

Building a Cutting-Edge Kubernetes Developer Platform at NVIDIA

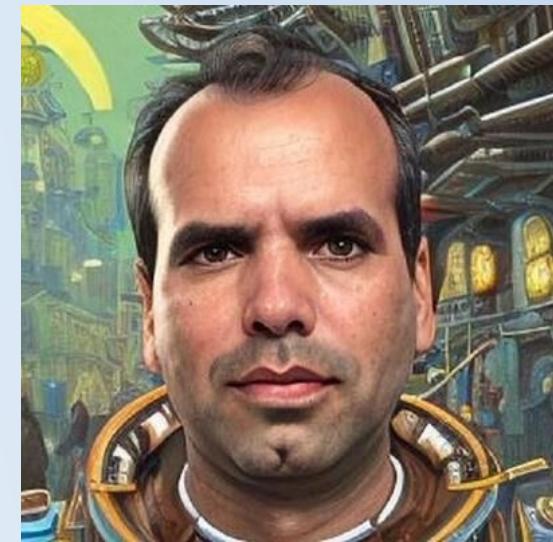


ArgoCon
NORTH AMERICA

November 12, 2024
Salt Lake City



Feng Zhou
Sr Staff Engineer
NVIDIA

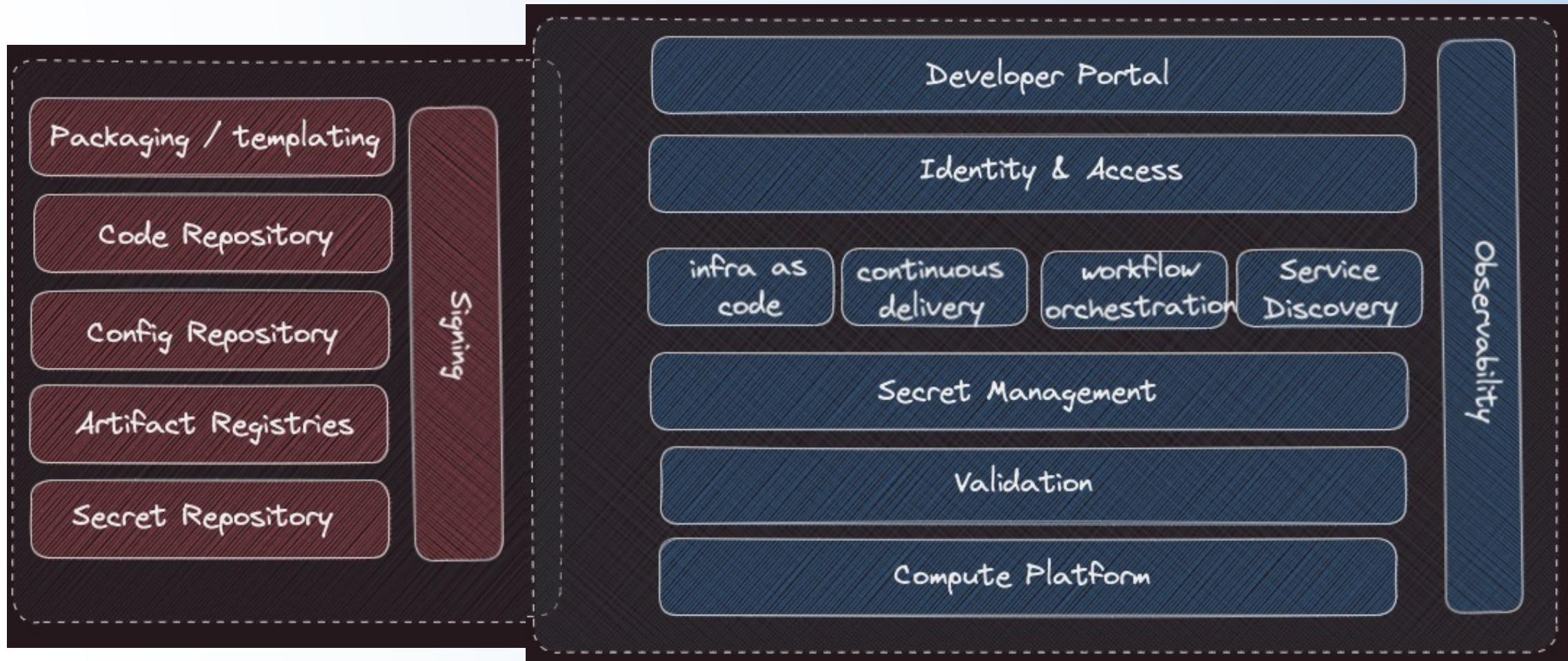


Carlos Santana
Sr Specialist Architect
AWS

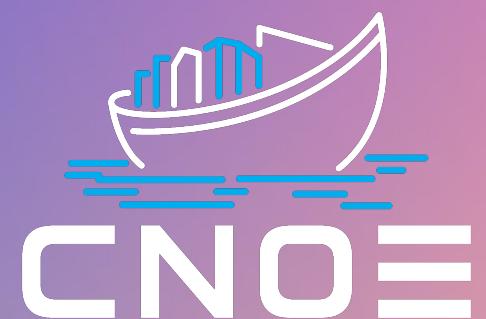
Internal Developer Platform - Capabilities



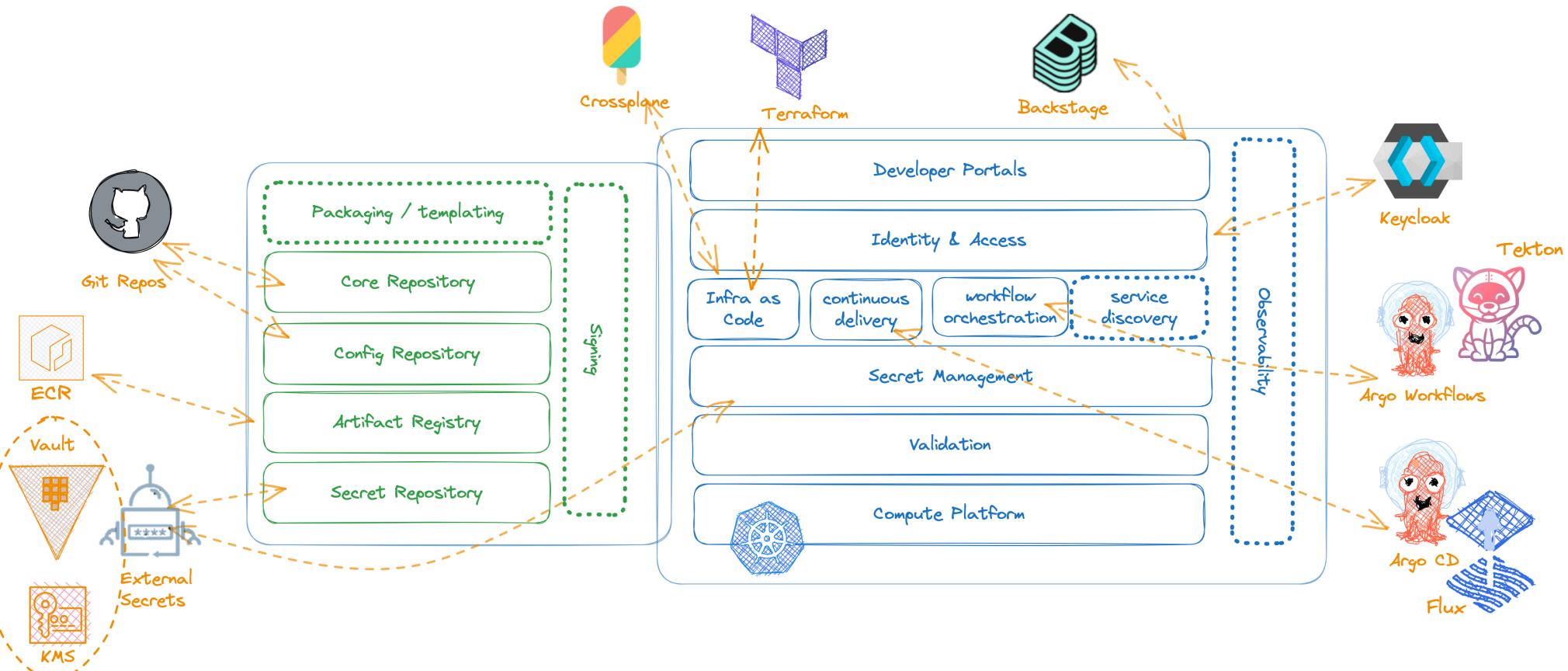
ArgoCon
NORTH AMERICA



**CNOE is open community collaboration
with the goal to help facilitate platform
engineering through the sharing of guidance,
tooling for internal developer platform (IDP)**



Reduced Technology Choices



Why an Internal Developer Platform?

Why?

- Easy (EASY) containerized app deploy
- Fast deployment time – reduced repeat work
- Extension of what Cloud Foundation offer (from Infra to Platform)
- Provide key components of success that developers can't easily do:
 - Security & Monitoring & Identity & Cost Management
- Standardization: features/addons with no effort

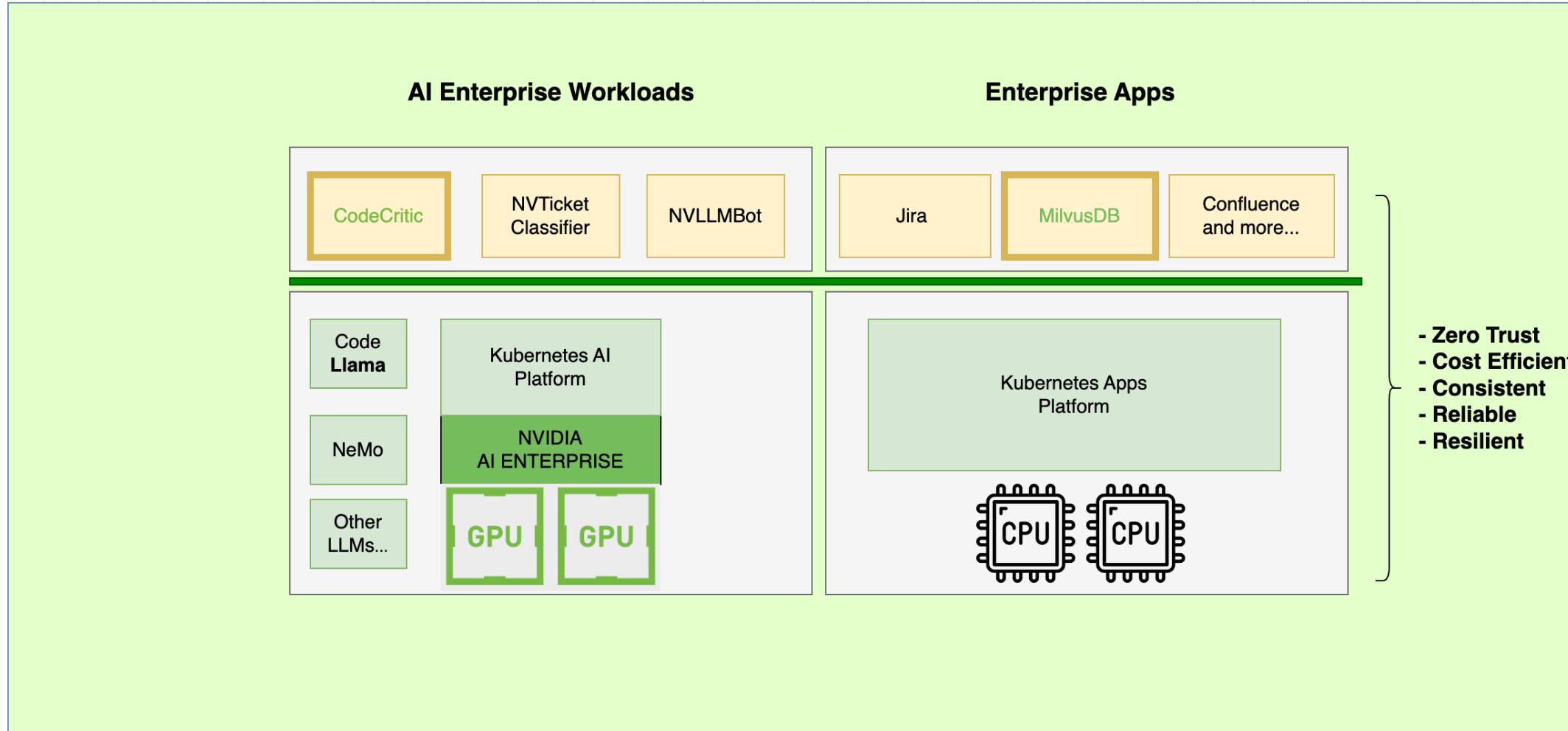
What?

- A Goto place (prebuilt) to click & deploy
- CI/CD ready to go
- Can deploy both app (helm chart) and cloud resources
- Self-service style request
- Runs in multiple env, multiple cloud
- From Simple to Complex apps

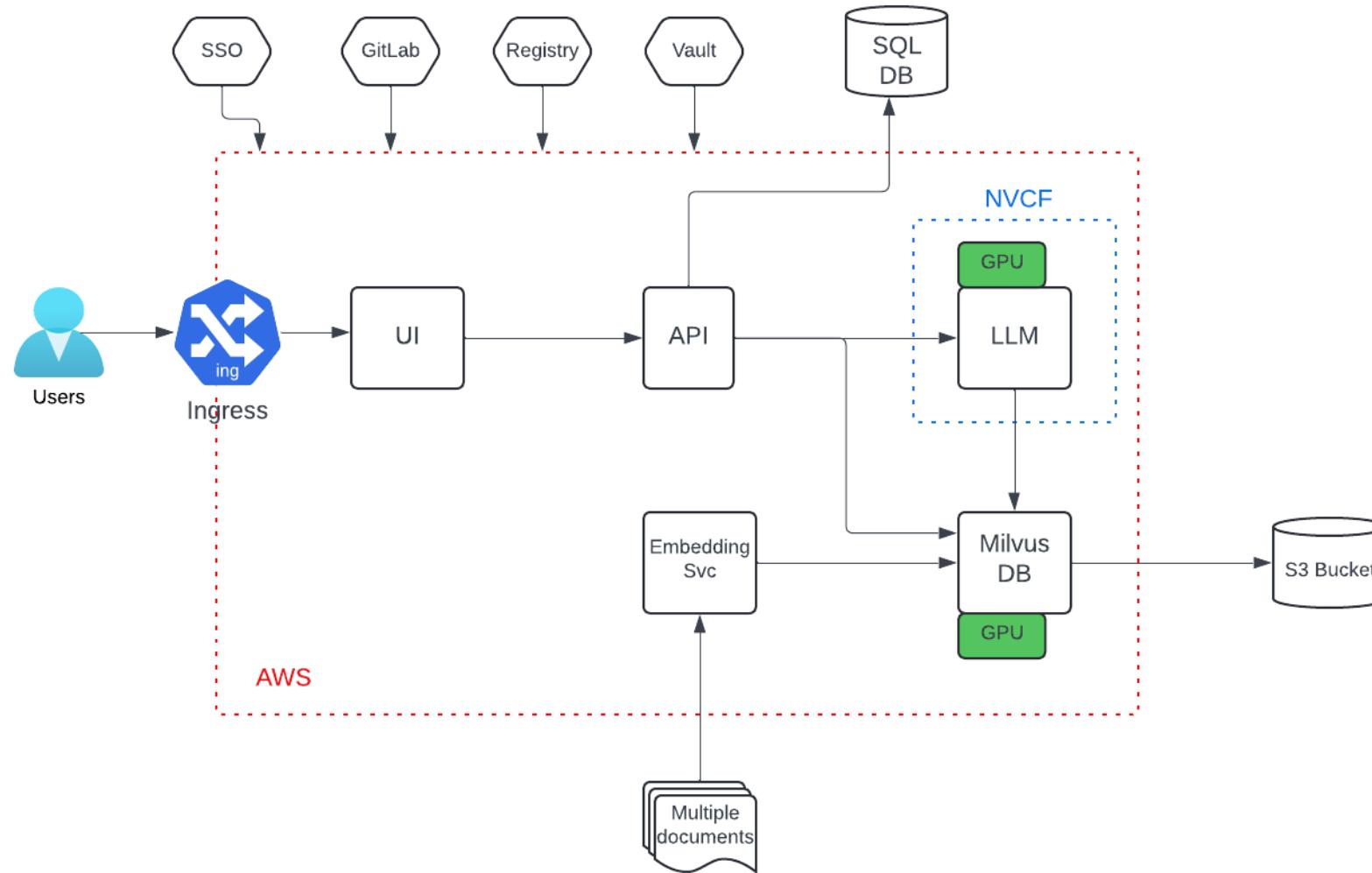
Who for?

- Enterprise App
- AI + LLM App
- Milvus Database hosting
- Example: Code Critic, JIRA, Mandap, Nvbugs/Nvbot
- 30+ apps rapidly growing

Architecture

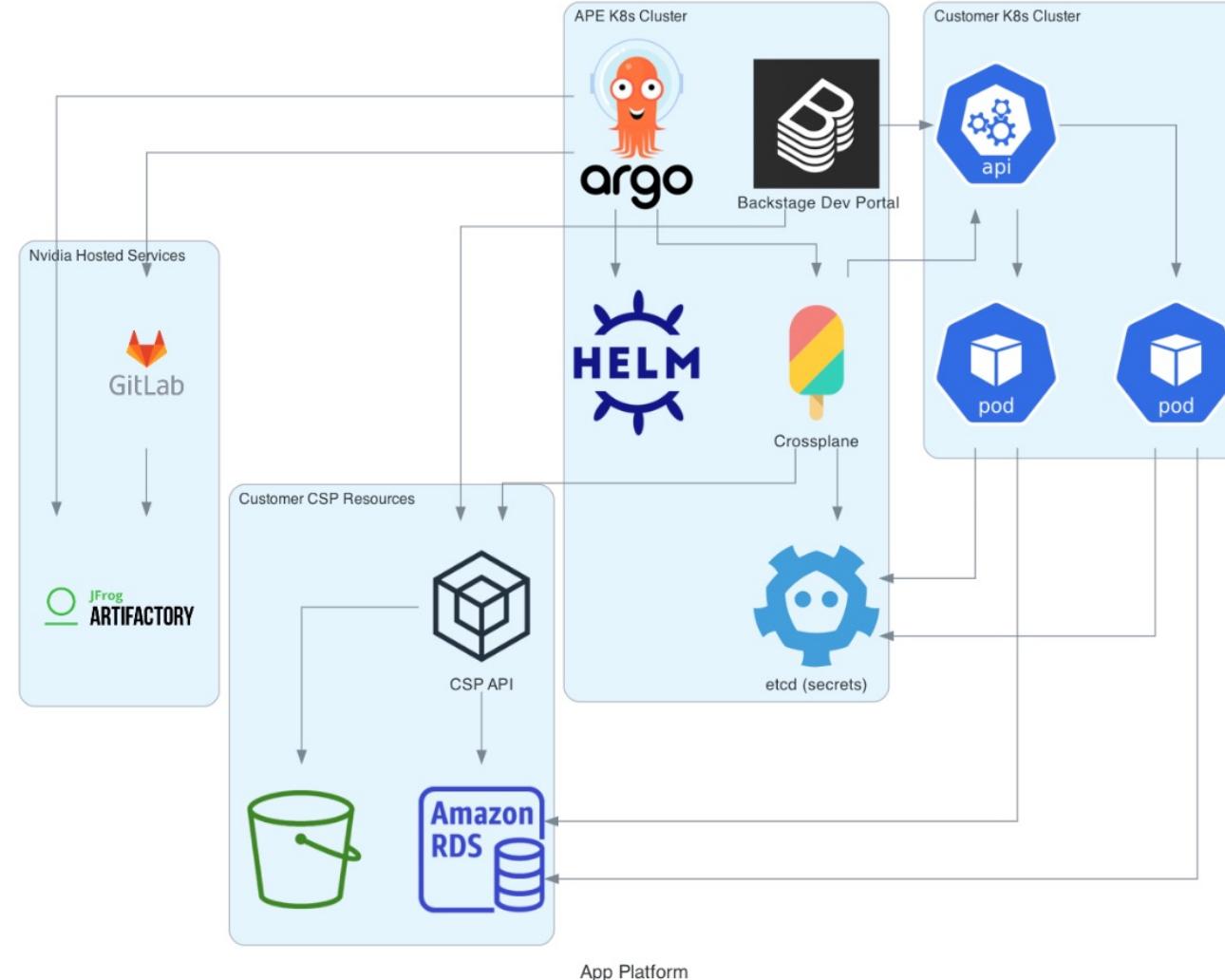


Typical AI Bot Service Use Case



High Level Design II

- From 0 to N

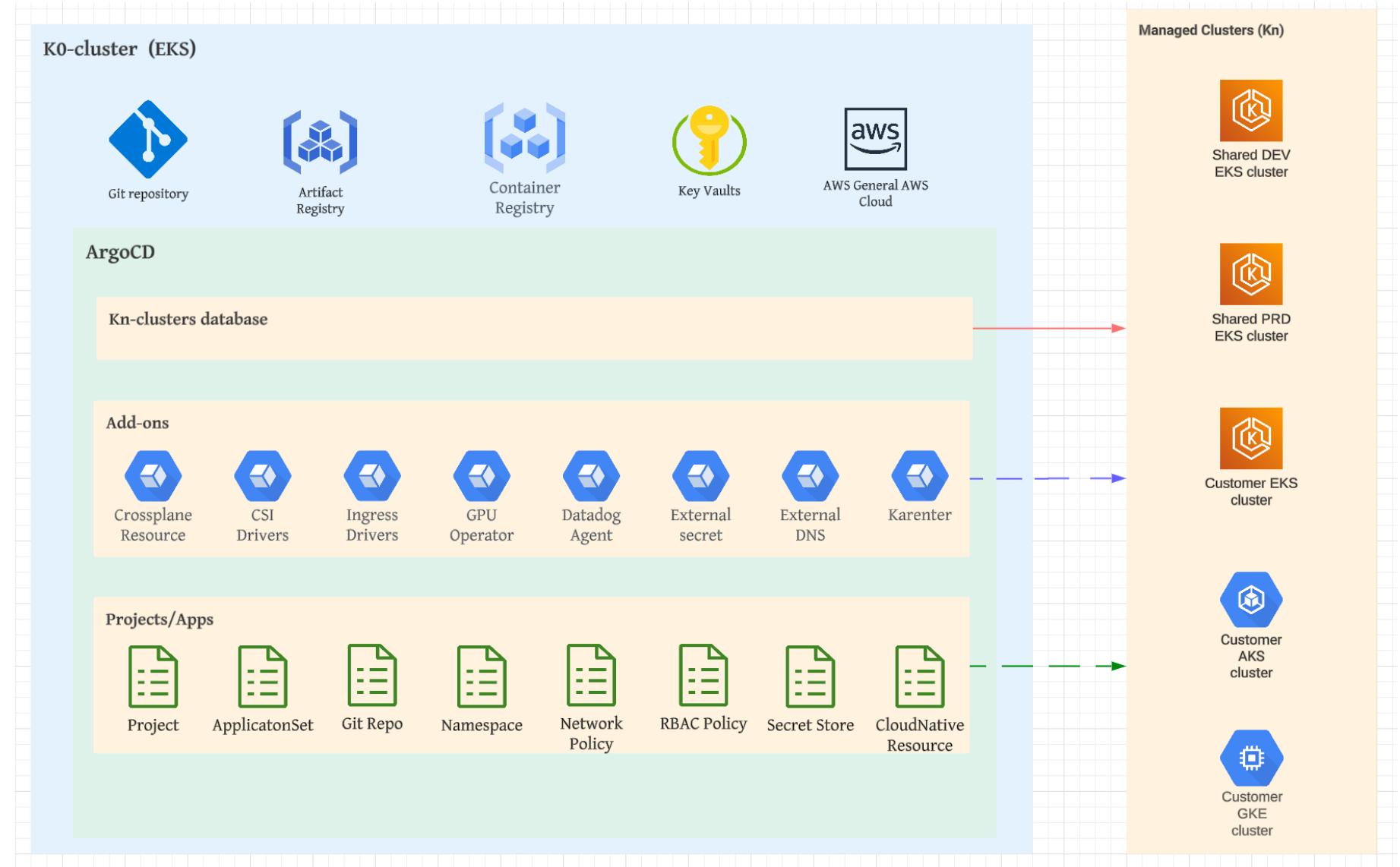


Warp I: Overall Architecture



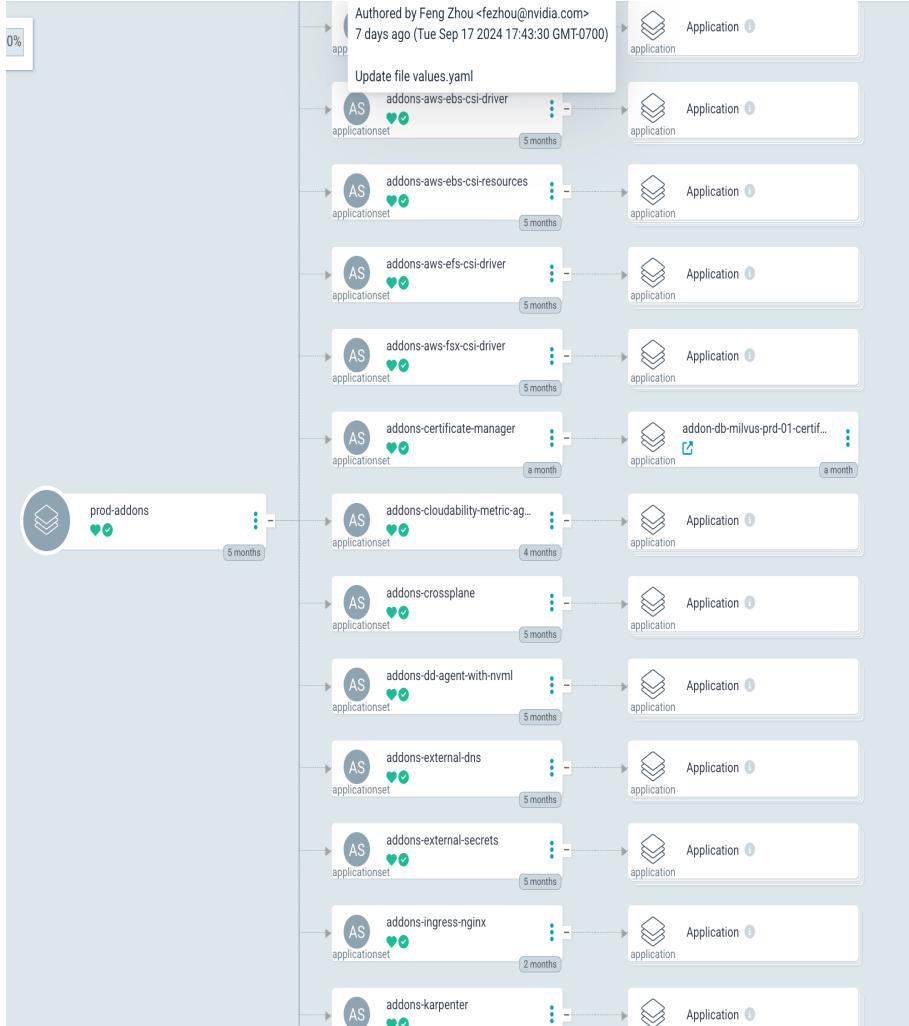
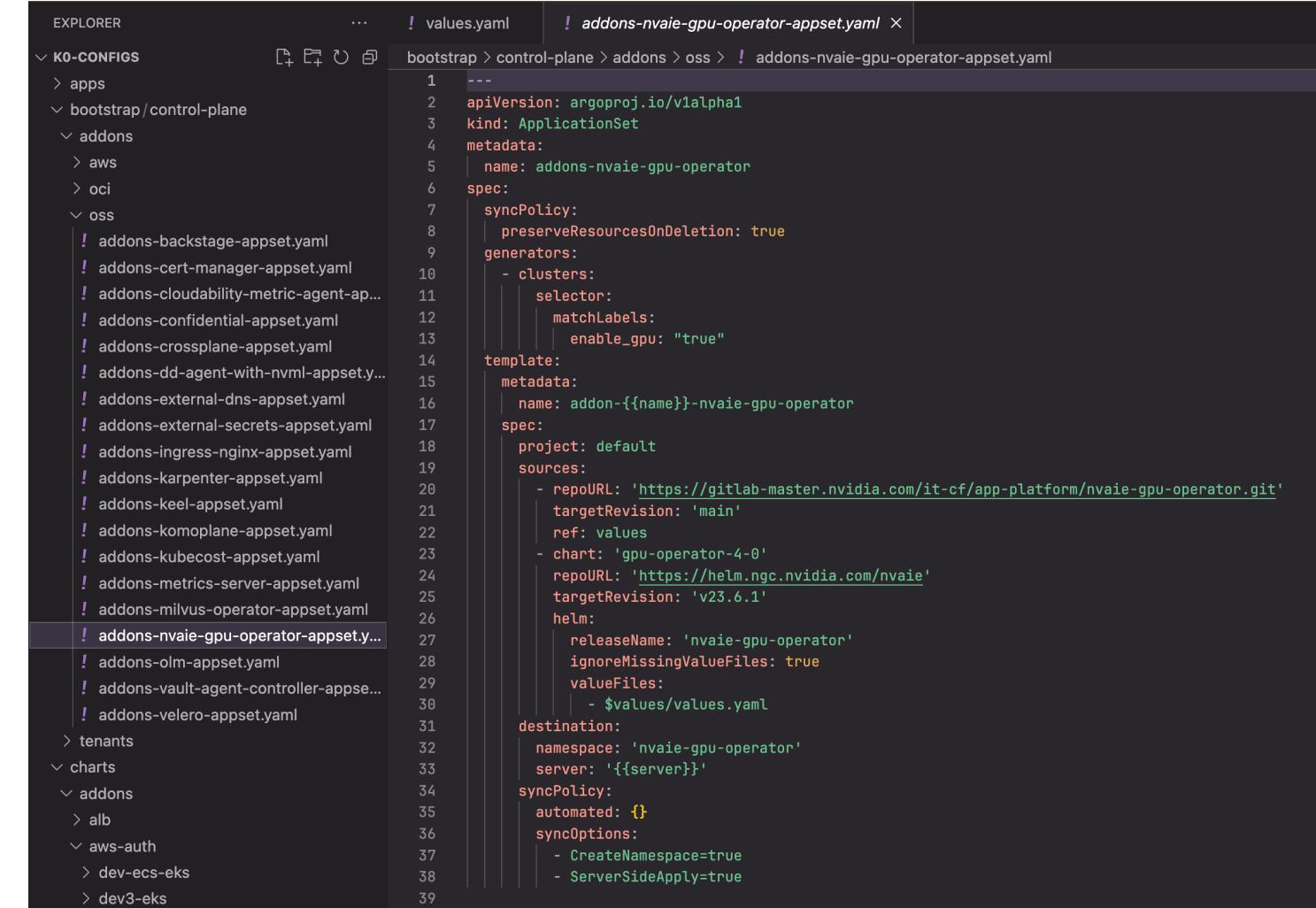
ArgoCD/Crossplane/Git based infra deployment with K0 & Kn clusters

K0=Control Plane Cluster
Kn = Workload Clusters



Addons via ApplicationSets

Based on AWS EKS Blueprints GitOps Bridge patterns

```

EXPLORER ... ! values.yaml ! addons-nvae-gpu-operator-appset.yaml X
bootstrap > control-plane > addons > oss > ! addons-nvae-gpu-operator-appset.yaml
1 --- apiVersion: argoproj.io/v1alpha1
2 kind: ApplicationSet
3 metadata:
4   name: addons-nvae-gpu-operator
5 spec:
6   syncPolicy:
7     | preserveResourcesOnDeletion: true
8   generators:
9     - clusters:
10       selector:
11         matchLabels:
12           enable_gpu: "true"
13   template:
14     metadata:
15       name: addon-{{name}}-nvae-gpu-operator
16   spec:
17     project: default
18   sources:
19     - repourl: "https://gitlab-master.nvidia.com/it-cf/app-platform/nvae-gpu-operator.git"
20       targetRevision: 'main'
21       ref: values
22     - chart: 'gpu-operator-4-0'
23       repourl: "https://helm.ngc.nvidia.com/nvae"
24       targetRevision: 'v23.6.1'
25       helm:
26         releaseName: 'nvae-gpu-operator'
27         ignoreMissingValueFiles: true
28         valueFiles:
29           - $values/values.yaml
30   destination:
31     namespace: 'nvae-gpu-operator'
32     server: '{{server}}'
33   syncPolicy:
34     automated: {}
35   syncOptions:
36     - CreateNamespace=true
37     - ServerSideApply=true
38
39

```

<https://github.com/gitops-bridge-dev>

Warp II: Onboarding projects/apps



structured approach with project YAML + deployer helm charts



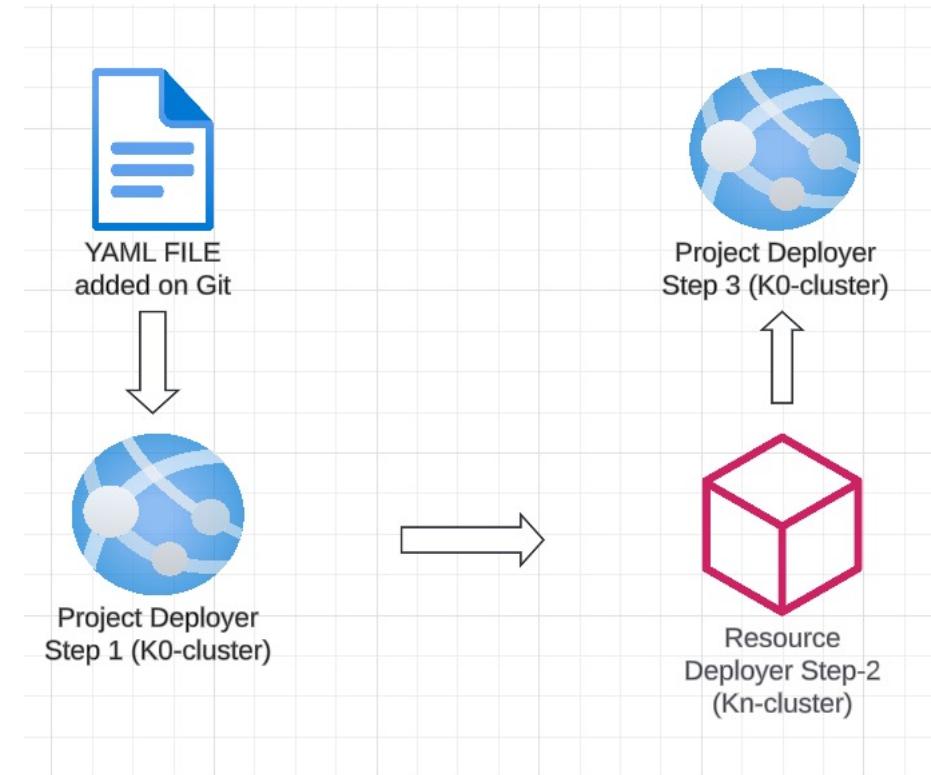
Actual Deploy Process



Projects are defined as YAML and deployed with 2 helm chart apps

```
1 department: IT
2 dl: ape
3 name: demo-app-1
4 inspect_id: XXXX-YYYY-ZZZZ
5 onboarded_by: somebody
6 primary_owner: somebody1
7 secondary_owner: somebody2
8 < clusterGenerator:
9   - clusters:
10     selector:
11       matchLabels:
12         cluster_name: dev-cluster-1
13     values:
14       env: dev
15       targetRevision: dev
16 < vaults:
17   dev:
18     server: https://test.vault.nvidia.com
19     mount_point: TestMount
20     namespace: test-ns
21     role: demo-app-role
22     version: v2
23 < repos:
24   - name: demo-app-1
25     type: git
26     url: https://gitlab-master.nvidia.com/my-test-repos/demo-app-1.git
27 < apps:
28   - code:
29     path: p1/app1
30     repoType: git
31     repoURL: https://gitlab-master.nvidia.com/my-test-repos/demo-app-1.git
32 < metadata:
33   name: app1
34   sync_wave: '10'
35
```

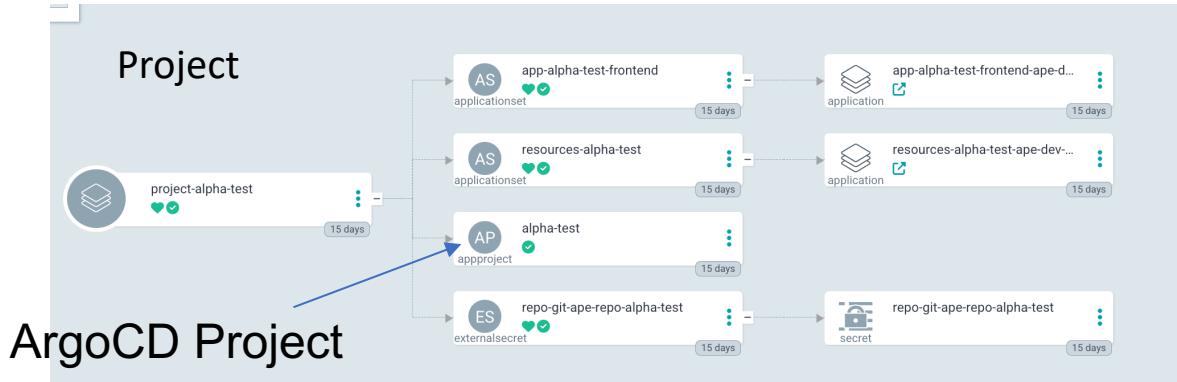
Argo CD Application Sets



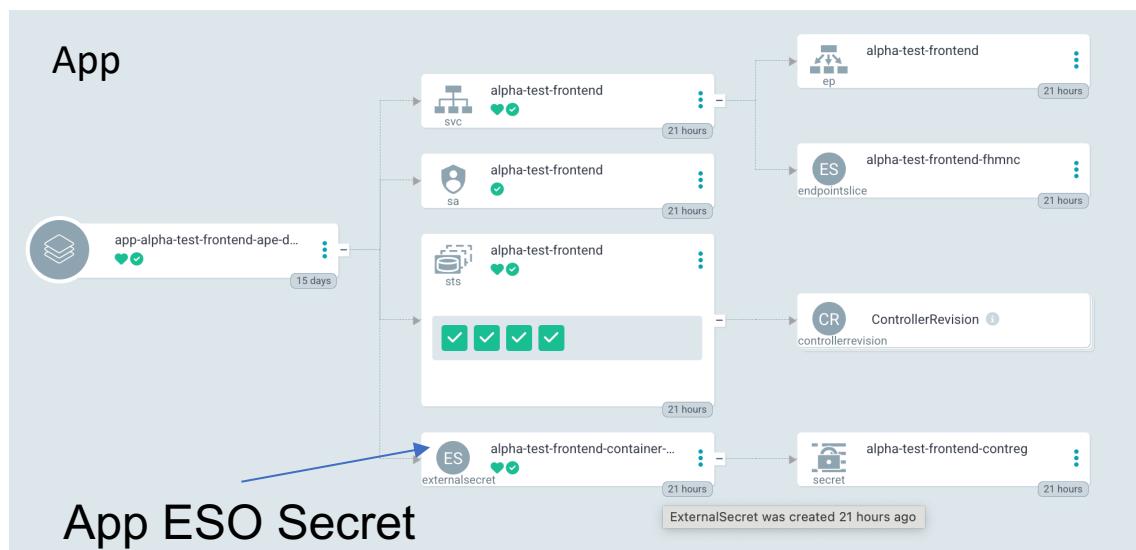
Example of deployed app



View of the project in ArgoCD UI



ArgoCD Project



App ESO Secret

Platform Team offers an App Helm Chart

```
# This is app configuration for app: frontend under project: alpha-test.
sysdev-prd:
  deployment:
    enabled: false
  statefulset:
    enabled: true
    replicaCount: 4
  autoscaling:
    enabled: false
  code:
    language: python
    serviceName: alpha-test-frontend
    version: 0.0.1
  useLatestImage: true
  container:
    image:
      name: projects/alpha-test/frontend
      tag: latest
    livenessProbe:
      enable: true
      initialDelaySeconds: 0
      path: /
      scheme: HTTP
    name: demo-app
    readinessProbe:
      enable: true
      initialDelaySeconds: 0
      path: /
      scheme: HTTP
    startupProbe:
      enable: true
      initialDelaySeconds: 0
      path: /
      scheme: HTTP
  datadog:
    enabled: true
    libVersion: v2.6.2
  deploymentEnv: dev
```

The Platform is Opinionated

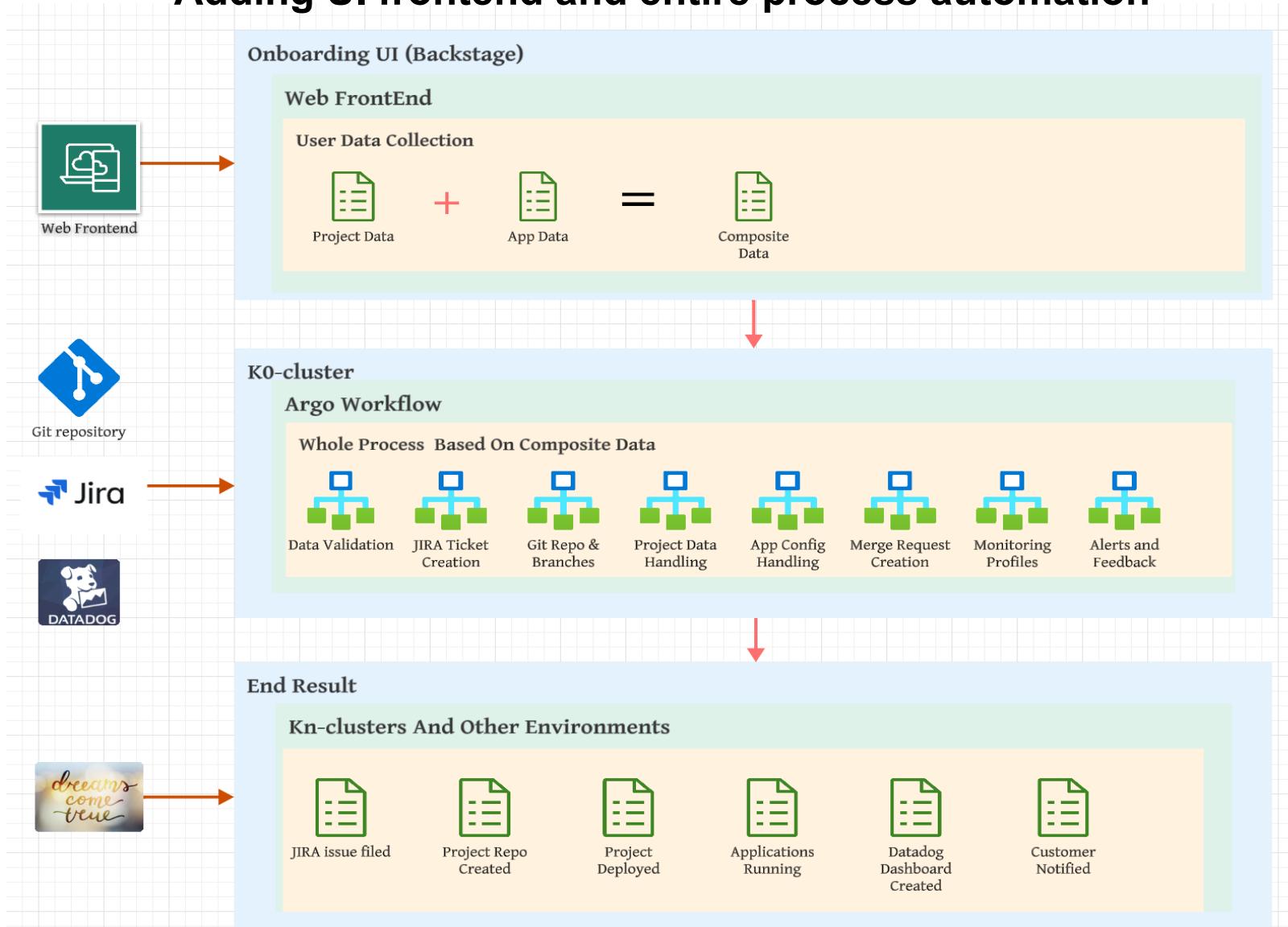
Assumptions are made to best support our customers

- The platform is opinionated
 - What are the essential components
 - How services/resources should be organized
 - How best to deploy cloud native resources
- The application is opinionated
 - Three types of application supported
 - Plain manifests
 - Helm chart + values files (app team responsible for helm chart)
 - Simple docker container – helper helm chart needed
 - What helper helm chart can do (platform team responsible for helm chart)
 - Ingress config (shared ALB)
 - Vault secret handling
 - SSO integration

Warp III: E2E deployment with Argo Workflow



Adding UI frontend and entire process automation



NVIDIA Onboarding UI & Workflow



Sysdev Fusion Onboarding

Onboard a new project and application with Fusion template

1 Project Configuration - General Information 2 Project Configuration - Vault Information 3 App Configuration 4 Review

Project Name
Unique name of the project. Should not contain spaces. This project name will be used to create the GitLab repository and Kubernetes namespace.

Host
gitlab-master.nvidia.com
The host where the repository will be created

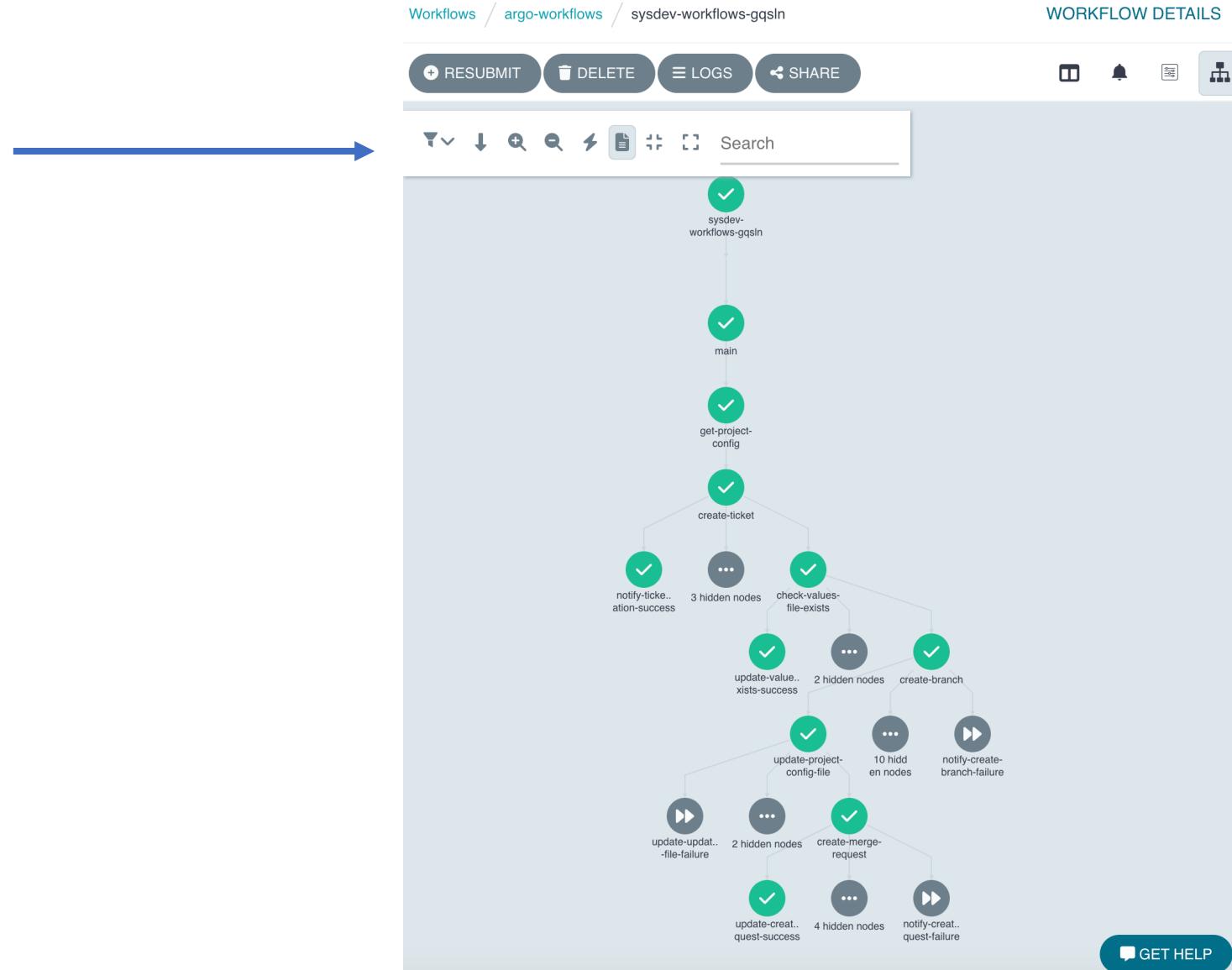
Owner Available
ape-repo/projects
GitLab namespace where this repository will belong to. It can be the name of organization, group, subgroup, user, or the project.

Repository*
The name of the repository

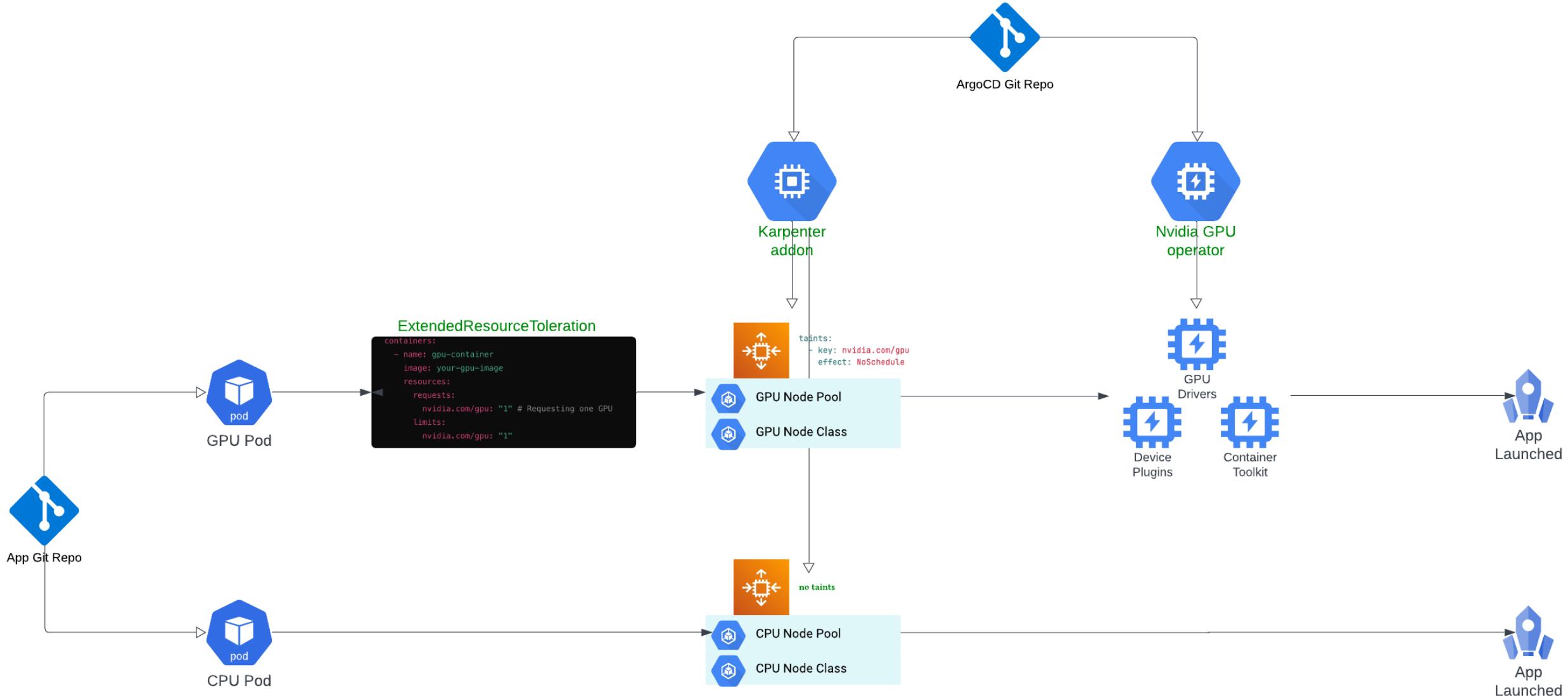
Select environments (Each environment corresponds to a different Kubernetes Cluster).
 dev
 stg
 prd

nSpect program ID
Existing nSpect program ID

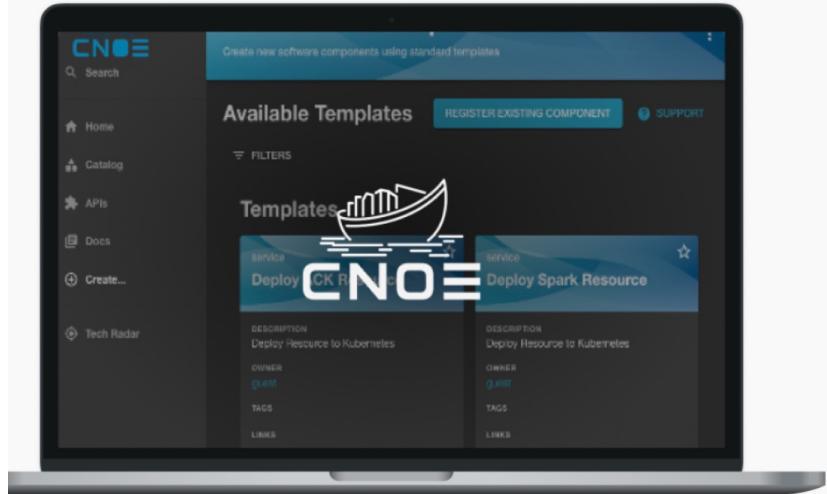
Select DL for access control*
Distribution List (DL) to access the project on-boarded to Fusion.



Getting GPUs For Your App



IDP Demo with CNOE



Leverage Open Source
Technology and build
your Developer
Platform now!

[Get Started](#)

[What is CNOE](#)

[Contribute](#)



CHOOSE STACK



BUILD STACK



TEST STACK



ADOPT STACK



SCALE STACK

<https://cnoe.io>

<https://github.com/cnoe-io/idpbuilder>

<https://github.com/cnoe-io/stacks>

Stack for today

Backstage

Argo CD

Argo Workflows

Argo Events

External Secret Operator

Git and OCI (gitea)

Keycloak

NGINX Ingress

kind

> idpbuilder create --help

Benefits

<h2>Convenience</h2> <p>Easy deploy of App, even without AWS acct</p> <p>Declarative infra, git -> deploy in seconds</p> <p>Argo UI consolidated view of workloads status & troubleshoot</p> <p>Cluster maintenance taken care of</p> <p>Reduced hours to deploy/development cycle</p>	<h2>Features</h2> <p>Loads of add-ons already configured</p> <p>Able to create cloud resource with Crossplane declaratively</p> <p>Able to work on multi-cluster (multi-cloud in future)</p> <p>Cost management/Backup/Performance testing/Monitoring</p> <p>Able to build in automation for app deploy & security</p>
<h2>Security</h2> <p>Base security: SSO based access Argo and K8s namespace</p> <p>Network isolation/NodePool Isolation/mTLS</p> <p>Encrypted data in rest/transit, encrypted network traffic</p> <p>Centralized container image & pod scanning</p> <p>Cloud provider enforced best practice applied</p>	<h2>Observability</h2> <p>Base monitoring profile setup by default</p> <p>Better view of your app performance (datadog APM)</p> <p>Better view of pod resource usage (datadog & Aptio)</p> <p>higher density cluster usage save \$\$\$</p>