DevOps Task: Jenkins & Kubernetes Deployment

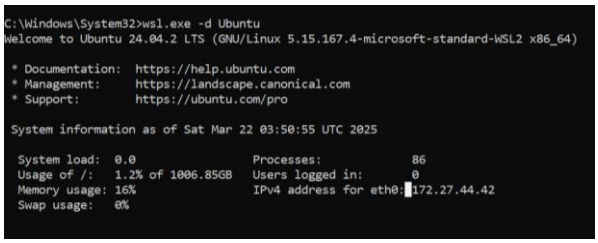
Step 1: Setting Up WSL and Ubuntu

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



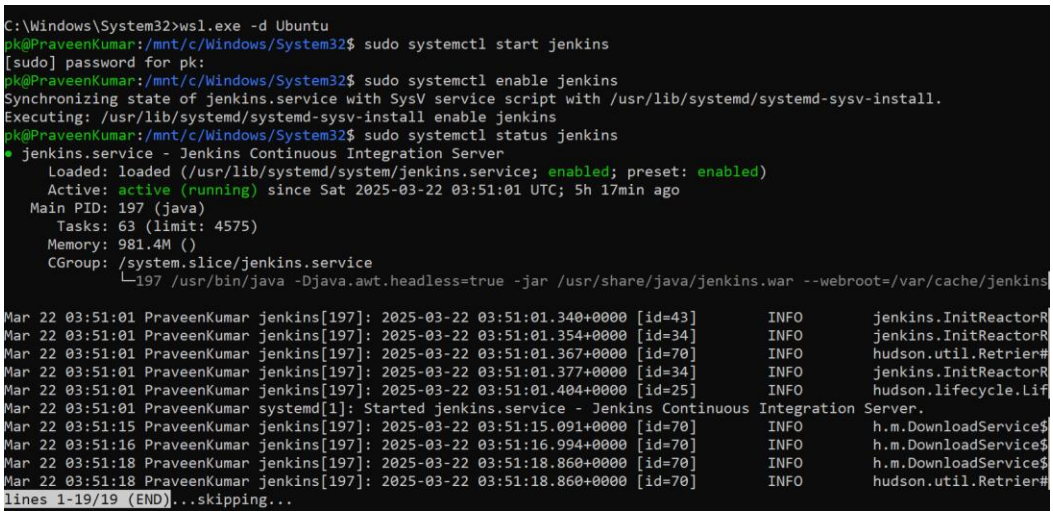
Step 2: Installing and Configuring Jenkins

1. Start the Jenkins service:

   sudo systemctl start Jenkins

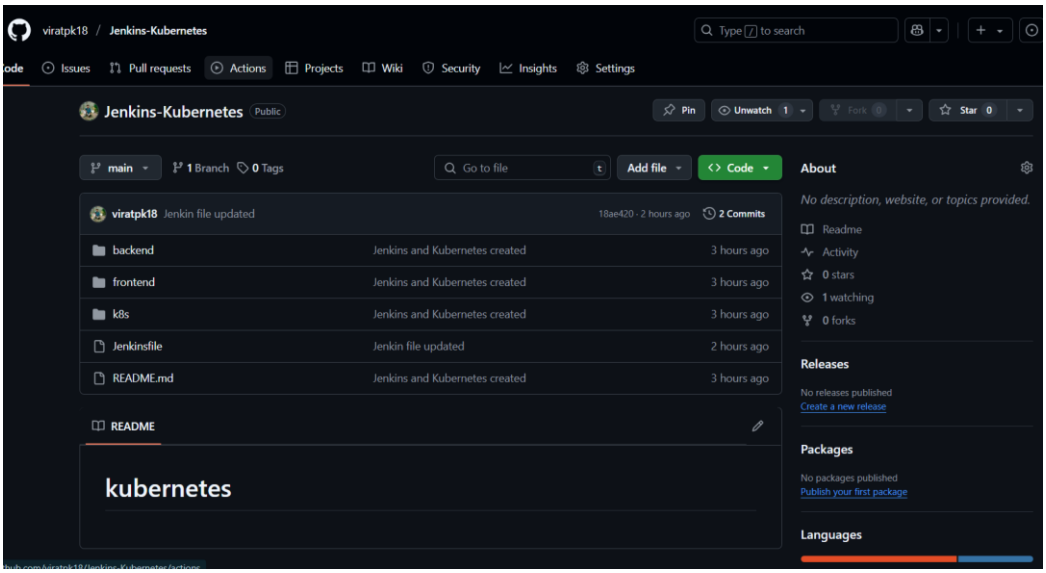
2. Enable Jenkins to start on boot:

   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/viratpk18/Jenkins-Kubernetes.git
cd Jenkins-Kubernetes
```

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

```
git clone https://github.com/viratpk18/kubernetes.git
```

```
pk@PraveenKumar:/mnt/c/Windows/System32$ git clone https://github.com/viratpk18/Jenkins-Kubernetes.git
Cloning into 'Jenkins-Kubernetes'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.
pk@PraveenKumar:/mnt/c/Windows/System32$ cd Jenkins-Kubernetes
pk@PraveenKumar:/mnt/c/Windows/System32/Jenkins-Kubernetes$ ls
README.md
pk@PraveenKumar:/mnt/c/Windows/System32/Jenkins-Kubernetes$ git clone https://github.com/viratpk18/kubernetes.git
Cloning into 'kubernetes'...
remote: Enumerating objects: 18, done.
remote: Counting objects: 100% (18/18), done.
remote: Compressing objects: 100% (15/15), done.
remote: Total 18 (delta 1), reused 18 (delta 1), pack-reused 0 (from 0)
Receiving objects: 100% (18/18), done.
Resolving deltas: 100% (1/1), done.
pk@PraveenKumar:/mnt/c/Windows/System32/Jenkins-Kubernetes$ ls
README.md  kubernetes
pk@PraveenKumar:/mnt/c/Windows/System32/Jenkins-Kubernetes$ ls
README.md  backend  frontend  k8s  kubernetes
```

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."
    }
  }
}
```

```
pk@PraveenKumar:/mnt/c/Windows/System32/Jenkins-Kubernetes$ nano Jenkinsfile
pk@PraveenKumar:/mnt/c/Windows/System32/Jenkins-Kubernetes$ cat Jenkinsfile
pipeline {
    agent any

    environment {
        FRONTEND_IMAGE = "viratpk18/frontend-app:latest"  // Update with your DockerHub username
        BACKEND_IMAGE = "viratpk18/backend-app:latest"
        FRONTEND_CONTAINER = "frontend-container"
        BACKEND_CONTAINER = "backend-container"
        REGISTRY_CREDENTIALS = "docker-praveen"  // Jenkins credentials ID for Docker login
    }

    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'
```

Step 6: Committing and Pushing the Jenkinsfile

1. Add and commit changes:

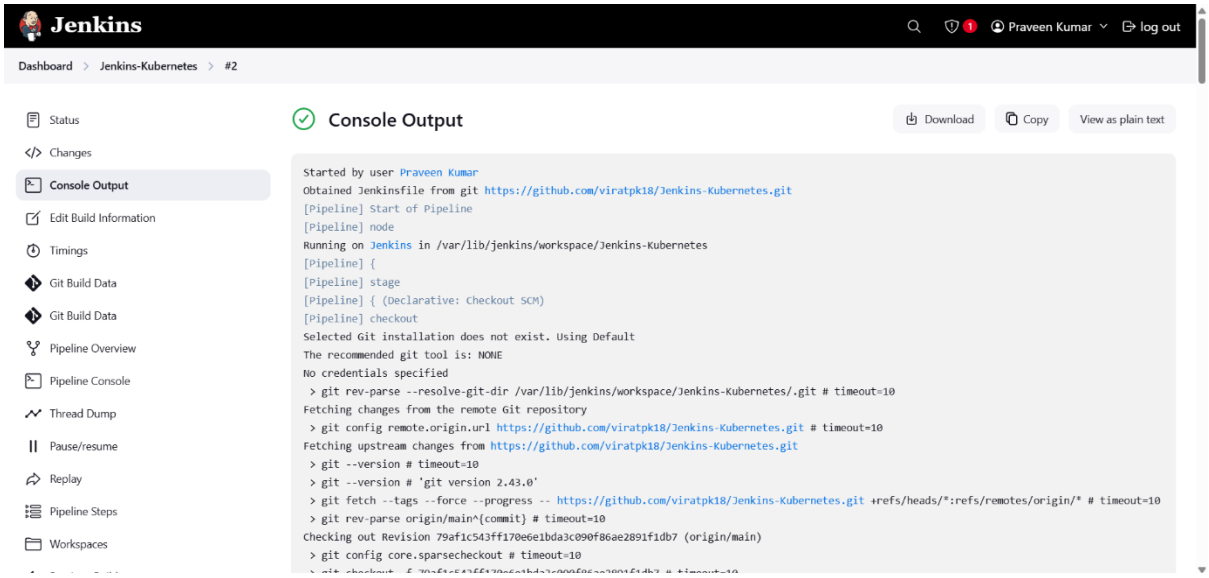
```
git add .
git commit -m "Jenkinsfile added"
```

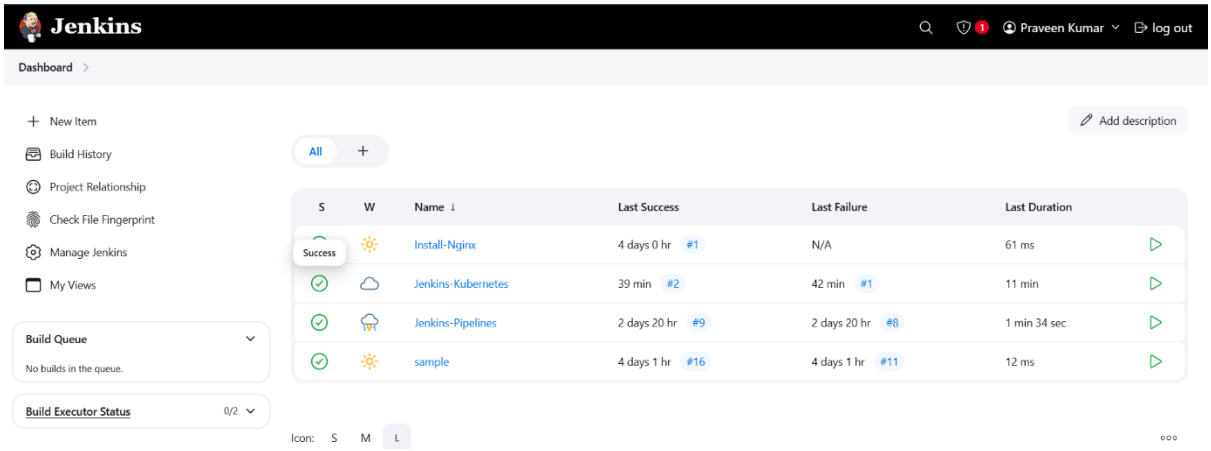
2. Push to GitHub:

```
git push https://github.com/viratpk18/Jenkins-Kubernetes.git
```

```
pk@PraveenKumar:/mnt/c/Windows/System32/Jenkins-Kubernetes$ git push https://viratpk18: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@github.com/viratpk18/Jenkins-Kubernetes.git
Enumerating objects: 18, done.
Counting objects: 100% (18/18), done.
Delta compression using up to 12 threads
Compressing objects: 100% (16/16), done.
Writing objects: 100% (18/18), 3.07 KiB | 27.00 KiB/s, done.
Total 18 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/viratpk18/Jenkins-Kubernetes.git
 + 0b6f9c2...2a549a1 main -> main (forced update)
```

Step7: Build the Jenkins





Step8: Check in Docker images