# // GITOPS BY EXAMPLE - REPOS, PATTERNS, PROMOTION, PROCESSES

Johannes Schnatterer, Cloudogu GmbH

**Slides**



@schnatterer@floss.social       in/jschnatterer
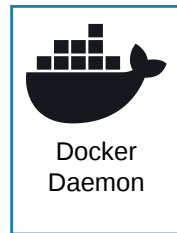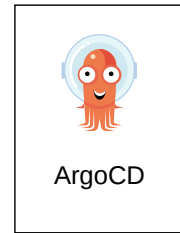
Version: 202410231818-851ab67
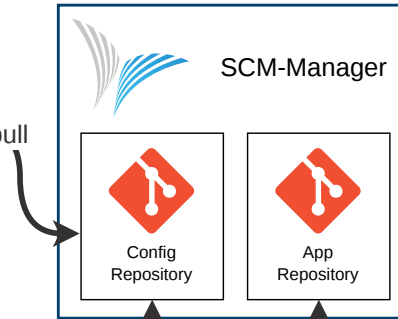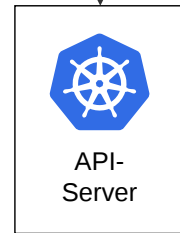
**Live example**

Developer's host

K3D Container

Docker Daemon —run→

ArgoCD —pull→ SCM-Manager

apply resources

API-Server

Config Repository

App Repository

push config and create pull request

pull

Jenkins —image push→ OCI Registry

3
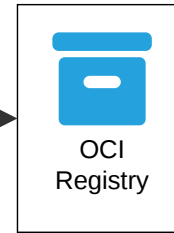
# Meet GOP

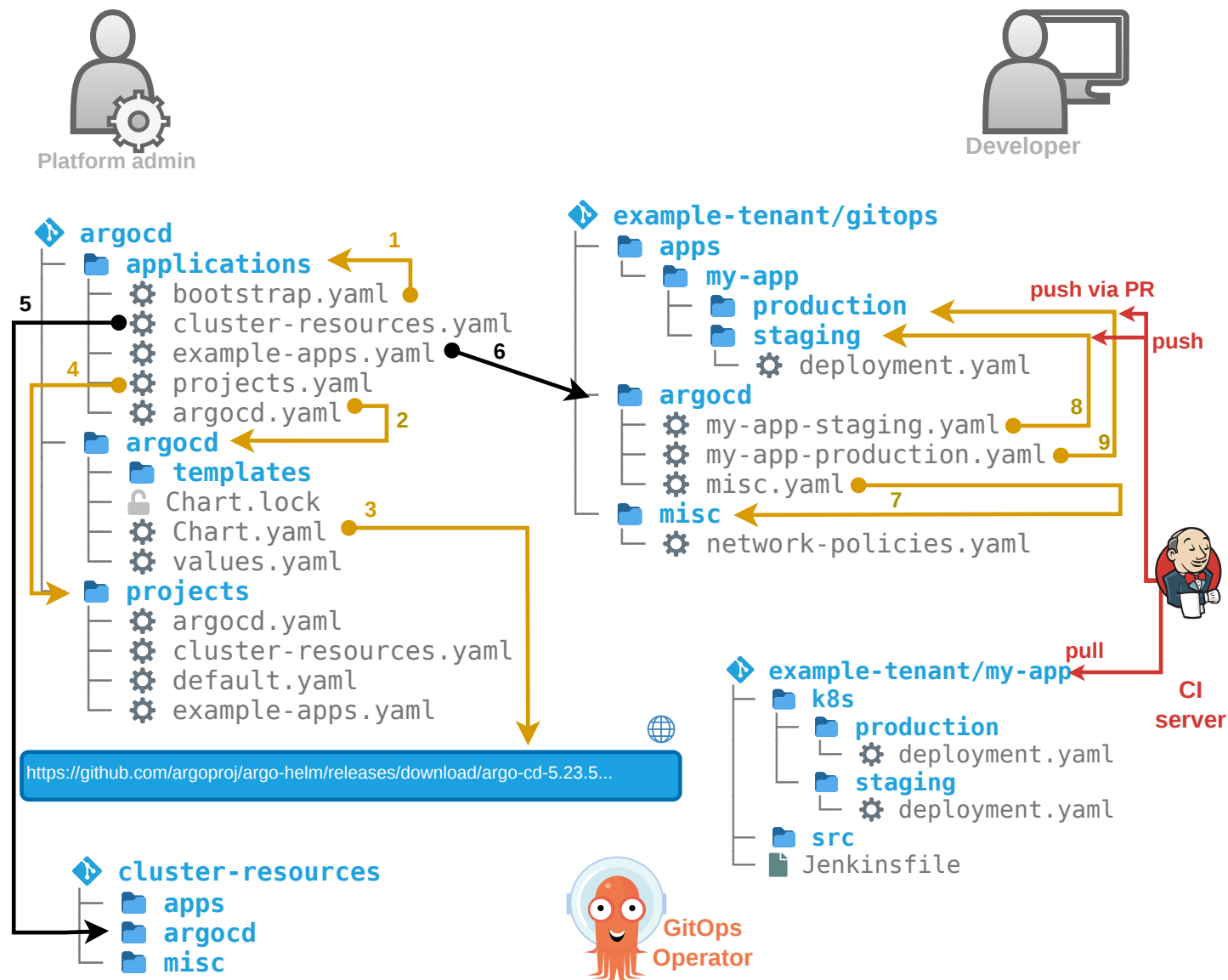a GitOps-based operational stack (platform)

```
VERSION='0.7.0'
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 --base-url=http://localhost --ingress-nginx --argocd
    # Other interesting options  --monitoring --vault=dev --mailhog
```

cloudogu/gitops-playground

# Example 1: GOP

- **Repo pattern:** Per team 🏭 per app
- **Operator pattern:** Instance per Cluster (Hub and Spoke)
- **Operator:** 🐙
- **Boostrapping:** `Helm`, `kubectl`
- **Linking:** 🐙 `Application`
- **Features:**
  - Operate ArgoCD with GitOps
  - Solution for cluster resources
  - Config update + replication via CI
  - Env per app pattern
  - Mixed repo patterns
- **Source:** 🐙 cloudogu/gitops-playground

**Platform admin**

**Developer**

**argocd**
- **applications**
  - bootstrap.yaml　1
  - cluster-resources.yaml　5
  - example-apps.yaml　6
  - projects.yaml　4
  - argocd.yaml　2
- **argocd**
  - **templates**
    - 🔓 Chart.lock
    - Chart.yaml　3
    - values.yaml
- **projects**
  - argocd.yaml
  - cluster-resources.yaml
  - default.yaml
  - example-apps.yaml

🌐 https://github.com/argoproj/argo-helm/releases/download/argo-cd-5.23.5...

**cluster-resources**
- **apps**
- **argocd**
- **misc**

**GitOps Operator**

**example-tenant/gitops**
- **apps**
  - **my-app**
    - **production**
    - **staging**
      - deployment.yaml
- **argocd**
  - my-app-staging.yaml　8
  - my-app-production.yaml　9
  - misc.yaml　7
- **misc**
  - network-policies.yaml

push via PR
push

**example-tenant/my-app**
- **k8s**
  - **production**
    - deployment.yaml
  - **staging**
    - deployment.yaml
- **src**
  - Jenkinsfile

pull

**CI server**

5

# More examples

# Example 3: ArgoCD autopilot

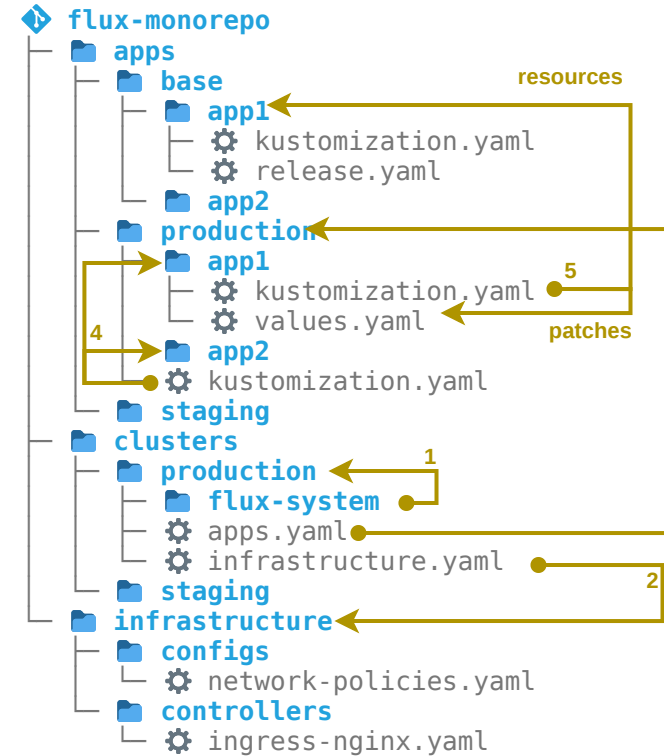- **Repo pattern:** Monorepo
- **Operator pattern:**
  Standalone / Hub and Spoke
- **Operator:** 🐙
- **Boostrapping:** `argocd-autopilot`
- **Linking:** 🐙 `Application`, `ApplicationSet`, K
- **Features:**
  - Operate ArgoCD with GitOps
  - Opinionated structure and YAML creation via CLI
- **Source:** ⧉ argoproj-labs/argocd-autopilot
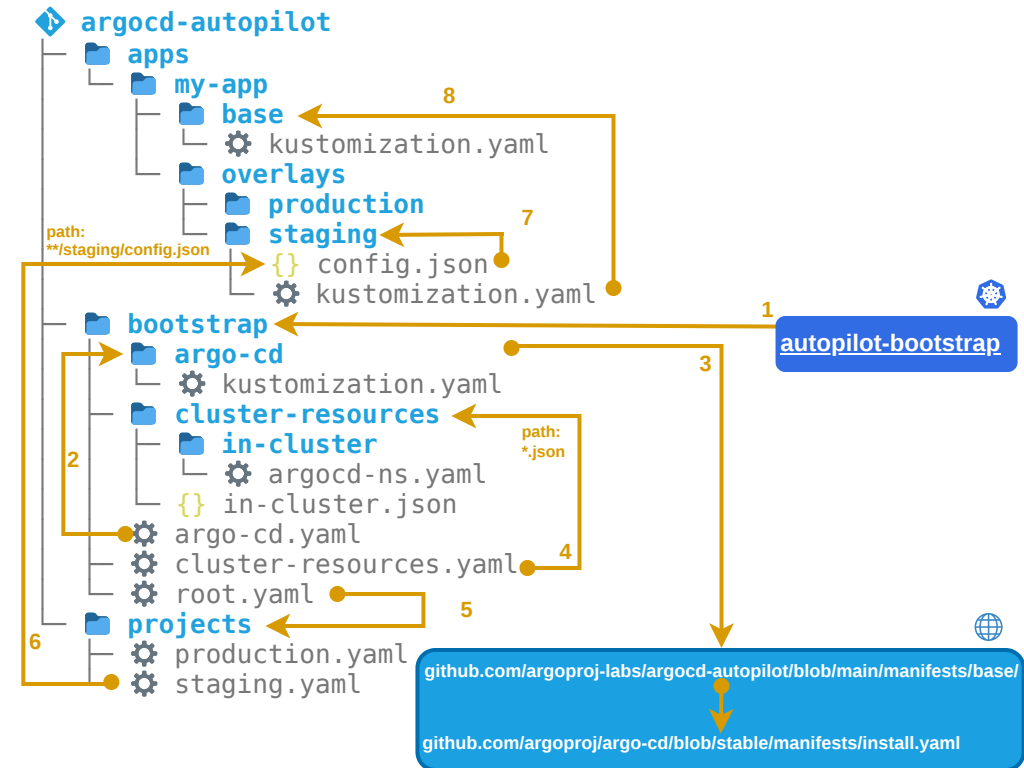
# Example 4: Flux Monorepo

- **Repo pattern:** Monorepo
- **Operator pattern:** Standalone
- **Operator:** 🪁 (🐙?)
- **Boostrapping:** `flux`
- **Linking:** 🪁 `Kustomization`, 🅺
- **Features:**
  - Cross-cutting infra
  - Operate Flux with GitOps
- **Source:**

  🐙 fluxcd/flux2-kustomize-helm-example#16

  🌐 fluxcd.io/flux/guides/repository-structure

```
flux-monorepo
├── apps
│   ├── base
│   │   ├── app1                          resources
│   │   │   ├── kustomization.yaml
│   │   │   └── release.yaml
│   │   └── app2
│   ├── production
│   │   ├── app1                              5
│   │   │   ├── kustomization.yaml
│   │   │   └── values.yaml
│   │   │                                 patches
│   │   ├── app2
│   │   └── kustomization.yaml
│   └── staging
├── clusters
│   ├── production                    1
│   │   ├── flux-system
│   │   ├── apps.yaml
│   │   └── infrastructure.yaml                3
│   └── staging                                  2
├── infrastructure
│   ├── configs
│   │   └── network-policies.yaml
│   └── controllers
│       └── ingress-nginx.yaml
```
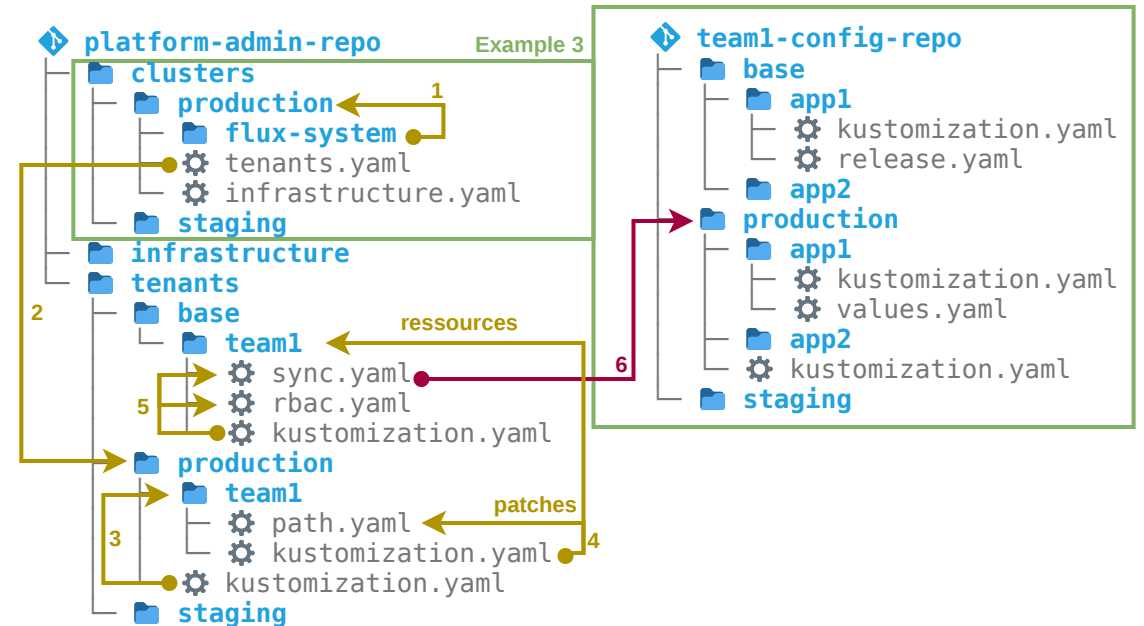
# Example 5: Flux repo per team

- **Repo pattern:** Repo per team
- **Operator pattern:** Standalone
- **Operator:** 🪁 (🐙?)
- **Boostrapping:** `flux`
- **Linking:** 🪁 `Kustomization`, Ⓚ
- **Features:** Ex 5 with repo for team
- **Source:**

  🐙 fluxcd/flux2-multi-tenancy

  🌐 fluxcd.io/flux/guides/repository-structure

# Example 6: ArgoCD and Flux alternative

- **Repo pattern:** Monorepo
- **Operator pattern:** Standalone
- **Operator:** 🐙 🔷
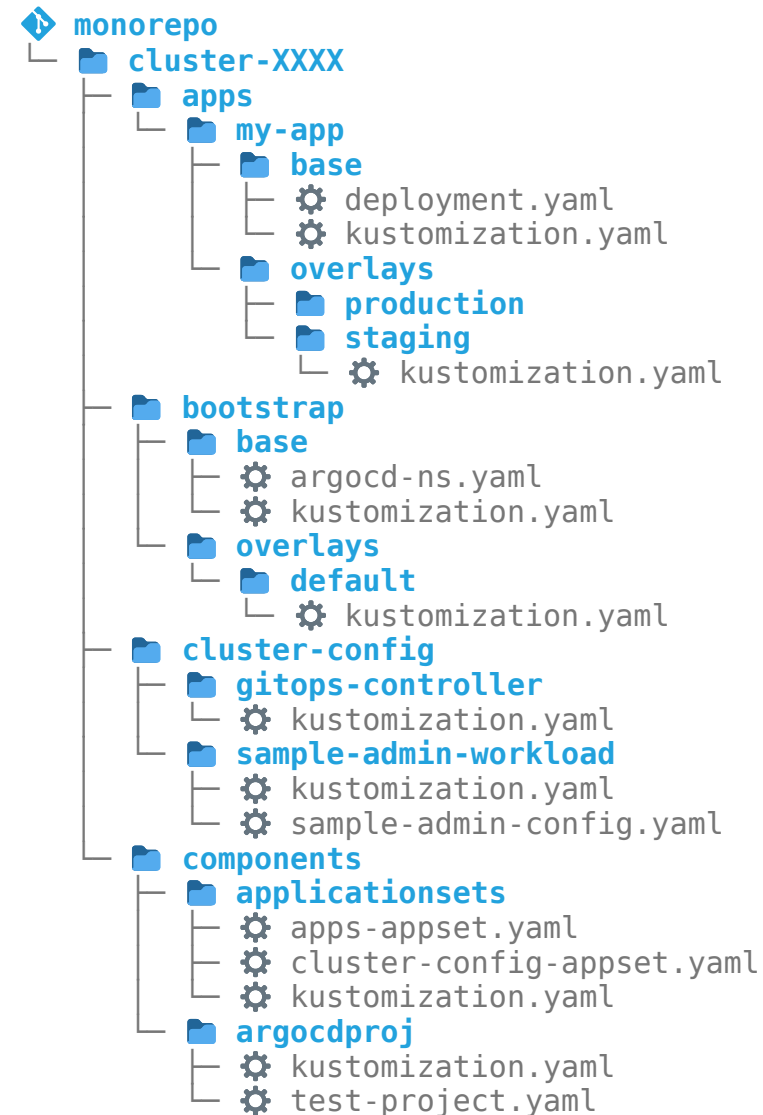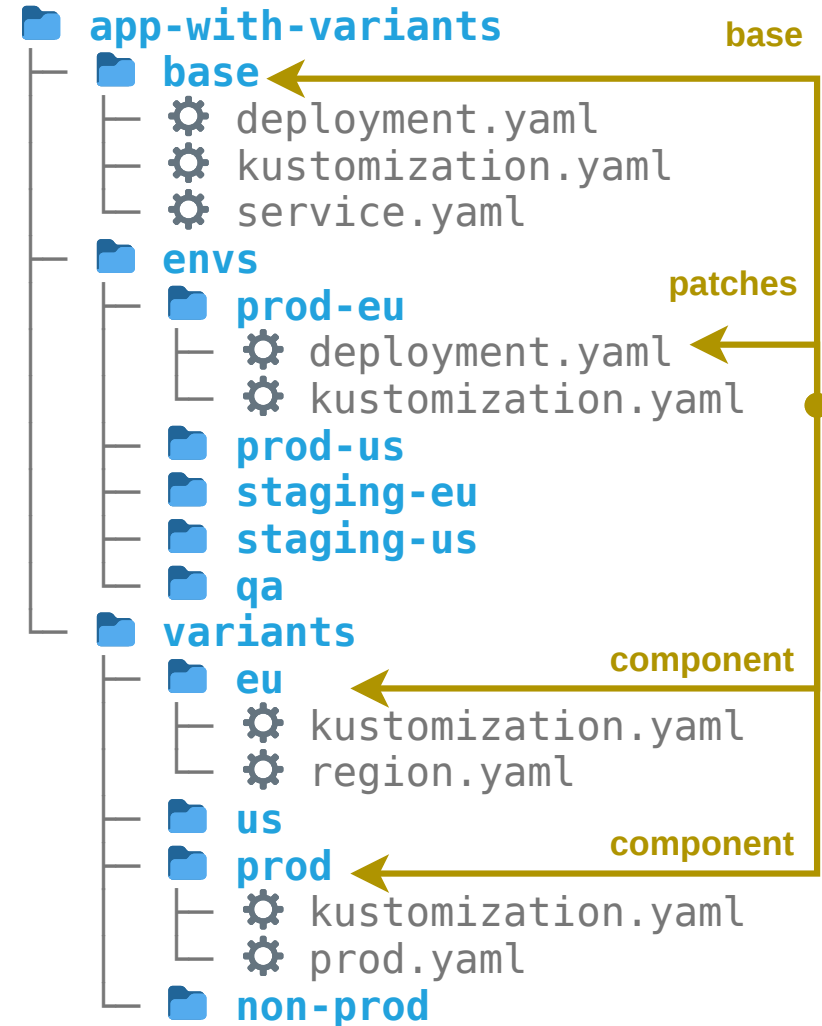- **Boostrapping:** `kubectl`
- **Linking:** 🐙 `Application`, `ApplicationSet` / 🔷 `Kustomization`, 🇰
- **Features:**
  - Cross-cutting infra and app(s)
  - ArgoCD **and** Flux examples
- **Source:**

  🔗 christianh814/example-kubernetes-go-repo

  📘 C. Hernandez - The Path to GitOps

```
 monorepo
 └── cluster-XXXX
     ├── apps
     │   └── my-app
     │       ├── base
     │       │   ├── deployment.yaml
     │       │   └── kustomization.yaml
     │       └── overlays
     │           ├── production
     │           └── staging
     │               └── kustomization.yaml
     ├── bootstrap
     │   ├── base
     │   │   ├── argocd-ns.yaml
     │   │   └── kustomization.yaml
     │   └── overlays
     │       └── default
     │           └── kustomization.yaml
     ├── cluster-config
     │   ├── gitops-controller
     │   │   └── kustomization.yaml
     │   └── sample-admin-workload
     │       ├── kustomization.yaml
     │       └── sample-admin-config.yaml
     └── components
         ├── applicationsets
         │   ├── apps-appset.yaml
         │   ├── cluster-config-appset.yaml
         │   └── kustomization.yaml
         └── argocdproj
             ├── kustomization.yaml
             └── test-project.yaml
```
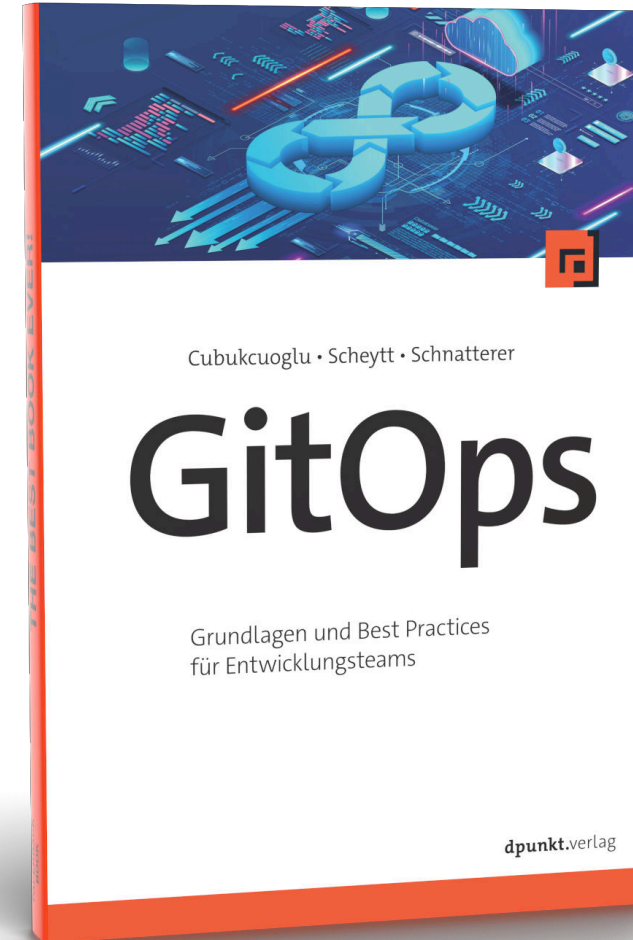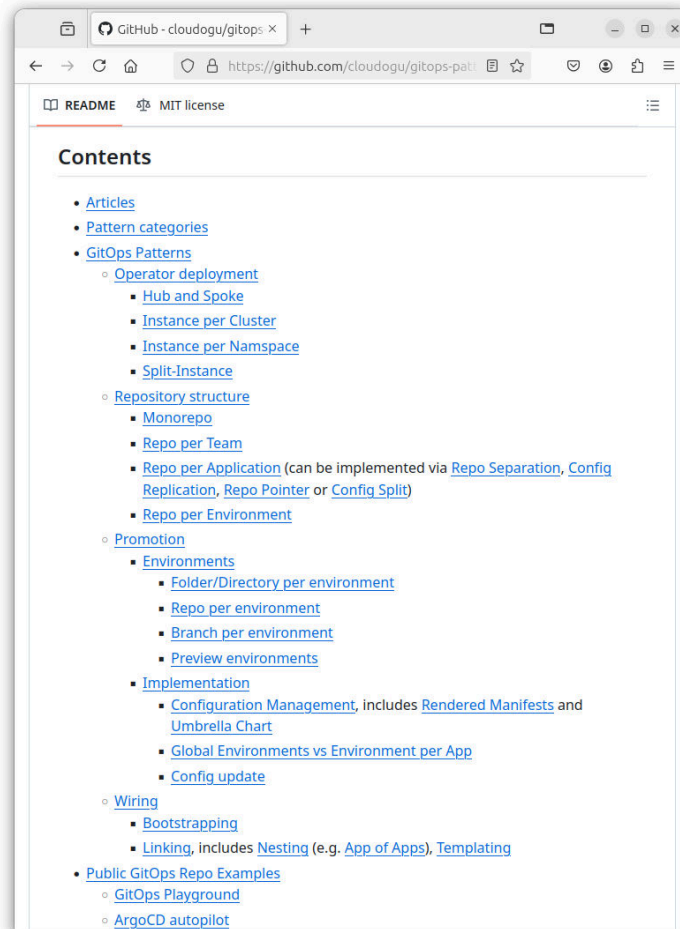
# Example 7: Environment variations

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

  🔗 kostis-codefresh/gitops-environment-promotion

```
📁 app-with-variants                    base
├── 📁 base
│   ├── ⚙ deployment.yaml
│   ├── ⚙ kustomization.yaml
│   └── ⚙ service.yaml
├── 📁 envs
│   ├── 📁 prod-eu                       patches
│   │   ├── ⚙ deployment.yaml
│   │   └── ⚙ kustomization.yaml
│   ├── 📁 prod-us
│   ├── 📁 staging-eu
│   ├── 📁 staging-us
│   └── 📁 qa
└── 📁 variants
    ├── 📁 eu                            component
    │   ├── ⚙ kustomization.yaml
    │   └── ⚙ region.yaml
    ├── 📁 us
    ├── 📁 prod                          component
    │   ├── ⚙ kustomization.yaml
    │   └── ⚙ prod.yaml
    └── 📁 non-prod
```

# More patterns + further reading
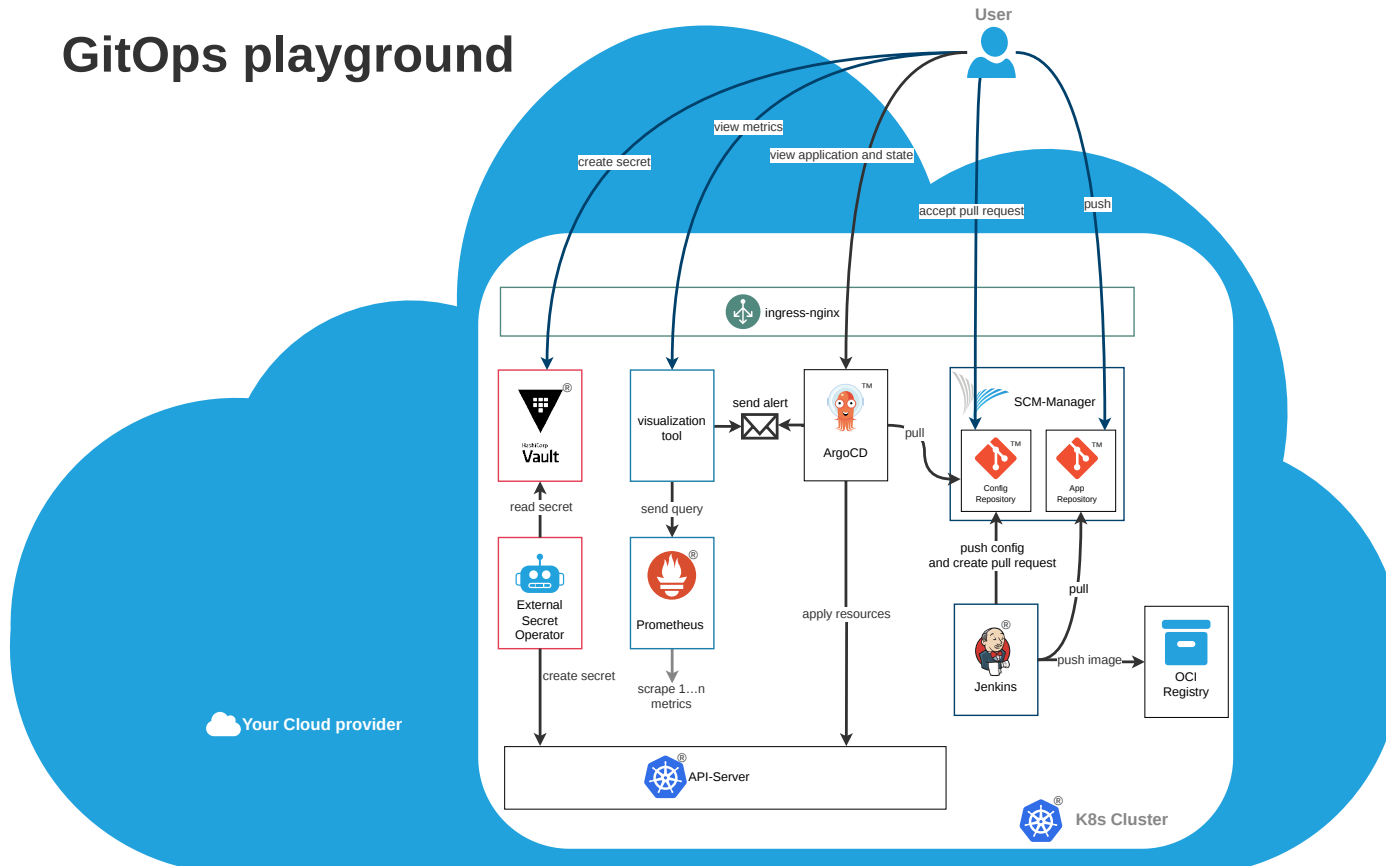


## Contents

cloudogu/gitops-patterns

# Johannes Schnatterer, Cloudogu GmbH

@schnatterer@floss.social    in/jschnatterer

cloudogu/gitops-playground

**GitOps playground**



Slides



©Cloudogu GmbH 2024: GitOps Playground© for use with Argo™, Git™, Jenkins®, Kubernetes®, Prometheus®, Vault® and SCM-Manager

# 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