

КИИИ домашна 9

Create deployments and services for two Pods:

```
PS D:\fakultet\3god\6 semestar\kiii\domasni\domasna 9> k3d cluster create k8sCluster4 -p "80:80@loadbalancer" -s 1 -a 1
INFO[0000] portmapping '80:80' targets the loadbalancer: defaulting to [servers:*,proxy agents:*,proxy]
INFO[0000] Prep: Network
INFO[0000] Created network 'k3d-k8sCluster4'
```

One for the app version 1.0 from the last homework

```
PS D:\fakultet\3god\6 semestar\kiii\domasni\domasna 9> cat deployment-Ver1.yml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
spec:
  replicas: 5
  selector:
    matchLabels:
      app: static-nginx
  revisionHistoryLimit: 5
  progressDeadlineSeconds: 300
  minReadySeconds: 10
  strategy:
    type: RollingUpdate
    rollingUpdate:
      maxUnavailable: 1
      maxSurge: 2
  template:
    metadata:
      labels:
        app: static-nginx
        ver: v1
    spec:
      containers:
      - name: static-nginx
        image: andre102/static-nginx:1.0
        ports:
        - containerPort: 8080
```

```
PS D:\fakultet\3god\6 semestar\kiii\domasni\domasna 9> kubectl apply -f deployment-Ver1.yml
deployment.apps/nginx-deployment created
```

```
PS D:\fakultet\3god\6 semestar\kiii\domasni\domasna 9> cat .\service-Ver1.yml
apiVersion: v1
kind: Service
metadata:
  name: sevice-v1
spec:
  type: ClusterIP
  selector:
    app: static-nginx
    ver: v1
  ports:
    - protocol: TCP
      port: 8080
      targetPort: 80
```

```
PS D:\fakultet\3god\6 semestar\kiii\domasni\domasna 9> kubectl apply -f service-Ver1.yml
service/sevice-v1 created
```

```
PS D:\fakultet\3god\6 semestar\kiii\domasni\domasna 9> kubectl describe service sevice-v1
Name:          sevice-v1
Namespace:     default
Labels:        <none>
Annotations:   <none>
Selector:      app=static-nginx,ver=v1
Type:          ClusterIP
IP Family Policy: SingleStack
IP Families:   IPv4
IP:            10.43.194.6
IPs:           10.43.194.6
Port:          <unset> 8080/TCP
TargetPort:    8080/TCP
Endpoints:     10.42.0.6:8080,10.42.0.7:8080,10.42.0.8:8080 + 2 more...
Session Affinity: None
Events:        <none>
```

One for the app version 2.0 from the last homework

```
PS D:\fakultet\3god\6 semestar\kiii\domasni\domasna 9> cat deployment-Ver2.yml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment-ver2
spec:
  replicas: 5
  selector:
    matchLabels:
      app: static-nginx
  revisionHistoryLimit: 5
  progressDeadlineSeconds: 300
  minReadySeconds: 10
  strategy:
    type: RollingUpdate
    rollingUpdate:
      maxUnavailable: 1
      maxSurge: 2
  template:
    metadata:
      labels:
        app: static-nginx
        ver: v2
    spec:
      containers:
      - name: static-nginx
        image: andre102/static-nginx:2.0
        ports:
        - containerPort: 8080
```

```
PS D:\fakultet\3god\6 semestar\kiii\domasni\domasna 9> kubectl apply -f deployment-Ver2.yml
deployment.apps/nginx-deployment-ver2 created
```

```
PS D:\fakultet\3god\6 semester\kiii\domasni\domasna 9> cat .\service-Ver2.yml
apiVersion: v1
kind: Service
metadata:
  name: sevice-v2
spec:
  type: ClusterIP
  selector:
    app: static-nginx
    ver: v2
  ports:
    - protocol: TCP
      port: 8080
      targetPort: 80
```

```
PS D:\fakultet\3god\6 semester\kiii\domasni\domasna 9> kubectl apply -f service-Ver2.yml
service/sevice-v2 created
```

```
PS D:\fakultet\3god\6 semester\kiii\domasni\domasna 9> kubectl describe service sevice-v2
Name:          sevice-v2
Namespace:     default
Labels:        <none>
Annotations:   <none>
Selector:      app=static-nginx,ver=v2
Type:          ClusterIP
IP Family Policy: SingleStack
IP Families:   IPv4
IP:            10.43.229.235
IPs:           10.43.229.235
Port:          <unset> 8080/TCP
TargetPort:    8080/TCP
Endpoints:     10.42.0.10:8080,10.42.0.9:8080,10.42.1.10:8080 + 2 more...
Session Affinity: None
Events:        <none>
```

```
PS D:\fakultet\3god\6 semester\kiii\domasni\domasna 9> kubectl get service
NAME                TYPE        CLUSTER-IP      EXTERNAL-IP  PORT(S)    AGE
kubernetes          ClusterIP   10.43.0.1        <none>       443/TCP    33m
sevice-v1           ClusterIP   10.43.194.6      <none>       8080/TCP   8m38s
sevice-v2           ClusterIP   10.43.229.235    <none>       8080/TCP   41s
```

```
PS D:\fakultet\3god\6 semester\kiii\domasni\domasna 9> kubectl get endpointslices
NAME                ADDRESSTYPE  PORTS  ENDPOINTS                                AGE
kubernetes          IPv4         6443   172.19.0.3                               35m
sevice-v1-hzlb5     IPv4         8080   10.42.1.7,10.42.0.8,10.42.1.6 + 2 more... 11m
sevice-v2-p8g7p     IPv4         8080   10.42.1.10,10.42.1.9,10.42.1.8 + 2 more... 3m19s
```

Create ingress pointing to the two apps:

```
PS D:\fakultet\3god\6 semestar\kiii\domasni\domasna 9> cat ingerss-complex.yml
apiVersion: traefik.containo.us/v1alpha1
kind: Middleware
metadata:
  name: middleware-all-ver
spec:
  stripPrefix:
    forceSlash: false
    prefixes:
      - /ver1
      - /ver2
---
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: ingress-all-ver
  annotations:
    ingress.kubernetes.io/ssl-redirect: "false"
    traefik.ingress.kubernetes.io/router.middlewares: default-middleware-all-ver@kubernetescrd
spec:
  rules:
  - http:
      paths:
      - path: /ver1
        pathType: Prefix
        backend:
```

```
      service:
        name: sevice-v1
        port:
          number: 8080
      - path: /ver2
        pathType: Prefix
        backend:
          service:
            name: sevice-v2
            port:
              number: 8080
  - host: ver1.201149.com
    http:
      paths:
      - path: /
        pathType: Prefix
        backend:
          service:
            name: sevice-v1
            port:
              number: 8080
  - host: ver2.201149.com
    http:
      paths:
      - path: /
        pathType: Prefix
        backend:
          service:
            name: sevice-v2
            port:
              number: 8080
```

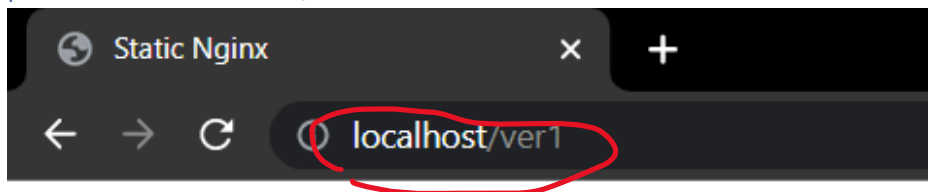
Deploy manifests and ingress

```
PS D:\fakultet\3god\6 semestar\kiii\domasni\domasna 9> kubectl apply -f ingeress-complex.yml
middleware.traefik.containo.us/middleware-all-ver created
ingress.networking.k8s.io/ingress-all-ver created
```

```
PS D:\fakultet\3god\6 semestar\kiii\domasni\domasna 9> kubectl get ing
NAME          CLASS    HOSTS                                ADDRESS          PORTS    AGE
ingress-all-ver  <none>  ver1.201149.com,ver2.201149.com  172.19.0.3,172.19.0.4  80      18s
PS D:\fakultet\3god\6 semestar\kiii\domasni\domasna 9>
```

Access the four ingress rules in your local browser

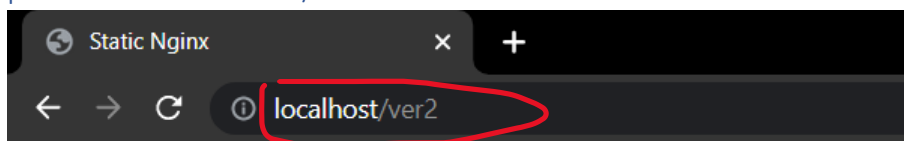
path based: localhost/ver1



KIII HOMEWORK 8

Andrej Bardakoski 201149

path based: localhost/ver2

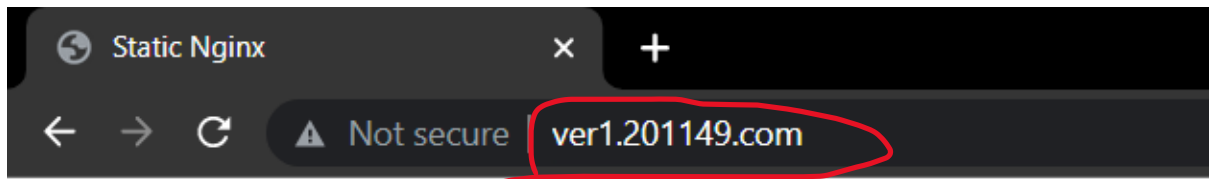


KIII HOMEWORK 8

Andrej Bardakoski 201149

Year: 2023

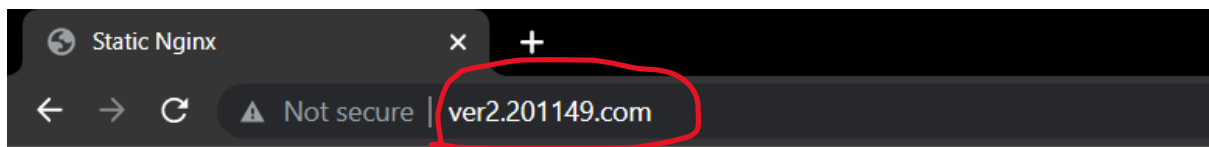
host based: ver1.<index>.com



KIII HOMEWORK 8

Andrej Bardakoski 201149

host based: ver2.<index>.com



KIII HOMEWORK 8

Andrej Bardakoski 201149

Year: 2023