GitOps The Big Picture:

UNDERSTANDING GITOPS

Kien Bui
DevOps & Platform Engineer

Overview

- Understanding What Gitops Is
- Understanding the Need for GitOps
- Understanding the Benefits of GitOps
- Understanding the Difference between GitOps and DevOps
- Understanding the Difference between GitOps and IaC
- GitOps Principles and Practices

Understanding What Gitops Is

GitOps

“GitOps is an operating model pattern for cloud native applications & Kubernetes storing application & declarative infrastructure code in Git as the source of truth used for automated continuous delivery.”

GitOps Is

GitOps is a fast-growing "operating model" pattern for cloud native applications commonly associated with Kubernetes

GitOps allows you to push code not containers

Git is the Source of Truth describing the desired state of your entire system

If it can be described it can be automated with GitOps such as:
apps, config, dashboards, monitoring etc..

GitOps Without Kubernetes

But, my organization does not have Kubernetes. Does GitOps still apply to us?

GitOps is not limited to Kubernetes. You can use GitOps with any system that can be observed and described declaratively. Currently the majority of pull-based GitOps operators were built for Kubernetes.

GitOps Journey

August 7th, 2017 the term GitOps was introduced by *Alexis Richardson* of Weaveworks in the '*Operations by Pull Request*' blog

Since the blog post, GitOps has grown seeing a buzz in the tech community online and places like KubeCon

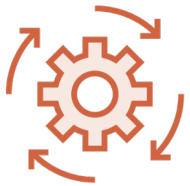
There have been many tools developed to support GitOps, numerous talks, articles, blogs, & even several books on the way about it

December 10th, 2020 the GitOps Working Group hosted its first meeting as a step towards governance, documentation, guidelines, & a clear definition for GitOps

Understanding the Need for GitOps

Use Cases for GitOps

Cloud Native App Management i.e. “CD” in CI/CD



Service Rollouts

Infrastructure management i.e. K8s clusters, fleets, microservices

Kubernetes App/Config Deployment Anti-Pattern

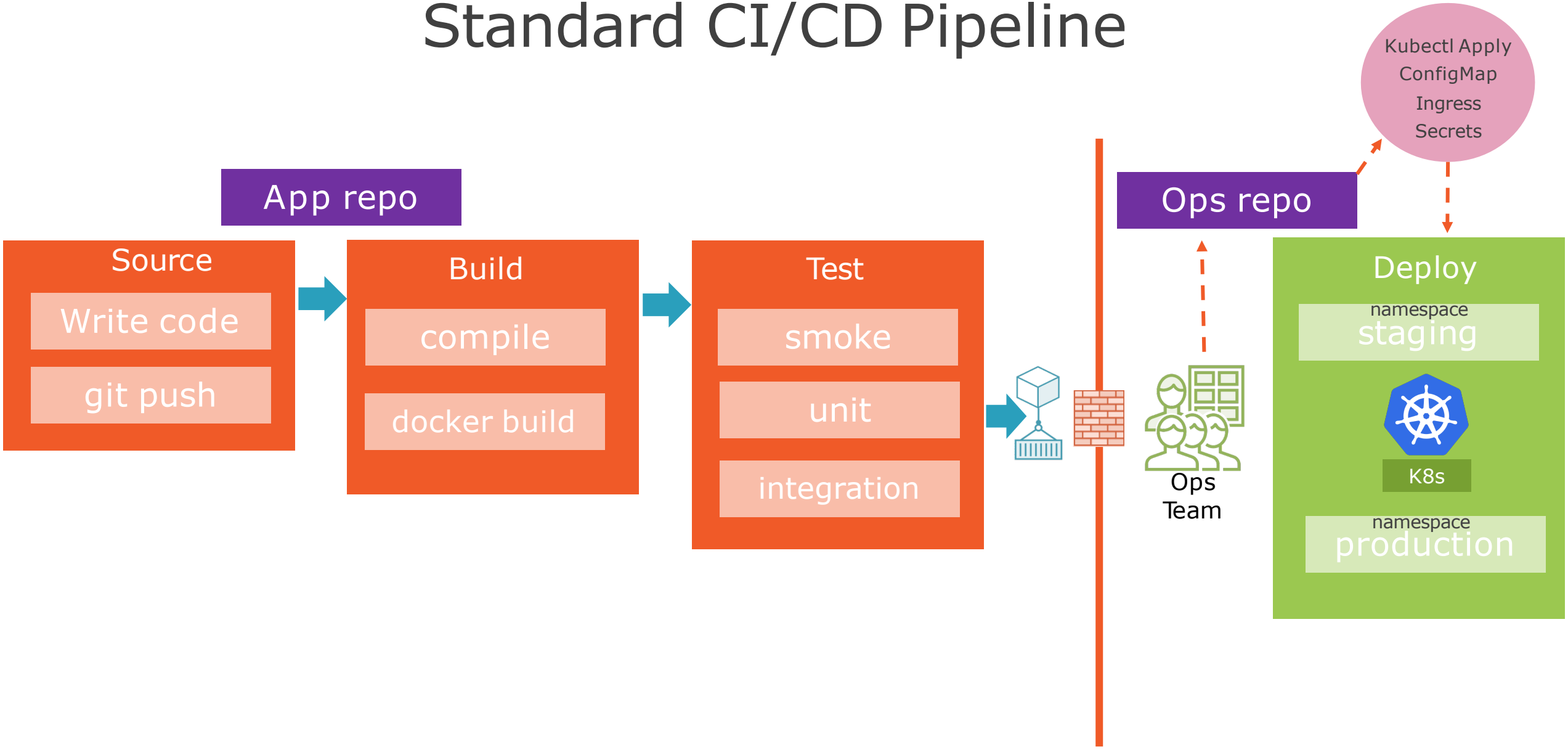
One of the problems with deployments to K8s is that a dev team will build an app, package it, then hand it to an ops team for deployment

The ops team will update IaC config scripts & will use them to deploy the app to K8s

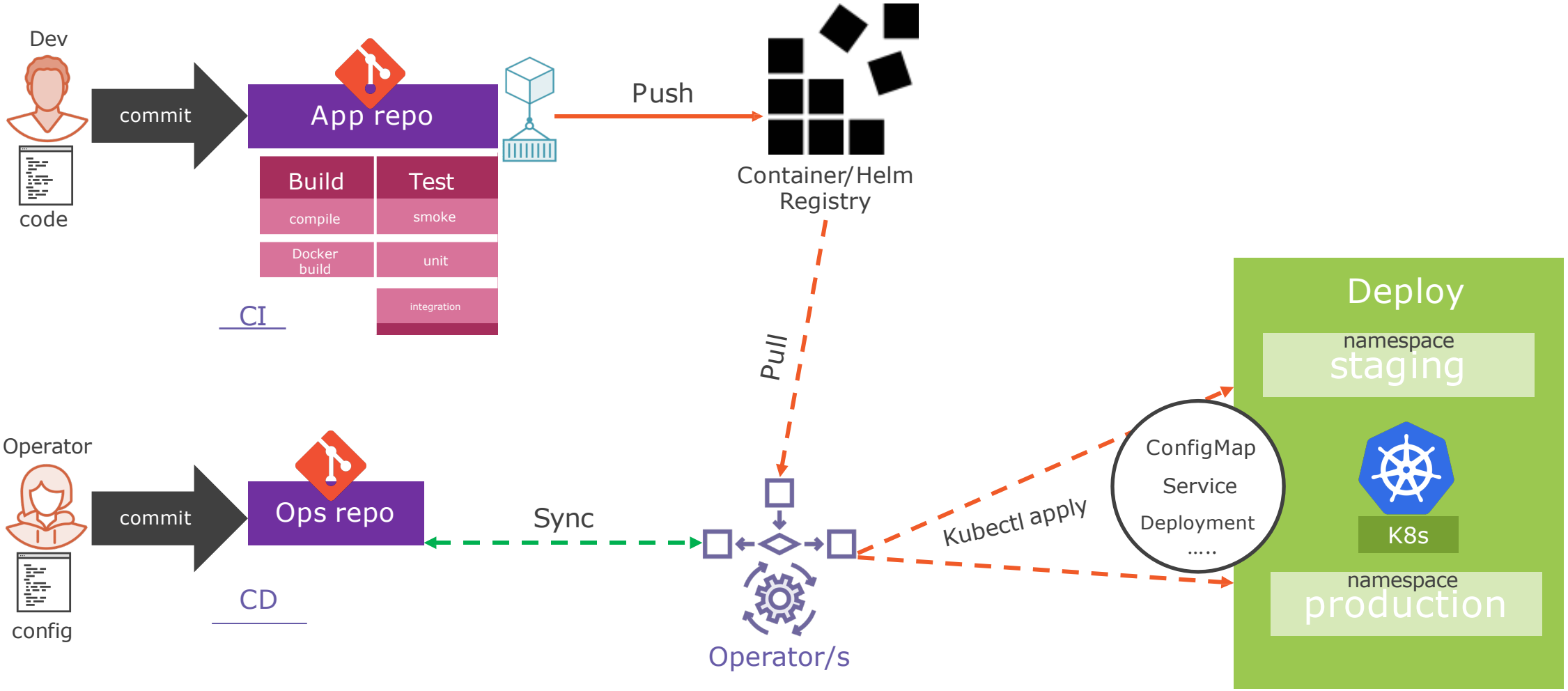
With this method the code in the repo is still somewhat disconnected from the live environment

When an update to the app or environment config is needed IaC config scripts are updated and applied manually to the live environment

Standard CI/CD Pipeline



GitOps Pipeline



Understanding the Benefits of GitOps

Benefits of GitOps

App & Ops in Git

Enhanced Developer Experience

Continuous Syncing

More Reliability

Simplified Operations

Increased Productivity

Full Audit Trail

Compliance & Stability

Everything as a code

Rollback, Consistency & Standardization

Continuous Security

Access shift, Security as Code, Credential & state segregation

Understanding the Difference between GitOps and DevOps

The Difference Between Gitops and DevOps

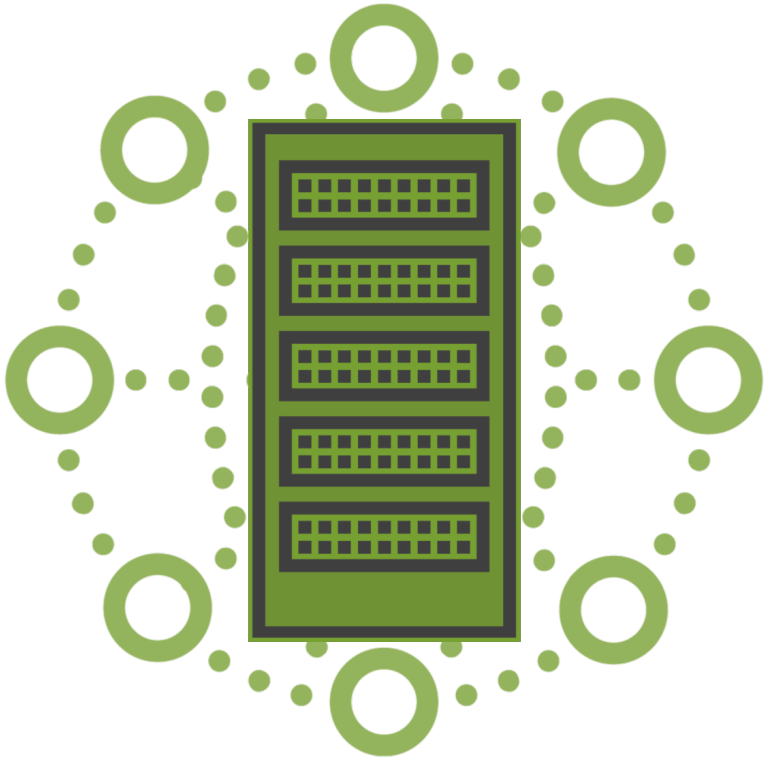
DevOps often takes a “Push” approach while GitOps takes a “Pull” approach to pipeline workflow

In DevOps the app & deployment pipelines are often separate with IaC scripts used for a static one-time deployment of the environment

GitOps replaces scripts of kubectl, Terraform, & Helm with an operator that handles operations tasks such as create, change, delete in a Kubernetes cluster based on what's described in Git

Understanding the Difference between GitOps and IaC

What’s The Difference Between GitOps and IaC



Declarative IaC actually plays a key role in GitOps as GitOps syncs the state of the live environment with the IaC in the Git repo

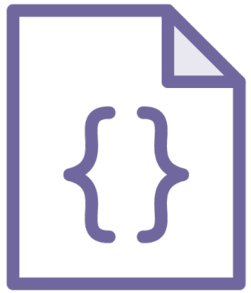
Declarative IaC is a part of GitOps – Terraform, K8s Manifests etc., Helm

GitOps applies the Git ecosystem & tooling to an infrastructure

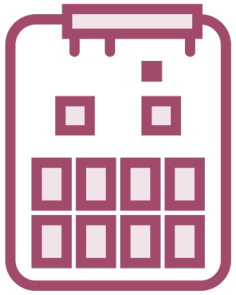
GitOps Principles and Practices

GitOps Principles

Git is the source of truth for entire system

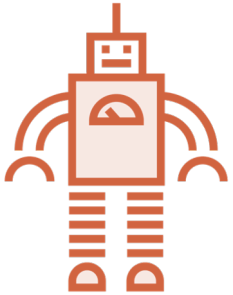


Desired system state is versioned in Git



System state described declaratively

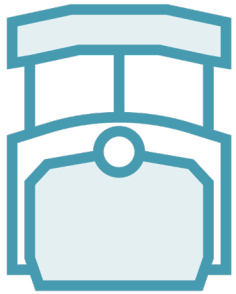
Git as the single place for operations (create, change, delete)



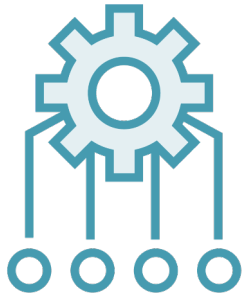
Autonomous Agents enforce desired state and alert on drift

Automated delivery of Approved system state changes

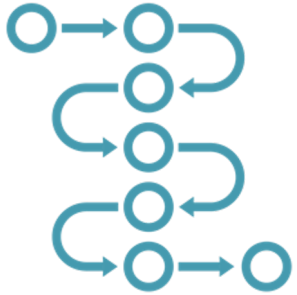
GitOps Practices



Pull over Push



At least 2 Repos per App. One for App Source Code &second for Config (manifests)



Ensure you Test

Have a plan for Secrets management

Have a plan for backups

Summary

In this module we covered:

- Learned what GitOps is, why its needed, & what the benefits of it are
- We looked at the differences between GitOps & DevOps/IaC as well as how they fit together
- We talked through the GitOps Principles and Practices

Why this is important:?

- Before adopting a new operating model pattern like GitOps it is key to understand what it is, how it applies to patterns you already know, as well as its principles
- This helps understand how it can apply to your environment and scenario