EXPERIMENT 8

AIM: Working with Kubernetes (Single Node Cluster)

Steps to Complete:

Step 1 - Start Minikube

Minikube has been installed and configured in the environment. Check that it is properly installed, by running the minikube version command:

`minikube version`

```
anuvagarg@Anuvas-MacBook-Air k8s % minikube version
minikube version: v1.27.0
commit: 4243041b7a72319b9be7842a7d34b6767bbdac2b
```

Start the cluster, by running the minikube start command:

`minikube start --wait=false`

Great! You now have a running Kubernetes cluster in your online terminal. Minikube started a virtual machine for you, and a Kubernetes cluster is now running in that VM.

```
anuvagarg@Anuvas-MacBook-Air devops lab % minikube start --wait=false
😄  minikube v1.27.0 on Darwin 12.5
❗  Kubernetes 1.25.0 has a known issue with resolv.conf. minikube is using a workaround that should work for most use cases.
❗  For more information, see: https://github.com/kubernetes/kubernetes/issues/112135
✨  Automatically selected the docker driver. Other choices: virtualbox, ssh
- Ensure your docker daemon has access to enough CPU/memory resources.
- Docs https://docs.docker.com/docker-for-mac/#resources

⛔  Exiting due to RSRC_INSUFFICIENT_CORES: Requested cpu count 2 is greater than the available cpus of 1
```

Step 2 - Cluster Info

The cluster can be interacted with using the kubectl CLI. This is the main approach used for managing Kubernetes and the applications running on top of the cluster.

Details of the cluster and its health status can be discovered via

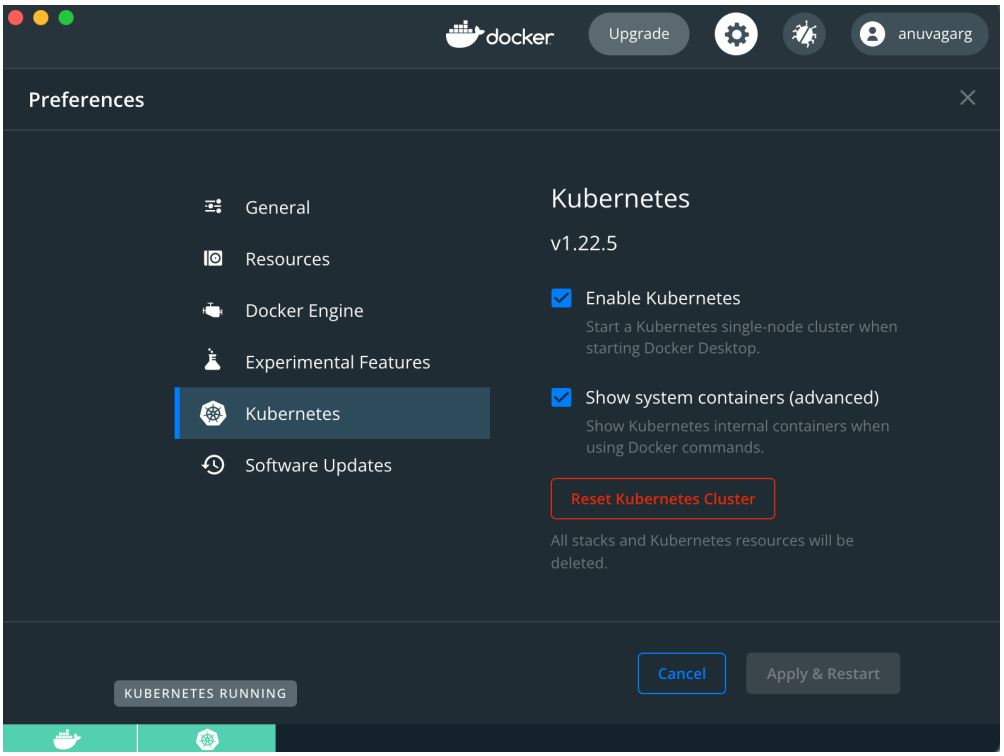
`kubectl cluster-info`

```
anuvagarg@Anuvas-MacBook-Air k8s % kubectl cluster-info
Kubernetes control plane is running at https://kubernetes.docker.internal:6443
CoreDNS is running at https://kubernetes.docker.internal:6443/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
```

`kubectl get nodes`

```
anuvagarg@Anuvas-MacBook-Air k8s % kubectl get nodes
NAME             STATUS   ROLES                  AGE     VERSION
docker-desktop   Ready    control-plane,master   9m34s   v1.22.5
```




```
anuvagarg@Anuvas-MacBook-Air k8s % ls -a
.               ..              Dockerfile      deployment.yml  pod.yml         replicaset.yml
  service.yml
anuvagarg@Anuvas-MacBook-Air k8s % kubectl apply -f pod.yml
pod/my-nginx-pod1 created
anuvagarg@Anuvas-MacBook-Air k8s % kubectl apply -f service.yml
service/my-nginx-service1 created
anuvagarg@Anuvas-MacBook-Air k8s % kubectl apply -f replicaset.yml
replicaset.apps/nginx-rs created
anuvagarg@Anuvas-MacBook-Air k8s % kubectl get all
NAME                 READY   STATUS              RESTARTS   AGE
pod/my-nginx-pod1    1/1     Running             0          59s
pod/nginx-rs-gbpkz   1/1     Running             0          17s
pod/nginx-rs-hnqr5   0/1     ContainerCreating   0          17s
pod/nginx-rs-lzm2h   0/1     ContainerCreating   0          17s
pod/nginx-rs-nvqmc   0/1     ContainerCreating   0          17s

NAME                         TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)        AGE
service/kubernetes           ClusterIP   10.96.0.1       <none>        443/TCP        37m
service/my-nginx-service1    NodePort    10.102.246.39   <none>        80:30005/TCP   49s

NAME                          DESIRED   CURRENT   READY   AGE
replicaset.apps/nginx-rs      5         5         2       17s
anuvagarg@Anuvas-MacBook-Air k8s % kubectl delete -f pod.yml
pod "my-nginx-pod1" deleted
anuvagarg@Anuvas-MacBook-Air k8s % kubectl delete -f service.yml
service "my-nginx-service1" deleted
anuvagarg@Anuvas-MacBook-Air k8s % kubectl delete -f replicaset.yml
replicaset.apps "nginx-rs" deleted
anuvagarg@Anuvas-MacBook-Air k8s % kubectl get all
NAME                 TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
service/kubernetes   ClusterIP   10.96.0.1    <none>        443/TCP   39m
```