# Viva Questions and Answers

## 1. Git

- Q: What is Git and why is it used?

A: Git is a distributed version control system used to track changes in source code during software development. It allows multiple developers to work on the same codebase simultaneously, manages code history, supports branching and merging, and ensures collaboration without conflicts.

- Q: How do you initialize a Git repository?

A: You can initialize a Git repository in a project directory by running `git init`. This creates a `.git` folder where Git stores all version history and configuration.

- Q: What is the difference between git add, git commit, and git push?

A: `git add` stages changes for the next commit, `git commit` records the changes in the local repository with a message, and `git push` uploads the committed changes to a remote repository like GitHub.

- Q: What does git status show?

A: `git status` displays the state of the working directory and staging area. It shows which files are modified, staged, or untracked, helping you know what is ready to be committed.

- Q: How can you view the commit history?

A: Use `git log` to view a list of previous commits along with details like commit ID, author, date, and commit message.

- Q: What is a merge conflict and how do you resolve it?

A: A merge conflict occurs when Git cannot automatically merge code due to conflicting changes in the same part of a file. To resolve it, you need to manually edit the file to fix the conflicts, then add and commit the resolved file.

## 2. GitHub / GitLab

- Q: What is the difference between Git and GitHub/GitLab?

A: Git is a tool for version control, while GitHub and GitLab are web-based platforms for hosting Git repositories, enabling collaboration, pull requests, issue tracking, and CI/CD integration.

- Q: What is a remote repository?

A: A remote repository is a version of your project hosted on the internet or another network. It's where collaborators can push and pull changes.

- Q: How do you clone a repository from GitHub/GitLab?

A: Use `git clone <repository-URL>`. This copies the repository, including all files, history, and branches, to your local system.

- Q: What is a Pull Request (PR)?

A: A Pull Request is a way to propose changes in a repository. It allows reviewers to discuss, review, and approve code before it is merged into the main branch.

- Q: Have you created or contributed to any project on GitHub/GitLab?

A: Yes, I created a personal notes app project on GitHub, and also contributed by fixing bugs in an open-source calculator repository.

- Q: What are branches and why are they used?

A: Branches are parallel versions of a repository that allow you to work on different features or fixes without affecting the main codebase. They make development more organized and prevent conflicts.

## 3. Jenkins

- Q: What is Jenkins and how is it useful in DevOps?

A: Jenkins is an open-source automation server used in DevOps to automate the building, testing, and deploying of software. It supports continuous integration and delivery, reducing manual work and improving deployment efficiency.

- Q: What is a Jenkins pipeline?

A: A Jenkins pipeline is a series of automated steps that define the process of building, testing, and deploying code. It can be scripted using Jenkinsfile, allowing for complex workflows.

- Q: What is the role of plugins in Jenkins?

A: Plugins extend Jenkins' functionality. For example, Git plugin helps with code checkout, Docker plugin helps build containers, and Slack plugin can send build notifications.

- Q: How do you trigger a build in Jenkins?

A: Builds can be triggered manually, through a scheduled time (cron jobs), after a Git commit (webhooks), or by another project.

- Q: Can Jenkins integrate with GitHub? If yes, how?

A: Yes, Jenkins can integrate with GitHub using webhooks and the Git plugin. When changes are pushed to GitHub, Jenkins is notified and can automatically start a build process.

## 4. Docker

- Q: What is Docker and why is it popular in DevOps?

A: Docker is a platform used to develop, ship, and run applications inside lightweight containers. It's popular because it ensures consistency across environments, speeds up deployment, and reduces dependency issues.

- Q: What is the difference between a container and a virtual machine?

A: Containers share the host OS kernel and are more lightweight, while virtual machines run separate OS instances on a hypervisor, consuming more resources.

- Q: What is a Dockerfile?

A: A Dockerfile is a script containing instructions on how to build a Docker image. It includes commands to install software, copy files, and configure the environment.

- Q: How do you run a container using Docker?

A: Use `docker run <image-name>` to create and start a container from an image.

- Q: What command is used to list running containers?

A: `docker ps` lists all currently running containers along with their IDs, names, and status.

## 5. Ansible

- Q: What is Ansible and what is it used for?

A: Ansible is an open-source automation tool used for configuration management, application deployment, and task automation. It simplifies managing multiple systems simultaneously.

- Q: What is an Ansible playbook?

A: A playbook is a YAML file that defines a series of tasks to be executed on managed hosts. It allows infrastructure to be described as code.

- Q: How is Ansible agentless?

A: Unlike other tools, Ansible doesn't require any agent software on client machines. It uses SSH to connect and execute tasks remotely.

- Q: What is the inventory file in Ansible?

A: The inventory file lists the target servers (hosts) and groups on which Ansible should run tasks. It can be in INI, YAML, or dynamic format.

- Q: Have you written or executed any simple playbook?

A: Yes, I wrote a playbook to install and start Apache on a remote Ubuntu server, which helped me understand basic automation.

## Generic Questions

- Q: What is the purpose of using version control systems?

A: Version control systems like Git track changes in code, support collaboration among developers, enable rollbacks, and provide a history of project evolution. They are essential for managing projects efficiently.

- Q: How do DevOps practices help in software development?

A: DevOps bridges the gap between development and operations. It improves collaboration, automates testing and deployment, reduces errors, and enables faster and more reliable software delivery.

- Q: What is CI/CD in simple terms?

A: CI/CD stands for Continuous Integration and Continuous Deployment/Delivery. CI automates testing when code is merged, and CD automates deployment to production, ensuring fast and consistent delivery of software updates.

- Q: Can you explain the workflow of your practical project?

A: In my project, I built a Notes App using React and Node.js. I used Git for version control and GitHub for hosting. Docker was used to containerize the backend. Jenkins was configured to run automated builds on code commits. The app was deployed using Docker Compose, and updates were managed through CI/CD pipelines.