GIT & GITHUB

(PART 1)

Date 10/10/2022

⭐ GIT Topic:-

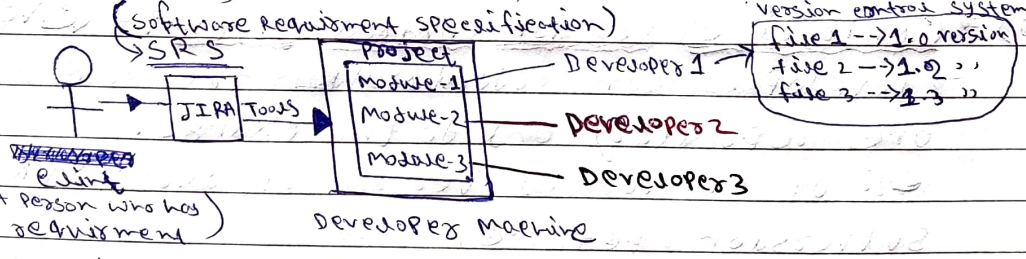
1) Introduction, 2) What is version control system(VCS) and types of VCS, 3) Git software installation, 4) Git project Architecture, 5) Git commands operation 6) Git commands execution (a) using command line, b) using IDE's like Eclipse/STS/IntelliJ Idea (Spring Tool Suit (STS)) 7) Github account creation (a) Public repository creation b) private repository creation, 8) Git folder structure 9) Git Branching strategy. (a) Developer branch, b) master branch, c) release branch, d) hotfix branch, e) fork branch, b) Project Review Process (PR process), code reviews, code merge, (11) Realtime problems with Git & how to fix them (12) FAQs

1) Introduction:-

⇒ GIT:- Git is stands for version control system(VCS) It was created by Linus Torvalds in (2005) and it is maintained by Junio Hamano.

⇒ GIT is used for:-

a) Tracking code changes, b) Tracking who made the changes like history of files. c) coding collaborations.



⇒ So, keeping track of the changes made to the file by the developer as per the changes made by the client requirment would be Difficult in Developer machine.

• To resolve this problem we need to use "version control system(VCS)".

2) ⇒ What is VersionControlSystem (VCS)?

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

ie, for every source code change in a file a new version will be created.

eg:- JDK 1.0V, JDK 1.1V, JDK 1.2V ....

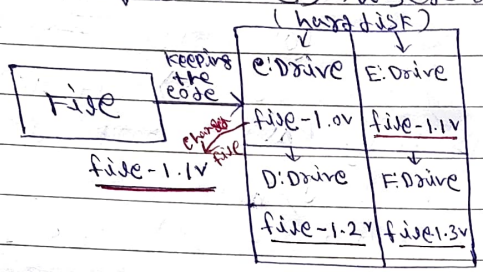
Spring 1.X, Spring 2.X, Spring 3.X

2) ⇒ **Types of VCS:-**

There are 03 type VCS → a) Local, b) Centralized c) Distributed version control system.

a) ⇒ **Local version control System (LVCS):-**

→ It is used to maintain the file version and retreive the files based on the specific version.



* our basic file change, rename in our own device and save it to our own drives is the ex. of LVCS.

Drawbacks:-

1. It is easy to forget in which driver you are in and accidently write the data to the wrong file or copy from other files.
2. If the harddisk is corrupted there would a possible loss of secure data.
3. By mistake we can delete few files also.

To overcome the drawback of LVCS system we have "Centralized version control system"

b) ⇒ **Centralized version control system:- (CVCS):-**

→ Developers can collaborate the code in one repository and do the change.

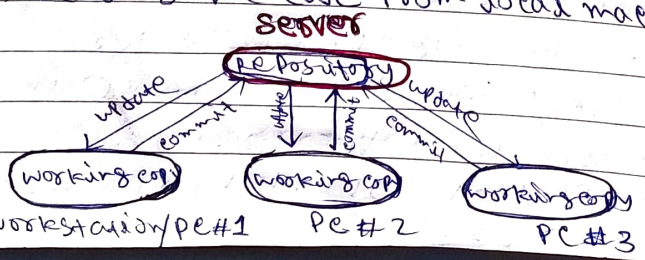
eg:- of centralized version softwares: SVN, subversion, peforce, ...

→ centralized version server will have single server that contains all the version files.

→ For many years this has been the standard version control system. (Get latest version not other version)

→ More no of developer would connect to CVS to checkout the files.

comment ⇒ checkout → taking the code from repo to local machine
push → sending the code from local machine to repository.



Advantage:-

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

Drawback:- Because, no backup avail

1. Single point of failure (SPF) would represent the CVCS.
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 there is every possibility of loss of data.

c) ⇒ Distributed version control System (DVCS):-

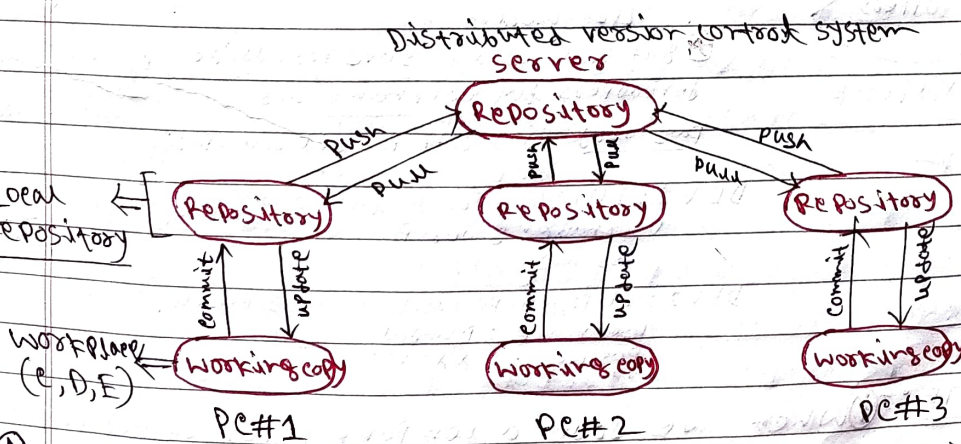
→ Eg: Git, Mercurial, Darcs, Bazaar, etc.....

→ If the main server goes off, still there is a local repository which would have maintained the copy of the repo where the entire code is available (history of versions)

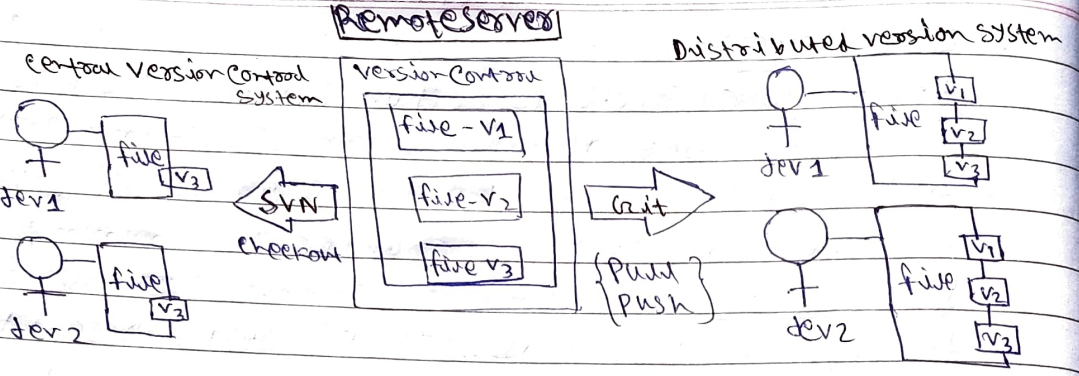
→ If the remote Repository is down, then developer can do changes in the local Repo and when the main server/Repo is up the code can be Pushed to remote repo from local Repo

⇒ Developer not only get the latest version but also the complete history of the files

⇒ Push will not only happen with latest version/ snapshot of the file rather they will push the old files also.



NOTE:- (i) In Lves CVCS setting up the complete history of changes is not possible. (ii) It is possible to get only the latest version, but not the entire history. Eg: SVN, (iii) Push will not happen w.r.t version Push will happen only with the latest change.



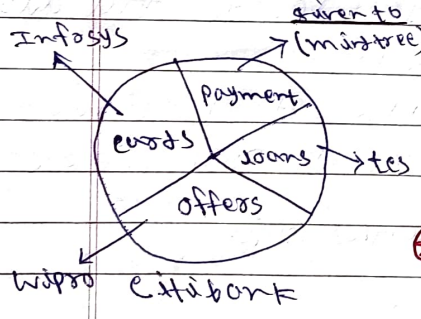
3) Installation of git software:-

1) Download a git software from the following link
   https://git-scm.com/downloads/win
2) There are 2 type of git software
   1) Git server
   2) Git client

⇒ Git server:

→ It is a repository
→ It is the 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, ...

- Suppose we are creating a project below ↓



Q) How to connect to github?
Ans:- You have to connect with
- url → http://repo.citibank:9999
- username →
- password →

GitServer (GitHub, GitLab)

cards (source code)	Payment (source code)
offers (source code)	Loans (source code)

* git url will be save for all the developers but username and password will diff for all developers:

DEV 1: http://repo.citibank:9999/projects/offers
username: swaroop
password: xxxxxx

DEV 2: http://repo.citibank:9999/projects/offers
username: mandal
password: xxxxxx

* NOTE: (i) When we join a company team leader or manager will share the url, username, password. Every developer will connect to git server and get source code from the

Git ⇒ client tool → Git server / Git client [where the client should provide url, username and password]

Github ⇒ server software where repositories/projects will be maintained online.

Git server and do the changes locally and then move the code from the local repository to the main repository with the versions

(ii) Git server physical location where it is installed can't be seen. it would be installed on the cloud platform like AWS, Azure or on any datacenter. this why we use http:///

Q) Where should we provide url, username & password?

Ans:- It is a TO type these details we need Git client.

⇒ Git client :-

→ It is a tool which is used to connect to our Gitserver

→ if we install Git client (Git S/W) we get the following tools for free →

a) Git bash, b) Git gui, c) Git cmd

→ Git client is a .exe file which can installed with just few clicks. (Goto search bar on windows > type Git → you can see Git → bash, cmd, Gui.

⇒ Practical start ⇐

(*) Now open Git bash → and linux commands are required.

→ Git Gui ⇒ Graphical user interface where all the actions will be done through clicks.

→ Git CMD ⇒ command line tools where developer should provide url, username, and password.

(*) When you open Git bash you see the default location to :→ C:/Users/Swarup mandal

Linux/Bash commands :-

→ How to change directory in Git bash?

→ pwd → shows the current directory address.

→ cd D: → For change directory (folder location) use cd command and cd D: means change directory to D driver of my computer

→ copy & paste in bash :- (i) you can't copy and paste anything to bash by keyboard shortcuts.

(ii) you have to do right click and then select copy, past...

(iii) If you copy and paste by Right click by defult some time you get a →(^[[200~ ..... ~). If you ↵ it

it will show error. So, Remove it carefully.

(4) → cd E:/Swarup Mandal ⇒ bash show to many arguments Error. Because in bash space of any thing will understand a different argument/ commands. How to resolve it?

⇒ (1) cd E:/ swarup\ mandal → By a Backword Slash or word ent or space.

(2) cd ① E:/Swarup mandal ② or cd ⑪ E:/ Swarup mandal ⑫ → By using Single Quote ' or double Quote or Both side

→ ls ⇒ used to list directories or files.

ls -a ⇒ List of all files in the directory including Hidden files

ls -l ⇒ List of all the file long listing and there size in the current directory.

ls -la ⇒ combination of -a and -l Both together

→ rm ⇒(remove) is used to delet files. rm <filename>

rm -i <file name>...[Option] ⇒ Remove file in the directory but let user confirm before deleting it by yes or no option in commandline.

rm -r ⇒ Remove non-empty directories including all the files with in them.

rm -f <file name> ⇒ Remove files or directories without prompting even if they are write-protected - the f stands for force

⇒ mv ⇒ (move) → used to move one or more file or file from on location to another location.

syntax ⇒ mv [Options/files] [source destination/]

⇒ Rename file name:- mv file1 file2

⇒ cp (copy) ⇒ lets you copy files or directory within the file system. Syntax:- cp [Options] source Destination

copy files ⇒ cp file1.text file1_final.txt

copy directory and (Preserve ownership):- copy all the content from file1 to file1_final

cp -Rp myDir/ myDirBackup

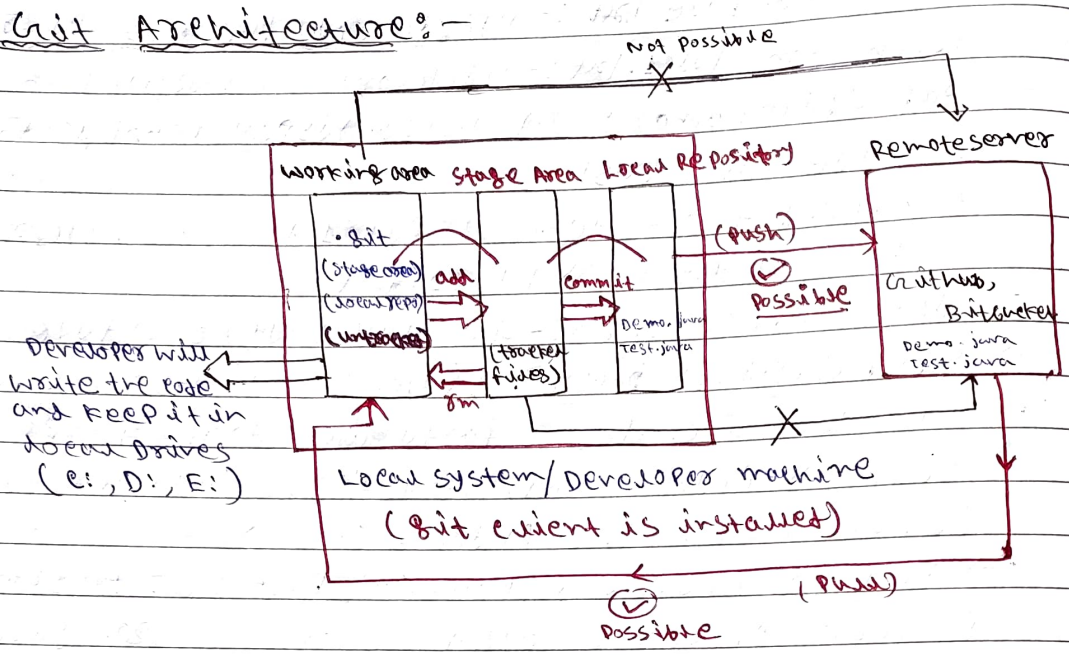
⇒ mkdir [Option] [Directory] ⇒ creat a file

⇒ touch (new empty file): touch <file name>

touch -c file1.text → If file (file1.txt) already exists, the this will update the file's time stamps, otherwise it will do nothing.

touch -a file.txt :- update only the access time stamp of the file.
touch -m file1.txt :- update only the modification time of the file
⇒ cat <filename> :- It will display what is inside in a file on bash.
Append the content of file1.text to file2.txt →
cat file1.txt >> file2.txt
⇒ less <filename> :- Display fussy of a file one page at a time
⇒ grep [Options] pattern [file...] :- is useful when you wish to search for a particular string files.
→ grep -v Andrew employee.txt ⇒ show all the line that donot match the Andrew string.
→ grep -r Andrew dirname/ ⇒ Recursively search for pattern Andrew in all files in the spe. diree. dirname
→ grep -i Andrew employee.txt ⇒ performs a case-insensitive search (Andrew and andrew Both)
⇒ open a folder > Right click) click on "Git Bash Here" or, open a folder > Right on folder directory address as " git bash" ↵ ⇒ To open git bash with folder adress.
⇒

4) Git Architecture:-



So, There are 3 regions:-

a) **Workplace** => It is a place where developers maintain there 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 stage area, we commit it to the local repo with some standard message, from local repo we "push" to main Repo by providing url, username and password.

5) Why is use of Stage Area in DVCS?

(PART-2)

5) Git commands & operations :- (case sensitive)

⇒ 1) git version :- This command is used to check the version of git
Syntax :- git version or git --version

⇒ 2) git help :- If we want to see the list of commands then we can use git help. Syntax : git help
NOTE :- This command is useful to get the documentation of any command. eg : git config --help or git help config. To show on bash → git config -help

⇒ 3) git config :- It is used when the git software is used for first time. The command will set the developer identity like name, emailid, ...
This configuration information will be used by git software for every push operation encountered.
> git config --list // This command is used to provide the list of configuration. It shows like -
diff.astextplain.textconv=astextplain
user.name = Swarup-mandal
user.email = swarup@gmail.com → etc. ... →
> git config, --list, --show-origin,
It will show the origin of .gitconfig.
(five:C:/Users/Swarup Mandal/.gitconfig) → to remove previous .gitconfig file goto address and delete the .gitconfig file and check again.

⇒ // To set the username and email :-
> git config --global user.name "Swarup-Mandal"
> git config --global user.email "swarupmandal07@gmail.com"
⇒ 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 work place and inside the folder the source code would be placed.
→ Normally this is the first command which we execute to set up the git for operation like - clone, push, pull...
→ This command internally creates one folder called .git
→ .git is used by git software to identify the folder which should participate in pushing to "local" and "remote" repositories.

Note :- To change the directory use cd <directoryname>
To check in which directory we are current working → pwd

⇒ > **git status** → This command is used to check the status of the working directory.

D:/gitsession/workplace_1 (master) ⇒ Demo.java
on branch master
No commits yet
Untracked files:
(use "git add <file>..." to include in what will be committed)
Demo.java
nothing added to commit but untracked files present (use "git add" to track)

→ git status normally will give outputs in the following ways :-

a. **untracked file (red color)** ⇒ it means the files are present still in working area and this files can't be commited to "local repository" nor to "remote repositor."

b. **tracked file (green color)** ⇒ it means the files are moved from working area to stage area. So, these files can be commited to "local repo" and to "remote repository".

c. **modified files (red color)** ⇒ it means the files ~~can be commited~~ are present still in working area and these file can be staged or it can also be restored back to the normal phase.

⇒ git add → To send the code from workspace to stage area we use the following command → git add <file name>

→ If we want to push all the files from workspace to stage area, we use the following commands →

> git add . or git add --a

→ It is also possible to unstage the files from staged area to work space, using the following command

>Syntax:- git rm --cached <file name> | git rm --cached * (for all files)

→ To restore the old file we use the following command

Syntax:- git restore <file name>

⇒ git commit → The file which are ready for commit should be in stage area, to perform commit operation we use the following command →

>syntax: git commit -m "some-messges"

e.g#1: git commit -m "First commit" // This file commit all the files present in stage area

e.g#2: git commit -m "second commit" <file name> // This will commit only that file into local repository.

⊛ Steps followed to create a remote repository and Push it to remote repository - - - →

1) Open github.com by providing the credentials

2) creat a new repository and enter some name (repository name) and click on create repository.

⊛ Note:- By default there may be possible of having main branch selected. So, change the main branch to master (or, creat master branch) Because, main> main direct Push not possibe

3) To perform push operation we need to use the following command →

→ git branch -M (move) main (it will create and shift master to main)

→ git remote add origin https://github.com/nitinTechnology/WORKSPACE.git

→ git Push -u origin main

↳ Push by user credential Pushing to main

→ (this command makes both create branch and move to that also.

8) Defference between pull and clone?

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

(fetch)

Syntax:- git pull

→ git clone => it is used to clone the repository to the working directory of the developer.

Syntax:- git clone <url>