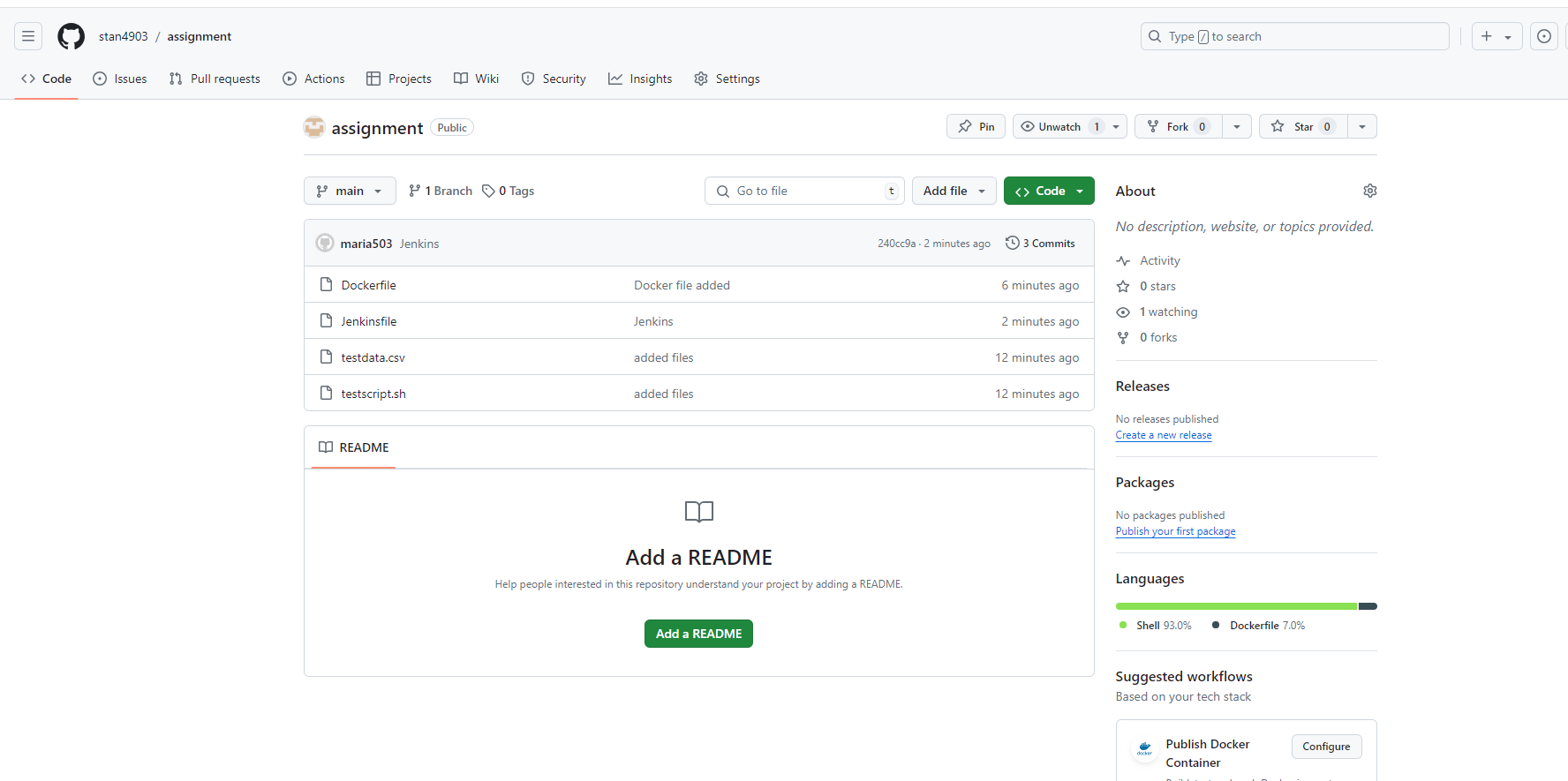
To Achieve this in CI/CD Pipeline, I’ll use GIT, Docker and Jenkins

1. Initially I’ll add all the scripts in GIT
2. I’ll Create a docker file which contains test script(bash) , I’ll build the docker image and run the container
3. For continuous Integration, I’ll use Jenkins

**GIT**

https://github.com/stan4903/assignment │

 **Docker File Creation –** testscript-assignment

FROM alpine:3.14

RUN apk --no-cache add curl jq bash

COPY testscript.sh /usr/local/bin/testscript.sh

COPY testdata2.csv.csv /usr/local/bin/testdata2.csv.csv

RUN chmod +x /usr/local/bin/testscript.sh

ENTRYPOINT ["/usr/local/bin/testscript.sh"]

**Build docker image**

docker build -t testscript-assignment .

**Running Docker container**

docker run --rm testscript-assignment 01

Using Jenkins Pipeline

pipeline {

    agent any

    environment {

        DOCKER\_IMAGE = 'testscript-assignment'

        UNIQUE\_KEY = '01'

    }

    stages {

        stage('Clone Repository') {

            steps {

                git url: ' https://github.com/stan4903/assignment.git', branch: 'main'

            }

        }

        stage('Build Docker Image') {

            steps {

                script {

                    // Build the Docker image

                    sh "docker build -t ${DOCKER\_IMAGE} ."

                }

            }

        }

        stage('Run Tests') {

            steps {

                script {

                    // Run the Docker container with the provided unique key

                    sh "docker run --rm ${DOCKER\_IMAGE} ${UNIQUE\_KEY}"

                }

            }

        }

    }

    post {

        always {

            // Clean up Docker resources if needed

            sh "docker rmi ${DOCKER\_IMAGE}"

        }

        success {

            echo 'The script ran successfully!'

        }

        failure {

            echo 'The script failed. Check the logs for more details.'

        }

    }

}