GIT

By Rahul Kumar Ojha

Introduction

Git is an Open Source Distributed Version Control System.
Now that’s a lot of words to define Git.

Let me break it down and explain the wording:

- **Control System:**
    a. This basically means that Git is a content tracker.

Introduction Cont….

- **Version Control System:**

  a. The code which is stored in Git keeps changing as more code is added.

  b. Also, many developers can add code in parallel. So Version Control System helps in handling this by maintaining a history of what changes have happened.

Introduction Cont….

- **Distributed Version Control System:**

  a. Git has a remote repository which is stored in a server and a local repository which is stored in the computer of each developer.

  b. This means that the code is not just stored in a central server, but the full copy of the code is present in all the developers' computers.

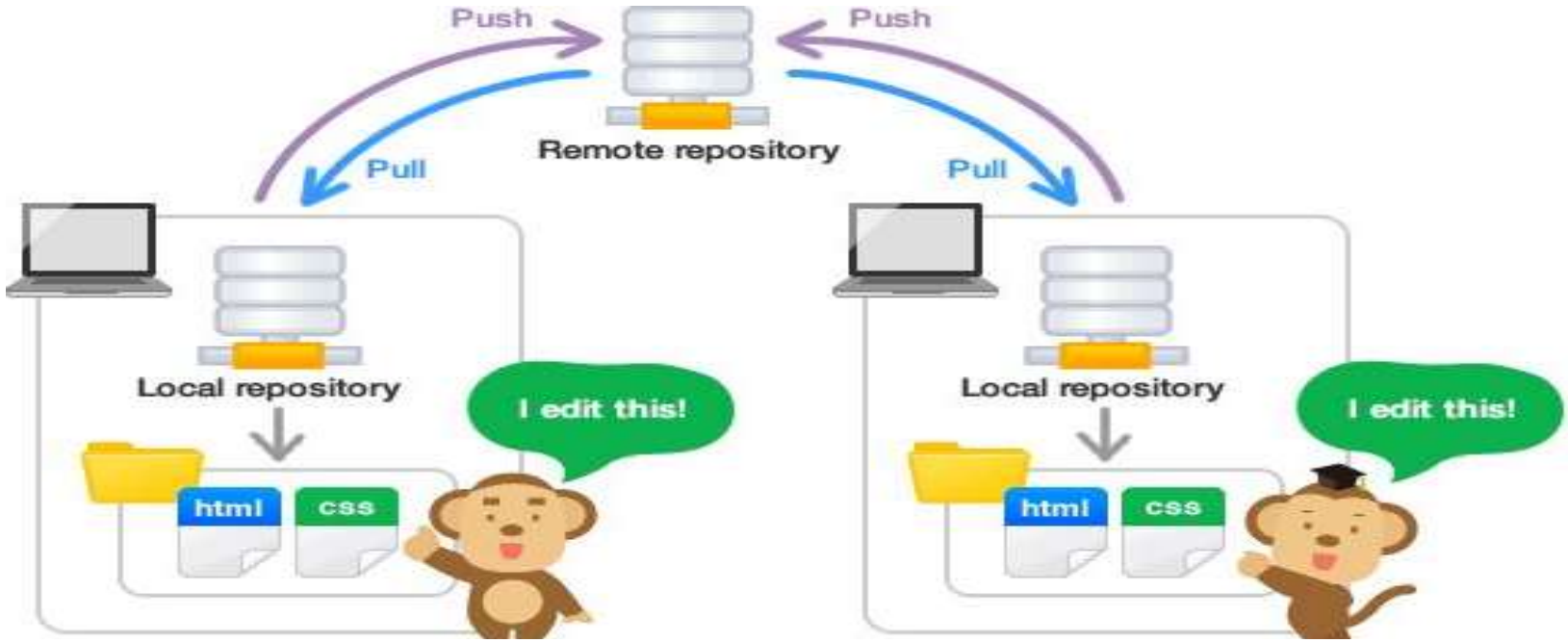
  c. Git is a Distributed Version Control System since the code is present in every developer's computer.

Installation

- `Ubuntu: sudo apt install git`

RePository

A repository (usually abbreviated to “repo”) is a location where all the files for a particular project are stored.



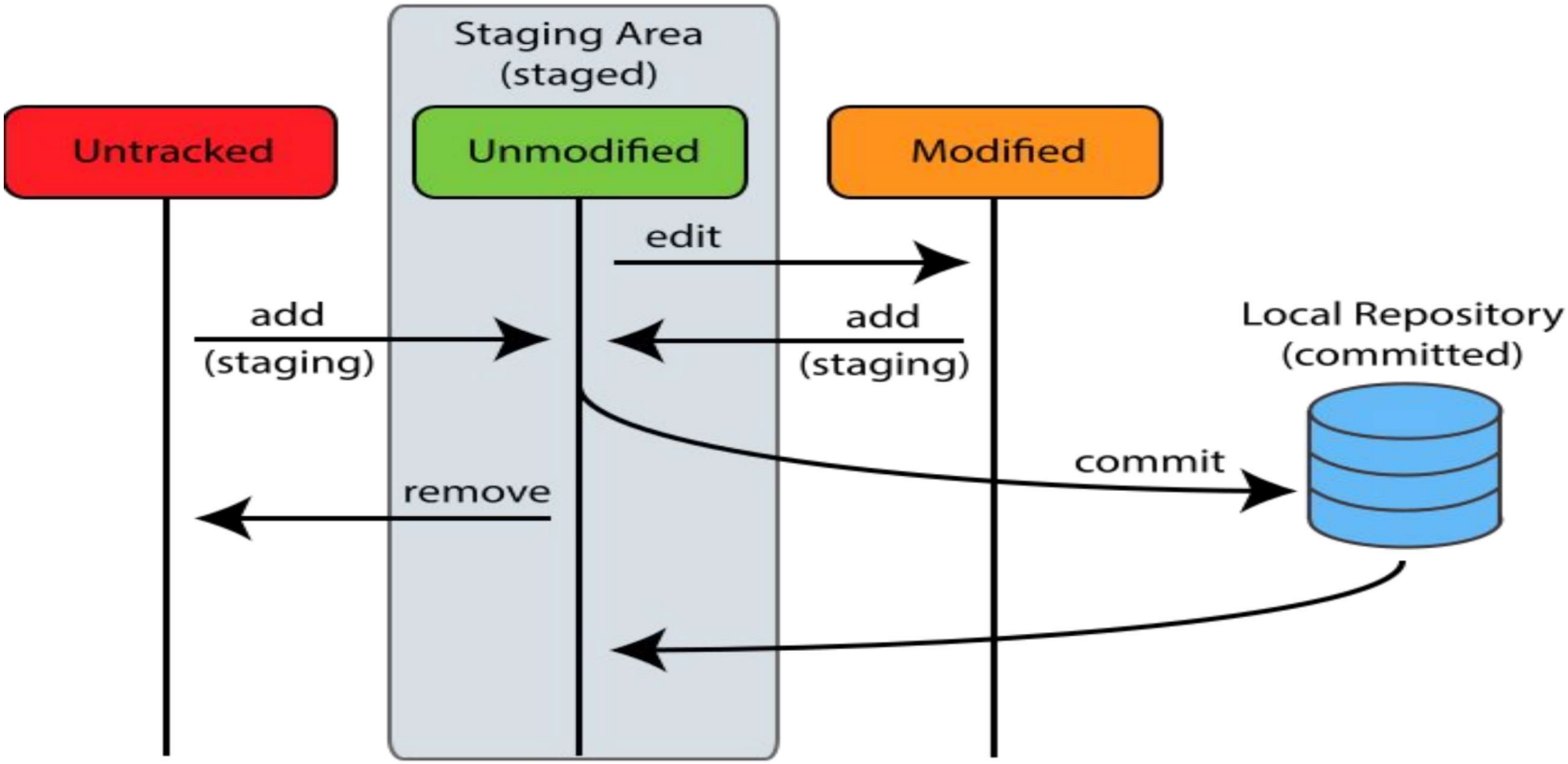
Initializing repository

- `git init`

Know status of your changes

- `git status`

The Three Stages Of Git



Stage your changes

- `git add`

COMMIT YOUR CHANGES

- `git commit -m “My first commit”`
- SHA
- Git config

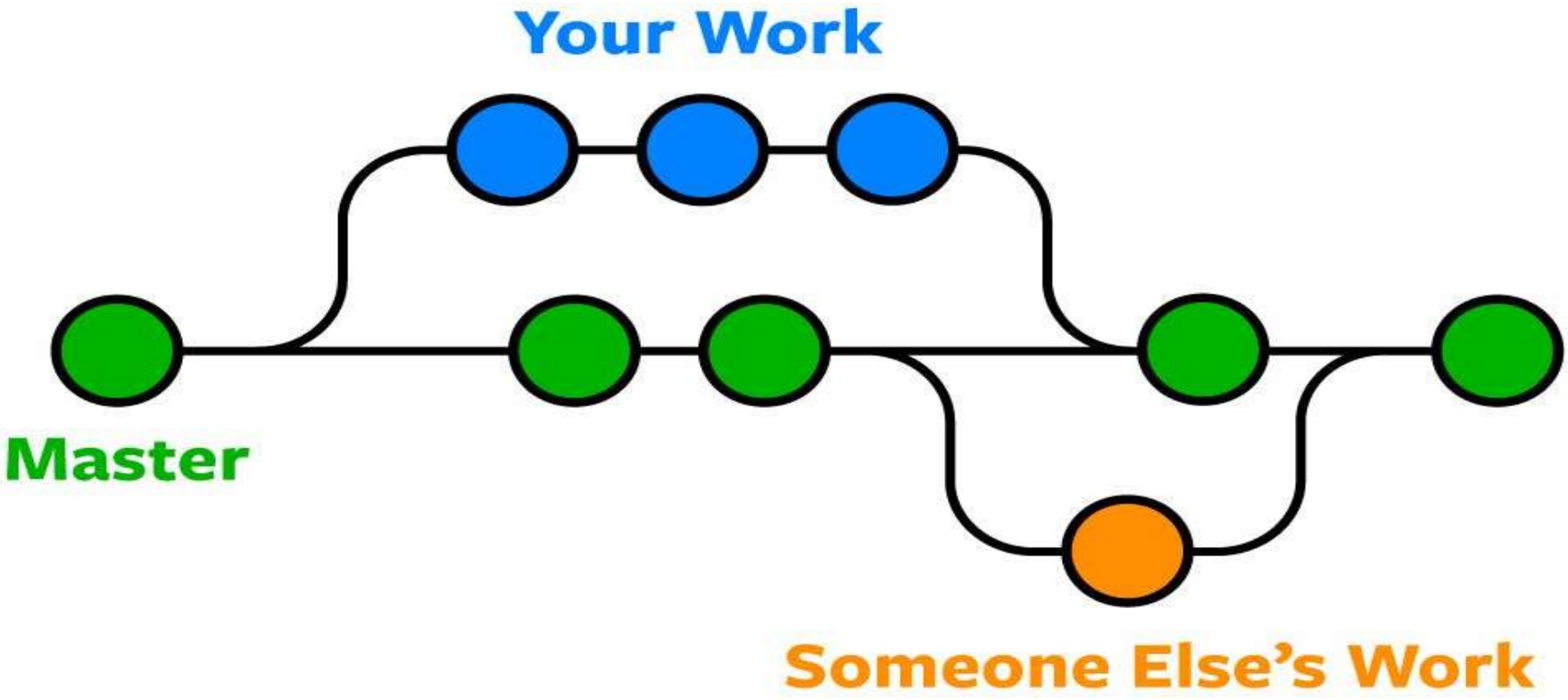
Lets Try Out Some more Commands

- `git reset`
- `git diff`
- `git checkout`
- `git reset --soft HEAD~`
- `git reset --hard HEAD~`
- `git stash`

Git branch

- A branch represents an independent line of development.
- You can think of them as a way to request a brand new working directory, staging area, and project history.
- New commits are recorded in the history for the current branch.

GIT BRANCH: LETS VISUALIZE IT



GIT BRANCH COMMANDS

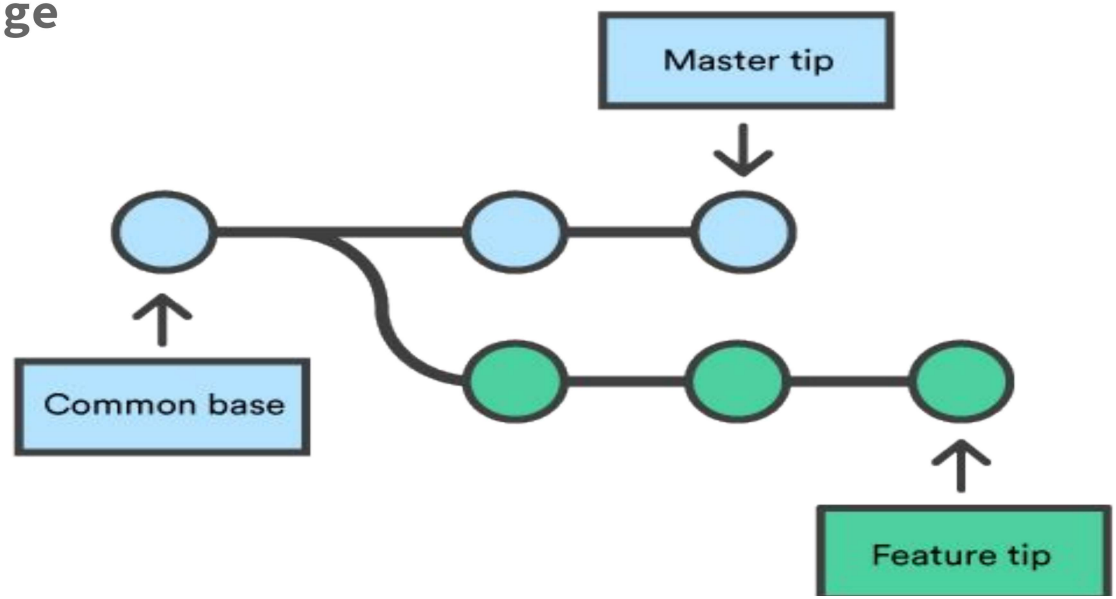
- `git branch`
- `git branch -m “new_branch_name”`
- `git checkout branch_name`
- `git checkout -b branch_name`
- `git branch -D branch_name`
- `git branch -d branch_name`

GIT MERGE

- Merging is Git's way of putting a forked history back together again.
- The git merge command lets you take the independent lines of development created by git branch and integrate them into a single branch.

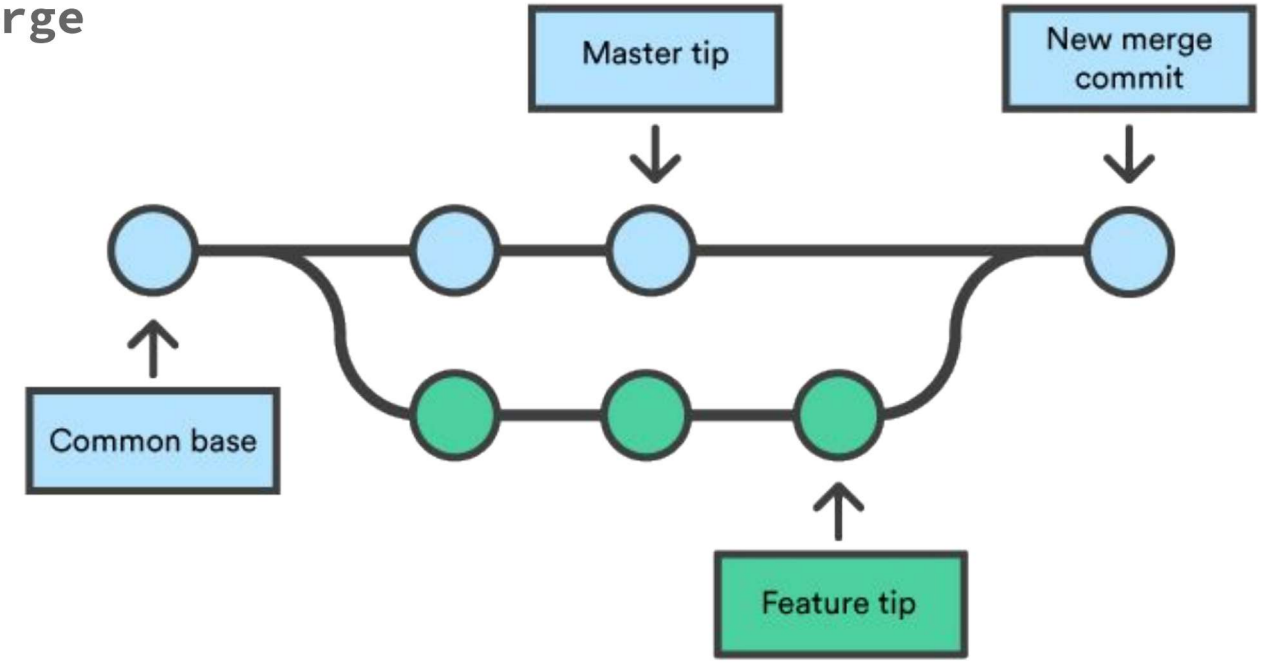
GIT MERGE CONT...

Before Merge



GIT MERGE CONT...

After Merge



Merge Conflicts

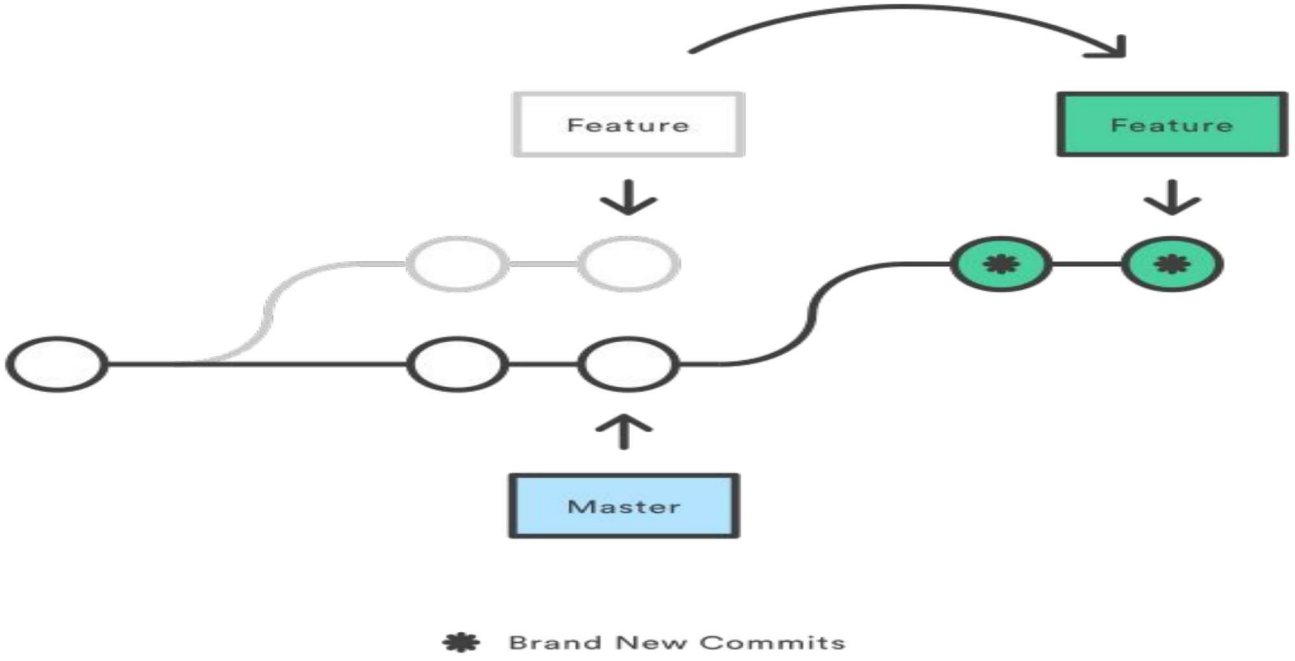
- Conflicts generally arise when two people have changed the same lines in a file, or if one developer deleted a file while another developer was modifying it.
- In these cases, Git cannot automatically determine what is correct.
- Conflicts only affect the developer conducting the merge, the rest of the team is unaware of the conflict. Git will mark the file as being conflicted and halt the merging process. It is then the developers' responsibility to resolve the conflict.

REBASE

- Rebasing is the process of moving or combining a sequence of commits to a new base commit.
- Rebasing is most useful and easily visualized in the context of a feature branching workflow.

REBASE CONT...

The general process can be visualized as the following:



MERGE VS REBASE

- **rebase** also solves the same problem as git **merge**
- git **merge** creates a new “merge commit” in the feature branch that ties together the histories of both branches.
- While git **rebase** moves the entire feature branch to begin on the tip of the master branch, effectively incorporating all of the new commits in master.
- Rebase does not use merge commit, rebasing re-writes the project history by creating brand new commits for each commit in the original branch.

MERGE VS REBASE CONT...

- If you want to update public branch you should use git **merge.**
- If you want to update your feature branch you should use git **rebase**

Git repository hosting services

- Github
- Bitbucket
- Gitlab

Lets Create a REPO on GITHUB

- Clone Repo
- Add remote to already existing repo.

Git PUSH & PULL

- `git push origin branch_name`
- `Git pull origin branch_name`

Lets CREATE a PULL Request

THANKYOU