.

Lab Exercise 8

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

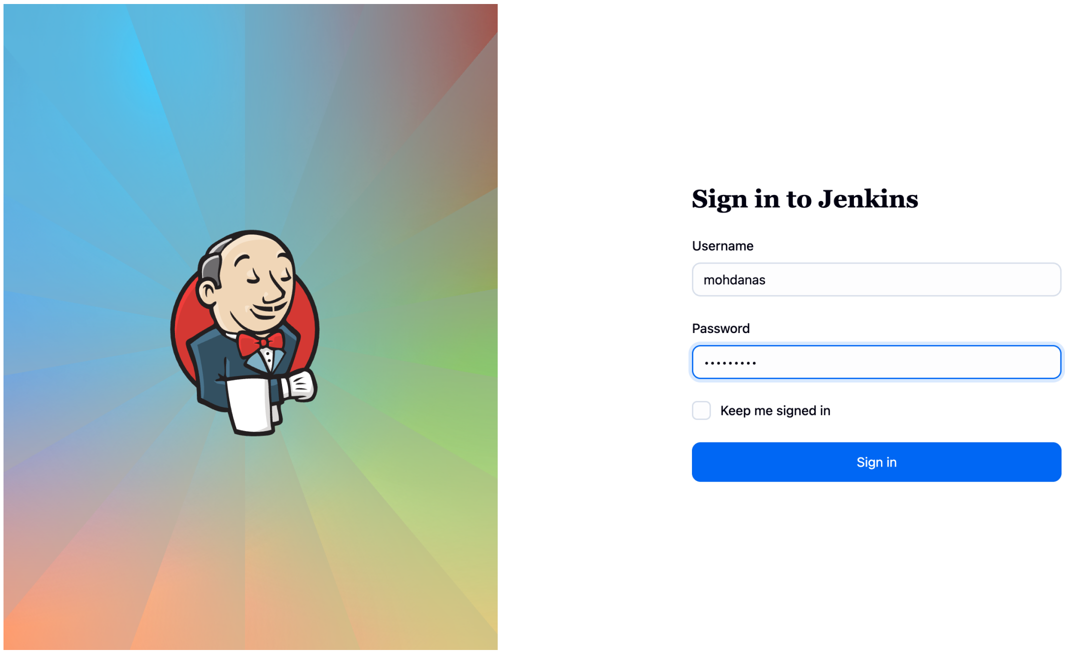
Prerequisites: You need to have a Jenkins up and running.

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. Log in to Jenkins using your credentials



Sign in to Jenkins

Username: mohdanas

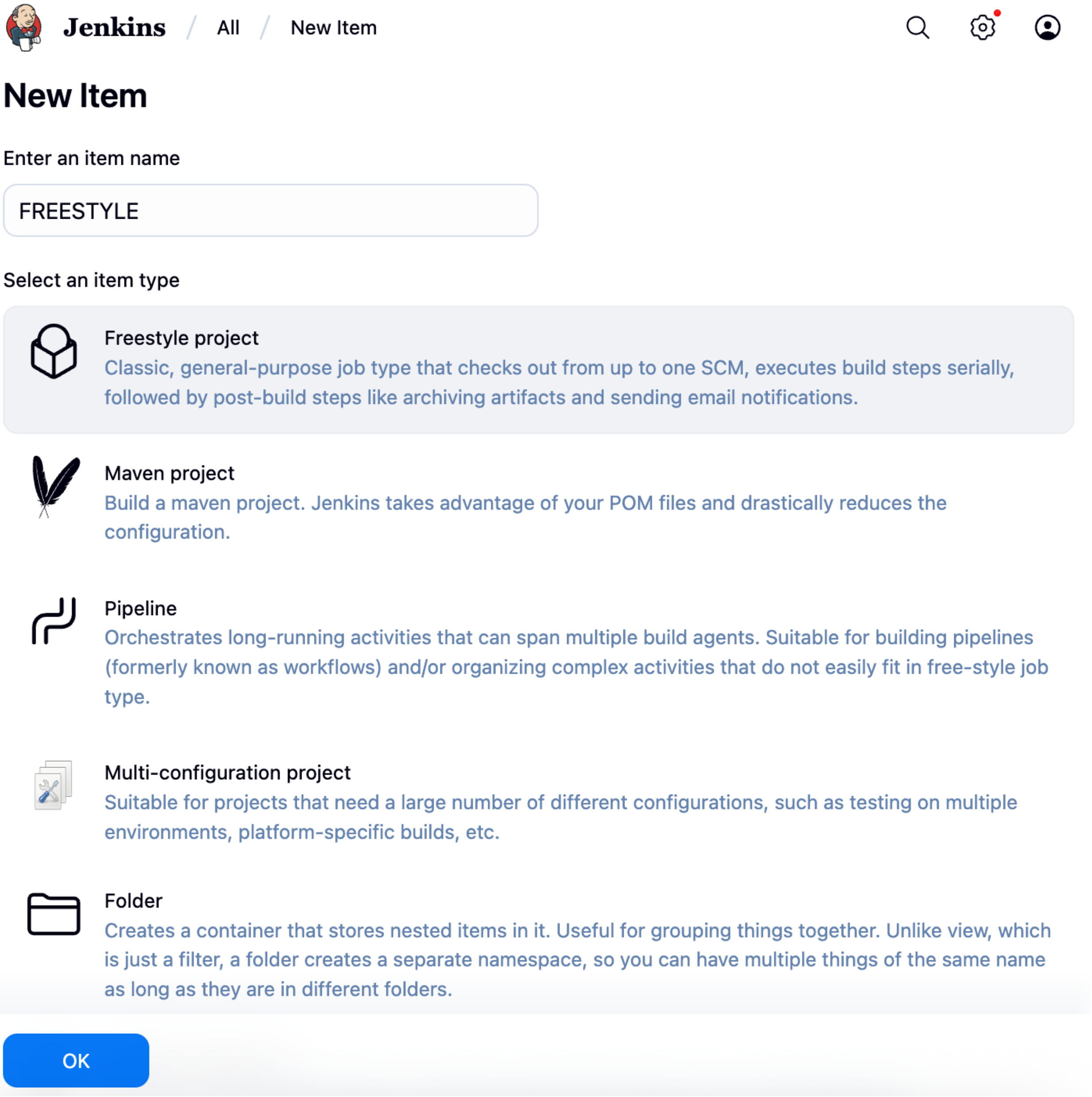
Password: •••••••••

□ Keep me signed in

.

2. In the Jenkins dashboard, click on **New Item**

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



New Item

Enter an item name

FREESTYLE

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.

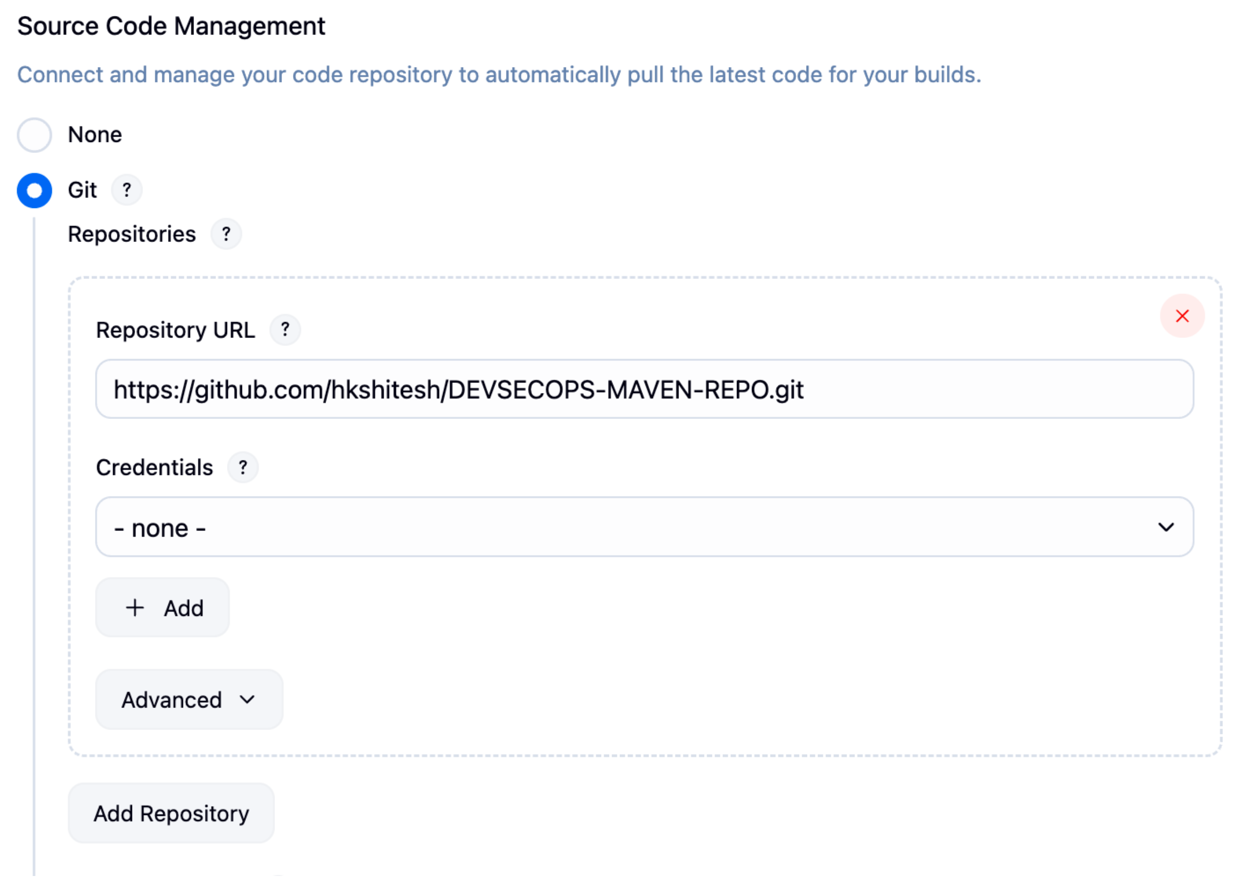
Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

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 folders.

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

.



Source Code Management

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

○ None

● Git ?

Repositories ?

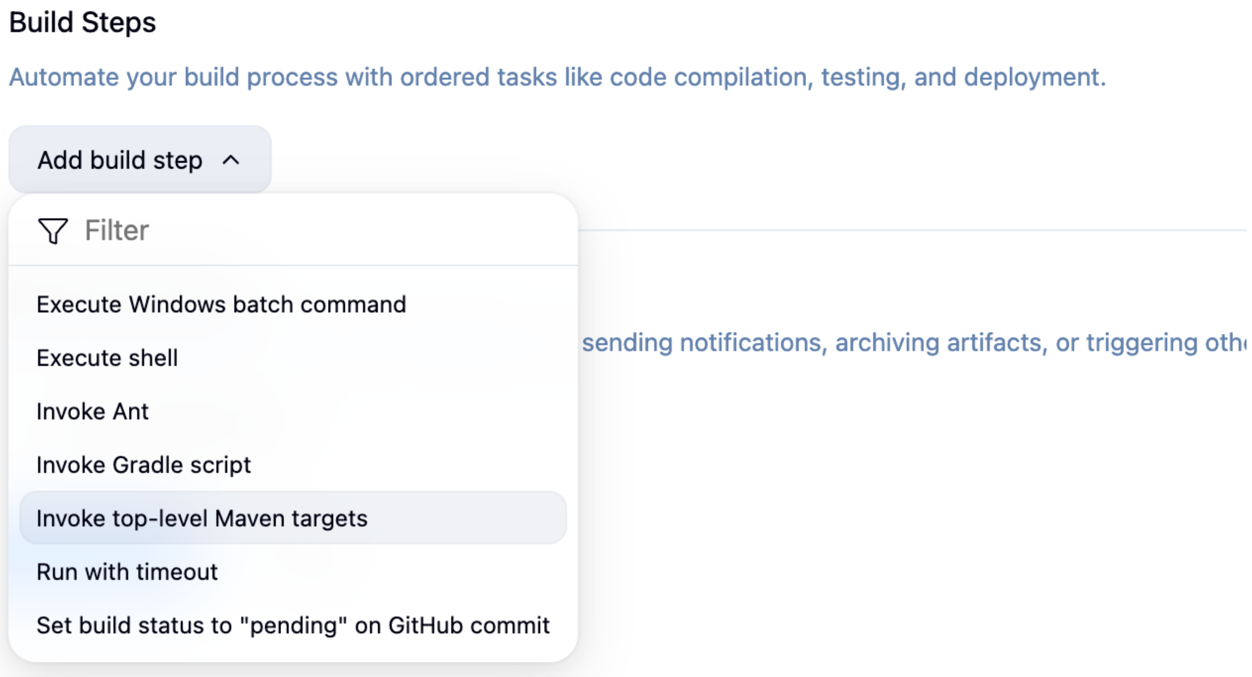
Repository URL ?

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

Credentials ?

- none -

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



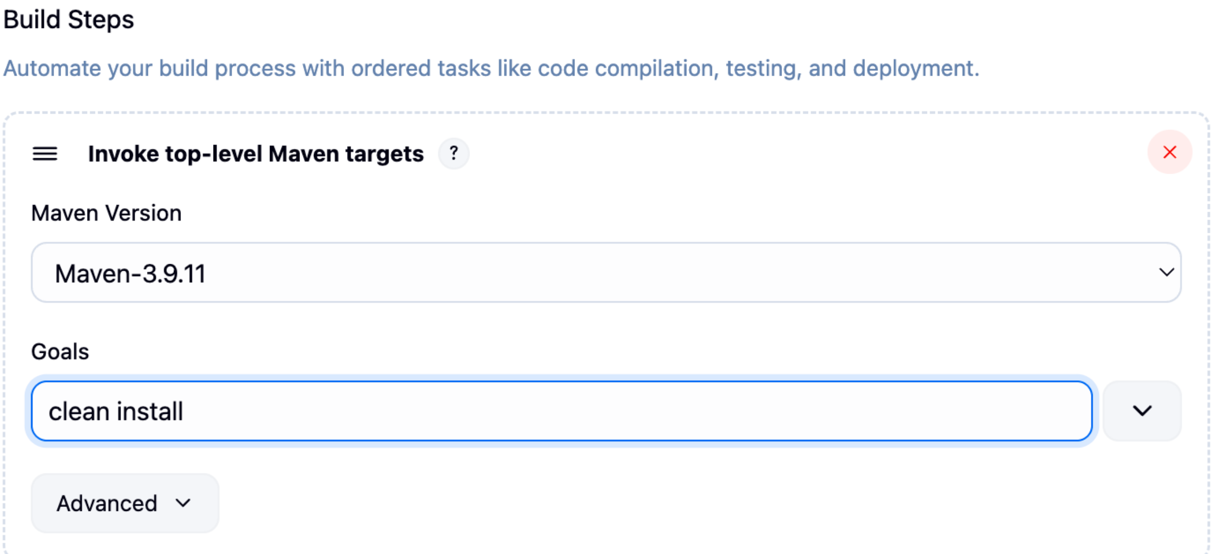
Build Steps

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

sending notifications, archiving artifacts, or triggering oth

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

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



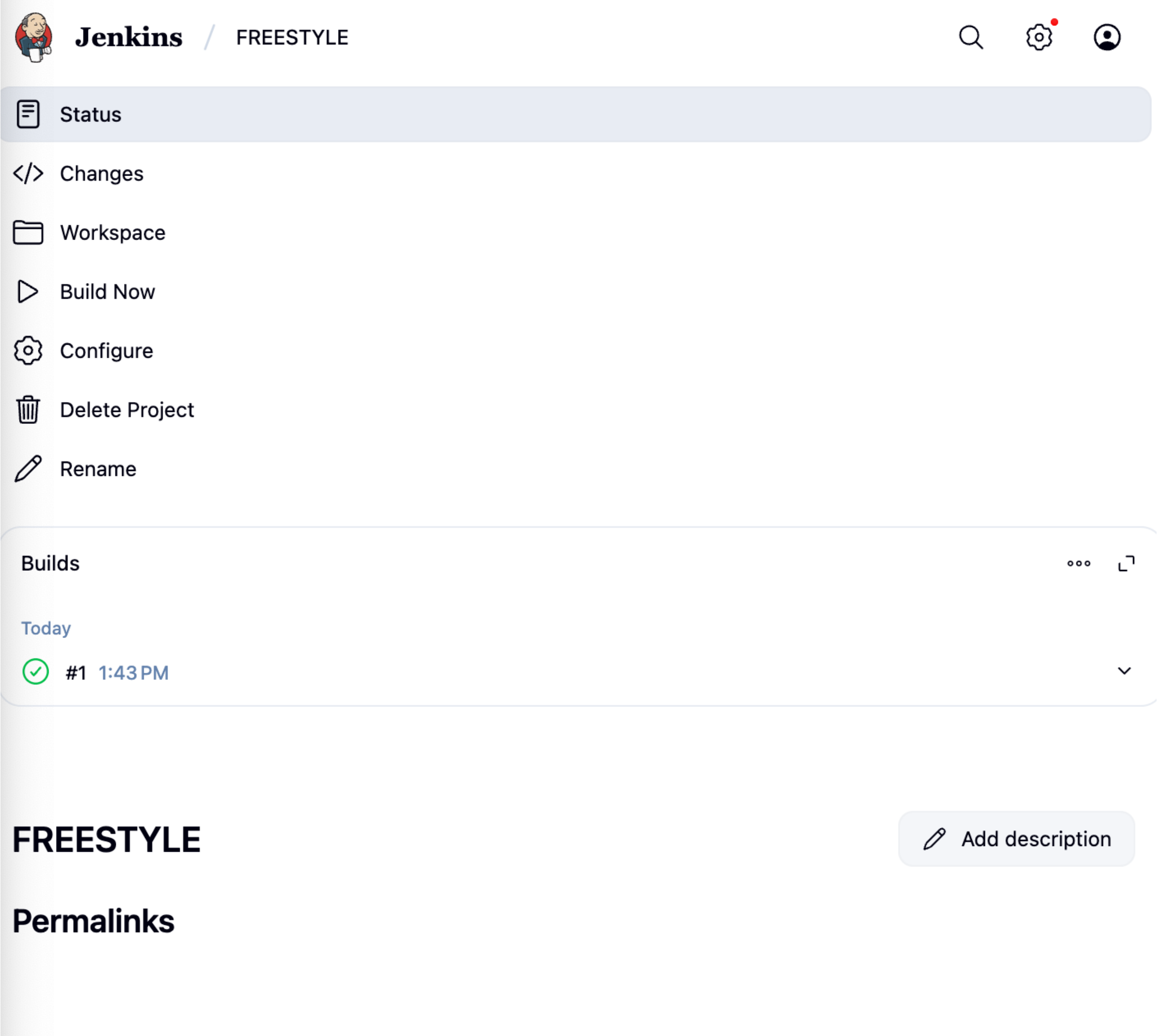
Build Steps

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



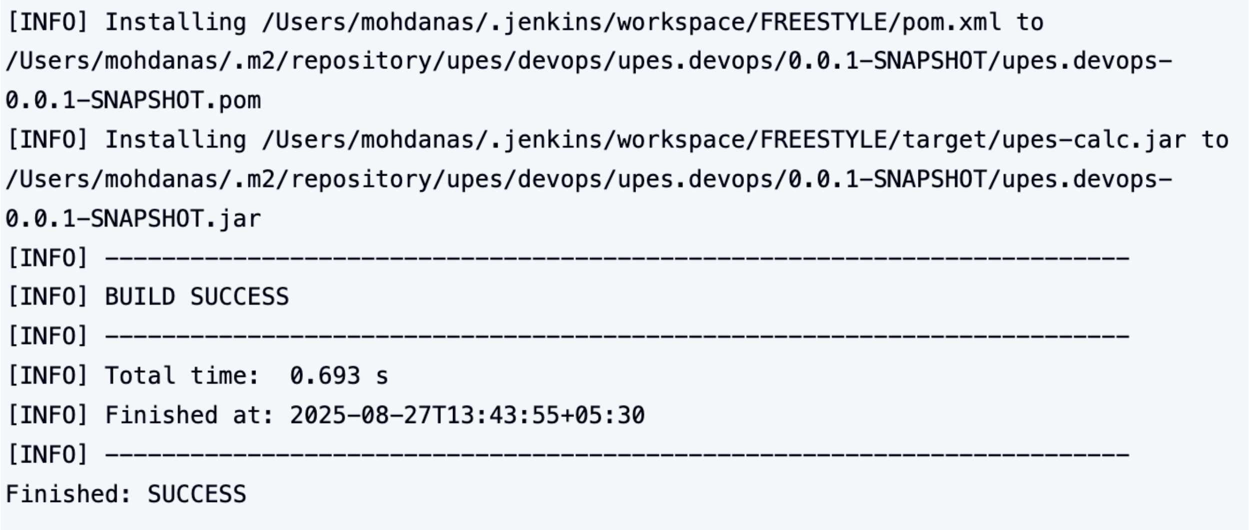
#1 1:43 PM

FREESTYLE

Permalinks

.

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.



```
[INFO] Installing /Users/mohdanas/.jenkins/workspace/FREESTYLE/pom.xml to
/Users/mohdanas/.m2/repository/upes/devops/upes.devops/0.0.1-SNAPSHOT/upes.devops-
0.0.1-SNAPSHOT.pom
[INFO] Installing /Users/mohdanas/.jenkins/workspace/FREESTYLE/target/upes-calc.jar to
/Users/mohdanas/.m2/repository/upes/devops/upes.devops/0.0.1-SNAPSHOT/upes.devops-
0.0.1-SNAPSHOT.jar
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  0.693 s
[INFO] Finished at: 2025-08-27T13:43:55+05:30
[INFO] ------------------------------------------------------------------------
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.