**Continuous Integration questions**

**Q1. What is meant by Continuous Integration?**

It is a development practice that requires developers to integrate code into a shared repository several times a day. Each check-in is then verified by an automated build, allowing teams to detect problems early.  
Example:

1. Developers check out code into their private workspaces.
2. When they are done with it they commit the changes to the shared repository (Version Control Repository).
3. The CI server monitors the repository and checks out changes when they occur.
4. The CI server then pulls these changes and builds the system and also runs unit and integration tests.
5. The CI server will now inform the team of the successful build.
6. If the build or tests fails, the CI server will alert the team.
7. The team will try to fix the issue at the earliest opportunity.
8. This process keeps on repeating.

**Q2. Why do you need a Continuous Integration of Dev & Testing?**

Continuous Integration of Dev and Testing improves the quality of software, and reduces the time taken to deliver it, by replacing the traditional practice of testing after completing all development. It allows Dev team to easily detect and locate problems early because developers need to integrate code into a shared repository several times a day (more frequently). Each check-in is then automatically tested.

**Q3. What are the success factors for Continuous Integration?**

* Maintain a code repository
* Automate the build
* Make the build self-testing
* Everyone commits to the baseline every day
* Every commit (to baseline) should be built
* Keep the build fast
* Test in a clone of the production environment
* Make it easy to get the latest deliverables
* Everyone can see the results of the latest build
* Automate deployment

**Q4. 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**.**

**Q5. 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 them 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.

**Q6. Explain how can create a backup and copy files in Jenkins?**

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.

**Q7. Explain how you can setup 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”.  
Then 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...

**Q8. Mention some of the useful plugins in Jenkins.**

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

**Q9. How will you secure Jenkins?**

The way I secure Jenkins is mentioned below. If you have any other way of doing it, please mention it in the comments section below:

* 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.

**Q10. 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 <plugin>.hpi.pinned
* Start Jenkins

**Q11. How you would make software deployable?**

The ability to script the reconfiguration and installation of software systems is essential towards automated and controlled change.  Older products and systems are supposing that the changes would be minor and infrequent and so the automated changes become difficult, even though there is an increasing style for new software to enable this. In order to expose settings and configuration in a way accessible to automation, the professional has  to work with concepts such as  scripted installation, separation of concerns, infrastructure as a code, command – tools, test harnesses, dependency injection and inversion of control.

**Q12.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

**Q13. Explain the working of HTTP?**

You can simply answer .Like other protocols, HTTP also works on the client-server model. A web server software which responds to the request is called a server and a web browser which initiates the request is called a client. HTTP enhances its request and response with the help of intermediates such as tunnel, proxy or gateway. URL helps in allocating the resources that are requested using HTTP. The connection to the application layer port of HTTP is provided by TCP.

**Q14. Which SCM tools Jenkins supports?**

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

**Q15. What is the port number of Jenkins?**

Jenkins by default runs on port 8080

You can change it by editing /etc/sysconfig/Jenkins or by command --httpport=<any free port>

**Q16.** **Mention what are the commands you can use to start Jenkins manually?**

To start Jenkins manually open Console/Command line, then go to your Jenkins installation directory. Over there you can use the below commands:

To start Jenkins: **jenkins.exe start**  
to stop Jenkins: **jenkins.exe stop**  
to restart Jenkins: **jenkins.exe restart**

**Q17.** **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 specified time ( crons )
* Manual Build Requests

**Q18.  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.

**Q19. Explain Jenkins Architecture**

Jenkins uses Master/Slave Architecture. It is distributed architecture.

Role of Master:-

* Schedule build jobs
* Dispatch builds to the slaves for the actual job execution
* Monitor the slaves and build results
* Can also execute build jobs directly.

**Q20. What is Role of slave?**

Execute build jobs dispatched by master

**Q21. What is executor?**

Executor is a separate stream of builds to be run on a node in parallel.

A node can have one or more executors

**Q22. What is plugins?**

A plugin, like plugins on any other system, is a piece of software that extends the core functionality of the core Jenkins server.

**Q23. Explain a brief overview of Manage Jenkins section?**

It contains

* Configure system. Configure global settings and path like java, maven path.
* Configure Global Security. Secure Jenkins, define who is allowed to access/use the system.
* Manage Plugins. Helps to add, disable, and enable plugins that can extend the functionality of Jenkins.
* System Information. Displays various environment information to assist trouble shooting.
* System Logs. It captures output from java.util.logging output related to Jenkins.

**Q24. How to automatically build in Jenkins?**

Using Poll SCM which makes sure that whenever a code is committed it will pull from GitHub.  
19) What are the different build triggers?

->Poll SCM

->Build Periodically

->Build after other projects are built

->Build when a change is pushed to GitHub