

```
# app.py
from flask import Flask, jsonify

app = Flask(__name__)

@app.route('/wise-cobra-50')
def wise_cobra():
    return jsonify(message="Everything is OK")
```

```
if __name__ == '__main__':
    app.run(host='0.0.0.0', port=5093)
```

- Create a Docker image for the Flask application and push it to a local or public container Registry.

command: `$ docker build -t flask-app .`

```
# Dockerfile
FROM python:3.9-slim
WORKDIR /app
COPY . /app
RUN pip install flask
EXPOSE 5093
CMD ["python", "app.py"]
```

```
vardhan@ubuntu:~/Documents/DevOps/lab$ docker build -t flask-app .
[+] Building 10.5s (10/10) FINISHED                                docker:default
=> [internal] load build definition from Dockerfile                0.0s
=> => transferring dockerfile: 160B                                0.0s
=> [internal] load metadata for docker.io/library/python:3.9-slim 3.7s
=> [auth] library/python:pull token for registry-1.docker.io      0.0s
=> [internal] load .dockerignore                                  0.0s
=> => transferring context: 2B                                      0.0s
=> [1/4] FROM docker.io/library/python:3.9-slim@sha256:caaf1af9e23adc6149e5d20662b267ead9505868ff07c7673dc4a7166951cfea 2.4s
=> => resolve docker.io/library/python:3.9-slim@sha256:caaf1af9e23adc6149e5d20662b267ead9505868ff07c7673dc4a7166951cfea 0.0s
=> => sha256:75a2bc32319e3d6a5bc12888b074f86338a9e3a4304d99d1dddc07155b8ba76e 14.93MB / 14.93MB 1.1s
=> => sha256:caaf1af9e23adc6149e5d20662b267ead9505868ff07c7673dc4a7166951cfea 10.41kB / 10.41kB 0.0s
=> => sha256:467c454a5863379d6dc810bdfbc963e877c1a887a11dddc2fca9702d2fd2f2fa 1.75kB / 1.75kB 0.0s
=> => sha256:47e5116b5e1de854ad39b1868d3ea37555ff2ed7edb8dccc618db2427d490ae 5.28kB / 5.28kB 0.0s
=> => sha256:fd674058ff8f8cfa7fb8a20c006fc0128541cbbad7f7f7f28df570d08f9e4d92 28.23MB / 28.23MB 0.9s
=> => sha256:96df0e5e81799ba220e250fc3d2c1da017b41302df5954c705bec1407dcab03 3.32MB / 3.32MB 1.8s
=> => extracting sha256:fd674058ff8f8cfa7fb8a20c006fc0128541cbbad7f7f7f28df570d08f9e4d92 0.6s
=> => sha256:d381bf0bf064697fb23da0e7d50d080411236ffcd2853432e12fd0f6b372778 248B / 248B 2.1s
=> => extracting sha256:96df0e5e81799ba220e250fc3d2c1da017b41302df5954c705bec1407dcab03 0.1s
=> => extracting sha256:75a2bc32319e3d6a5bc12888b074f86338a9e3a4304d99d1dddc07155b8ba76e 0.3s
=> => extracting sha256:d381bf0bf064697fb23da0e7d50d080411236ffcd2853432e12fd0f6b372778 0.0s
=> [internal] load build context                                  0.0s
=> => transferring context: 3.73kB                                  0.0s
=> [2/4] WORKDIR /app                                           0.1s
=> [3/4] COPY . /app                                           0.0s
=> [4/4] RUN pip install flask                                  4.1s
=> => exporting to image                                           0.1s
=> => exporting layers                                           0.1s
=> => writing image sha256:022304fe2f43594570936729cfad34c7cdc29fa2549b48786653d5e866cecfac 0.0s
=> => naming to docker.io/library/flask-app                      0.0s
vardhan@ubuntu:~/Documents/DevOps/lab$
```

- Deploy the Flask application to the Kubernetes cluster using a ReplicaSet with the following specifications:

- The ReplicaSet should run 5 replicas of the Flask application.

Command: `$ kubectl get pods`

```
vardhan@ubuntu:~/Documents/DevOps/lab$ kubectl get pods
NAME                READY   STATUS    RESTARTS   AGE
flask-app-rs-7plz4  1/1     Running   0           83s
flask-app-rs-cbdp5  1/1     Running   0           4s
flask-app-rs-cxzgt  1/1     Running   0           4s
flask-app-rs-dvmwx  1/1     Running   0           83s
flask-app-rs-kxz9w  1/1     Running   0           83s
vardhan@ubuntu:~/Documents/DevOps/lab$
```

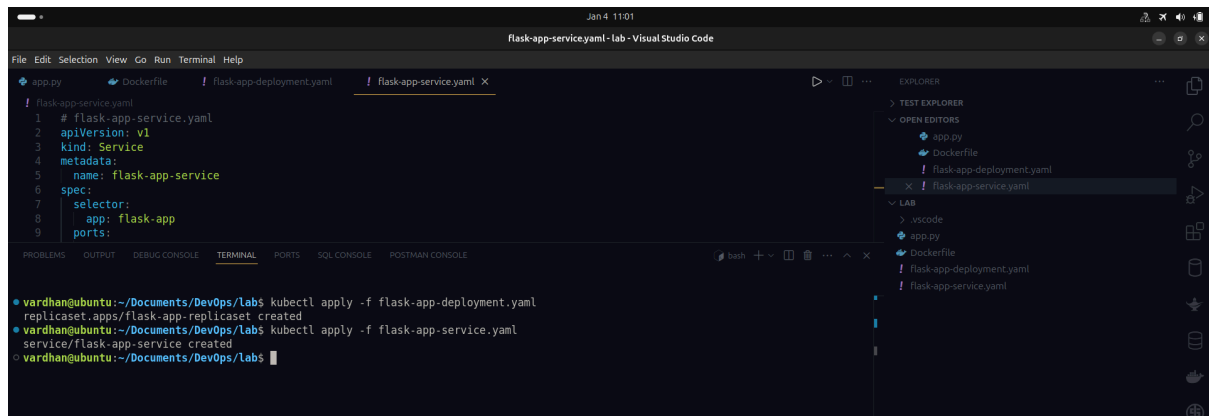
- Use the label app=flask-app to manage the pods.
- Expose the application on port 5093 within the cluster.

```
# flask-app-deployment.yaml
apiVersion: apps/v1
kind: ReplicaSet
metadata:
  name: flask-app-rs
spec:
  replicas: 5
  selector:
    matchLabels:
      app: flask-app
  template:
    metadata:
      labels:
        app: flask-app
    spec:
      containers:
        - name: flask-app
          image: flask-app:latest
          imagePullPolicy: Never
          ports:
            - containerPort: 5093
```

```
# flask-app-service.yaml
apiVersion: v1
kind: Service
metadata:
  name: flask-app-service
spec:
  selector:
    app: flask-app
  ports:
    - protocol: TCP
      port: 5093
      targetPort: 5093
      nodePort: 30093
  type: NodePort
```

Run commands:

```
$ kubectl apply -f flask-app-deployment.yaml
$ kubectl apply -f flask-app-service.yaml
```



Initialize minikube, build Image and Verify the ReplicaSet:

```
$ minikube docker-env
$ eval $(minikube docker-env)
$ docker build -t flask-app .
```

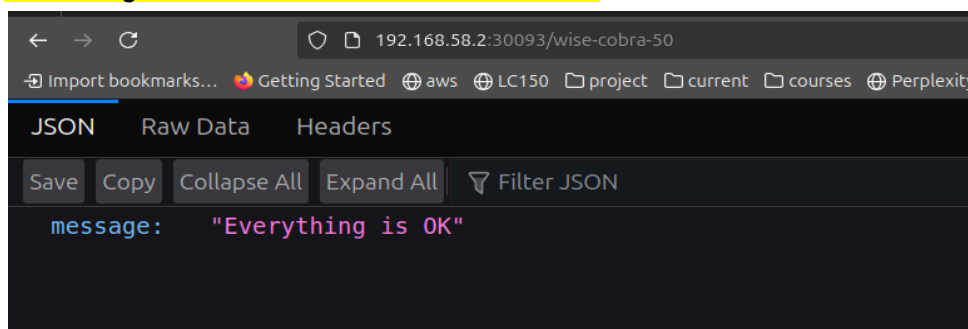
- Test the deployment by accessing the /wise-cobra-50 endpoint through a Kubernetes Service of type NodePort.

Get Ip of minikube :

```
$ minikube ip
192.168.52.2
```

Since node port is : 30093

Accessing : 192.168.52.2:30093/wise-cobra-50



- Scale the ReplicaSet to 7 replicas and demonstrate that the new pods are successfully Created.

Commands:

```
$ kubectl scale rs flask-app-rs --replicas=7
$ kubectl get rs
$ kubectl get pods
```

The screenshot shows a Visual Studio Code interface with a terminal window open. The terminal displays the following commands and output:

```
vardhan@ubuntu:~/Documents/DevOps/Lab$ kubectl scale rs flask-app-rs --replicas=7
replicaset.apps/flask-app-rs scaled
vardhan@ubuntu:~/Documents/DevOps/Lab$ kubectl get rs
NAME          DESIRED   CURRENT   READY   AGE
flask-app-rs  7         7         7       3m36s
vardhan@ubuntu:~/Documents/DevOps/Lab$ kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
flask-app-rs-7plz4  1/1     Running   0          3m42s
flask-app-rs-bkncw  1/1     Running   0          12s
flask-app-rs-cbdp5  1/1     Running   0          2m23s
flask-app-rs-cxztg  1/1     Running   0          2m23s
flask-app-rs-dvmwx  1/1     Running   0          3m42s
flask-app-rs-j6kn4  1/1     Running   0          12s
flask-app-rs-kxz9w  1/1     Running   0          3m42s
```

The right sidebar shows the Explorer view with the following files:

- app.py
- Dockerfile
- flask-app-deployment.yaml
- flask-app-service.yaml

- Delete one of the pods manually and verify that the ReplicaSet automatically recreates it to maintain the desired state.

Commands :

```
$ kubectl delete pod flask-app-rs-7plz4
```

```
$ kubectl get pods
```

After deleting the pod `flask-app-rs-7plz4` another pod with name `flask-app-rs-tljhv` is created.

The screenshot shows a Visual Studio Code interface with a terminal window open. The terminal displays the following commands and output:

```
vardhan@ubuntu:~/Documents/DevOps/Lab$ kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
flask-app-rs-7plz4  1/1     Running   0          3m42s
flask-app-rs-bkncw  1/1     Running   0          12s
flask-app-rs-cbdp5  1/1     Running   0          2m23s
flask-app-rs-cxztg  1/1     Running   0          2m23s
flask-app-rs-dvmwx  1/1     Running   0          3m42s
flask-app-rs-j6kn4  1/1     Running   0          12s
flask-app-rs-kxz9w  1/1     Running   0          3m42s
vardhan@ubuntu:~/Documents/DevOps/Lab$ kubectl delete pod flask-app-rs-7plz4
pod "flask-app-rs-7plz4" deleted
vardhan@ubuntu:~/Documents/DevOps/Lab$
vardhan@ubuntu:~/Documents/DevOps/Lab$ kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
flask-app-rs-bkncw  1/1     Running   0          98s
flask-app-rs-cbdp5  1/1     Running   0          3m49s
flask-app-rs-cxztg  1/1     Running   0          3m49s
flask-app-rs-dvmwx  1/1     Running   0          5m8s
flask-app-rs-j6kn4  1/1     Running   0          98s
flask-app-rs-kxz9w  1/1     Running   0          5m8s
flask-app-rs-tljhv  1/1     Running   0          33s
vardhan@ubuntu:~/Documents/DevOps/Lab$
```