Git & GitHub

15-Oct-2019

AGENDA

01 Introduction to Version Control System

02 Version Control – What & Why

03 Types of Version Control System

04 Version Control tools

05 Git Installation and GitHub account creation

06 Git Features

07 Git Workflow

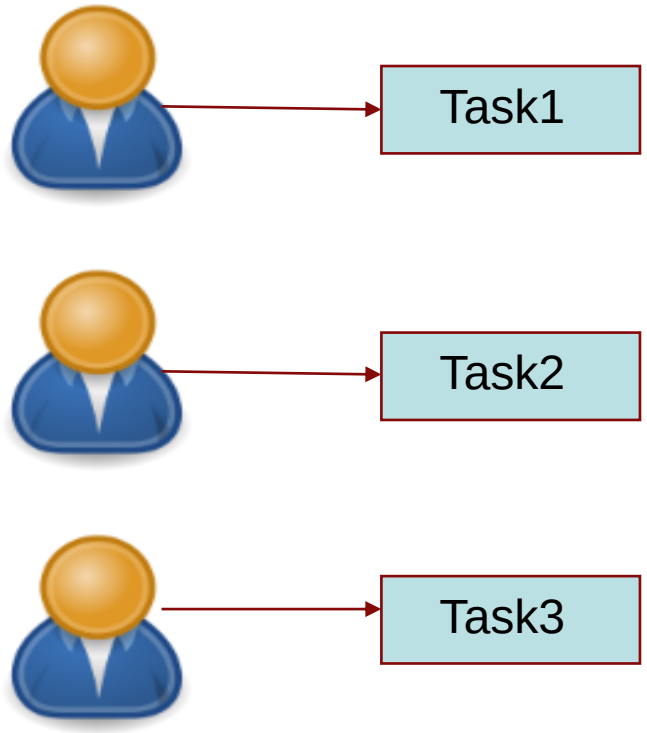
08 Git Operations & Commands

Version Control System

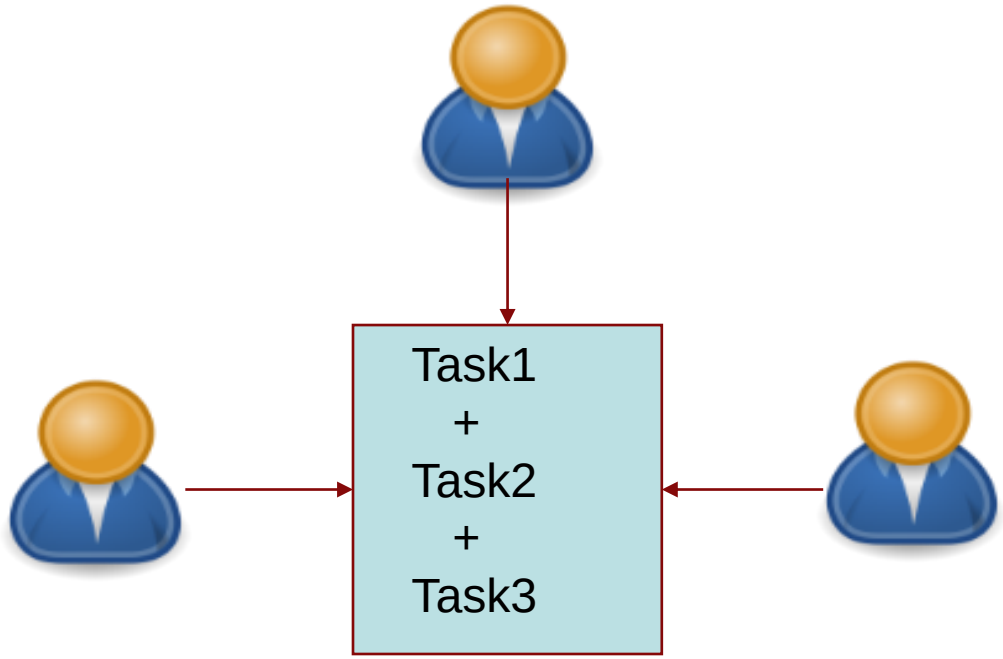
- ✓ Version control system is the management of changes to documents, software programs, websites and other collection of information
- ✓ These changes are usually termed as “versions”

1. Collaboration

Before



After



2. Storing Versions

- Snapshots of all versions are properly stored and documented.
- Versions are also named accurately

3. Backup

- In any case if central server crashes , a backup is always available in local servers.

4. Analyse

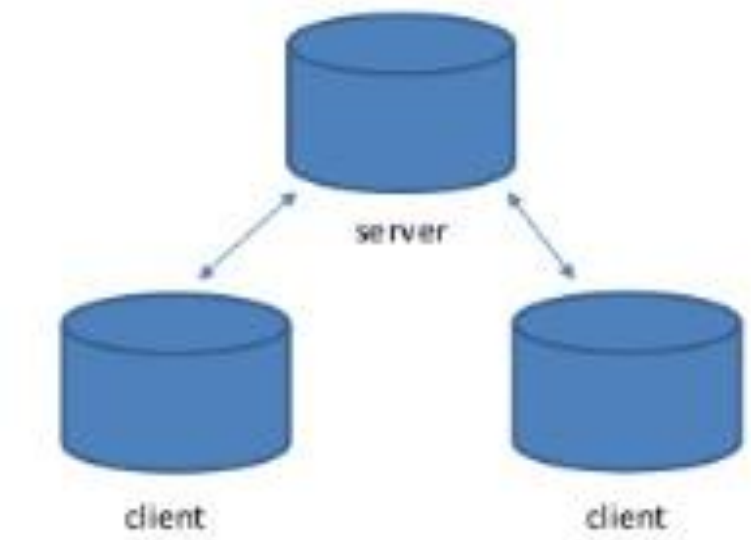
When you change versions

- VCS provides you proper description
- What exactly was changed
- When it was changed
- Who has changed

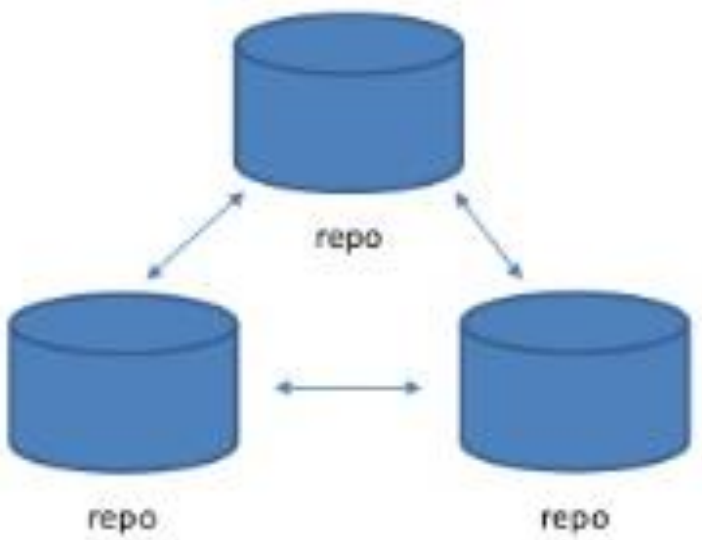
Hence, it is easy to analyse how project is evolved between versions

Types of Version Control System

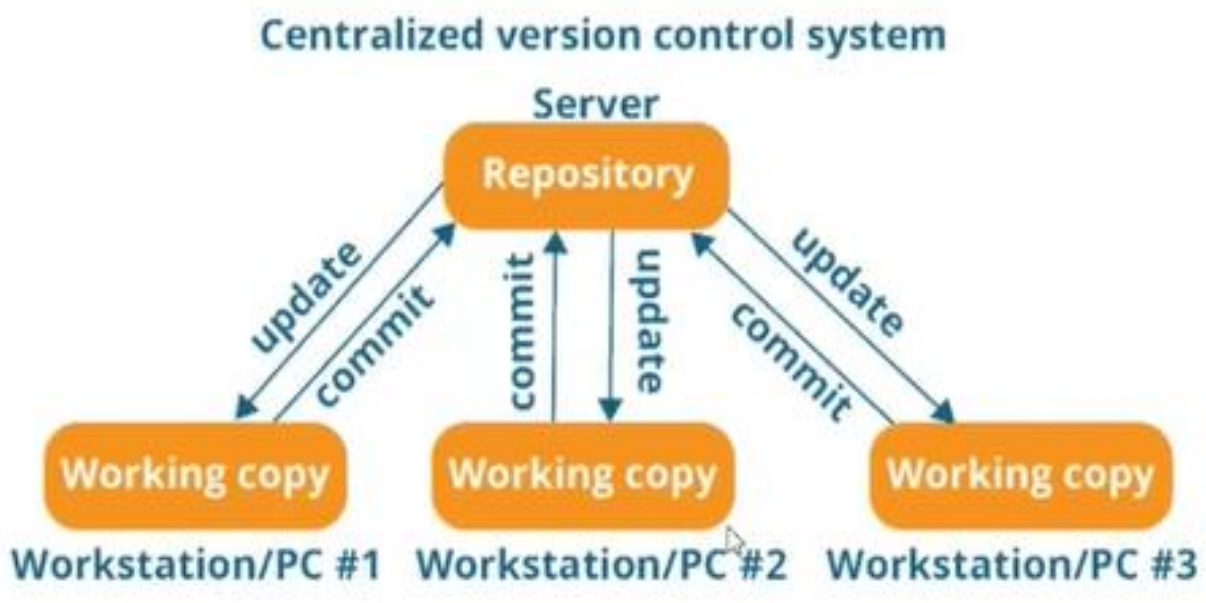
Centralized Version Control



Distributed Version Control



Centralised Version Control System(CVCS)



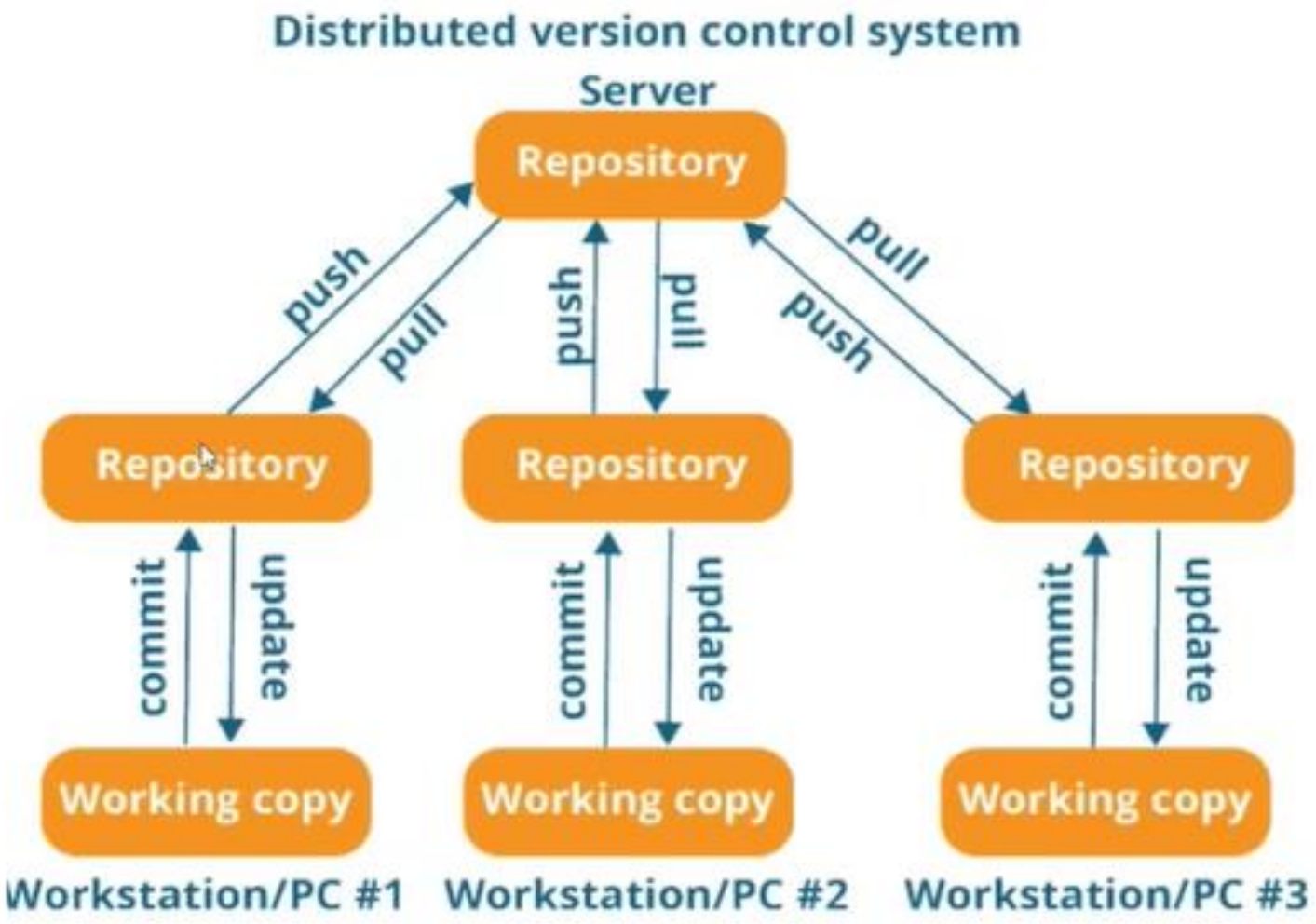
Centralized Version Control System

- CVCS uses a central server to store all files.
- It works on a single repository to which users can directly access a central server.
- Central server could be local or remote machine directly connected to each of the programmer's workstation.

Drawbacks of Centralized VCS

- Centralized repository is not locally available.
- Since everything is centralized, in any case of the central server getting crashed or corrupted will result in losing the entire data of the project.

Distributed Version Control System(DVCS)



Distributed Version Control System

- In Distributed VCS, every contributor has a local copy or "clone" of the main repository.
- Developer can update their local repositories with new data from the central server by an operation called "**pull**" and affect changes to the main repository by an operation called "**push**" from their local repository

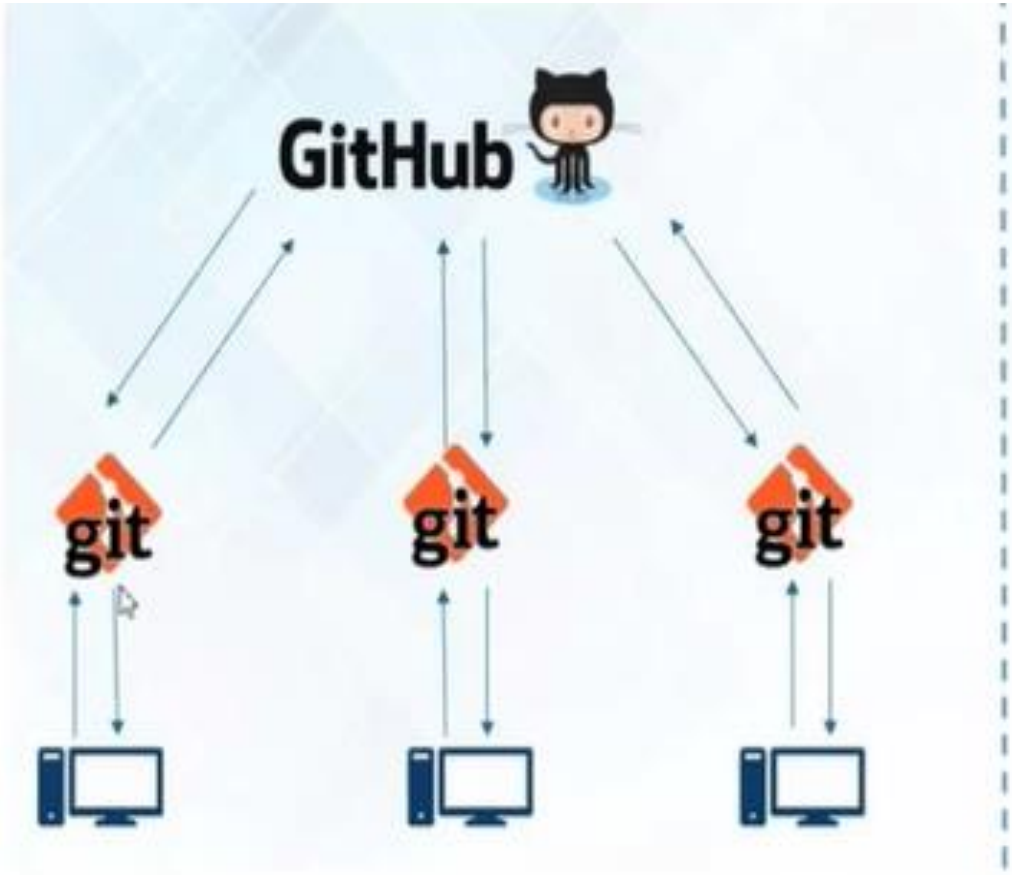
Advantages - DVCS

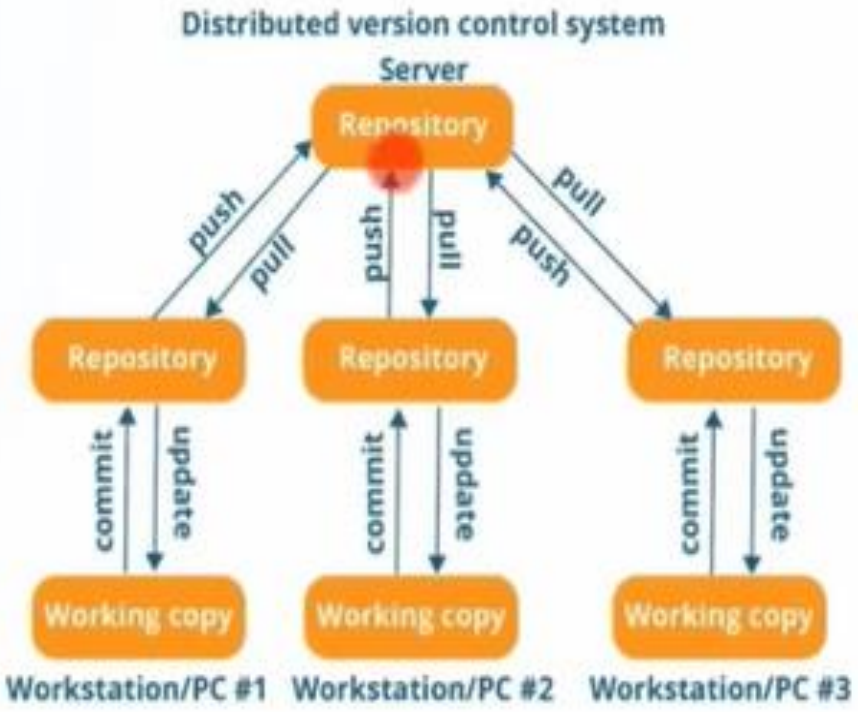
- All Operations are very fast because tools need to access the Hard Drive only.
- Committing new change-sets can be done locally without manipulating the data on the main repository
- If the central server gets crashed at any point of time, the lost data can be easily recovered from any one of the contributor's local repositories.

Version Control Tools

Interest over Time Graph

GIT & GITHUB

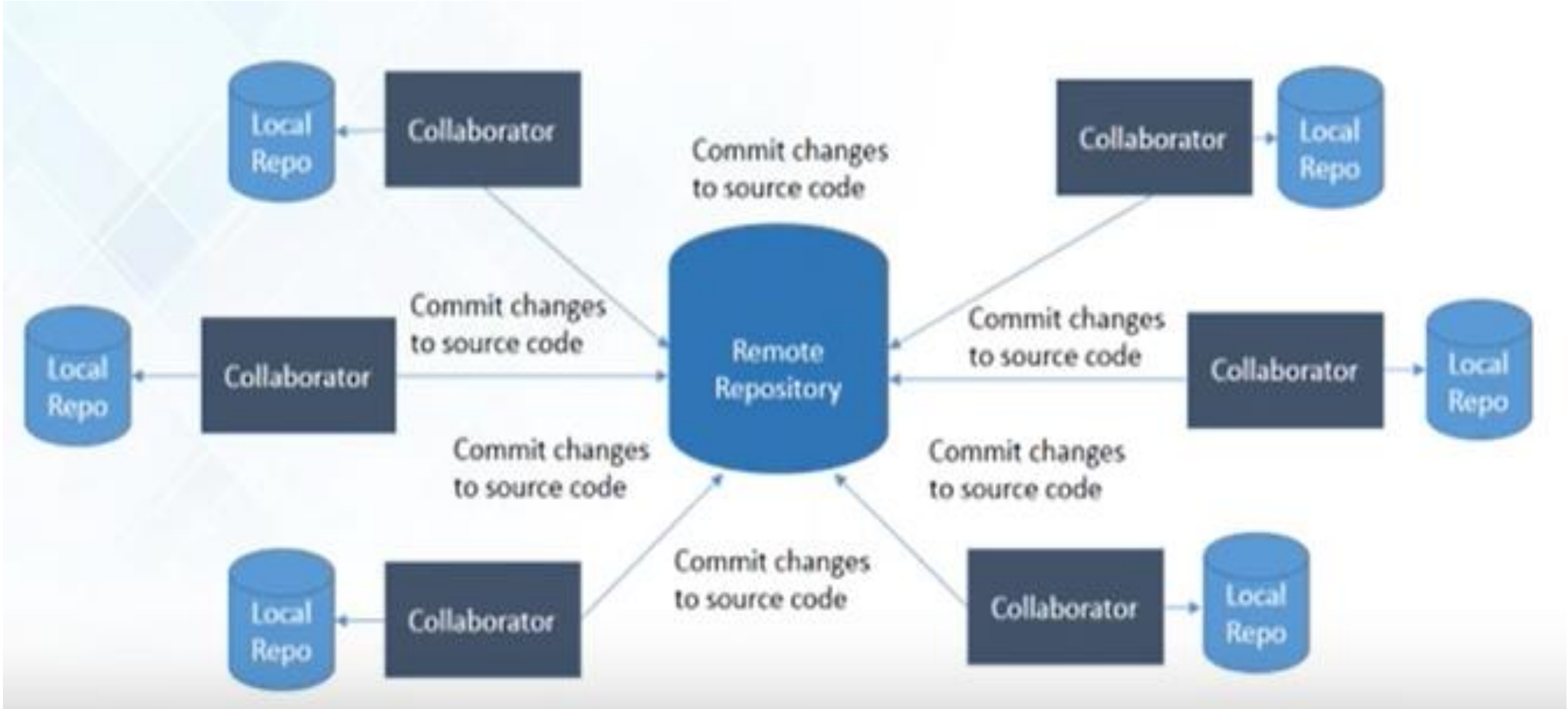




GIT Workflow

What is Git ?

Git is a distributed version control tool that supports distributed non linear workflows by providing data assurance for developing quality software.



GIT Installation

&

GitHub Account Creation

Git Features

What is a Repository?

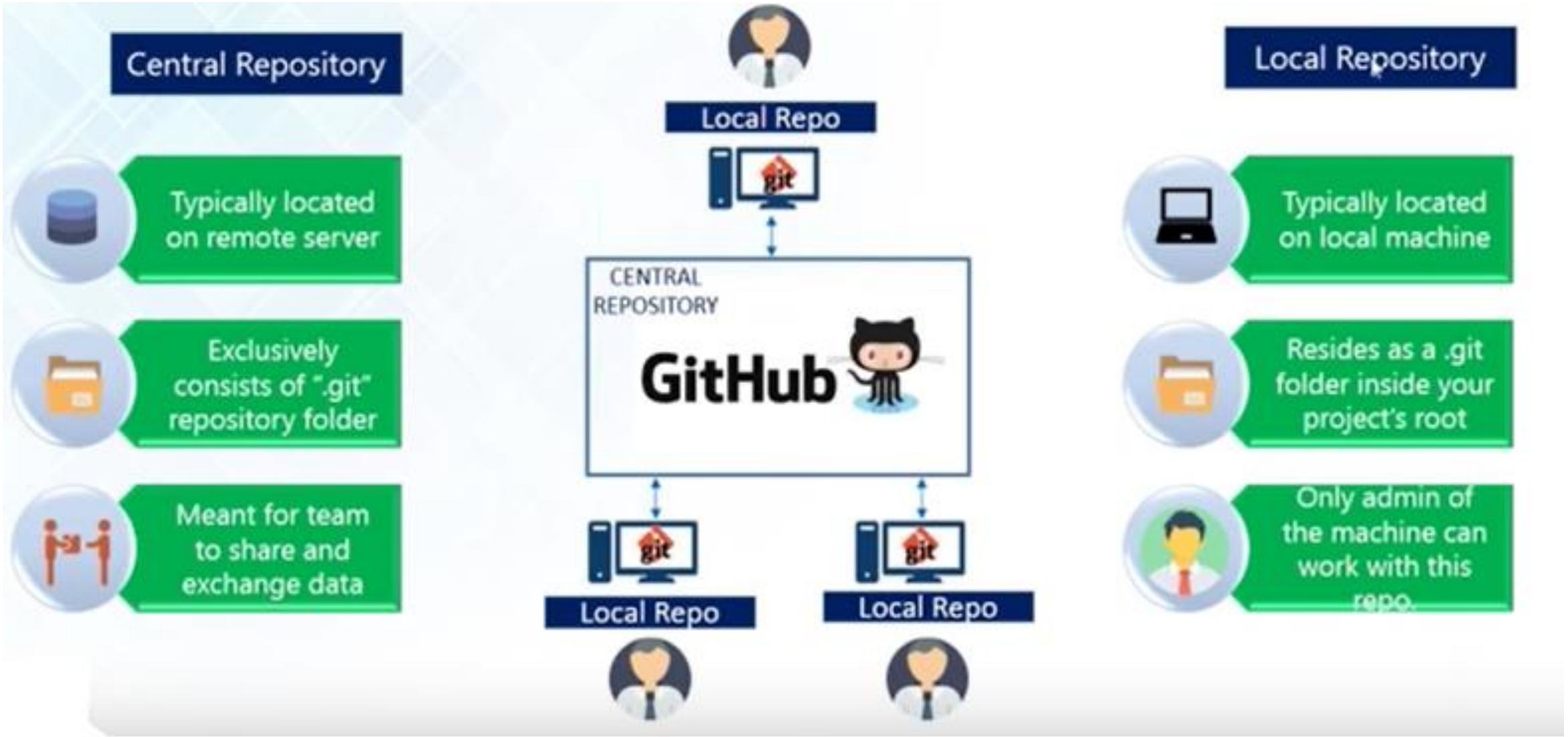
A directory or storage space where your projects can live. It can be local to a folder on your computer, or it can be a storage space on GitHub or any other online host. You can keep Code files, text files, Image files, inside a repository

There are 2 types of repositories:

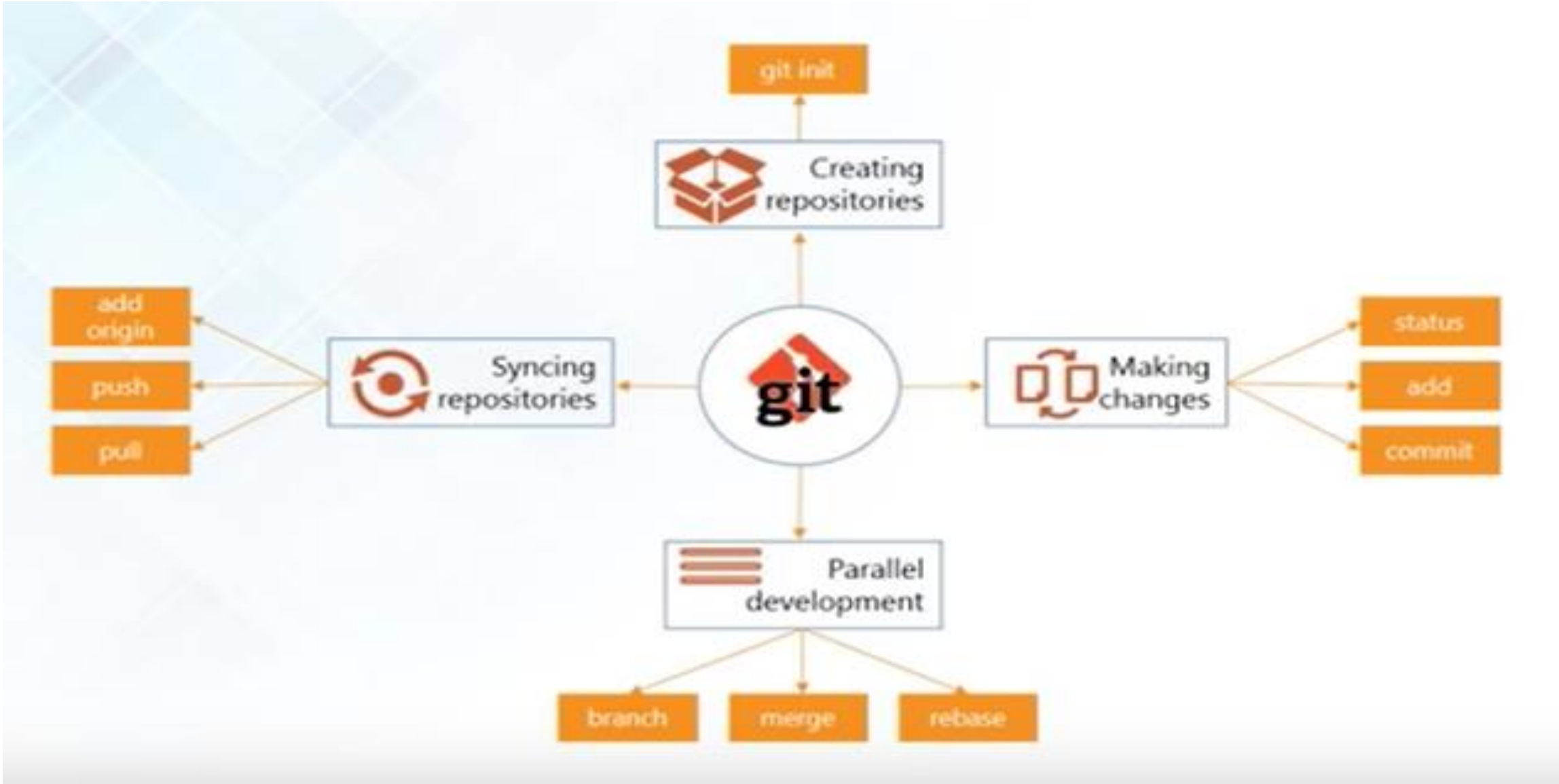
1. Central Repository

2. Local Repository

Central & Local Repository



Git Operations & Commands



Creating Repositories

Create Central Repository on GitHub (http://github.com)


```
git init
```

Install git on Local system and use “git init” command to create Local repository

OR

```
git clone
```

Clone your repository from GitHub

Syncing Repositories

- Use " git add origin <Repo URL>" to add remote repository
- "git pull" command to pull files from remote repository
- "git push" command to push changes into central repo

Making Changes

git status

- Tells you which files are added to index and are ready to commit

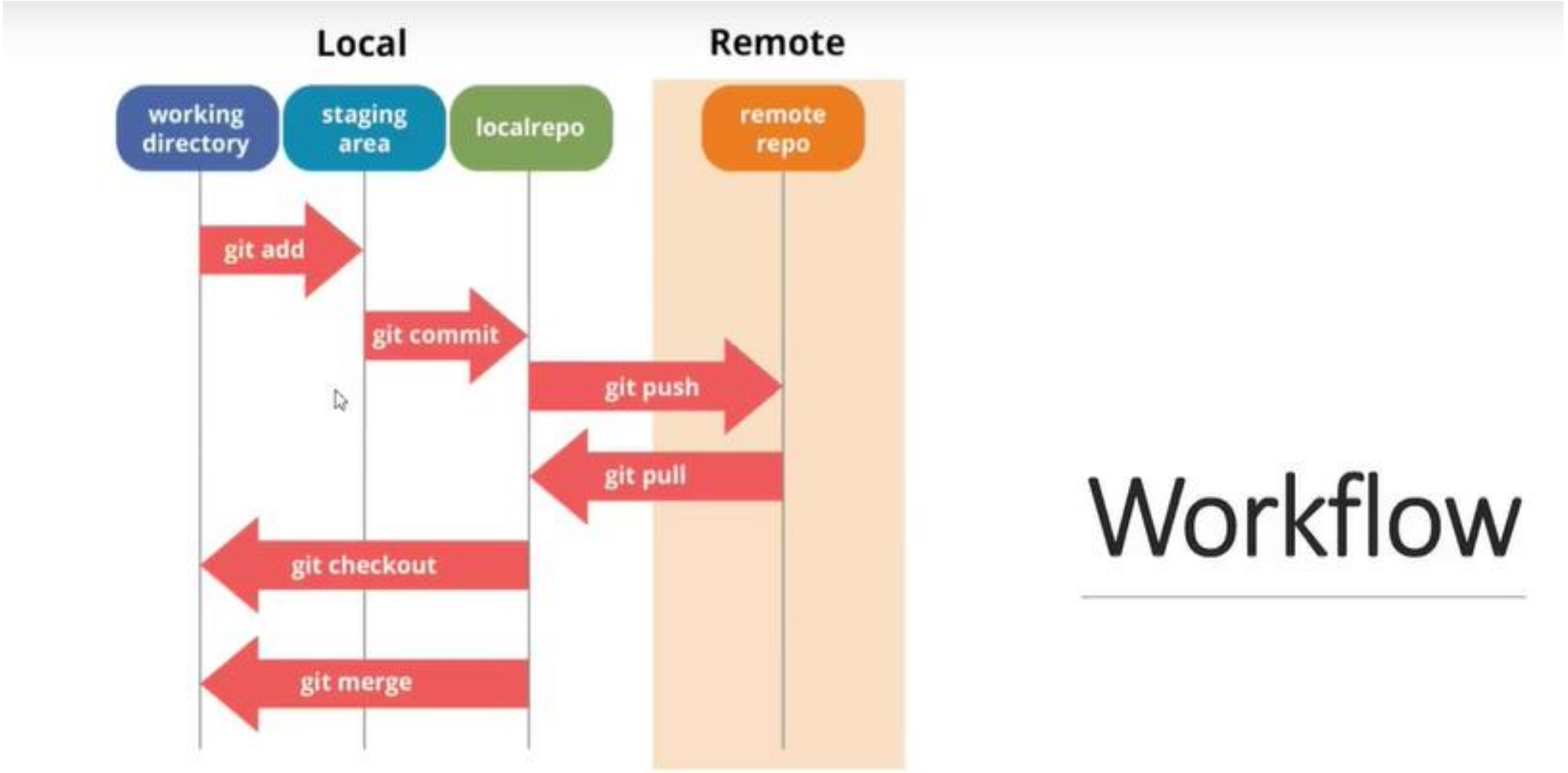
git add

- “git add” lets you add files to index

git commit

- “git commit” refers to recording snapshots of the repo at a given time.
- Committed snapshots never change unless done explicitly

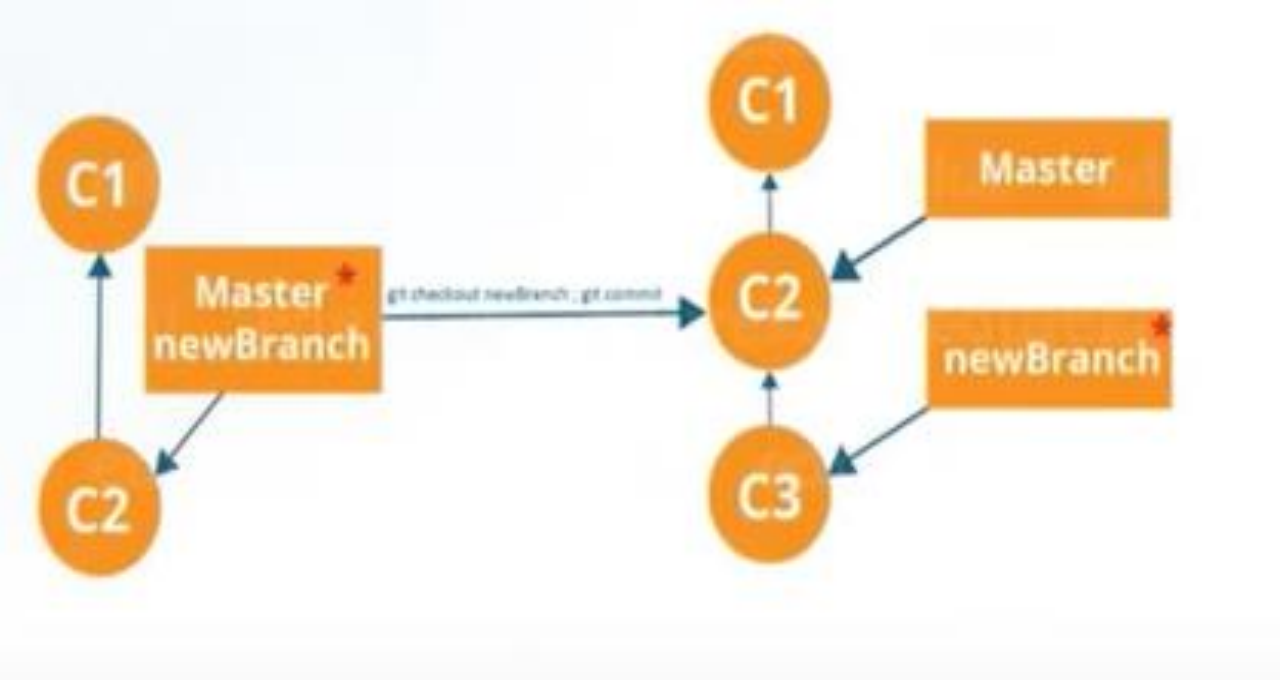
Git Flow



Workflow

Parallel Development - Branching

- Branches are pointers to a specific commit
- Branches are of two types
  - Local Branches
  - Remote-tracking Branches

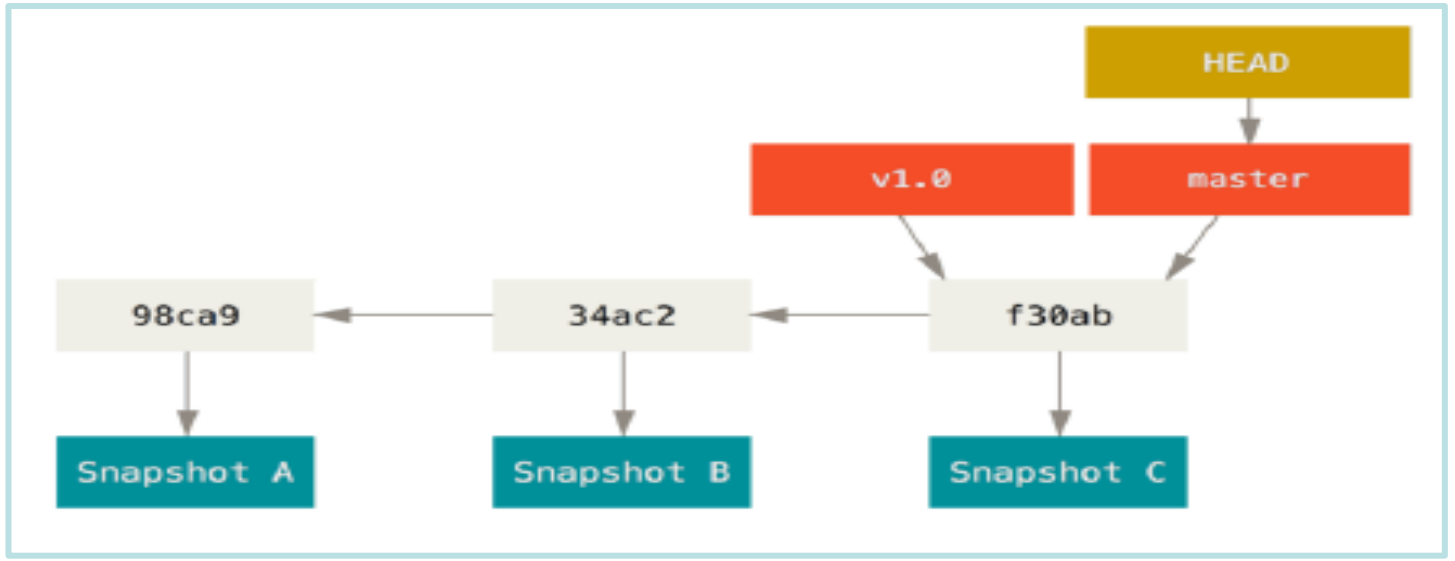


Parallel Development – Branching(Contd..)

Understanding Git Branching

1. A branch in Git is simply a lightweight movable pointer to one of these commits
2. .The default branch name in Git is master.
3. As you start making commits, you're given a master branch that points to the last commit you made.
4. Every time you commit, it moves forward automatically.
5. The "master" branch in Git is not a special branch. It is exactly like any other branch. The only reason nearly every repository has one is that the *git init* command creates it by default and most people don't change it.

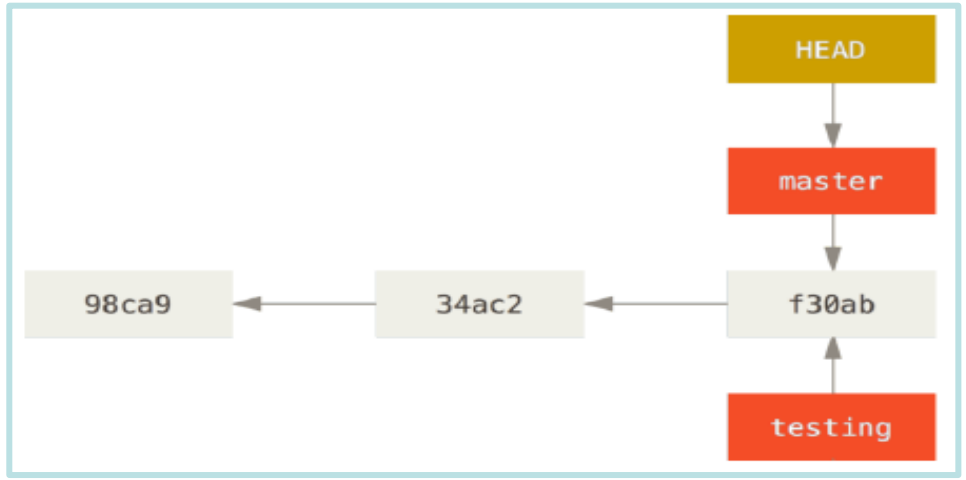
.

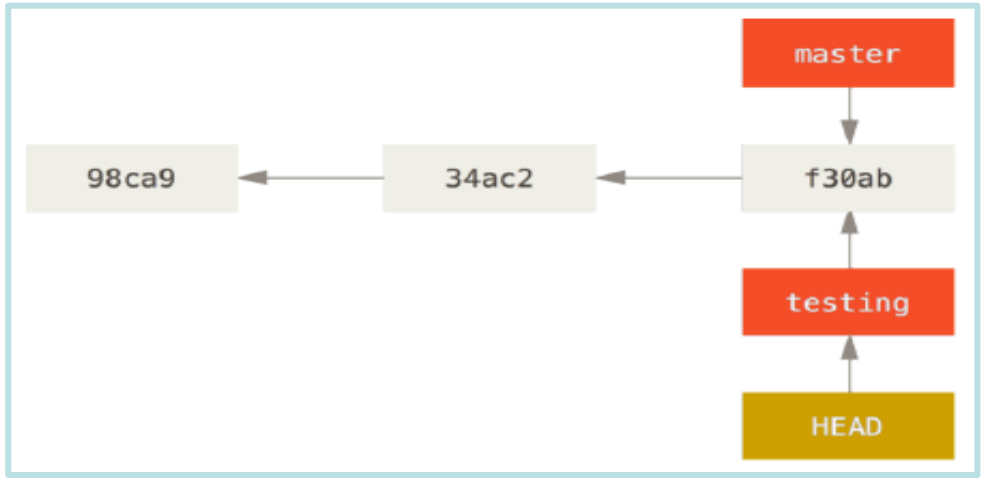


How to create branch in Git & GitHub ?

Creating new Branch

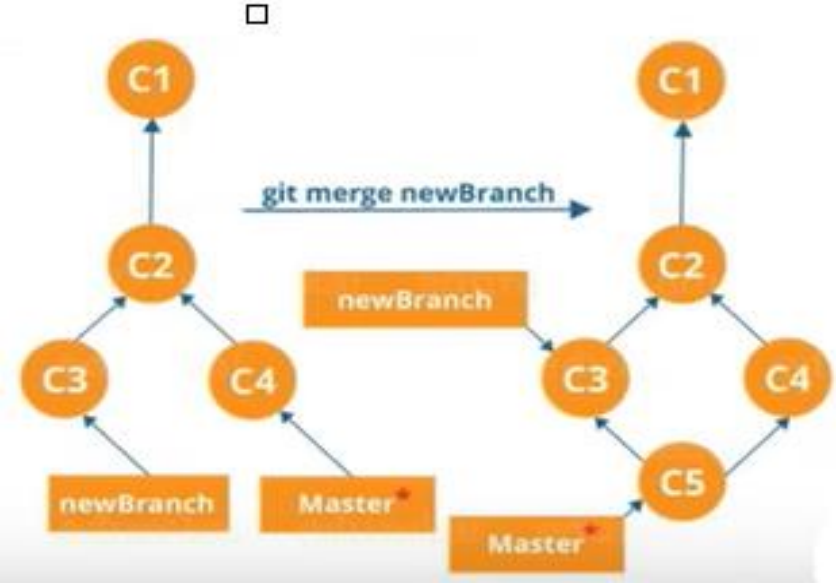
1. A new branch is created using *git branch <branch name>* command
2. This creates a new pointer to the same commit you're currently on
3. How does Git know what branch you're currently on? It keeps a special pointer called HEAD
4. This is a lot different than the concept of HEAD in other VCSs ,In Git, this is a pointer to the local branch you're currently on
5. The *git branch* command only *created* a new branch – it didn't switch to thatbranch.
6. To switch to an existing branch, *git checkout <branch name>* command is used .
7. This moves HEAD to point to requested branch





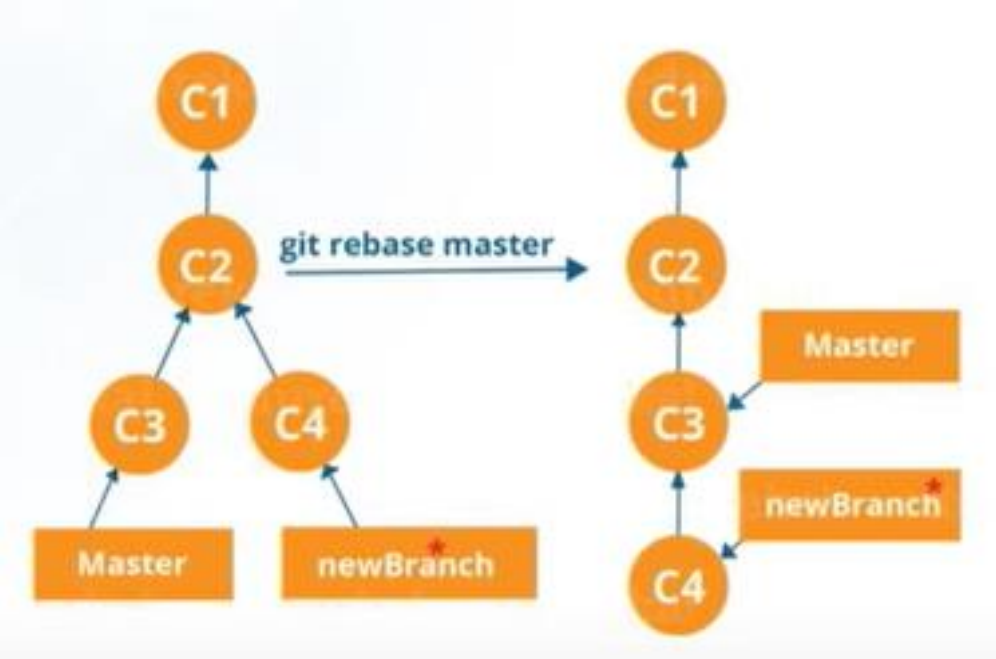
Parallel Development - Merging

- Merging is a way to combine the work of different branches together
- Allows to branch off, develop a new feature and combine it back again

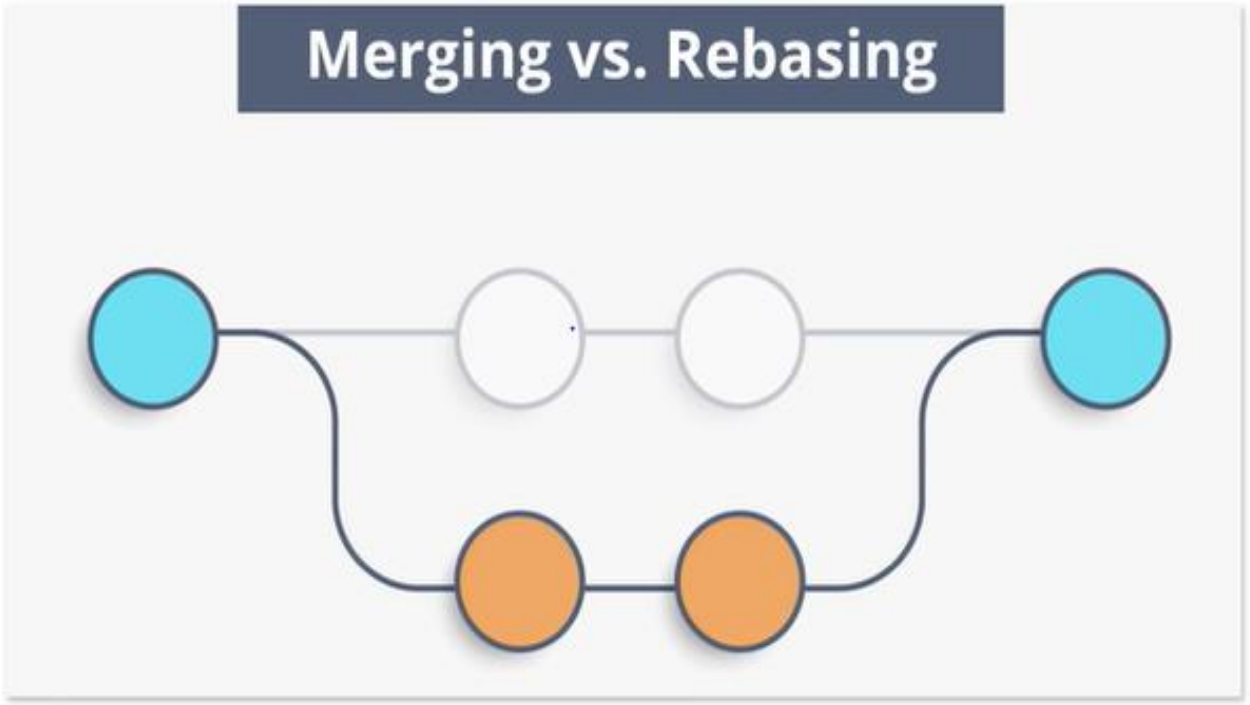


Parallel Development - Rebasing

- Rebasing is also a way of combining the work between different branches.
- It can be used to make linear sequence of commits.

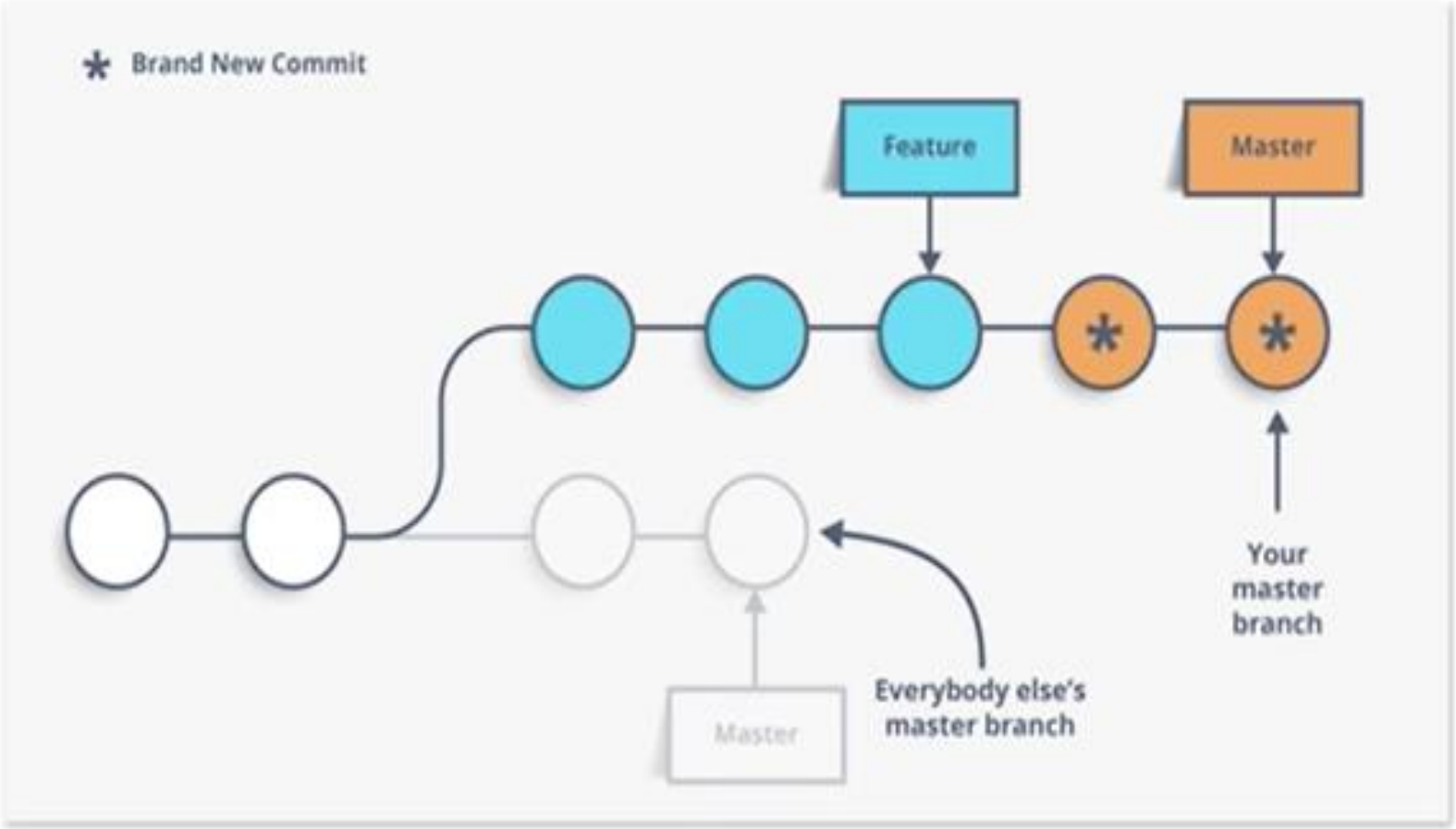


Git Merge vs Git Rebase – which one to use?



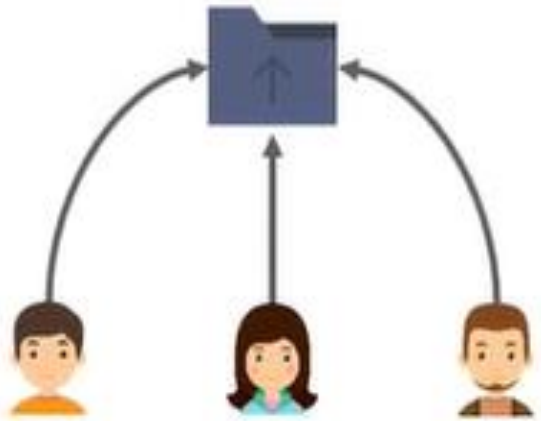
When not to do Rebase?

- Do not rebase commits that exist outside your repository.

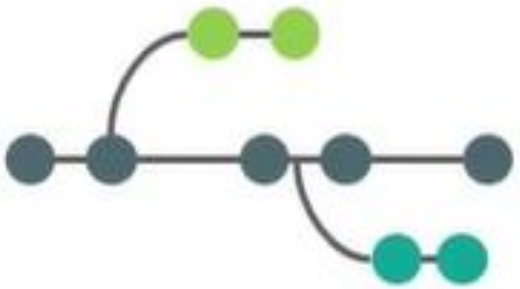


Git Workflow

Centralized WorkFlow

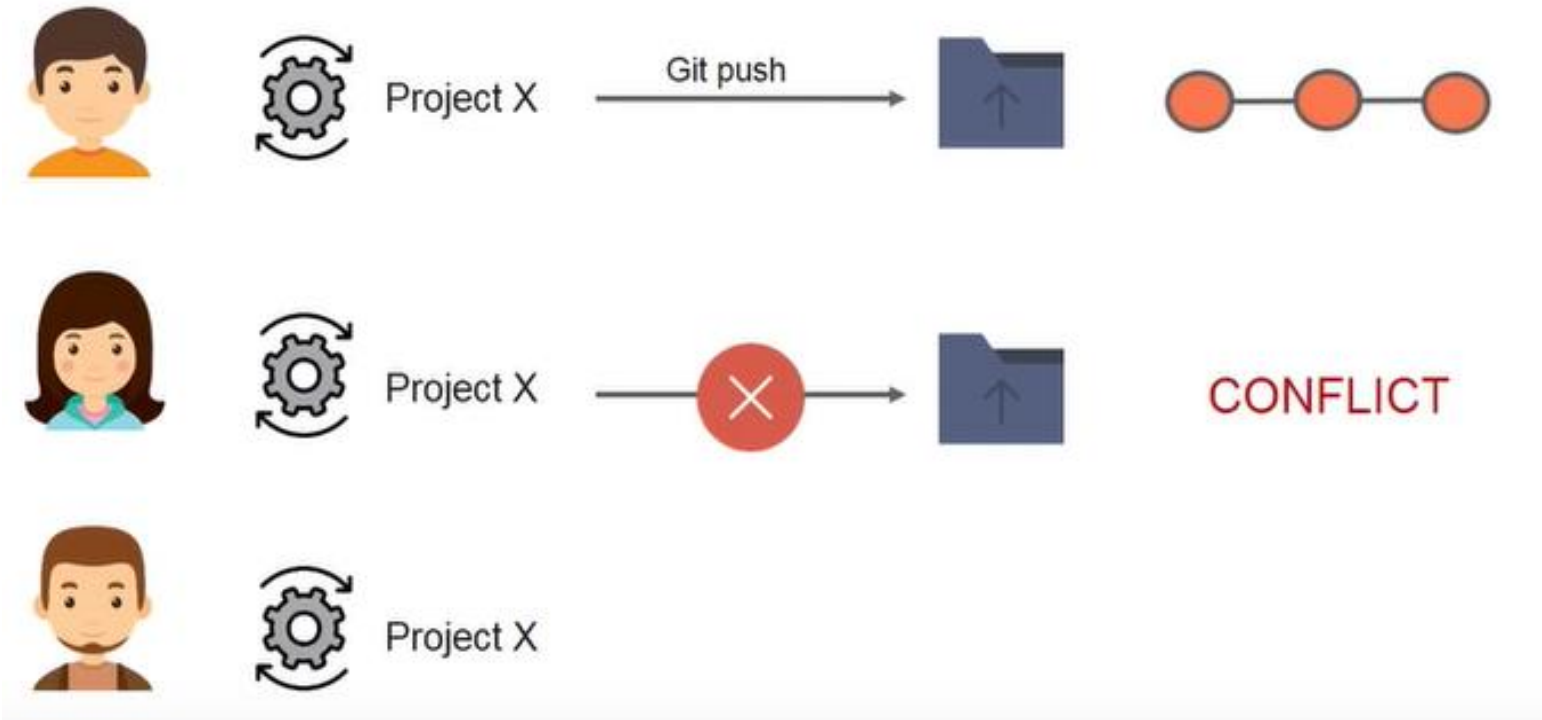


Feature Branch WorkFlow



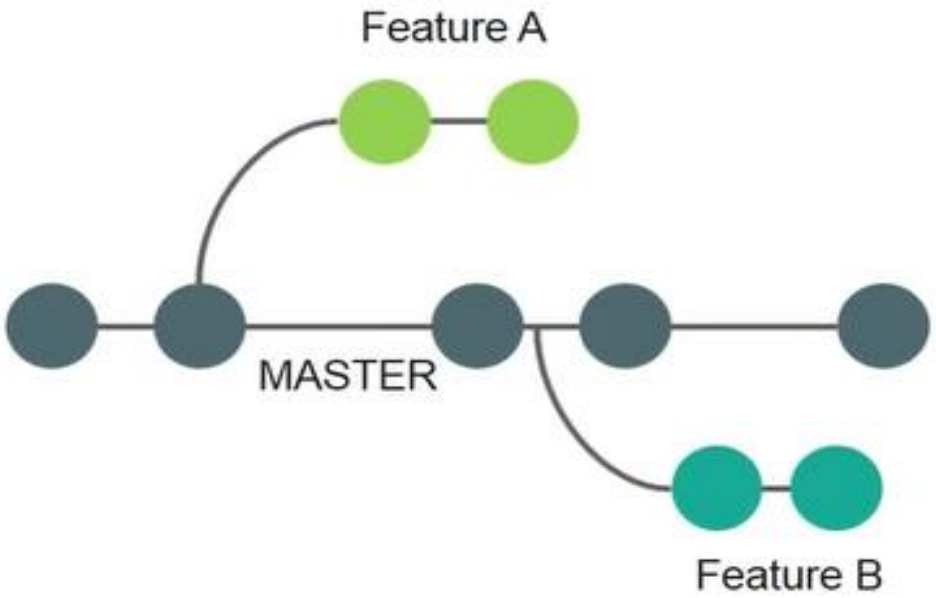
GitFlow WorkFlow

Centralised Workflow

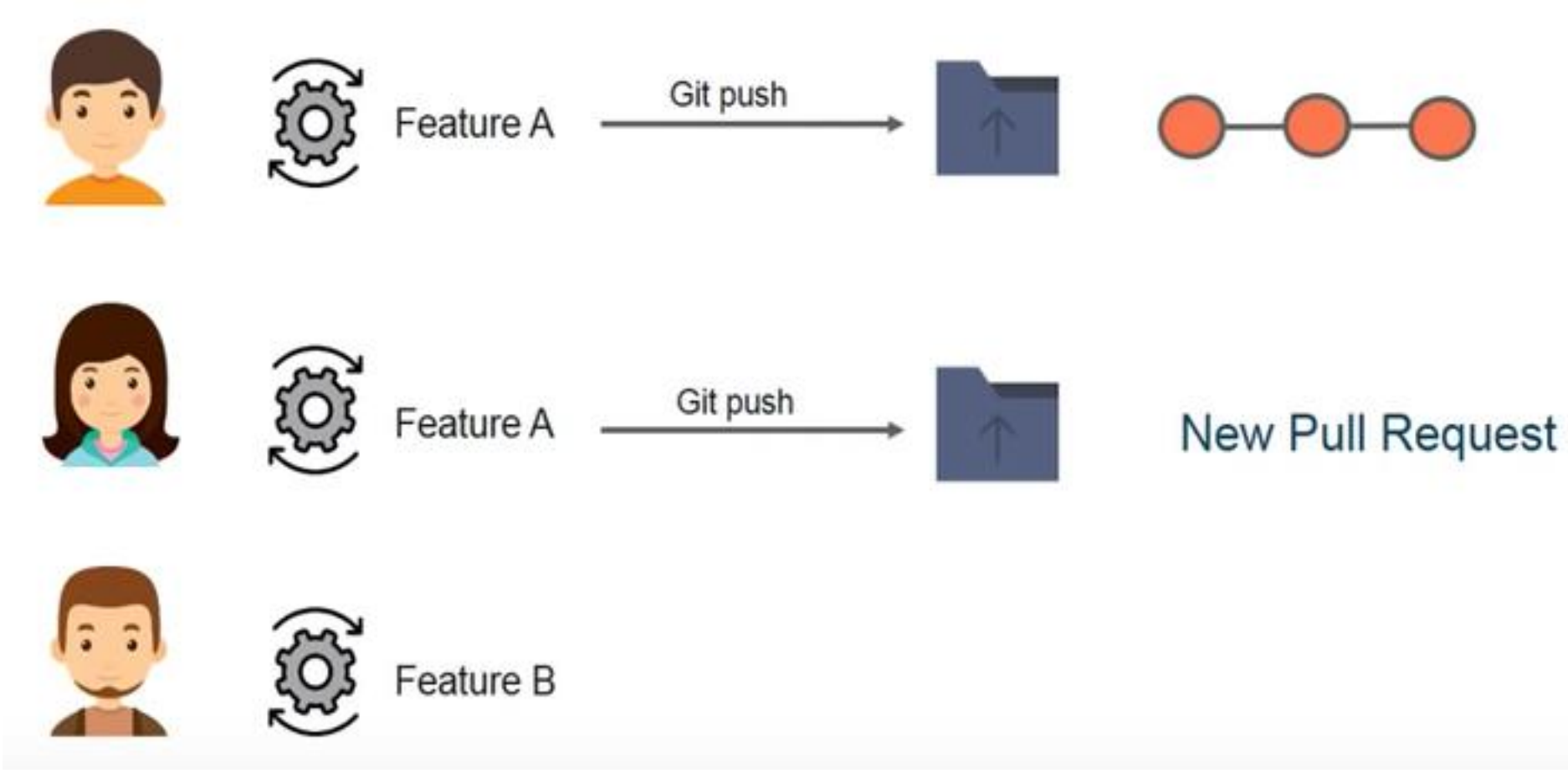


Resolving Conflicts
- Demo

Feature Branch Workflow



Centralised Workflow



GIT COMMANDS

git config :
to configure git and setting preferences

git commit :
It take content that is staged and record a permanent snapshot and move branch pointer to current commit

git diff :
to see differences between any two directory trees, commit/staging/working

git log :
to show the reachable recorded history of a
project from the most recent commit snapshot backwards, by default of current branch

git help :
to view all the documentation about a command

git status :
It shows status of files in working directory and staging area

git checkout :
switch branches and check content out into working directory

git tag :
give a permanent bookmark to a specific point in the code history.

git init :
to turn a directory into new git repository

git clone :
It is a wrapper around several other commands,(init, remote add, fetch, checkout). It create a new directory, connect to remote repo, fetch the content from remote repo, the checkout latest commit into working directory

git rm :
to remove files from working directory and staging area opposite to git add command

git branch :
to create a new branch, delete branches and rename branches

git pull :
combination of the git fetch and git merge commands, fetch from the remote and then immediately try to merge it into the current branch

git add :
to add content from working directory into staging area for next commit

git merge :
merge one or more branches into the branch have checked out, hen advance the current branch to the result of the merge

git push :
communicate with another repository, compare it with local database for difference, and then pushes the difference into the other repository

git fetch :
To communicates with a remote repository and fetches down all the information that is in repository and not in current one and stores it in local database

git clean :
to remove files from working directory and staging area opposite to git add command