Name: **Vishal S** USN: **1BM21IS206** Section: **7 D**

Q 75.1: Minikube single node cluster

Steps:

- Install Minikube and start a Kubernetes cluster on your local machine.

- Write a Python Flask application that serves a simple API endpoint at /brave-falcon-46,

returning the message: {"health": "All systems operational"}.

- Create a Docker image for the Flask application and push to your docker hub public container

registry.

- Deploy the application to the Minikube cluster using kubectl with the following requirements:

- Use a Kubernetes Deployment with 2 replicas of the Flask application.

- Expose the Deployment using a Service of type NodePort.

- Configure the Service to expose the application on port 32484 on the Minikube host.

- Verify the deployment by accessing the /brave-falcon-46 endpoint through the exposed

NodePort.

- Demonstrate the scaling of the application by increasing the replicas to 3 and verifying the

updated Deployment.

1. **Install Minikube** <https://minikube.sigs.k8s.io/docs/start/>
2. **Start minikube**

‘’’ minikube start ‘’’



1. **Create app.py**

‘’’ from flask import Flask

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

@app.route('/brave-falcon-46')

def health\_check():

return {"health": "All systems operational"}

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

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

1. **Dockerfile**

‘’’ # Use an official Python runtime as a parent image

FROM python:3.8-slim

# Set the working directory in the container

WORKDIR /app

# Copy the current directory contents into the container at /app

COPY . /app

# Install Flask

RUN pip install flask

# Make port 5000 available to the world outside this container

EXPOSE 5000

# Define environment variable

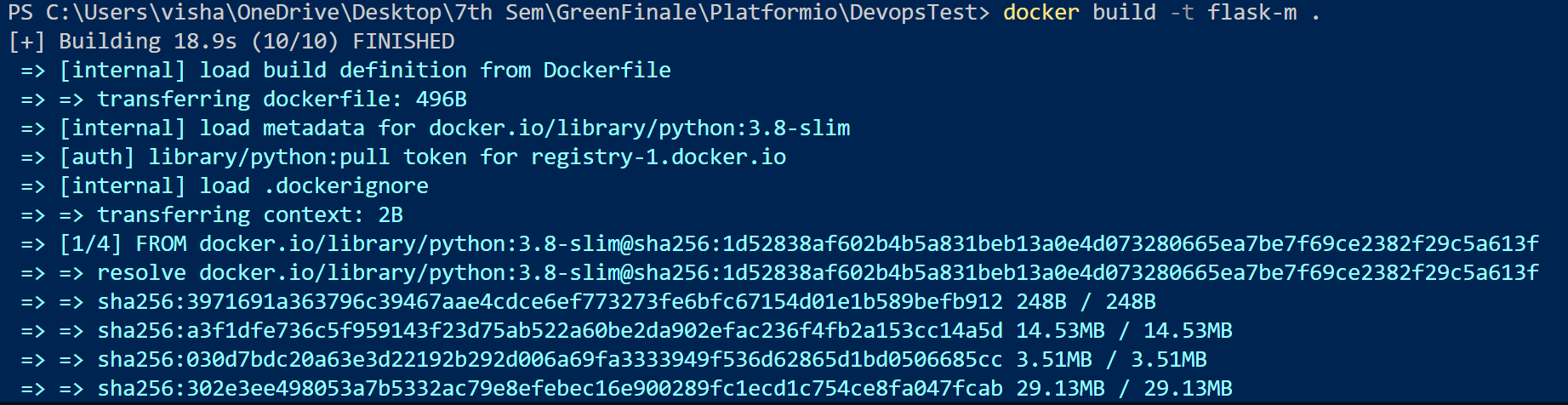
ENV NAME World

# Run app.py when the container launches

CMD ["python", "app.py"] ‘’’

1. **Build Docker image**

‘’’ docker build -t flask-mini ‘’’



1. **Push image to dockerhub**

‘’’ docker push vishalsantosh2003/flask-m ‘’’

1. **Deployment.yaml**

apiVersion: apps/v1

kind: Deployment

metadata:

  name: flask-app-deployment

spec:

  replicas: 2

  selector:

    matchLabels:

      app: flask-app

  template:

    metadata:

      labels:

        app: flask-app

    spec:

      containers:

      - name: flask-m

        image: vishalsantosh2003/flask-m

        ports:

        - containerPort: 5000

1. **Service.yaml**

apiVersion: v1

kind: Service

metadata:

  name: flask-app-service

spec:

  type: NodePort

  selector:

    app: flask-app

  ports:

  - protocol: TCP

    port: 5000

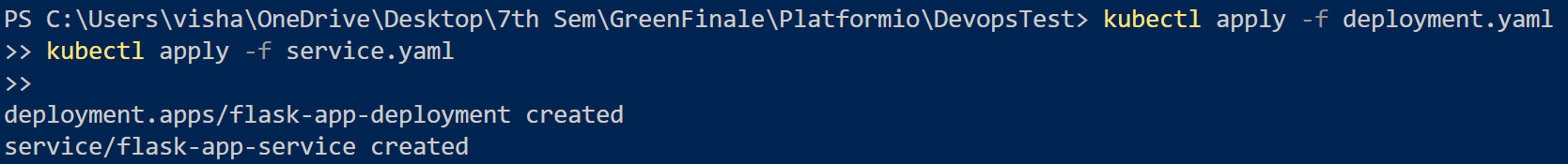
    targetPort: 5000

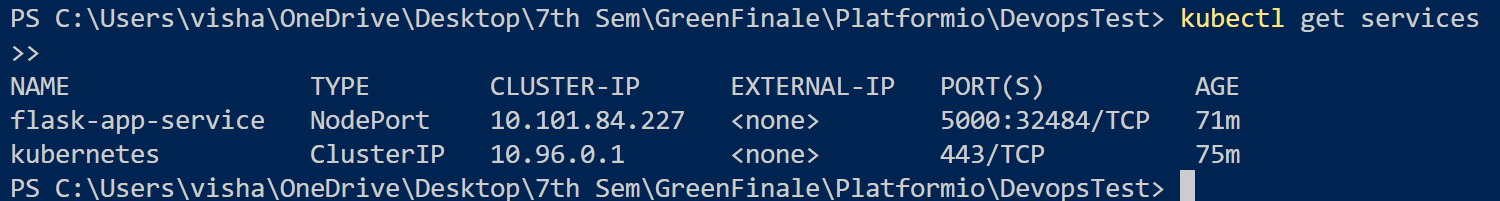
    nodePort: 32484

1. **Apply the deployment and service:**

‘’’ kubectl apply -f deployment.yaml

kubectl apply -f service.yaml ‘’’



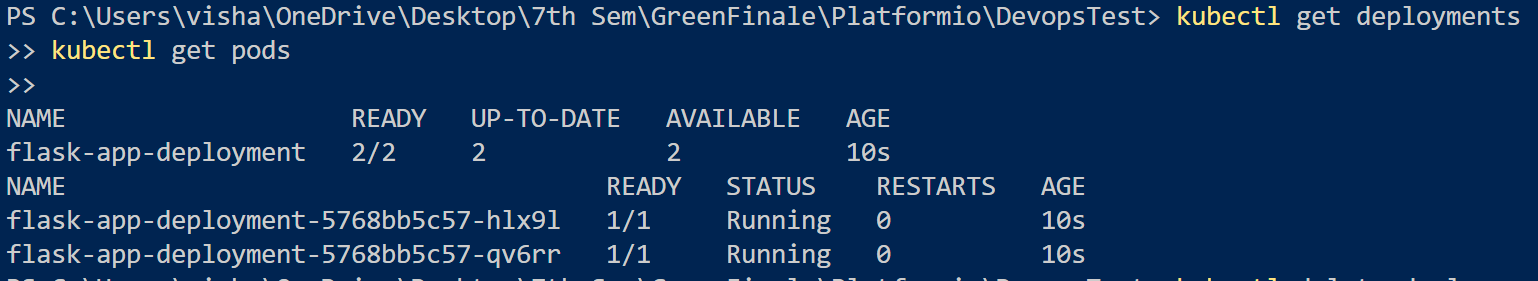


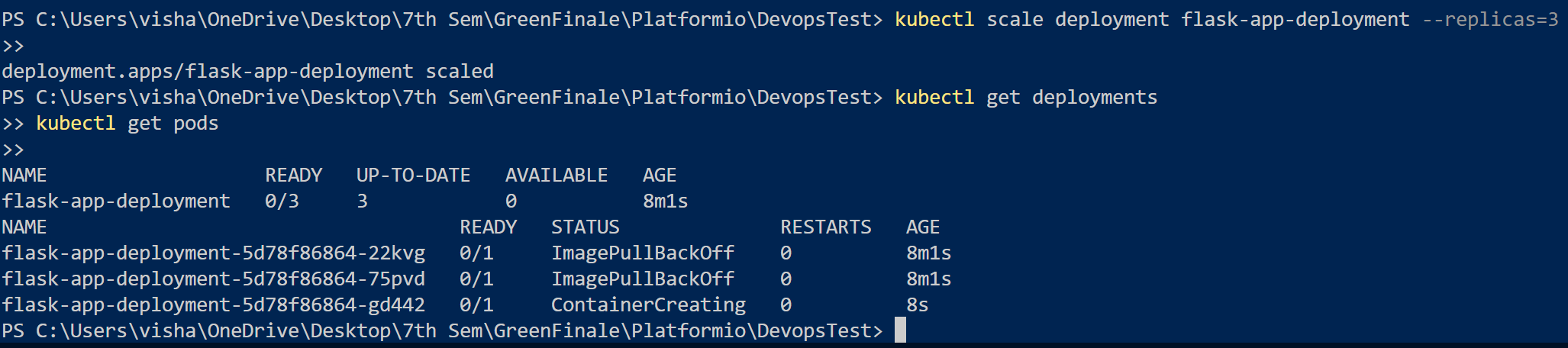
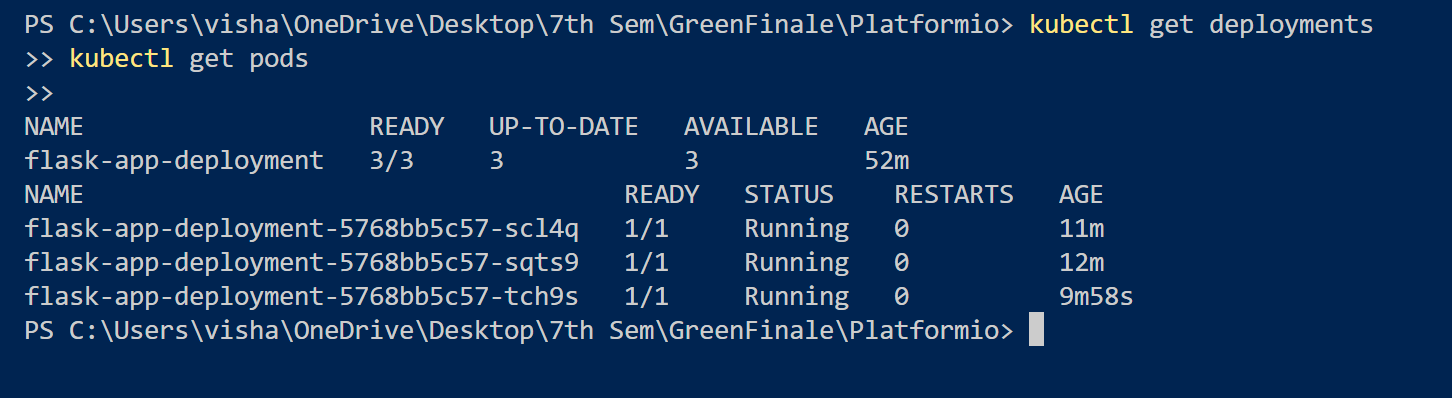
1. **Get minikube ip**

‘’’ minikube ip’’’

Output: 192.168.49.2

1. **Initial deployment with 2 replicas:**

****

1. **Scale the deployment to 3 replicas and verify the updated deployment:**
2. **Output: Access the Flask application at http:/192.168.49.2:32484/brave-falcon-46**

