

GIT & GITHUB

Page No.:

Date: / /

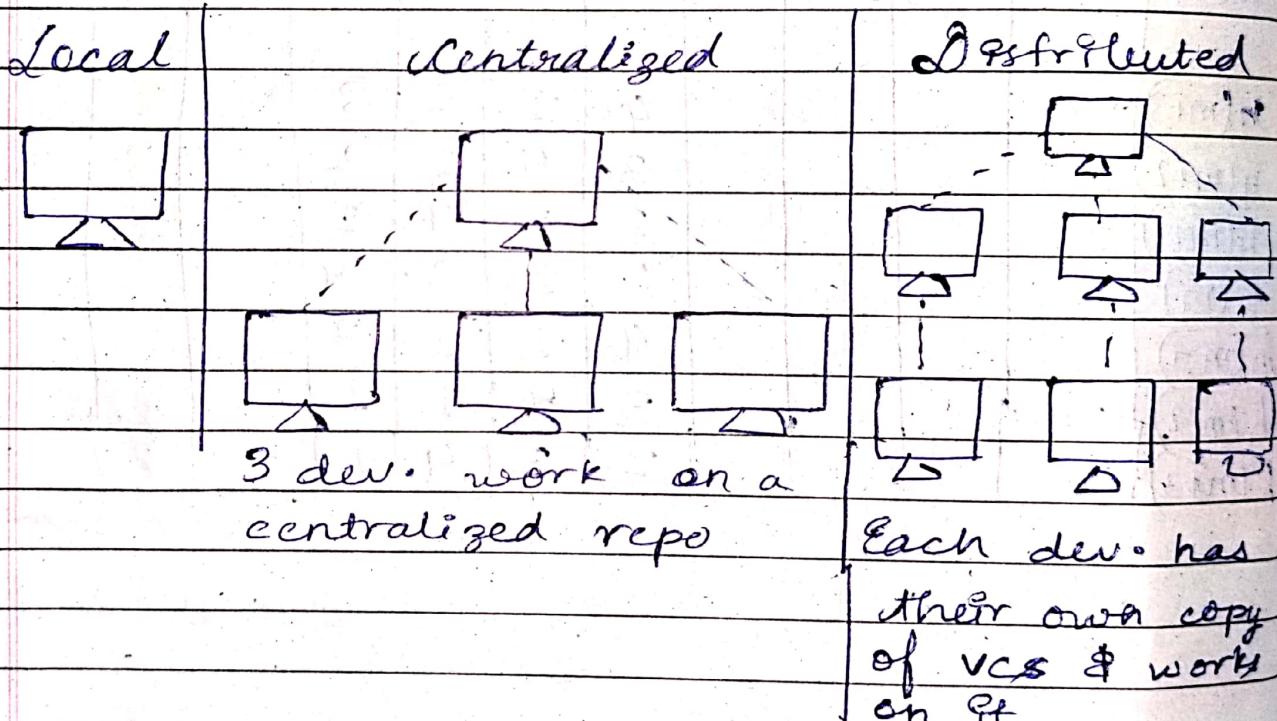
What is version control?

Version control is a system that records changes to a file or set of files over time so that we can recall specific versions later.

These versions are recorded in a repository & can be recalled from the same.

There are 3 version control systems.

- Local
- Centralized
- Distributed



Uses:

→ Collaboration: Shared workspace & Real time updates

→ Manage versions: All versions of code are preserved

Rollback

Rollbacks: Easy roll-back from current version

Reduce Downtime: Reverse faulty update & save time

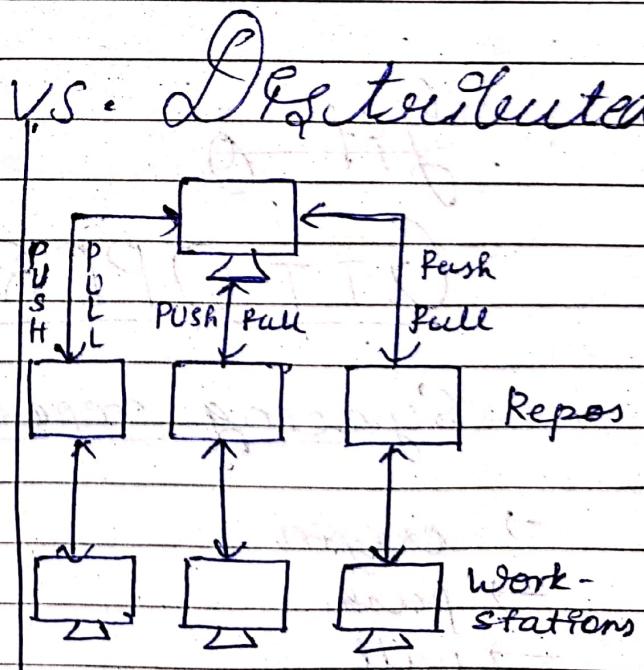
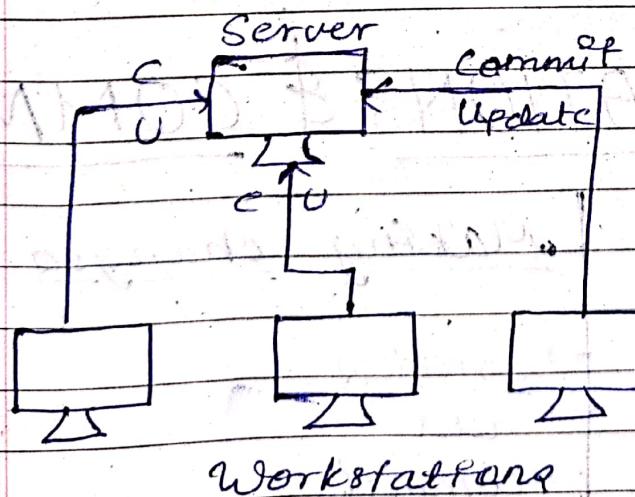
Analyse Project: Analyze & compare versions

~~ed over~~

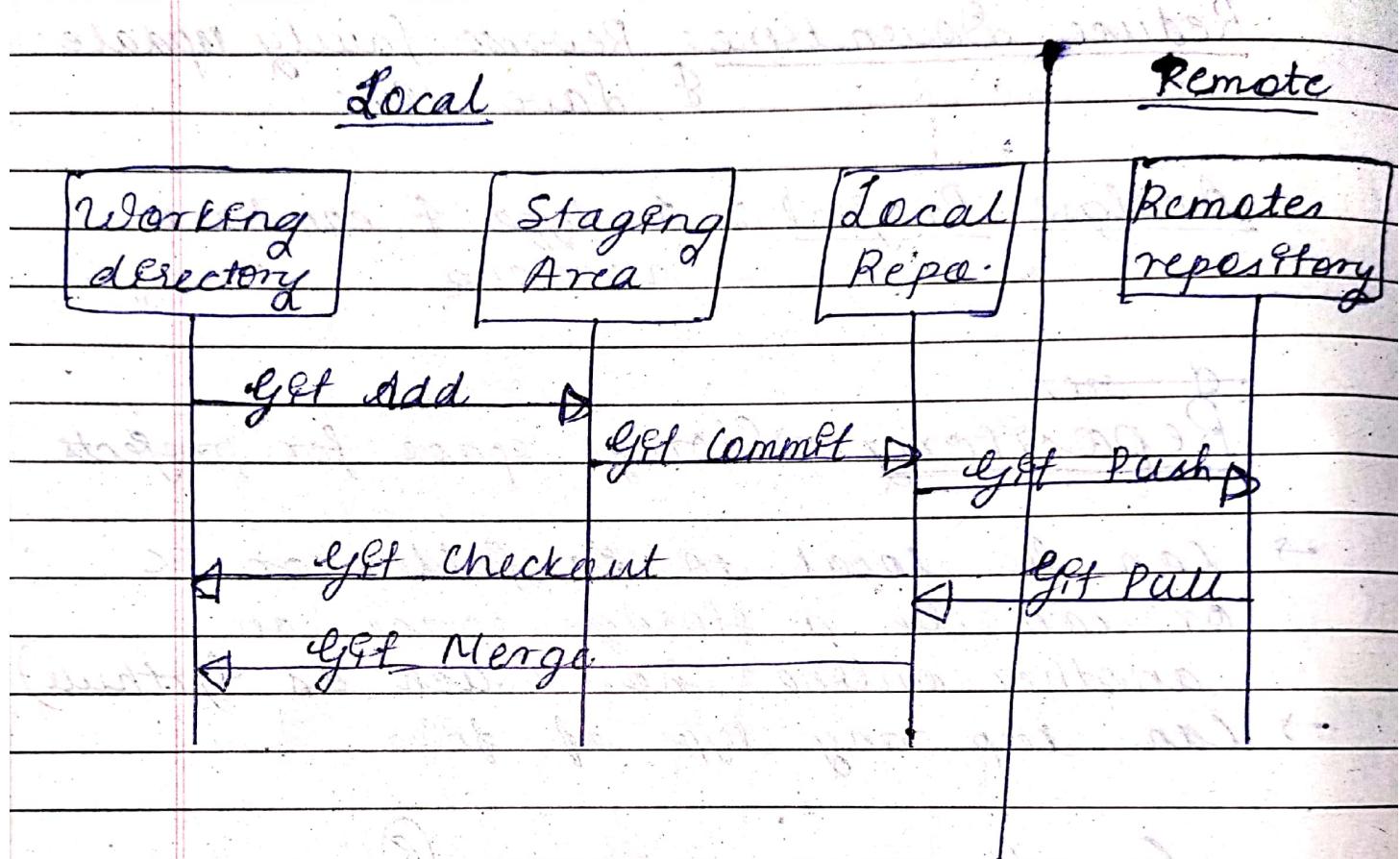
Repository :- Storage space for projects

- Can be local to a folder or PC or can be a storage space on another online host (such as GitHub)
- Can keep any type of files

Centralized vs. Distributed



git: → Distributed VCS
 → Team mem. can work on their local repos & merge changes onto source



git

GIT OPERATIONS & COMMANDS

Syncing repos

- origin
- push
- pull

Making changes

- status
- add
- commit

Creating repos

→ get init

Parallel development

- branch
- merge
- release

get init: Create a new get repo on your local machine is called local repo.

get clone: Used to create a copy of org. repo on your local machine.

get fork: Create copy of original repo on your GitHub account.

→ get clone

→ get origin: Link any remote repo with your local repo.

Syntax: get remote add origin <repo link>

→ get pull: Copy all files from master branch of remote repo to local repo.

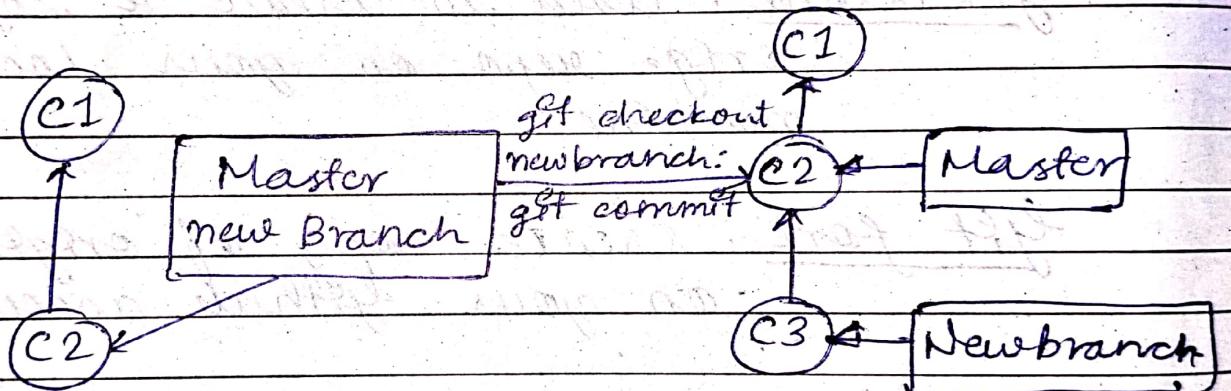
Syntax: get pull origin master

git push: Lets you push your local changes onto central repos

Syntax: git push origin master

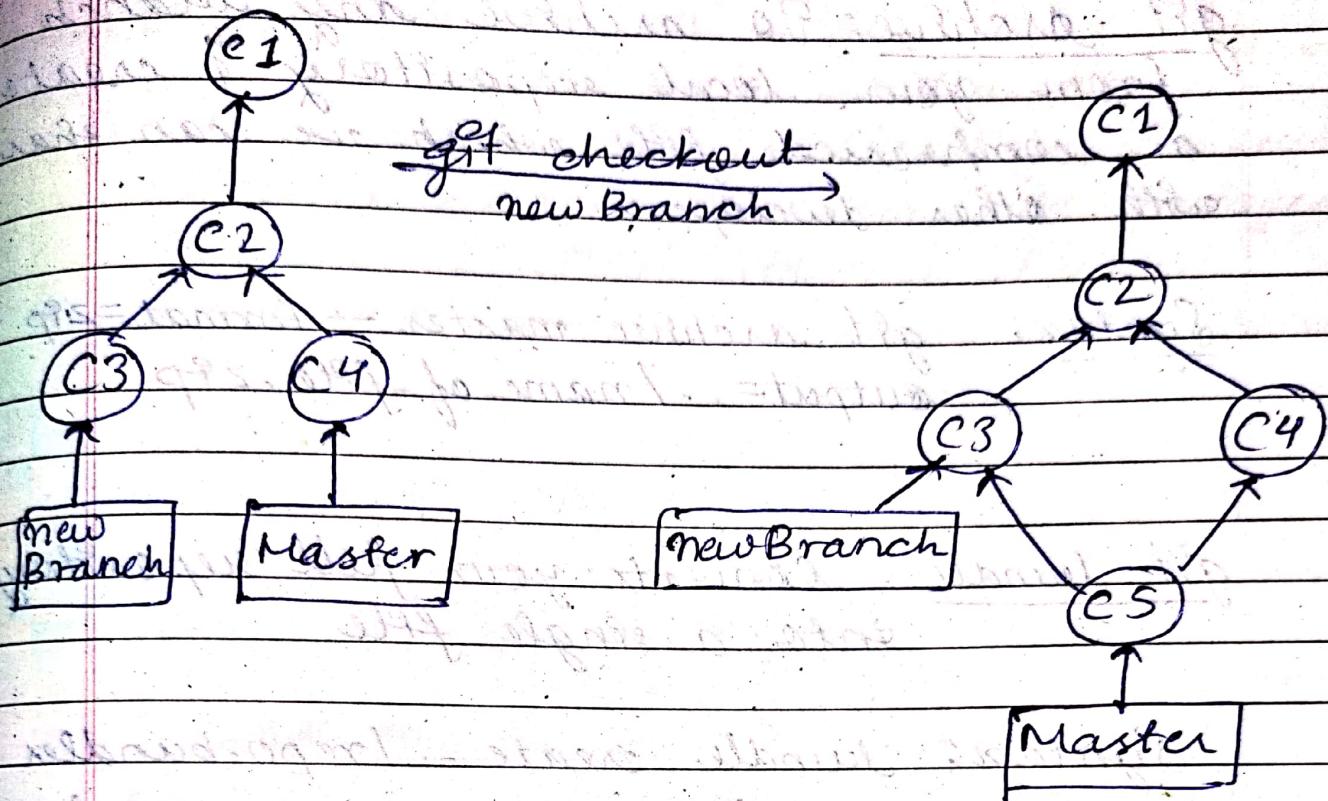
Branching:

Unlike other VCS, git does not create a copy of existing files for new branch. It points to snapshot of the changes you have made in system.



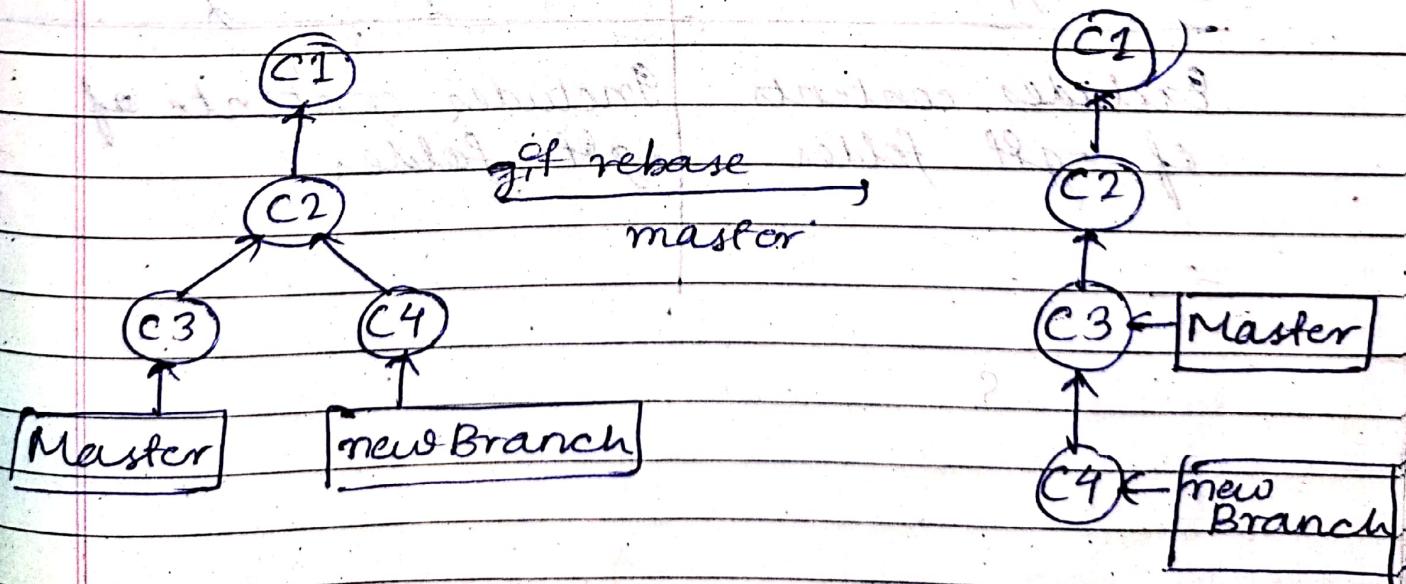
git add
git commit
git branch <branch-name>

Merge: Integrates changes made in diff. branches to one single branch. Main



Syntax: `git merge <branch-name>`

Release: Used when changes made in one branch needs to be reflected in another branch



`git release master`

git archive: To archive any branch from your local repository & create a compressed file which we can share with other developers.

Syntax: `git archive master --format=zip -o output=.. /name-of-file.zip`

git bundle: Converts your full repository into a single file

Syntax: `bundle create -l repo.bundle master`

Archive

Sends only specified contents to std. o/p

Excludes contents of .git folder

Bundle

Creates a single file containing both file contents & history

Includes contents of .git folder

git stash: Stash means to store (changes) safely in a hidden place (the stash stack). Stashing the current working directory's staged or unstaged changes & then storing them in the stash stack reverts current working directory to the last commit.

Syntax: `git stash`
`git stash apply`

git
 Revision control system to manage source code & code history

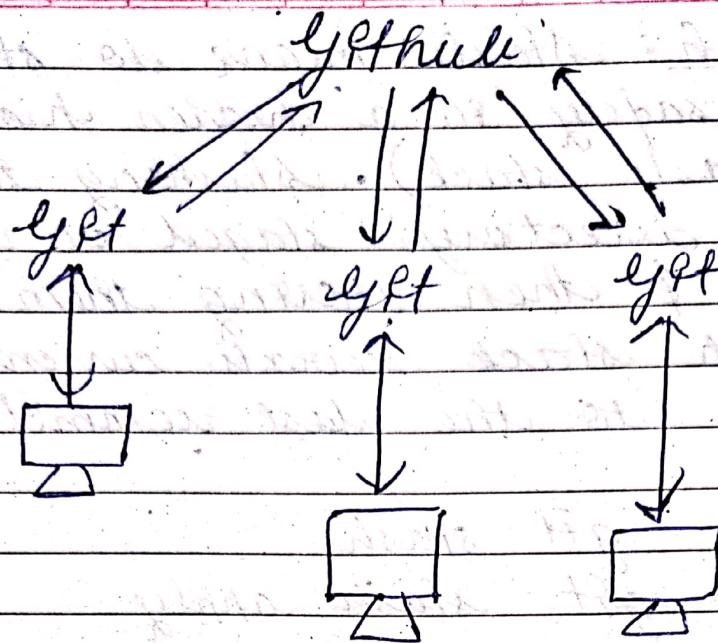
Installed & maintained on your local system.

git is the tool git is the service for projects that use git

Pushes or pulls data from central server

It's a core hosting platform for version control system.

Allows you to host a central repo on a remote server



git commit : Records changes to one or more files

- Assigns each commit a unique ID, called a SHA or hash, that identifies spec. changes

Syntax:

git commit -m "Commit message"
git commit -m "<Message>"

Open & merge pull requests:

- Notifies dev. about the changes you have made pushed to a branch in repo
- Act. & review changes.
- You can create a pull req. in 2 ways

Syntax: git merge <branch-name>

Configure existing project as Remote Repo.

- Change to directory where your repos are stored
 - Use ⇒ "git remote add" command
 - git takes 2 arguments
 - A remote name for e.g. origin
 - A remote URL, for e.g. https://
https://github.com/username/repo.git

E.g.

5) `git remote add origin https://github.com/`

ff Set a new remote

5 gibt demote -v

Verify new remote

Before doing this you need to first initialize a folder as git

- Create a new repo on GitHub
- Open git bash
- Change current working directory to local project directory
- Use "git init" to initialize

Syntax: `git add`

- Add files in new local repo

⇒ `git add .`

Adds file in local repo & stages them for commit

To unstage a file, use '`git reset HEAD YOUR-FILE`'.

- This stages files for the first commit

- Commit staged files

⇒ `$ git commit -m "First commit"`

Commits the tracked changes & prepares them to be pushed to a remote repository.

To remove this commit & modify the file, use '`git reset --soft HEAD-1`' & commit & add the file again

- Copy the remote repo URL

⇒ `git remote add origin <remote-repo URL>`

Sets new remote

⇒ `git remote -v`

Verifies new remote URL

Show a remote repo:

- "git remote" command is used to see which remote servers are configured.
- "-v" option shows the URL's that git has stored for the short name to be used when reading & writing to that remote.
- This command will list all the remote servers present.

Rename a remote repo:

- You can run git remote rename to change a remote's short name.
- It also changes all your remote-tracking branch names.
- For e.g. you want to change origin to repo1.

E.g. git remote rename origin repo1