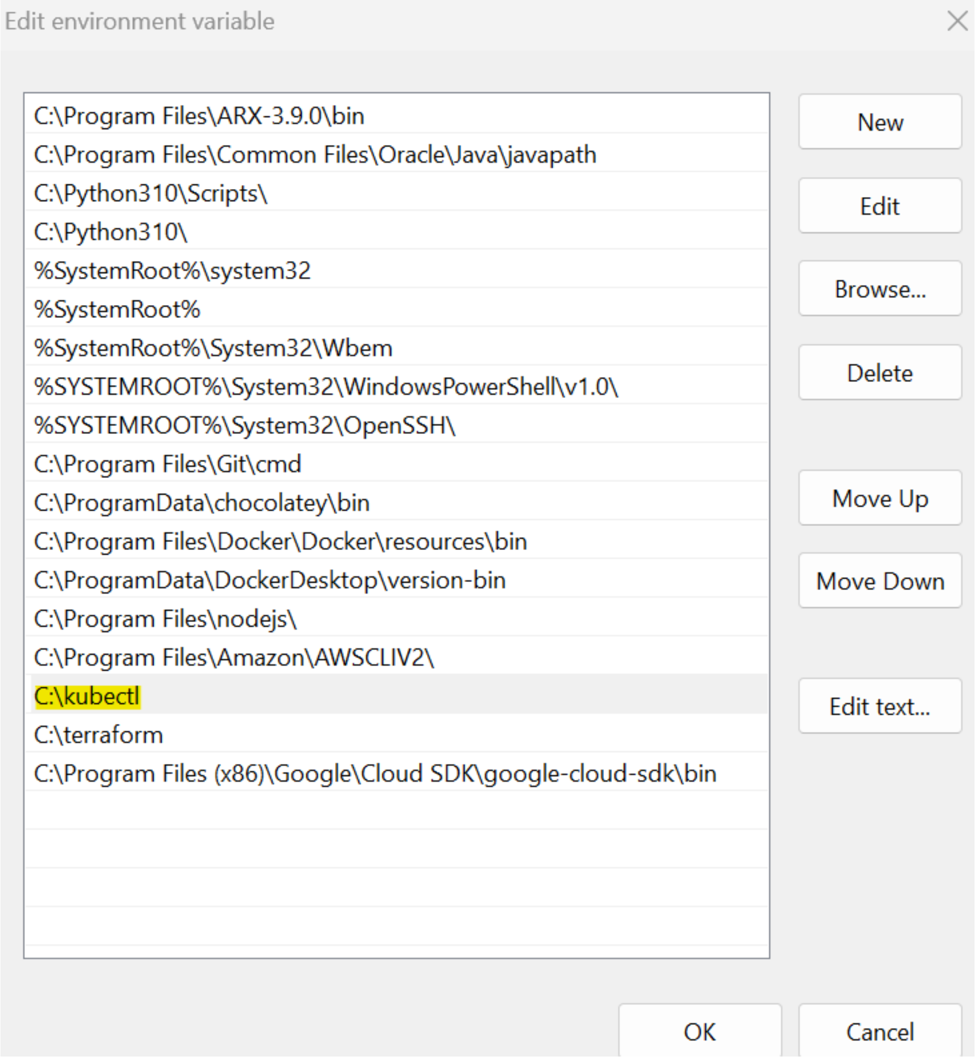
7.1P: CREATING A KUBERNETES CLUSTER FOR A CONTAINERISED APPLICATION

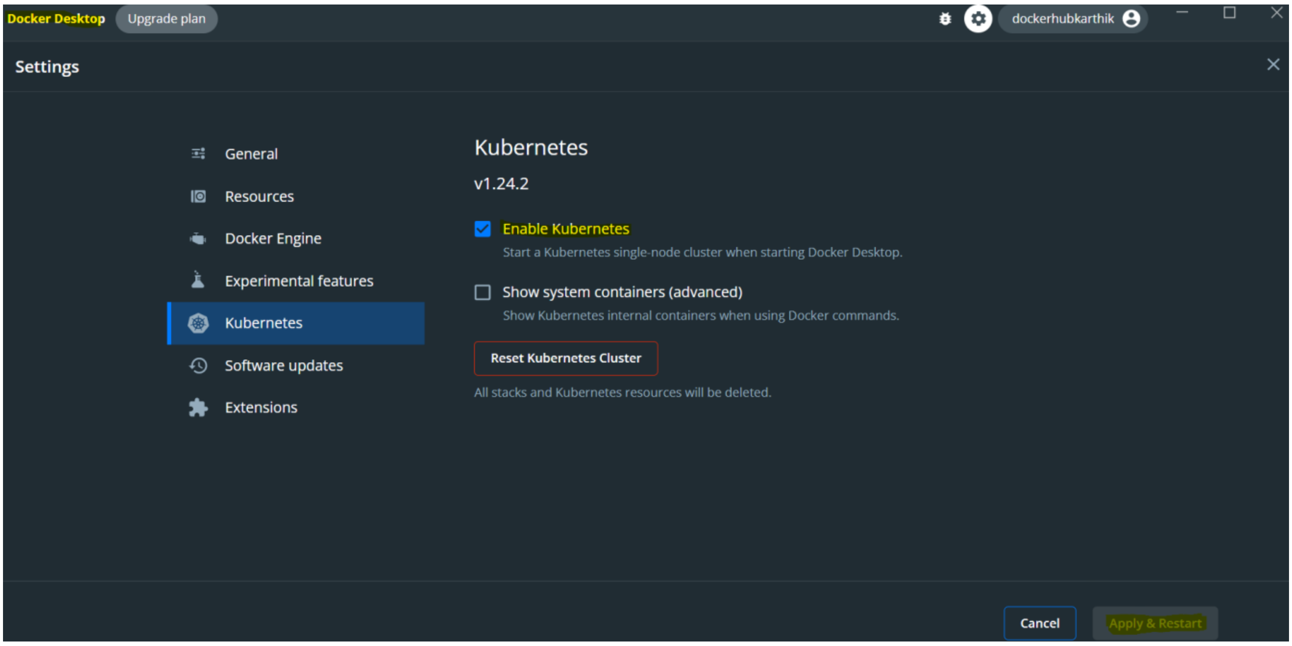
• Setup the Kubernetes Cluster

1. To setup the kubectl,
   - Downloaded kubectl from https://kubernetes.io/docs/tasks/tools/install-kubectl-windows/
   - Set the path environment variable to point to kubectl folder



```
C:\Program Files\ARX-3.9.0\bin
C:\Program Files\Common Files\Oracle\Java\javapath
C:\Python310\Scripts\
C:\Python310\
%SystemRoot%\system32
%SystemRoot%
%SystemRoot%\System32\Wbem
%SYSTEMROOT%\System32\WindowsPowerShell\v1.0\
%SYSTEMROOT%\System32\OpenSSH\
C:\Program Files\Git\cmd
C:\ProgramData\chocolatey\bin
C:\Program Files\Docker\Docker\resources\bin
C:\ProgramData\DockerDesktop\version-bin
C:\Program Files\nodejs\
C:\Program Files\Amazon\AWSCLIV2\
C:\kubectl
C:\terraform
C:\Program Files (x86)\Google\Cloud SDK\google-cloud-sdk\bin
```

2. Enabled Kubernetes in Docker Desktop application > Apply & Restart



Kubernetes

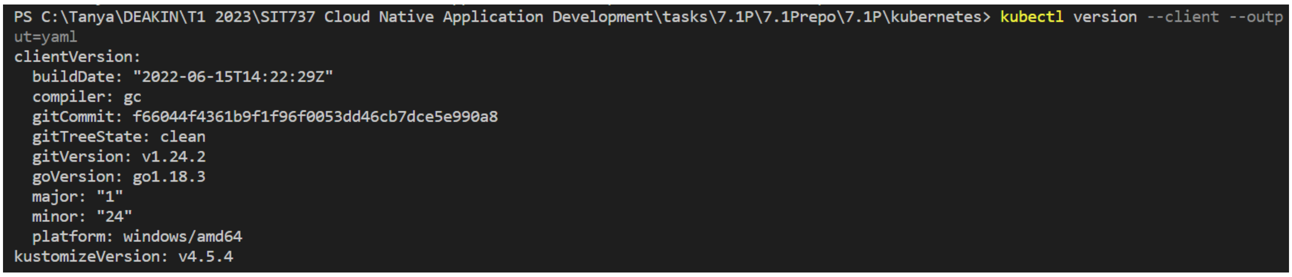
v1.24.2

☑ Enable Kubernetes
Start a Kubernetes single-node cluster when starting Docker Desktop.

□ Show system containers (advanced)
Show Kubernetes internal containers when using Docker commands.

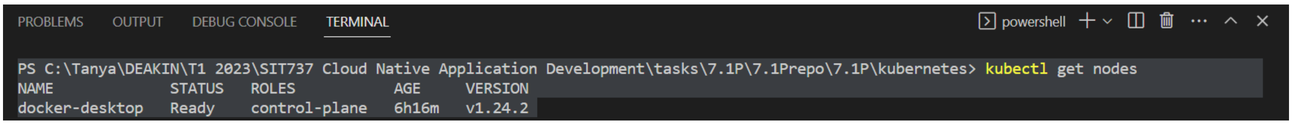
All stacks and Kubernetes resources will be deleted.

3. The below ensures that Kubernetes is enabled –
   Command → kubectl version –client –output=yaml



```
PS C:\Tanya\DEAKIN\T1 2023\SIT737 Cloud Native Application Development\tasks\7.1P\7.1Prepo\7.1P\kubernetes> kubectl version --client --outp
ut=yaml
clientVersion:
  buildDate: "2022-06-15T14:22:29Z"
  compiler: gc
  gitCommit: f66044f4361b9f1f96f0053dd46cb7dce5e990a8
  gitTreeState: clean
  gitVersion: v1.24.2
  goVersion: go1.18.3
  major: "1"
  minor: "24"
  platform: windows/amd64
kustomizeVersion: v4.5.4
```

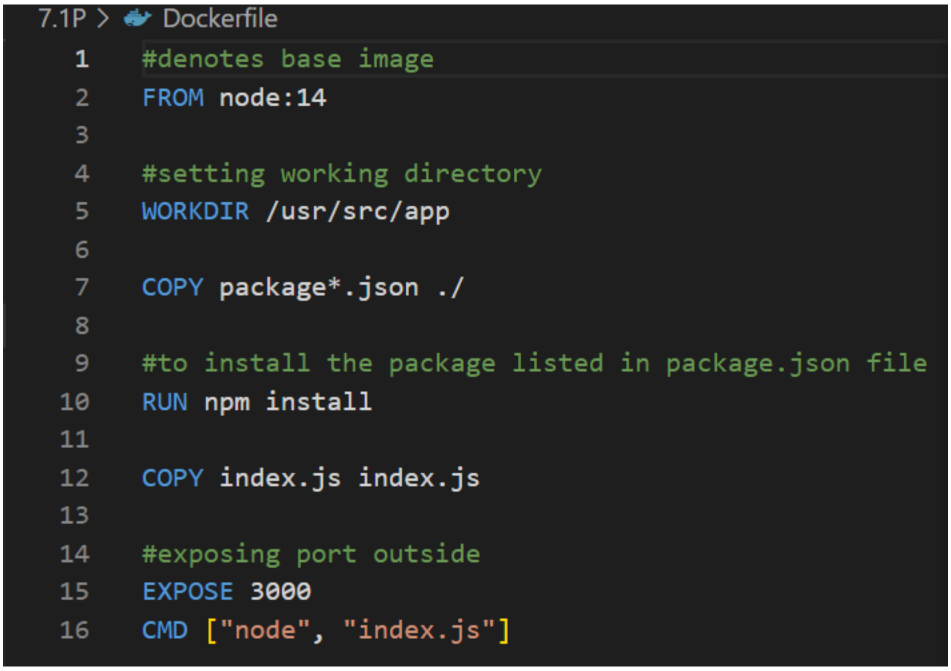
Also checked → kubectl get nodes



```
PS C:\Tanya\DEAKIN\T1 2023\SIT737 Cloud Native Application Development\tasks\7.1P\7.1Prepo\7.1P\kubernetes> kubectl get nodes
NAME             STATUS   ROLES           AGE     VERSION
docker-desktop   Ready    control-plane   6h16m   v1.24.2
```

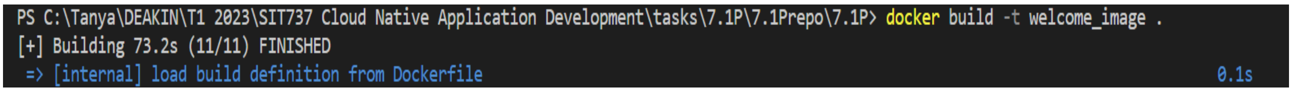
- **Create the Docker Image**

Created the docker file for Node.js application first –



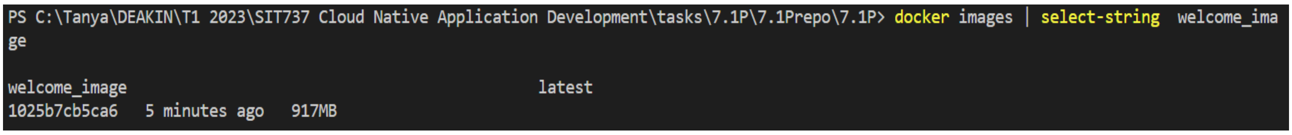
```
#denotes base image
FROM node:14

#setting working directory
WORKDIR /usr/src/app

COPY package*.json ./

#to install the package listed in package.json file
RUN npm install

COPY index.js index.js

#exposing port outside
EXPOSE 3000
CMD ["node", "index.js"]
```

Then, used the command → docker build -t welcome_image .



```
PS C:\Tanya\DEAKIN\T1 2023\SIT737 Cloud Native Application Development\tasks\7.1P\7.1Prepo\7.1P> docker build -t welcome_image .
[+] Building 73.2s (11/11) FINISHED
 => [internal] load build definition from Dockerfile                                                             0.1s
```

This confirms the creation of the image, command → docker images | select-string welcome_image



```
PS C:\Tanya\DEAKIN\T1 2023\SIT737 Cloud Native Application Development\tasks\7.1P\7.1Prepo\7.1P> docker images | select-string  welcome_ima
ge

welcome_image                                       latest
1025b7cb5ca6    5 minutes ago    917MB
```

8. Push the Docker image to a registry

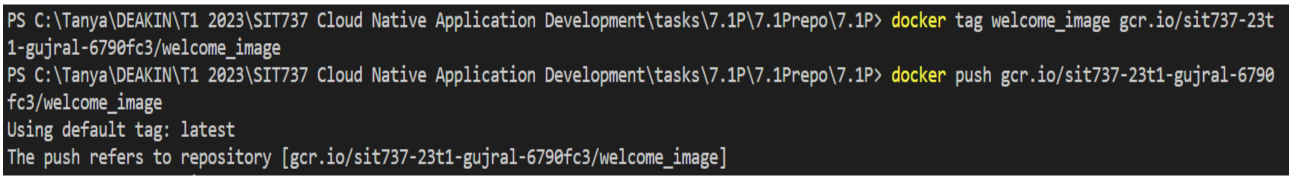
Tagged the image using → docker tag <image_name> gcr.io/<project_id>/<image_name>

Command used → docker tag welcome_image gcr.io/sit737-23t1-gujral-6790fc3/welcome_image

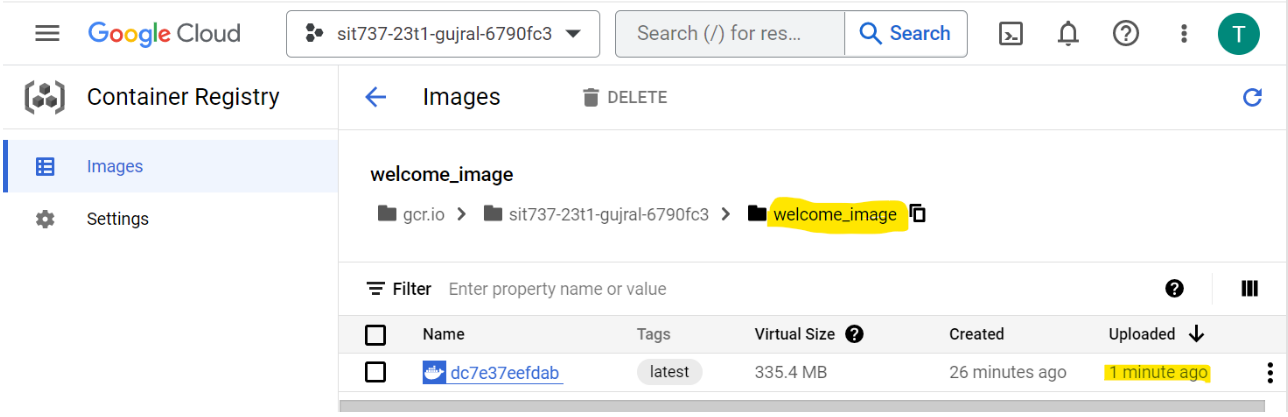
And,

Pushed the image using → docker push gcr.io/<project_id>/<image_name>

Command used → docker push gcr.io/sit737-23t1-gujral-6790fc3/welcome_image



```
PS C:\Tanya\DEAKIN\T1 2023\SIT737 Cloud Native Application Development\tasks\7.1P\7.1Prepo\7.1P> docker tag welcome_image gcr.io/sit737-23t
1-gujral-6790fc3/welcome_image
PS C:\Tanya\DEAKIN\T1 2023\SIT737 Cloud Native Application Development\tasks\7.1P\7.1Prepo\7.1P> docker push gcr.io/sit737-23t1-gujral-6790
fc3/welcome_image
Using default tag: latest
The push refers to repository [gcr.io/sit737-23t1-gujral-6790fc3/welcome_image]
```



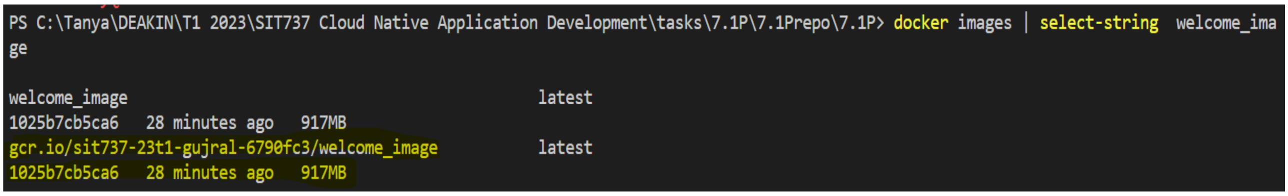
welcome_image

gcr.io > sit737-23t1-gujral-6790fc3 > welcome_image

Filter Enter property name or value

□	Name	Tags	Virtual Size	Created	Uploaded
□	dc7e37eefdab	latest	335.4 MB	26 minutes ago	1 minute ago

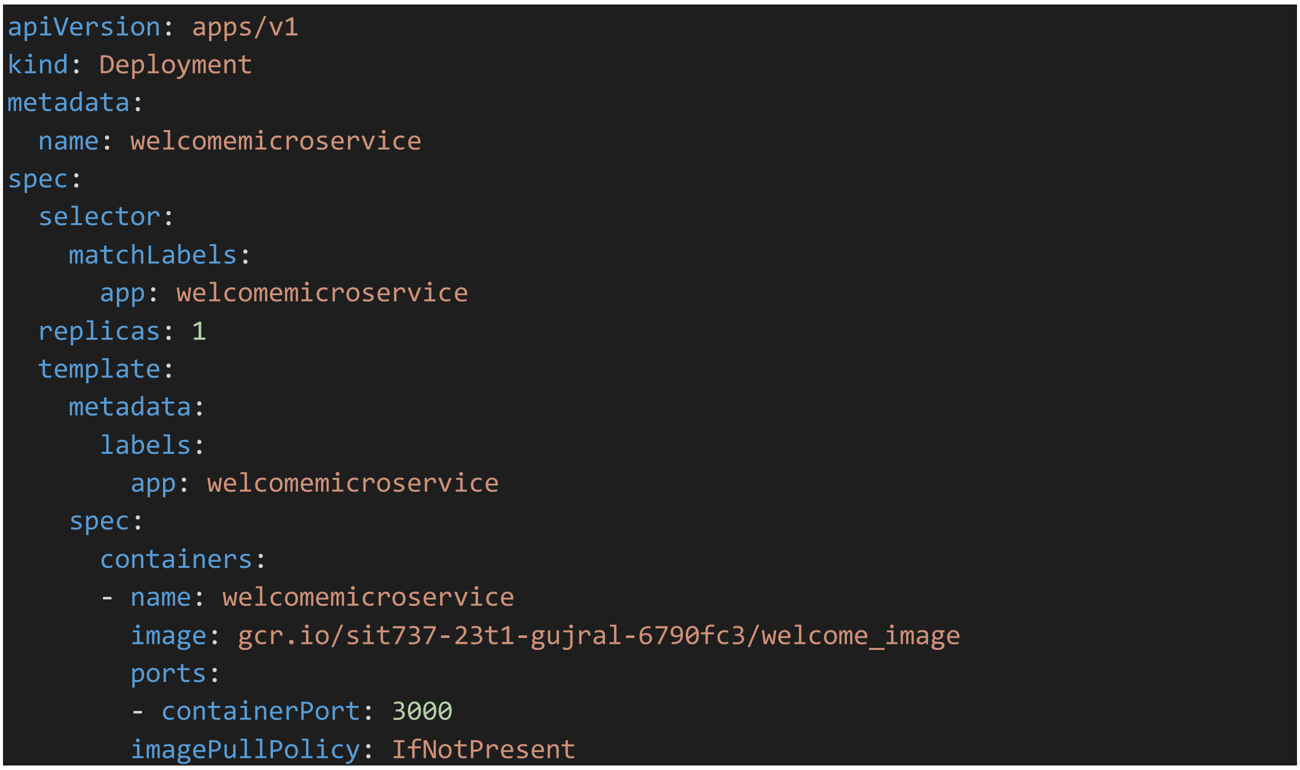
Confirmed the push to GCR



```
PS C:\Tanya\DEAKIN\T1 2023\SIT737 Cloud Native Application Development\tasks\7.1P\7.1Prepo\7.1P> docker images | select-string  welcome_ima
ge

welcome_image                                        latest
1025b7cb5ca6   28 minutes ago   917MB
gcr.io/sit737-23t1-gujral-6790fc3/welcome_image      latest
1025b7cb5ca6   28 minutes ago   917MB
```

- **Create the Kubernetes Deployment**

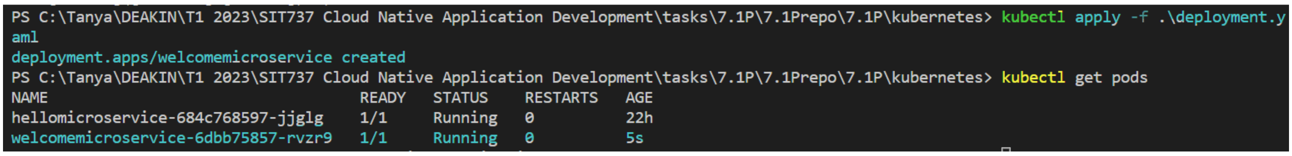
Kubernetes deployment configuration file for the application -



```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: welcomemicroservice
spec:
  selector:
    matchLabels:
      app: welcomemicroservice
  replicas: 1
  template:
    metadata:
      labels:
        app: welcomemicroservice
    spec:
      containers:
      - name: welcomemicroservice
        image: gcr.io/sit737-23t1-gujral-6790fc3/welcome_image
        ports:
        - containerPort: 3000
        imagePullPolicy: IfNotPresent
```

created the deployment service using the command → kubectl apply -f .\deployment.yaml

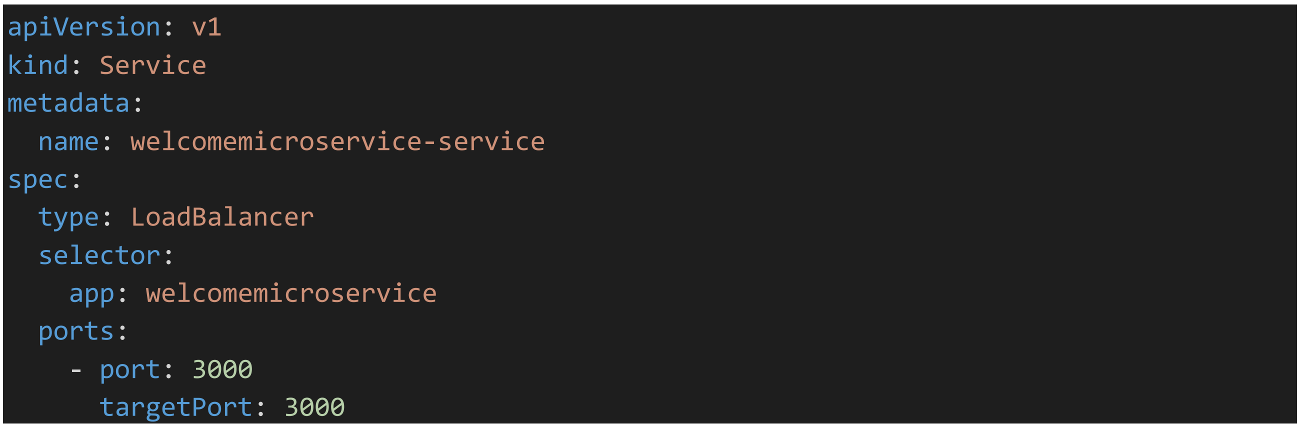
and checked using → kubectl get pods



```
PS C:\Tanya\DEAKIN\T1 2023\SIT737 Cloud Native Application Development\tasks\7.1P\7.1Prepo\7.1P\kubernetes> kubectl apply -f .\deployment.y
aml
deployment.apps/welcomemicroservice created
PS C:\Tanya\DEAKIN\T1 2023\SIT737 Cloud Native Application Development\tasks\7.1P\7.1Prepo\7.1P\kubernetes> kubectl get pods
NAME                                   READY   STATUS    RESTARTS   AGE
hellomicroservice-684c768597-jjglg     1/1     Running   0          22h
welcomemicroservice-6dbb75857-rvzr9    1/1     Running   0          5s
```

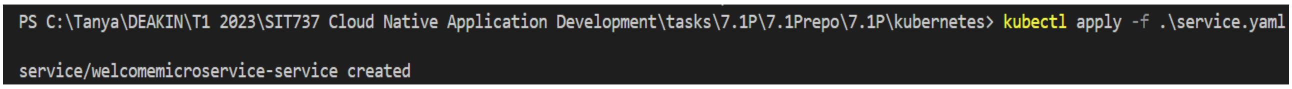
- **Create the Kubernetes Service**

Kubernetes service configuration file for the application –



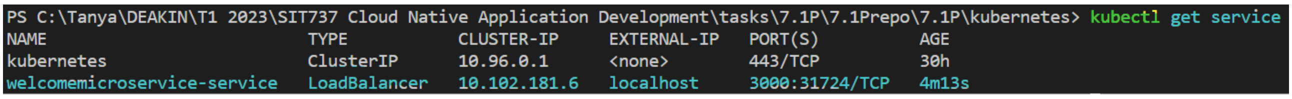
```yaml
apiVersion: v1
kind: Service
metadata:
  name: welcomemicroservice-service
spec:
  type: LoadBalancer
  selector:
    app: welcomemicroservice
  ports:
    - port: 3000
      targetPort: 3000
```

created the deployment service using the command → kubectl apply -f .\service.yaml



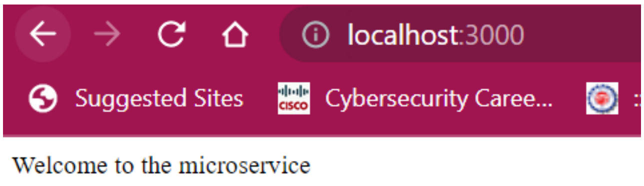
```
PS C:\Tanya\DEAKIN\T1 2023\SIT737 Cloud Native Application Development\tasks\7.1P\7.1Prepo\7.1P\kubernetes> kubectl apply -f .\service.yaml

service/welcomemicroservice-service created
```

and checked using → kubectl get service



```
PS C:\Tanya\DEAKIN\T1 2023\SIT737 Cloud Native Application Development\tasks\7.1P\7.1Prepo\7.1P\kubernetes> kubectl get service
NAME                          TYPE           CLUSTER-IP     EXTERNAL-IP   PORT(S)          AGE
kubernetes                    ClusterIP      10.96.0.1      <none>        443/TCP          30h
welcomemicroservice-service   LoadBalancer   10.102.181.6   localhost     3000:31724/TCP   4m13s
```

- Screenshots or video of the deployed application running on the Kubernetes cluster



Welcome to the microservice