

Git Operations (GitOps)

[Go up to Terms and Definitions](#)

Discussion

Git Operations (GitOps) applies [Git](#) and [Declarative Configuration](#) practices to the operation and management of software systems, applications, infrastructure, or services.

In a GitOps approach, a [Version Control System](#) contains the authoritative declaration of the desired operational state.

GitOps applies four core principles:

- Declarative
- Versioned and immutable
- Pulled automatically
- Continuously reconciled

A GitOps implementation:

- Represents the desired state declaratively
- Records the desired state in a Version Control System
- Identifies each revision of the desired state
- Detects changes to the recorded desired state
- Retrieves approved changes automatically
- Compares the actual operational state with the desired state
- Reconciles differences between the actual state and the desired state
- Reports reconciliation status or failure

GitOps uses Git repository history to provide:

- Change identification
- Revision history
- Reviewable changes
- Rollback targets
- [Provenance](#)
- [Reproducibility](#)
- [Traceability](#)

Within Crucible documentation, GitOps provides an operational approach for applying approved Declarative Configuration from Git repositories to managed environments and reconciling those environments with the recorded desired state.

Definition

operational approach that records declarative desired state in a Version Control System and

automatically reconciles actual operational state with that desired state

Source

Adapted from:

- OpenGitOps, *GitOps Principles*
- Cloud Native Computing Foundation, *Cloud Native Glossary: GitOps*

Note

The term **GitOps** combines **Git** and **Operations**.

GitOps uses [Git](#) as the conventional Version Control System, but its architectural principles depend on controlled versioning, immutability, automatic retrieval, and continuous reconciliation rather than on a particular repository-hosting product.

GitOps differs from a general Git-based [Workflow](#). A Git-based Workflow uses Git to coordinate work. GitOps additionally requires declarative desired state and automated reconciliation of actual state with that desired state.

GitOps differs from a [CI/CD Pipeline](#). A CI/CD Pipeline builds, tests, packages, or delivers changes. A GitOps mechanism observes approved desired state and reconciles an operational environment with that state. A CI/CD Pipeline and a GitOps mechanism can participate in the same Workflow.

GitOps does not require a particular:

- Git-hosting service
- Repository provider
- Branch strategy
- Reconciliation tool
- Deployment platform
- Infrastructure provider

Example

An approved change to Declarative Configuration is merged into a designated Git branch. A GitOps mechanism detects the new repository state, retrieves the change, compares the declared desired state with the actual deployment state, applies the required changes, and reports the reconciliation result.

From:

<https://wiki.didosolutions.com/> - **Dido Solutions, Inc Wiki**

Permanent link:

https://wiki.didosolutions.com/dido/99_annexes/annex-b-terms-and-definitions/g/gitops

Last update: **2026/07/30 13:54**

