

Lab Exercise 8 – Setting Up Maven

Build Job In Jenkins

Name:-Vansh Bhatt

SapId:- 500125395

R.No:- R2142231689

Batch:- DevOps B1

To:- Hitesh Kumar Sharma Sir

Objective: To set up Maven build job in Jenkins for automating the build process, enabling continuous integration to enhance the software development lifecycle

Tools required: Jenkins

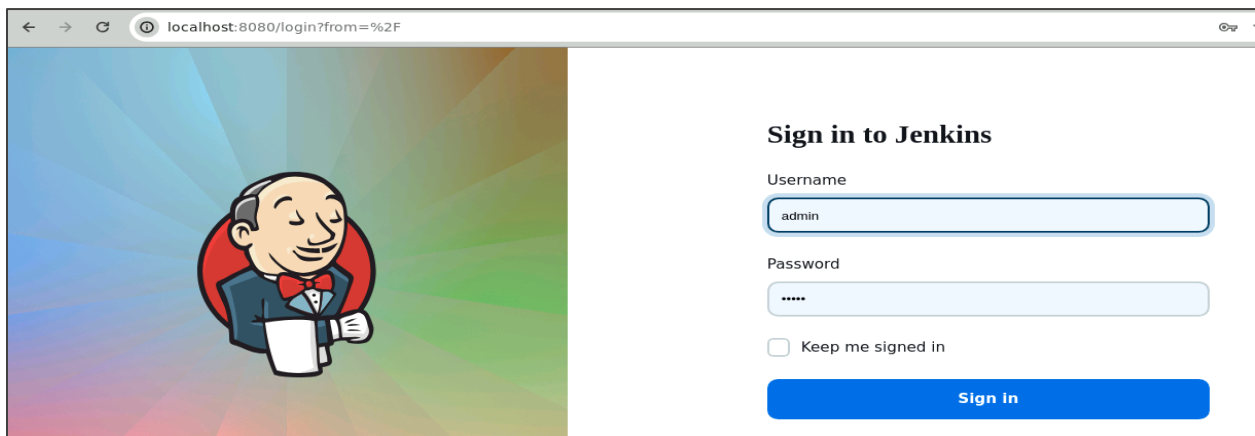
Prerequisites: You need to have a Jenkins up and running.

Steps to be followed:

1. Log in to Jenkins CI tool and configure Maven freestyle job

Step 1: Log in to Jenkins CI tool and configure Maven freestyle job

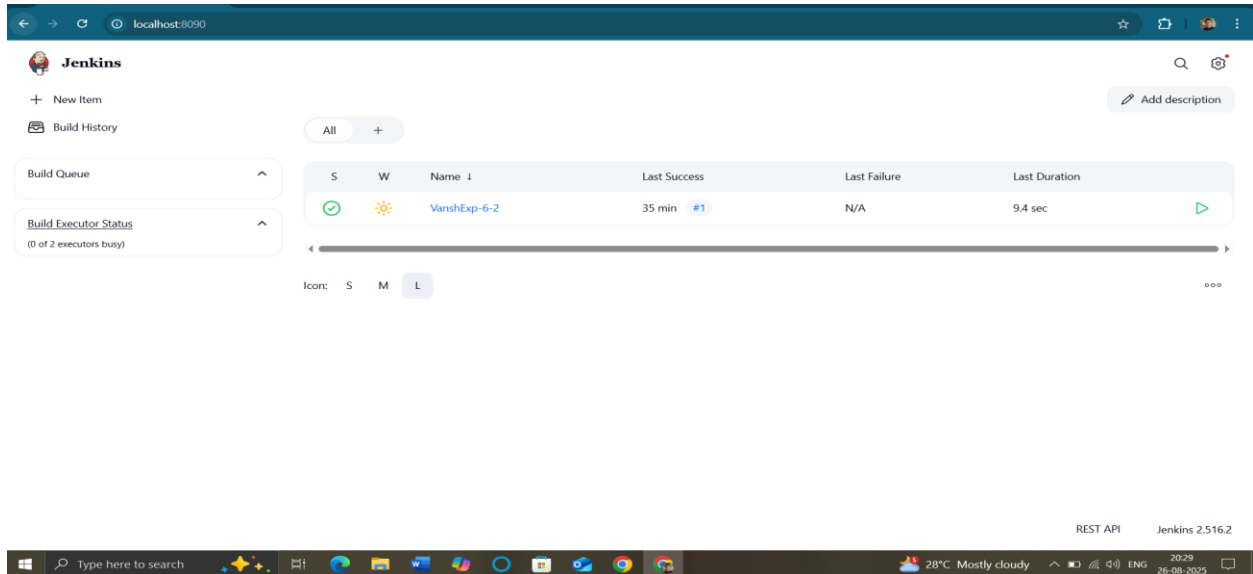
1.1 Log in to Jenkins using your credentials



The screenshot shows the Jenkins login interface in a web browser. The address bar displays 'localhost:8080/login?from=%2F'. On the left, there is a cartoon illustration of a man in a tuxedo. On the right, the title 'Sign in to Jenkins' is followed by input fields for 'Username' (containing 'admin') and 'Password' (masked with dots). Below these fields is a checkbox labeled 'Keep me signed in' and a blue 'Sign In' button.

Note: The credentials for accessing Jenkins in the lab are Username: **admin** and Password: **admin**.

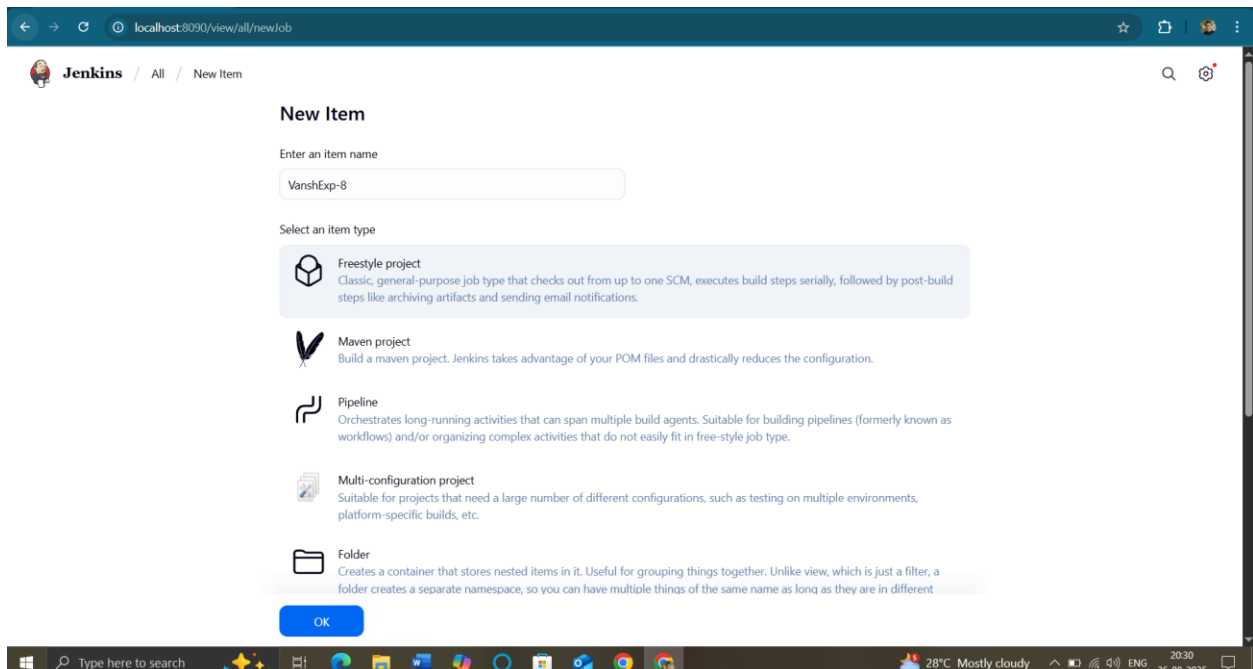
1.2 In the Jenkins dashboard, click on **New Item**



The screenshot shows the Jenkins dashboard at localhost:8090. On the left sidebar, there are links for 'New Item' and 'Build History'. The main area displays a table of jobs. The table has columns for 'S' (Status), 'W' (Icon), 'Name', 'Last Success', 'Last Failure', and 'Last Duration'. A single job is listed: 'VanshExp-6-2' with a status of 'Success' (green checkmark), a sun icon, a last success time of '35 min #1', and a last duration of '9.4 sec'. Below the table, there are tabs for 'Icon: S M L'. At the bottom right, it says 'REST API' and 'Jenkins 2.516.2'.

S	W	Name	Last Success	Last Failure	Last Duration
✓	☀	VanshExp-6-2	35 min #1	N/A	9.4 sec

1.3 Select the **Freestyle project** while creating a Jenkins job, provide a custom job name, and click on **OK**



The screenshot shows the 'New Item' form in Jenkins. The 'Enter an item name' field contains 'VanshExp-8'. Under 'Select an item type', the 'Freestyle project' option is selected. The description for 'Freestyle project' is: 'Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.' Other options include 'Maven project', 'Pipeline', 'Multi-configuration project', and 'Folder'. At the bottom, there is a blue 'OK' button.

New Item

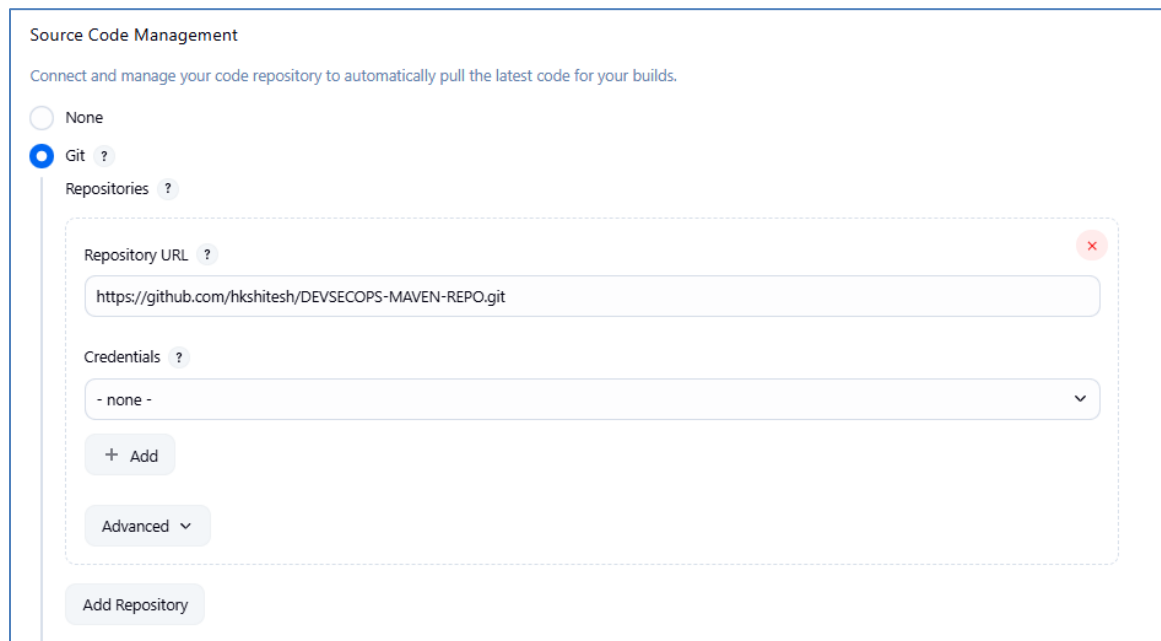
Enter an item name
VanshExp-8

Select an item type

- Freestyle project**
Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.
- Maven project
Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.
- Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) 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 multiple environments, platform-specific builds, etc.
- Folder
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different

OK

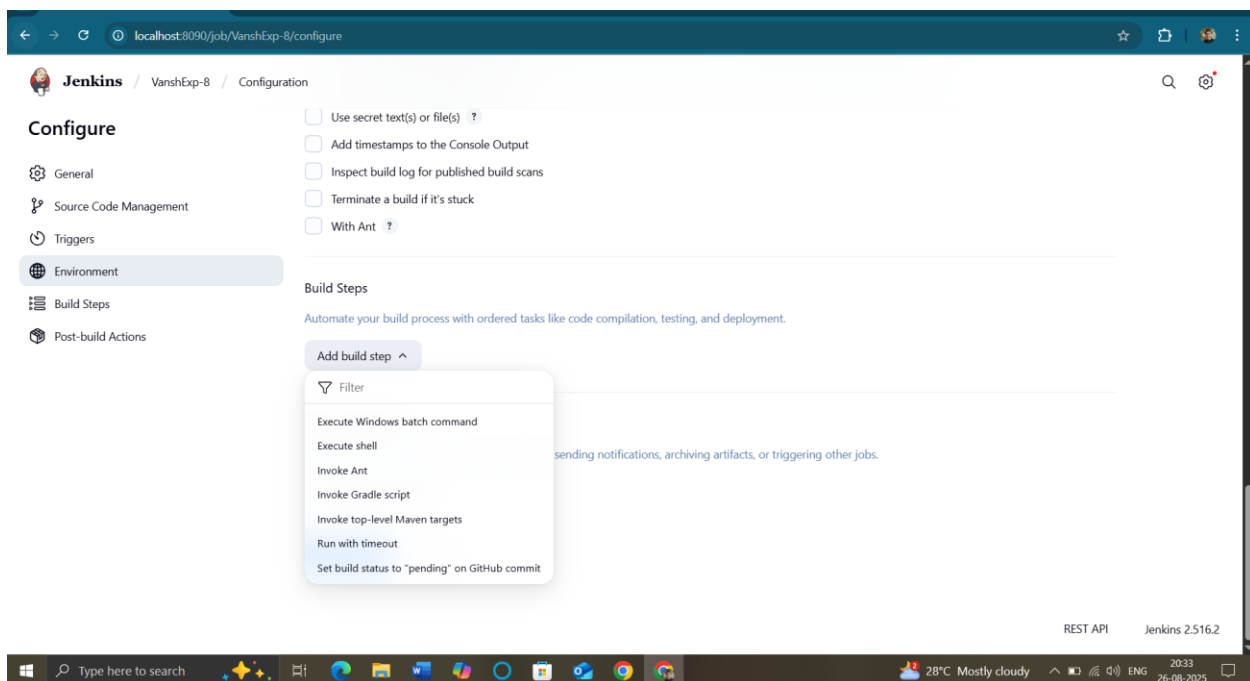
1.4 Now, in the Configure page, navigate to **Source Code Management** in the left navigation bar, select **Git**, and then provide the Git repository URL



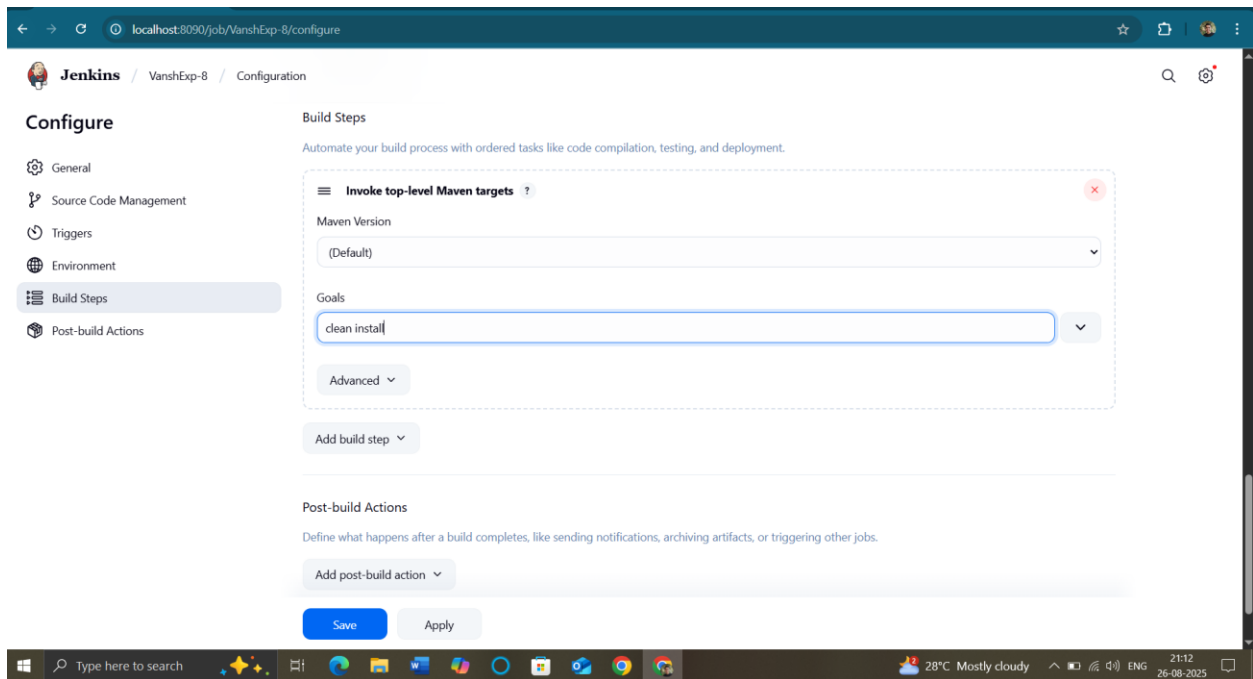
The screenshot shows the 'Source Code Management' configuration page in Jenkins. At the top, it says 'Connect and manage your code repository to automatically pull the latest code for your builds.' Below this, there are two radio buttons: 'None' and 'Git'. The 'Git' option is selected. Under the 'Git' section, there is a 'Repositories' heading. A dashed box contains a 'Repository URL' field with the value 'https://github.com/hkshitesh/DEVSECOPS-MAVEN-REPO.git', a 'Credentials' dropdown menu set to '- none -', an 'Add' button, and an 'Advanced' dropdown. Below the dashed box is an 'Add Repository' button.

Note: Here, the repository URL is <https://github.com/hkshitesh/DEVSECOPS-MAVEN-REPO.git>.

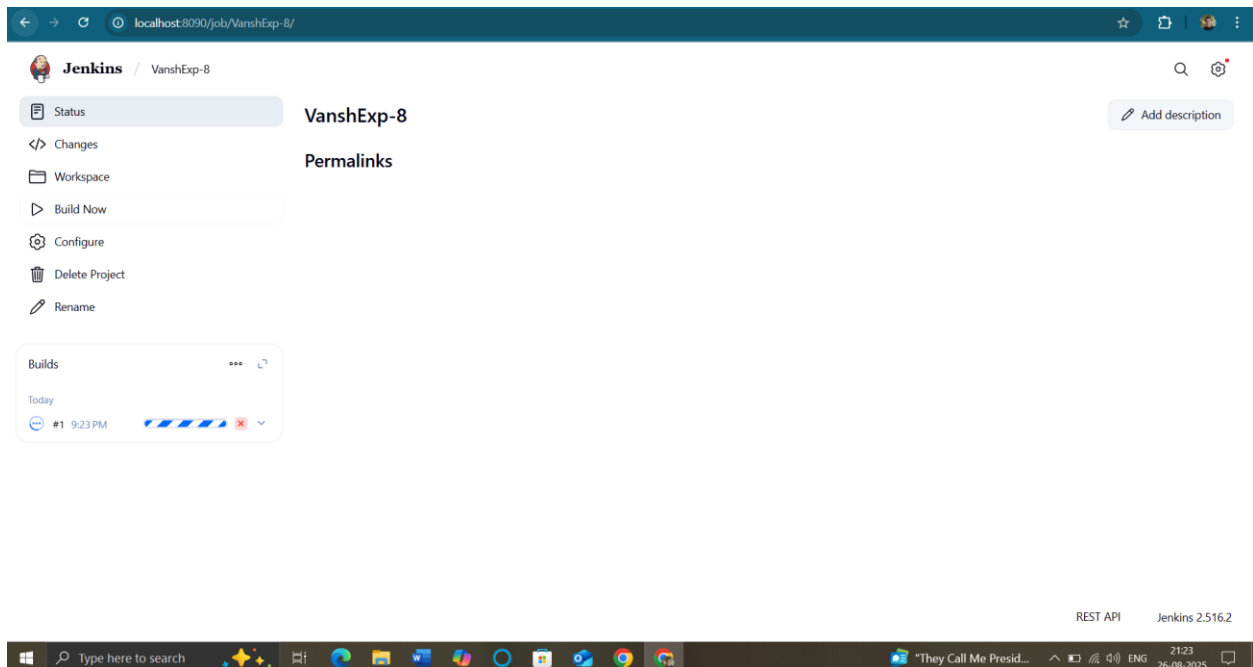
1.5 Now, navigate to **Build Steps**, click on **Add build step**, and then select the option **Invoke top-level Maven targets**

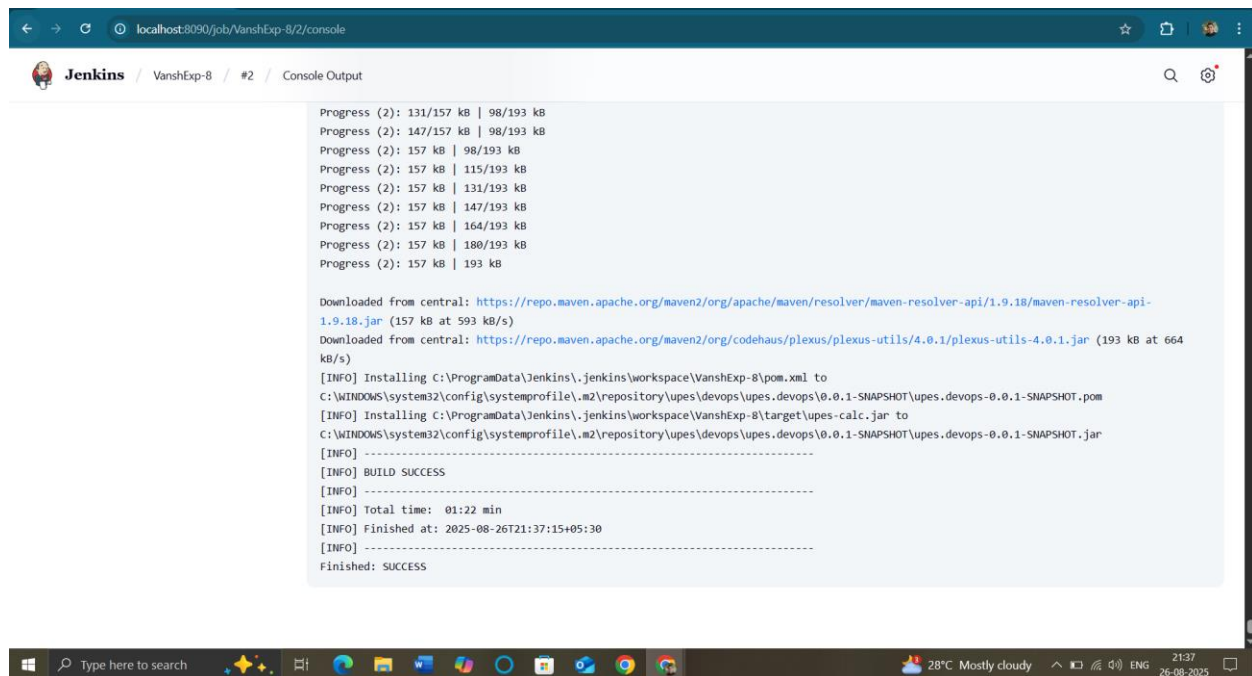


1.6 Provide clean install under **Goals** section and then click on **Save**



1.7 You will be navigated to the project after clicking on Save. Now, click on **Build Now** to initiate a new build, and the build logs will display the progress of the build process.





You can see that the build is configured successfully.

By following these steps, you have successfully set up Maven build job in Jenkins for automating the build process, enabling continuous integration to enhance the software development lifecycle.