=================================================================================

When you are configuring HPA (horizontal pod autoscaling) then only we will get autoscaling feature

==================================================================================

Autoscaling

It is the process of increasing/decreasing the infrastructure resources based on the demand

Autoscaling mainly can done in 2 ways

-> Horizontal Scaling means increasing number of instances/servers/pods.

-> vertical Scaling means increasing capacity of single system.

HPA : Horizontal POD Autoscaling

VPA : Vertical POD Autoscaling (we don't use this)

=================================

HPA - Horizontal Pod autoscaling

=================================

It is used to scaleup or down to no of pod replicas based on the observed metrics (cpu or memory utilization)

To give the metrics to HPA we need to configure the metrics’ server

The HPA will interact with metric server

-> HPA will interact with "Metric Server" to identify CPU/Memory utilization of POD.

-> Metrics server is an application that collect metrics from objects such as pods, nodes according to the state of CPU, RAM and keeps them in time.

To get node metrics

$ kubectl top nodes

To get pod metrics

$ kubectl top pods

Note: By default, metrics server is not available in our k8s cluster

==================================

Step-1 To setup the metric server

==================================

1) clone git repo

# git clone https://github.com/suffixscope/k8s-metrics-server.git

2) check the cloned repo

# cd k8s-metrics-server

# ls deploy/1.8+/

3) apply manifest files from manifest-server directlry

# kubectl apply -f deploy/1.8+/

Note: it will create service account, role, role binding all the stuff

we can see metric server running in kube-system name space

# kubectl get all -n kube-system

check the top nodes using metric server

# kubectl top nodes

check the top pods using metric server

# kubectl top pods

If you are getting response, then you can confirm your metrics servers are running

====================================================

Step-2 To deploy a sample application in the cluster

====================================================

This manifest yml is provided by k8s for pod creating for a sample application

# vim sampleapp.yml

---

apiVersion: apps/v1

kind: Deployment

metadata:

name: hpa-demo-deployment

spec:

selector:

matchLabels:

run: hpa-demo-deployment

replicas: 1

template:

metadata:

labels:

run: hpa-demo-deployment

spec:

containers:

- name: hpa-demo-deployment

image: k8s.gcr.io/hpa-example

ports:

- containerPort: 80

resources:

limits:

cpu: 500m

requests:

cpu: 200m

...

#kubectl apply -f sampleapp.yml

=======================

Step-3 create a service

=======================

# vim serv.yml

---

apiVersion: v1

kind: Service

metadata:

name: hpa-demo-deployment

labels:

run: hpa-demo-deployment

spec:

ports:

- port: 80

selector:

run: hpa-demo-deployment

...

# kubectl apply -f serv.yml

=====================

Step-4 - create a HPA

=====================

#vim hpa.yml

---

apiVersion: autoscaling/v1

kind: HorizontalPodAutoscaler

metadata:

name: hpa-demo-deployment

spec:

scaleTargetRef:

apiVersion: apps/v1

kind: Deployment

name: hpa-demo-deployment

minReplicas: 1

maxReplicas: 10

targetCPUUtilizationPercentage: 50

...

#kubectl apply -f hpa.yml

=============================

Step-5 : Increase the Load

=============================

# kubectl get hpa

# 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-deployment; done"

Note: After executing load generator open new git bash and connect to eks vost vm and monitor hpa with below commands

# kubectl get hpa -w

we can monitor the events of my pods

# kubectl describe deploy hpa-demo-deployment

We can view

# kubectl get hpa

# kubectl get events

# kubectl top pods

# kubectl get hpa