

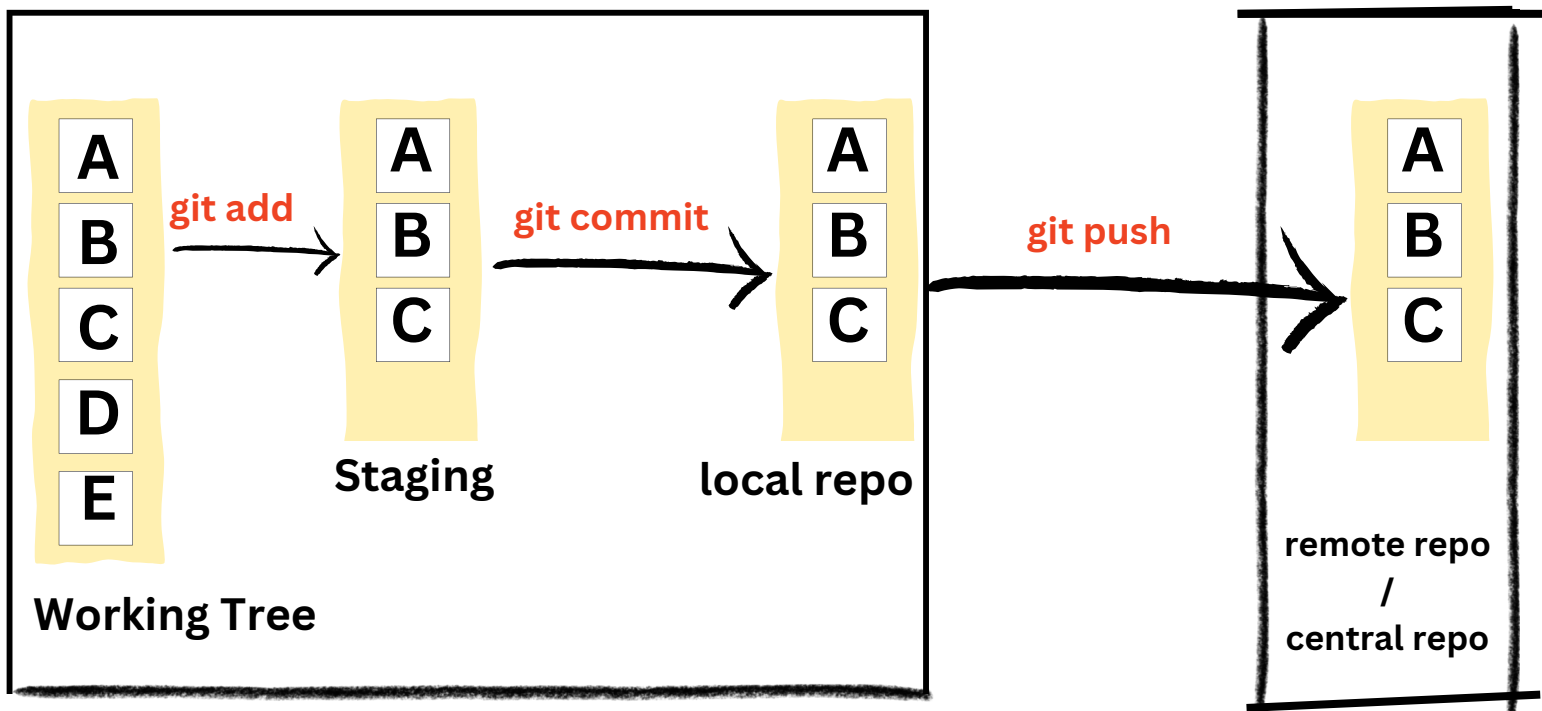


Source code management tools is [GitHub](#)

## Advantage of source code

- Integrate developer code
- work hierarchy

## GitHub Architecture



- **git init** (working tree)
- **git add .** (add file /staging)
- **git commit -m “first commit”** (local repo)
- **git branch main** (main branch)
- **git remote add origin URL**
- **git push** (remote/central repo)
- **git clone URL** (clone the repository when your doing first time)

how to remove file **ex A.txt** ?

- **git rm A.txt**

if someone is changed code in remote/central repo how do you take changes in local repo?

- **git pull**

## Overview to create branch

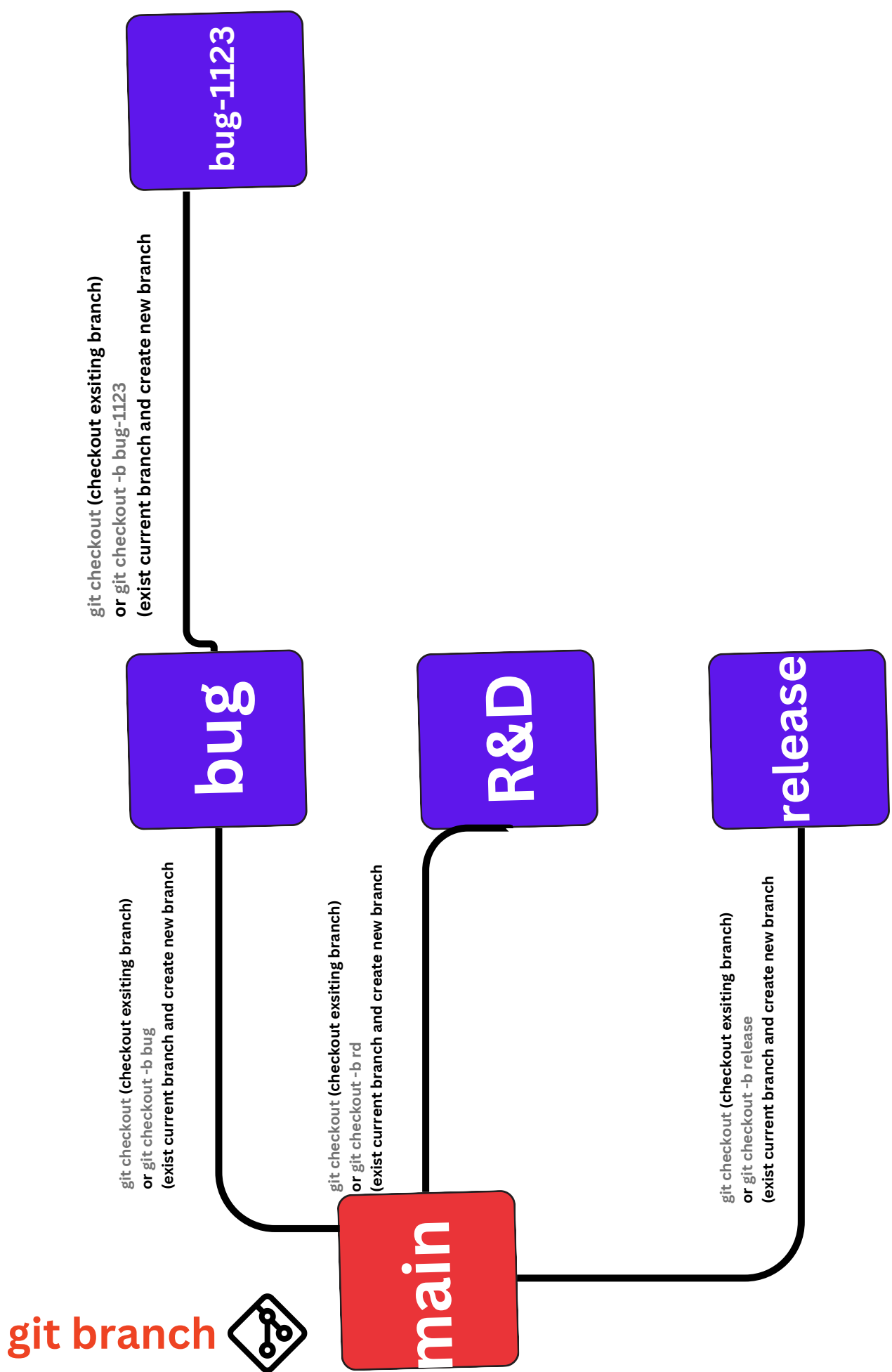
**git checkout** (checkout current branch)

**git checkout -b branch\_name**

(checkout current branch and create new branch)

**git push -u origin branch\_name**

whatever you branch created push by using below command  
git push -u origin bug-1123



## pull conflict

pull conflict occur when **developer 1** working ex line **number 2** and **developer 2** also working **line number 2** when **developer 2** pull code he get conflict you need to resolve manually

- **note in companies we do first git pull and then git push to avoid conflicts**

## merge conflict

when you merge two branches in Git, and both branches have changes in the same line of code in a file, Git cannot automatically resolve which version of the code to keep. This results in a merge conflict.

so, conflicts occur in 2 places

when we doing **pull , merge request**

## UNDO from GitHub

go your github you will see one id like this ex: **888938e**

git revert **888938e**  It will undo all the changes what you done

## UNDO from local repository

**git reset --soft HEAD~1**

 **1 → reset latest one if it is 2 reset 2nd latest**

**soft reset means it will UNDO from local repository but in working tree your files/code is still present**

**git reset --hard HEAD~1**

 **1 → reset latest one if it is 2 reset 2nd latest**

**hard reset means it will UNDO from local repository and also removes your files/code from working tree completely**

# Git forking

A fork is a copy of a repository. Forking a repository allows you to freely experiment with changes without affecting the original project.

## What is a Collaborator in GitHub?

A collaborator is someone who has been granted direct access to a GitHub repository by the repository owner. Unlike forking (where you work on a separate copy), collaborators can push changes directly to the repository.



**git pull**

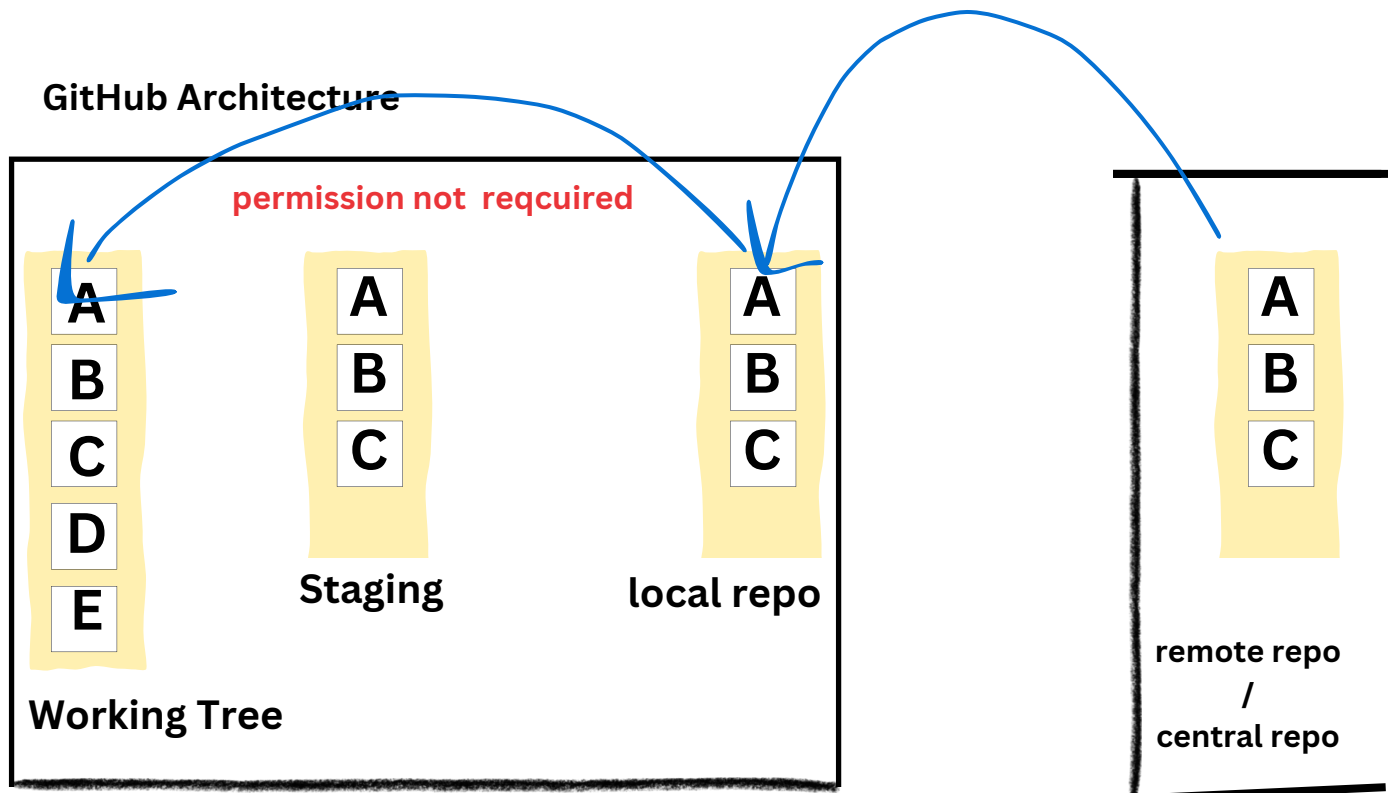


**git fetch**



**git pull**

when your doing git pull you will get code in local repo and then automatically code merge to your working tree as shown above

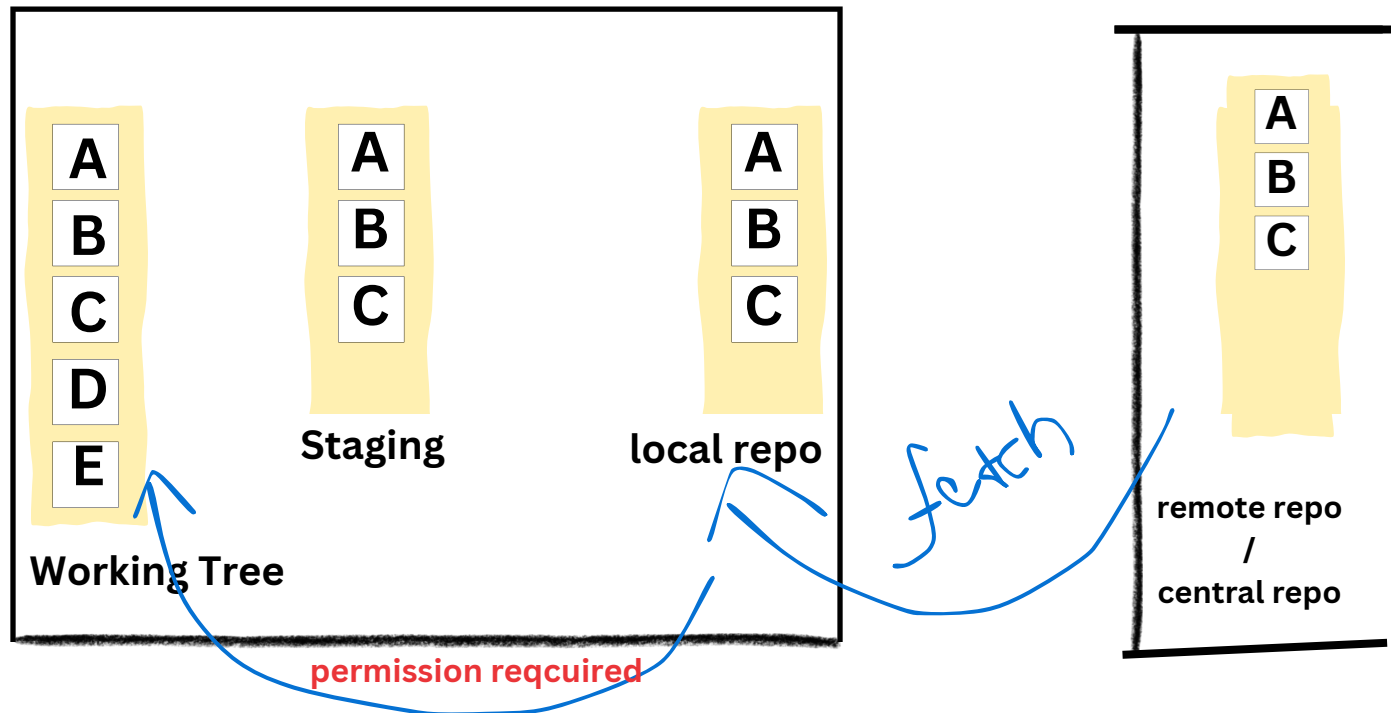


**git pull**



# git fetch

## GitHub Architecture



when your doing git fetch you will get code in local repo and then required permission to get code merge to your working tree as shown above

**git fetch**

**git merge**

## git stash

-> git stash is a command used in Git to temporarily save changes that are not yet committed to work on something else without losing your modifications.

-> when you run git stash, all your modified but unstaged or staged changes are temporarily stored in a hidden area

**example:**

**git stash** - to store changes in temp area

**git stash apply** - to get the changes from temp area back

## git cherrypic:

-> git cherry-pick is used to apply a specific commit from one branch to another without merging the entire branch.

Interview Question?

You have done today 5 commits, but you want to merge only 3rd commit to particular branch, how will you do?

Answer: git cherrypic

Note: git merge will merge all the commit to a particular branch