Building Java Projects with Maven - Task 5
Ragul R R
Date: 07.02.2025

Overview

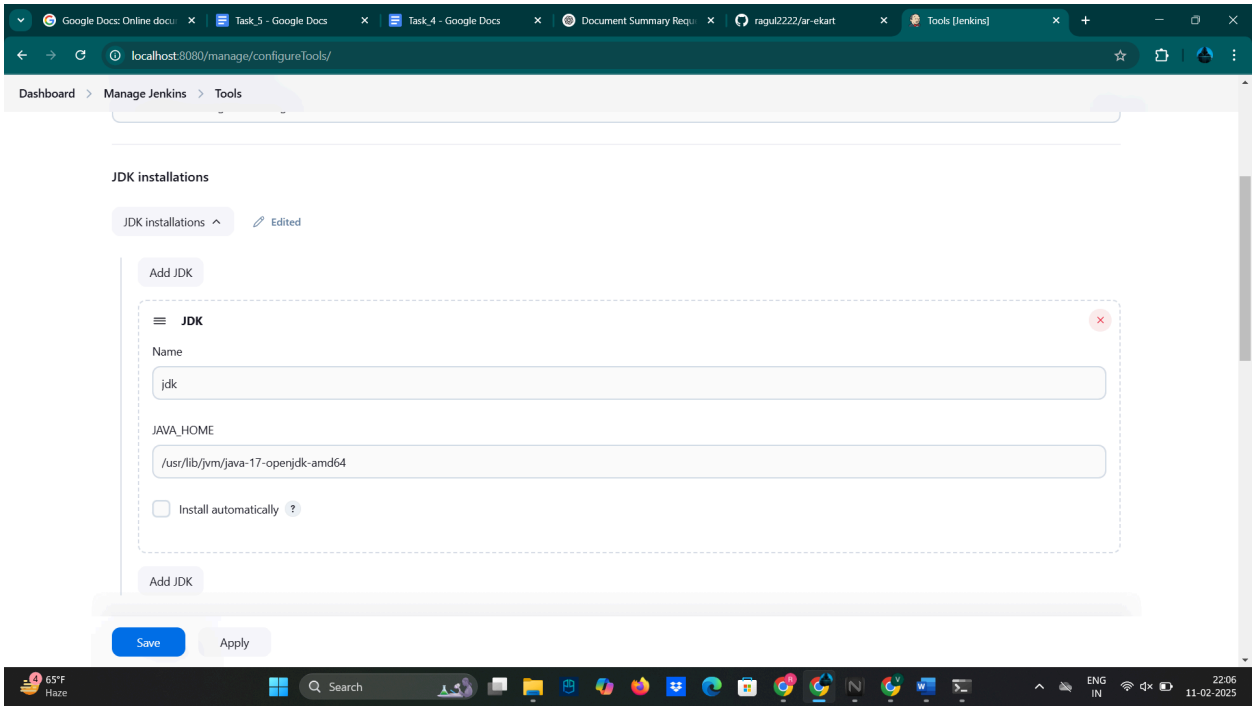
Maven is a robust project management tool designed for Java applications. Just as a manager coordinates tasks, resources, and deadlines, Maven efficiently manages dependencies, builds, tests, and deployments.

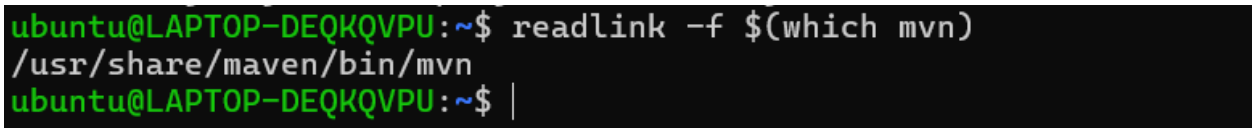
Step 1: Installing Java and Maven on Ubuntu

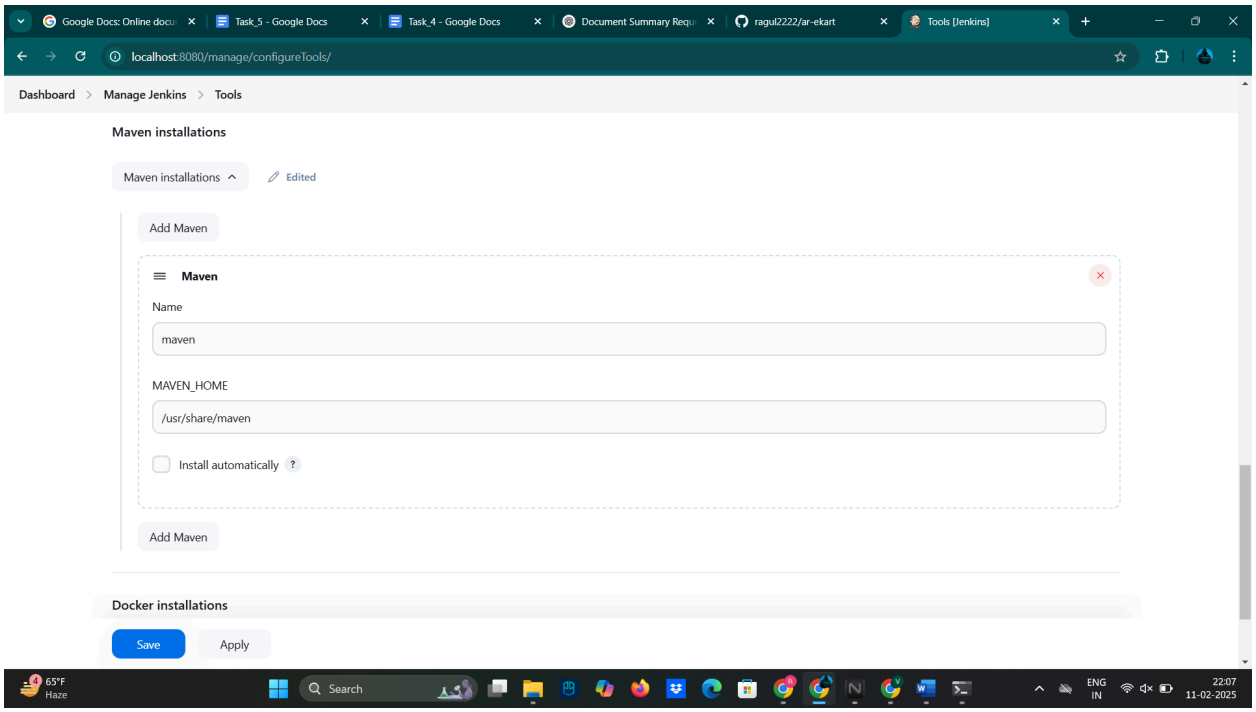
Before proceeding, ensure that both Java and Maven are installed on your Ubuntu system.

```
ubuntu@LAPTOP-DEQKQVPU:~$ 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@LAPTOP-DEQKQVPU:~$
```

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

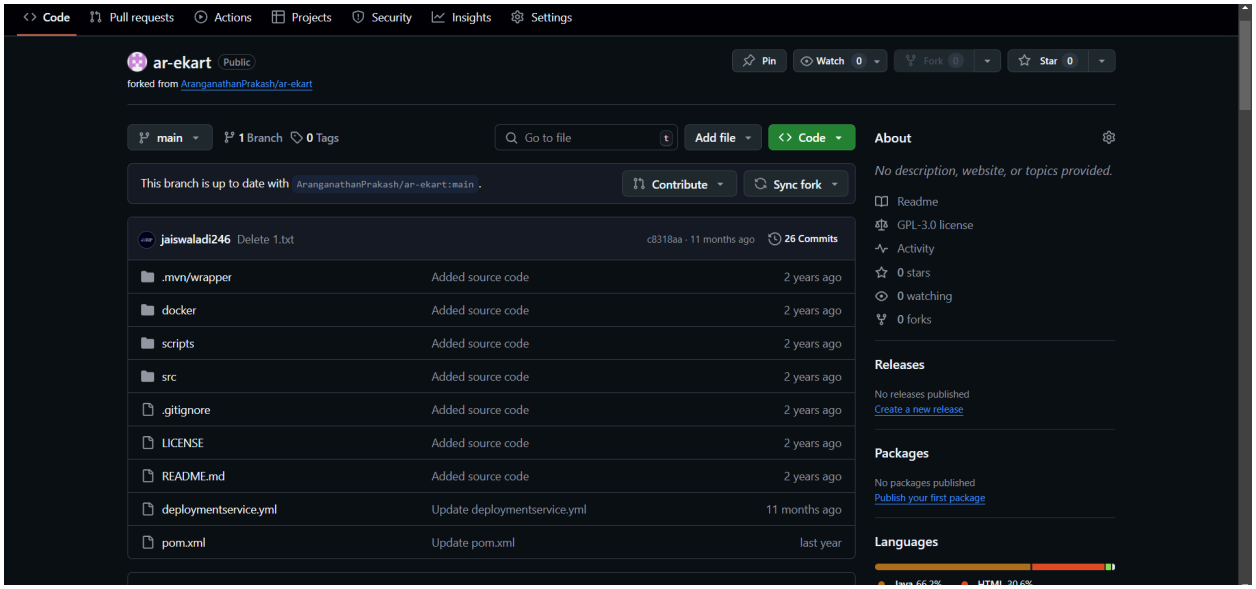






Step 2: Forking the eKart Repository on GitHub

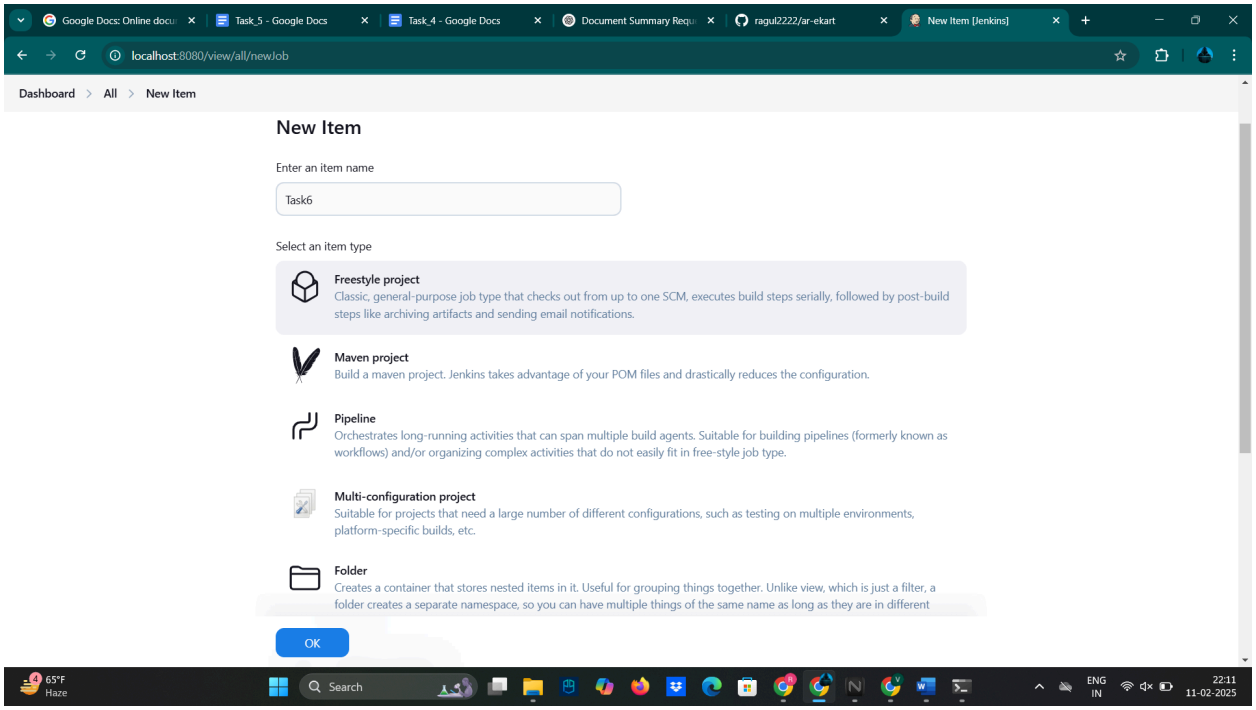
Fork the eKart repository to your personal GitHub account for seamless modification and integration.



Step 3: Configuring Jenkins

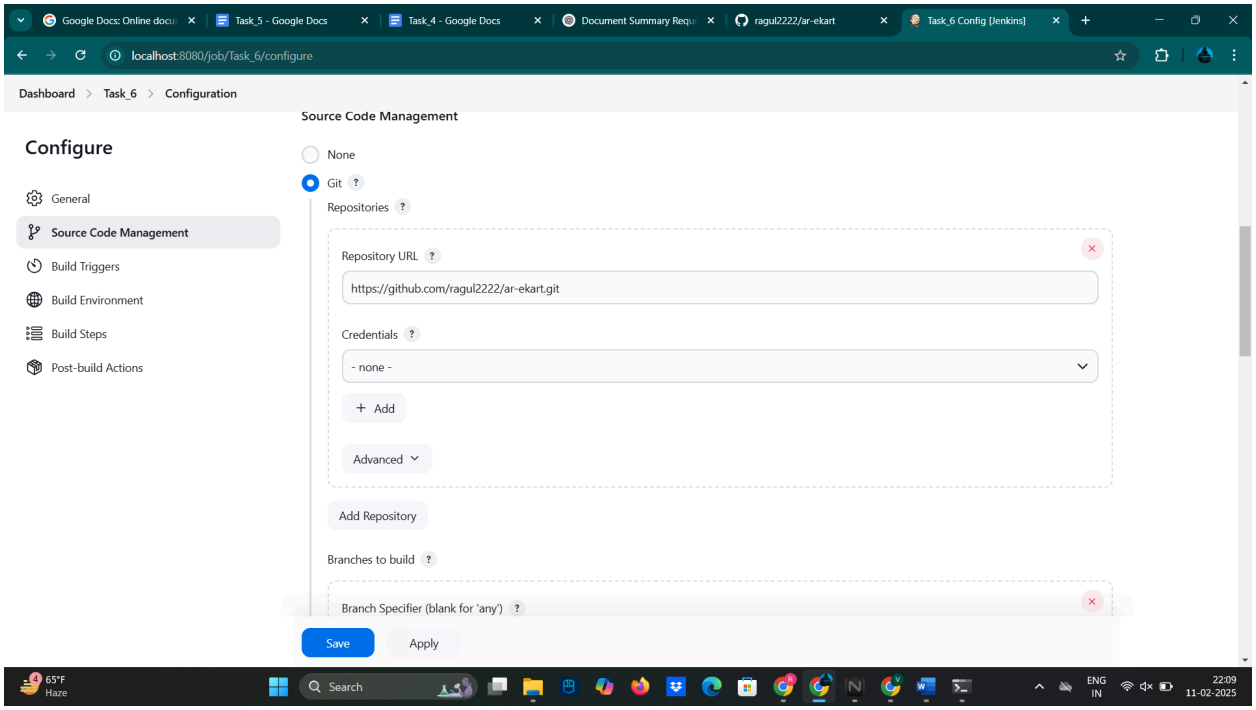
Creating a New Jenkins Job:

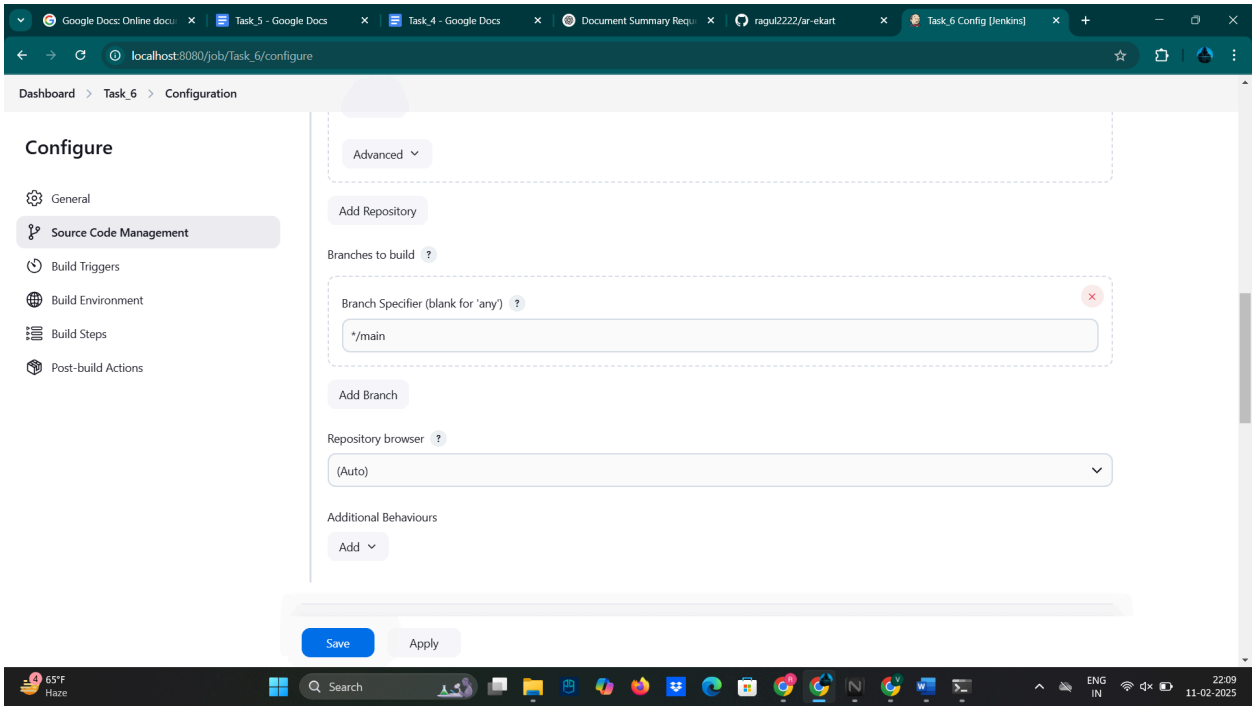
1. Open Jenkins in your browser.
2. Click **New Item** → Choose **Freestyle Project**.
3. Name the project **task6** and click **OK**.

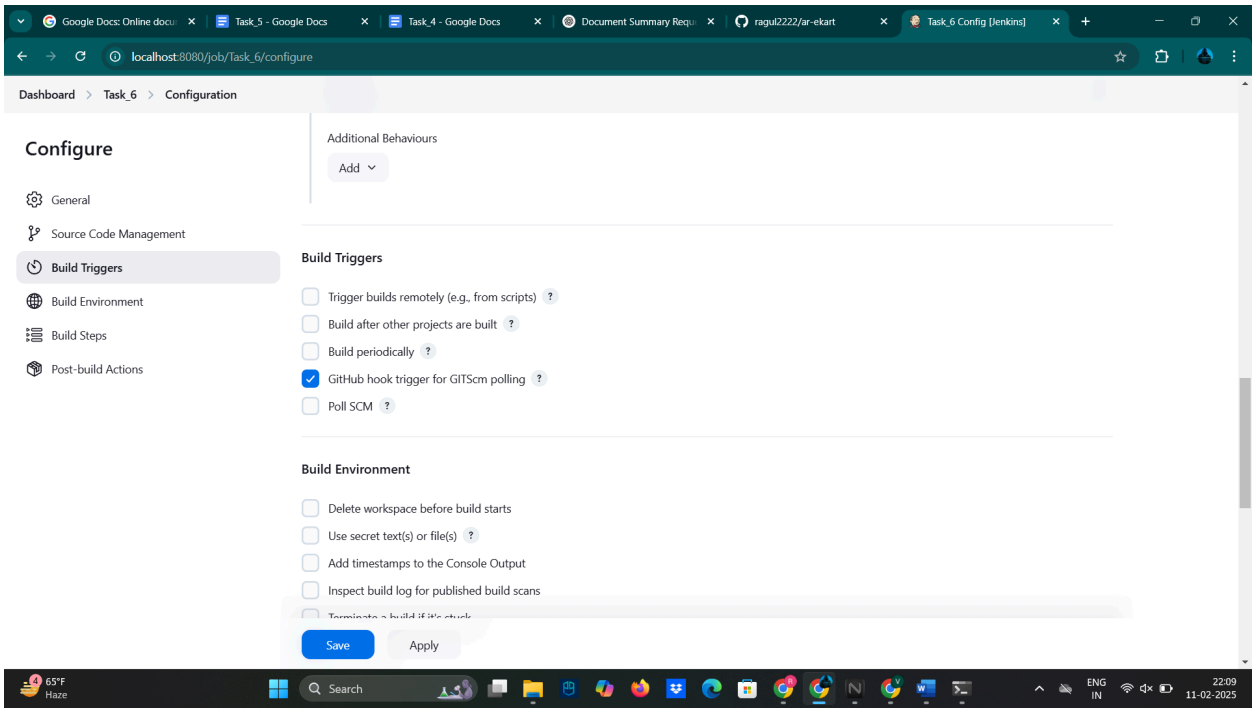


Job Configuration:

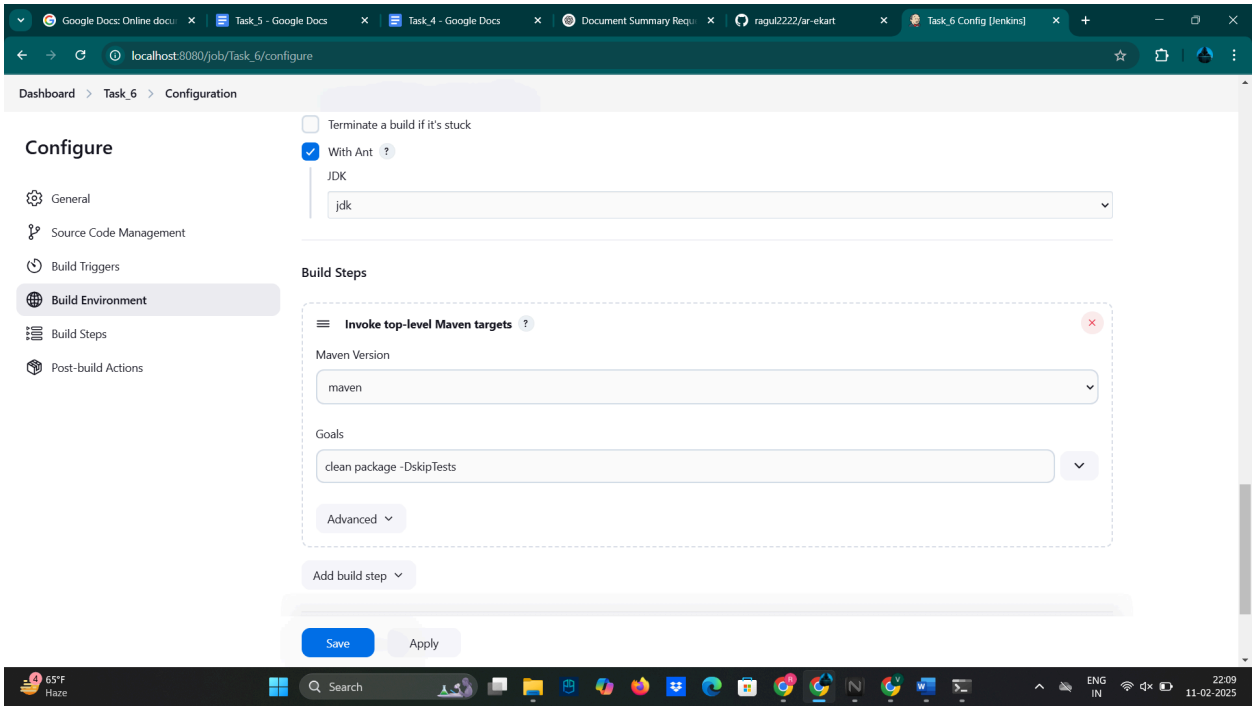
- **Setting Up Build Tools:**
    - Navigate to **Global Tool Configuration**.
    - Verify Java and Maven are installed, and configure if necessary.
- **Configuring GitHub Repository:**
    - Under **Source Code Management**, select **Git**.
    - Paste the forked repository URL.
    - Set the branch to **main**.


Google Docs: Online docu
Task_5 - Google Docs
Task_4 - Google Docs
Document Summary Requ
ragul2222/ar-ekart
Task_6 Config [Jenkins]
localhost:8080/job/Task_6/configure
Dashboard
Task_6
Configuration
Configure
General
Source Code Management
Build Triggers
Build Environment
Build Steps
Post-build Actions
Source Code Management
None
Git
Repositories
Repository URL
https://github.com/ragul2222/ar-ekart.git
Credentials
- none -
Add
Advanced
Add Repository
Branches to build
Branch Specifier (blank for 'any')
Save
Apply
65°F
Haze
Search
ENG
IN
22:09
11-02-2025





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



Enter the following command:
clean package -DskipTests

- Click **Build Now** to execute.

Step 4: Navigating Jenkins Workspace

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

1.

List all available projects:
ls

2.

Navigate to the project folder:
cd Maven_task5/target

3.

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

4.

Step 5: Deploying via Docker and Kubernetes

Building and Pushing Docker Image:

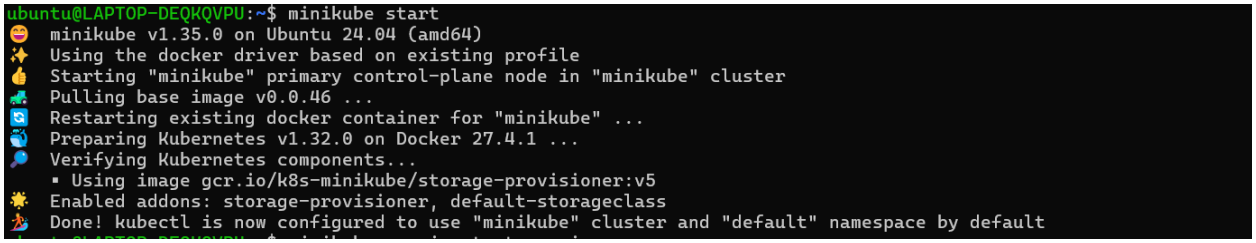
docker build -t test -f docker/Dockerfile
docker push rrragul/sample

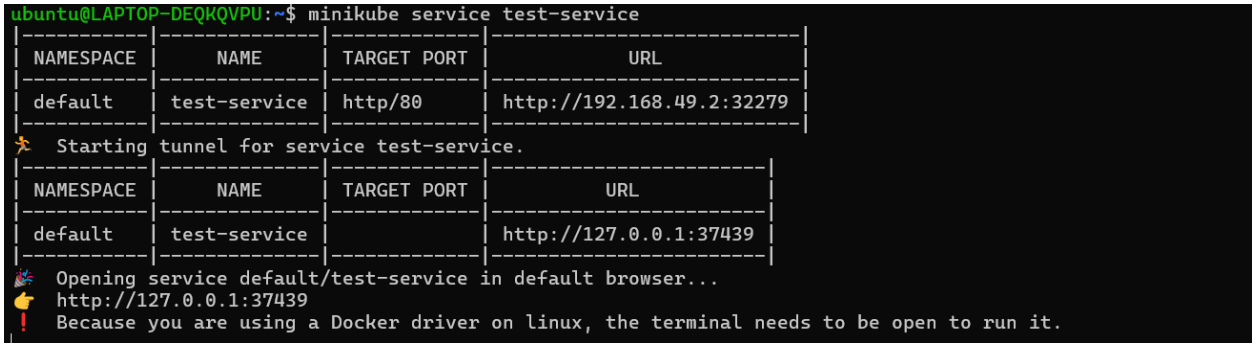
Deploying on Kubernetes:

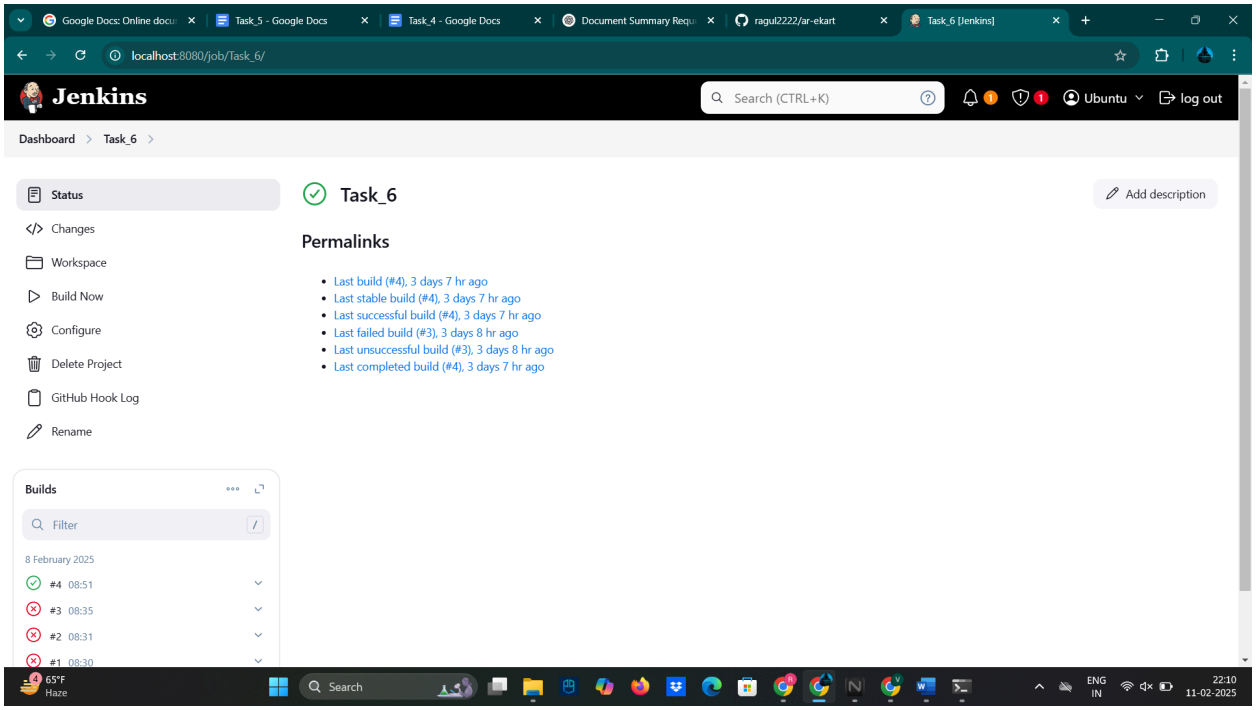
kubectl create deployment maven --image=test --port=80
kubectl expose deployment maven --type=NodePort --port=80 --target-port=8070

Verifying Deployment:

docker images | grep nivethitha24/mave # Confirm Docker image creation
kubectl get pods # List active pods
minikube service maven # Retrieve service URL







Jenkins Configuration & Output