Roberto Morassi del Blanco

A0265239X

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

**Task A2.1**

**Commands to create the cluster**

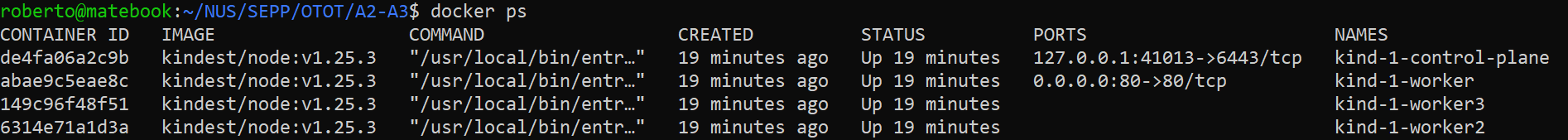
1. “kind create cluster --name kind-1 --config k8s/kind/cluster-config.yaml” creates the cluster from the configuration file and launches it.

Text

Description automatically generated

**Commands to verify the cluster**

1. The nodes can be seen running with the command “docker ps”.

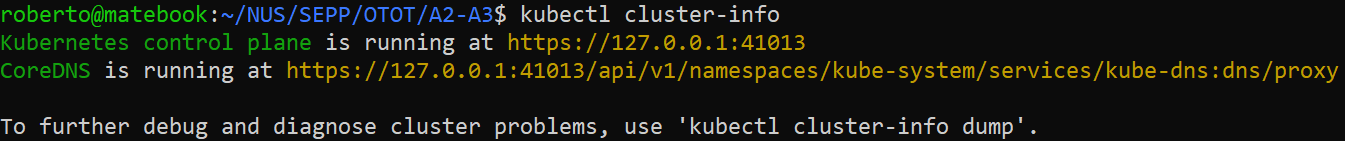


1. “kubectl get nodes” can also be used for the same purpose.

Text

Description automatically generated

1. “kubectl cluster-info” can let us know the IP Address and ports of clusters that are currently running. We see that the open ports match with what is displayed in “docker ps”.



**Task A2.2**

**Commands to create and verify Deployment in Cluster**

1. We run “kubectl apply -f k8s/manifests/k8/backend-deployment.yaml” to apply the deployment into the cluster. Note that the file was modified to point to the 8080 port which aligns with the port of “app” created in A1.



1. We run “kubectl get po -lapp=backend –watch” to check that the deployment is working properly. Since the status went from “ContainerCreating” to “Running”, we can confirm that the “app” image was correctly pulled from the online repository successfully (uploaded manually for this assignment).

A screenshot of a computer

Description automatically generated with medium confidence

**Commands to create and verify Service in Cluster**

1. Run “kubectl apply -f k8s/manifests/k8/backend-service.yaml” to create the service in our cluster.



1. Run “kubectl get service” to confirm the service is running correctly.

Text

Description automatically generated

**Task A2.3**

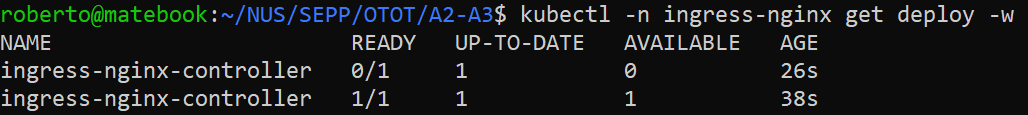
**Commands create ingress-controller and verify it works**

1. We run “kubectl apply -f https://raw.githubusercontent.com/kubernetes/ingress-nginx/main/deploy/static/provider/kind/deploy.yaml” to pull the ingress controller file from the internet and apply it to our cluster.

Text

Description automatically generated

1. We run “kubectl -n ingress-nginx get deploy -w” to check when the ingress controller starts working and check if it is functioning correctly.



**Commands create Ingress Object and verify it works**

1. Run “kubectl apply -f k8s/manifests/k8/backend-ingress.yaml” to configure Ingress Object.



1. Run “kubectl get ingress -w” to make sure the Ingress Object is running and configured correctly.

Text

Description automatically generated

1. Run “kubectl get nodes -L ingress-ready” to check if the ingress is configured correctly.

Text

Description automatically generated

**Screenshot proof of the websites working as usual**

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application, Word

Description automatically generated