



Aim: To Create a new build job in Jenkins. 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:

1. It is a free and open-source automation tool
2. Jenkins provides a vast number of plugins
3. It is easy to set up and install on multiple operating systems
4. Provides pipeline support
5. Fast release cycles
6. 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>

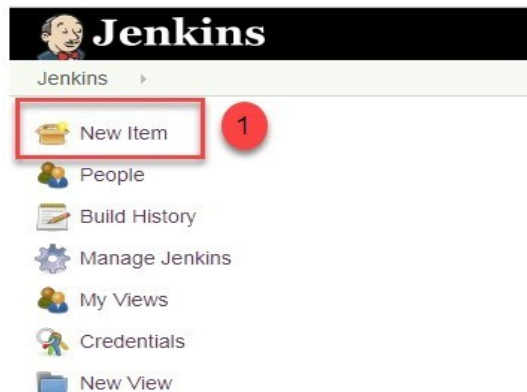
Step 2: Create New Item

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



Vidyavardhini's College of Engineering and Technology

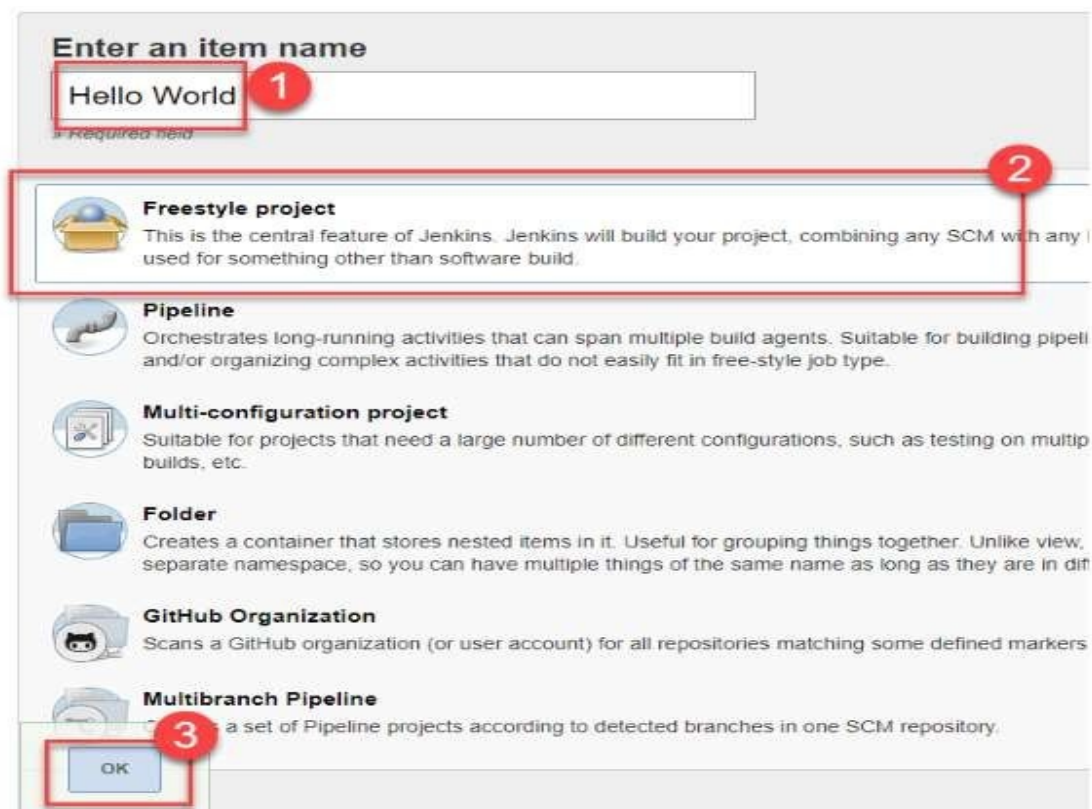
Department of Artificial Intelligence & Data Science



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.



Vidyavardhini's College of Engineering and Technology

Department of Artificial Intelligence & Data Science

The image shows the 'General' tab of a Jenkins job configuration. The 'Description' field is highlighted with a red box and contains the text 'Hello world java test program'. Below the description, there are several checkboxes: 'Discard old builds', 'GitHub project', 'This project is parameterized', 'Throttle builds', 'Disable this project', and 'Execute concurrent builds if necessary'. A 'Preview' link is visible next to the description field. The 'Advanced...' button is at the bottom right.

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.

The image shows the 'Source Code Management' tab of a Jenkins job configuration. The 'Git' radio button is selected and highlighted with a red box. Below it, the 'Repository URL' field is highlighted with a red box and contains the text 'https://github.com/kriru/firstJava.git'. The 'Credentials' dropdown is set to '- none -'. The 'Branches to build' section shows the 'Branch Specifier (blank for 'any')' set to '*/master'. The 'Add Repository' and 'Add Branch' buttons are visible at the bottom right.

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.



Vidyavardhini's College of Engineering and Technology

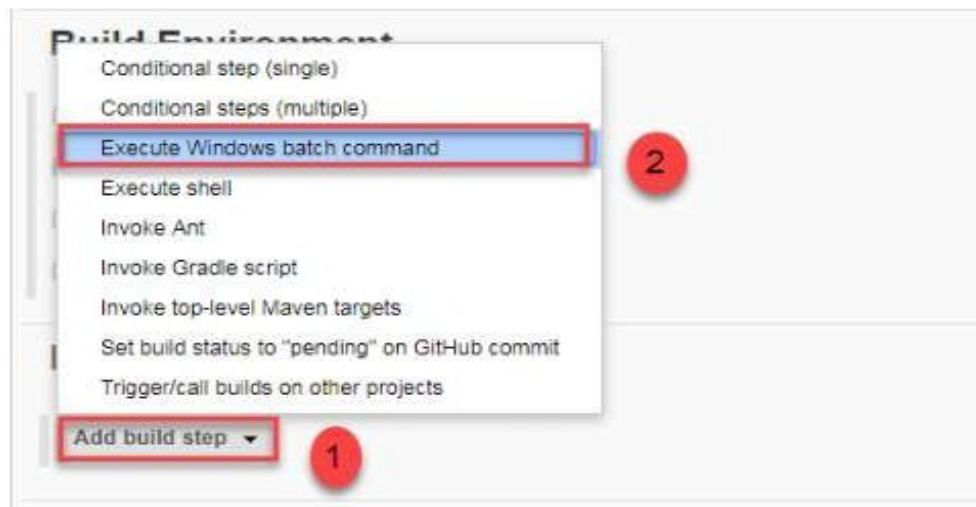
Department of Artificial Intelligence & Data Science

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**,

- Click on “**Add build step**”
- 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,

- Click **Apply**
- **Save** the project.

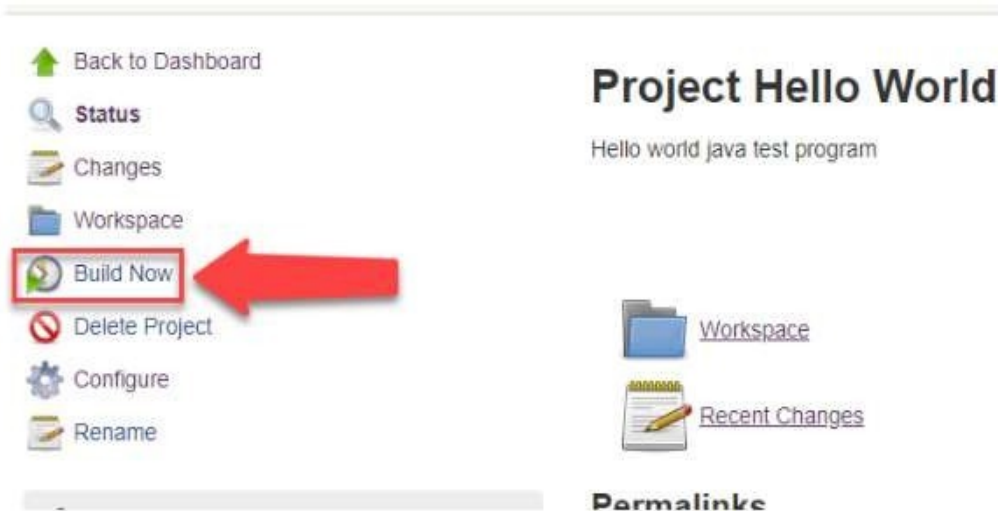


Vidyavardhini's College of Engineering and Technology

Department of Artificial Intelligence & Data Science

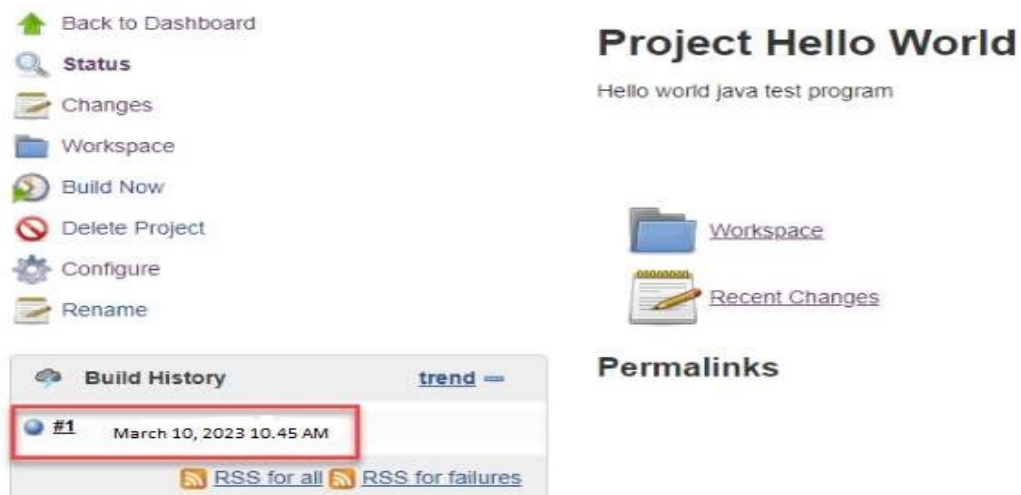
Step 8: Build Source code

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.



Vidyavardhini's College of Engineering and Technology

Department of Artificial Intelligence & Data Science

The screenshot shows the Jenkins web interface for a build named 'Hello World' (#1). On the left sidebar, the 'Console Output' tab is selected and highlighted with a red box. A green dashed arrow points from this tab to the console output area. The console output displays the following commands and their results:

```
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?

Jenkins supports various SCM tools, including Git, SVN (Subversion), Mercurial, CVS, Perforce, and others.

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

- Builds can be scheduled in Jenkins using various methods, including:
 - Periodic builds at specified intervals (e.g., every hour, daily, weekly).
 - Poll SCM to trigger builds based on changes in the source code repository.
 - Trigger builds manually by users.
 - Integration with external systems or triggers using plugins.