Git Workshop

Jaime Canizales

City University of New York

jaime.canizales@hunter.cuny.edu

January 14, 2025

Overview

What is Git?

What is Git?

Git is a distributed version control system used for tracking changes in source code during software development. It allows multiple developers to collaborate on projects by managing code versions, enabling branching, merging, and keeping a history of changes.

Key Features of Git

- `Version Control` - Tracks changes and allows you to revert to previous versions.
- `Distributed System` - Every developer has a full copy of the repository, increasing reliability.
- `Branching and Merging` - Developers can create branches for new features or bug fixes and merge them back into the main codebase.
- `Collaboration` - Teams can collaborate without overwriting each other's work.
- `Speed and Efficiency` - Git handles large projects efficiently.

What is a Git Repo?

What is a Git Repo?

A Git repository is a storage space where all the files, history of changes, and version control data for a project are kept. It tracks the complete history of a project, allowing developers to manage and collaborate on code efficiently.

You can identify a git repo by the presence of a .git repo. If you delete the .git repo, the current repo is no longer a git repo.

Draw Software Architecture objectf git.

Draw graph representation of git repo(edges are changes).

Types of Git Repositories

1. `Local Repository` - Stored on your own computer, containing the full project history and working files.
2. `Remote Repository` - Hosted on a server (e.g., GitHub, GitLab, Bitbucket) for collaboration with other developers.

Structure of Git Repositories

- `Working Directory:` The actual files you are currently working on.
- `Staging Area (Index):` A space where changes are prepared before committing.
- `.git Directory:` Contains all the metadata, configurations, and object database (commit history).
- `Commits:` Snapshots of changes in the project over time.

Installing Git

- Git Install Link

Common Git Commands(starter)

- `git init:` Initializes a new Git repository.
- `git remote add origin <link>:` associate a local repo witha remote repo.
- `git clone:` Creates a copy of an existing repository.
- `git add:` Stages changes for a commit.
- `git rm:` removes staged changes
- `git commit:` Records a snapshot of all the tracked files in your directory(between commits only the set of changes is saved)

Common Git Commands(collaborative)

- `git fetch:*` That downloads the latest changes from a remote repository without merging them into your current branch.
- `git push:*` Uploads commits to a remote repository.
- `git pull:*` Fetches and integrates changes from a remote repository.
- `git branch:*` Lists, creates, or deletes branches.(a pointer to tree)
- `git checkout:*` Used to switch branches or restore files in your working directory.

Common Git Commands(combining)

- `git merge:*` Combine branches(in your checkout branch, creates another commit that adds all changes from the commit specified in argument)(perserves history)
- `git rebase:*` Used to move or combine a sequence of commits to a new base commit. It creates a cleaner, linear project history by integrating changes from one branch onto another. Rewrites history to make it look like the changes happened sequentially.
- `git cherry-pick:` That allows you to apply a specific commit from one branch to another without merging the entire branch.

Common Git Commands(show info)

- `git log:` Show all commits on terminal.
- `git status:` Shows the status of changes. (think of staging area)
- `git tag:` Used to mark specific points in a repository's history—typically for releases (e.g., v1.0, v2.0). Tags are like bookmarks for important commits.
- `git describe:` Generates a human-readable name for the current commit based on the nearest tag, the number of commits since that tag, and the abbreviated commit hash.

What is Github?

What is Github?

GitHub is a cloud-based platform for version control and collaborative software development. It uses Git for tracking changes in code but adds powerful tools for collaboration, project management, and deployment. Git is often used with platforms like GitHub, GitLab, and Bitbucket for collaborative development.

HEAD

- **Defintion:** The symbolic name for the currently checked out commit.
- Normally HEAD points to a branch name.
- Detached head state just means head points to commit instead of a branch.
- **Relative Refs:** $\sim x$: go up x edges, $\wedge x$: go to xth parents

Other Common terms in git

- **Remote Branch:** Is a branch that exist locally, and reflects the state of the remote repository.(upstream refers to this)
- **origin:** Common name used to refer to the remote repository.
-

.gitignore

.gitignore

The .gitignore file tells Git which files and directories to ignore in a project. It prevents unnecessary or sensitive files from being tracked in version control.**Show Example**

ssh keys

Problem Statement

SSH keys are a secure way to authenticate with GitHub without using a username and password every time you interact with your repositories. They use a public-key cryptography system to establish a secure connection between your computer and GitHub.

SETUP LINK

how to deal with more than one git account on one computer

1. git clone <repo>
2. git remote set-url origin git@github.com-PRTEST:JimsPrTest/git-workshop.git
3. git config user.name "your_user_name"
4. git config user.email "your_email"
5. git push origin –tags

- Run all commands from above from terminal
- Vscode with Git Example
- Forking and Pull Request Example(consider .ssh/config file)

Learning Github Resources Links

- Official Git Documentation
- GitHub Skills
- Git Basics Tutorial
- Interactive Git Branching Tutorial
- Git and GitHub for Beginners - Crash Course