Aim: To Create a new build job in Jenkins.

Objective: The objective of creating a new build job in Jenkins is to set up an automated process that fetches the latest source code from a version control repository, compiles the code, executes tests

Theory:

What is a Jenkins Freestyle Project?

Jenkins Freestyle Project is a repeatable build job, script, or pipeline that contains steps and post-build actions. It is an improved job or task that can span multiple operations. It allows you to configure build triggers and offers project-based security for your Jenkins project. It also offers plugins to help you build steps and post-build actions.
The types of actions you can perform in a Jenkins build step or post-build action are quite limited. There are many standard plugins available within a Jenkins Freestyle Project to help you overcome this problem.

Fig 5.1 How to Create a Job in Jenkins

Features of Jenkins:

Some of the crucial features of Jenkins are the following:

- It is a free and open-source automation tool
- Jenkins provides a vast number of plugins
- It is easy to set up and install on multiple operating systems
- Provides pipeline support
- Fast release cycles
- Easy upgrades

<u>Steps to Create a New Build Job in Jenkins:</u>

Step 1: Login to Jenkins

To create a Jenkins freestyle job, log on to your Jenkins dashboard by visiting your Jenkins installation path. Usually, it will be hosted on localhost at http://localhost:8080

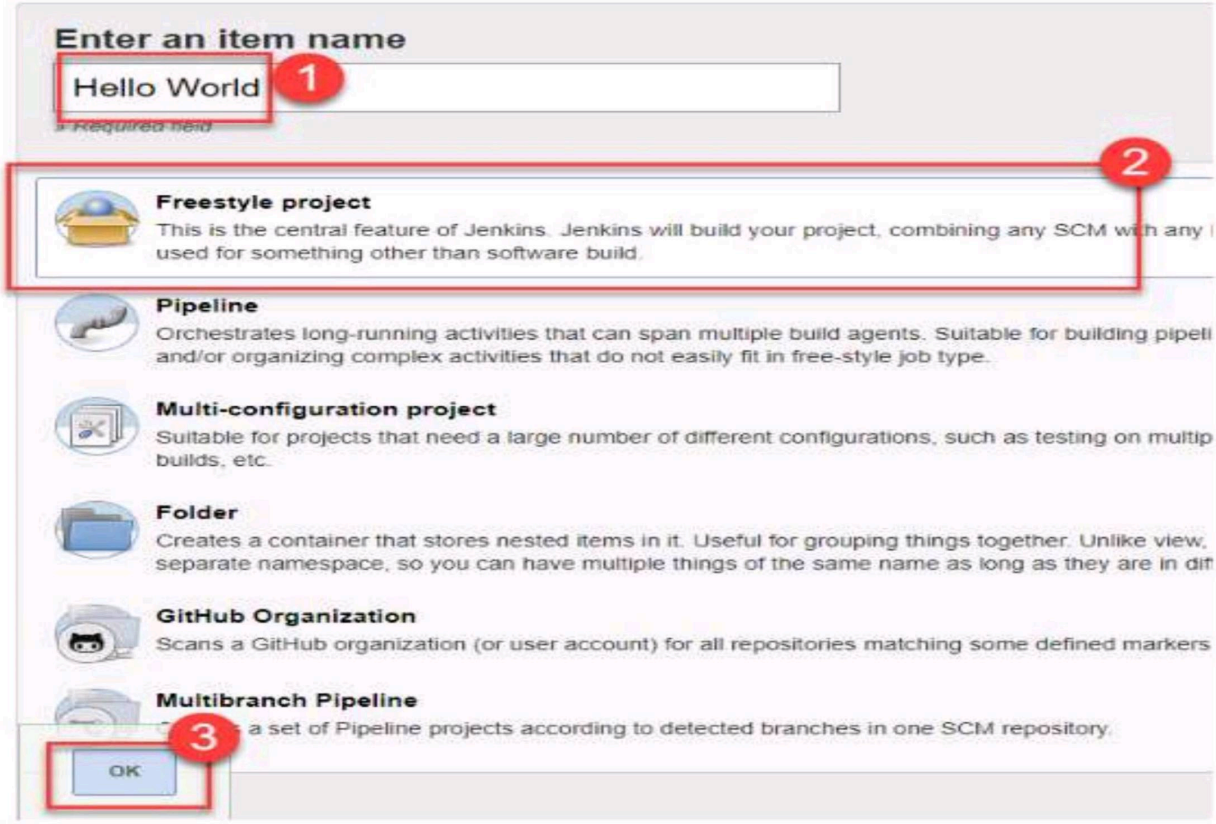
Step 2: Create New Item

Click on "**New Item**" at the top left-hand side of your dashboard.

Step 3: Enter Item details

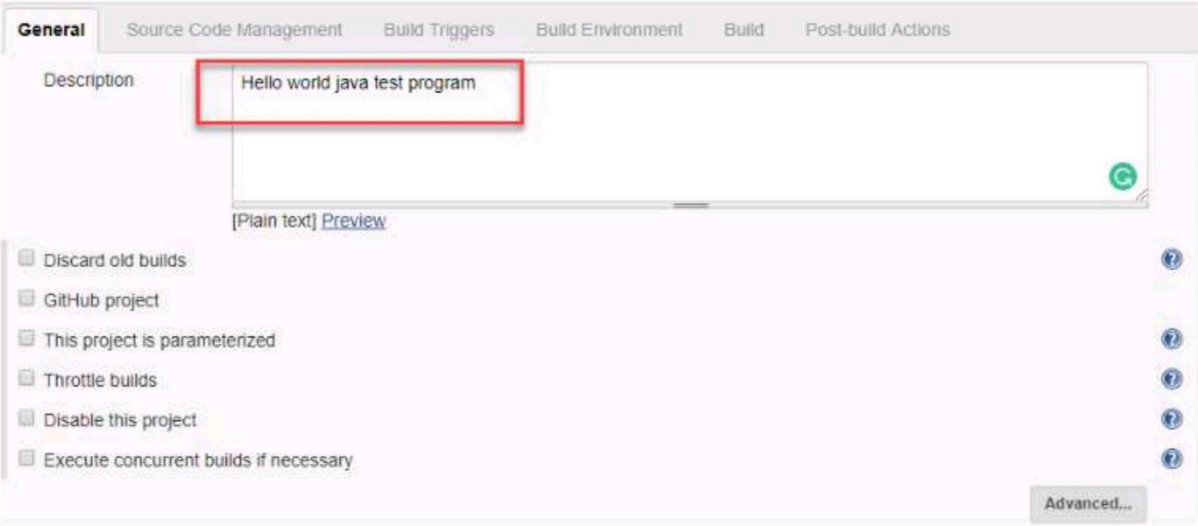
In the next screen,

1. Enter the name of the item you want to create. We shall use the "Hello world" for this demo.
2. Select Freestyle project
3. Click Okay



Step 4: Enter Project details

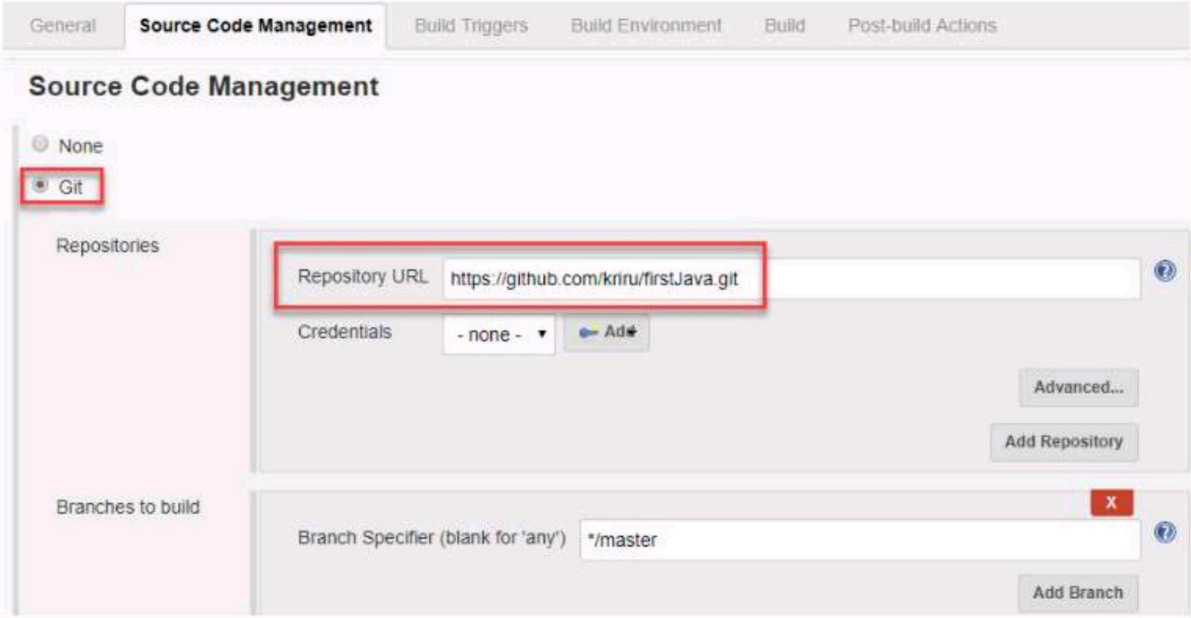
Enter the details of the project you want to test.



Step 5: Enter repository URL

Under Source Code Management, Enter your repository URL. We have a test repository located at https://github.com/kriru/firstJava.git

It is also possible for you to use a local repository.



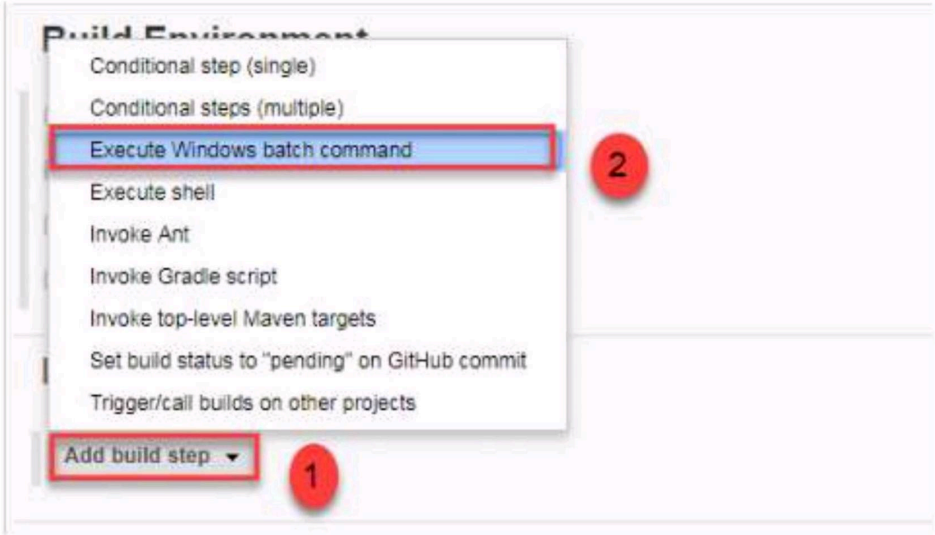
If your GitHub repository is private, Jenkins will first validate your login credentials with GitHub and only then pull the source code from your GitHub repository.

Step 6: Tweak the settings

Now that you have provided all the details, it's time to build the code. Tweak the settings under the **build** section to build the code at the time you want. You can even schedule the build to happen periodically, at set times.

Under **build**,

1. Click on "**Add build step**"
2. Click on "**Execute Windows batch command**" and add the commands you want to execute during the build process.



In the command window, enter the following commands and then click on the Save button.

Javac HelloWorld.java
Java HelloWorld

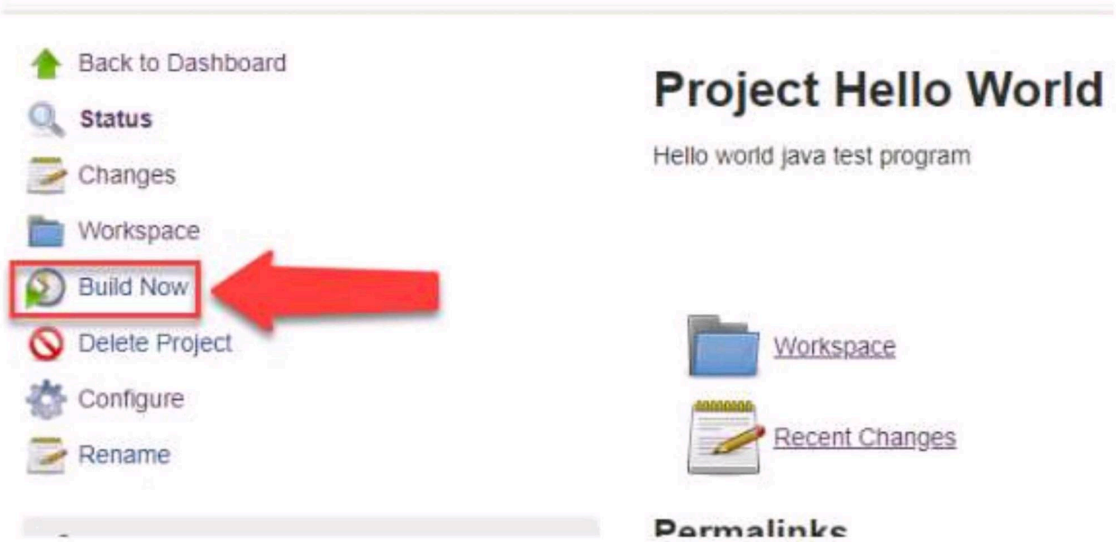
Step 7: Save the project

When you have entered all the data,

1. Click **Apply**
2. **Save** the project.

Step 8: Build Source code

Now, in the main screen, Click the **Build Now** button on the left-hand side to build the source code.



Step 9: Check the status
After clicking on **Build now,** you can see the status of the build you run under **Build History**.

Step 10: See the console output
Click on the **build number** and then Click on **console output** to see the status of the build you run. It should show you a success message.

have successfully created a new build job in Jenkins to automate the process of fetching the latest source code, compiling it, and executing tests. This will help ensure that the code is continuously

Conclusion:

1. Which SCM tools Jenkins supports?

Jenkins supports a variety of Source Code Management (SCM) tools, including AccuRev, CVS, Subversion, Git, Mercurial, Perforce, Clearcase, and RTC. These tools enable Jenkins to manage and automate the build, test, and deployment processes for different software projects. Jenkins provides plugins for each of these SCM tools, which extend its functionality and allow it to integrate seamlessly with the development environment.

For instance, the Git plugin allows Jenkins to interact with Git repositories, enabling features like polling for changes, building from a specific branch, and merging code changes. Similarly, the Subversion plugin enables Jenkins to interact with Subversion repositories, providing features like checkout, update, and commit operations.

2. What are the various ways in which build can be scheduled in Jenkins?
   1. Build periodically: This is the most common way to schedule a build in Jenkins. It allows you to specify a schedule using a cron-like syntax. For example, you can schedule a build to

run every hour by entering `@hourly`, or at 4 PM every day by entering `0 16 * * *`. Jenkins also supports predefined aliases for common intervals, such as `@daily`, `@weekly`, and `@monthly`.

2. Schedule Build Plugin: This plugin adds the capability to schedule a build for a later point in time. It allows you to select a date and time for the build, and adds it to the build queue with a specified quiet period.
3. Quiet Period: A quiet period is a time period during which Jenkins will not start a new build if a previous build is still in progress. This can be useful for preventing conflicts or resource contention between builds.
4. Hash-based scheduling: Jenkins supports hash-based scheduling, which calculates the build start time based on the hash code of the project name. This can be useful for scheduling multiple builds to run at different times, even if they are scheduled for the same time.
5. Build Triggers: Jenkins supports various build triggers, such as SCM changes, manual builds, and timer-based triggers. These triggers can be configured to start a build based on specific events or conditions.