

*****What is Git & VCS (version control system)**

- Git is a free open-source version control system.
- Version control software keeps track of every modification of the code in a special kind of database.
- Vcs system helps term of rollback to previous version in case of any issue with specific version.

***** Need of version control system**

types of vcs :- Basically two types of vcs.

1. Centralized version control system (CVCS)
2. Distributed version control system

Centralized version control system :- CVCS uses a central server to store all files and enables team collaboration.

CVCS works on a single repository to which users can directly access a control server.

Centralized version control system (CVCS)

Repository (server)

Working Copy

Workstation pc1

- Here each workstation is connected with the central code repository.
- Drawbacks: - It is not locally available.
- Crash of cvcs will result in losing the entire data of the project.

Distributed version control system :-

- In distributed vcs every contributor has a local copy or "clone" of the main repository.

- Users can change and commit local repo without any interference.
- Users can update their local repo from the central server.
- Users can update the central server from their repo.
- Operations in dvcs are fast.
- New changes can be made locally without manipulation.
- If the central server gets crashed at any point of time, the lost data can be easily recovered from any one-off contributors' local repositories.

GIT Key Terminology: -

Git Repository: - Git repository contains files, history, config managed by GIT.

The git repository

Working dir

Repository

~/projects/my.project

./git

Text

V1 txt, v2 text

Stages of GIT:-

Working dir: - Area of live files also known as untracked area of GIT.

Staging Area: - Staging area is when git starts tracking and saving changes that occur in files.

Git Direcotry:- Also called "local repo" is yours .git repo its area where git save everything.

Remote Repository: - GITHUB

Remote repository is stored on a code hosting service like GitHub or on an internal server.

Branch In Git:- Branch in Git is a way to keep developing and coding a new feature or modification to the software and still not effecting the main part of the project.

—

*****Install GIT on linux machine: -**

Git is open source and can be installed on all major os like Linux, mac, windows.

Here we are going to **install git in the ubuntu machine.**

sudo su

#apt-get update

To install git

#apt-get install git -y

To check version

#git --version

*****Install GIT on windowos machine: -**

- Go to the browser search download git.
- open the git website and download the file.
- After download run the setup file in machine
- Check the version.

c://Git -version

Note: - If you want to open git then searches gitbash.

Configure User information in GIT:-

Go to windows system open git bash.

Check the version by GIT --version cmd.

Note: - user configuration is global configuration

```
#git config --global user.name "Raushan kumar"
```

```
#git config --global email.name "raushan123@gmail.com"
```

To check user configuration details: -

```
#git config --list
```

Create First GIT Repository: -

```
#mkdir project
```

```
#cd project
```

Initialize the GIT by

```
#git init
```

GIT COMMIT:-

GIT STATUS: - it is a command which provides the details of current working details.

```
#git add (file name)
```

```
#git add first.file.txt
```

```
#git status
```

Note: - with the help of add. cmd we can move all working to staging area.

To this cmd we can commit working dir to staging area.

```
#git add .
```

We have other cmd for commit with commits: -

```
#git commit -m "first file commit "
```

To diff cmd we can see the updated data in the file.

```
#git diff
```

***** Git tracking & git Logs:-**

First check the working dir on git with the help of cmd also we can create new project so working to create new project.

```
#touch myfile
```

```
#ls
```

To check the status:-

```
#git status
```

After commit the code by commit cmd.

```
#git commit -m "my commit for second file"
```

```
#git status
```

GIT LOGS: -

With the help of git log we can get to know about the commit details or commit file metadata.

#git log :- If we run this cmd then shows the logs of all commit details but in case we want to check the particular single or author commit then run below cmd.

```
#git log --author="Raushan kumar"
```

we can get the details of commit with the help email also.

```
#git log --author="raushan123@gmail.com"
```

*****Explore GIT with GITHUB: -**

- Git hub is a website and cloud-based service that helps developers store and manage their code as well as track and control changes to their code.
- GIT hub is remote repository for your code base.

How to create account in github:-

Login into GitHub ----create a new repository.

Owner
Repository Name

Raushan
Git Training

Description: -

Public
private

Add a README.file

add.gitignore

template gitbook

create.

Note:- if you want to sync with local git to remote git server firstly login into github ---goto the code area form the will get clone option ,https, ssh, download zip other.

will use https copy that link after that come to the local server check the current path select the path where you want to crrate a clone of remote server in your machine.

Basically git works an root folder come to the root folder and run this cmd.

#git clone -b main (paste the link here)

Check to remote repo

#ls

#git status

We can create a file an local server that file we will upload to the remote server or central server.

In that way will use push or pull cmd.

#vi text.file

insert some data in that folder and save it :wq

#git status

Note:- if we want to add or message in cmd area.

This cmd commit with changes & message (commit for track file)

#git commit -am "commit message"

For push the code to central server:-

#git push

First time ask username and password of git account.

***Setup authentication in local github:-

Login into your github account -----settings---sshkey and GPG keys. (open this here will get some option like your personal profile , nes ssh key,new gpg key with the help of ssh & gpg keys for the setup.

Open ssh keys:- gener ssh key (click on it) here will get some option like about ssh checking for existing ssh keys generating a new ssh key and adding it to the ssh-agent adding a new.ssh key to your github account.

For jenerate the SSH key:-

1.Open gitbash

2. Your github email address.

```
#ssh-keygen -t ed25519 -c"raushan123@gmail.com"
```

After that they ask confirmation then just **press** enter.

will get some details:-

Enter file in which yo save the key (rakumar/.ssh/id_ed25519) :- **press enter**

Enter passphrase (empty for no phassphrase)

After that you get username or ssh id like (id_ed25519) :- **enter**

generate the ssh key and copy:-

firstly goto that location like **(rakumar/.ssh/id)ed25519)**

Copy the sshkey come back to github account ----setting--ssh and gpg keys---inside the ssh keys click on new ssh key

sshkeyss/addnew

title:- windows ssh key

Key :- paste the ssh key----add ssh key

After that we need to test for that again we need to goto github ---setting---ssh and gpg keys--ssh keys----generate sshkeys (click on that then you will get the option for testing ssh connection -----click on testing your ssh connection).

Here will get step for the test ssh connection

Open git bash

enter the following for test.

#ssh -T git@github.com

Before running this cmd you should an the clone repo in your local machine.

After verification try to push or pull something from your local repo to remote server (github).

#git push

Branch In Git:-

To check branch in local repo

#git branch

To create new branch:-

git branch develop (branch name)

To check create branch

#git branch

If you want to switch the branch:-

#git switch branch name

#git branch develop

create some file or dir here in develop branch

#cat file.txt g1 23 e3

check the status (here will get untarck file)

#git status

To cehck file or dir

#ll

To move the file from working dir to staging area.

#git add .

After check the stauts

#git status

To permanent save we have to commit the file by below cmd:--

#git commit -am "commit simple file a develop"

After that push the code to github server by cmd:-

#git push

First time we got error like (fatal:- the current branch develop has no upstream barnch to push the current branch and set the remote as upstream use).

then we have to use below cmd:-

#git push origin develop

After run this cmd go to your github server referesh then you will get your develop branch details like file or dir.

NOTE:- If you want to verify develop branch file or dir refelecting or not in master barnch.

#git switch master

#ll

Here some file are mssing like **file.txt g1 23 e3**

After creation of develop branch if we create some file or dir in master branch it will not reflect in develop branch.

create some file in master and verify:-

come to master branch by git switch master

Come to the master branch

#git switch master

Create file and some dir

#touch h1 b3 4

To check the status

#git status

To Move in staging area

#git add .

#git commit -am "test commit"

#git push

After check into the develop branch file will not refelect.

Note:- if we are going to push the code by master branch then we dont need to used **git push origin** (mastername).

- All the branches are sync with master branch.
- We can merge the all branch with master branch.

Note:- If you want to merge the code of two different branches like master or developer.

Go to your github account--verify first where to where you want to merge the code.

Here we have two branches main or develop so we want to merge the code from develop to main.

Go to the pull request---create new pull request--compare the changes.

Base:- develop branch ---compare ---main branch

Create pull request

title:- New file.txt

create pull request

Review

merge pull request ----confirm

Check the files in develop branch

If you want to work on a merge file then we have to pull the latest code from develop branch.

Cmd:- Git pull origin develop

come to develop branch then run this cmd.

branch develop:-

#git pull origin develop

How to add, commit in GIT:-

Create one dir and go inside it.

#sudo su

#Git init

touch myfile (put some data)

git status

#Git add .

#git commit -m "1st commit"

#git status

git log

#git show <commit id>

#git push origin (branch name)

#git pull origin (branch name)

Merge branches in GIT:-

- Isolating features into different branches is crucial practice for any serious developer.
- At some point a piece of code will reach a state where you will want to integrate it with the rest of the project. this is where the git merge command comes in.

Before merge :-

c0 c1 c2 **c3** c4----hotfix

after merge:-

c0 c1 c2 c3 **c4**--- master

Practically we are going to do that:- go to your develop branch and do ls create some file or put some data.

After that come to master branch run merge cmd

#git branch master

#git merge develop

Here will get all file inside the develop branch

#ls

Note:- if we want to upload this code on github then we have to commit this code.

#git add .

#git status

#git commit

#git push

Revert in previous state in GIT:-

- Revert is all about to undo the changes you did in repo.
- In GIT this can be done via **RESET** and **REVERT**
- RESET- Practically user can think of it as a "rollback".
- Reset points local environment back to a previous commit.
- Get reset is a powerful command that is used to undo local changes to the state of a get repo.
- To reset staging area:-

git reset <filename>

To reset the changes from both staging area and working dir at a time.

If we want to delete the add. file or current working dir file then we can use this cmd

git reset --hard

REVERT:-

- The Net effect of the git revert command is similar to reset, but its approach is different.
- The revert command helps you undo an existing commit to.
- Revert adds a new commit at the end of the chain to "cancel" changes.
- REVERT or RESET?
- If User have already pushed commit to the remote repo , a revert is a mice way to cancel out changes.
- Git workflow works well for picking up additional commits at the end of a branch, but it can be challenging if a set of commits is no longer seen in the chain when someone resets the branch pointer back.
- It commits in local then reset is good, it commits is pushed then revert is good option.

Practical: -

Git reset:-

firstly, check the all online: -

to check log:-

git log --online

create one file or put some data in that file

touch file1

vi file1

i am raushan.

:wq

git status

git add .

git commit - am "first commit"

again edit file1 and enter some more data along with previous data.

vi file1

i am employee of estuate,

:wq

git add

git commit -am "second commit"

To check the latest commit:-

git log --oneline

after commit we fell ilke second commit is not correct in that way we will use **RESET** for previous code line or commit.

If you want to reset your second commit then use this cmd firstly find the commit id with the help of commit.

git log --oneline

copy the second git commit id

git reset --soft (commit id)

git reset --soft 492930e

To soft cmd it will delete from the commit history log only if you want to delete from dir as well as then we have to used hard cmd like

get reset --hard 492930e

After that check log it is reset or not.

git log --oneline

How to work with GIT REVERT:-

Firstly modify into the file

vi file1

along with previous code.

nikhil is my best friend.

:wq

git commit -am "commit revert"

git push

edit again in file1

vi file1

nikhil dish is chicken.

:wq

git commit -am "commit revert2"

git push

after push check the file in your github server.

git log --oneline

If you want to delete the commit of "commit revert" from remote server use below cmd

To this cmd we can delete the latest commit to the remote server.

git revert HEAD

git push

If we run this cmd they will create another commit with previous commit in that commit they will delete the content as well as.

git log --oneline

Merge branches in GIT:-

To merge the branches master to develop:-

Git merge develop

To check the details of merge:-

git log --oneline --graph

Merge branch with message cmd:-

git merge branchname -m "message"

git merge main -m "merging main to develop"

If you want to merge develop to main come to develop mode:-

git switch develop

git merge main -m "merge develop to main"

git add .

git commit

git log --oneline --graph

Git Conflict:-

- When same file having different content in different branches if you do merge, conflict occurs resolve conflict then add and commit.
- Conflict occurs when you merge branches.

Git Stash:-

- Suppose you are implementing a new feature for your product, your code is in progress and suddenly a customer escalation comes because of this, you have to keep aside your new feature work few hours.
- you cannot commit your partial code and also cannot throw away your changes so you need some temporary storage, when you can store your partial charges and later an commit it.
- To stash an item (only applies to modified files nt new files).

GIT stash commands:-

To stash an item:-

git stash

To see stash ite list:-

git stash list

To apply stashed items: -

git stash apply (stash@{0})

To clear the stash items:-

git stash clear/drop

Stash Untracked files & GIT stashed POP:-

Come to terminal run below cmd to check untrack status: -

git status

we can get the details of track or untracked file with the help of below cmd:-

git stash -a

o/p:- stash the data

If you want to continue your work o pending or stash file used another method or cmd also.

git stash pop

To check list:-

git stash list

To pop will retrieve the file or delete from the stash area but apply cmd retrieve the data file noe delete from the stash.

