**Building using Maven**

**Objective**

Maven is like a project manager for Java applications. Just like a manager organizes tasks, resources, and deadlines, Maven organizes dependencies, builds, tests, and deployments, ensuring everything runs smoothly and efficiently.

**Procedure**

Step 1: Install Java and Maven on Ubuntu

Step 2: Fork the eKart Repository on GitHub Step 3: Configure Jenkins Create a New Job in Jenkins

1. Open Jenkins in your browser.
2. Click on New Item → Select Freestyle Project → Name it Maven\_task5 → Click OK.

Configure the Job • Set up Build Tools:

o Under Global Tool Configuration, add Java and Maven if not configured.

• Set GitHub Repository:

* Go to Source Code Management → Select Git.
* Paste the forked repository URL. o Set the branch to main.

• Add Build Command: o Go to Build → Add Build Step → Select Invoke top-level Maven targets. o Enter: clean package -DskipTests o Then Build Now.

Step 4: Navigate to Jenkins Workspace cd /var/lib/jenkins/workspace ls # List available projects cd Maven\_task5 cd target ls # Verify generated artifacts (e.g.,

.jar file)

Step 5: Check Docker Image and Kubernetes Deployment docker build -t test -f docker/Dockerfile docker push shandeep04/password kubectl create deployment maven --image=test –port 80 kubectl expose deployment maven -type=NodePort --port=80 --target-port=8070 docker images | grep shandeep04/mave # Verify Docker image is built kubectl get pods # Check running pods minikube service maven # Get the service URL

















