JENKINS INTERVIEW QUESTIONS:

1. What Is Jenkins Used For?

Jenkins is used for automating software development tasks such as code compilation, testing, code quality checks, artifact creation, and deployment. It streamlines the development process, ensuring reliability and quality by automating repetitive tasks in a DevOps context.

2. How To Trigger a Build In Jenkins Manually?

To manually trigger a build in Jenkins:

- Access the Jenkins Dashboard.
- Select the specific Jenkins job.
- Click “Build Now” to start the manual build.
- Provide build parameters if necessary.
- Confirm and monitor the build progress in real time.
- Review the build results on the job’s dashboard.
- Access build artifacts if applicable.
- Trigger additional builds as needed.

3. What Is The Default Path For The Jenkins Password When You Install It?

The default path for the Jenkins password when you install it can vary depending on your operating system and how you installed Jenkins. Here are the general default locations for the Jenkins password:

If you installed Jenkins manually, you might need to check the Jenkins home directory, which is often located at /var/lib/jenkins.

4. How To Integrate Git With Jenkins?

To integrate Git with Jenkins:

- Install the “Git Plugin” in Jenkins through the plugin manager.
- Configure Git in the global tool configuration, ensuring automatic installation is enabled.
- Create or configure a Jenkins job, selecting Git as the version control system.
- Specify the Git repository URL and, if necessary, credentials for authentication.
- Define the branches to monitor and build.
- Set up build triggers as needed.
- Save the job configuration and trigger builds manually or automatically based on your settings.
- Monitor build progress and results in the Jenkins dashboard.

5. What Does “Poll SCM” Mean In Jenkins?

In Jenkins, “poll SCM” means periodically checking a version control system (e.g., Git) for changes. You can schedule how often Jenkins checks for updates. When changes are detected, Jenkins triggers a build, making it a key feature for continuous integration, scheduled tasks, and automated response to code changes.

6. How To Schedule Jenkins Build Periodically (hourly, daily, weekly)? Explain the Jenkins schedule format?

To schedule Jenkins builds periodically at specific intervals, you can use the built-in scheduling feature. Jenkins uses a cron-like syntax for scheduling, allowing you to specify when and how often your builds should run. Here's a detailed explanation of the Jenkins schedule format and how to schedule builds:

Here's what each field means:

- Minute (0 – 59): Specifies the minute of the hour when the build should run (e.g., 0 for the top of the hour, 30 for the half-hour).
- Hour (0 – 23): Specifies the hour of the day when the build should run (e.g., 1 for 1 AM, 13 for 1 PM).
- Day of the month (1 – 31): Specifies the day of the month when the build should run (e.g., 1 for the 1st day of the month, 15 for the 15th day).
- Month (1 – 12): Specifies the month when the build should run (e.g., 1 for January, 12 for December).
- Day of the week (0 – 7): Specifies the day of the week when the build should run (e.g., 0 or 7 for Sunday, 1 for Monday, and so on).

7. What Is A Jenkins Agent?

A Jenkins agent, also called a Jenkins slave or node, is a separate machine or resource that collaborates with a Jenkins master to execute jobs and build tasks. Agents enable parallel and distributed builds, scaling Jenkins' capacity.

They register with the master, get assigned jobs, execute them on their own hardware or VMs, and report back results. Agents can run on various platforms, making it possible to test and build in different environments.

8. What Is The Default Port Number For Jenkins?

The default port number for Jenkins is 8080. When you access the Jenkins web interface via a web browser, you typically use the URL: http://your_jenkins_server:8080/.

9. Types of build triggers in Jenkins?

Types of build triggers in Jenkins include:

- SCM Polling Trigger: Monitors source code repositories for changes and triggers builds.
- Scheduled Build Trigger: Runs jobs on a predefined schedule using cron-like syntax.
- Webhook Trigger: Listens for external events or notifications to start builds.
- Upstream/Downstream Trigger: Triggers downstream jobs based on the success of upstream jobs, creating build pipelines.
- Manual Build Trigger: Requires manual user intervention to start a job.
- Dependency Build Trigger: Triggers jobs when another job is completed, regardless of success or failure.
- Parameterized Trigger: Passes parameters from one job to another during triggering.
- Pipeline Trigger: Allows custom triggering logic within Jenkins Pipelines.

10. What is the language used to write the Jenkins CI/CD pipeline?

Jenkins CI/CD pipelines are typically written using a domain-specific language called Groovy. Specifically, Jenkins uses the Jenkins Pipeline DSL (Domain-Specific Language), which is an extension of Groovy tailored for defining and orchestrating continuous integration and continuous delivery pipelines.

11. What is Declarative and Scripted Syntax?

Jenkins Pipelines support two syntax flavours. Declarative and Scripted. Declarative syntax provides a simplified and structured way to define pipelines, while Scripted syntax allows for more fine-grained control and scripting capabilities

12. What is pipeline as a code?

Jenkins Pipelines support two syntax flavours. Declarative and Scripted. Declarative syntax provides a simplified and structured way to define pipelines, while Scripted syntax allows for more fine-grained control and scripting capabilities

13. What is the difference between Continuous Delivery and Continuous Deployment?

Continuous Delivery (CD) and Continuous Deployment (CD) are two distinct practices in the DevOps and software development lifecycle, but they are closely related. Here are the key differences between the two:

the main difference between Continuous Delivery and Continuous Deployment is the level of automation and human intervention in the final deployment to the production environment. Continuous Delivery stops short of fully automated production deployment and includes a manual approval step, while Continuous Deployment automates the entire process, releasing code changes to production as soon as they pass automated tests. The choice between the two practices depends on an organization's risk tolerance, release strategy, and the need for manual validation.

14. Explain about Master-Slave Configuration in Jenkins?

A Master-Slave configuration in Jenkins, also known as a Jenkins Master-Agent configuration, is a setup that allows Jenkins to distribute and manage its workload across multiple machines or nodes. In this configuration, there is a central Jenkins Master server, and multiple Jenkins Agent nodes (slaves) that are responsible for executing build jobs. This architecture offers several advantages, including scalability, parallelism, and the ability to run jobs in diverse environments.

Components:

- **Jenkins Master:**

a. The Jenkins Master is the central server responsible for managing and coordinating the entire Jenkins environment.
b. It hosts the Jenkins web interface and handles the scheduling of build jobs, job configuration, and the storage of build logs and job history.
c. The Master communicates with Jenkins Agents to delegate job execution and collects the results.

- **Jenkins Agent (Slave):**

a. Jenkins Agents, often referred to as Jenkins Slaves or nodes, are remote machines or virtual instances that perform the actual build and testing tasks.
b. Agents can run on various operating systems and environments, enabling the execution of jobs in different configurations.
c. Agents are registered with the Jenkins Master and are available to accept job assignments.

15. What is a Freestyle project in Jenkins?

A Freestyle project in Jenkins is a basic and user-friendly job type. It allows users to configure build jobs using a graphical interface without scripting. It's suitable for simple build and automation tasks, supporting various build steps, post-build actions, and integration with plugins. While it's easy to use, it may not be ideal for complex workflows, unlike Jenkins Pipeline jobs, which offer more flexibility and scripting capabilities.

16. Explain about the multibranch pipeline in Jenkins?

A Multibranch Pipeline in Jenkins is a feature for managing CI/CD pipelines for multiple branches in a version control repository. It automatically creates pipelines for each branch or pull request, uses Jenkinsfiles to define pipeline configurations, supports parallel builds, and cleans up unused jobs. It simplifies managing and automating pipelines across various code branches and pull requests, streamlining the CI/CD process.

17. What is a Pipeline in Jenkins?

A Jenkins Pipeline is a series of code-defined steps that automate the Continuous Integration and Continuous Delivery (CI/CD) process. It allows you to define and manage your entire software delivery pipeline as code, using a declarative or scripted syntax. Pipelines cover continuous integration, delivery, and deployment, with support for parallel and sequential stages.

18. What is the global tool configuration in Jenkins?

Global Tool Configuration in Jenkins refers to the centralized configuration of software tools and installations that can be used by all Jenkins jobs and pipelines across the Jenkins master server. It allows Jenkins administrators to set up and manage tool installations such as JDKs, build tools (e.g., Maven, Gradle), version control systems (e.g., Git, Subversion), and other utilities in a consistent and organized manner.

19. Write a sample Jenkins pipeline example?

Here's a simple Jenkins pipeline example written in Declarative Pipeline syntax.

```
pipeline {
    agent any


    stages {
        stage('Build') {
            steps {
                echo 'Building...'
                // Add your build commands here
            }
        }
        stage('Test') {
```

```
        steps {
            echo 'Testing...'
            // Add your test commands here
        }
    }
    stage('Deploy') {
        steps {
            echo 'Deploying...'
            // Add your deployment commands here
        }
    }
  }

  post {
    success {
        echo 'Pipeline succeeded!'
    }
    failure {
        echo 'Pipeline failed!'
    }
  }
}
```

20. Explain the node step in Jenkins pipelines and its significance?

The “node” step in Jenkins pipelines is significant for two main reasons:

Parallelization: It allows tasks in the pipeline to run concurrently on different agents, significantly speeding up the pipeline execution. This is crucial for identifying issues quickly and delivering software efficiently.

Flexibility in Agent Selection: It provides the flexibility to choose different agent types, such as Docker containers, cloud-based agents, on-premises agents, or Kubernetes pods.

21. What is the difference between Poll SCM and Webhook?

Poll SCM:

Pull Mechanism: The CI/CD server pulls information from the VCS to determine if there are new changes. If it detects new changes, it triggers the build or pipeline execution.

Webhook:

Push Mechanism: In a webhook setup, the VCS system sends an HTTP POST request to a predefined URL (the webhook endpoint) whenever there is a code change or commit. This means that the VCS actively notifies the CI/CD system about changes.

22. What is CI and CD in Jenkins interview questions?

CI/CD (Continuous Integration/Continuous Deployment) pipeline is an automated process in software development. It involves integrating code changes frequently, running automated tests, and deploying code to production quickly and consistently.

23. Which are Jenkins job types?

Jenkins provides different types of jobs (or projects). Some of them are:

- Pipeline.
- Multibranch Pipeline.
- Freestyle.
- Multi-configuration (matrix)
- External job

24. What is Jenkinsfile?

Jenkins file is a text file that has a definition of a Jenkins pipeline and is checked into the source control repository. It enables code review and iteration on the pipeline. It also permits an audit trail for the pipeline.

25. Why is Jenkins used with Selenium?

sing Selenium allows Jenkins's testing whenever there are any software changes or any changes in the environment. When the Selenium test suite is integrated with Jenkins, the testing part is also automated as part of the build process.

26. Name the two components that Jenkins is mostly integrated with?

Jenkins is typically integrated with these two components:

- Version Control systems like Git and SVN (Apache Subversion)
- Build tools like Maven

27. What are the features of Jenkins?

- Free open source.
- Easy installation on various operating systems.
- Build Pipeline Support.
- Workflow Plugin.
- Test harness built around JUnit.
- Easy upgrades.
- Rapid release cycle.
- Easy configuration setup.
- Extensible with the use of third-party plugins.

28. Mention some of the important plugins in Jenkins?

- Gits
- Maven 2 Project
- HTML Publisher
- Join
- Green Balls
- Amazon EC2

29. What is Groovy in Jenkins?

Groovy is the default scripting language that is being used in the development of JMeter Version 3.1.

Currently Apache Groovy is the dynamic object-oriented programming language that is used as a scripting language for the Java platform

30. What is a build in Jenkins?

A build in Jenkins is a process where source code (from a version control system) is fetched, compiled, and packaged into a deployable format like JAR, WAR, or Docker Image.