Lesson 03 Demo 14

Configuring Horizontal Pod Autoscaling (HPA)

Objective: To create and configure horizontal pod autoscaling to optimize performance and implement efficient resource utilization

Tools required: kubeadm, kubectl, kubelet, and containerd

Prerequisites: A Kubernetes (refer to Demo 01 from Lesson 01 for setting up a cluster)

Steps to be followed:

1. Create HPA in the master node
2. Check the deployment
3. Verify the HPA

Step 1: Create HPA in the master node

1.1 On the master node, enter the **nano app-hpa.yaml** command to create a YAML file

1.2 Add the following code in the YAML file:

```
apiVersion: v1
kind: Service
metadata:
  name: php-apache
spec:
  ports:
  - port: 80
    protocol: TCP
    targetPort: 80
  selector:
    run: php-apache
---
apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    run: php-apache
  name: php-apache
spec:
  replicas: 1
  selector:
    matchLabels:
      run: php-apache
  template:
    metadata:
      labels:
        run: php-apache
    spec:
      containers:
      - image: k8s.gcr.io/hpa-example
        name: php-apache
        ports:
        - containerPort: 80
        resources:
          requests:
            cpu: 200m
```

```
  GNU nano 6.2                                                    app-hpa.yaml *
---
apiVersion: v1
kind: Service
metadata:
  name: php-apache
spec:
  ports:
  - port: 80
    protocol: TCP
    targetPort: 80
  selector:
    run: php-apache
---
apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    run: php-apache
  name: php-apache
spec:
  replicas: 1
  selector:
    matchLabels:
      run: php-apache
  template:
    metadata:
      labels:
        run: php-apache
    spec:
      containers:
      - image: k8s.gcr.io/hpa-example
        name: php-apache
        ports:
        - containerPort: 80
        resources:
          requests:
            cpu: 200m
```

1.3 Create the HPA using the following command:
kubectl create -f app-hpa.yaml

```
labsuser@master:~$ kubectl create -f app-hpa.yaml
service/php-apache created
deployment.apps/php-apache created
labsuser@master:~$
```

Step 2: Check the deployment

2.1 Verify the pod status using the following command:
kubectl get pods

```
labsuser@master:~$ kubectl get pods
NAME                          READY   STATUS    RESTARTS      AGE
frontend-6xkgb                1/1     Running   1 (10m ago)   23m
frontend-7q6qg                1/1     Running   1 (10m ago)   23m
frontend-bltgs                1/1     Running   1 (10m ago)   23m
php-apache-5f9f45d488-fg59l   1/1     Running   0             26s
```

2.2 Check the HPA deployment using the following command:
kubectl get deployment

```
labsuser@master:~$ kubectl get deployment
NAME         READY   UP-TO-DATE   AVAILABLE   AGE
php-apache   1/1     1            1           64s
labsuser@master:~$
```

2.3 Run the following command to get the SVC:
kubectl get svc

```
labsuser@master:~$ kubectl get svc
NAME         TYPE        CLUSTER-IP     EXTERNAL-IP   PORT(S)   AGE
kubernetes   ClusterIP   10.96.0.1      <none>        443/TCP   133m
php-apache   ClusterIP   10.96.172.52   <none>        80/TCP    84s
labsuser@master:~$
```

2.4 Run the **nano hpa.yaml** command to create a YAML file

```
labsuser@master:~$ nano hpa.yaml
```

2.5 Add the following code to the YAML file:

```
apiVersion: autoscaling/v1
kind: HorizontalPodAutoscaler
metadata:
  creationTimestamp: null
  name: php-apache
spec:
  maxReplicas: 10
  minReplicas: 1
  scaleTargetRef:
    apiVersion: apps/v1
    kind: Deployment
    name: php-apache
  targetCPUUtilizationPercentage: 50
status:
  currentReplicas: 0
  desiredReplicas: 0
```

```
  GNU nano 6.2                                                    hpa.yaml *
apiVersion: autoscaling/v1
kind: HorizontalPodAutoscaler
metadata:
  creationTimestamp: null
  name: php-apache
spec:
  maxReplicas: 10
  minReplicas: 1
  scaleTargetRef:
    apiVersion: apps/v1
    kind: Deployment
    name: php-apache
  targetCPUUtilizationPercentage: 50
status:
  currentReplicas: 0
  desiredReplicas: 0
```

2.6 Run the following command to create the HPA:
kubectl create -f hpa.yaml

```
labsuser@master:~$ kubectl create -f hpa.yaml
horizontalpodautoscaler.autoscaling/php-apache created
labsuser@master:~$
```

Step 3: Verify the HPA

3.1 Run the following command to verify the HPA:
kubectl get hpa

```
labsuser@master:~$ kubectl get hpa
NAME         REFERENCE                TARGETS   MINPODS   MAXPODS   REPLICAS   AGE
php-apache   Deployment/php-apache    0%/50%    1         10        1          29s
labsuser@master:~$
```

3.2 Run the following command to create a pod load generator:
kubectl run load-generator --image=busybox -- /bin/sh -c "while sleep 0.01; do wget -q -O- http://php-apache; done"

```
labsuser@master:~$ kubectl run load-generator --image=busybox -- /bin/sh -c "while sleep 0.01; do wget -q -O- http://php-apache; done"
pod/load-generator created
labsuser@master:~$
```

With this step, we verify that the HPA is reacting to CPU usage as expected and is responsive and functional.

3.3 Run the following command to delete the pod:
kubectl delete pod load-generator

```
labsuser@master:~$ kubectl delete pod load-generator
pod "load-generator" deleted
```

The pod is successfully deleted. This step is crucial to prevent any unnecessary load on your system after the test.

By following these steps, you have successfully created and configured horizontal pod autoscaling.