

2019

WK 31 • 210-155

## Lecture-10

## Git and GitHub

JULY

29

Git

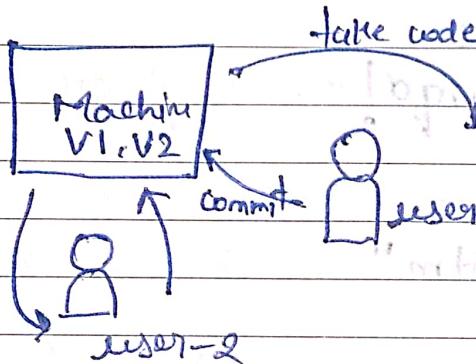
a version control system  
own on local Machine.

Version stored in local machine. Localized VCS

Centralised VCS

Distributed VCS

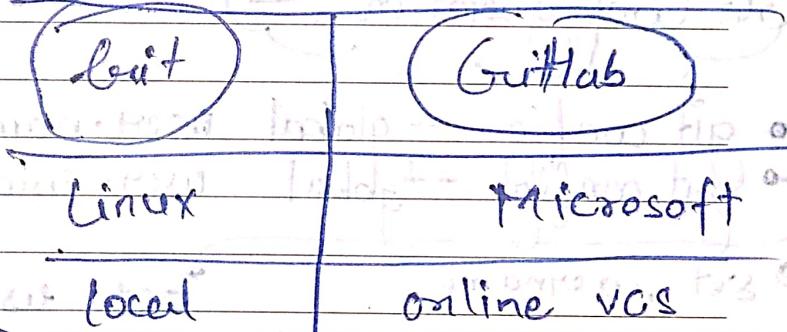
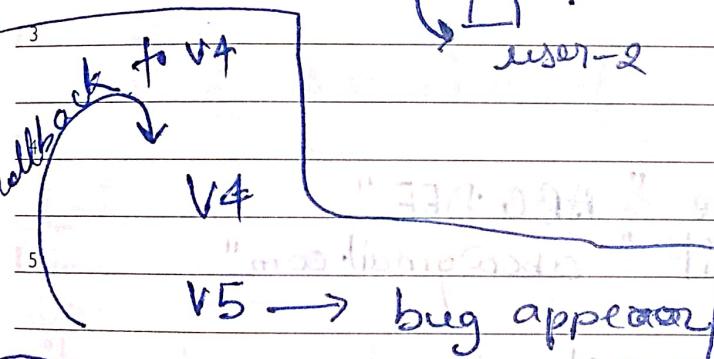
one single machine but multiple  
users can work on version.



git is distributed VCS.

uses clone step

with all versions  
and do change it  
and commit it.



pwd (print working directory)

ls (list files)

mkdir (make directory)

cd (change directory)

touch (create new file)

rm 'Hello.py' (remove file)

cd.. (go back one directory)

We will upload this steps to

2019 AUGUST

Mon	5	12	19	26
Tue	6	13	20	27
Wed	7	14	21	28
Thu	1	8	15	22
Fri	2	9	16	23
Sat	3	10	17	24
Sun	4	11	18	25

# 30

JULY - 9 AM - 'folder' (Delete folder)

TUESDAY

2011

WK 31 • 211

- `cd 'Hello.text'` (read file). → open file.
- `vi 'Hello.text'`
  - ↳ open file in editor in cmd.
  - 10 ↳ press i to insert text → edit text
  - ↳ if not want to save then → no save file
  - 11 ↳ press esc
  - ↳ write :q!
  - 12 ↳ to save your work → save file.
  - 13 ↳ press esc
  - 14 ↳ write :wq!
- watch html on browser
  - ↳ start "Hello.htm"

## Git Configuration:-

git config --global user.name "ABC DEF"  
git config --global user.email "abc@gmail.com"

set username      set usermail

- `git config user.email` (show email)
- `git config user.name` (show username)
- `git config --list` (show list of configuration)
- `git clear` (clear screen).

These steps are needed to setup git before initialization of git.

	1	8	15	22	29
Mon	1	8	15	22	29
Tue	2	9	16	23	30
Wed	3	10	17	24	31
Thu	4	11	18	25	x
Fri	5	12	19	26	
Sat	6	13	20	27	
Sun	7	14	21	28	

\* MOST AWARDED BRAND \*



2019

WK 31 • 212-153

Git repo works, only locally

JULY  
WEDNESDAY

31

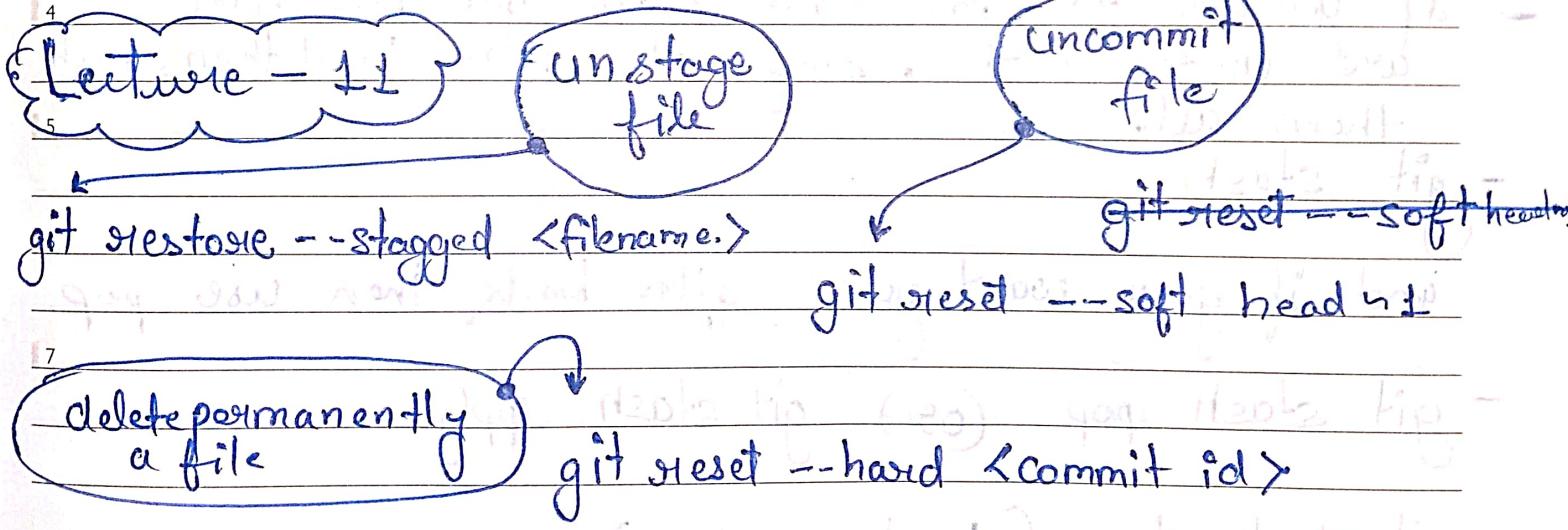
- git init (initialize empty git repo)
- ls -a (show hidden files and folders)
- git status (show current repo state).
- git add <filename> (stage the file)
- git add . (track all files at once).
- git rm -r --cached . (unstage the files)
- git commit -m " <your msg>" (final submit to git)
- git log (show stacktrace log).

deletion process

Head shows the newly committed log.

- git reset --hard HEAD^n (uncommit the latest commit)  
↳ this also removes the latest changes from files.

n is number of commit



01

AUGUST  
THURSDAY

Number of commits

2019  
WK 31 • 213-102

- git reset --soft head  $u_1$

- git reset --soft head  $u_2$

$\rightarrow$  --soft  $\rightarrow$  take the commit to the staging area.

- git reset --soft <commit Id>

$\rightarrow$  this Id is not going to be staged area but every commit before this will go to staged area. This id is excluded.

$\rightarrow$  --hard  $\rightarrow$  will delete those files permanently.

- if you want to show tree clean but few files are unstaged then stage the files and then stash them all

- git stash

$\rightarrow$  and if you want your files back then use pop.

- git stash pop (091) git stash apply

- git stash clear (to clear stash)

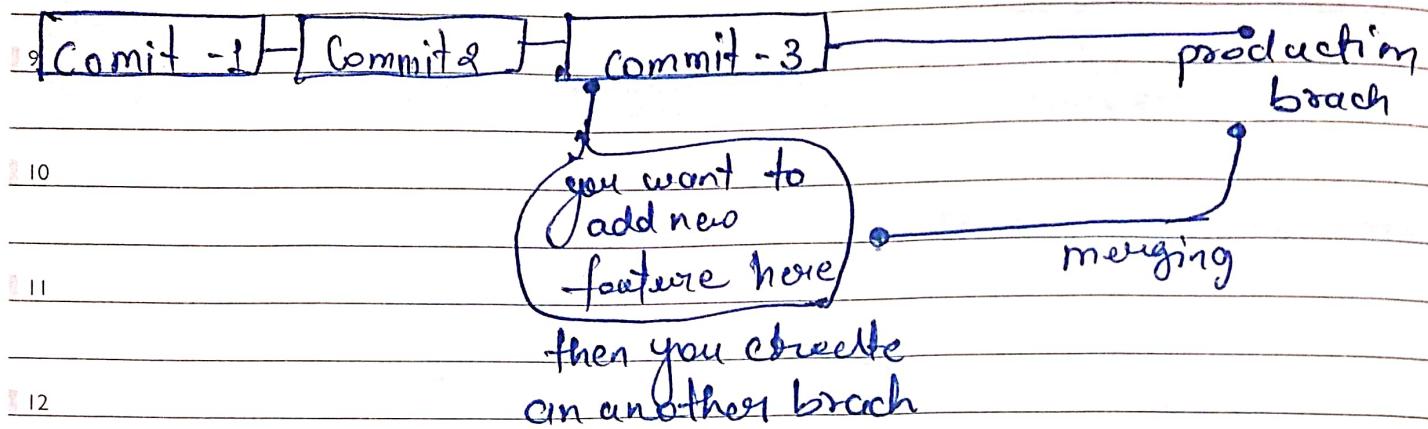
02

AUGUST  
FRIDAY

## git branches

2019

WK 31 • 214-151



Branch is a pointer that parallelly running and user make changes to new branch and then merge it with production branch

- `git branch` (show branches) \* Show current branch
- `git branch new-branch` (create new branch)
- <sup>4</sup> goto another branch
  - `git checkout new-branch.` (09)
  - `git switch new-branch.`
- <sup>6</sup> multiple command (;) using this b/w commands.
- <sup>7</sup> goto main branch to merge all branch
  - `git merge new-branch`
- Create and switch to new branch at once.
  - `git checkout -b new-2`

2019 AUGUST

Mon	5	12	19	X 26
Tue	6	13	20	27

if two people merging single file then merge conflict occurred. Here auto-merging will failed.

2019

WK 31 • 215-150

AUGUST  
SATURDAY

03

One will fix the issue, and then commit and then merge.

- delete a branch

→ git branch -d new-branch

and then commit and then merge.

gitbase → clean and manage the commits.

- git rebase -i <id of C4>

→ remove 'pick' with 's' (means squash)

→ ex- {pick C1}

{s C2}

{s C3}

want to merge this three commits.

C1
C2
C3
C4
C5

→ press esc

→ !x! + enter

→ enter msg by pressing **ins** button to insert.

3 commits merged

→ esc + **wq!** to save.

→ rebase done.

C1	C2	C3	C4

SUNDAY 04

gitHub → post a repo online by connecting git with gitHub.

- Create new **repo**

a directory online to contain **pushed codes**.

2019 SEPTEMBER						
	Mon	Tue	Wed	Thu	Fri	Sat
Mon	30	2	9	16	23	
Tue		3	10	17	24	
Wed		4	11	18	25	
Thu		5	12	19	26	
Fri		6	13	20	27	
Sat		7	14	21	28	
Sun	1	8	15	22	29	

05

AUGUST  
MONDAY

- git remote (check url of repo)   
 online repo url.

2019

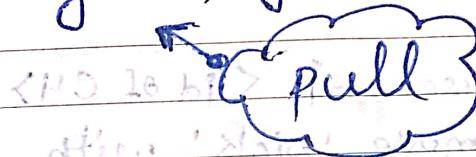
WK 32 • 217-148

- git remote add origin <remote url>  
→ a name for remote url  
this can be any name.

- git push origin master. (pushing code online)

Let's say you have added new file online on GitHub  
that doesn't created locally so, you have to  
make a pull request.

- git pull origin master



fetch → this is same as pull request but little  
different. pull do fetch then merge.

Here you fetch manually and merge manually then.

- git fetch origin master (fetch changes)

- git merge origin/master (merge with changes)

pull will do it automatically in one command.

fork → copy an online repo to your account.  
then clone it locally.

- git clone <paste repo url>

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
Tue																										
Wed																										
Thu	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
Fri																										
Sat	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
Sun	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29

2019

WK 32 • 218-147

after cloning → do changes → do commit →  
do push to main branch.

AUGUST

TUESDAY

06

After this you wants to contribute to original repo where  
you forked the repo.

- go to your online repo.
- click on contribute.
- click on open pull request.
- click on create a pull request.
  - write your msg
  - created pull request

bingoo !! you have contributed.