Homework 4

Kubernetes Hands-On – Minikube Deployment

CS 7319 – Software Architecture and Design
Instructor: Dr. Isaac Chow
Due Date: September 21, 2025

Objective

Implement a Java Spring Boot REST API or with the language of your choice that serves **five (5) inspirational quotes**—selected **randomly** on each request—from reputable sources (such as famous people). This app is containerized with Docker and deployed on **Minikube** via a **Kubernetes Deployment (4 replicas)** exposed through a **Service.**

Functional Requirements

1. **Endpoint:** GET /api/quotes returns a JSON array of **exactly 4** quotes chosen at random from a local pool of at least **10** quotes.
2. **Schema:** Each quote object includes text (quotation), and author (attribution).
3. **Port:** Application listens on **8080** (align with container/Kubernetes settings).
4. **Homepage:** A minimal static page at / that fetches and displays the five quotes from /api/quotes.

Containerization & Kubernetes

1. **Dockerize** the service and confirm it runs locally.
2. **Deploy to Minikube** using a **Deployment with 4 replicas** and a **Service** for access (NodePort is acceptable in Minikube).
3. **Verification:** Show endpoint access on Minikube.

Deliverables

1. Screenshots showing successful runs

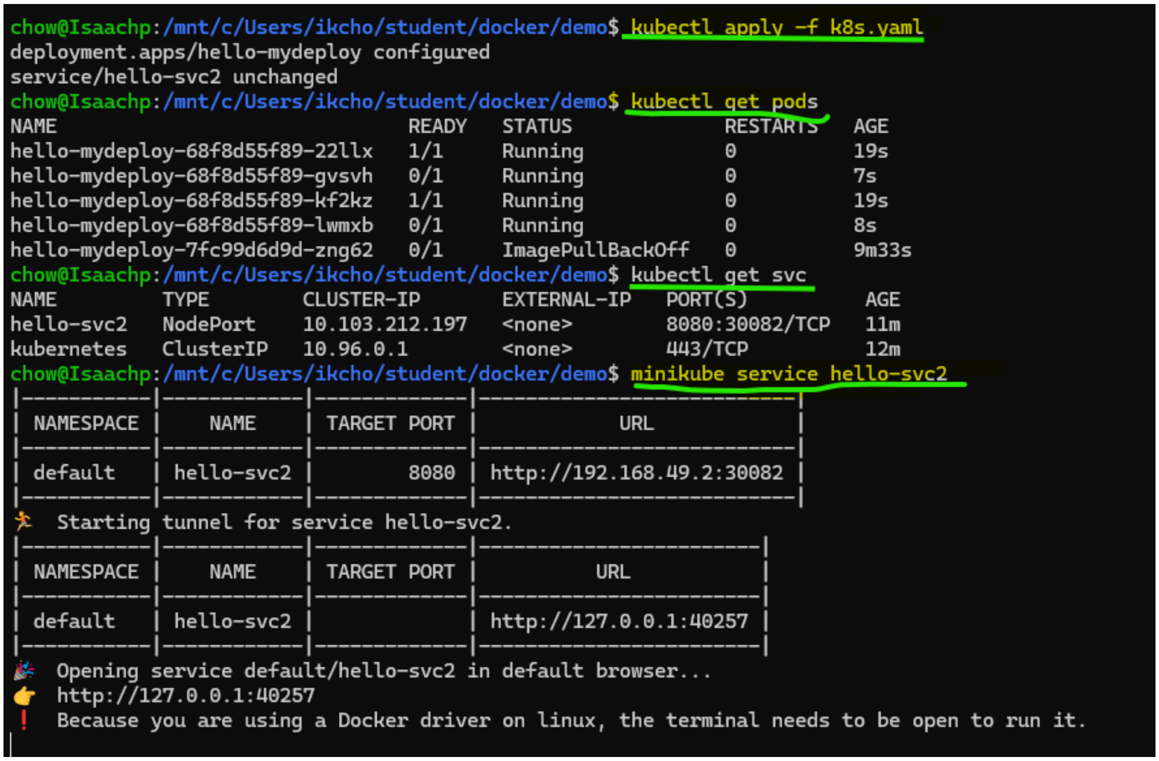
2. Source Code and Dockerfile

4. Kubernetes manifests (e.g. k8s.yaml)

Useful Commands for this homework:

```
$ eval $(minikube docker-env)
$ docker build -t myhello:latest .
$ docker login
$ docker tag myhello:latest chow1668/myhello:latest
                                                        #tag your image for Docker Hub
$ docker push chow1668/myhello:latest
                                                        #push the image to Docker Hub
```

```
$ kubectl apply -f k8s.yaml
$ kubectl get pods
$ kubectl get svc
$ minikube service hello-svc2
```



```
chow@Isaachp:/mnt/c/Users/ikcho/student/docker/demo$ kubectl apply -f k8s.yaml
deployment.apps/hello-mydeploy configured
service/hello-svc2 unchanged
chow@Isaachp:/mnt/c/Users/ikcho/student/docker/demo$ kubectl get pods
NAME                              READY   STATUS             RESTARTS   AGE
hello-mydeploy-68f8d55f89-22llx   1/1     Running            0          19s
hello-mydeploy-68f8d55f89-gvsvh   0/1     Running            0          7s
hello-mydeploy-68f8d55f89-kf2kz   1/1     Running            0          19s
hello-mydeploy-68f8d55f89-lwmxb   0/1     Running            0          8s
hello-mydeploy-7fc99d6d9d-zng62   0/1     ImagePullBackOff   0          9m33s
chow@Isaachp:/mnt/c/Users/ikcho/student/docker/demo$ kubectl get svc
NAME         TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)          AGE
hello-svc2   NodePort    10.103.212.197   <none>        8080:30082/TCP   11m
kubernetes   ClusterIP   10.96.0.1        <none>        443/TCP          12m
chow@Isaachp:/mnt/c/Users/ikcho/student/docker/demo$ minikube service hello-svc2
|-----------|------------|-------------|---------------------------|
| NAMESPACE |    NAME    | TARGET PORT |            URL            |
|-----------|------------|-------------|---------------------------|
| default   | hello-svc2 |        8080 | http://192.168.49.2:30082 |
|-----------|------------|-------------|---------------------------|
🏃  Starting tunnel for service hello-svc2.
|-----------|------------|-------------|------------------------|
| NAMESPACE |    NAME    | TARGET PORT |          URL           |
|-----------|------------|-------------|------------------------|
| default   | hello-svc2 |             | http://127.0.0.1:40257 |
|-----------|------------|-------------|------------------------|
🎉  Opening service default/hello-svc2 in default browser...
👉  http://127.0.0.1:40257
❗  Because you are using a Docker driver on linux, the terminal needs to be open to run it.
```

Useful kubectl Commands

Once your API is containerized, deployed via Kubernetes/Minikube with 4 replicas, and exposed via a Service, you'll need these commands:

- **Apply Kubernetes manifests** (Deployment + Service):
  - kubectl apply -f k8s.yaml
- **Check what's running**:
  - kubectl get pods
  - kubectl get deploy
  - kubectl get svc
- **Review replication and rollout status**:
  - kubectl rollout status deployment/<deployment-name>
- **Scale replicas as needed** (beyond the original 4):
  - kubectl scale deployment/<deployment-name> --replicas=6

- **Inspect resource details**:
  - kubectl describe pods
  - kubectl describe svc
- **Access application endpoint** (in Minikube):
  - minikube service <service-name> --url

Other Helpful Commands

- List pods across all namespaces:
  - kubectl get pods --all-namespaces
- Get more detail (-o wide):
  - kubectl get pods -o wide
- View logs:
  - kubectl logs <pod-name>
- Stop the app entirely (scale to zero)
  - kubectl scale deployment/<deployment-name> --replicas=0