1. What is Jenkins?

Jenkins is an open-source free automation tool used to build and test software projects. The tool makes it painless for developers to integrate changes to the project. Jenkins' primary focus is to keep track of the version control system and initiate and monitor a build system if there are any changes. It keeps an eye on the entire process and provides reports and notifications to alert.

Some typical reasons as to why Jenkins is so widely used are:

* Developers and testers use Jenkins to detect defects in the software development lifecycle and automate the testing of builds.
* They use it to continuously monitor the code in real-time and integrate changes into the build.
* Jenkins as it turns out, is a great fit for building a [CI/CD pipeline](https://www.simplilearn.com/tutorials/jenkins-tutorial/ci-cd-pipeline) because of its plugin-capabilities, and simple-to-use nature.

2. What are the features of Jenkins?

Some of the crucial features of Jenkins are the following:

* It is a free and open-source automation tool
* Jenkins provides a vast number of plugins
* It is easy to set up and install on multiple operating systems
* Provides pipeline support
* Fast release cycles
* Easy upgrades

3. What is Groovy in Jenkins?

* Apache Groovy is a dynamic object-oriented programming language used as a scripting language for [Java](https://www.simplilearn.com/tutorials/java-tutorial/what-is-java) platforms.
* Groovy is used to orchestrate the Jenkins pipeline and enables different teams to contribute to the work in different languages.
* Groovy's syntax is very similar to that of Java, making it more seamless with the Java interface.
* The language has several features like Java compatibility and Development support.

Also Read: [All You Need to Know for Choosing the First Programming Language to Learn](https://www.simplilearn.com/tutorials/programming-tutorial/first-programming-language)

4. How do you install Jenkins?

Follow the steps mentioned below to [install Jenkins](https://www.simplilearn.com/tutorials/jenkins-tutorial/jenkins-installation-on-windows):

* Install Java
* Install Apache Tomcat Server
* Download Jenkins war File
* Deploy Jenkins war File

5. Which commands can be used to begin Jenkins?

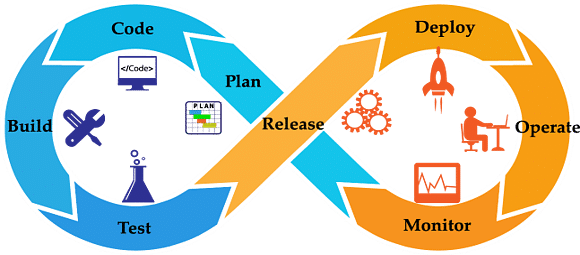
Here are the commands used to start Jenkins:

* Open the command prompt
* After the command prompt opens, browse to the directory where Jenkins war is present
* Then run the following command:

D:\>Java -jar Jenkins.war

6. What is "Continuous Integration" with reference to Jenkins?

* [Continuous Integration](https://www.simplilearn.com/tutorials/devops-tutorial/continuous-integration) is a development practice where the codes can be integrated into a shared repository.
* The practice uses automated verifications for the early detection of code problems.
* Continuous Integration triggers the build to find and identify bugs present in the code.
* It adds consistency to the build process.
* It’s a means to build things faster and prevents broken code.



7. What are the differences between Continuous Integration, Continuous Delivery, and Continuous Deployment?

|  |  |  |
| --- | --- | --- |
| Continuous Integration | Continuous Delivery | Continuous Deployment |
| Continuous Integration (CI) is a DevOps software development practice that permits developers to combine/merge the changes to their code in the central repository to run automated builds and tests. | Continuous Delivery (CD) refers to the building, testing, and delivering improvements to the software code. The most critical part of the CD is that the code is always in a deployable state. | Continuous Deployment (CD) is the ultimate stage in the DevOps pipeline. It  refers to automatic release of any developer changes from the repository to the production stage. |

8. What is a CI/CD pipeline?

CI/CD Pipeline or Continuous Integration/ Continuous Delivery is considered the DevOps approach's backbone. The pipeline is responsible for building codes, running tests, and deploying new software versions.

9. What is a Jenkins pipeline?

* The pipeline represents the continuous delivery and continuous integration of all the jobs in the SDLC and DevOps life cycle.
* The Jenkins pipeline is a set of plugins that support implementation and integration of continuous delivery pipelines into Jenkins. It connects this pipeline in a particular format by Jenkins.
* The Jenkins pipeline solves several problems like the maintenance of thousands of jobs and maintaining deployment with needing to resort to other powerful methods.

10. Name the three different types of pipelines in Jenkins?

The three different types of Jenkins pipelines are:

* CI/CD pipeline
* Scripted pipeline
* Declarative pipeline

11. How can you set up a Jenkins job?

To set up a Jenkins job, you may follow these steps:

* Select New item from the menu
* Next, enter a name for the job and select a free-style job
* Click on OK to create a new job
* Hence, the next page that appears will allow you to configure your job.

12. What are the requirements for using Jenkins?

To use Jenkins, you require the following:

* A source code repository that can be accessed, for example, a Git repository.
* A build script, for example, a Maven script.

13. Name the two components that Jenkins is mostly integrated with.

Jenkins is typically integrated with these two components:

1. Version Control systems like Git and SVN (Apache Subversion)
2. Build tools like [Maven](https://www.simplilearn.com/tutorials/maven-tutorial/what-is-maven)

14. Name some of the useful plugins in Jenkins.

Some of the plugins in Jenkins include:

* [Maven 2 project](https://www.simplilearn.com/tutorials/maven-tutorial/maven-project-in-eclipse)
* [Amazon EC2](https://www.simplilearn.com/tutorials/aws-tutorial/aws-ec2)
* Copy artifact
* Join
* HTML publisher
* Green Balls

15. How can you create a backup and copy files in Jenkins?

* Jenkins stores all the settings, builds scripts, and logs in the home directory.
* Then, if you want to create a backup of this Jenkins set up all you have to do is copy this directory.
* The job directory may also be copied to clone a job or rename the directory.

Intermediate Level Jenkins Interview Questions

16. How can you deploy a custom build of a core plugin?

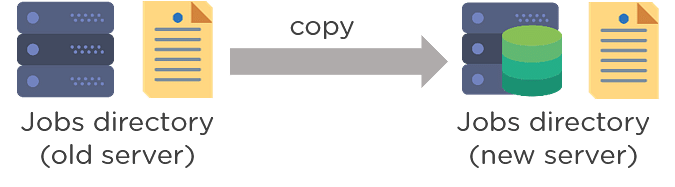
If you wish to deploy a custom build of a core plugin, you follow the following steps:

* Stop Jenkins
* Then copy the custom HPI to $Jenkins\_Home/plugins
* After that, delete the previously expanded plugin directory
* Next, make an empty file called <plugin>.hpi.pinned
* Finally, start Jenkins

17. What could be the steps to move or copy Jenkins from one server to another?

There are multiple ways to move or copy Jenkins from one server to another:

* You may move a job from one Jenkins installation to another just by copying the corresponding job directory.
* You may make a copy of an already existing job by making a clone of the job directory with an uncommon name.
* You may also just rename a current job by renaming a directory.



18. Name some more continuous Integration tools other than Jenkins.

Some of the top continuous integration tools other than Jenkins are:

* TeamCity
* Travis CI
* Go CD
* Bamboo
* GitLab CI
* CircleCI
* Codeship

19. Assume that you have a pipeline. The first job that you performed was successful, but the second one failed.  What would you do now?

You don't have to worry, and you just have to restart the pipeline from the point where it failed by doing 'restart from stage.'

20. Explain the process in which Jenkins works?

Here’s the process in which Jenkins works:

* Jenkins checks changes in repositories regularly, and developers must secure their code regularly.
* Once the changes are defined, Jenkins detects them and uses them to prepare a new build.
* After that, Jenkins will transverse through various stages in its usual pipeline. As one stage completes, the process will move further on to the next stage.
* If a stage fails, the Jenkins build will stop there, and the software will email the team using it. When completed successfully, the code implements itself in the proper server so that testing begins.
* After the successful testing phase, Jenkins shares the results with the team using it.

21. 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.

22. Differentiate between Maven, Ant, and Jenkins.

|  |  |  |
| --- | --- | --- |
| Maven | ANT | Jenkins |
| 1. Build tool 2. Perform build operations | 1. Build tool 2. Perform build operations | 1. Continuous Integration tool 2. Jenkins may run unit tests and deploy applications |

23. Differentiate between Bamboo and Jenkins?

|  |  |  |
| --- | --- | --- |
| S.no | Bamboo | Jenkins |
| 1 | Commercial tool | Open-source tool |
| 2 | Dedicated development team | Huge global community |
| 3 | Comparatively more user friendly | Less user-friendly |
| 4 | Many built-in features and plugins are available in the Atlassian marketplace. | Many plugins to perform different functions |

24. What is the difference between Jenkins and Hudson?

|  |  |  |
| --- | --- | --- |
| S.no | Jenkins | Hudson |
| 1 | Jenkins is a free open-source Continuous Integration server. | Hudson is an extensible Continuous Integration server |
| 2 | Forked by Hudson | Continuous Integration tool |
| 3 | Used by companies like Netflix, Facebook, eBay, Instacart, LinkedIn, etc. | Used by companies like Logo Yazilim, TableAir UAB and OptoSweden AB. |
| 4 | Jenkins supports a lot of plugins | Hudson supports a lesser number of plugins |

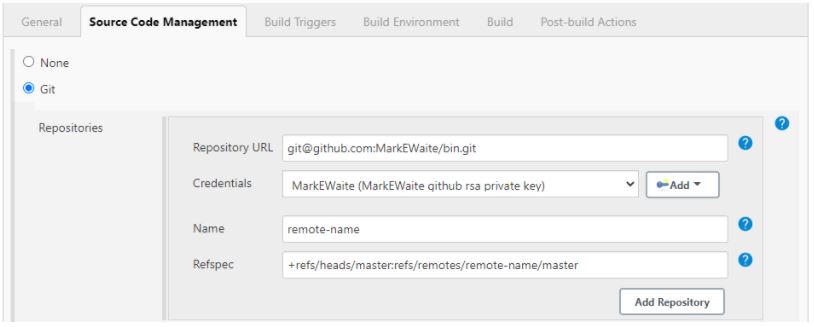
25. Why is Jenkins used with Selenium?

Using [Selenium](https://www.simplilearn.com/tutorials/selenium-tutorial/what-is-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. What is the process to integrate Git with Jenkins?

To integrate Git with Jenkins, you can follow the following steps:

* First, create a new Jenkins job and open the Jenkins dashboard.
* Now, enter the desired project name and select the job type.
* Click on OK.
* Then enter the project information.
* After that, visit the 'Source Code Management' tab.



Source: <https://plugins.jenkins.io/git/>

* If the Git plugin is pre-installed in Jenkins, there will be 'Git'.
* If it is not installed, you must reinstall the plugins (GitHub plugin, GitHub Branch Source plugin, GitHub API plugin, Git client plugin, etc.).
* After we install the plugins, restart Jenkins.
* To check if Git is installed, you can go to Command Prompt and type Git, and you would see various options like usage, version, help, etc.

27. Explain Kubernetes, and how can you integrate Jenkins with Kubernetes?

* [Kubernetes](https://www.simplilearn.com/tutorials/kubernetes-tutorial/what-is-kubernetes) is a portable and open-source platform that is used for managing workloads and services that are containerized.
* With the help of Kubernetes, the group of hosts running the Linux containers can be easily and efficiently managed.
* To manage a Continuous Delivery (CD) pipeline, the most efficient way is to deploy Jenkins with Kubernetes Engine.
* Kubernetes enables the creation of multiple container instances to satisfy more fault tolerance.
* Kubernetes deploy plug may be used with Jenkins for Continuous Deployment.Free Course: Introduction to Kubernetes

28. What is DSL Jenkins?

DSL stands for Domain Specific Language. Jenkins job DSL is a plugin that allows us to define jobs in the programmatic form with minimal effort. You can describe your jobs in Jenkins using a Groovy Based Language. They designed Jenkins job DSL plugin to create versions of the job, manage the records

29. What is the process to configure Third-party tools in Jenkins?

The process to configure Third-party tools in Jenkins can be seen in four significant steps:

* Install the third-party software
* Then install a Jenkins plugin supporting the third-party tool
* Now, configure the tool from the Manage Jenkins section
* Finally, your plugin is ready to be used

30. What are some of the default environmental variables in Jenkins?

Some of the Jenkins environmental variables are:

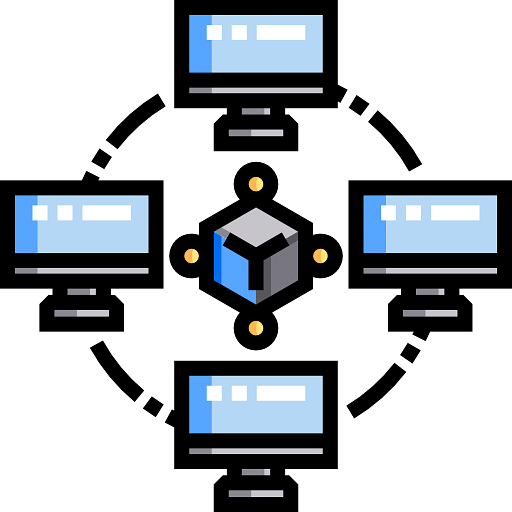
* $JOB\_NAME - The name that you give your job when it is first set up.
* $NODE\_NAME - This is the name of the node on which the current build is running.
* $WORKSPACE - Refers to the path of the workspace
* $BUILD\_URL - Indicates the URL where the results of the builds can be found.
* $JENKINS\_URL - This is set to the URL of the Jenkins master that is responsible for running the build.

Advance Level Jenkins Interview Questions

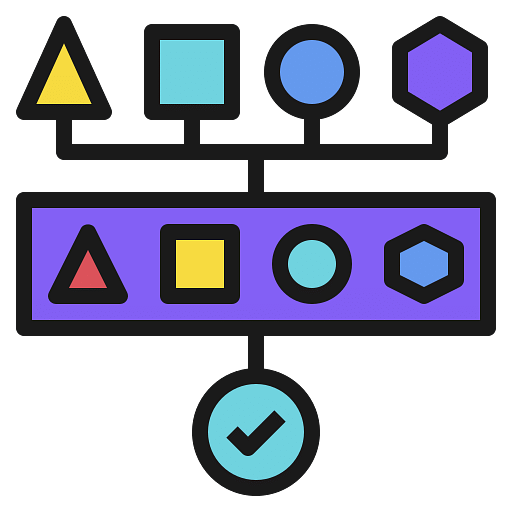
31. What are some of the critical aspects of the Jenkins pipeline?

Some of the Jenkins Pipeline key aspects are:

* Pipeline: User-defined model of a CD pipeline. Pipeline's code takes the role of defining the entire build process, including building, testing, and delivering an application.
* Node: A machine as a part of the Jenkins environment which is capable of executing a pipeline.



* Step: An individual task that communicates to Jenkins about what to do at a particular point in time
* Stage: This defines distinct subset of tasks that are conceptually unique and performed through the pipeline (build, test, deploy stages)



32. Let's say there is a broken build in the Jenkins project, then what can be done?

Initially, you will have to open the console output where the broken builds are created and then figure out if there are any file changes that were missed. In case there are no issues found there, then you will need to update your local workspace, replicate the problem, and then try to solve it.

33. How to deploy a custom build of a core plugin?

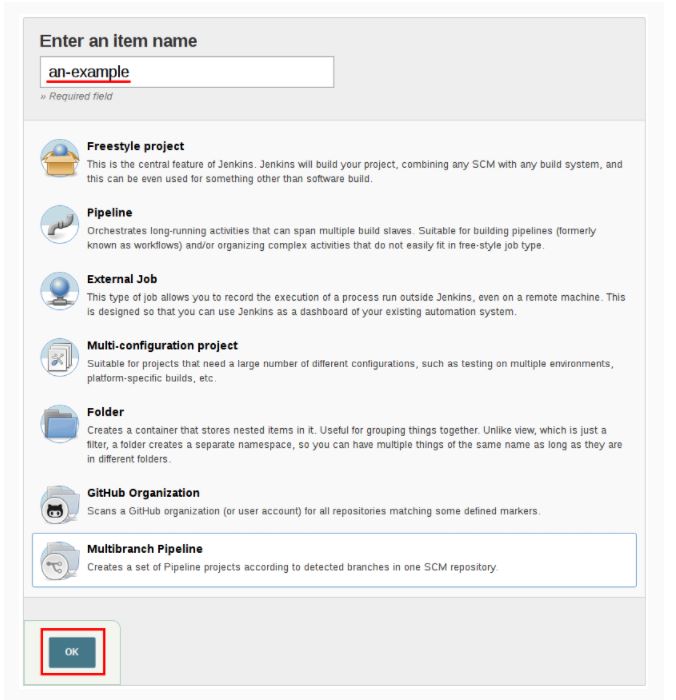
The steps to deploy a custom build of a core plugin are:

* First, copy the .hpi file to $JENKINS\_HOME/plugins
* Then remove the plugin's development directory
* Next, create an empty file called <plugin>.hpi.pinned
* Finally, restart Jenkins and use your custom build of a core plugin

34. What is the process of making a Multibranch Pipeline in Jenkins?

To create a Multibranch Pipeline in Jenkins, follow the following steps:

* Open the Jenkins dashboard and create a new item by clicking on 'new item'
* Enter the project name and, from the options, select 'Multibranch pipeline'
* Click on OK



Source: <https://www.jenkins.io/doc/book/pipeline/multibranch/>

* Then select the repository location, branch source (GitHub/Bitbucket), and add the branch source credentials.
* Save the project
* Now, Jenkins automatically creates new Multibranch Pipelines for repositories
* Then to connect to the GitHub repo, we need the HookURL
* To get this URL from the repository settings, add this HookURL to the Webhooks section
* Once the jobs are created, Jenkins will automatically trigger the build

35. How can the parameters be defined in Jenkins?

In Jenkins, a build can take many input parameters to execute.

* To define parameters for the job, select the “this project is parameterized” box.
* The drop down “Add Parameter” is enabled with the parameter types list. Any number of parameters may be added in the list.

There are several parameter types provided in the list.

36. Explain the ways to configure Jenkins node agent to communicate with Jenkins master?

There are two ways to configure Jenkins node agent to communicate with Jenkins master:

* 1. Browser–If we launch the Jenkins node agent from a browser, a Java Web Start or JNLP file is downloaded. The downloaded file launches a new process on the client machine to run jobs.
  2. Command-line–If you want to start the node agent using the command line, you need the executable agent.jar file. When this file runs, it launches a client's process to communicate with the Jenkins master to run build jobs.

37. What is the use of the JENKINS\_HOME directory?

* JENKINS\_HOME directory is the place where all the settings, logs, and configurations are stored. It stores all this information in XML files.
* The directory contains a subdirectory for every Jenkins build job being operated.
* Every directory has two subdirectories:  builds and workspace., and some other files as well.
* These sub directories are important, as the workspace directory is located at the place where Jenkins is building the project, and it contains the source code.
* The builds directory stores the history of all the builds performed for this job.



38. Explain a backup plugin and its uses.

It includes job configs, plugins, logs, plugin configuration, etc. Jenkins provides a backup plugin which can be used to get critical backup configuration. This is most important when there is a failure; it prevents the loss of any settings.

39. What do you understand by a trigger concerning a pipeline?

A trigger is something that defines when and how the pipelines should be executed. There may be several triggers like a pull request trigger that is used to deploy a pull request, or there may be a stage trigger that is used in configuring how each stage in the release will be triggered.

40. What are the three security mechanisms Jenkins uses to authenticate users?

The three mechanisms are as follows:

* Jenkins uses an internal database to store user data and credentials.
* Jenkins can use a lightweight Directory Access Protocol (LDAP) server to authenticate users.
* We can configure Jenkins to employ the application server's authentication mechanism upon which we deploy it.

Conclusion

By now, you should be well prepared for a Jenkins interview. You now know the answers to the top 40 questions from a beginner's level to an advanced level.

In addition to these Jenkins interview questions, there are also many other helpful [DevOps interview questions](https://www.simplilearn.com/tutorials/devops-tutorial/devops-interview-questions) that you can check out and supplement your DevOps knowledge.

If you have any questions for us, please let us know in the comment section of the Jenkins Interview Questions, and we will get our experts to answer it for you.

However, if you want to learn more about Jenkins and DevOps in general, you can check this fantastic set of[DevOps Certification Training courses](https://www.simplilearn.com/devops/?source=GhPreviewCoursepages) that will be valuable in helping you land that coveted job in development.

**1. What is Jenkins?**

Jenkins is a self-contained, open-source automation server that can be used to automate all sorts of tasks related to building, testing, and delivering or deploying software. Jenkins can be installed through native system packages, Docker, or even run standalone by any machine with a Java Runtime Environment (JRE) installed.

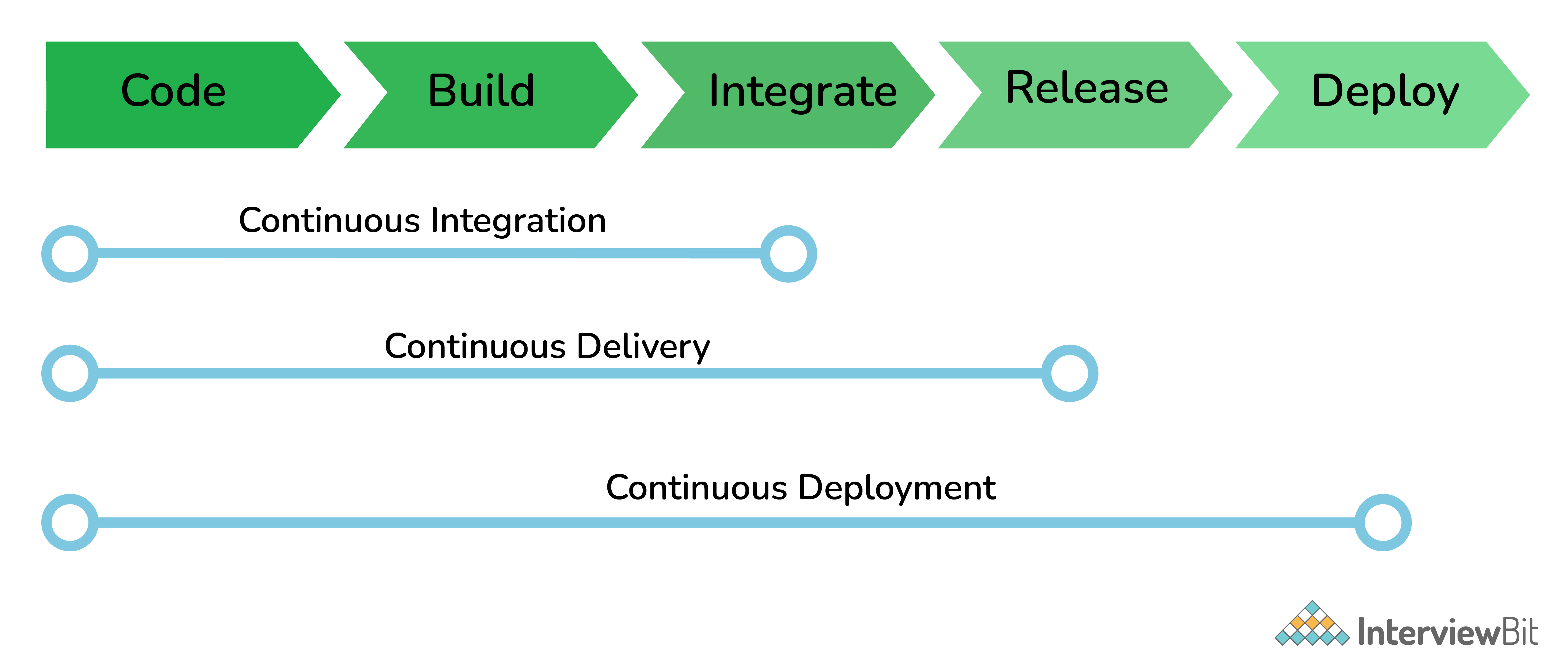
**2. Tell me something about Continuous Integration, Continuous Delivery, and Continuous Deployment?**

**Continuous Integration:** A software development process where the changes made to software are integrated into the main code as and when a patch is ready so that the software will be always ready to be - built, tested, deployed, monitored - continuously.

**Continuous Delivery:** This is a Software Development Process where the continuously integrated (CI) changes will be tested & deployed continuously into a specific environment, generally through a manual release process, after all the quality checks are successful

**Continuous Deployment:** A Software Development practice where the continuously integrated (CI) changes are deployed automatically into the target environment after all the quality checks are successful

Based on the level of automation, the above three paradigms can be better represented as below -

CI/CD and Continuous Deployment

**3. What are the common use cases Jenkins is used for?**

Jenkins being open-source automation can be used for any kind of software-based automation. Some of the common use-cases include but not limited to -

* Software build jobs
* Sanity/Smoke/CI/Regression test jobs
* Web/Data Scraping related jobs
* Code coverage measurement jobs
* General-purpose automation
* Reverse Engineering jobs
* Key Decoding jobs & many other jobs where software automation will be applicable.

**You can download a PDF version of Jenkins Interview Questions.**

[**Download PDF**](javascript:void(0))

**4. What are the ways to install Jenkins?**

Jenkins can be installed using -

1. Native System Package Manager like - apt (Linux), brew (Mac), etc.
2. Docker (popular docker images for Jenkins is available for different platforms like Unix/Mac/Windows in the docker registry)
3. Kubernetes (available as a helm chart and can be installed on our Kubernetes clusters)
4. Standalone (on any machine with a Java Runtime Environment installed)

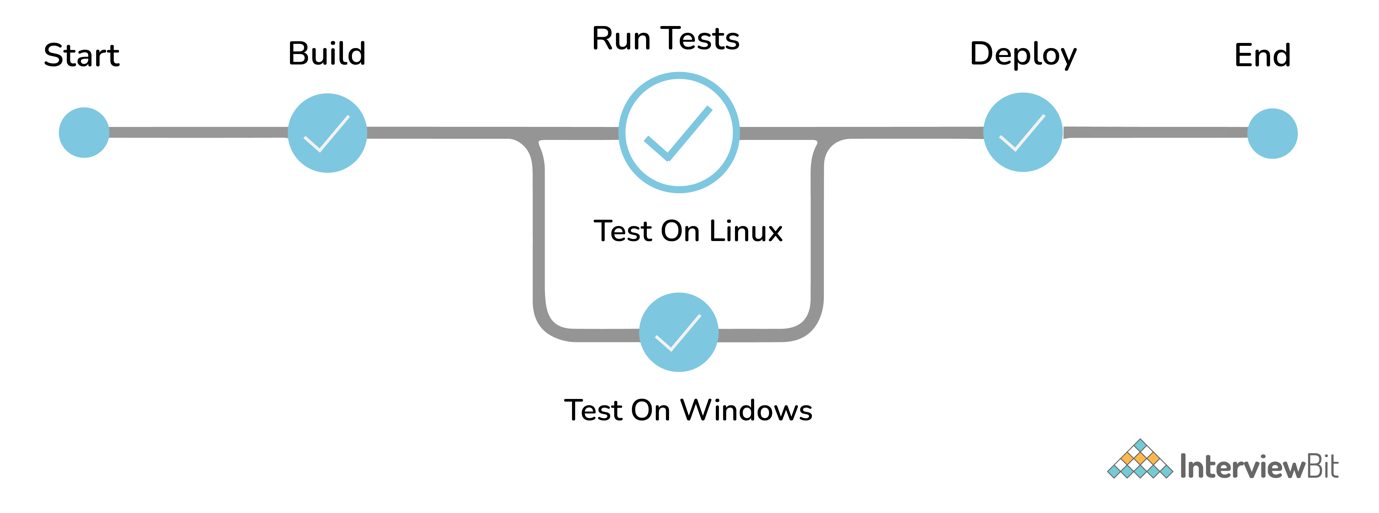
For more detailed installation instructions refer [official documentation](https://www.jenkins.io/doc/book/installing/)

**5. What is a Jenkins job?**

A Job/Project is the fundamental unit of a logical work (like a software build, an automation task, test execution, etc) using the Jenkins automation server and other required plugins, configurations & infrastructures.

Jobs can be of different types like - a freestyle project, a multi-configuration project, a pipeline project, a multi-branch project, etc.

**6. What is a Jenkins Pipeline?**

Jenkins Pipeline

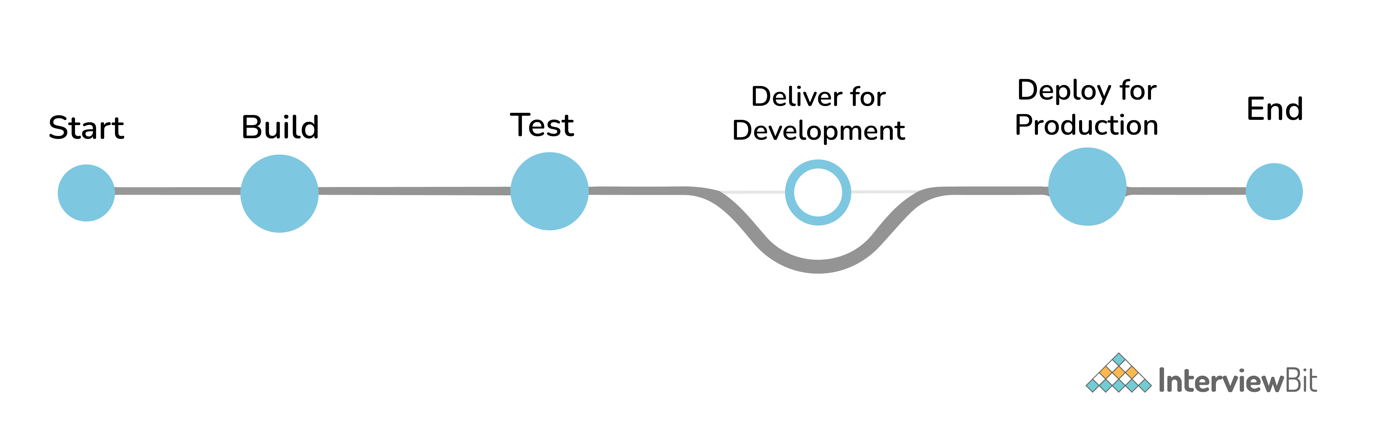
The pipeline is a special type of Jenkins job - simply a sequence of steps controlled by a defined logic - which Orchestrates long-running activities that can span across multiple build agents. It is suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that cannot be easily achieved using a freestyle job.

Understand how to create your first pipeline [here](https://www.jenkins.io/doc/pipeline/tour/hello-world/)

**7. What are the types of Jenkins pipelines?**

Jenkins Pipelines can be either - a Declarative pipeline or a Scripted Pipeline. Declarative pipeline makes use of numerous, generic, predefined build steps/stages (i.e. code snippets) to build our job according to our build/automation needs whereas, with Scripted pipelines, the steps/stages can be custom-defined & used using a groovy syntax which provides better control & fine-tuned execution levels.

**8. Explain Jenkins Multibranch Pipeline?**

Jenkins Multibranch Pipeline

It is a pipeline job that can be configured to Create a set of Pipeline projects according to the detected branches in one SCM repository. This can be used to configure pipelines for all branches of a single repository e.g. if we maintain different branches (i.e. production code branches) for different configurations like locales, currencies, countries, etc.

**9. How do you store credentials in Jenkins securely?**

Credentials can be stored securely in Jenkins using the Credentials plugin, which stores different types of credentials like - Username with a password, SSH username with the private key, AWS Credentials, Jenkins Build Token, Secret File/Text, X509 & other certificates, Vault related credentials securely with proper encryption & decryption as and when required.

**10. How can we stop a scheduled job from being executed temporarily?**

Disable the job from the job details page to temporarily stop all scheduled executions & other factors/events from triggering the job and enable it back to resume the job schedules/triggers. If a job is not required permanently, we can delete the job from the jobs list view page.

**Intermediate Questions**

**11. What are the ways to trigger a Jenkins Job/Pipeline?**

There are many ways we can trigger a job in Jenkins. Some of the common ways are as below -

* Trigger an API (POST) request to the target job URL with the required data.
* 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.

**12. What is Jenkins Build Cause?**

Build Cause is a text attribute that represents what made a job's build to be triggered, say it could be a Jenkins User (from UI), Timer for Scheduled jobs, Upstream jobs for a job which was triggered by upstream job, etc. This is mainly used to identify the nature of the builds - be it nightly, manual, automated, etc.

**13. 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.

**14. What are the credential types supported by Jenkins?**

In Jenkins, credentials are a set of information used for authentication with internal/external services to accomplish an action. Jenkins credentials are provisioned & managed by a built-in plugin called - Credentials Binding - plugin. Jenkins can handle different credentials as follows -

* 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.

And as we can guess, this can be extended to several other extensible credential types like - AWS credential, Azure secrets, etc. using commonly available plugins.

**15. What are the Scopes of Jenkins Credentials?**

Jenkins credentials can be of one of the two scopes - Global & 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.

**16. What is a Jenkins Shared Library and how it is useful?**

As an organization starts using more and more pipeline jobs, there is a chance for more and more code being duplicated in every pipeline job, since a part of the build/automation processes will be the same for most of the jobs. In such a situation, every other new upcoming job should also duplicate the same piece of code. To avoid duplications, the Jenkins project brought in the concept of Shared Libraries, to code - DRY - Don't Repeat Yourself.

Shared libraries are a set of code that can be common for more than one pipeline job and can be maintained separately. Such libraries improve the maintenance, modularity & readability of the pipeline code. And it also speeds up the automation for new jobs.

**17. How Jenkins jobs can be Triggered/Stopped/Controlled programmatically?**

Jenkins Remote Access API can be used to do things like -

* Retrieving information about jobs, views, nodes, builds, etc. from Jenkins for programmatic consumption.
* Trigger a build (both parameterized & non-parameterized), stop/abort a build, enable/disable a Job, group/remove jobs into/from views, etc.
* Create/copy/modify/delete jobs.

and many other programming language-specific functionalities. It has wrappers for main programming languages like - Python, Ruby & Java. It can be triggered via CURL as below -

**Jobs without parameters**

Simply an HTTP POST on JENKINS\_URL/job/JOBNAME/build.

**Jobs with parameters**

Simple example - sending "String Parameters":

curl JENKINS\_URL/job/JOB\_NAME/buildWithParameters  --user USER:TOKEN --data id=123 --data verbosity=high

**18. How to get the Jenkins version programmatically in Jobs/Pipelines or nodes other than master?**

To check the version of Jenkins, load the top-level page or any top-level Remote Access API path like the '.../api/\*' page and then check for the 'X-Jenkins' response header.

This contains the version number of Jenkins, like "1.404". This is also a good way to check if an URL is a Jenkins URL.

**19. What happens when a Jenkins agent is offline and what is the best practice in that situation?**

When a job is tied to a specific agent on a specific node, the job can only be run on that agent and no other agents can fulfill the job request. If the target node is offline or all the agents on that particular node are busy building other jobs, then the triggered job has to wait until the node comes online or an agent from that node becomes available to execute the triggered build request.

As a result, a triggered job may sometimes wait indefinitely without knowing that the target node is offline. So, it is always the best practice to tie the jobs to a group of nodes & agents, referred to with a 'Label'. Once a job is tied to a Label, instead of a specific node/agent, any of the nodes/agents falling under the label can fulfill a build request, when a job is triggered. This way we can reduce the overall turn-around time of the builds.

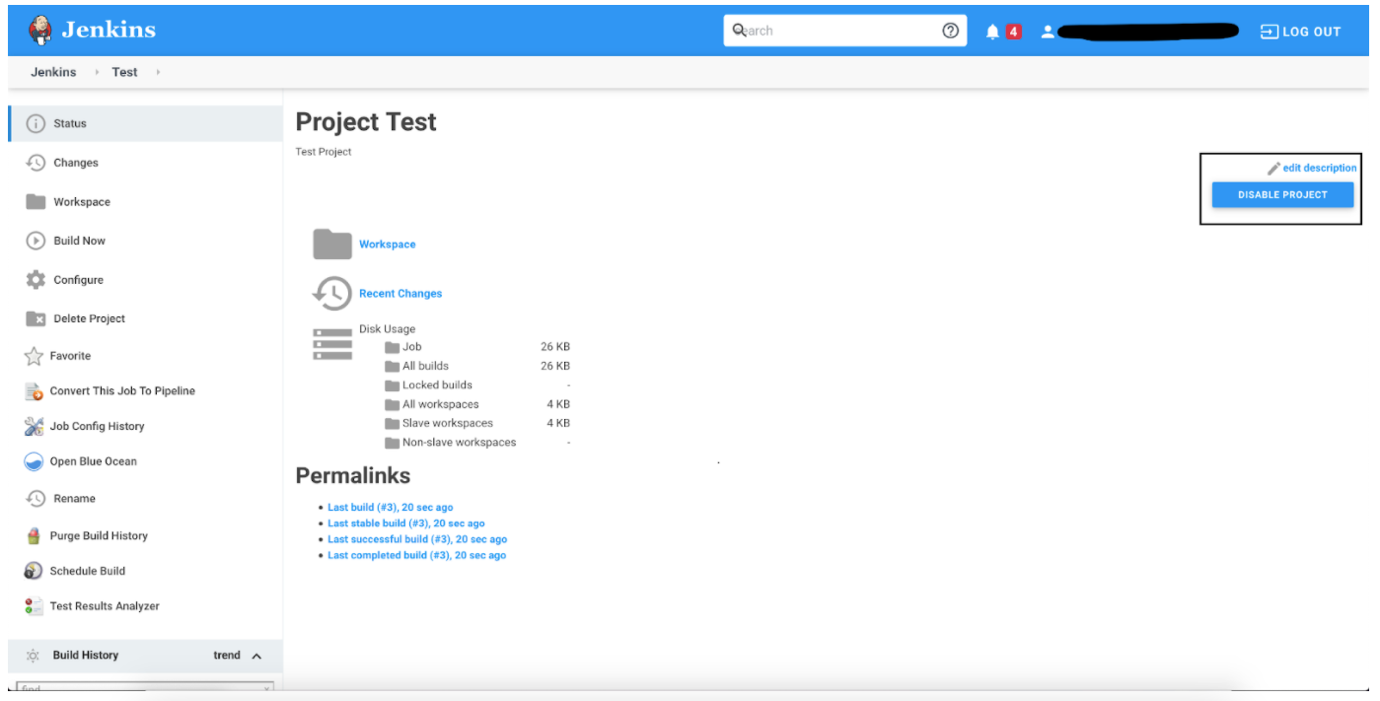
Even then if a job is waiting for more time for the nodes/agents, then it is time to consider adding more nodes/agents.

**20. 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. Blue Ocean’s main features include -

* Sophisticated visualizations of continuous delivery (CD) Pipelines, allowing for fast and intuitive comprehension of your Pipeline’s status.
* Pipeline editor - makes the creation of Pipelines approachable by guiding the user through an intuitive and visual process to create a Pipeline.
* Personalization to suit the role-based needs of each member of the team.
* Pinpoint precision when intervention is needed and/or issues arise. Blue Ocean shows where in the pipeline attention is needed, facilitating exception handling and increasing productivity.
* Native integration for branch and pull requests, enables maximum developer productivity when collaborating on code with others in GitHub, Bitbucket, etc.

**Conventional UI - Job Details Page**

Conventional UI Jenkins

**21. What is the Jenkins User Content service?**

Jenkins has a mechanism known as "User Content", where administrators can place files inside the **$JENKINS\_HOME/userContent** folder and these files are served from yourhost/jenkins/userContent.

This can be thought of as a mini HTTP server to serve images, stylesheets, and other static resources that you can use from various description fields inside Jenkins.

**Advanced Interview Questions**

**22. How is continuous integration achieved using Jenkins?**

Continuous integration is a process where a developer’s code changes are constantly integrated into the main code and the same will be tested automatically and the results of the tests will decide whether the change is ready for deployment. In this process -

* Developer Makes a change - commit/pull\_request - in feature/dev branch
* Source Control Management system generates appropriate events
* SCM Specific Jenkins Plugins like Git/SVN will detect those events from the configured repositories and these events will be used to Trigger - build/dependent/test - jobs on Jenkins
* After the Test/Dependent jobs are completed, the change/patch will be labeled according to the status of the test job
* Based on the Status (i.e. readiness of a change to be merged with the main branch), the Continuous Delivery or Continuous Deployment strategy/tool will take it forward.

**23. 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 plugin called - Copy artifact Plugin. Based on the configured pattern, the files/directories matching the configured patterns will be archived for a Jenkins build, which can be used for future references. In the pipeline, it can be configured as follows -

archiveArtifacts artifacts: 'output/\*\*/\*'

**24. How to configure inclusions & exclusions in Artifacts Archival?**

Artifact archival takes in a pattern for matching target files. Similarly, it also takes in a pattern (ANT build system pattern for matching files) for exclusion as well which will be ignored while selecting the files for archival.

For e.g.  
archiveArtifacts artifacts: 'output/\*.txt', excludes: 'output/specific\_file.txt'

The above command will archive all the text files from the output folder except specific\_file.txt

**25. How can we share information between different build steps or stages in a Jenkins Job?**

Every build step or stage will be running in its process and hence sharing information between two different build steps is not so direct. We can use either a File, a Database Entry, an Environment Variable, etc. to share info from one build step to another or a post-build action.

**26. How code coverage is measured/tracked using Jenkins in a CI environment?**

Using language-specific code coverage plugins like JaCoCo, CodeCov, etc or generic tools/plugins like Sonarqube which will add the code coverage data to builds with some minor tweaks in the code and the same can be displayed as a graph in Jenkins.

**27. Default Environment Variables by Jenkins & How to introduce custom environment variables?**

Jenkins provides several environment variables by default like - **BRANCH\_NAME, BUILD\_NUMBER, BUILD\_TAG, WORKSPACE, etc.**

**28. 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.

**29. 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. Hence it needs its corresponding Jenkins plugin as well. 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.

**30. How to create & use a Shared Library in Jenkins?**

Basic requirements for a Jenkins shared library to be used in a Pipeline Code are -

* A Repository with pipeline shared library code in SCM.
* An appropriate SCM Plugin configuration for the Jenkins instance.
* Global Shared Library should be configured in Jenkins Global configuration.
* Include the Shared Library in the Pipeline Code and use the methods defined in the Jenkins Shared Library.

**E.g.**

#!/urs/bin/env groovy  
@Library('fs\_jenkins\_shared\_library@v2.0.7')\_

**31. How to install a Custom Jenkins Plugin or a Version of Plugin Not available in Jenkins Update Center?**

Generally, it is the best practice to use the latest version of a plugin. But there are ways to install custom plugins or outdated versions of a published plugin. Jenkins Plugins are exported using a .hpi file and the same can be installed in multiple ways -

**Using the Jenkins CLI**

java -jar jenkins-cli.jar -s http://localhost:8080/ install-plugin SOURCE ... [-deploy] [-name VAL] [-restart]

The above command Installs a plugin either from a file, an URL or from the update center.

* SOURCE: If this points to a local file, that file will be installed. If this is an URL, Jenkins downloads the URL and installs that as a plugin. Otherwise, the name is assumed to be the short name of the plugin in the existing update center (like "findbugs") and the plugin will be installed from the update center.
* -deploy: Deploy plugins right away without postponing them until the reboot.
* -name VAL: If specified, the plugin will be installed as this short name (whereas normally the name is inferred from the source name automatically).
* -restart: Restart Jenkins upon successful installation.

**Advanced Installation - via - Web UI**

Assuming a .hpi file has been downloaded, a logged-in Jenkins administrator may upload the file from within the web UI:

* Navigate to the Manage Jenkins > Manage Plugins page in the web UI.
* Click on the Advanced tab.
* Choose the .hpi file under the Upload Plugin section.
* Upload the plugin file.
* Restart the Jenkins instance

**Advanced Installation - via - On the master**

Assuming a .hpi file has been explicitly downloaded by a systems administrator, the administrator can manually place the .hpi file in a specific location on the file system.

Copy the downloaded .hpi file into the JENKINS\_HOME/plugins directory on the Jenkins controller (for example, on Debian systems JENKINS\_HOME is generally /var/lib/jenkins).

The master will need to be restarted before the plugin is loaded and made available in the Jenkins environment.

**32. How to download the Console log for a particular Jenkins build programmatically?**

**Using the Jenkins CLI - console - command**

java -jar jenkins-cli.jar console JOB [BUILD] [-f] [-n N]

Produces the console output of a specific build to stdout, as if you are doing 'cat build.log'

* JOB: Name of the job
* BUILD: Build number or permalink to point to the build. Defaults to the last build
* -f: If the build is in progress, append console output as it comes, like tail -f
* -n N: Display the last N lines.

**E.g.**

ssh -l <ssh\_username> -p <port\_no> <Jenkins\_URL> console <JOB\_NAME>

**33. What is Jenkins Remote Access API?**

Jenkins provides remote access API to most of its functionalities (though some functionalities are programming language-dependent). Currently, it comes in three flavors -

* XML
* JSON with JSONP support
* Python

Remote access API is offered in a REST-like style. That is, there is no single entry point for all features, and instead, they are available under the ".../api/" URL where the "..." portion is the data that it acts on.

For example, if your Jenkins installation sits at interviewbit.com, visiting /api/ will show just the top-level API features available – primarily a listing of the configured jobs for this Jenkins instance.

Or if we want to access information about a particular build, e.g. https://ci.jenkins.io/job/Infra/job/jenkins.io/job/master/lastSuccessfulBuild/, then go to https://ci.jenkins.io/job/Infra/job/jenkins.io/job/master/lastSuccessfulBuild/api/ and you’ll see the list of functionalities for that build.

**34. What is In-process Script Approval and how it works?**

Jenkins, and several plugins, allow users to execute Groovy scripts in Jenkins. To protect Jenkins from the execution of malicious scripts, these plugins execute user-provided scripts in a Groovy Sandbox that limits what internal APIs are accessible.

This protection is provided by the Script Security plugin. As soon as an unsafe method is used in any of the scripts, the "In-process Script Approval" action should appear in "Manage Jenkins" to allow Administrators to make a decision about which unsafe methods, if any, should be allowed in the Jenkins environment.

This in-process script approval inherently improves the security of the overall Jenkins ecosystem.

**35. Can we monitor Jenkins using common Observability tools?**

Common monitoring platforms like **DataDog, Prometheus, JavaMelody** & few others - have their corresponding Jenkins plugin, which when configured, sends Metrics to the corresponding Monitoring platform, which can then be Observed with the latest tools & technologies. The same can be configured with Alarms & Notifications for immediate attention when something goes wrong.

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

Jenkins installs "ping thread" on every remote connection, such as Controller/Agent connections, regardless of its transport mechanism (such as SSH, JNLP, etc.). The lower level of the Jenkins Remoting Protocol is a message-oriented protocol, and a ping thread periodically sends a ping message that the receiving end will reply to. The ping thread measures the time it takes for the reply to arrive, and if it’s taking excessive time (currently 4 minutes and configurable), then it assumes that the connection was lost and initiates the formal close down.

This is to avoid an infinite hang, as some of the failure modes in the network cannot be detected otherwise. The timeout is also set to a long enough value so that a temporary surge in the load or a long garbage collection pause will not trip off the close-down.

Ping thread is installed on both controller & agent; each side pings the other and tries to detect the problem from their sides.

The ping thread time out is reported through java.util.logging. Besides, the controller will also report this exception in the agent launch log. Note that some agent launchers, most notably SSH agents, writes all stdout/stderr outputs from the agent JVM into this same log file, so you need to be careful.

**Conclusion**

Though these are not the complete possibilities of Jenkins, we tried to cover some of the commonly asked interview questions on core Jenkins. We also need to understand that the Jenkins Update Center is enriched with thousands of useful plugins that enhance the supported functionalities of Jenkins.

Before appearing for an interview, make sure to install a Jenkins Server on any of the supported platforms - either locally or on the cloud, install the most common plugins (suggested by Jenkins itself & other commonly used plugins). Try creating & building a normal freestyle project with Git or any other SCM integration plugin and try to execute some code from the connected Git Repository.

Also, try creating a pipeline project with JenkinsFile and a global shared Jenkins library and build the job successfully. This will help us learn how Jenkins actually works with some hands-on issues.

**Basic Interview Questions**

**Q1. What is the difference between Jenkins and Bamboo?**

|  |  |  |
| --- | --- | --- |
| **Parameters** | **Jenkins** | **Bamboo** |
| **Open Source** | Jenkin is open-source | Bamboo is not open source |
| **Pricing Logic** | Jenkin is completely free | It charges for the number of build agents required |
| **Operating System** | Windows, Ubuntu, Red Hat, Mac OS | Windows, Linux, Solaris |
| **Browsers** | Chrome, Firefox, Internet Explorer | Firefox, Chrome, Safari, Edge |
| **Plugin Support** | Yes, It supports a lot of plugins | It does not support many plugins as compared to Jenkins |
| **Support** | Being open-source, it has a lot of support from communities | It has less support as compared to Jenkins |

**Q2. What is Jenkins?**

Jenkins is an open-source automation tool written in Java with plugins built for Continuous Integration purposes. Jenkins is used to build and test your software projects continuously making it easier for developers to integrate changes to the project, and making it easier for users to obtain a fresh build. It also allows you to continuously deliver your software by integrating with a large number of testing and deployment technologies.

**Q3. Define the process of Jenkins.**

* First, a developer commits the code to the source code repository. Meanwhile, the Jenkins server checks the repository at regular intervals for changes.
* Soon after a commit occurs, the Jenkins server detects the changes that have occurred in the source code repository. Jenkins will pull those changes and will start preparing a new build.
* If the build fails, then the concerned team will be notified.
* If the build is successful, then Jenkins deploys the build in the test server.
* After testing, Jenkins generates feedback and then notifies the developers about the build and test results.
* It will continue to check the source code repository for changes made in the source code and the whole process keeps on repeating.

****

**Q4. What are the benefits of using Jenkins?**

I will suggest you include the following benefits of Jenkins if you can recall any other benefit apart from the below-mentioned points you can include that as well.

* At the integration stage, you can cache build failures.
* For each change in the source code, you generate an automatic build report notification.
* To notify developers about build report success or failure, Jenkins integrates with the LDAP mail server.
* Achieves continuous integration agile development and test driven development.
* With simple steps, you can automate the maven release project.
* Easy tracking of bugs at an early stage in a development environment than production.

**Q5.** **What are the pre-requisites for using Jenkins?**

The answer to this is pretty straightforward. To use Jenkins you require:

* A source code repository which is accessible, for instance, a Git repository.
* A working build script, e.g., a Maven script, checked into the repository.

**Q6. What is the relation between Hudson and Jenkins?**

You can just say Hudson was the earlier name and version of current Jenkins. After some issues, they renamed the project from Hudson to Jenkins.

**Q7.** **Mention some of the useful plugins in Jenkins**

Below I have mentioned some important Plugins:

* Maven 2 project
* Git
* Amazon EC2
* HTML publisher
* Copy artifact
* Join
* Green Balls

Diagram

Description automatically generated

These Plugins I feel are the most useful plugins, if you want to include any other Plugin that is not mentioned above, you can add that as well, but make sure you first mention the above-stated plugins and then add your own.

**Q8. How do you install Jenkins?**

To install Jenkins, you just need to follow these five steps:

1. **Install Java Version 8** – Jenkins is a Java based application, hence Java is a must.
2. **Install Apache Tomcat Version 9** – Tomcat is essential to deploy Jenkins war file.
3. **Download Jenkins war File** – This war is must to install Jenkins.
4. **Deploy Jenkins war File** – You deploy Jenkins war file using Tomcat to run Jenkins.
5. **Install Suggested Plugins** – Install a list of plugins suggested by Jenkins.

Once the installation is complete, you will be able to see the Jenkins dashboard.

**Q9. What are the two components that you can integrate Jenkins with?**

According to me, the integration of Jenkins is possible with the following:

* Version Control system like GIT, SVN.
* Build tools like Apache Maven.

If you have anything else in your mind then mention that as well but make sure you include the above two components in your answer.

**Q10. What is Maven? What is the benefit of integrating Maven with Jenkins?**

[Maven](https://maven.apache.org/) is a build management tool. It uses a simple pom.xml to configure all the dependencies needed to build, test and run the code. Maven manages the full lifecycle of a test project. Once integrated with Jenkins, the maven Webdriver will build the project and execute all tests efficiently.

**Q11.** **Mention what are the commands you can use to start Jenkins manually.**

For this answer I will suggest you go with the below-mentioned flow:  
To start Jenkins manually open Console/Command line, then go to your Jenkins installation directory. Over there you can use the below commands:

Start Jenkins: **jenkins.exe start**  
Stop Jenkins: **jenkins.exe stop**  
Restart Jenkins: **jenkins.exe restart**

**Q12.** **Which SCM tools Jenkins supports?**

Here are some of the Source Code Management tools supported by Jenkins:

* AccuRev
* CVS
* Subversion
* Git
* Mercurial
* Perforce
* Clearcase
* RTC

**Q13. How will you define Post in Jenkins?**

**Post** is a section that contains several additional steps that might execute after the completion of the pipeline. The execution of all the steps within the condition block depends upon the completion status of the pipeline. The condition block includes the following conditions – **changed success, always, failure, unstable and aborted.**

**Q14. What are Parameters in Jenkins?**

Parameters are supported by Agent section and they are used to support various use-cases pipelines. Parameters are defined at the top-level of the pipeline or inside an individual stage directive.

**Q15. What is Groovy?**

Groovy from Apache is a language designed for the Java platform. It is the native scripting language for Jenkins. Groovy-based plugins enhance Jenkins with great interfaces and build reports that are of dynamic and consistent nature.

**Intermediate Questions**

**Q16. How Can You Clone A Git Repository Via Jenkins?**

If you want to clone a Git repository via Jenkins, you have to enter the e-mail and user name for your Jenkins system. Switch into your job directory and execute the “git config” command for that.

**Q17.** **Explain how you can set up Jenkins job.**

My approach to this answer will be to first mention how to create Jenkins job.

Go to Jenkins top page, select “New Job”, then choose “Build a free-style software project”.

Now you can tell the elements of this freestyle job:

* Optional SCM, such as CVS or Subversion where your source code resides.
* Optional triggers to control when Jenkins will perform builds.
* Some sort of build script that performs the build (ant, maven, shell script, batch file, etc.) where the real work happens.
* Optional steps to collect information out of the build, such as archiving the artifacts and/or recording javadoc and test results.
* Optional steps to notify other people/systems with the build result, such as sending e-mails, IMs, updating issue tracker, etc..

**Q18. How to create a backup and copy files in Jenkins?**

The answer to this question is really direct.

To create a backup all you need to do is to periodically back up your JENKINS\_HOME directory. This contains all of your build jobs configurations, your slave node configurations, and your build history. To create a back-up of your Jenkins setup, just copy this directory. You can also copy a job directory to clone or replicate a job or rename the directory.

**Q19. How will you secure Jenkins?**

The way I secure Jenkins is mentioned below if you have any other way to do it than mention that:

* Make sure that the global security is on.
* Check if Jenkins is integrated with my company’s user directory with an appropriate plugin.
* Ensure that the matrix/Project matrix is enabled to fine-tune access.
* Automate the process of setting rights/privileges in Jenkins with custom version controlled script.
* Limit physical access to Jenkins data/folders.
* Periodically run security audits on the same.

**Q20.** **Explain how you can deploy a custom build of a core plugin?**

Below are the steps to deploy a custom build of a core plugin:

* Stop Jenkins.
* Copy the custom HPI to **$Jenkins\_Home/plugins**.
* Delete the previously expanded plugin directory.
* Make an empty file called **<plugin>.hpi.pinned**.
* Start Jenkins.

**Q21. What you do when you see a broken build for your project in Jenkins?**

There can be multiple answers to this question I will approach this task in the following way:

I will open the console output for the broken build and try to see if any file changes were missed. If I am unable to find the issue that way, then I will clean and update my local workspace to replicate the problem on my local and try to solve it.

If you do it in a different way then just mention that in your answer.

**Q22.** **What are the various ways in which build can be scheduled in Jenkins?**

You can schedule a build in Jenkins in the following ways:

* By source code management commits
* After completion of other builds
* Can be scheduled to run at a specified time (crons)
* Manual Build Requests

**Q23. What is the use of Pipelines in Jenkins?**

Pipeline plugin is used in Jenkins for making the Jenkins Pipeline, which gives us the view of stages or tasks to perform one after the other in the pipeline form. It models a series of related tasks. Pipelines help the teams to review, edit and iterate upon the tasks. Pipelines are durable and it can optionally stop and wait for human approval as well to start the next task. A pipeline is extensible and can perform work in parallel. It supports complex CD requirements.

**Q24. Explain the terms Agent, post-section, Jenkinsfile**

**Agent:**It is directive to tell Jenkins to execute the pipeline in a particular manner and order.

**Post-section:**If we have to add some notification and to perform other tasks at the end of a pipeline, post-section will definitely run at the end of every pipeline’s execution.

**Jenkinsfile:**The text file where all the definitions of pipelines are defined is called Jenkinsfile. It is being checked in the source control repository.

**Q25. Do you know about cloud computing? How can Jenkins fit into a cloud computing environment? Explain with an example.**

Let us take the example of AWS cloud service. Cloud computing services use the CI/CD model so that they can push their work to the customers and constantly receive feedback. Jenkins is used to automating the CI/CD pipelines. For example, a lot of Jenkins plugins are available for many of the AWS services like Amazon EC2 and ECS.

**Q26. What is Kubernetes? How can you integrate Jenkins with Kubernetes?**

[Kubernetes](https://kubernetes.io/) is a container orchestration tool. With Kubernetes, one can create multiple container instances to achieve more fault tolerance. You can use the Kubernetes deploy plugin to use it with Jenkins for continuous deploy.

**Q27. Have you run automated tests on Jenkins? How is it done?**

Yes, this can be done easily. Automated tests can be run through tools like Selenium or maven. Developers can schedule the test runs. Jenkins displays the test results and sends a report to the developers.

**Q28.  Let us say, you have a pipeline. The first job was successful, but the second failed.  What should you do next?**

You just need to restart the pipeline from the point where it failed by doing ‘restart from stage’.

**Q29. What is the use of JENKINS HOME directory?**

All the settings, logs and configurations are stored in the JENKINS\_HOME directory.

**Q30. What is a backup plugin? Why is it used?**

This is a helpful plugin that backs up all the critical settings and configurations to be used in the future. This is useful in cases when there is a failure so that we don’t lose the settings.

**Q31. What is a trigger? Give an example of how the repository is polled when a new commit is detected.**

Triggers are used to define when and how pipelines should be executed.

When Jenkins is integrated with an SCM tool, for example, Git, the repository can be polled every time there is a commit.

* The Git plugin should be first installed and set up.
* After this, you can build a trigger that specifies when a new build should be started. For example, you can create a job that polls the repository and triggers a build when a change is committed.

**Q32. How do you define parameters for a build in Jenkins?**

A build can take several input parameters to execute. For example, if you have multiple test suites, but you want to run only one. You can set a parameter so that you are able to decide which one should be run. To have parameters in a job, you need to specify the same while defining the parameter. The parameter can be anything like a string, a file or a custom.

**Q33. What are the ways to configure Jenkins node agent to communicate with Jenkins master?**

There are 2 ways to start the node agent –

* **Browser** – if Jenkins node agent is launched from a browser, a JNLP (Java Web Start) file is downloaded. This file launches a new process on the client machine to run jobs.
* **Command-line** – to start the node agent using the command line, the client needs the executable agent.jar file. When this file is run, it simply launches a process on the client to communicate with the Jenkins master to run build jobs.

**Q34. How does Jenkins authenticate users?**

**There are 3 ways –**

* The default way is to store user data and credentials in an internal database.
* Configure Jenkins to use the authentication mechanism defined by the application server on which it is deployed.
* Configure Jenkins to authenticate against LDAP server.

**Q35.  How can you use a third-party tool in Jenkins?**

Below are the steps used for working with a third-party tool in Jenkins.

* First install the third-party software
* Download the plug-in that supports the third-party tool.
* Configure the third-party tool in the admin console.
* Then use the required plug-in from the Jenkins build job.

For different third-party tools, the procedure may vary slightly, because of the difference in configuration settings.

**Q36. What are the types of pipelines in Jenkins?**

There are 3 types –

1. CI CD pipeline (Continuous Integration Continuous Delivery)
2. Scripted pipeline
3. Declarative pipeline

**Q37. What syntax does Jenkins use to schedule build job or SVN polling?**

The cron syntax.

Cron syntax is represented using five asterisks each separated by a space. The syntax is as follows – [minutes] [hours] [day of the month] [month] [day of the week]. Example, if you want to set up a cron for every Monday at 11.59 pm, it would be 59 11 \* \* 1

**Q38. What is DevOps and in which stage does Jenkins fit in?**

DevOps is a software development practice that blends software development (Dev) with the IT operations (Ops) making the whole development lifecycle simpler and shorter by constantly delivering builds, fixes, updates, and features. Jenkins plays a crucial role because it helps in this integration by automating the build, test and deployment process.

**Q39. Do you know any other Continuous Integration tools? How is Jenkins better than any of those?**

There are many other CI tools, and the most prominent ones are –

* TeamCity
* Bamboo
* Perforce
* Circle CI
* Go
* ThoughtWorks
* Integrity
* Travis CI

There are many more. We cannot say if Jenkins is better than each because each has its own unique features. For example, TeamCity offers great .NET support but is complex and costly, Travis CI is free just like Jenkins and has good documentation too. Bamboo too offers efficient and faster builds but is not completely free and so on.

**Q40. Name a Jenkins environment variable you have used in a shell script or batch file.**

There are numerous environment variables that are available by default in any Jenkins build job. A few commonly used ones include:

* $JOB\_NAME
* $NODE\_NAME
* $WORKSPACE
* $BUILD\_URL
* $JOB\_URL

Note that, as new Jenkins plug-ins are configured, more environment variables become available. For example, when the Jenkins Git plug-in is configured, new Jenkins Git environment variables, such as $GIT\_COMMIT and $GIT\_URL, become available to be used in scripts.

**Q41. What is Continuous Integration In Jenkins?**

In software development, multiple developers or teams work on different segments of the same web application. So in this case, you have to perform integration testing by integrating all modules. In order to do that an automated process for each piece of code is performed on a daily bases so that all your codes get tested. This process is known as continuous integration.

**Q42. How do you achieve continuous integration using Jenkins?**

Here are the steps –

* All the developers commit their source code changes to the shared Git repository.
* Jenkins server checks the shared Git repository at specified intervals and detected changes are then taken into the build.
* The build results and test results are shared to the respective developers
* The built application is displayed on a test server like Selenium and automated tests are run.
* The clean and tested build is deployed to the production server.

**Q43. What is a DSL Jenkins?**

The Jenkins “Job DSL / Plugin” is made up of two parts – first, The Domain Specific Language (DSL) itself that allows users to describe jobs using a Groovy-based language, and second, a Jenkins plugin which manages the scripts and the updating of the Jenkins jobs which are created and maintained as a result.

**Q44. How do you create Multibranch Pipeline in Jenkins?**

The Multibranch Pipeline project type enables you to implement different Jenkinsfiles for different branches of the same project. In a Multibranch Pipeline project, Jenkins automatically discovers, manages and executes Pipelines for branches that contain a Jenkinsfile in source control.

**Q.45 What are the types of jobs or projects in Jenkins?**

These are the types of jobs/projects in Jenkins –

* Freestyle project
* Maven project
* Pipeline
* Multibranch pipeline
* External Job
* Multi-configuration project
* Github organization

**Q46. What is blue ocean in Jenkins?**

It is a project that was started with the purpose to rethink the user experience of Jenkins, modeling and presenting the process of software delivery by surfacing information that’s important to development teams. This is done with as few clicks as possible, while still staying true to the extensibility that is core to Jenkins. While this project is in the alpha stage of development, the intent is that Jenkins users can install Blue Ocean side-by-side with the Jenkins Classic UI via a plugin

**Advanced Questions**

**Q47. What is Continuous Testing?**

Continuous Testing is the process where you execute automated tests as part of the software delivery pipeline. This is done so that you get the feedback on the business risks associated with software as early as possible. It consists of evolving and extending test automation to address the increased complexity and pace of modern application development and delivery.

Continuous Testing means that testing takes place on a continuous basis without any disruption of any kind. In a Continuous DevOps process, a software change is continuously moving from Development to Testing to Deployment. The code undergoes continuous development, delivery, testing and deployment.

**Q48.** **Explain how you can move or copy Jenkins from one server to another?**

I will approach this task by copying the jobs directory from the old server to the new one. There are multiple ways to do that, I have mentioned it below:

You can:

* Move a job from one installation of Jenkins to another by simply copying the corresponding job directory.
* Make a copy of an existing job by making a clone of a job directory by a different name.
* Rename an existing job by renaming a directory. Note that if you change a job name you will need to change any other job that tries to call the renamed job.

**Q49. How do you integrate Git with Jenkins?**

These are the steps to integrate [Git](https://bit.ly/1LBvRP9) with Jenkins –

**1**.Click on the **Manage Jenkins** button on your Jenkins dashboard:

**Graphical user interface, text, application

Description automatically generated**

**2**. Click on **Manage Plugins.**

**Graphical user interface, text, application

Description automatically generated**

**3.**In the Plugins Page

1. Select the GIT Plugin
2. Click on **Install without restart.**The plugin will take a few moments to finish downloading depending on your internet connection, and will be installed automatically.
3. You can also select the option **Download now and Install after restart.**
4. You will now see a “No updates available” message if you already have the Git plugin installed.

**4.**Once you install the plugins , go to **Manage Jenkins**on your Jenkins dashboard. You will see your plugins listed among the rest.

**Graphical user interface, text, application, email

Description automatically generated**

**Q50. How can you temporarily turn off Jenkins security if the administrative users have locked themselves out of the admin console?**

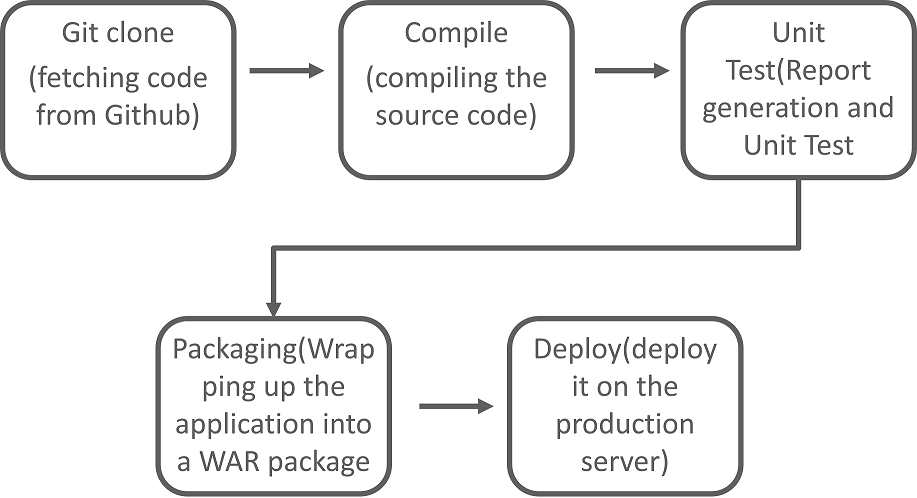
The JENKINS\_HOME folder contains a file named config.xml. When you enable the security, this file contains an XML element named useSecurity that changes to true. If you change this setting to false, security will be disabled the next time Jenkins is restarted.

<useSecurity>false</useSecurity>

However, we must understand that disabling security should always be both a last resort and a temporary measure. Once you resolve the authentication issues, make sure that you re-enable Jenkins security and reboot the CI server.

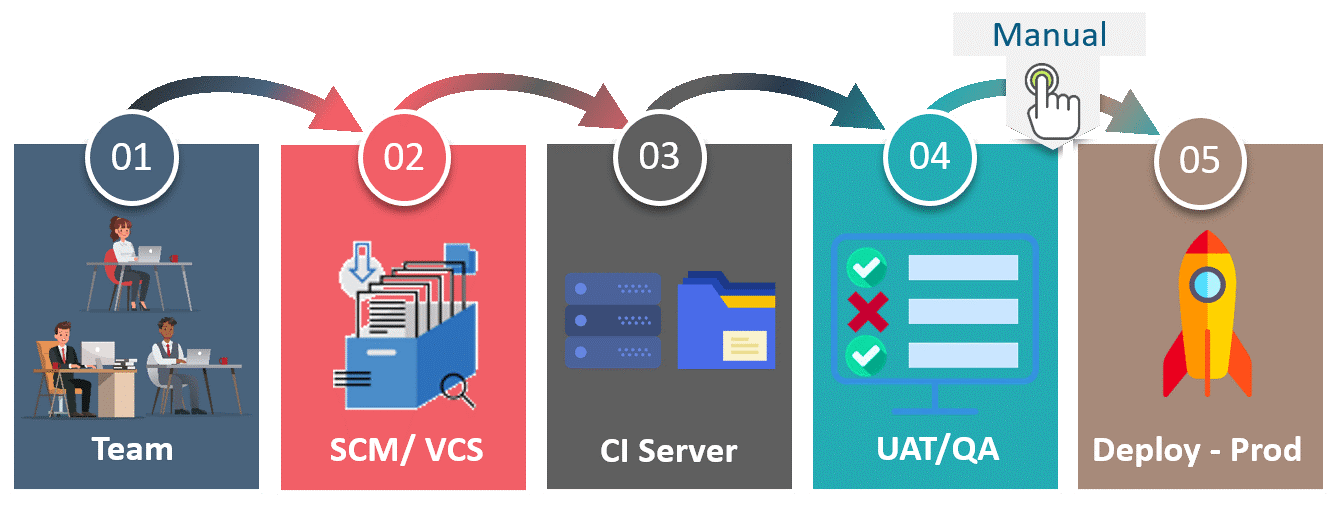
**Q51. Can you define a Continuous Delivery Workflow?**

The flowchart below shows the Continuous Delivery Workflow. Hope it will be much easier to understand with visuals.

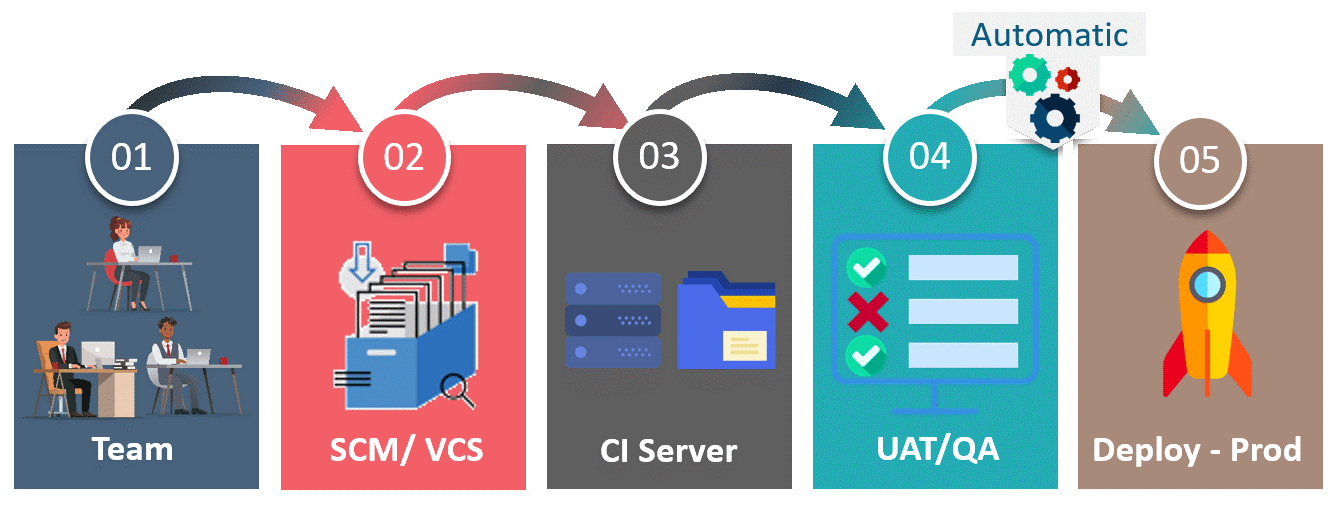
****

**Q52. What is the difference between Continuous Delivery and Continuous Deployment?**

**Continuous Delivery: (Manual Deployment to Production. Does not involve every change to be deployed.)**

Continuous Delivery is a software development practice where you build software in such a way that the software can be released to the production at any time. You achieve Continuous Delivery by continuously integrating the products built by the development team, running automated tests on those built products to detect problems and then push those files into production-like environments to ensure that the software works in production.

**Continuous Deployment: (Automated Deployment to Production. Involves deploying every change automatically)**

Continuous deployment means that every change that you make, goes through the pipeline, and if it passes all the tests, it automatically gets deployed into production. So, with this approach, the quality of the software release completely depends on the quality of the test suite as you have automated everything.

**Q53. What do you mean by Pipeline as a Code?**

**Pipeline as Code** describes a set of features that allow Jenkins users to define pipelined job processes with code, stored and versioned in a source repository. These features allow Jenkins to discover, manage, and run jobs for multiple source repositories and branches — eliminating the need for manual job creation and management.

To use Pipeline as Code, projects must contain a file named Jenkinsfile in the repository root, which contains a “Pipeline script.”

Additionally, one of the enabling jobs needs to be configured in Jenkins:

* **Multibranch Pipeline**: build multiple branches of a *single* repository automatically
* **Organization Folders**: scan a **GitHub Organization** or **Bitbucket Team** to discover an organization’s repositories, automatically creating managed Multibranch Pipeline jobs for them

Once you have prepared yourself with these Jenkins interview questions, then no one can stop you from getting your dream job.

I have included the frequently asked Docker interview questions. If you have more questions in your mind just type it in the comment box below and we will reply you ASAP. Before going for the interview I will suggest you to [***check out this Jenkins blog series***](https://www.edureka.co/blog/what-is-jenkins/).

**1. What is Jenkins?**

Jenkins is a self-contained, open-source automation server that can be used to automate all sorts of tasks related to building, testing, and delivering or deploying software. Jenkins can be installed through native system packages, Docker, or even run standalone by any machine with a Java Runtime Environment (JRE) installed.

**2) Tell me something about Continuous Integration, Continuous Delivery, and Continuous Deployment?**

Continuous Integration: A software development process where the changes made to software are integrated into the main code as and when a patch is ready so that the software will be always ready to be – built, tested, deployed, monitored – continuously.

Continuous Delivery: This is a Software Development Process where the continuously integrated (CI) changes will be tested & deployed continuously into a specific environment, generally through a manual release process, after all the quality checks are successful

Continuous Deployment: A Software Development practice where the continuously integrated (CI) changes are deployed automatically into the target environment after all the quality checks are successful

**3) What are the common use cases Jenkins is used for?**

Jenkins being open-source automation can be used for any kind of software-based automation. Some of the common use-cases include but not limited to –

* Software build jobs
* Sanity/Smoke/CI/Regression test jobs
* Web/Data Scraping related jobs
* Code coverage measurement jobs
* General-purpose automation
* Reverse Engineering jobs
* Key Decoding jobs & many other jobs where software automation will be applicable.

**4) What are the ways to install Jenkins?**

Jenkins can be installed using –

* Native System Package Manager like – apt (Linux), brew (Mac), etc.
* Docker (popular docker images for Jenkins is available for different platforms like Unix/Mac/Windows in the docker registry)
* Kubernetes (available as a helm chart and can be installed on our Kubernetes clusters)
* Standalone (on any machine with a Java Runtime Environment installed)

**5) What is a Jenkins job?**

A Job/Project is the fundamental unit of a logical work (like a software build, an automation task, test execution, etc) using the Jenkins automation server and other required plugins, configurations & infrastructures.

Jobs can be of different types like – a freestyle project, a multi-configuration project, a pipeline project, a multi-branch project, etc.

**6. What are the types of Jenkins pipelines?**

Jenkins Pipelines can be either – a Declarative pipeline or a Scripted Pipeline. Declarative pipeline makes use of numerous, generic, predefined build steps/stages (i.e. code snippets) to build our job according to our build/automation needs whereas, with Scripted pipelines, the steps/stages can be custom-defined & used using a groovy syntax which provides better control & fine-tuned execution levels.

**7) How do you store credentials in Jenkins securely?**

Credentials can be stored securely in Jenkins using the Credentials plugin, which stores different types of credentials like – Username with a password, SSH username with the private key, AWS Credentials, Jenkins Build Token, Secret File/Text, X509 & other certificates, Vault related credentials securely with proper encryption & decryption as and when required.

**8) How can we stop a scheduled job from being executed temporarily?**

Disable the job from the job details page to temporarily stop all scheduled executions & other factors/events from triggering the job and enable it back to resume the job schedules/triggers. If a job is not required permanently, we can delete the job from the jobs list view page.

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

There are many ways we can trigger a job in Jenkins. Some of the common ways are as below –

* Trigger an API (POST) request to the target job URL with the required data.
* 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.

**10) What is Jenkins Build Cause?**

Build Cause is a text attribute that represents what made a job’s build to be triggered, say it could be a Jenkins User (from UI), Timer for Scheduled jobs, Upstream jobs for a job which was triggered by upstream job, etc. This is mainly used to identify the nature of the builds – be it nightly, manual, automated, etc.

**11) 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.

**12) What are the credential types supported by Jenkins?**

In Jenkins, credentials are a set of information used for authentication with internal/external services to accomplish an action. Jenkins credentials are provisioned & managed by a built-in plugin called – Credentials Binding – plugin. Jenkins can handle different credentials as follows –

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.  
And as we can guess, this can be extended to several other extensible credential types like – AWS credential, Azure secrets, etc. using commonly available plugins.

**13) What are the Scopes of Jenkins Credentials?**

Jenkins credentials can be of one of the two scopes – Global & 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.

**14) What is a Jenkins Shared Library and how it is useful?**

As an organization starts using more and more pipeline jobs, there is a chance for more and more code being duplicated in every pipeline job, since a part of the build/automation processes will be the same for most of the jobs. In such a situation, every other new upcoming job should also duplicate the same piece of code. To avoid duplications, the Jenkins project brought in the concept of Shared Libraries, to code – DRY – Don’t Repeat Yourself.

Shared libraries are a set of code that can be common for more than one pipeline job and can be maintained separately. Such libraries improve the maintenance, modularity & readability of the pipeline code. And it also speeds up the automation for new jobs.

**15) How do you store credentials in Jenkins securely?**

Credentials can be stored securely in Jenkins using the Credentials plugin, which stores different types of credentials like – Username with a password, SSH username with the private key, AWS Credentials, Jenkins Build Token, Secret File/Text, X509 & other certificates, Vault related credentials securely with proper encryption & decryption as and when required.

**16) What is Jenkins and why is it used?**

Jenkins is a continuous integration (CI) tool for real-time testing and reporting of smaller builds in a large chunk of code. It is written in Java. It is used because it helps developers and testers work in tandem to detect and close defects early in the software development lifecycle and encourage automated testing of builds.

**17) List some features of Jenkins.**

Features of Jenkins are –

* Free and open source
* Excellent community and documentation
* Exhaustive set of plugins and integrations
* Easy to set up, install and use on any platform because it is based on Java
* Supports distributed builds due to master-slave architecture, thus reducing the load on the CI server

**18) What are the advantages of using Jenkins?**

Advantages of Jenkins are –

* Provides great collaboration between development and operations team, making it into a single DevOps team
* Code errors can be detected as early as possible.
* Code deployment is easy and happens in minutes, along with the generation of reports.
* Automation of integration work, thereby reducing the number of integration issues.

**19) How did Jenkins come into existence?**

Originally called Hudson, Jenkins came into existence when Oracle took over the ownership to continue development of the product, by renaming it as Jenkins.

**20) How is continuous integration achieved using Jenkins?**

Here are the steps –

* All the developers commit their source code changes to the shared Git repository.
* Jenkins server checks the shared Git repository at specified intervals and detected changes are then taken into the build.
* The build results and test results are shared to the developers
* The built application is displayed on a test server like Selenium and automated tests are ran.
* The clean and tested build is deployed to the production server.

**21) Do you know any other continuous integration tools? How is Jenkins better than any of those?**

There are many other CI tools, the prominent ones being –

TeamCity  
Bamboo  
Perforce  
Circle CI  
Go  
ThoughtWorks  
Integrity  
Travis CI  
There are many more. We cannot say if Jenkins is better than each because each have their own unique features. For example, TeamCity offers great .NET support but is complex and costly, Travis CI is free just like Jenkins and has good documentation too. Bamboo offers efficient and faster builds but is not completely free and so on.

**22) What is DevOps and in which stage does Jenkins fit in?**

DevOps is a software development practice which blends software development (Dev) with the IT operations (Ops) making the whole development lifecycle simpler and shorter by constantly delivering builds, fixes, updates, and features. Jenkins plays a great role because it helps in this integration by automating the build, test and deployment process.

**23) What are the system requirements to install Jenkins?**

The minimum configuration required is –

* 256MB of RAM
* 1 GB of drive space
* Java
* Web browser

**24) Give some important plugins in Jenkins.**

Here you go –

Maven 2  
Gits  
Amazon EC2  
Join  
Copy artifact  
Green Balls  
HTML Publisher

**25) What is Groovy?**

Groovy from Apache is a language for Java platform. It is the native scripting language for Jenkins. Groovy-based plugins enhance Jenkins with great interfaces and build reports that are dynamic and consistent.

**26) Give a simple use case/scenario to explain how Jenkins works.**

Let us say a developer is working on some code changes and eventually commits them to the repository.  
Jenkins server, which constantly checks for changes in the repository, detects the change and pulls the changes to trigger a build.  
The build can fail, in which case the developer is informed with reports.  
If the build passes, it is deployed on to the test server.  
Once the testing is complete, a test report is generated and sent to the developers. This process continues till all the tests are successful, after which code is deployed to production.

**27) Can you start Jenkins using command line? How?**

Yes, using jenkins.exe start

**28) What are the SCM tools that Jenkins supports?**

The SCM or Source Code Management tools Jenkins supports are SVN, Clearcase, CVS, Git, AccuRev, Perforce, RTC, Mercurial.

**29) What is a job in Jenkins?**

A job or build job is a task or step in the entire build process. It could be compiling the source code, running unit tests, deploying the application to the web server and so on.

**30) How can you create a job?**

On the dashboard page, you can just select a ‘New Job’. When you create a job, you can choose options such as the SCM, triggers to control, the build script and notifications.

**31) What is meant by Jenkins pipeline?**

A pipeline is a group of interlinked jobs done one after the other in a sequence. To integrate and implement continuous delivery pipelines, Jenkins pipelines provides a combination of plugins. The instructions to be performed are given through code.

**32) What are the types of pipelines in Jenkins?**

There are 3 types –

* CI CD pipeline (Continuous Integration Continuous Delivery)
* Scripted pipeline
* Declarative pipeline

**33) How do you install Jenkins?**

To install Jenkins, make sure the following are installed –

Java (version 8)  
Apache Tomcat (version 9)  
Download the Jenkins war file and deploy it using Tomcat. You can choose to install the plugins suggested by Jenkins during the installation itself. Once the installation is done, you will be able to see the Jenkins dashboard.

**34) Explain what is continuous integration?**

In software development, when multiple developers or teams are working on different segments of same web application, we need to perform integration test by integrating all modules. In order to do that an automated process for each piece of code is performed on daily bases so that all your code get tested.

**35) What is the requirement for using Jenkins?**

To use Jenkins you require

A source code repository which is accessible, for instance, a Git repository  
A working build script, e.g., a Maven script, checked into the repository

**36) Explain how you can move or copy Jenkins from one server to another?**

Slide a job from one installation of Jenkins to another by copying the related job directory  
Make a copy of an already existing job by making clone of a job directory by a different name  
Renaming an existing job by renaming a directory.

**37) Mention what are the commands you can use to start Jenkins manually?**

To start Jenkins manually, you can use either of the following

(Jenkins\_url)/restart: Forces a restart without waiting for builds to complete  
(Jenkin\_url)/safeRestart: Allows all running builds to complete

**38) Mention some of the useful plugins in Jenkin?**

Some of the important plugins in Jenkin includes

* Maven 2 project
* Amazon EC2
* HTML publisher
* Copy artifact
* Join
* Green Balls

**39) Explain how you can deploy a custom build of a core plugin?**

To deploy a custom field of a core plugin, you have to do following things

* Stop Jenkins
* Copy the custom HPI to $Jenkins\_Home/plugins
* Delete the previously expanded plugin directory
* Make an empty file called .hpi.pinned
* Start Jenkins

**40) Explain how can create a backup and copy files in Jenkins?**

Jenkins saves all the setting, build artifacts and logs in its home directory, to create a back-up of your Jenkins setup, just copy this directory. You can also copy a job directory to clone or replicate a job or rename the directory.

**41) Explain how you can clone a Git repository via Jenkins?**

To clone a Git repository via Jenkins, you have to enter the e-mail and user name for your Jenkins system. For that, you have to switch into your job directory and execute the “git config” command.

**42) Explain how you can set up Jenkins job?**

To create a project that is handled via jobs in Jenkins. Select New item from the menu, once this done enter a name for the job and select free-style job. Then click OK to create new job in Jenkins. The next page enables you to configure your job.

**43) Mention what are the two components Jenkins is mainly integrated with?**

Jenkin is mainly integrated with two components

Version Control system like GIT, SVN  
And build tools like Apache Maven.

**44) Why should we use Jenkins?**

Jenkins is often used to consistently design and validate software applications. It makes it easy for developers to implement updates to the code and for users to get the latest version quickly. Jenkins uses plugins to accomplish Continuous Integration. Plugins essentially allow DevOps stages to be integrated.

**45) How does Jenkins achieve Continuous Integration?**

Continuous Integration is a programming process that requires developers to commit updates to the source code in a public repository many times a day or at frequent intervals. A repository is created for any commitments made. As a result, any challenges can be spotted by the team early on. Moreover, the Continuous Integration platform performs various other tasks, such as deploying the development application to the test server, informing the relevant teams about the construct and test performance, etc. Jenkins uses plugins to accomplish Continuous Integration. Plugins generally allow DevOps stages to be integrated.

**46) Explain DevOps and how Jenkins is used in it?**

Azure DevOps (formerly known as Visual Studio Team Services – VSTS) is a series of tools for creating, evaluating, and distributing software. Developing and installing software has become even more effective with Azure DevOps. It is not only a series of software for automating CI-CD using the Microsoft stack but it can also be conveniently integrated with a variety of other third-party tools.

**47) List the system requirements for the installation of Jenkins.**

Jenkins requires a few basic requirements to be installed on your Windows system.

Hardware Prerequisites

Jenkins needs at least 256 MB of RAM on your computer or laptop to run.  
Jenkins can take up at least 1 GB of space on your hard disc.  
Software Prerequisites

Since Jenkins is based on Java, you’ll need the most recent update of either the Java Development Kit (JDK) or the Java Runtime Environment (JRE).

**48) Explain the process of installation of Jenkins**

The measures below should be taken to install Jenkins effectively:

* Go to https://www.jenkins.io/download/ and choose the platform you want to use.
* Unzip the downloaded package from your local computer’s download spot. Double-click the jenkins.msi file that has been unzipped.
* Press next on the Jenkins setup pad.
* Selecting the directory where you want the Jenkins instance to be saved (the default location is C: Program Files (x86)Jenkins).
* Click on the install option.
* When the installation is completed, click on finish.

**49) Name the Jenkins suite’s essential plugins?**

The Jenkin suite’s essential plugins are Docker, Jira, Slack Notification, Maven, Amazon E2C, jUnit, Pipeline, Mailer, and Greenballs.

**50) Explain what Groovy means?**

Jenkins uses a domain-specific language (DSL) called “Groovy” inside a Pipeline Project (read plugin), which will describe a new pipeline as just a script. That single script may articulate a flow that would typically take several “standard” Jenkins jobs chained around.

Groovy will work in conjunction with Java, and the two languages’ syntaxes are very close. While writing Groovy, when you forget the grammar, you can write in Java syntax. Groovy may be used as one of the Java platform’s scripting languages. Groovy scripts could be named from inside Java, reducing the amount of time spent on Java development.

**51) Explain Jenkins’ working in a simple use-case?**

Let’s assume a developer is operating on a code and contributes it to the repository.

Jenkins server scans for adjustments in the repository detects the code and pulls it to start a build.  
If the build malfunctions, the results are sent to the developer to make changes.  
If it succeeds, the build is deployed to a test server.  
A test report is produced and submitted to the developer when the testing is completed. This method is repeated until the code passes all the checks, after which it is deployed to output.

**52) How does one start using Jenkins from the command line?**

The Jenkins Web application ARchive (WAR) file can be started from the command line in the following manner:

* Download the latest stable Jenkins WAR file to an appropriate directory on your machine.
* Open up a terminal/command prompt window to the download directory.
* Run the command java -jar Jenkins.War.
* Browse to http://localhost:8080 and wait until the unlock Jenkins page appears.
* Continue with the post-installation setup wizard below.