Keane Chan Jun Yu
A0205678W

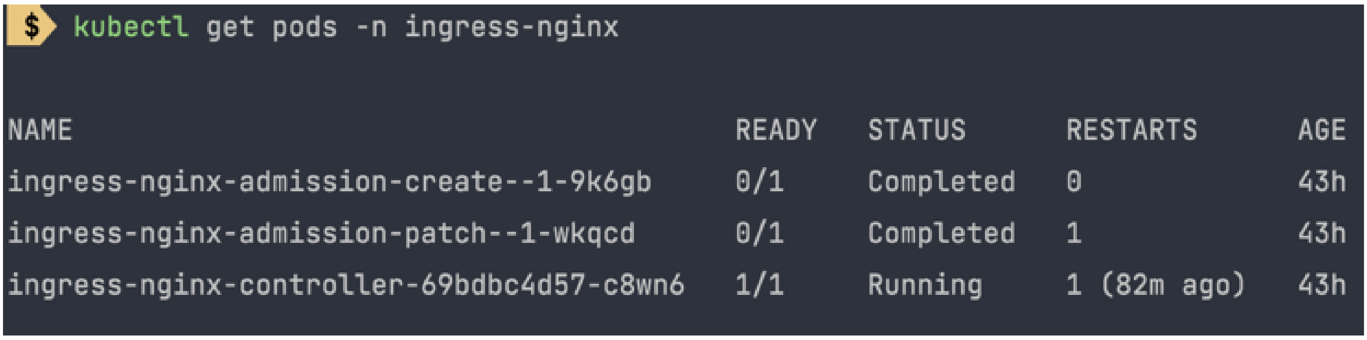
GitHub Repo: https://github.com/keanecjy/CS3219-TaskA3

CS3219-TaskA3 - Ingress Controller

Deployment

Verify that NGINX Ingress controller is up

```
kubectl get pods -n ingress-nginx
```

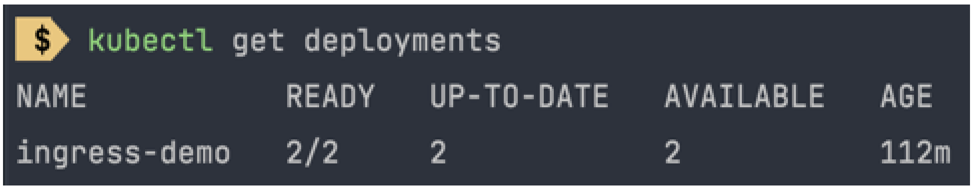


```
$ kubectl get pods -n ingress-nginx

NAME                                        READY   STATUS      RESTARTS       AGE
ingress-nginx-admission-create--1-9k6gb     0/1     Completed   0              43h
ingress-nginx-admission-patch--1-wkqcd      0/1     Completed   1              43h
ingress-nginx-controller-69bdbc4d57-c8wn6   1/1     Running     1 (82m ago)    43h
```

Deploy and create service

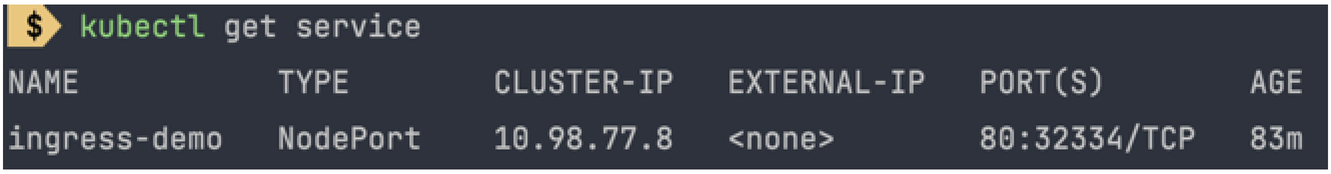
```
kubectl apply -f deployment.yml
```

```
kubectl get deployments
```



```
$ kubectl get deployments
NAME           READY   UP-TO-DATE   AVAILABLE   AGE
ingress-demo   2/2     2            2           112m
```

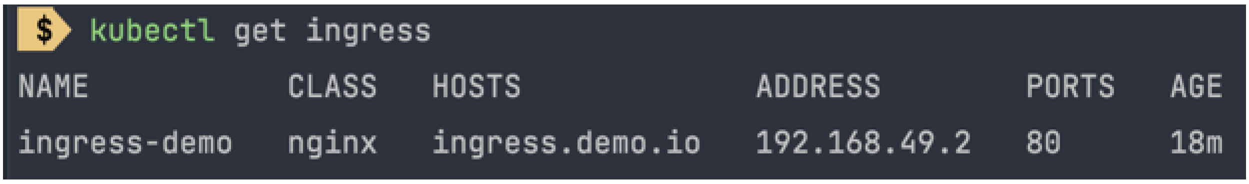
```
kubectl get service
```



```
$ kubectl get service
NAME           TYPE       CLUSTER-IP    EXTERNAL-IP   PORT(S)        AGE
ingress-demo   NodePort   10.98.77.8    <none>        80:32334/TCP   83m
```

Add Ingress controller. It can take awhile for the ip address to be set.

```
kubectl apply -f ingress.yml
kubectl get ingress
```

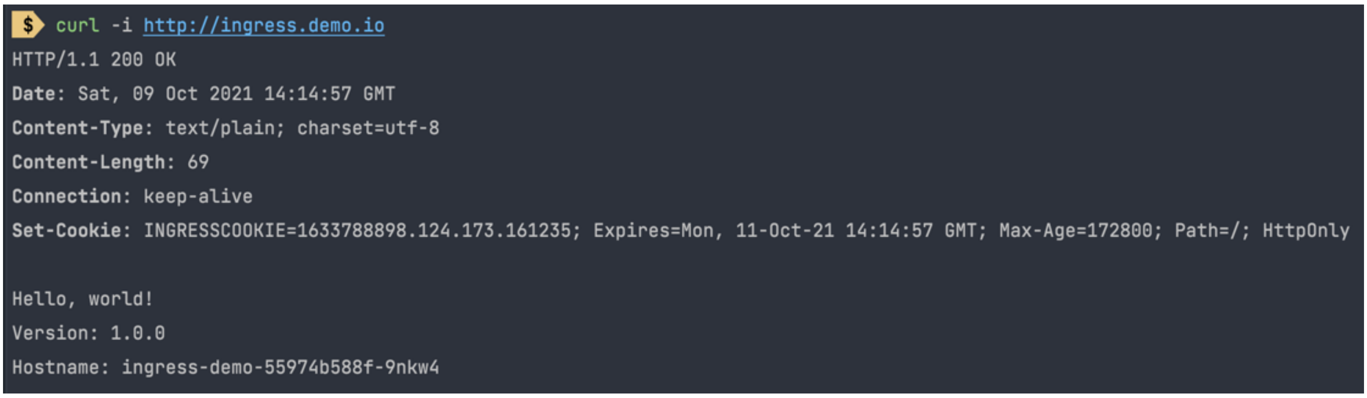


```
$ kubectl get ingress
NAME           CLASS   HOSTS             ADDRESS        PORTS   AGE
ingress-demo   nginx   ingress.demo.io   192.168.49.2   80      18m
```

Using this ip address, go to /etc/hosts file and add `<IP_ADDRESS> ingress.demo.io` at the bottom of the file.

Now we can access the endpoint at http://ingress.demo.io

```
curl -i http://ingress.demo.io
```



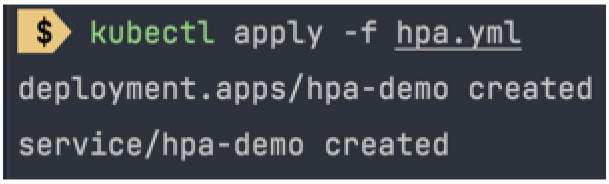
```
$ curl -i http://ingress.demo.io
HTTP/1.1 200 OK
Date: Sat, 09 Oct 2021 14:14:57 GMT
Content-Type: text/plain; charset=utf-8
Content-Length: 69
Connection: keep-alive
Set-Cookie: INGRESSCOOKIE=1633788898.124.173.161235; Expires=Mon, 11-Oct-21 14:14:57 GMT; Max-Age=172800; Path=/; HttpOnly

Hello, world!
Version: 1.0.0
Hostname: ingress-demo-55974b588f-9nkw4
```

As seen here, we see that there is a cookie set in the field. Given the same cookie, the routing will always direct back to the same webpage.

CS3219-TaskA3 - Horizontal pod auto-scaler

Create deployment, expose it and attach service

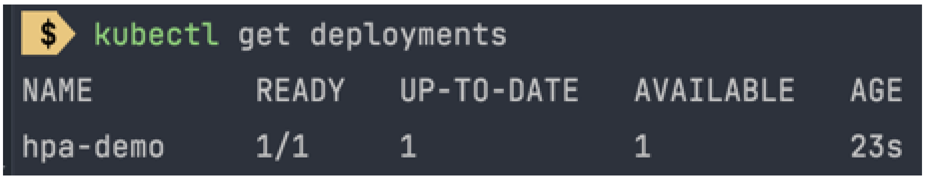
```
kubectl apply -f hpa.yml
```



```
$ kubectl apply -f hpa.yml
deployment.apps/hpa-demo created
service/hpa-demo created
```

Check if deployment exist

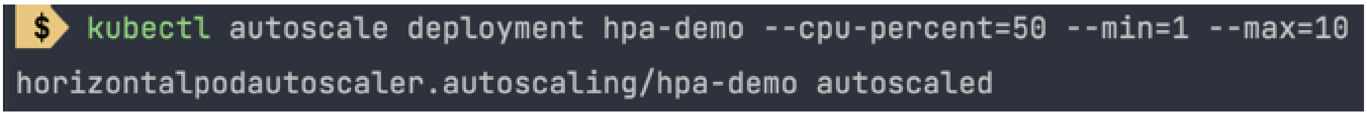
```
kubectl get deployments
```



```
$ kubectl get deployments
NAME         READY   UP-TO-DATE   AVAILABLE   AGE
hpa-demo     1/1     1            1           23s
```

Create hpa

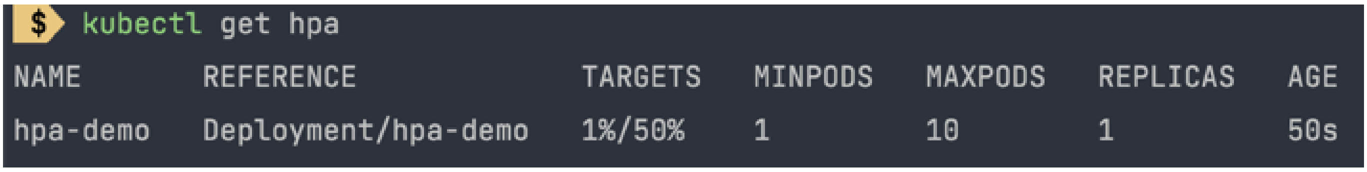
```
kubectl autoscale deployment hpa-demo --cpu-percent=50 --min=1 --max=10
```



```
$ kubectl autoscale deployment hpa-demo --cpu-percent=50 --min=1 --max=10
horizontalpodautoscaler.autoscaling/hpa-demo autoscaled
```

Check status of autoscaler (might take some time to show %). We should expect 0% since we are not sending any requests to the server.

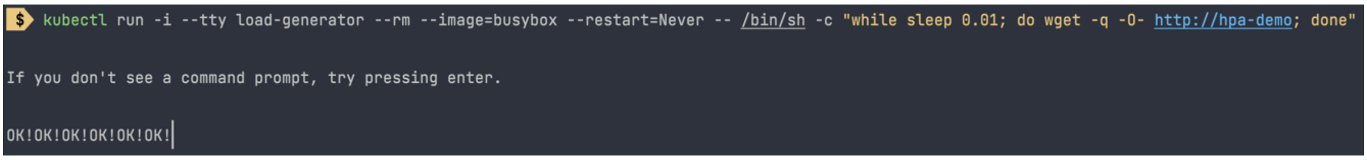
```
kubectl get hpa
```



```
$ kubectl get hpa
NAME       REFERENCE             TARGETS   MINPODS   MAXPODS   REPLICAS   AGE
hpa-demo   Deployment/hpa-demo   1%/50%    1         10        1          50s
```

Increase load on the service

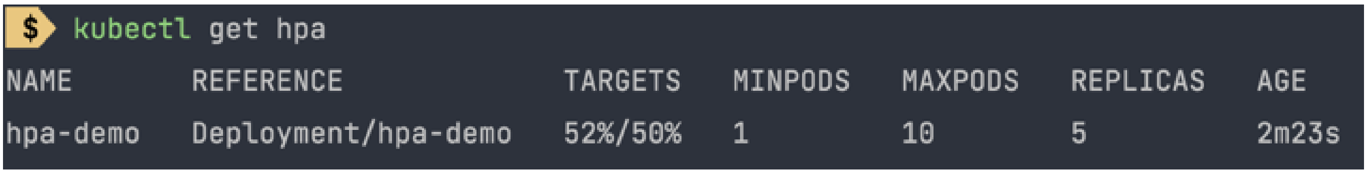
```
kubectl run -i --tty load-generator --rm --image=busybox --restart=Never -
- /bin/sh -c "while sleep 0.01; do wget -q -O- http://hpa-demo; done"
```



```
$ kubectl run -i --tty load-generator --rm --image=busybox --restart=Never -- /bin/sh -c "while sleep 0.01; do wget -q -O- http://hpa-demo; done"

If you don't see a command prompt, try pressing enter.

OK!OK!OK!OK!OK!OK!
```

Open up a new terminal. After 1 minute, we should see higher cpu load:

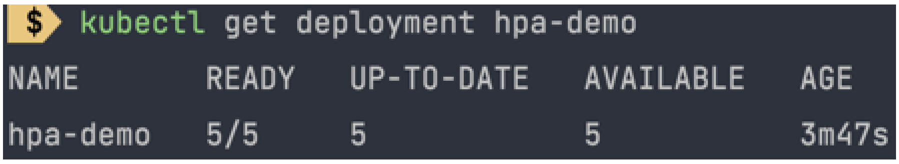
```
kubectl get hpa
```



```
$ kubectl get hpa
NAME       REFERENCE             TARGETS   MINPODS   MAXPODS   REPLICAS   AGE
hpa-demo   Deployment/hpa-demo   52%/50%   1         10        5          2m23s
```

We should also see more replicas created for the deployment

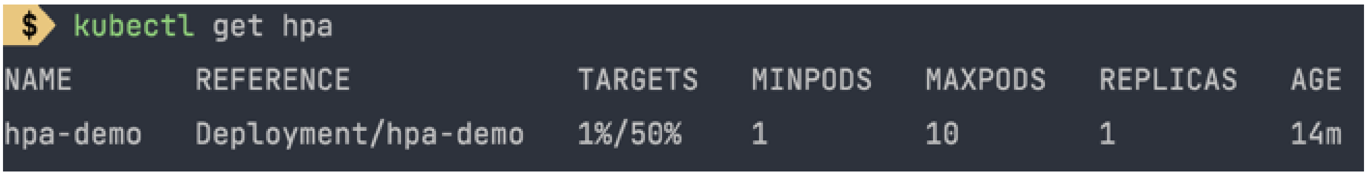
```
kubectl get deployment hpa-demo
```



```
$ kubectl get deployment hpa-demo
NAME       READY   UP-TO-DATE   AVAILABLE   AGE
hpa-demo   5/5     5            5           3m47s
```

Now we stop the container with the busybox image using Ctrl-C (SIGINT). After 1 minute or so, cpu utilization will drop back to 0%

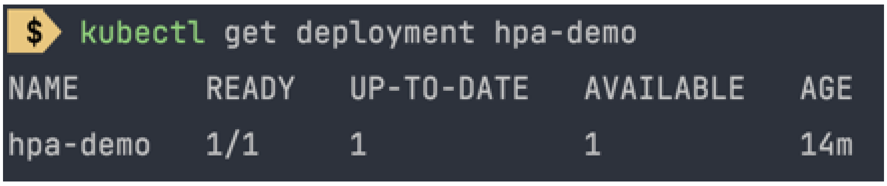
```
kubectl get hpa
```



```
$ kubectl get hpa
NAME       REFERENCE             TARGETS   MINPODS   MAXPODS   REPLICAS   AGE
hpa-demo   Deployment/hpa-demo   1%/50%    1         10        1          14m
```

After several minutes, the HPA will autoscale and decrease the no. of replicas down to 1.

```
kubectl get deployment hpa-demo
```



```
$ kubectl get deployment hpa-demo
NAME       READY   UP-TO-DATE   AVAILABLE   AGE
hpa-demo   1/1     1            1           14m
```

Resources

1. https://kubernetes.io/docs/tasks/run-application/horizontal-pod-autoscale-walkthrough/