// GITOPS - CONTINUOUS OPERATIONS WITH KUBERNETES

Johannes Schnatterer, Cloudogu GmbH

@jschnatterer

Version: 202107141319-8052d90

Agenda

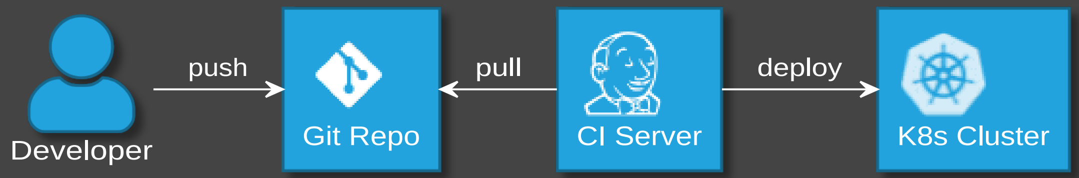
- What is GitOps?
- Where can it be used?
- How can it be used?
- What challenges arise?

What is GitOps?

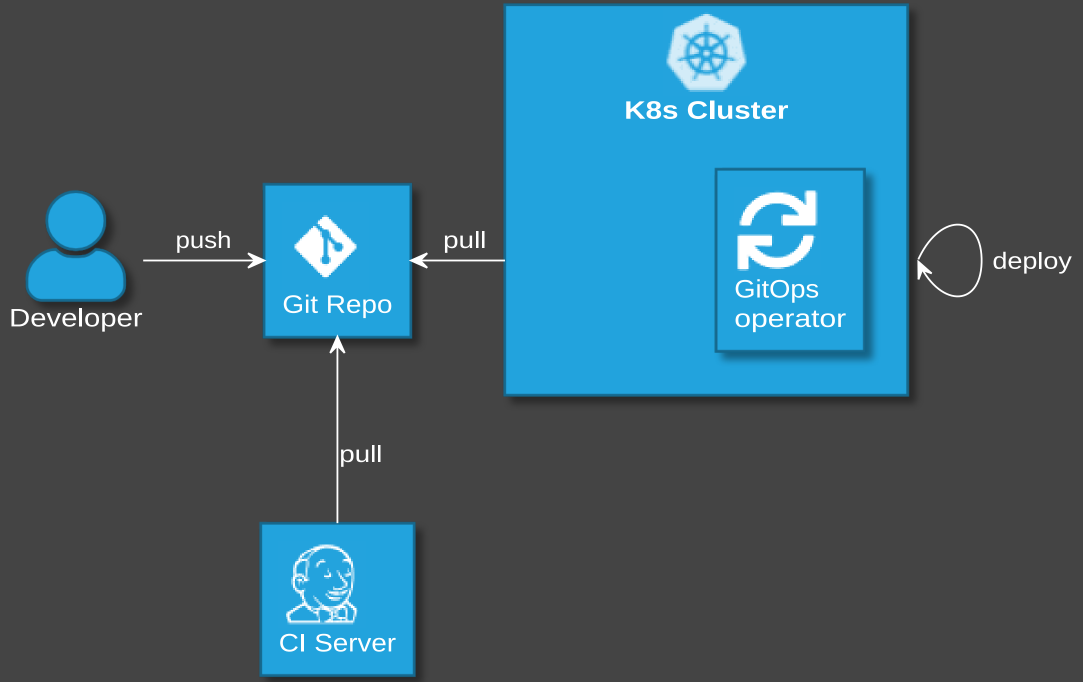
- Operating model
- Origin: blog post by Weaveworks, August 2017

> Use developer tooling to drive operations
>
> 🌐 weave.works/blog/gitops-operations-by-pull-request

"Classic" Continuous Delivery ("CIOps")



GitOps



GitOps Principles

1. The principle of declarative desired state
2. The principle of immutable desired state versions
3. The principle of continuous state reconciliation
4. The principle of operations through declaration

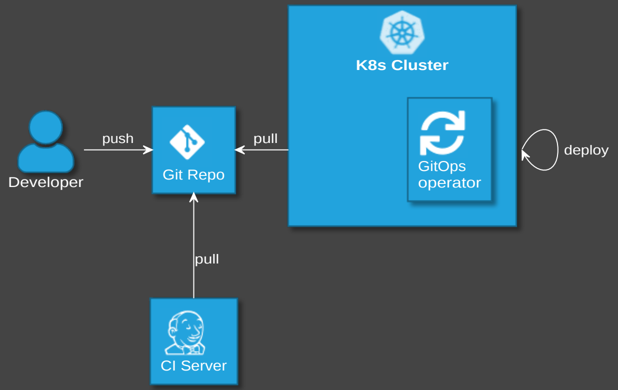
github.com/open-gitops/documents/blob/main/PRINCIPLES.md

GitOps vs DevOps

- DevOps is about collaboration of formerly separate groups (mindset)
- GitOps focuses on ops (operations model)
- GitOps can be used with or without DevOps

Advantages of GitOps

- (Almost) no access to cluster from outside
- No credentials on CI server
- Forces 100% declarative description
  - auditable
  - automatic sync of cluster and git
- Enterprise: Accessing git is simpler (no new firewall rules)

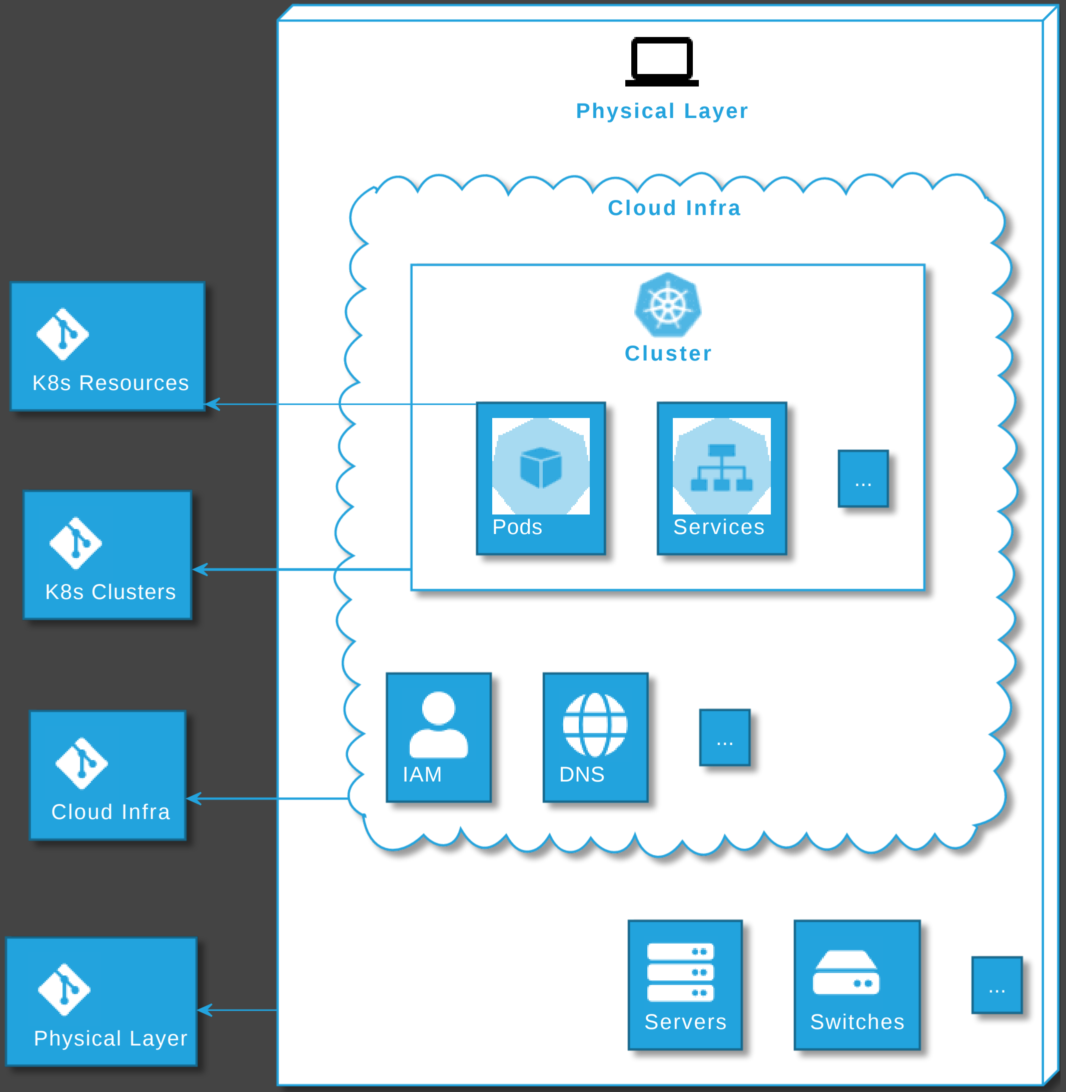


What can GitOps be used for?

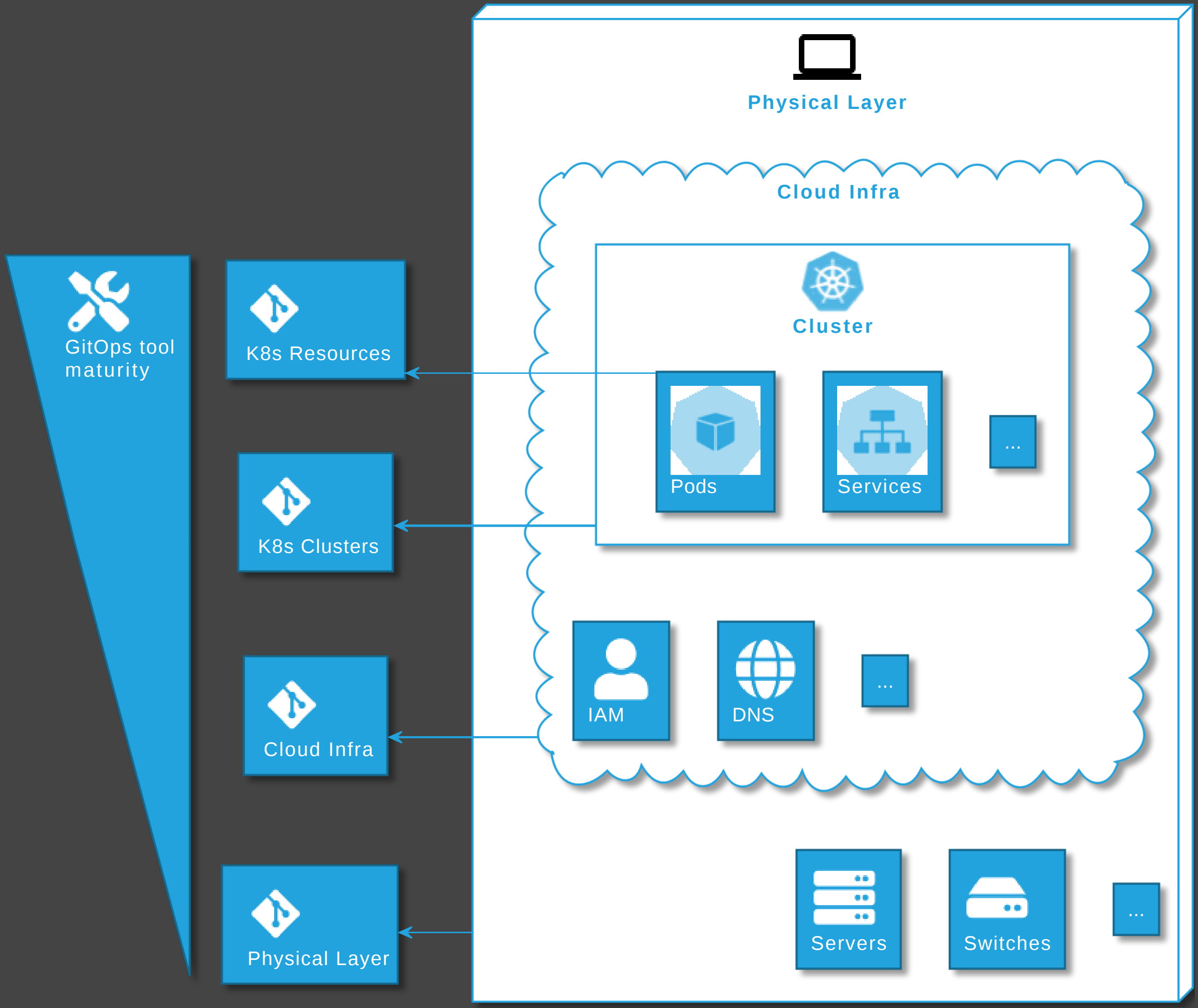
📜 GitOps History in a nutshell

- grew up operating applications on Kubernetes,
- is now rising above it, operating clusters
  and other (cloud) infrastructure

A GitOps Vision



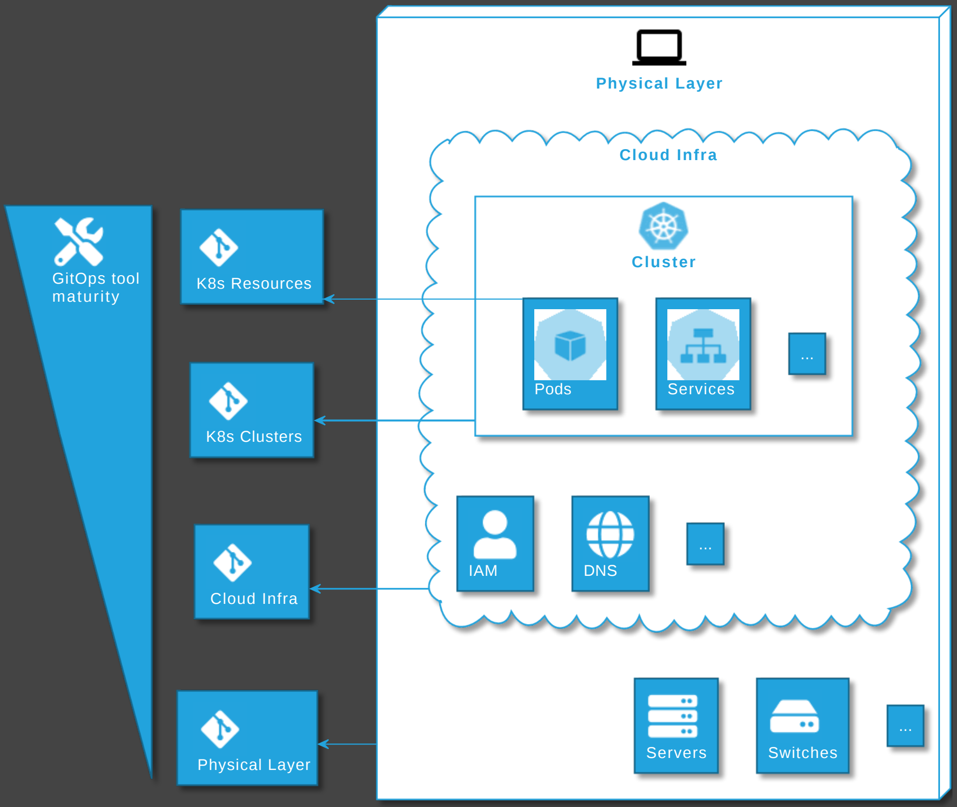
GitOps reality



How can GitOps be used?

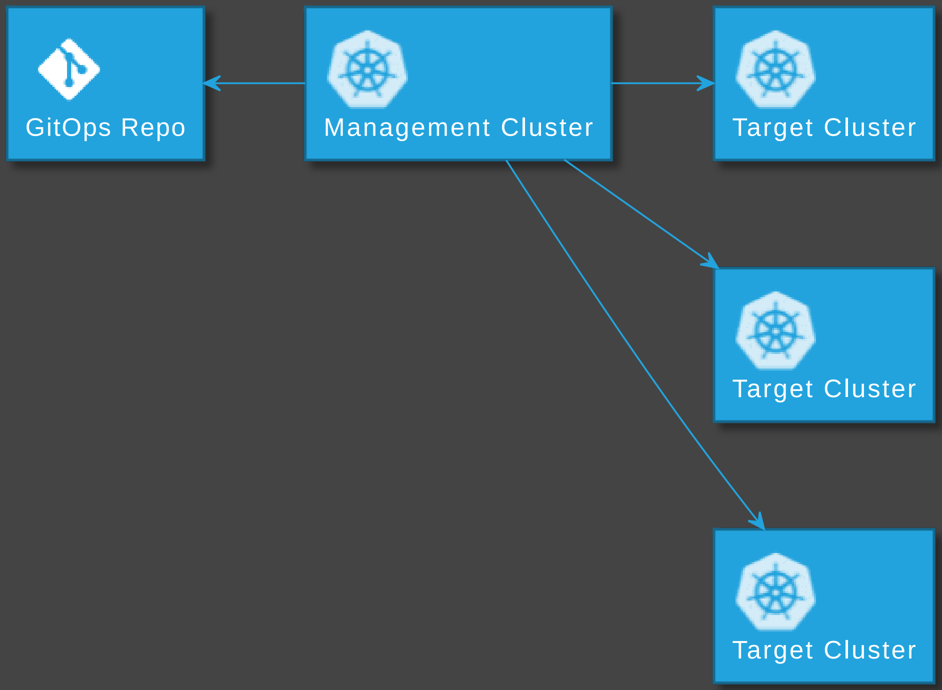
Categories

- Tools for Kubernetes AppOps
- Tools for Kubernetes ClusterOps
- Supplementary GitOps tools



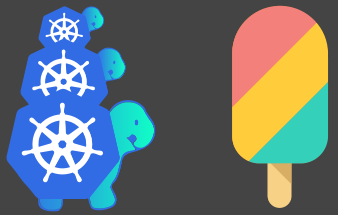
GitOps Tools for Kubernetes AppOps

Operate Kubernetes with Kubernetes

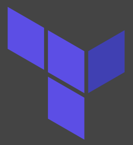


GitOps Tools for Kubernetes ClusterOps

+



|



-

Cloud or Operator

-
-
- rancher/terraform-controller

Supplementary GitOps tools

Secrets

- bitnami-labs/sealed-secrets
- Soluto/kamus
- mozilla/sops + K8s integration
- Operators for Key Management Systems

Others

- Backups
- Deployment Strategies - Progressive Delivery
- ...

See also

☁️ cloudogu.com/blog/gitops-tools (iX 4/2021)

- General tool comparison,
- tips on criteria for tool selection,
- comparison of ArgoCD v1 and Flux v2

What challenges arise with GitOps?

More Infra ...

- GitOps Operator: One or more custom controllers
- Helm, Kustomize Controllers
- Operators for Supplementary tools (secrets, etc.)
- Monitoring/Alerting systems
- ...

… higher cost

- Maintenance/patching (vendor dependency)
- Resource consumption
- Error handling
  - failing late and silently
  - monitoring/alerting required
  - reason might be difficult to pinpoint
  - operators cause alerts (OOM errors, on Git/API server down, etc.)

Day two questions

- POC is simple
- Operations in prod has its challenges
  - How to realize staging?
  - Role of CI server?
  - How to structure repos?
  - How to delete resources?
  - How to realize local dev env?
  - ...

Implementing stages

Idea 1: Staging Branches

- Develop ➡️ Staging
- Main ➡️ Production

❌ Logic for branching complicated and error prone (merges)

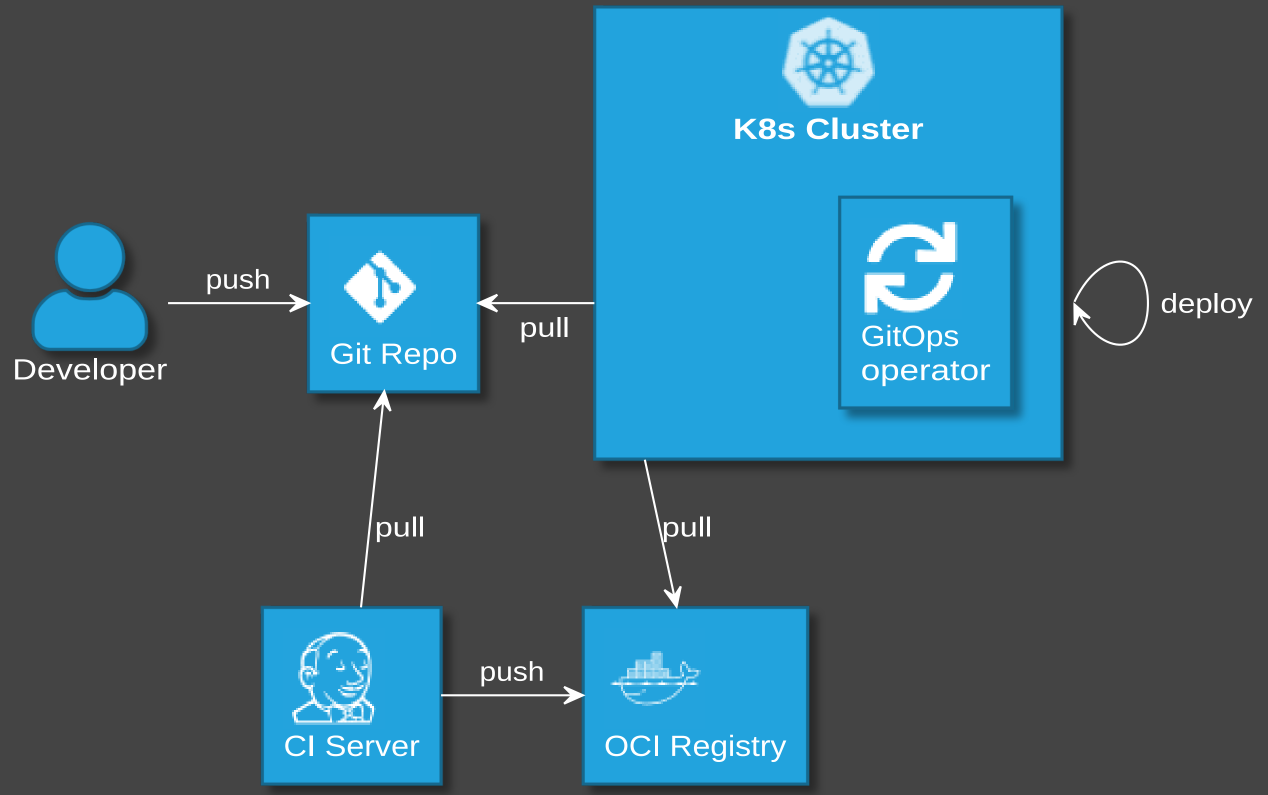
Idea 2: Staging folders

- On the same branch: One folder per stage
- Process:
  - Commit to staging folder only,
  - create short lived branches and pull requests for prod
- Risky, but can be automized

✅

- Logic for branching simpler
- Supports arbitrary number of stages

Role of CI server

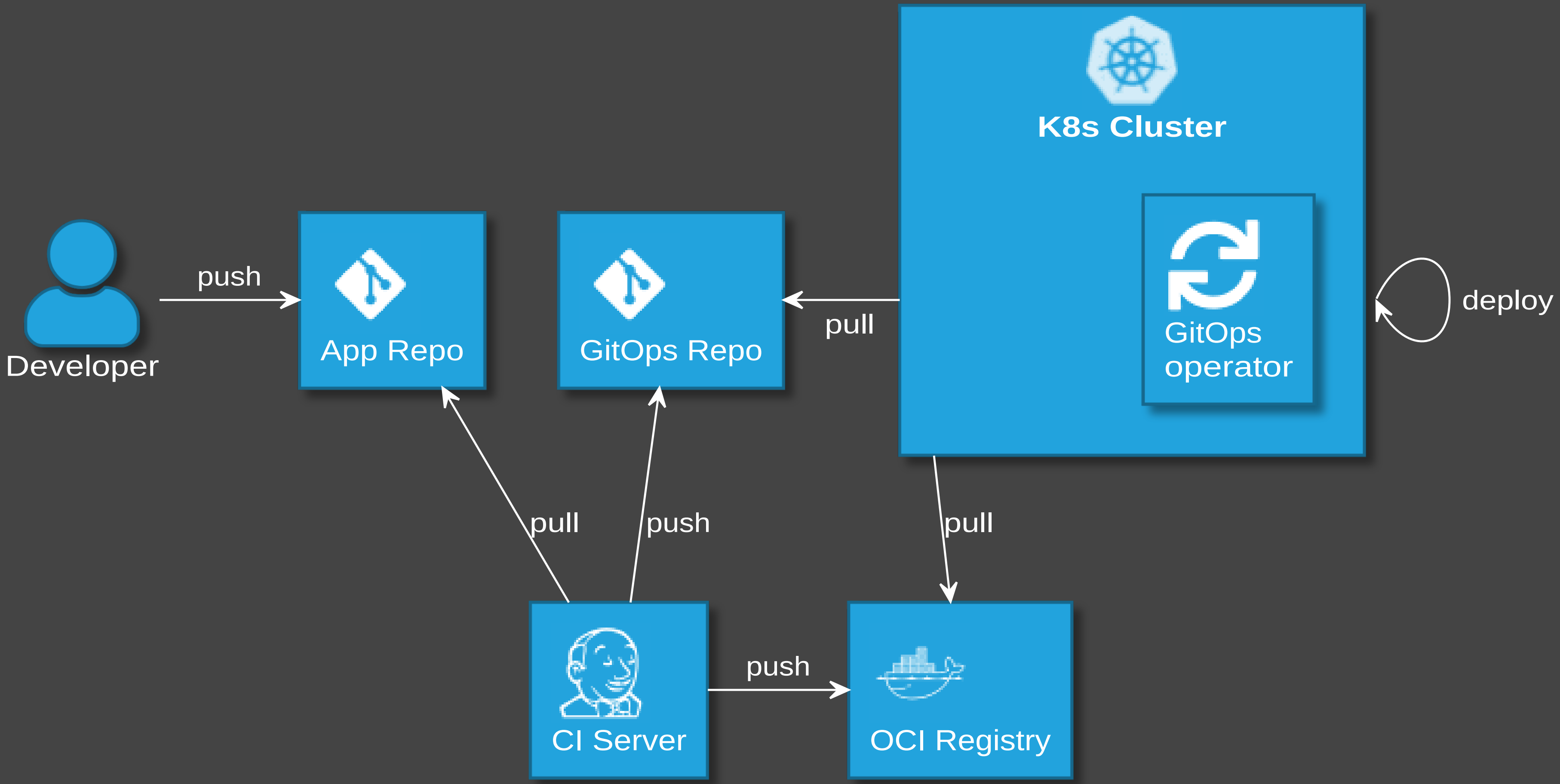


Number of repositories: application vs GitOps repo

- Good pratice: Keeping everything in app repo (code, docs, infra)
- GitOps: Put infra in separate repo!
  - Advantage: All cluster infra in one repo
  - Disadvantages:
    - Separated maintenance & versioning of app and infra code
    - Review spans across multiple repos
    - Local dev more difficult

Can’t we have both?

Yes, we can! Using a CI-Server



Disadvantages

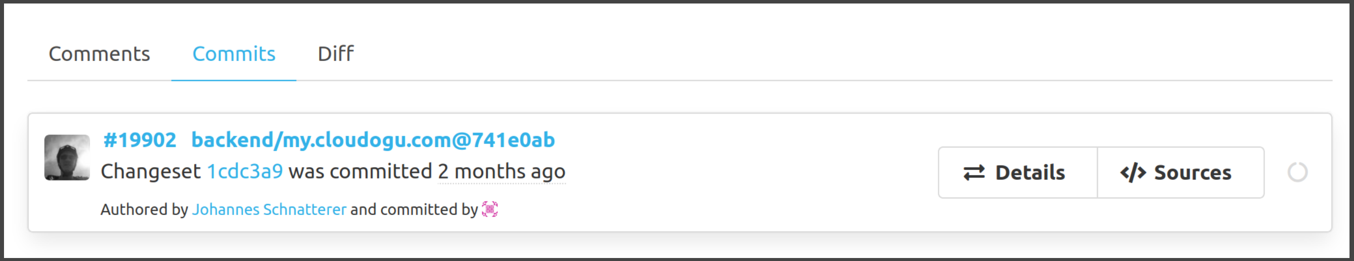
- Complexity in CI pipelines ➡️ efforts for development
- A lot can go wrong. Examples
  - Git Conflicts caused by concurrency
  - Danger of inconsistencies

➡️ Recommendation: Use a plugin or library

Example: cloudogu/gitops-build-lib

Advantages

- Fail early: static code analysis + policy check on CI server, e.g. yamlint, kubeval, helm lint, conftest
- Automated staging (e.g. PR creation, namespaces)
- Use IaC for local dev
- Write config files not inline YAML
  ➡️ Automatically converted to configMap
- Simplify review by adding info to PRs



#19902 backend/my.cloudogu.com@741e0ab

Changeset 1cdc3a9 was committed 2 months ago

Authored by Johannes Schnatterer and committed by

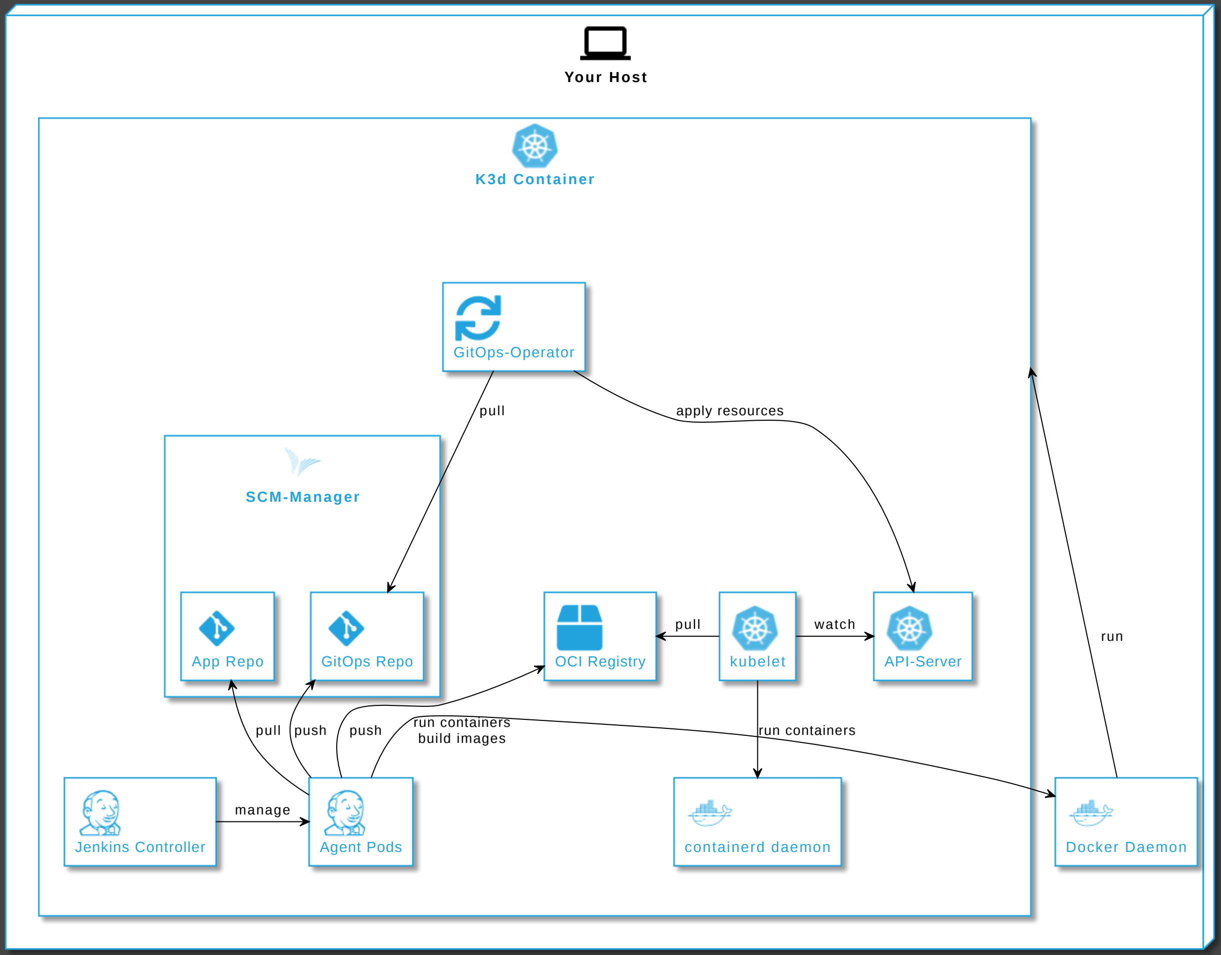
How to delete resources?

- ”garbage collection” (Flux) / ”resource pruning” (ArgoCD) disabled by default
- 📌 Enable from beginning ➡️ avoid manual interaction

Local development

- Option 1: Deploy GitOps operator and Git server on local cluster
  ➡️ complicated
- Option 2: Just carry on without GitOps.
  Easy, when IaC remains in app repo

Demo



cloudogu/gitops-playground

Conclusion

Personal Conclusion

After migrating to and operating with GitOps in production for > 1 year

- Smoother CI/CD,
  - *everything* declarative
  - faster deployment
- But: security advantages only when finished migration

GitOps experience distilled

+ Has advantages, once established

− Mileage for getting there may vary

Adopt GitOps?

- Greenfield
  - AppOps: Definitely
  - ClusterOps: Depends
- Brownfield: Depends

Johannes Schnatterer, Cloudogu GmbH

🌐 cloudogu.com/gitops

- ℹ️ GitOps Resources (intro, tool comparison, etc.)
- Links to GitOps Playground and Build Lib
- 💬 Discussions
- 🧑‍🏫 Trainings

Slides

Image sources

- What is GitOps? https://pixabay.com/illustrations/question-mark-important-sign-1872665/
- What can GitOps be used for? https://pixabay.com/photos/hammer-nails-wood-board-tool-work-1629587/
- How can GitOps be used? Tools: https://pixabay.com/photos/tools-knives-wrenches-drills-1845426/
- What challenges arise with GitOps? https://unsplash.com/photos/bJhT_8nbUA0