Running a service using a Kubernetes

Step-by-step process to create and manage a Kubernetes cluster using **Kind** and deploy an NGINX web server.

To check the version of the kubectl

- kubectl version –client

1. Create config.yaml file

- nano kind-config.yaml

-
```
kind: Cluster
apiVersion: kind.x-k8s.io/v1alpha4
nodes:
- role: control-plane
- role: worker
- role: worker
networking:
  podSubnet: "10.244.0.0/16"
  serviceSubnet: "10.96.0.0/12"
```

Create a cluster named mycluster

- kind create cluster --config kind-config.yaml --name mycluster

2. Create deployment.yaml file

- nano ngnix-deployment.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
  namespace: default
spec:
  replicas: 2
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:latest
        ports:
        - containerPort: 80
```

To apply : kubectl apply -f ngnix-deployment.yaml

3. Create nginx file

- nano nginx-service.yaml

-
  ```
  apiVersion: v1
  kind: Service
  metadata:
    name: nginx-service
    namespace: default
  spec:
    selector:
      app: nginx
    ports:
    - protocol: TCP
      port: 80
      targetPort: 80
    type: ClusterIP
  ```

To apply : kubectl apply -f ngnix-service.yaml

A temporary BusyBox pod is launched to connect to the NGINX service:
Bash

- kubectl run -i --tty --rm debug --image=busybox --restart=Never – sh

Inside the BusyBox pod:

- wget -O- nginx-service

Then exit

External Access:
To get IP address : docker inspect mycluster-worker | grep IPAddress

- curl http://172.18.0.4:30080/