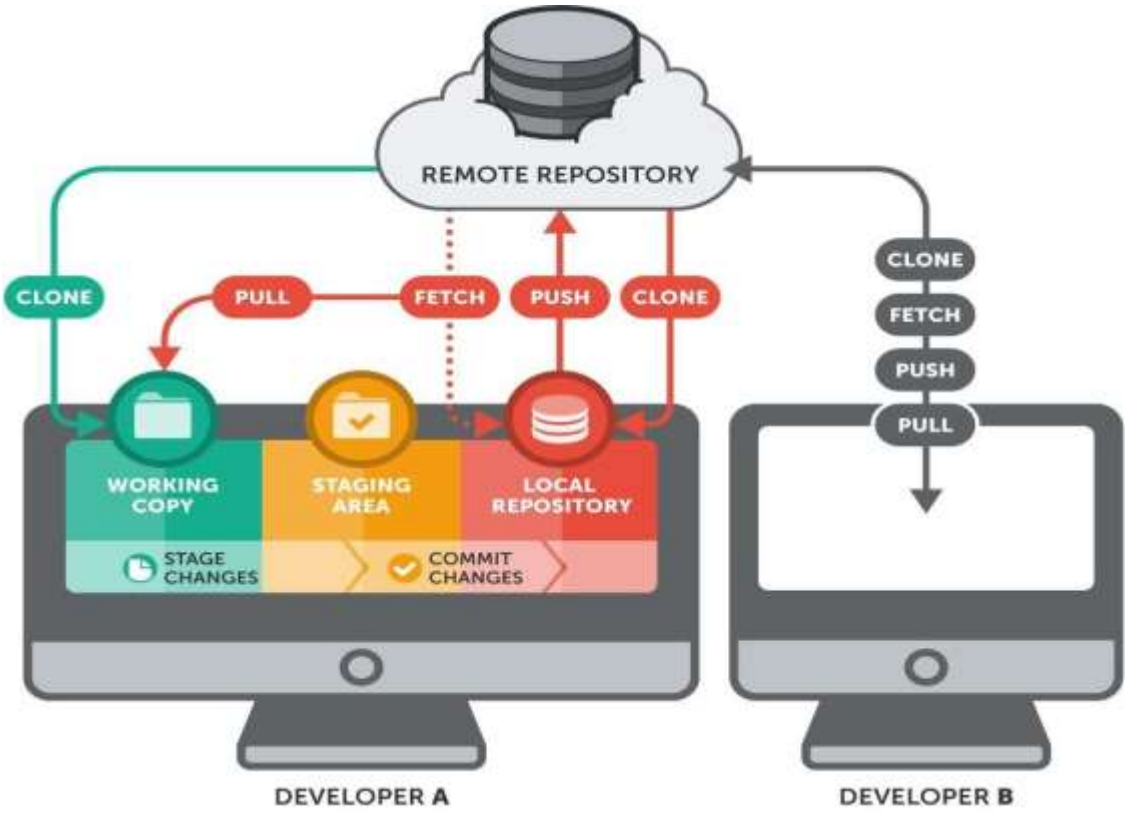
Week 2: Explore Git and GitHub commands.

A version control system (VCS) is a software tool that helps manage changes to source code over time. It allows multiple developers to collaborate on a project, track changes, and revert to previous versions when needed. Version control systems are essential in software development to maintain code integrity, facilitate teamwork, and manage the complexity of evolving projects.

Git is one of the most popular distributed version control systems. Introduced by Linus Torvalds in 2005, Git provides a distributed and decentralized model where each developer has a complete copy (clone) of the entire repository on their local machine. This allows developers to work offline, commit changes locally, and synchronize their work with others using remote repositories. **GitHub** is a web-based hosting service for Git repositories.

Why Git in DevOps?

- Supports Continuous Integration (CI) workflows
- Enables collaboration and rollback
- Integrates with build and deployment tools (Jenkins, Docker)



1. Working Directory

Definition: The working directory is where you make changes to your files. It contains the files you are currently editing and working on.

Example: If you have a project folder on your computer with files like register.html, sample.txt, the folder and its contents are your working directory.

2. Staging Area

Definition: The staging area (also known as the index) is a place where you prepare changes before committing them.

Example: Suppose you modified register.html and sample.txt. If you only want to commit changes made to register.html for now, you would add register.html to the staging area using git add register.html. The changes to sample.txt remain in the working directory but are not yet staged.

3. Local Repository

Definition: The local repository is where Git stores your commits, branches, and history on your local machine. It's the place where you commit your changes and manage your project.

Example: After staging and committing changes, the Git records are stored in your local repository. You can check the history of your commits with git log, and your commits are saved in your local.git directory.

4. Remote Repository

Definition: The remote repository is a version of your project stored on a server or a platform like GitHub, GitLab, or Bitbucket. It allows you to share and collaborate on your project with others.

Example: If you push your local commits to a repository on GitHub using git push origin master, you are updating the remote repository at https://github.com/username/repository.git with your local changes.

1. Install Git: Download and install Git for Windows from the official site:

https://git-scm.com/downloads/win

- During installation: Choose "Use Git from the command line"
- Keep other default settings,
- After installation, open Command Prompt (CMD) or Git Bash

2. Create a GitHub Account:

Sign up at: https://github.com

Use your official/student email ID.

3. Create or Navigate to a Project Folder

Navigate to your application folder

C:\Users\CC5\Documents\1265\

4. Configure Git (First Time Only)

These commands set your name and email for commits:

git config --global user.name "Your Name"

git config --global user.email "your-email@example.com"

--global: applies to all projects on your system

--local: applies only to the current project folder

5. Verify configuration:

git config --list

You should see your name and email listed.

6. Initialize a Local Git Repository

git init

This creates a hidden .git folder. Git will now track changes in this folder.

7. Check File Status

git status

Shows untracked, modified, or staged files.

8. Stage Files for Commit

selecting the changed files you want to include in your next commit.

git add app.py # add a specific file

git add . # add all files

9. Commit Changes

git commit -m "Committed all files”

saving a snapshot of your code with a message

10. Connect to Remote GitHub Repository

Step 1: Create a GitHub Repo

Example: https://github.com/sriludone/Sree.git

Step 2: Add the remote

git remote add origin https://github.com/sriludone/Sree.git

Step 3: Verify remote

git remote -v

11. Push Code to GitHub

First Push:

git push -u origin master

This command uploads your local master branch to the remote repository (GitHub) and sets a tracking link so you can use git push next time without full details.

-u means --set-upstream

It tells Git to remember the connection between your local branch and the remote branch.

Later pushes:

git push

12. Pull Latest Changes from GitHub

git pull

Fetches the latest changes from the remote repository and merges them into your current local branch.

13. Fetch Changes

git fetch

downloads the latest changes from the remote repository, but does not apply (merge) them into your code.

13. Branching in Git: A branch in Git is like a separate version of your project. It allows you to make changes or add new features without affecting the main version of your project.

Main Branch (Master): This is your original branch with all the documents you currently have.

New Branch: You create a new branch to add your new files. This branch is like a separate copy of your book where you work on the new chapter.

- View branches:

  git branch

  shows all the branches in your local Git repository.

- Create a new branch:

  git branch new

- Switch between branches:

  git checkout main

- Merge branch into current:

  git merge new

15. Delete or Rename Files

- Delete from Git:

  git rm filename.txt

- Rename file:

  git mv oldname.txt newname.txt

15. Get Help for Any Command

git help <command>

git help commit

git help push

16. Clone an Existing Repository (From GitHub)

git clone https://github.com/sriludone/app.git

This creates a full local copy of the GitHub repository.