Desplegando servicio Java en Kubernetes, 2/2

incsteps.com

1. Crear Cluster (en local)

```
curl -s \
https://raw.githubusercontent.com/k3d-io/k3d/main/install.sh | bash
k3d cluster create -p "8081:80@loadbalancer" my-cluster
```

el puerto 8081 **local** se "linka" con el 80 **interno** del cluster

2. Deployment

Casi siempre desplegamos una aplicación y un servicio para acceder a ella

deployment.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: "hello"
spec:
  selector:
    matchLabels:
      app: "hello"
  template:
    metadata:
      labels:
        app: "hello"
    spec:
      containers:
        - name: "hello"
          image: incsteps/helloworld #(1)
          ports:
            - name: http
              containerPort: 8080
---
apiVersion: v1
kind: Service
metadata:
  name: "hello"
spec:
  selector:
    app: "hello"
  type: NodePort
  ports:
    - protocol: "TCP"
      port: 8080
```

1. Utiliza tu namespace

3. Ingress (para acceder desde la "calle")

ingress.yml

```
apiVersion: traefik.containo.us/v1alpha1
kind: IngressRoute
metadata:
  name: hello
spec:
  entryPoints:
    - web
  routes:
    - match: PathPrefix(`/hello/`)
      kind: Rule
      services:
        - name: hello
          port: 8080
      middlewares:
        - name: hello
---
apiVersion: traefik.containo.us/v1alpha1
kind: Middleware
metadata:
  name: hello
spec:
  stripPrefix:
    prefixes:
      - /hello/
```

(k3d usa traefik como LoadBalancer, otros proveedores usan otros)

4. Probando nuestro servicio

Abre un navegador en http://localhost:8081/hello/

Authors :

https://www.linkedin.com/in/jagedn
Mentors juniors by telling old "war" stories

Ir al repositorio de pildoras