Experiment 8

AIM: Kubernetes - Minikube installation and fundamentals

Start the cluster

```
$ minikube start --wait=false
* minikube v1.8.1 on Ubuntu 18.04
* Using the none driver based on user configuration
* Running on localhost (CPUs=2, Memory=2460MB, Disk=145651MB) ...
* OS release is Ubuntu 18.04.4 LTS
* Preparing Kubernetes v1.17.3 on Docker 19.03.6 ...
  - kubelet.resolv-conf=/run/systemd/resolve/resolv.conf
* Launching Kubernetes ...
* Enabling addons: default-storageclass, storage-provisioner
* Configuring local host environment ...
* Done! kubectl is now configured to use "minikube"
```

Get cluster details

```
$ kubectl cluster-info
Kubernetes master is running at https://172.17.0.75:8443
KubeDNS is running at https://172.17.0.75:8443/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy
```

view the nodes in the cluster

```
$ kubectl get nodes
NAME       STATUS   ROLES    AGE   VERSION
minikube   Ready    master   62s   v1.17.3
$
```

Deploy containers

```
$ kubectl create deployment first-deployment --image=katacoda/docker-http-server
deployment.apps/first-deployment created
$ kubectl get pods
NAME                               READY   STATUS              RESTARTS   AGE
first-deployment-666c48b44-blg5l   0/1     ContainerCreating   0          1s
$
```

Expose container via NodePort

```
$ kubectl expose deployment first-deployment --port=80 --type=NodePort
service/first-deployment exposed
```

find the allocated port and executes a HTTP request

```
$ export PORT=$(kubectl get svc first-deployment -o go-template='{{range.spec.ports}}{{if .nodePort}}{{.nodePort
}}{{"\n"}}{{end}}{{end}}')
$ echo "Accessing host01:$PORT"
Accessing host01:31113
$ curl host01:$PORT
<h1>This request was processed by host: first-deployment-666c48b44-blg5l</h1>
```

Enable the dashboard using Minikube

```
$ minikube addons enable dashboard
* The 'dashboard' addon is enabled
```

Make the Kubernetes Dashboard available

```
$ kubectl apply -f /opt/kubernetes-dashboard.yaml
namespace/kubernetes-dashboard configured
service/kubernetes-dashboard-katacoda created
```

To see the progress of the Dashboard starting, watch the Pods within the *kube-system* namespace

```
$ kubectl get pods -n kubernetes-dashboard -w
NAME                                         READY   STATUS    RESTARTS   AGE
dashboard-metrics-scraper-7b64584c5c-hphq4   1/1     Running   0          43s
kubernetes-dashboard-79d9cd965-kmpzq         1/1     Running   0          43s
```