Git a 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.
There are multiple version control systems as CVS, SVN, GIT, Mercurial etc.

Functions of a Version Control System:

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

There are two types of Version Control System:

- Centralised version control system (CVCS).
- Distributed/Decentralised version control system (DVCS).

Git falls under distributed version control system.

Distributed Version Control System VS Centralised version control system

- Centralised 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.

About Git :

- Git is a distributed version-control system for tracking changes in source code during software development.
- It is designed for coordinating work among programmers, but it can be used to track changes in any set of files.
- Its goals include speed, data integrity, and support for distributed, non-linear workflows.
- Git was created by Linus Torvalds in 2005 for development of the Linux kernel, with other kernel developers contributing to its initial development.

- Its current maintainer since 2005 is Junio Hamano.
- As with most other distributed version-control systems, and unlike most client–server systems, every Git directory on every computer is a full-fledged repository with complete history and full version-tracking abilities, independent of network access or a central server.
- Git is free and open-source software distributed under the terms of the GNU General Public License version 2.

Advantages of Git :

Free and open source

Git is released under GPL's open source license.

It is available freely over the internet. You can use Git to manage property projects without paying anything.

As it is an open source, we can download its source code and also perform changes according to your requirements.

Fast and small

As most of the operations are performed locally, it gives a huge benefit in terms of speed.

Git does not rely on the central server; that is why, there is no need to interact with the remote server for every operation.

The core part of Git is written in C, which avoids runtime overheads associated with other high-level languages.

Though Git mirrors entire repository, the size of the data on the client side is small.

This illustrates the efficiency of Git at compressing and storing data on the client side.

Implicit backup

The chances of losing data are very rare when there are multiple copies of it. Data present on any client side mirrors the repository, hence it can be used in the event of a crash or disk corruption.

Security

Git uses a common cryptographic hash function called secure hash function (SHA1), to name and identify objects within its database. Every file and commit is check-summed and retrieved by its checksum at the time of checkout. It implies that, it is impossible to change file, date, and commit message and any other data from the Git database without knowing Git.

No need of powerful hardware

In case of CVCS, the central server needs to be powerful enough to serve requests of the entire team. For smaller teams, it is not an issue, but as the team size grows, the hardware limitations of the server can be a performance bottleneck. In case of DVCS, developers don't interact with the server unless they need to push or pull changes. All the heavy lifting happens on the client side, so the server hardware can be very simple indeed.

Easier branching

CVCS uses cheap copy mechanism, If we create a new branch, it will copy all the codes to the new branch, so it is time-consuming and not efficient. Also, deletion and merging

of branches in CVCS is complicated and time-consuming. But branch management with Git is very simple. It takes only a few seconds to create, delete, and merge branches.
DVCS Terminologies

Local Repository

Every VCS tool provides a private workplace as a working copy. Developers make changes in their private workplace and after commit, these changes become a part of the repository. Git takes it one step further by providing them a private copy of the whole repository. Users can perform many operations with this repository such as add file, remove file, rename file, move file, commit changes, and many more.

About GitHub

- GitHub Inc. is a web-based hosting service for version control using Git.
- It is mostly used for computer code.
- It offers all of the distributed version control and source code management (SCM) functionality of Git as well as adding its own features.
- It provides access control and several collaboration features such as bug tracking, feature requests, task management, and wikis for every project.
- GitHub offers plans for enterprise, team, pro and free accounts which are commonly used to host open-source software projects.
- From January 2019, GitHub now offers unlimited private repositories to all plans, including free accounts.
- As of June 2018, GitHub reports having over 28 million users and 57 million repositories (including 28 million public repositories), making it the largest host of source code in the world.

Features provided by GitHub

- **Github Gist :** Allows GitHub users to share pieces of code or other notes.

- **GitHub Flow :** Is a lightweight, branch-based workflow for regularly updated deployments.

- **GitHub Pages :** Are static webpages to host a project, pulling information directly from an individual's or organization's GitHub repository.

- **GitHub Desktop :** Enables users to access GitHub from Windows or Mac desktops, rather than going to GitHub's website.

- **GitHub Student Developer Pack :** Is a free offering of developer tools that is limited to students, and includes cloud resources, programming tools and support, and GitHub access.