Version Control Systems

Dr Kawal Jeet

Version Control Systems

- Category of software tools that help a software team manage changes to source code over time.
- Keeps track of every modification to the code in a special kind of database.
- If a mistake is made, developers can turn back the clock and compare earlier versions of the code to help fix the mistake while minimizing disruption to all team members.
- It protects source code from both catastrophe and the casual degradation of human error.

Version Control Systems

- Tracks every individual change by each contributor and helps preventing concurrent work from conflicting.
- Changes made in one part of the software can be incompatible with those made by another developer working at the same time.
- Any change can introduce new bugs on its own and new software can't be trusted until it's tested.
- CVS, SVN, GIT etc.

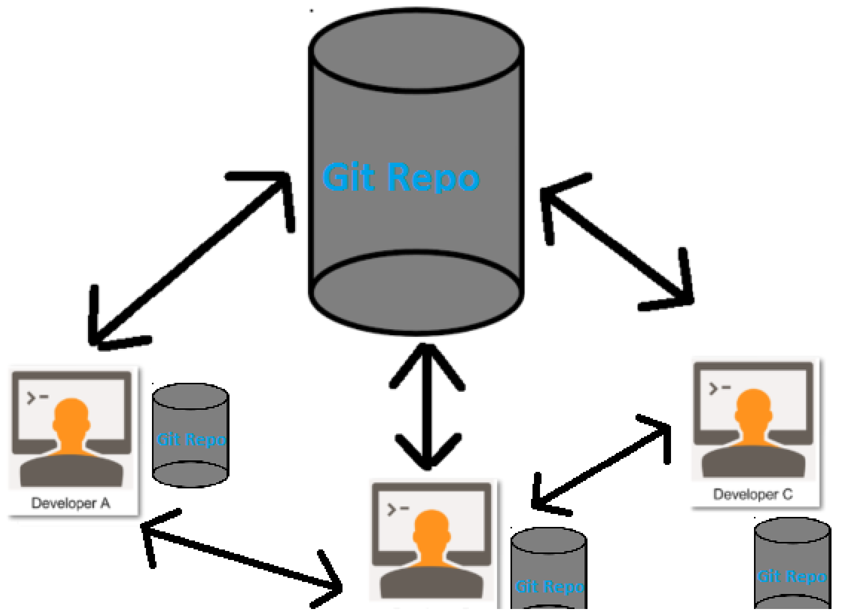
Applications of VCS

Anything you want continuously changes and improves

- Source code
- Website contents
- Documentation
- Automated testing
- Book

Distributed Version Control Systems

- Each user has local repository that can be synchronized with central one.
- Open source software
- Has a vibrant community.
- Could be successfully used for large and small projects.



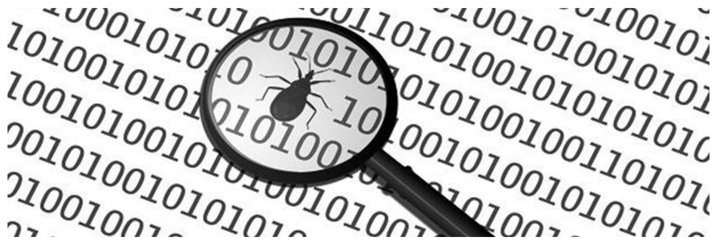
Git: Most used VCS

- Git is software that runs locally.
- Online hosts - GitHub or Bitbucket store a copy of the files and their revision history.
- Could access via a command line, or a desktop app that has a GUI or IDE eclipse (Sourcetree, GitHub Desktop).
- A Git repository (or repo for short) contains all the project files and the entire revision history.
- It contains a .git subfolder, which contains all the Git metadata for tracking changes.

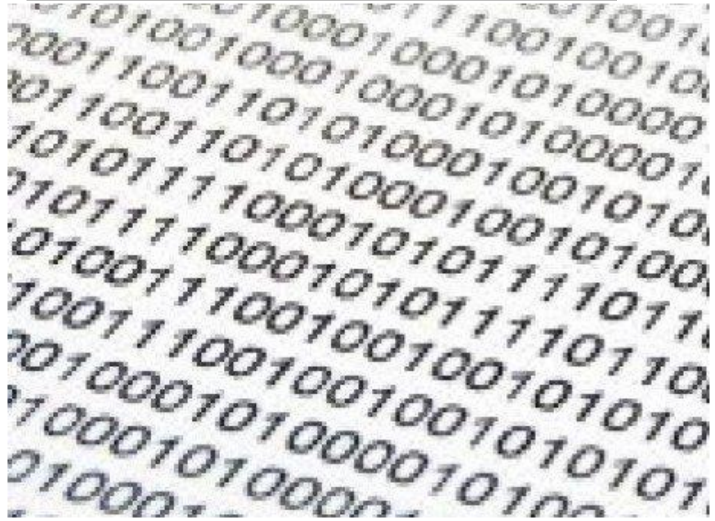
Manage versions

- Let we have an application with 50 files
- We want to improve
- Git manages version of project
- Each version is commit

Commit A with bug
(50 files)



Commit B without bug
(50 files with changes in 1 file)



Remote Repositories

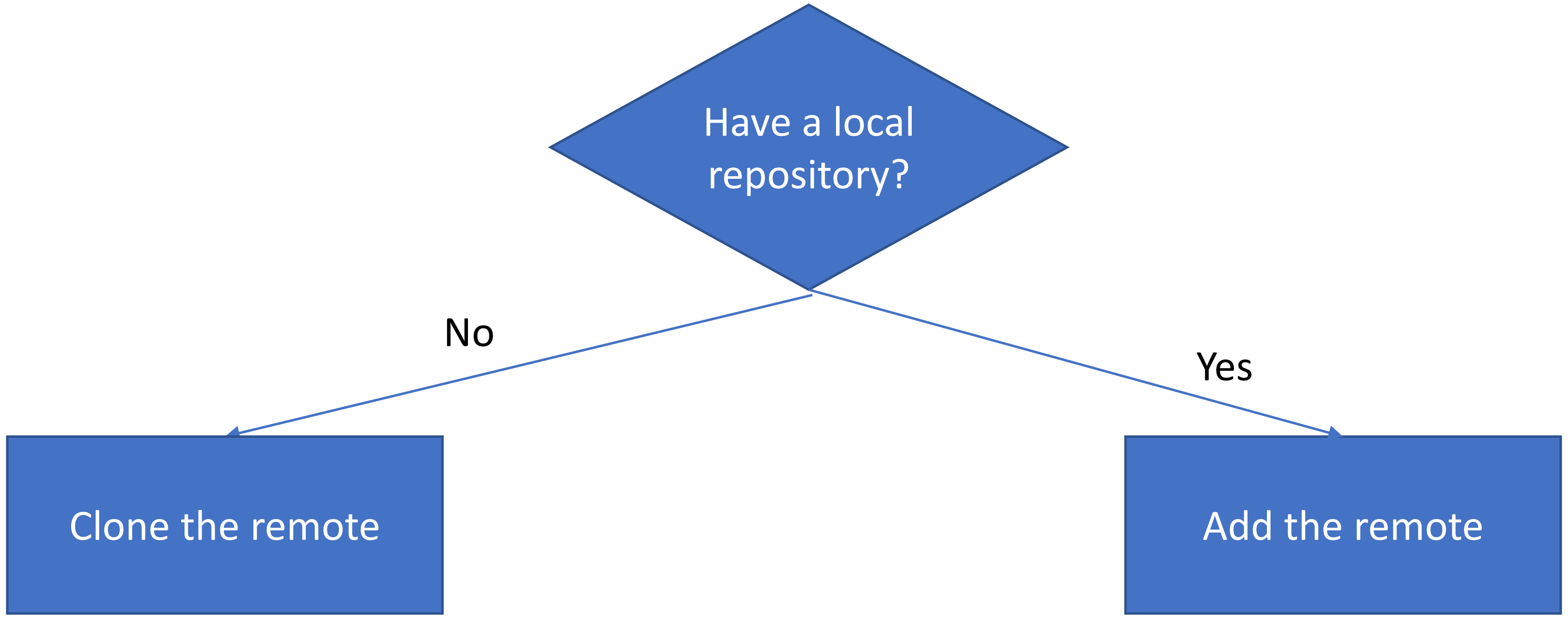
GitHub & Bitbucket

Gives you a centrally located place where you can upload your changes and download changes from others, letting you collaborate more easily with other developers.

Remote Repository

- A professionally managed repository that is hosted in a data center or in the cloud.
- It often acts as the central source of truth or official state of the project.
- Because nobody works with the repository locally, there is usually no working tree or staging area on a remote repository.
- The root directory of a remote repository is like the ".git" directory in a local repository.
- By convention, remote repository names end with ".git".

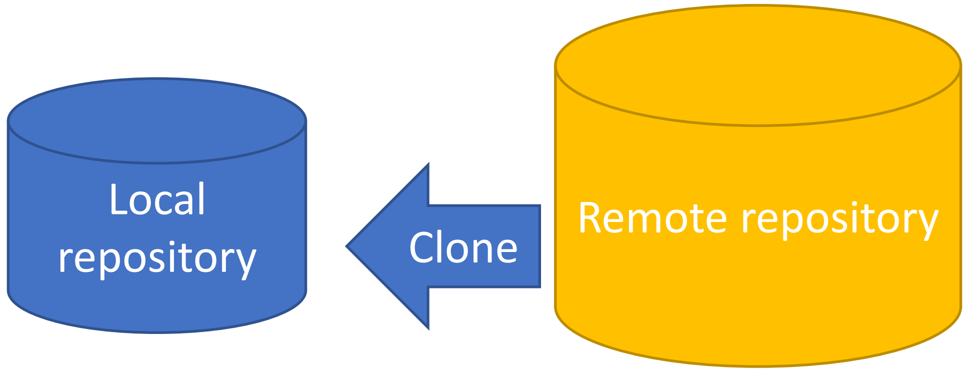
- **Version Control**



Cloning

- Suppose I am your project leader, and I created a remote repository on Git Hub for you (my team) to share and add features.

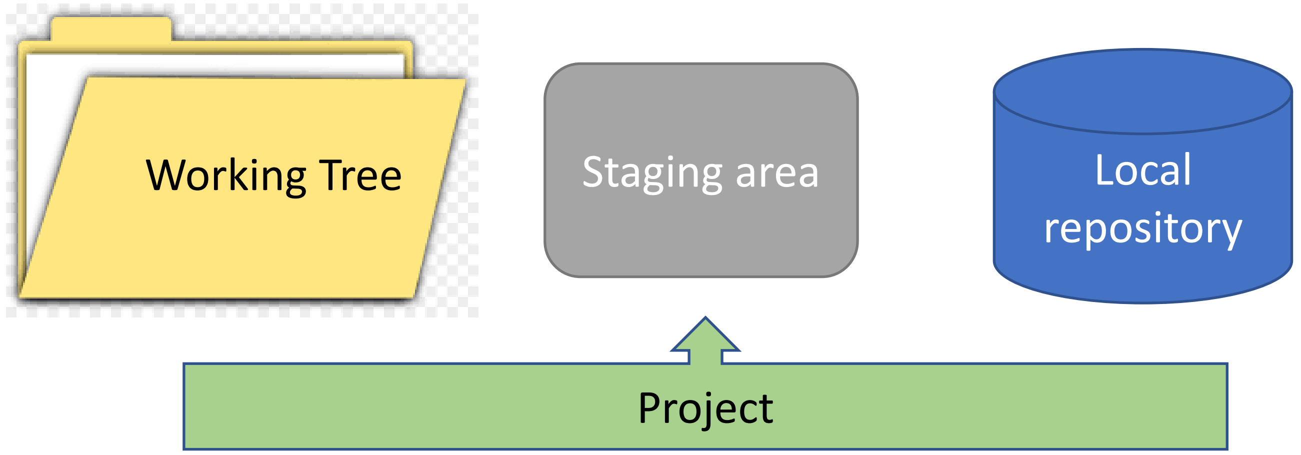
- Cloning a repository pulls down a full copy of all the repository data that GitHub has at that point in time, including all versions of every file and folder for the project.



Reference of remote repository is included in Local repository (Synchronization)

Locations in Git

- The working tree is the location on your computer that contains the directories and files of a single commit.
- The staging area contains a list of files that are planned to be included in the next commit that you make.
- The local repository contains all the commits that have been made for the project. These commits represent the version history of the project.
- Project directory: the working tree, staging area, and local repository are commonly all contained in a single directory on your local computer.



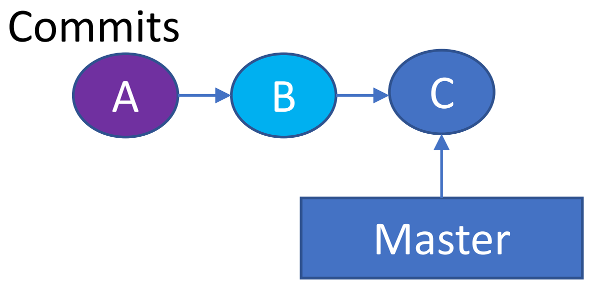
Commit

- It is called Snapshot of project at various times
- Each unique files are stores only once
- Collection of commit contains the history of the project
- Nearly Every Operation Is Local

Commit Contd…

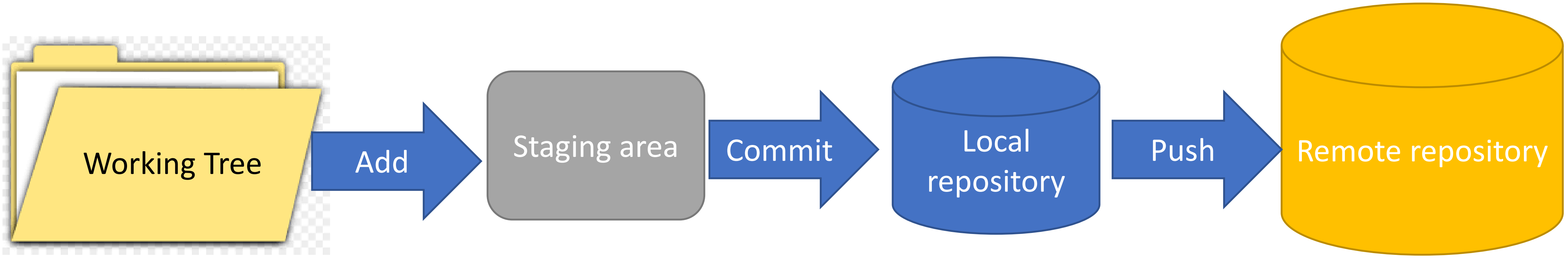
- All commits belong to branch
  - An independent line of development of branch

- By default there is single branch called master



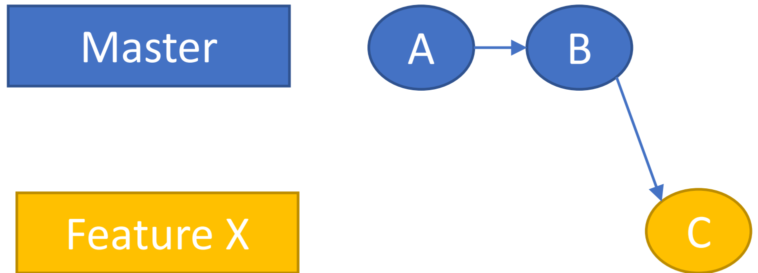
Push Command

- It adds new objects and references to the remote repository.
- Push add commit for a branch to remote repository
- A successful push synchronises remote and local repositories.
- They are copies of each other.
- Pushing is good backup of local repository.
- Other team members can see and synchronize their work.



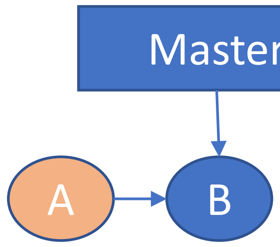
Creating branches

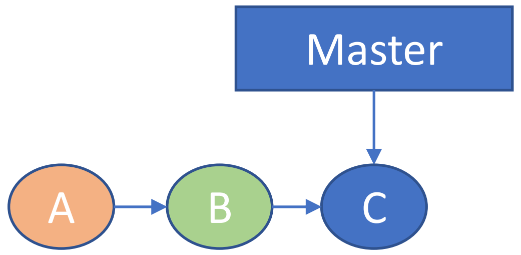
- Maintain stable project when you are making changes
  - Make a separate branch independent of master branch
- If you have an idea for a change, you can create a branch and test your idea.
- Branches enable experimentation.
- Later, you can throw out your branch or merge it into the official project.
- The master branch is not aware of feature branch
- For master branch latest commit is B whereas its C in actual.



Master branch

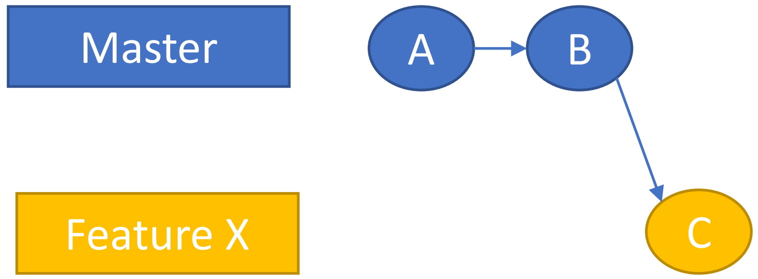
- Master is the default branch name in Git.
- Floating commit pointer.
- The "master" branch was created by Git automatically for us when we started the project.
- Every time you commit, it moves forward automatically.

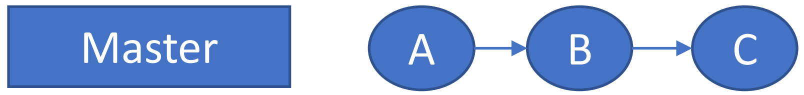




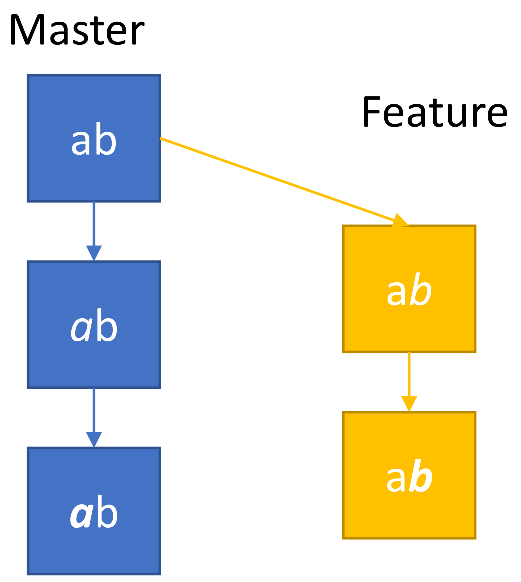
Git Branch Merge

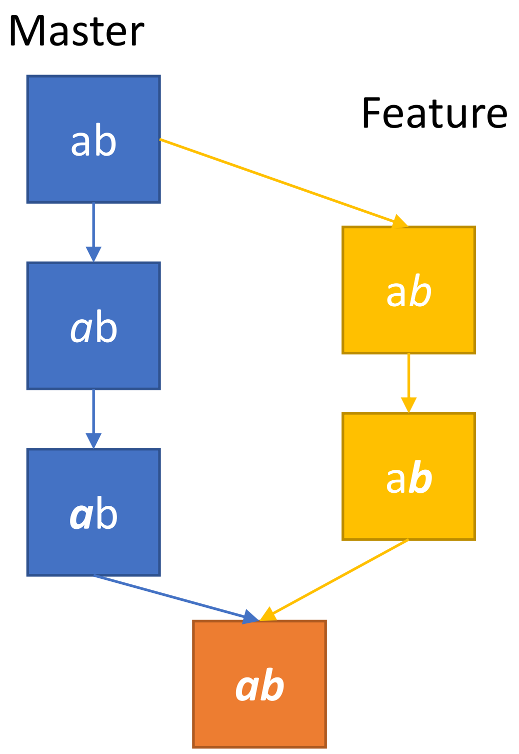
- Merging independent branches
- Before merge Master branch has no idea about feature C
- Team members can discuss, review and approve your changes
- Can require passing automated testing



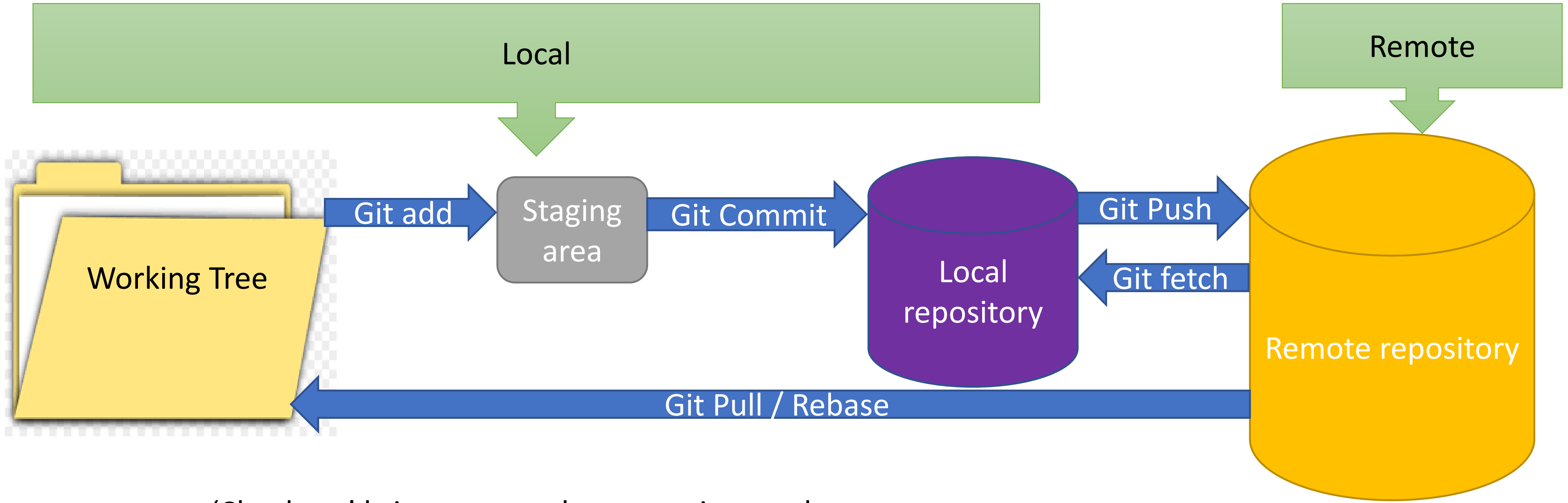


Merge





- It integrates changes from one branch to another.
- The merge commit represents every change that has occurred on feature since it branched from master



‘Checkout’ brings you to that commit snapshot stage.

Thank You