CKA Lab Part 9 - Networking

Lab 1 - Create a ClusterIP

- Create a deployment consisting of three nginx containers.

```
kubectl run nginx --image=nginx --replicas=3 --port=80
```

- Create a service of type “Cluster IP” which is exposed on port 8080 that facilitates connections to the aforementioned deployment, which listens on port 80.

```
kubectl expose deployment nginx --port=8080 --target-port=80
```

- Test the connectivity by spinning up a pod

```
kubectl run busybox --image=busybox -- "sleep" "100000"
kubectl exec -it busybox-58654df677-rp47x sh
wget http://10.100.200.227:8080http://10.100.200.227:8080
Connecting to 10.100.200.227:8080 (10.100.200.227:8080)
index.html           100%
```

**Lab 2 - Create a LoadBalancer **

- Note, for this to work your environment must be able to provision load balancers (GKE, AKE, etc)
- Create a deployment consisting of three nginx containers.

```
kubectl run nginx --image=nginx --replicas=3 --port=80
```

- Create a service of type “Load Balancer” which is exposed on port 8080 that facilitates connections to the aforementioned deployment, which listens on port 80.

```
kubectl expose deployment nginx --port=8080 --target-port=80 --type=LoadBalancer
```

- Test the connectivity by accessing the website

Lab 3 - Create a NodePort

- Create a deployment consisting of three nginx containers.

```
kubectl run nginx-nodeport --image=nginx --replicas=3 --port=80
```

- Create a service of type “NodePort” which is exposed on port 30010 that facilitates connections to the aforementioned deployment, which listens on port 80.

```
apiVersion: v1
kind: Service
metadata:
 creationTimestamp: 2019-04-28T15:31:50Z
 labels:
   run: nginx-nodeport
 name: nginx-nodeport
 namespace: default
 spec:
 clusterIP: 10.100.200.236
 externalTrafficPolicy: Cluster
 ports:
 - nodePort: 30010
   port: 30010
   protocol: TCP
   targetPort: 80
 selector:
   run: nginx-nodeport
```

```
 sessionAffinity: None
 type: NodePort
status:
 loadBalancer: {}
```

- Test the connectivity by spinning up a pod

Lab 4 - Create a Ingress Resource

Create a ingress resource “website-ingress” that directs traffic to the following conditions:

- Base domain website.com
  - Default backend service is “default-service” on port 80
  - /backend directs traffic to “backend-service” on port 443
  - /test directs traffic to “test-service” on port 8000

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
 name: website-ingress
 annotations:
   nginx.ingress.kubernetes.io/rewrite-target: /
spec:
 rules:
 - host: website.com
   http:
     paths:
     - path: /
       backend:
         serviceName: default-service
         servicePort: 80
     - path: /backend
       backend:
         serviceName: backend-service
         servicePort: 443
     - path: /test
       backend:
         serviceName: test-service
         servicePort: 8000
```