Github Repo :

https://github.com/architsharm/dsml_git_demo

https://github.com/architsharm/Spoon-Knife

Start at 9:05

=> Welcome to MLOPs =>

=> • Python
Math based >

=> MLOPs => • All tools & Service associated with DS

1) Git & Github
2) Streamlit
3) Flask
4) ECS
5) CI/CD
6) Containers &
7) ML System Design
8) Sagemaker
9) PySpark

=> What is Git ??

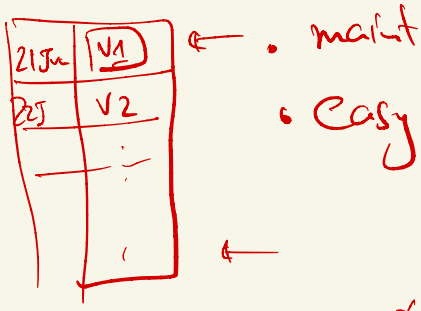
VCS => Version Control System

=> • model 1 => model 2 => model 3

→ model1.pj ←
model2.pj ←
model1_1.pj ←
⋮

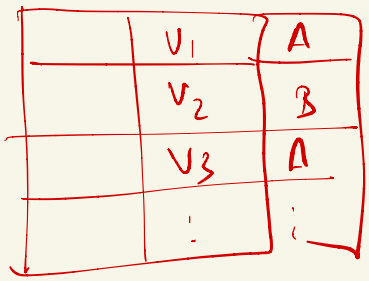
Checkpoints
↳ Save a point in history

⇒ → . a good way to record checkpoint in development cycle.

⇒ 

← . maintain every check point }
. Easy to retrieve

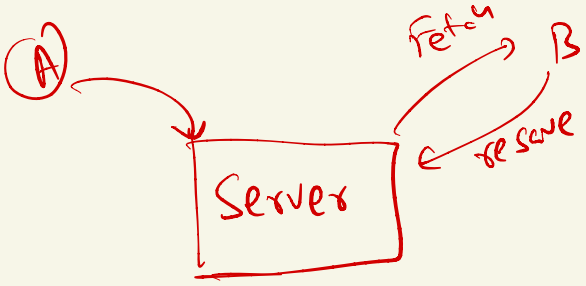
→ . Multiple people work on a project



⇒ . if we add a Creator tag

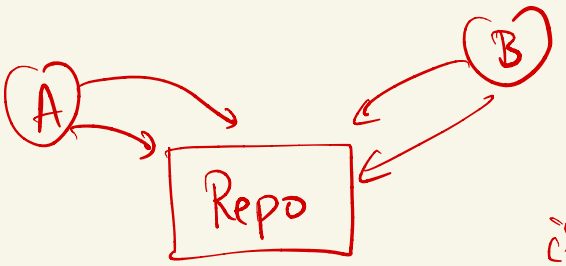
→ You will store this file (modelfile) Somewhere

→ A worked on project V1 (Record)
↓
Save it
↓
B will download → V2 → Save it (Record)



⇒ [Centeralised version Control]

* Distributed Version Control



Both A & B can work simentaneously

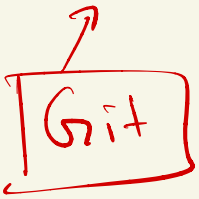
↓

if some save something

↓

other person has option to get those changes

Distributed Version Control System



⇒ what if A & B are at different location →

online platform that supports Git came →

⇒ Github ⇒ hub of Git

BitBucket → Atlassian

⇒ Git

Distributed VCS

Github

- Code sharing platform that uses git
- Open source repo hub
  - Bit Bucket
  - Git lab
  - aws Code Commit

⇒ Github → . online website ⇐
- Github Desktop ⇐
- Github Cmdline ⇐

⇒ Git basic Terminologies →

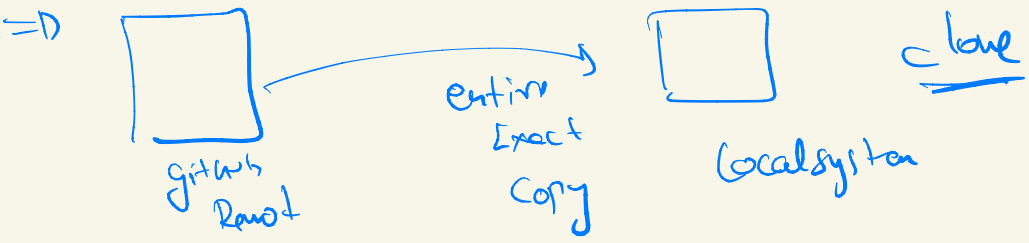
(1) Repository → a folder where all your Codefiles would be.
- Local → The folder which is on your own machine
- Remote → folder which has been uploaded on a server.

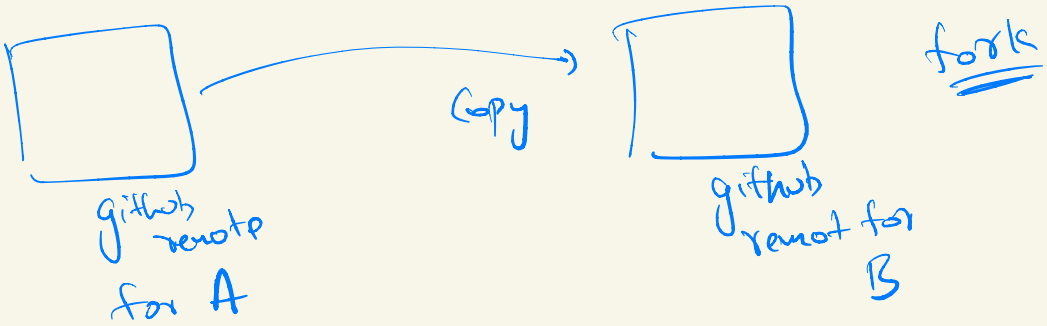
(2) CMDs →

1) Commit → . a checkpoint to save the code
2) Push → Send data from a local repo to a remote

3) Pull → bring data from a remote repo to Local system

4) Clone → making a Copy of any repo
↳ exact

=> 



=> Readme → helps people in understanding what the Code is about

=> Gitignore →
- You have password stored
- You have a very large setup file
- local db file
-

↳ List files | Subfolder which you do not want to Share with remote repo

- Security
- Size limit
- Cached images

⇒ Publish repo → first time send the data/create a remote repo

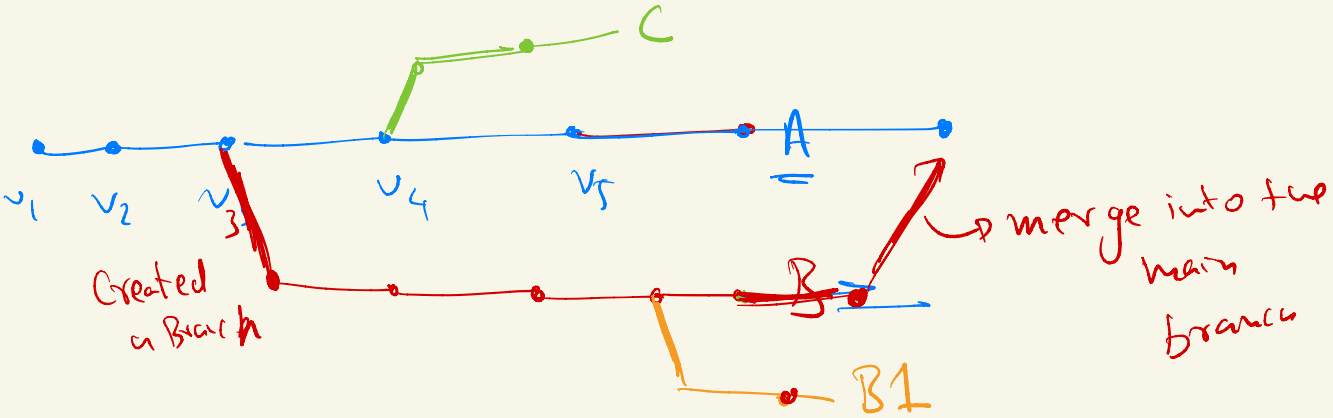
---

⇒ local repo → made changes → pushed → Remote

⇒ Multiple people can collaborate ⇒

Server → everyone can have their own copy of repo

→ Git Branch → Copy of the main code.



→ A & B will always make changes which does not contradict each other

def eval ( ): → rmse ⟷ def eval ( ) Std_err

⇒ when B ⟶ A
want to merge with

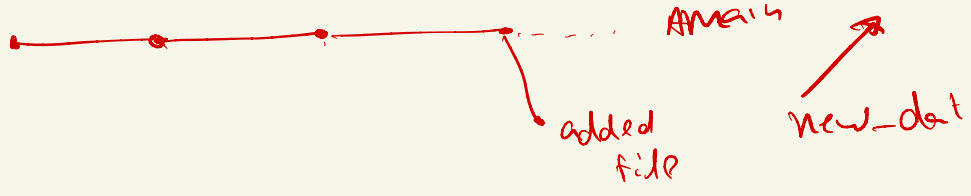
→ B create Pull Request
- Resolve any Conflict →
- Merge after you resolve Conflict

⇒ Shift from one branch to another
↳ checkout

Remote
main

Local
- main
- new_data

Publish new_data to remote



→ merge new-data into main

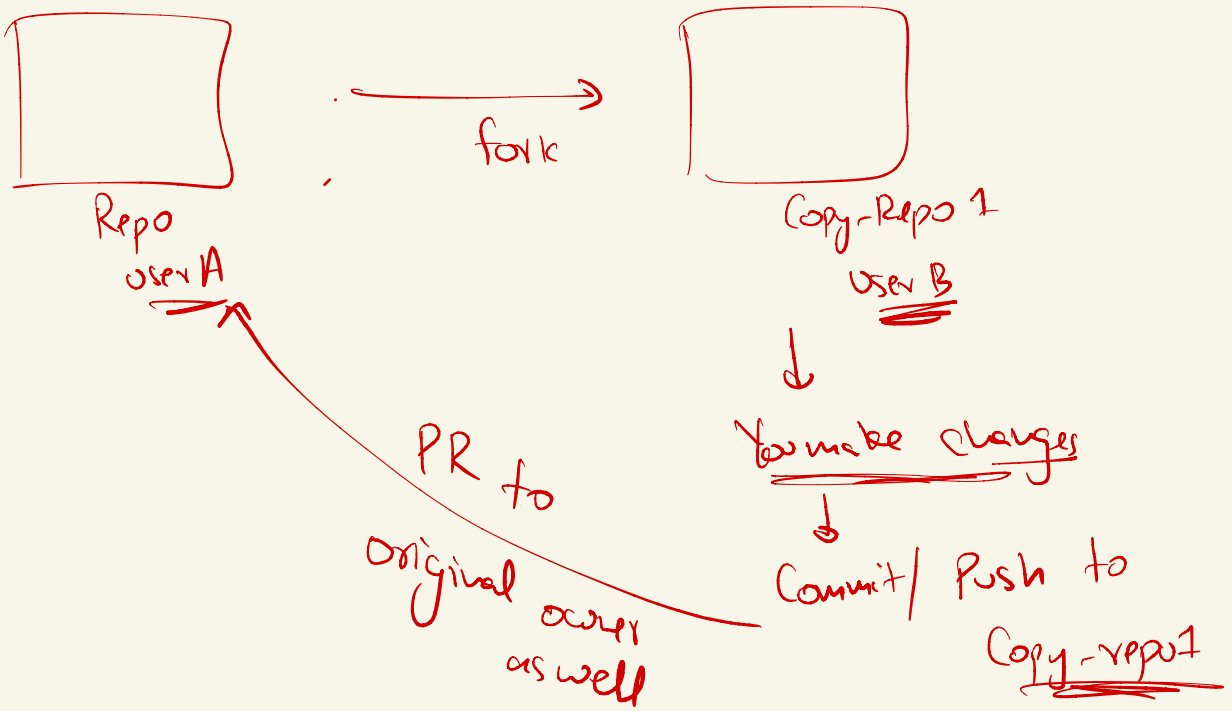
→ PR

⇒ main_branch → change new → hello archit
→ hello vishwa

new-data → change

⇒ Open Source Contribution →

- access to directly change a repo



⇒ For first time → you would need to clone a remote repo → local repo.

⇒ Clone any remote repo → UserB/random

→ a local repo with Name "Random"

UserA

Publish to remote to "Random"

⇒ You can't directly contribute to UserB's code

Qui2 1) Default branch git → main

2) Clone → Copy of the code in local

instance of repo → it is connected to original repo

Fork

3)

⇒ Git Stash → any uncommited change → you store somewhere

→ reset the code

⇒ Code → fileA, fileB → Stash → fileA, fileB

Code → [ ]

=> Git reset → • Delete all local changes →**

=> Git terminal

- git clone →
  push →
  pull →
  merge →
  commit →
  checkout →