**CASE STUDY - INTRODUCTION TO KUBERNETES**

**You have just joined a startup Ventura Software as a Devops Lead Engineer. The company relies on a Monolithic Architecture for its product. Recently, the senior management was hired. The new CTO insists on having a Microservice Architecture. The Development Team, is working on breaking the Monolith. Meanwhile, you have been asked to host a Test Application on Kubernetes, to understand how it works.**

**Following things have to be implemented:**

**1. Deploy an Apache2 deployment of 2 replicas**

**2. Sample code has been checked-in at the following Git-Hub repo:** [**https://github.com/hshar/website.git**](https://github.com/hshar/website.git)**.**

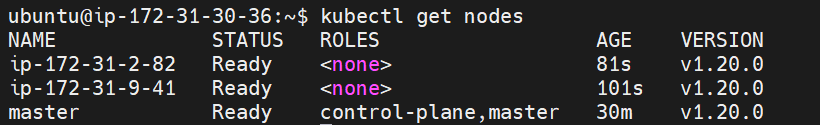
**You have to containerize this code, and push it to Docker Hub. Once done, deploy it on Kubernetes with 2 replicas**

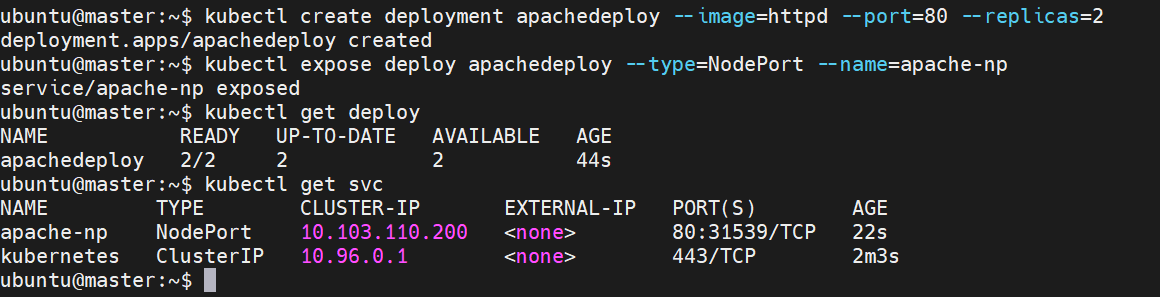
**3. Deploy Ingress with the following rules:**

**i) \*/apache\* should point to the apache pods**

**ii) \*/custom\* should point to the GitHub application**

1. **Deploy an Apache2 deployment of 2 replicas**

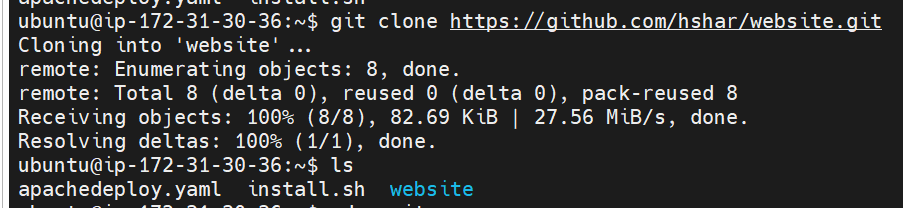
* **kubectl get nodes**
* **kubectl create deployment apachedeploy --image=httpd --port=80 --replicas=2**
* **kubectl expose deploy apachedeploy --type=NodePort --name=apache-np**
* **kubectl get deploy**
* **kubectl get svc**

****

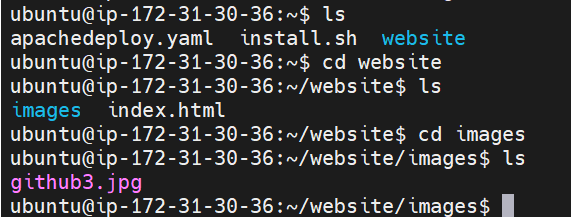
1. **Sample code has been checked-in at the following Git-Hub repo:** [**https://github.com/hshar/website.git**](https://github.com/hshar/website.git)**.**

**You have to containerize this code, and push it to Docker Hub. Once done, deploy it on Kubernetes with 2 replicas**

* **git clone** [**https://github.com/hshar/website.git**](https://github.com/hshar/website.git)
* **ls**

****

* **cd website**
* **ls**
* **cd images**
* **ls**

****

* **sudo nano dockerfile**

FROM ubuntu

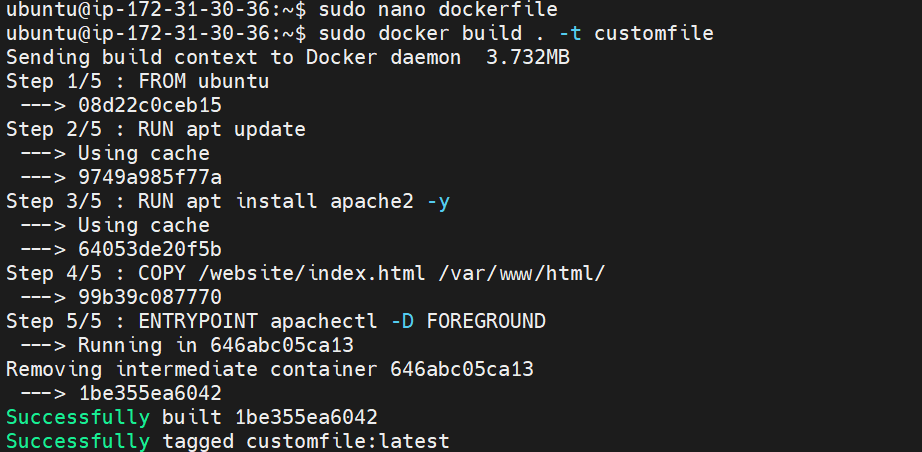
RUN apt update

RUN apt install apache2 -y

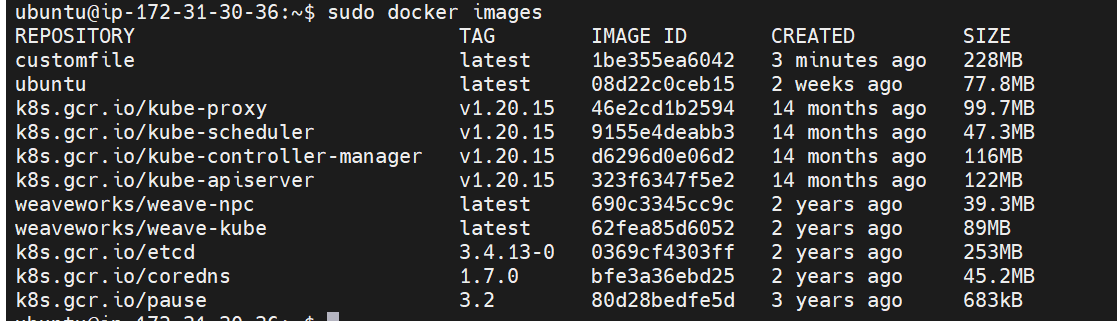
COPY website/index.html /var/www/html/

ENTRYPOINT apachectl -D FOREGROUND

* **sudo docker build . –t customfile**

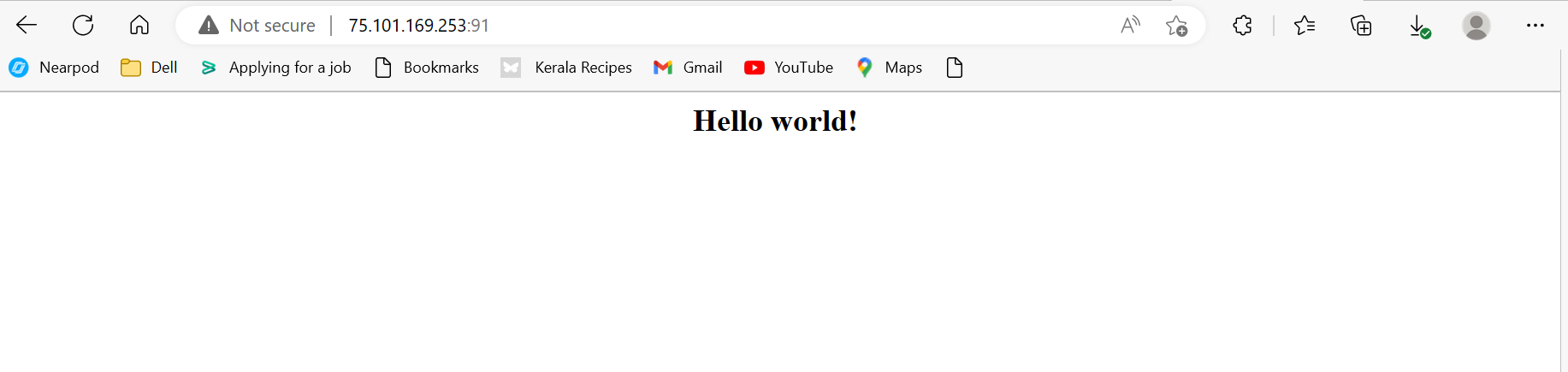
****

* **sudo docker images**

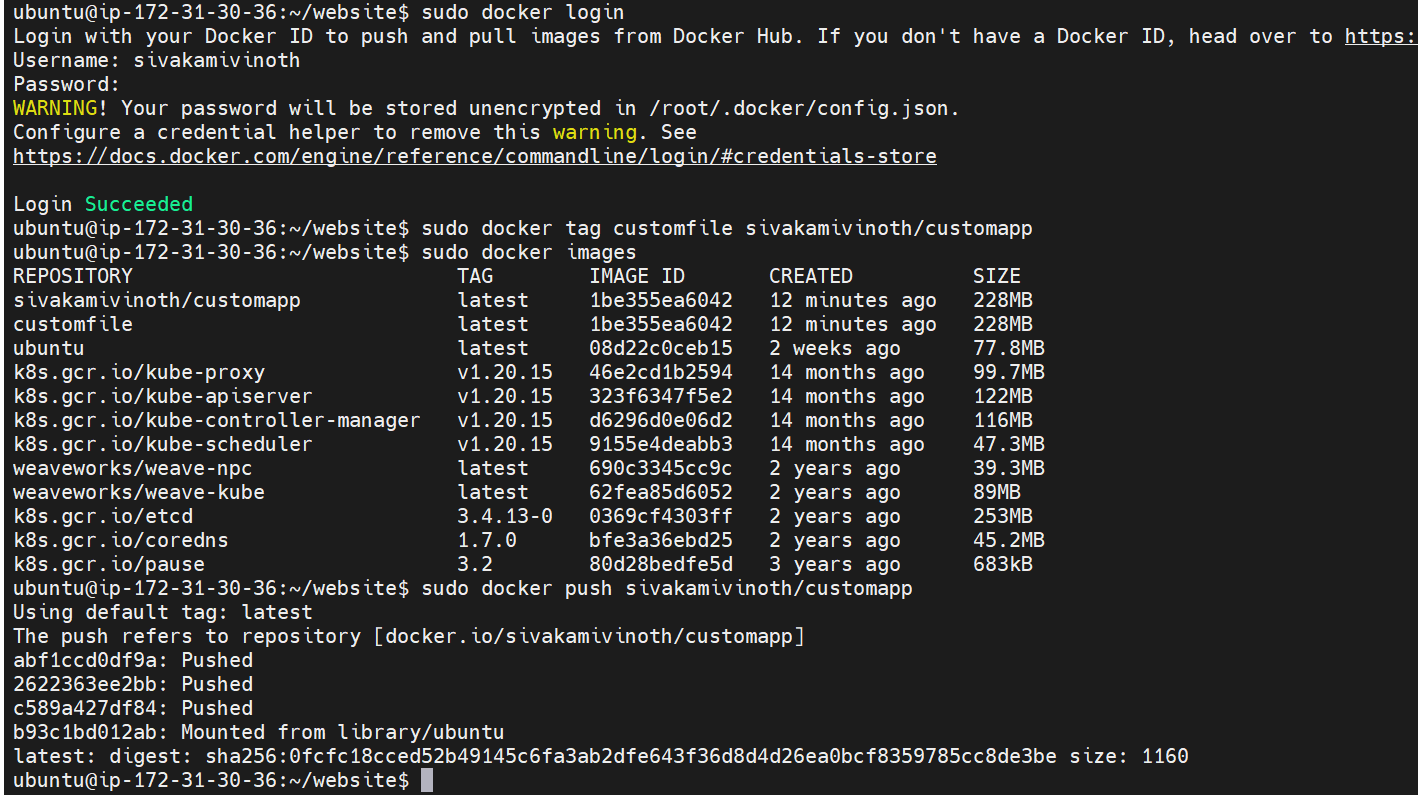
****

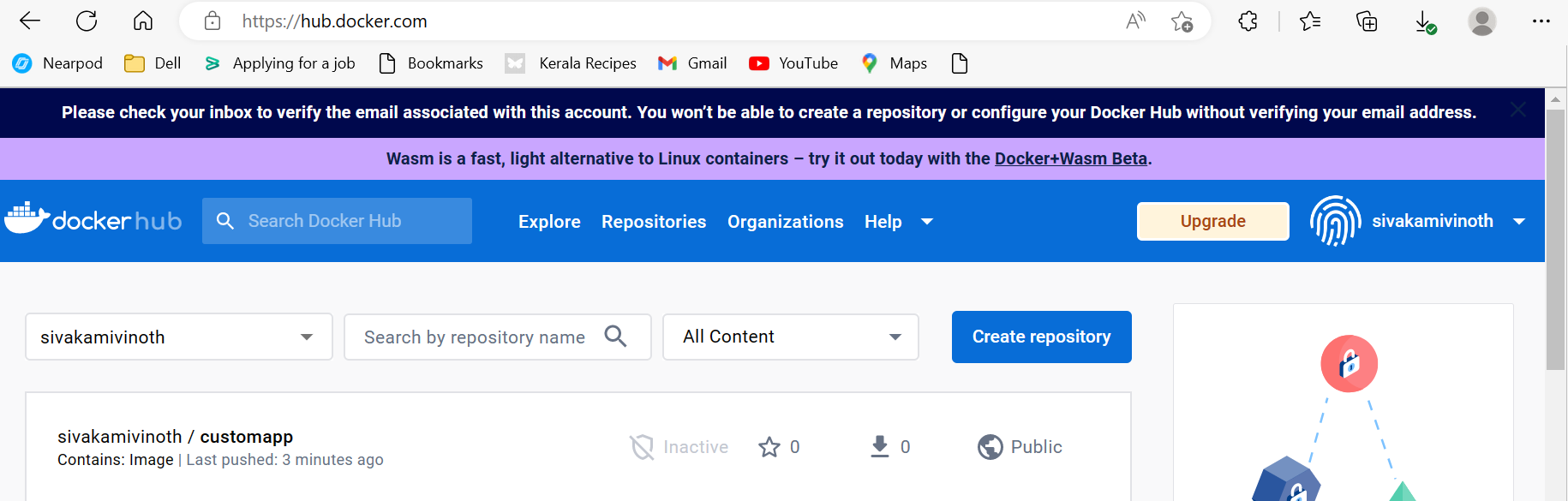
* **sudo docker run –itd –p 91:80 customfile**

****

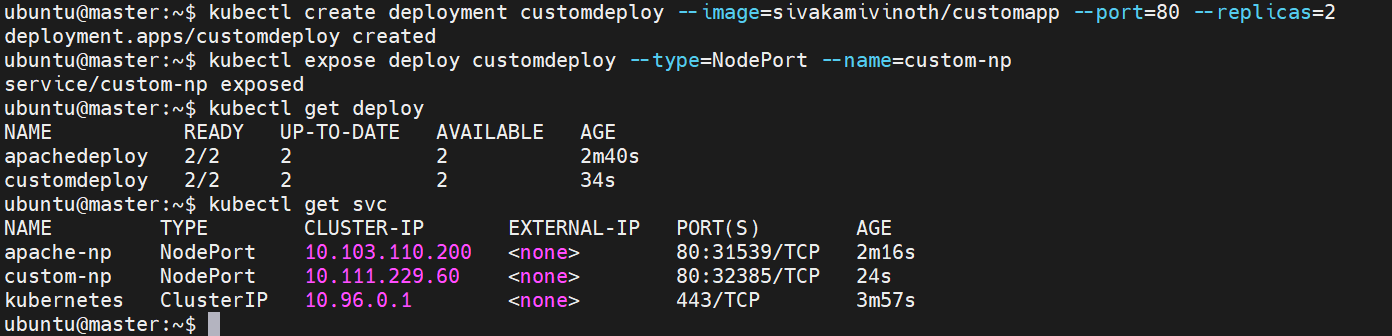
****

* **sudo docker login**
* **sudo docker tag customfile sivakamivinoth/customapp**
* **sudo docker images**
* **sudo docker push sivakamivinoth/customapp**

****

****

* **kubectl create deployment customdeploy --image=sivakamivinoth/customapp --port=80 --replicas=2**
* **kubectl expose deploy customdeploy --type=NodePort --name=custom-np**
* **kubectl get deploy**
* **kubectl get svc**

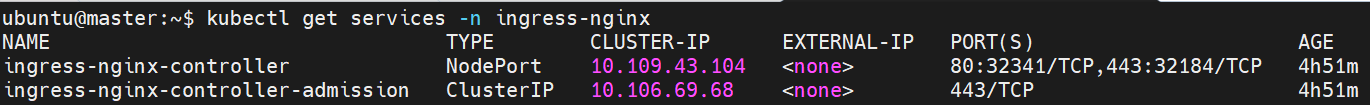
****

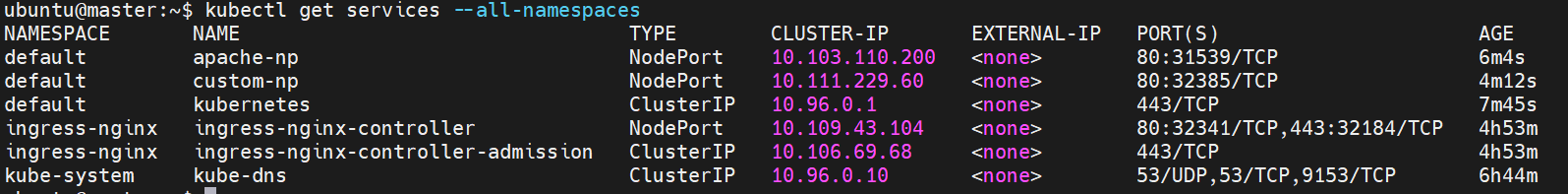
**3. Deploy Ingress with the following rules:**

**i) \*/apache\* should point to the apache pods**

**ii) \*/custom\* should point to the GitHub application**

* **kubectl apply -f** [**https://raw.githubusercontent.com/kubernetes/ingress-nginx/controller-v1.1.1/deploy/static/provider/baremetal/deploy.yaml**](https://raw.githubusercontent.com/kubernetes/ingress-nginx/controller-v1.1.1/deploy/static/provider/baremetal/deploy.yaml)
* **kubectl get services -n ingress-nginx**
* **kubectl get services --all-namespaces**

****

****

* **nano myingress.yaml**

apiVersion: networking.k8s.io/v1

kind: Ingress

metadata:

name: myingress

annotations:

kubernetes.io/ingress.class: nginx

ingress.kubernetes.io/rewrite-target: /

spec:

rules:

- host: "ec2-34-200-246-190.compute-1.amazonaws.com" # app1.com

http:

paths:

- path: "/"

pathType: Prefix

backend:

service:

name: custom-np

port:

number: 80

- host: "ec2-3-236-129-79.compute-1.amazonaws.com" # app2.com

http:

paths:

- path: "/"

pathType: Prefix

backend:

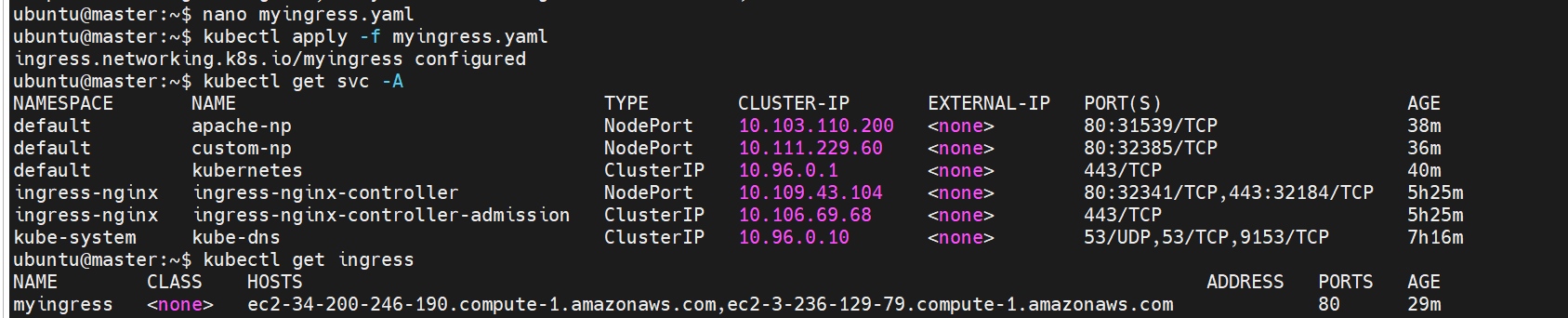
service:

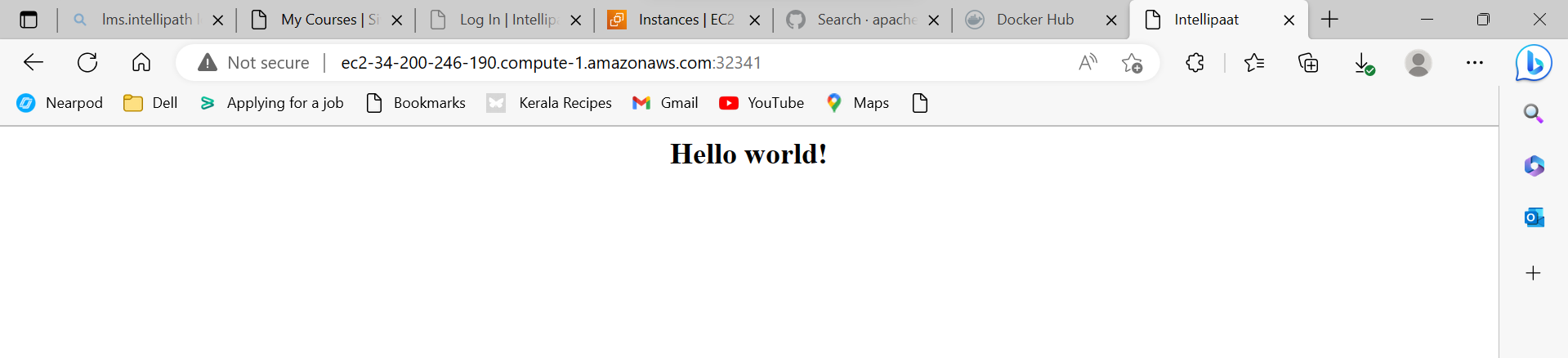
name: apache-np

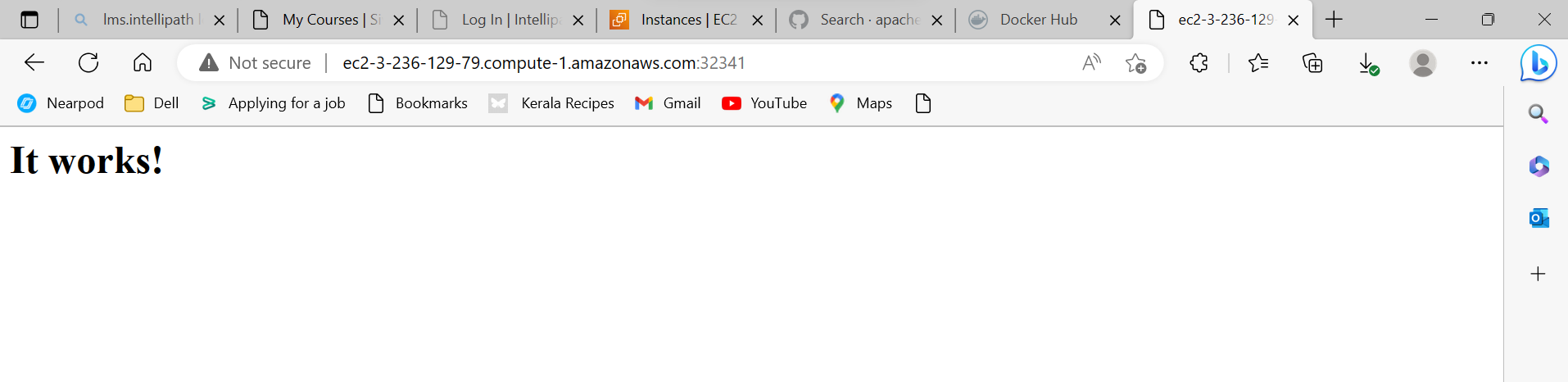
port:

number: 80

* **kubectl create -f myingress.yaml**
* **kubectl get svc –A**
* **kubectl get ingress**

****

****

****