DevOps Task 5: Jenkins & Kubernetes Deployment

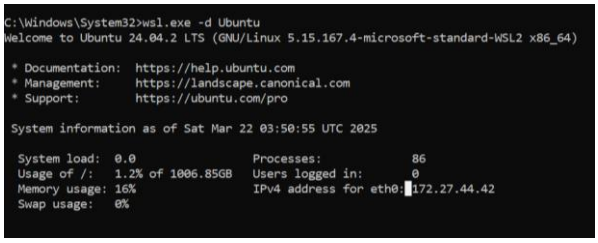
**Step 1:** Setting Up WSL and Ubuntu

1. Open **Windows Subsystem for Linux (WSL)** and start **Ubuntu**:

   ```
   wsl.exe -d Ubuntu
   ```

2. Verify system information and update packages:

   ```
   sudo apt update
   sudo apt upgrade
   ```




**Step 2:** Installing and Configuring Jenkins

1. Start the Jenkins service:

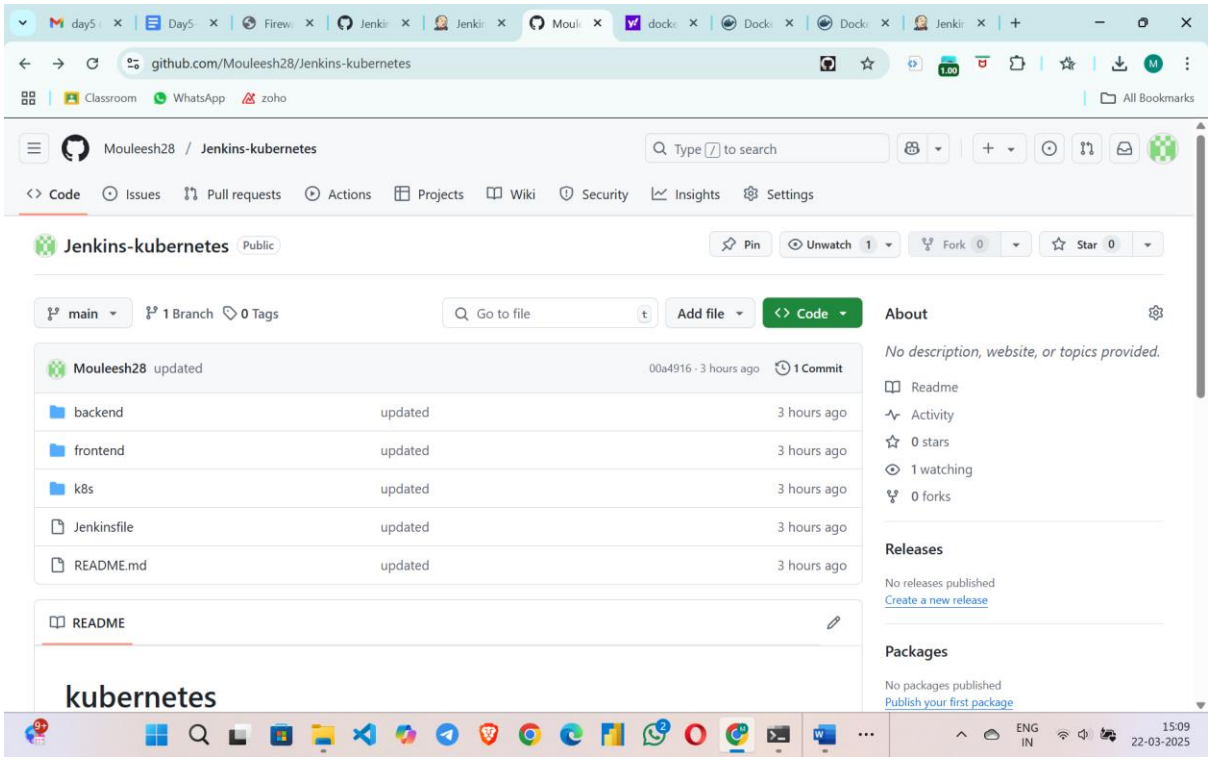
   ```
   sudo systemctl start Jenkins
   ```

2. Enable Jenkins to start on boot:

   ```
   sudo systemctl enable Jenkins
   ```

**Step 3:** Create new repository

1. Add Readme.md file



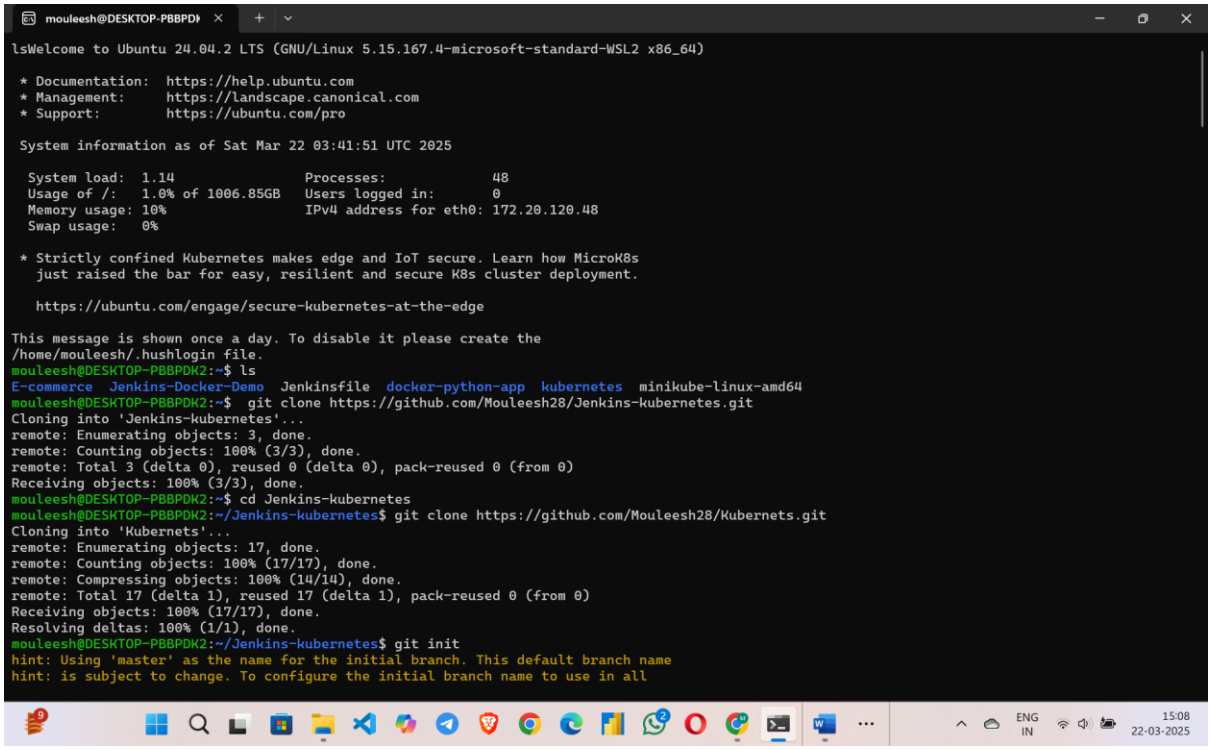

**Step 4:** Cloning Repositories

1. Clone the **Jenkins-Kubernetes** repository:

```
git clone https://github.com/Mouleesh28/Jenkins-Kubernetes.git
cd Jenkins-Kubernetes
```

2. Clone the **Kubernetes** configuration repository:

```
git clone https://github.com/Mouleesh28/kubernetes.git
```




**Step 5:** Creating the Jenkins Pipeline

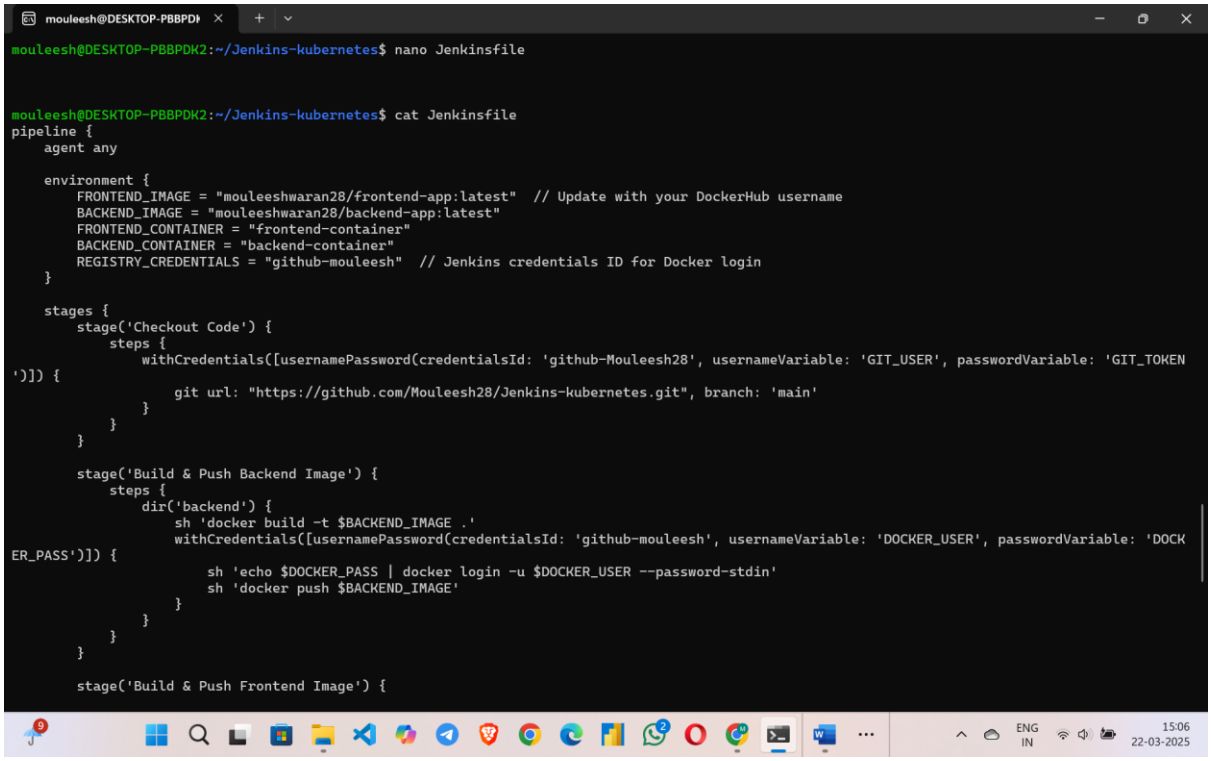
1. Open **Jenkinsfile** using a text editor:

```
nano Jenkinsfile
```

2. Add the following pipeline script:

```
pipeline {
    agent any
    environment {
        FRONTEND_IMAGE = "viratpk18/frontend-app:latest"
        BACKEND_IMAGE = "viratpk18/backend-app:latest"
        FRONTEND_CONTAINER = "frontend-container"
        BACKEND_CONTAINER = "backend-container"
        REGISTRY_CREDENTIALS = "docker-praveen"
    }
    stages {
        stage('Checkout Code') {
            steps {
                withCredentials([usernamePassword(credentialsId: 'github-pk',
usernameVariable: 'GIT_USER', passwordVariable: 'GIT_TOKEN')]) {
                    git url:
"https://$GIT_USER:$GIT_TOKEN@github.com/viratpk18/Jenkins-
Kubernetes.git", branch: 'main'
                }
            }
        }
```

```
        stage('Build & Push Backend Image') {
            steps {
                dir('backend') {
                    sh 'docker build -t $BACKEND_IMAGE .'
                    withCredentials([usernamePassword(credentialsId: 'docker-praveen',
usernameVariable: 'DOCKER_USER', passwordVariable: 'DOCKER_PASS')]) {
                        sh 'echo $DOCKER_PASS | docker login -u $DOCKER_USER --
password-stdin'
                        sh 'docker push $BACKEND_IMAGE'
                    }
                }
            }
        }
        stage('Build & Push Frontend Image') {
            steps {
                dir('frontend') {
                    sh 'docker build -t $FRONTEND_IMAGE .'
                    withCredentials([usernamePassword(credentialsId: 'docker-praveen',
usernameVariable: 'DOCKER_USER', passwordVariable: 'DOCKER_PASS')]) {
                        sh 'echo $DOCKER_PASS | docker login -u $DOCKER_USER --
password-stdin'
                        sh 'docker push $FRONTEND_IMAGE'
                    }
                }
            }
        }
        stage('Deploy Containers') {
            steps {
                sh 'docker run -d -p 5000:5000 --name $BACKEND_CONTAINER
$BACKEND_IMAGE'
                sh 'docker run -d -p 3000:3000 --name $FRONTEND_CONTAINER
$FRONTEND_IMAGE'
            }
        }
    }
    post {
        success {
            echo "Deployment successful!"
        }
        failure {
            echo "Deployment failed."
        }
    }
}
```


mouleesh@DESKTOP-PBBPDK
mouleesh@DESKTOP-PBBPDK2:~/Jenkins-kubernetes$ nano Jenkinsfile
mouleesh@DESKTOP-PBBPDK2:~/Jenkins-kubernetes$ cat Jenkinsfile
pipeline {
agent any
environment {
FRONTEND_IMAGE = "mouleeshwaran28/frontend-app:latest" // Update with your DockerHub username
BACKEND_IMAGE = "mouleeshwaran28/backend-app:latest"
FRONTEND_CONTAINER = "frontend-container"
BACKEND_CONTAINER = "backend-container"
REGISTRY_CREDENTIALS = "github-mouleesh" // Jenkins credentials ID for Docker login
}
stages {
stage('Checkout Code') {
steps {
withCredentials([usernamePassword(credentialsId: 'github-Mouleesh28', usernameVariable: 'GIT_USER', passwordVariable: 'GIT_TOKEN
')]) {
git url: "https://github.com/Mouleesh28/Jenkins-kubernetes.git", branch: 'main'
}
}
}
stage('Build & Push Backend Image') {
steps {
dir('backend') {
sh 'docker build -t $BACKEND_IMAGE .'
withCredentials([usernamePassword(credentialsId: 'github-mouleesh', usernameVariable: 'DOCKER_USER', passwordVariable: 'DOCK
ER_PASS')]) {
sh 'echo $DOCKER_PASS | docker login -u $DOCKER_USER --password-stdin'
sh 'docker push $BACKEND_IMAGE'
}
}
}
}
stage('Build & Push Frontend Image') {
ENG
IN
15:06
22-03-2025

**Step 6:** Committing and Pushing the Jenkinsfile

1. Add and commit changes:

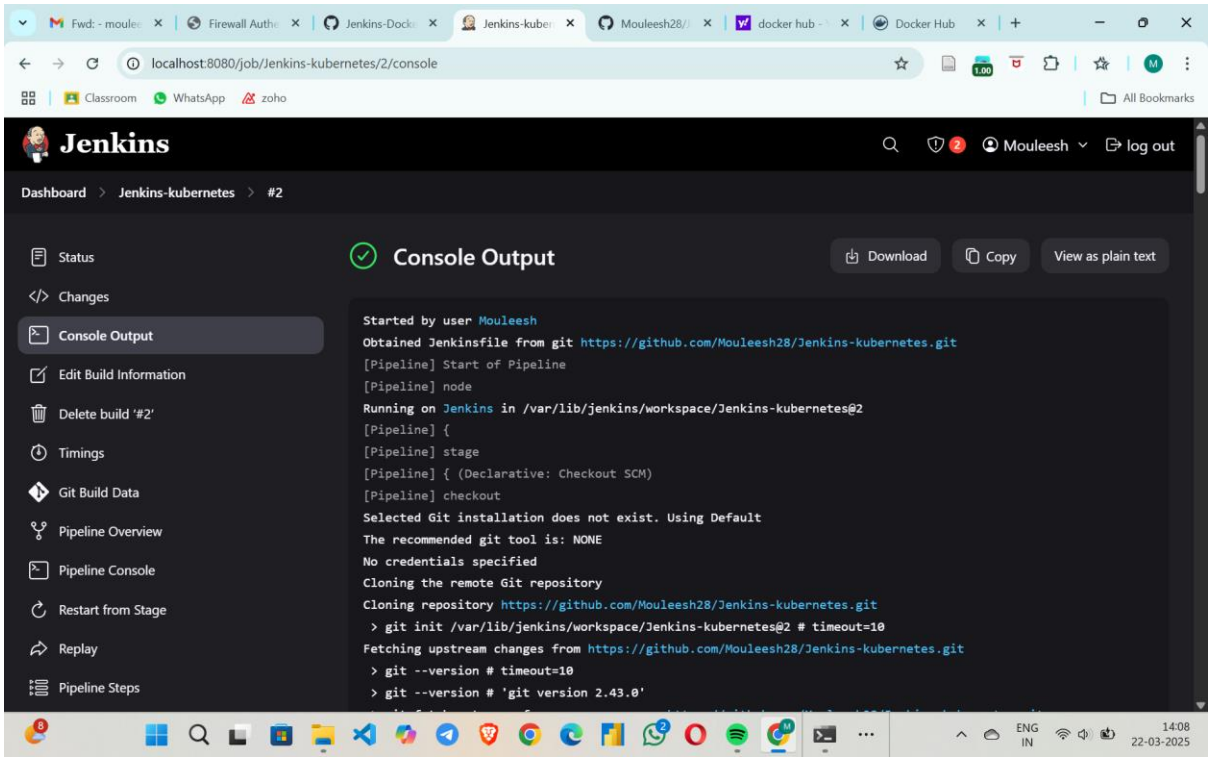
```
git add .
git commit -m "Jenkinsfile added"
```

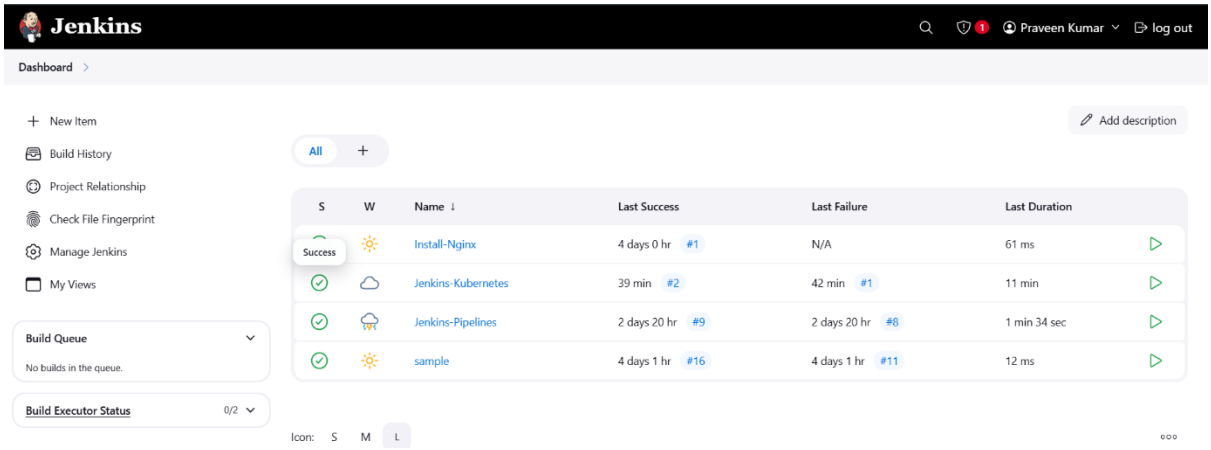
2. Push to GitHub:

```
git push https://github.com/Mouleesh28/Jenkins-Kubernetes.git
```

```
mouleesh@DESKTOP-PBBPDK2:~/Jenkins-kubernetes$ git push https://Mouleesh28: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@github.com/Mouleesh28/Jenkin
s-kubernetes.git --force
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 920 bytes | 920.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/Mouleesh28/Jenkins-kubernetes.git
   00a4916..7cfca1c  main -> main
```

**Step7:** Build the Jenkins






**Step8:** Check in Docker images