GitOps in practice: Argo CD, Flux and policy-as-code

26 October 2026 · 5 min read · blogs.iictp.org/gitops-in-practice

GitOps keeps the desired state of a system in Git and uses an automated agent to make the real system match it. Changes happen by pull request, and the history of Git becomes the audit trail.

TL;DR The short version

- GitOps keeps a system's desired state in Git and uses an agent to make the real system match it.
- Argo CD and Flux are two widely used CNCF tools that do this for Kubernetes.
- Every change is reviewed and recorded, rollback is a revert, and policy as code can block unsafe changes before they apply.

Read online, listen to it, or change the text size: https://blogs.iictp.org/gitops-in-practice

The principles

- The desired state is declared, for example as Kubernetes manifests.
- It is stored in Git and versioned.
- An agent in the cluster pulls changes and applies them.
- The agent keeps correcting drift.

The tools

Argo CD and Flux are two widely used CNCF projects that do this for Kubernetes. They watch a repository and reconcile the cluster with it.

Benefits

- Every change is reviewed and recorded.
- Rollback is a revert.
- Rebuilding an environment is repeatable.

Adding policy

Policy-as-code tools check changes against rules before they are applied, for example blocking images from unapproved registries. Keep secrets out of Git, and use a secrets manager instead.

Related credential: Agile Delivery Practitioner. Explore the exam facts and syllabus.

Sources and further reading

1. OpenGitOps. GitOps Principles. opengitops.dev.
2. Argo CD. argo-cd.readthedocs.io.

Published by IICTP, an independent technical credential body, on 26 October 2026. The current version of this post is at https://blogs.iictp.org/gitops-in-practice. More guides: blogs.iictp.org.