

Lab Exercise 8

Setting up Maven Build Job in Jenkins

Name: Viraj Bhidola

Sap id: 500121825

Enrolment no.: R2142231731

Batch 2 DevOps

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

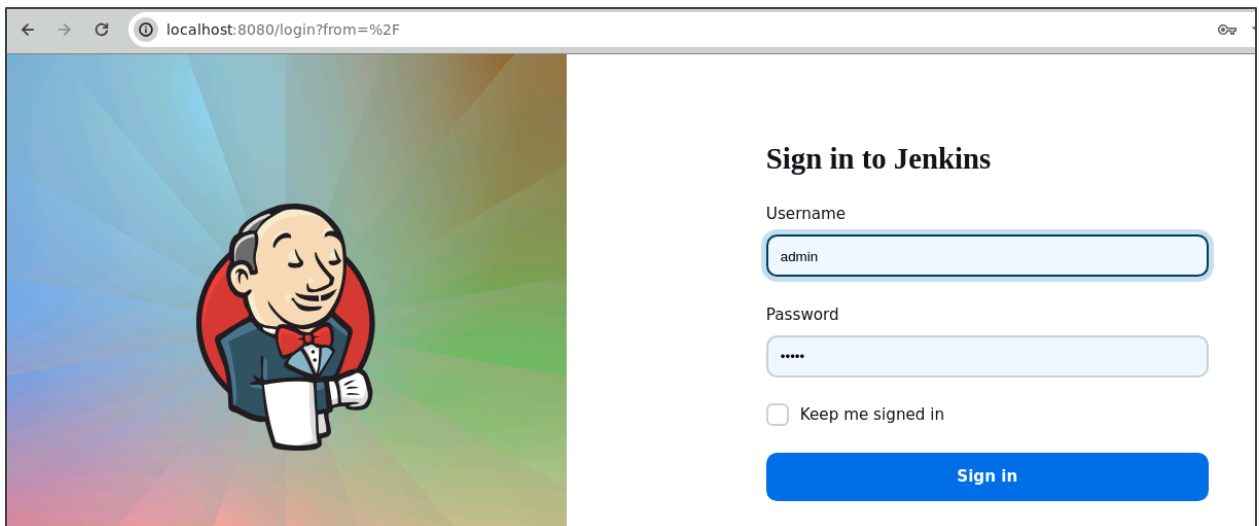
Prerequisites: You need to have a Jenkins up and running.

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 window showing the Jenkins login page. The browser's address bar displays 'localhost:8080/login?from=%2F'. The page features a colorful background on the left with a cartoon character of a man in a tuxedo. On the right, there is a 'Sign in to Jenkins' section with a 'Username' field containing 'admin', a 'Password' field with masked characters, a 'Keep me signed in' checkbox, and a blue 'Sign In' button.

← → ↻ 🔒 localhost:8080/login?from=%2F

Sign in to Jenkins

Username
admin

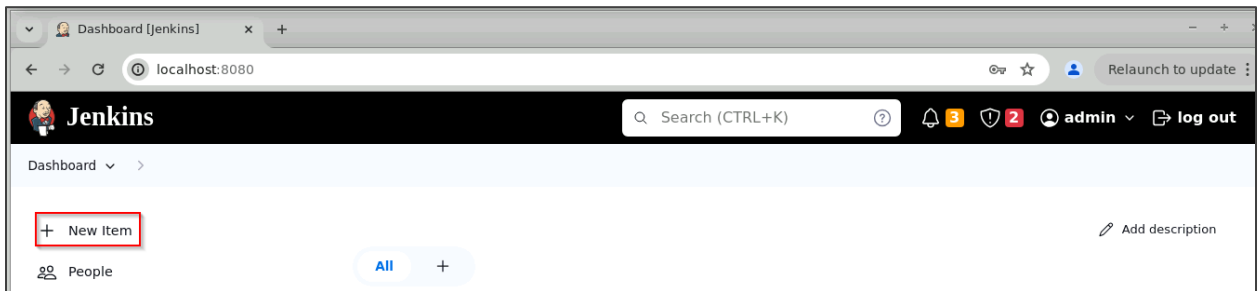
Password
.....

☐ Keep me signed in

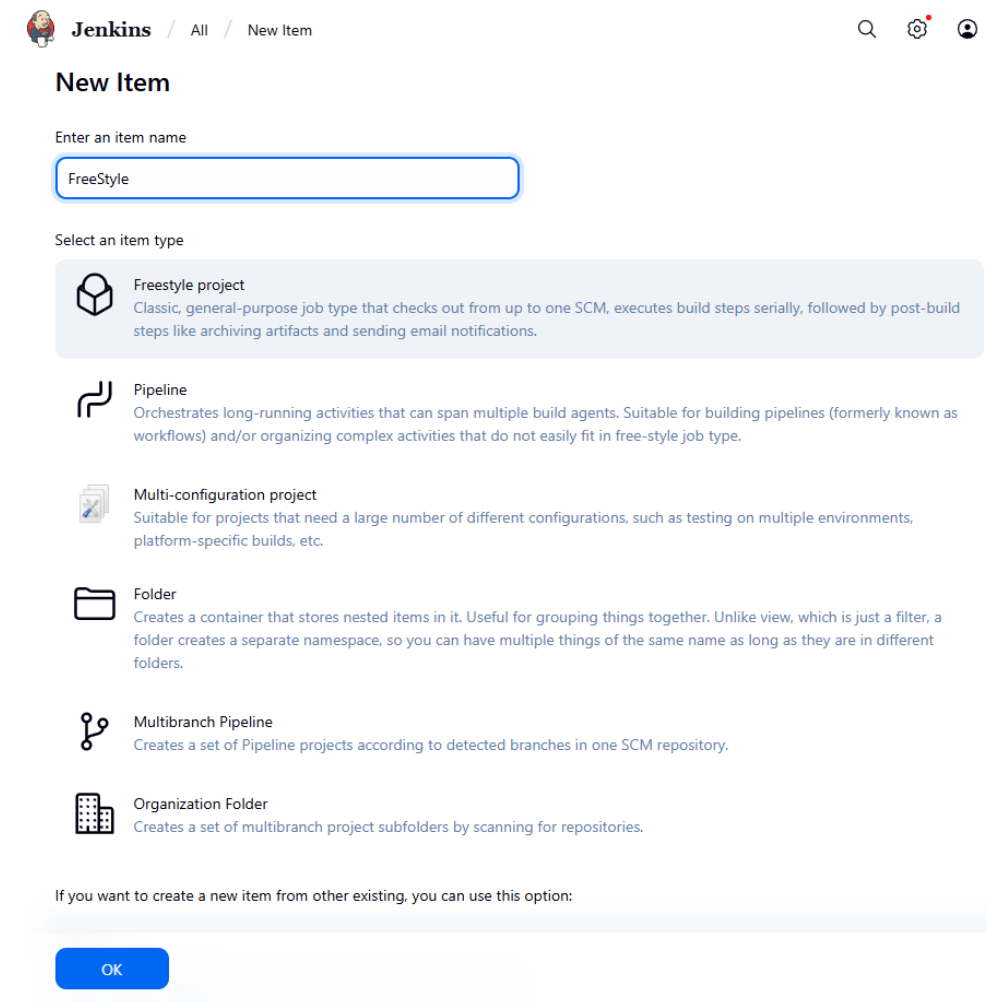
Sign In

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

 **Jenkins** / FreeStyle / Configuration

Configure

General

Source Code Management

Triggers

Environment

Build Steps

Post-build Actions

☐ Execute concurrent builds if necessary ?

Advanced ▼

Source Code Management

Connect and manage your code repository to automatically pull the latest code for your builds.

☐ None

☒ Git ?

Repositories ?

Repository URL ?

https://github.com/hkshitesh/DEVSECOPS-MAVEN-REPO.git

Credentials ?

- none -

+ Add

Advanced ▼

Add Repository

Branches to build ?

Branch Specifier (blank for 'any') ?

Save

Apply

1.5

0

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

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

The screenshot shows the Jenkins FreeStyle Configuration page. The left sidebar contains a 'Configure' section with a list of tabs: General, Source Code Management, Triggers, Environment (selected), Build Steps, and Post-build Actions. The main content area is divided into two sections: 'Environment' and 'Build Steps'. The 'Environment' section has a description and six checkboxes: 'Delete workspace before build starts', 'Use secret text(s) or file(s)', 'Add timestamps to the Console Output', 'Inspect build log for published build scans', 'Terminate a build if it's stuck', and 'With Ant'. The 'Build Steps' section has a description and an 'Add build step' button. A dropdown menu is open from the 'Add build step' button, showing a list of build steps: 'Execute Windows batch command', 'Execute shell', 'Invoke Ant', 'Invoke Gradle script', 'Invoke top-level Maven targets' (highlighted), 'Run with timeout', and 'Set build status to "pending" on GitHub commit'. The bottom right corner of the page shows 'REST API' and 'Jenkins 2.516.2'.

Jenkins / FreeStyle / Configuration

Configure

- General
- Source Code Management
- Triggers
- Environment**
- Build Steps
- Post-build Actions

Environment

Configure settings and variables that define the context in which your build runs, like credentials, paths, and global parameters.

- ☐ Delete workspace before build starts
- ☐ Use secret text(s) or file(s) ?
- ☐ Add timestamps to the Console Output
- ☐ Inspect build log for published build scans
- ☐ Terminate a build if it's stuck
- ☐ With Ant ?

Build Steps

Automate your build process with ordered tasks like code compilation, testing, and deployment.

Add build step ^

- Filter
- Execute Windows batch command
- Execute shell
- Invoke Ant
- Invoke Gradle script
- Invoke top-level Maven targets**
- Run with timeout
- Set build status to "pending" on GitHub commit

REST API Jenkins 2.516.2

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

The screenshot shows the Jenkins 'Configure' page for a FreeStyle project. The left sidebar contains navigation links: General, Source Code Management, Triggers, Environment, Build Steps (selected), and Post-build Actions. The main content area is titled 'Build Steps' and includes a description: 'Automate your build process with ordered tasks like code compilation, testing, and deployment.' A build step named 'Invoke top-level Maven targets' is configured with 'Maven Version' set to 'Maven-3.9.11' and 'Goals' set to 'clean install'. Below this, there is an 'Advanced' dropdown and an 'Add build step' button. The 'Post-build Actions' section is also visible, with an 'Add post-build action' button. At the bottom, there are 'Save' and 'Apply' buttons. The footer indicates 'REST API' and 'Jenkins 2.516.2'.

Jenkins / FreeStyle / Configuration

Configure

- General
- Source Code Management
- Triggers
- Environment
- Build Steps**
- Post-build Actions

Build Steps

Automate your build process with ordered tasks like code compilation, testing, and deployment.

Invoke top-level Maven targets ?

Maven Version

Maven-3.9.11

Goals

clean install

Advanced

Add build step

Post-build Actions

Define what happens after a build completes, like sending notifications, archiving artifacts, or triggering other jobs.

Add post-build action

Save Apply

REST API Jenkins 2.516.2

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.

The top screenshot shows the Jenkins web interface for a project named 'FreeStyle'. The left sidebar contains navigation links: Status, Changes, Workspace, Build Now, Configure, Delete Project, and Rename. The main area displays 'FreeStyle' with a 'Permalinks' section and a 'Builds' table showing a single build #1 from 10:59 PM.

The bottom screenshot shows the 'Console Output' for build #1. The output text is as follows:

```
Downloaded from central:
https://repo.maven.apache.org/maven2/org/apache/maven/resolver/maven-
resolver-api/1.9.18/maven-resolver-api-1.9.18.jar (157 kB at 1.6
MB/s)
Progress (1): 147/193 kB
Progress (1): 164/193 kB
Progress (1): 180/193 kB
Progress (1): 193 kB

Downloaded from central:
https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-
utils/4.0.1/plexus-utils-4.0.1.jar (193 kB at 1.2 MB/s)
[INFO] Installing
C:\ProgramData\Jenkins\.jenkins\workspace\FreeStyle\pom.xml to
C:\WINDOWS\system32\config\systemprofile\.m2\repository\upes\devops\upes-
SNAPSHOT\upes.devops-0.0.1-SNAPSHOT.pom
[INFO] Installing
C:\ProgramData\Jenkins\.jenkins\workspace\FreeStyle\target\upes-
calc.jar to
C:\WINDOWS\system32\config\systemprofile\.m2\repository\upes\devops\upes-
SNAPSHOT\upes.devops-0.0.1-SNAPSHOT.jar
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 52.171 s
[INFO] Finished at: 2025-08-27T23:19:59+05:30
[INFO] -----
Finished: SUCCESS
```

At the bottom right of the console output window, it says 'REST API Jenkins 2.516.2'.

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.