KubeCon CloudNativeCon North America 2026

#KubeCon #CloudNativeCon

Robust CI as a Safety Net: Moving Fast at Scale with AI-Generated GitOps

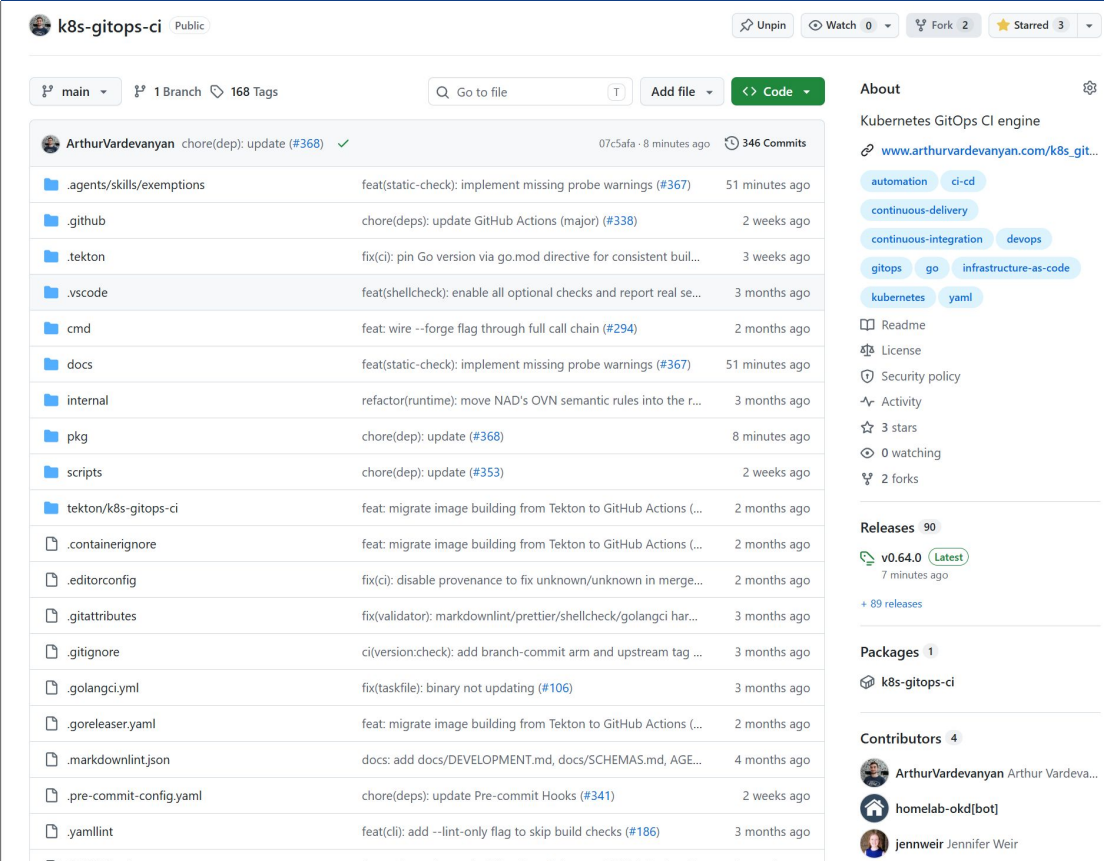
Arthur Vardevanyan

Jennifer Weir



k8s-gitops-ci

KubeCon CloudNativeCon North America 2026

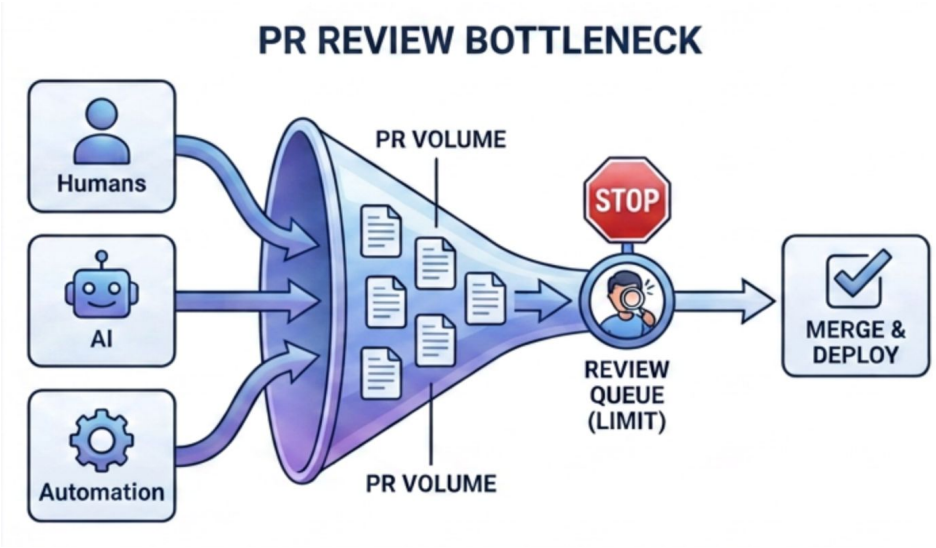


Disclaimer

The views and opinions expressed in this presentation are our own and do not necessarily reflect those of our employer.

Problem Statement

- **The Environment:** Operating a large-scale Kubernetes platform across hundreds of clusters using GitOps.
- **The Problem:** A high volume of daily pull requests from a mix of humans, AI, and automated systems.
- **The Bottleneck:** This overload makes the code review process inefficient, unreliable, and difficult to scale.



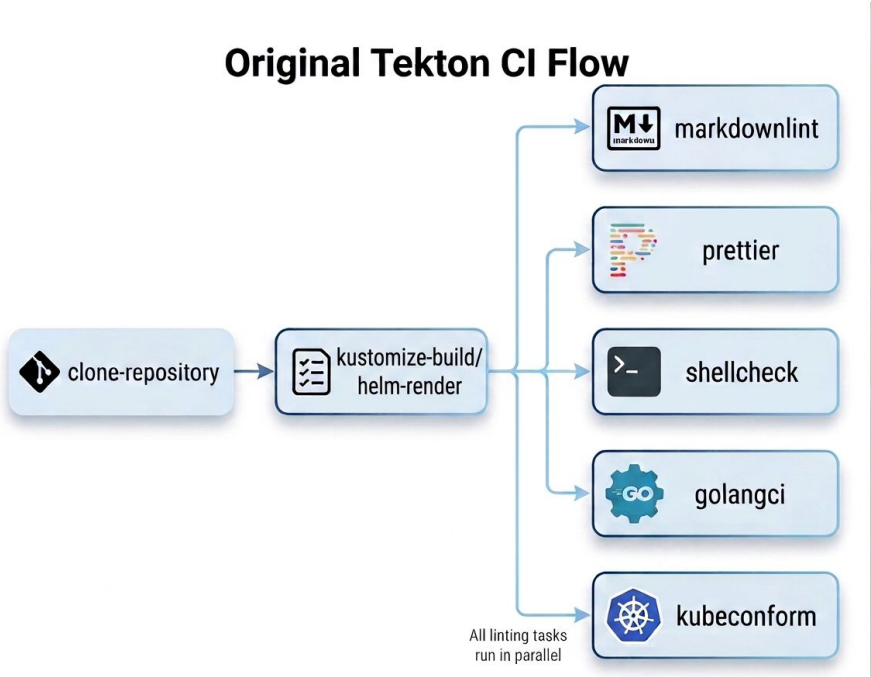
Operational Problems

Pre-CI (before 2023)

- Lack of automated validation
- Broken Kustomize, bad manifests, typos
- Slow due to excessive rework & inaccuracy

Original Tekton CI (2023 – early 2026)

- Multi-task, modular Tekton pipeline with many containers
- Lots of moving parts with no unit tests
- Security concerns via potential CI injection
- Running locally was challenging



The Modern Problem: AI

KubeCon CloudNativeCon North America 2026

FOUNDATION



STRONG KNOWLEDGE CONTEXT

All effective AI automation begins with a solid foundation of contextual knowledge.

MULTIPLIER



MORE OUTPUT (GOOD & BAD)

AI dramatically raises the ceiling on output, but CI must raise the floor on quality.

PLAUSIBILITY TRAP

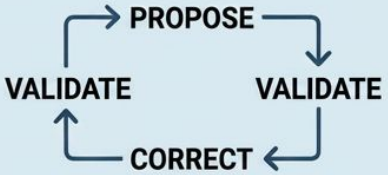


PLAUSIBLE ≠ CORRECT

Deterministic guardrails are required for non-deterministic AI authors.

ENGINE

AGENTIC LOOP

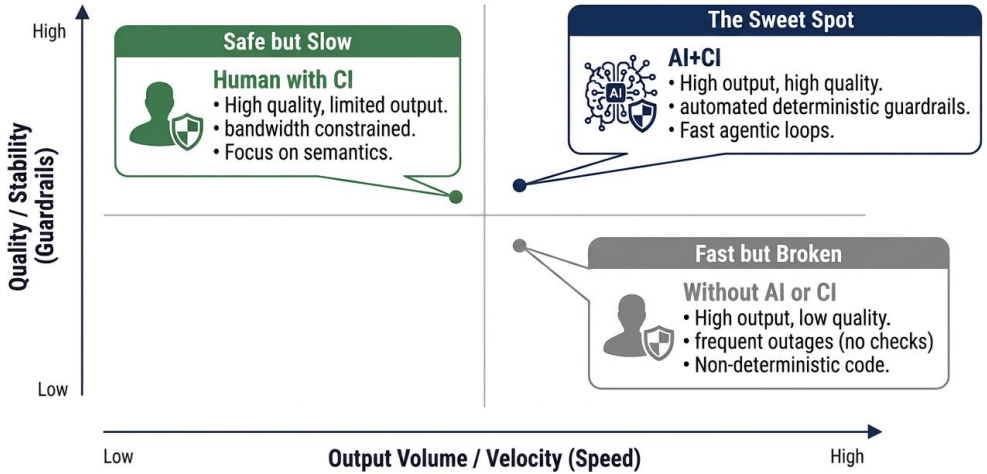


AGENTIC LOOP

Fast feedback in seconds, not minutes, is critical for AI agent loops.

Robust CI as a Safety Net

- Best practices do not change, but the scale does.
- As AI increases output, CI provides the deterministic guardrails needed to ensure quality.
- This allows reviewers to focus on the content, not the semantics.
- It provides the fast feedback required for AI's iterative loops.
- Ultimately, CI acts as a universal safety net that runs anywhere.



KubeCon CloudNativeCon North America 2026

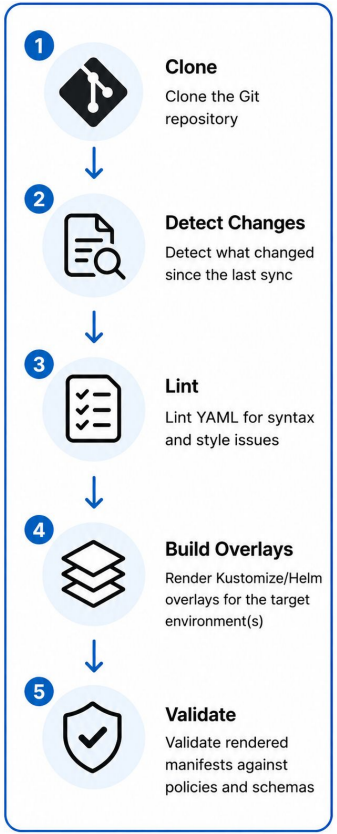
Modernized Solution: k8s-gitops-ci

https://github.com/ArthurVardevanyan/k8s-gitops-ci

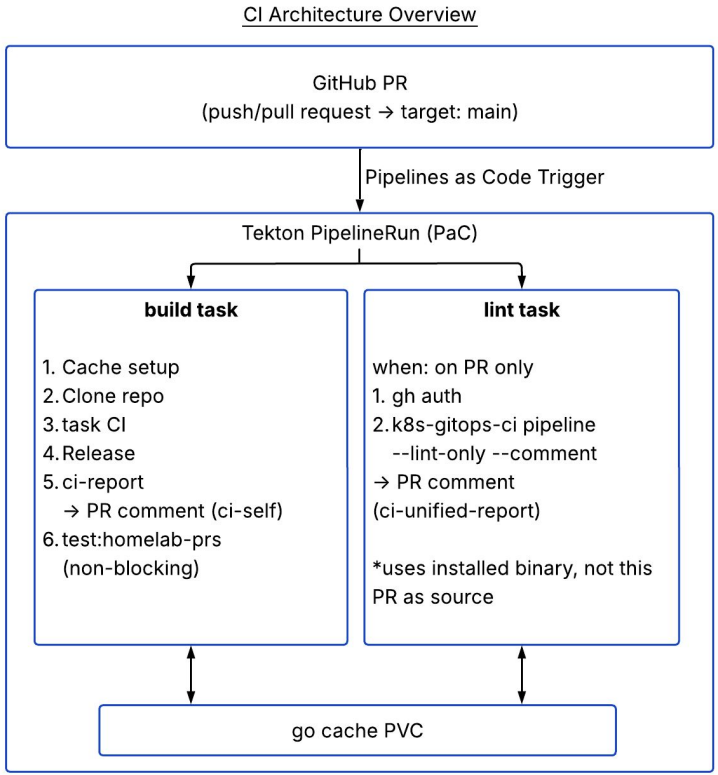


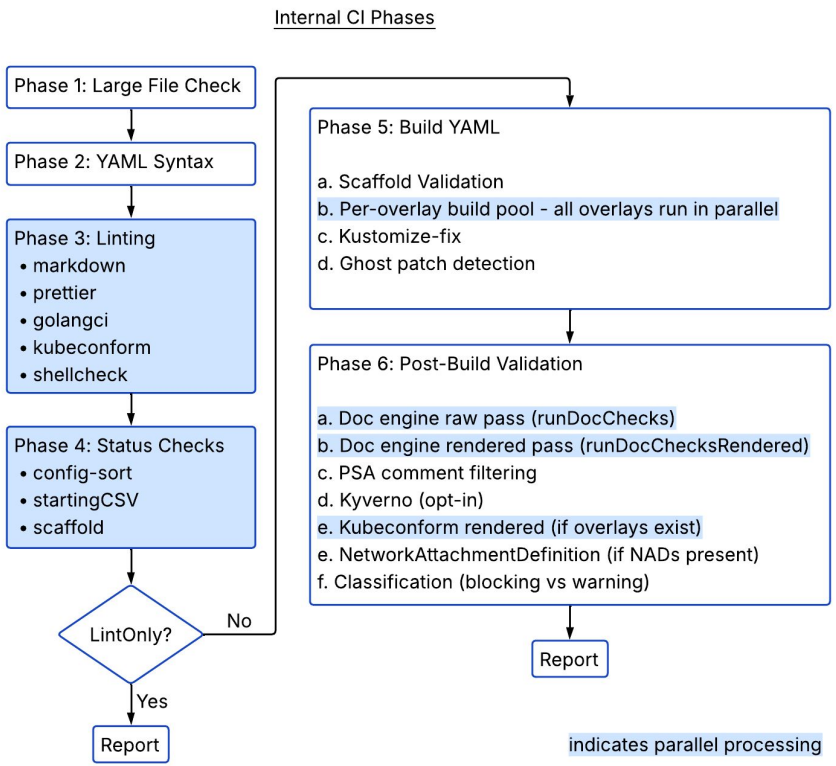
Packaged as a Single Go Binary

- Unit testing framework
- Change-aware validation
- Native-first & shell-out only when needed
  - Native Go: Kustomize build, Helm template, YAML parsing, kubeconform, most validators
  - Shell-out (only for external tools): scafctl scaffolding, kyverno, prettier, shellcheck, git
  - CEL validation evaluates x-kubernetes-validations rules extracted from CRD schemas using apiserver CEL libraries
- Built for speed
  - Independent checks run concurrently
  - Deduplication: identical YAML validated once

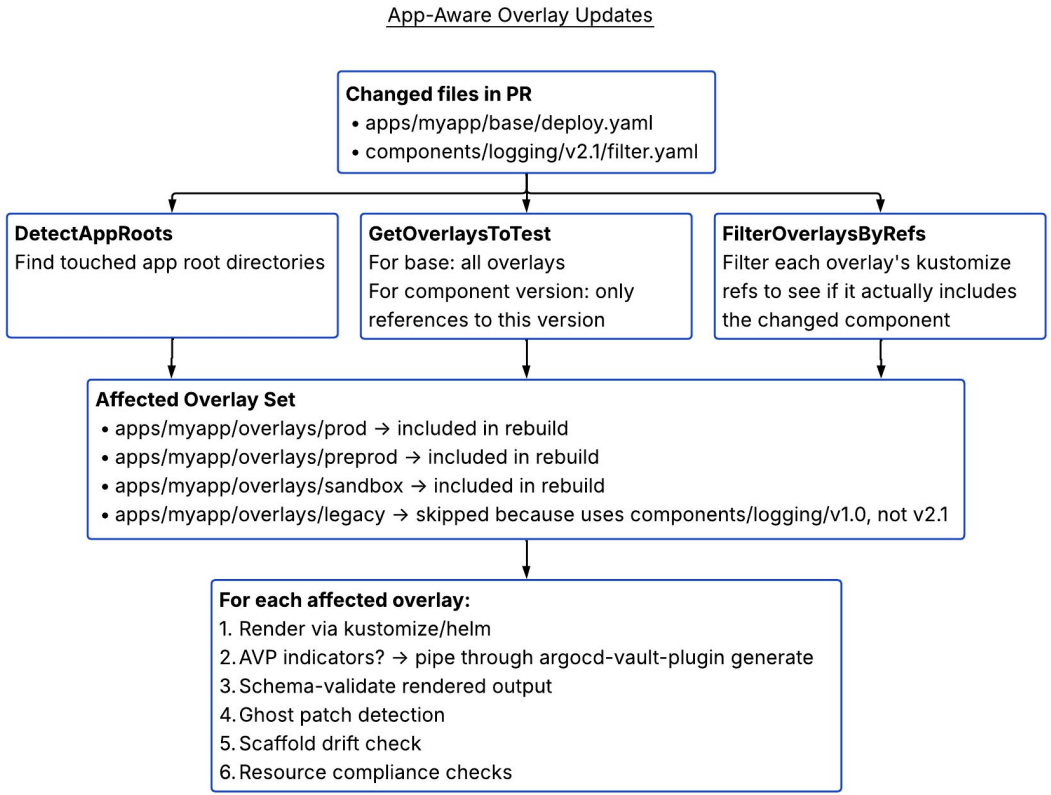


Architecture Design



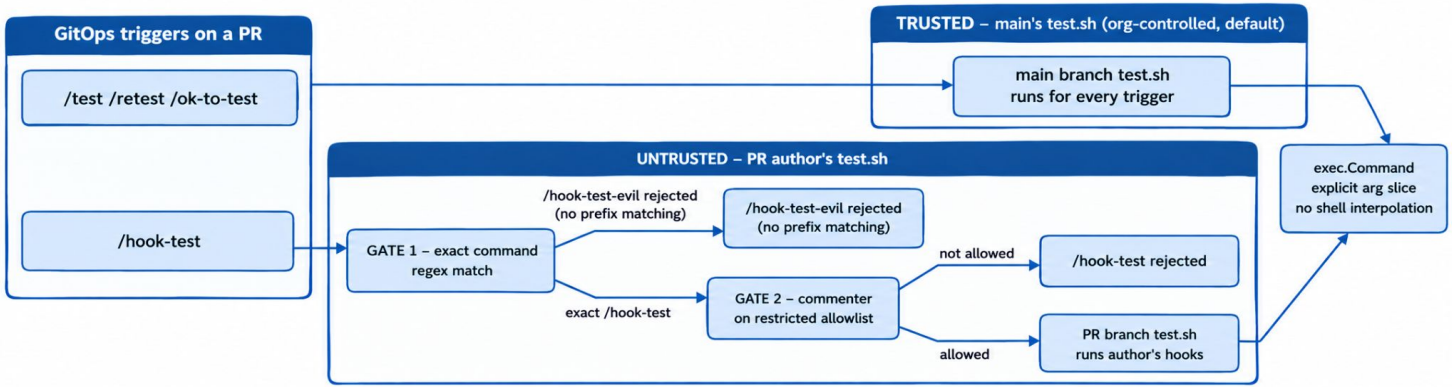


Overlay Discovery



/hook-test security model

- test.sh (arbitrary hook shell functions) is read from main by default — PR authors cannot inject hooks into the CI run validating their own PR.
- **/hook-test is the only path to a PR's own test.sh**
  - exact-command regex match (no /hook-test-evil prefix match)
  - restricted to certain people
  - all other gitops comments (/test, /retest, /ok-to-test) use main
- exec.Command never uses shell interpolation — args passed as explicit slices



Runtime Validation Logic

- **Static vs Runtime**
  - **Schemas only check shape, not values**
    - No enums, ranges, or patterns. Names are not validated at all.
  - **Validating Admission Rules**
    - Not expressed in schema definitions.
- **We port the API server's own rules**
  - Kubernetes does not ship them as a library, reimplemented against the typed structs.
- **Catches real rejections early**
  - Bad values, negative replicas, duplicate ports, mismatched selectors, invalid cron syntax.
- **Every rule cites verified upstream source**
  - CI digests the cited function; if upstream changes, the build fails.
- **Extensible**
  - Pattern can extend to other operators with runtime validators
- **Always blocking, no exemptions**

Exemption Mechanism Security

ℹ️ Exemptions are visible on PR comments via an INFO log line and a non-blocking sub-check

1. gitops-ci.k8s.io/exempt-<check-id> annotations are inline/self-granting

```yaml
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: system:controller:operator-lifecycle-manager
  annotations:
    gitops-ci.k8s.io/exempt-rbac-wildcards: "apiGroups, resources, nonResourceURLs,
rules:
  - apiGroups:
      - "*"
    resources:
      - "*"
```

```
#!/bin/bash

export EXEMPTIONS=(
  "check=kubeconform,file=install-config.yaml"
  "check=kubeconform,file=vm-config.yaml"
  "check=kubeconform,file=node-config/gpu-1.yaml"
  "check=kubeconform,file=node-config/worker-1.yaml"
  "check=kubeconform,file=node-config/worker-2.yaml"
  "check=kubeconform,file=node-config/worker-3.yaml"
  "check=placeholder,file=install-config.yaml"
)
```

2. EXEMPTIONS selectors read from main (out-of-band merged PR before taking affect)

`cel`	The CEL validation step (`pkg/validator/phases.go`) — validates `x-kubernetes-validations` from embedded CRD schemas .	**No — deliberately non-exemptable.** CEL rules are enforced by the Kubernetes API server at admission time, so an exemption in CI would not prevent the cluster from rejecting the manifest.

https://github.com/ArthurVardevanyan/k8s-gitops-ci/blob/main/docs/EXEMPTIONS.md

Extensible By Design

A generic validation engine, extended cleanly for organization-specific conventions and behavior

Configuration Overrides

- Package-level settings overwrite default behavior/naming
- Applied once at startup, before any pipeline stage runs
- Enables an org to set its own defaults

Pluggable Provider Interfaces

- Well-defined interfaces; orgs supply custom implementations
- Report branding, comment handling, cluster metadata
- Sensible, generic defaults when no customization exists

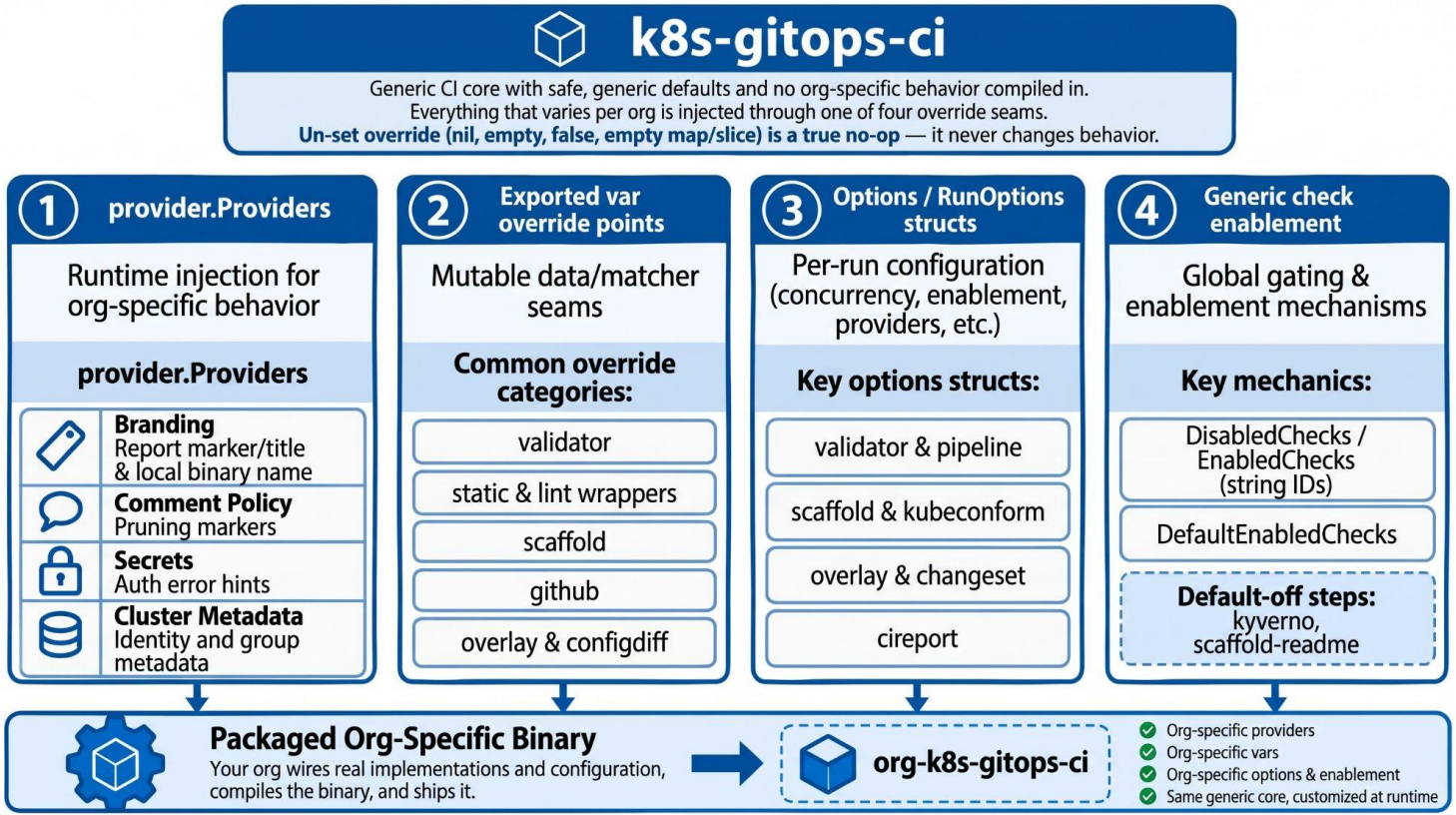
Embedded Policy & Schema Data

- Validation schemas supplied and independently versioned
- Bundled into org's build; no modification to shared engine
- Keeps org-specific rules isolated and auditable

Clean, One-Way Dependency

- Org layer depends on the core -- never the reverse
- Org-specific rules stay local while updates occur upstream
- Shared core updates are simple, low-risk version bumps

Organization Level Overrides



Examples

KubeCon CloudNativeCon North America 2026

GitOps CI Results

Last Updated: 2026-08-21T21:26:21Z

▶ ✅ Expand: PR Checks

▶ ✅ Expand: Linting

▶ ✅ Expand: Static Checks

▶ ✅ Expand: Kustomize Build

▶ ✅ Expand: Scaffold Validation

▶ ✅ Expand: Scaffold Drift Protection

▶ ℹ️ Expand: Resource Compliance

▶ ✅ Expand: NetworkAttachmentDefinition Validation

▶ ✅ Expand: CI Notes

```
k8s-gitops-ci pipeline --url="https://github.com/ArthurVardevanyan/HomeLab" --pr=650 --assume-openshi
```


GitOps CI Results

Last Updated: 2026-08-21T21:26:21Z

▶ ✅ Expand: PR Checks

▶ ✅ Expand: Linting

▼ ✅ Expand: Static Checks

▶ ✅ Large File

▶ ✅ YAML Syntax

▶ ✅ Config Sort Order

▶ ✅ Starting CSV

▶ ✅ Scaffold Table

▼ ✅ Expand: Kustomize Build

▶ ✅ Overlay Build

▶ ✅ Hooks

▶ ✅ Kustomize Fix

▶ ✅ Ghost Patches

▶ ✅ Expand: Scaffold Validation

▶ ✅ Expand: Scaffold Drift Protection

Examples

GitOps CI Results

Last Updated: 2026-08-26T13:29:23Z

▶ ✅ Expand: PR Checks

▶ ❌ Expand: Linting

▶ ✅ Expand: Static Checks

▶ ❌ Expand: Kustomize Build

▶ ✅ Expand: Scaffold Validation

▶ ✅ Expand: Scaffold Drift Protection

▶ ❌ Expand: Resource Compliance

▶ ✅ Expand: CI Notes

```
k8s-gitops-ci pipeline --url="https://github.com/jennweir/homelab" --pr=166
```

▼ ❌ Expand: Resource Compliance

If the affected resource is being modified in this PR, these issues **must** be corrected.
Otherwise, these are non-blocking warnings for pre-existing issues.

▼ ⚠️ Image Digest Pinning (3 finding(s))

Container images not pinned to a SHA256 digest.

Kind	Name	Image	Overlays
Deployment	cert-manager-webhook	quay.io/jetstack/cert-manager-webhook:v1.18.2	`k8s/manifests/platform/cert-manager/base/v1.18.2/deployment-cert-manager-webhook.yaml`
Deployment	cert-manager-cainjector	quay.io/jetstack/cert-manager-cainjector:v1.18.2	`k8s/manifests/platform/cert-manager/base/v1.18.2/deployment-cert-manager-cainjector.yaml`

▼ ❌ PSA Namespace Labels (1 finding(s))

Namespaces missing required Pod Security Admission labels.

Namespace	Missing	Source File(s)	Overlays
cert-manager	: Namespace "cert-manager" missing PSA labels: [pod-security.kubernetes.io/enforce, pod-security.kubernetes.io/enforce-version, pod-security.kubernetes.io/warn, pod-security.kubernetes.io/warn-version, pod-security.kubernetes.io/audit, pod-security.kubernetes.io/audit-version]	`k8s/manifests/platform/cert-manager/base/v1.21.1/namespace.yaml`	1

KubeCon CloudNativeCon
North America 2026

Feedback

Robust CI as a Safety Net: Moving Fast at Scale with AI-Generated GitOps

