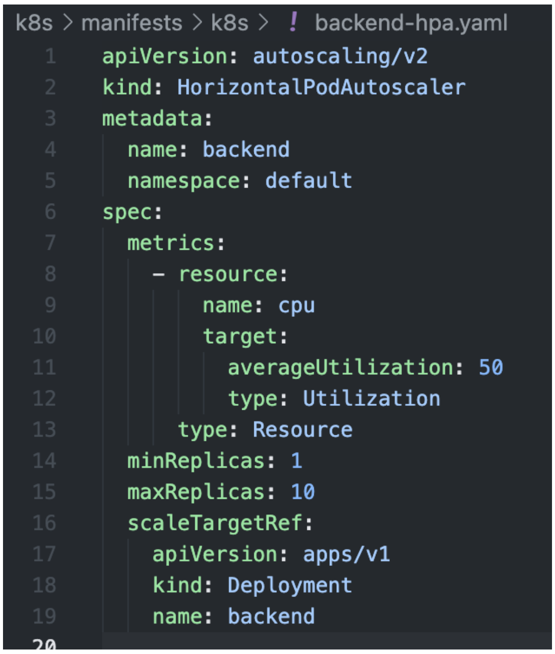
Name: Chen Hsiao Ting
Matriculation Number: A0222182R
Link to GitHub repository: https://github.com/hsiaotingluv/CS3219-OTOT-TaskA2-A3
Link to Demo video:
https://drive.google.com/file/d/1q6xOcmwVU5KD8ske_ENULFgm_kKD_omM/view?usp=sharing

# Instructions on how to create k8s objects

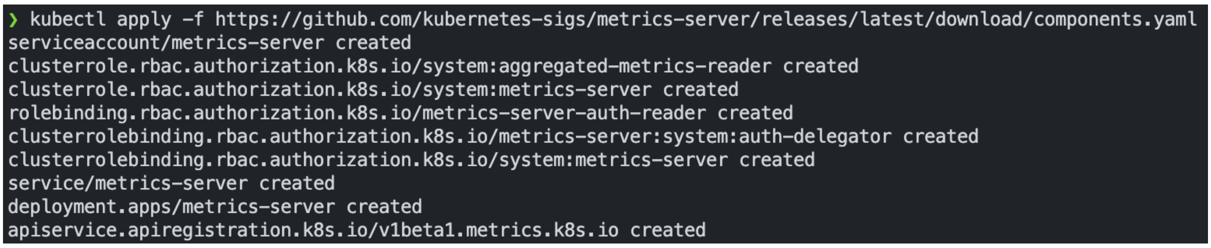
## Task A3.1: Deploy a metrics-server and HorizontalPodAutoscaler

1. Add the relevant HorizontalPodAutoscaler manifest



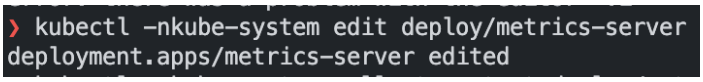
```yaml
apiVersion: autoscaling/v2
kind: HorizontalPodAutoscaler
metadata:
  name: backend
  namespace: default
spec:
  metrics:
    - resource:
        name: cpu
        target:
          averageUtilization: 50
          type: Utilization
      type: Resource
  minReplicas: 1
  maxReplicas: 10
  scaleTargetRef:
    apiVersion: apps/v1
    kind: Deployment
    name: backend
```

2. Create the metrics-server and verify it works



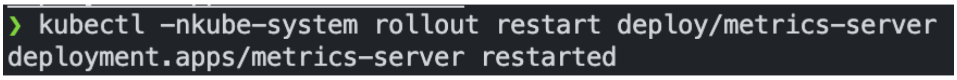
```
❯ kubectl apply -f https://github.com/kubernetes-sigs/metrics-server/releases/latest/download/components.yaml
serviceaccount/metrics-server created
clusterrole.rbac.authorization.k8s.io/system:aggregated-metrics-reader created
clusterrole.rbac.authorization.k8s.io/system:metrics-server created
rolebinding.rbac.authorization.k8s.io/metrics-server-auth-reader created
clusterrolebinding.rbac.authorization.k8s.io/metrics-server:system:auth-delegator created
clusterrolebinding.rbac.authorization.k8s.io/system:metrics-server created
service/metrics-server created
deployment.apps/metrics-server created
apiservice.apiregistration.k8s.io/v1beta1.metrics.k8s.io created
```

- run `kubectl apply -f https://github.com/kubernetes-sigs/metrics-server/releases/latest/download/components.yaml` to create metrics-server



```
❯ kubectl -nkube-system edit deploy/metrics-server
deployment.apps/metrics-server edited
```

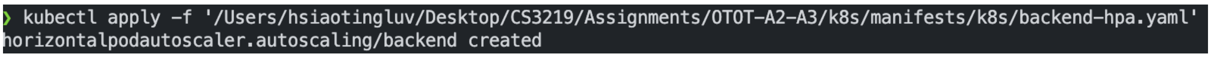
- run `kubectl -nkube-system edit deploy/metrics-server` to manually edit the Deployment manifest to add a flag `--kubelet-insecure-tls` to `deployment.spec.containers[].args[]`



```
❯ kubectl -nkube-system rollout restart deploy/metrics-server
deployment.apps/metrics-server restarted
```

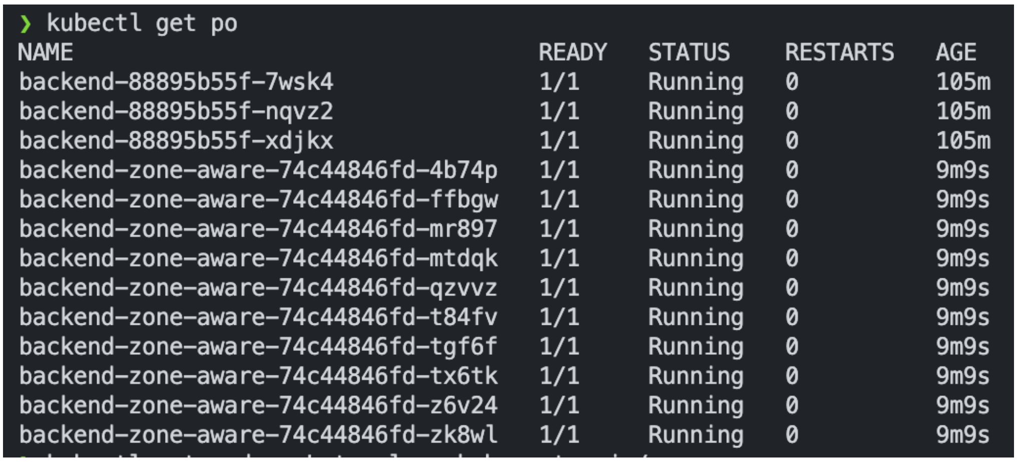
- restart the Deployment using `kubectl -nkube-system rollout restart deploy/metrics-server`

## 3. Apply the HPA and verify that it works



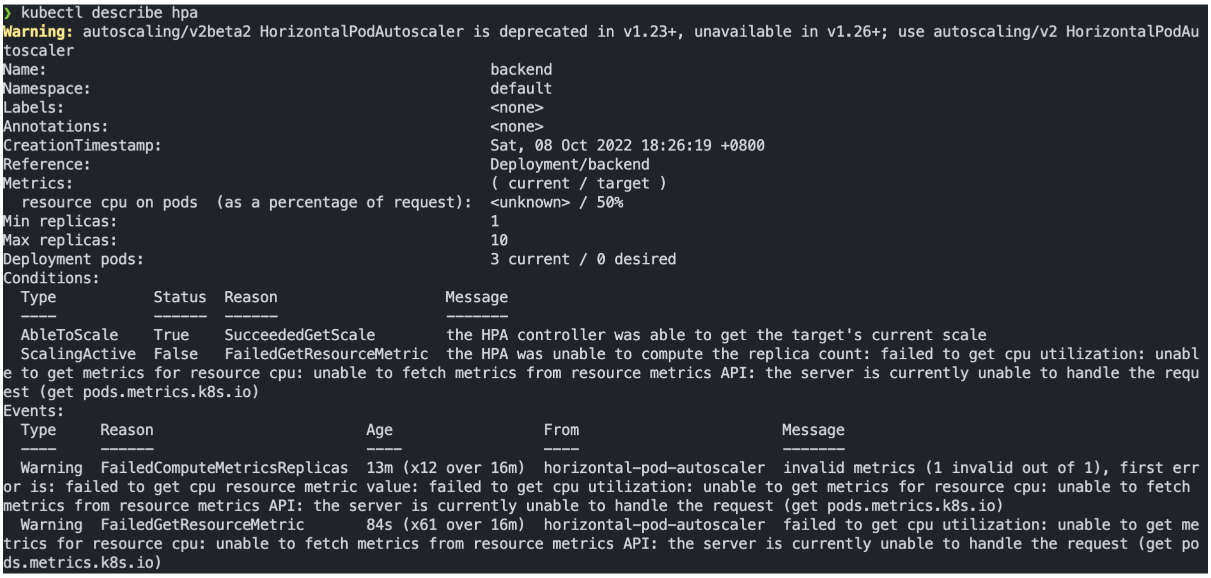
```
❯ kubectl apply -f '/Users/hsiaotingluv/Desktop/CS3219/Assignments/OTOT-A2-A3/k8s/manifests/k8s/backend-hpa.yaml'
horizontalpodautoscaler.autoscaling/backend created
```

- run `kubectl apply -f backend-hpa.yaml` to apply HPA



```
❯ kubectl get po
NAME                                 READY   STATUS    RESTARTS   AGE
backend-88895b55f-7wsk4              1/1     Running   0          105m
backend-88895b55f-nqvz2              1/1     Running   0          105m
backend-88895b55f-xdjkx              1/1     Running   0          105m
backend-zone-aware-74c44846fd-4b74p  1/1     Running   0          9m9s
backend-zone-aware-74c44846fd-ffbgw  1/1     Running   0          9m9s
backend-zone-aware-74c44846fd-mr897  1/1     Running   0          9m9s
backend-zone-aware-74c44846fd-mtdqk  1/1     Running   0          9m9s
backend-zone-aware-74c44846fd-qzvvz  1/1     Running   0          9m9s
backend-zone-aware-74c44846fd-t84fv  1/1     Running   0          9m9s
backend-zone-aware-74c44846fd-tgf6f  1/1     Running   0          9m9s
backend-zone-aware-74c44846fd-tx6tk  1/1     Running   0          9m9s
backend-zone-aware-74c44846fd-z6v24  1/1     Running   0          9m9s
backend-zone-aware-74c44846fd-zk8wl  1/1     Running   0          9m9s
```

- run `kubectl get po`

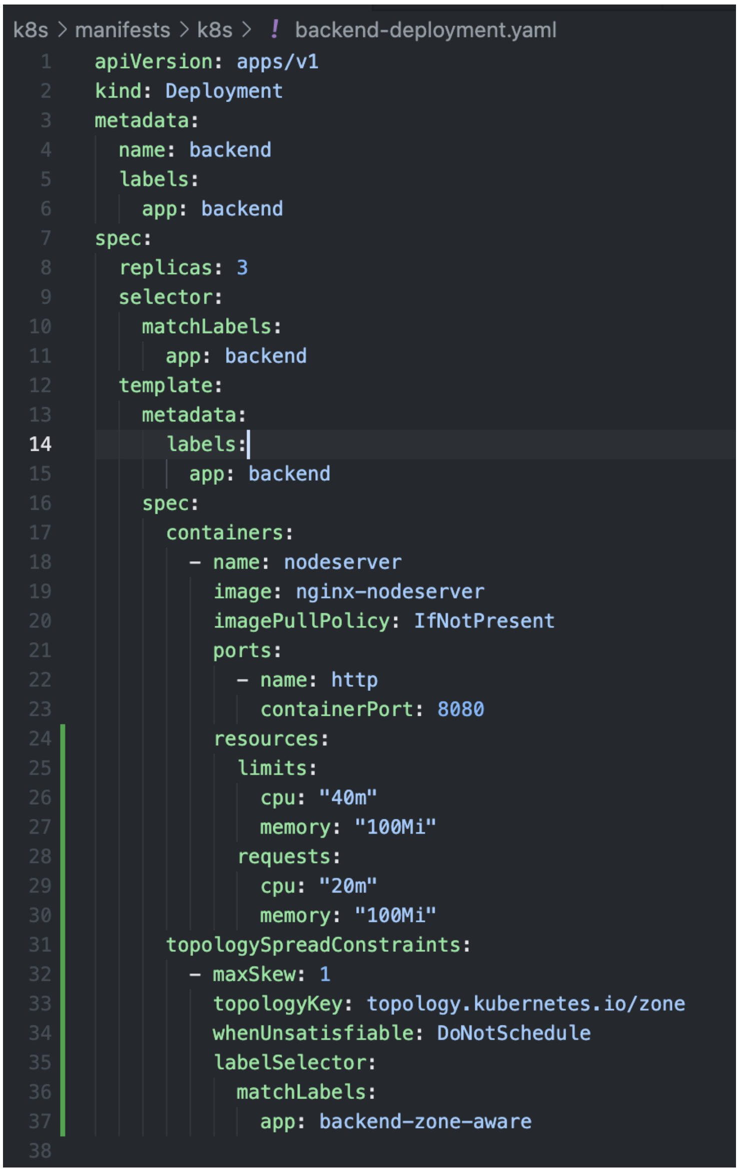


```
❯ kubectl describe hpa
Warning: autoscaling/v2beta2 HorizontalPodAutoscaler is deprecated in v1.23+, unavailable in v1.26+; use autoscaling/v2 HorizontalPodAu
toscaler
Name:                                                  backend
Namespace:                                             default
Labels:                                                <none>
Annotations:                                           <none>
CreationTimestamp:                                     Sat, 08 Oct 2022 18:26:19 +0800
Reference:                                             Deployment/backend
Metrics:                                               ( current / target )
  resource cpu on pods  (as a percentage of request):  <unknown> / 50%
Min replicas:                                          1
Max replicas:                                          10
Deployment pods:                                       3 current / 0 desired
Conditions:
  Type            Status  Reason                   Message
  ----            ------  ------                   -------
  AbleToScale     True    SucceededGetScale        the HPA controller was able to get the target's current scale
  ScalingActive   False   FailedGetResourceMetric  the HPA was unable to compute the replica count: failed to get cpu utilization: unabl
e to get metrics for resource cpu: unable to fetch metrics from resource metrics API: the server is currently unable to handle the requ
est (get pods.metrics.k8s.io)
Events:
  Type     Reason                        Age                 From                       Message
  ----     ------                        ----                ----                       -------
  Warning  FailedComputeMetricsReplicas  13m (x12 over 16m)  horizontal-pod-autoscaler  invalid metrics (1 invalid out of 1), first err
or is: failed to get cpu resource metric value: failed to get cpu utilization: unable to get metrics for resource cpu: unable to fetch
metrics from resource metrics API: the server is currently unable to handle the request (get pods.metrics.k8s.io)
  Warning  FailedGetResourceMetric       84s (x61 over 16m)  horizontal-pod-autoscaler  failed to get cpu utilization: unable to get me
trics for resource cpu: unable to fetch metrics from resource metrics API: the server is currently unable to handle the request (get po
ds.metrics.k8s.io)
```

- run `kubectl describe hpa`

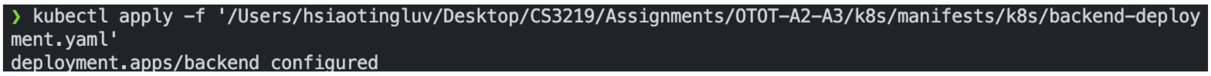
## Task A3.2: deploy another version of your A2 Deployment in a zone-aware manner

1. Add the relevant Deployment manifest



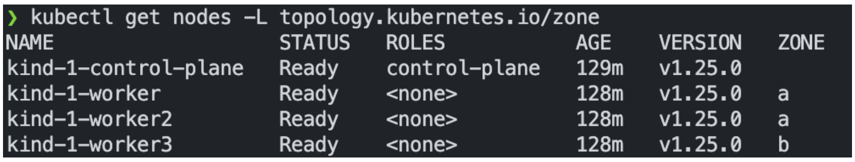
```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: backend
  labels:
    app: backend
spec:
  replicas: 3
  selector:
    matchLabels:
      app: backend
  template:
    metadata:
      labels:
        app: backend
    spec:
      containers:
        - name: nodeserver
          image: nginx-nodeserver
          imagePullPolicy: IfNotPresent
          ports:
            - name: http
              containerPort: 8080
          resources:
            limits:
              cpu: "40m"
              memory: "100Mi"
            requests:
              cpu: "20m"
              memory: "100Mi"
      topologySpreadConstraints:
        - maxSkew: 1
          topologyKey: topology.kubernetes.io/zone
          whenUnsatisfiable: DoNotSchedule
          labelSelector:
            matchLabels:
              app: backend-zone-aware

```

2. Apply the Deployment and verify it works



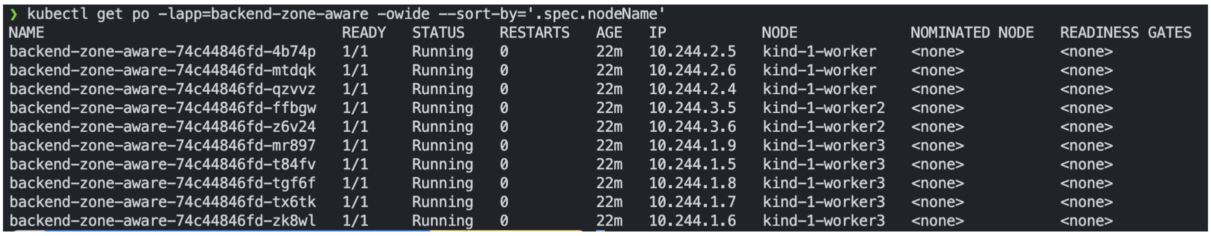
```
❯ kubectl apply -f '/Users/hsiaotingluv/Desktop/CS3219/Assignments/OTOT-A2-A3/k8s/manifests/k8s/backend-deploy
ment.yaml'
deployment.apps/backend configured
```

- run `kubectl apply -f backend-deployment.yaml` to reapply Zone Aware Deployment manifest



```
❯ kubectl get nodes -L topology.kubernetes.io/zone
NAME                  STATUS   ROLES           AGE    VERSION   ZONE
kind-1-control-plane  Ready    control-plane   129m   v1.25.0
kind-1-worker         Ready    <none>          128m   v1.25.0   a
kind-1-worker2        Ready    <none>          128m   v1.25.0   a
kind-1-worker3        Ready    <none>          128m   v1.25.0   b
```

- run `kubectl get nodes -L topology.kubernetes.io/zone` to verify
- As you can see, each worker node is labeled with key "topology.kubernetes.io/zone" and a letter zone "a" or "b".



```
❯ kubectl get po -lapp=backend-zone-aware -owide --sort-by='.spec.nodeName'
NAME                                  READY   STATUS    RESTARTS   AGE   IP            NODE             NOMINATED NODE   READINESS GATES
backend-zone-aware-74c44846fd-4b74p   1/1     Running   0          22m   10.244.2.5    kind-1-worker    <none>           <none>
backend-zone-aware-74c44846fd-mtdqk   1/1     Running   0          22m   10.244.2.6    kind-1-worker    <none>           <none>
backend-zone-aware-74c44846fd-qzvvz   1/1     Running   0          22m   10.244.2.4    kind-1-worker    <none>           <none>
backend-zone-aware-74c44846fd-ffbgw   1/1     Running   0          22m   10.244.3.5    kind-1-worker2   <none>           <none>
backend-zone-aware-74c44846fd-z6v24   1/1     Running   0          22m   10.244.3.6    kind-1-worker2   <none>           <none>
backend-zone-aware-74c44846fd-mr897   1/1     Running   0          22m   10.244.1.9    kind-1-worker3   <none>           <none>
backend-zone-aware-74c44846fd-t84fv   1/1     Running   0          22m   10.244.1.5    kind-1-worker3   <none>           <none>
backend-zone-aware-74c44846fd-tgf6f   1/1     Running   0          22m   10.244.1.8    kind-1-worker3   <none>           <none>
backend-zone-aware-74c44846fd-tx6tk   1/1     Running   0          22m   10.244.1.7    kind-1-worker3   <none>           <none>
backend-zone-aware-74c44846fd-zk8wl   1/1     Running   0          22m   10.244.1.6    kind-1-worker3   <none>           <none>
```

- run `kubectl get po -lapp=backend-zone-aware -owide --sort-by='.spec.nodeName'` to verify if the pods are evenly across the zones