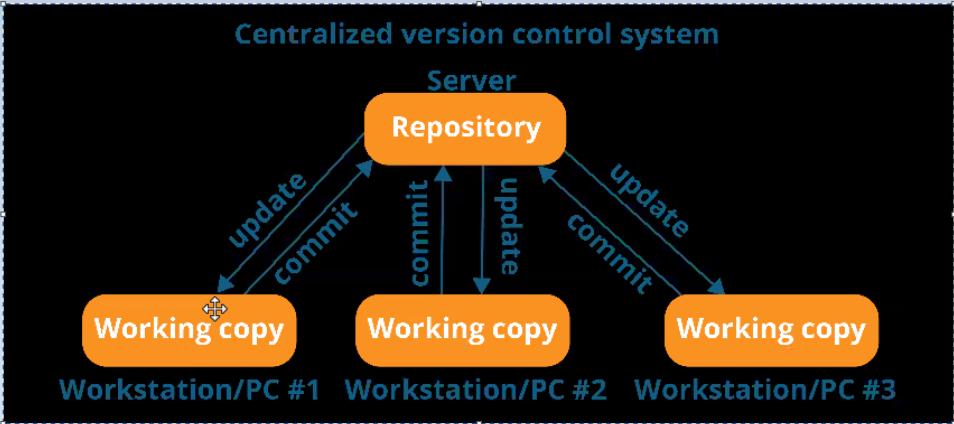
**Git and GitHub**

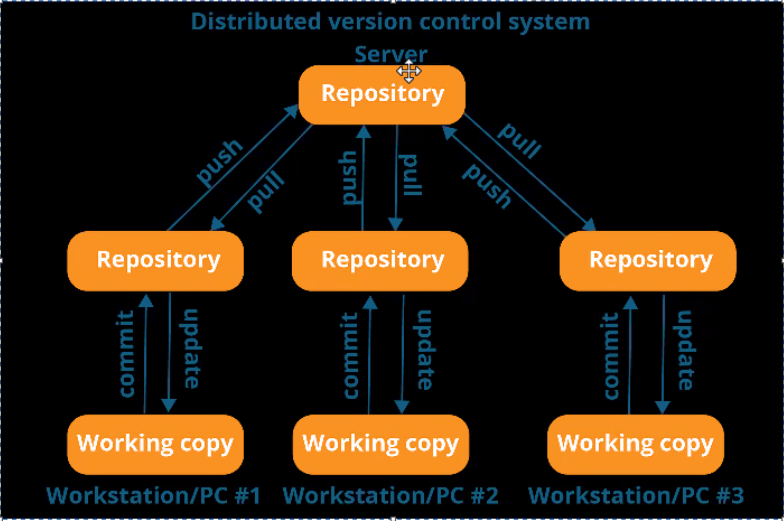
Git is something where we can maintain the version.

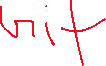
1.centralized version control



We have issue with centralized system and we are not working.

2.distrubuted version control system







* Git –version
* Git init 🡪 initialization
* Git status 🡪 to know the status of the file
* Git add filename 🡪will add particular file
* **Git add .** or **git add -A** or **git add \***🡪 (.) means current folder and add everything
* **Git rm --cached filename** 🡪 it will delete the file which is added
* **Git rm --cached foldername -r** 🡪 recursive
* **Git rm - -cached \*** 🡪 delete everything
* **Git rm - - cached filename -f** 🡪 remove from the staging
* **Git restore filename 🡪** wont remove from staging and revert the change we have done
* **Git commit -m “message”** 🡪 -m means message
* **Git log** 🡪 to see log message after commit
* **Git ls-files 🡪** To see the file which is in committing phase
* **Git show 🡪** difference will show when file is present in committed phase
* **git diff** 🡪 it will show the differene when data is there is stage environment
* **git reset commit id 🡪** oly commit will get changes but file will remain same
* **git reset - -hard commitID 🡪** commit will delete and also change the file
* **git remote -v 🡪** To see what all the permission we have
* **git remote add origin httplink**
* **git push origin main -f 🡪**for forcefull push
* **git clone http link 🡪** we can clone from the

**Parag trainer code :**

* git status - it lists all  the modified files which are ready to be added to the local repository
* git config --global user.email "ilearniexcel@gmail.com"
* git config --global user.name "coderepositor"
* git config --list - It will show the entire configruation parameter set for the local repository
* git add <filename> : It will add the file into the staging area
* git add \* : It will move all the files from working directory to staging area
* git add -A: It will move all the untracked files
* git rm --cached <filename> : It will unstage the file from staging area
* git rm --cached <filename> -r : Recursively removes the file from cached folder to working directory
* git rm --cached \* : It will unstage all the files from staging area to working directory
* git rm --cached <filename> -f : It will unstage all the files from staging area to working directory forcefully
* git restore <filename> - It updates the working directory file with staging area file and it is not unstaged
* git commit -m "Initial version": It will commit all the files from staging area to the local repository
* git commit - It will open the message file to enter the details about the version you are committing
* git log : it will show all the commit logs made for the working directory
* git ls-files: it will show all the files in the local repository(commited state)
* git reset <commit id>
* git reset --hard <commit id>
* git show: It will show the difference between the files between the current commit and the previous commit
* git show <commit id>: It will show the difference between your latest version vs commit id provided
* git diff : It will show the difference between working directory file and latest commit
* git diff --staged: Difference between staged file and the latest commit
* git remote add origin <repositories link> : It will connect local repository to the remote repository
* git remote: To find how many remote repositories connected to my local repository
* git remote -v : To check the repository links
* git push origin master/main
* git pull <git repository link>: It will download the data from the git repository
* git clone <url> : it will will obtain a repository from an git url
* git branch : List down all the branches
* git branch <branchname> : crates the branch with the given name
* git checkout <branchname> will switched to the branch
* git checkout -b <branchname> : It will create the branch and checkout at the same time
* git diff master : It will compare the files between current branch and master
* git merge <branchname> : It will merge all the changes of the given branchname to the master branch
* git branch -d <branchname>: It will delete the branch
* git pull <git repository link>: It will download the data from the git repository
* git log --pretty=short  
  git log --pretty=fuller  
  git log --pretty='format:%C(auto)%h (%s, %ad)'