Monitoring Kubernetes Applications

Relevant Documentation

- Resource metrics pipeline
- Tools for Monitoring Resources
- metrics-server GitHub

Exam Tips

- The Kubernetes metrics API provides metric data about container performance.
- You can view Pod metrics using `kubectl top pod`.

Lesson Reference

Log in to the **control plane node**.

Install `metrics-server`. This command uses a yaml file with a small tweak to make it easier to get metrics-server running in our environment.

```
kubectl apply -f https://raw.githubusercontent.com/ACloudGuru-Resources/content-cka-resources/master/metrics-server-components.yaml
```

Note: It may take a few minutes for metrics-server to gather initial metrics and become available.

Create a Pod that uses a detectable amount of CPU.

This setup uses a `resource-consumer` image that is designed to consume an approximate amount of CPU for testing purposes. Since it receives instructions via HTTP request, a sidecar container makes a request to the main container instructing it on how much CPU to consume.

```
vi resource-consumer-pod.yml
```

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: resource-consumer-pod
spec:
  containers:
  - name: resource-consumer
    image: gcr.io/kubernetes-e2e-test-images/resource-consumer:1.5
  - name: busybox-sidecar
    image: radial/busyboxplus:curl
    command: ['sh', '-c', 'until curl localhost:8080/ConsumeCPU -d "millicores=100&durationSec=3600"; do sleep 5; done && while true; do sleep 10; done']
```

```
kubectl apply -f resource-consumer-pod.yml
```

Use `kubectl top` to view metrics for Pods.

```
kubectl top pod

kubectl top pod -n default
```

View metrics by node.

```
kubectl top node
```