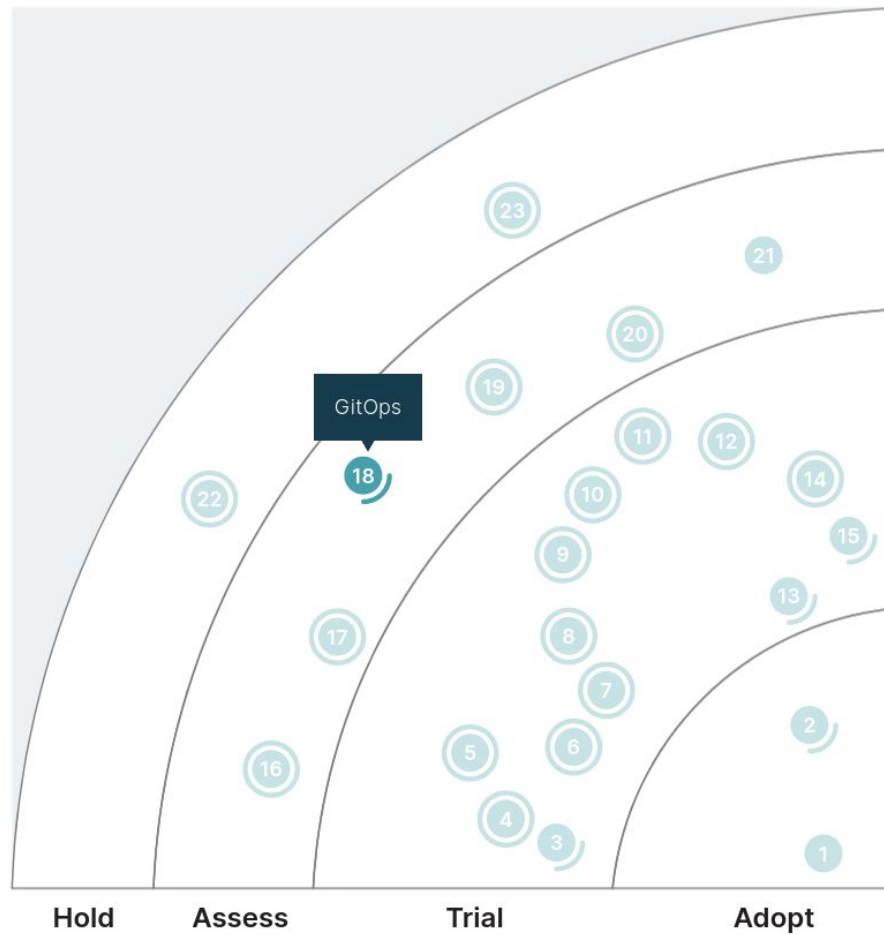


GitOps

Make changes by committing to a git repo



SEP
2023

Assess ?

GitOps is a technique for deploying applications via the control loop ↗ pattern. An operator keeps the deployed application synchronized with configuration, usually a Git repository. When we last wrote about GitOps, the community had yet to agree on a definition of the term. At the time, we were concerned about common interpretations of the technique that included approaches like "branch per environment" for configuration, which may lead to snowflakes as code ↗. Moreover, the messaging around GitOps as an alternative to continuous delivery was confusing. Since then, the four GitOps principles ↗ have clarified the scope and nature of the technique. When you peel away the hype and confusion, GitOps is a useful technique that takes advantage of the functionality of a Kubernetes cluster and creates opportunities to separate concerns between configuring an application and the implementation of the deployment process. Some of our teams have implemented GitOps as part of their continuous delivery setup with positive experiences, which is why we recommend assessing it.

GitOps Principles

v1.0.0

1 Declarative

A system managed by GitOps must have its desired state expressed **declaratively**.

2 Versioned and Immutable

Desired state is **stored** in a way that enforces immutability, versioning and retains a complete version history.

3 Pulled Automatically

Software agents automatically pull the desired state declarations from the source.

4 Continuously Reconciled

Software agents **continuously** observe actual system state and **attempt to apply** the desired state.

GET INVOLVED

(from:

<https://opengitops.dev/>)

a repo we can look at:

<https://github.com/engineering-llc/ctf-workshop>

Pros:

- Automatically fixing drift
- Single source of truth
- Helps with promoting between environments

Cons:

- Can slow down the iteration cycle

Tech Radar rating: **Assess**

My rating: **Adopt**