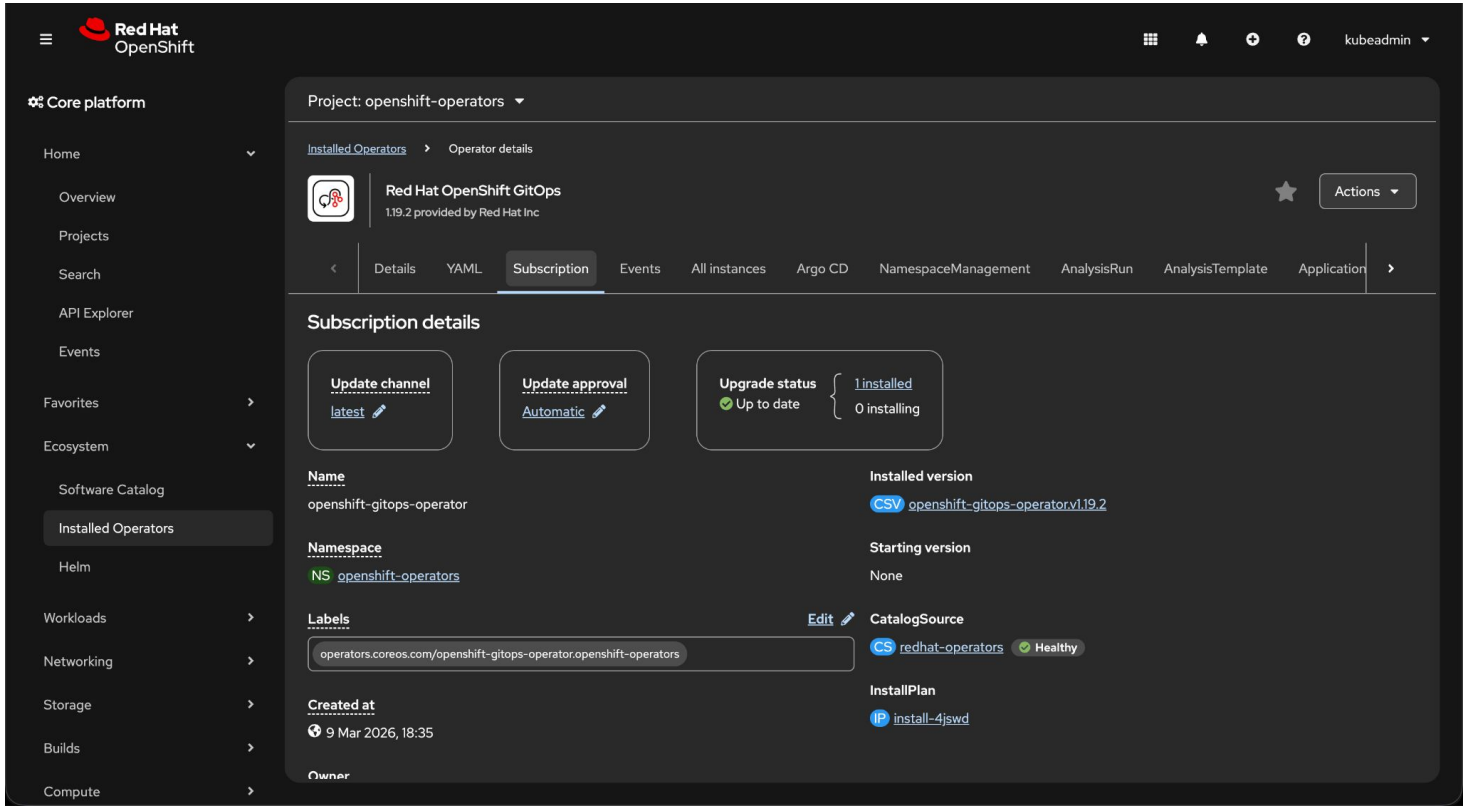
This Is Fine... Once.

It's Expensive Forever.

OpenShift + OLM



OpenShift + OLM



It’s Not Fewer Clicks.

It’s Fewer Unknowns .

Curated Catalogs Reduce Blast Radius.

- Security posture alignment
- Predictable upgrades
- Supportable combinations
- Reduced “random YAML from the internet”

Kubernetes Optimizes For Flexibility.

OpenShift Optimizes For Stability at Scale.

Both are valid.

Different goals.

The Real Value

OpenShift removes decisions.

- Reduced cognitive load
- Reduced variance
- Reduced entropy
- Faster path to boring production
- No hidden costs

Opinionated Platforms Win at Scale.

Questions?

Find me on LinkedIn here!