**Experiment 5: Deploy a Simple AI Application using Kubernetes and Manage Pods and Services**

**Learning Objective:** Students should be able to understand how to deploy a simple AI application using Kubernetes and manage its lifecycle, including creating deployments, services, and managing pods.

**Tools:** Kubernetes, Docker, Python, Minikube, VS Code

**Theory:**

Kubernetes is an open-source container orchestration system for automating the deployment, scaling, and management of containerized applications. It ensures high availability and scalability while abstracting the complexities of managing multiple containers.

In this experiment, we deploy a simple AI application using Kubernetes. The AI application is a Flask-based service that classifies text using a Naïve Bayes model. We first containerize the application using Docker, push the image to a repository, and then deploy it using Kubernetes. The Kubernetes cluster manages the application’s lifecycle, ensuring that the service remains available and scalable.

#### ****Step 1: Create a Simple AI Application****

Create a Python script app.py:

from flask import Flask, request, jsonify

from sklearn.feature\_extraction.text import CountVectorizer

from sklearn.naive\_bayes import MultinomialNB

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

# Training data

train\_data = ["hello world", "AI is amazing", "Kubernetes is powerful", "ML is the future"]

labels = [0, 1, 1, 1] # 0 = General, 1 = AI-related

vectorizer = CountVectorizer()

X\_train = vectorizer.fit\_transform(train\_data)

model = MultinomialNB()

model.fit(X\_train, labels)

@app.route('/predict', methods=['POST'])

def predict():

data = request.json.get("text")

X\_test = vectorizer.transform([data])

prediction = model.predict(X\_test)[0]

return jsonify({"prediction": "AI-related" if prediction == 1 else "General"})

if \_\_name\_\_ == '\_\_main\_\_':

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

#### ****Step 2: Create a Dockerfile****

FROM python:3.9

WORKDIR /app

COPY app.py requirements.txt /app/

RUN pip install -r requirements.txt

EXPOSE 5000

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

Create requirements.txt:

* Flask
* scikit-learn

#### ****Step 3: Build and Push Docker Image****

docker build -t my-ai-app

docker tag my-ai-app <ansh>/my-ai-app:latest

docker push <ansh>/my-ai-app:latest

#### ****Step 4: Deploy the Application on Kubernetes****

**Create deployment.yaml:**

apiVersion: apps/v1

kind: Deployment

metadata:

name: ai-app-deployment

spec:

replicas: 2

selector:

matchLabels:

app: ai-app

template:

metadata:

labels:

app: ai-app

spec:

containers:

- name: ai-app

image: <ansh>/my-ai-app:latest

ports:

- containerPort: 5000

**Create service.yaml:**

apiVersion: v1

kind: Service

metadata:

name: ai-app-service

spec:

selector:

app: ai-app

ports:

- protocol: TCP

port: 80

targetPort: 5000

type: LoadBalancer

#### ****Step 5: Apply Kubernetes Configurations****

kubectl apply -f deployment.yaml

kubectl apply -f service.yaml

#### ****Step 6: Check Deployment Status****

kubectl get deployments

kubectl get pods

kubectl get services

#### ****Step 7: Test the AI Application****

**Get the external IP:**

kubectl get service ai-app-service

**Test with curl:**

curl -X POST http://<EXTERNAL-IP>:80/predict -H "Content-Type: application/json" -d '{"text": "AI is great"}'

#### ****Step 8: Scale and Manage Pods****

* Scale Up to 3 Pods:

kubectl scale deployment ai-app-deployment --replicas=3

* Delete Deployment:

kubectl delete deployment ai-app-deployment

* Delete Service:

kubectl delete service ai-app-service

### ****Conclusion:****

In this experiment, we successfully deployed a simple AI application using Kubernetes. We learned how to create a Docker container, push the image to a repository, deploy it using Kubernetes, and manage its lifecycle. By scaling pods and managing services, we demonstrated Kubernetes' ability to provide high availability and resilience in containerized environments.

### ****Viva Questions:****

1. What is Kubernetes?
2. What is the purpose of a Kubernetes Deployment?
3. How does a Kubernetes Service work?
4. What command is used to scale up pods in Kubernetes?
5. How do you delete a running pod in Kubernetes?

**For Faculty Use:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Correction Parameters** | **Formative Assessment**  **[40%]** | **Timely completion of Practical**  **[ 40%]** | **Attendance/ Learning**  **Attitude [20%]** |  |
| **Marks**  **Obtained** |  |  |  |