**Jenkins interview questions:**

**What is Jenkins?**

* Jenkins is an open source automation server. Jenkins is a continuous integration tool.
* Jenkins helps to automate the non-human part of software development process, with continuous integration and facilitating technical aspects of continuous delivery.

**Why do we use Jenkins?**

* Jenkins is an open-source continuous integration software tool for testing and reporting on isolated changes in a larger code base in real time.
* The Jenkins software enables developers to find and solve defects in a code base rapidly and to automate testing of their builds.

**What is the process of Jenkins?**

* First of all, a software developer commits code to the SCR (Source Code Repository). Now, Jenkins server will check the repository on regular intervals for changes.
* As soon as changes are committed, they are analyzed by the server and Jenkins will pull those changes and start making new build instantly.
* In case, the build is not successful then the concerned team will be notified. At the same time, if the build is successful then the build will be deployed to the test server.
* Once the testing is complete, Jenkins will send the feedback and developers are notified for the new build or test results.
* This is not a one-time process, but Jenkins server keeps on monitoring the repository regularly and the same process will be repeated as discussed above.

**What is Jenkins File and why it is needed in DevOps?**

The Jenkins pipeline details are stored in a file that is named as the Jenkins file and it is further connected to the source code repository in general.

**What is Maven and what is Jenkins?**

* Maven is a build tool, in short a successor of ant. It helps in build and version control. However, Jenkins is continuous integration system, where in maven is used for build. Jenkins can be used to automate the deployment process.

**What is meant by continuous integration in Jenkins?**

* Continuous integration is a process in which all development work is integrated as early as possible.
* The resulting artifacts are automatically created and tested.
* This process allows to identify errors as early as possible. Jenkins is a popular open source tool to perform continuous integration and build automation.

**Why do we use Jenkins with selenium?**

* Running Selenium tests in Jenkins allows you to run your tests every time your software changes and deploy the software to a new environment when the tests pass.
* Jenkins can schedule your tests to run at specific time.

**What is a CI CD pipeline?**

* A continuous integration and deployment pipeline (CD/CI) is such an important aspect of a software project. It saves a ton of manual, error-prone deployment work.
* It results in higher quality software for continuous integration, automated tests, and code metrics.

**What is build pipeline in Jenkins?**

* Job chaining in Jenkins is the process of automatically starting other job(s) after the execution of a job.
* This approach lets you build multi-step build pipelines or trigger the rebuild of a project if one of its dependencies is updated.

**What is a Jenkins pipeline?**

* Is workflow with the group of events/jobs that are chained and integrated with each other in a sequence.
* Based on a Domain Specific Language (DSL) in Groovy, the Pipeline plugin makes pipelines scriptable and it is an incredibly powerful way to develop complex, multi-step DevOps pipelines.
* By setting the schedule period to 15 13 \* \* \* you tell Jenkins to schedule the build every day of every month of every year at the 15th minute of the 13th hour of the day.
* The Jenkins Pipeline plugin is a game changer for Jenkins users.

**Example:**

Once code is committed to git, then it will go to CI, then it il will go to development environment, from development environment, it go to pipeline, then commit, build, Test, Stage, Deploy, QA environment, etc.

**What is Multibranch pipeline?**

Branch indexing is way how branches are there present in the repository and creates build for every branch..

Jenkins uses a cron expression, and the different fields are:

MINUTES Minutes in one hour (0-59)

HOURS Hours in one day (0-23)

DAYMONTH Day in a month (1-31)

MONTH Month in a year (1-12)

DAYWEEK Day of the week (0-7) where 0 and 7 are sunday

Example:

If you want to schedule your build every 5 minutes, this will do the job: \*/5 \* \* \* \*

If you want to schedule your build every day at 8h00, this will do the job: 0 8 \* \* \*

**The steps for schedule jobs in Jenkins:**

click on "Configure" of the job requirement

scroll down to "Build Triggers" - sub title

Click on the checkBox of Build periodically

Add time schedule in the Schedule field, for example, @midnight

The format is as follows:

MINUTE (0-59), HOUR (0-23), DAY (1-31), MONTH (1-12), DAY OF THE WEEK (0-6)

The letter H, representing the word Hash can be inserted instead of any of the values. It will calculate the parameter based on the hash code of you project name.

This is so that if you are building several projects on your build machine at the same time, let’s say midnight each day, they do not all start their build execution at the same time. Each project starts its execution at a different minute depending on its hash code.

You can also specify the value to be between numbers, i.e. H(0,30) will return the hash code of the project where the possible hashes are 0-30.

Examples:

Start build daily at 08:30 in the morning, Monday - Friday: 30 08 \* \* 1-5

Weekday daily build twice a day, at lunchtime 12:00 and midnight 00:00, Sunday to Thursday: 00 0,12 \* \* 0-4

Start build daily in the late afternoon between 4:00 p.m. - 4:59 p.m. or 16:00 -16:59 depending on the projects hash: H 16 \* \* 1-5

Start build at midnight: @midnight or start build at midnight, every Saturday: 59 23 \* \* 6

Every first of every month between 2:00 a.m. - 02:30 a.m.: H(0,30) 02 01 \* \*

**Poll SCM vs build periodically:**

Poll SCM periodically polls the SCM to check whether changes were made (i.e. new commits) and builds the project if new commits where pushed since the last build, whereas build periodically builds the project periodically even if nothing has changed.