Git

Go up to Terms and Definitions

Discussion

Git is a distributed Version Control System used to record, organize, compare, and exchange changes to files and directories maintained in a repository.

A Git repository contains identifiable revisions of managed Artifacts and preserves relationships among those revisions.

Git represents recorded repository states as commits. Git workflows commonly use:

- Repositories
- Commits
- Branches
- Tags
- References
- Working trees
- Staging
- Merging
- Rebasing
- Fetching
- Pulling
- Pushing
- Cloning

Git records repository states and the relationships among commits. This history enables users and automated processes to:

- Identify changes
- Compare repository states
- Trace the ancestry of a change
- Retrieve an identified repository state
- Restore or reuse an identified repository state
- Reproduce managed content associated with an identified commit

Within Crucible documentation, Git provides repository and change-history capabilities used by:

- Git-based workflows
- GitOps practices
- CI/CD Pipelines
- Configuration management
- Automated build operations
- Automated deployment operations
- Automated validation operations
- Provenance

- Reproducibility
- Traceability

Definition

distributed Version Control System that records identifiable repository states and preserves the relationships among those states

Source

Adapted from:

- Git Project documentation
- Version Control System

Note

The name **Git** is not an acronym and should not be written as **GIT**.

Git is a type of Version Control System.

Git is distinct from repository-hosting services such as GitHub, GitLab, and Bitbucket. Those services can host and manage Git repositories, but they are not Git itself.

Git does not require a single permanent central repository. A project can designate one repository as an authoritative or shared repository through policy and workflow conventions.

A Git-based workflow uses Git repository states and operations to coordinate change. The existence of a Git repository alone does not define:

- The workflow
- Approval rules
- Branch strategy
- Merge strategy
- Automation
- Build behavior
- Deployment behavior

Example

A contributor commits a Declarative Configuration change to a Git branch. A review process approves and merges the change into a designated branch. A CI/CD Pipeline identifies the resulting commit and initiates automated build, validation, and deployment operations associated with that repository

state.