Lab 11 : Resources and Limits

1. Create a pod (pod.yaml) with the requests (cpu: 200m, mem: 50Mi) and limits (cpu: 500m, mem: 60Mi) .

Shell

```
apiVersion: v1
kind: Pod
metadata:
  name: limits
  labels:
    name: app
spec:
  containers:
  - name: app
    image: paulbouwer/hello-kubernetes:1.10
    ports:
      - containerPort: 8080
    resources:
      requests:
        cpu: '200m'
        memory: '50Mi'
      limits:
        cpu: '500m'
        memory: '60Mi'
```

Shell

```
k apply -f limits.yaml
```

2. Expose the pod as service inside the cluster on port 3000.

Shell

```
apiVersion: v1
kind: Service
metadata:
  name: limits-service
spec:
  type: ClusterIP
  selector:
    name: app
  ports:
  - port: 80
    targetPort: 8080
```

Shell

```
k apply -f limits-service.yaml
```

3. Deploy a pod with image nicolaka/netshoot as helper.

🔥nicolaka/netshoot - Docker Image

Shell

```
kubectl run tmp-shell --rm -i --tty --image nicolaka/netshoot -- /bin/bash
```

4. Stress the pod by sending continuous requests to the service using the command below. Check the status of the pod after some time.

Shell

```
kubectl run tmp-shell --rm -i --tty --image nicolaka/netshoot -- /bin/bash
tmp-shell:~# fortio load -qps 10000 -t 0 http://limits-service.default.svc.cluster.local:80
```

Shell

```
k get po -w |  grep limits
```

Note: Observe the OOMKilled status.

5. Delete the resources.