Answers:

#1) create deployment file- deployment.yaml

apiVersion: apps/v1

kind: Deployment

metadata:

name: app-deployment

labels:

app: app

spec:

replicas: 2

selector:

matchLabels:

app: app

template:

metadata:

labels:

app: app

spec:

containers:

- name: app

image: sirfragalot/docker-demo:dcus

ports:

- containerPort: 80

Create service file: service.yaml

apiVersion: v1

kind: Service

metadata:

name: app-service

spec:

selector:

app: app

type: NodePort

ports:

- port: 8080

nodePort: 30080

Text

Description automatically generated

Graphical user interface, text

Description automatically generated

Graphical user interface, application

Description automatically generated

#2) Create namespace test and deploy k8s cluster in it

Text

Description automatically generated

deployment.yaml

apiVersion: apps/v1

kind: Deployment

metadata:

name: app-deployment

namespace: test

labels:

app: app

spec:

replicas: 2

selector:

matchLabels:

app: app

template:

metadata:

labels:

app: app

spec:

containers:

- name: app

image: sirfragalot/docker-demo:dcus

ports:

- containerPort: 80

Service.yaml

apiVersion: v1

kind: Service

metadata:

name: app-service

namespace: test

spec:

selector:

app: app

type: NodePort

ports:

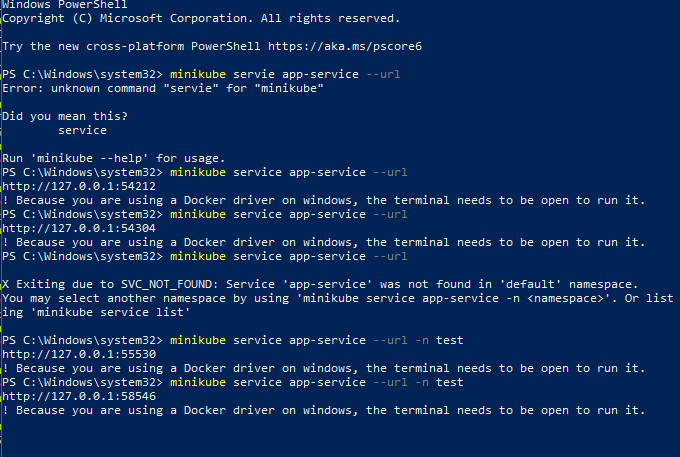
- port: 8080

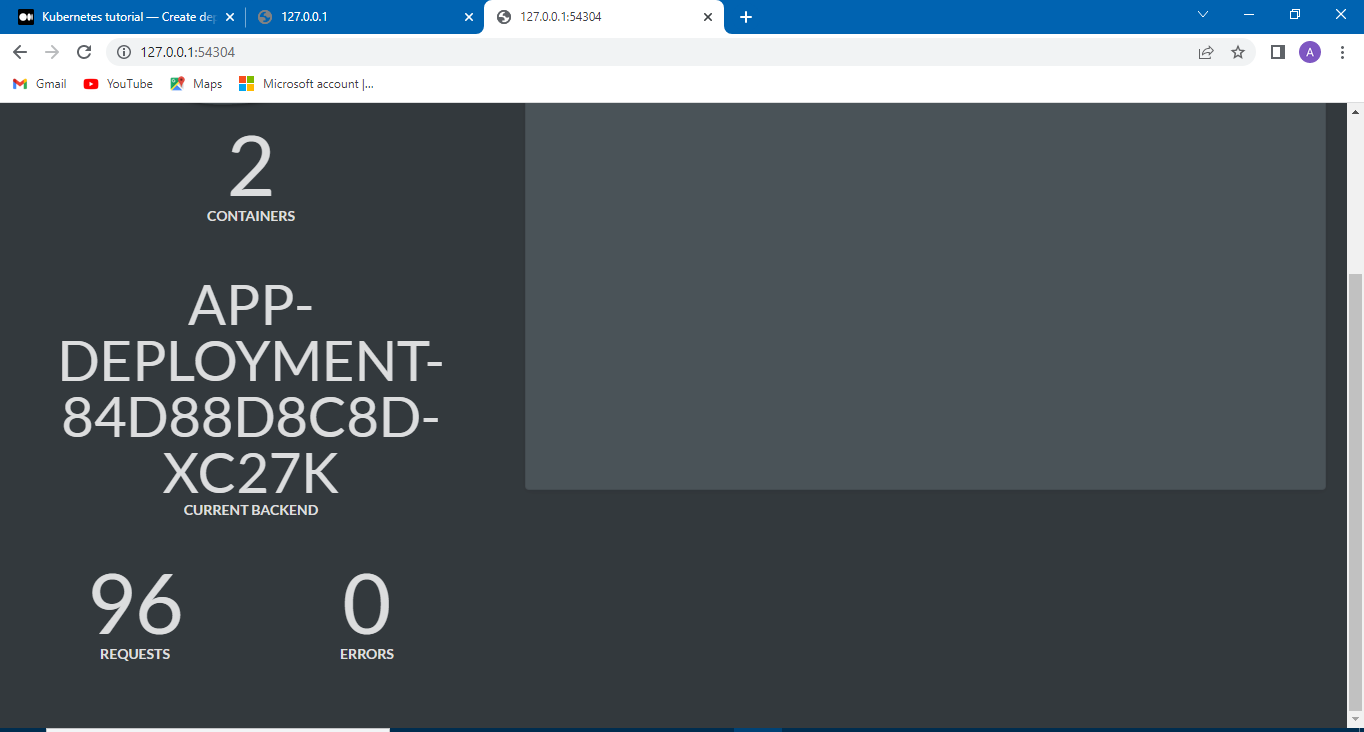
nodePort: 30080

#3) Expose deployment on local machine

Minikube is running on docker image and using docker driver so it is making tunnel to application port no to random port number as you can see in below screenshot.

We can access application on localhost:54304 url.





#4)Scale replicas upto 5.

Deployment.yaml

apiVersion: apps/v1

kind: Deployment

metadata:

name: app-deployment

namespace: test

labels:

app: app

spec:

replicas: 5

selector:

matchLabels:

app: app

template:

metadata:

labels:

app: app

spec:

containers:

- name: app

image: sirfragalot/docker-demo:dcus

ports:

- containerPort: 80

Service.yaml

apiVersion: v1

kind: Service

metadata:

name: app-service

namespace: test

spec:

selector:

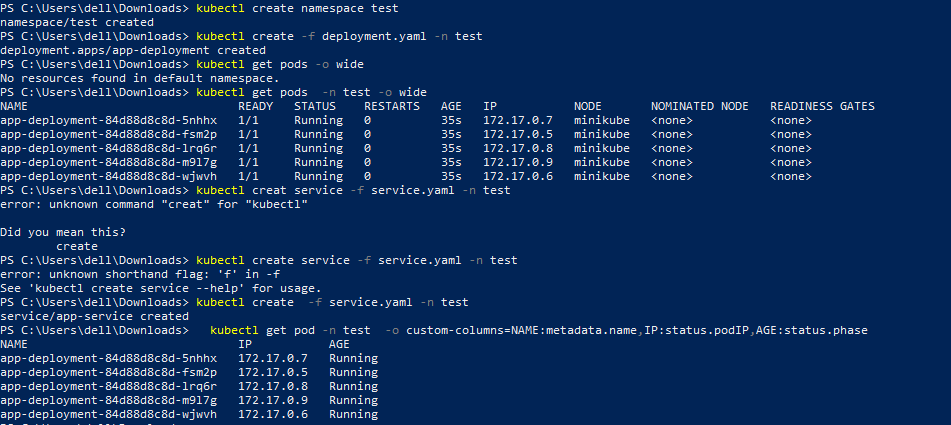
app: app

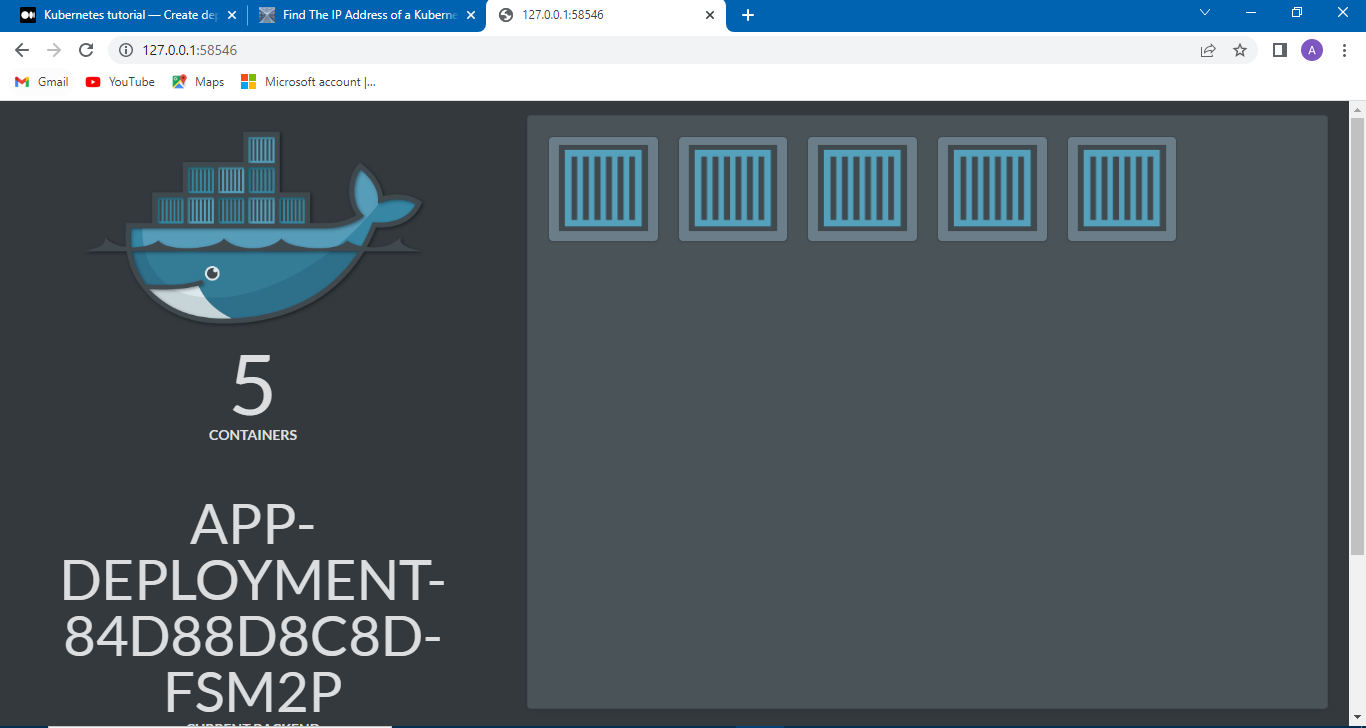
type: NodePort

ports:

- port: 8080

nodePort: 30080





#5) show only pods Ips and health under the headers IP and Health.

kubectl get pod -n test -o custom-columns=NAME:metadata.name,IP:status.podIP,AGE:status.phase

