

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
```

```
C:\Windows\System32>wsl.exe -d Ubuntu
Welcome to Ubuntu 24.04.2 LTS (GNU/Linux 5.15.167.4-microsoft-standard-WSL2 x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

System information as of Sat Mar 22 03:50:55 UTC 2025

System load:  0.0               Processes:    86
Usage of /:   1.2% of 1006.85GB Users logged in: 0
Memory usage: 16%              IPv4 address for eth0: 172.27.44.42
Swap usage:   0%
```

Step 2: Installing and Configuring Jenkins

1. Start the Jenkins service:

```
sudo systemctl start Jenkins
```

2. Enable Jenkins to start on boot:

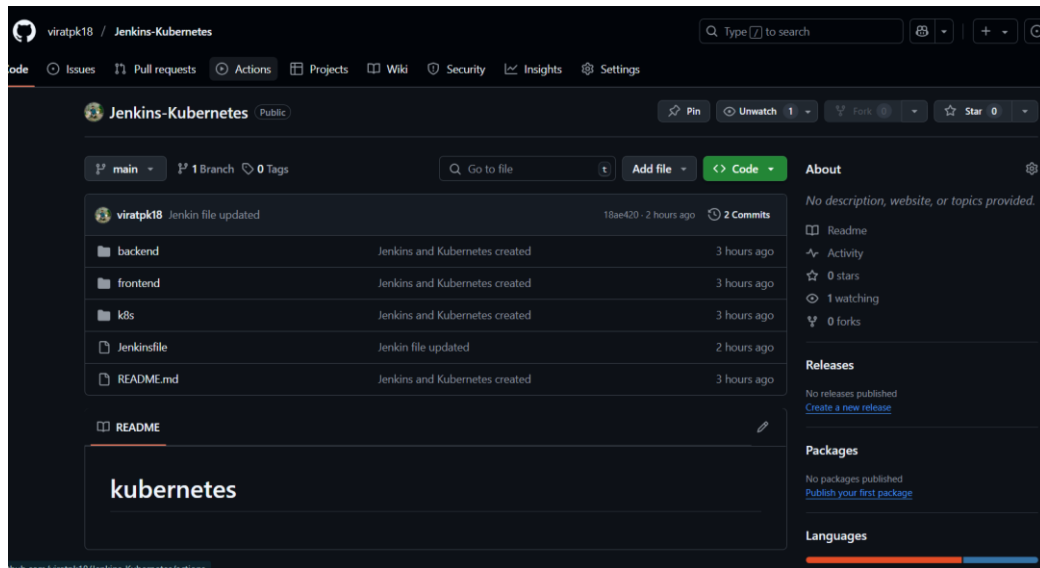
```
sudo systemctl enable Jenkins
```

```
C:\Windows\System32>wsl.exe -d Ubuntu
pk@PraveenKumar:/mnt/c/Windows/System32$ sudo systemctl start jenkins
[sudo] password for pk:
pk@PraveenKumar:/mnt/c/Windows/System32$ sudo systemctl enable jenkins
Synchronizing state of jenkins.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable jenkins
pk@PraveenKumar:/mnt/c/Windows/System32$ sudo systemctl status jenkins
● jenkins.service - Jenkins Continuous Integration Server
   Loaded: loaded (/usr/lib/systemd/system/jenkins.service; enabled; preset: enabled)
   Active: active (running) since Sat 2025-03-22 03:51:01 UTC; 5h 17min ago
     Main PID: 197 (java)
       Tasks: 63 (limit: 4575)
      Memory: 981.4M ( )
      CGroup: /system.slice/jenkins.service
              └─197 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=/var/cache/jenkins

Mar 22 03:51:01 PraveenKumar jenkins[197]: 2025-03-22 03:51:01.340+0000 [id=43] INFO jenkins.InitReactorR
Mar 22 03:51:01 PraveenKumar jenkins[197]: 2025-03-22 03:51:01.354+0000 [id=34] INFO jenkins.InitReactorR
Mar 22 03:51:01 PraveenKumar jenkins[197]: 2025-03-22 03:51:01.367+0000 [id=70] INFO hudson.util.Retrier#
Mar 22 03:51:01 PraveenKumar jenkins[197]: 2025-03-22 03:51:01.377+0000 [id=34] INFO jenkins.InitReactorR
Mar 22 03:51:01 PraveenKumar jenkins[197]: 2025-03-22 03:51:01.404+0000 [id=25] INFO hudson.lifecycle.Lif
Mar 22 03:51:01 PraveenKumar systemd[1]: Started jenkins.service - Jenkins Continuous Integration Server.
Mar 22 03:51:15 PraveenKumar jenkins[197]: 2025-03-22 03:51:15.091+0000 [id=70] INFO h.m.DownloadService$
Mar 22 03:51:16 PraveenKumar jenkins[197]: 2025-03-22 03:51:16.994+0000 [id=70] INFO h.m.DownloadService$
Mar 22 03:51:18 PraveenKumar jenkins[197]: 2025-03-22 03:51:18.860+0000 [id=70] INFO h.m.DownloadService$
Mar 22 03:51:18 PraveenKumar jenkins[197]: 2025-03-22 03:51:18.860+0000 [id=70] INFO hudson.util.Retrier#
lines 1-19/19 (END)...skipping...
```

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."
  }
}
}

```

```

PKSPraaveenKumar@mnt/c/windows/System32/Jenkins-Kubernetes$ nano Jenkinsfile
PKSPraaveenKumar@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)
```

Step 7: Build the Jenkins

The screenshot shows the Jenkins web interface. The top navigation bar includes the Jenkins logo, a search icon, a shield icon, a notification icon, the user name 'Praveen Kumar', and a 'log out' button. The breadcrumb trail is 'Dashboard > Jenkins-Kubernetes > #2'. On the left sidebar, the 'Console Output' tab is selected. The main content area displays the console output for a build, which includes the following text:

```
Started by user Praveen Kumar  
Obtained Jenkinsfile from git https://github.com/viratpk18/Jenkins-Kubernetes.git  
[Pipeline] Start of Pipeline  
[Pipeline] node  
Running on Jenkins in /var/lib/jenkins/workspace/Jenkins-Kubernetes  
[Pipeline] {  
[Pipeline] stage  
[Pipeline] { (Declarative: Checkout SCM)  
[Pipeline] checkout  
Selected Git installation does not exist. Using Default  
The recommended git tool is: NONE  
No credentials specified  
> git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/Jenkins-Kubernetes/.git # timeout=10  
Fetching changes from the remote Git repository  
> git config remote.origin.url https://github.com/viratpk18/Jenkins-Kubernetes.git # timeout=10  
Fetching upstream changes from https://github.com/viratpk18/Jenkins-Kubernetes.git  
> git --version # timeout=10  
> git --version # 'git version 2.43.0'  
> git fetch --tags --force --progress -- https://github.com/viratpk18/Jenkins-Kubernetes.git +refs/heads/*:refs/remotes/origin/* # timeout=10  
> git rev-parse origin/main(commit) # timeout=10  
Checking out Revision 79ef1c543ff170e6e1bda3c090f86ae2891f1db7 (origin/main)  
> git config core.sparsecheckout # timeout=10  
> git checkout -f 79ef1c543ff170e6e1bda3c090f86ae2891f1db7 # timeout=10
```

The screenshot shows the Jenkins Dashboard. The top navigation bar is identical to the previous screenshot. The breadcrumb trail is 'Dashboard >'. On the left sidebar, the 'Build Queue' section is expanded, showing 'No builds in the queue.' and the 'Build Executor Status' as '0/2'. The main content area displays a table of builds with the following columns: 'S', 'W', 'Name', 'Last Success', 'Last Failure', and 'Last Duration'. The table contains four rows of build data:

S	W	Name	Last Success	Last Failure	Last Duration
Success	Sun	Install-Nginx	4 days 0 hr #1	N/A	61 ms
Success	Cloud	Jenkins-Kubernetes	39 min #2	42 min #1	11 min
Success	Cloud	Jenkins-Pipelines	2 days 20 hr #9	2 days 20 hr #8	1 min 34 sec
Success	Sun	sample	4 days 1 hr #16	4 days 1 hr #11	12 ms

Step8: Check in Docker images

