

Top Jenkins Interview Questions

Enlisted below are some of the most popular Jenkins Interview Questions that have been answered by experts.

Q #1) What is Jenkins?

Answer: Jenkins is a free open source Continuous Integration tool and automation server to monitor continuous integration and delivery. It is written in Java.

It is known as an automated Continuous Delivery tool that helps to build and test the software system with easy integration of changes to the system. Jenkins follows Groovy Scripting.

Also, it enables developers to continuously check in their code and also analyze the post-build actions. The automation testers can use to run their tests as soon as the new code is added or code is modified.

Q #2) What are the features of Jenkins?

Answer: Jenkins comes with the following features:

1. Free open source.
2. Easy installation on various operating systems.
3. Build Pipeline Support.
4. Workflow Plugin.
5. Test harness built around JUnit.
6. Easy upgrades.
7. Rapid release cycle.
8. Easy configuration setup.
9. Extensible with the use of third-party plugins.

Q #3) What are the advantages of Jenkins? Why we use Jenkins?

Answer: Jenkins is used to continuously monitor the large code base in real-time. It enables developers to find bugs in their code and fix them. Email notifications are made to the developers regarding their check-ins as a post-build action.

Advantages of Jenkins are as follows:

- Build failures are cached during the integration stage.
- Notifies the developers about build report status using LDAP (Lightweight Directory Access Protocol) mail server.
- Maven release project is automated with simple steps.
- Easy bug tracking.
- Automatic changes get updated in the build report with notification.
- Supports Continuous Integration in agile development and test-driven development.

Q #4) Mention some of the important plugins in Jenkins?

Answer: Plugins in Jenkins includes:

- Gits
- Maven 2 Project
- HTML Publisher
- Copy Artcraft
- Join

- Green Balls
- Amazon EC2

Q #5) What is Continuous Integration in Jenkins?

Answer: Continuous integration is the process of continuously checking-in the developer's code into a version control system and triggering the build to check and identify bugs in the written code.

This is a very quick process and also gives them a chance to fix the bugs. Jenkins is one such continuous integration tool.

In software development, multiple developers work on different software modules. While performing integration testing all the modules are being integrated together. It is considered as the development practice to integrate the code into the source repository

Whenever the programmer/developer makes any change to the current code, then it automatically

gets integrated with the system running on the tester's machine and makes the testing task easy and speedy for the system testers.

Continuous Integration comprises of:

- Development and Compilation
- Database Integration
- Unit Testing
- Production Deployment
- Code Labeling
- Functional Testing
- Generating and Analyzing Reports

Q #6) What is the difference between Hudson and Jenkins?

Answer: There is no difference between Hudson and Jenkins. Hudson was the former name of Jenkins, after going through several issues the name was changed to Jenkins.

Q #7) What is Groovy in Jenkins?

Answer: Groovy is the default scripting language that is being used in the development of JMeter Version 3.1.

Currently Apache Groovy is the dynamic object-oriented programming language that is used as a scripting language for the Java platform. Apache Groovy comes with some useful features such as Java Compatibility and Development Support.

Q #8) Which command is used to start Jenkins?

Answer: You can follow the below-mentioned steps to start Jenkins:

1. Open Command Prompt
2. From the Command Prompt browse the directory where Jenkins. war resides
3. Run the command given below:

D:\>Java -jar Jenkins.war

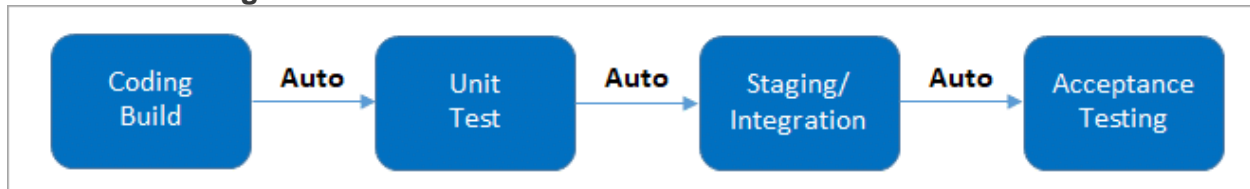
Q #9) What is Jenkinsfile?

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

Q #10) What is the difference between Continuous Integration, Continuous Delivery, and Continuous Deployment?

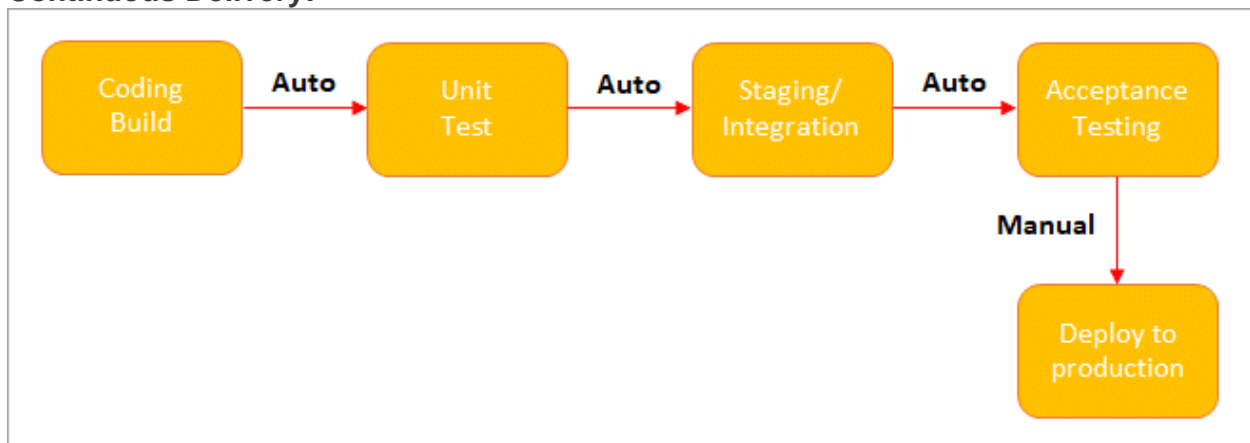
Answer: The diagrammatic representation given below can elaborate on the differences between Continuous Integration, Continuous Delivery, and Continuous Deployment more precisely.

Continuous Integration:



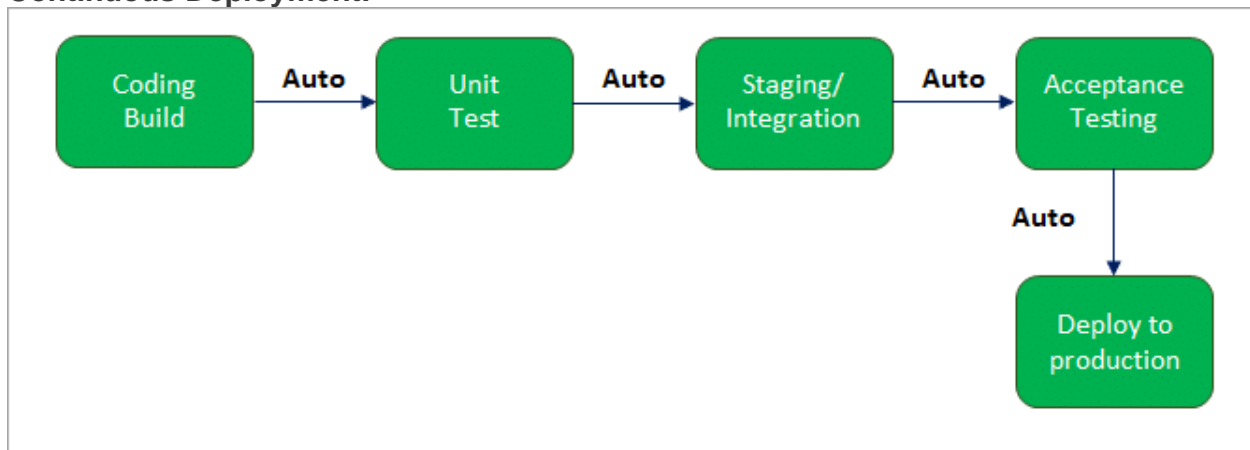
(It involves keeping the latest copy of the source code at a commonly shared hub where all the developers can check to fetch out the latest change in order to avoid conflict.)

Continuous Delivery:



(Manual Deployment to Production. It does not involve every change to be deployed.)

Continuous Deployment:



(Automated Deployment to Production. Involves every change to be deployed automatically.)

Q #11) What is Jenkins Pipeline? What is a CI CD pipeline?

Answer: The pipeline can be defined as the suite of plugins supporting the implementation and integration of continuous delivery pipelines in Jenkins.

Continuous integration or continuous delivery pipeline consists of build, deploy, test, release pipeline. The pipeline feature saves a lot of time and error in maintaining the builds. Basically, a pipeline is a group of build jobs that are chained and integrated in sequence.

Q #12) What are Scripted Pipelines in Jenkins?

Answer: Scripted Pipeline follows Groovy Syntax as given below:

```
Node {  
  
    }
```

In the above syntax, the **node** is a part of the Jenkins distributed mode architecture, where there are two types of node, **Master** which handle all the tasks in the development environment and the **Agent** is being used to handle multiple tasks individually.

Q #13) 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 */  
  
}
```

Blocks must contain Sections, Directives, steps or assignments.

```
pipeline {  
  
    agent any  
  
    stages {  
  
        stage('Build') {  
  
            steps {  
  
                // Statements...  
  
            }  
  
        }  
  
    }  
  
}
```

```

stage ('Test') {

    steps {

        // Statements...

    }

}

}

```

The above code has 3 major elements

- **Pipeline:** The block of script contents.
- **Agent:** Defines where the pipeline will start running from.
- **Stage:** The pipelines contain several steps enclosed in the block called Stage.

Q #14) What is SCM? Which SCM tools are supported in Jenkins?

Answer:

- SCM stands for Source Control Management.
- SCM module specifies the source code location.
- The entry point to SCM is being specified as ***jenkins_jobs.scm***.
- The job specified with 'scm' attribute accepts multiple numbers of SCM definitions.

The SCM can be defined as:

scm:

name: eloc – scm

scm:

git:

url: ssh://Jenkins.org/eloc.git

Jenkins supported SCM tools include:

- CVS
- Git
- Perforce
- AccuRev
- Subversion
- Clearcase

- RTC
- Mercurial

Q #15) Which CI Tools are used in Jenkin?

Answer: Jenkins supported the following CI tools:

1. Jenkins
2. GitLab CI
3. Travis CI
4. CircleCI
5. Codeship
6. Go CD
7. TeamCity
8. Bamboo

Q #16) Which commands can be used to start Jenkins manually?

Answer: You can use the following commands to start Jenkins manually:

1. **(Jenkins_url)/restart:** To force restart without waiting for build completion.
2. **(Jenkin_url)/safeRestart:** Waits until all the build gets completed before restarting.

Q #17) Which Environmental Directives are used in Jenkins?

Answer: Environmental Directives is the sequence that specifies pairs of the key-values called Environmental Variables for the steps in the pipeline.

Q #18) What are Triggers?

Answer: Trigger in Jenkins defines the way in which the pipeline should be executed frequently. PollSCM, Cron, etc are the currently available Triggers.

Q #19) What is Agent Directive in Jenkins?

Answer: The Agent is the section that specifies the execution point for the entire pipeline or any specific stage in the pipeline. This section is being specified at the top-level inside the pipeline block.

Q #20) 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.

Q #21) What is the difference between Maven, Ant, and Jenkins?

Answer: Maven vs Jenkins:

Maven is a build tool like Ant. It consists of a pom.xml file which is specified in Jenkins to run the code. Whereas, Jenkins is used as a continuous integration tool and automates the deployment process. The reports of the builds can be used to set a mark for continuous delivery as well.

The below table enlists the differences between Maven, Ant, and Jenkins in a comparative way:

Maven	Ant	Jenkins
It is a Build Automation Tool.	Java Library/Command Line Tool.	Continuous Integration Tool.
Defines how the software is built and describes the software dependencies.	Drives build process.	Automates the software development process with Continuous Integration and facilitates Continuous Delivery.
Supports projects written in C#, Ruby.	Supports projects written in C and C++.	Supports version control tools like Git, AccuRev.
Executes Unit Tests as a part of the normal build cycle.	Supports single file execution introduced with Java 11.	Can execute Apache Ant and Apache Maven.

Q #22) How will you define Post in Jenkins?

Answer: **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**.

Q #23) What are Parameters in Jenkins?

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

Q #24) How you can set up a Jenkins job?

Answer: **Setting up a new job in Jenkins is elaborated below with snapshots:**

Step 1: Go to the Jenkins Dashboard and log in with your registered login credentials.

← → ↻

 **Jenkins**

Jenkins ▶

User:

Password:

☐ Remember me on this computer

log in


Step 2: Click on the **New Item** that is shown in the left panel of the page.


← → ↻ localhost:8080


Jenkins


search


Jenkins ▶

 [New Item](#)

 [People](#)

 [Build History](#)

 [Manage Jenkins](#)

 [My Views](#)

 [Credentials](#)

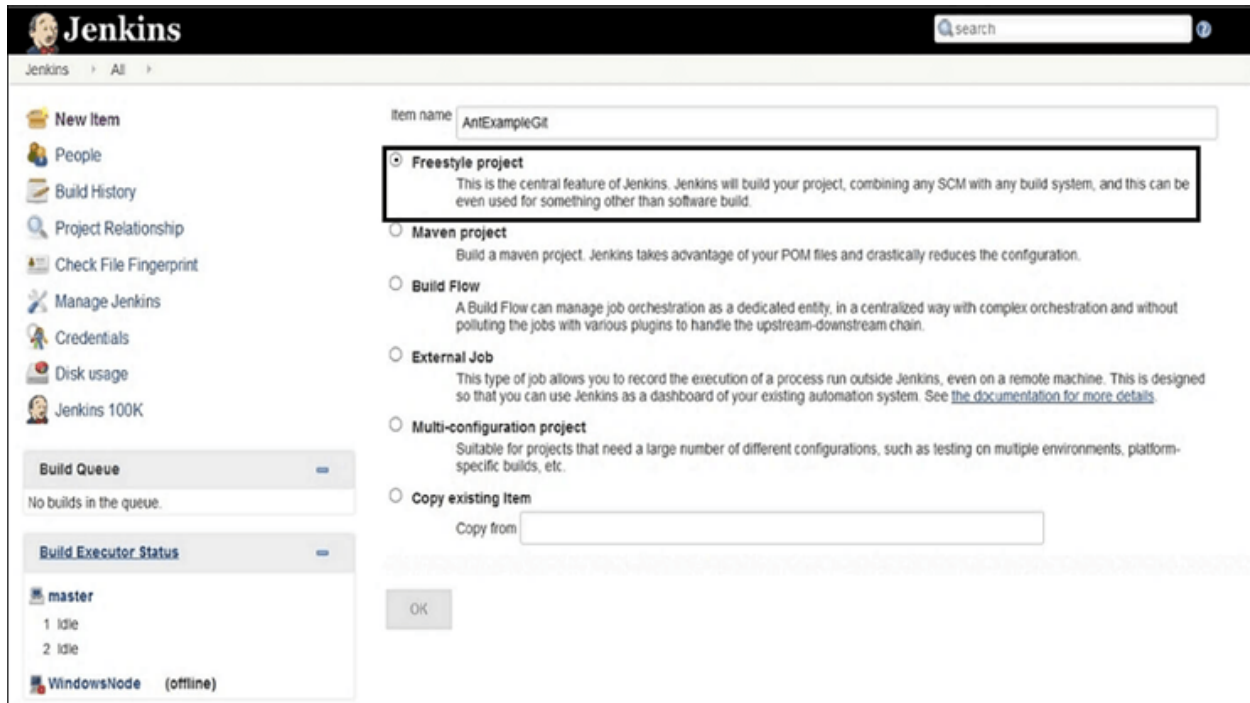
Build Queue

No builds in the queue.

Build Executor Status

1	Idle
2	Idle

Step 3: Click on the **Freestyle Project** from the given list on the upcoming page and specify the item name in the text box.



The image shows the Jenkins 'New Item' configuration page. The 'Item name' field is set to 'AntExampleGit'. The 'Freestyle project' option is selected and highlighted with a red box. Other options include 'Maven project', 'Build Flow', 'External Job', 'Multi-configuration project', and 'Copy existing item'. The left sidebar shows various Jenkins links like 'New Item', 'People', 'Build History', etc. The bottom left shows the 'Build Queue' and 'Build Executor Status' sections.

Jenkins

Item name:

☒ **Freestyle project**
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

☐ **Maven project**
Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.

☐ **Build Flow**
A Build Flow can manage job orchestration as a dedicated entity, in a centralized way with complex orchestration and without polluting the jobs with various plugins to handle the upstream-downstream chain.

☐ **External Job**
This type of job allows you to record the execution of a process run outside Jenkins, even on a remote machine. This is designed so that you can use Jenkins as a dashboard of your existing automation system. [See the documentation for more details](#)

☐ **Multi-configuration project**
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

☐ **Copy existing item**
Copy from:

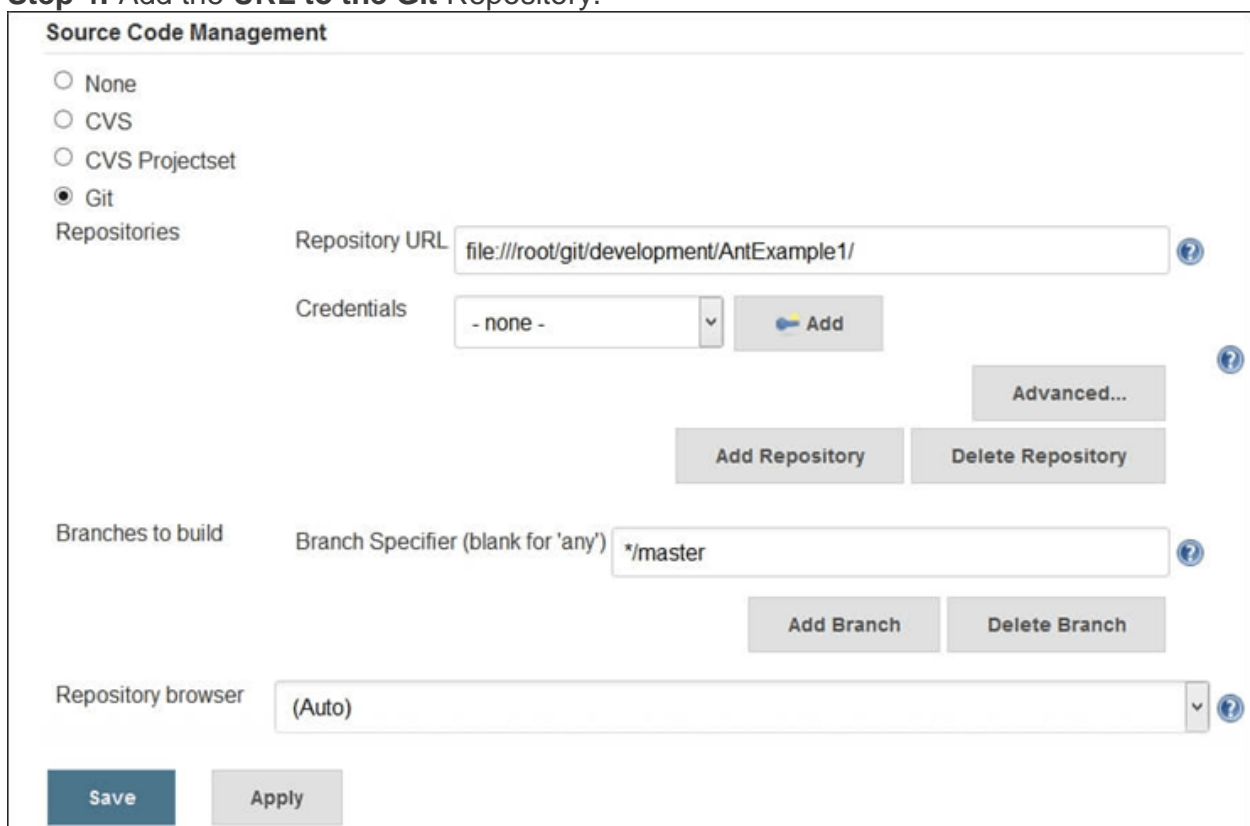
Build Queue
No builds in the queue.

Build Executor Status

master
1 Idle
2 Idle

WindowsNode (offline)

Step 4: Add the URL to the Git Repository.



The image shows the 'Source Code Management' section of the Jenkins configuration page. The 'Git' option is selected under 'None', 'CVS', 'CVS Projectset', and 'Git'. The 'Repository URL' field is set to 'file:///root/git/development/AntExample1/'. The 'Credentials' dropdown is set to '- none -'. The 'Branches to build' section has a 'Branch Specifier (blank for 'any')' set to '*/master'. The 'Repository browser' dropdown is set to '(Auto)'. There are buttons for 'Add Repository', 'Delete Repository', 'Add Branch', and 'Delete Branch'. At the bottom, there are 'Save' and 'Apply' buttons.

Source Code Management

☐ None
☐ CVS
☐ CVS Projectset
☒ **Git**

Repositories

Repository URL:

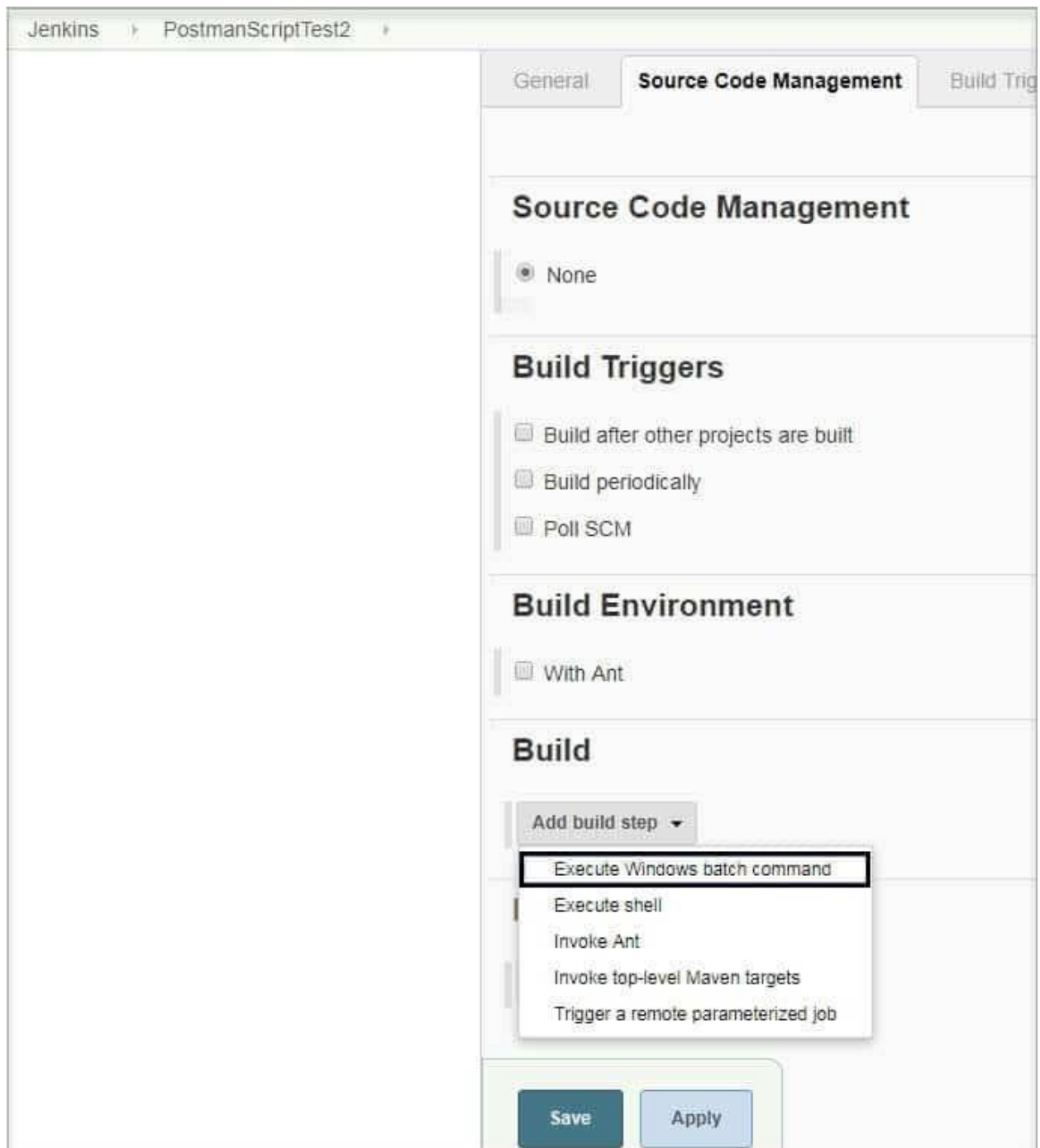
Credentials:

Branches to build

Branch Specifier (blank for 'any'):

Repository browser:

Step 5: Go to the Build section and click on the Add build step => Execute Windows batch command.



Step 6: Enter the command in the **command window** as shown below.

Step 7: After saving all the settings and changes click on **Build Now**.

Step 8: To see the status of the build click on **Console Output**.

Q #25) What are the two components (pre-requisites) that Jenkins is mainly integrated with?

Answer: Jenkins integrates with:

1. Build tools/ Build working script like Maven script.
2. Version control system/Accessible source code repository like Git repository.

Q #26) How can You Clone a Git Repository via Jenkins?

Answer: To create a clone repository via Jenkins you need to use your login credentials in the Jenkins System.

To achieve the same you need to enter the Jenkins job directory and execute the **git config** command.

Q #27) How 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.

Q #28) How to create a backup and copy files in Jenkins?

Answer: In Jenkins, all the settings, build logs and configurations are stored in the JENKINS_HOME directory. Whenever you want to create a backup of your Jenkins you can back up **JENKINS_HOME** directory frequently.

It consists of all the job configurations and slave node configurations. Hence, regularly copying this directory allows us to keep a backup of Jenkins.

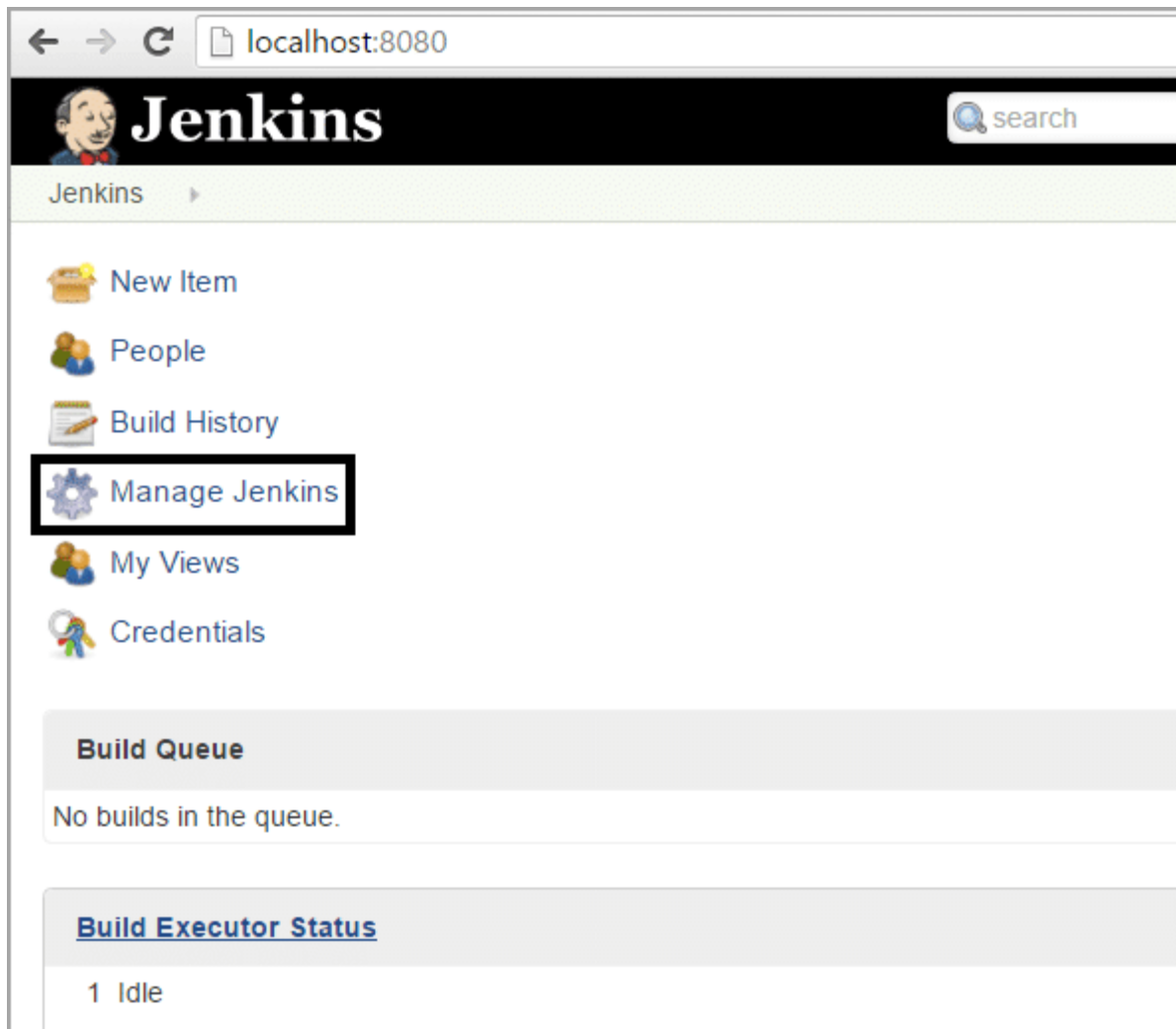
You can maintain a separate backfile and copy it whenever you need the same. If you want to copy the Jenkins job, then you can do so by simply replicating the job directory.

Q #29) What is the use of Backup Plugin in Jenkins? How to use it?

Answer: Jenkins Backup Plugin is used to back up the critical configurations and settings in order to use them in the future in case of any failure or as per the need of time.

The following steps are followed to back up your settings by using the Backup Plugin.

Step 1: Go to the Jenkins Dashboard and click on **Manage Jenkins**.

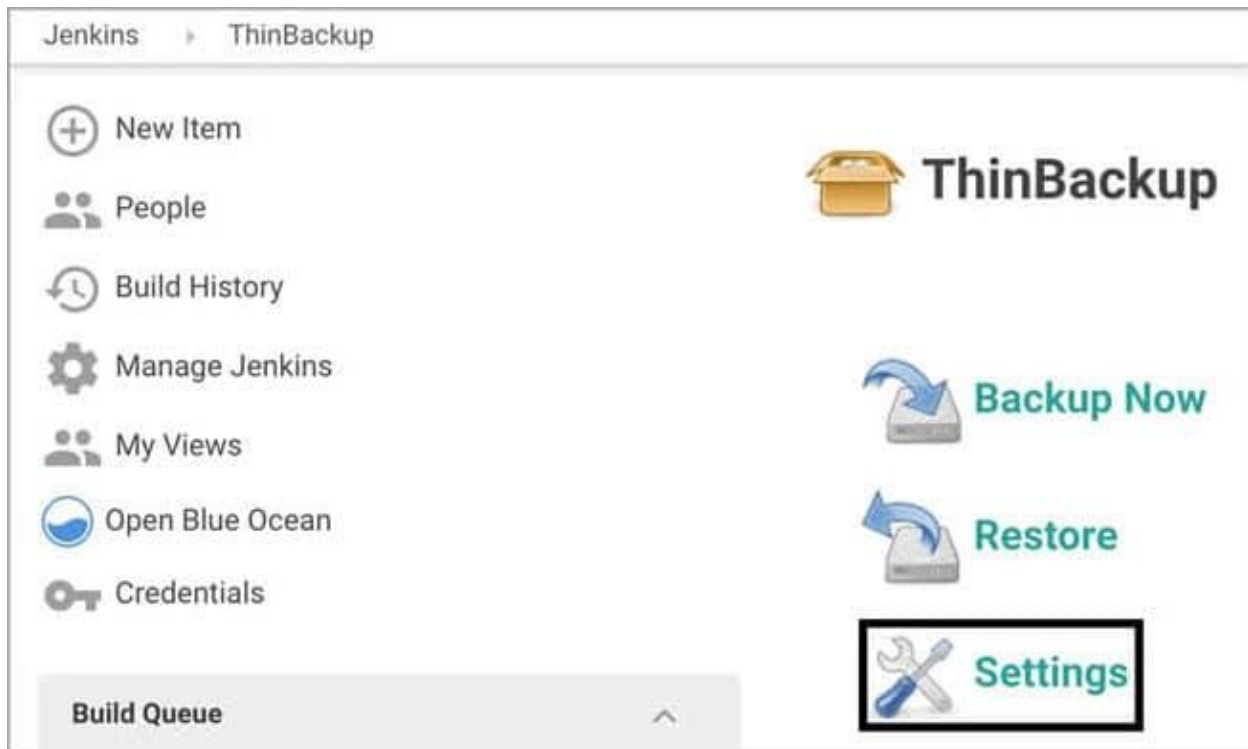


Step 2: Click on **Manage Plugins** that appears on the next page.

Step 3: Go to **Available Tab** on the next page and search for **ThinBackup**.

Step 4: Once you choose the available option, it will start installing.

Step 5: Once it is installed the following screen will appear, from there choose **Settings**.



Step 6: Enter the necessary details like backup directory along with other options as shown on the below screen and save the settings. The backup will be saved to the specified **Backup Directory**.



thinBackup Configuration

Backup settings

Backup directory

lib/jenkins/backup

The directory does not exist, but will be created

Backup schedule for full backups

E H**1-5

Backup schedule for differential backups

E H**1-10

Max number of backup sets

10

Files excluded from backup (regular expression)

☒ Wait until Jenkins/Hudson is idle to perform a backup

Force Jenkins to quiet mode after specified minutes

100

☒ Backup build results

☒ Backup build archive

☐ Backup only builds marked to keep

Step 7: Go to the previous page to test whether the backup is happening or not by clicking on **Backup Now** as shown in the below image.

Jenkins > ThinBackup

New Item

People

Build History

Manage Jenkins

My Views

Open Blue Ocean

Credentials

Backup Now

Restore

Settings

Build Queue ^

Step 8: At last, you can check the Backup Directory specified in the ThinBackup Settings. (Step 6) to check the whole backup

Q #30) What is Flow Control in Jenkins?

Answer: In Jenkins, flow control follows the pipeline structure (scripted pipeline) that are being executed from the top to bottom of the Jenkins file.

Q #31) 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).

Q #32) 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.

Q #33) How can you define a Continuous Delivery Workflow?

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

Q #34) What are the various ways in which the build can be scheduled in Jenkins?

Answer: The build can be triggered in the following ways:

1. After the completion of other builds.
2. By source code management (modifications) commit.
3. At a specific time.
4. By requesting manual builds.

Q #35) Why is Jenkins called a Continuous Delivery Tool?

Answer: We have seen the Continuous Delivery workflow in the previous 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.

In this way, Jenkins follows a **Continuous Delivery approach** and is called the **Continuous Delivery Tool**.

Q #36) Give any simple example of Jenkins script.

Answer: This is a Jenkins declarative pipeline code for Java:

```
pipeline {
    agent

    stages {
        stage('Building your first asset') {
            agent

            steps {
                echo 'Build asset'
            }
        }
    }

    stage('Test') {
        agent
```

```
steps {  
  
    echo 'Building project 1'  
  
}  
  
}  
  
}
```

These are some of the frequently asked questions in the Jenkins interview, you can practice them as part of your preparation for the interview.

We advise you to practice some Jenkins scripts before attending an interview. Also for detailed information and processes in Jenkins please check **Jenkins**.

Conclusion

The above-mentioned interview questions are being asked during Jenkins Software Configuration Management and DevOps interview as Jenkins is also considered as an important tool in the DevOps program.

The real-time Jenkins interview questions listed in this article can be referred by beginners, professionals as well as job seekers who are looking for a good opportunity or a technical position in Jenkins.