cloudogu

// GITOPS ADVENTURE: REPO STRUCTURES

Johannes Schnatterer, Cloudogu GmbH

Slides

@schnatterer@floss.social in/jschnatterer

Version: 202507012040-0d6fab0

🙋 What is your profession?

🤓 Software Engineer / Developer

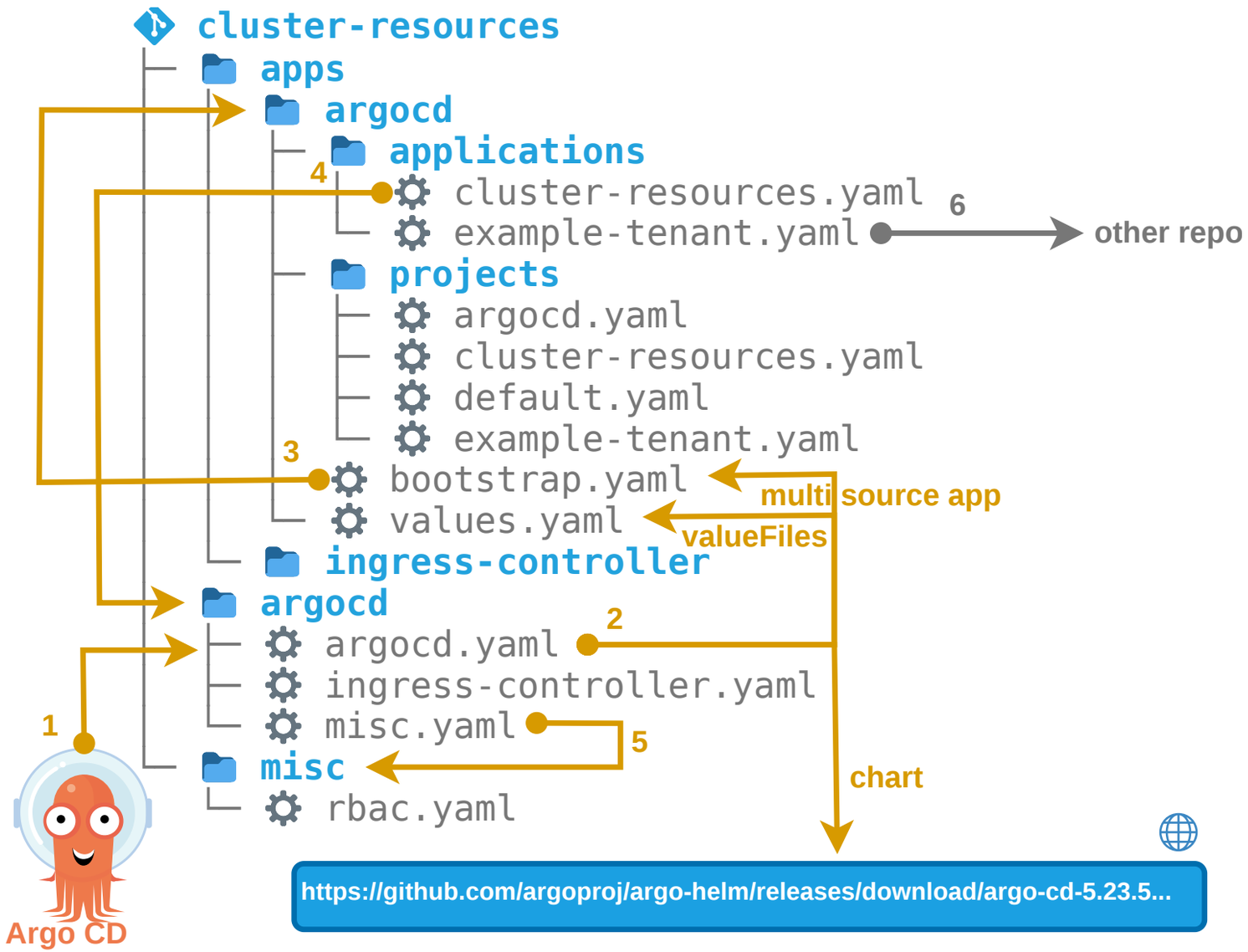
🙋 What is your profession?

🛠️ Platform Engineer / Ops person

🙋 What is your profession?

🤷 None of the above

Basic repo structure



Bootstrapping in vanilla cluster

- Deploy Argo CD + repo secret
- Apply `projects` + app `argocd`

```
# The only imperative commands you'll ever need
helm template argocd argo/argo-cd --version .. --namespace argocd --values .. \
  | kubectl apply -f-

kubectl apply ..
```

➡️ GitOps

IDP Bootstrapping made simple: GOP

Creates a complete GitOps-based operational stack / IDP on your Kubernetes clusters

cloudogu/gitops-playground

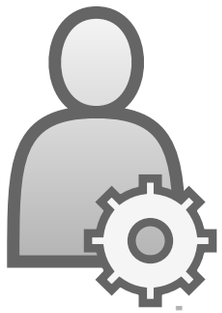
Also runs locally

```
VERSION='cec82a7' # Preview for 0.12.0
bash <(curl -s \
  "https://raw.githubusercontent.com/cloudogu/gitops-playground/$VERSION/scripts/init-cluster.sh") \
   && docker run --rm -t -u $(id -u) \
    -v ~/.config/k3d/kubeconfig-gitops-playground.yaml:/home/.kube/config \
    --net=host \
    ghcr.io/cloudogu/gitops-playground:$VERSION --yes --argocd --ingress-nginx --base-url=http://localhost
# More IDP-features: --monitoring --vault=dev --cert-manager --mailhog
# More features for developers: --jenkins --registry --content-examples
```

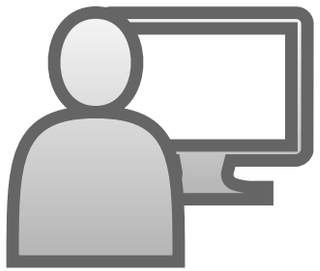
Try new repo structure with GOP:

github.com/cloudogu/gitops-talks/tree/4b91d711/src/repo-examples/basic-repo-structure

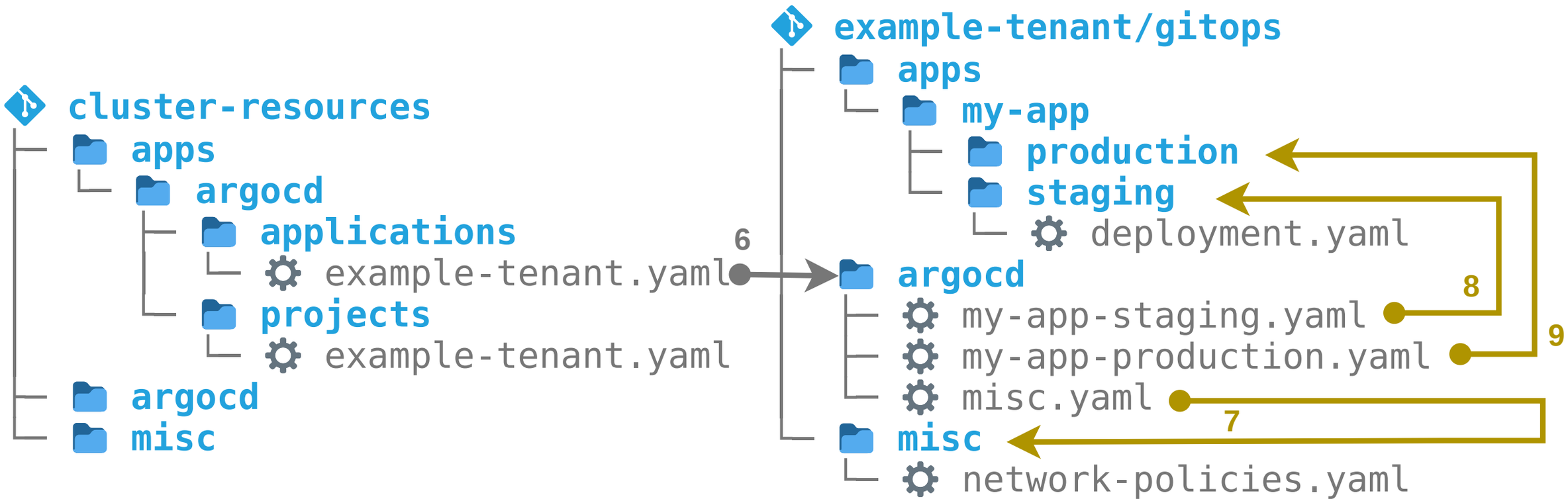
Multi-Tenancy



Platform engineer

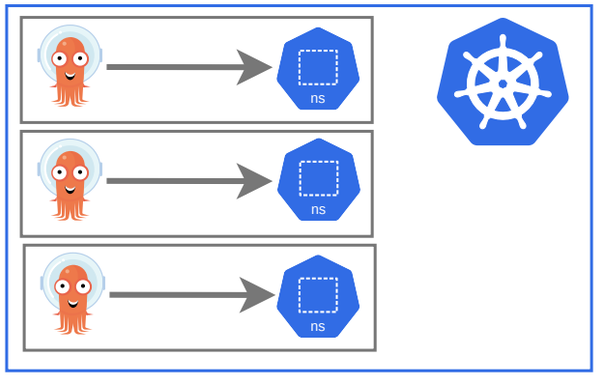


Software engineer

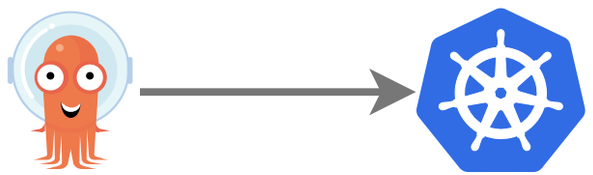


Options for multi-tenancy

- Shared
- Dedicated
  - Instance per Namespace
  - Instance per Cluster



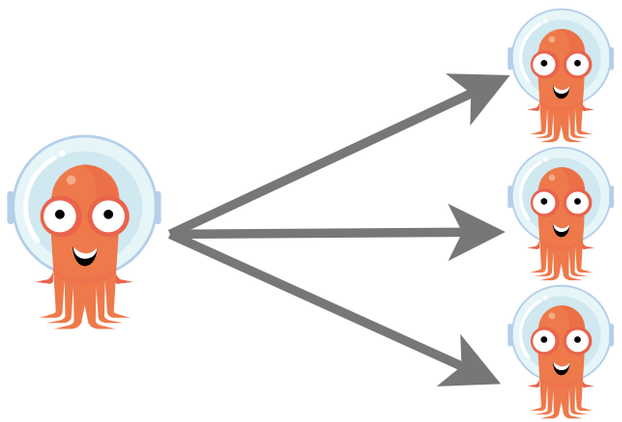
Instance per Namespace



Instance per Cluster

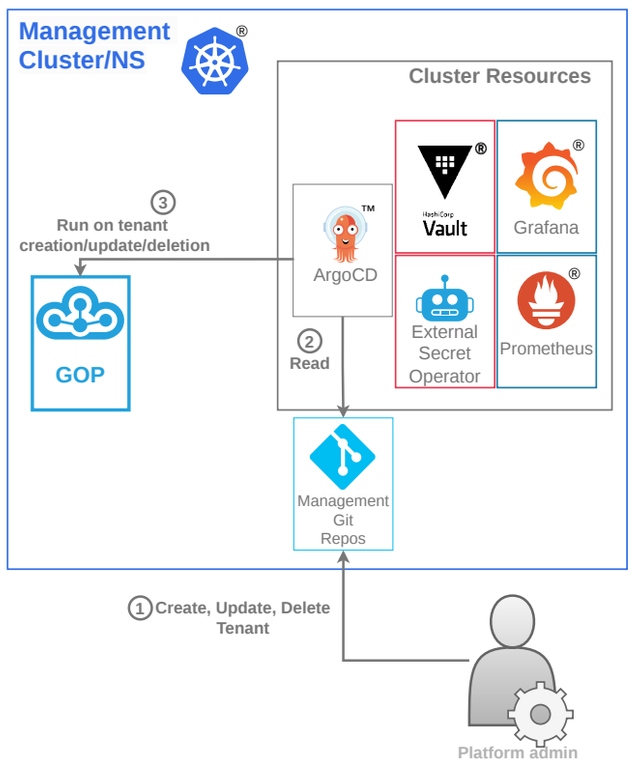
Managing dedicated instances

- Standalone
- Hub and Spoke
  - 💡 Argo CD AppSets

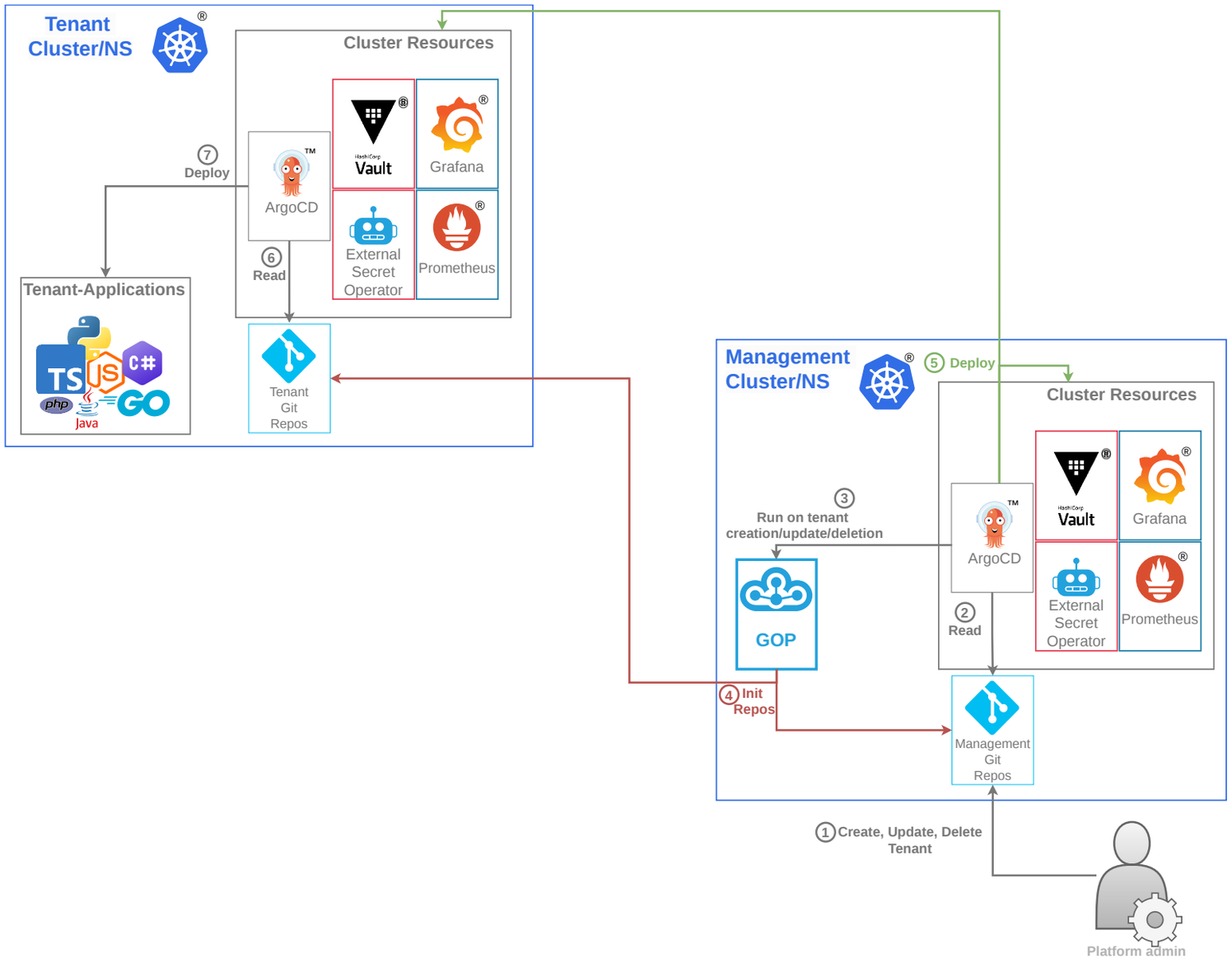


Hub and Spoke

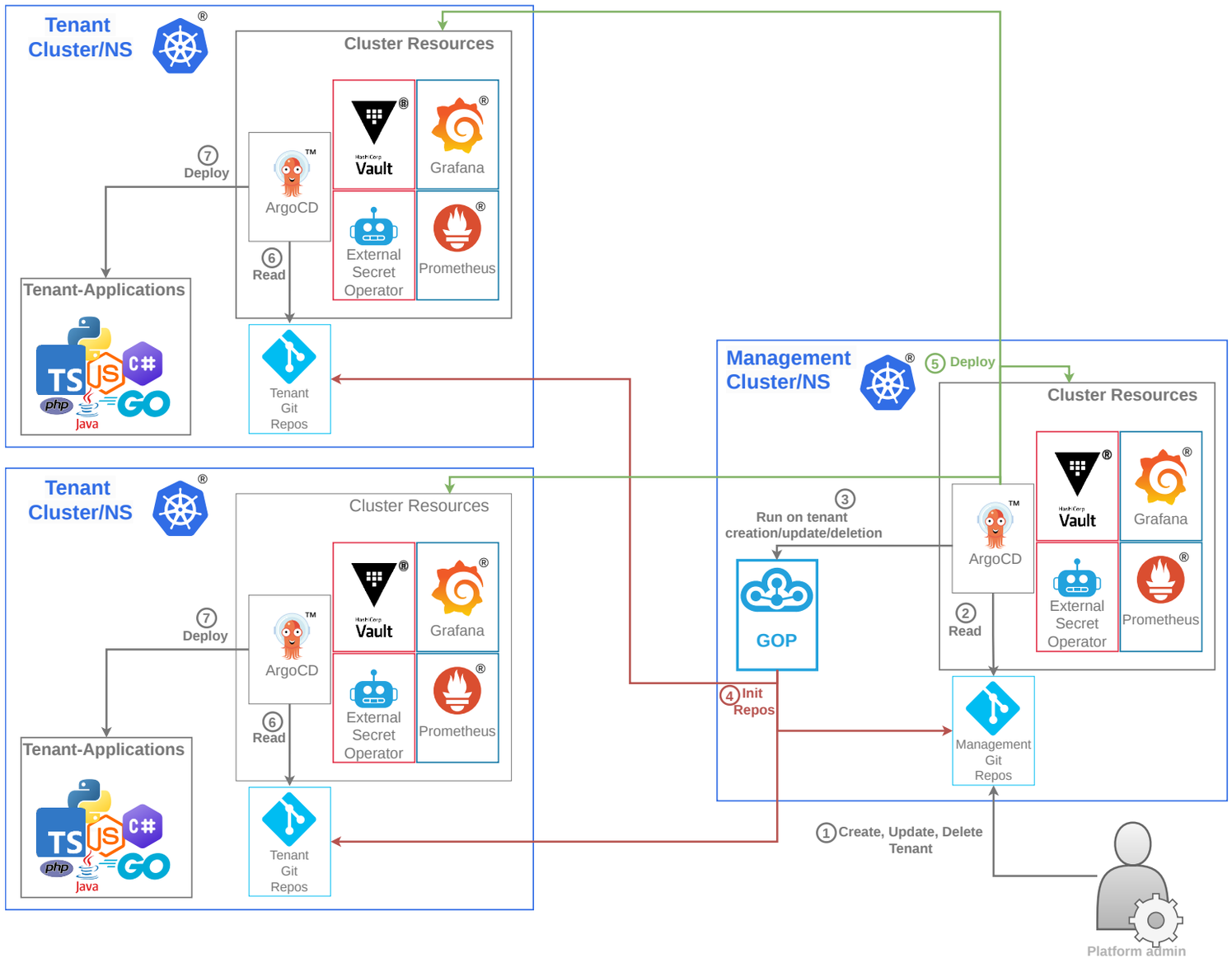
IDP as a Service using Argo CD AppSets

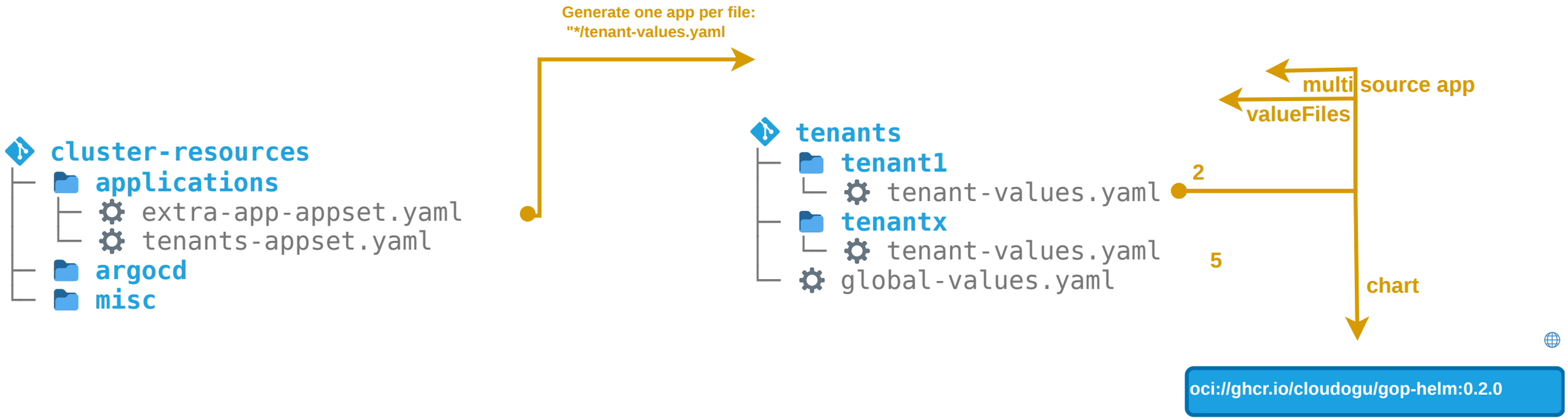


IDP as a Service using Argo CD AppSets



IDP as a Service using Argo CD AppSets

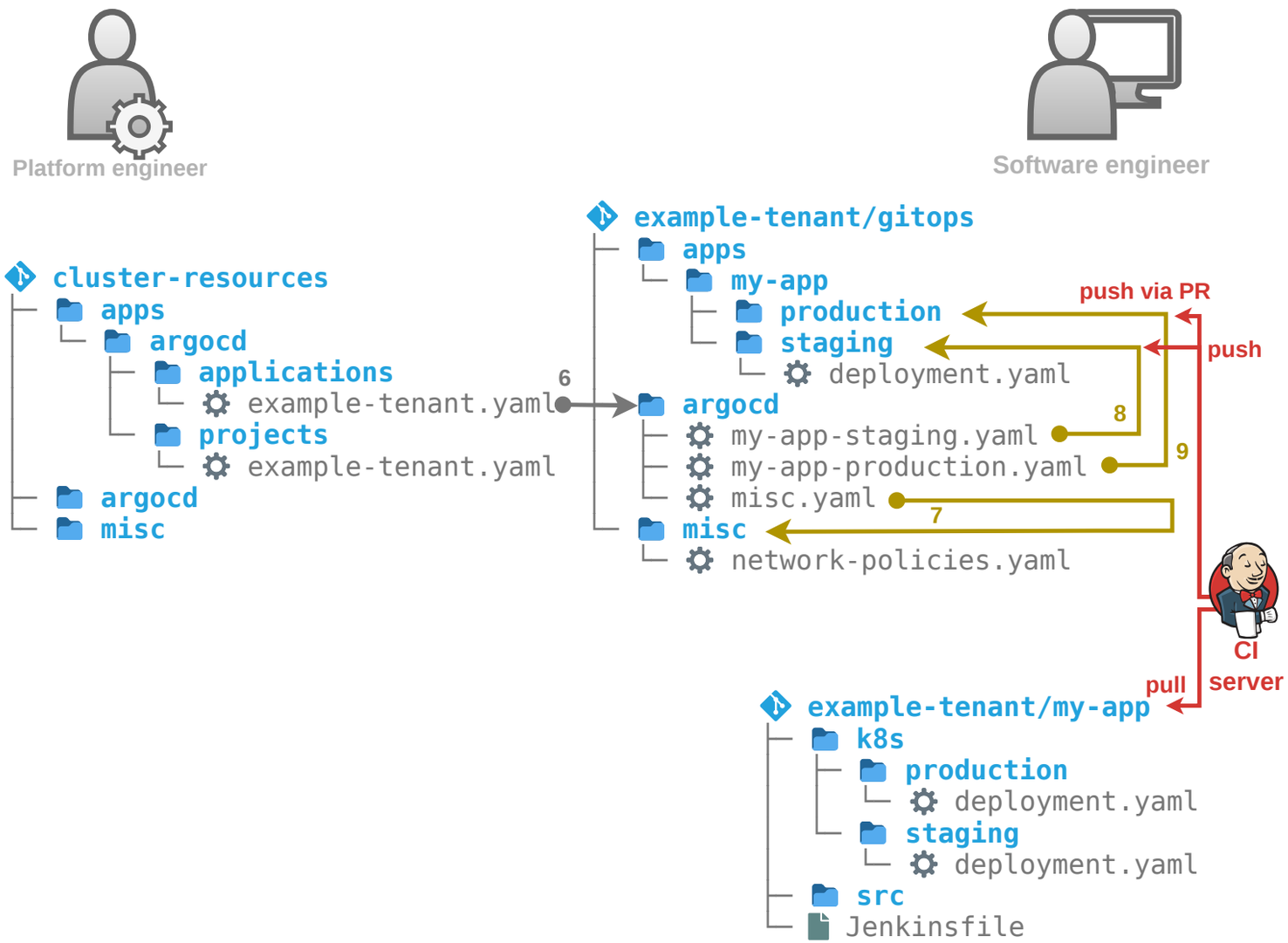




Try with GOP:

github.com/cloudogu/gitops-talks/tree/4b91d711/src/repo-examples/centralized-management-repo

Developers

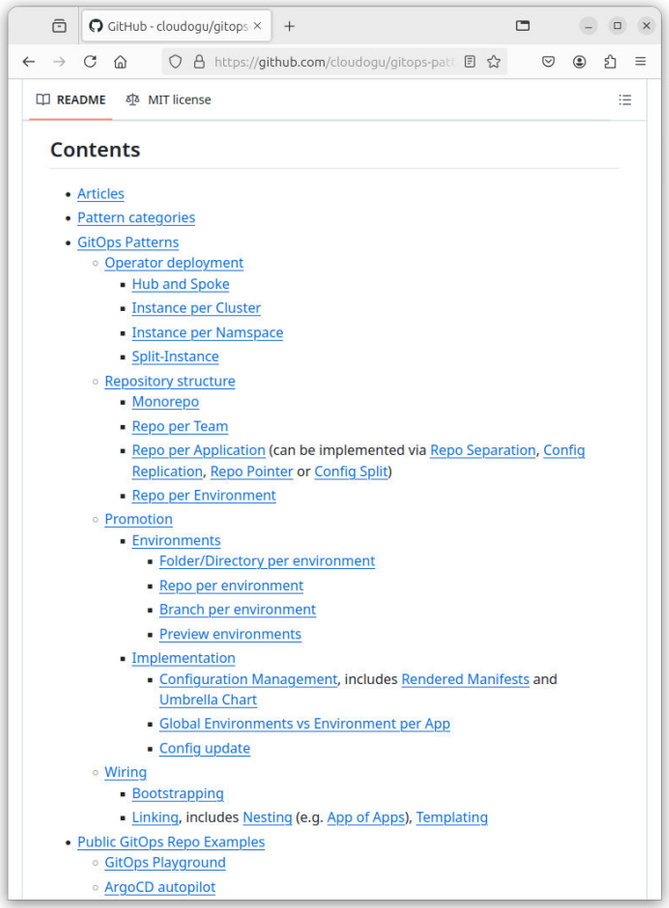

Platform engineer
Software engineer
cluster-resources
apps
argocd
applications
example-tenant.yaml
projects
example-tenant.yaml
argocd
misc
6
example-tenant/gitops
apps
my-app
production
staging
deployment.yaml
argocd
my-app-staging.yaml
my-app-production.yaml
misc.yaml
misc
network-policies.yaml
push via PR
push
8
9
7
CI
server
pull
example-tenant/my-app
k8s
production
deployment.yaml
staging
deployment.yaml
src
Jenkinsfile

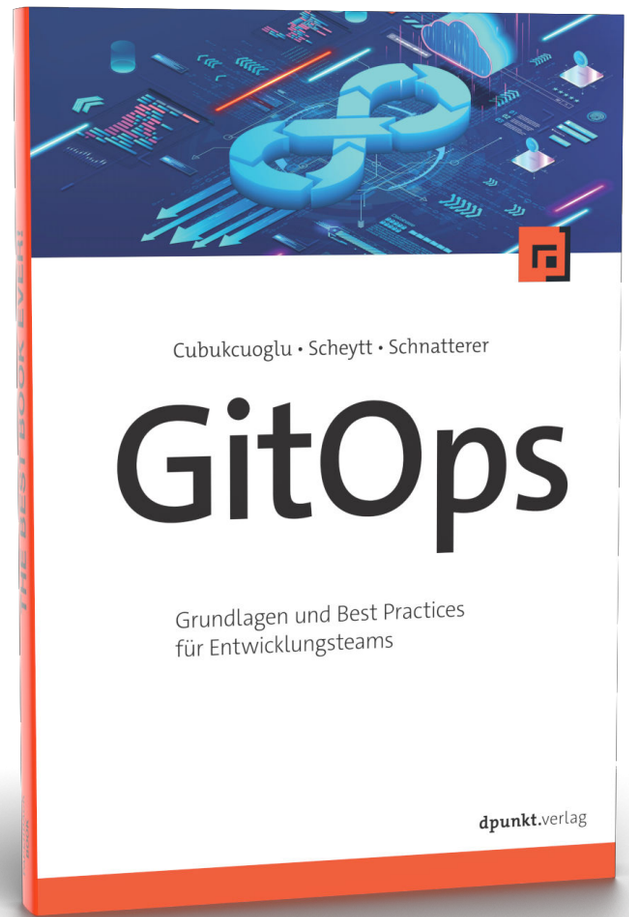
Alternatives for promotion

- **CI** 
  - \+ Can provide lots of features (PRs, hydration, validation, etc.)
  - − Depends on CI and SCM
- **Renovate** 
  - \+ Generic solution
  - − Less specific K8s or GitOps-related options
- **Image Updaters** 
  - \+ Good integration to GitOps operator
  - − No PRs (so far)
- **argoproj-labs/gitops-promoter** 
  - \+ First Party Argo CD solution (also does hydration)
  - − Not stable, not for flux
- **akuity/kargo** 
  - \+ Declarative Stage promotions, provides UI
  - − Yet another tool

More examples

Further reading

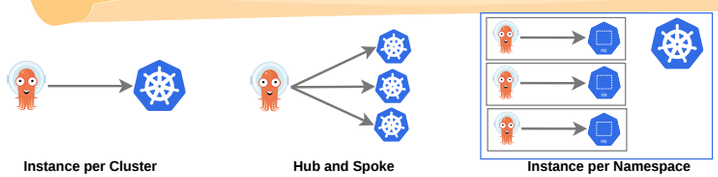


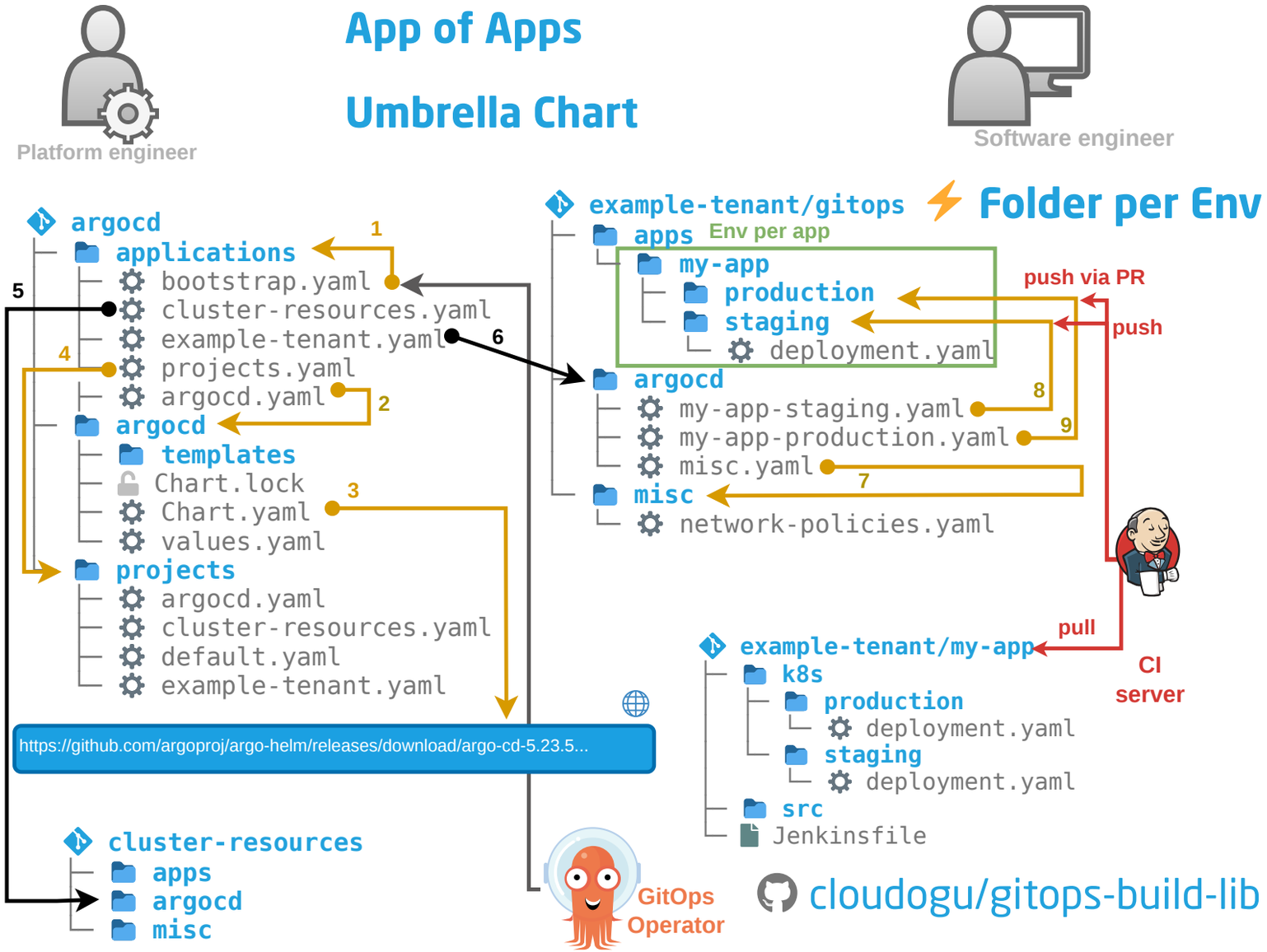


cloudogu/gitops-patterns

Example 1: GOP

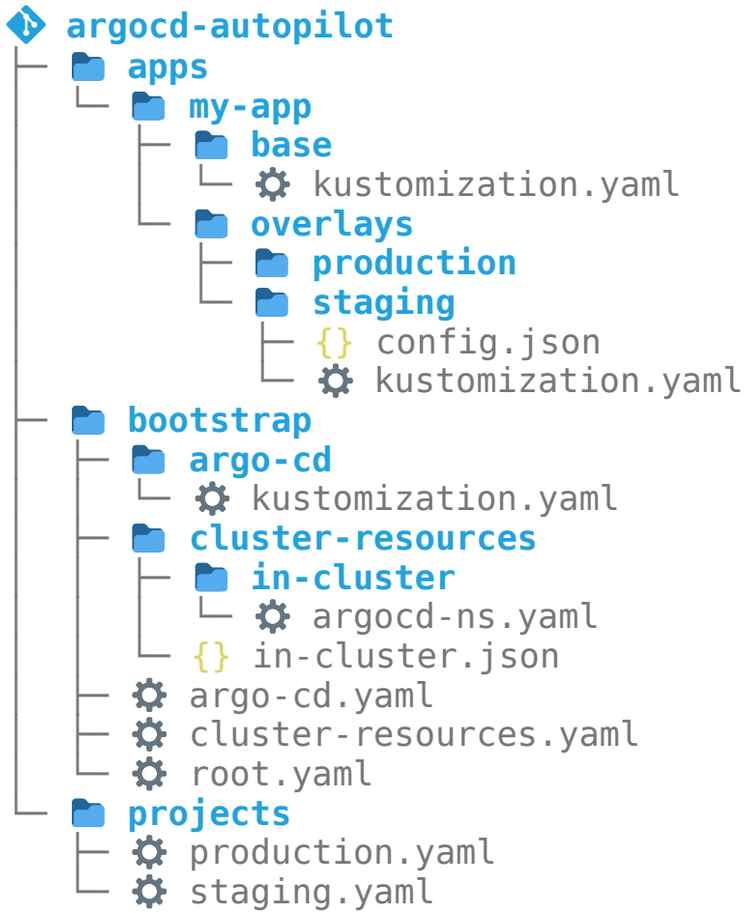
- **Repo pattern:** Per team per app
- **Operator pattern:** Instance per Cluster (Hub and Spoke, Instance per Namespace)
- **Operator:**
- **Bootstrapping:** GOP (`Helm`, `kubectl`)
- **Linking:** `Application`
- **Features:**
  - Operate ArgoCD with GitOps
  - Solution for cluster resources
  - Env per app
  - CI: Config update + replication, rendered Manifest
  - Multi-tenancy: N tenants on 1 cluster (namespace envs)
- **Source:** cloudogu/gitops-playground





Example 2: ArgoCD autopilot

- **Repo pattern:** Monorepo
- **Operator pattern:**
  Instance per Cluster / Hub and Spoke
- **Operator:**
- **Bootstrapping:** `argocd-autopilot`
- **Linking:** `Application`, `ApplicationSet`,
- **Features:**
  - Operate ArgoCD with GitOps
  - Solution for cluster resources
  - Env per app Pattern
  - Create structure and YAML via CLI
- **Source:**
  argoproj-labs/argocd-autopilot
  gitops-book/argocd-autopilot-example



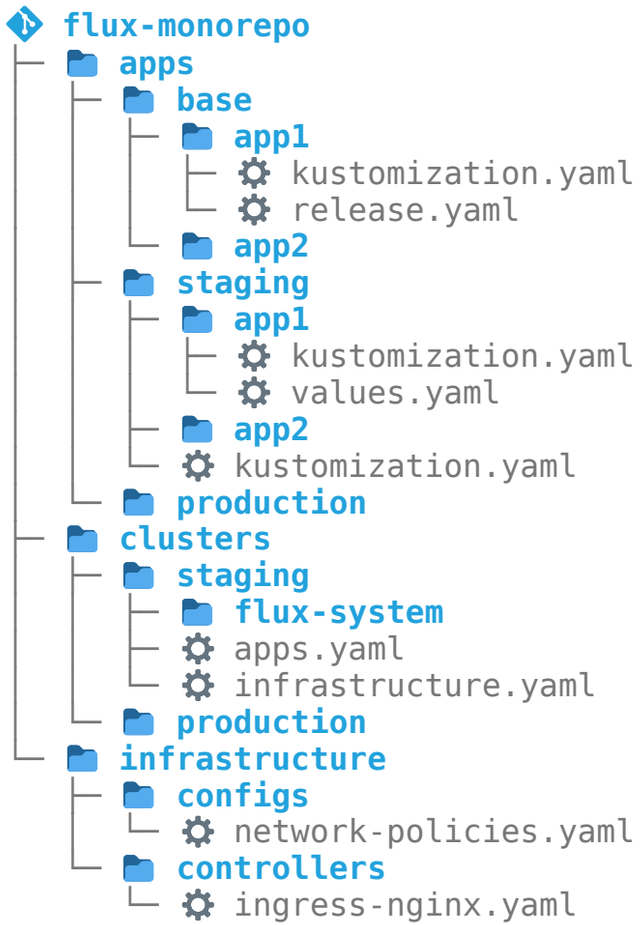
Example 3: Flux Monorepo

- **Repo pattern:** Monorepo
- **Operator pattern:** Instance per Cluster
- **Operator:**
- **Bootstrapping:** `flux`
- **Linking:** `Kustomization`,
- **Features:**
  - Operate Flux with GitOps
  - Solution for cluster resources
- **Source:**

fluxcd/flux2-kustomize-helm-example#16

fluxcd.io/flux/guides/repository-structure

gitops-book/flux2-kustomize-helm-example



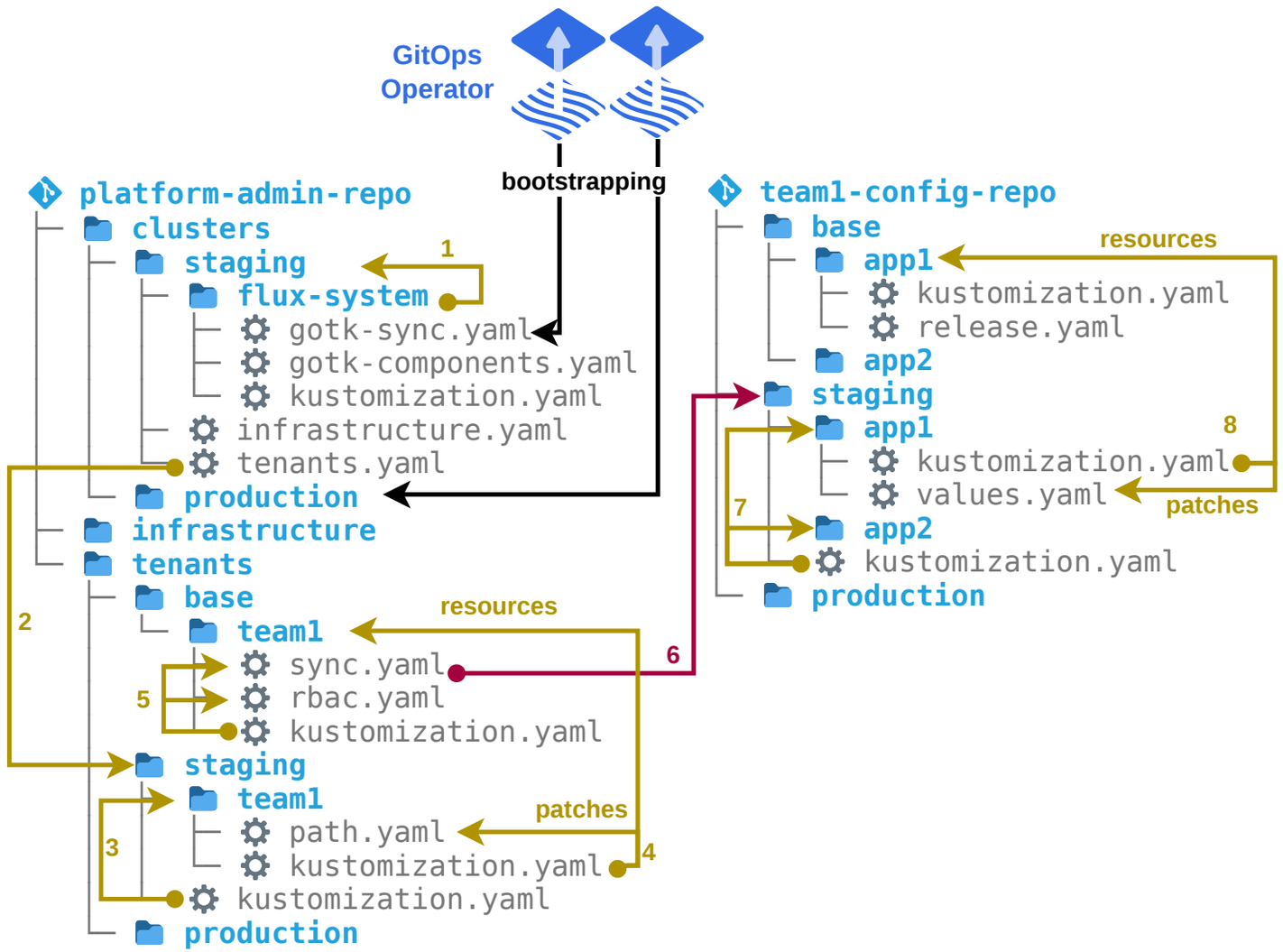
Example 4: Flux repo per team

- **Repo pattern:** Repo per team
- **Operator pattern:** Instance per Cluster
- **Operator:**
- **Bootstrapping:** `flux`
- **Linking:** `Kustomization`,
- **Features:**
  - Operate Flux with GitOps
  - Solution for cluster resources
  - Multi-tenancy: N tenants on 1 cluster per env (cluster envs)
- **Source:**

fluxcd/flux2-multi-tenancy

fluxcd.io/flux/guides/repository-structure

gitops-book/flux2-multi-tenancy



Example 4b: Flux D1 Reference Architecture

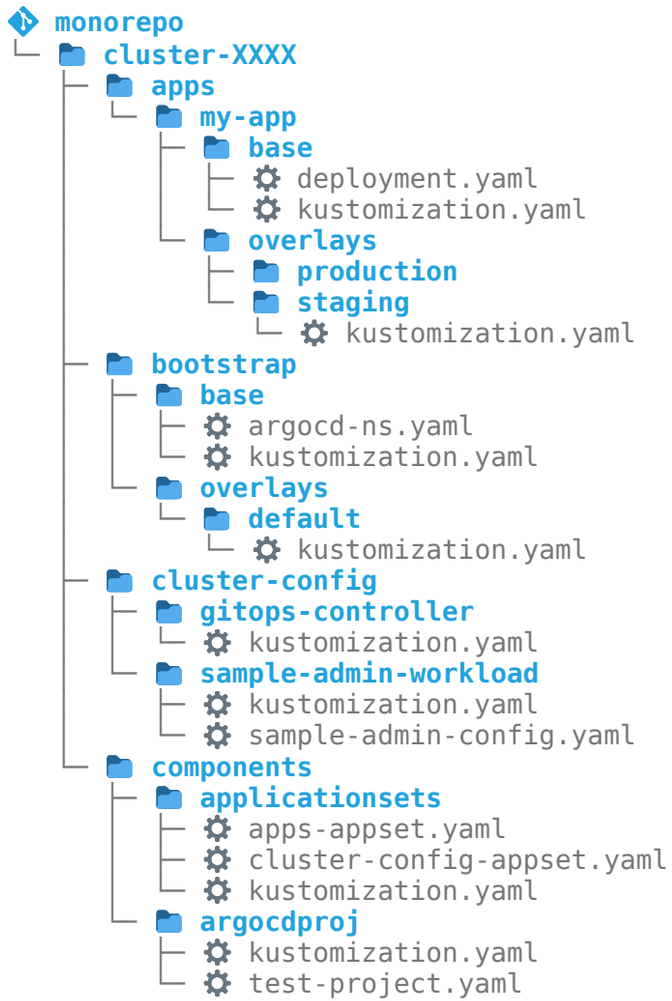
controlplaneio-fluxcd/d1-infra

Example 5: ArgoCD and Flux alternative

- **Repo pattern:** Monorepo
- **Operator pattern:** Instance per Cluster
- **Operator:**
- **Bootstrapping:** `kubectl`
- **Linking:** `Application`, `ApplicationSet` / `Kustomization`,
- **Features:**
  - Solution for cluster resources
  - ArgoCD **and** Flux examples
- **Source:**

  christianh814/example-kubernetes-go-repo

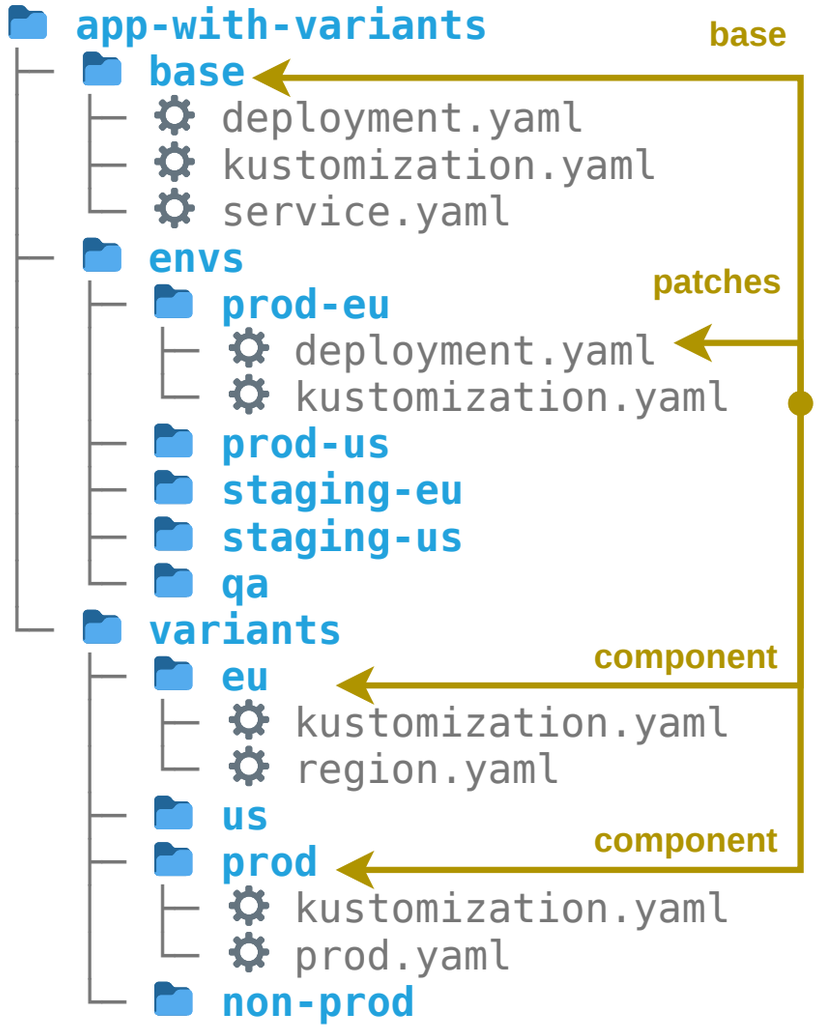
  C. Hernandez - The Path to GitOps



Example 6: Env variations single app

- **Operator:** () ()
- **Linking:** 
- **Features:**
  - Env variants for a single app
  - Promotion "via `cp`"
  - Env per app Pattern
- **Source:**

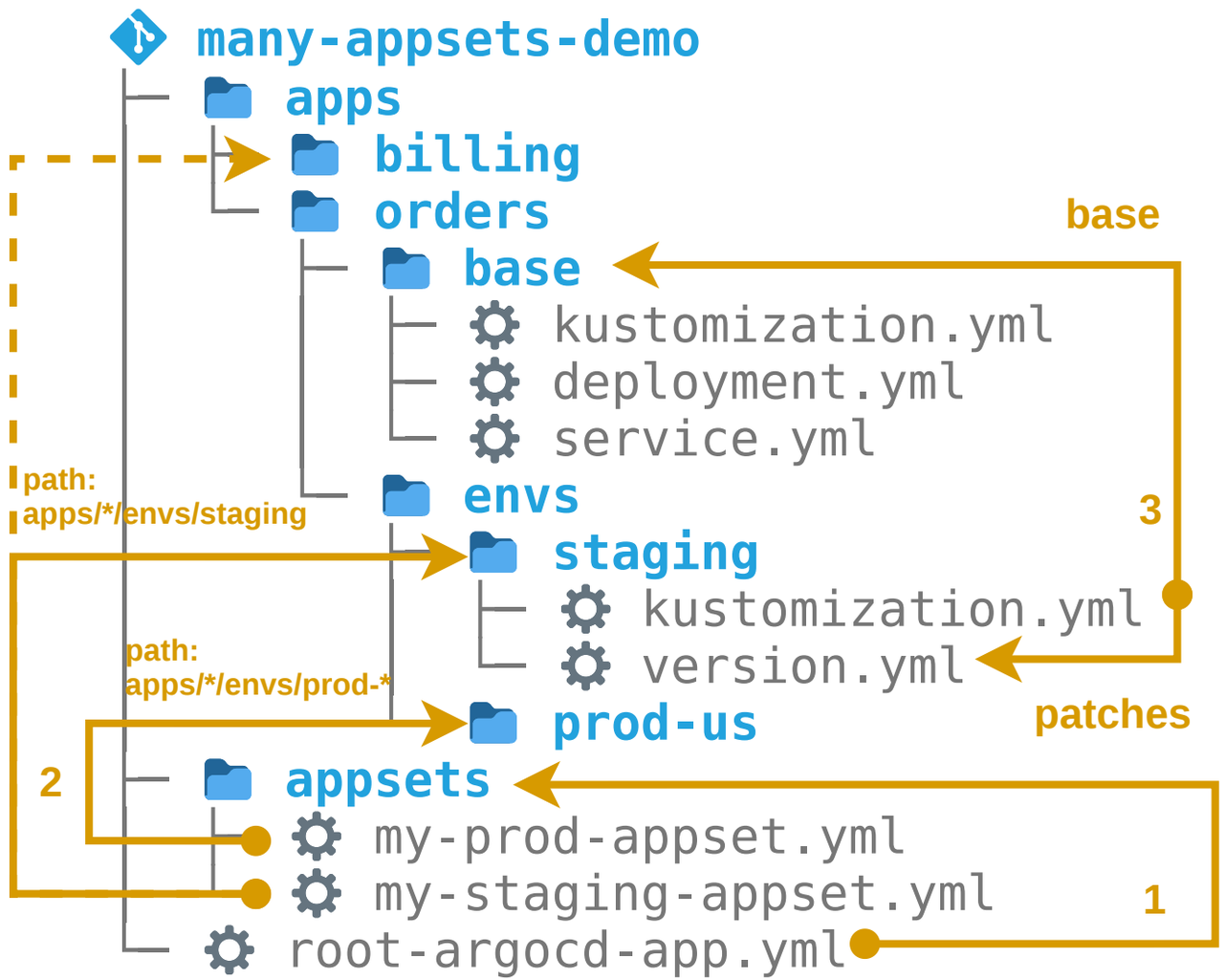
  kostis-codefresh/gitops-environment-promotion



Example 7: Env variations multiple apps

- **Operator:**
- **Linking:** K , `ApplicationSet`, (`Application`)
- **Features:**
  - Env variants for multiple apps with few appSets
  - Promotion "via `cp`"
  - Env per app Pattern
- **Source:**

kostis-codefresh/many-appsets-demo



Key Takeaways

- Bootstrapping: The only imperative step before GitOps
- Linking: K
  - `Kustomization`
  - `Application`, `ApplicationSet`
- Multi-tenancy: Repos, operators, clusters, namespaces

➡️ Use examples as inspiration to solving your use case

Johannes Schnatterer, Cloudogu GmbH

@schnatterer@floss.social

in/jschnatterer

Slides

cloudogu/gitops-playground

Image sources

- Live Example: https://unsplash.com/photos/assorted-color-hot-air-balloons-during-daytime-DuBNA1QMpPA
- More examples: https://unsplash.com/photos/XZc4f2XZc84
- coloured-parchment-paper background by brgfx on Freepik: https://www.freepik.com/free-vector/coloured-parchment-paper-designs_1078492.htm