MINIPROJECT-2

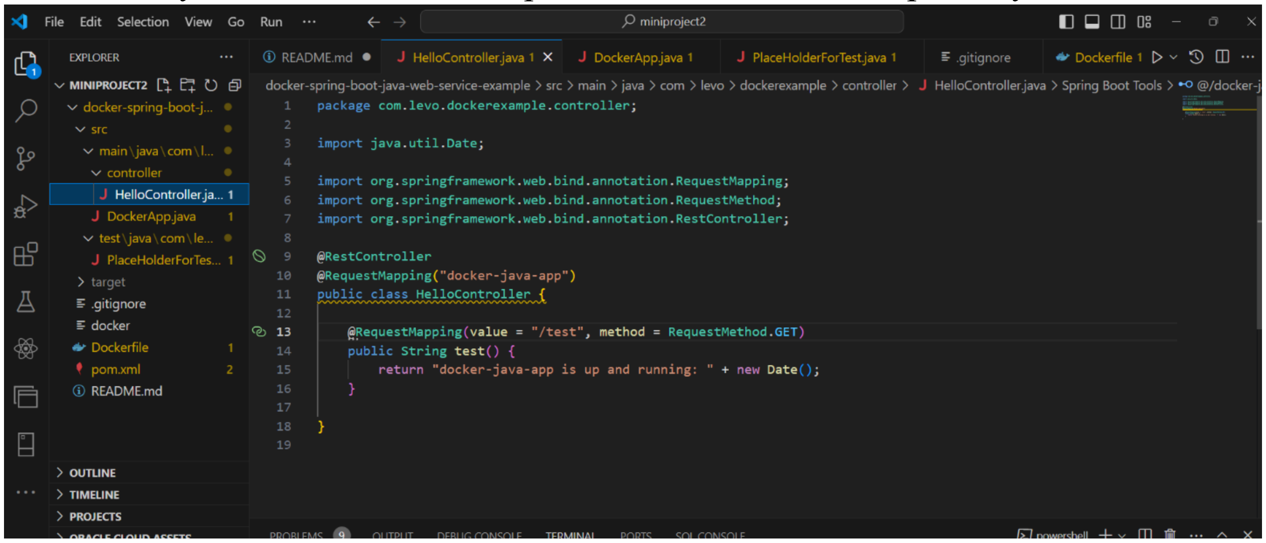
Student Name: Abhishek Sharma **UID: 23MCC20083**
Branch: MCA (CC & DEVOPS) **Section/Group: 23MCD-1 /A**
Semester: 3rd semester **Date of Performance:25/10/2024**
Subject Name: Devops Process Automation **Subject Code: 23CAH-731**

1.Aim/Overview of the practical:

- **Build and push a docker image from a Jenkins pipeline.**

2.Step for the Practical:

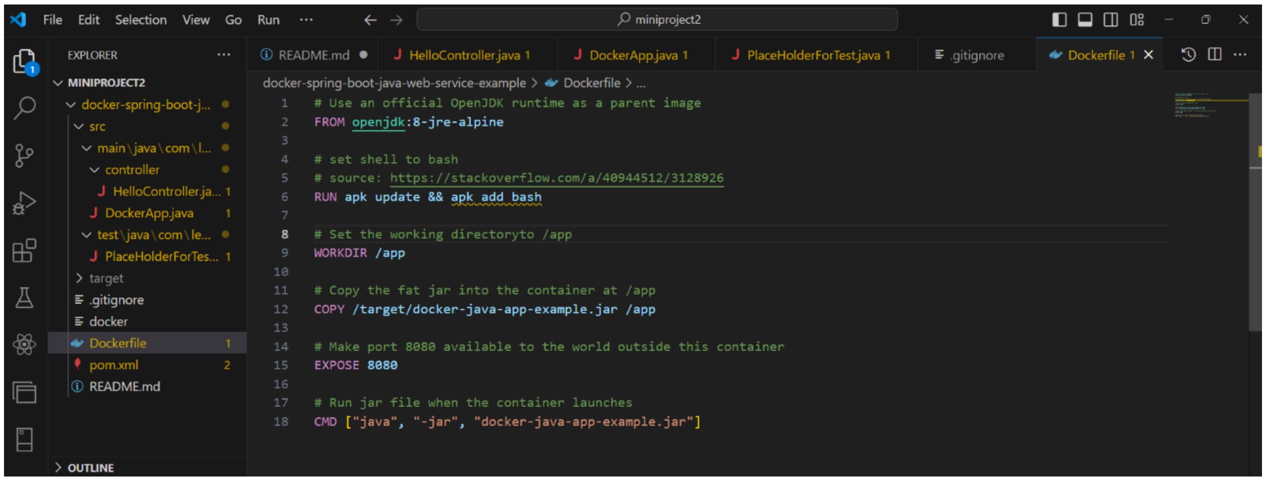
- Firstly, create a website and push it into the GitHub repository.



docker-spring-boot-java-web-service-example > src > main > java > com > levo > dockerexample > controller > HelloController.java > Spring Boot Tools

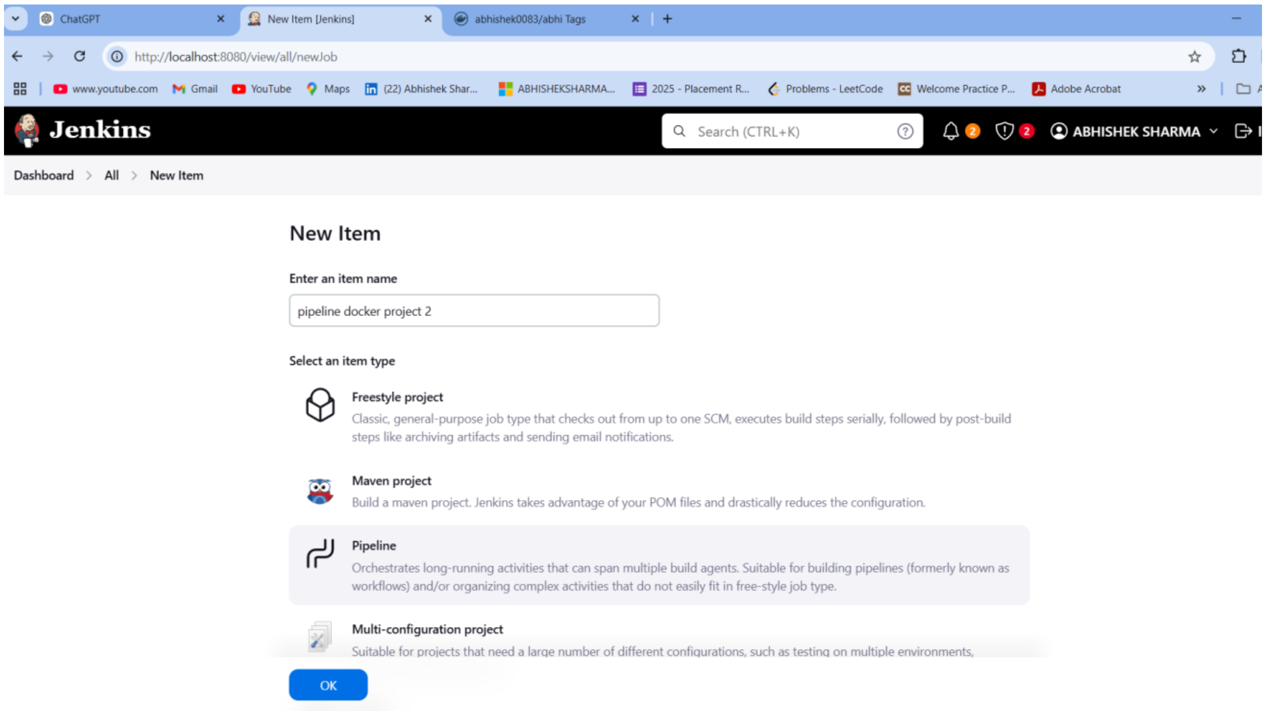
```java
package com.levo.dockerexample.controller;

import java.util.Date;

import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.RestController;

@RestController
@RequestMapping("docker-java-app")
public class HelloController {

    @RequestMapping(value = "/test", method = RequestMethod.GET)
    public String test() {
        return "docker-java-app is up and running: " + new Date();
    }

}
```

- Create a Docker file for build the docker image and push it to the docker hub.
- To build this project make sure you have downloaded and configure all the plugins related to Docker.
- In the Docker Hub create a username and password, after login create a repository for pushing the docker image through the Jenkins pipeline.
- If the docker repository is private then gave a credential to the Jenkins through global manager.
- Also docker Hub credential.



```
# Use an official OpenJDK runtime as a parent image
FROM openjdk:8-jre-alpine

# set shell to bash
# source: https://stackoverflow.com/a/40944512/3128926
RUN apk update && apk add bash

# Set the working directoryto /app
WORKDIR /app

# Copy the fat jar into the container at /app
COPY /target/docker-java-app-example.jar /app

# Make port 8080 available to the world outside this container
EXPOSE 8080

# Run jar file when the container launches
CMD ["java", "-jar", "docker-java-app-example.jar"]
```

- Now open the github and copy the URL of the repository to use in Jenkins CI/CD pipeline for
- Open the Jenkins through localhost and create a pipeline project



http://localhost:8080/view/all/newJob

Dashboard > All > New Item

New Item

Enter an item name

pipeline docker project 2

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.

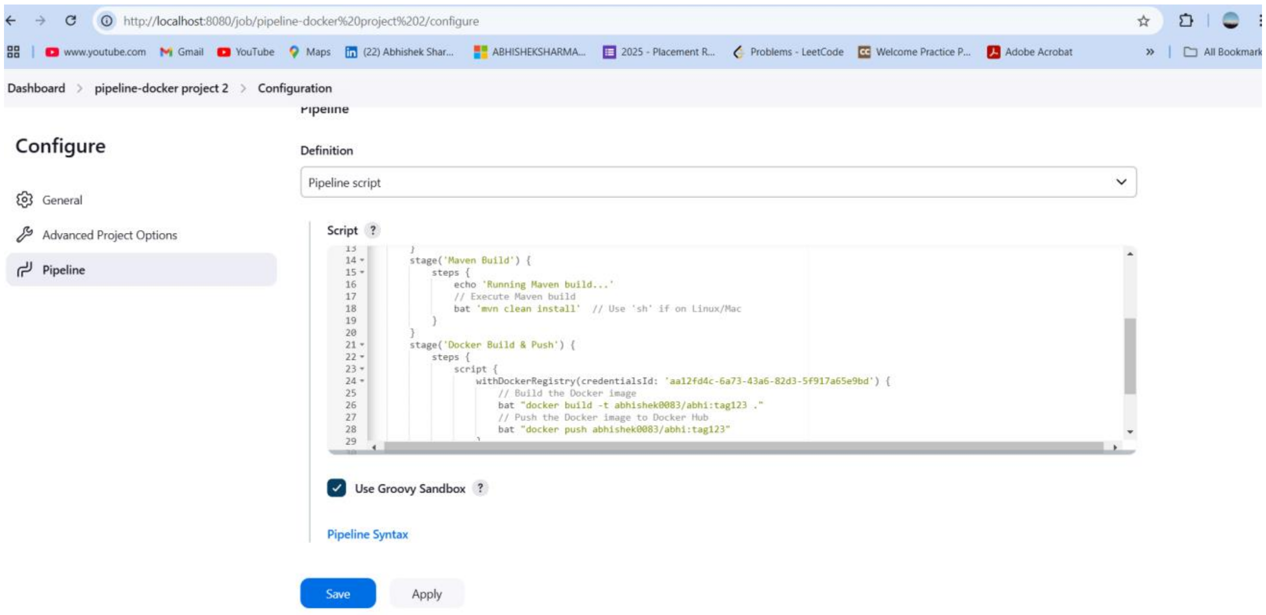
Maven project
Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.

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,

OK

- After that add description about the project and in the github section add repository URL.
- Now below sections add Jenkins script to build and Push.



http://localhost:8080/job/pipeline-docker%20project%202/configure

Dashboard > pipeline-docker project 2 > Configuration

Configure

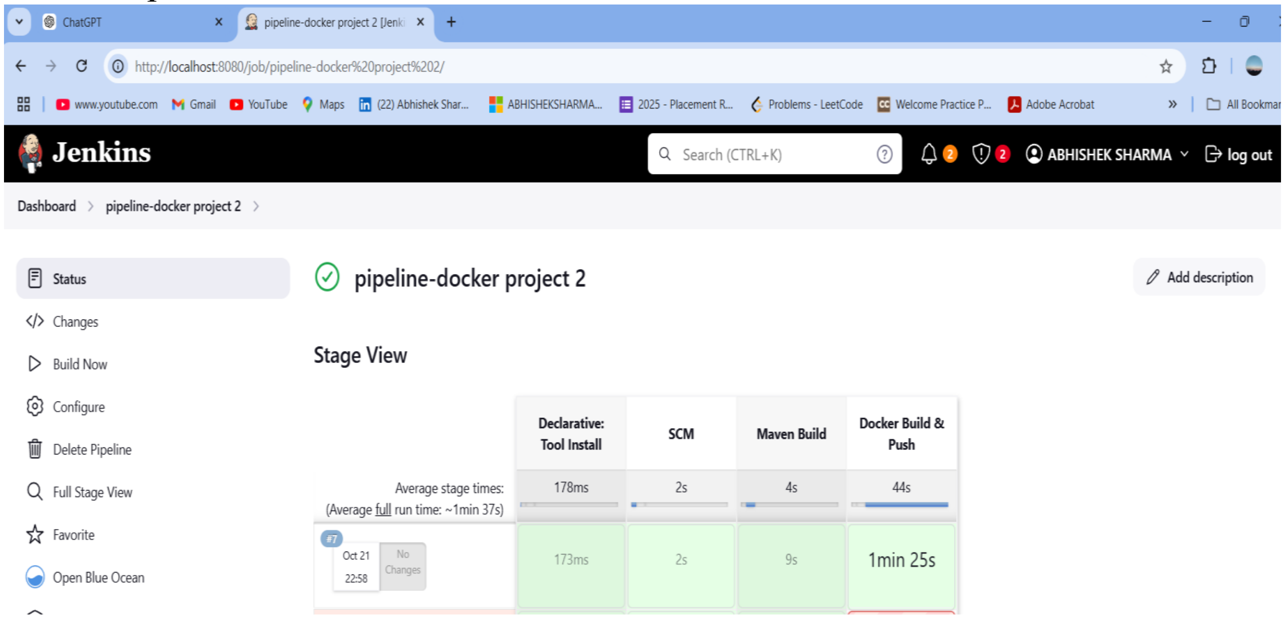
Definition: Pipeline script

Script

```
        }
        stage('Maven Build') {
            steps {
                echo 'Running Maven build...'
                // Execute Maven build
                bat 'mvn clean install'  // Use 'sh' if on Linux/Mac
            }
        }
        stage('Docker Build & Push') {
            steps {
                script {
                    withDockerRegistry(credentialsId: 'aa12fd4c-6a73-43a6-82d3-5f917a65e9bd') {
                        // Build the Docker image
                        bat "docker build -t abhishek0083/abhi:tag123 ."
                        // Push the Docker image to Docker Hub
                        bat "docker push abhishek0083/abhi:tag123"
```

[✓] Use Groovy Sandbox

- Save and build now the project or in the stage view section we can see the output.



http://localhost:8080/job/pipeline-docker%20project%202/

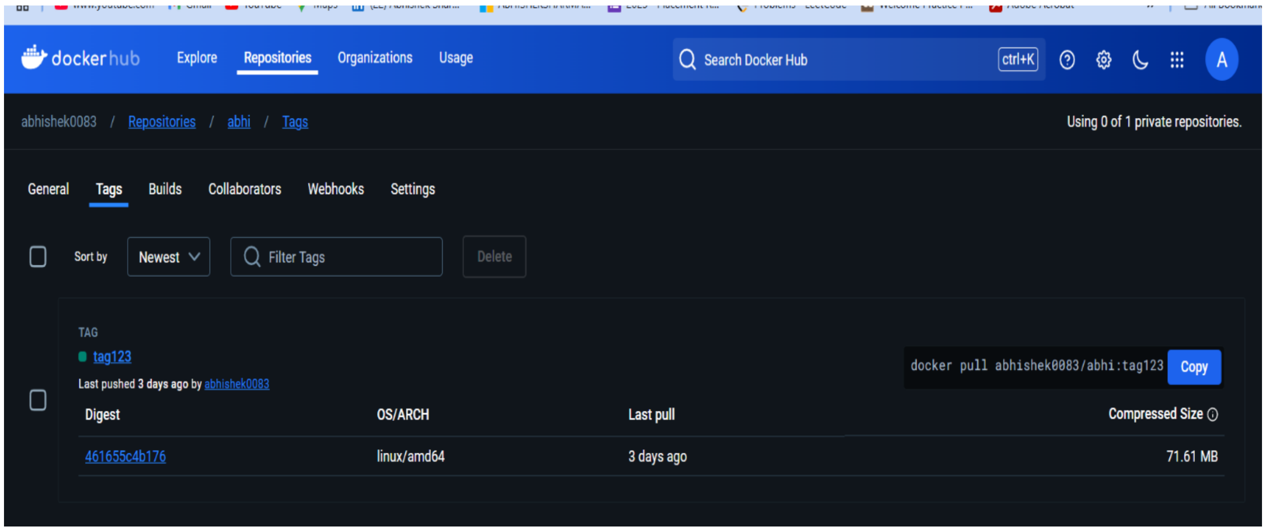
Dashboard > pipeline-docker project 2

pipeline-docker project 2

Stage View

	Declarative: Tool Install	SCM	Maven Build	Docker Build & Push
Average stage times: (Average full run time: ~1min 37s)	178ms	2s	4s	44s
#7 Oct 21 22:58 No Changes	173ms	2s	9s	1min 25s

- To see the output open the docker Hub.
- Check whether the image is pushed or not.



abhishek0083 / Repositories / abhi / Tags

Using 0 of 1 private repositories.

TAG
tag123
Last pushed 3 days ago by abhishek0083

```
docker pull abhishek0083/abhi:tag123
```

Digest	OS/ARCH	Last pull	Compressed Size
461655c4b176	linux/amd64	3 days ago	71.61 MB

6. Repository Link: *https://github.com/abhishek0083/DevopsMiniProject2*

3.Learning Outcomes (What I have learned)

I Learned to: -

- How to build and push docker image into docker hub using a Jenkins pipeline.