GitHub Link: https://github.com/HaydenDuong/SIT323\_Cloud\_Native\_Application\_Development/tree/main/6.1P

1/ Setup the Kubernetes Cluster

* Navigate to Docker Desktop application.
* Find the Setting option on the top-right of the navigation bar of the application, it will display the following (Image 1).

A screenshot of a computer

AI-generated content may be incorrect.

Image 1

* Select “Kubernetes” option from the left-side list (Image 2).

A screenshot of a computer

AI-generated content may be incorrect.

Image 2

* Enabling Kubernetes by:
  + Select “Enable Kubernetes”.
  + In “Cluster settings”, select “Kubeadm”.
  + Click “Apply & restart”.
  + Wait for short period of time for Docker to download the necessary files to construct “Kubeadm” – done when near the bottom-left corner shown “Kubernetes running”.
* Check the cluster information by type in command “kubectl clutser-info” into Command Prompt. (Image 3)

A screen shot of a computer

AI-generated content may be incorrect.

2/ Create the Docker Image

* In task 5.1P, I had created an Docker Image based on task 4.2C, named “calculator:v1” (Image 4):

A screen shot of a computer

AI-generated content may be incorrect.

Image 4

* Create another image of the same application will add another 1.35GB – which is not a good choice, thus, I “recycled” the old Docker Image from task 5.1P.

3/ Create the Kubernetes Deployment

* There are 2 ways to do this:
  + Through command “kubectl create deployment <name-of-deployment> --image=<name-of-Docker-Image>”.
  + Through writing a YAML file – which is allow user to have more control over the deployment.
* For this task, I chose to write YAML file and its structure is as follows (Image 5):

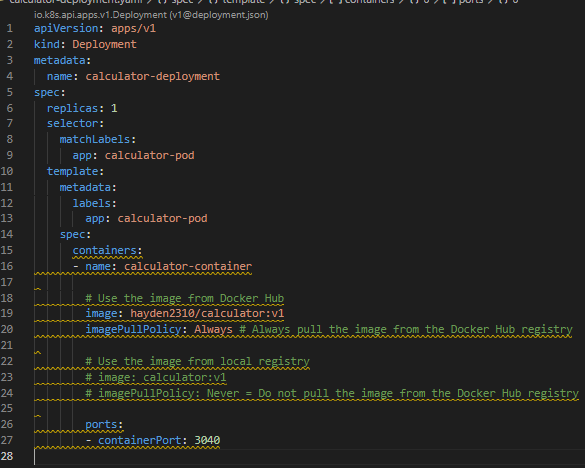


Image 5

* Input command: “kubectl apply -f <name-of-YAML-file>” (Image 6)



Image 6

* Check the status of this newly generated deployment through “kubectl get deployments” (Image 7):

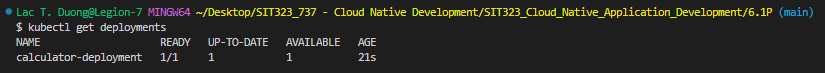


Image 7

4/ Create the Kubernetes Service

* Similar to K8s Deployment, a Service can be created by one of these two ways:
  + “kubectl expose <name-of-the-running-deployment> --type=<Option> --port 80 –target-port <port-that-the-application-running>” – this method will let kubernetes randomly choose a port (30000 – 32767) to allow external applications to accessing to the calculator app inside the container (not flexible)
  + Write a YAML file (allow easily configuration later on)
* The structure of the YAML is as follows (Image 8):

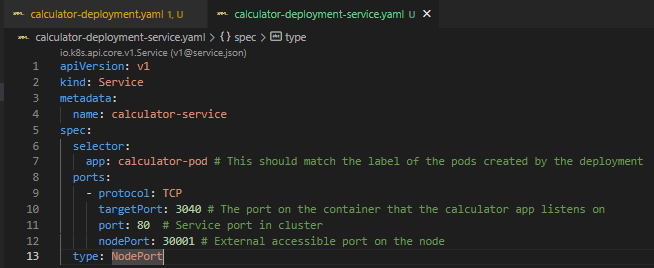


Image 8

* Apply this YAML file with “kubectl apply -f <name-of-YAML-file>”
* Check the generated service with “kubectl get services” (Image 9)

A screen shot of a computer

AI-generated content may be incorrect.

Image 9

* Test the application via //http:localhost:30001/add?a=10&b=10

