

ASSIGNMENT 10

IT_41_Himanshu Shukla

Create a Jenkins pipeline to build and push Docker Image. Dockerfile, Jenkinsfile and the application(for which image will be built) shall be stored in a GitHub Repository.

1.create a Jenkins file and do as follow

Dashboard > Jenkinsfile_hsk > Configuration

Configure

- General
- Advanced Project Options
- Pipeline

Build Triggers

- ☐ Build after other projects are built ?
- ☐ Build periodically ?
- ☒ GitHub hook trigger for GITScm polling ?
- ☐ Poll SCM ?
- ☐ Quiet period ?
- ☐ Trigger builds remotely (e.g., from scripts) ?

Advanced Project Options

Advanced ▾

Dashboard > Jenkinsfile_hsk > Configuration

Configure

- General
- Advanced Project Options
- Pipeline

Pipeline

Definition

Pipeline script from SCM ▾

SCM ?

Git ▾ ?

Repositories ?

Repository URL ?

https://github.com/himanshuhs01/Docker_java.git

Credentials ?

himanshuhs01/***** (Docker_hsk) ▾

Add ▾

Here in credential put your docker username and docker password and then go to Dashboard>Manage Jenkins>Credential and copy the Id

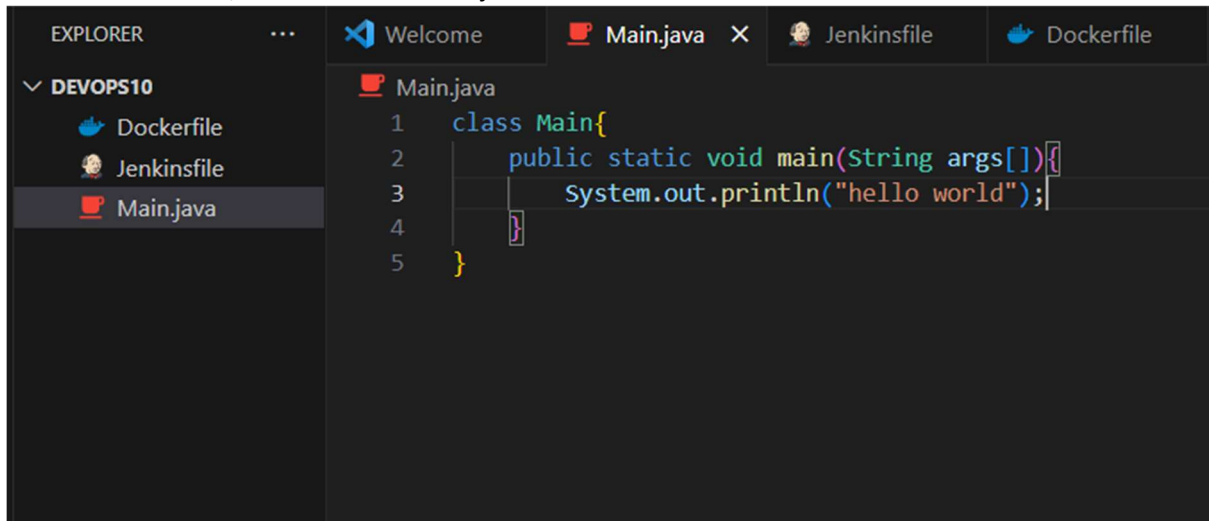
Dashboard > Manage Jenkins > Credentials

Credentials

T	P	Store	Domain	ID
		System	(global)	bf153ce8-4048-4ca4-b781-830c87ec605b
		System	(global)	235da7d6-f1f4-4fe6-bbf0-2bffa1578079
		System	(global)	863e2693-013e-42e8-b46e-bc5f02277301

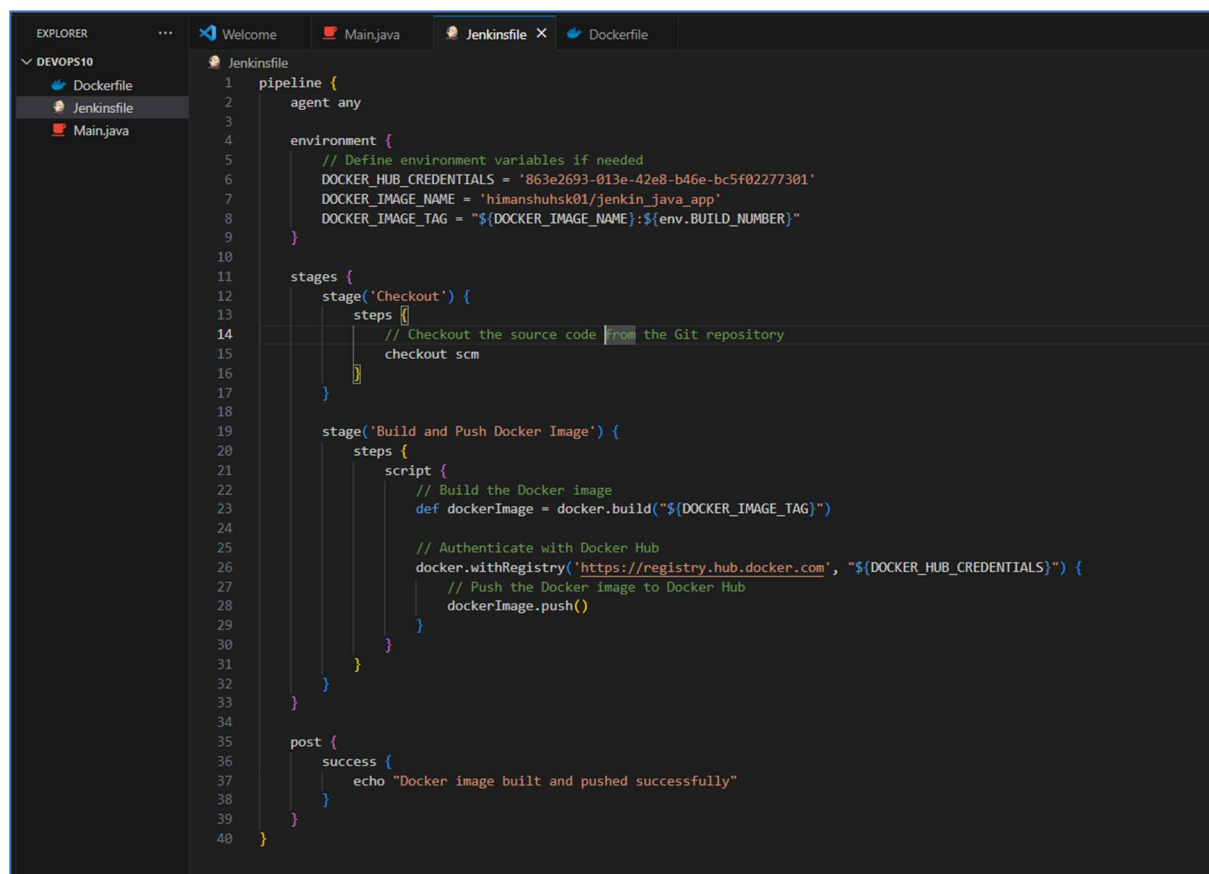
Stores scoped to Jenkins

2 create Dockerfile,Jenkinsfileand Main.java in vs code

A screenshot of the Visual Studio Code editor. The Explorer sidebar on the left shows a project named 'DEVOPS10' with three files: 'Dockerfile', 'Jenkinsfile', and 'Main.java'. The 'Main.java' file is selected and open in the main editor. The code in 'Main.java' is as follows:

```
1 class Main{
2     public static void main(String args[]){
3         System.out.println("hello world");
4     }
5 }
```

Now paste the Id in line 6 in Docker hub credentials and in image name put your username/filename

A screenshot of the Visual Studio Code editor. The Explorer sidebar on the left shows the same project 'DEVOPS10' with 'Dockerfile', 'Jenkinsfile', and 'Main.java'. The 'Jenkinsfile' file is selected and open in the main editor. The code in 'Jenkinsfile' is as follows:

```
1 pipeline {
2     agent any
3
4     environment {
5         // Define environment variables if needed
6         DOCKER_HUB_CREDENTIALS = '863e2693-013e-42e8-b46e-bc5f02277301'
7         DOCKER_IMAGE_NAME = 'himanshuhs01/jenkin_java_app'
8         DOCKER_IMAGE_TAG = "${DOCKER_IMAGE_NAME}:${env.BUILD_NUMBER}"
9     }
10
11     stages {
12         stage('Checkout') {
13             steps {
14                 // Checkout the source code from the Git repository
15                 checkout scm
16             }
17         }
18
19         stage('Build and Push Docker Image') {
20             steps {
21                 script {
22                     // Build the Docker image
23                     def dockerImage = docker.build("${DOCKER_IMAGE_TAG}")
24
25                     // Authenticate with Docker Hub
26                     docker.withRegistry('https://registry.hub.docker.com', "${DOCKER_HUB_CREDENTIALS}") {
27                         // Push the Docker image to Docker Hub
28                         dockerImage.push()
29                     }
30                 }
31             }
32         }
33     }
34
35     post {
36         success {
37             echo "Docker image built and pushed successfully"
38         }
39     }
40 }
```

```
pipeline {
    agent any
```

```
environment {
    // Define environment variables if needed
    DOCKER_HUB_CREDENTIALS = '863e2693-013e-42e8-b46e-bc5f02277301'
    DOCKER_IMAGE_NAME = 'himanshuhs02/jenkin_java_app'
```

```

    DOCKER_IMAGE_TAG = "${DOCKER_IMAGE_NAME}:${env.BUILD_NUMBER}"
}

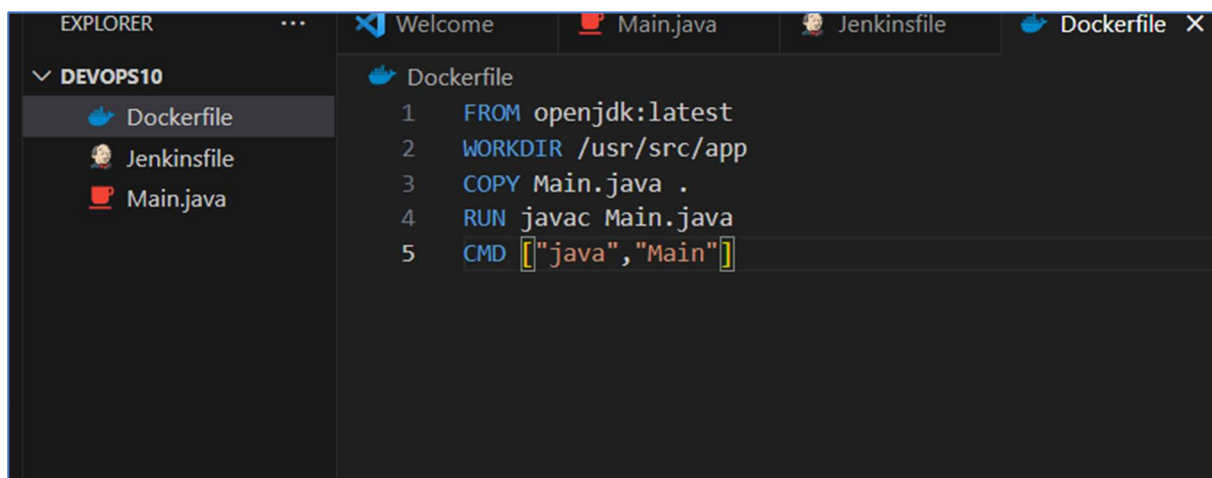
stages {
    stage('Checkout') {
        steps {
            // Checkout the source code from the Git repository
            checkout scm
        }
    }

    stage('Build and Push Docker Image') {
        steps {
            script {
                // Build the Docker image
                def dockerImage = docker.build("${DOCKER_IMAGE_TAG}")

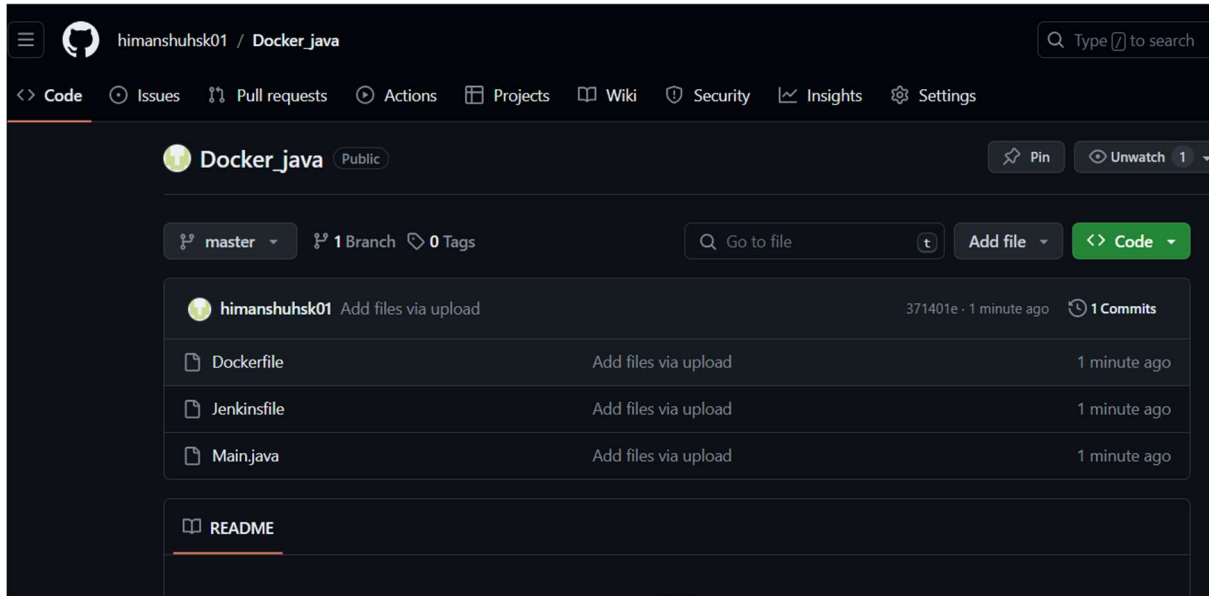
                // Authenticate with Docker Hub
                docker.withRegistry('https://registry.hub.docker.com', "${DOCKER_HUB_CREDENTIALS}") {
                    // Push the Docker image to Docker Hub
                    dockerImage.push()
                }
            }
        }
    }
}

post {
    success {
        echo "Docker image built and pushed successfully"
    }
}
}

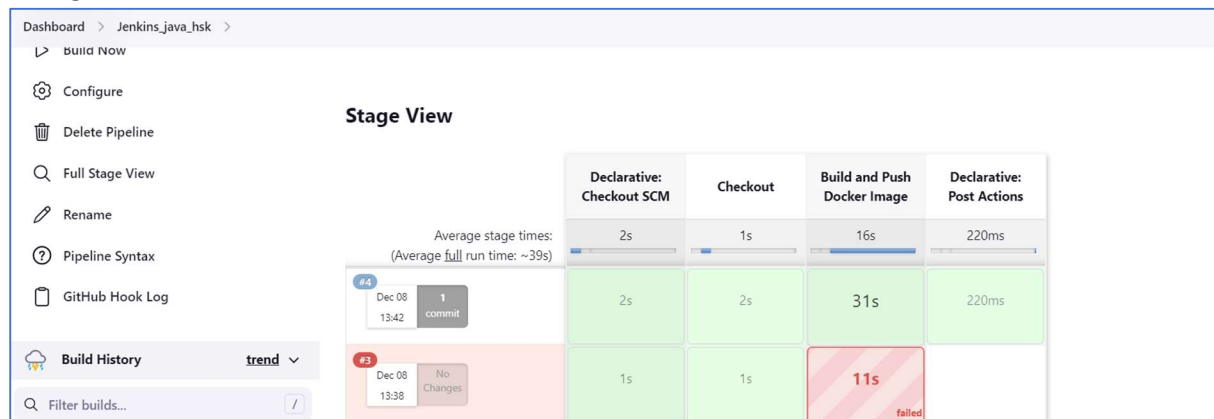
```



Push into github which you have created (here branch name is master)



Now go to localhost and build it



Here the it is pushed successfully

