What is version control?

• What is staging area?

• What do mean by repository?

• What care is to be taken when merging two branches?

• What is difference between git and github?

• How do you push your project into remote repository?

• Is it possible to revert changes after commit? Is so, how?

17. What is Continuous Integration?

18. What is CI/CD?

19. Which tools can be plugged with Jenkins?

• What is pipeline?

• What is declarative pipeline?

• What is scripted pipeline?

• What is Tomcat server?

• What is a web server?

• What are the other web servers available for deployment?

• What is a Container?

• Why Learn Docker?

What are docker images?

• What is a dockerfile?

• What is Docker hub?

• How do you create a docker container from an image?

• What is configuration management?

• Why we need configuration management?

• What are the other tools for configuration management?

1. **What is version control?** Version control is a system that records changes to a file or set of files over time so that you can recall specific versions later.
2. **What is a staging area?** The staging area is a part of Git, a version control system. It is a file, generally contained in your Git directory, that stores information about what will go into your next commit.
3. **What do you mean by repository?** A repository in version control is a central place where data, usually in the form of code, is stored and managed.
4. **What care is to be taken when merging two branches?** When merging two branches, ensure conflicts are resolved, the merge is tested before finalizing, and the branch history is understood to avoid unintended changes.
5. **What is the difference between Git and GitHub?** Git is a version control system tool used to track changes in source code during software development. GitHub is a hosting service for Git repositories, offering collaboration features.
6. **How do you push your project into a remote repository?** Use the command **git push [remote-name] [branch-name]**, where **[remote-name]** is typically **origin**, and **[branch-name]** is your branch's name.
7. **Is it possible to revert changes after a commit? If so, how?** Yes, you can revert changes using commands like **git revert** to create a new commit that undoes the changes or **git reset** to alter the commit history based on your requirements.
8. **What is Continuous Integration?** Continuous Integration (CI) is a development practice where developers integrate code into a shared repository frequently, preferably several times a day.
9. **What is CI/CD?** CI/CD stands for Continuous Integration and Continuous Deployment/Delivery. CI refers to the automation process for developers, while CD automates the delivery of applications to selected infrastructure environments.
10. **Which tools can be plugged with Jenkins?** Jenkins can integrate with numerous tools such as Git, Maven, Docker, Puppet, Ansible, and many others to support building, deploying, and automating any project.
11. **What is a pipeline?** In DevOps, a pipeline is a set of automated processes that allow developers and DevOps professionals to reliably and efficiently compile, build, and deploy their code to their production compute platforms.
12. **What is declarative pipeline?** A declarative pipeline in Jenkins is defined in a more simplified and more opinionated syntax with a specific structure to automate the pipeline.
13. **What is scripted pipeline?** Scripted pipeline is a traditional way of writing the Jenkins pipeline as code. It offers tremendous flexibility and a more granular level of control over the pipeline configuration.
14. **What is Tomcat server?** Apache Tomcat is an open-source Java servlet container that implements several Java EE specifications including Java Servlet, JavaServer Pages (JSP), and WebSocket, and provides a "pure Java" HTTP web server environment for Java code to run.
15. **What is a web server?** A web server is a system that delivers content or services to end-users over the internet, handling HTTP requests from clients.
16. **What are the other web servers available for deployment?** Other popular web servers include Apache HTTP Server, Nginx, IIS (Internet Information Services), and Lighttpd.
17. **What is a Container?** A container is a standard unit of software that packages up code and all its dependencies so the application runs quickly and reliably from one computing environment to another.
18. **Why learn Docker?** Learning Docker is beneficial because it ensures consistent environments from development to production, streamlines deployment, and scales efficiently.
19. **What are Docker images?** Docker images are the blueprints of containers, defining the code and dependencies that the container will run.
20. **What is a Dockerfile?** A Dockerfile is a text document that contains all the commands a user could call on the command line to assemble an image.
21. **What is Docker Hub?** Docker Hub is a service provided by Docker for finding and sharing container images with your team and the Docker community.
22. **How do you create a docker container from an image?** Use the command **docker run [options] image [command] [arg...]**.
23. **What is configuration management?** Configuration management is a systems engineering process for establishing and maintaining consistency of a product's performance, functional, and physical attributes with its requirements, design, and operational information throughout its life.
24. **Why do we need configuration management?** It helps in automating the provisioning, deployment, and remote tasks in software management processes, ensuring that the current design and build state of the system is known, good, and trusted; it avoids configuration drift.
25. **What are the other tools for configuration management?** Other popular tools include Ansible, Puppet, Chef, and SaltStack.