

Lesson 06 Demo 03

Setting up Maven Build Job in Jenkins

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

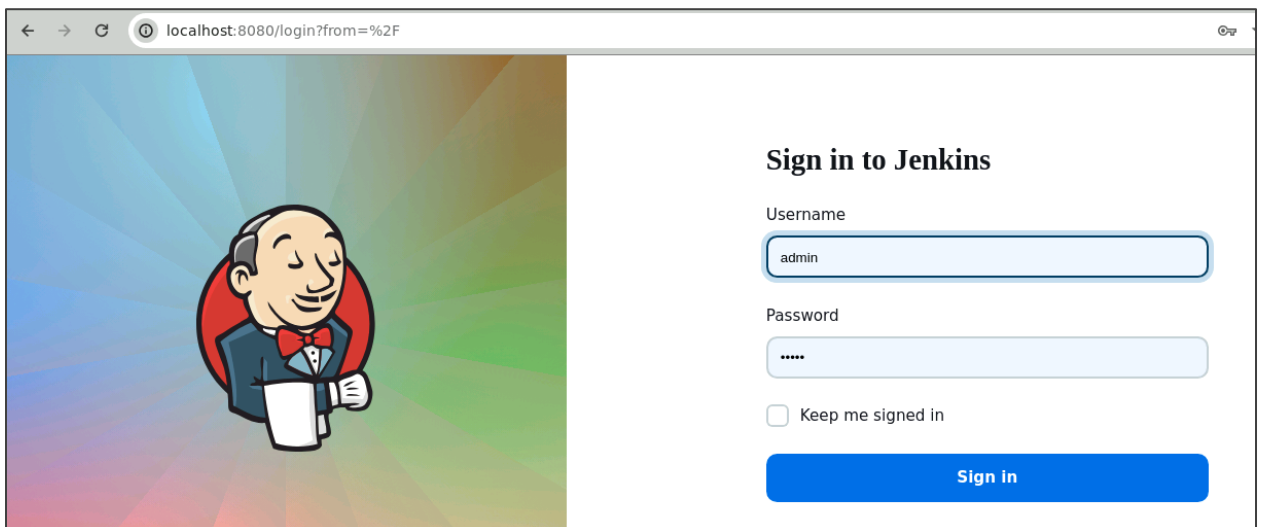
Prerequisites: You need to have a Jenkins up and running.

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

A screenshot of a web browser showing the Jenkins login page. The browser's address bar displays 'localhost:8080/login?from=%2F'. The page features a large illustration of Mr. Jenkins, a cartoon character with a balding head, wearing a blue suit, a red bow tie, and holding a white rolled-up document. To the right of the illustration is the 'Sign in to Jenkins' form. The form includes a 'Username' field with 'admin' entered, a 'Password' field with masked characters '.....', an unchecked checkbox for 'Keep me signed in', and a blue 'Sign In' button at the bottom.

← → ↻ 📄 localhost:8080/login?from=%2F

Sign in to Jenkins

Username
admin

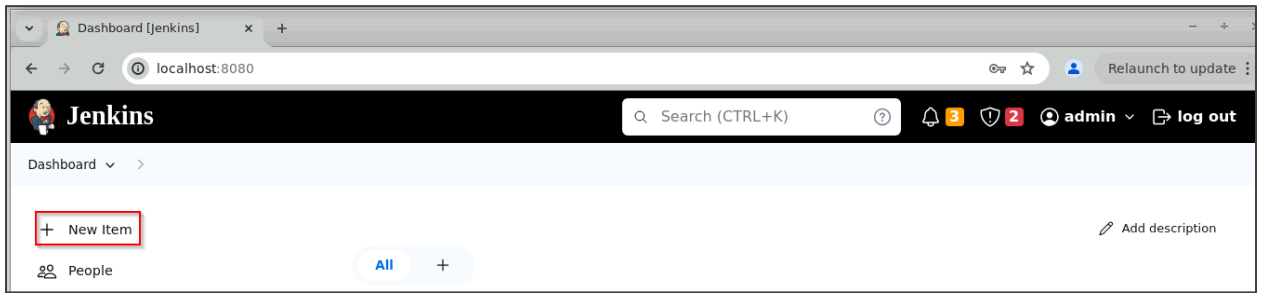
Password
.....

☐ Keep me signed in

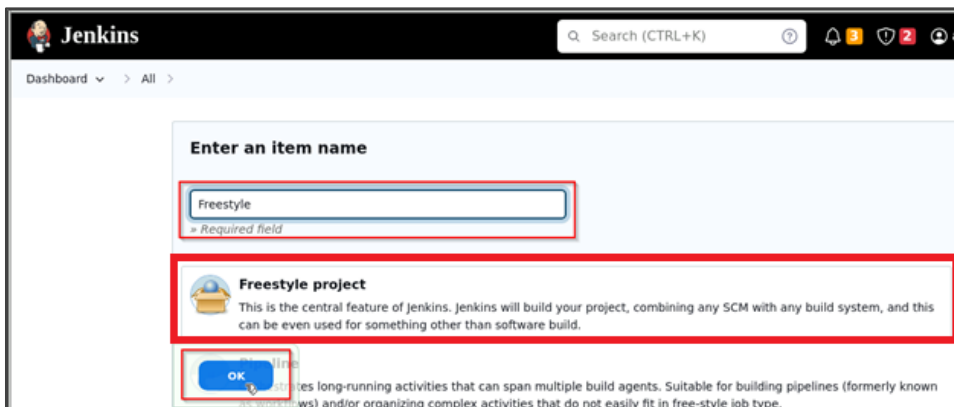
Sign In

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

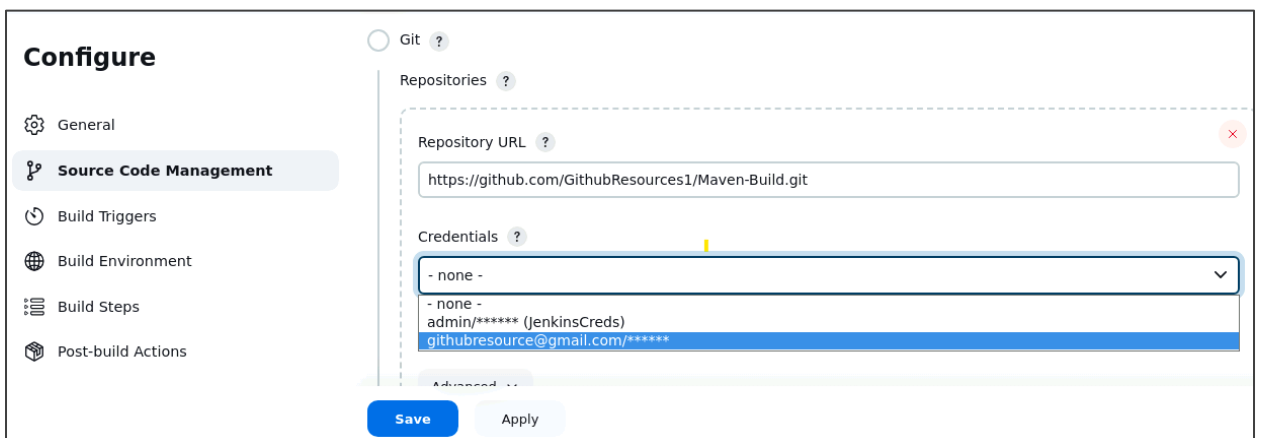
1.2 In the Jenkins dashboard, click on **New Item**



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

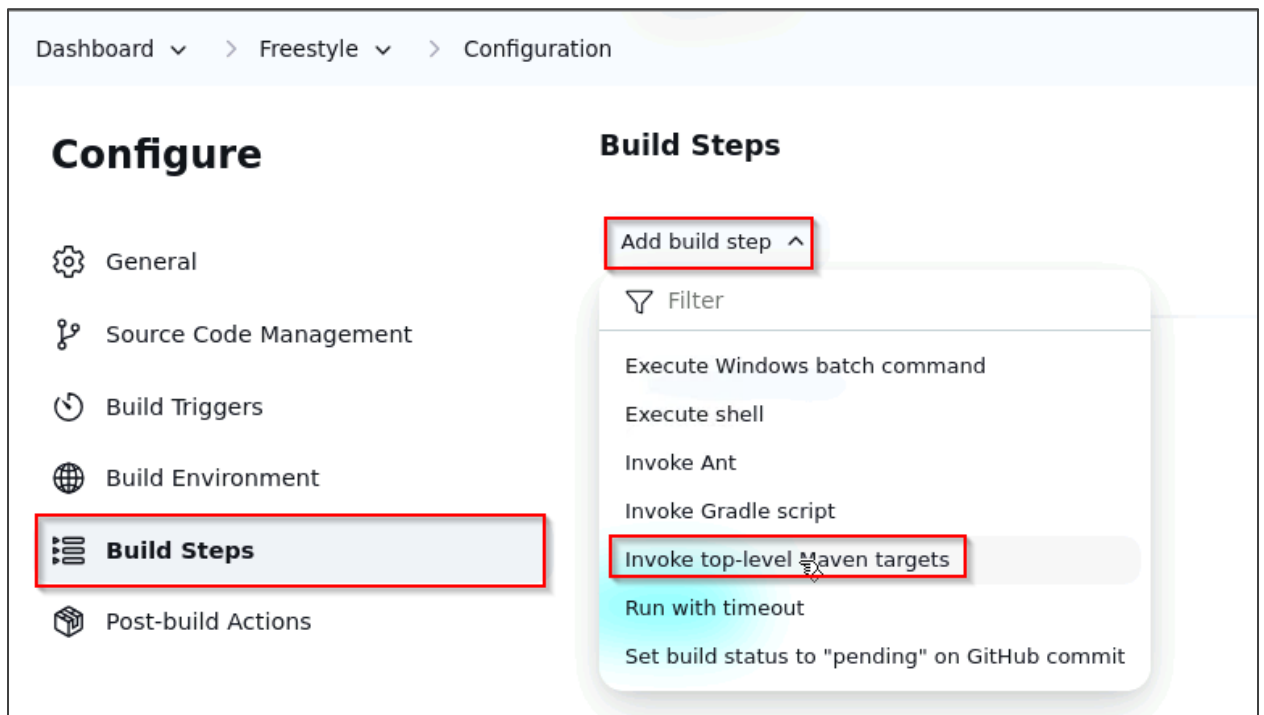


1.4 Now, in the Configure page, navigate to **Source Code Management** in the left navigation bar, select **Git**, and then provide the Git repository URL

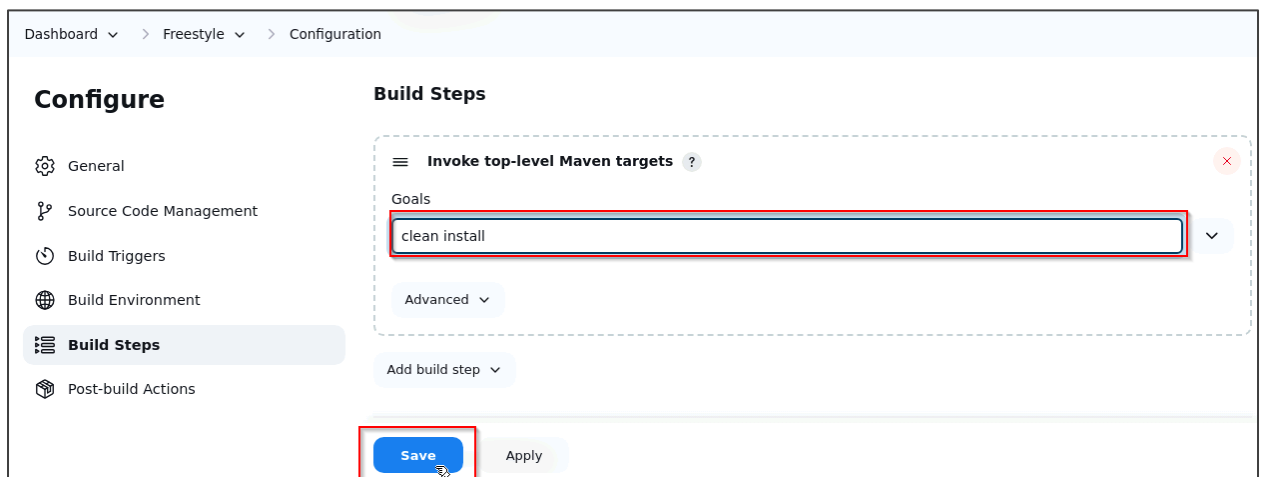


Note: Here, the repository URL is <https://github.com/GithubResources1/Maven-Build.git>.

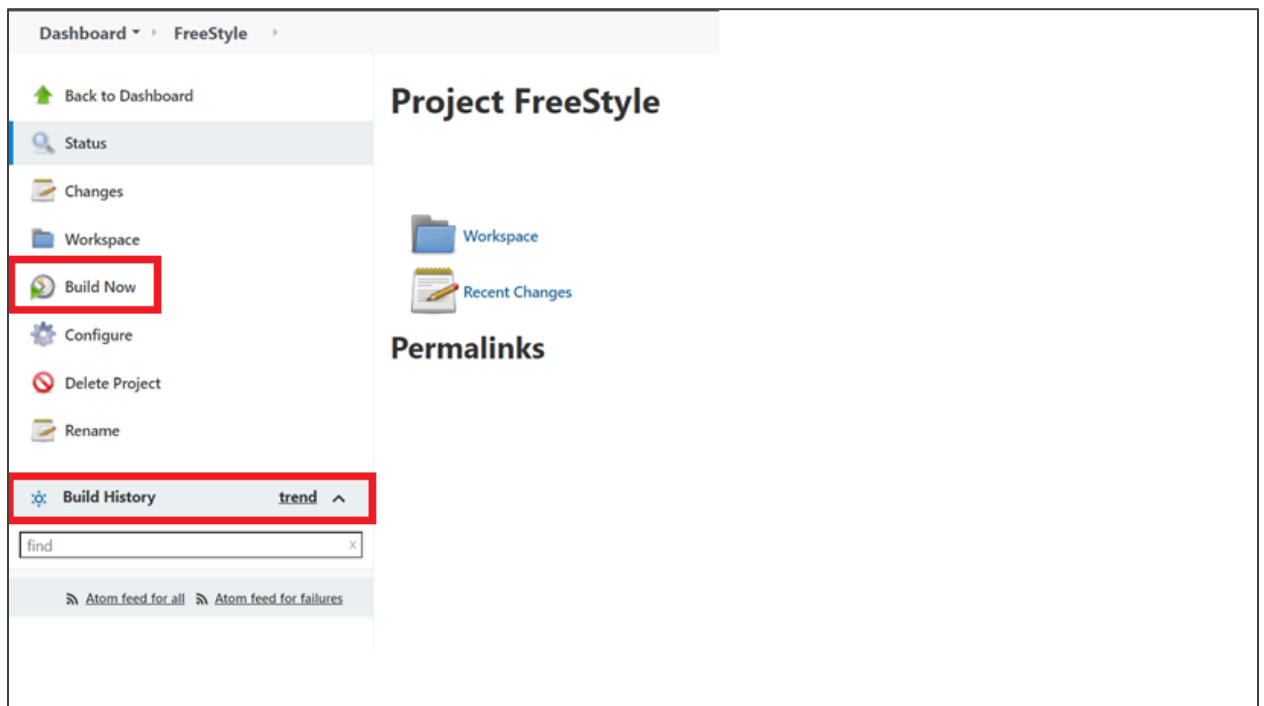
1.5 Now, navigate to **Build Steps**, click on **Add build step**, and then select the option **Invoke top-level Maven targets**



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.



```
01:26:18 [01:34mINFO]m
01:26:18 [01:34mINFO]m Results:
01:26:18 [01:34mINFO]m
01:26:18 [01:34mINFO]m [1,32mTests run: 1, Failures: 0, Errors: 0, Skipped: 0]m
01:26:18 [01:34mINFO]m
01:26:18 [01:34mINFO]m
01:26:18 [01:34mINFO]m [1m--- [0;32maven-jar-plugin:3.0.2:jar[0m [1m(default-jar)[0m @ [1mdemo-docker[0;1m ---[0m
01:26:19 [01:34mINFO]m Building jar: /var/lib/jenkins/workspace/FreeStyle/target/demo-docker-0.0.1-SNAPSHOT.jar
01:26:19 [01:34mINFO]m
01:26:19 [01:34mINFO]m [1m--- [0;32mspring-boot-maven-plugin:2.0.5.RELEASE:repackage[0m [1m(default)[0m @ [1mdemo-docker[0;1m ---[0m
01:26:20 [01:34mINFO]m
01:26:20 [01:34mINFO]m [1m--- [0;32maven-install-plugin:2.5.2:install[0m [1m(default-install)[0m @ [1mdemo-docker[0;1m ---[0m
01:26:20 [01:34mINFO]m Installing /var/lib/jenkins/workspace/FreeStyle/target/demo-docker-0.0.1-SNAPSHOT.jar to /var/lib/jenkins/.m2/repository/com/example/demo-docker/0.0.1-SNAPSHOT/demo-docker-0.0.1-SNAPSHOT.jar
01:26:20 [01:34mINFO]m Installing /var/lib/jenkins/workspace/FreeStyle/pom.xml to /var/lib/jenkins/.m2/repository/com/example/demo-docker/0.0.1-SNAPSHOT/demo-docker-0.0.1-SNAPSHOT.pom
01:26:20 [01:34mINFO]m [1m-----[0m
01:26:20 [01:34mINFO]m [1m]BUILD SUCCESS[0m
01:26:20 [01:34mINFO]m
01:26:20 [01:34mINFO]m Total time: 14.415 s
01:26:20 [01:34mINFO]m Finished at: 2021-07-21T19:56:20Z
01:26:20 [01:34mINFO]m [1m-----[0m
01:26:20 Finished: SUCCESS
```

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.