// INTRODUCTION TO GITOPS – A NEW AGE OF AUTOMATION?

Johannes Schnatterer, Cloudogu GmbH

@jschnatterer

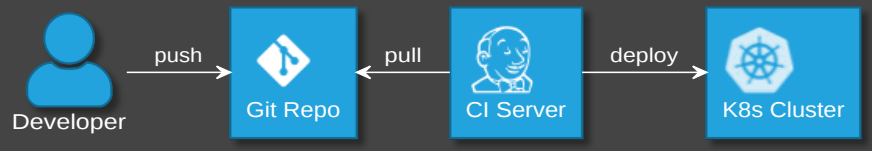
Version: 202104161337-035c88f

Agenda

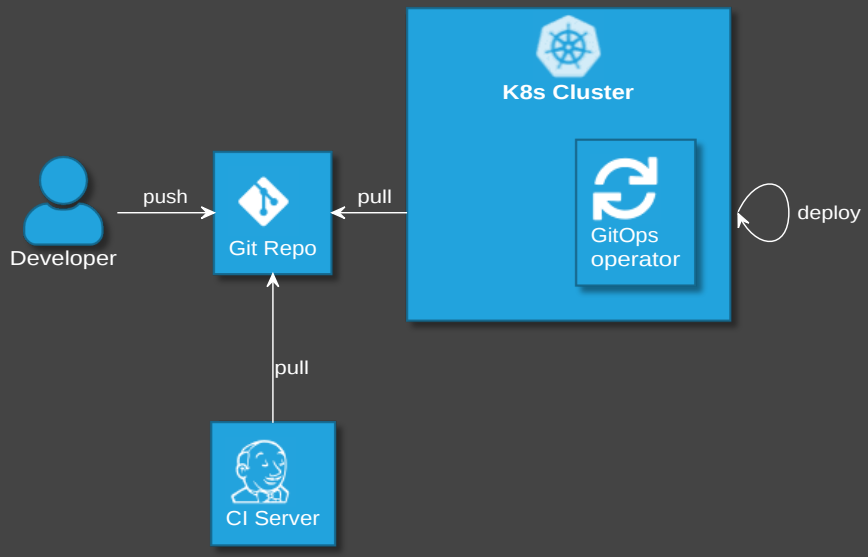
- What is GitOps?
- Where can it be used?
- How can it be used?
- What challenges arise?

What is GitOps?

"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 state reconciliation
4. The principle of operations through declaration

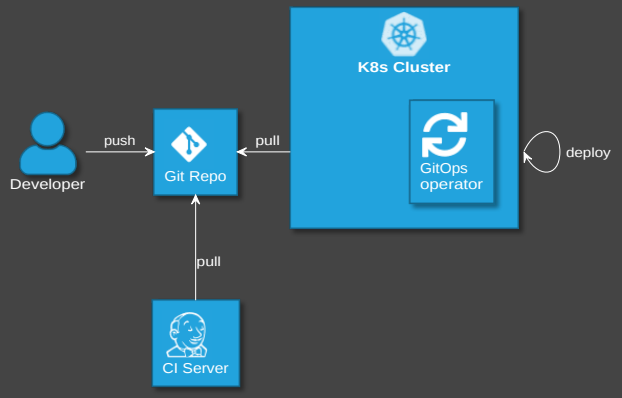
WIP!

github.com/gitops-working-group/gitops-working-group/pull/48

hackmd.io/arwvV8NUQX683uBM3HzyNQem

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?

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

How can GitOps be used?

Demo

github.com/cloudogu/k8s-gitops-playground

What challenges arise with GitOps?

For dev and ops

Personal Conclusion

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

- Smoother CI/CD,
  - *everything* declarative
  - faster deployment
  - force sync desired state ↔️ actual state
- But: security advantages only when finished migration
- A lot of potential ahead!

GitOps experience distilled

+ Has advantages, once established

— Mileage for getting there may vary

Adopt?

- Greenfield
  - Kubernetes AppOps: Definitely
  - Cloud Infra: 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
- 👩‍🏫 Training

Slides