

JENKINS QUESTIONS

1. What is Jenkins?

Jenkins can be thought of as a free and open-source automation tool that is used for continuous integration. It automates some of the software development processes, such as building, testing, and deployment. It can also be integrated with a large number of testing and deployment technologies

2. What is Jenkins used for?

Jenkins is used in the automation of some of the software development processes. With Jenkins, one can continuously test software projects so that developers can integrate the changes to them. Jenkins facilitates continuous integration and delivery through its built-in plugins.

3. How to add a user in Jenkins?

- Go to Manage Jenkins
- Click on Create User
- Enter all the details
- Select Create User

4. What is Jenkins Pipeline?

Jenkins Pipeline, sometimes simply called Pipeline, is a set of plugins, which supports the integration and use of continuous delivery pipelines in Jenkins. The continuous delivery pipeline is the process where automated builds, tests, and deployments are planned and arranged as one release workflow. Simply put, it is the process the code changes will go through for delivering software from the version control to clients and users.

5. What is Jenkinsfile?

A Jenkinsfile is essentially a text file containing the steps for running a Jenkins Pipeline and is registered into the source control repository of a project.

6. What is continuous integration in Jenkins?

Continuous integration is the process in which all development work integration is done at the earliest and a code is continuously tested after a commit. This process allows one to discover bugs at an early stage and fix them. The Jenkins build server provides this functionality.

7. What are the software prerequisites that must be met before Jenkins is installed?

First, let us compare Jenkins with TeamCity:

Comparison	Jenkins	TeamCity
<i>Open Source</i>	Yes	No
<i>Default security</i>	No	Yes
<i>Individual validation</i>	No	Yes

Popularity	Widely used	Not so widely used
-------------------	-------------	--------------------

The software prerequisites for installing Jenkins are that first we need to install Java Development Kit. We also need to install the Jakarta Enterprise Edition. Jenkins also comes with an embedded Jetty Runtime that can be used if WebSphere or Tomcat is not available.

8. How to take a backup of your Jenkins build jobs?

Within the XML configuration, each Jenkins build is stored. When this folder is copied, the configuration of all the build jobs that are managed by the Jenkins master is backed up. If we can perform a Jenkins Git integration, then it is good. When we copy the contents of the folder, we will see that the build jobs described in the folder will be restored when the Jenkins server is started the next time.

9. State some of the advantages of using Jenkins.

Here are some of the most important advantages of Jenkins:

- We will get an automated build report every time a change is made to the source code.
- We will be able to achieve continuous integration with agile methodology principles.
- We can automate the maven release project with a few simple steps.
- Bugs can be easily tracked at an early development stage.

10. What are the requirements for using Jenkins?

Some of the requirements for using Jenkins are as follows:

- A source code repository like a Git repository
- A build script like a Maven script that is checked into the repository

11. What are some of the useful plugins in Jenkins?

Some of the important plugins that are used with Jenkins are:

- Git repository
- Amazon EC2
- HTML Publisher
- JDK Parameter Plugin
- Configuration Slicing Plugin

12. How to integrate Git with Jenkins?

Before we begin, we need to first have the GitHub Jenkins plugin installed.

1. Go to Manage Jenkins on the Jenkins dashboard, and then select Configure System
2. Choose Add GitHub Server in the GitHub section
3. Add the GitHub token as credentials, and Save
4. Open the Jenkins project
5. Tick off the GitHub project checkbox, and give the GitHub Repository path for the Project URL
6. Under Source Code Management, check the Git box, and in Repositories, give the GitHub Repository URL
7. Under Build Triggers, check the 'Build when a change is pushed to GitHub' box
8. Set a webhook to Jenkins after the plugin installation is done
9. Go to Settings and then to Integrations & Services from the GitHub repository

10. Select Add Service, and then add Jenkins (GitHub plugin)
11. The URL of the Jenkins machine should be given as Jenkins Hook URL. Add /github-webhook/

13. Name some of the SCM tools that are supported by Jenkins.

Some of the important SCM tools that are supported by Jenkins include:

- Git
- Subversion
- CVS
- Mercurial

14. What is the relation between Hudson and Jenkins?

Initially, Jenkins was called Hudson. However, due to some reasons, the name was changed from Hudson to Jenkins.

15. If there is a broken build in your Jenkins project, then what will you do?

First, we need to open the console output where the broken build is created and then see if there are any file changes that were missed. If we do not find any issues in this manner, then we can update our local workspace and replicate the problem and then try to solve it.

16. How to schedule builds in Jenkins?

Some steps for scheduling builds in Jenkins are as follows:

- First, we should have a source code management commit.
- We have to complete the other builds.
- Then, we have to schedule it to run at a specified time.
- We need to then give a manual build request.

17. How to configure Jenkins with Maven?

To configure Jenkins with Maven:

1. On the Jenkins dashboard, go to Manage Jenkins, and select Configure System
2. Scroll down until the Maven section is seen, and then go to Add Maven
3. Uncheck the 'Install automatically' box
4. Add any name for the setting and the location of the MAVEN_HOME
5. After saving, one can create a job with the 'Maven project' option

18. How to create a slave node in Jenkins?

To create a slave node in Jenkins:

1. Go to Manage Jenkins, and scroll down to Manage Nodes
2. Click on New Node
3. Set the node name, choose the Dumb slave option, and then click on OK
4. Enter the node slave machine details, and click on Save

19. How do you install Jenkins?

To install Jenkins, you need to download the WAR file from the Jenkins website and run it with `java -jar jenkins.war` on Linux/Mac machines. On Windows, you can download the MSI installer file which installs Jenkins as a Windows service

20. Which commands can be used to begin Jenkins?

The main commands to start Jenkins are `java -jar jenkins.war`, which runs it in the foreground or `service start jenkins` if installed as a service on Linux/Mac machines. On Windows, it can be started from the Services console.

21. What is a CI/CD pipeline?

A CI/CD pipeline refers to the automated process of continuously building, testing, and deploying code changes to ensure code quality, and faster delivery of features to end users and to reduce the risk of failed deployments.

INTERMEDIATE QUESTIONS:

22. Which architecture is recommended for a scalable Jenkins environment?

Distributed builds architecture is prescribed for a scalable Jenkins environment because the agents perform the main work and the master maintains the coordination. In this way, the focus is on both the master and the agent. In addition, the master provides the GUI and API endpoints.

23. How to change port for Jenkins?

Jenkins, by default, runs on port number 8080. For changing the existing port from 8080 to the desired port number, we can take the following steps:

1. Run Jenkins using the command line
2. Execute the command, `java -jar -httpPort=desired_port jenkins.war`

If Jenkins is installed using the Windows package, then:

1. Go to Program Files/Jenkins directory
2. Open Jenkins.xml in the editor

Find `-httpPort=8080` and replace the port 8080 with the new port number

24. How to configure and use third-party tools in Jenkins?

These are the steps used for working with a third-party tool in Jenkins:

- First, we have to install the third-party software.
- We need to have the plugin that supports the third-party tool.
- Then, we must configure the third-party tool in the admin console.
- We can then use the plug-in from the Jenkins build job.

25. How to set up a Jenkins job?

First, we need to create a Jenkins job by going to the Jenkins top page and then selecting a 'New Job' and building a freestyle software project. Some of the elements of a freestyle project include the following:

- We need a CVS or a subversion where our source code will reside.
- When Jenkins performs the builds, we will need the optional triggers.
- We need a build script like a Maven or Ant where the script is actually built.

26. What are the steps included in a Jenkins pipeline?

A complete Jenkins pipeline includes building a project from the source code, putting it through a variety of units, integrating, testing for user acceptance and performance, and then finally deploying the packaged application on an application server.

So, the steps in a Jenkins pipeline can be listed as below:

- Build
- Test
- Deploy

27. What is the process for creating a backup and copy files in Jenkins?

If we want to create a backup for our file, then we need to regularly backup our Jenkins_Home directory. This will include all the build jobs configuration, all the slave node configuration, and the build history. If we want to create a Jenkins backup, we can copy a job directory to a clone or can rename the directory.

28. How will you create Pipeline in Jenkins?

The Jenkins Pipeline code is written into a text file called Jenkinsfile, which in turn is checked into a project's source control repository.

1. Click on the New Item option on the Jenkins dashboard
2. Create a name for the new item
3. Choose the 'Pipeline' project, and click on OK
4. In the Pipeline tab, put the Jenkinsfile code
5. Click on Apply and Save
6. Select Build Now, which will start building the Pipeline

After the Pipeline is set up, any new branches or pull requests that are created in the repository will be automatically detected by Jenkins, and it will start running Pipelines for them.

29. How to configure Docker in Jenkins?

To configure Docker in Jenkins, we have to follow the below steps:

1. First, we need to click on Manage Jenkins on the Jenkins dashboard.
2. We will then select Manage Plugins on the Configuration page.
3. Next, we need to click on Available once in the tabbed interface, and this will show all the Jenkins plugins available for installation.
4. Then, we will search for a Docker plugin in the search box. There are multiple Docker plugins to choose from.
5. Finally, we will select the Docker plugin.

30. When can you use the GitHub plugin in Jenkins?

Whenever one wants to integrate Jenkins with GitHub projects, the GitHub plugin can be used. It is used to enable the scheduling of a build, pulling data and code files from the GitHub repository to the Jenkins machine, and triggering every build automatically on the Jenkins server after each commit on the Git repository. This saves time and allows one to incorporate the specific project into the CI process.

31. Name two ways a Jenkins node agent can be configured to communicate back with the Jenkins master.

These are the mechanisms for starting a Jenkins node agent:

- From the browser window, launch a Jenkins node agent
- From the command line, launch a Jenkins node agent

When we launch a Jenkins node agent, it will download a JNLP file. A new process is launched on the client machine by JNLP when it runs.

32. How to turn off Jenkins Security if the administrative users have locked out of the admin console?

There is a folder that contains a file named config.xml. We need to change the settings to false for the security to be disabled when Jenkins is started the next time.

33. From one server to another, how do you copy or move your Jenkins jobs?

First, we need to copy our jobs directory from the old to the new server. There are multiple ways to do it. We can either move the job from the installation by simply copying the corresponding job directory or we can make a clone of the job directory by making an existing job's copy. For this, we need to have a different name, which we can rename later.

34. How to deploy from Jenkins?

1. On the Jenkins dashboard, go to Manage Jenkins and then to Manage Plugins. In the Available section, tick off the 'Deploy to Container Plugin' box and install it. Then, restart the Jenkins server
2. The next step is to go to the Build project and click on Configure. Choose the 'Deploy war/ear to a container' option
3. In the Deploy war/ear to a container section, enter the server details on which the files need to be deployed, and press Save

After a successful build, the necessary files will get deployed to the required container.

35. What are the features of Jenkins?

Jenkins is an open-source automation server with features like continuous integration, continuous delivery, artifact management, code quality monitoring, and reporting. It supports version control tools like Git, SV, etc., and can be integrated with bug/issue tracking tools. It provides flexibility through plugins.

36. Explain the process in which Jenkins works.

Jenkins polls the configured source code management (SCM) tool like Git for new code changes. When a new change is detected, Jenkins pulls the latest code, compiles it, and runs automated tests. If tests pass, the code is deployed to staging or production environments. Plugins can be used to customize the process for specific needs.

ADVANCED JENKINS QUESTIONS FOR EXPERIENCED

37. What is the process for securing Jenkins?

First, we need to ensure global security. Then, we have to make sure that Jenkins is integrated with the user directory through an appropriate plugin. The project matrix is enabled for fine tuning the access using the custom version-controlled script for automating the process of rights and privileges in Jenkins. The access to Jenkins data or folder is limited. We will run security audits on it.

38. What is Groovy in Jenkins?

Groovy is a scripting language for Jenkins that allows dynamically creating/modifying jobs and workflows in Jenkins pipelines. It provides a domain-specific language (DSL) to define Jenkins pipelines as code. This makes pipelines more readable, maintainable, and reusable.

39. Explain a backup plugin and its uses.

Backup plugins like Jenkins Backup take backups of Jenkins configurations, jobs, plugins, and other data to external storage like Amazon S3. This helps restore Jenkins after failures like data loss or server crashes. The backups can be scheduled periodically or triggered manually. It ensures business continuity in case of disasters.

40. What is a Ping Thread in Jenkins and how it works?

The Ping Thread is used by the Jenkins master to monitor the health and availability of Jenkins nodes.

The Ping Thread runs periodically on the Jenkins master and pings each registered slave node to check connectivity. If a slave does not respond within the configured timeout, it is marked offline. This prevents jobs from being assigned to unavailable nodes. It ensures that only live nodes are used for job executions, quickly detecting node failures without relying on job failures.

JENKINS INTERVIEW QUESTIONS FOR DEVOPS

41. How do you implement a Blue-Green Deployment using Jenkins?

Blue-Green Deployment, in Jenkins, can be set up by utilizing the Jenkins pipeline and deployment plugins. This approach involves creating two environments, with one environment being active at a given time. Once testing is completed in the Green environment, the traffic is smoothly transitioned from the environment to the Green one, minimizing downtime and potential risks.

42. Explain how Jenkins integrates with container orchestration tools like Kubernetes.

Jenkins works seamlessly with Kubernetes using the Kubernetes plugin, which allows for the setup of Jenkins agents within Kubernetes pods. This integration streamlines the process of deploying, scaling, and managing containerized applications. With this setup, Jenkins pipelines can efficiently build, test, and deploy directly to Kubernetes clusters.

43. How do you secure sensitive data (like passwords and API keys) in Jenkins?

To ensure the security of information, in Jenkins, the Credentials Plugin is utilized. This plugin securely stores passwords, tokens, and keys. For an added layer of security, it is advisable to integrate with secret managers such as HashiCorp Vault. This integration allows for secret management and access control within Jenkins jobs and pipelines.

44. Discuss the use of Jenkins Shared Libraries for pipeline reuse.

Shared libraries in Jenkins allow for the sharing of scripts between Jenkins files. This promotes the reuse of code and helps reduce duplication. By storing used pipeline code in version control repositories, teams can establish pipeline processes and enhance maintainability across various projects.

JENKINS INTERVIEW QUESTIONS FOR TESTERS

45. How can Jenkins be used to automate testing in a CI/CD pipeline?

Jenkins streamlines the testing process by integrating with a range of testing tools and frameworks. As part of a CI/CD pipeline, Jenkins initiates automated tests following every commit, utilizing tools such as JUnit or Selenium. The outcomes of these tests are thorough. Jenkins can be customized to alert the team in case of any failures. This ensures feedback. Allows for quick problem resolution.

46. Explain how Jenkins supports Test-Driven Development (TDD).

Jenkins simplifies the testing procedure by integrating with testing tools and frameworks. As a part of the CI/CD pipeline, Jenkins automatically triggers tests after each commit, making use of tools like JUnit or Selenium. The results of these tests are thoroughly examined. Additionally, Jenkins can be personalized to notify the team in case of any failures, ensuring feedback and facilitating problem resolution.

47. How does Jenkins assist in performance testing?

Jenkins assists in conducting performance tests by integrating with tools like JMeter or LoadRunner. It can schedule and execute performance tests, collect results, and generate reports. This enables the inclusion of performance testing within the CI/CD pipeline, guaranteeing consistent application performance and scalability.