

# Building Java Projects with Maven

Varshni D S

Date: 07.02.2025

## Introduction

Maven is a project management tool for Java applications. Just like a manager oversees tasks, resources, and deadlines, Maven efficiently handles dependencies, builds, tests, and deployments.

## Step 1: Install Java and Maven on Ubuntu

Ensure Java and Maven are installed on your Ubuntu system before proceeding.

jdk-23 will receive updates under these terms, and jdk-23.0.2, which is made superseded by jdk-24.

Linux	macOS	Windows
Product/file description		
File size		
Download		
x64 Compressed Archive	228.77 MB	<a href="https://download.oracle.com/java/23/latest/jdk-23_windows-x64_bin.zip">https://download.oracle.com/java/23/latest/jdk-23_windows-x64_bin.zip</a> (sha256)
x64 Installer	205.27 MB	<a href="https://download.oracle.com/java/23/latest/jdk-23_windows-x64_bin.exe">https://download.oracle.com/java/23/latest/jdk-23_windows-x64_bin.exe</a> (sha256)
x64 MSI Installer	204.02 MB	<a href="https://download.oracle.com/java/23/latest/jdk-23_windows-x64_bin.msi">https://download.oracle.com/java/23/latest/jdk-23_windows-x64_bin.msi</a> (sha256)

```
ubuntu@Harz-PC:~$ java -version
openjdk version "17.0.14" 2025-01-21
OpenJDK Runtime Environment (build 17.0.14+7-Ubuntu-124.04)
OpenJDK 64-Bit Server VM (build 17.0.14+7-Ubuntu-124.04, mixed mode, sharing)
ubuntu@Harz-PC:~$ mvn -version
Apache Maven 3.9.9 (8e8579a9e76f7d015ee5ec7bfc97d260186937)
Maven home: /mnt/c/Program Files/apache-maven-3.9.9-bin/apache-maven-3.9.9
Java version: 17.0.14, vendor: Ubuntu, runtime: /usr/lib/jvm/java-17-openjdk-amd64
Default locale: en, platform encoding: UTF-8
OS name: "linux", version: "5.15.167.4-microsoft-standard-wsl2", arch: "amd64", family: "unix"
ubuntu@Harz-PC:~$
```

```
ubuntu@Harz-PC:~$ readlink -f $(which java)
/usr/lib/jvm/java-17-openjdk-amd64/bin/java
```

Use default maven global settings

JDK installations

JDK installations ^ Edited

Add JDK

**JDK**

Name

java

JAVA\_HOME

/usr/lib/jvm/java-17-openjdk-amd64

☐ Install automatically ?

Save Apply

```
ubuntu@Harz-PC:~$ readlink -f $(which mvn)
/usr/share/maven/bin/mvn
```

Maven installations

Maven installations ^ Edited

Add Maven

**Maven**

Name

maven\_task5

MAVEN\_HOME

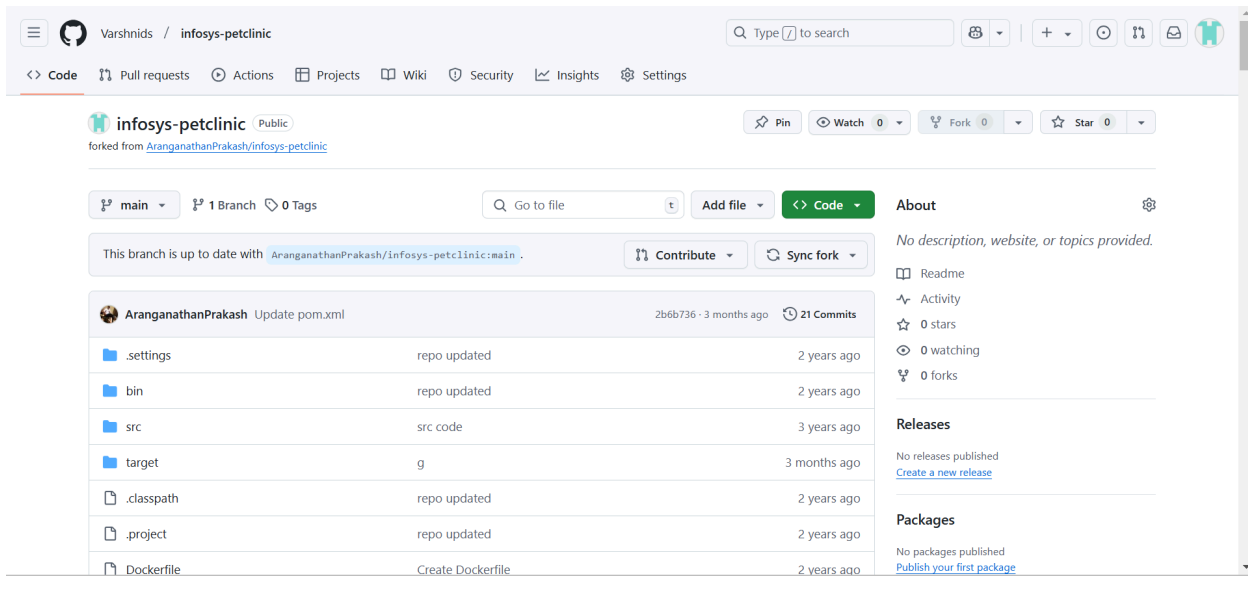
/usr/share/maven

☐ Install automatically ?

Save Apply

## Step 2: Fork the eKart Repository on GitHub

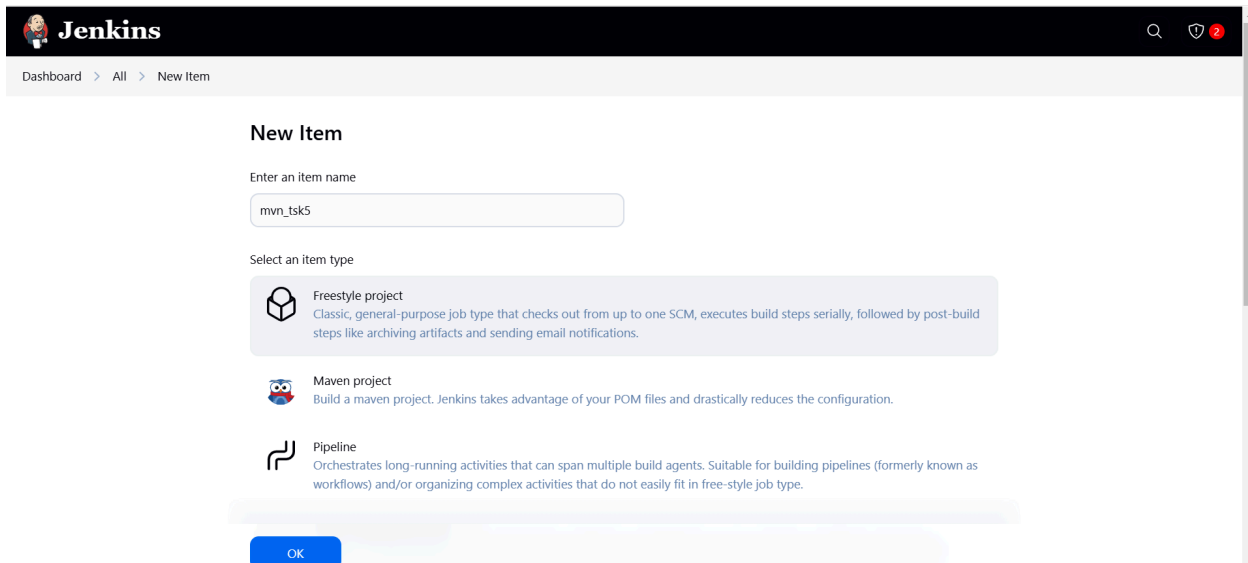
Fork the repository to your own GitHub account for modification and integration.



## Step 3: Configure Jenkins

### Create a New Job in Jenkins:

1. Open Jenkins in a web browser.
2. Click **New Item** → Select **Freestyle Project**.
3. Name it **Maven\_task5** and click **OK**.



### Configure the Job:

- **Build Tools Setup:**

- Navigate to **Global Tool Configuration**.
- Ensure Java and Maven are added if not already configured.
- **Set Up GitHub Repository:**
  - Under **Source Code Management**, select **Git**.
  - Paste the forked repository URL.
  - Set the branch to **main**.

Dashboard > mvn\_tsk5 > Configuration

### Configure

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

#### 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/Varshnids/infosys-petclinic.git

Credentials ?

- none -

+ Add

Advanced ▾

Add Repository

Save Apply

Dashboard > mvn\_tsk5 > Configuration

### Configure

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

#### Source Code Management

Advanced ▾

Add Repository

Branches to build ?

Branch Specifier (blank for 'any') ?

\*/main

Add Branch

Repository browser ?

(Auto)

Additional Behaviours

Add ▾

Save Apply

- **Add Build Command:**
  - Go to **Build** → **Add Build Step** → Select **Invoke top-level Maven targets**.

Dashboard > mvn\_task5 > Configuration

Configure

General

Source Code Management

Triggers

Environment

Build Steps

Post-build Actions

Triggers

Set up automated actions that start your build based on specific events, like code changes or scheduled times.

☐ Build after other projects are built ?

☐ Build periodically ?

☒ GitHub hook trigger for GITScm polling ?

☐ Poll SCM ?

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

Save

Apply

Dashboard > mvn\_task5 > Configuration

Configure

General

Source Code Management

Triggers

Environment

Build Steps

Post-build Actions

With Ant ?

JDK

java

Build Steps

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

≡ Invoke top-level Maven targets ?

Maven Version

maven\_task5

Goals

clean package -DskipTests

Advanced ▾

Save

Apply

Enter the command:  
clean package -DskipTests

- ☐
- ☐ Click **Build Now**.

## Step 4: Navigate to Jenkins Workspace

Move to the Jenkins workspace directory:  
cd /var/lib/jenkins/workspace

1.

List the available projects:

```
ls
```

2.

Navigate to the specific project folder:

```
cd Maven_task5/target
```

3.

Verify the generated artifacts (e.g., `.jar` file):

```
ls
```

4.

---

## Step 5: Check Docker Image and Kubernetes Deployment

### Build and Push Docker Image:

```
docker build -t test -f docker/Dockerfile
```

```
docker push varshnids057/sample
```

### Deploy with Kubernetes:

```
kubectl create deployment maven --image=test --port=80
```

```
kubectl expose deployment maven --type=NodePort --port=80 --target-port=8070
```

### Verify Deployment:

```
docker images | grep nivethitha24/mave # Check Docker image
```

```
kubectl get pods # List running pods
```

```
minikube service maven # Get service URL
```

Dashboard > mvn\_tsk5 >

Status

</> Changes

Workspace

Build Now

Configure

Delete Project

GitHub Hook Log

Rename

mvn\_tsk5

Permalinks

- Last build (#3), 3 min 4 sec ago
- Last stable build (#3), 3 min 4 sec ago
- Last successful build (#3), 3 min 4 sec ago
- Last completed build (#3), 3 min 4 sec ago

Add description

Builds

Filter

Today

#3 8:16 am

```
ubuntu@Harz-PC:~$ minikube start
minikube v1.35.0 on Ubuntu 24.04 (amd64)
Using the docker driver based on existing profile
Starting "minikube" primary control-plane node in "minikube" cluster
Pulling base image v0.0.46 ...
Updating the running docker "minikube" container ...
Failing to connect to https://registry.k8s.io/ from inside the minikube container
To pull new external images, you may need to configure a proxy: https://minikube.sigs.k8s.io/docs/reference/networking/proxy/
Preparing Kubernetes v1.32.0 on Docker 27.4.1 ...
Verifying Kubernetes components...
  Using image gcr.io/k8s-minikube/storage-provisioner:v5
Enabled addons: storage-provisioner, default-storageclass
Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
```

```
ubuntu@Harz-PC:~$ minikube service test-service
|-----|
| NAMESPACE | NAME       | TARGET PORT | URL                |
|-----|
| default   | test-service | http/80      | http://192.168.49.2:31874 |
|-----|
Starting tunnel for service test-service.
|-----|
| NAMESPACE | NAME       | TARGET PORT | URL                |
|-----|
| default   | test-service |             | http://127.0.0.1:41531 |
|-----|
Opening service default/test-service in default browser...
http://127.0.0.1:41531
! Because you are using a Docker driver on linux, the terminal needs to be open to run it.
```

## Jenkins Configuration & Output



OnePlus 9 5G



5.4 inch display

399

Add to Cart

Iphone 13 mini



5.4 inch display

399

Add to Cart

Samsung s21 ultra



5.4 inch display

399

Add to Cart

xiomi mi 11



5.4 inch display

399

Add to Cart

OnePlus 9 5G



Iphone 13 mini

Samsung s21 ultra



xiomi mi 11

