Jenkins

1] What is Jenkins ?

It is a continuous integration tool written in Java.Jenkins is an open source tool with plugin built for continuous integration purpose. The principle functionality of Jenkins is to keep a track of version control system and to initiate and monitor a build system if changes occur. It monitors the whole process and provides reports and notifications to alert.

2] 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.

3] What is the requirement for 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.

4] What are the advantages of Jenkins?

At integration stage, build failures are cached

For each code commit changes an automatic build report notification generates

To notify developers about build report success or failure, it is integrated with LDAP mail server Achieves continuous integration agile development and test driven development

With simple steps, maven release project is automated

Easy tracking of bugs at early stage in development environment than production

5] 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.

6] 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**

7] 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**

8] 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 <plugin>.hpi.pinned**
- **Start Jenkins**

9] 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.

10] 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.

11] 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.

12] 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.**

13] Why continuous integration is important ?

Two important reasons:

- **Defects found early cost less to fix :** When a defect is found immediately after a developer codes it, it takes 10x times less time to fix it compared to finding the defect a month later.
- **Reduced Time to Market :** Software is always tested. So, it is always ready to move to further environments.

14] How a Continuous Integration implemented ?

Different tools for supporting Continuous Integration are **Hudson, Jenkins and Bamboo.** Jenkins is the most popular one currently. They provide integration with various version control systems and build tools.

15] What is the difference between Maven, Ant and Jenkins ?

Maven and Ant are Build Technologies whereas Jenkins is a continuous integration tool.

16] Which SCM tools Jenkins supports ?

AccuRev, CVS, Subversion, Git, Mercurial, Perforce, Clearcase and RTC.

17] What are the various ways in which build can be scheduled in Jenkins ?

Builds can be triggered by source code management commits.

Can be triggered after completion of other builds.

Can be scheduled to run at specified time (crons)

Manual Build Requests

18] What is the relation between hudson and Jenkins ?

Hudson was the earlier name and version of current Jenkins. After some issue , the project name was changed from Hudson to Jenkins.

19] What you do to make sure that your project build doesn't break in Jenkins ?

I make sure that I perform successful clean install on my local machine with all unit tests.

Then I make sure that I check in all code changes.

Then I do a Synchronize with repository to make sure that all required config and POM changes and any difference is checked into the repository.

20] What you do when you see a broken build for your project in Jenkins ?

I will open the console output for the build and will try to see if any file changes were missed.

If not able to find the issue that way, Will clean and update my local workspace to replicate the problem on my local and will try to solve it.

21] What is continuous Integration ?

Continuous Integration is a software development practice where members of a team integrate their work (Code) frequently, usually each person integrates at least daily - leading to multiple integrations per day. By having scheduled automated builds we can determine if are there any integration issues , also by running the Unit Testcases as part of these builds and verify if is the existing functionality is broken. Automated tools such as CruiseControl, Jenkins, Hudson, Bamboo, BuildMaster, AnthillPro or Teamcity offer this scheduling automatically.

Here are the principles of "Continuous Integration":

Maintain a code repository : Code should be maintained in a Version Control such as CVS, SVN, VSS, ClearCase etc., which could allow multiple developers to work collaboratively in parallel by Versioning the Files (code).

Automate the build :

Should Have scheduled automated builds by using the tools like "Hudson,TeamCity, CruiseControl" etc., which could automatically checkout(Get) the code from Code Repository and build.

Make the build self-testing :

During the build process, after compiling the code we can make the Code Self-Testing by executing the Unit Test cases such as JUNIT or Cactus or EasyMock etc. and confirms that nothings is broken.

Automate deployment :

Once build and self-testing process is done, we should have automated Deployments. Tools such as "Hudson,TeamCity, CruiseControl" do complete "Continuous Integation" for you, but you have to have some build script such as ANT or MAVEN scripts to perform some of these Tasks.

Here is the sequence of tasks that could be done by above tools.

i) Having Scheduled Builds. (Daily or Hours)

i) Checkout the code.

ii) Compile the Code.

iii) Running Unit Test-Casess.

iv) FTP code to different hosts

v) Deploy the artifacts.

22] What is version control ?

A "Version control" system (also known as a Revision Control System) is a repository of files, often the files for the source code of computer programs, with monitored access. Every change made to the source is tracked, along with who made the change, why they made it, and references to problems fixed, or enhancements introduced, by the change.

Version control systems are essential for any form of distributed, collaborative development, as different teams want to design, develop and deploy software in parallel and merge them easily.

Key Terms :

Check in :

Saving your charges into Version Control .

Check Out :

Getting the latest version from Version Control.

Conflict: :

In general the "Version Control" merges your changes with existing Code automatically, and merges them automatically , but sometimes the "Versions Control" detect Conflicts, and the "Version Control" would not be able resolve, it would require a manual intervention to resolve the Conflict.

Merge: :

Combining multiple changes made to different working copies of the same files in the source repository. Merging is a strategy for managing conflicts by letting multiple developers work at the same time (with no locks on files), and then incorporating their work into one combined version.

Trunk:

"Trunk" refers to the unnamed branch (version) of a file tree under revision control. The trunk is usually meant to be the base of a project on which development progresses.

Branching: :

Branching, in revision control and software configuration management, is the duplication of an object under revision control (such as a source code file, or a directory tree) so that modifications can happen in parallel along both branches.

23] What is SVN ?

SVN is a Sourcecode Version Control, you could manage the files and folders in SVN by using the Command Line commands as shown below or You could also use the some SVN clients like SmartSVN or Tortoisesvn.

svn diff:

Shows line-level details of a particular change

svn log:

Shows you broad information: log messages with date and author information attached to revisions and which paths changed in each revision

svn cat:

Retrieves a file as it existed in a particular revision number and displays it on your screen.

svn list:

Displays the files in a directory for any given revision.

24] What is JIRA and DOORs ?

JIRA : JIRA is the project tracker for teams building great software. JIRA sits at the center of your development team, connecting the people and the work being done. Track bugs and tasks, link issues to related source code, plan agile development, monitor activity, report on project status, and more.

DOORS: DOORS, a leading solution for requirements management, provide capabilities including: A collaborative requirements management environment that allows all stakeholders to actively participate in the requirements process.

The ability to manage changing requirements Powerful life cycle traceability to help teams align their efforts with the business needs and measure the impact that changes will have on everything from business goals to development Support for requirements-driven testing with built-in test tracking tool and integrations with Rational Quality Manager and HP Quality Center Integrates with enterprise architecture, product portfolio management, model-driven development, quality management, and change and release management solutions from IBM and third party companies.

25] What is Hudson ? How to use this ?

Hudson is a continuous integration (CI) tool written in Java, which runs in a servlet container, such as Apache Tomcat or the Weblogic application server. It supports SCM tools including CVS, Subversion, Git, Perforce and Clearcase and can execute Apache Ant and Apache Maven based projects, as well as arbitrary shell scripts and Windows batch commands.

26] Explain Jenkins -pipeline highlights ?

Pipeline item type for new jobs (instead of Freestyle)

Entire pipeline as text code in SCM (GitHub)

Multiple SCM repositories in each job

Pausable: Jobs can wait for manual user input before continuing

Jobs share global library to share scripts, functions, variables for DRY (Do not Repeat Yourself) – Reusable components and flow

Extendable DSL with loops, logic

Visualized: Pipeline StageView provides status at-a-glance dashboard and trending

Parallel execution of arbitrary build states

Jobs starting in one agent can switch (be joined) to another (fork/join)

27] What is CI (Continuous integration) & CD (Continuous Delivery) in Jenkins ?

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.

What is the requirement for using **JenkinsCode done->Unit Tests ->Integration->Acceptance Test->Deployment** fully process automatically through jenkins is a part of Continuous delivery or continuous deployment process.

28] 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**

29] Mention some of the useful plugins in Jenkin.

- **Powershell script**
- **Window execute script**
- **Env inject**
- **VSTS plugin**
- **Maven 2 project**
- **Amazon EC2**
- **HTML publisher**
- **Copy artifact**
- **Publish artifact**

30] 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.**

31] How do all these tools work together ?

Developers develop the code and this source code is managed by Version Control System tools like Git etc.

1. Developers send this code to the Git repository and any changes made in the code is committed to this Repository.
2. Jenkins pulls this code from the repository using the Git plugin and build it using tools like Ant or Maven.
3. Configuration management tools like puppet deploys & provisions testing environment and then Jenkins releases this code on the test environment on which testing is done using tools like selenium.
4. Once the code is tested, Jenkins send it for deployment on the production server (even production server is provisioned & maintained by tools like puppet).

32] How to make sure that your project builds does not break in Jenkins ?

You must follow these steps to make sure that your project builds doesn't break in Jenkins:

1. First, perform successful clean install on your local machine with all unit tests.
2. Check all your code changes.
3. Synchronize with repository to make sure that all required config and POM changes and any difference is checked into the repository.

33] Explain some Jenkins Job with example ?

Freestyle Project:

Freestyle build jobs are general-purpose build jobs, which provides maximum flexibility. The freestyle build job is the most flexible and configurable option, and can be used for any type of project. It is relatively straightforward to set up, and many of the options we configure here also appear in other build jobs.

Multiconfiguration Job:

The “multi configuration project” (also referred to as a “matrix project”) allows you run the same

build job on different environments. It is used for testing an application in different environments, with different databases, or even on different build machines.

Monitor an External Job:

The “Monitor an external job” build job lets you keep an eye on non-interactive processes, such as cron jobs.

34] Which SCM tools Jenkins supports ?

Below are Source code management tools supported by Jenkins:

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

35] Explain how you can set up 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..

36] 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:

- **Ensure global security is on.**

- **Ensure that Jenkins is integrated with my company's user directory with appropriate plugin.**
- **Ensure that 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 same.**

37] 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.**

38] What is the relation between Hudson and Jenkins ?

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

39] 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.

40] Explain how you can move or copy Jenkins from one server to another ?

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.

41] What are the various ways in which build can be scheduled in Jenkins ?

By source code management commits

- After completion of other builds
- Can be scheduled to run at specified time (crons)
- Manual Build Requests

42] 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.