



How to Start Your Startup Fast on Kubernetes



Presenting Sponsor



Headline Sponsors



Title Sponsors



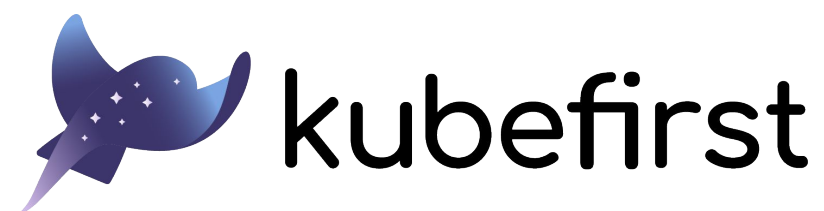
Member Sponsors

Baden-Wuerttemberg International
Big Compass
British Consulate-General
Canvas Community Credit Union
Caruso Ventures
Checkr
Credera
DAT Solutions
Davis Graham & Stubbs LLP
The Doyle Group
Granicus
Greenhouse Software
Kalen Jesse Photography
Red Rocks Community College
SmartMomGig
StickerGiant

Partner Sponsors

Bennett Thrasher
Colorado Sun
Contentful
Denver Economic Development & Opportunity
Endeavor
Denver Place
Geosnapshot
Industry Denver
Jake Jabs Center for Entrepreneurship | CU Denver Business School
Molson Coors
Metropolitan State University
Newmark
Park Central
Strat Labs
Sweater Ventures
Tea With Tae
Xactly

Developer Track Sponsor



John Dietz & Jared Edwards



john@kubefirst.io



jared@kubefirst.io

Kubernetes Wins



kubefirst.io

CNCF Cloud Native Interactive Landscape

The Cloud Native Community is THRIVING!

App Definition and Development

Database: KV, V, etc.
Streaming & Messaging: cloudevents, NATS, etc.
Application Definition & Image Build: HELM, Backstage, KubeVirt, etc.
Continuous Integration & Delivery: argo, flux, etc.

Automation & Configuration: Kubescape, BOSH, etc.
Container Registry: Quay, etc.
Security & Compliance: Falco, in-toto, Kyverno, etc.
Key Management: spiffe, SPIFFE, etc.

CLOUD NATIVE Landscape

This landscape is intended as a map through the previously uncharted terrain of cloud native technologies. There are many routes to deploying a cloud native application, with CNCF Projects representing a particularly well traveled path.

l.cncf.io

Orchestration & Management

Scheduling & Orchestration: kubernetes, etc.
Coordination & Service Discovery: Consul, etcd, etc.
Remote Procedure Call: gRPC, etc.
Service Proxy: envoy, etc.
API Gateway: Kong, etc.
Service Mesh: Linkerd, etc.

Members

Grid of member logos.

Serverless

Grid of serverless logos.

Runtime

Cloud Native Storage: MinIO, etc.
Container Runtime: containerd, etc.
Cloud Native Network: Cilium, etc.

Observability and Analysis

Monitoring: Prometheus, etc.
Logging: fluentd, etc.
Tracing: Jaeger, etc.
Chaos Engineering: Litmus, etc.
Continuous Optimization: etc.

Platform

Certified Kubernetes - Distribution: etc.
Certified Kubernetes - Hosted: etc.
Certified Kubernetes - Installer: etc.
PaaS/Container Service: etc.

Special

Kubernetes Certified Service Provider: etc.
Kubernetes Training Partner: etc.
Certified CNFs: etc.

CD Foundation Landscape

Grid of CD Foundation logos.

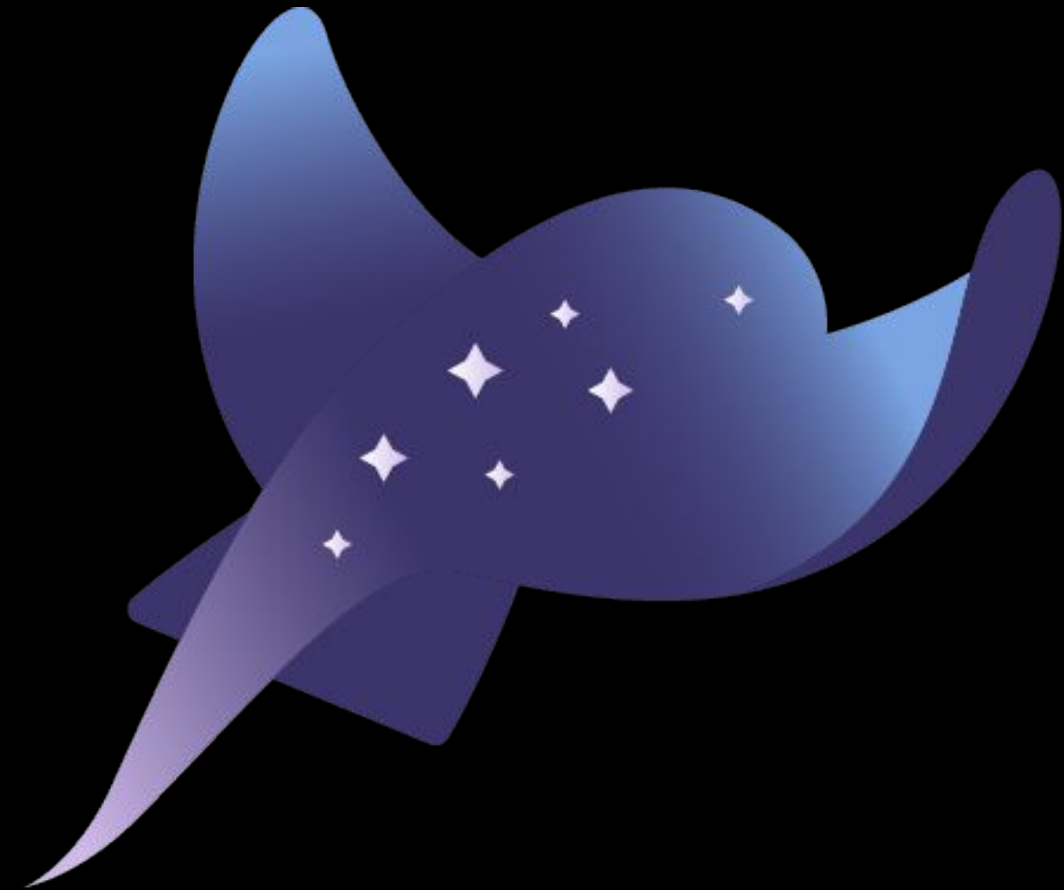
just pick the purple ones?



Your first 8 cloud-native platform needs (in order)

- ❖ git provider
- ❖ infrastructure as code
- ❖ gitops driver
- ❖ secrets manager
- ❖ artifact repository
- ❖ continuous integration
- ❖ authentication & authorization
- ❖ observability

how fun!
let's see it!



1. Git Provider

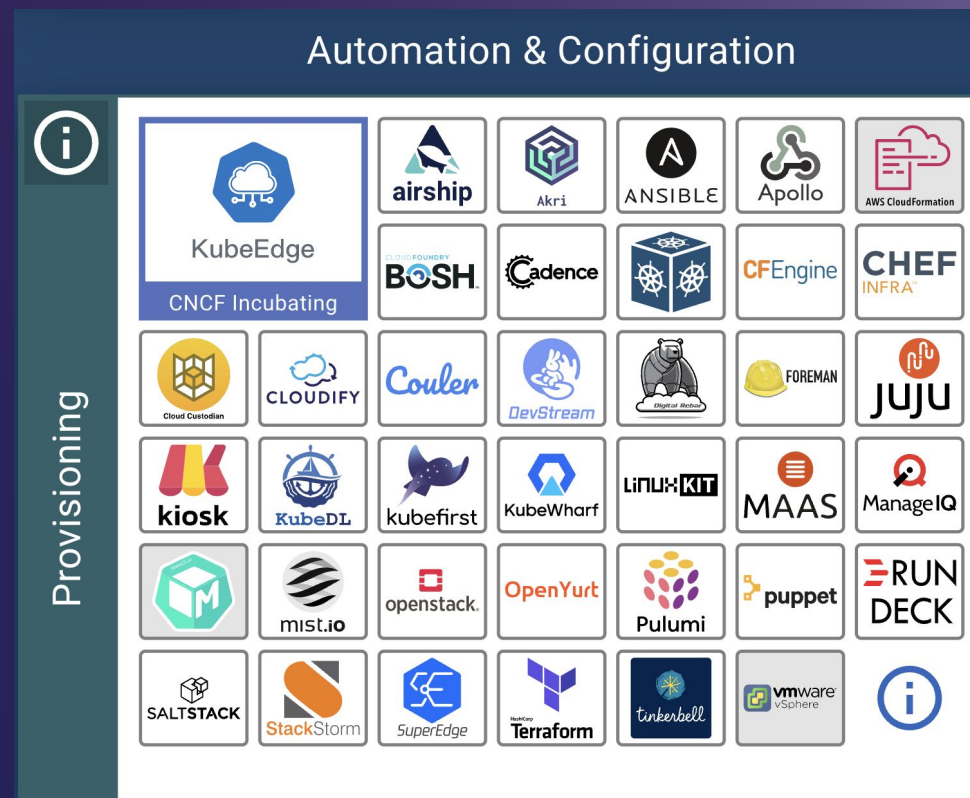


GitHub



GitLab

2. Infrastructure as Code

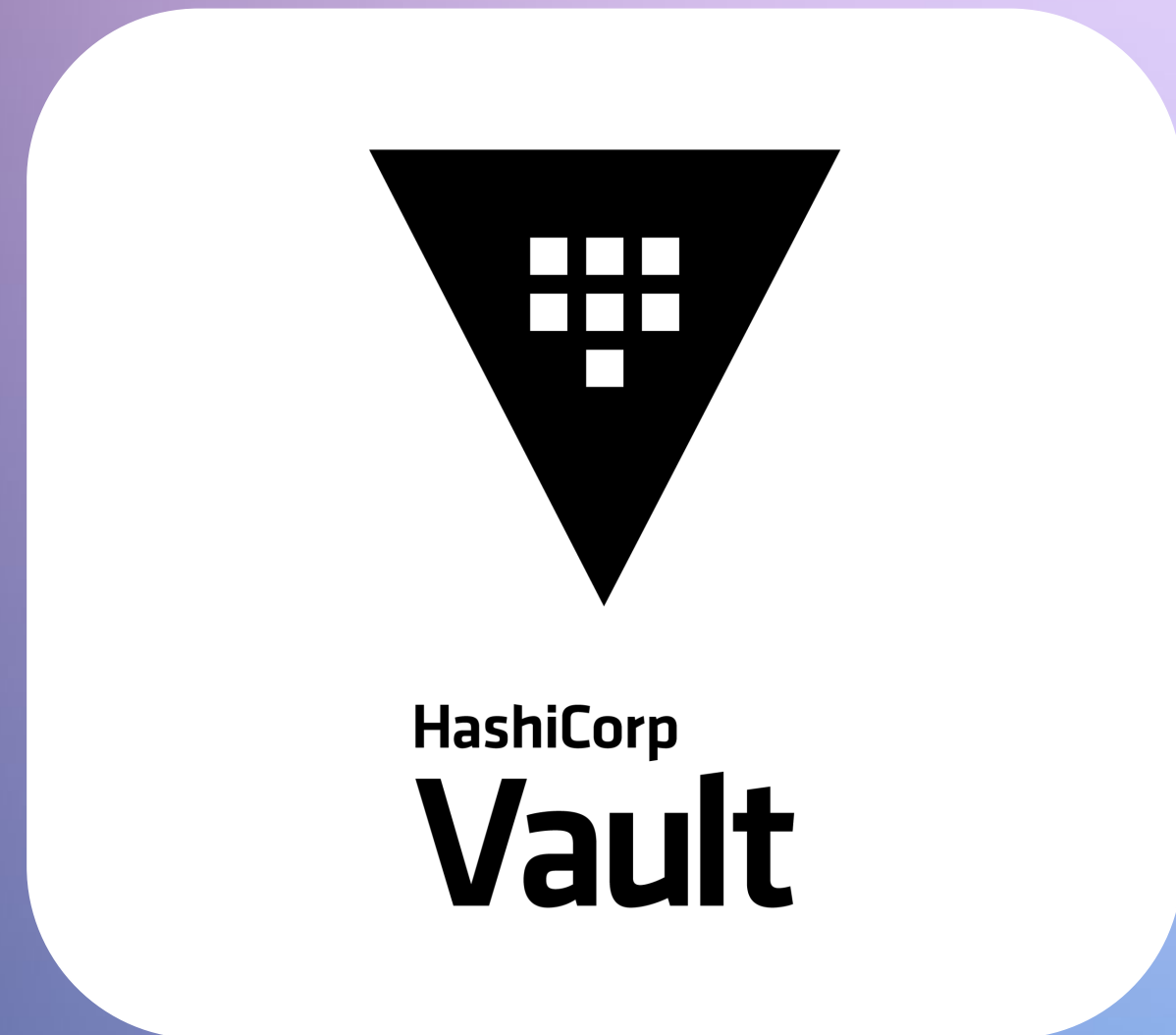


3. GitOps Driver

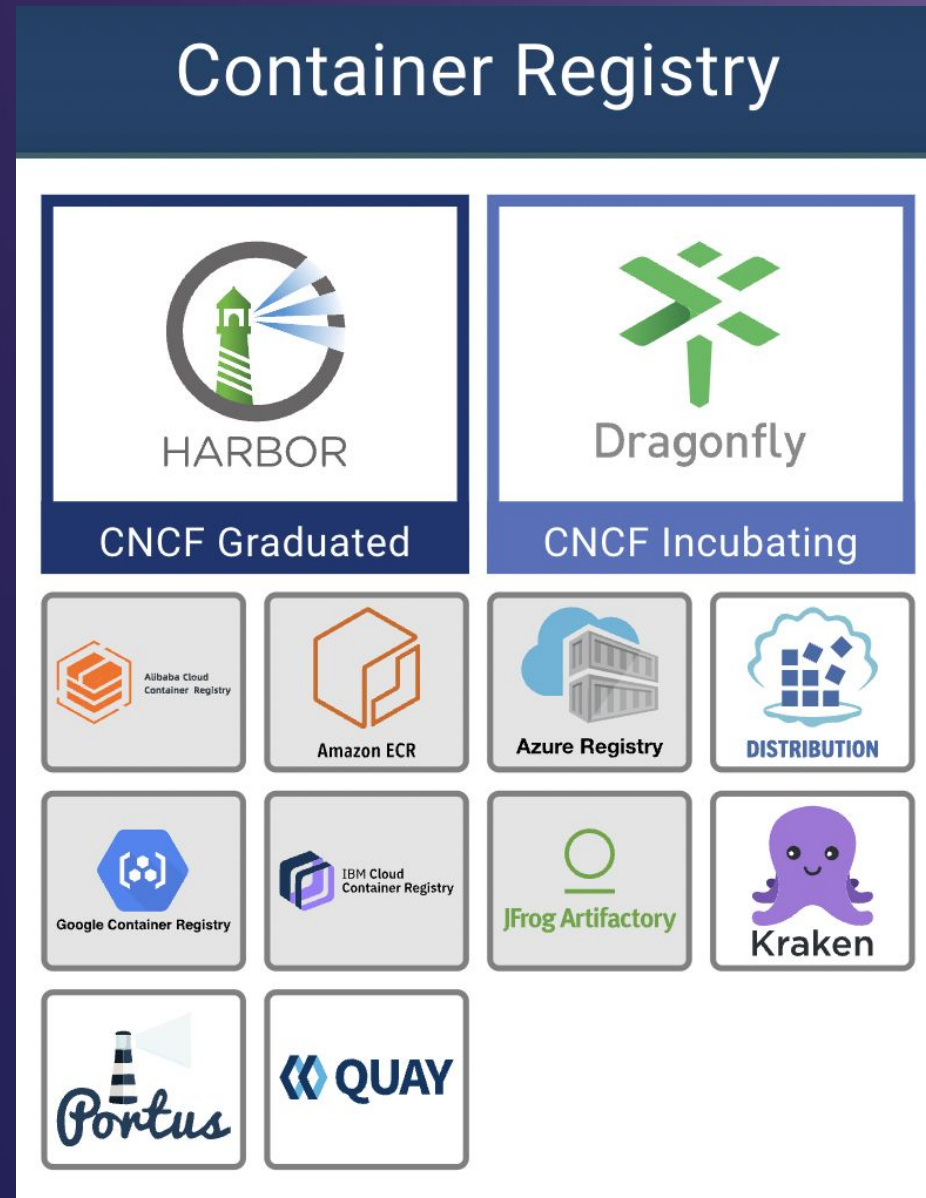


4. Secrets Manager

- ❖ secrets storage
- ❖ ephemeral credentials
- ❖ pki infrastructure
- ❖ identity provider
- ❖ oidc provider
- ❖ cloud access leasing
- ❖ single sign on
- ❖ secure and open source
- ❖ scalable and recoverable



5. Artifact Repository



6. Continuous Integration



GitHub Actions



argo



7. Authentication and Authorization

- ❖ establish roles
- ❖ establish access controls and iam
- ❖ bind roles to cloud access
- ❖ bind roles to kubernetes access
- ❖ bind roles to platform tools access
- ❖ automate engineering onboarding and offboarding
- ❖ enforce two factor authentication

least
privilege!

8. Observability



- ❖ logging
- ❖ application performance monitoring
- ❖ resource utilization
- ❖ monitoring
- ❖ dashboarding
- ❖ anomaly detection
- ❖ incident management

Recap

- ❖ establish git
- ❖ put your iac in git and automate it
- ❖ establish the gitops driver
- ❖ gitops your secrets manager
- ❖ terraform your artifact repository
- ❖ gitops your continuous integration
- ❖ terraform authentication & authorization
- ❖ gitops your observability

Scenarios To Avoid

- ❖ Don't establish CI without a secrets manager
- ❖ Don't start scripting: leverage gitops
- ❖ Don't run local terraform: automate it
- ❖ Don't delay single sign on



don't start
with tech debt



kubefirst open-source platform



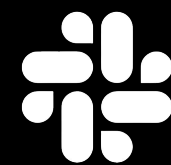
Thank You
Denver!!!



kubefirst



github.com/kubefirst/kubefirst



bit.ly/kubefirst-slack



kubefirst.io