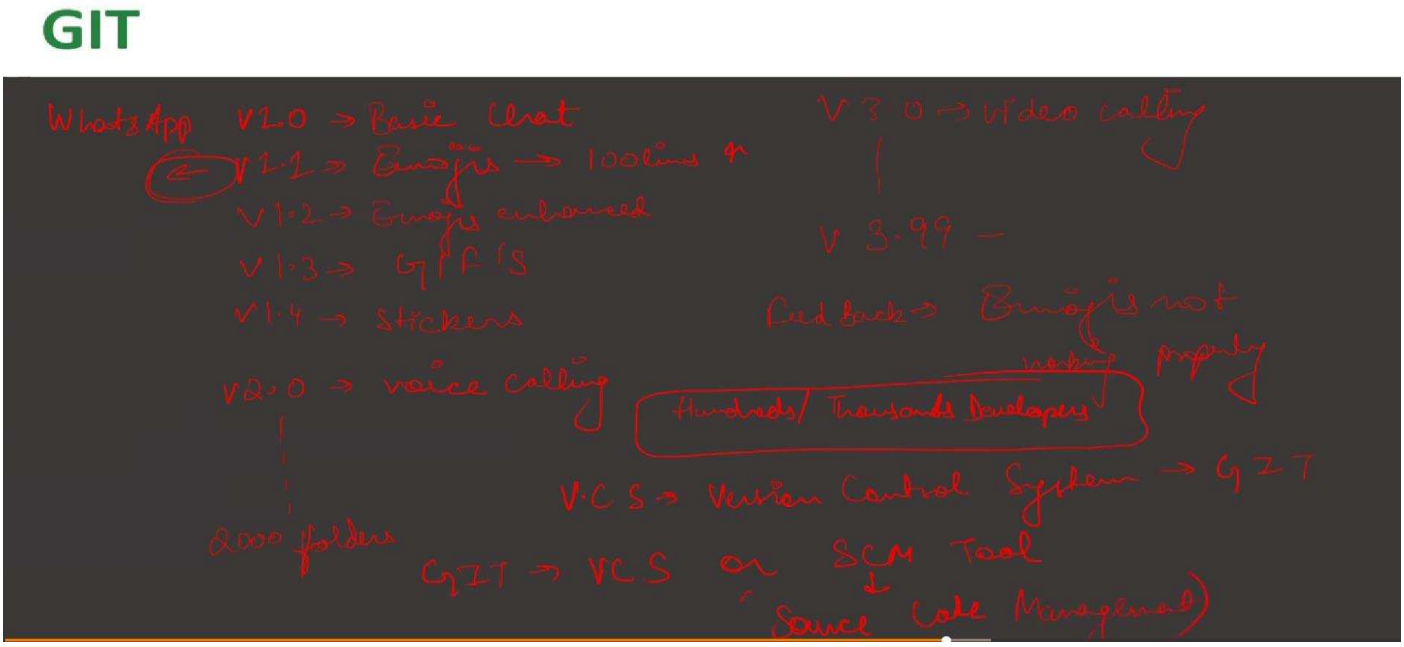
❖ What is GIT?

Git is an **open-source distributed version control system**. It is designed to handle minor to major projects with high speed and efficiency. It is developed to co-ordinate the work among the developers. The version control allows us to track and work together with our team members at the same workspace.

Git is foundation of many services like **GitHub** and **GitLab**, but we can use Git without using any other Git services. Git can be used **privately** and **publicly**.

Git was created by **Linus Torvalds** in **2005** to develop Linux Kernel. It is also used as an important distributed version-control tool for **the DevOps**.

Git is easy to learn and has fast performance. It is superior to other SCM tools like Subversion, CVS, Perforce, and ClearCase.



Why Use Git?

You need to know that around 70% of developers worldwide use Git for development. Some of the prominent reasons for using Git are:

- *Developers can work together from anywhere.*
- *Developers can see the full history and can compare the previous and new changes of the project.*
- *Developers can retreat to earlier versions of a project.*

Working with Git

When a folder is initialised with Git, it becomes a repository—a special location where Git logs all changes made to a hidden folder. In that folder, each time you change, add, or remove a file, Git takes note of the change and marks the file as "modified." You can choose which modified files you want to save by staging them, so don't worry. Consider staging as getting the changes ready for a particular snapshot that you want to keep. Once the staged changes are to your satisfaction, commit them, and Git will keep a permanent copy of those files in its history.
Git is great because it maintains a complete record of each commit you make, allowing you to see

What is Github?

GitHub is a hosting service for Git repositories and if you have a project hosted on GitHub, you can access and download that project with commands on any computer you have access and make your changes and push the latest version back to GitHub.
GitHub allows you to store your repo on their platform. It is also comes with GitHub, ability to collaborate with other developers from any location.

Version Control System

Version Control System (VCS) is a software that helps software developers to work together and maintain a complete history of their work.

Listed below are the functions of a VCS –

- Allows developers to work simultaneously.
- Does not allow overwriting each other's changes.
- Maintains a history of every version.

Following are the types of VCS –

- Centralized version control system (CVCS).
- Distributed/Decentralized version control system (DVCS).

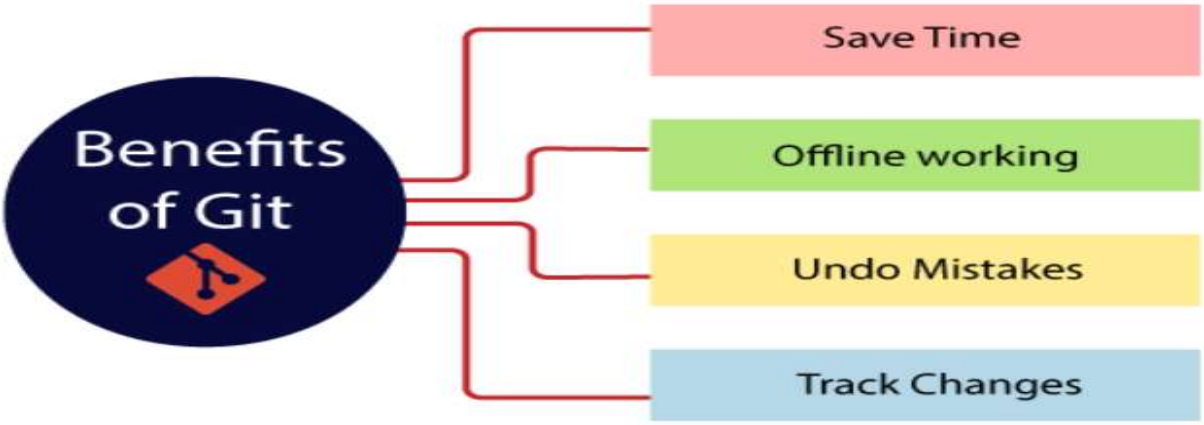
In this chapter, we will concentrate only on distributed version control system and especially on Git. Git falls under distributed version control system.

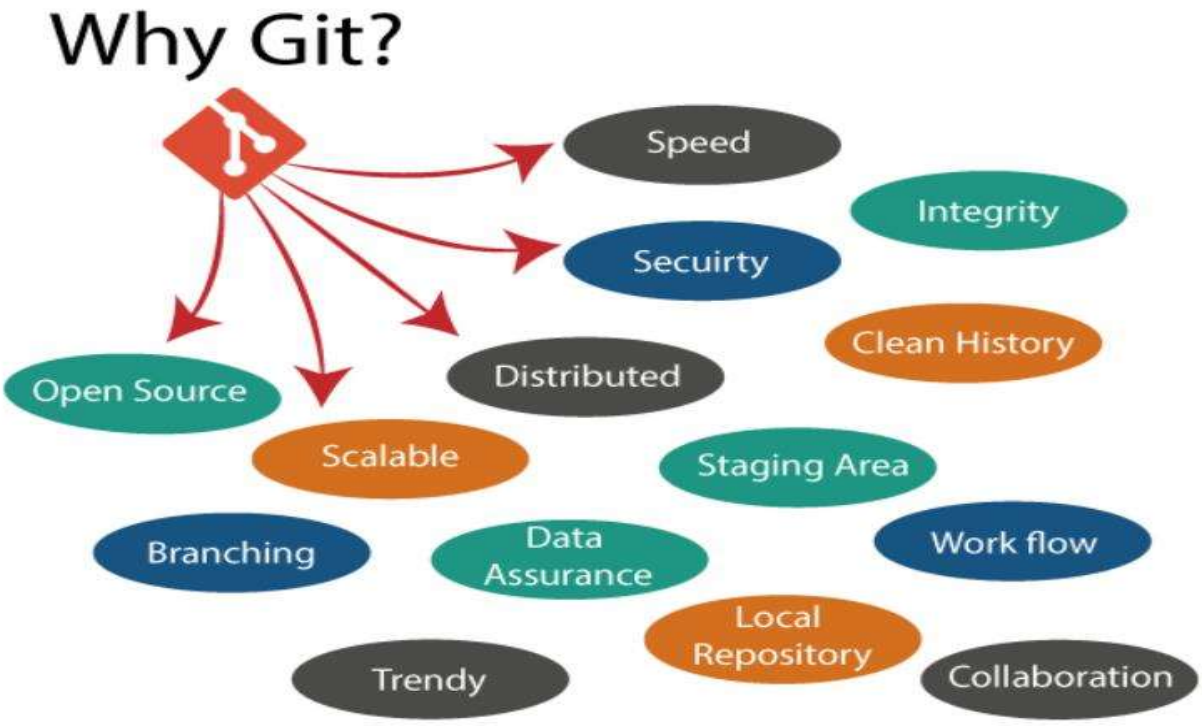
Distributed Version Control System

Centralized version control system (CVCS) uses a central server to store all files and enables team collaboration. But the major drawback of CVCS is its single point of failure, i.e., failure of the central server. Unfortunately, if the central server goes down for an hour, then during that hour, no one can collaborate at all. And even in a worst case, if the disk of the central server gets corrupted and proper backup has not been taken, then you will lose the entire history of the project. Here, distributed version control system (DVCS) comes into picture.

DVCS clients not only check out the latest snapshot of the directory but they also fully mirror the repository. If the server goes down, then the repository from any client can be copied back to the server to restore it. Every checkout is a full backup of the repository. Git does not rely on the central server and that is why you can perform many operations when you are offline. You can commit changes, create branches, view logs, and perform other operations when you are offline. You require network connection only to publish your changes and take the latest changes.


Scalability
Security
Open Source
Features of Git
Speed


Benefits of Git
Save Time
Offline working
Undo Mistakes
Track Changes


Why Git?
Speed
Integrity
Secuirty
Clean History
Open Source
Distributed
Scalable
Staging Area
Branching
Data Assurance
Work flow
Local Repository
Trendy
Collaboration