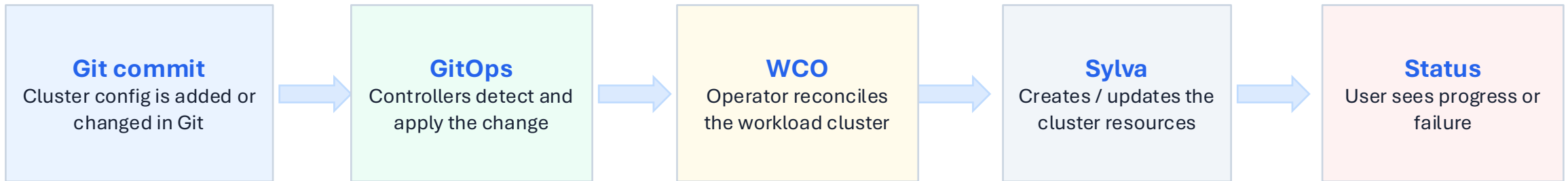


Background: WCO + GitOps deployment



Simple meaning

The user describes the desired workload cluster in Git. WCO works to make the real cluster match that desired state.

Why observability matters

A failed deployment can involve several layers: SWC, SylvaUnitsRelease, Flux, HelmRelease, GitRepository and CAPI resources.

GitOps Time Machine

Deployment history + early failure prediction for Sylva workload clusters

Git is source of truth

Purpose

Help workload teams quickly understand what changed, which Git commit caused a problem, and whether a live deployment is becoming risky.

What History and Prediction will do

1. History

- Records every important workload-cluster deployment state
- Links each deployment state to the related Git revision
- Shows which Git change introduced the issue
- Tracks recovery after a revert or configuration fix
- Provides a timeline for comparing healthy and failed deployments

Tools planned to be used:

- Go, Controller-runtime, kubebuilder, Kubernetes CRD's

2. Prediction

- Watch live deployment signals
- Detect repeated failures
- Compare with previous failure patterns
- Show Low / Medium / High risk before final failure

Tools planned to be used:

- Lightweight prediction model like “scikit-learn” which is a Python AI and machine learning library

GitOps Time Machine Explains

- What Changed
- Which Git commit caused it
- Has it happened before?
- Is the deployment likely to fail?