

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).

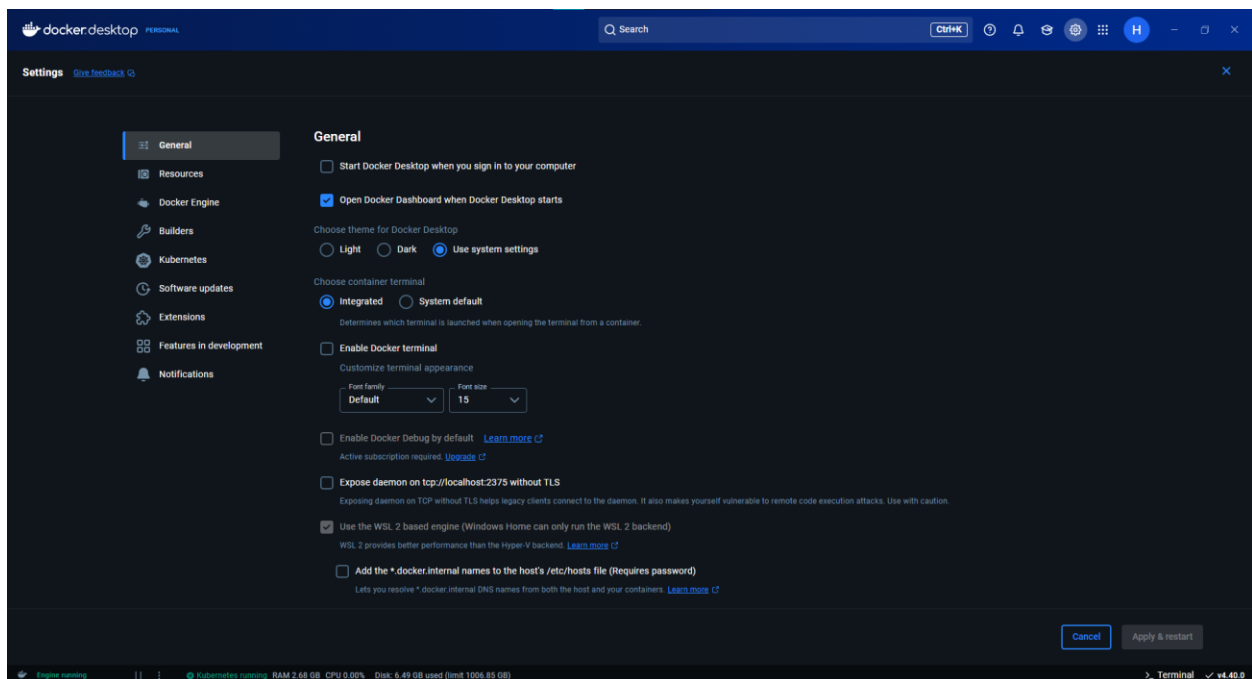


Image 1

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

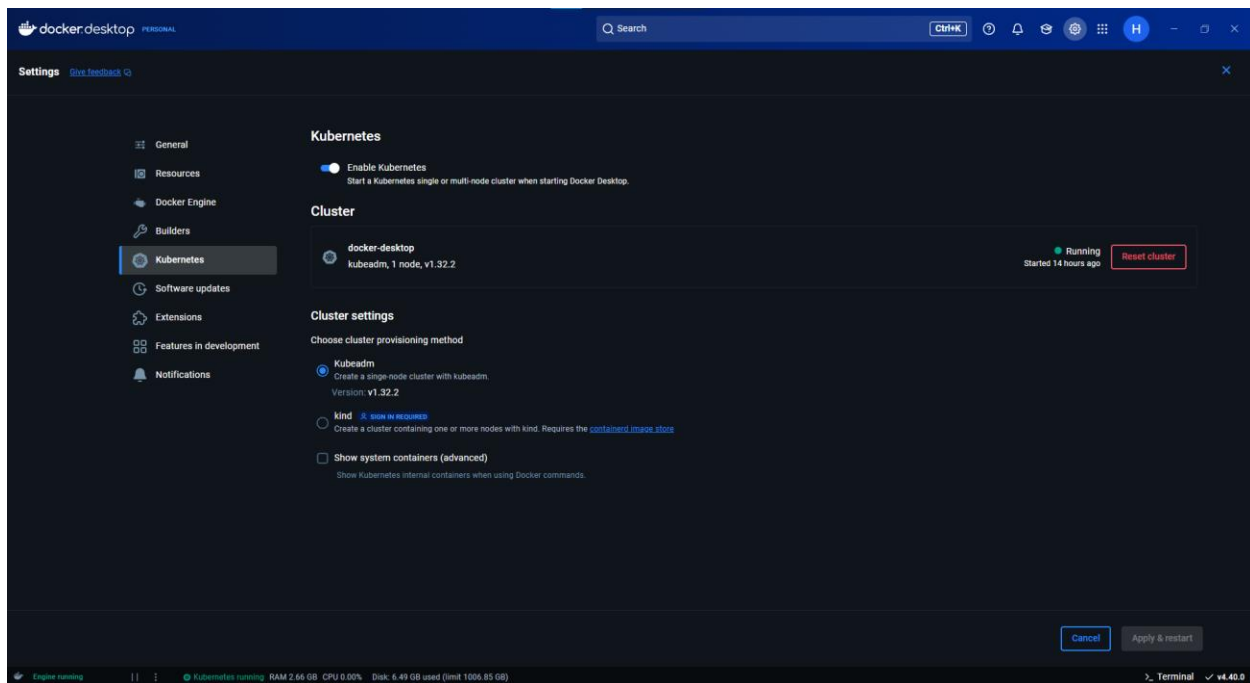
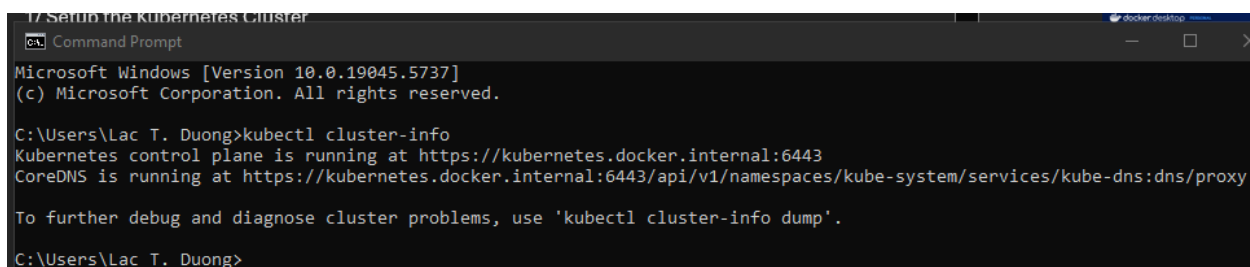


Image 2

- 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 cluster-info” into Command Prompt. (Image 3)



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):

```
Lac T. Duong@legion-7 MINGW64 ~/Desktop/SIT323_737 - Cloud Native Development/SIT323_Cloud_Native_Application_Development/6.1P (main)
```

```
$ docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
australia-southeast2-docker.pkg.dev/sit323-25t1-duong-102550c/calculator-repo/calculator	v1	4d19093419e0	12 days ago	1.35GB
hayden2310/calculator	v1	4d19093419e0	12 days ago	1.35GB
calculator	v1	4d19093419e0	12 days ago	1.35GB
docker/desktop-kubernetes	kubernetes-v1.32.2-cni-v1.6.0-critools-v1.31.1-cri-dockerd-v0.3.16-1-debian	eeef9515fbfb	7 weeks ago	412MB
registry.k8s.io/kube-apiserver	v1.32.2	85b7a174738b	2 months ago	97MB
registry.k8s.io/kube-scheduler	v1.32.2	d8e673e7c998	2 months ago	69.6MB
registry.k8s.io/kube-proxy	v1.32.2	f1332858868e	2 months ago	94MB
registry.k8s.io/kube-controller-manager	v1.32.2	b6a454c5a800	2 months ago	89.7MB
registry.k8s.io/etcd	3.5.16-0	a9e7e6b294ba	7 months ago	150MB
registry.k8s.io/coredns/coredns	v1.11.3	c69fa2e9cbf5	8 months ago	61.8MB
registry.k8s.io/pause	3.10	873ed7510279	10 months ago	736KB
docker/labs-k8s-toolkit-extension	0.0.46	b4b9195545cc	16 months ago	132MB
docker/desktop-vpnkit-controller	d331cb22850be8cd97c84a9cfeca44a1afb6e	556080075b3d	23 months ago	36.2MB
docker/desktop-storage-provisioner	v2.0	99f89471f470	3 years ago	41.5MB

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):

```

calculator-deployment.yaml / ( / spec / ( / template / ( / spec / ( / containers / ( / 0 / ( / ports / ( / 0
io.k8s.api.apps.v1.Deployment (v1@deployment.json)
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: calculator-deployment
5  spec:
6    replicas: 1
7    selector:
8      matchLabels:
9        app: calculator-pod
10   template:
11     metadata:
12       labels:
13         app: calculator-pod
14     spec:
15       containers:
16         - name: calculator-container
17
18       # Use the image from Docker Hub
19       image: hayden2310/calculator:v1
20       imagePullPolicy: Always # Always pull the image from the Docker Hub registry
21
22       # Use the image from local registry
23       # image: calculator:v1
24       # imagePullPolicy: Never = Do not pull the image from the Docker Hub registry
25
26       ports:
27         - containerPort: 3040
28

```

Image 5

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

```

• Lac T. Duong@Legion-7 MINGW64 ~/Desktop/SIT323_737 - Cloud Native Development/SIT323_Cloud_Native_Application_Development/6.1P (main)
$ kubectl apply -f calculator-deployment.yaml
deployment.apps/calculator-deployment created

```

Image 6

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

```

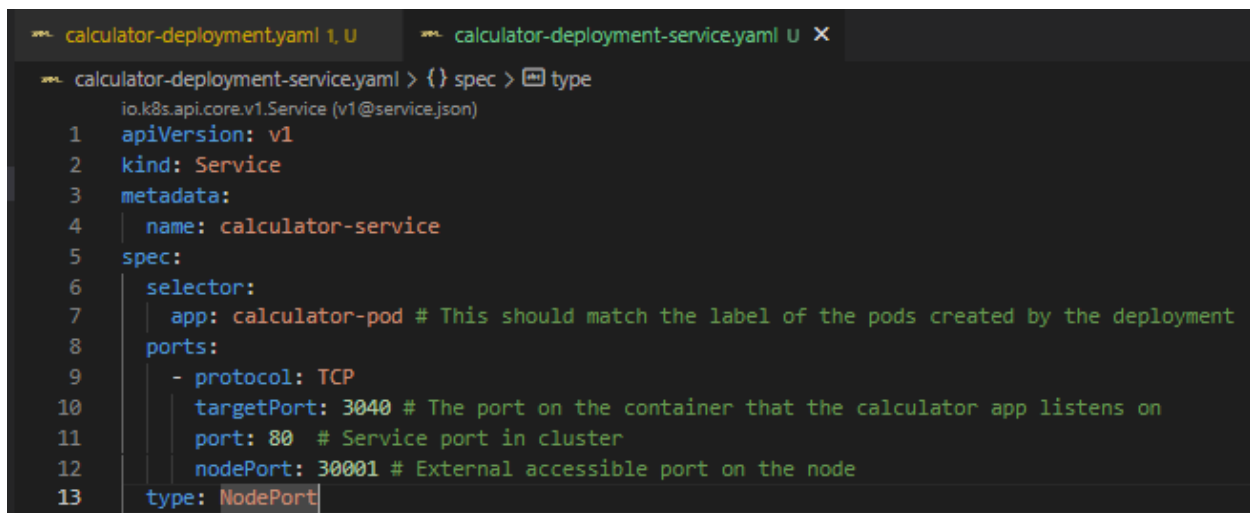
• Lac T. Duong@Legion-7 MINGW64 ~/Desktop/SIT323_737 - Cloud Native Development/SIT323_Cloud_Native_Application_Development/6.1P (main)
$ kubectl get deployments
NAME                READY   UP-TO-DATE   AVAILABLE   AGE
calculator-deployment 1/1     1            1           21s

```

Image 7

4/ Create the Kubernetes Service

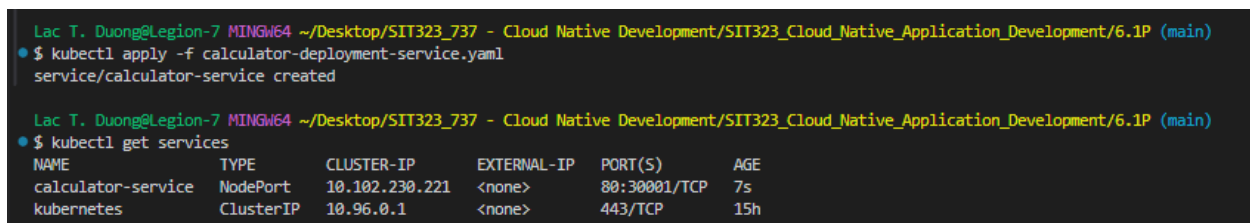
- Similar to K8s Deployment, a Service can be created by one of these two ways:
 - “kubectl expose <name-of-the-running-deployment> --type=<Option> --port 80 --target-port <port-that-the-application-running>” – this method will let kubernetes randomly choose a port (30000 – 32767) to allow external applications 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):



```
calculator-deployment-service.yaml > {} spec > type
io.k8s.api.core.v1.Service (v1@service.json)
1  apiVersion: v1
2  kind: Service
3  metadata:
4    name: calculator-service
5  spec:
6    selector:
7      app: calculator-pod # This should match the label of the pods created by the deployment
8    ports:
9      - protocol: TCP
10        targetPort: 3040 # The port on the container that the calculator app listens on
11        port: 80 # Service port in cluster
12        nodePort: 30001 # External accessible port on the node
13    type: NodePort
```

Image 8

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



```
Lac T. Duong@Legion-7 MINGW64 ~/Desktop/SIT323_737 - Cloud Native Development/SIT323_Cloud_Native_Application_Development/6.1P (main)
$ kubectl apply -f calculator-deployment-service.yaml
service/calculator-service created

Lac T. Duong@Legion-7 MINGW64 ~/Desktop/SIT323_737 - Cloud Native Development/SIT323_Cloud_Native_Application_Development/6.1P (main)
$ kubectl get services
NAME                TYPE        CLUSTER-IP    EXTERNAL-IP  PORT(S)          AGE
calculator-service  NodePort    10.102.230.221 <none>        80:30001/TCP     7s
kubernetes          ClusterIP   10.96.0.1     <none>        443/TCP          15h
```

Image 9

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

