**GIT & GITHUB**

**Version control**

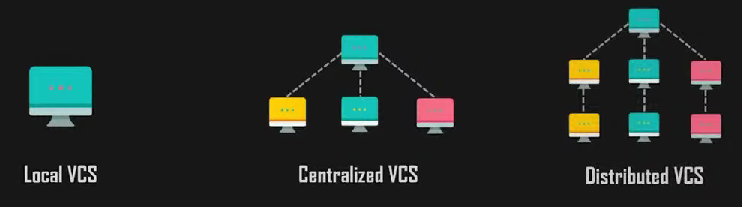
version control is a system that records changes to a file or a set of files over time so that you can recall specific versions later. These versions are recorded in a repository and can be recalled from the same. There are **local, Centralized** and **distributed version** control systems.

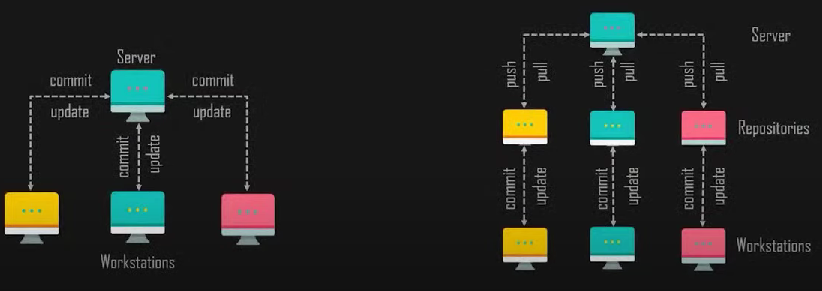
* Collaboration: Shared workspace & real-time updates
* Manage versions: All versions of code are preserved
* Roll backs: easy rollback from current version
* Reduce downtime: reverse faulty update & save time
* Analyse project: analyse and compare versions

**Repository**

A repository(repo) is 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 another online host(such as Github). You can keep code files, text files, image files, you name it inside a repository.

**Centralised vs distributed vcs**

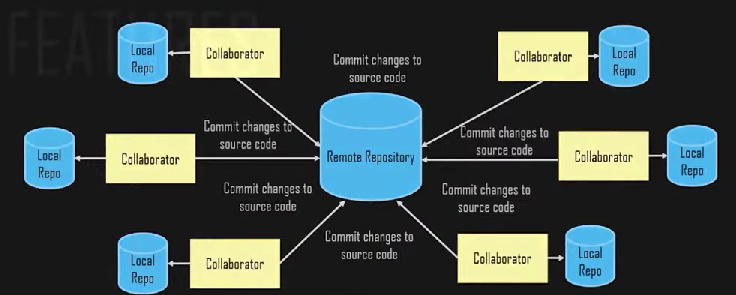




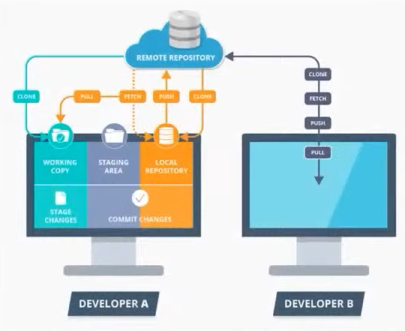
**Git**

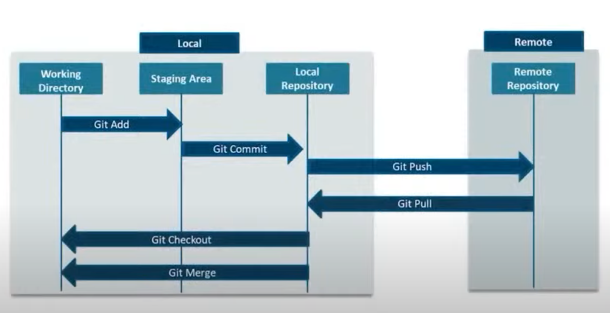
Git is a distributed version control tool that supports distributed non-linear workflows by priding data assurance for developing quality software. It lets you and your team of developers work together on the same project from anywhere. Team members can work on files and easily merge their changes into one source.

* Economical: Released under GPL's license , it is free & open source
* Non-linear: supports non-linear development of software
* Snapshots: Records changes made to a file rather than file itself
* Distributed: Every user has his own copy of the repository data stored locally
* Robustness: Nearly every task in Git is undo-able
* Integrity: no changes can be made without git recording it
* Branching: every collaborator's working directory is a branch by itself
* Speed



**Git working**





**Git operations & commands**

**Create repo**

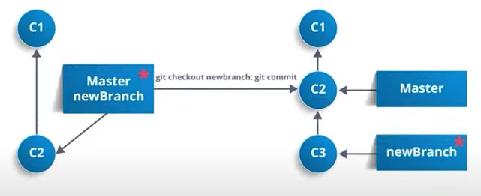
1. Git init- this command creates a new repository
2. Git clone- create a copy of the original repository on your local machine
3. Git fork- create a copy of the original repository on your github account

**Sync repo**

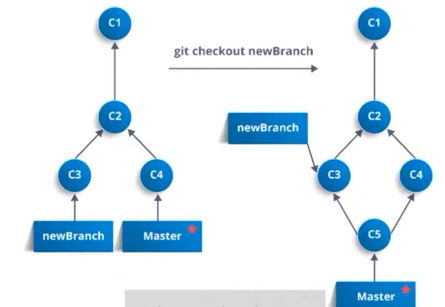
1. Git origin-lets you add remote repository(git remote add origin <repo\_link>)
2. Git pull- lets you copy all the files from the master branch of remote repository to your local repository(git pull origin master)
3. Git push- lets you push your local changes into central repository(git push origin master)

**Parallel development**

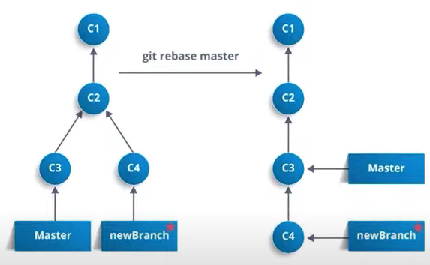
1. Branch- Branching is an integral part of any version control system. Unlike other vcs, git doesn't create a copy of existing files for new branch. It points to snapshot of the changes you have made in system(git branch <branch\_name>)



1. Merge- it integrates the changes made in different branches to one single branch.(git merge <branch\_name>)



1. Rebase- it is used when changes made in one branch needs to be reflected in another branch.(git rebase master)



**Make changes**

1. Git status-tells you which files are added to index and are ready to commit(git status)
2. Git add-lets you add files to your index(git add <options> <file\_name>)
3. Git commit-it refers to recording snapshots of the repository at a given time.

Commited snapshots will never change unless done explicitly(git commit-m <message>)

**Other commands**

1. Archive - archive repository(git archive master --format=zip -output=../name\_file.zip)
2. Bundle- bundle repository.(git bundle create ../repo.bundler master)
3. Stash- stash uncommited changes.(git stash, git stash apply)
4. Show remote repository(git remote)
5. Rename repository(git remote rename)
6. Remove repository(git remote rm , git remote remove)
7. Compare remote repos(git diff <masterbranch\_path> <remotebranch\_path>)



**Github**

* Web based git repository hosting service
* Easy management of code
* Open-source software for version control
* Effective collaboration
* Bug tracker

**Need for Github**

* Developers need a web/cloud based code hosting platform
* Useful for version control
* Enables effective collaboration
* Download projects and files in one go
* Easy evaluation of each other's work

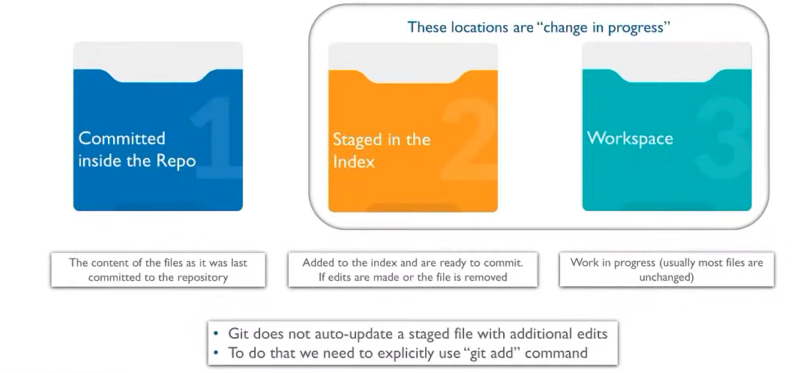
**Advantages**

* Immensely powerful community
* The largest shared repository
* Easy version control
* Secure cloud storage

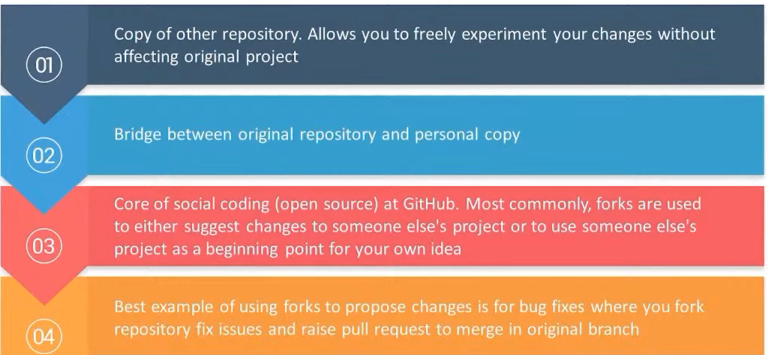
**Git vs github**

* Git is the tool and github is the service for projects that use git
* Git pushes or pulls data from the central server. Github is a core hosting platform for version control collaboration
* Github is a company that allows you to host a central repository in a remote server

**File locations pre & post commit**



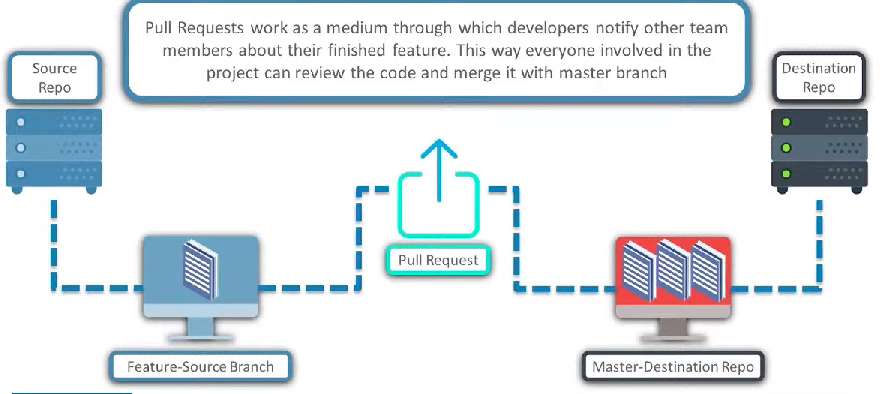
**Fork**

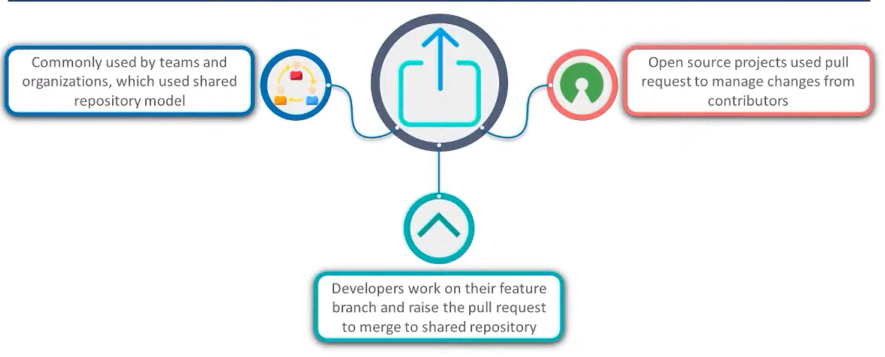


**Upstream & downstream**



**Pull request**





**Git & github flow**

**Git**

Works with different branches to manage each phase of the software development

It is perfect when one/two developers are working on the same feature

**Github**

Lightweight, branching workflow

Supports teams that deploy often

Cent

ered around small changes