**GIT :**

VERSION CONTROL

VCS

SCM

LOCAL, CENTRAL AND DITRIBUTED REPOSITORY

WHAT IS GIT

WORKING COPY , LOCAL REPOSITORY AND CENTRAL REPOSITORY

ADVANTAGES & DISADVANTAGES

BRANCHES IN GIT

BRANCHING AND BRANCHING STRATEGIES

Readme.md files

.gitignore

License

Gitbash

Pwd

Ls

Mkdir devops

Cd devops

Ls -a it will give u all files name and u will get an .git file

Ls ~a for direct home directory

Git init for initialize

Cd d+tab shows all folder with d

Cd .. for going previous folder

Cd . for current directory

Cd ../.. for going directly to home directory

Cd – for going to the previous one from where u have come

Cd for direct going to the home directory

Cd .ssh (if no directory is there) u can create

Ssh-keygen ( use for creating an ssh folder) and just do enter 2-3 times

To increase size in vm do ctl and move mouse

Cat file path for getting pub key private key and then go to the github and go to repo and setting and left go to ssh and add new ssh key

Path c:/user/administrator/devops

Git config global –user.name “kaushaldarji7182”

Git config global –user.email “kaushaldarji7182@gamil.com”

Git add .

Git add -A all the files will get add

If we want only specific when write git add data.txt

Git status for checking its properly trakced or not

Git commit -m “ init version “

And the n do git push to push file in the repo

Git push

Git pull , fetch and …

Git help for finding all the commands and their use

Git add -h for help

Git diff and git diff -w for the difference

Git diff for finding changes difference

Git diff –word-diff for word to word check changes

Git diff - –stat for finding statistic of changes how many changes is being made

Git diff –numstat for finding how many lines and words has being changed

Git commit then it will go the terminal then press key I and add commit msg and after editing press esc key and then execute :wq

Key I for insert mode

Key esc for command mode

:w for writing file

:wq for saving file

:q! for quit without saving

:wq! For proper saving and quit

In Git, **ORT** stands for **"Ostensibly Recursive's Twin"**, which is a **merge strategy** introduced in Git 2.33.

It is a **high-performance replacement** for the older **recursive merge strategy** used by Git when combining branches. The name "ORT" reflects its role as a twin to the older "recursive" strategy.