GIT

GIT classified into 2 types they are

1> CVCS (centralized version control system)

A Centralized Version Control System (CVCS) is a version control model where a single central repository stores all versions of a project, and developers must connect to it to access, commit, or update changes.

How CVCS Works:-

* A central server stores the project history and codebase.
* Developers check out files from the server.
* Changes are made locally and then committed back to the server.
* The latest version is always maintained in the central repository.
- It's popular tool was SVN (subversion control system)

Advantages of CVCS:-

> Easier to manage – Since all versions are stored in one place, administrators can control access and monitor changes easily.
> Faster for small teams – Works well for teams where everyone is online and connected to the central server.
> Consistent backups – The server has a complete history, so data is safe (as long as the server is maintained).

Disadvantages of CVCS:-

> Single point of failure – If the central server crashes, no one can access the code.
> Requires internet connectivity – Developers must be connected to commit or retrieve changes.
> Less flexibility – Branching and merging are more difficult than in Distributed Version Control Systems (DVCS) like Git.

2> DVCS (Decentralized version control system OR Distributed version control system)

A Distributed Version Control System (DVCS) is a version control model where every developer has a full copy of the repository, including its history. Unlike Centralized Version Control Systems (CVCS), a DVCS does not rely on a single central server to store all versions of a project.

How DVCS Works:-

* A developer clones (copies) the entire repository, including history, to their local machine.
* Changes are made and committed locally, without needing an internet connection.
* Developers can create branches and merge changes independently.
* When ready, changes are pushed to a remote repository (e.g., GitHub, GitLab, Bitbucket) for collaboration.
- It's popular tool was GIT

Advantages of DVCS

> Works offline – Developers can commit changes locally without needing a network connection.
> Faster performance – Since commits and history are stored locally, operations like diff, log, and revert are much quicker.
> No single point of failure – If a central server goes down, developers still have full copies of the repository.
> Efficient branching and merging – Branching is lightweight and easy, making it ideal for collaborative projects.
> Better collaboration – Multiple developers can work independently before integrating changes.

Disadvantages of DVCS

> Higher storage requirements – Every developer has a full copy of the repository, which can be large.
> Steeper learning curve – Tools like Git can be more complex for beginners.
> Potential conflicts – Since multiple copies exist, managing conflicts requires proper workflows (e.g., pull requests, rebasing).

** A repository is a storage location for a project's files and history in a version control system. It tracks changes, allows collaboration, and enables rollback to previous versions.

Types of Repositories:-

1. Local Repository
- Exists on a developer’s computer.
- Created using Git (git init).
- Changes are tracked locally before pushing to a remote repository.
2. Remote Repository/ Global Repository
- Hosted on platforms like GitHub, GitLab, or Bitbucket.
- Used for collaboration and backups.
- Developers clone, pull, and push updates to it.

```
   1. HTTPS (Hyper Text Transfer Protocal Security):-
- Most commonly used protocol in modern VCS like GitHub and GitLab.
- Secure with HTTPS encryption.
- Easier to configure (does not require special ports).
- Often slower than other protocols due to overhead encryption.
   2. SSH (Secure Shell):-
- Secure and faster than HTTP/HTTPS.
- Requires SSH keys for authentication.
- Common in remote repositories like GitHub and GitLab.
- Preferred for automation scripts and CI/CD pipelines.
     GIT COMMANDS:-
git --version:- Check installed Git version.
git config --global user.name "Your Name":- Set your Git username.
git config --global user.email "you@example.com":-  Set your Git email.
git config --list:- View current Git configuration.
git init:-  Initialize a new Git repository.
git commit -m "Initial commit"
git commit -m "Added new file"
git commit -m "Your commit message":-   Commit changes with a message.
git commit -am "Message":-  Stage & commit tracked files in one step.


```