

Final Project

Đề bài: Xây dựng và triển khai một ứng dụng guestbook đơn giản, ứng dụng bao gồm:

- Có **frontend web** (form nhập text + submit).
- Triển khai bằng **Kubernetes Deployments + Pods**.
- Thực hiện **Horizontal Pod Autoscaling (HPA)**.
- Thử nghiệm **Rolling Updates & Rollbacks**.

Bước 1: Kiểm tra môi trường và công cụ

Bật terminal, clone repo chứa các file cần thiết về:

```
[ ! -d 'guestbook' ] && git clone https://github.com/ibm-developer-skills-net/work/guestbook
```

Vào thư mục project và kiểm tra các file bằng lệnh ls

```
theia@theiaopenshift-huyhieu14220:/home/project/guestbook$ ls
LICENSE  README.md  v1  v2
```

Bước 2: Xây dựng app

- Vào thư mục v1 và sửa lại file docker như sau:

```
FROM golang:1.18 AS builder
```

```
WORKDIR /app
```

```
COPY main.go .
```

```
RUN go mod init guestbook
```

```
RUN go mod tidy
```

```
RUN go build -o main main.go
```

```
FROM ubuntu:18.04
```

```
COPY --from=builder /app/main /app/guestbook
```

```
COPY public/index.html /app/public/index.html
```

```
COPY public/script.js /app/public/script.js
```

```
COPY public/style.css /app/public/style.css
```

```
COPY public/jquery.min.js /app/public/jquery.min.js
```

```
WORKDIR /app
```

```
CMD ["/guestbook"]
```

```
EXPOSE 3000
```

- Export namespace như một môi trường để sử dụng

```
export MY_NAMESPACE=sn-labs-$USERNAME
```

- Build app bằng docker

```
docker build . -t us.icr.io/$MY_NAMESPACE/guestbook:v1
```

- Push image vào registry

```
docker push us.icr.io/$MY_NAMESPACE/guestbook:v1
```

- Sửa lại namespace trong file yaml

```
apiVersion: apps/v1
```

```
kind: Deployment
```

```
metadata:
```

```
  name: guestbook
```

```
  labels:
```

```
    app: guestbook
```

```
spec:
```

```
  replicas: 1
```

```
selector:
  matchLabels:
    app: guestbook
strategy:
  rollingUpdate:
    maxSurge: 25%
    maxUnavailable: 25%
  type: RollingUpdate
template:
  metadata:
    labels:
      app: guestbook
  spec:
    containers:
      - image: us.icr.io/sn-labs-huyhieu14220/guestbook:v1
        imagePullPolicy: Always
        name: guestbook
        ports:
          - containerPort: 3000
            name: http
    resources:
      limits:
        cpu: 50m
      requests:
        cpu: 20m
    replicas: 1
```

- **Mở một đường dẫn (port-forward)** từ cổng `3000` trên máy cá nhân đến cổng `3000` trên **một Pod thuộc Deployment** `guestbook`

```
kubectl apply -f deployment.yml
kubectl port-forward deployment.apps/guestbook 3000:3000
```

- Trang web sẽ chạy như sau:

Guestbook - v1

SUBMIT

https://huyhieu14220-3000.theiaopenshiftnext-0-labs-prod-theiaopenshift-4-tor01.proxy.cognitiveclass.ai/

[/env](#) [/info](#)

Bước 3: Autoscale ứng dụng

- Autoscale sử dụng kubectl

```
kubectl autoscale deployment guestbook --cpu-percent=5 --min=1 --max=10
```

```
kubectl run -i --tty load-generator --rm --image=busybox:1.36.0 --restart=Never -- /bin/sh -c "while sleep 0.01; do wget -q -O- <your app URL>; done"
```

- Chạy lệnh tạo tải (thay `<your app URL>` bằng URL app ở bước trước):

```
kubectl run -i --tty load-generator --rm --image=busybox:1.36.0 --restart=Never -- /bin/sh -c "while sleep 0.01; do wget -q -O- <your app URL>; done"
```

- Kiểm tra HPA

```
kubectl get hpa guestbook
```

Bước 4: Thực hiện rolling updates và rollback

- Build & push image mới

```
docker build . -t us.icr.io/$MY_NAMESPACE/guestbook:v1 && \
docker push us.icr.io/$MY_NAMESPACE/guestbook:v1
```

- Chỉnh CPU trong `deployment.yml`
 - Thay giá trị:

```
resources:
  requests:
    cpu: 5m
  limits:
    cpu: 2m
```

- Áp dụng thay đổi:

```
kubectl apply -f deployment.yml
```

- Xem lịch sử rollout

```
kubectl rollout history deployment/guestbook
kubectl rollout history deployments guestbook --revision=2
```

- Kiểm tra ReplicaSets hiện tại

```
kubectl get rs
```

- Hoàn tác (rollback) về revision 1

```
kubectl rollout undo deployment/guestbook --to-revision=1  
kubectl get rs
```