Book:
Learn Version Control With Git

GIT

By: Zeeshan Hanif Sir

3

9:56PM
19-07-2020
Sunday

Version Control System Part 1 :-

- also known as source control system or revision control.
- It is basically a management of document, programs, large web sites or other collection of information.
- tightly coupled with software development.

VCS Kind Of Database :-

- VCS options Git, Mercurial, SVN, Preforce.
- kind of database ble it has snapshot of your project at time you want.

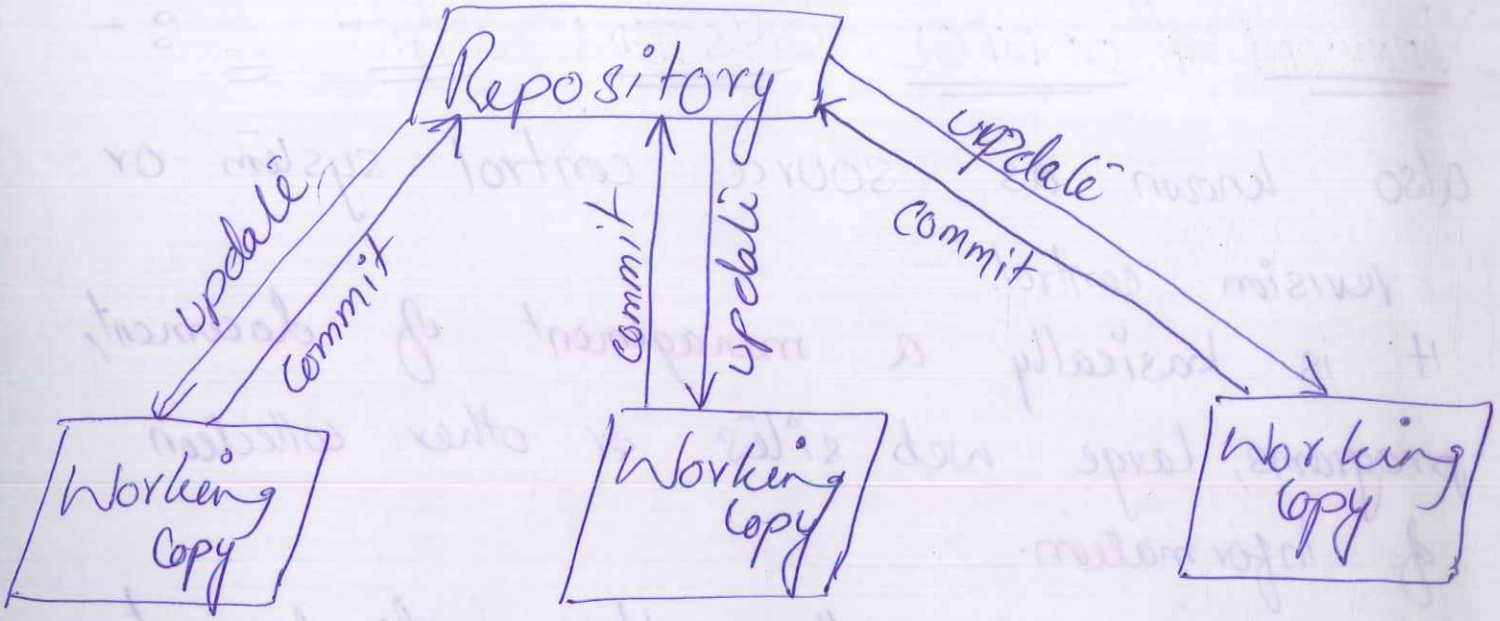
- VCS is INDEPENDENT :-

It is independent of kind of project / technology / framework / editor.

- Why Use VCS : Collaboration

- Git is distributed VCS

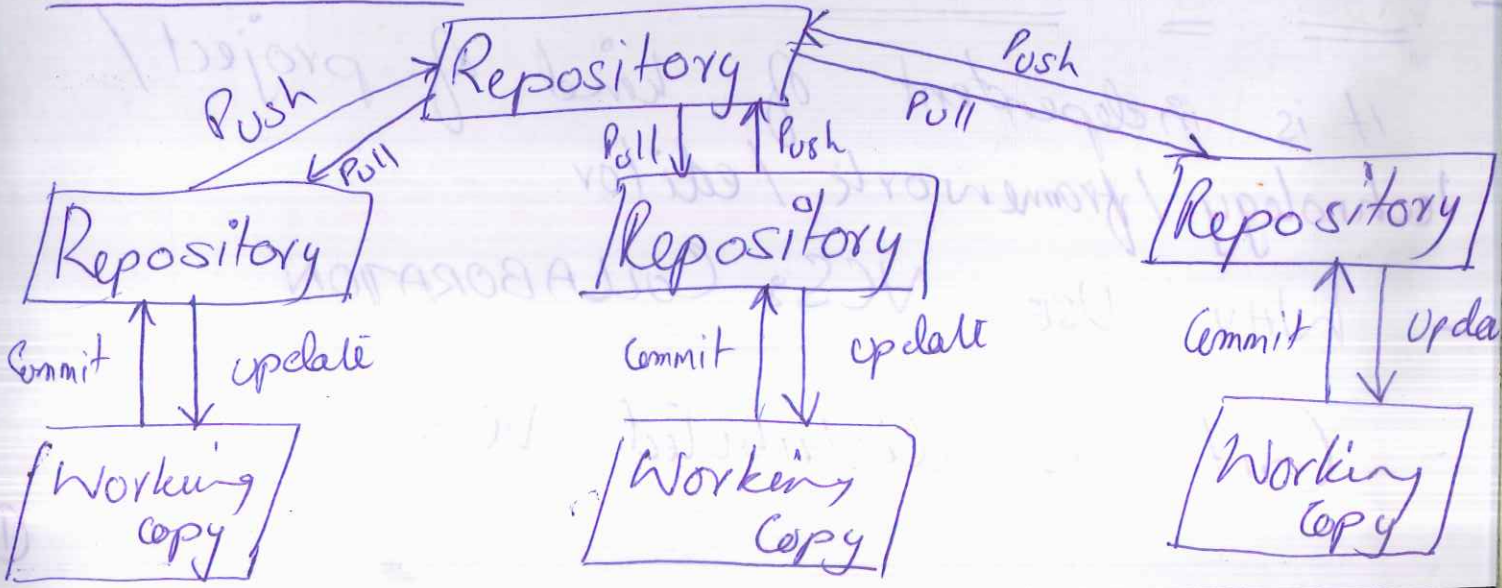
CENTRALISE VCS



- Every ~~op~~ operation is on central server.

- Drawback:
  - Network or internet is required
  - Server/machine lost mean data lost.

DISTRIBUTED VERSION CONTROL SYSTEM



21-Jul-2020
TUE 10:42 PM

DVCS ADVANTAGES :-

(1) All operation are speedy since data is being access from hard drive not from remote server. (You do not need internet conn.)

(2) Changes if ~~so~~ are committed so you still have a ~~change~~ chance to push or revoke chances

(3) If central server crashes, so any User's copy can be utilized for a master copy.

INSTALLATION AND BASIC WORKFLOW :-

- Go to www.git-scm.com to download Git
  - www.syntevo.com for Smart Git

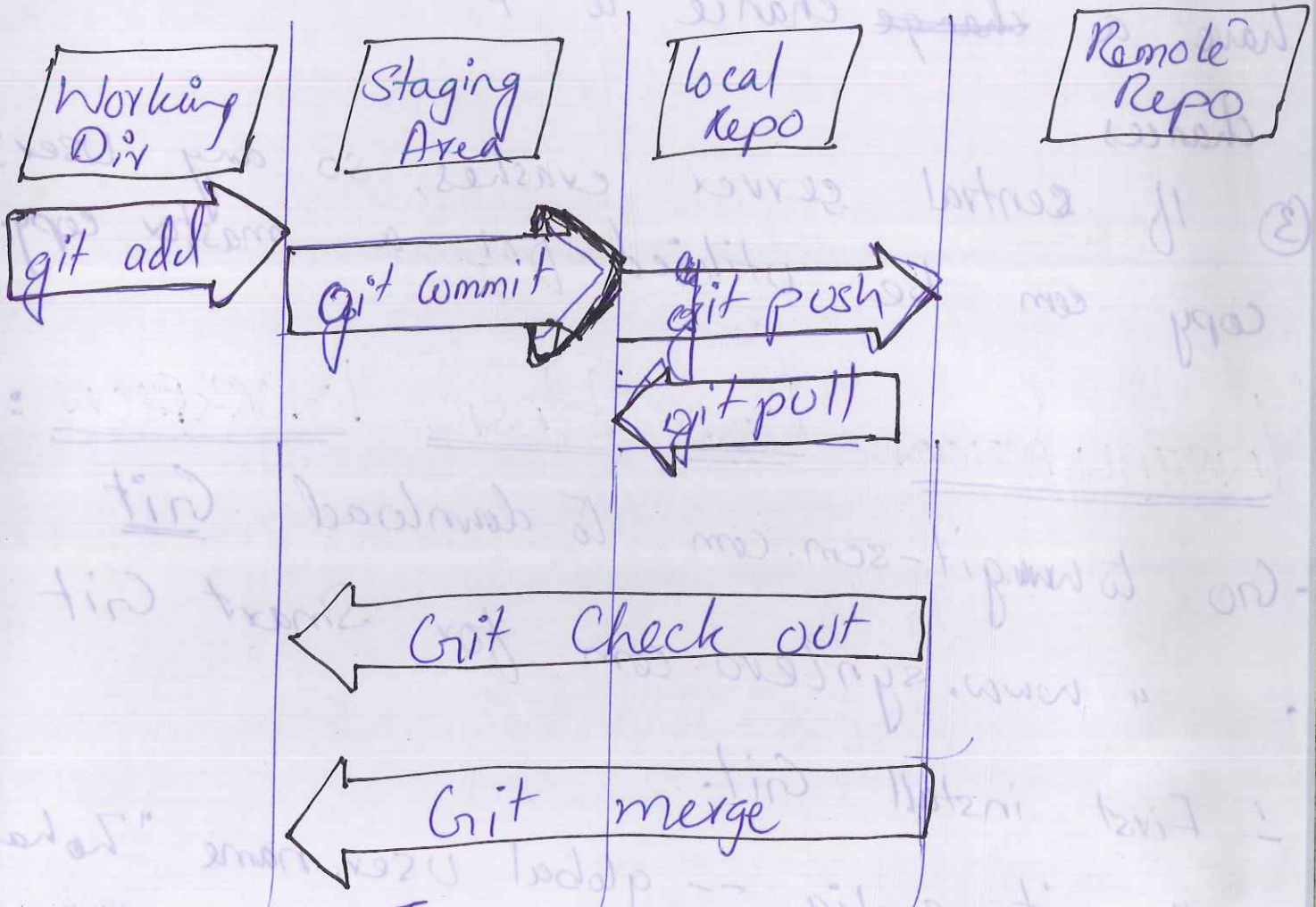
! First install Git.

- # git config --global user.name "Zohaib"
  # git config --global user.email "Email"

SMART GIT:
- Select non-commercial user only.

GIT OPERATIONS :-

1) Intialization.
2) Add
3) Commit
4) Pull
5) Push



IMPORTANT TERMS

1) Repository.
2) Working directory.
3) File status. (Tracked/Uncracked

23-7-2020
8:38PM
THURSDAY

GIT INSTALLATION

- Download GIT :-
  - go to https://git-sem.com/download/linux

For GUI :-
www.syntevo.com/smartgit.

Installation of Git :-
Next, Next. ~ next.

SETTING UP GIT :-

git config --global user.name="ZOHAIB ATTAR"

git config --global user.email="Zohaib4986@gmail.com"

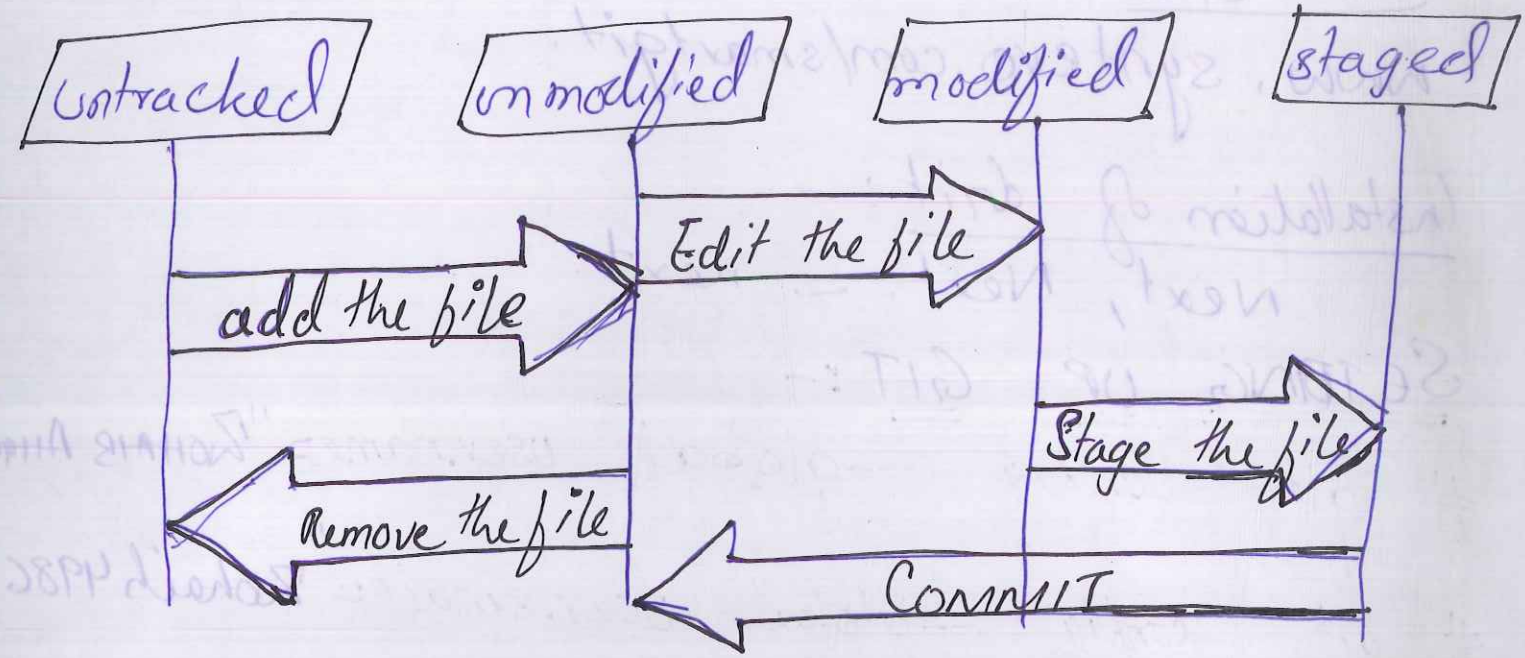
SETTING UP SMART GIT :-
- Non commercial

GIT OPERATION :-
- Initialize
- Add
- Commit
- Pull
- Push

ADVANCE OPERATIONS :-
1. Branching
2. Merging.
3. Rebashing

IMPORTANT TERMS :-

1) Repository.
2) Working directory
3) File status.



4) Staging Area

BASIC WORKFLOW :-

- Open terminal and create directory

C:\Repo\myproject

git init (To initialize repo in directory)
This will create .git hidden folder

- To add file

git add first.txt second.txt

- Commit Message:

git commit -m "implemented new feature"

- Git history

git log

Git status :- (# git status)

To check changes

To ADD :-

git add .

OR

git add *.txt

OR

git add first.txt

COMMIT: PAST, PRESENT & FUTURE

- If C1, C2 are the commits then C2 will be HEAD, current represents Branch (master), and next commit will future commits.

COMMIT HASH :-

Every commit provided with a unique identifier of 40 bits.

UNSTAGE FILE

By git reset command, it will remove from staging area. along with changes if not committed.

git reset.

git reset --hard (remove from staging area and discard)

* IGNORING FILES :-

Some files or folder if exist in Git repo then you may ignor them.

E.g
- .DS_store (MAC)
- Node modules
- Build
- logs

So you will create a file without name having extension .gitignore

E.g
```
*.doc
*.js
third.txt
build (folder)
```

26-July-20
8:41 AM
SUNDAY

BRANCHING & MERGING :-

Basically a concept of branch is to seperate working space from another. like if you making project of new web site and suddenly you need change old (already deployed web site) then you change you branch; add, changes and submit changes and then revert back previously working branch.

BRANCH COMMANDS

⊙ show list of branches

git branch

⊙ Show list of branches with some details

git branch -v

⊙ Create new branch with new name

git branch new-dev

⊙ Switch to new 'new-dev' branch.

git checkout 'new-dev'

⊙ Merge branch into current active branch

git merge new-dev

⊙ Show commit difference in two branches

#git log new-dev .. master

OR

#git log master .. new-dev

STASH :-

- A clipboard or temporary place where git files get save and these changes are not committed to repo.
- Later you can restore the changes from stash in your working copy and continue working where you left.
- you can create as many stashes as you want.

#git stash #git stash save <name>

#git stash apply stash_name stash@{1}

git stash list # git stash pop

By Sir Zeeshan Hanif

Remote Repositories

26-7-2020 Sunday 6:05 PM

Online services
- Github
- bitbucket Bucket
- Gitlab
- Many other

So you don't need to maintain individual/own central server.

Cloning a repository

git clone url

Command For Remote Repo :-

git push
↳ push changes to remote repository

git fetch
↳ Fetch changes from remote repository.

git merge
↳ Merge changes that was fetch by git fetch command.

Instead of git fetch & git merge we can use

git pull

↳ Fetch and merge changes from remote repository.

git remote -v

↳ Show remote urls

git remote show origin.

↳ Show details of origin.

git remote add myremote

↳ This command will add remote repo to local repository.

PUBLISH LOCAL REPO TO GITHUB

git remote add origin url
repo name

git push -u origin master
repo name branch

-u flag establishes a tracking connection b/w remote and our local

COMMAND FOR REMOTE REPO

git push 'remote repo name' 'branch'

like

git push origin master.

git log remote/branch or origin/master.

PUBLISH LOCAL BRANCHES

git branch work
branch name.

git checkout work

git push -u origin work

8:25 AM TUE
28-JULY-2020

ADVANCE GIT

GIT WORKFLOW:-

It is possible while working with git that your code need to be tested and verified before going into production so git advance gives us a feature that in which a developer request the authority to push request and after review.

Pull request is a github feature

Fork

If you dont have write access of repository then you fork means a copy of repository is create in your github account so that you can made changes ~~and then~~ like open source project.

- You can request pull request if so its on owner discretion that whether they accept you changes or reject.

DELETE BRANCH :-

git branch -d contact-form

For remote branch

git branch -dr origin/contact-form.

To delete branch from git hub

git push origin --delete test.

UNDOING LOCAL CHANGES :-

git checkout head <file to restore>

git reset --hard Head

UNDOING COMMITTED CHANGES

git revert <commit hash>

git reset --hard <commit hash>

WED 8:32AM
29-JUL-2020

ADVANCE GIT -- REBASE

ALTERNATE TO MERGE -- REBASE

- merging is the easiest and common way to integrate changes.
- Rebasing is quite a bit more complex than merging.

MERGE
- Fast forword : (simpliest one) changes only one branch at a time
- Merge commit: creat a additional commit of merger

REBASE :- Add changes in straight line.

git rebase <branch name>

PITFALL / DISADV. :-

If a commit like C3 that is in b/w C2 & C4 will be added as new commit

BENEFITS :-

All the commit being done branch A will be seems ifor branch A.

THU 1:13 PM
30-Jul-2020

MARKDOWN

- Markdown is a lightweight markup language that you can use to add formatting elements to plaintext documents.
- Created by John gruber in 2004.
- Often used for 'Readme' files

GFM (Github flavoured Markdown)

Heading :-
It has 6 heading for big heading use '#' single hash or for smaller one use 6 '#' six hashes.

Styling Text :-
Bold ** text ** or __ text __
Italic * text * or _ text _

Quote text :-
use > to quote

Quote Code :-
use ` text ` to format text in distinct block.

1-A

- Links :-

use [text] to make it URL like

[Github Pages](https://pages.github.com/).

List :-

Oreder list use '-' hypen will convert into bulled

Nested :-

1. Hello
   - World

Task list :-

use - [ ] or - [x] - checked

uncheck