7.2C: Interacting with Kubernetes

In this task I am going to demonstrate how I created Kubernetes dashboard and show its interaction with my application.

Work done in brief -

I. First I created the dashboard on my existing welcome application which was a basic node js application
II. I checked the dashboard state for the exiting application
III. Then I edited the node js application to incorporate a new image in it as required in the task
IV. I observed the subsequent changes picked up in the dashboard

(elaborated in detail in the walkthrough)

Instructions –

1. Import the 7.1P folder in Visual Studio Code
2. Get the bearer token for login
   Command → kubectl -n kubernetes-dashboard create token admin-user
3. Access the dashboard at http://localhost:8001/api/v1/namespaces/kubernetes-dashboard/services/https:kubernetes-dashboard:/proxy/#/log/default/welcomemicroservice-8fcc9cc-4b8c8/pod?namespace=default&container=welcomemicroservice
4. Enter the generated token to login

(For any further exploration in the code, the below Walkthrough section will demonstrate the steps)

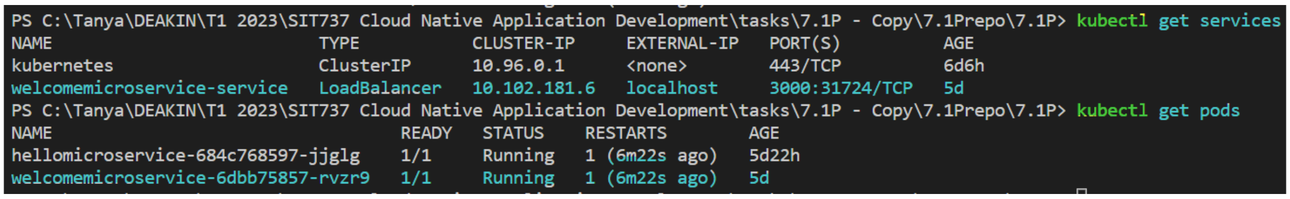
Detailed walkthrough for the task below -

1. CREATING THE DASHBOARD

- With my existing node.js application, I first checked the status of my application.

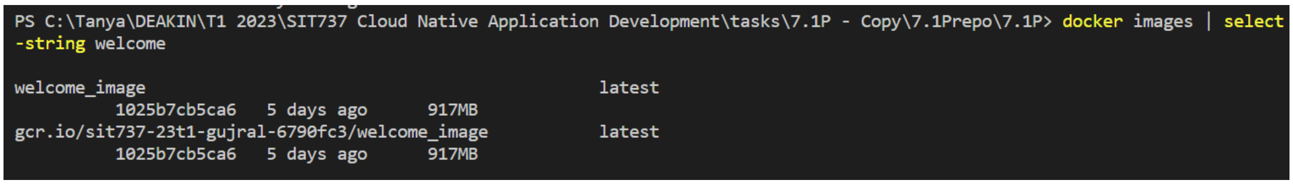
Commands used → 'kubectl get services', and 'kubectl get pods'

The highlighted output is for my microservice which is a simple 'Welcome to the microservice' application.



```
PS C:\Tanya\DEAKIN\T1 2023\SIT737 Cloud Native Application Development\tasks\7.1P - Copy\7.1Prepo\7.1P> kubectl get services
NAME                           TYPE           CLUSTER-IP      EXTERNAL-IP   PORT(S)          AGE
kubernetes                     ClusterIP      10.96.0.1       <none>        443/TCP          6d6h
welcomemicroservice-service    LoadBalancer   10.102.181.6    localhost     3000:31724/TCP   5d
PS C:\Tanya\DEAKIN\T1 2023\SIT737 Cloud Native Application Development\tasks\7.1P - Copy\7.1Prepo\7.1P> kubectl get pods
NAME                                  READY   STATUS    RESTARTS        AGE
hellomicroservice-684c768597-jjglg    1/1     Running   1 (6m22s ago)   5d22h
welcomemicroservice-6dbb75857-rvzr9   1/1     Running   1 (6m22s ago)   5d
```

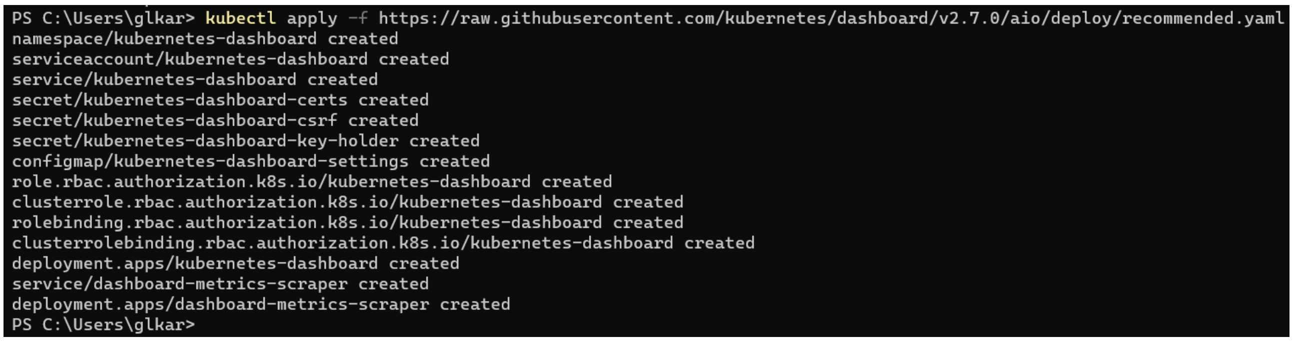
- I made sure to check the docker image. Welcome_image is the one being used in this scenario



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

welcome_image                                             latest
          1025b7cb5ca6   5 days ago      917MB
gcr.io/sit737-23t1-gujral-6790fc3/welcome_image           latest
          1025b7cb5ca6   5 days ago      917MB
```

- Now a crucial step, deploying the dashboard ui (Source - https://kubernetes.io/docs/tasks/access-application-cluster/web-ui-dashboard/)
  Since, the Dashboard UI is not deployed by default, I used the below command to deploy it –

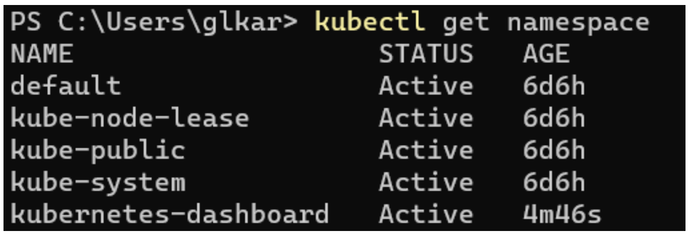
kubectl apply -f
https://raw.githubusercontent.com/kubernetes/dashboard/v2.7.0/aio/deploy/recommended.yaml



```
PS C:\Users\glkar> kubectl apply -f https://raw.githubusercontent.com/kubernetes/dashboard/v2.7.0/aio/deploy/recommended.yaml
namespace/kubernetes-dashboard created
serviceaccount/kubernetes-dashboard created
service/kubernetes-dashboard created
secret/kubernetes-dashboard-certs created
secret/kubernetes-dashboard-csrf created
secret/kubernetes-dashboard-key-holder created
configmap/kubernetes-dashboard-settings created
role.rbac.authorization.k8s.io/kubernetes-dashboard created
clusterrole.rbac.authorization.k8s.io/kubernetes-dashboard created
rolebinding.rbac.authorization.k8s.io/kubernetes-dashboard created
clusterrolebinding.rbac.authorization.k8s.io/kubernetes-dashboard created
deployment.apps/kubernetes-dashboard created
service/dashboard-metrics-scraper created
deployment.apps/dashboard-metrics-scraper created
PS C:\Users\glkar>
```

It created all the 14 objects that can be seen in the above screenshot. The yaml of the 14 objects can be seen at -
https://raw.githubusercontent.com/kubernetes/dashboard/v2.7.0/aio/deploy/recommended.yaml

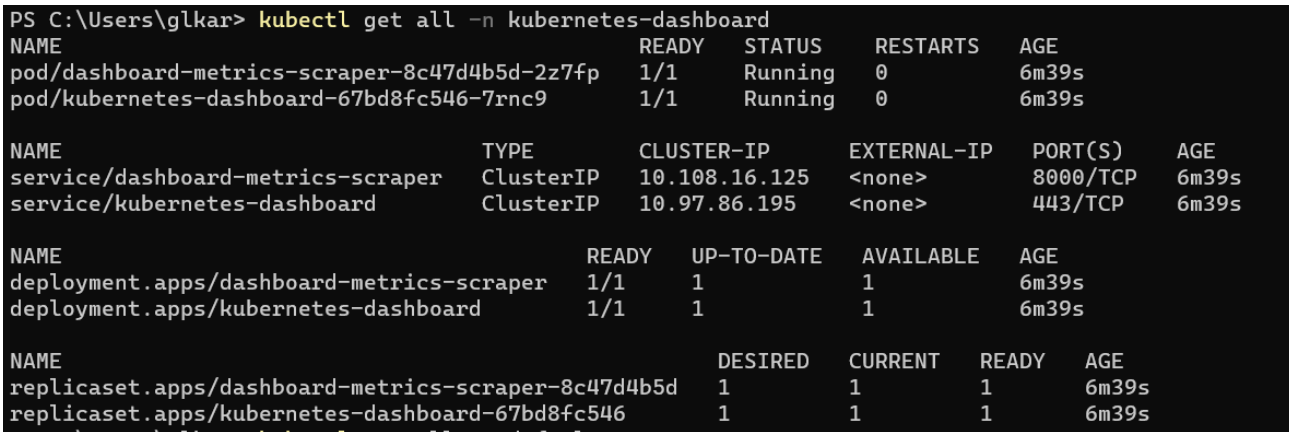
- Referring the above yaml, I came to know that the kubernetes-dashboard object is in the namespace Kubernetes-dashboard.
- An attempt to check all the namespaces -



```
PS C:\Users\glkar> kubectl get namespace
NAME                   STATUS   AGE
default                Active   6d6h
kube-node-lease        Active   6d6h
kube-public            Active   6d6h
kube-system            Active   6d6h
kubernetes-dashboard   Active   4m46s
```

- To check whether all the 14 objects required for the Kubernetes dashboard are in 'running' state, I checked kubectl with get all and specifying the namespace as Kubernetes-dashboard
  Command → kubectl get all -n Kubernetes-dashboard

  We can observe the deployment object and the service object here among others



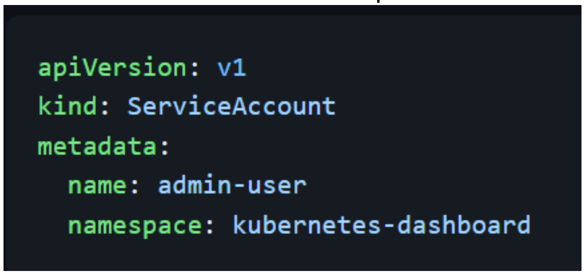
```
PS C:\Users\glkar> kubectl get all -n kubernetes-dashboard
NAME                                            READY   STATUS    RESTARTS   AGE
pod/dashboard-metrics-scraper-8c47d4b5d-2z7fp   1/1     Running   0          6m39s
pod/kubernetes-dashboard-67bd8fc546-7rnc9       1/1     Running   0          6m39s

NAME                                TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)    AGE
service/dashboard-metrics-scraper   ClusterIP   10.108.16.125   <none>        8000/TCP   6m39s
service/kubernetes-dashboard        ClusterIP   10.97.86.195    <none>        443/TCP    6m39s

NAME                                        READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/dashboard-metrics-scraper   1/1     1            1           6m39s
deployment.apps/kubernetes-dashboard        1/1     1            1           6m39s

NAME                                                  DESIRED   CURRENT   READY   AGE
replicaset.apps/dashboard-metrics-scraper-8c47d4b5d   1         1         1       6m39s
replicaset.apps/kubernetes-dashboard-67bd8fc546       1         1         1       6m39s
```

2. Creating a user to access dashboard UI

(Source: https://github.com/kubernetes/dashboard/blob/master/docs/user/access-control/creating-sample-user.md)

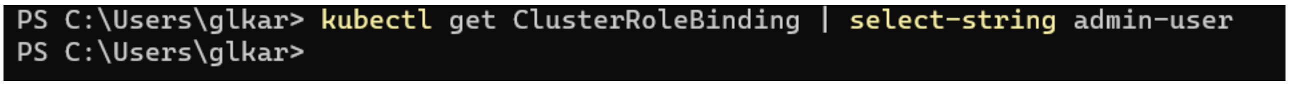
- The objective here was that in order to access the dashboard, I created a new user, gave it the kind 'ServiceAccount' and granted it admin permissions. This user will then need a bearer token tied to itself and then it can access the dashboard.

- Created Service Account in the code project as a yaml file service-account.yaml with the name admin-user in the namespace kubernetes-dashboard with the lines -



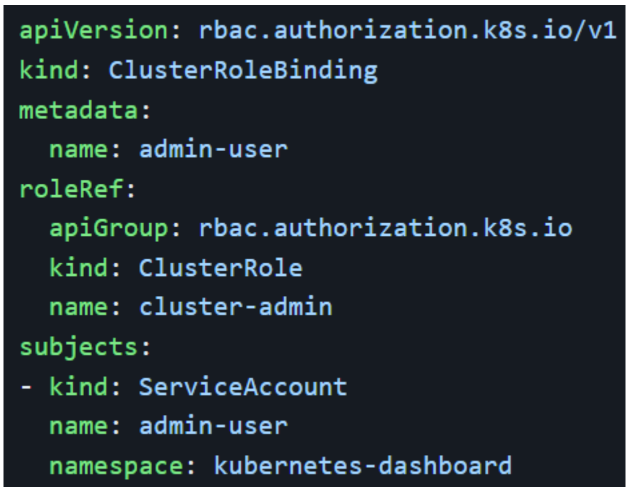
```
apiVersion: v1
kind: ServiceAccount
metadata:
  name: admin-user
  namespace: kubernetes-dashboard
```

- Then checked the ClusterRoleBinding for the ServiceAccount which did not exist as checked below -



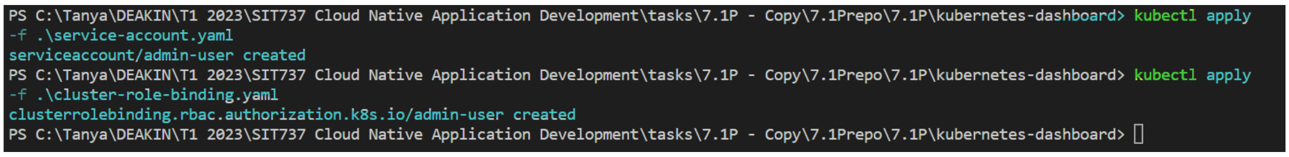
```
PS C:\Users\glkar> kubectl get ClusterRoleBinding | select-string admin-user
PS C:\Users\glkar>
```

- So, created the new ClusterRoleBinding admin-user under the project as a new yaml file cluster-role-binding.yaml and granted the privileges. Code -



```yaml
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: admin-user
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: cluster-admin
subjects:
- kind: ServiceAccount
  name: admin-user
  namespace: kubernetes-dashboard
```

- Now running kubectl apply on both the yaml files –



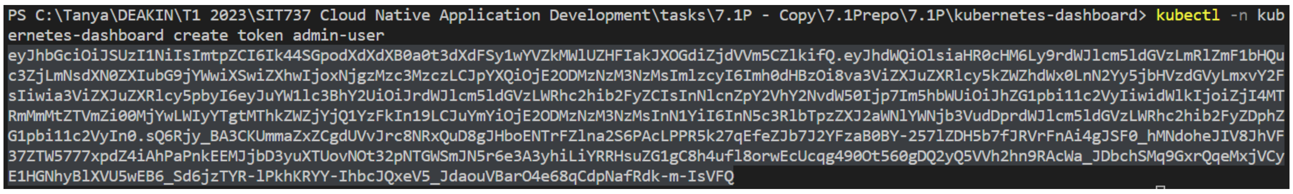
```
PS C:\Tanya\DEAKIN\T1 2023\SIT737 Cloud Native Application Development\tasks\7.1P - Copy\7.1Prepo\7.1P\kubernetes-dashboard> kubectl apply
-f .\service-account.yaml
serviceaccount/admin-user created
PS C:\Tanya\DEAKIN\T1 2023\SIT737 Cloud Native Application Development\tasks\7.1P - Copy\7.1Prepo\7.1P\kubernetes-dashboard> kubectl apply
-f .\cluster-role-binding.yaml
clusterrolebinding.rbac.authorization.k8s.io/admin-user created
PS C:\Tanya\DEAKIN\T1 2023\SIT737 Cloud Native Application Development\tasks\7.1P - Copy\7.1Prepo\7.1P\kubernetes-dashboard> ▯
```

3. Getting bearer token for admin-user

[Source : https://github.com/kubernetes/dashboard/blob/master/docs/user/access-control/creating-sample-user.md]

The bearer token needed for login is created as -

Command → kubectl -n kubernetes-dashboard create token admin-user
It takes into account that the admin-user is in namespace kubernetes-dashboard, indicated with -n namespace



```
PS C:\Tanya\DEAKIN\T1 2023\SIT737 Cloud Native Application Development\tasks\7.1P - Copy\7.1Prepo\7.1P\kubernetes-dashboard> kubectl -n kub
ernetes-dashboard create token admin-user
eyJhbGciOiJSUzI1NiIsImtpZCI6Ik44SGpodXdXdXB0a0t3dXdFSy1wYVZkMWlUZHFIakJXOGdiZjdVVm5CZlkifQ.eyJhdWQiOlsiaHR0cHM6Ly9rdWJlcm5ldGVzLmRlZmF1bHQu
c3ZjLmNsdXN0ZXIubG9jYWwiXSwiZXhwIjoxNjgzMzc3MzczLCJpYXQiOjE2ODMzNzM3NzMsImlzcyI6Imh0dHBzOi8va3ViZXJuZXRlcy5kZWZhdWx0LnN2Yy5jbHVzdGVyLmxvY2F
sIiwia3ViZXJuZXRlcy5pbyI6eyJuYW1lc3BhY2UiOiJrdWJlcm5ldGVzLWRhc2hib2FyZCIsInNlcnZpY2VhY2NvdW50Ijp7Im5hbWUiOiJhZG1pbi11c2VyIiwidWlkIjoiZjI4MT
RmMmMtZTVmZi00MjYwLWIyYTgtMThkZWZjYjQ1YzFkIn19LCJuYmYiOjE2ODMzNzM3NzMsInN1YiI6InN5c3RlbTpzZXJ2aWNlYWNjb3VudDprdWJlcm5ldGVzLWRhc2hib2FyZDphZ
G1pbi11c2VyIn0.sQ6Rjy_BA3CKUmmaZxZCgdUVvJrc8NRxQuD8gJHboENTrFZlna2S6PAcLPPR5k27qEfeZJb7J2YFzaB0BY-257lZDH5b7fJRVrFnAi4gJSF0_hMNdoheJIV8JhVF
37ZTW5777xpdZ4iAhPaPnkEEMJjbD3yuXTUovNOt32pNTGWSmJN5r6e3A3yhiLiYRRHsuZG1gC8h4ufl8orwEcUcqg490Ot560gDQ2yQ5VVh2hn9RAcWa_JDbchSMq9GxrQqeMxjVCy
E1HGNhyBlXVU5wEB6_Sd6jzTYR-lPkhKRYY-IhbcJQxeV5_JdaouVBarO4e68qCdpNafRdk-m-IsVFQ
```

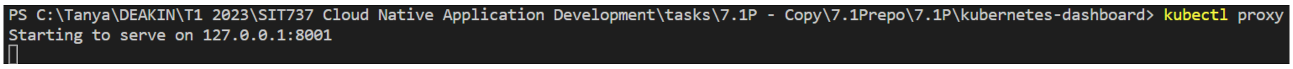
- In my case, the token was –

eyJhbGciOiJSUzI1NiIsImtpZCI6Ik44SGpodXdXdXB0a0t3dXdFSy1wYVZkMWlUZHFIakJXOGdiZjdVVm5
CZlkifQ.eyJhdWQiOlsiaHR0cHM6Ly9rdWJlcm5ldGVzLmRlZmF1bHQuc3ZjLmNsdXN0ZXIubG9jYWwiX
SwiZXhwIjoxNjgzMzc3MzczLCJpYXQiOjE2ODMzNzM3NzMsImlzcyI6Imh0dHBzOi8va3ViZXJuZXRlcy5k
ZWZhdWx0LnN2Yy5jbHVzdGVyLmxvY2FsIiwia3ViZXJuZXRlcy5pbyI6eyJuYW1lc3BhY2UiOiJrdWJlcm5l
dGVzLWRhc2hib2FyZCIsInNlcnZpY2VhY2NvdW50Ijp7Im5hbWUiOiJhZG1pbi11c2VyIiwidWlkIjoiZjI4
MTRmMmMtZTVmZi00MjYwLWIyYTgtMThkZWZjYjQ1YzFkIn19LCJuYmYiOjE2ODMzNzM3NzMsInN1
YiI6InN5c3RlbTpzZXJ2aWNlYWNjb3VudDprdWJlcm5ldGVzLWRhc2hib2FyZDphZG1pbi11c2VyIn0.sQ
6Rjy_BA3CKUmmaZxZCgdUVvJrc8NRxQuD8gJHboENTrFZlna2S6PAcLPPR5k27qEfeZJb7J2YFzaB0BY-
257lZDH5b7fJRVrFnAi4gJSF0_hMNdoheJIV8JhVF37ZTW5777xpdZ4iAhPaPnkEEMJjbD3yuXTUovNOt
32pNTGWSmJN5r6e3A3yhiLiYRRHsuZG1gC8h4ufl8orwEcUcqg490Ot560gDQ2yQ5VVh2hn9RAcWa_J

DbchSMq9GxrQqeMxjVCyE1HGNhyBlXVU5wEB6_Sd6jzTYR-lPkhKRYY-IhbcJQxeV5_JdaouVBarO4e68qCdpNafRdk-m-IsVFQ

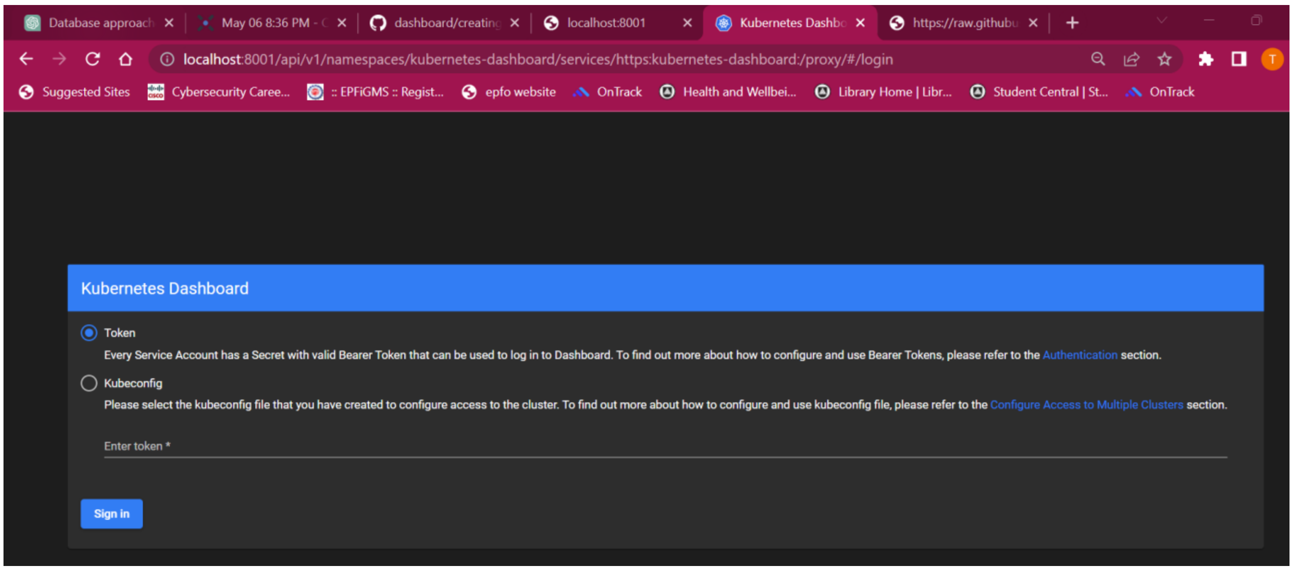
4. Getting access to the Dashboard

- Command needed was – kubectl proxy to enable access to the dashboard



```
PS C:\Tanya\DEAKIN\T1 2023\SIT737 Cloud Native Application Development\tasks\7.1P - Copy\7.1Prepo\7.1P\kubernetes-dashboard> kubectl proxy
Starting to serve on 127.0.0.1:8001
```

- Dashboard url was hit - http://localhost:8001/api/v1/namespaces/kubernetes-dashboard/services/https:kubernetes-dashboard:/proxy/#/log/default/welcomemicroservice-8fcc9cc-4b8c8/pod?namespace=default&container=welcomemicroservice



localhost:8001/api/v1/namespaces/kubernetes-dashboard/services/https:kubernetes-dashboard:/proxy/#/login

Kubernetes Dashboard

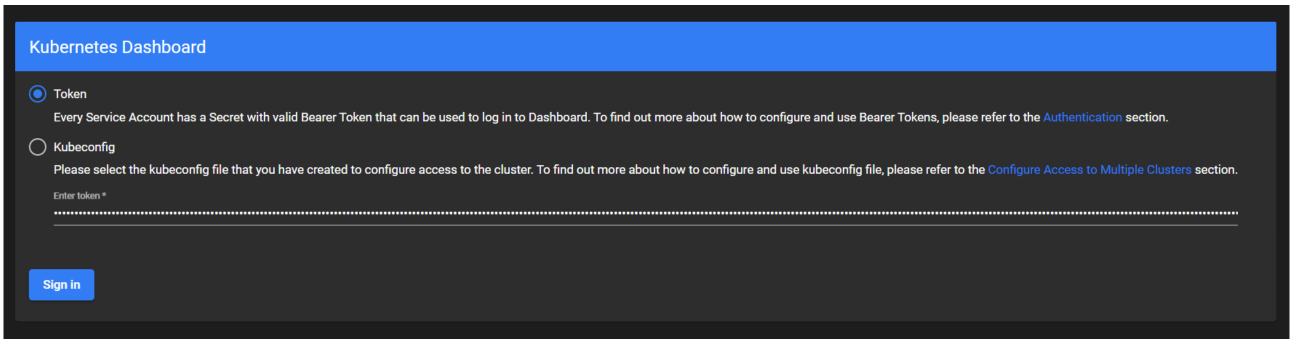
Token
Every Service Account has a Secret with valid Bearer Token that can be used to log in to Dashboard. To find out more about how to configure and use Bearer Tokens, please refer to the Authentication section.

Kubeconfig
Please select the kubeconfig file that you have created to configure access to the cluster. To find out more about how to configure and use kubeconfig file, please refer to the Configure Access to Multiple Clusters section.

Enter token *

Sign in

- Token was pasted and sign in was clicked -



Kubernetes Dashboard

Token
Every Service Account has a Secret with valid Bearer Token that can be used to log in to Dashboard. To find out more about how to configure and use Bearer Tokens, please refer to the Authentication section.

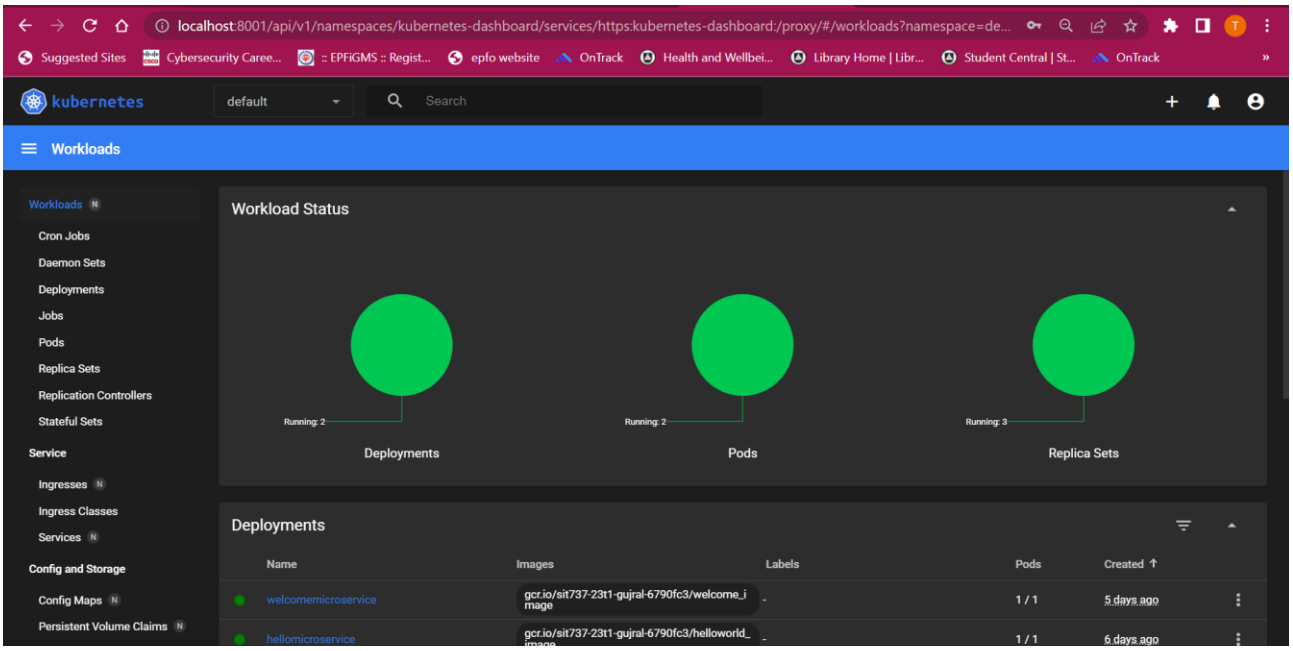
Kubeconfig
Please select the kubeconfig file that you have created to configure access to the cluster. To find out more about how to configure and use kubeconfig file, please refer to the Configure Access to Multiple Clusters section.

Enter token *
••

Sign in

- Below observation about the dashboard -

Workloads page of the dashboard showing the ‘welcomeservice’


localhost:8001/api/v1/namespaces/kubernetes-dashboard/services/https:kubernetes-dashboard:/proxy/#/workloads?namespace=de...

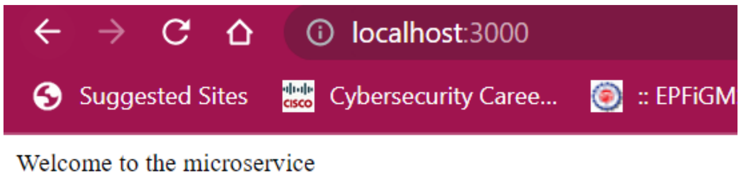
Workloads

Workload Status

Deployments

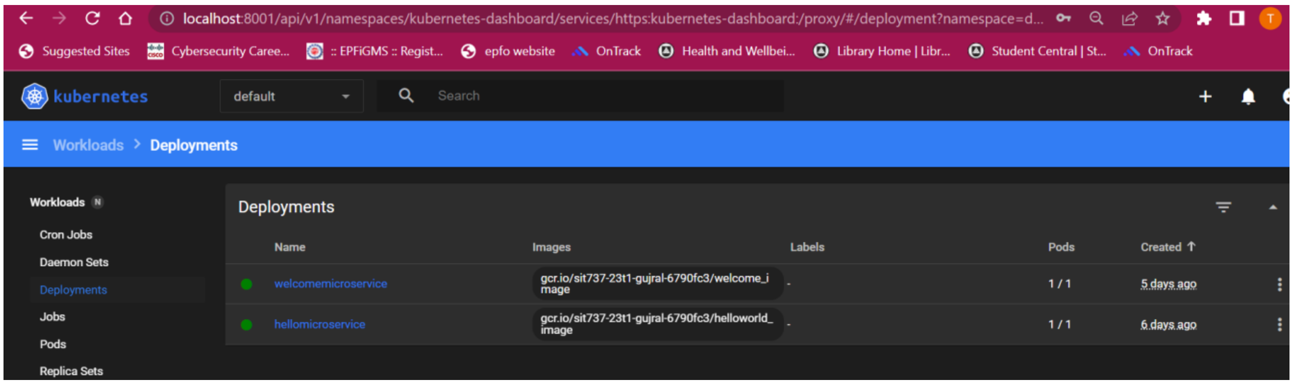
Name	Images	Labels	Pods	Created ↑
welcomemicroservice	gcr.io/sit737-23t1-gujral-6790fc3/welcome_image	-	1 / 1	5 days ago
hellomicroservice	gcr.io/sit737-23t1-gujral-6790fc3/helloworld_image	-	1 / 1	6 days ago

Launched the application / a running view of the old container. The message 'Welcome to the microservice' can be seen –


localhost:3000

Welcome to the microservice

- The 'Deployments' view on the Dashboard shows the welcomemicroservice again -

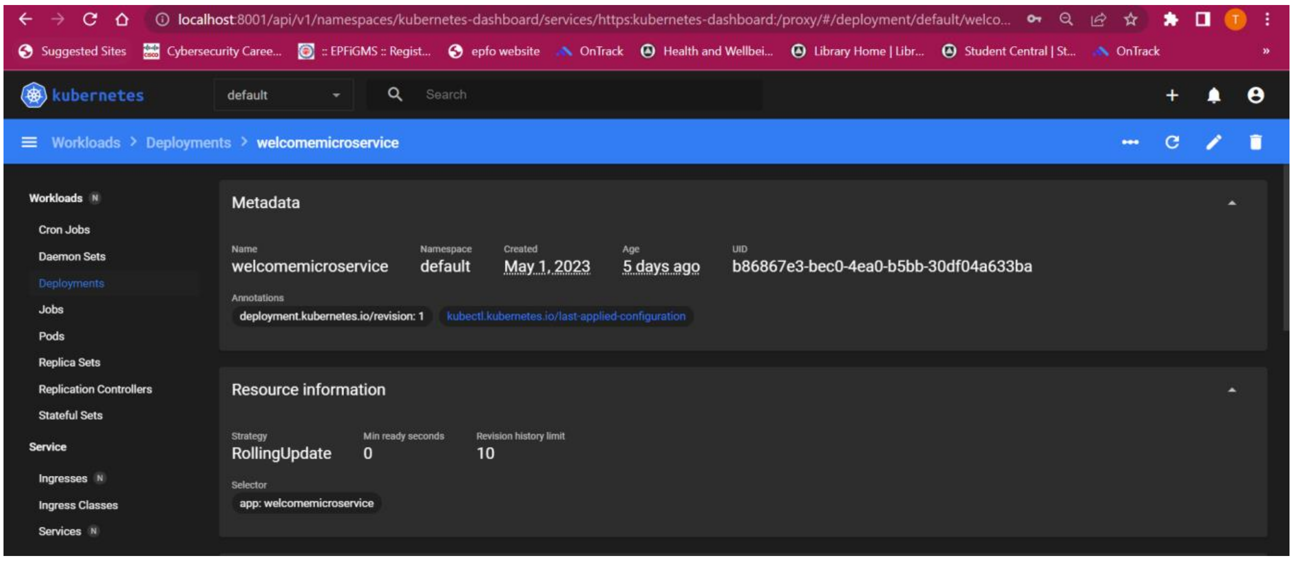

localhost:8001/api/v1/namespaces/kubernetes-dashboard/services/https:kubernetes-dashboard:/proxy/#/deployment?namespace=d...

Workloads > Deployments

Deployments

Name	Images	Labels	Pods	Created ↑
welcomemicroservice	gcr.io/sit737-23t1-gujral-6790fc3/welcome_image	-	1 / 1	5 days ago
hellomicroservice	gcr.io/sit737-23t1-gujral-6790fc3/helloworld_image	-	1 / 1	6 days ago

-Under Deployments > welcomemicroservice shows the following -



localhost:8001/api/v1/namespaces/kubernetes-dashboard/services/https:kubernetes-dashboard:/proxy/#/deployment/default/welco...

Suggested Sites | Cybersecurity Caree... | :: EPFiGMS :: Regist... | epfo website | OnTrack | Health and Wellbei... | Library Home | Libr... | Student Central | St... | OnTrack

kubernetes | default | Search

Workloads > Deployments > welcomemicroservice

Workloads N
Cron Jobs
Daemon Sets
Deployments
Jobs
Pods
Replica Sets
Replication Controllers
Stateful Sets
Service
Ingresses N
Ingress Classes
Services N

Metadata

Name	Namespace	Created	Age	UID
welcomemicroservice	default	May 1, 2023	5 days ago	b86867e3-bec0-4ea0-b5bb-30df04a633ba

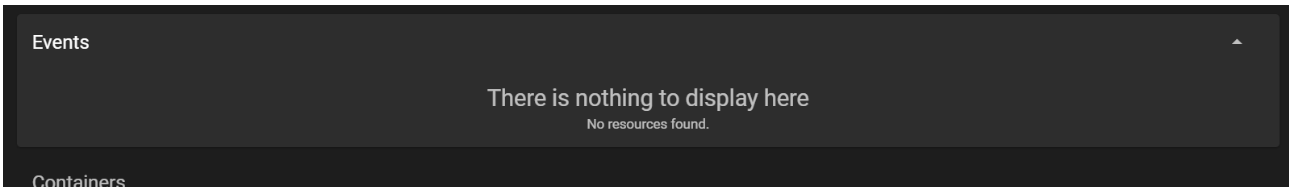
Annotations
deployment.kubernetes.io/revision: 1 | kubectl.kubernetes.io/last-applied-configuration

Resource information

Strategy	Min ready seconds	Revision history limit
RollingUpdate	0	10

Selector
app: welcomemicroservice

It shows no events. (This will change when I deploy the new image) -

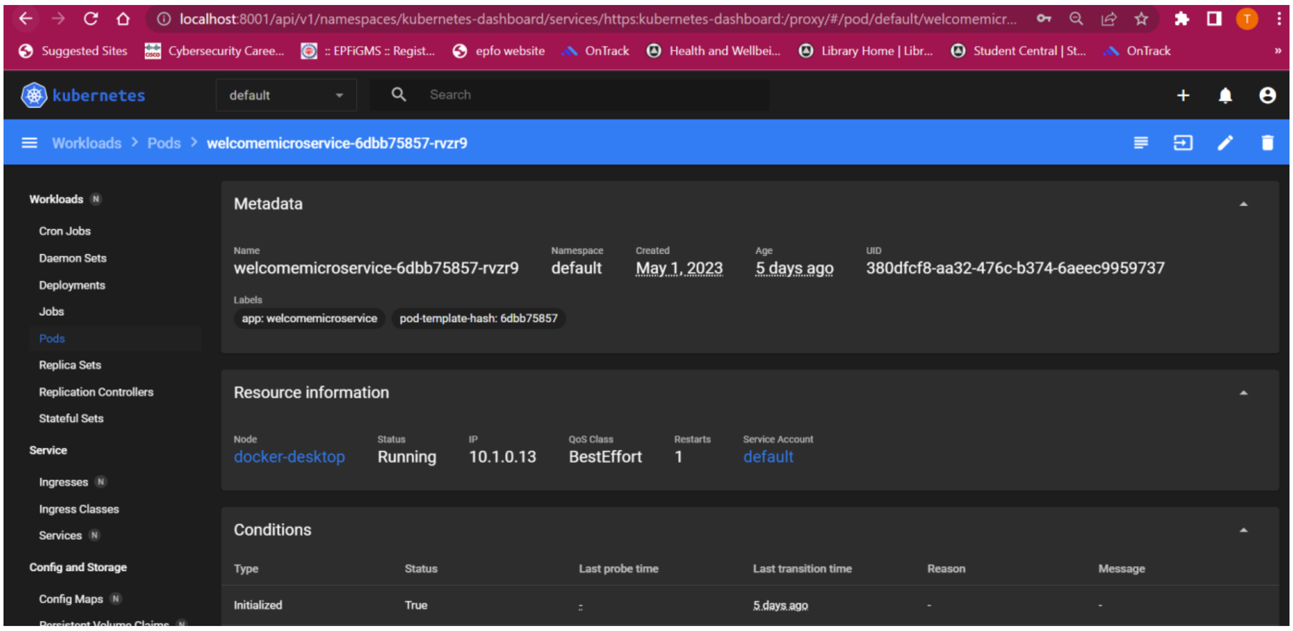


Events

There is nothing to display here
No resources found.

Containers

- The 'Pods' view shows the pod for the welcomemicroservice as below -



localhost:8001/api/v1/namespaces/kubernetes-dashboard/services/https:kubernetes-dashboard:/proxy/#/pod/default/welcomemicr...

Suggested Sites | Cybersecurity Caree... | :: EPFiGMS :: Regist... | epfo website | OnTrack | Health and Wellbei... | Library Home | Libr... | Student Central | St... | OnTrack

kubernetes | default | Search

Workloads > Pods > welcomemicroservice-6dbb75857-rvzr9

Workloads N
Cron Jobs
Daemon Sets
Deployments
Jobs
Pods
Replica Sets
Replication Controllers
Stateful Sets
Service
Ingresses N
Ingress Classes
Services N
Config and Storage
Config Maps N

Metadata

Name	Namespace	Created	Age	UID
welcomemicroservice-6dbb75857-rvzr9	default	May 1, 2023	5 days ago	380dfcf8-aa32-476c-b374-6aeec9959737

Labels
app: welcomemicroservice | pod-template-hash: 6dbb75857

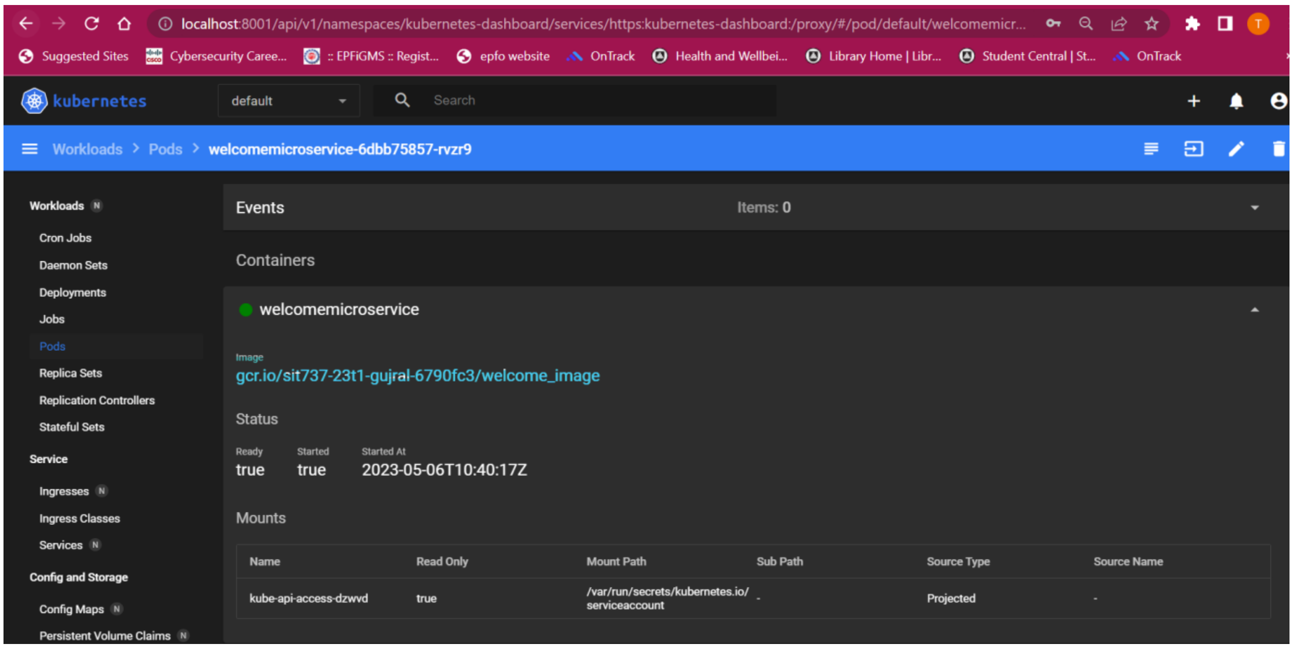
Resource information

Node	Status	IP	QoS Class	Restarts	Service Account
docker-desktop	Running	10.1.0.13	BestEffort	1	default

Conditions

Type	Status	Last probe time	Last transition time	Reason	Message
Initialized	True	-	5 days ago	-	-

- It show the old image which was on GCR -


localhost:8001/api/v1/namespaces/kubernetes-dashboard/services/https:kubernetes-dashboard:/proxy/#/pod/default/welcomemicr...

Suggested Sites | Cybersecurity Caree... | :: EPFiGMS :: Regist... | epfo website | OnTrack | Health and Wellbei... | Library Home | Libr... | Student Central | St... | OnTrack

kubernetes | default | Search

Workloads > Pods > welcomemicroservice-6dbb75857-rvzr9

Events — Items: 0

Containers

welcomemicroservice

Image
gcr.io/sit737-23t1-gujral-6790fc3/welcome_image

Status

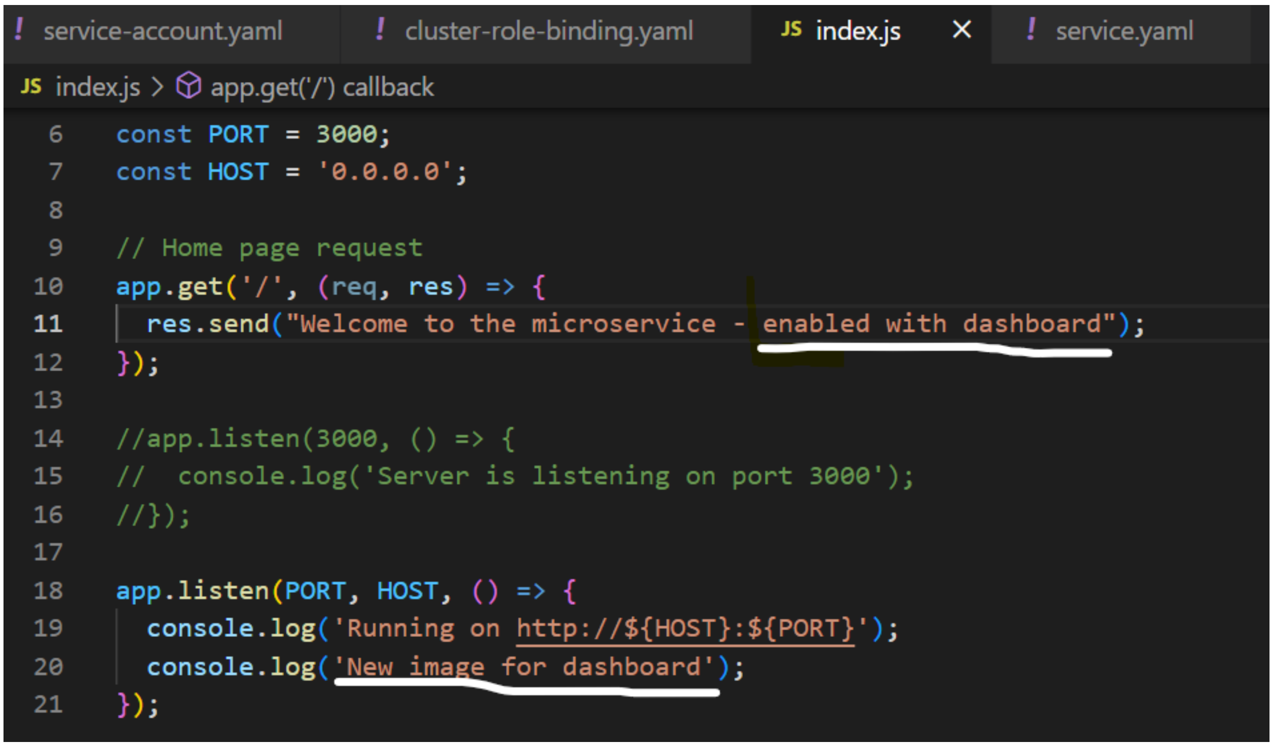
Ready	Started	Started At
true	true	2023-05-06T10:40:17Z

Mounts

Name	Read Only	Mount Path	Sub Path	Source Type	Source Name
kube-api-access-dzwvd	true	/var/run/secrets/kubernetes.io/serviceaccount	-	Projected	-

5. Created new image –

As required in the task to create new image, I changed the code in index.js with two altered string messages highlighted as below -



```
const PORT = 3000;
const HOST = '0.0.0.0';

// Home page request
app.get('/', (req, res) => {
  res.send("Welcome to the microservice - enabled with dashboard");
});

//app.listen(3000, () => {
//  console.log('Server is listening on port 3000');
//});

app.listen(PORT, HOST, () => {
  console.log('Running on http://${HOST}:${PORT}');
  console.log('New image for dashboard');
});
```

Res.send is updated to give the message “Welcome to the microservice - **enabled with dashboard**”

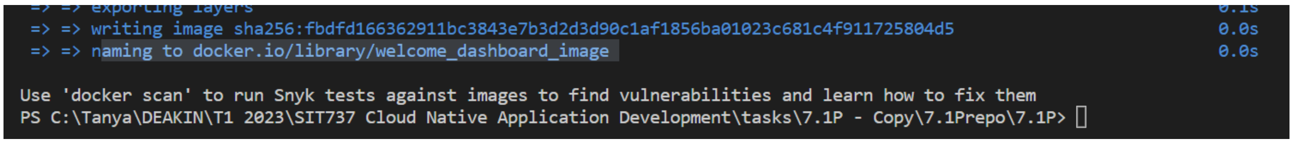
And

Console.log has an additional message “**New image for dashboard** ”

We will notice these on the dashboard soon.

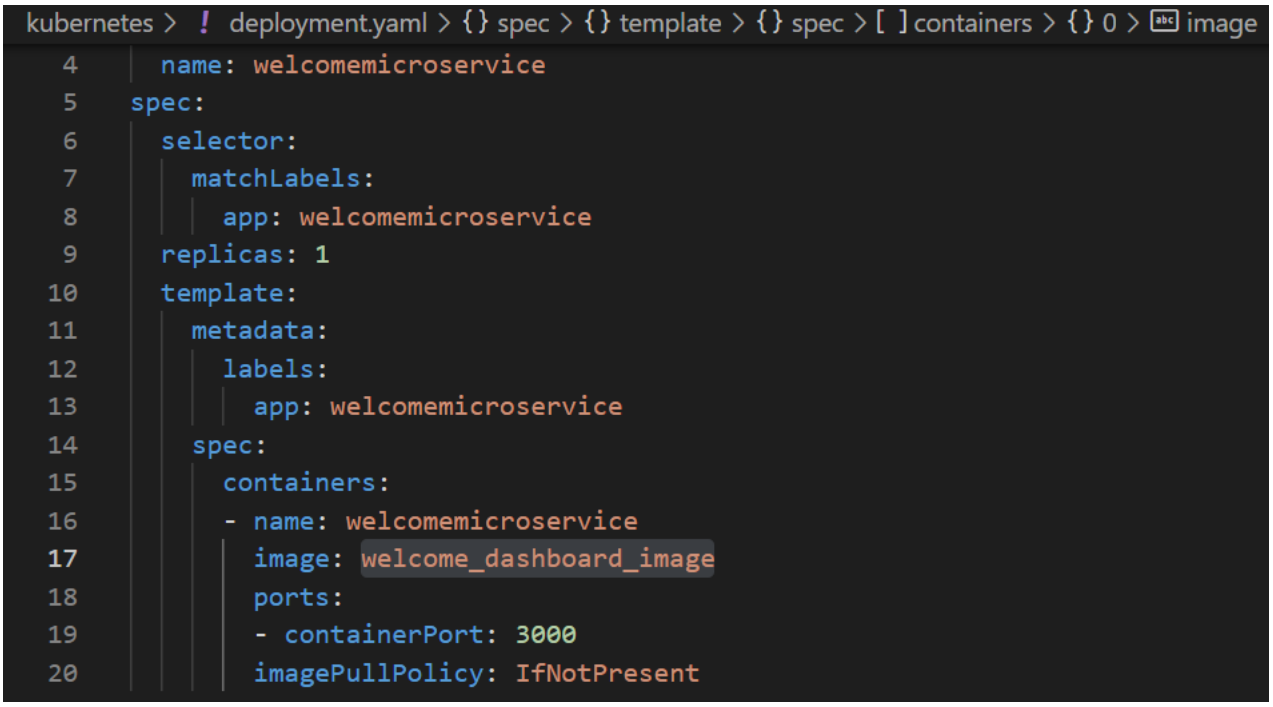
- Gave the new image name as - welcome_dashboard_image and built it -

  Command → docker build -t welcome_dashboard_image .



```
 => => writing image sha256:fbdfd166362911bc3843e7b3d2d3d90c1af1856ba01023c681c4f911725804d5    0.0s
 => => naming to docker.io/library/welcome_dashboard_image                                       0.0s

Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them
PS C:\Tanya\DEAKIN\T1 2023\SIT737 Cloud Native Application Development\tasks\7.1P - Copy\7.1Prepo\7.1P>
```

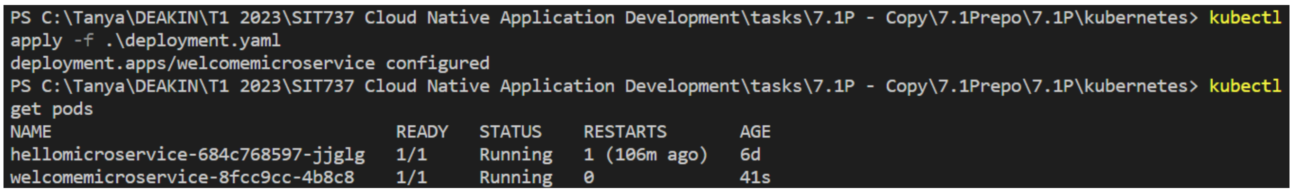
- Changed the deployment configuration file to contain the new image name as 'welcome_dashboard_image'
  For this, in the project, Edited Kubernetes> deployment.yaml configuration file with the new image as below -



kubernetes > ! deployment.yaml > {} spec > {} template > {} spec > [ ] containers > {} 0 > image

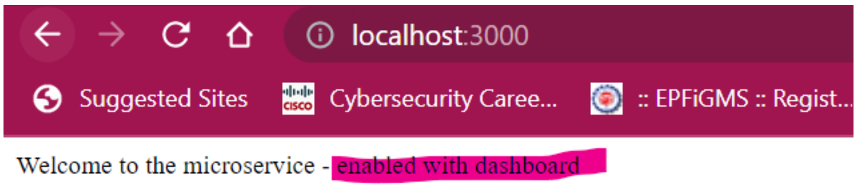
```
  name: welcomemicroservice
spec:
  selector:
    matchLabels:
      app: welcomemicroservice
  replicas: 1
  template:
    metadata:
      labels:
        app: welcomemicroservice
    spec:
      containers:
      - name: welcomemicroservice
        image: welcome_dashboard_image
        ports:
        - containerPort: 3000
        imagePullPolicy: IfNotPresent
```

- Then did kubectl apply with the altered deployment configuration file. The kubectl get pods returned the new one marked 41 seconds ago -



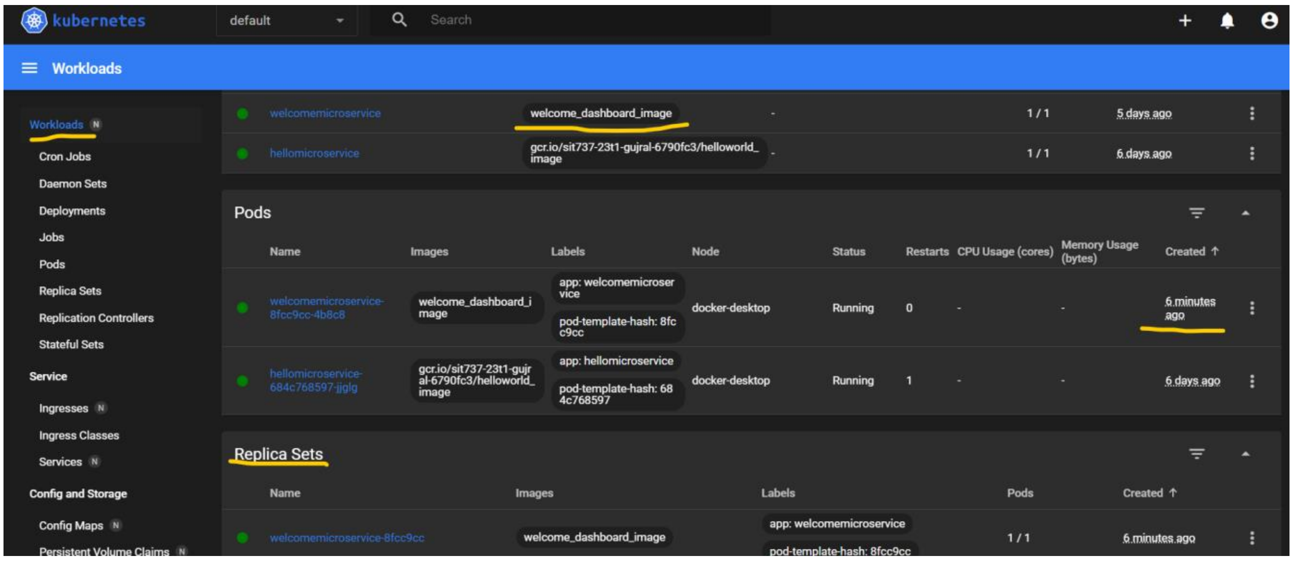
```
PS C:\Tanya\DEAKIN\T1 2023\SIT737 Cloud Native Application Development\tasks\7.1P - Copy\7.1Prepo\7.1P\kubernetes> kubectl
apply -f .\deployment.yaml
deployment.apps/welcomemicroservice configured
PS C:\Tanya\DEAKIN\T1 2023\SIT737 Cloud Native Application Development\tasks\7.1P - Copy\7.1Prepo\7.1P\kubernetes> kubectl
get pods
NAME                                  READY   STATUS    RESTARTS        AGE
hellomicroservice-684c768597-jjglg    1/1     Running   1 (106m ago)    6d
welcomemicroservice-8fcc9cc-4b8c8     1/1     Running   0               41s
```

- Hit the browser to see the changed message as below -



Welcome to the microservice - enabled with dashboard

- Launched the dashboard again at - http://localhost:8001/api/v1/namespaces/kubernetes-dashboard/services/https:kubernetes-dashboard:/proxy/#/log/default/welcomemicroservice-8fcc9cc-4b8c8/pod?namespace=default&container=welcomemicroservice
- The changes can be seen in all the layouts – Workloads, Deployment, Pods, Logs of the pod as highlighted in the below screenshots –
- Workloads show the new image name ‘welcome_dashboard_image’ -



welcomemicroservice	welcome_dashboard_image	-	1 / 1	5 days ago
hellomicroservice	gcr.io/sit737-23t1-gujral-6790fc3/helloworld_image	-	1 / 1	6 days ago

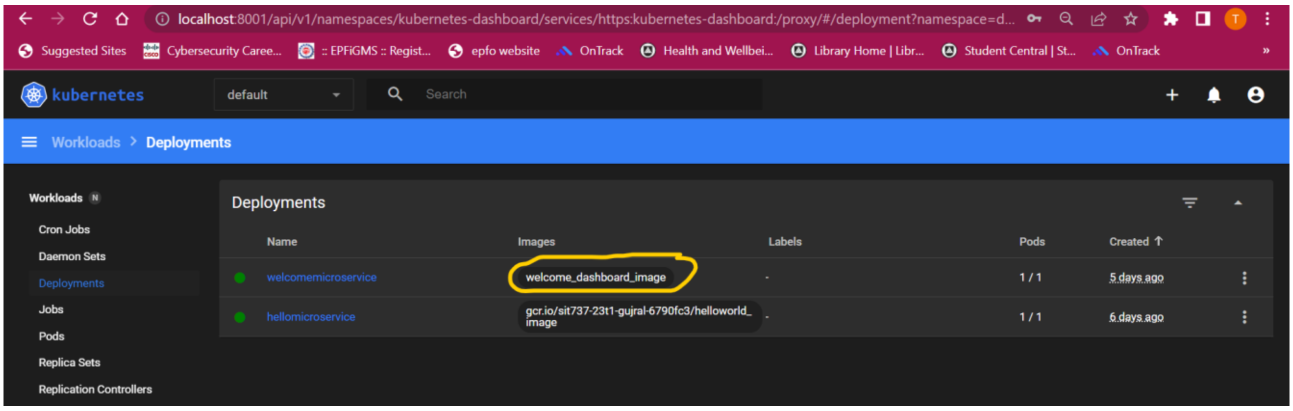
Pods

Name	Images	Labels	Node	Status	Restarts	CPU Usage (cores)	Memory Usage (bytes)	Created ↑
welcomemicroservice-8fcc9cc-4b8c8	welcome_dashboard_image	app: welcomemicroservice pod-template-hash: 8fcc9cc	docker-desktop	Running	0	-	-	6 minutes ago
hellomicroservice-684c768597-jjglg	gcr.io/sit737-23t1-gujral-6790fc3/helloworld_image	app: hellomicroservice pod-template-hash: 684c768597	docker-desktop	Running	1	-	-	6 days ago

Replica Sets

Name	Images	Labels	Pods	Created ↑
welcomemicroservice-8fcc9cc	welcome_dashboard_image	app: welcomemicroservice pod-template-hash: 8fcc9cc	1 / 1	6 minutes ago

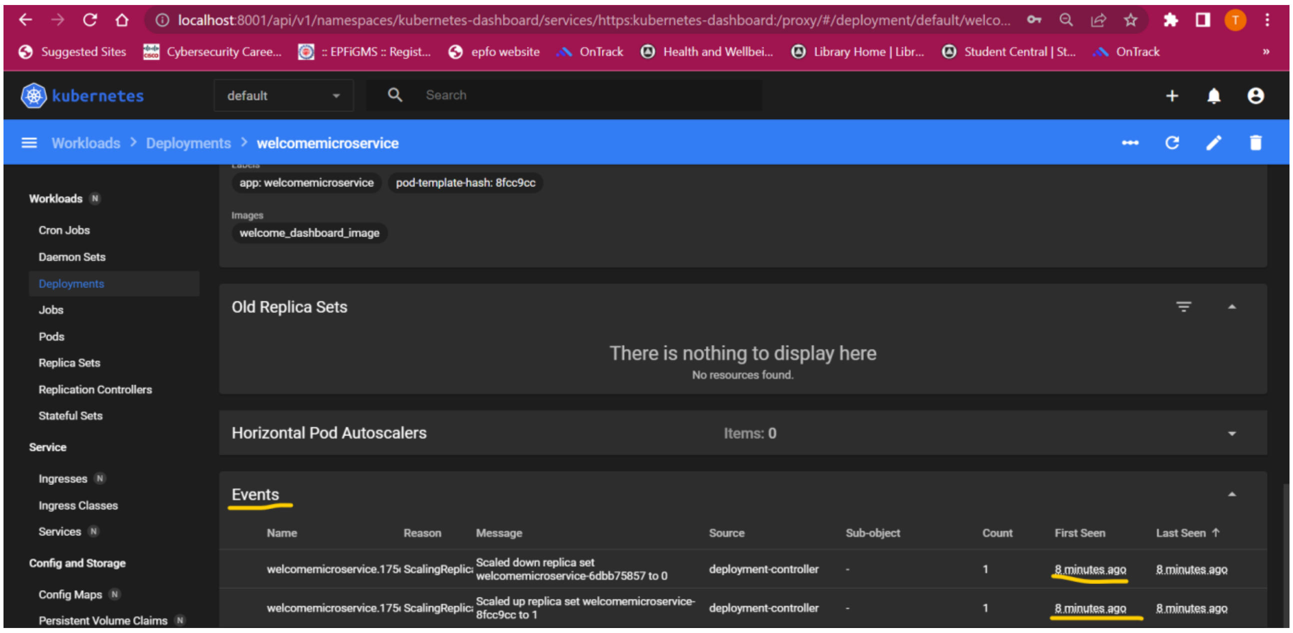
- Deployments shows the new image name -



Deployments

Name	Images	Labels	Pods	Created ↑
welcomemicroservice	welcome_dashboard_image	-	1 / 1	5 days ago
hellomicroservice	gcr.io/sit737-23t1-gujral-6790fc3/helloworld_image	-	1 / 1	6 days ago

- Events can be seen updated as scaled down replica set and scaled up replica set -



app: welcomemicroservice   pod-template-hash: 8fcc9cc

Images
welcome_dashboard_image

Old Replica Sets

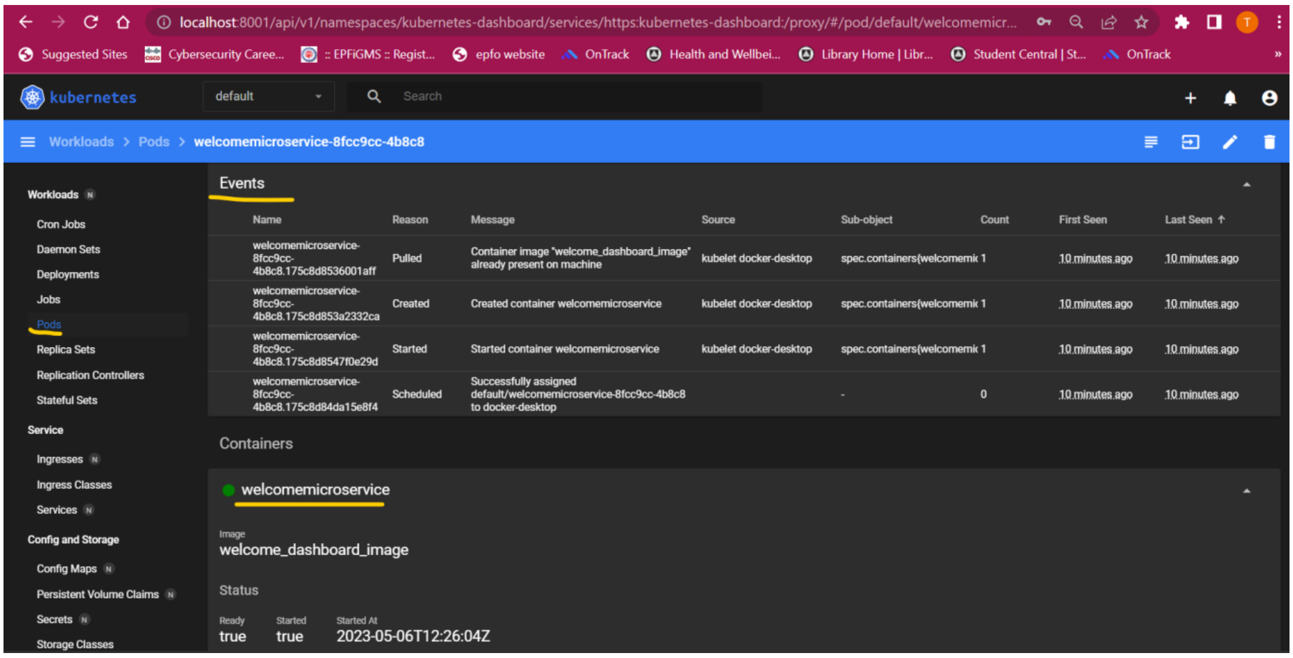
There is nothing to display here
No resources found.

Horizontal Pod Autoscalers   Items: 0

Events

Name	Reason	Message	Source	Sub-object	Count	First Seen	Last Seen ↑
welcomemicroservice.175	ScalingReplic	Scaled down replica set welcomemicroservice-6dbb75857 to 0	deployment-controller	-	1	8 minutes ago	8 minutes ago
welcomemicroservice.175	ScalingReplic	Scaled up replica set welcomemicroservice-8fcc9cc to 1	deployment-controller	-	1	8 minutes ago	8 minutes ago

- Pods events also reflect the change of the image from old to new -



Events

Name	Reason	Message	Source	Sub-object	Count	First Seen	Last Seen ↑
welcomemicroservice-8fcc9cc-4b8c8.175c8d8536001aff	Pulled	Container image "welcome_dashboard_image" already present on machine	kubelet docker-desktop	spec.containers{welcomemic	1	10 minutes ago	10 minutes ago
welcomemicroservice-8fcc9cc-4b8c8.175c8d853a2332ca	Created	Created container welcomemicroservice	kubelet docker-desktop	spec.containers{welcomemic	1	10 minutes ago	10 minutes ago
welcomemicroservice-8fcc9cc-4b8c8.175c8d8547f0e29d	Started	Started container welcomemicroservice	kubelet docker-desktop	spec.containers{welcomemic	1	10 minutes ago	10 minutes ago
welcomemicroservice-8fcc9cc-4b8c8.175c8d84da15e8f4	Scheduled	Successfully assigned default/welcomemicroservice-8fcc9cc-4b8c8 to docker-desktop		-	0	10 minutes ago	10 minutes ago

Containers

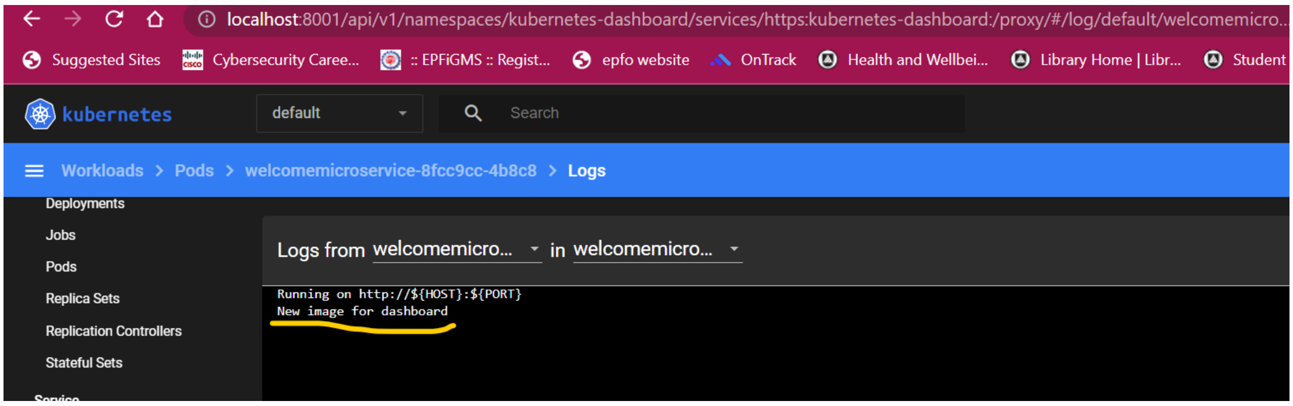
welcomemicroservice

Image
welcome_dashboard_image

Status

Ready	Started	Started At
true	true	2023-05-06T12:26:04Z

- New log for the new pod records the console.log message added ‘New image for dashboard’ –


localhost:8001/api/v1/namespaces/kubernetes-dashboard/services/https:kubernetes-dashboard:/proxy/#/log/default/welcomemicro...
Suggested Sites
Cybersecurity Caree...
:: EPFiGMS :: Regist...
epfo website
OnTrack
Health and Wellbei...
Library Home | Libr...
Student
kubernetes
default
Search
Workloads > Pods > welcomemicroservice-8fcc9cc-4b8c8 > Logs
Deployments
Jobs
Pods
Replica Sets
Replication Controllers
Stateful Sets
Logs from welcomemicro... in welcomemicro...

```
Running on http://${HOST}:${PORT}
New image for dashboard
```