

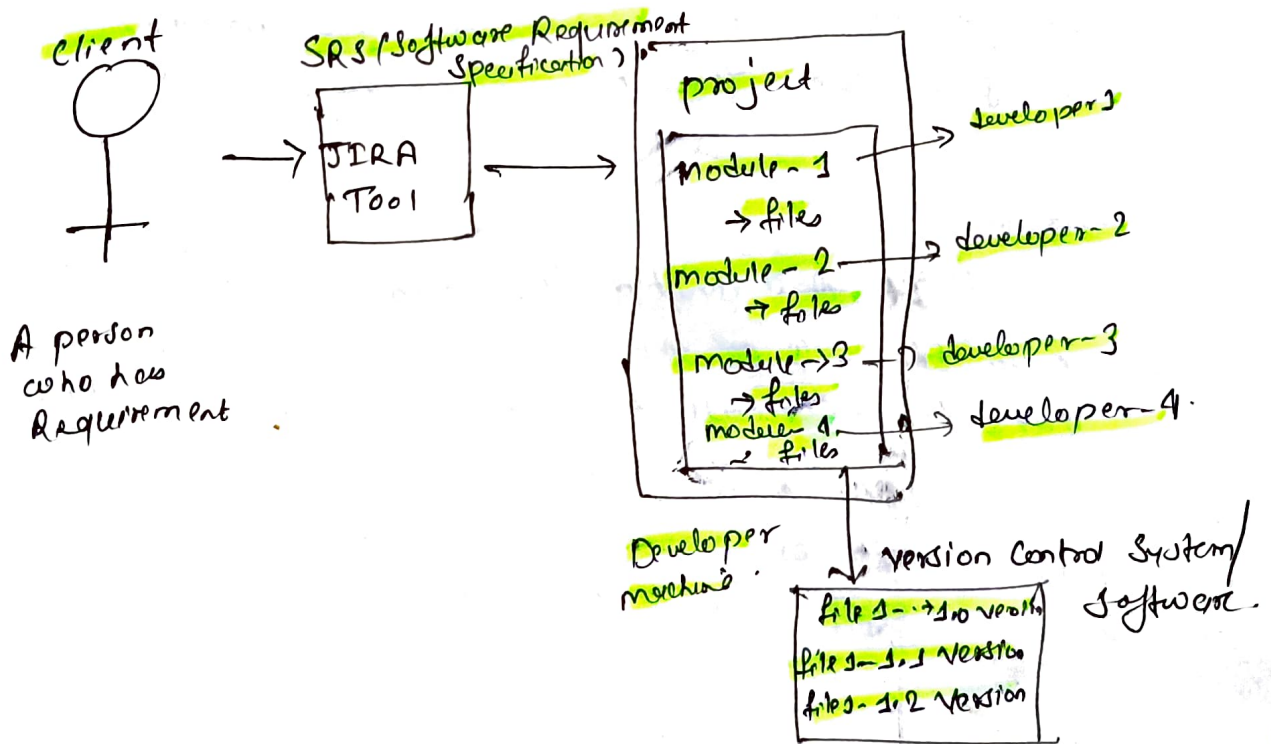
Gift
Hub

Introduction to Git ..

→ Git is a popular version control system (VCS).
→ It was created by Linus Torvalds in 2005
and it is maintained by Junio Hamano.

Git is used for .

- (i) Tracking Code changes.
- (ii) Tracking who made the changes like history of files.
- (iii) Coding collaboration.



→ Suppose there is one client who has the requirement to make the software. So, he will tell his requirement to the developer and the developer will write the code.

According to the Client Requirement, in developer machine there will be different modules of project which is handled by different developers. and developer has to ^{do} changes according to client requirement. So, "Keeping track of changes made to the file by the developer as per the changes made by the client requirement would be difficult in developer machine. To resolve this problem we need to use 'Version Control System'."

Notes

Version Control System \rightarrow It keep track of all the changes made in code, who has made changes, when the change is made etc.

Definition:

"It is a system that record changes made to the file or set of files over the time, so that we can recall the specific version later."

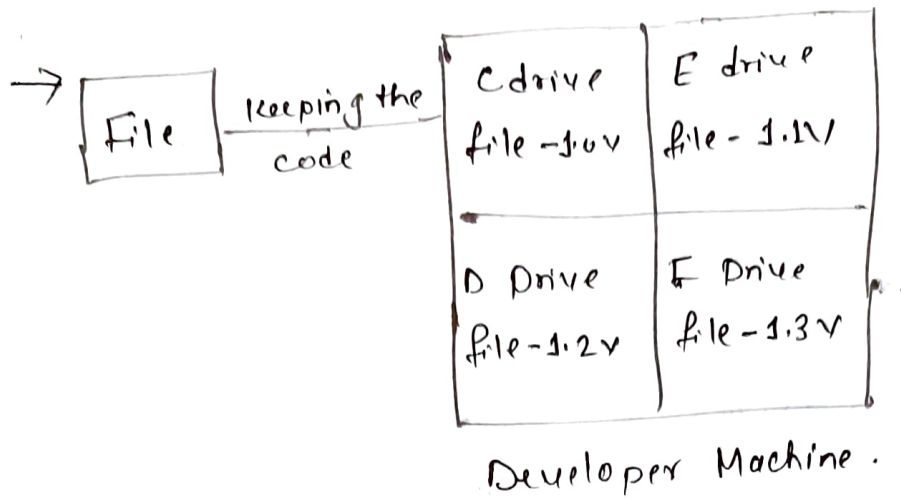
i.e. for every source code change in a file a new version will be created.

eg: JDK 1.0.4, JDK 1.1.4, JDK 1.2.4...

* Type of VCS

- Local Version Control system
- Centralized Version Control system
- Distributed Version Control system.

a) Local Version Control System.



→ In this we keep our files locally in our system and following problems may occur.

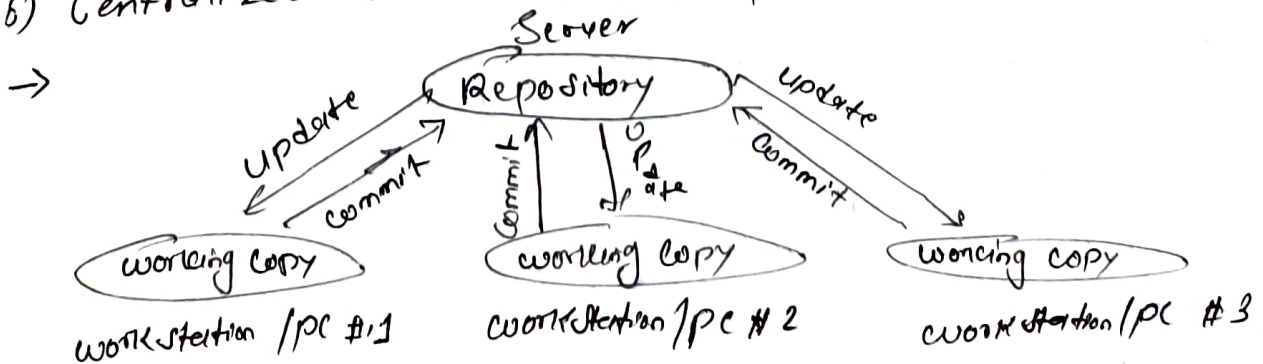
1. It is easy to forget in which driver you are in and accidentally write the data to the wrong file or copy from other files.

2. If the hard disk is corrupted there would be a possible loss of secured data.

3. By mistake we can delete few files also.

So, to overcome this problem "Centralized Version Control System" came into picture.

(b) Centralized Version Control System?



Benefits:

→ Developers can collaborate the code in one repository and do the change.
eg:- SVN, Subversion.

→ It has a single server that contain all the server files.

→ More no of developers would connect to CVS to checkout the file.

Note:

Checkout → taking the code from repository to local machine.

push → sending the code from local machine to repository (CVS)

Advantage:

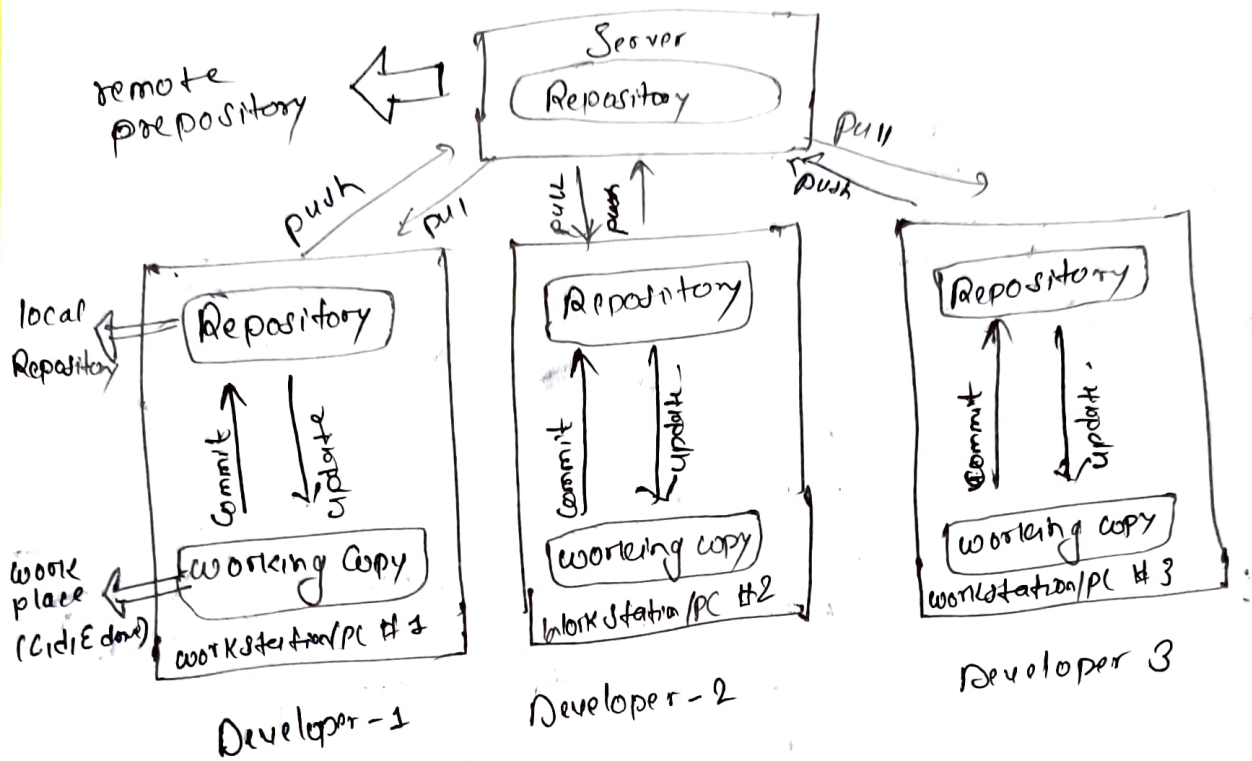
1. Everyone know to certain degree what everyone else on the project is doing.
2. Administrator will have full control over which can do what and it is easier to manage.

Draw back:

1. Single point of Failure (SPF) would represent the centralized system.
2. If the server goes down due to network traffic, during that hour nobody can collaborate at all or save changes to the server.
3. If the hard disk of the centralized system gets corrupted and proper backup haven't been taken then there is every possibility of loss of data.

Single point of Contact.

Distributed Version Control System.



→ In this a developer is working in a system and he will commit the code in local repository and from there it will go to remote repository.

So,
(i) If the ~~remote~~ main server goes off, still there is a local repository which would have maintained the copy of the repository where the entire code is available (history of versions).

(ii) If the remote repository is down, developer can do changes in the local repository and when the main repository is up the code can be pushed to remote repository from local repository.

Note: In LACS and CVS getting up the complete history of changes is not possible. It is possible to get only the latest version, but not the entire history.

Push will not happen with version rather than push will happen only with the latest change.

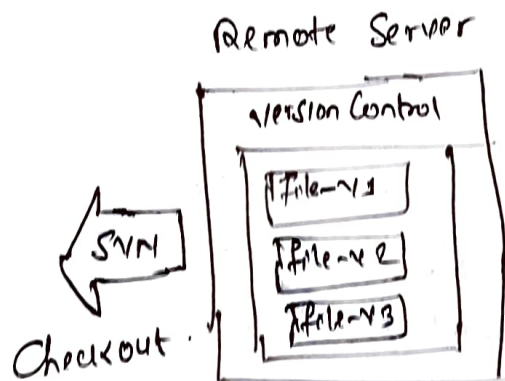
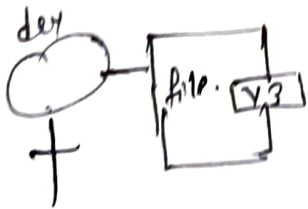
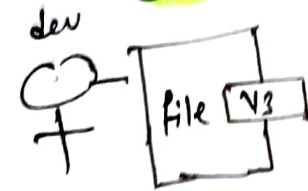
Version history:

file \rightarrow 1.0 v

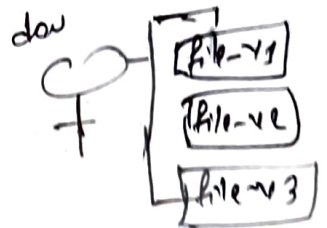
file \rightarrow 1.1 v

file \rightarrow 1.2 v

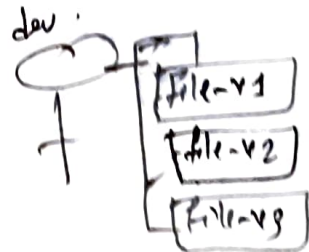
SVN VS GIT



CVS



CVS
pull or
clone



Installation of git Software.

1. Download a git software from the following link.
<https://git-scm.com/download/win>

There are two types of Git Software.

1. Git Server.
2. Git Client.

Git Server .

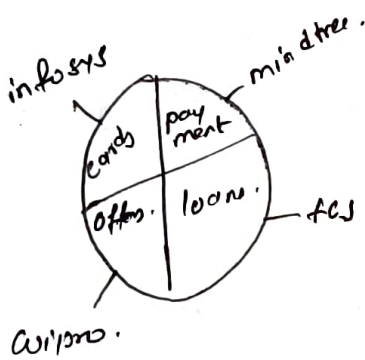
- It is a repository .
- It is a largest host of source code in the world .
- It is used to store/manage the source code of the project .
- Some of the Git server tools are : Github, BitBucket .
GitLab etc .

Note :

Github

It is a server that holds large number of projects .

GitServer (Github, GitLab)



How to connect to github?
url → `http://repo.citibank:9999`
url →
Username →
password →
git url will be same for all the developers but username and password will be different

Cards (source code)	payment (source code)
offer (source code)	Loan (source code)

not allowed

Dev 1: `http://repo.citibank:9999/project/card`
Username: nithin
pw: xxxxxx

Dev 2: `http://repo.citibank:9999/project/card`
Username: kamaaluddin
pw: xxxxxx

Note : When we join company team leader or managers will share the url, username and password. Every developer will connect to git server and get source code from the git

server and do the changes locally and then move the code from the local repository to the main repository with the version.

Git server physical location where it is installed can't be seen, it would be installed on cloud platform like AWS, Azure or on any data centre.

Where. Should we provide url, username, pw?

↳ To type these details we need git client.

Git Client.

↳ It is a tool which is used to connect to our git server.

If we install git client (git cli) we get the following tools for free.

- a. git bash. \Rightarrow linux command are required.
- b. git gui \Rightarrow Graphical user interface where all the actions will be done through clicks
- c. git cmd. \Rightarrow Command line tool where developer should provide url, username and password.

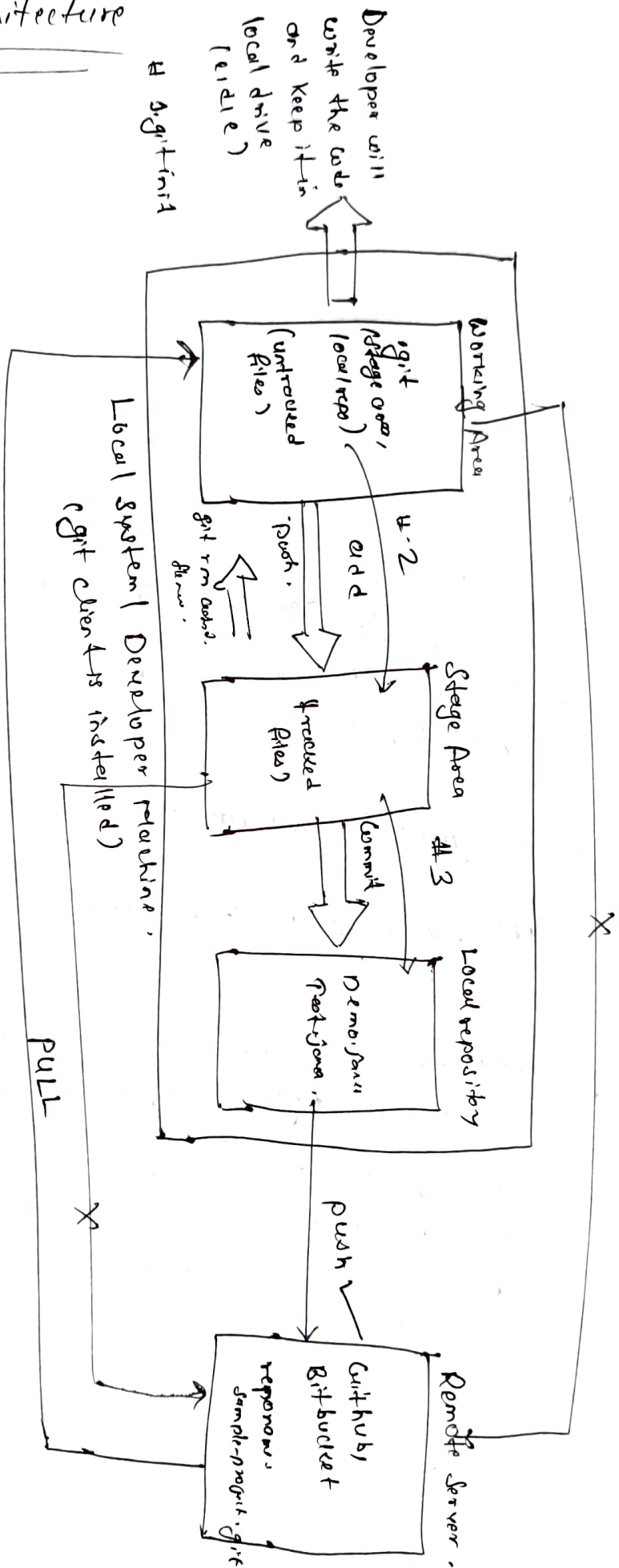
Git VS Github.

git \rightarrow client tool

github \rightarrow server software where repositories will be maintained.

Git

Architecture



Explanation:

A developer will write a code in his local machine, and that code ~~is~~ we ~~do~~ cannot push to remote repository. In order to ~~code~~ ^{push} we ~~has~~ need to add code to stage area. Again from stage area we have to push the code and to local repository and local repository ^{commit} to the remote repository. But we can directly pull the code from remote server to our local system by using pull command.

Note:

git → Client tool where the client should provide url, username, and password.
github → Server software where repositories will be maintained.

There are 3 regions in GIT architecture:

- a. Workplace → It is a place where developers maintain the source code.
- b. Stage area → Once the code is ready, then it will be added to stage area (indication to git software).
- c. Local repository → Once the code is in the stage area, we commit it to the local repository with some standard message, from local repository we repush to main repository by providing url, username, and password.

Git Commands. (They are case sensitive.)

1. git version
2. git help.
3. git config.
4. git init
5. git clone.
6. git add
7. git status
8. git rm
9. git restore
10. git commit
11. git log
12. git show
13. git push
14. git push pull
15. git branch
16. git checkout
17. git stash.

also: git merge, git rebase

1. git version

↳ This command is used to check the version of git.

Syntax: git --version.

2. git help

↳ If we want to see the list of commands, then we can use git help.

Syntax: git help .

In order to get help of specific command we can use like this .

git help init

It will give documentation of init command likewise this it will give documentation of all the command.

3. git Config .

↳ It is used when the git software is used for the first time .

↳ This command will set the developer identity like name , email id . . .

↳ This configuration information will be used by git software for every push operation encountered .

Note:- git config --list --show-origin .

> git config --list // this command is used to provide the list of configuration .

// to set the username and email .

> git config --global user.name "Kamaluddin - Ansoni"

> git config --global user.name email "Kamaluddin981@gmail.com" .

Note:- global ⇒ It indicates the user can work with git commands from different drives of computer .

important operations associated with git .

git init

⇒ Normally a folder will be created in the developers workplace and inside the folder the source code would be place .

⇒ This command literally internally creates one folder called .git .

⇒ .git is used by git software to identify the folder which should be participate in pushing to "local" and "remote" repositories .

⇒ normally this is the first command which we execute to setup the git for operations like clone, push, pull .

Syntax : `git init` .

Note: To change the directory use this command .

Cd D : Syntax: `cd _____`

To check in which directory we are currently present we use pwd command .

git status

→ This command is used to check the status of the working directory

01 Untracked files (red colour) ⇒ It means the files are present in working area and these files cannot be committed to "local repository" or "remote repository" .

b. tracked files \Rightarrow It means the files are moved from working (green colour) area to stage area so, these files can be committed to "local repository" and to "remote repository".

Note:

It is also possible to unstage the files from staged area to workspace, using the following command.

Syntax: git rm --cached <file-name>.

* To send the ^{specific file} code from workspace to stage area we use following command.

Syntax: git add <file name>

If we want to push all the files from ^{working} stage area to ~~working place~~ stage area, we use the following command.

Syntax: git add.

git add - - a.

c. modified files (red colour).

\Rightarrow It means the files are present still in working area and these files can be staged or it can also be restored back to the normal phase.

Syntax: git restore <file name>

^{to} ^{old}
This is used to restore the file

* The files which are created to commit should be in stage area, to perform commit operation we use following command.

Syntax: git commit -m <some-message>

eg # 1. `git commit -m "first commit"` // This file commit all the files present in stage area.

eg # 2. `git commit -m "second commit" file name`
↳ This will commit only that file into local repository.

Steps followed to create a git repository and push it to remote repository.

1. open github.com by providing credentials.
2. Create a new repository and enter some name (repository name) and click on the Create repository.
3. To perform push operation we need to use the following command.

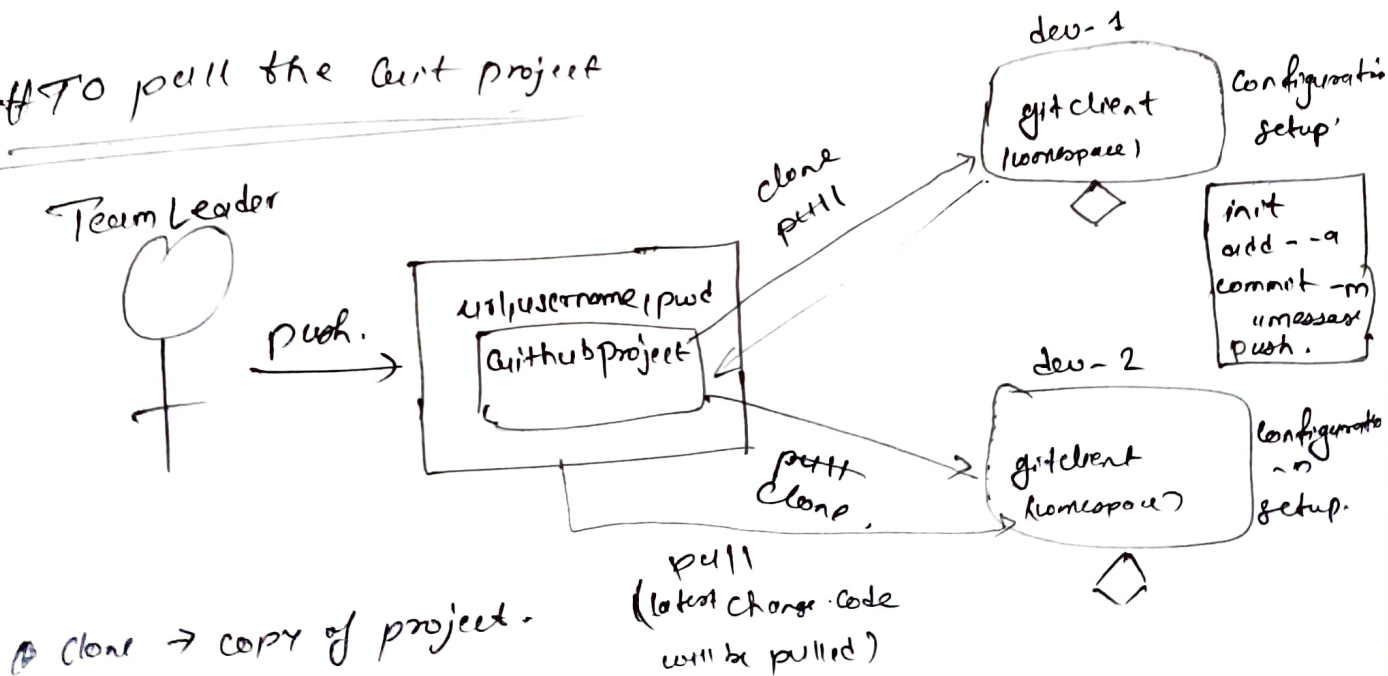
`git branch -M main`

`git remote add origin (URL you need to give URL)`

`git push -u origin main`

This will show in URL

How to pull the Git project



↳ In order to complete the pull the project from remote repository. First we will push done the project and do some changes, and again we will follow the same process and again the developer can pull the updated project and make some changes.

Notes:

git pull → It is used to fetch the latest changes made in remote repository to working directory.

vs

git clone → It is used to clone the repository to the working directory of the developer.

Syntax: `git clone <URL>`

Syntax: `git pull`.