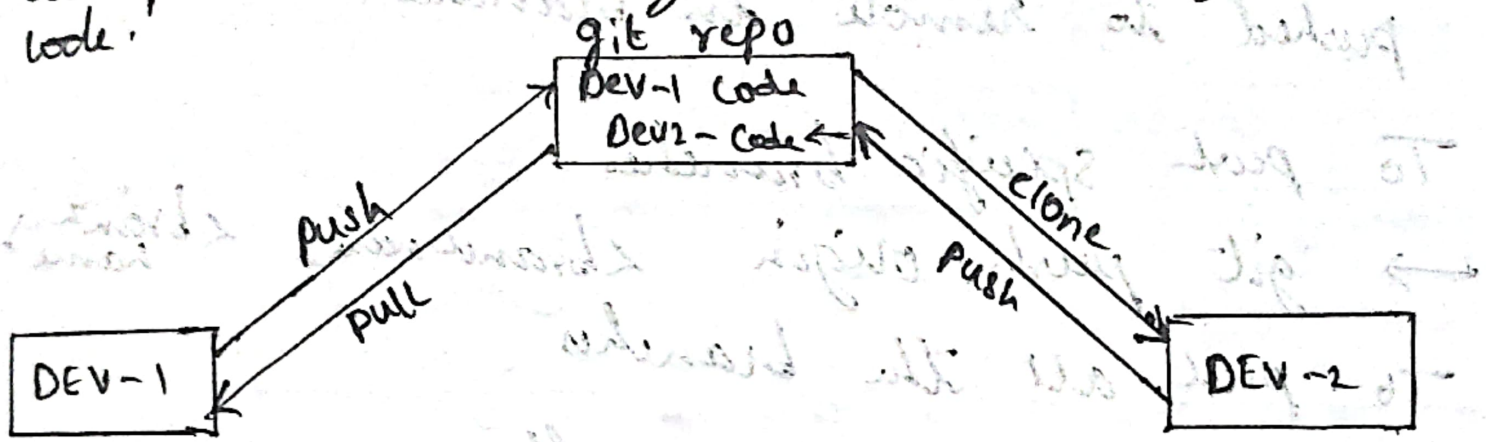
GitHub

→ central
→ cloud based platform for version control that allows developers to store, manage and track changes in their code.



Push :- upload local commits to remote repository (GitHub)
git push origin master

Pull :- fetch updates from a remote repository and merges into local branch
git pull origin master

clone :- creates a copy of existing remote repository on your local machine
git clone https://.....

Local to remote connection using HTTPS

→ git remote add origin <https>
only one repository at a time.

For every push it will asks the credentials for https. For SSH it wont.

To check which repository we are connected

→ git remote -v

To push our local repo/commits to remote

→ git push origin <branchname>

it will ask you username and password

password will be tokens

Settings → developer settings → Token (classic) → Generate new token → classic → enter token name

Select repo box → Generate token

Now you can see the branch that you have pushed to remote in branches

To push specific branches

→ git push origin <branch-name> <branch name>

To push all the branches

→ git push origin --all

if we want the changes which is done by another developer to local

→ git pull origin <branchname>

For pull, in public wont ask password.

Pull = Fetch + Merge

Fetch = downloads the latest changes from the remote repository but don't merge.

Merge = After fetching, git automatically merge the changes into your current branch

Session - 14 28/03/25

To get remote repository to our local
→ git clone <https URL>
if it is public it download automatically, if it is private have to enter credentials (tokens)
when you download the repo from the github, you will be main branch bydefault.
If you want to see all the branches for that repo.
→ git branch -a
Now you can see other branches and you can checkout to them.

* For Fetch → git fetch origin <branch name>
For merge → git merge origin <branch name>
Pull = Fetch + Merge git merge origin/main

connecting github to local through SSH
we will do it by SSH-keygen which will downloads both public & private keys.
In our local we will have private and will place the public-key into git-hub then the connection is established and it wont ask credentials for every push

In root
→ SSH-Keygen
→ ll -a (.ssh will have pub & pri keys)
place the public key in Github
Goto → github → settings → on left SSH & GPG keys → SSH keys → New SSH key → name & place the key → Add SSH → it will ask password

To disconnect repo → git remote rm origin

you can directly add your files inside your repo in github

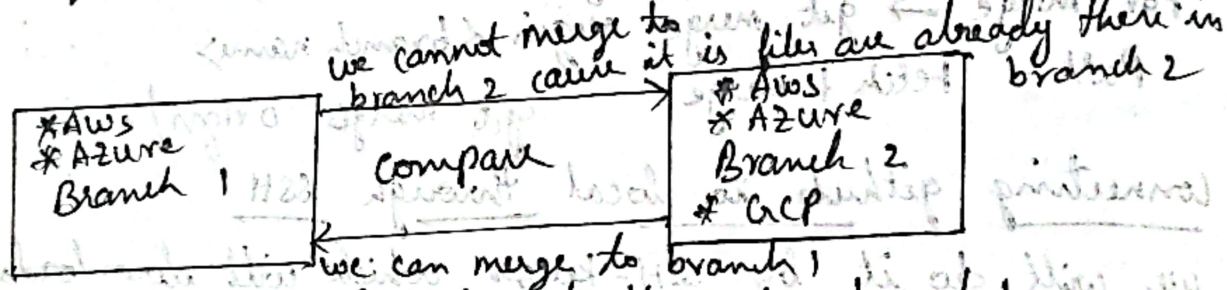
go to repo → add file → upload (or create files after selecting files from your pc → commit

Pull Before Push

means whenever you want to push into github you have to pull the changes to local first and then push your changes to github.

Both should have balanced changes

In git, we use merge to get the changes from one branch to another branch. In Github after the commit, compare & pull request



Click on compare and pull request

base → destination
compare → source

we can see a message that able to merge

and click on create pull request → merge PR → commit

you can also raise a pull request in PR Icon

when merging what if conflicts arised ?

Same scenario like having same file but different data and try to merge. it will show you that merge conflict. & even though you can ~~merge~~ create PR.

click on PR → Resolve Conflicts → do the change in file → mark as resolved → Commit merge → merge pull request → Commit merge

you can delete and restore the branch in branches section (you cannot restore branches after deletion after some time).

Repository settings
↓
danger zone
↓
* Change visibility → Public (or) private
* Transfer ownership → we can transfer the ownerships to others.
↓
specify an organization (or) username (github id)
they will receive a mail
you want to undo → Abort Transfer

* Archive
* Delete

To Rename a Repo
click the Repo → Settings → you can Rename

29/03/25

Session - 15

Fork:- copying of a repository. (only public repo)

Search the username → go to user's → repositories → select the repo → fork → give a name & untick copy the main branch only → Create fork
user will know that repo is forked.

Git Tags :- mark specific points to group of commits.

→ git tag -m "message" <message-tag/tagname>

To view list of tags

→ git tag (or) git log

To know what is that tag

→ git show <tagname>

we can also push the tag

→ git push origin <tagname>