

SIT323 – Task 7.1P

Once Kubernetes was installed (by enabling it on Docker Desktop), the Dashboard UI was deployed:

```
D:\DeakinUniversity\SIT323\sit323_737-2023-t1-prac7p>kubectl apply -f https://raw.githubusercontent.com/kubernetes/dashboard/v2.7.0/aio/deploy/recommended.yaml
namespace/kubernetes-dashboard unchanged
serviceaccount/kubernetes-dashboard unchanged
service/kubernetes-dashboard unchanged
secret/kubernetes-dashboard-certs unchanged
secret/kubernetes-dashboard-csrf configured
Warning: resource secrets/kubernetes-dashboard-key-holder is missing the kubectl.kubernetes.io/last-applied-configuration annotation which is required by kubectl apply. k
ubectl apply should only be used on resources created declaratively by either kubectl create --save-config or kubectl apply. The missing annotation will be patched automa
tically.
secret/kubernetes-dashboard-key-holder configured
configmap/kubernetes-dashboard-settings unchanged
role.rbac.authorization.k8s.io/kubernetes-dashboard unchanged
clusterrole.rbac.authorization.k8s.io/kubernetes-dashboard unchanged
rolebinding.rbac.authorization.k8s.io/kubernetes-dashboard unchanged
clusterrolebinding.rbac.authorization.k8s.io/kubernetes-dashboard unchanged
deployment.apps/kubernetes-dashboard unchanged
service/dashboard-metrics-scraper unchanged
deployment.apps/dashboard-metrics-scraper unchanged
```

Then, a Service Account was created, using two separate YAML files:

```
! dashboard-adminuser.yaml
1  apiVersion: v1
2  kind: ServiceAccount
3  metadata:
4    name: admin-user
5    namespace: kubernetes-dashboard
```

```
! cluster_role_binding.yaml
1  apiVersion: rbac.authorization.k8s.io/v1
2  kind: ClusterRoleBinding
3  metadata:
4    name: admin-user
5  roleRef:
6    apiGroup: rbac.authorization.k8s.io
7    kind: ClusterRole
8    name: cluster-admin
9  subjects:
10 - kind: ServiceAccount
11   name: admin-user
12   namespace: kubernetes-dashboard
```

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19044.2846]
(c) Microsoft Corporation. All rights reserved.

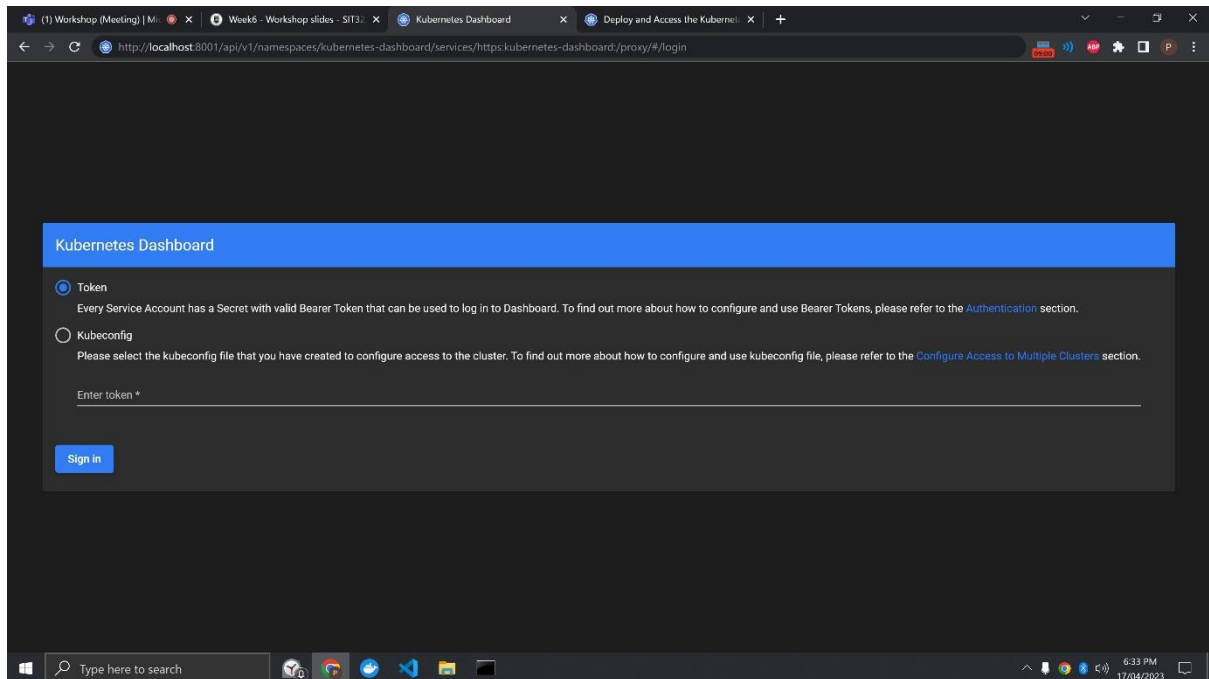
D:\DeakinUniversity\SIT323\sit323_737-2023-t1-prac7p>kubectl apply -f dashboard-adminuser.yaml
serviceaccount/admin-user created

D:\DeakinUniversity\SIT323\sit323_737-2023-t1-prac7p>kubectl apply -f cluster_role_binding.yaml
clusterrolebinding.rbac.authorization.k8s.io/admin-user created
```

Dashboard was then hosted on localhost, using kubectl proxy:

```
D:\DeakinUniversity\SIT323\sit323_737-2023-t1-prac7p>kubectl proxy
Starting to serve on 127.0.0.1:8001
```

Dashboard was visited, at which point a token was required:



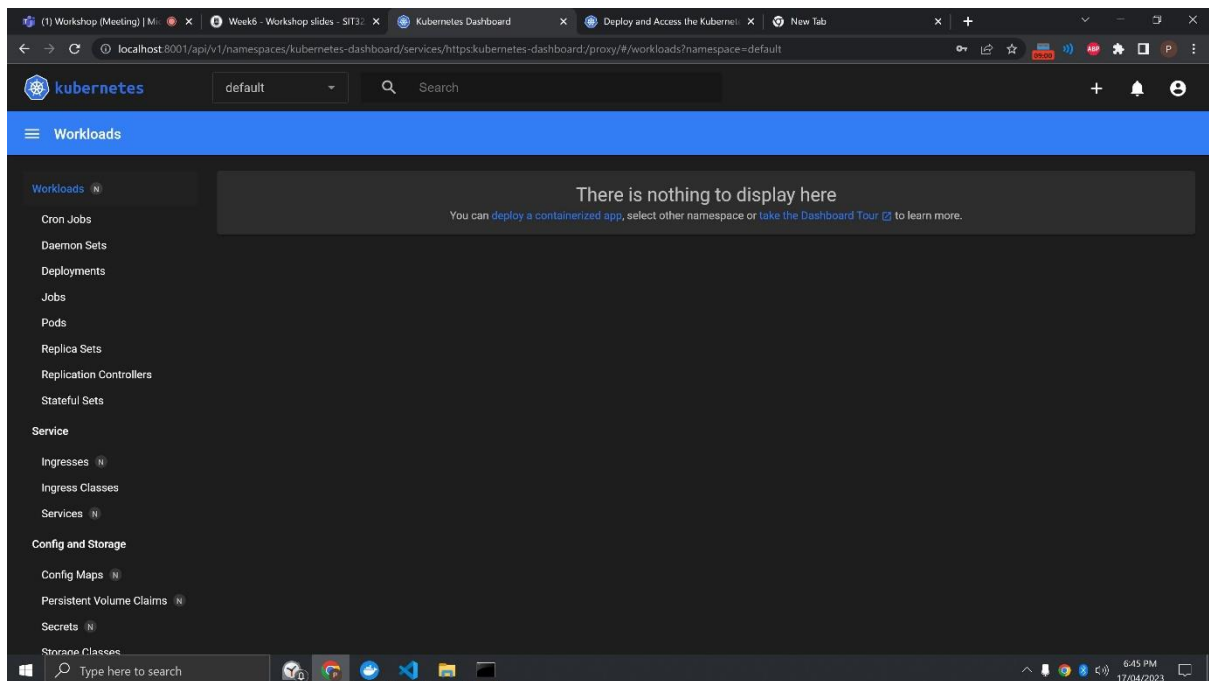
Token was created:

```

p2-beakn1university51t32p33g1t323_737_2021_1-prac7-qbuksbel_1_kubermetes_dashboard_create_token_admin user
y2hbk610135u211w1st7c61gpk1g1t323hbk1t291q7hMQG01NXh5b0e1ZhoYf1c5u10kUp1uY1Yn1fTfW81f0.eyJhdW01c3JlMmVsdXN0ZXBubG9jYmV1X
y2h2Xhwl3ojXjgX2k1BfZc21c3pYX0101gY20DE3M1cYEMXD51mnc1zyl61mhdhB6108rva3v12X1ZuX1c5kZVhZhdw0k1n2Yv5jhbVzdgV5yLmWZv5f1w1a3V1ZuXJR1c3pYb1gY201c3BHV2101jrdw1c3
m51dGz4UWRhZ2h1BfZc21c3Zc1n1c2pYhZV1Nvdw50157p1m5hB0w101g1Zhg61p11c1Y11w1dW1k1J1c3U1N4Y4T1n2Yv5Zc3S0B9Z0K1dW1KfMDE2t1RnMwU3j1c3B2Y1n1c3L1n1c3L1c3U1mY1J1c3EMXD51n1Y1
1161n5C3r1b1p1Zc21a2w1M1c3pYb1Vudp1dpr1m1c51dGz4UWRhZ2h1BfZc21pZdph3Zg61h1c1Y2v1n1.NRk81ANcm1jG6hZnA2eXa2daC17F1dNkVMMwYw4S0J6f--gq-389hT1LW0f5h36D01hAXg--8-FHFFt76
30Se8B0vCrm1m3oBg1AEPV13P--McM69h01L1TevTm1Cp5aCvNbDk106L10f1SPR4Q0zGomX0R1L1K981J3CkUd6g3hCPVAA3H81N1Bwq71EYgYSW1BrL9X1j5FMHcYt5epZapT8A0Wf8uBmq0P7gVKD5R0xxc1b3
p2jz--0g25Ju1-mmWu1XugY91CL_A5Q94Ur4f7p2--gp2jp1m_u4dQwgDw81Fm-PQ_RRDhdp1zhU5-Gne3Y0vduXq13pgh
d2-beakn1university51t32p33g1t323_737-2021_1-prac7p3

```

Dashboard was accessed:



Four pods, and a single replicaset, were created using two separate YAML files.

```
! createPod.yaml
1  apiVersion: v1
2  kind: Pod
3  metadata:
4    name: mypod
5    labels:
6      run: mypod
7  spec:
8    containers:
9      - image: alamfaisal654/node-web-app:latest
10        name: nodewebapp
11        ports:
12          - containerPort: 8080
13        dnsPolicy: ClusterFirst
14        restartPolicy: Always
15
```

```
! createReplicaSet.yaml
1  apiVersion: apps/v1
2  kind: ReplicaSet
3  metadata:
4    name: myreplicaset
5  labels:
6    app: myapp
7    tier: frontend
8  spec:
9    # modify replicas according to your case
10   replicas: 3
11   selector:
12     matchLabels:
13       tier: frontend
14   template:
15     metadata:
16       labels:
17         tier: frontend
18     spec:
19       containers:
20         - name: mynode
21           image: alamfaisal654/node-web-app:latest
22           ports:
23             - containerPort: 8080
```

The four pods and the replicaset were then successfully applied:

```
D:\DeakinUniversity\SIT323\sit323_737-2023-t1-prac7p>kubect1 apply -f createReplicaSet.yaml
replicaset.apps/myreplicaset created

D:\DeakinUniversity\SIT323\sit323_737-2023-t1-prac7p>kubect1 apply -f createPod.yaml
pod/mypod created
```

The screenshot shows the Kubernetes Dashboard interface. The 'Workload Status' section displays two green circles: one for 'Pods' labeled 'Running: 4' and another for 'Replica Sets' labeled 'Running: 1'. Below this, the 'Pods' table lists four pods, all in a 'Running' state. The 'Replica Sets' table shows one replicaset, 'myreplicaset', also in a 'Running' state.

Name	Images	Labels	Node	Status	Restarts	CPU Usage (cores)	Memory Usage (bytes)	Created ↑
mypod	alanfaizal654/node-web-app:latest	run: mypod	docker-desktop	Running	0	-	-	3 minutes ago
myreplicaset-v99fb	alanfaizal654/node-web-app:latest	tier: frontend	docker-desktop	Running	0	-	-	3 minutes ago
myreplicaset-xvsc1	alanfaizal654/node-web-app:latest	tier: frontend	docker-desktop	Running	0	-	-	3 minutes ago
myreplicaset-zgprt	alanfaizal654/node-web-app:latest	tier: frontend	docker-desktop	Running	0	-	-	3 minutes ago

Name	Images	Labels	Pods	Created ↑
myreplicaset	alanfaizal654/node web-app:latest	app: myapp tier: frontend	3 / 3	3 minutes ago

Finally, the Deployment Kubernetes object was created and applied:

```
D:\DeakinUniversity\SIT323\sit323_737-2023-t1-prac7p>kubect1 apply -f createDeployment.yaml
deployment.apps/mydeployment created
```

The screenshot shows the Kubernetes Dashboard interface after creating a deployment. The 'Workload Status' section now includes three green circles: 'Deployments' (Running: 1), 'Pods' (Running: 4), and 'Replica Sets' (Running: 1). The 'Deployments' table shows 'mydeployment' in a 'Running' state. The 'Pods' and 'Replica Sets' tables remain the same as in the previous screenshot.

Name	Images	Labels	Pods	Created ↑
mydeployment	alanfaizal654/node-web-app:latest	app: myapp tier: frontend	3 / 3	41 seconds ago

Name	Images	Labels	Node	Status	Restarts	CPU Usage (cores)	Memory Usage (bytes)	Created ↑
mypod	alanfaizal654/node-web-app:latest	run: mypod	docker-desktop	Running	0	-	-	4 minutes ago
myreplicaset-v99fb	alanfaizal654/node web-app:latest	tier: frontend	docker-desktop	Running	0	-	-	4 minutes ago
myreplicaset-xvsc1	alanfaizal654/node-web-app:latest	tier: frontend	docker-desktop	Running	0	-	-	4 minutes ago
myreplicaset-zgprt	alanfaizal654/node-web-app:latest	tier: frontend	docker-desktop	Running	0	-	-	4 minutes ago

Name	Images	Labels	Pods	Created ↑
myreplicaset	alanfaizal654/node-web-app:latest	app: myapp tier: frontend	3 / 3	4 minutes ago

A larger image of the dashboard is provided below, for visual clarity:

The screenshot displays the Kubernetes Dashboard interface. The top navigation bar shows the 'Workloads' section selected. The left sidebar lists various Kubernetes resources, including Workloads, Cron Jobs, Daemon Sets, Deployments, Jobs, Pods, Replica Sets, Replication Controllers, Stateful Sets, Service, Ingresses, Ingress Classes, Services, Config and Storage, Cluster, Cluster Role Bindings, Cluster Roles, Events, Namespaces, Network Policies, Nodes, Persistent Volumes, Role Bindings, Roles, and Service Accounts.

The main content area is divided into four sections:

- Workload Status:** A summary view showing the status of Deployments, Pods, and Replica Sets. Each resource is represented by a green circle and a line indicating the number of running instances: Deployments (Running: 1), Pods (Running: 4), and Replica Sets (Running: 1).
- Deployments:** A table listing the deployment 'mydeployment' with details such as Name, Images, Labels, Pods, and Created time.
- Pods:** A table listing four pods: 'mypod', 'myreplicaset-995fb', 'myreplicaset-kx6c4', and 'myreplicaset-zghrt'. Each pod entry includes details such as Name, Images, Labels, Node, Status, Restarts, CPU Usage, Memory Usage, and Created time.
- Replica Sets:** A table listing the Replica Set 'myreplicaset' with details such as Name, Images, Labels, Pods, and Created time.

The bottom of the dashboard shows the Windows taskbar with the search bar and system tray icons.

Deployment, pods, and replicaset, all successful.