JENKINS INTERVIEW QUESTIONS

1) What is Jenkins and why should we use it?

Jenkins is a leading open-source, free automation tool that is used to develop and test software projects.

Reasons why Jenkins is widely used:

- -Used to detect faults in software development and systematize the testing of buildss
- -Used to constantly monitor the code simultaneously and add changes to the build.
- -Jenkins consists of an installer package for most operating systems.
- -Used to keep the team updated and synchronized about the changes incorporated.
- -Used to build CI/CD pipelines since it has plugin capabilities and is easy to use.

2) What are the features of Jenkins?

Jenkins features include:

- -It is an open-source automation tool and it is free.
- -Gives pipeline support
- -installation is easy on systems with multiple operations.
- -large number of plugins
- -Jenkins upgrades effortlessly.
- -speedy release cycle
- -Configuration setup is easy.

3) What are the prerequisites to use Jenkins?

The requirements to use Jenkins are:

- -First requirement is an accessible source code repository, for instance, a Git repository.
- -A build script in working condition, example- Maven script checked into the repository.

4) How do we manually restart Jenkins?

The following commands are used to restart Jenkins manually:

- -(jenkins_url)/safe restart — Wait until all the builds are completed before restarting.
- -(jenkins_url)/restart-To force restart without waiting for build completion.

5) What does "continuous integration" mean?

Continuous Integration is the continuous process of checking the code made by developers into a version control system numerous times. The build is automated in the process to inspect and detect bugs in the developed code. Continuous integration comprises of:

6) what is Jenkins Pipeline?

Jenkins Pipeline is collection of features of Jenkins. They are installed as plugins that allows delivery of pipeline implementation continuously.

7) What are the advantages of using Jenkins?

The advantages of using Jenkins are:

- -User-friendly, free, and it is an open source
- -Trouble-free Installation
- -Code deployment is convenient and takes very less time. It simultaneously generates reports.
- -Helps in collaboration between the operation and development teams.
- -Free of cost
- -Detection of code errors at an early stage
- -Reduced integration issues due to automation of integration work
- -Rich plugin ecosystem
- -Platform independence

8) What are the components that you can integrate Jenkins with?

Jenkins is mainly integrated with the following components:

- -First is the version control system, for example: GIT, SVN
- -Another one is build tools. An example is Apache Maven.

9) How does Hudson relate to Jenkins?

Jenkins was previously known as Hudson. It is a continuous integration tool and is open source, written in Java.

10) Explain how to set up Build jobs in Jenkins.

The following steps will help you to build jobs in Jenkins:

Step 1 -First, open the Jenkins dashboard and click on the New Item.

Step 2 -Enter the item name and choose the “Freestyle project option”.

Step 3 -Specify the details of the job.

Step 4 -Next, specify the location of files that need to be built.

Step 5 -In case the repository is hosted on GitHub, enter the URL of that repository here.

Step 6 -Build section and click on the Add build step.

Step 7 -In the command window, enter the following commands and then click on the Save button.

(Javac HelloWorld.java

Java HelloWorld)

Step 8 -Now click on the “Build” option to see whether the job is defined well or not.

Step 9 -Once the build is scheduled, it will run.

Step 10 -To see details of the build, click on the Console Output link.

11) What is Artifact Archival & how to do it in Pipelines?

Artifacts are the exportable/storable/archivable results of a specific job build. This can be configured using a plugins. Based on the configured pattern, the files/directories matching the configured patterns will be archived for a Jenkins build.

Nexus, JFrog and S3 these are frequently used artifactory services that we can integrate in Jenkins

12) How can a job configuration be reset to an earlier version/state?

From the Job details page, we can use Job Config History to — See diff, Review & Revert the Job configs from the history of changes we have made to a particular job. This will be super useful when a job is misconfigured by someone by mistake, it can be reviewed and reverted easily to any of its earlier states.

13) How to do Global Tools Configuration in Jenkins?

Global Tools are tools that need to be installed outside the Jenkins environment and need to be controlled from within the Jenkins environment

Steps to using a Global Tool generally include -

- Install the tool Plugin into the Jenkins instance, to include the global tool into a list of global tools used by Jenkins.
- Install the tool in the Jenkins instance or provide away (maybe a command to download and) install the tool during runtime.
- Go to Manage Jenkins -> Global Tools Configuration and Scroll through the tool list and configure the global tool-specific configurations.
- Make use of the installed global Tool in your job/pipeline.

14) What are the ways to trigger a Jenkins Job/Pipeline?

There are many ways we can trigger a job in Jenkins.

- Trigger it manually from the Jenkins web application.
- Trigger it using Jenkins CLI from the master/slave nodes.
- Time-based Scheduled Triggers like a cron job.
- Event-based Triggers like SCM Actions (Git Commit, Pull Requests), WebHooks, etc.
- Upstream/Downstream triggers by other Jenkins jobs.

15) How Jenkins knows when to execute a Scheduled job/pipeline and how it is triggered?

Jenkins master will have the cron entries set up for the jobs as per the scheduled Job's configurations. As and when the time for a particular job comes, it commands agents (based on the configuration of the job) to execute the job with required configurations.

16) What are the credential types supported by Jenkins?

Secret text — A token such as an API token, JSON token, etc.

Username and password — Basic Authentication can be stored as a credential as well.

Secret file — A secret file used to authenticate some secure data services & security handshakes.

SSH Username with a private key — An SSH public/private key pair for Machine to Machine authentication.

Certificate — a PKCS#12 certificate file and an optional password.

Docker Host Certificate Authentication credentials.

17) What are the credential types supported by Jenkins?

Global and System

Global: the credential will be usable across all the jobs configured in the Jenkins instance (i.e. for all jobs). This is more suited for user Jobs (i.e. for the freestyle, pipeline, or other jobs) to authenticate itself with target services/infrastructures to accomplish the purpose of the job)

System: This is a special scope that will allow the Jenkins itself (i.e. the core Jenkins functionalities & some installed plugins) to authenticate itself to external services/infrastructures to perform some defined tasks. E.g. sending emails, etc.

18) What is the Blue Ocean?

Blue Ocean is the redefined user experience for Jenkins. Designed from the ground up for Jenkins Pipeline, it is still compatible with freestyle jobs, Blue Ocean reduces clutter and increases clarity

19) What is meant by CI-CD?

CI-CD refers to the practice of combining continuous integration and continuous deployment, which is a key component of DevOps practices.

20) What is Jenkins Pipeline?

Jenkins Pipeline is a suite of plugins that enables implementation of continuous delivery pipelines within Jenkins. A Jenkins pipeline defines a set of steps and stages that are executed automatically to build, test, and deploy software applications.

21) How do you configure the job in Jenkins?

To configure a job in Jenkins, you need to navigate to the job's configuration page and specify the source code repository, build triggers, build steps, and post-build actions. You can also configure build parameters, environment variables, and security settings.

22) Where do you find errors in Jenkins?

You can find errors in Jenkins by checking the build logs and console output. The build logs provide detailed information about the build steps and any errors that occurred during the build process. The console output displays the build progress and any error messages generated during the build.

23) In Jenkins how can you find log files?

In Jenkins, you can find log files by navigating to the job's build history and selecting the build for which you want to view the log files. From there, you can access the console output and build logs.

24) Jenkins workflow and write a script for this workflow?

```
pipeline {
  agent any


  stages {
    stage('Checkout code') {
      steps {
        checkout([$class: 'GitSCM',
             url: 'https://github.com/my-repo.git'])
      }
    }
    stage('Build code') {
      steps {
        sh 'YOUR BUILD COMMAND HERE'
      }
    }
    stage('Test code') {
      steps {
        sh 'YOUR TEST COMMAND HERE'
      }
    }
    stage('Deploy code') {
      steps {
        sh 'YOUR DEPLOY COMMAND HERE'
      }
    }
  }
}
```

25) How to create continuous deployment in Jenkins?

To create continuous deployment in Jenkins, you need to create a Jenkins pipeline that includes the necessary steps to build, test, and deploy your software application. You can use Jenkins plugins to automate various stages of the deployment process, such as building and deploying the application.

26) How build job in Jenkins?

To build a job in Jenkins, you need to navigate to the job’s configuration page and specify the source code repository, build triggers, build steps, and post-build actions. Then, you can manually start a build by clicking the “Build Now” button, or you can configure the job to build automatically when code changes are pushed to the source control repository.

27) Why we use pipeline in Jenkins?

We use pipelines in Jenkins because they provide a way to automate the entire build-test-deploy process. Pipelines help to ensure that software applications are consistently built, tested, and deployed in a consistent and repeatable manner.

28) Is Only Jenkins enough for automation?

Jenkins is a popular tool for automation, but it may not be enough for all automation needs, depending on the complexity of your software applications and your organization’s specific requirements. Other tools, such as Ansible and Puppet, may be needed to automate additional aspects of the software development lifecycle.

29) Explain diff stages in CI-CD setup

The stages in a typical CI-CD setup could include:

Code commit

Build

Test

Deploy to staging

Test in staging

Deploy to production

30) Name some of the plugins in Jenkin?

Some popular plugins in Jenkins include:

Pipeline: a suite of plugins that helps to implement and integrate continuous delivery pipelines

Git Plugin: allows Jenkins to interact with Git repositories

Maven Plugin: enables the building of Java projects with Apache Maven

JUnit Plugin: allows Jenkins to display test results for JUnit tests

Slack Plugin: integrates Jenkins with Slack, allowing for notifications and status updates to be sent to a Slack channel

Email-ext Plugin: allows Jenkins to send customizable build notifications via email

Artifactory Plugin: integrates Jenkins with JFrog Artifactory to manage binary artifacts

Docker Plugin: allows Jenkins to build, test, and deploy Docker containers.

SSHAgent: Used to make a connection between the servers

Deploy to container: used to deploy the application in tomcat web servers

Role-based authorisation strategy: used to provide roles to users