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

Since version 2.54, Jenkins requires an installation of the Java Development Kit (JDK). JAVA_HOME should also be configured prior to installation. Version 1.8 of the JDK is the minimum that Jenkins will support.

Jenkins also requires the services of a Jakarta Enterprise Edition (EE) web profile-compliant servlet engine in order to run, although Jenkins comes with an embedded Jetty runtime that can be used if an existing Tomcat, WildFly or WebSphere application server is not available.

2. What is the syntax Jenkins uses to schedule items such as build jobs and SVN polling?

Jenkins uses the cron syntax to schedule various items within the tool.

The cron syntax is represented by five asterisks, with each one separated by a space. The first asterisk represents *minutes*, the second represents *hours*, the third the *day of the month*, the fourth the *month itself* and the fifth the *day of the week*. For example, to schedule a build job to pull from GitHub every Friday at 5:30 p.m., the syntax would be: 30 17 * * 4.

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

There are a number of 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.

4. Name three security mechanisms Jenkins uses to authenticate users.

Jenkins can authenticate users in one of three ways:

1. Jenkins can use an internal database to store user data and credentials. (This is the default.)

2. Jenkins can be configured to authenticate against a Lightweight Directory Access Protocol server.
3. Jenkins can be configured to employ the authentication mechanism used by the application server upon which it is deployed.

5. Describe the standard process to configure and use third-party tools within Jenkins?

The process to use a third-party tool, such as Artifactory, Node, SonarQube or Git typically follows a four-step process.

1. The third-party software must be installed.
2. A Jenkins plug-in that supports the third-party tool must be installed through the Jenkins admin console.
3. The third-party tool must be configured in the Tools tab of the Manage Jenkins section of the admin console.
4. Finally, the plug-in can be used from within a Jenkins build job. The plug-in will then facilitate communication between the Jenkins build job and the third-party tool.

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

The tool provides two mechanisms for starting a Jenkins node agent:

1. Launch a Jenkins node agent from a browser window.

2. Launch a Jenkins node agent from the command line.

When a Jenkins node agent is launched from a browser, a JNLP file is downloaded. When it runs, the JNLP file launches a new process on the client machine that runs Jenkins jobs.

To launch from the command line, the *agent.jar* file is required on the client. This executable JAR file is run from the command line, along with a reference to the slave agent's JNLP file that is hosted on the server.

7. How do you take a backup of your Jenkins build jobs in order to prepare for disaster recovery?

Each Jenkins build stores its configuration as XML in a subdirectory of the *JENKINS_HOME\jobs* folder. By copying this folder to a secondary location, the configuration of all of the build jobs managed by the Jenkins master will be backed up as a result. Checking this folder into a source code management tool like Git isn't a bad idea either. By simply copying the contents of this folder to a new Jenkins server instance, all of the build jobs described in this folder will be restored the next time the Jenkins server is started.

It's also worth noting that a history of all build jobs that have been run is written to this folder as well. To really impress the

interviewer asking this advanced Jenkins interview question for DevOps, mention that using a *.gitignore* file and ignoring job-related files will ensure that any Git repository that stores Jenkins project configuration data will make sure only configuration data is stored. Build job history should be archived elsewhere and not stored in a source code repository.

8. Name three steps or stages a typical Jenkins pipeline might include.

A full-blown Jenkins pipeline will build a project from source code, put it through a variety of unit, integration, performance and user acceptance tests, and then, finally, if every test succeeds, deploy a packaged application to an application server, Nexus repository or Docker container. So, three fundamental stages would be:

1. Build
2. Test
3. Deploy

9. 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 security is enabled, this file contains an XML element named useSecurity that will be set to true. By changing this

setting to false, security will be disabled the next time Jenkins is restarted.

```
<useSecurity>false</useSecurity>
```

10. Polling a Git repository for new commits is considered a Jenkins anti-pattern. What is a sound alternative to SVN polling?

Constantly polling a source code management tool like Git or Subversion to check if a new commit has been issued is a waste of clock cycles and should be avoided.

A better approach is to reverse this process and have the source code tool trigger a Jenkins build when new commits happen. With GitHub or GitLab, it is relatively easy to configure a post-commit hook that runs every time a commit is successful. When provided with the URL of the Jenkins build, the post-commit hook can easily trigger a Jenkins build, eliminating the need to have Jenkins constantly poll the source code repository.

11.Which commands can be used to begin Jenkins?

Commands that can be used to start Jenkins are:

- 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
```

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

- 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 problems.
- Continuous Integration triggers the build to find and identify bugs present in the code.

13.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.

14.Name some of the useful plugins in Jenkins.

Some of the plugins in Jenkins include:

- Maven 2 project
- Amazon 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.

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

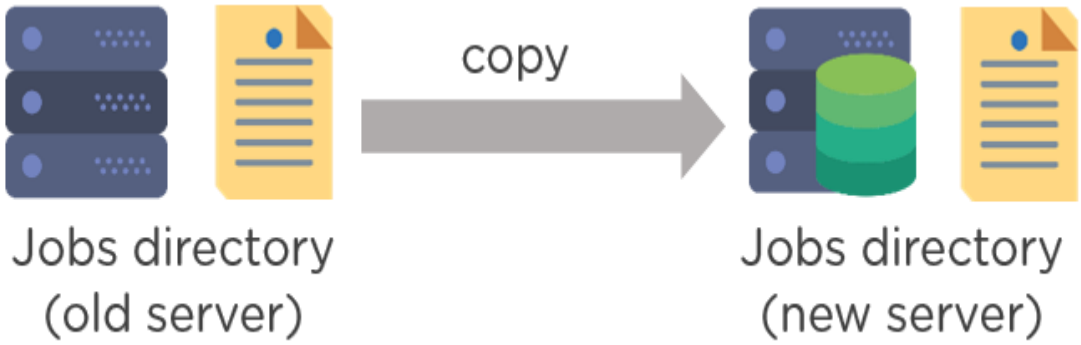
If you wish to deploy a custom build of a core plugin, you may follow the following steps:

- 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?

In order to move or copy Jenkins from one server to another, there may be multiple ways:

- 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. 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.'

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

- 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.

20. 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

21. 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.

22. 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)

23. 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.

24. 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.

25. 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.

26. 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.

27. 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

28.What are the possible use cases for Jenkins?

Once you are sure on the definition of Jenkins, let us discuss the use cases mentioned below –

- 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 builds 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.

29). How will you define the Jenkins agent?

An agent in Jenkins describes the complete pipeline or a particular stage where execution will take place or where the agent is located. This section is available at the higher level inside the pipeline block, but the usage of stage-level is generally optional.

30). How can you define the agent directives in Jenkins?

The objective of agent directive in Jenkins is to check how or where the pipeline can be executed. Most of the times, the agent is needed for all possible pipelines

31). How will you define the Post in Jenkins?

The post is one more additional step that will run upon the completion of a pipeline block. The post can support a plenty of post-conditional blocks and makes execution much easier and logical in DevOps.

32). Where the bulk work can be described in the Jenkins?

When one or more stage directives are defined together, the stage is the place where bulk work can be described in detail. Make sure that for each stage directive, there is one stage defined properly to manage the discrete parts of the continuous delivery process like deploy, manage, or build etc.

33). How can you define the environment directive in Jenkins?

With the help of environmental directives, we can specify a complete sequence of the key-value pairs that could be defined as the environment variables too for different steps where the environment directive is located within the pipeline.

34). Define the triggers in Jenkins.

To re-trigger the pipeline in multiple automated ways, triggers are defined in the Jenkins. A few pipelines are integrated with sources like GitHub, BitBucket, or other triggers first then they are implemented to perform a specific action.

35). What do you mean by the input directives in Jenkins?

With the help of input directives, you could prompt the inputs with the help of input steps. The stage is paused after different options have been applied and before you enter to the stage agent to evaluate its condition. When the input is approved, you could continue with the stage ahead. The parameters that given as the part of input submission will be available in the environment for the rest of the stage.

36). How can you define the Control Flows in Pipeline?

When scripted pipelines are executed in a sequential order from top to the downwards then it is called as control flows in the pipeline.

AFTER INTERVIEW

37) What is Jenkinsfile?

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

38)What are Declarative Pipelines in Jenkins?

Answer: Declarative Pipelines are the newest additions to Jenkins that simplify the groovy syntax of Jenkins pipelines (top-level pipeline) with some exceptions, such as:

No semicolon to be used as a statement separator. The top-level pipeline should be enclosed within block viz;

The common syntax is:

```
pipeline {

/* Declarative Pipeline */

}
```

39) How to make sure that your project build does not break in Jenkins?

Answer: You need to follow the below-mentioned steps to make sure that the Project build does not break:

1. Clean and successful installation of Jenkins on your local machine with all unit tests.
2. All code changes are reflected successfully.
3. Checking for repository synchronization to make sure that all the differences and changes related to config and other settings are saved in the repository.

40)w can you secure Jenkins?

Answer: Securing Jenkins is a little lengthy process, and there are two aspects of securing Jenkins:

(i) Access Control which includes authenticating users and giving them an appropriate set of permissions, which can be done in 2 ways.

- **Security Realm** determines a user or a group of users with their passwords.

- **Authorization Strategy** defines what should be accessible to which user. In this case, there might be different types of security based on the permissions granted to the user such as Quick and simple security with easy setup, Standard security setup, Apache front-end security, etc.

(ii) Protecting Jenkins users from outside threats.

41)What is the solution if you find a broken build for your project?

Answer: To resolve the broken build follow the below-mentioned steps:

- Open console output for the build and check if any file change has missed.

OR

- Clean and update your local workspace to replicate the problem on the local system and try to resolve it (In case you couldn't find out the issue in the console output).

42) What are the basic requirements for installing Jenkins?

Answer: For installing Jenkins you need the following system configuration:

1. Java 7 or above.
2. Servlet 3.1
3. RAM ranging from 200 MB to 70+ GB depending on the project build needs.
4. 2 MB or more of memory.

43)Why is Jenkins called a Continuous Delivery Tool?

Answer: We have seen the Continuous Delivery workflow in the p1revious question, now let's see the step by step process of why Jenkins is being called as a Continuous Delivery Tool:

1. Developers work on their local environment for making changes in the source code and push it into the code repository.
2. When a change is detected, Jenkins performs several tests and code standards to check whether the changes are good to deploy or not.
3. Upon a successful build, it is being viewed by the developers.
4. Then the change is deployed **manually** on a staging environment where the client can have a look at it.
5. When all the changes get approved by the developers, testers, and clients, the final outcome is saved **manually** on the production server to be used by the end-users of the product.

44)How to restart Jenkins manually?

Ans: To restart Jenkins manually, you can use any one of the following commands:

- (jenkins_url)/safe restart - Allows all running jobs to complete. New jobs will remain in the queue to run after the restart is complete.

- (jenkins_url)/restart - Forces a restart without waiting for builds to complete.

45)What is the difference between Jenkins, Maven, and Ant?

Jenkins	**Maven**	**Ant**
Continuous Integration Tool	Build automation tool	Command Line/Java Library Tool.
Automates software development process through continuous integration and facilitates continuous delivery.	Describes software dependencies and explains how the software is built	Drives build process
Supports version control tools like Git, AccuRev.	Supports projects written in C#, Ruby.	Supports projects written in C and

		C++.

46) How can you deploy a custom build of a core plugin?

Ans: To deploy a custom build of a core plugin, you have to do the following:

- 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.

47) What are the Parameters in Jenkins?

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

48) What is the use of the Agent Directive?

Ans: The agent directive specifies Jenkins how and where to execute the entire pipeline. The directive is specified at the top-level inside the pipeline block and stage-level usage is optional.

49) What is flow control in Jenkins?

Ans: Flow control supports the pipeline structure (scripted pipeline) for executing the top to bottom of the Jenkins file.

50) What are the ways to configure Jenkins node agent to communicate with Jenkins master?

Ans: Below mentioned ways help you to start the node agent –

- Command-line – For this, the client needs the executable agent.jar file. It helps to launch the process on the client to communicate with the Jenkins master to run build jobs.
- Browser – This file launches a new process on the client machine to run jobs.

51): What is a Jenkins Pipeline?

Ans: Jenkins Pipeline is a set of features of Jenkins, which are installed as plugins to enable continuous delivery pipeline implementation. These are the automated processes to get the software from source control through deployment to end-users.

52): What is the use of the JENKINS HOME directory?

Ans: JENKINS_HOME directory is used to store all the settings, configurations, and logs

53)What is Groovy in Jenkins?

Ans: Apache Groovy is the default scripting language used for the JVM platform. It has some useful features like DSL support, dynamic typing, and closures.

Some of the major features of using Groovy are:

- Groovy is a dynamic and agile language, offers seamless integration with all the existing Java libraries and objects.
- We can use scopes to define Collections or Arrays.
- Allows you to add or remove collections

54)What are the ways by which a build can be scheduled in Jenkins?

Ans: 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 (crowns)
- Manual Build Requests

55)How will you secure Jenkins?

Ans: The following ways will help you to secure Jenkins:

- Check the global security is on.
- Make sure Jenkins is integrated with the appropriate login with my company's user directory.
- The project matrix/matrix is allowed to fine-tune access.
- Automate the process of setting rights/privileges with custom version controlled script in Jenkins.

- Limit physical access to Jenkins data/folders.
- Run security audits periodically on the same.

56)What are declarative pipelines?

Ans: A declarative pipeline is a new feature in Jenkins that maintains the pipeline as a code and makes it easier to write and read. It is defined within a block labeled pipeline.

Syntax:

The common syntax is:

```
1 pipeline {
2 /* Declarative Pipeline */
3 }
```

Structure of the declarative pipeline:

- any – It represents the whole pipeline will run on any available agent.
- docker – This is to run the pipeline in the Docker environment.
- none – It indicates all the stages under the block will have to be declared with the agent separately.
- label – This is just a label for the Jenkins environment

57)What is DSL Jenkins?

- **Ans:** Job DSL is one of the most important plugins for Jenkins that allows managing configuration as code. It is made up of two parts. First, Domain Specific Language (DSL) that enables users to define jobs utilizing a Groovy-

based language, and second, a Jenkins plugin that manages the scripts and updates the created Jenkins jobs.

58.What do you understand by the term Parallel in Jenkins?

In case of the declarative Pipeline, a plenty of nested stages can be defined together and they are executed in parallel. The parallel stages generally don't contain any agent or tools because they are not relevant here.

59.Why do we use Jenkins with selenium?

Answer: Running Selenium tests in Jenkins will allow to run tests every time software changes and deploy the software to a new environment when the tests will pass. Jenkins will schedule tests to run at specific time.

What is build pipeline in Jenkins?

Answer: In **Jenkins**, Job chaining is the process of automatically starting other job(s) after the execution of a job. This approach will build **multi-step build pipelines** or trigger the rebuild of a project, if one of its dependencies will be updated.

60. What is a Jenkins pipeline?

Answer: The **Jenkins Pipeline plugin** is based on a Domain Specific Language (DSL) in Groovy, the Pipeline plugin will make pipelines scriptable. This is an incredibly powerful way for developing complex, multi-step **DevOps pipelines**.

61.What is the tool used for provisioning and configuration?

Answer: Ansible is an agent-less configuration management and orchestration tool. In Ansible, the configuration modules will be called as Playbooks. Like other tools, Ansible will be used to cloud provision.

62.What is blue ocean in Jenkins?

Answer: Blue Ocean is a project which will model and present the process of software delivery by surfacing information which is important for development teams with as few clicks as possible, while still will be staying true to the extensibility which is core to Jenkins.

63.How do you configuring automatic builds in Jenkins?

Answer: Builds in Jenkins will be triggered periodically on a schedule. It is specified in configuration, or when source will change in the project have been detected, or they will be automatically triggered by requesting the URL: http://YOURHOST/jenkins/job/PROJECTNAME/build

64.What are the advantages of Jenkins?

Answer: Following are some advantages of using Jenkins or by any matter any integration tools:

1. it saves developer time: Most of the integration task is being handled by Jenkins via automatic integration; the developer time

will be focused on development activities mostly.
2. Improved software quality: Since the software is can be tested immediately after any code check-in, it will keep the quality check frequently, thus improves overall software quality.
3. Faster Delivery: Jenkins automatically perform continuous integration, which will lead to very early detection of bugs / defects and hence it will lead to faster delivery of software.
4. Decreased development time: Since most of the integration work will be automated by Jenkins, it will lead to the faster development of application.
5. Easily portable: Since Jenkins will be developed using Java, it will easily portable to other platforms.

65.Explain role-based strategy plugin?

Answer: We will create 'Global roles', 'Project roles' and 'Slave roles' and assigning roles to users by using role-based strategy,
1. **Global roles:** admin, Job creator, anonymous, etc. will allow to set Overall, slave, job, Rn, View and SCM permissions on a global basis.
2. **Project roles:** Allow to set only Job and Run permissions on a project basis.
3. **Slave roles:** Allow to set node-related permissions.

66.How your jenkins come to know which version to choose when two versions of Java are configured?

Answer: When we will start build jenkins it is asked to choose on which version of Java the build/job has to run.

67.Which plugin is required to deploy a .war file into application server?

Answer: 'Deploy to Container Plugin' is required to deploy a.war file into application server.

68.What is upstream and downstream projects in jenkins?

Answer: Nothing but dependency of jobs.

- Upstream: if we will set job1 is upstream for job2. Then job1 will be built to build job2.
- Downstream: if we will set job2 is downstream in job1. Then if we can build job1 then automatically job2 can also be build.

> In 'Build triggers' section we will set the upstream projects.
> In 'Post-build' section we will set downstream projects

69.How to re-execute a parameterized build without entering the parameter value when the job fails?

Answer: 'Rebuild plug-in' will allow the user to rebuild a parametrized build without entering the parameters again.

70. How can you pass parameters from one job to another job?

Answer: When we will create new/existing job go to –> General tab –
–> enable 'This project will be parameterized' –> select 'Run

parameter' from drop down –> in project tab select the project name (from which project it will take the parameters).

71.How master system will communicate to slave system?

Answer: using 'slave.jar' file in slave system master system will communicate to slave system. This file must be present in slave to communicate with.

72. How do you make a job to run only on slave node?

Answer: Go to job configuration tab –> under 'General' section enable 'Restrict where this project will be run' then we will provide the lable name of the node.

73How do you make all the jobs to be run only on slave node?

Answer: Go to –>'Manage jenkins' –> 'configure system' –> at 'Labels' mention the node system label.

74.What is Quite period in Jenkins?

Answer: Time gap between the builds. In seconds.

What is SCM Checkout retry count?

Answer: It can check the code in SCM tool twice in between the builds, if the value is 2.

75.How do you delete old builds?

Answer: Delete old builds in following way
go to –> General tab –> enable 'Discard old Builds' checkbox. Here we will set 'Days to keep builds' and 'Max # of builds to keep'

76.What is difference between Build periodically and Poll SCM?

Answer: Build periodically – It can trigger builds as per the schedule, even if we haven't changed anything.
Answer: Poll SCM – It can check for changes before triggering any build, if there will be any changes to the previous version than only build can be triggered. Otherwise it will not the build.

77.How will you secure Jenkins?

Answer: Following way Jenkins can be secured:
1. Ensure global security is on.
2. Ensure that matrix based is enabled to fine tune access.
3. Automate the process of setting rights/privileges in Jenkins with custom version-controlled script.
4. Limit physical access to Jenkins data/folders.
5. Periodically run security audits on same.

78.How will you give the customized/custom build numbers in jenkins?

Answer: By using Next_Build_Number plugin, we will change the build number of a jenkins.

- We will only change the build numbers for failed builds, not for success builds.

79. Why we use Nodes in jenkins?

Answer: 1. Single jenkins server will not handle the whole workload of large and heavier build projects.

2. Sometimes we may required several different environments to test the builds. This will not be done by a single server.

80.How do you send the notifications whether build fails or success?

Answer: Method to send the notifications whether build fails or success

1. Configure Email Notification in Configure system under Mange Jenkins.

2. Email notifications will be configured in job configuration under post-build section.