



// GITOPS: IS THIS SOMETHING FOR ME - OR NOT?

Johannes Schnatterer, Cloudogu GmbH



in in/jschnatterer @ @schnatterer@floss.social

Version: 202404301240-2284f7e

Agenda

- Basics
- Tools
- Challenges







GitOps basics

August 2017: GitOps - Operations by Pull Request

weave.works/blog/gitops operations by pull request

Use developer tooling to drive operations

GitOps Principles

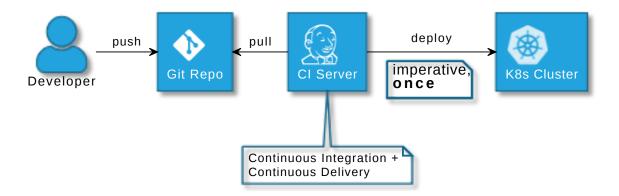
The desired state of a GitOps managed system must be:

1. Declarative

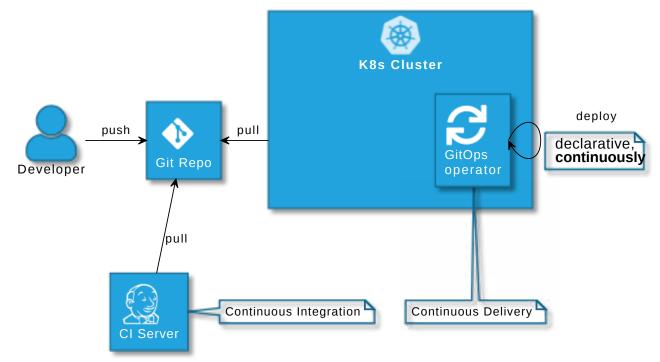
G

- 2. Versioned and Immutable
- 3. Pulled Automatically
- 4. Continuously Reconciled
- github.com/open-gitops/documents/blob/main/PRINCIPLES.md

"Classic" Continuous Delivery ("ClOps")

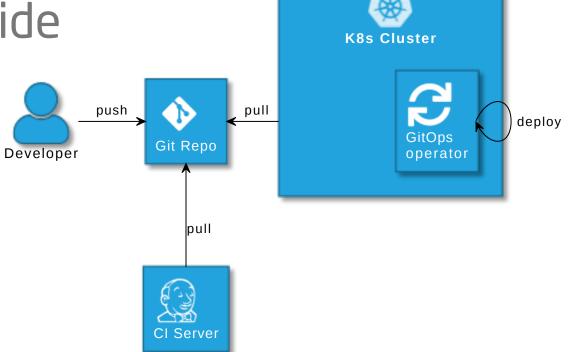


GitOps



Advantages of GitOps

- No access to cluster from outside
- No credentials on CI server
- Forces declarative description
- Config is auditable
- Self-healing
- Scalability one repo many applications
- Innovations provided by tools





GitOps operators/controllers











Supplementary GitOps tools

- Secrets Management
- Backup + Restore
- Deployment Strategies Progressive Delivery





















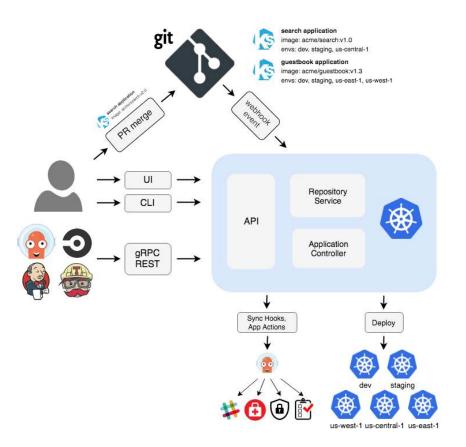


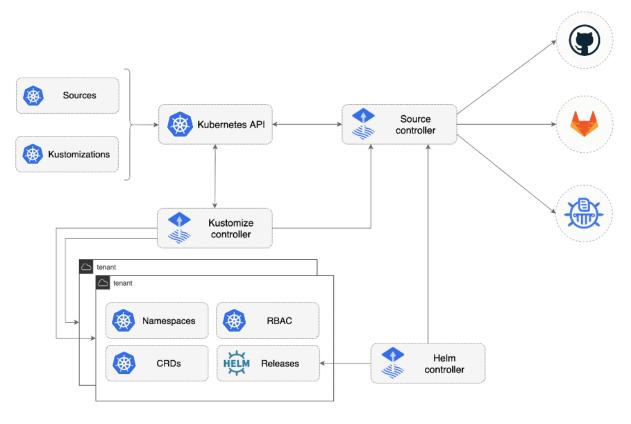
Downsides

- More infra necessary
- Steep learning curve
- Lots of day two questions

GitOps infra

- GitOps Operator comprises several applications
- Cause ops efforts: maintenance, alerts





Learning curve

- New concepts + tools for devs and platform teams
- Adapt deployment process + error handling
- Migrate applications

Day two questions

- How to realize local dev env?
- How to delete resources?
- How to structure repos and folders?
- How to realize different stages/environments?
- How to do monitoring and alerting?
- Role of Cl server?

Local development

- App development: Carry on without GitOps.
 Setting up GitOps operator, SCM, etc. complicated
- Integration: Run GitOps+k8s-based ops platform locally
 - cloudogu/gitops-playground
 - 2024-06-20 Workshop at CloudLand



Structuring repos

- No standard for structures (intentionally)
- How many repos?
- Within repo: folder/branch structure for stage, team, app
- More considerations:
 - Topology: GitOps controller (s) 😂 Cluster(s) / Namespaces
 - GitOps controller-specific config / CRDs
 - Multi-tenancy

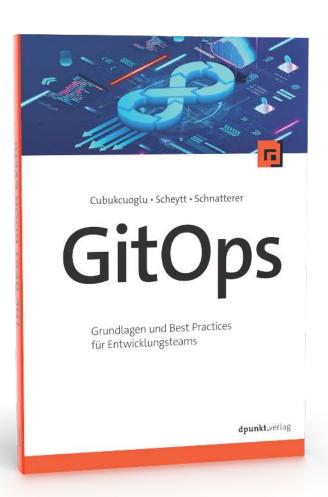
Promotion

How to model environments/stages?

- Use branches or folders?
- Operations by Pull Request: (How to) use PRs?
- How to integrate CM tools?
- How to automate?
- (How to) use preview environments?

Further reading

- Usual Slides (Continuous Lifecycle 23)
- Blog post series (Cloudogu Blog)
- GitOps book
- Cloudogu/gitops-patterns





Is GitOps something for me - or not?

- + Has advantages, once established
- Don't underestimate learning curve and migrations

Johannes Schnatterer, Cloudogu GmbH

@ @schnatterer@floss.social in in/jschnatterer

Cloudogu/gitops-playground

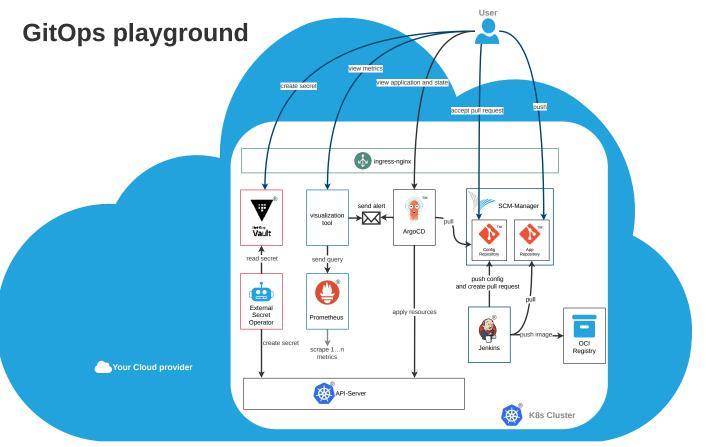




Image sources

- Basics:
 - https://pixabay.com/illustrations/question-mark-important-sign-1872665/
- Tools:
 - https://pixabay.com/photos/tools-knives-wrenches-drills-1845426/
- Challenges:
 - https://unsplash.com/photos/bJhT_8nbUA0
- Is GitOps something for me?
 - https://pixabay.com/illustrations/question-mark-question-response-1020165/