1. GIT- An Overview

Git is a distributed version control system that helps developers track changes in their codebase, collaborate with others, and manage project history.

Some of the terms related to Git:

* **Repository:** A repository is a collection of files. It can be local or remote.
* **Commit:** The changes made to the code at a given time.
* **Clone:** Copying a repository from a remote server to our local machine.
* **Pull:** Fetching changes from a remote repository and merging them into the current branch.
* **Push:** Uploading your local changes to a remote repository.
* **Branch:** A branch is a parallel repository version, allowing you to work on different features without affecting the main codebase.
* **Merge:** Combining changes from one branch to another.
* **Merge Conflict:** This occurs when Git is unable to automatically merge changes from different branches.
* **Pull Request:** A request to merge changes from one branch into another.
* **Fork:** Creating a personal copy of someone else's project on a remote server.

2. Basic Git Commands

1. Initialize Repository: `git init`

2. Clone Repository: `git clone <URL of the repository>`

3. Check Status: `git status`

4. Create a branch: `git branch <branch-name>`

5. Pull changes: `git pull origin <branch name>`

6. Push changes: `git push origin <branch name>`

7. Delete branch: ` git branch -d <branch-name>`

3. GITHUB

GitHub is a web-based platform built on top of Git, providing additional features and functionalities for collaborative software development.

1. Repository: A repository on GitHub is similar to a Git repository, containing files, commit history, and branches.
2. Fork: Forking a repository creates a personal copy of someone else's project under your GitHub account.
3. Pull Request: A Pull Request is a way to propose changes to a repository.
4. Issues: Issues can be assigned, labeled, and linked to Pull Requests.
5. Collaborations: Collaborators are individuals or teams with access to a repository.

GITLAB

GitLab is a web-based DevOps lifecycle tool that provides source code management, continuous integration, and continuous deployment. Some of the few concepts in GitLab:

1. Repository: The GitLab repository is where your project's source code and history are stored.
2. Project: In GitLab, a project is a container for repositories.
3. Group: A group in GitLab is a collection of projects.
4. Fork: Similar to GitHub, forking in GitLab creates a personal copy of a project under your GitLab account.
5. Merge Request: Equivalent to a Pull Request in GitHub, an MR is a request to merge changes from one branch into another.
6. Issue Tracker: GitLab includes an issue-tracking system for managing tasks, bugs, and enhancements.

BITBUCKET

Bitbucket is a web-based platform that provides Git and Mercurial code repositories along with features for code collaboration, continuous integration, and deployment. Here are key concepts related to Bitbucket:

1. Repository: Similar to Git and GitLab, a Bitbucket repository is a container for source code, allowing for version control and collaboration.
2. Branches: Bitbucket supports branching, allowing developers to work on separate features or bug fixes without affecting the main codebase.
3. Pull Request: In Bitbucket, a Pull Request is a request to merge changes from one branch into another.
4. Fork: Similar to GitHub and GitLab, forking in Bitbucket creates a personal copy of a repository under your Bitbucket account.
5. Issues: Bitbucket’s issue-tracking system helps manage tasks, bugs, and enhancements associated with a repository.

Industrial Practices of Using Git

1. **Version Control:**
   * Branching Strategies**:** Companies often adopt branching strategies like Git Flow or GitHub Flow to manage feature development, releases, and hotfixes effectively.
   * Semantic Versioning: Semantic versioning is commonly used to manage software version numbers in a standardized way.
2. **Collaborative Development:**
   * Pull Request Workflow**:** Developers submit Pull Requests (or Merge Requests) for code review before merging changes into the main branch. This ensures quality control and collaboration.
   * Code Review Guidelines**:** Establishing clear code review guidelines helps maintain code quality and consistency across the codebase.
3. **Continuous Integration/Continuous Deployment (CI/CD):**
   * Automated Builds: CI/CD pipelines are set up to automate the build process, run tests, and deploy applications.
   * Integration Testing: Automated testing, including unit tests and integration tests, is integrated into the CI/CD process to catch bugs early.
4. **Documentation:**
   * Code Documentation: Encouraging developers to document code helps with onboarding, maintenance, and collaboration.
5. **Backup and Disaster Recovery:**
   * Repository Backups: Regular backups of Git repositories are maintained to prevent data loss in case of hardware failures or other unforeseen events.

5. Cloning a Repository to Local

1. Open the terminal or command prompt and navigate to the directory where you want to clone the repository.

2. Use the command git clone <URL of the repository> to clone.

3.Check the cloned repository using cd command.

