1st Tool of DevOps

- VCS – Version Control System
- SCM – Source Code Management
- LVCS – Local Version Control System – Primarily used by Solo Developers rather than teams. **Drawbacks of LVCS** – 1. Single Point of Failure, 2. Limited Collaboration, 3. Backup issues, 4. Complexity in merging changes, 5. Lack of Transparency
- CVCS – Centralized Version control System – SVN
- **Drawbacks of CVCS** – **Single Point of Failure**: If central server goes down, nobody can commit changes, and the entire team's work can be halted. Limited Offline access, Scalability issues, Security Risks, Dependency on the Central Server.
- DVCS – Distributed Version Control System – Git, GitHub, Bitbucket

Why we need VCS:

1. For collaboration
2. Storing versions (update release)
3. Figuring what happened (changes made)
4. Backup

Issues without version control

1. Once saved, all the changes made in the files are permanent and cannot be reverted back.
2. No record of what was done and by whom.
3. Downtime that can occur because of a faulty update could cost millions in losses.

What is version control?

- Version control is a system that documents changes made to a file or a set of files. It allows multiple users to manage multiple revisions of the same unit of information. It is a snapshot of our project over time.

Local Version Control (LCV)

- The practice of having the Version Database in the local computer.
- Local database keeps a record of the changes made to files in version database.

Local Version Control: Issue

Issue: Multiple people parallelly working on the same project
Solution: Centralized Version control

Centralized Version Control (CVC)

- Local Version Control's issues are resolved by Centralized Version Control
- In CVC, a central repository is maintained where all the versioned files are kept
- Now users can checkout and check-in files from their different computers at any time.

Centralized Version Control: Issue

- **Issue**: In case of central server failure whole system goes down
- **Solution**: Distributed Version Control System

Distributed Version Control System

- Version Database is stored at every user's local system and at the remote server
- Users manipulate the local files and then upload the changes to the remote server
- If any of the servers die, a client server can be used to restore

Introduction to Git

- What is Git?
  Git is an open-source Distributed Version Control System (DVCS) which records changes made to the files laying emphasis on speed, data integrity and distributed, non-linear workflows.

Command to get current version of git in bash	:	git - -version
Command to get help on git commands	:	git –help

The Git File Workflow

The Remote Repository is the server where all the collaborators upload changes made to the files.

Remote Repository (GitHub)
↓
Clone
↓
Working Area
↓
Staging/Index Area
↓
Local repository
↓
Push

Other steps involved are: pull, fetch

- Local Repository is user's copy of the Version Database
- The user accesses all the files through local repository and then push the change made to the "Remote Repository".
- "**Workspace**" is user's active directory.
- The user modifies existing files and creates new files in this space. Git tracks these changes compared to your Local Repository.

- **Stage – Staging Area- Index Area** is a place where all the modified files marked to be committed are placed.
- **Clone** command creates a copy of an existing Remote Repository inside the Local Repository.
- **Push** command pushes all the changes made in the Local Repository to the Remote Repository.
- **Pull** like Fetch, gets all the changes from remote repository and copies them to the Local Repository.
- **Pull** merges those changes to the current working directory
- **.git folder:** .git folder is the empty repository which will be present when we clone anything from remote repository.
- When we clone the repo for the first time an empty repository we will having 2 objects .git folder and readme.md which is optional and it will show when we checkmark it while creating a new repo in GitHub.

  What is .git folder?
- A Git repository is the . git/ folder inside a project. This repository tracks all changes made to files in your project, building a history over time. Meaning, if you delete the . git/ folder, then you delete your project's history.