Roberto Morassi del Blanco

A0265239X

<https://github.com/rmorassi/CS3219-OTOT-A2-A3>

**Task A3.1**

**Commands to create metrics-server and verify it works**

1. Run “kubectl apply -f https://github.com/kubernetes-sigs/metrics-server/releases/latest/download/components.yaml” to download the metrics-server from online and apply it to our clusters.

Graphical user interface, text

Description automatically generated with medium confidence

1. Execute “kubectl -nkube-system edit deploy/metrics-server” to edit the metrics-server and add the “--kubelet-insecure-tls” flag to deployment.spec.containers[].args[]. Below we can see how I added it.

Text

Description automatically generated

1. Run “kubectl -nkube-system rollout restart deploy/metrics-server” to restart metrics-server.



**Commands to create HPA and verify it works**

1. Run “kubectl apply -f k8s/manifests/k8/backend-hpa.yaml” to apply the HPA.



1. To test the HPA to see if it works, I tried stressing the system in different ways
   1. I initially only queried the app intermittently to not stress the servers too much. As seen in screenshot below, as the load is below target, and the system does not need to scale up (see number of deployed pods and desired pods – stays constant).

Text

Description automatically generated

* 1. I then queried very the app frequently to stress the servers above the target. As seen in the screenshot below, as the load increased above the target, the system scaled up (see number of deployed pods and desired pods – wants more pods).

Text

Description automatically generated

* 1. I then stopped querying the app frequently to destress the servers below the target. As seen in the screenshot below, as the load is decreased below the target, the system scaled down (see number of deployed pods and desired pods – wants less).

Text

Description automatically generated

1. We can confirm that the pods are running in different zones by executing “kubectl get nodes -L topology.kubernetes.io/zone”

Text

Description automatically generated

**Task A3.2**

**Commands to create the Deployment and verify it works**

1. We apply the zone aware deployment with the command “kubectl apply -f k8s/manifests/k8/backend-deployment-zone-aware.yaml”



1. We recall that “worker” and “worker2” are running in zone a while “worker3” is running in zone b. We run “kubectl get po -lapp=backend-zone-aware -owide --sort-by='.spec.nodeName'” to check how many pods are assigned to each node. We can see that there are 4 pods assigned to worker, 1 to worker 2, and 5 to worker3. This means that there are 5 pods running in zone a and 5 pods in zone b, and thus we know that the pods are uniformly distributed across the two zones.

