

Day 5: DevOps

Create GitHub repo:

"Add and commit all files to the remote repository."

```
bhuvan_kumar@bhuvan:~/kubernetes$ cd ..
bhuvan_kumar@bhuvan:~$ mkdir demo-kuber
mkdir: cannot create directory 'demo-kuber': File exists
bhuvan_kumar@bhuvan:~$ cd demo-kuber
bhuvan_kumar@bhuvan:~/demo-kuber$ cp frontend backend k8s demo-kuber
cp: target 'demo-kuber': No such file or directory
bhuvan_kumar@bhuvan:~/demo-kuber$ cp -r frontend backend k8s demo-kuber/
cp: target 'demo-kuber/': No such file or directory
bhuvan_kumar@bhuvan:~/demo-kuber$ cd ..
bhuvan_kumar@bhuvan:~$ cp -r frontend backend k8s demo-kuber/
cp: cannot stat 'frontend': No such file or directory
cp: cannot stat 'backend': No such file or directory
cp: cannot stat 'k8s': No such file or directory
bhuvan_kumar@bhuvan:~$ cd kubernetes
bhuvan_kumar@bhuvan:~/kubernetes$ cp -r frontend backend k8s demo-kuber/
bhuvan_kumar@bhuvan:~/kubernetes$ cd ..
bhuvan_kumar@bhuvan:~$ cd demo-kuber
bhuvan_kumar@bhuvan:~/demo-kuber$ git init
Reinitialized existing Git repository in /home/bhuvan_kumar/demo-kuber/.git/
bhuvan_kumar@bhuvan:~/demo-kuber$ git remote add origin https://github.com/bhuvanmsc/demo-kuber.git
error: remote origin already exists.
bhuvan_kumar@bhuvan:~/demo-kuber$ git remote -v
origin https://github.com/bhuvanmsc/demo-kuber.git (fetch)
origin https://github.com/bhuvanmsc/demo-kuber.git (push)
bhuvan_kumar@bhuvan:~/demo-kuber$ git remote -v
origin https://github.com/bhuvanmsc/demo-kuber.git (fetch)
origin https://github.com/bhuvanmsc/demo-kuber.git (push)
bhuvan_kumar@bhuvan:~/demo-kuber$ git add .
bhuvan_kumar@bhuvan:~/demo-kuber$ git commit -m "Initial commit"
Author identity unknown

*** Please tell me who you are.

Run

  git config --global user.email "you@example.com"
  git config --global user.name "Your Name"

to set your account's default identity.
Omit --global to set the identity only in this repository.

fatal: empty ident name (for <bhuvan_kumar@bhuvan.>) not allowed
bhuvan_kumar@bhuvan:~/demo-kuber$ git config --global user.email "bhuvanmsc2023@gmail.com"
bhuvan_kumar@bhuvan:~/demo-kuber$ ^C
bhuvan_kumar@bhuvan:~/demo-kuber$ git config --global user.name "bhuvanmsc"
bhuvan_kumar@bhuvan:~/demo-kuber$ git add .
bhuvan_kumar@bhuvan:~/demo-kuber$ git commit -m "Initial commit"
On branch main
Your branch is up to date with 'origin/main'.
```

Push local changes to the remote repository

```
bhuvan_kumar@bhuvan:~$ cd kubernetes
bhuvan_kumar@bhuvan:~/kubernetes$ cp -r frontend backend k8s demo-kuber/
bhuvan_kumar@bhuvan:~/kubernetes$ cd ..
bhuvan_kumar@bhuvan:~$ cd demo-kuber
bhuvan_kumar@bhuvan:~/demo-kuber$ git init
Reinitialized existing Git repository in /home/bhuvan_kumar/demo-kuber/.git/
bhuvan_kumar@bhuvan:~/demo-kuber$ git remote add origin https://github.com/bhuvanmsc/demo-kuber.git
error: remote origin already exists.
bhuvan_kumar@bhuvan:~/demo-kuber$ git remote -v
origin https://github.com/bhuvanmsc/demo-kuber.git (fetch)
origin https://github.com/bhuvanmsc/demo-kuber.git (push)
bhuvan_kumar@bhuvan:~/demo-kuber$ git remote -v
origin https://github.com/bhuvanmsc/demo-kuber.git (fetch)
origin https://github.com/bhuvanmsc/demo-kuber.git (push)
```

Create a Jenkins config file **Jenkinsfile**

A **Jenkinsfile** is a script that defines a **Jenkins pipeline**. It outlines how Jenkins should build, test, and deploy your code. Instead of manually configuring Jenkins jobs through the web interface, you can store the entire pipeline as code in your repository

```
pipeline {
  agent any
  environment {
    DOCKER_IMAGE = "bhuvankumar17/docker-app:latest" // Change this to your registry
    CONTAINER_NAME = "docker-running-app-1"
    REGISTRY_CREDENTIALS = "doc" // Jenkins credentials ID
  }
  stages {
    stage('Checkout Code') {
      steps {
        withCredentials([usernamePassword(credentialsId: 'bhuvanmsc', usernameVariable: 'GIT_USER', passwordVariable: 'GIT_TOKEN')]) {
          git url: "https://$GIT_USER:$GIT_TOKEN@github.com/bhuvanmsc/demo-kuber.git", branch: 'main'
        }
      }
    }
    stage('Build Docker Image') {
      steps {
        sh 'docker build -t $DOCKER_IMAGE .'
      }
    }
    stage('Login to Docker Registry') {
      steps {
        withCredentials([usernamePassword(credentialsId: 'docker-hub-credentials', usernameVariable: 'DOCKER_USER', passwordVariable: 'DOCKER_PASS')]) {
          sh 'echo $DOCKER_PASS | docker login -u $DOCKER_USER --password-stdin'
        }
      }
    }
    stage('Push to Container Registry') {
      steps {
        sh 'docker push $DOCKER_IMAGE'
      }
    }
  }
}
```

Enable and Start the Jenkins

```
bhuvan_kumar@bhuvan:~/demo-kuber$ sudo systemctl enable jenkins
[sudo] password for bhuvan_kumar:
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
bhuvan_kumar@bhuvan:~/demo-kuber$ sudo systemctl start jenkins
```

Vist the Jenkins in Port 8080

Jenkins Pipe Line

Create a Jenkins Pipeline

Push the Jenkins file to the Remote Repo

Install **Kubernetes plugin** in Jenkins

Dashboard > Manage Jenkins > Plugins

Plugins

Updates

Available plugins

Installed plugins

Advanced settings

Q

kubernetes plugin

Name	Enabled
Kubernetes Client API Plugin 6.10.0-251.v556f5f100500 Kubernetes Client API plugin for use by other Jenkins plugins. Report an issue with this plugin	<div><div></div></div>
Kubernetes Credentials Plugin 192.v4d5b_1c429d17 Common classes for Kubernetes credentials Report an issue with this plugin	<div><div></div></div>
Kubernetes plugin 4324.vfec199a_33512 This plugin integrates Jenkins with Kubernetes Report an issue with this plugin	<div><div></div></div>

Build the Jenkins

Jenkins

Dashboard > Manage Jenkins > Plugins

Plugins

Updates

Available plugins

Installed plugins

Advanced settings

Download progress

Download progress

Preparation

- Checking internet connectivity
- Checking update center connectivity
- Success

Kubernetes Client API

Success

Authentication Tokens API

Success

Kubernetes Credentials

Success

Kubernetes

Success

Loading plugin extensions

Success

→

[Go back to the top page](#)

(you can start using the installed plugins right away)

→

☐ Restart Jenkins when installation is complete and no jobs are running

REST API Jenkins 2.492.2

Dashboard > day5

Status

Changes

Build Now

Configure

Delete Pipeline

Stages

Rename

Pipeline Syntax

day5

Add description

Permalinks

- Last build (#4), 3 hr 54 min ago
- Last stable build (#4), 3 hr 54 min ago
- Last successful build (#4), 3 hr 54 min ago
- Last failed build (#3), 3 hr 58 min ago
- Last unsuccessful build (#3), 3 hr 58 min ago
- Last completed build (#4), 3 hr 54 min ago

Builds

Filter

Today

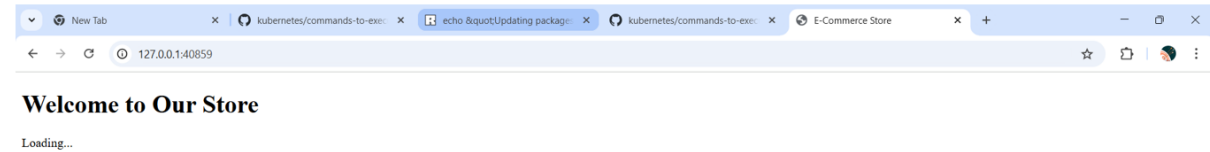
Success

 #4 4:45 am

Build successful

To test Frontend

```
bhuvan_kumar@bhuvan: ~/kubernetes$ minikube service frontend-service --url
http://127.0.0.1:40859
! Because you are using a Docker driver on linux, the terminal needs to be open to run it.
```



To Test backend

```
service/backend-service created
service/frontend-service created
bhuvan_kumar@bhuvan: ~/kubernetes$ kubectl apply -f k8s/configmap.yaml
configmap/backend-config created
bhuvan_kumar@bhuvan: ~/kubernetes$ kubectl get pods
NAME                READY   STATUS    RESTARTS   AGE
backend-dfd8d5579-hpkn5  1/1     Running   0           92s
frontend-6cfd7c46-lk22p  1/1     Running   0           31s
bhuvan_kumar@bhuvan: ~/kubernetes$ kubectl get svc
NAME                TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)          AGE
backend-service     ClusterIP   10.104.231.125  <none>        5000/TCP          32s
frontend-service    NodePort    10.103.7.217   <none>        3000:31640/TCP    32s
kubernetes           ClusterIP   10.96.0.1      <none>        443/TCP           11h
bhuvan_kumar@bhuvan: ~/kubernetes$ minikube service frontend-service --url
http://127.0.0.1:40859
! Because you are using a Docker driver on linux, the terminal needs to be open to run it.
^Chhuvan_kumar@bhuvan: ~/kubernetes$ kubectl get nodes -o wide
NAME                STATUS    ROLES          AGE   VERSION   INTERNAL-IP   EXTERNAL-IP   OS-IMAGE             KERNEL-VERSION        CONTAINER-RUNTIME
minikube            Ready    control-plane  11h   v1.32.0    192.168.49.2  <none>        Ubuntu 22.04.5 LTS   5.15.167.4-microsoft-standard-WSL2  docker://27.4.1
bhuvan_kumar@bhuvan: ~/kubernetes$ kubectl run test-pod --image=alpine --restart=Never -it -- sh
If you don't see a command prompt, try pressing enter.
/ #
/ # apk add curl
fetch https://dl-cdn.alpinelinux.org/alpine/v3.21/main/x86_64/APKINDEX.tar.gz
fetch https://dl-cdn.alpinelinux.org/alpine/v3.21/community/x86_64/APKINDEX.tar.gz
(1/9) Installing brotli-libs (1.1.0-r2)
(2/9) Installing c-ares (1.34.3-r0)
(3/9) Installing libunistring (1.2-r0)
(4/9) Installing libidn2 (2.3.7-r0)
(5/9) Installing nghttp2-libs (1.64.0-r0)
(6/9) Installing libpsl (0.21.5-r3)
(7/9) Installing zstd-libs (1.5.6-r2)
(8/9) Installing libcurl (8.12.1-r1)
(9/9) Installing curl (8.12.1-r1)
Executing busybox-1.37.0-r12.trigger
OK: 12 MiB in 24 packages
/ # curl http://backend-service:5000/products
[{"id":1,"name":"Smartphone","price":299.99},{ "id":2,"name":"Laptop","price":799.99},{ "id":3,"name":"Headphones","price":49.99},{ "id":4,"name":"Tablet","price":199.99}]
```

Note: We expect this kind of output because we are running this frontend on localhost.

— COMPLETED —