Containers & Kubernetes

Session #09

Monitoring and Operation

Kubernetes Dashboard

Auto-scalling: HPA

Lab

Monitoring and Operation

Best practices

Monitoring & Operation

- Containers must write logs to STDOUT or STDERR
  - Log files can be lost when container is removed
  - Common monitoring platform automatically stream stdout and stderr
- Pods must implement only one service/process
- Use services for internal communication between pods
- Use ingress to allow communication from outside the cluster

kubectl logs

Monitoring & Operation

```
kubectl logs <pod> [-n <namespace>]
```

- Shows pod stdout and stderr
- Flag **-f** blocks the console and show new lines

kubectl attach

Monitoring & Operation

```
kubectl attach [-it] <pod> [-c container] [-n <ns>]
```

- Attach to a process that is already running inside an existing container
- Adding `[-it]` flags allow to send commands to the pod

kubectl describe

Monitoring & Operation

```
kubectl describe pod <pod> [-n <namespace>]
```

- Shows details about pod
  - Metadata
  - Network
- Lists all events occurred during pod lifecycle
- First place to go when pod don’t have “Running” status

kubectl port-forward

Monitoring & Operation

```
kubectl port-forward pod <pod> [-n <ns>] hostport:podPort
kubectl port-forward svc <svc> [-n <ns>] hostport:podPort
```

- Maps a port on machine with pod port
- Allow to make direct requests
- When using service, maps directly to only on container (no load balancing)

kubectl top

Monitoring & Operation

```
kubectl top node <node>
kubectl top pod <pod> [-n <ns>]
```

- Display resource (CPU/memory) usage of the resources (nodes or pods)
- Due to the metrics pipeline delay, they may be unavailable for a few minutes since pod creation

Kubernetes Dashboard

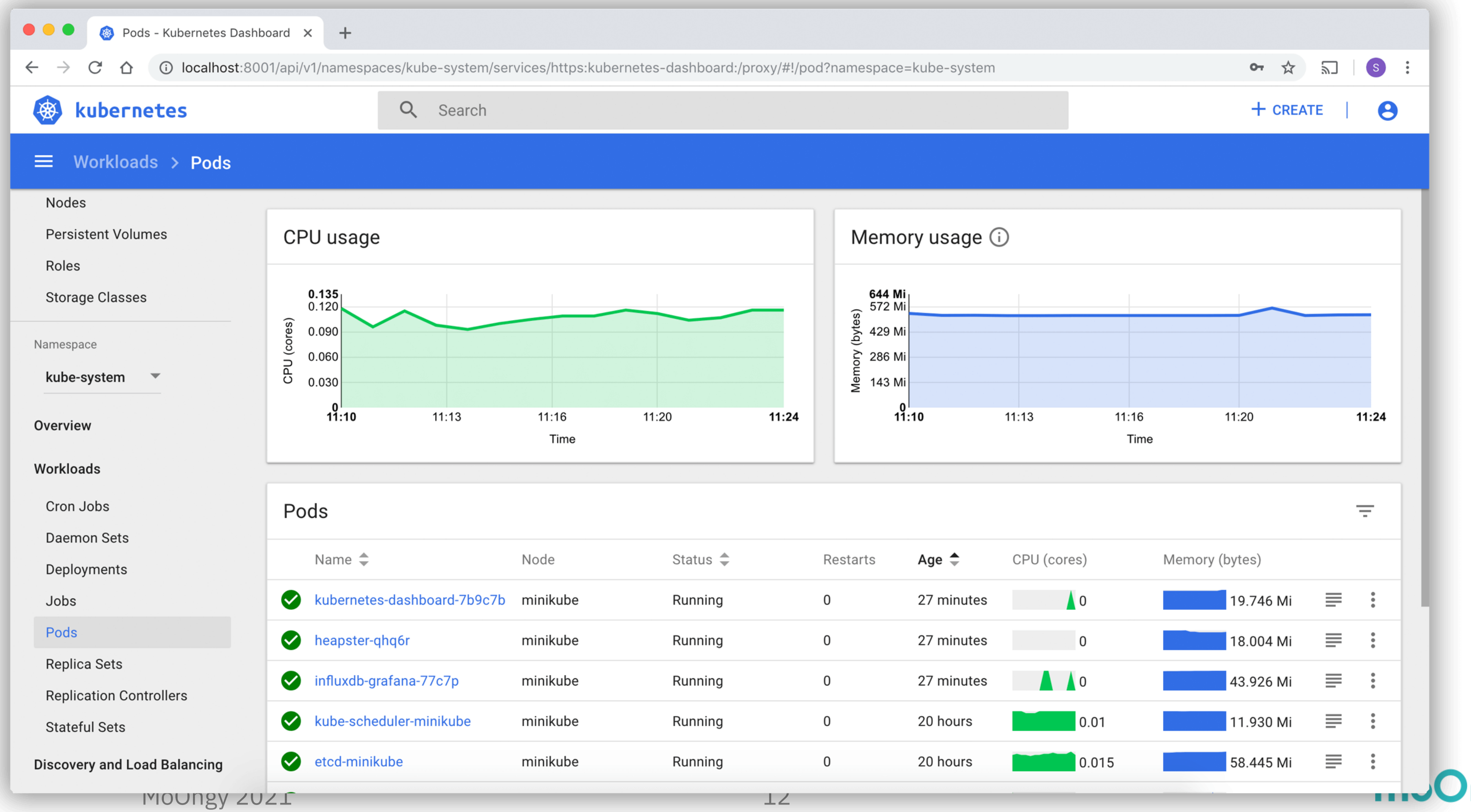
Kubernetes Dashboard

Observability

- Web-based Kubernetes user interface.
- You can use Dashboard to deploy containerized applications to a Kubernetes cluster, troubleshoot your containerized application, and manage the cluster resources.
- You can use Dashboard to get an overview of applications running on your cluster, as well as for creating or modifying individual Kubernetes resources. For example, you can scale a Deployment, initiate a rolling update, restart a pod or deploy new applications using a deploy wizard.
- Dashboard also provides information on the state of Kubernetes resources in your cluster and on any errors that may have occurred.

Kubernetes Dashboard

Observability



localhost:8001/api/v1/namespaces/kube-system/services/https:kubernetes-dashboard:/proxy/#!/pod?namespace=kube-system

Pods

	Name	Node	Status	Restarts	Age	CPU (cores)	Memory (bytes)
✓	kubernetes-dashboard-7b9c7b	minikube	Running	0	27 minutes	0	19.746 Mi
✓	heapster-qhq6r	minikube	Running	0	27 minutes	0	18.004 Mi
✓	influxdb-grafana-77c7p	minikube	Running	0	27 minutes	0	43.926 Mi
✓	kube-scheduler-minikube	minikube	Running	0	20 hours	0.01	11.930 Mi
✓	etcd-minikube	minikube	Running	0	20 hours	0.015	58.445 Mi

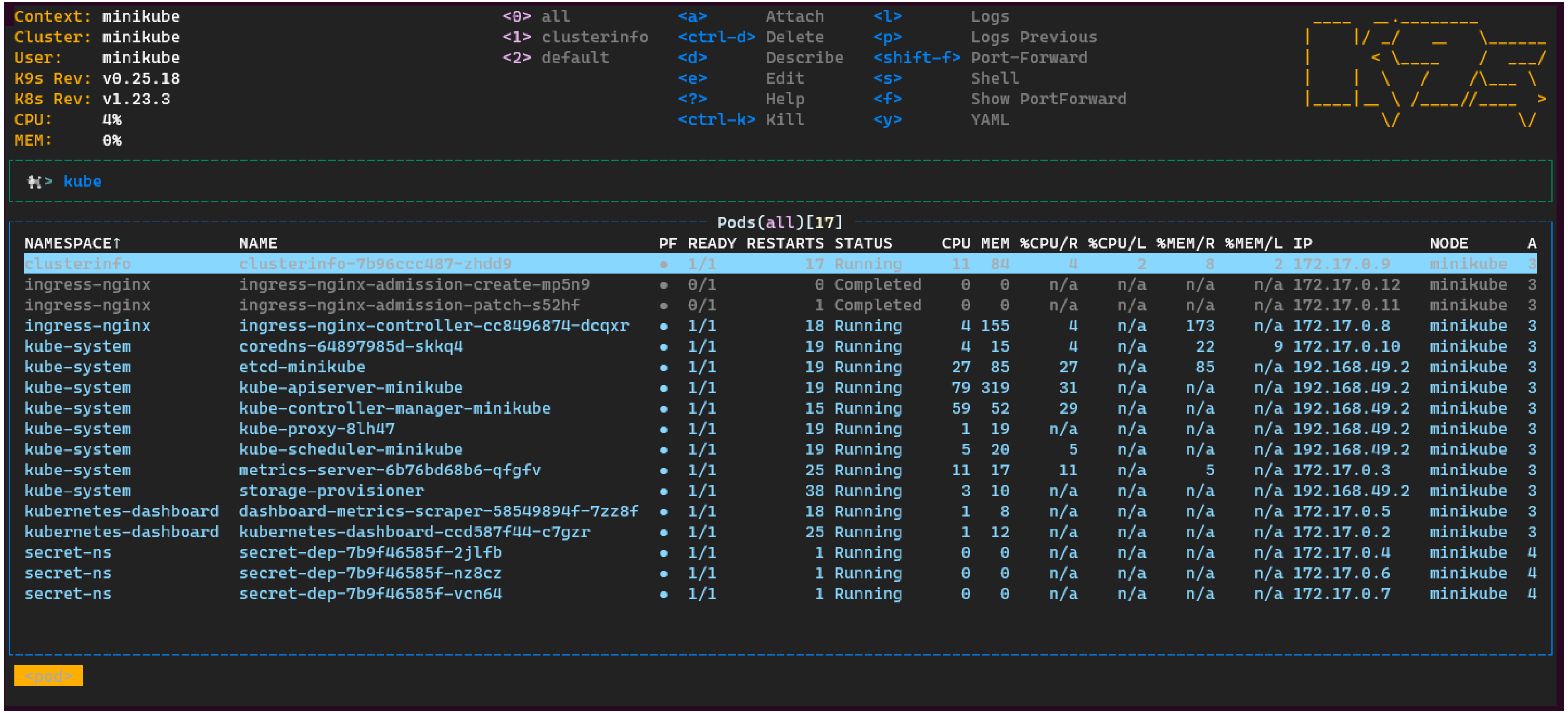
K9s

Observability

- Terminal based UI to interact with your Kubernetes clusters
- The aim of this project is to make it easier to navigate, observe and manage your deployed applications in the wild
- K9s continually watches Kubernetes for changes and offers subsequent commands to interact with your observed resources.

K9s

Observability



```
Context: minikube                                   <0> all            <a>      Attach     <l>       Logs              ____  __.________
Cluster: minikube                                   <1> clusterinfo    <ctrl-d> Delete     <p>       Logs Previous    |    |/ _/   __   \______
User:    minikube                                   <2> default        <d>      Describe   <shift-f> Port-Forward     |      < \____    /  ___/
K9s Rev: v0.25.18                                                      <e>      Edit       <s>       Shell            |    |  \   /    /\___ \
K8s Rev: v1.23.3                                                       <?>      Help       <f>       Show PortForward |____|__ \ /____//____  >
CPU:     4%                                                            <ctrl-k> Kill       <y>       YAML                     \/            \/
MEM:     0%

 > kube

                                                    Pods(all)[17]
 NAMESPACE↑             NAME                                      PF READY RESTARTS STATUS     CPU MEM %CPU/R %CPU/L %MEM/R %MEM/L IP            NODE     A
 clusterinfo            clusterinfo-7b96ccc487-zhdd9              ●  1/1         17 Running     11  84      4      2      8      2 172.17.0.9    minikube 3
 ingress-nginx          ingress-nginx-admission-create-mp5n9      ●  0/1          0 Completed    0   0    n/a    n/a    n/a    n/a 172.17.0.12   minikube 3
 ingress-nginx          ingress-nginx-admission-patch-s52hf       ●  0/1          1 Completed    0   0    n/a    n/a    n/a    n/a 172.17.0.11   minikube 3
 ingress-nginx          ingress-nginx-controller-cc8496874-dcqxr  ●  1/1         18 Running      4 155      4    n/a    173    n/a 172.17.0.8    minikube 3
 kube-system            coredns-64897985d-skkq4                   ●  1/1         19 Running      4  15      4    n/a     22      9 172.17.0.10   minikube 3
 kube-system            etcd-minikube                             ●  1/1         19 Running     27  85     27    n/a     85    n/a 192.168.49.2  minikube 3
 kube-system            kube-apiserver-minikube                   ●  1/1         19 Running     79 319     31    n/a    n/a    n/a 192.168.49.2  minikube 3
 kube-system            kube-controller-manager-minikube          ●  1/1         15 Running     59  52     29    n/a    n/a    n/a 192.168.49.2  minikube 3
 kube-system            kube-proxy-8lh47                          ●  1/1         19 Running      1  19    n/a    n/a    n/a    n/a 192.168.49.2  minikube 3
 kube-system            kube-scheduler-minikube                   ●  1/1         19 Running      5  20      5    n/a    n/a    n/a 192.168.49.2  minikube 3
 kube-system            metrics-server-6b76bd68b6-qfgfv           ●  1/1         25 Running     11  17     11    n/a      5    n/a 172.17.0.3    minikube 3
 kube-system            storage-provisioner                       ●  1/1         38 Running      3  10    n/a    n/a    n/a    n/a 192.168.49.2  minikube 3
 kubernetes-dashboard   dashboard-metrics-scraper-58549894f-7zz8f ●  1/1         18 Running      1   8    n/a    n/a    n/a    n/a 172.17.0.5    minikube 3
 kubernetes-dashboard   kubernetes-dashboard-ccd587f44-c7gzr      ●  1/1         25 Running      1  12    n/a    n/a    n/a    n/a 172.17.0.2    minikube 3
 secret-ns              secret-dep-7b9f46585f-2jlfb               ●  1/1          1 Running      0   0    n/a    n/a    n/a    n/a 172.17.0.4    minikube 4
 secret-ns              secret-dep-7b9f46585f-nz8cz               ●  1/1          1 Running      0   0    n/a    n/a    n/a    n/a 172.17.0.6    minikube 4
 secret-ns              secret-dep-7b9f46585f-vcn64               ●  1/1          1 Running      0   0    n/a    n/a    n/a    n/a 172.17.0.7    minikube 4

<pod>
```

moongy.

Demo: Kubernetes Dashboard

Auto-Scalling: HPA

Motivation

HPA

- Kubernetes can handle several replicas of the same pods
    - ReplicaSets handle replication
    - Services handle load balancing between them
- However, if the demand of a service starts to grow, the number of replicas deployed may be not sufficient to handle requests
- Number of replicas can be changed manually but it's not scalable
- Kubernetes have a HorizontalPodAutoscaller (HPA) object to handle scalability of a Deployment automatically

HorinzontalPodAutoscaler

HPA

- Horizontal scaling means that the response to increased load is to deploy more Pods
- HPA defines a minimum and maximum number of replicas
- If the load increases, and the number of Pods is below the configured maximum, the HPA instructs the Deployment to scale up
- If the load decreases, and the number of Pods is above the configured minimum, the HPA instructs the workload resource to scale down
- HPA uses an interval (default is 15 seconds) to check if some change is needed

Metrics

HPA

- To make the decision about scaling, HPA uses metrics about pods resources (CPU, Memory) utilization
- Metric target can be set as a percentage or an average value (preferable)
- Value used for check need to scale up/down is the average utilization of all pods
- Is mandatory that pods have resources (limits) defined

```
desiredReplicas = ceil[currentReplicas * (currentMetricValue /
                          desiredMetricValue)]
```

moOngy.

Demo: HPA

Lab

Lab 9: Monitoring and Operation

Github

Lab 09 - Monitoring and Operation | docker-kubernetes-training (tasb.github.io)

moOngy.
Minds on the move

Rua Sousa Martins, nº 10
1050-218 Lisboa | Portugal