

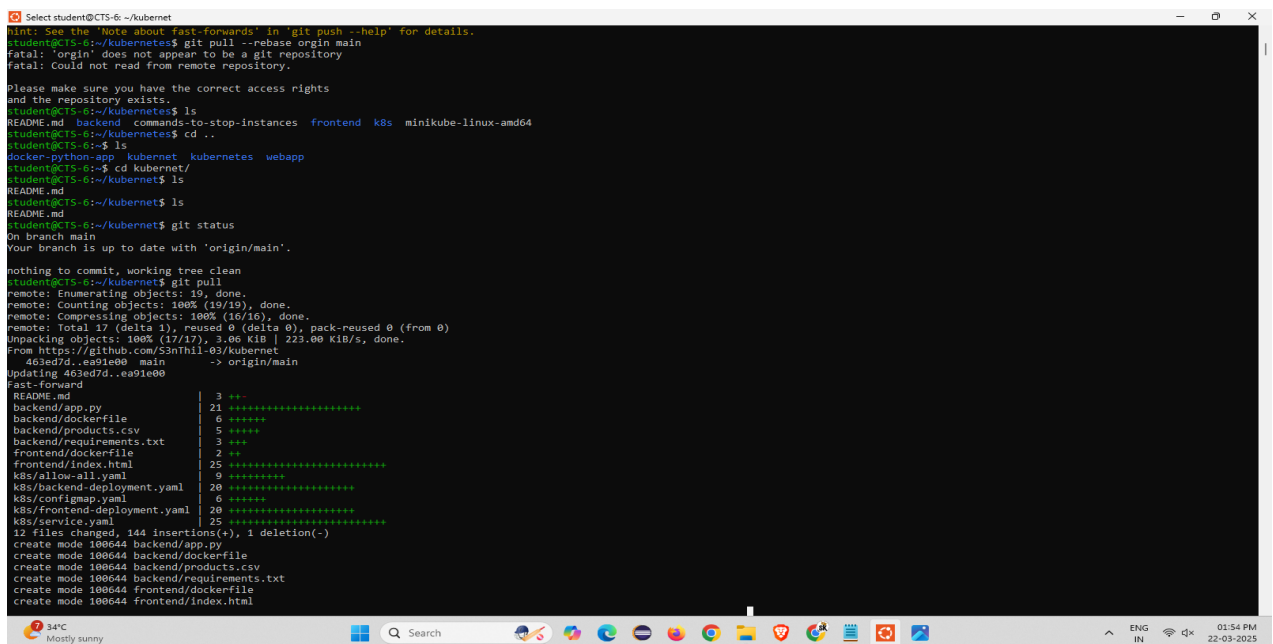
NAME: BALAJI HARIHARAN N S

ROLLNO: 24MCR008

DEVOPS TRAINING

DAY 5 CONFIGURING PIPELINE

Step 1: Create github repository kubernetes and push the cd kubernetes files as frontend ,backend and k8s



```
Select student@CTS-6: ~/kubernetes
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
student@CTS-6:~/kubernetes$ git pull --rebase origin main
fatal: 'origin' does not appear to be a git repository
fatal: Could not read from remote repository.

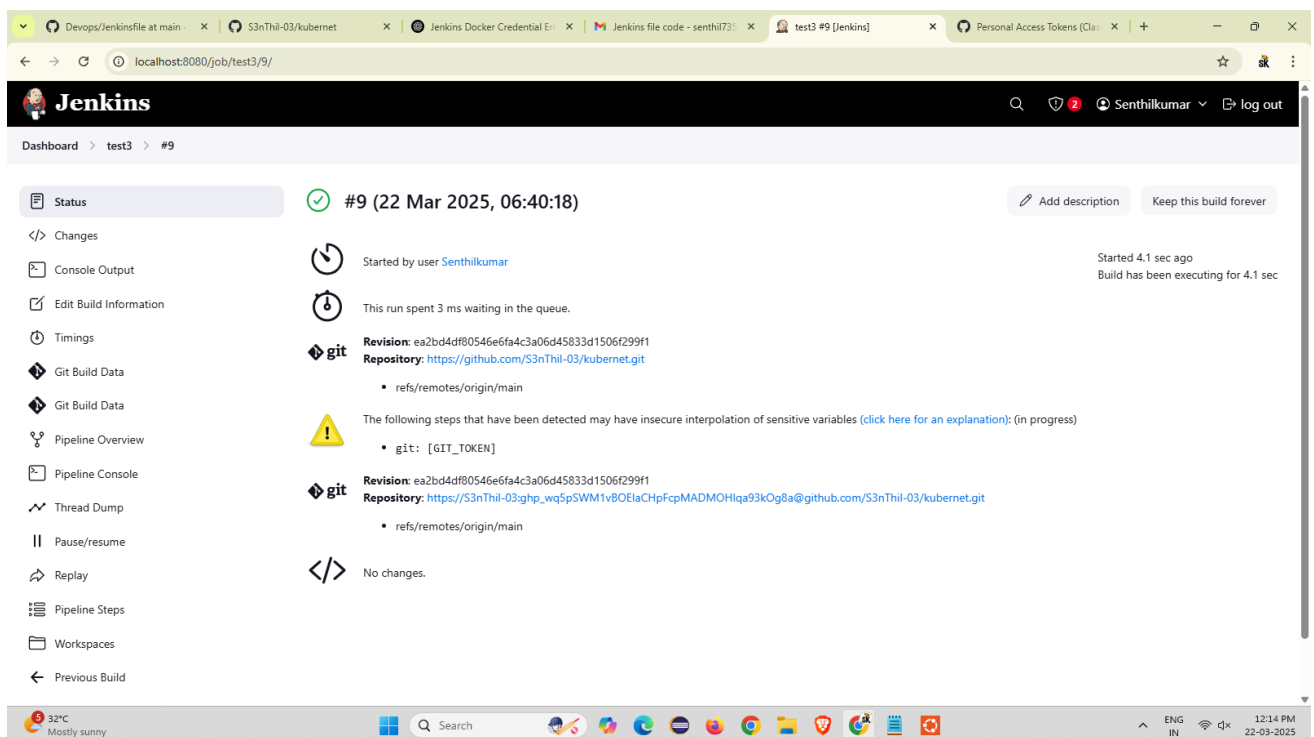
Please make sure you have the correct access rights
and the repository exists.
student@CTS-6:~/kubernetes$ ls
README.md  backend  commands-to-stop-instances  frontend  k8s  minikube-linux-amd64
student@CTS-6:~/kubernetes$ cd ..
student@CTS-6:~$ ls
docker-python-app  kubernetes  kubernet  webapp
student@CTS-6:~/kubernet/$ ls
README.md
student@CTS-6:~/kubernet/$ ls
README.md
student@CTS-6:~/kubernet/$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
student@CTS-6:~/kubernet/$ git pull
remote: Enumerating objects: 19, done.
remote: Counting objects: 100% (19/19), done.
remote: Compressing objects: 100% (16/16), done.
remote: Total 17 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (17/17), 3.06 KiB | 223.00 KiB/s, done.
From https://github.com/53nThil-03/kubernetes
  463ed7d..ea91e00  main    -> origin/main
Updating 463ed7d..ea91e00
Fast-forward
 README.md          | 3 ++
 backend/app.py     | 21 +++++
 backend/dockerfile | 6 +++++
 backend/products.csv | 5 +++++
 backend/requirements.txt | 3 +++
 frontend/dockerfile | 2 ++
 frontend/index.html | 25 +++++
 k8s/allow-all.yaml | 9 +++++
 k8s/backend-deployment.yaml | 20 +++++
 k8s/configmap.yaml | 6 +++++
 k8s/frontend-deployment.yaml | 20 +++++
 k8s/service.yaml | 25 +++++
 12 files changed, 144 insertions(+), 1 deletion(-)
create mode 100644 backend/app.py
create mode 100644 backend/dockerfile
create mode 100644 backend/products.csv
create mode 100644 backend/requirements.txt
create mode 100644 frontend/dockerfile
create mode 100644 frontend/index.html
```

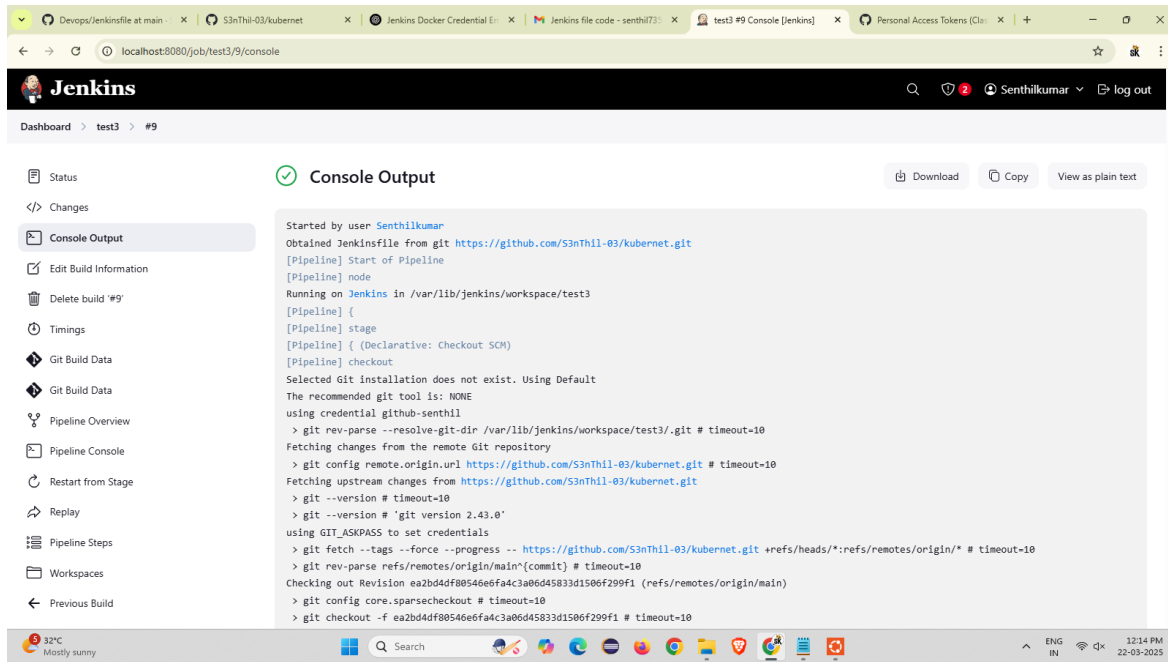
Step 2: create Jenkinsfile and add the following code and push into the github repo kubernetes

```
Select student@CTS-6: ~/Kubernetes
id https://github.com/S3nThil-03/kubern...
$ curl -s https://raw.githubusercontent.com/S3nThil-03/kubern...
$ cat jenkinsfile
pipeline {
    agent any
    environment {
        // Define image names
        BACKEND_IMAGE = "senthil7387/docker-backend:latest"
        FRONTEND_IMAGE = "senthil7387/docker-frontend:latest"
        // Define container names
        BACKEND_CONTAINER = "docker-running-backend"
        FRONTEND_CONTAINER = "docker-running-frontend"
    }
    stages {
        stage("Checkout Code") {
            steps {
                withCredentials([usernamePassword(credentialId: 'github-senthil', usernameVariable: 'GIT_USER', passwordVariable: 'GIT_TOKEN')]) {
                    git url: "https://$GIT_USER:$GIT_TOKEN@github.com:S3nThil-03/kubern...", branch: 'main'
                }
            }
        }
        stage("Build Backend Docker Image") {
            steps {
                // Use the Backend directory as the build context
                sh "docker build -t $BACKEND_IMAGE -f backend/dockerfile backend"
            }
        }
        stage("Build Frontend Docker Image") {
            steps {
                // Use the Frontend directory as the build context
                sh "docker build -t $FRONTEND_IMAGE -f frontend/dockerfile frontend"
            }
        }
        stage("Login to Docker Registry") {
            steps {
                withCredentials([usernamePassword(credentialId: 'dockerhub-senthil', usernameVariable: 'DOCKER_USER', passwordVariable: 'DOCKER_PASS')]) {
                    sh "echo $DOCKER_PASS | docker login -u $DOCKER_USER --password-stdin"
                }
            }
        }
        stage("Push Docker Images") {
            steps {
                sh "docker push $BACKEND_IMAGE"
                sh "docker push $FRONTEND_IMAGE"
            }
        }
        stage("Stop & Remove Existing Containers") {
            steps {
                script {
                    sh "...
                    if [ $(docker ps -aq -f name=$BACKEND_CONTAINER) ]; then
                        docker stop $BACKEND_CONTAINER || true
                        docker rm $BACKEND_CONTAINER || true
                    fi
                    if [ $(docker ps -aq -f name=$FRONTEND_CONTAINER) ]; then
                        docker stop $FRONTEND_CONTAINER || true
                        docker rm $FRONTEND_CONTAINER || true
                    fi
                    sh "...
                }
            }
        }
    }
}
```

Step 3: Open Jenkins create a item in pipeline and click ok and go to configure add the repo url and credentials and click build



Step 4 : open console output and check build is complete or not.



Step 5 : go to dashboard > manage Jenkins > plugins and install the Kubernetes once it all download success will shown.

