

Lab 6. Setup Build Jobs

Background

The freestyle build job is a highly flexible and easy-to-use option. You can use it for any type of project; it is easy to set up, and many of its options appear in other build jobs.

Objectives

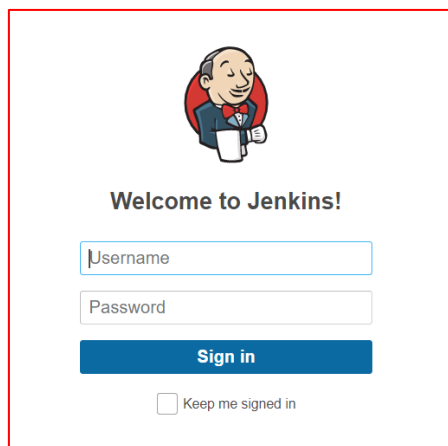
- Create a Job in Jenkins.
- Pull source code from GitHub.
- Build the Project
- Set Environment Variables.

Pre-Requisite

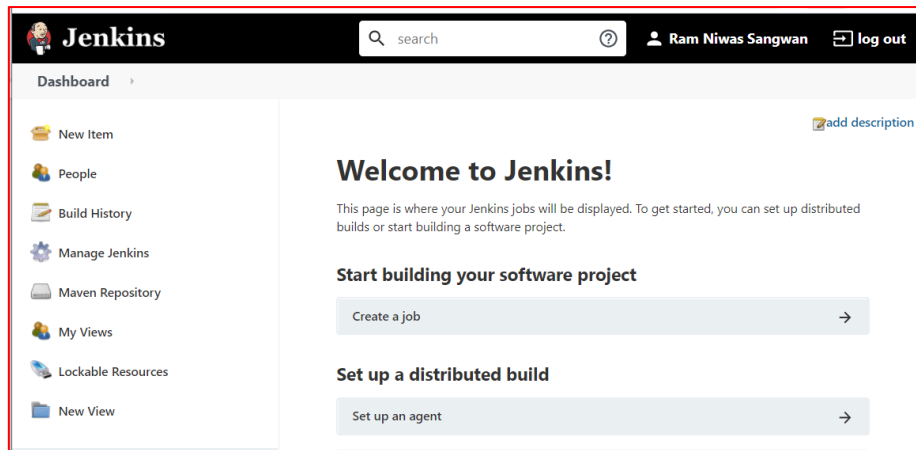
- OL 7 Virtual Machine with Jenkins Installed.
- Internet Access.

Sequence 1. Setup Build Jobs in Jenkins.

1. Log on to your Jenkins dashboard by visiting your Jenkins installation path. Usually, it will be hosted on localhost at <http://10.10.0.100:8080>



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



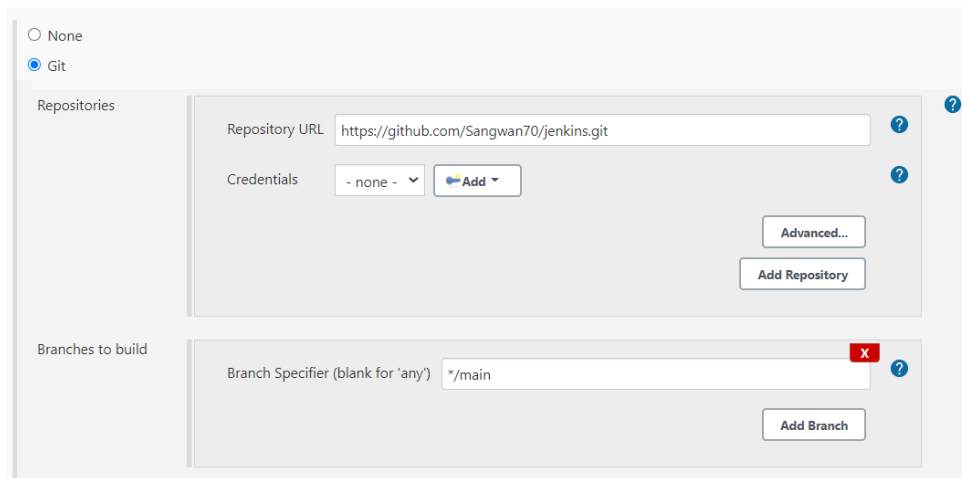
3. In the next screen, enter the name of the item you want to create. We shall use the "HelloWorld" for this demo.
 - Select Freestyle project
 - Click OK

This is a dialog box titled 'Enter an item name'. It features a text input field containing 'HelloWorld' and a 'Required field' message below it. Below the input field, there are two project type options, each with an icon and a description: 'Freestyle project' (represented by a box and arrow icon) and 'Maven project' (represented by a cat icon). The 'Freestyle project' description states it's the central feature of Jenkins, combining any SCM with any build system. The 'Maven project' description states it builds a maven project using POM files to reduce configuration.This is a continuation of the project selection dialog box. It shows two more options: 'GitHub Organization' (with a GitHub logo icon) and 'Multibranch Pipeline' (with a pipeline icon). The 'GitHub Organization' description says it scans a GitHub organization for repositories matching defined markers. The 'Multibranch Pipeline' description says it creates a set of Pipeline projects based on detected branches in one SCM repository. At the bottom left, there is a blue 'OK' button.

4. Enter the details of the project you want to test.

This screenshot shows the 'General' tab of the Jenkins Job Configuration page. The 'Description' field is a large text area containing 'Hello World from DevOps Team at Oracle India'. Below the text area, there's a '[Plain text] Preview' link. At the bottom, there are two checkboxes: 'Discard old builds' and 'GitHub project', both of which are currently unchecked. A help icon (?) is visible next to the 'GitHub project' checkbox.

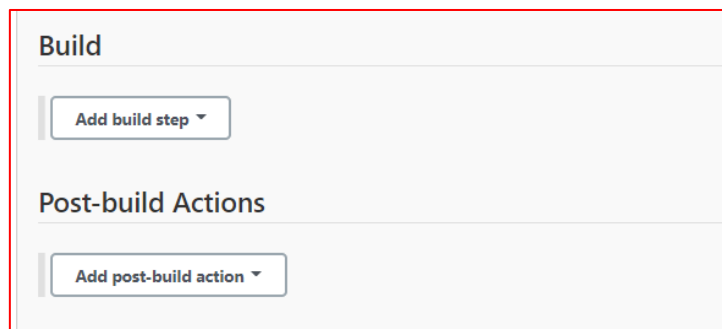
- Under Source Code Management, Enter your repository URL. We have a test repository located at <https://github.com/Sangwan70/jenkins.git>
Change the Branch Specifier to “*/main”



The screenshot shows the Jenkins configuration page for Source Code Management. On the left, there are two radio buttons: "None" and "Git", with "Git" selected. Below this is a section titled "Repositories". It contains a "Repository URL" text field with the value "https://github.com/Sangwan70/jenkins.git", a "Credentials" dropdown menu showing "- none -" with an "Add" button next to it, and an "Advanced..." button. Below the "Repositories" section is a section titled "Branches to build". It contains a "Branch Specifier (blank for 'any')" text field with the value "*/main" and an "Add Branch" button. There are blue question mark icons next to the "Repository URL" and "Branch Specifier" fields.

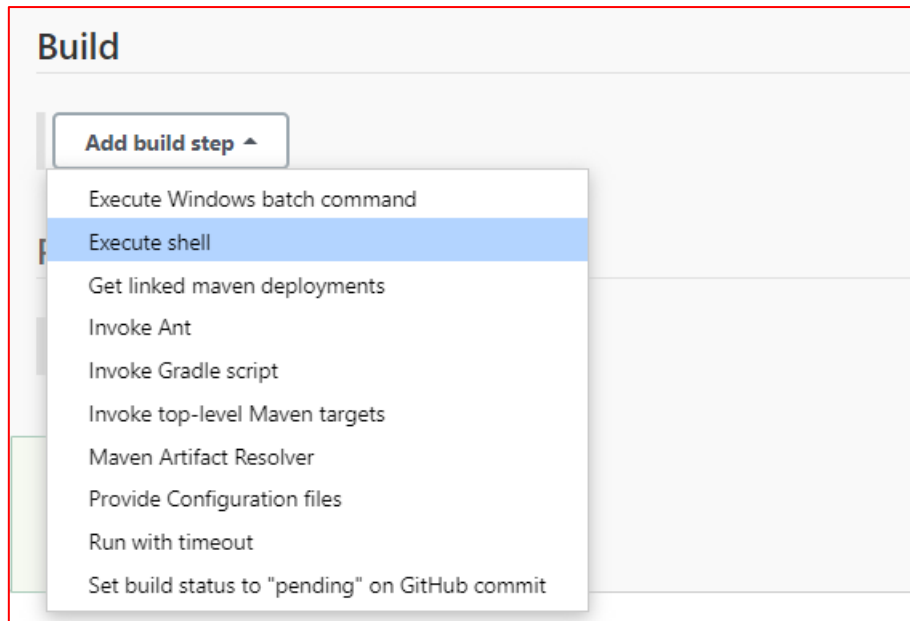
It is also possible for you to use a local repository. If your GitHub repository is private, Jenkins will first validate your login credentials with GitHub and only then pull the source code from your GitHub repository.

- Now that you have provided all the details, it is time to build the code. Under build, Click on "Add build step"



The screenshot shows the Jenkins configuration page for the Build section. It has a title "Build" and a button "Add build step". Below this is a section titled "Post-build Actions" with a button "Add post-build action". The entire configuration area is enclosed in a red rectangular border.

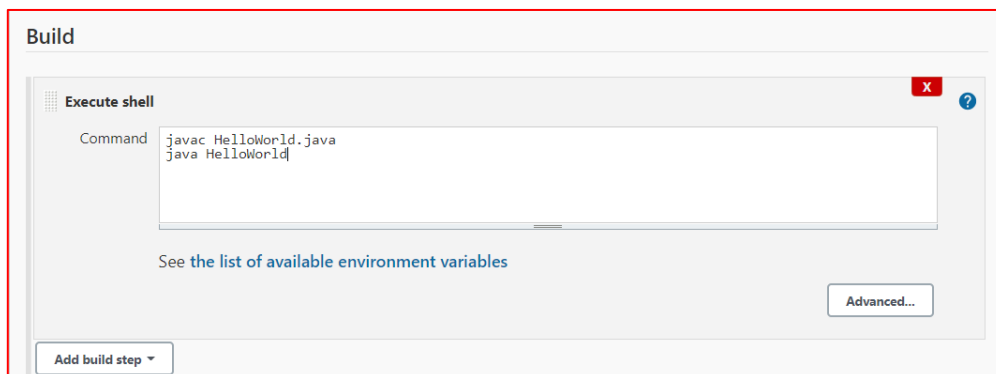
- Click on "Execute shell" and add the commands you want to execute during the build process.



Here, I have added the java commands to compile the java code.

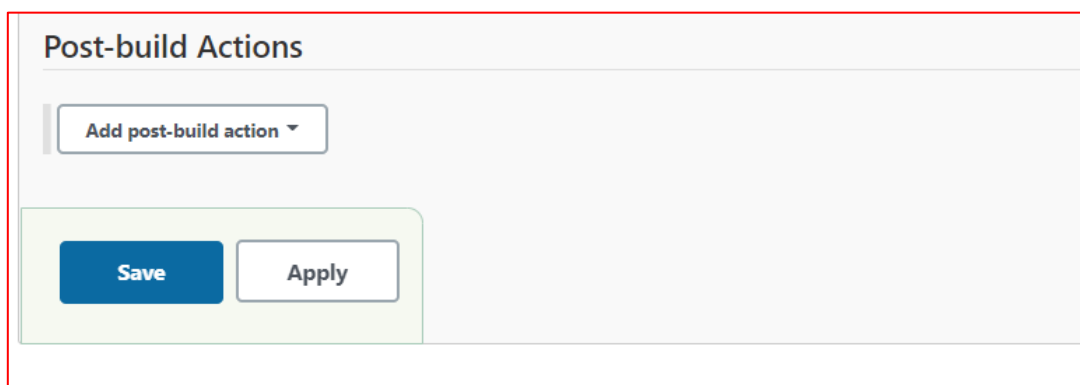
javac HelloWorld.java

java HelloWorld

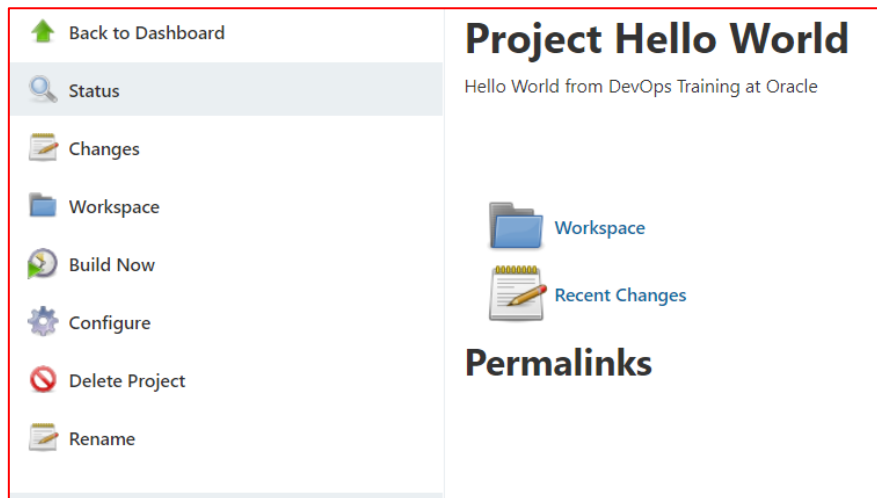


8. When you have entered all the data,

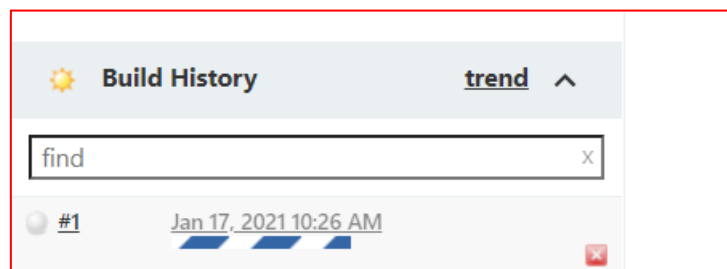
- Click **Apply**
- **Save** the project.



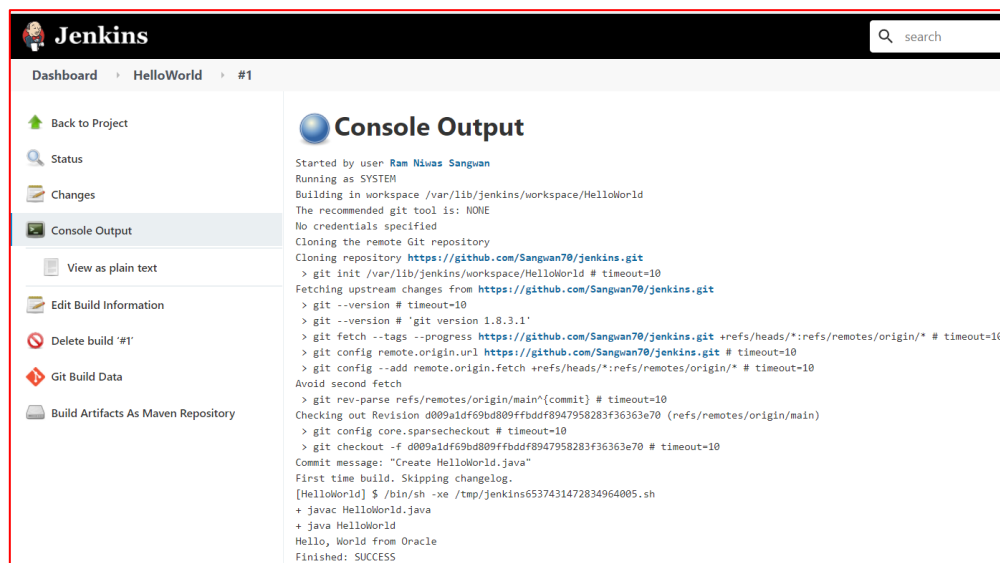
9. Now, in the main screen, click the **Build Now** button on the left-hand side to build the source code.



10. After clicking on Build now, you can see the status of the build you run under Build History.

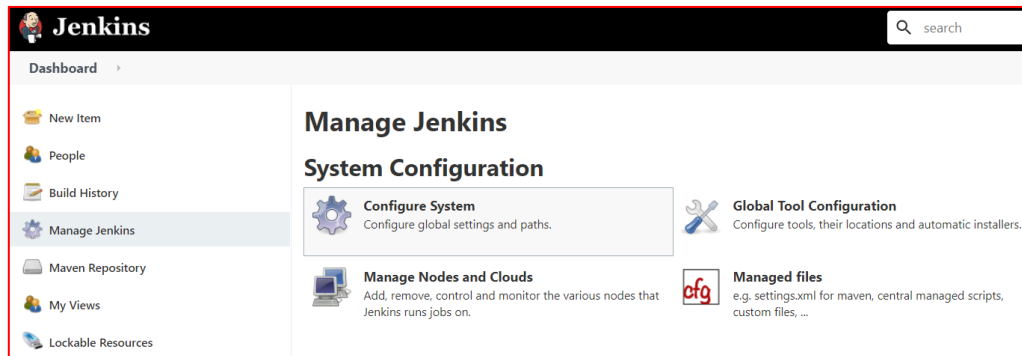


11. Click on the build number and then Click on console output to see the status of the build you run. It should show you a success message.

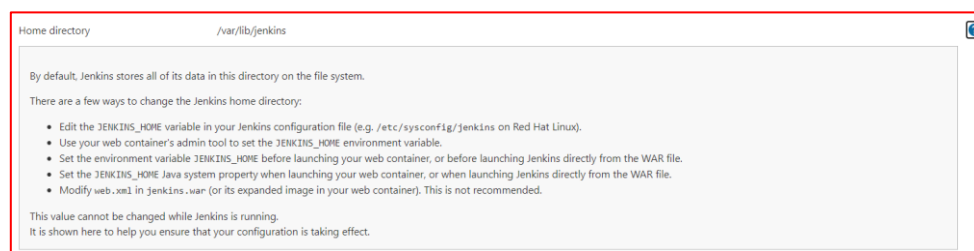


We have executed a HelloWorld program hosted on GitHub. Jenkin pulls the code from the remote repository and builds continuously at a frequency you define.

12. To configure **Environment Variables**, settings and paths, we need to go to Configure System in the Jenkins dashboard.
- Go to the Jenkins dashboard.
 - Click on Manage Jenkins.
 - Click on Configure System.
 - Verify the Home directory available:



- In the Configure System page, we can define Environment variables too so that it can be used during the execution of build jobs.



13. For example, we can the ANDROID SDK path in Environment variables in to set Continuous Integration for Android Apps.

