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.

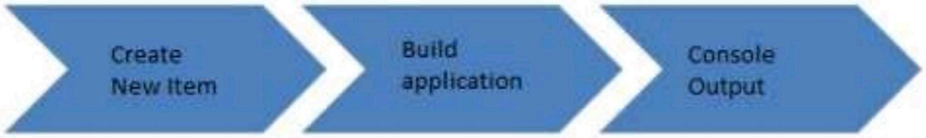


Fig 5.1 How to Create a Job in Jenkins

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

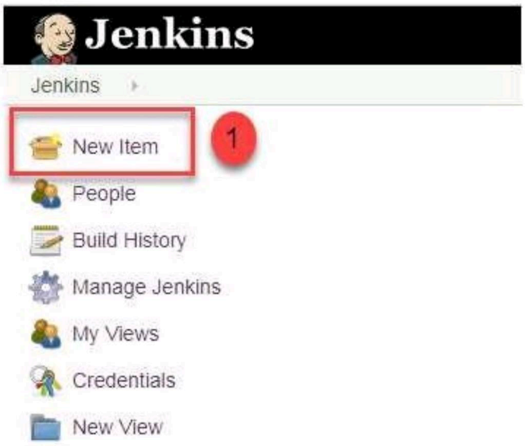
Steps to Create a New Build Job in Jenkins:

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

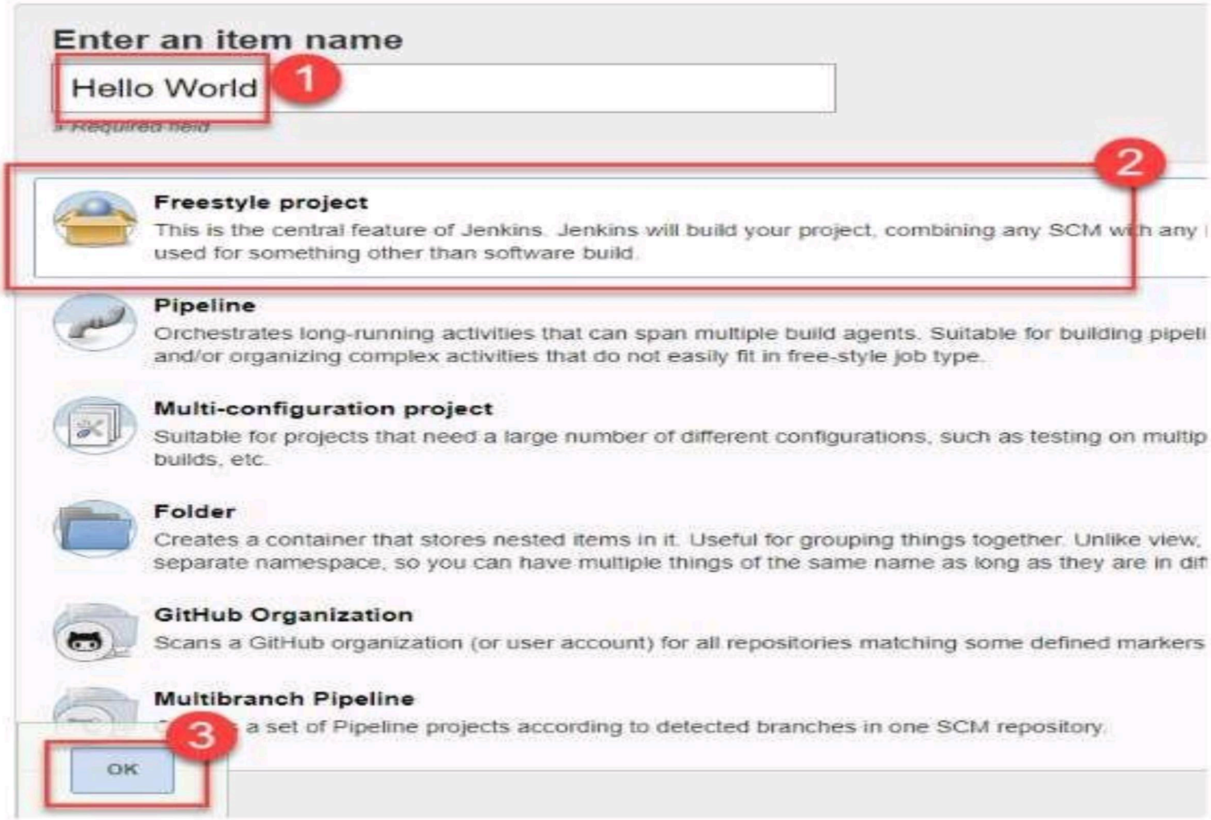
Click on "**New Item**" at the top left-hand side of your dashboard.



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



Enter an item name

Hello World

» Required field

Freestyle project
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any
used for something other than software build.

Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipeli
and/or organizing complex activities that do not easily fit in free-style job type.

Multi-configuration project
Suitable for projects that need a large number of different configurations, such as testing on multip
builds, etc.

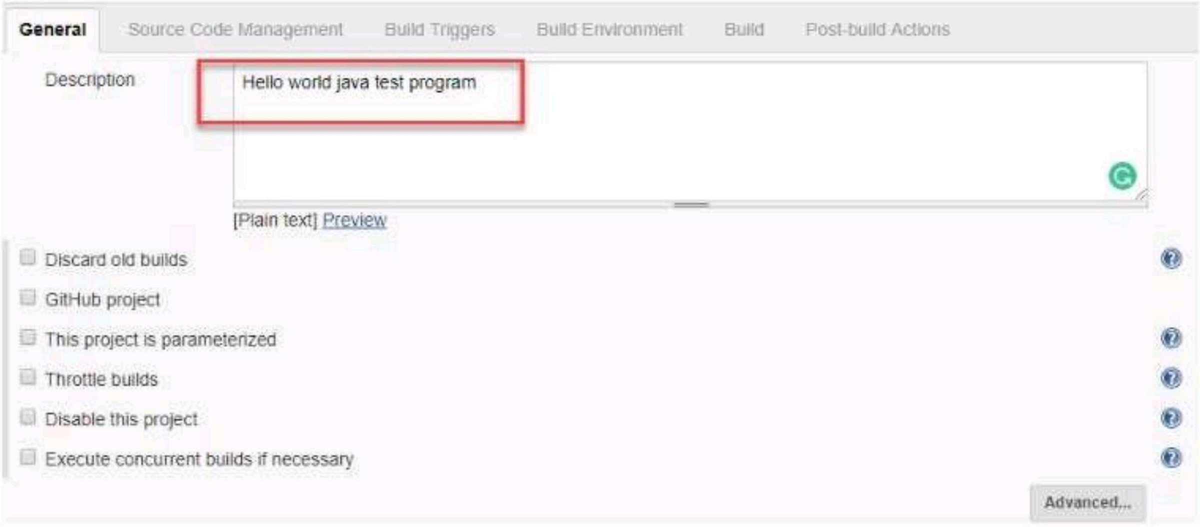
Folder
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view,
separate namespace, so you can have multiple things of the same name as long as they are in dif

GitHub Organization
Scans a GitHub organization (or user account) for all repositories matching some defined markers

Multibranch Pipeline
a set of Pipeline projects according to detected branches in one SCM repository.

Step 4: Enter Project details

Enter the details of the project you want to test.



Description: Hello world java test program

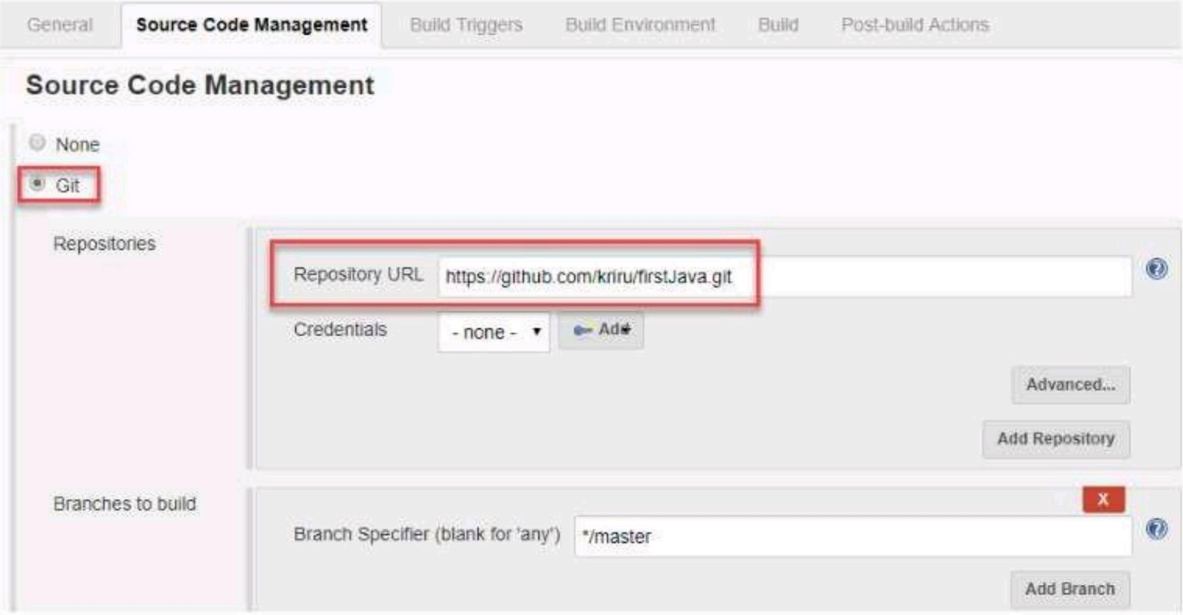
[Plain text] Preview

□ Discard old builds
□ GitHub project
□ This project is parameterized
□ Throttle builds
□ Disable this project
□ Execute concurrent builds if necessary

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.



Source Code Management

○ None
● Git

Repositories

Repository URL: https://github.com/kriru/firstJava.git

Credentials: - none -

Branches to build

Branch Specifier (blank for 'any'): */master

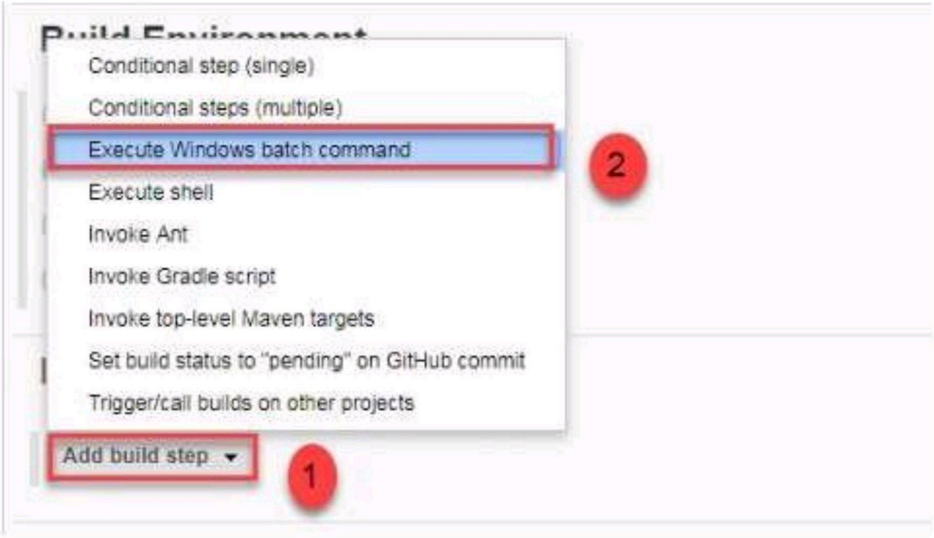
If your GitHub repository is private, Jenkins will first validate your login credentials with GitHub and only then pull the source code from your GitHub repository.

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.

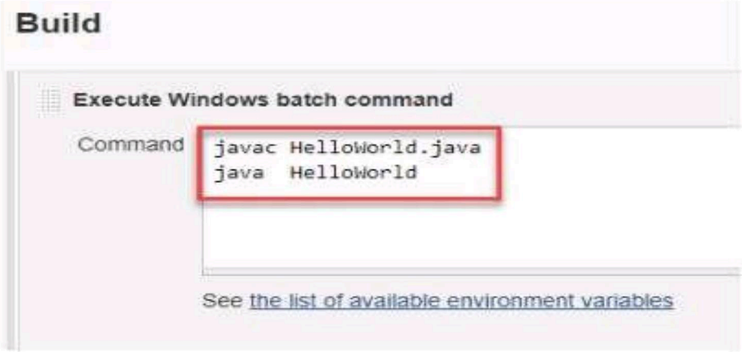
Under **build**,

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.


Build Environment
Conditional step (single)
Conditional steps (multiple)
Execute Windows batch command
Execute shell
Invoke Ant
Invoke Gradle script
Invoke top-level Maven targets
Set build status to "pending" on GitHub commit
Trigger/call builds on other projects
Add build step
1
2

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

Javac HelloWorld.java
Java HelloWorld


Build
Execute Windows batch command
See the list of available environment variables

Command

```
javac HelloWorld.java
java  HelloWorld
```

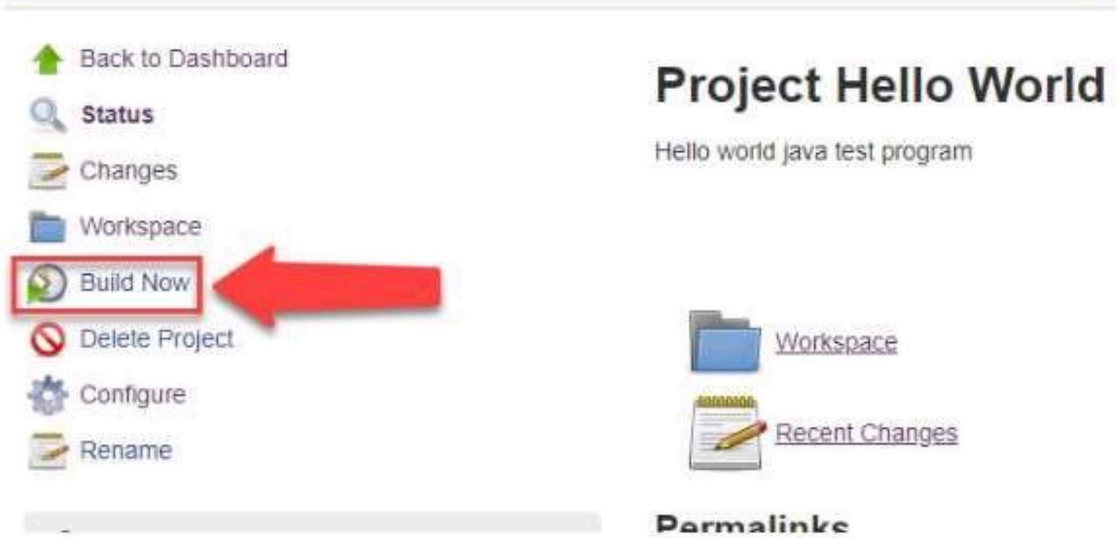
Step 7: Save the project

When you have entered all the data,

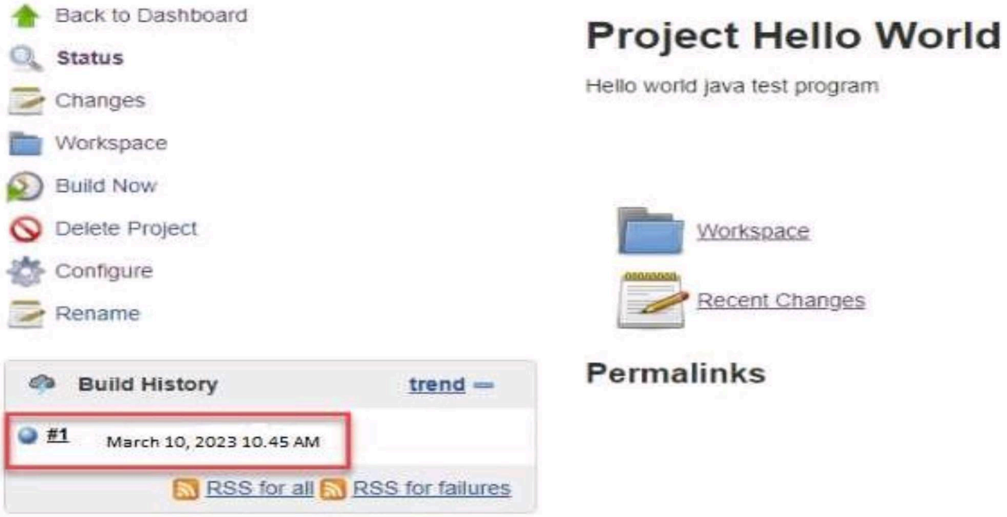
1. Click **Apply**
2. **Save** the project.

Step 8: Build Source code

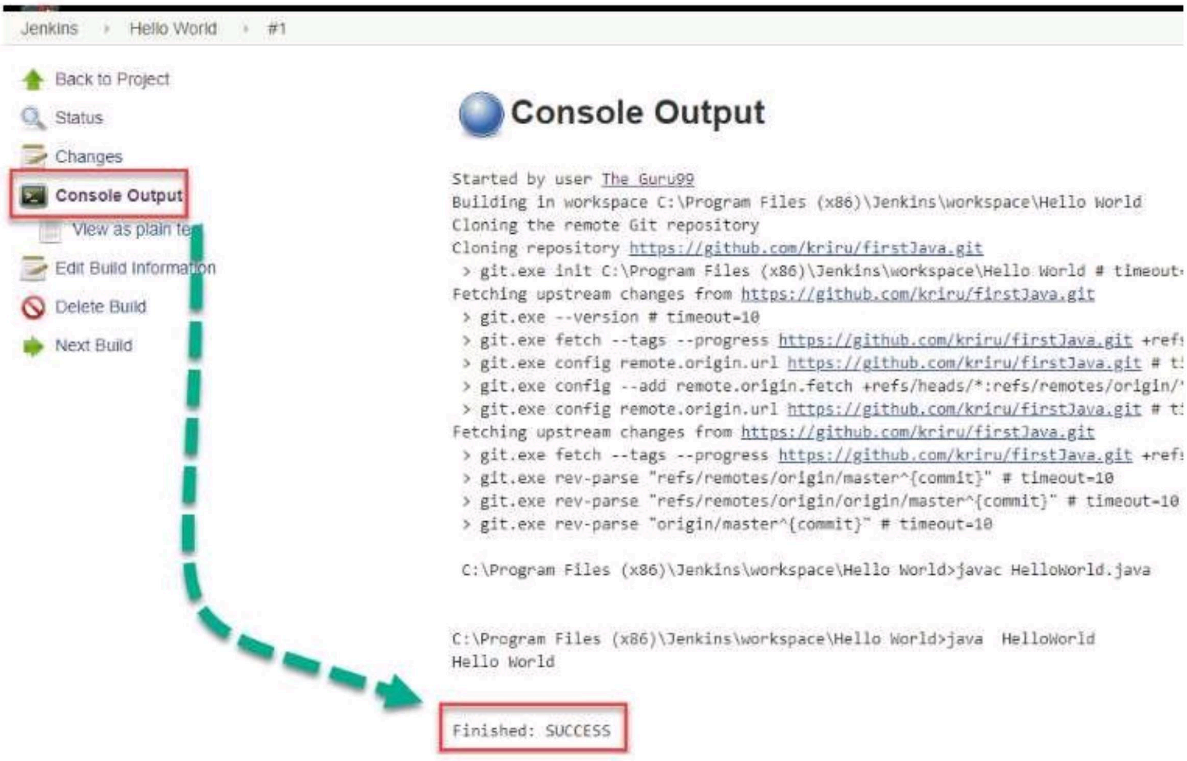
Now, in the main screen, Click the **Build Now** button on the left-hand side to build the source code.



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.



Console Output

```
Started by user The Guru99
Building in workspace C:\Program Files (x86)\Jenkins\workspace\Hello World
Cloning the remote Git repository
Cloning repository https://github.com/kriru/firstJava.git
 > git.exe init C:\Program Files (x86)\Jenkins\workspace\Hello World # timeout
Fetching upstream changes from https://github.com/kriru/firstJava.git
 > git.exe --version # timeout=10
 > git.exe fetch --tags --progress https://github.com/kriru/firstJava.git +ref
 > git.exe config remote.origin.url https://github.com/kriru/firstJava.git # t
 > git.exe config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/
 > git.exe config remote.origin.url https://github.com/kriru/firstJava.git # t
Fetching upstream changes from https://github.com/kriru/firstJava.git
 > git.exe fetch --tags --progress https://github.com/kriru/firstJava.git +ref
 > git.exe rev-parse "refs/remotes/origin/master^{commit}" # timeout=10
 > git.exe rev-parse "refs/remotes/origin/origin/master^{commit}" # timeout=10
 > git.exe rev-parse "origin/master^{commit}" # timeout=10

 C:\Program Files (x86)\Jenkins\workspace\Hello World>javac HelloWorld.java


C:\Program Files (x86)\Jenkins\workspace\Hello World>java  HelloWorld
Hello World

Finished: SUCCESS
```

Conclusion:

1. Which SCM tools Jenkins supports?

Ans: The SCM or Source Code Management tools Jenkins supports are SVN, Clearcase, CVS, Git, AccuRev, Perforce, RTC, Mercurial.

2. What are the various ways in which build can be scheduled in Jenkins?

Ans: 1. Builds can be triggered by source code management commits.

2. Builds can be triggered sequentially after completion of other builds.

3.Can be scheduled to run at a specified time using the CRON jobs.

4. Manual Build Requests.