1. **Install Kubernetes on Ubuntu**
   1. **Install Dependencies**

sudo apt update

sudo apt install -y apt-transport-https ca-certificates curl

* 1. **Install kubectl**

curl -LO "https://dl.k8s.io/release/$(curl -L -s https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl"

chmod +x kubectl

sudo mv kubectl /usr/local/bin/

kubectl version –client

* 1. **Install minikube**

curl -LO https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64

chmod +x minikube-linux-amd64

sudo mv minikube-linux-amd64 /usr/local/bin/minikube

* 1. **Start Minikube**

minikube start --driver=docker

* 1. **Check status**

kubectl cluster-info

1. **Create project**
   1. **Create folder**

mkdir flask-ci-cd

cd flask-ci-cd

* 1. **Create Flask app**

**nano app.py**

from flask import Flask

app = Flask(\_\_name\_\_)

@app.route('/')

def home():

return "Hello, World!"

if \_\_name\_\_ == "\_\_main\_\_":

app.run(host="0.0.0.0", port=5000)

* 1. **Create Docker File**

FROM python:3.9

WORKDIR /app

COPY requirements.txt requirements.txt

RUN pip install -r requirements.txt

COPY . .

CMD ["python", "app.py"]

* 1. **Create requirements.txt**

Flask

1. **Build and Push Docker Image**

docker build -t kpkm25/flask-ci-cd:latest .

docker login

docker push kpkm25/flask-ci-cd:latest

1. **Connect Kubernetes to Docker**

Create a secret for dockerhub(put in your details)

kubectl create secret docker-registry docker-hub-secret \

--docker-server=https://index.docker.io/v1/ \

--docker-username=kpkm25 \

--docker-password=YOUR\_DOCKER\_PASSWORD \

--docker-email=YOUR\_EMAIL

1. **Kubernetes Deployment**

Create k8s-deployment.yaml

apiVersion: apps/v1

kind: Deployment

metadata:

name: flask-app

spec:

replicas: 2

selector:

matchLabels:

app: flask-app

template:

metadata:

labels:

app: flask-app

spec:

imagePullSecrets:

- name: docker-hub-secret

containers:

- name: flask-app

image: kpkm25/flask-ci-cd:latest

ports:

- containerPort: 5000

---

apiVersion: v1

kind: Service

metadata:

name: flask-service

spec:

selector:

app: flask-app

ports:

- protocol: TCP

port: 80

targetPort: 5000

type: LoadBalancer

Apply the deployment:

kubectl apply -f k8s-deployment.yaml

Check if the pods are running:  
kubectl get pods

1. **How to stop and restart deployment**

kubectl delete -f k8s-deployment.yaml

kubectl apply -f k8s-deployment.yaml

1. **Jenkins Pipeline**

pipeline {

agent any

stages {

stage('Clone Code') {

steps {

git 'https://github.com/kpkm25/flask-ci-cd.git'

}

}

stage('Build Docker Image') {

steps {

sh 'docker build -t kpkm25/flask-ci-cd:latest .'

}

}

stage('Push to Docker Hub') {

steps {

withDockerRegistry([credentialsId: 'docker-creds', url: '']) {

sh 'docker push kpkm25/flask-ci-cd:latest'

}

}

}

stage('Deploy to Kubernetes') {

steps {

sh 'kubectl apply -f k8s-deployment.yaml'

}

}

}

}