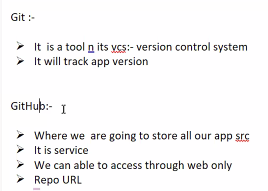
Git is a Local repository in our system

GIT stands for "Global Information Tracker"

In practical terms, Git is a "distributed version control system" used to track changes in source code

during software development. It helps multiple developers collaborate, keeps a history of changes, and allows you to revert to previous versions if needed.



==============================================================

create a file .gitignore in the server folder.

and add one line init,==> .env

So that it will not push sensitive information (.env)

=============================================================

**Install the Git :**

=====================

Download the Git for Windows installer from the official site:

👉 https://git-scm.com/download/win

Run the downloaded .exe file.

Follow the installation steps: (Keep other options as default)

Click Next through the setup.

Click Install and wait for the process to complete.

C:\Program Files\Git\bin (If Git exists there, it means it's installed but not added to the PATH.)

D:\>git --version

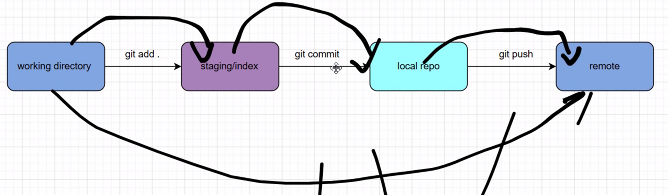
git version 2.48.1.windows.1

Add following two paths in the Environment variables:

C:\Program Files\Git\bin

C:\Program Files\Git\cmd

===================================================================



**Git Commands:**

========================================

Git (Distributed VCS)

-----------------------

1. Github account ( username, email)

2. install git in the local system

**Configuration (One time enough)**

--------------

$git config --list (github account, origin)

- logout github from local system

$git config --global --unset user.name

$git config --global --unset user.email

$git config --list

-Login GitHub/GitLab from Local System

$git config --global user.name "suneethabulla"

$git config --global user.email "suneethabulla@gmail.com"

Git Basic Commands

-------------------

1. Push

--------

step1: create repository in github (my-first-repo)

step2: create folder in git workspace in the local system

$mkdir my-first-repo

$cd my-first-repo

step3: push modifications in local folder to remote repo

$echo "<h1>this is index page">>index.html

convert folder local vcs

git init

git add .

git commit -m "commit"

git branch main/master

git remote add origin repo\_url

git push -u origin main

//git remote remove origin

2. branch

---------

git branch //Lists all local branches in your repo

gti branch -a //It displays the current branch

git branch v1 //Creates a new branch v1

git checkout v1 //to redirect to the corresponding branch v1

echo "<h1>updated file</h2>" >> index.html

git add .

git status

git commit -m "file updated"

git push -u origin v1

-----------------------------------------------------

3. merge (to add two branches)

Two possibilities

main--> v1 (Must be in the main repo)

v1 --> main

Merging:

===================================================

Direct Merging: (Some times may not work usually)

----------------

git checkout master # Step 1: Switch to master

git merge v1 # Step 2: Merge v1 into master

Indirect Merge:

--------------

>git ls-tree master --name-only //to display files in the current branch master

add.txt

>git ls-tree v1 --name-only //to display files in the current branch v1

README.md

add.txt

sub.txt

>git checkout master //be in the master branch

>git checkout v1 -- README.md sub.txt //To copy the files( README.md sub.txt files) into master from v1 branch without switching branches.

>git push origin master //pushed to github master.

>git ls-tree master --name-only //to display files in the current branch master

README.md

add.txt

sub.txt

=======================================

---------------------------------------------------------

4. delete branch (git branch -D branch\_name)

To remove the branch v1

-git checkout main

-git branch -D v1

------------------------------------------------------

5. pull (Applicable when work with collaborative repositories)

Fetches changes from a remote repository and merges them into your local branch.

Use case: To update your local repository with the latest changes from the remote repo (e.g., changes made by teammates).

Direction: Remote → Local

>git pull origin master

-----------------------------------------------------------

6. clone (To download the project from the Github)

git clone repo\_url

-------------------------------------------------------

7. log (TO get the log information)

git log

-------------------------------------------------------

8. rm (To delete a files from the repo)

git rm "temp.js"

git commit -m "Delete temp.js from main branch"

-----------------------------------------------

9. Git Actions

=================

It is used to make automated pipeline for git repositories.

GitHub Actions allows you to:

------------------------------

Automate everything around your Git process

Improve code quality through CI (tests, lint)

Enable CD for seamless deployments

Reduce manual effort with smart, event-driven workflows

--------------------------------------------------

Eevnts --> Workflow --> Actions

=====================================================

**Git Commands Explanation:**

=================================================

1. creates and initializes the repository.

2. It adds all files folders of current project into repository.

git add filename.txt => to add specific file into repository

3. git config user.email kodavali.lakshmi@gmail.com

4. git config --global user.name kln@sairam

providing github username and email, it will connect with github

5. git remote add origin https://github.com/Y23\_MSWD\_DoorStep\_7894.git

6. To know the status of git connectivity

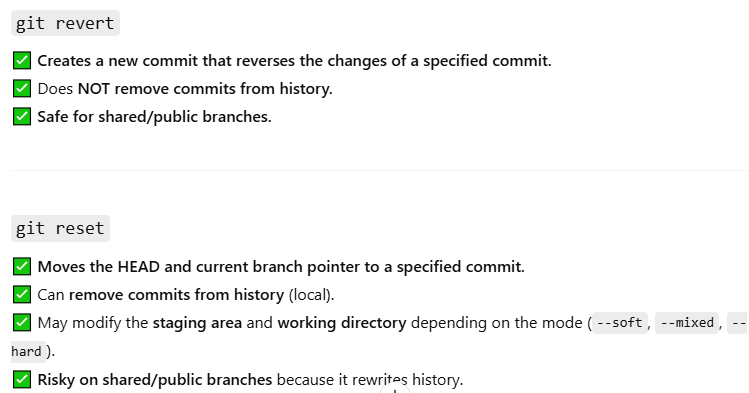
7. To commit(save) the folder in local git. -m (message)

8. -u (update) push into either either main or master

===============================================================

=============================================================

git reset .



    git logs  
  
  git log

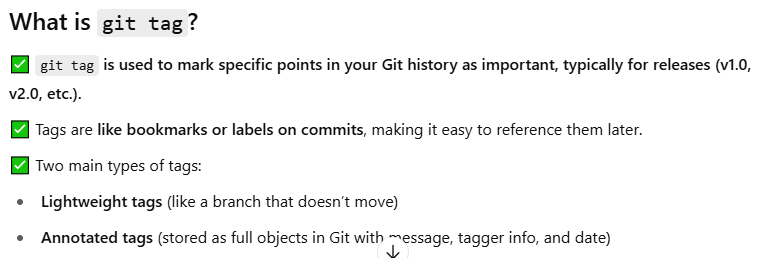
commit 3b12650e7407596de6e7ebc4c8bbf2c2c6f0f677 (HEAD -> master, origin/master)  
  
  git revert 6e5a74263cc5916025413767e7f986703b9623ee

Syntax: git revert <commit-hash>

It removes files from the local repository

Does **NOT remove commits from history.**

  git tag -a signup -m "siginup is created" 6e5a74263cc5916025413767e7f986703b9623ee



git tag v1.0

git tag -a v1.1 -m "Release version 1.1"

git tag //list all tags

git show v1.0 //show tag details

git push origin v1.4 // pushes with release number.

git tag -d v1.0 //delete a tag

