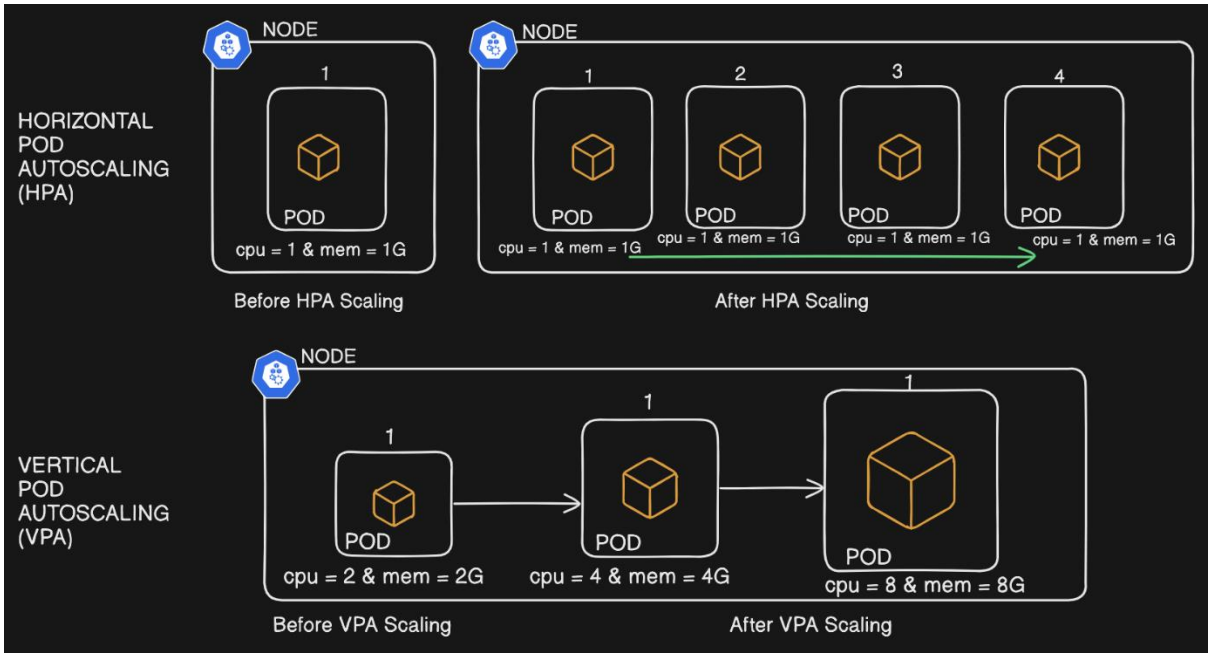
HORIZONTAL POD AUTOSCALER (HPA) & VERTICAL POD AUTOSCALER (VPA) IN KUBERNETES

HPA (Horizontal Pod Autoscaler) and **VPA (Vertical Pod Autoscaler)** are mechanisms that manage the scaling of resources for applications running within the cluster.



1. HPA - Horizontal Pod Autoscaler

- **Purpose**: HPA automatically adjusts the number of pod replicas in a deployment, replicaset, or statefulset based on observed CPU, memory usage, or other custom metrics.
- **Usage**: Best suited for stateless applications where multiple instances of the application can run simultaneously without data synchronization issues.

➢ **When to Use**:

✓ When you expect varying loads and want to handle traffic surges.

✓ For applications where scaling horizontally (adding more pods) is easier than increasing resources of a single pod.

➢ **Advantages**:

✓ Increased resilience and load balancing since multiple pods can share the workload.

✓ Ideal for applications with distributed or stateless architectures.

2. VPA - Vertical Pod Autoscaler

- **Purpose**: VPA automatically adjusts the resource requests and limits (CPU, memory) of existing pods based on observed usage patterns, thereby "scaling up" the resource capabilities of each pod individually.
- **Usage**: Ideal for stateful applications where a single instance with more resources is more efficient than multiple replicas.

➢ **When to Use**:

✓ For applications where horizontal scaling isn't feasible, such as those requiring shared storage or maintaining a unique state.

HORIZONTAL POD AUTOSCALER (HPA) & VERTICAL POD AUTOSCALER (VPA) IN KUBERNETES

- ✓ For workloads that require more memory or CPU over time but don't need additional replicas.

➢ **Advantages**:

- ✓ Can help reduce resource underutilization by right-sizing pods.
- ✓ Prevents resource overcommitment by adjusting limits based on actual usage patterns.

Which One is Better: HPA or VPA?

- **Use Cases**: HPA is generally better for **stateless** and **scalable applications**, while VPA is beneficial for stateful applications or those that benefit from increasing the power of individual instances rather than adding new ones.
- **Combined Approach**: Often, HPA and VPA can be used together. HPA handles the number of replicas, while VPA ensures each pod has the correct resources. However, their combined use should be carefully tested, as scaling horizontally while also changing individual pod sizes can introduce complexity.

Summary

- Use **HPA** for workloads with fluctuating traffic and load that can be distributed across multiple instances.
- Use **VPA** for applications that benefit from increased resources per pod rather than additional replicas.

Hands-on (HPA)

Deployment and Service manifest

autoscaling > ! deploy.yaml

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: php-apache
spec:
  selector:
    matchLabels:
      run: php-apache
  template:
    metadata:
      labels:
        run: php-apache
    spec:
      containers:
      - name: php-apache
        image: registry.k8s.io/hpa-example
        ports:
        - containerPort: 80
        resources:
          limits:
            cpu: 500m
          requests:
            cpu: 200m
---
apiVersion: v1
kind: Service
metadata:
  name: php-apache
  labels:
    run: php-apache
spec:
  ports:
  - port: 80
  selector:
    run: php-apache


#to increase the load in CPU run this command, you will get this command in k8's doc --> hpa --> got to bottom --> example --> find increase the load on cpu
#kubectl run -i --tty load-generator --rm --image=busybox:1.28 --restart=Never -- /bin/sh -c "while sleep 0.01; do wget -q -O- http://php-apache; done"

```

HORIZONTAL POD AUTOSCALER (HPA) & VERTICAL POD AUTOSCALER (VPA) IN KUBERNETES

Horizontal pod autoscaling manifest, where ever the CPU utilization cross 50% new pod will be created.

autoscaling > ! scale.yaml

```yaml
apiVersion: autoscaling/v2
kind: HorizontalPodAutoscaler
metadata:
  name: php-apache
spec:
  scaleTargetRef:
    apiVersion: apps/v1
    kind: Deployment
    name: php-apache
  minReplicas: 1
  maxReplicas: 10
  metrics:
  - type: Resource
    resource:
      name: cpu
      target:
        type: Utilization
        averageUtilization: 50
```

```
manoj -->
manoj -->
manoj -->kubectl get pods
No resources found in default namespace.
manoj -->
manoj -->kubectl apply -f deploy.yaml
deployment.apps/php-apache created
service/php-apache created
manoj -->
manoj -->kubectl get pods
NAME                          READY   STATUS    RESTARTS   AGE
php-apache-d87b7ff46-bxk7d    1/1     Running   0          6s
manoj -->
manoj -->kubectl get svc
NAME         TYPE        CLUSTER-IP     EXTERNAL-IP   PORT(S)   AGE
kubernetes   ClusterIP   10.96.0.1      <none>        443/TCP   27d
php-apache   ClusterIP   10.96.216.60   <none>        80/TCP    17s
manoj -->
manoj -->
```

pod and service running

HPA Object created

```
manoj -->
manoj -->
manoj -->kubectl autoscale deploy php-apache --cpu-percent=50 --min=1 --max=10
horizontalpodautoscaler.autoscaling/php-apache autoscaled
manoj -->
manoj -->kubectl get hpa
NAME         REFERENCE               TARGETS              MINPODS   MAXPODS   REPLICAS   AGE
php-apache   Deployment/php-apache   cpu: <unknown>/50%   1         10        0          11s
manoj -->
manoj -->kubectl get hpa
NAME         REFERENCE               TARGETS       MINPODS   MAXPODS   REPLICAS   AGE
php-apache   Deployment/php-apache   cpu: 0%/50%   1         10        1          28s
manoj -->
manoj -->kubectl get hpa
NAME         REFERENCE               TARGETS       MINPODS   MAXPODS   REPLICAS   AGE
php-apache   Deployment/php-apache   cpu: 0%/50%   1         10        1          41s
manoj -->
manoj -->
```

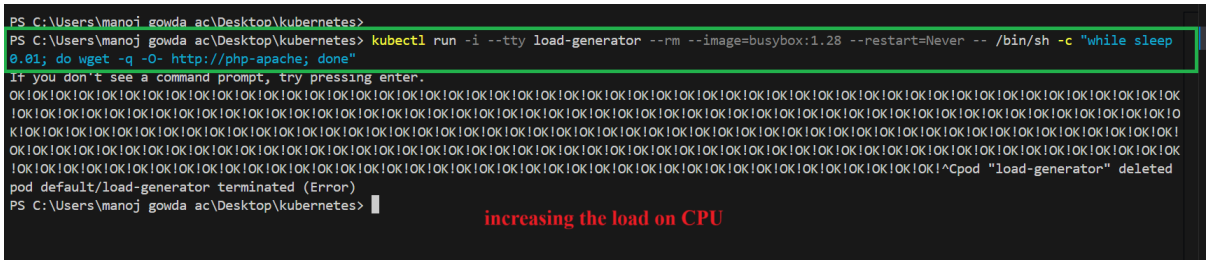
HPA object created for deployment

min of 1 pod, max of 10 pod will be create when load on CPU increases above 50%

after the load, CPU load decreases to below 50% then replicas will downsize to min of 1 pod

HORIZONTAL POD AUTOSCALER (HPA) & VERTICAL POD AUTOSCALER (VPA) IN KUBERNETES

Applying stress to increase the load on CPU



We can see when CPU utilization crossed 50% new pod got created as we specified in HPA manifest.