k8s pod Auto Scaler

Pod auto scaler HPA and VPA need metric server to work as we can scale based on the node and pod resource usage so we need a metric server to get HPA and VPA to be working.

Metric Server

- Metrics Server collects resource metrics from Kubelet and exposes them in Kubernetes API server through Metrics API for use by Horizontal Pod Autoscaler and Vertical Pod Autoscaler.
- kubectl top command use Metrics API to list the resource utilization of all pods.
- Metrics Server is not meant for non-autoscaling purposes like we won’t forward these metrics
- data to monitoring tools.

STEP 1: To install all the resources related to metric server (https://github.com/kubernetes-sigs/metrics-server)

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

Note: 1. List all the pods in kube-system namespace

```
kubectl get pods -n kube-system
```

Check whether the metric-server related pod is running or not and if not running due Readiness probe failure then edit the deployment manifest and add the below command section in pod template.

```
kubectl -n kube-system edit deploy metrics-server
```

```
 containers:
- args:
- --cert-dir=/tmp
- --secure-port=4443
- --kubelet-preferred-address-types=InternalIP,ExternalIP,Hostname
- --kubelet-use-node-status-port
- --metric-resolution=15s
command:
- /metrics-server
- --kubelet-insecure-tls
- --kubelet-preferred-address-types=InternalIP
```

Horizontal autoscaler

Horizontal Pod Auto-Scaler (HPA)

- HPA is used to automatically scale the number of pods based on deployments, replicasets, statefulsets or other objects, based on CPU, Memory threshold.
- Automatic scaling of the horizontal pod does not apply to objects that cannot be scaled. ex: DaemonSets.
- We need metric server as a soruce for autoscalling.

Demo

STEP 1: create the below resources

```
Kubectl apply -f deployment.yml
Kubectl apply -f service.yml
```

STEP 2: Check the deployment

```
kubectl get pods
```

STEP 3: Create HPA

```
kubectl autoscale deployment php-apache --cpu-percent=80 --min=1 --max=4
kubectl describe hpa
```

STEP 4: Create a app to put load on php deployment

```
Kubectl apply -f load-deployment.yml
```

To check the load process

```
kubectl exec -it <load-app pod name>  -- ps
```

STEP 5: Check hpa status

```
kubectl get hpa -w
```

Note: Replicas will increase in sometime when load increases and delete the load-deployment replicas will decrease in sometime

Vertical Pod Auto-Scaler (VPA)

- vpa automatically adjusts the CPU and Memory attributes for your Pods.
  - basically vpa will recreate your pod with the suitable CPU and Memory attributes.
  - when we describe vpa, it will show recommendations for the Memory/CPU requests, Limits and it can also automatically update the limits.

```
apiVersion: autoscaling.k8s.io/v1
kind: VerticalPodAutoscaler
metadata:
  name: my-app-vpa
spec:
  targetRef:
	apiVersion: "apps/v1"
	kind:       Deployment
	name:       my-app
  updatePolicy:
	updateMode: "Auto"
```

Horizontal / Vertical Cluster Auto-Scaler

- Cluster Autoscaler is a tool that automatically adjusts the size of the Kubernetes cluster when one of the following conditions is true:
  1. some pods failed to run in the cluster due to insufficient resources,
  2. some nodes in the cluster that have been overloaded for an extended period and their pods can be placed on other existing nodes.

- Cluster autoscaler tools are mostly provided by public cloud providers.