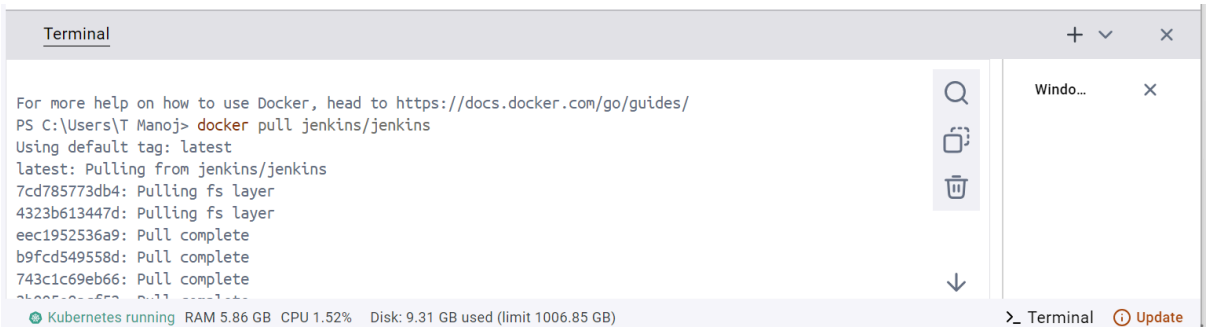
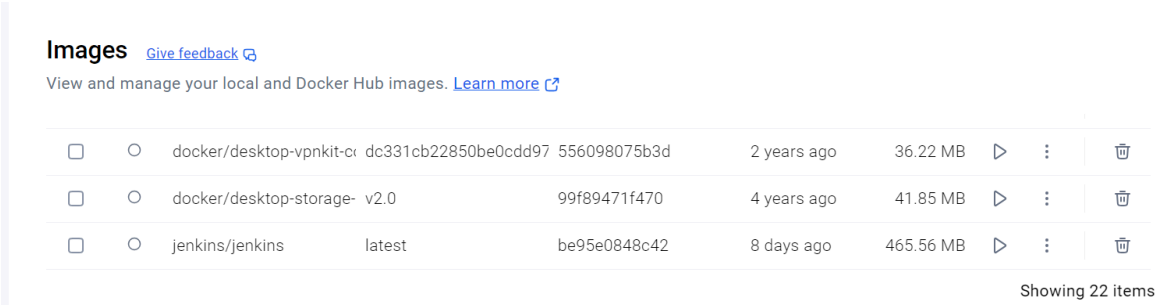
NAME : T MANOJ

EMPID : LYAKE2KHS

Setting up Jenkins with Docker

1. Go to hub.docker.com and open the Jenkins repository
2. Copy the pull command



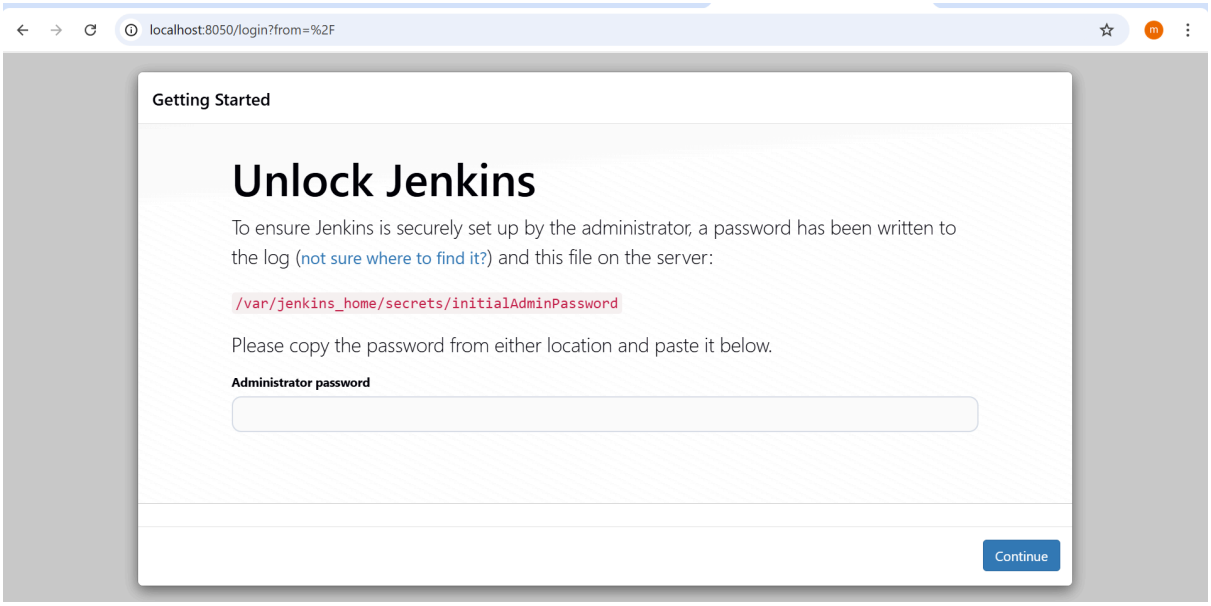


3. Run the command in your Docker desktop terminal

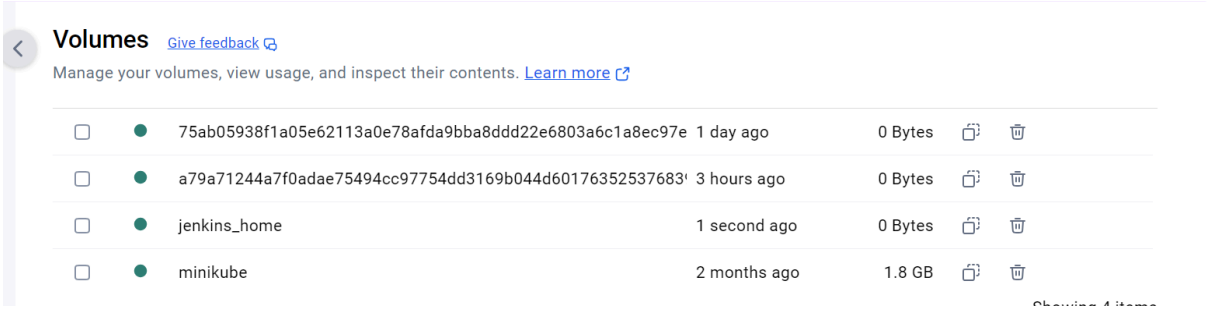
To create and run the Jenkins container:

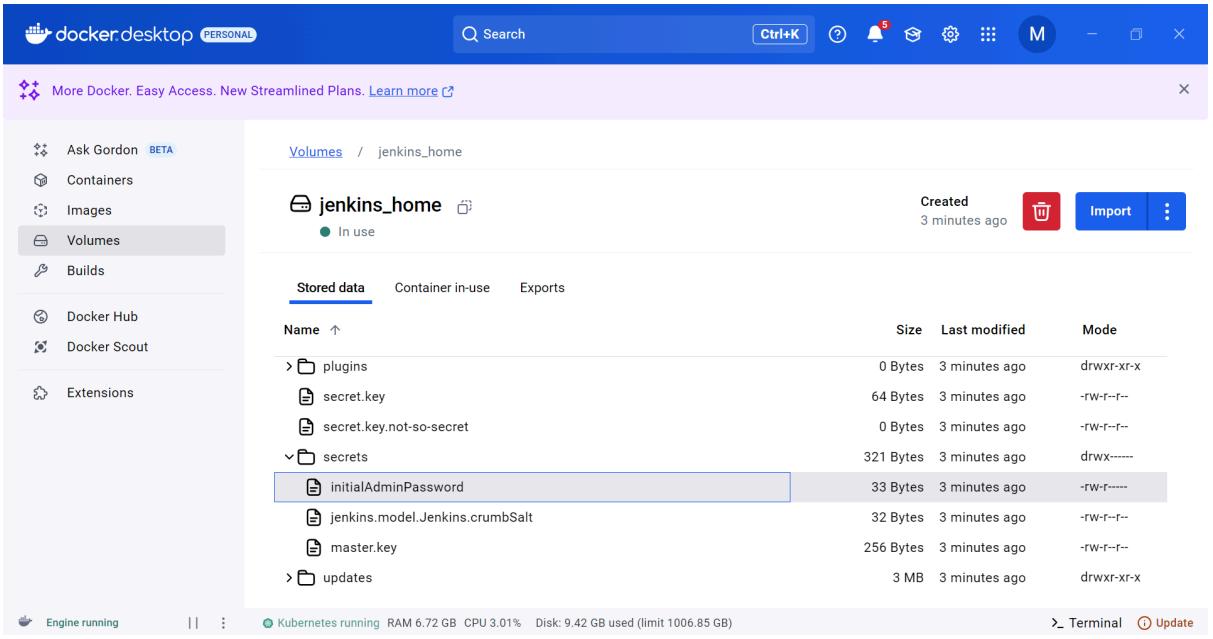
```
 docker run -p 8050:8080 -p 50000:50000 -v
jenkins_home:/var/jenkins_home jenkins/jenkins
```

4. Open Chrome and navigate to `localhost:8050`



5. Get the initial admin password:
    - Navigate to Volume > secrets > initial admin password
    - Save the password to your local system
    - View and copy the password
    - Paste it into Jenkins setup



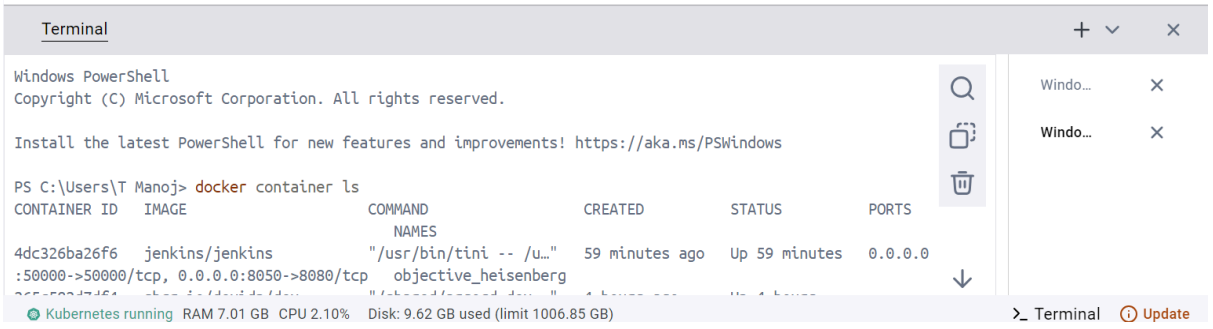


Setting up VS Code and Jenkins Pipeline

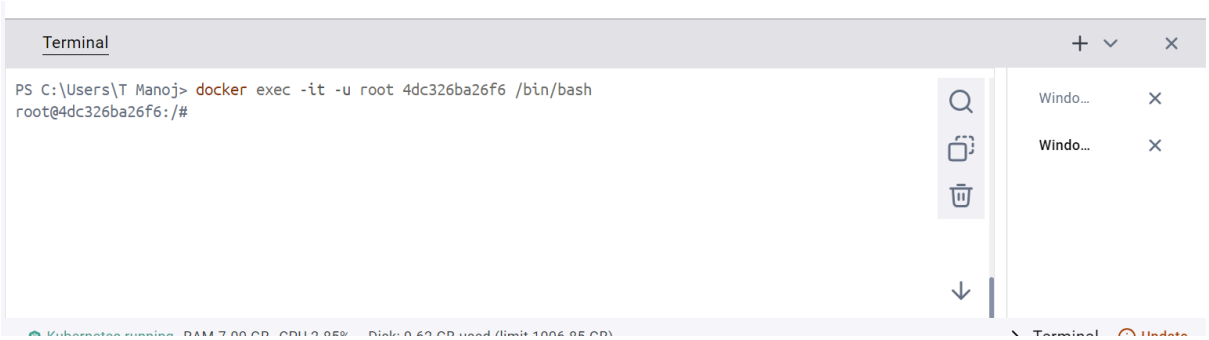
1. Open VS Code and install the Jenkins extension
2. Create a Jenkinsfile

Creating a Pipeline in Jenkins

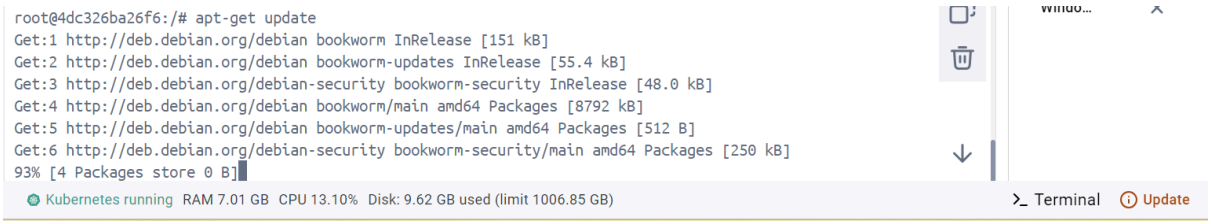
1. Write the pipeline script
2. Open a new terminal without closing the previous one
3. Run `docker container ls` to get the container ID

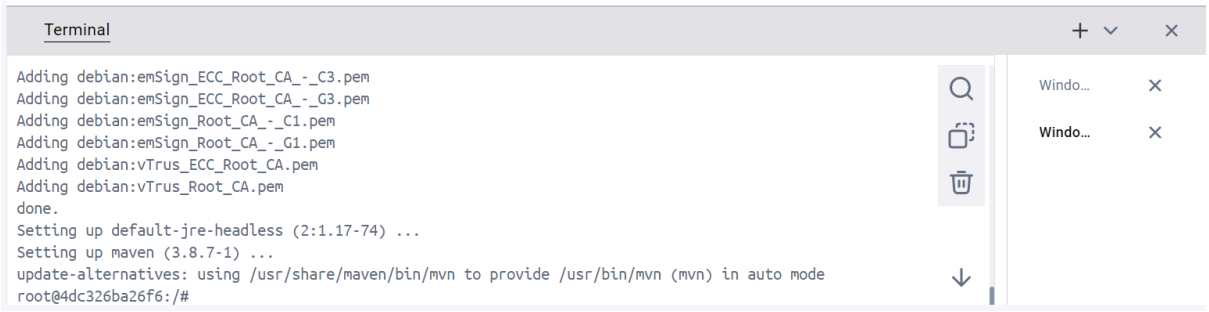


4. Access the container: `docker exec -it -u root [containerID] /bin/bash`



5. Update and install Maven (build tool)

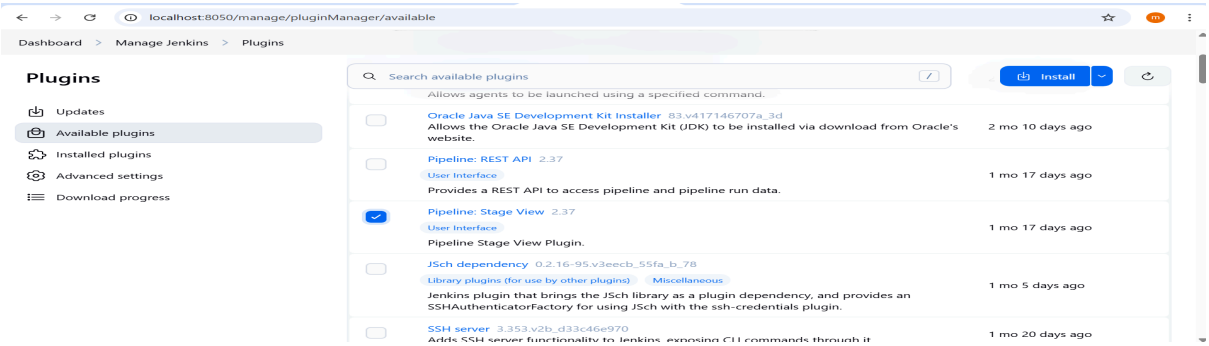


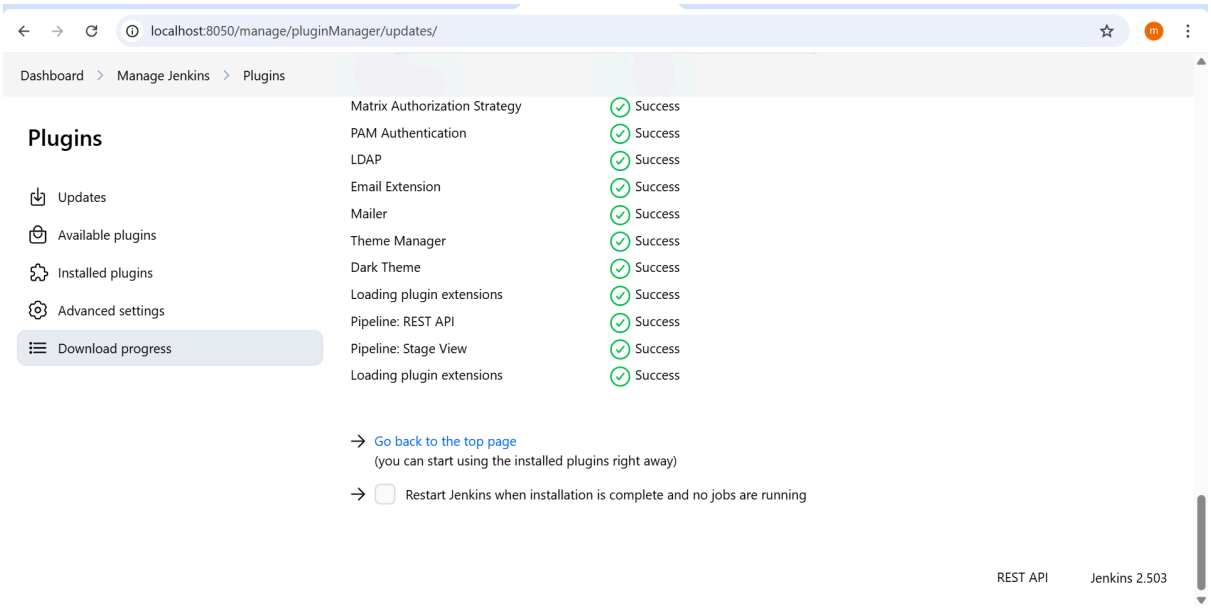


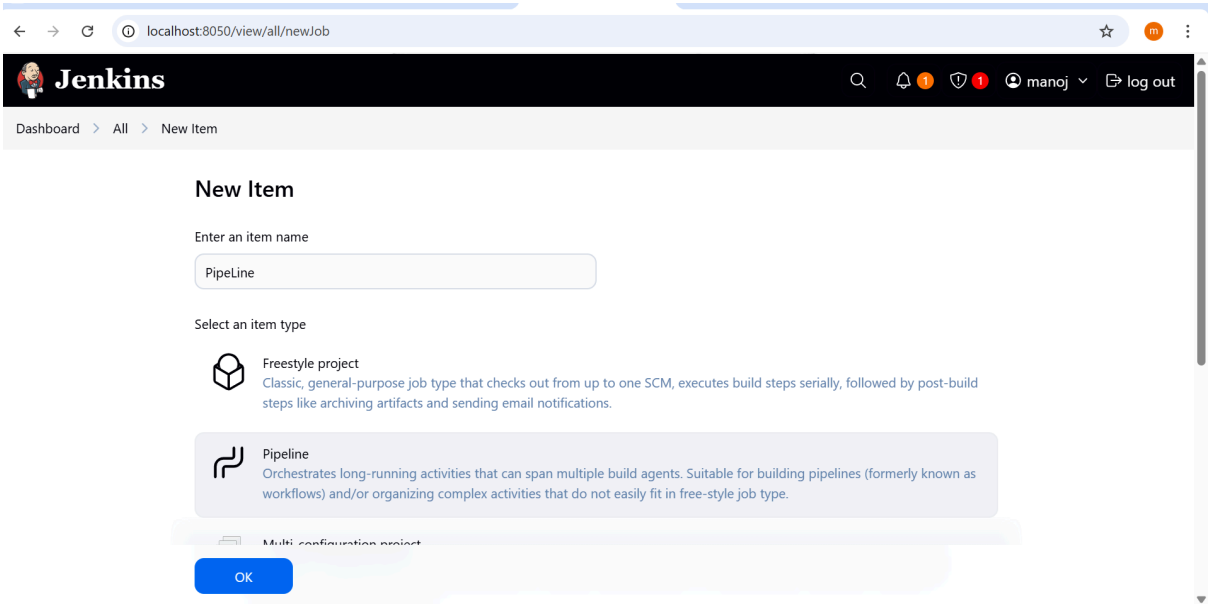
6. Exit the container

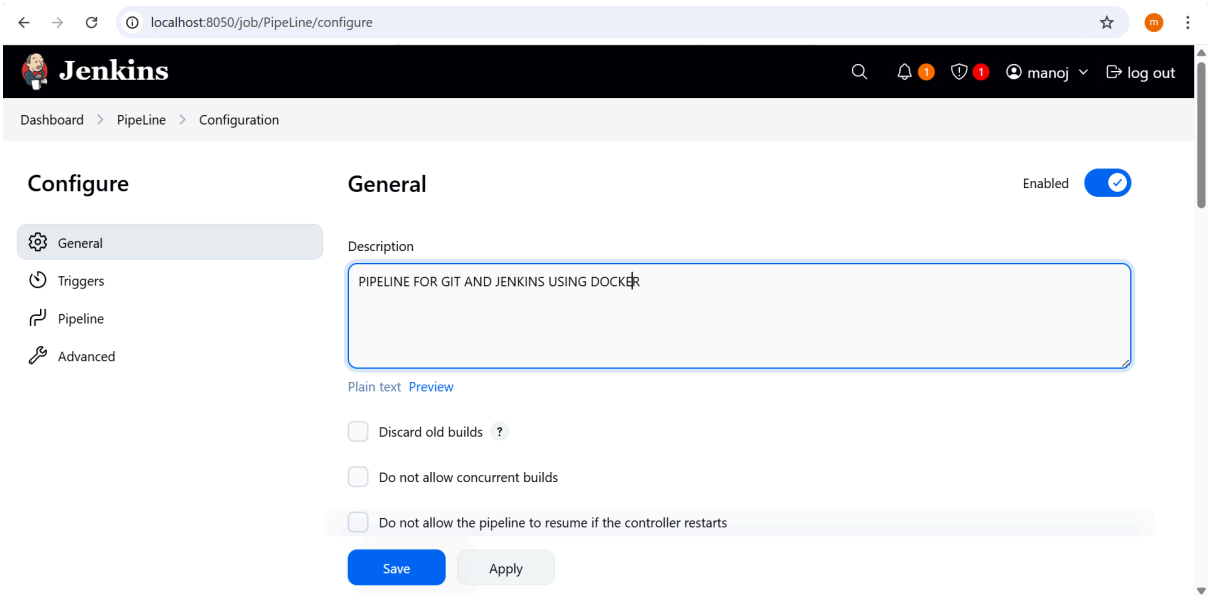
Setting up the Jenkins Job

1. Install the "Pipeline: Stage View" plugin




localhost:8050/manage/pluginManager/updates/
Dashboard > Manage Jenkins > Plugins
Plugins
Updates
Available plugins
Installed plugins
Advanced settings
Download progress
Matrix Authorization Strategy Success
PAM Authentication Success
LDAP Success
Email Extension Success
Mailer Success
Theme Manager Success
Dark Theme Success
Loading plugin extensions Success
Pipeline: REST API Success
Pipeline: Stage View Success
Loading plugin extensions Success
Go back to the top page
(you can start using the installed plugins right away)
Restart Jenkins when installation is complete and no jobs are running
REST API
Jenkins 2.503

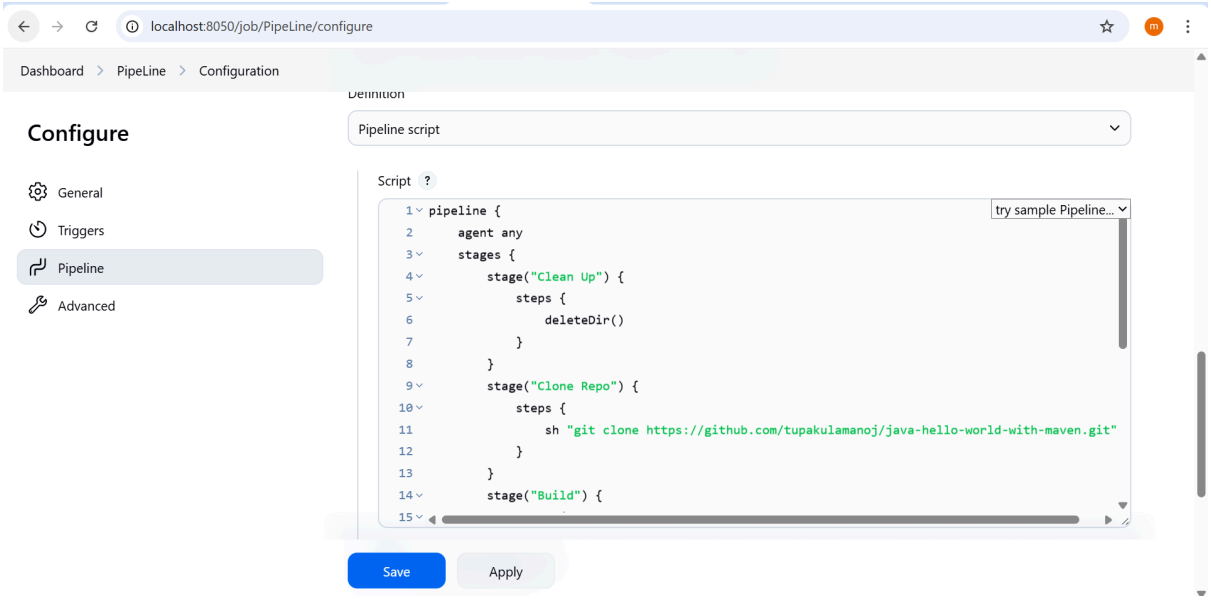

localhost:8050/view/all/newJob
Jenkins
manoj
log out
Dashboard > All > New Item
New Item
Enter an item name
PipeLine
Select an item type
Freestyle project
Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.
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.
OK



2. Create a job with this pipeline script:

```
pipeline {
    agent any
    stages {
        stage("Clean Up") {
            steps {
                deleteDir()
            }
        }
        stage("Clone Repo") {
            steps {
                sh "git clone
https://github.com/tupakulamanoj/java-hello-world-with-maven.git"
            }
        }
        stage("Build") {
            steps {
                dir("java-hello-world-with-maven") {
                    sh "mvn clean install"
                }
            }
        }
        stage("Test") {
            steps {
                dir("java-hello-world-with-maven") {
                    sh "mvn test"
                }
            }
        }
```

```
    }
}
```



3. Run and view the pipeline execution