Git and GitHub – Plain English Explanation

What Is Git?

Git is the tool on your computer that tracks changes to files. It works even without internet and lets you commit, branch, merge, and undo mistakes. Think of Git as Time Machine for your project.

What Is GitHub?

GitHub is a website (a cloud platform) that stores Git repositories online. It adds collaboration tools like pull requests, permissions, issue tracking, and CI/CD (GitHub Actions).

One-Sentence Difference

Git is the local change-tracking tool.

GitHub is the online place where Git repositories live and teams collaborate.

Analogy

Git = Microsoft Word installed on your computer.

GitHub = OneDrive/SharePoint where you upload documents to collaborate.

Git is the editor. GitHub is the shared drive.

How They Work Together

1. You edit files locally using Git.

2. You commit changes.

3. You push them to GitHub.

4. GitHub stores them, and CI/CD pipelines run.

Git is the engine; GitHub is the garage, motorway, and pit crew.

In Terraform Context

Git tracks your Terraform code changes locally.

GitHub stores your repo and runs GitHub Actions (Plan/Apply).

Together they provide safety and control

The Big Picture

GitHub is simply a safe way to manage changes to files and code. Think of it like a shared Word document where multiple people can make changes — but without overwriting each other or losing history.

The Main Branch

The 'main' branch is the official version of your project. It is the version that CI/CD tools use and the one considered stable and correct.

What Is a Branch?

A branch is a copy of your project where you can make changes without affecting the main branch. It is like creating a duplicate of a document so you can edit freely.

What Do You Do on a Branch?

You add features, edit files, test things, and experiment safely. Anything you do on your branch does not impact the main version.

What Is a Pull Request?

A pull request (PR) is a request to merge the changes from your branch into the main branch. It allows review, discussion, and automated checks before merging changes.

What Does 'Merge' Mean?

Merging takes the changes from your branch and applies them to the main branch, making them part of the official version.

Why Use PRs with Terraform + GitHub Actions?

PRs run Terraform plan, allowing you to preview infrastructure changes before applying them. Merging the PR triggers Terraform apply, safely updating cloud resources.

Why Not Just Commit to Main?

Direct commits to main risk breaking deployments, causing unintended changes, or introducing errors. PRs help enforce review, safety, and predictability.

The Simplest GitHub Workflow

1. Create a branch
2. Make your changes
3. Commit and push
4. Open a PR
5. Review and merge
6. Delete the branch afterward

What Your Lab Teaches

Create a branch, push changes, open a PR, run plan, merge, run apply, and verify results. This reflects real-world DevOps workflows.