**GIT**

**INDEX:**

1. **Introduction to Git – Syllabus**
2. [What is Version Control](#xps4ck9nncgm)
3. Types of Version Control (Centralized vs Distributed)
4. What is Git
5. Features of Git
6. Advantages of Git
7. Git Installation
8. Git Configuration (git config)
9. Git Basic Workflow
10. Git Architecture (Working Directory, Staging Area, Repository)
11. Important Git Terminologies (Repository, Commit, Branch, Merge, Clone, Remote)
12. Difference between Git and GitHub

Introduction to GIT:

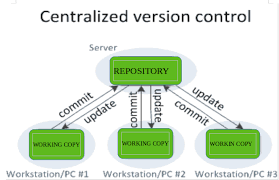
**Version Control System**:

* Version Control is a system that records changes to files over time so you can track history, collaborate with others, and revert to an earlier version if needed
* In Software development, it’s mainly used to manage source code, allowing multiple developers to work on the same project without overwriting each other’s changes
* VCS tracks documentation versions and hence can be traced or tracked accordingly, and can be reverted to previous versions
* VCS tracks history, keeping track of aspects such as what was changed, who changed it, when it was changed, and a short message explaining *why*
* VCS allows you to revert to the previous version by restoring files to an earlier state in history
* VCS is used by
  + **Software Developers**: To manage source code, track bugs, add features, and collaborate on projects.
  + **DevOps / System Admin**: To version control infrastructure and configuration files (e.g., using Git + Terraform).
  + **Technical Writers & Documentation Teams**
  + **Data Scientists / ML Engineers:** To track versions of datasets, Jupyter notebooks, and models.
  + **Designers / Artists**
  + **Students & Researchers** -
  + and many others
* Version control is also referred to as **Source Code management (SCM)**, **Revision control management (RCM)**, and **Source Control**. All these terms mean the same core idea: **managing and tracking changes to files (especially code) over time**.
* The most popular version control tool is **GIT**
* If we don’t use a Version Control System (VCS):
  + We can’t track changes or history.
  + Team members may overwrite each other’s work.
  + It’s hard to undo mistakes.
  + File management becomes messy (project\_v1\_final\_final2).
  + Collaboration and experimentation are very difficult.

**Types of Version Control (Centralized vs Distributed)**

**Centralized**

* Centralized Version Control System, also called Client-Server Version Control System, is a version control model where there is a single central server that stores the complete codebase and its history
* Developers or clients connect to this server to check out, commit, and update code
* **Single central repository**: All project files and version history are stored on one server
* **Clients connect to the server**: Developers download (check out) the latest version from the server and commit changes back to it.
* **Network dependency**: You need network dependencies to the server to commit or retrieve updates
* **Examples**: CVS, Subversion (SVN), Perforce.
* It works like a **"hub-and-spoke"** system — the **server is the hub**, and all developers (clients) connect to it to collaborate.



* **Server (Central Repository):** the main machine that stores the entire project's code and history; it acts like a library where everyone checks in and out from a single place
* **Clients (Developers’ Machines):** Each developer connects to the central server over a network (LAN or Internet). And usually, the client keeps only the latest snapshot or the files instead of the full history, makes changes to it, and commits it to the server. Any operation like commit, update, branch, or merge requires communication with the server.

**How the connection works (step-by-step):**

* **Checkout/Update**: A developer requests files → server sends them the latest version.
* **Edit**: Developer makes changes locally.
* **Commit**: Developer pushes changes back to the server → server updates the central repository.
* **Sync with others**: Other developers must update (pull from the server) to see those new changes.