GIT : A distributed version control system

Istapriya Jagravi
21ME10039

Contents

Introduction

- Version control (or revision control, or source control) is all about managing multiple versions of documents, programs, web sites, etc.
- Git is a Version Control System (VCS) designed to make it easier to have multiple versions of a code base, sometimes across multiple developers or teams
- It allows you to see changes you make to your code and easily revert them.

Git-ing To The Point

What exactly does Git do for us?

Basic commands

- Git config : Define author name to be used for all commits in current repo. Devs commonly use —global flag to set config options for current user.
- Git init : Create empty Git repo in specified directory. Run with no arguments to initialize the current directory as a git repository
- Git status : List which files are staged, unstaged, and untracked.
- Git add : Stage all changes in for the next commit. Use a . in last to add all the files.
- Git commit -m “message” : Commit the staged snapshot, but instead of launching a text editor, use as the commit message.
- Git log : Display the entire commit history using the default format. For customization see additional options.

Branching

- Branches are extremely essential part of git.Think of branching as alternative timelines of a project
- They enable us to create separate contexts where we can try new things, or even work on multiple ideas in parallel
- If we make changes on one branch, they do not impact the other branches (unless we merge the changes)
- Command : git branch <branch-name>

Terminologies :

- Master branch: Branch from which we started our repository.
- Head : Points to a particular branch reference.

Few Advanced Commands

Git switch

- Allows to switch current HEAD branch.
- Provides a simpler alternative to the classic "checkout" command.
- Command : git switch <branch-name>
- "-c" parameter can be used along with it, if we want to create a local branch in one go.

Git stash

- The git stash command enables you to switch branches without committing the current branch.
- Most common uses are git stash save and git stash pop.
- Acts as a mechanism to locally change version files without those versions being seen by other developers who share the same git repository.

Git diff

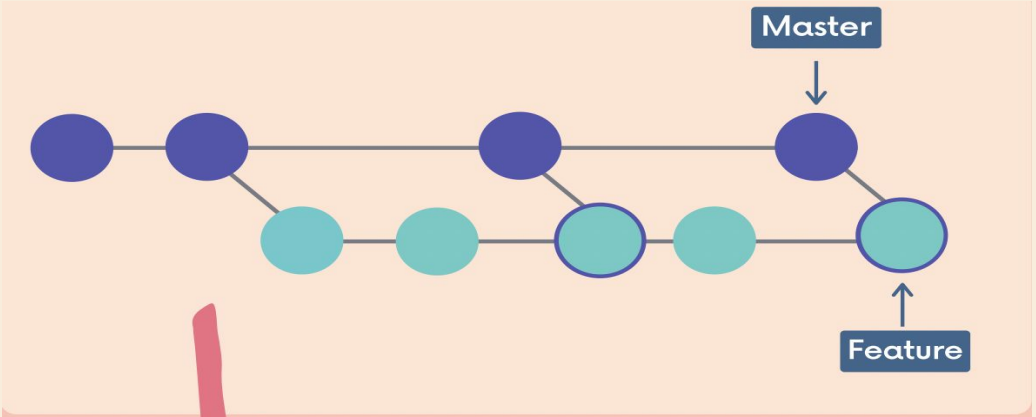
- Diff command is used in git to track the difference between the changes made on a file.
- It can be staged or unstaged.
- git diff- shows the changes between the Working Directory and the Staging Area
- git diff staged- shows the changes between the Working Directory and the last commit.

Git rebase : scariest command ?

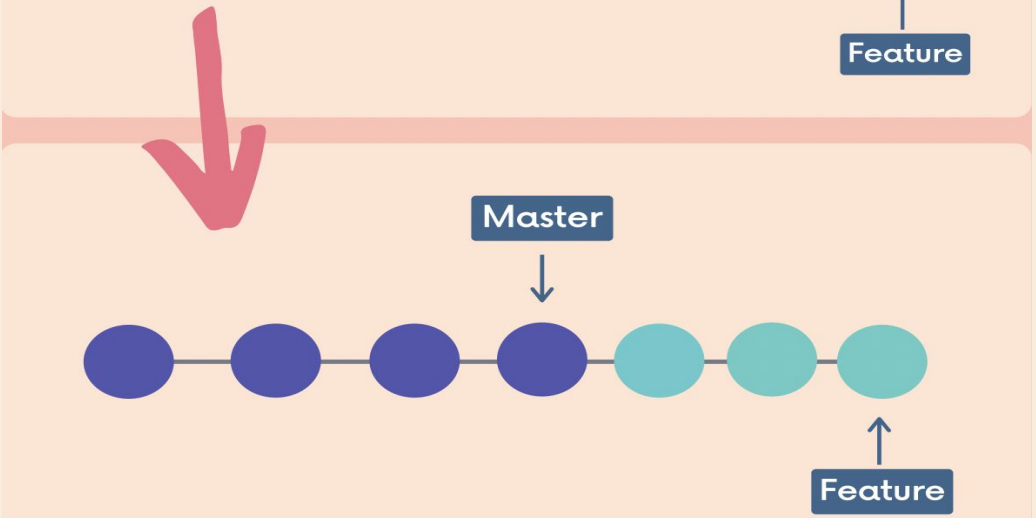
- Alternative to merge
- Rebasing is most useful and easily visualized in the context of a feature branching workflow.
- It is changing the base of your branch from one commit to another making it appear as if you'd created your branch from a different commit.
- It's very important to understand that even though the branch looks the same, it's composed of entirely new commits.

Rebase vs Merging

Merging :



Rebasing:



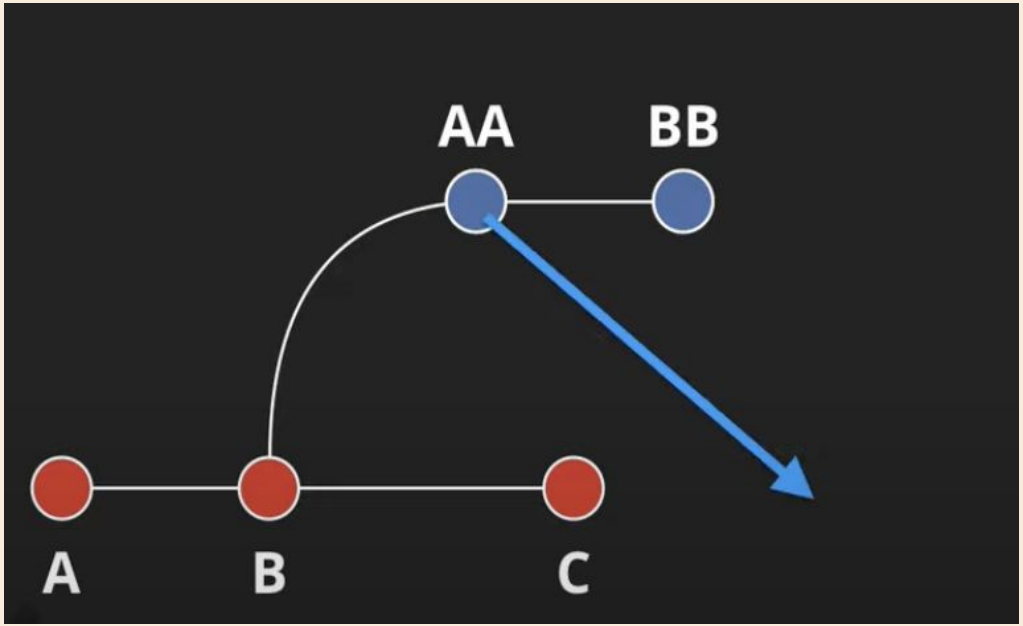
Git reflog

- Reflog is a mechanism to record when the tip of branches are updated.
- This command is to manage the information recorded in it.
- Every action you perform inside of Git where data is stored, you can find it inside of the reflog

Git cherry-pick

- Git allows you to integrate selected, individual commits from any branch into your current HEAD branch, through cherry pick command.
- The main motive of a cherry-pick is to apply the changes introduced by some existing commit.
- Cherry-pick is a useful tool, but always it is not a good option. It can cause duplicate commits and some other scenarios where other merges are preferred instead of cherry-picking.

Git cherry-pick



Git bisect

- Helps track down the commit where the code works and the commit where it does not.
- Hence, tracks down the commit that introduced the bug into the code.
- Finds the faulty commit by performing a binary search on the commits to reduce the time taken to find the faulty commit.

And that brings us to the end.....

THANK YOU !