Monitoring is an important part of managing any application infrastructure. In this lesson, we will discuss how to view the resource usage of pods and nodes using the `kubectl top` command.

Relevant Documentation

- https://kubernetes.io/docs/tasks/debug-application-cluster/resource-usage-monitoring/

Lesson Reference

Here are some sample pods that can be used to test `kubectl top`. They are designed to use approximately 300m and 100m CPU, respectively.

```
apiVersion: v1
kind: Pod
metadata:
  name: resource-consumer-big
spec:
  containers:
  - name: resource-consumer
    image: gcr.io/kubernetes-e2e-test-images/resource-consumer:1.4
    resources:
      requests:
        cpu: 500m
        memory: 128Mi
  - name: busybox-sidecar
    image: radial/busyboxplus:curl
    command: [/bin/sh, -c, 'until curl localhost:8080/ConsumeCPU -d "millicores=300&durationSec=3600"; do sle
```

```
apiVersion: v1
kind: Pod
metadata:
  name: resource-consumer-small
spec:
  containers:
  - name: resource-consumer
    image: gcr.io/kubernetes-e2e-test-images/resource-consumer:1.4
    resources:
      requests:
        cpu: 500m
        memory: 128Mi
  - name: busybox-sidecar
    image: radial/busyboxplus:curl
    command: [/bin/sh, -c, 'until curl localhost:8080/ConsumeCPU -d "millicores=100&durationSec=3600"; do sle
```

Here are the commands used in the lesson to view resource usage data in the cluster:

```
kubectl top pods
kubectl top pod resource-consumer-big
kubectl top pods -n kube-system
kubectl top nodes
```