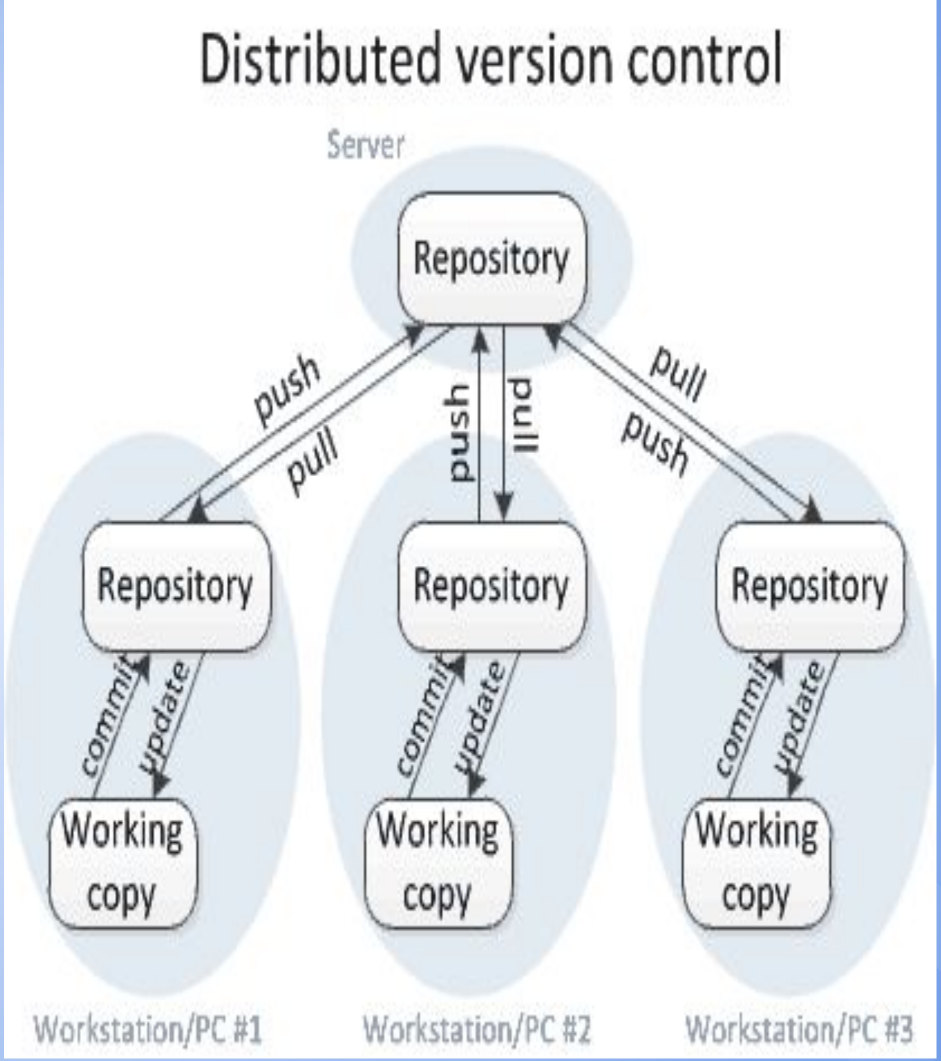
Introduction to git

Harindra

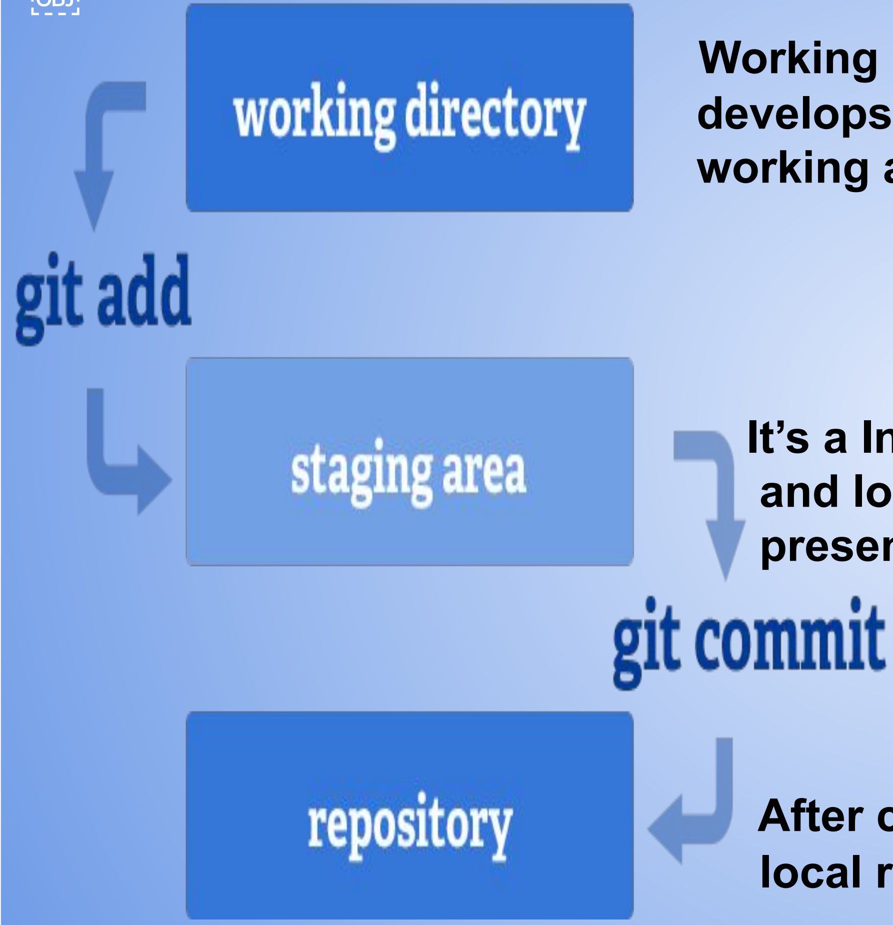
Git is a version controlling , which we can store older and latest version of code on remote repository. At the end of the day all the developers upload their code into version controlling system. version controlling accept all the code coming from the various developers and create one integrate project out of these uploads. The process of uploading nothing but **check-in.** The next day when developers download code from the Vcs they can find the code created by all the team members and the process of downloading is nothing but **check-out.**



GIT Advantage : -

- We can easily track who did made changes on the code.
- At any point of time we have flexibility to switch from one version to another.
- No need continuous network connectivity, it requires only while the time of push the code.
- If anything happens to remote server, already we have backup on each individual developer they can push the code again.

We have 3 stages



Working area is the location where file created and develops there code here and the file present in the working area is called untracked files

It’s a Intermediate buffer zone between working area and local repo before committing the files would be present at staging or index area the file present the staging area is nothing but staged file.

After committing the files would present at local repo these files are called committed files.

Git Commands

git init

--> To initialise git repo, if we do git init by default we get master branch

git status

--> Shows the files present working directory as well as staging area

git add [filename]

--> Moving particular file from working area to staging area

git commit -m [MSG][filename]

--> To commit particular file (moving file from staging area to local repo)

git commit -m [MSG]

--> Move all files present in staging area to local repo

git reset [filename]

--> Get back file from staging area to working area

git config --global user.email “ your name "

git config --global user.email " your email "

Git Commands

git reset --soft [commit id]

--> Get back file from local repo to staging area

git reset --mixed [commit id]

--> Bring back file from local repo to working area

git log --oneline

--> Shows the committed history in single line

git log --oneline -3

--> Shows the only latest 3 commits

git show [commit id]

--> It displays the list of files in that commit

git branch flow : - If we enable the branches, developers can enable to work on multiple branches and they can work multiple functionalities at a time

git branch [branch name] --> It creates branch

git branch --> Will display the list of available branches

git branch -a --> Display the branches of local as well as remote branch

git checkout [branch name] --> Switch into particular branch

git checkout -b [branch name] --> It create and switch into that branch

git branch -d [branch name] --> Delete the specified branch

git merge [branch name] --> Merge all the commits into particular branch

git cherry-pick [commit id] --> We can pick specific commit into our branch

git rebase --> It creates linear commit history and its a fast forward merge i mean future branch commit history will append to next to the head of master branch commit history

--> We can **rearrange** the **commit** history

ex : - git rebase -i HEAD~9 [number will be total -1, suppose we have 10 commits we have to pass 9]

git stash --> We can stash the particular the file and we can work on another file, i mean if we create or add or commit file it won't disturb the stash file

git stash --> The files present at staging area these will be stashed mean stash is a temporary location files saved here

git stash pop --> Unstash the files

git squash --> We can merge multiple commit into single

TAGS

Tags are just like bookmark,to specify the versions in the Repo

git tag [tag name] --> we can tag the latest committed file

`git tag -a v1.4 -m "my version 1.4"`--> we can pass the tags to particular or old commit by giving the tag name by providing commit-id (we can also call annotated tag)

git show [tag name] --> get the info about that tag

git tag --> displays the list of tags

git tag -d [tag name] --> remove tag locally

git push --delete origin [tag name] --> remove tag remote

git pull, git fetch, git clone

git clone :- It clones the complete repo if we do multiple time it downloads again and again entire repo

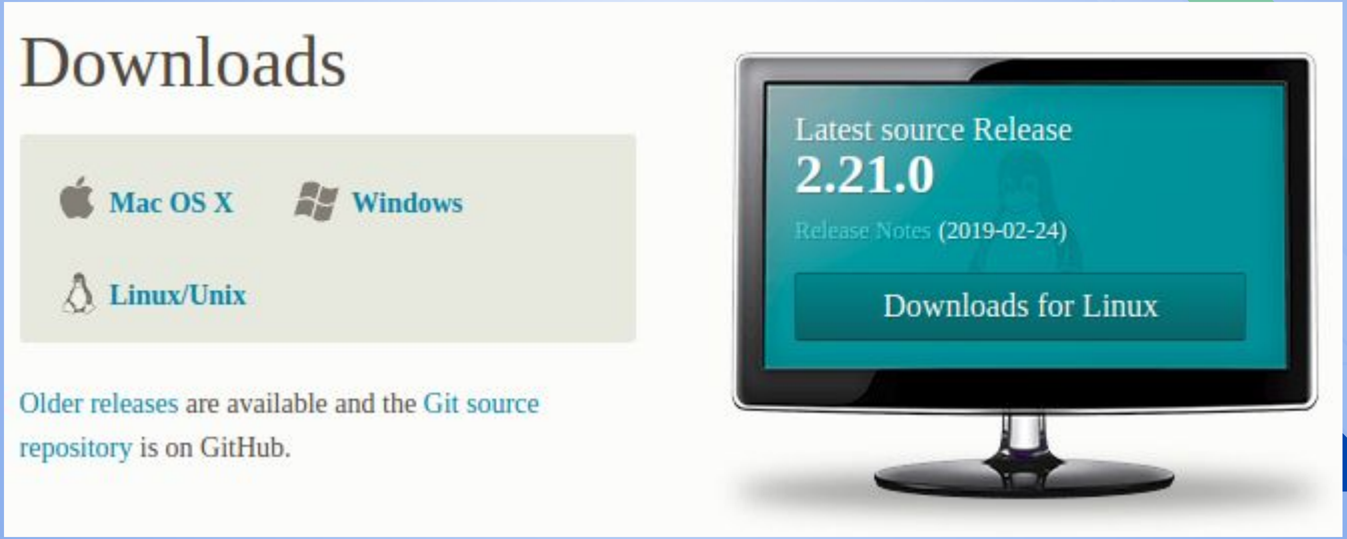
git fetch : - To get file only the updated one,if we do git fetch, 1st it will create one remote branch called **remote/origin/master** the updated file would be present at this branch we have to switch and we can the the file now, after that we have to merge with master branch to get the updates

git pull : - To get file only the updated one,if we do git pull it directly merge with master and display the modifications

Git Download

In browser just search for Git download we get download options for different OS, choose what you need, run the downloaded git executable file click next ...next ...next

https://git-scm.com/downloads



Downloads

Older releases are available and the Git source repository is on GitHub.

Latest source Release
2.21.0
Release Notes (2019-02-24)