03/12/24

GIT

[Global Information Tracking]

VCS : version control system (or) SCM : Source code management

GIT : It is a source code management tool.

↳ where we can manage end to end source code.

GIT	GITHUB
• Peace of software • GIT is a version control tool.	• Central location (or) repository where code will be stored & managed.

*** Today Task:

✓ Install Git bash...

✓ To check version : git --version

• Installed git bash & checked version

• Ran the basic Linux commands in GitBash.

* Pwd : present working directory

⇒ To create files using Linux commands

* Touch file2

⇒ To create multiple files at a time

* touch file3 file4 file5

⇒ To clear the screen

clear (or) ctrl + l

⇒ To create folders

mkdir dir3

(make directory)

pwd
touch
mkdir
ls
ls -la
ll
ls -lrth
rm -rf
clear
history

① What is GIT ? Global information Tracker

- GIT is a distributed Version control system that is used to track changes in source code during software development.
- It permits multiple developers to work on a project together without interrupting each other's changes.

② What is a repository ?

A Git repo is a folder where Git tracks all the changes to your project files.

③ What is the difference between git init & git clone ?

"git init" develops a new Empty Git repository in the present directory, while "git clone" copies an existing remote repository, containing all files, branches & history on our local machine.

④ What is git add ?

"git add" stages changes (in your working directory), telling Git to include them in next commit.

⑤ What are key features of GIT ?

Distributed Version Control : Every developer has a full copy of repository, including the history.

⑤ Branching & Merging : Easy to create, switch & merge branches.

Lightweight & fast : Git is efficient, even with large Projects.

Data Integrity

collaboration

History tracking

undo mistakes

⑥ What are the different states in Git ?

- Modified : changes made but not staged.
- staged : changes prepared for the next commit
- committed : changes saved in the repo.

⑦ How do you delete a branch in Git ?

- Locally git branch -d <br-name>
- Remotely git push origin -d <br-name>

⑧ What is git pull & git push

git pull : It downloads changes from a remote repository & merges them into your local branch.

git push : uploads your local changes to remote repository.

⇒ To delete files & folders

rm -rf
(remove, recursively, forcefully)

Ex: rm -rf file1

⇒ We can view the contents in different views

ls
ll
ls -lrth
ls -la (to view hidden files)

⇒ change the directory

cd <dirname>

4/12/24

To open the file

cat <filename>
↳ Display the file contents

① ⇒ git init : initialise empty Git repository
↳ Global Information Tracker
↳ It will manage code, version control system (VCS)

② How to configure username & email :

⇒ git config --global user.name <name>

git config --global user.email <emailaddress>

How to move code from 3 phases :

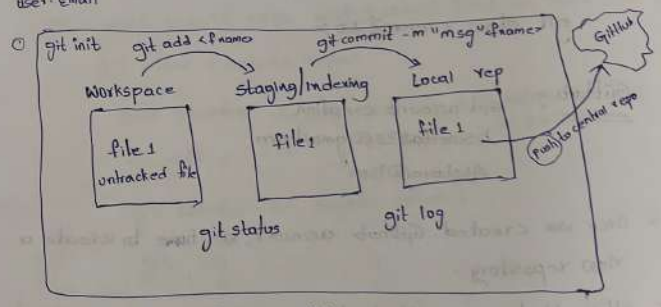
1) Workspace (file1) — If we want to move file to staging, git add <file name>
2) Staging / Indexing — git commit -m "msg" filename
3) Local repository

Finally we have to move central Repository which is in GitHub → use push

If we want to know the status then use

git status

② user.name
user.email



③ We have to create one file

touch java → it is in workspace stage which is untracked file...

After
To know the status enter

git status

④ We have to move the file in to staging using below
git add Java

⑤ After this we have to move local repo
git commit -m "my first commit" java

⑥ git status

⑦ If we want to know the history the, Enter below command
git log
↳ will get commit (or) tracking ID

Using this Tracking ID, we know who did this commit
git show <commit ID>

Github : Git account creation
bswetha235@gmail.com
[redacted]@1302

> Once we created Github account, we have to create a new repository.

> After that open git bash
git clone <Repo url>
↳ we get from github.
touch Swetha
git add . (or) filename
↳ means all files

git commit -m "This is swetha from BIS"
git status
git log
git show <commit ID>
git push
↳ It will create file in github account

# # # # # # # # #

• Master branch is the default branch in local Git, but in Github, it is main.

25/12/24 :

⇒ cloned remote repo and pushed files to main repo.
Git clone <remote repo>
Touch swetha → Edit this file with notebad (or) vi editor...
git add .
git commit -m "swetha commit"
git log
git pull

6/12/24 : Branching concept

Branch : copy of data where we can work on changes without affecting the orginal code.
↳ Why do we create branch means to Enhance the features or fix bugs without affecting the main code.

> Several types of branches are there ..

1) Main (or) Master : Primary & stable branch
2) Feature branch : for enhancement
3) Bugfix branch : to fix bugs in the project
4) Release branch : used to prepare for a new release
5) Hotfix branch : To Address critical issues ...

> New branch : git branch <br.name>
> Switch to a branch : git checkout <br.name>
> Create & switch to a new Branch : git checkout -b <br.name>
> List all branches git branch
> Delete branch git branch -d <br.name>
(or) git branch -D <br.name> → forcefully
> Merge a branch git ~~branch~~ merge <br.name>

> push a branch to remote
git ~~branch~~ push origin <br.name>
↳ reference to your repository

Today's 6th dec task:

1) Create a folder in central repo and add files to that folder. ...
2) create a branch in the local repo & merge
3) Create a branch in the remote repo without merge we have push to central repo.

9/12/24 :

Git conflict : A Git conflict happens when two people change the same part of a file, and Git doesn't know which change to keep. → Requires manual action.

> Commit ID also known as SHA (Secure Hash Algorithm), Commit Hash, checksum ..

Task : 1) Created merge commit ID
2) We can commit specific commit ID using cherry-pick command.
3) Git conflict

Git Merge : It combines changes from one branch into another branch to unify their histories.

> It is used to integrate features or updates ...

10/12/24 :

git branch -r
↳ to see remote branches

Head : is a pointer, it's pointing to current branch.

> Git ~~bra~~ checkout Swetha
↳ It is not moving Swetha branch because it has file & branch so git got confuse. So we have to use below command...
> Git checkout Swetha --

Fork : is a copy of an Existing repo that is created under your own account.
↳ used in open-source projects..
↳ used in collabarative development.

Task :

1) We have to merge 2 branches using by compare & pull requests.

2) fork (open source project)

11/12/24 :

Alias in Git : (→ Extra reference (or) shortcut) It allows us to create a custom commands that or Easier (or) faster to type.

* Replace long git commands with shorter ones..

Ex:

git config --global alias.st status
(st can be alias for git status)

How to remove alias ? command ?

git config --global --unset alias.st

To see all configured alias :

git config --global --get-regexp alias
↳ Regular Expression

How to protect our branch : unwanted accidental merges can avoid

Go to project repo settings → Branches (left side) → select add classic branch protection rule.

Origin : is a reference to central repo..

Amend: is used to modify the most recent commit, to fix a mistake (or) update the commit message.

* We can only change the latest commit message.

Ex:

Git commit --amend -m "add files"

(Amend commit)

↑ we can modify this

git commit --amend -m "add files"

(like I missed a file to add (or) change the message)

** Suppose, If we forget to add one more file to commit in previous.

git commit --amend -m "add file"

* Git commit --amend → doesn't create a new commit instead of it modifies the most recent commit.

*** git commit --amend

↓

This command doesn't create a new commit but modifies the previous one. (commit)

Revert: revert is a command used to undo changes from a specific commit.

git revert "commit ID"

yes, revert creates a new commit.

git revert is a command used to undo changes made by a specific commit by creating a new commit that reverses those changes. It keeps the orginal commit history intact. (unchanged (or) not altered)

> We can directly go from workspace to Local repo (we can miss staging command) only for modified files..

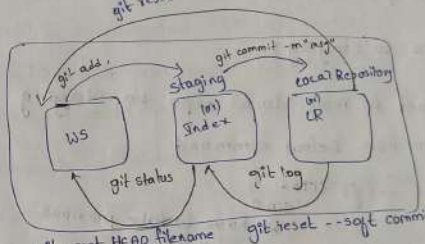
git commit -am "message"

(commit message) (not newly created files)

↳ automatically

This command will not work for new files, you must use git add for those first.

Reset:



Workspace → Localrepo

Yes, we can commit from workspace to local repo only for modified files not newly created files

touch file1

git add.

git commit -m "msg"

Vi file1

:wq! (adding code & exit)

git status → workspace to modify

No, need to git add.. We can directly commit as it a modified file

git commit -am "msg"

Git reset : is a commit to undo changes by moving the HEAD pointer to a specific commit. It can unstage changes or discard them entirely, depending on the type of reset (hard, soft, mixed).

Staging to workspace :

git reset Head <f.name>
↳ filename

If we want to move the commit from staging to workspace use above command.

Local repo to staging :

If we want to move local repo to staging then we use below command

git reset --soft <CID>
↳ (n-1) We have to take previous commit ID

Local repo to workspace :

git reset --mixed <CID>
↳ previous CID

git reset --hard
↳ It will permanently deletes uncommited changes

git reset --hard <CID> → permanently deletes that specific commit.
Previous commit

Cherry-pick :

> The git cherry-pick command is used to apply specific commit from one branch to another.

> It allows you to pick individual commits from a branch & apply them to your current branch without merging the entire branch.

git cherry-pick <commit ID>

> What is a detached HEAD ? → this happens when you checkout a commit or tag directly

A detached HEAD occurs when HEAD points to a specific commit instead of latest commit in a branch.

> What is git rebase ?

It is a poweerful feature that allows you to integrate changes from one branch into another branch, creating a linear history..

> Difference between git merge and rebase ?

The main difference between git merge and git rebase is how they integrate/combine changes from one branch in to another.

- Git Merge : It combines branches by creating a new commit, preserving (keep) the history of both branches
creates new merge commit

Git Rebase : It moves or rebase the commits of one branch on top of another, creating a linear history. clean & linear history
No merge commit ID creates
rewrites the commit history

Use case :

- Use merge when you want to preserve the complete branch history.
- Use Rebase for a cleaner, linear history.

> What is a branch in Git ?

Branch is a copy of data where we can work on changes without affecting the orginal code.

> How do you rename a branch ?

Rename the current branch :

```
git branch -m <new-name>
```

Rename another branch :

```
git branch -m <old-name> <new-name>
```

> How do you compare branches in Git ?

Use 'git diff' command

```
git diff <br.1> <br.2>
```

> What is the purpose of the git branch --merged command ?

This command list all branches that have been fully merged in to the current branch.

> How do you view the commit history ?

```
git log
(or)
git log --oneline
```

Dec 13/12th :

TAG : Where we can mark important milestone..
→ reference (or) information (or) marking the milestone
↳ label

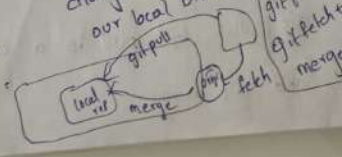
Git Rebase :
→ history will reapplied & altered
→ there is no merge commit id
→ combines changes from one branch to on top of another.

git pull
↓
commit ID
Vi Editor open avuthadi

git pull --rebase
↳ clean history
↳ no commit id

git fetch
It will downloads the changes from remote repo but not merge in local branch

git pull :
It will download & merge the changes from remote repo to our local branch



· gitignore : Specifies which files & directories Git should'n't track.

gib clean : It permanently deletes untracked files.

git commit --amend : It allows us to update the most recent commit by including additional changes or modifying it's commit message.

It doesn't create a new commit but replaces the last one.

⇒ Difference b/w Git fetch & Git pull ?

Git fetch : It downloads updates from the remote repository, such as new commits, branches & tags.

However, it does not automatically merge those changes in to your current branch.

Then you have to choose to merge or rebase those changes manually.

Use case : you use git fetch when you want to see what changes are available from the remote repository before you integrate them in to your local branch.

git pull : Git pull is a combination of git fetch and git merge.

It will download the changes from the remote repository and automatically merges them into your current branch.

⇒ What is a git commit ?

git commit is a snapshot of your changes at a particular point in time.

⇒ What does the 'git status' command do?

git status shows the current state of your working directory and staging area.

- which files are modified, but not yet staged for commit.
- which files are staged and ready to be commited.
- which files are untracked, meaning they are not being tracked by Git yet.

**** If there are changes on the remote branch that you don't have locally, and you try to push your changes, Git will prevent the push & prompt you to pull first, to avoid overwriting remote work.

> What is a git clean?

git clean is a command used to remove untracked files and directories from your working directory.

- git clean -n
  ↳ It previews what will be removed
- git clean -f
  ↳ Remove untracked files
- git clean -fd
  ↳ Remove untracked directories
- git clean -fx
  ↳ Remove Ignored files.

** The files and directories removed by git clean cannot be recovered easily. So use the -n option to check before running.

GitHub: Webbased Repository Hosting service.

16/12/24

Git stash: Temparorily storage (or) Store Something safely.

If we're working on one task suddenly we got another task which is priority but the previous task was at 50% done then we can store this uncomplete task is in stash.. complete the priority task & bring back the task in following ways:

- .git/refs/stash

⇒ git stash save "label name"

apply	pop	drop
git stash apply	git stash pop	git stash drop
↓	↓	↓
file will be there in stash & our work directory	file will not be there in staph, it will come in our staging stage	file will be deleted from stash... (Removed from stash, it will not move to our staging area)
(copy & paste)	(cut & paste)	

→ git stash list

git ignore: git shouldn't track, specified files & folders..

⇒ Centralised Version control System (CVCS)

- Ex: Svn (Subversion)
- user directly do updates/changes will made on central repo.
- Requires internet
- If it is down, then we don't have any copy...

Distributed VCS (DVCS)

- Ex: Git
- We are cloning the remote repo locally, do changes are made in local after then we will push to RR.
- No need internet (internet required only when we clone & push)
- We can work distributedly..
- Even if it is down (or) deleted we have code in local.