1. What is Git?

- It is a popular version control system which is created by Linus Torvalds in2005 and maintained by Junio Humano.
- It is used for tracking the history of project or files, tracking code changes and coding collaboration.

2. What do you understand by the term 'Version Control System'?

- It is a software tool
- It helps track changes to file and directories over time.
- It enables multiple contributors to work on a project simultaneously, managing different versions of file and keeping a history of modifications.
- It also allows users to revert to previous states of the project, compare, and collaborate more effectively.

3. What is GitHub?

- GitHub is a web-based platform that serves as a hosting service for version control using Git.
- It allows developers to collaborate on projects, manage and track changes to their code, and coordinate work within a team.
- GitHub provides features such as repositories for hosting code, pull requests for proposing and discussing changes, issue tracking, and project management tools.
- Developers can use GitHub to store their projects, share them with others, and contribute to open-source projects.
- It facilitates collaboration by allowing multiple contributors to work on the same project simultaneously, providing a centralized place for managing code changes and documentation.

4. Mention some popular Git hosting services.

There are following Git hosting services.

a) GitHub: One of the most widely used platforms, offering both public and private repositories. It provides a user-friendly interface, collaboration tools, and integration with various third-party services.

b) GitLab: Offers a web-based Git repository manager with features like code review, continuous integration, and container registry. GitLab can be used both as a hosted service and as a self-hosted solution.

c) BitBacket: Provides Git and Mercurial repository hosting. It offers free repositories with a limited number of contributors for small teams and integrates with other Atlassian tools like Jira for project management.

d) AWS CodeCommit: A fully managed source control service by Amazon Web Services (AWS) that hosts secure and scalable Git repositories. It integrates well with other AWS services.

e) Microsoft Azure Repos: Part of the Azure DevOps Services, it provides Git version control with unlimited private repositories. It integrates seamlessly with other Azure DevOps tools for continuous integration and delivery.

f) SourceForge: An older platform that hosts not only Git but also other version control systems. It offers features like bug tracking, mailing lists, and forums.

g) Gitea: A self-hosted Git service written in Go. It is lightweight and easy to install, making it a good option for those who prefer to host their Git repositories.

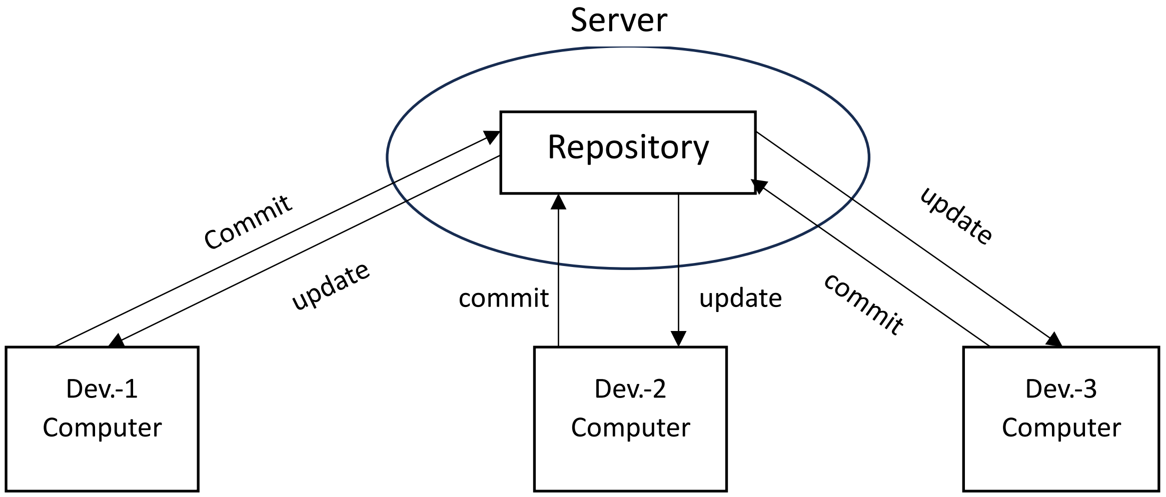
5. Different types of version control systems.

There are two types of version control systems:

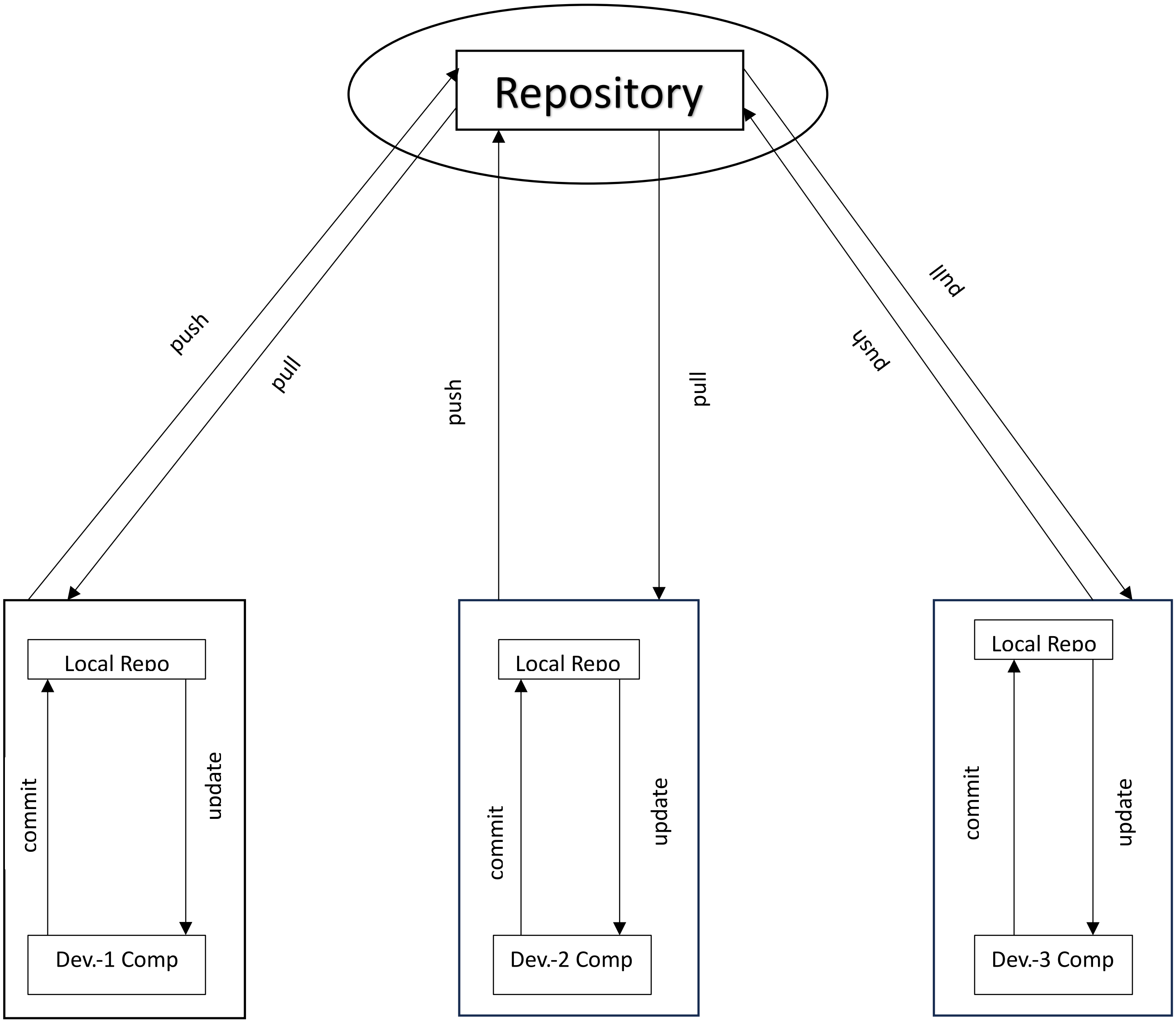
a) Centralized version control systems

- In CVCS, there is a central server that contains the repository, and all team members interact with this central repository to access and update files.
- Team members check out files from the central repository, make changes locally, and then commit those changes back to the central server.

  Ex- CVS, SubVersion, Perforce.



1. Centralized Version Control System



2. Distributed Version Control System

b) Distributed Version Control systems

- In DVCS, each team member has a complete copy of the repository, including the entire history of the project.
- Team members can work independently, commit changes locally, and later synchronize with other repositories as needed.
- DVCS allows for more flexibility and autonomy, as team members can work offline and commit changes to their local repository before pushing them to a central server or sharing with others.
  Ex- Git, Mercurial, Dars, Bazaar etc

6. What benefits come with using Git?

Using Git as a version control system provides several benefits for software development and collaboration. Some key advantages include:

- Distributed Version Control: Git is a distributed version control system (DVCS), allowing each team member to have a complete copy of the repository. This enables offline work, independence, and flexibility in development.
- Branching and Merging: Git provides powerful and efficient branching and merging capabilities. Developers can easily create branches for new features or bug fixes, work on them independently, and later merge changes back into the main branch.
- Speed and Performance: Git is designed to be fast and performant. Operations like committing changes, branching, and merging are typically quick, contributing to a smooth development workflow.
- Data Integrity: Git uses a content-addressable file system, ensuring the integrity of the data. Each change is uniquely identified by a hash, and any corruption or loss of data can be easily detected.
- Collaboration and Remote Work: Git facilitates collaboration among developers, regardless of their physical location. It allows multiple contributors to work on the same project simultaneously, share code changes, and collaborate seamlessly.
- Open Source and Community Support: Git is open source, widely adopted, and has a large and active community. This community support ensures continuous improvement, the availability of resources, and a wealth of third-party tools and integrations.
- Flexible Workflow: Git is adaptable to various workflows. Whether using a centralized approach, feature branching, or Gitflow, developers can choose a workflow that best fits their project requirements.
- Integration with Other Tools: Git integrates well with a variety of development tools and services. Platforms like GitHub, GitLab, and Bitbucket provide additional features for code hosting, collaboration, and continuous integration.
- Security: Git has robust security features, including user authentication, access controls, and the ability to sign commits. It helps protect the codebase and ensures that only authorized users can make changes.
- Version History and Rollbacks: Git maintains a complete history of changes to the codebase. This allows developers to track the evolution of the project, understand who made specific changes, and easily roll back to previous states if needed.

7. What is a Git repository?

A Git repository is a storage location where a collection of files and directories, along with their complete version history, is managed by the Git version control system. It serves as a central hub for a project, allowing developers to track changes, collaborate, and maintain the integrity of their codebase.

8. How can you initialize a repository in Git?

To create a new repository, I will use the git init command. git init is a one-time command which I use during the initial setup of a new repository. Executing this command will create a new .git subdirectory in your current working directory. This will also create a new main branch.