

КИИИ Лаб 9

Class Assignment / Homework

- Create deployments and services for two Pods:
 - One for the app version 1.0 from the last homework
 - One for the app version 2.0 from the last homework
- Create ingress pointing to the two apps:
 - path based: localhost/ver1
 - path based: localhost/ver2
 - host based: ver1.<index>.com
 - host based: ver2.<index>.com
- Deploy manifests and ingress
- Access the four ingress rules in your local browser

STEP 1: Create manifests yaml

Deployment-1.yaml:

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-static-site-v1
spec:
  replicas: 5
  selector:
    matchLabels:
      app: my-static-site-v1
  template:
    metadata:
      labels:
        app: my-static-site-v1
    spec:
      containers:
        - name: my-static-site
          image: beratahmetaj/my-static-site:1.0
          ports:
            - containerPort: 80
```

Deployment-2.yaml:

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-static-site-v2
spec:
  replicas: 5
  selector:
    matchLabels:
      app: my-static-site-v2
```

```
template:
  metadata:
    labels:
      app: my-static-site-v2
  spec:
    containers:
      - name: my-static-site
        image: beratahmetaj/my-static-site:2.0
        ports:
          - containerPort: 80
```

Step 2: Create 2 service manifests

service-v1.yaml:

```
apiVersion: v1
kind: Service
metadata:
  name: my-static-site-v1
spec:
  selector:
    app: my-static-site-v1
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
```

service-v2.yaml:

```
apiVersion: v1
kind: Service
```

```
metadata:
  name: my-static-site-v2
spec:
  selector:
    app: my-static-site-v2
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
```

STEP 3: Create Ingress yaml

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: my-static-site-ingress
  annotations:
    nginx.ingress.kubernetes.io/rewrite-target: /
spec:
  rules:
    - host: "localhost"
      http:
        paths:
          - path: /ver1
            pathType: Prefix
            backend:
              service:
                name: my-static-site-v1
                port:
                  number: 80
          - path: /ver2
            pathType: Prefix
            backend:
              service:
```

```
        name: my-static-site-v2
        port:
          number: 80
- host: "ver1.<index>.com"
  http:
    paths:
      - path: /
        pathType: Prefix
        backend:
          service:
            name: my-static-site-v1
            port:
              number: 80
- host: "ver2.<index>.com"
  http:
    paths:
      - path: /
        pathType: Prefix
        backend:
          service:
            name: my-static-site-v2
            port:
              number: 80
```

STEP 4: run

```
kubectl apply -f deployment-v1.yaml
kubectl apply -f deployment-v2.yaml
kubectl apply -f service-v1.yaml
kubectl apply -f service-v2.yaml
kubectl apply -f ingress.yaml
```

Ingress rule

```
minikube addons enable ingress
```

```
C:\Users\berat>minikube ip
W0602 19:52:24.667985 8336 main.go:291] Unable to resolve the current Docker CLI context "default": context "default"
: context not found: open C:\Users\berat\.docker\contexts\meta\37a8eec1ce19687d132fe29051dca629d164e2c4958ba141d5f4133a3
3f0688f\meta.json: The system cannot find the path specified.

X Exiting due to GUEST_STATUS: Unable to get control-plane node minikube host status: state: unknown state "minikube": c
ocker container inspect minikube --format={{.State.Status}}: exit status 1
stdout:

stderr:
error during connect: this error may indicate that the docker daemon is not running: Get "http://%2F%2F.%2Fpipe%2Fdocke
_engine/v1.45/containers/minikube/json": open //./pipe/docker_engine: The system cannot find the file specified.

C:\Users\berat>
```

192.168.99.100 ver1.ver1.12345.com.com

192.168.99.100 ver2.ver1.12345.com.com