Configure autoscaling in your cluster (Horizontal scaling)

Horizontal Pod Autoscaling (HPA) in Kubernetes automatically scales the number of pod replicas in a deployment, replica set, or stateful set based on observed CPU utilization (or other select metrics).

We will use following steps for this configuration.

1. At first, we required Metrics Sever
To check Metrics Server we use:

$ kubectl get deployment metrics-server -n kube-system

If not installed, then at first we will have to install it.

2. Creating a Deployment

Here I am using NGINX deployment

nginx-deployment.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
spec:
  replicas: 1
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx
        resources:
          requests:
            cpu: 100m
```

```
        limits:
          cpu: 200m
        ports:
        - containerPort: 80
```

Apply it:

$ kubectl apply -f nginx-deployment.yaml

3. Creating a Horizontal Pod Autoscaler

For this, we can use `kubectl` or YAML.

Using kubectl:

$ kubectl autoscale deployment nginx-deployment --cpu-percent=50 --min=1 --max=5

- This will scale the deployment between 1 and 5 pods, targeting 50% average CPU usage.

Using YAML:

nginx-hpa.yaml

```
apiVersion: autoscaling/v2
kind: HorizontalPodAutoscaler
metadata:
  name: nginx-hpa
spec:
  scaleTargetRef:
    apiVersion: apps/v1
    kind: Deployment
    name: nginx-deployment
  minReplicas: 1
  maxReplicas: 5
  metrics:
  - type: Resource
    resource:
      name: cpu
      target:
```

```
    type: Utilization
    averageUtilization: 50
```

Apply it:

```
$ kubectl apply -f nginx-hpa.yaml
```

4. Check the HPA Status

```
$ kubectl get hpa
```

After this, we will get HPA status.