EE 6254 : DEVOPS ENGINEERING
ASSIGNMENT 02

NAME   : UDAYANTHIKA K.D.S.

REG NO.   : EG/2020/3942

SEMESTER : 06

DATE   : 22/04/2024

1. Figure out and write the required codes to dockerize your application

- Selected a application.
- Wrote Dockerfile and docker-compose.yml to containerize the application.
- Manually tested the Dockerization process to ensure functionality.

The Dockerfile:

```dockerfile
# Use the latest version of the Node.js image as the base image
FROM node:latest

# Set the working directory inside the container to /usr/src/app
WORKDIR /usr/src/app

# Copy the contents of the local "nodeapp" directory to the root directory of the
container
COPY nodeapp/* /

# Run the npm install command to install the dependencies specified in package.json
RUN npm install

# Expose port 3000 to allow incoming connections to the container
EXPOSE 3000

# Start the application by running the "npm start" command
CMD [ "npm", "start" ]
```
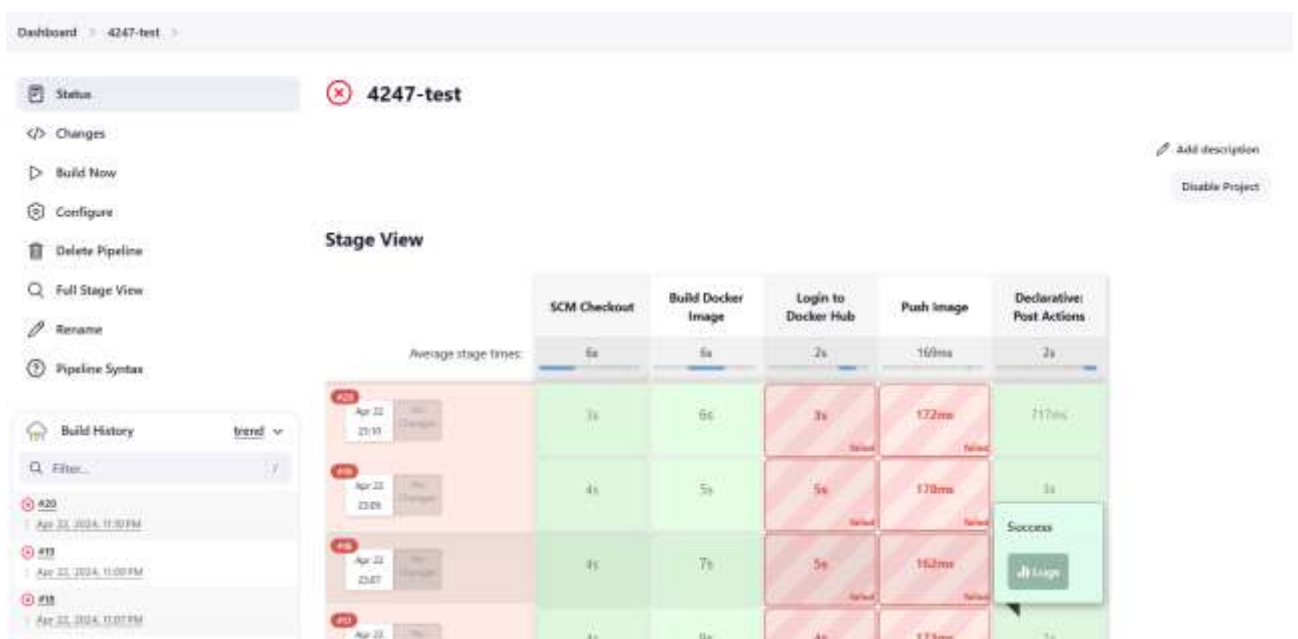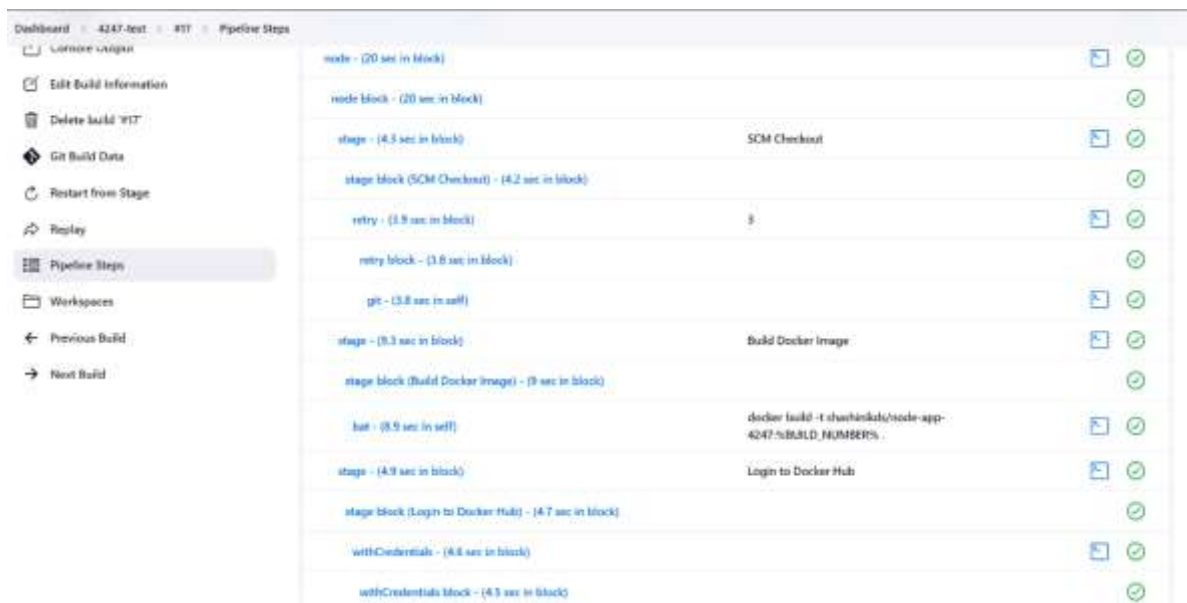
2. Put your codes to the remote repository

- Created a GitLab repository

  The GitHub Link: https://github.com/ShashiniUK/EG_2020_4247_Devops

- Pushed the application code, Dockerfile, docker-compose.yml, and Jenkins pipeline script to the repository.

+ New Item

People

Build History

Project Relationship

Check File Fingerprint

Manage Jenkins

My Views

Build Queue ∨

No builds in the queue

Build Executor Status ∨

1 Idle

2 Idle

All +

Add description

S	W	Name ↓	Last Success	Last Failure	Last Duration	
⊗	☁	4247-test	N/A	12 min #20	17 sec	▷
⊗	☁	Devops	N/A	1 hr 29 min #1	15 sec	▷

Icon: S M L

Icon legend   🔊 Atom feed for all   🔊 Atom feed for failures   🔊 Atom feed for just latest builds

REST API   Jenkins 2.440.3

3. Write a pipeline to automate the dockerize process using groovy. Groovy code also should be kept inside the repository

```groovy
pipeline {
    agent any


    stages {
        stage('SCM Checkout') {
            steps {
                retry(3) {
                    git branch: 'main', url: 'https://github.com/ShashiniUK/EG_2020_4247_Devops'
                }
            }
        }
        stage('Build Docker Image') {
            steps {
                bat 'docker build -t shashinikds/node-app-4247:%BUILD_NUMBER% .'
            }
        }
        stage('Login to Docker Hub') {
            steps {
                withCredentials([string(credentialsId: 'testdockerass', variable: 'testDockerPass_4247')]) {
                    bat'docker login -u shashinikds -p ${testDockerPass_4247}'
                    // some block
                }
                withCredentials([string(credentialsId: 'Devops4247', variable: 'test-dockerhub4247')]) {
                    bat'docker login -u shashinikds -p ${test-dockerhub4247}'


                }


            }
        }
        stage('Push Image') {
            steps {
                bat 'docker push adomicarts/node-app-4247:%BUILD_NUMBER%'
            }
        }
    }
    post {
        always {
            bat 'docker logout'
        }
    }
}
```

Docker