Lesson 5 : Application Autoscaling

Configure a horizontal pod autoscaler for an application.

Manually scale deployment using YAML manifest

- Specifies number of pod replicas for your deployment or deployment config:
  - increased or decreased to meet your application needs.
  - A **replica set** or **replication controller (RC)** will scale your application
- The definition in your deployment or DC's YAML:

```
apiVersion: apps/v1
kind: Deployment
...output omitted...
spec:
  replicas: 1 (1)
  selector:
    matchLabels:
      deployment: scale (2)
  strategy: {}
  template: (3)
    metadata:
      labels:
        deployment: scale (4)
    spec:
      containers:
```

1. Number of replicas / pods to run
2. Selector used by RC to count and scale to desired state
3. A template for pod that RC creates
4. Labels created by template must match those selector

Manually scale deployment using **oc scale** command

- Scale existing deployment config to 3 replicas

```
[student@workstation ~]$ oc scale dc/postgresql --replicas=3
Warning: extensions/v1beta1 Scale is deprecated in v1.2+, unavailable in v1.16+
deploymentconfig.apps.openshift.io/postgresql scaled
```

- Can safely ignore deprecation warning
- Verify the scaling

```
[student@workstation ~]$ oc get dc/postgresql
NAME         REVISION   DESIRED   CURRENT   TRIGGERED BY
postgresql   2          3         3         config,image(postgresql:12-el8)
[student@workstation ~]$ oc get pods
NAME                  READY   STATUS      RESTARTS   AGE
postgresql-1-deploy   0/1     Completed   0          85m
postgresql-2-deploy   0/1     Completed   0          84m
postgresql-2-gcgzx    1/1     Running     0          84m
postgresql-2-pk89n    1/1     Running     0          2m19s
postgresql-2-shbnr    1/1     Running     0          2m19s
```

The Horizontal Pod AutoScaler (HPA)

- The autoscaler works as a control loop with a default of 15 seconds for the sync period.
- OpenShift can autoscale a deployment based on current load on the application pods, by using HorizontalPodAutoscaler (HPA)resource type.
- Controller manager during sync period queries CPU, memory utilization or both for each pod that is targeted by HPA
- HPA uses performance metrics collected by OpenShift Metrics subsystem.
- To implement HPA, all targeted pods must have resource request set

Autoscaling Pods using **oc autoscale** command

- Configure following DC to scale up to max of 10 replicas of pods when cpu hit 80%

```
$ oc autoscale dc/hello --min 1 --max 10 --cpu-percent 80
```

- To get information about HPA in current project:

```
$ oc get hpa
NAME   REFERENCE            TARGETS          MINPODS  MAXPODS  REPLICAS  ...
hello  DeploymentConfig/hello   7%/80%       1        10       1         ...
scale  Deployment/scale  <unknown>/80%      1        10       1         ...
```

- HPA initially has value of <unknown> . It might take up to five minutes to calc current usage. If it continue to show <unknown>, it indicate resource requests is not set. Hence HPA will not scale the pod

Autoscaling Pods using **YAML manifest** file

```
apiVersion: autoscaling/v2
kind: HorizontalPodAutoscaler
metadata:
  name: hello
spec:
  minReplicas: 1          (1)
  maxReplicas: 10         (2)
  metrics:
  - resource:
      name: cpu
      target:
        averageUtilization: 80   (3)
        type: Utilization
    type: Resource
  scaleTargetRef:         (4)
    apiVersion: apps/v1
    kind: Deployment
    name: hello
```

$ oc apply -f hpa.yml

1. Minimum number of pods.
2. Maximum number of pods.
3. Ideal average CPU usage for each pod. If the global average CPU usage is above that value, then the horizontal pod autoscaler starts new pods. If the global average CPU usage is below that value, then the horizontal pod autoscaler deletes pods.
4. Reference to the name of the deployment resource

Autoscaling Pods based on memory

```
apiVersion: autoscaling/v2
kind: HorizontalPodAutoscaler
metadata:
  name: hello
spec:
  minReplicas: 1
  maxReplicas: 10
  metrics:
  - resource:
      name: memory
      target:
        averageUtilization: 80
...output omitted...
```

- only via YAML manifest file

Guided Exercise: Application Autoscaling

You should be able to:

- You should be able to manually scale up a deployment, configure a horizontal pod autoscaler resource, and monitor the autoscaler.

Lab: Configure Applications for Reliability

You should be able to:

- Add resource requests to a Deployment object.
- Configure probes.
- Create a horizontal pod autoscaler resources.