How To Expose Multiple Applications on Amazon EKS Using a Single Application Load Balancer

## Deploy a hello, world app

1. Create a Deployment using the following command:

$ kubectl create deployment web --image=gcr.io/google-samples/hello-app:1.0

The output should be:

deployment.apps/web created

1. Expose the Deployment:

$ kubectl expose deployment web --type=NodePort --port=8080

The output should be:

service/web exposed

Verify the Service is created and is available on a node port:

$ kubectl get service web

The output is similar to:

NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE

web NodePort 10.104.133.249 <none> 8080:31637/TCP 12m

$ vi example-ingress.yaml

**apiVersion**: networking.k8s.io/v1

**kind**: Ingress

**metadata**:

**name**: example-ingress

**annotations**:

**nginx.ingress.kubernetes.io/rewrite-target**: /$1

**spec**:

**rules**:

- **host**: hello-world.info

**http**:

**paths**:

- **path**: /

**pathType**: Prefix

**backend**:

**service**:

**name**: web

**port**:

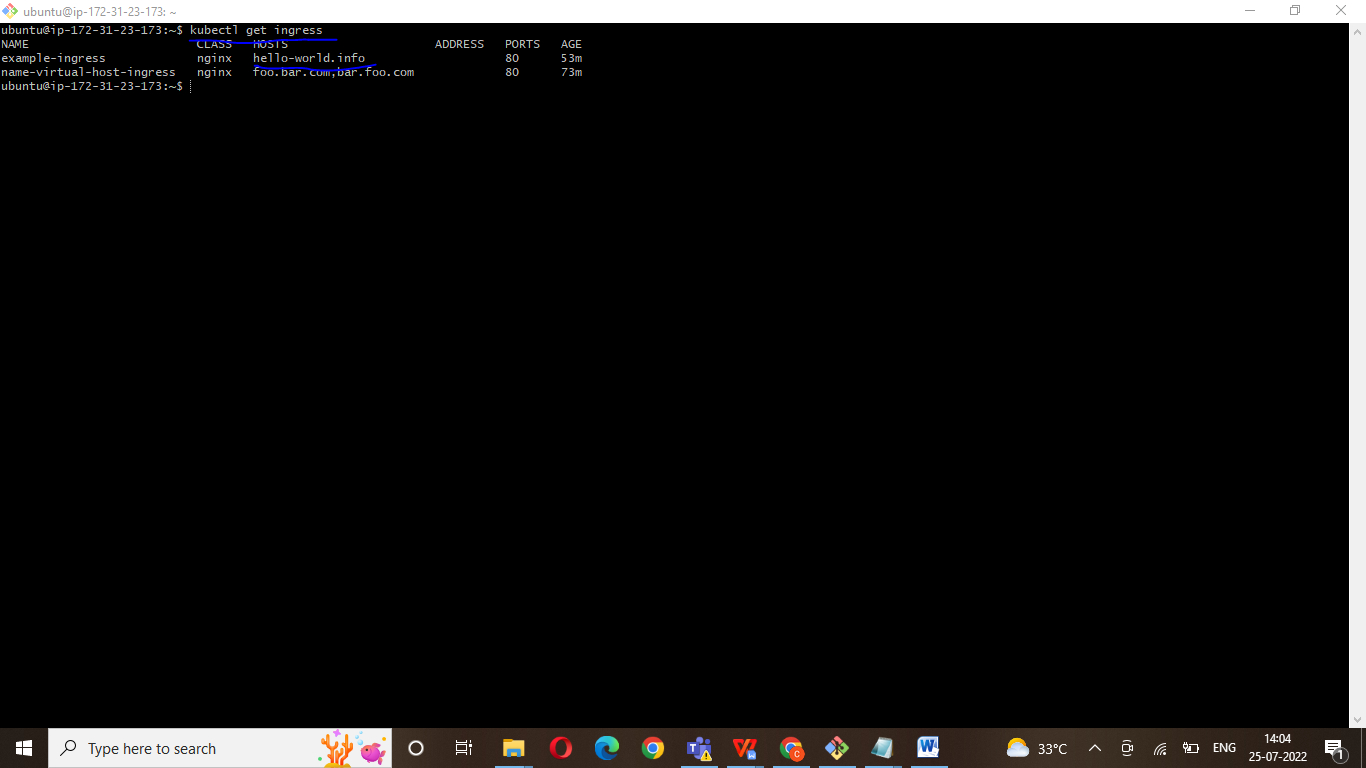
**number**: 8080

kubectl apply -f <https://k8s.io/examples/service/networking/example-ingress.yaml>

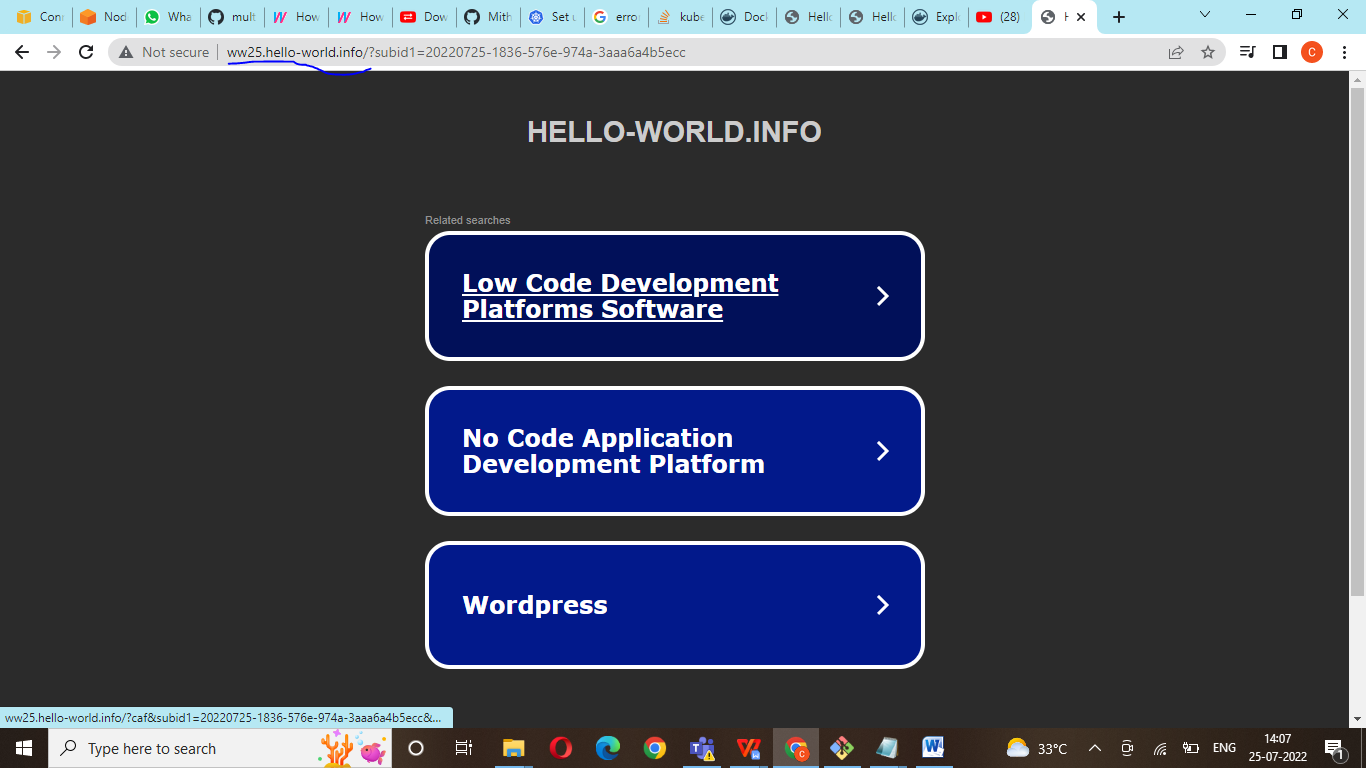
The output should be:

ingress.networking.k8s.io/example-ingress created

$ kubectl get ingress



url : hello-world.info



## Create a second Deployment

1. Create another Deployment using the following command:

$ kubectl create deployment web2 --image=gcr.io/google-samples/hello-app:2.0

1. The output should be:
2. deployment.apps/web2 created
3. Expose the second Deployment:

$ kubectl expose deployment web2 --port=8080 --type=NodePort

The output should be:

service/web2 exposed

apiVersion: networking.k8s.io/v1

kind: Ingress

metadata:

name: example-ingress

annotations:

nginx.ingress.kubernetes.io/rewrite-target: /$1

spec:

rules:

- host: hello-world.info

http:

paths:

- path: /

pathType: Prefix

backend:

service:

name: web

port:

number: 8080

- path: /v2

pathType: Prefix

backend:

service:

name: web2

port:

number: 8080

$kubectl apply -f example-ingress.yaml

You should see:

ingress.networking/example-ingress configured

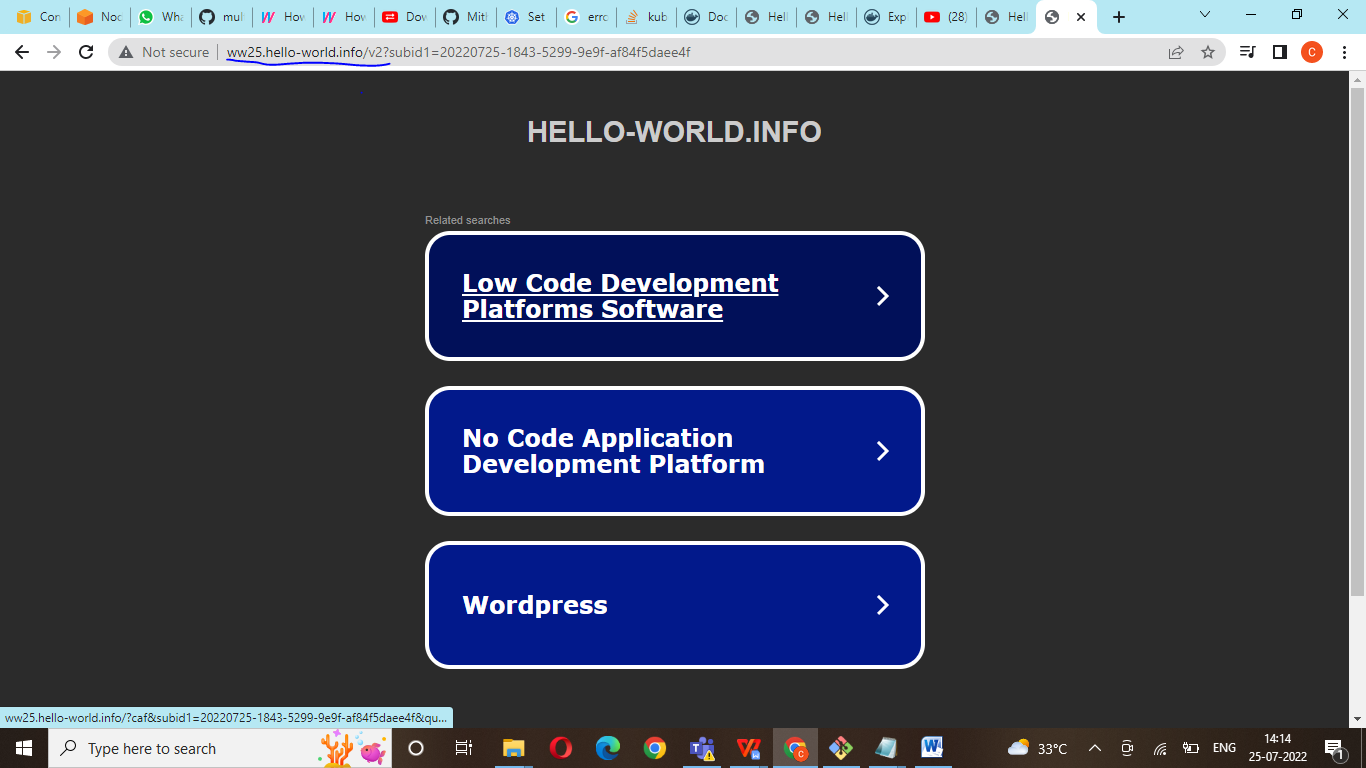
## Test your Ingress

1. Access the 1st version of the Hello World app.

$ curl hello-world.info

Access the 2nd version of the Hello World app.

$ curl hello-world.info/v2



<https://kubernetes.io/docs/tasks/access-application-cluster/ingress-minikube/>

https://cloud.google.com/kubernetes-engine/docs/tutorials/http-balancer