

OTOT Task A2-3 Report

Name: Pham Ba Thang

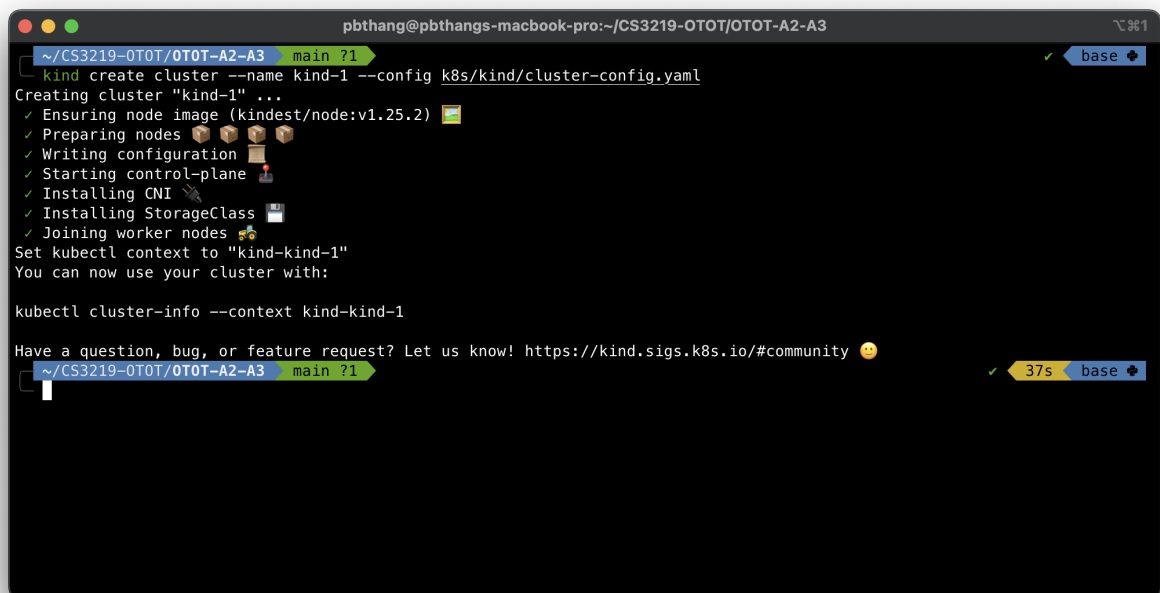
Matric No: A0219715B

Repo: <https://github.com/pbthang/OTOT-A2-A3>

Task A2.1

- Create a cluster using **kind**:

```
kind create cluster --name kind-1 --config k8s/kind/cluster-  
config.yaml
```



A terminal window screenshot showing the execution of the command `kind create cluster --name kind-1 --config k8s/kind/cluster-config.yaml`. The output shows the cluster creation process with progress bars and icons for each step: Ensuring node image (kindest/node:v1.25.2), Preparing nodes, Writing configuration, Starting control-plane, Installing CNI, Installing StorageClass, and Joining worker nodes. The terminal also shows the command `kubectl cluster-info --context kind-kind-1` and a link to the kind community page. The terminal window has a title bar with the user 'pbthang' and the file path '~/CS3219-OTOT/OTOT-A2-A3'.

- Verify cluster and nodes are running:

```
kubectl cluster-info --context kind-kind-1  
kubectl get nodes --context kind-kind-1 -o wide
```

```

~/CS3219-OTOT/OTOT-A2-A3 main 72
k cluster-info
Kubernetes control plane is running at https://127.0.0.1:57786
CoreDNS is running at https://127.0.0.1:57786/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.

~/CS3219-OTOT/OTOT-A2-A3 main 72
k get nodes -o wide
NAME                STATUS    ROLES    AGE     VERSION   INTERNAL-IP   EXTERNAL-IP   OS-IMAGE             KERNEL-VERSION   CONTAINER-RUNTIME
kind-1-control-plane Ready    control-plane 4m7s    v1.25.2   172.20.0.5    <none>         Ubuntu 22.04.1 LTS   5.10.124-linuxkit containerd://1.6.8
kind-1-worker       Ready    <none>       3m47s    v1.25.2   172.20.0.4    <none>         Ubuntu 22.04.1 LTS   5.10.124-linuxkit containerd://1.6.8
kind-1-worker2      Ready    <none>       3m47s    v1.25.2   172.20.0.2    <none>         Ubuntu 22.04.1 LTS   5.10.124-linuxkit containerd://1.6.8
kind-1-worker3      Ready    <none>       3m47s    v1.25.2   172.20.0.3    <none>         Ubuntu 22.04.1 LTS   5.10.124-linuxkit containerd://1.6.8

```

Task A2.2

- Create a deployment:

```
kubectl apply -f k8s/manifests/backend-deployment.yml
```

- Verify that deployments/pods is running:

```
kubectl get deploy/backend --watch
kubectl get po -lapp=backend --watch
```

```

~/CS3219-OTOT/OTOT-A2-A3 main 72
k apply -f k8s/manifests/backend-deployment.yml
deployment.apps/backend created

~/CS3219-OTOT/OTOT-A2-A3 main 72
k get deploy/backend --watch
NAME    READY   UP-TO-DATE   AVAILABLE   AGE
backend 0/3      3             0           16s
backend 1/3      3             1           24s
backend 2/3      3             2           26s
backend 3/3      3             3           28s

~/CS3219-OTOT/OTOT-A2-A3 main 72
k get po -lapp=backend --watch
NAME                                READY   STATUS    RESTARTS   AGE
backend-54db4b9667-2k55w             1/1     Running   0           45s
backend-54db4b9667-6bhwf             1/1     Running   0           45s
backend-54db4b9667-brjgk             1/1     Running   0           45s

```

- Create Ingress controller (nginx-ingress-controller) and verify it

```
kubectl apply -f https://raw.githubusercontent.com/kubernetes/ingress-nginx/main/deploy/static/provider/kind/deploy.yaml
kubectl -n ingress-nginx get deploy -w
```

```

pbthang@pbthangs-macbook-pro:~/CS3219-OTOT/OTOT-A2-A3
~/CS3219-OTOT/OTOT-A2-A3 main ?2
kubectll apply -f https://raw.githubusercontent.com/kubernetes/ingress-nginx/main/deploy/static/provider
/kind/deploy.yaml
namespace/ingress-nginx created
serviceaccount/ingress-nginx created
serviceaccount/ingress-nginx-admission created
role.rbac.authorization.k8s.io/ingress-nginx created
role.rbac.authorization.k8s.io/ingress-nginx-admission created
clusterrole.rbac.authorization.k8s.io/ingress-nginx created
clusterrole.rbac.authorization.k8s.io/ingress-nginx-admission created
rolebinding.rbac.authorization.k8s.io/ingress-nginx created
rolebinding.rbac.authorization.k8s.io/ingress-nginx-admission created
clusterrolebinding.rbac.authorization.k8s.io/ingress-nginx created
clusterrolebinding.rbac.authorization.k8s.io/ingress-nginx-admission created
configmap/ingress-nginx-controller created
service/ingress-nginx-controller created
service/ingress-nginx-controller-admission created
deployment.apps/ingress-nginx-controller created
job.batch/ingress-nginx-admission-create created
job.batch/ingress-nginx-admission-patch created
ingressclass.networking.k8s.io/nginx created
validatingwebhookconfiguration.admissionregistration.k8s.io/ingress-nginx-admission created
~/CS3219-OTOT/OTOT-A2-A3 main ?2 ✓ base
~/CS3219-OTOT/OTOT-A2-A3 main ?2 ✓ base

```

```

pbthang@pbthangs-macbook-pro:~/CS3219-OTOT/OTOT-A2-A3
~/CS3219-OTOT/OTOT-A2-A3 main ?2 ✓ base kind-kind-1 *
kubectll -n ingress-nginx get deploy -w
NAME READY UP-TO-DATE AVAILABLE AGE
ingress-nginx-controller 1/1 1 1 3m55s
^C
~/CS3219-OTOT/OTOT-A2-A3 main ?2 1 x base

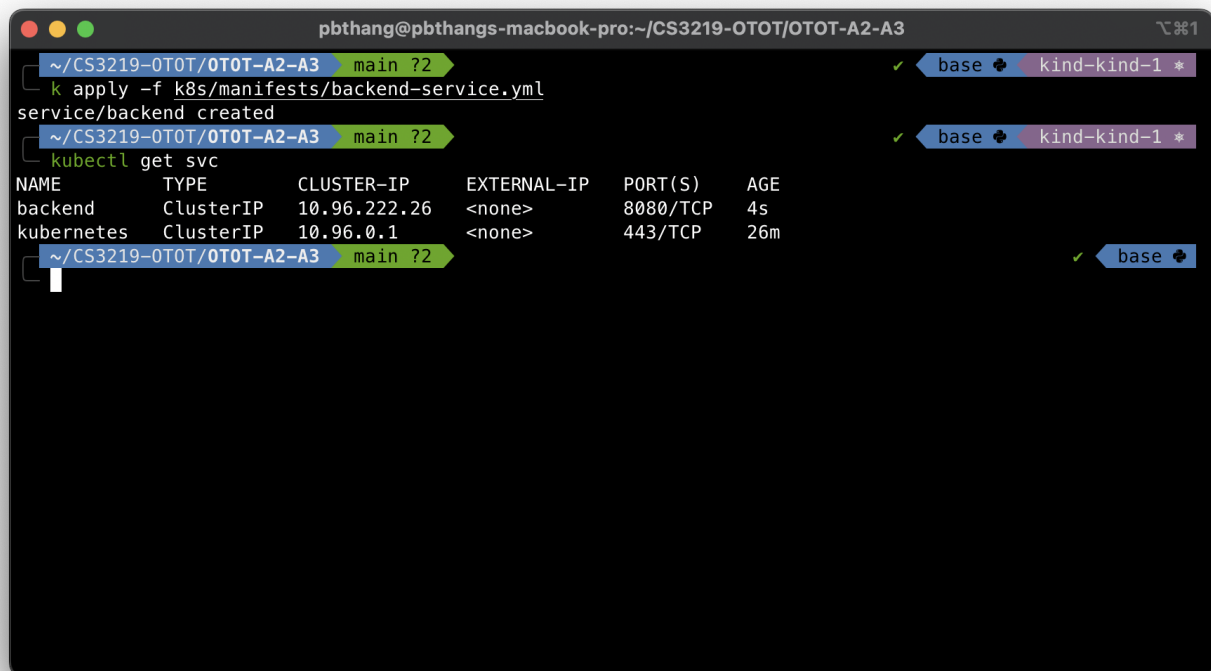
```

- Create a Service and verify it

```

kubectll apply -f k8s/manifests/backend-service.yml
kubectll get svc

```

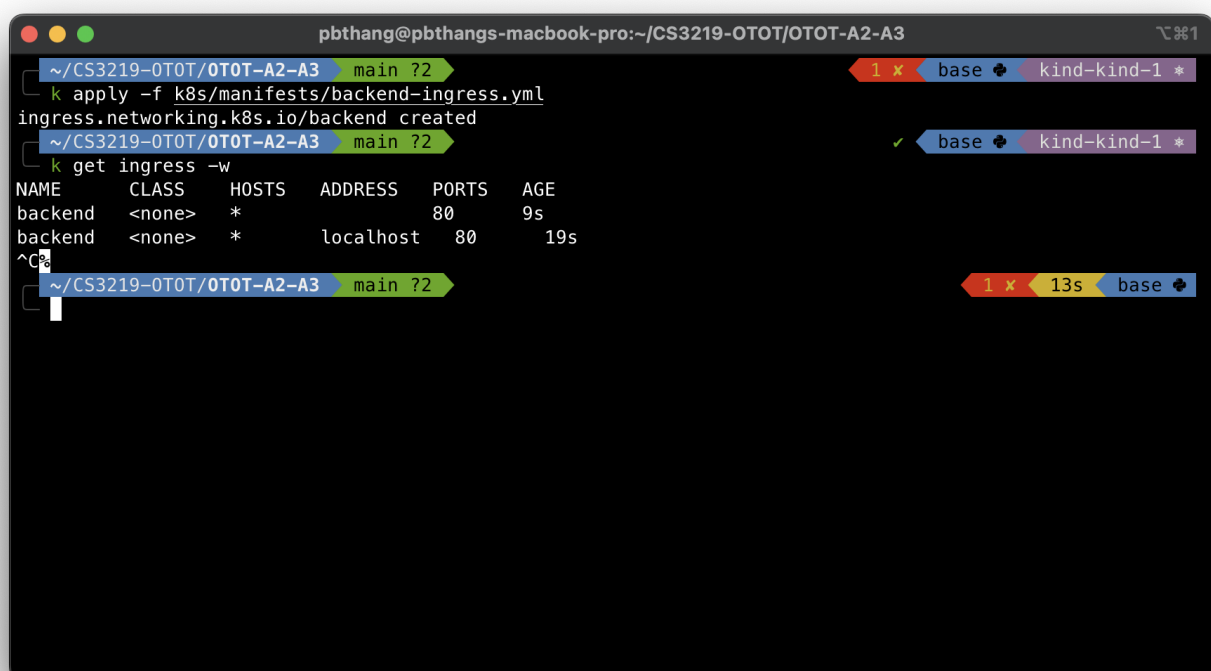


```
pbthang@pbthangs-macbook-pro:~/CS3219-OTOT/OTOT-A2-A3
~/CS3219-OTOT/OTOT-A2-A3 main ?2
k apply -f k8s/manifests/backend-service.yml
service/backend created
~/CS3219-OTOT/OTOT-A2-A3 main ?2
kubectl get svc
NAME          TYPE          CLUSTER-IP    EXTERNAL-IP    PORT(S)    AGE
backend       ClusterIP     10.96.222.26  <none>         8080/TCP   4s
kubernetes    ClusterIP     10.96.0.1     <none>         443/TCP    26m
```

Task A2.3

- Create an Ingress and verify it

```
kubectl apply -f k8s/manifests/backend-ingress.yml
kubectl get ingress -w
```



```
pbthang@pbthangs-macbook-pro:~/CS3219-OTOT/OTOT-A2-A3
~/CS3219-OTOT/OTOT-A2-A3 main ?2
k apply -f k8s/manifests/backend-ingress.yml
ingress.networking.k8s.io/backend created
~/CS3219-OTOT/OTOT-A2-A3 main ?2
k get ingress -w
NAME          CLASS    HOSTS    ADDRESS    PORTS    AGE
backend       <none>   *        localhost  80       9s
backend       <none>   *        localhost  80       19s
^C
```

If you succeed, then you should see this on <http://localhost>:

