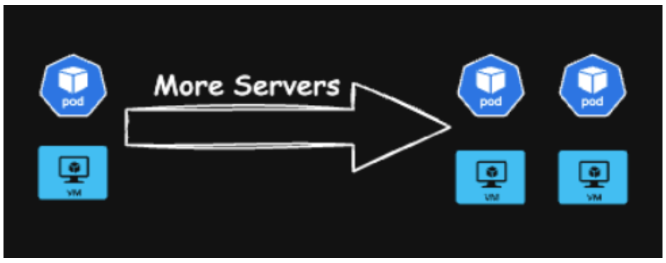
KUBERNETES HPA & VPA CONCEPTS

SCALING: Adjusting the resources based on varying load on application.

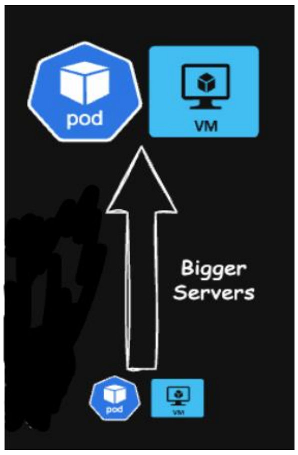
Pods can be scaled in following 2 ways in K8s:

1) **Horizontal Scaling**: Increasing or Decreasing the number of pods or number of VMs based on application load.
   - **Manual Scaling:** By changing replicas count in deployment.yaml
   - **Automatic Scaling:** By using Horizontal pods Autoscaler



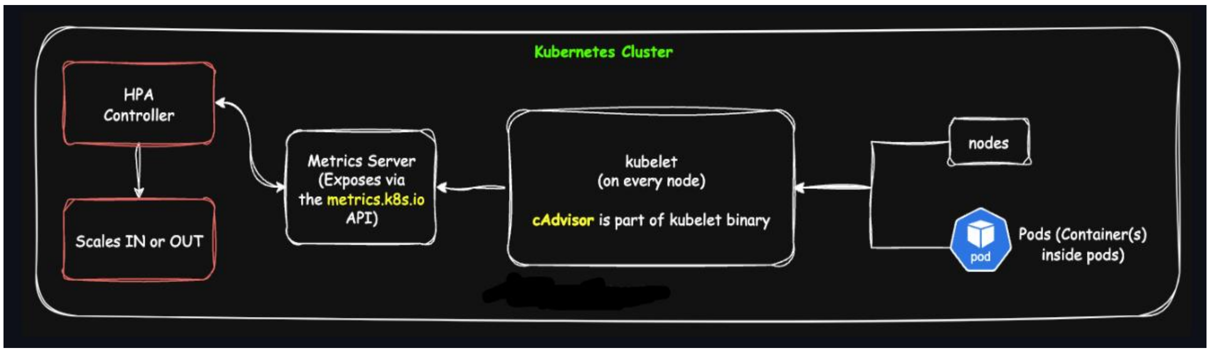
Horizontal Scaling

2) **Vertical Scaling**: Increasing or decreasing the resources(CPU, Memory etc) of the same pod or VM based on traffic to application.
   - **Manual Scaling:** By changing container requests and limits in deployment.yaml
   - **Automatic Scaling:** By using Vertical pods Autoscaler



Vertical Scaling

1) Horizontal Pods Autoscaler(HPA): It is a Kubernetes object that automatically **scales out(increase)** or **sclaes in(decrease)** the number of pods based on Resource usage or Custom metrics. It runs as a control loop, checking metrics every 15 seconds.


Kubernetes Cluster
HPA Controller
Scales IN or OUT
Metrics Server (Exposes via the metrics.k8s.io API)
kubelet (on every node)
cAdvisor is part of kubelet binary
nodes
pod
Pods (Container(s) inside pods)
[redacted]

Flow with Default Resource Metrics (CPU & Memory)

Note: Metric Server is the pre-requisite for HPA

Practical:

Deployment.yaml:

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deploy
spec:
  replicas: 1
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx
        resources:
          requests:
            cpu: "100m"
          limits:
            cpu: "200m"
        ports:
        - containerPort: 80
```

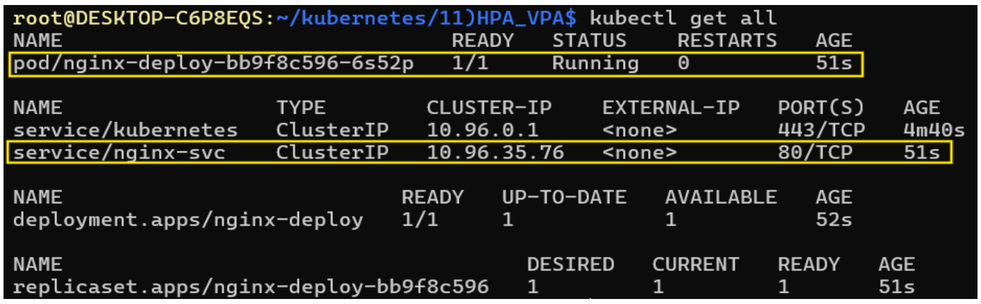
Service .yaml:

```
apiVersion: v1
kind: Service
metadata:
  name: nginx-svc
spec:
  selector:
    app: nginx
  ports:
  - protocol: TCP
    port: 80
    targetPort: 80
  type: ClusterIP
```

Apply;

```
kubectl apply -f nginx-deployment.yaml
kubectl apply -f nginx-Service .yaml
```

Verify;

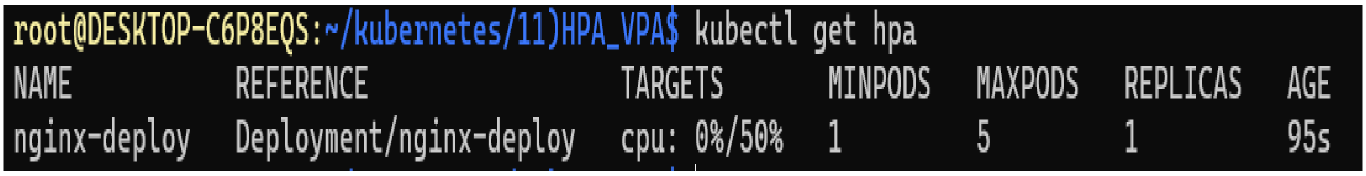


```
root@DESKTOP-C6P8EQS:~/kubernetes/11)HPA_VPA$ kubectl get all
NAME                                 READY   STATUS    RESTARTS   AGE
pod/nginx-deploy-bb9f8c596-6s52p     1/1     Running   0          51s

NAME                     TYPE        CLUSTER-IP    EXTERNAL-IP   PORT(S)   AGE
service/kubernetes       ClusterIP   10.96.0.1     <none>        443/TCP   4m40s
service/nginx-svc        ClusterIP   10.96.35.76   <none>        80/TCP    51s

NAME                           READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/nginx-deploy   1/1     1            1           52s

NAME                                     DESIRED   CURRENT   READY   AGE
replicaset.apps/nginx-deploy-bb9f8c596   1         1         1       51s
```

Lets create a HPA object:

```
kubectl autoscale deployment nginx-deploy --
cpu='50%' --min=1 --max=5
```

It states that at least 1 replica must run, and if CPU usage exceeds 50% of the request, scale up to a maximum of 5 replicas.



```
root@DESKTOP-C6P8EQS:~/kubernetes/11)HPA_VPA$ kubectl get hpa
NAME           REFERENCE                 TARGETS       MINPODS   MAXPODS   REPLICAS   AGE
nginx-deploy   Deployment/nginx-deploy   cpu: 0%/50%   1         5         1          95s
```

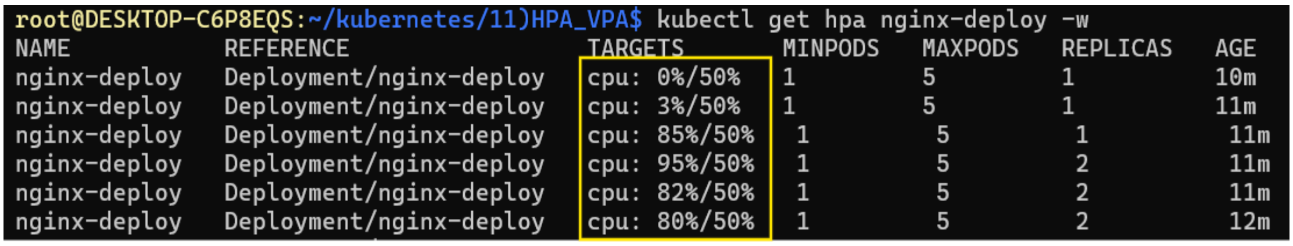
Lets apply some CPU load on the pods using below command from 2 different tabs:

```
kubectl run -it --rm load-generator-1 --image=busybox -- /bin/sh -c "while true; do wget -q -O- http://nginx-svc; done"
```

Behavior:

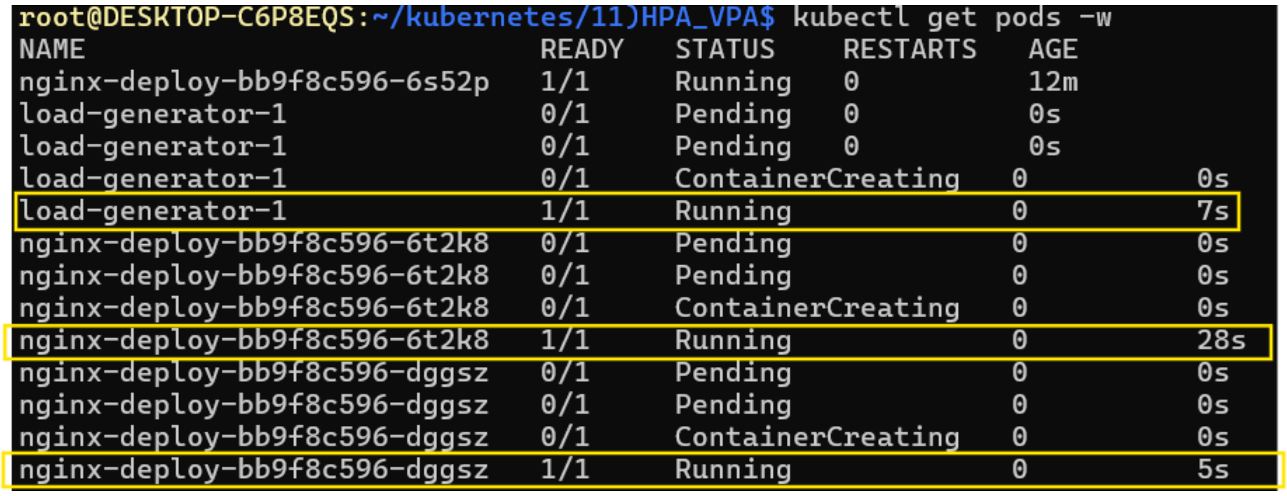
- As load increases, CPU utilization rises.
- HPA will scale from 1 pod up to max 5 if needed.
- Once load decreases (stop load-generator), HPA scales back down.

CPU Utilization:



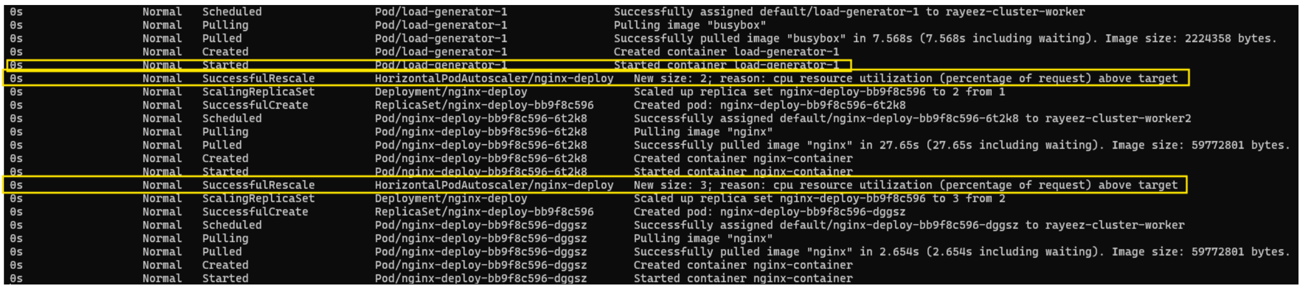
```
root@DESKTOP-C6P8EQS:~/kubernetes/11)HPA_VPA$ kubectl get hpa nginx-deploy -w
NAME           REFERENCE                 TARGETS       MINPODS   MAXPODS   REPLICAS   AGE
nginx-deploy   Deployment/nginx-deploy   cpu: 0%/50%   1         5         1          10m
nginx-deploy   Deployment/nginx-deploy   cpu: 3%/50%   1         5         1          11m
nginx-deploy   Deployment/nginx-deploy   cpu: 85%/50%   1         5         1          11m
nginx-deploy   Deployment/nginx-deploy   cpu: 95%/50%   1         5         2          11m
nginx-deploy   Deployment/nginx-deploy   cpu: 82%/50%   1         5         2          11m
nginx-deploy   Deployment/nginx-deploy   cpu: 80%/50%   1         5         2          12m
```

Scaling out of Replicas:



```
root@DESKTOP-C6P8EQS:~/kubernetes/11)HPA_VPA$ kubectl get pods -w
NAME                           READY   STATUS    RESTARTS   AGE
nginx-deploy-bb9f8c596-6s52p   1/1     Running   0          12m
load-generator-1               0/1     Pending   0          0s
load-generator-1               0/1     Pending   0          0s
load-generator-1               0/1     ContainerCreating   0          0s
load-generator-1               1/1     Running             0          7s
nginx-deploy-bb9f8c596-6t2k8   0/1     Pending             0          0s
nginx-deploy-bb9f8c596-6t2k8   0/1     Pending             0          0s
nginx-deploy-bb9f8c596-6t2k8   0/1     ContainerCreating   0          0s
nginx-deploy-bb9f8c596-6t2k8   1/1     Running             0          28s
nginx-deploy-bb9f8c596-dggsz   0/1     Pending             0          0s
nginx-deploy-bb9f8c596-dggsz   0/1     Pending             0          0s
nginx-deploy-bb9f8c596-dggsz   0/1     ContainerCreating   0          0s
nginx-deploy-bb9f8c596-dggsz   1/1     Running             0          5s
```

Kubectl Events:



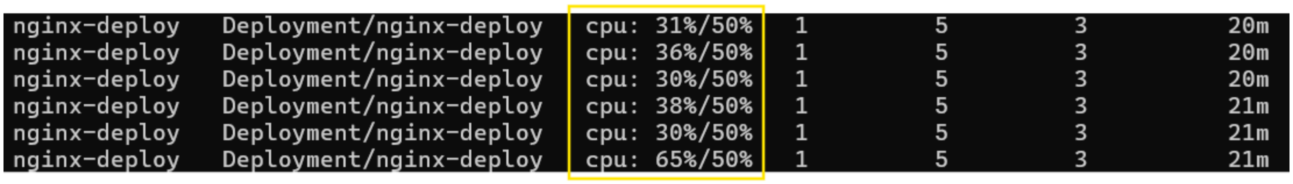
```
0s          Normal   Scheduled             Pod/load-generator-1                        Successfully assigned default/load-generator-1 to rayeez-cluster-worker
0s          Normal   Pulling               Pod/load-generator-1                        Pulling image "busybox"
0s          Normal   Pulled                Pod/load-generator-1                        Successfully pulled image "busybox" in 7.568s (7.568s including waiting). Image size: 2224358 bytes.
0s          Normal   Created               Pod/load-generator-1                        Created container load-generator-1
0s          Normal   Started               Pod/load-generator-1                        Started container load-generator-1
0s          Normal   SuccessfulRescale     HorizontalPodAutoscaler/nginx-deploy   New size: 2; reason: cpu resource utilization (percentage of request) above target
0s          Normal   ScalingReplicaSet     Deployment/nginx-deploy                  Scaled up replica set nginx-deploy-bb9f8c596 to 2 from 1
0s          Normal   SuccessfulCreate      ReplicaSet/nginx-deploy-bb9f8c596        Created pod: nginx-deploy-bb9f8c596-6t2k8
0s          Normal   Scheduled             Pod/nginx-deploy-bb9f8c596-6t2k8         Successfully assigned default/nginx-deploy-bb9f8c596-6t2k8 to rayeez-cluster-worker2
0s          Normal   Pulling               Pod/nginx-deploy-bb9f8c596-6t2k8         Pulling image "nginx"
0s          Normal   Pulled                Pod/nginx-deploy-bb9f8c596-6t2k8         Successfully pulled image "nginx" in 27.65s (27.65s including waiting). Image size: 59772801 bytes.
0s          Normal   Created               Pod/nginx-deploy-bb9f8c596-6t2k8         Created container nginx-container
0s          Normal   Started               Pod/nginx-deploy-bb9f8c596-6t2k8         Started container nginx-container
0s          Normal   SuccessfulRescale     HorizontalPodAutoscaler/nginx-deploy   New size: 3; reason: cpu resource utilization (percentage of request) above target
0s          Normal   ScalingReplicaSet     Deployment/nginx-deploy                  Scaled up replica set nginx-deploy-bb9f8c596 to 3 from 2
0s          Normal   SuccessfulCreate      ReplicaSet/nginx-deploy-bb9f8c596        Created pod: nginx-deploy-bb9f8c596-dggsz
0s          Normal   Scheduled             Pod/nginx-deploy-bb9f8c596-dggsz         Successfully assigned default/nginx-deploy-bb9f8c596-dggsz to rayeez-cluster-worker
0s          Normal   Pulling               Pod/nginx-deploy-bb9f8c596-dggsz         Pulling image "nginx"
0s          Normal   Pulled                Pod/nginx-deploy-bb9f8c596-dggsz         Successfully pulled image "nginx" in 2.654s (2.654s including waiting). Image size: 59772801 bytes.
0s          Normal   Created               Pod/nginx-deploy-bb9f8c596-dggsz         Created container nginx-container
0s          Normal   Started               Pod/nginx-deploy-bb9f8c596-dggsz         Started container nginx-container
```

Haseeb Ullah

Applied Load generator-2;

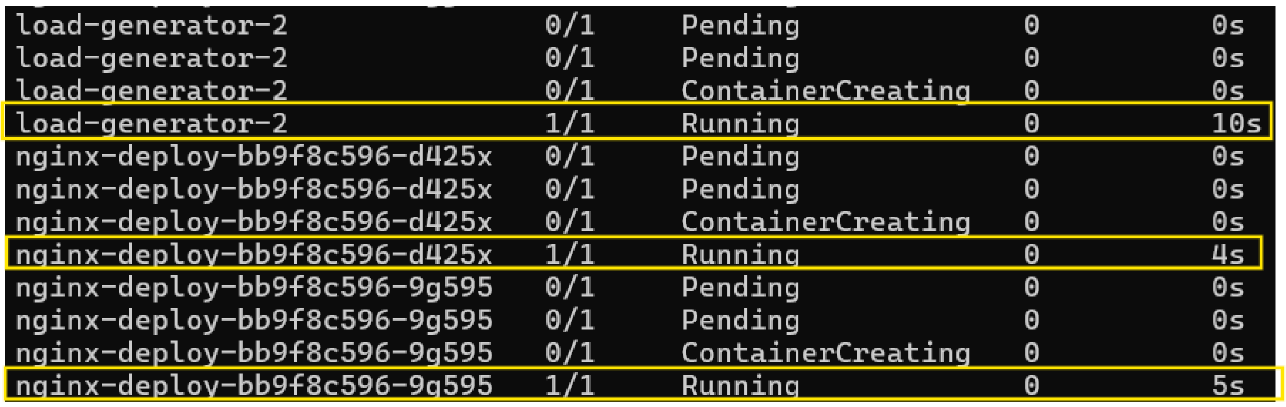
```
kubectl run -it --rm load-generator-2 --
image=busybox -- /bin/sh -c "while true; do
wget -q -O- http://nginx-s
vc; done"
```

CPU Utilization:



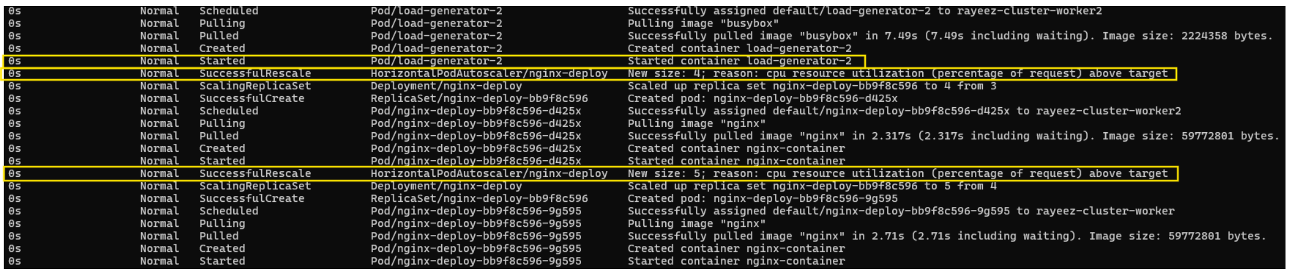
```
nginx-deploy   Deployment/nginx-deploy   cpu: 31%/50%   1   5   3   20m
nginx-deploy   Deployment/nginx-deploy   cpu: 36%/50%   1   5   3   20m
nginx-deploy   Deployment/nginx-deploy   cpu: 30%/50%   1   5   3   20m
nginx-deploy   Deployment/nginx-deploy   cpu: 38%/50%   1   5   3   21m
nginx-deploy   Deployment/nginx-deploy   cpu: 30%/50%   1   5   3   21m
nginx-deploy   Deployment/nginx-deploy   cpu: 65%/50%   1   5   3   21m
```

Scaling out of Replicas:



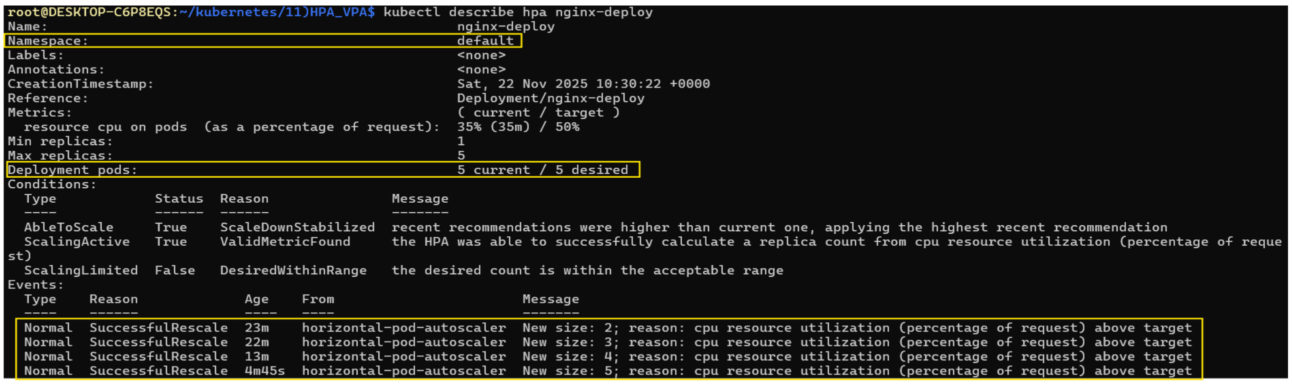
```
load-generator-2                0/1   Pending             0   0s
load-generator-2                0/1   Pending             0   0s
load-generator-2                0/1   ContainerCreating   0   0s
load-generator-2                1/1   Running             0   10s
nginx-deploy-bb9f8c596-d425x    0/1   Pending             0   0s
nginx-deploy-bb9f8c596-d425x    0/1   Pending             0   0s
nginx-deploy-bb9f8c596-d425x    0/1   ContainerCreating   0   0s
nginx-deploy-bb9f8c596-d425x    1/1   Running             0   4s
nginx-deploy-bb9f8c596-9g595    0/1   Pending             0   0s
nginx-deploy-bb9f8c596-9g595    0/1   Pending             0   0s
nginx-deploy-bb9f8c596-9g595    0/1   ContainerCreating   0   0s
nginx-deploy-bb9f8c596-9g595    1/1   Running             0   5s
```

Kubectl Events:



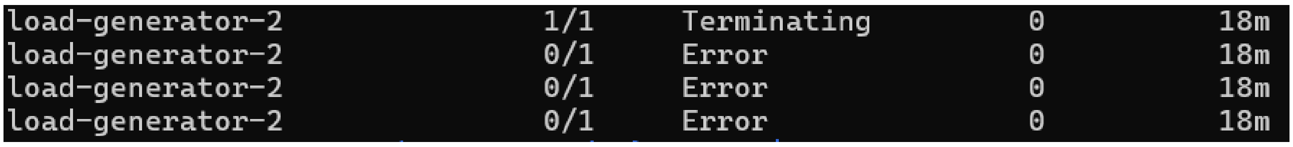
```
0s   Normal   Scheduled            Pod/load-generator-2                        Successfully assigned default/load-generator-2 to rayeez-cluster-worker2
0s   Normal   Pulling              Pod/load-generator-2                        Pulling image "busybox"
0s   Normal   Pulled               Pod/load-generator-2                        Successfully pulled image "busybox" in 7.49s (7.49s including waiting). Image size: 2224358 bytes.
0s   Normal   Created              Pod/load-generator-2                        Created container load-generator-2
0s   Normal   Started              Pod/load-generator-2                        Started container load-generator-2
0s   Normal   SuccessfulRescale    HorizontalPodAutoscaler/nginx-deploy        New size: 4; reason: cpu resource utilization (percentage of request) above target
0s   Normal   ScalingReplicaSet    Deployment/nginx-deploy                     Scaled up replica set nginx-deploy-bb9f8c596 to 4 from 3
0s   Normal   SuccessfulCreate     ReplicaSet/nginx-deploy-bb9f8c596           Created pod: nginx-deploy-bb9f8c596-d425x
0s   Normal   Scheduled            Pod/nginx-deploy-bb9f8c596-d425x            Successfully assigned default/nginx-deploy-bb9f8c596-d425x to rayeez-cluster-worker2
0s   Normal   Pulling              Pod/nginx-deploy-bb9f8c596-d425x            Pulling image "nginx"
0s   Normal   Pulled               Pod/nginx-deploy-bb9f8c596-d425x            Successfully pulled image "nginx" in 2.317s (2.317s including waiting). Image size: 59772801 bytes.
0s   Normal   Created              Pod/nginx-deploy-bb9f8c596-d425x            Created container nginx-container
0s   Normal   Started              Pod/nginx-deploy-bb9f8c596-d425x            Started container nginx-container
0s   Normal   SuccessfulRescale    HorizontalPodAutoscaler/nginx-deploy        New size: 5; reason: cpu resource utilization (percentage of request) above target
0s   Normal   ScalingReplicaSet    Deployment/nginx-deploy                     Scaled up replica set nginx-deploy-bb9f8c596 to 5 from 4
0s   Normal   SuccessfulCreate     ReplicaSet/nginx-deploy-bb9f8c596           Created pod: nginx-deploy-bb9f8c596-9g595
0s   Normal   Scheduled            Pod/nginx-deploy-bb9f8c596-9g595            Successfully assigned default/nginx-deploy-bb9f8c596-9g595 to rayeez-cluster-worker
0s   Normal   Pulling              Pod/nginx-deploy-bb9f8c596-9g595            Pulling image "nginx"
0s   Normal   Pulled               Pod/nginx-deploy-bb9f8c596-9g595            Successfully pulled image "nginx" in 2.71s (2.71s including waiting). Image size: 59772801 bytes.
0s   Normal   Created              Pod/nginx-deploy-bb9f8c596-9g595            Created container nginx-container
0s   Normal   Started              Pod/nginx-deploy-bb9f8c596-9g595            Started container nginx-container
```

```
Kubectl describe hpa nginx-deploy
```



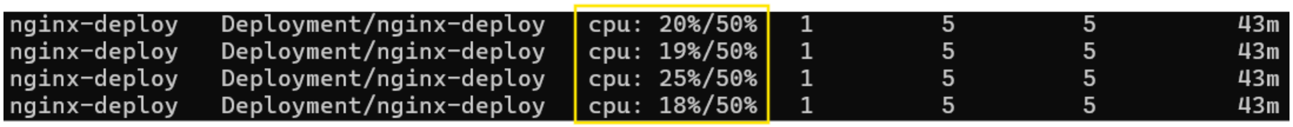
```
root@DESKTOP-C6P8EQS:~/kubernetes/11)HPA_VPA$ kubectl describe hpa nginx-deploy
Name:                                                  nginx-deploy
Namespace:                                             default
Labels:                                                <none>
Annotations:                                           <none>
CreationTimestamp:                                     Sat, 22 Nov 2025 10:30:22 +0000
Reference:                                             Deployment/nginx-deploy
Metrics:                                               ( current / target )
  resource cpu on pods  (as a percentage of request):  35% (35m) / 50%
Min replicas:                                          1
Max replicas:                                          5
Deployment pods:                                       5 current / 5 desired
Conditions:
  Type            Status  Reason                 Message
  ----            ------  ------                 -------
  AbleToScale     True    ScaleDownStabilized    recent recommendations were higher than current one, applying the highest recent recommendation
  ScalingActive   True    ValidMetricFound       the HPA was able to successfully calculate a replica count from cpu resource utilization (percentage of reque
st)
  ScalingLimited  False   DesiredWithinRange     the desired count is within the acceptable range
Events:
  Type    Reason             Age    From                       Message
  ----    ------             ----   ----                       -------
  Normal  SuccessfulRescale  23m    horizontal-pod-autoscaler  New size: 2; reason: cpu resource utilization (percentage of request) above target
  Normal  SuccessfulRescale  22m    horizontal-pod-autoscaler  New size: 3; reason: cpu resource utilization (percentage of request) above target
  Normal  SuccessfulRescale  13m    horizontal-pod-autoscaler  New size: 4; reason: cpu resource utilization (percentage of request) above target
  Normal  SuccessfulRescale  4m45s  horizontal-pod-autoscaler  New size: 5; reason: cpu resource utilization (percentage of request) above target
```

Lets Terminate Load generator-2 and observe scaling in behaviour;



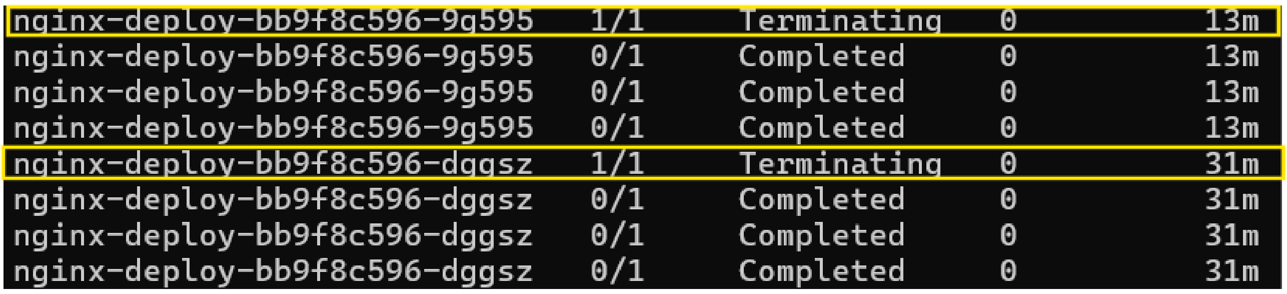
```
load-generator-2                  1/1     Terminating          0            18m
load-generator-2                  0/1     Error                0            18m
load-generator-2                  0/1     Error                0            18m
load-generator-2                  0/1     Error                0            18m
```

CPU Utilization:



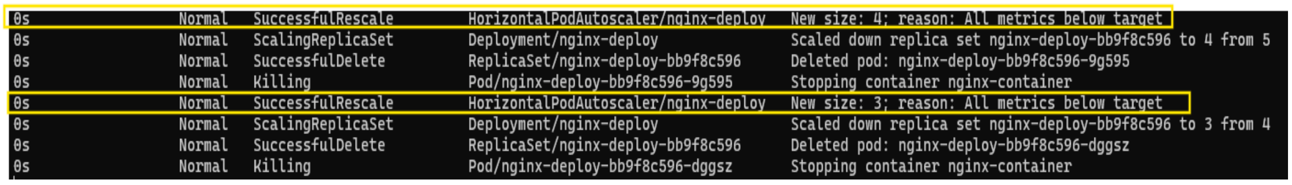
```
nginx-deploy   Deployment/nginx-deploy   cpu: 20%/50%   1    5    5    43m
nginx-deploy   Deployment/nginx-deploy   cpu: 19%/50%   1    5    5    43m
nginx-deploy   Deployment/nginx-deploy   cpu: 25%/50%   1    5    5    43m
nginx-deploy   Deployment/nginx-deploy   cpu: 18%/50%   1    5    5    43m
```

Scaling in of Replicas:



```
nginx-deploy-bb9f8c596-9g595   1/1   Terminating   0        13m
nginx-deploy-bb9f8c596-9g595   0/1   Completed     0        13m
nginx-deploy-bb9f8c596-9g595   0/1   Completed     0        13m
nginx-deploy-bb9f8c596-9g595   0/1   Completed     0        13m
nginx-deploy-bb9f8c596-dggsz   1/1   Terminating   0        31m
nginx-deploy-bb9f8c596-dggsz   0/1   Completed     0        31m
nginx-deploy-bb9f8c596-dggsz   0/1   Completed     0        31m
nginx-deploy-bb9f8c596-dggsz   0/1   Completed     0        31m
```

Kubectl Events:



```
0s          Normal   SuccessfulRescale   HorizontalPodAutoscaler/nginx-deploy   New size: 4; reason: All metrics below target
0s          Normal   ScalingReplicaSet   Deployment/nginx-deploy                Scaled down replica set nginx-deploy-bb9f8c596 to 4 from 5
0s          Normal   SuccessfulDelete    ReplicaSet/nginx-deploy-bb9f8c596      Deleted pod: nginx-deploy-bb9f8c596-9g595
0s          Normal   Killing             Pod/nginx-deploy-bb9f8c596-9g595       Stopping container nginx-container
0s          Normal   SuccessfulRescale   HorizontalPodAutoscaler/nginx-deploy   New size: 3; reason: All metrics below target
0s          Normal   ScalingReplicaSet   Deployment/nginx-deploy                Scaled down replica set nginx-deploy-bb9f8c596 to 3 from 4
0s          Normal   SuccessfulDelete    ReplicaSet/nginx-deploy-bb9f8c596      Deleted pod: nginx-deploy-bb9f8c596-dggsz
0s          Normal   Killing             Pod/nginx-deploy-bb9f8c596-dggsz       Stopping container nginx-container
```

Conclusion:

HPA automatically scales out and scales in the number of replicas on the cluster based on load/traffic received by the application.

2) Vertical Pods Autoscaler(VPA):

It automatically adjust the **CPU** and **memory requests** and **limits** of pods based on their actual usage.

- It increases or decreases the CPU & memory reservations of pods.
- Pods always have enough resources to perform efficiently.
- Wasted resources are minimized, avoiding over-provisioning.
- Removes guesswork. No need to manually tune CPU/memory requests.

Haseeb Ullah

Prequisite: VPA recommender, Updater, Admission controller and metric Server.

Practical:

deployment.yaml:

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deploy
spec:
  replicas: 2
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx
        resources:
          requests:
            cpu: "100m"
          limits:
            cpu: "200m"
        ports:
        - containerPort: 80
```

Service.yaml:

```
apiVersion: v1
kind: Service
metadata:
  name: nginx-svc
spec:
  selector:
    app: nginx
  ports:
  - protocol: TCP
    port: 80
    targetPort: 80
  type: ClusterIP
```

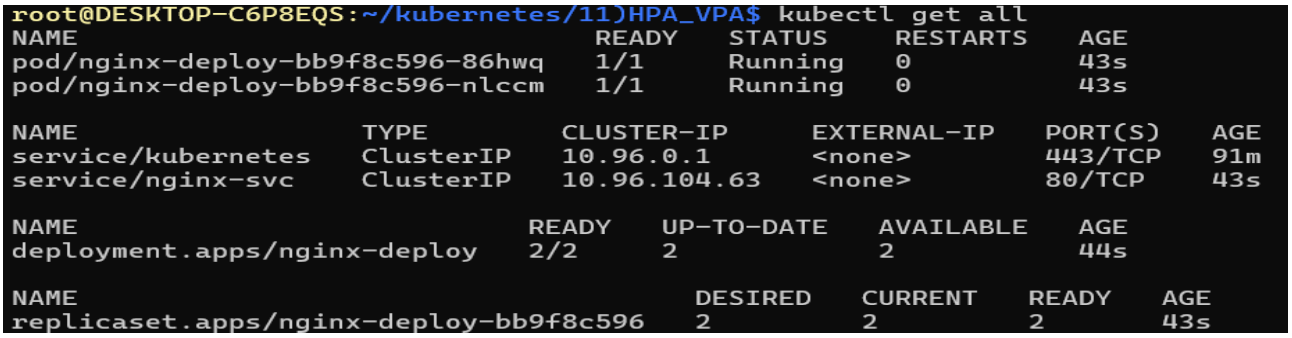
vpa.yaml:

```
apiVersion: autoscaling.k8s.io/v1
kind: VerticalPodAutoscaler
metadata:
  name: nginx-vpa
spec:
  targetRef:
    apiVersion: "apps/v1"
    kind:       Deployment
    name:       nginx-deploy
  updatePolicy:
    # updateMode options:
    # "Off"     - VPA only recommends
resources, does NOT apply them.
    # "Initial" - VPA sets recommended
resources at pod creation, no changes after.
    # "Auto"    - VPA automatically updates
resources and restarts pods as needed.
    updateMode: "Auto"
```

Apply;

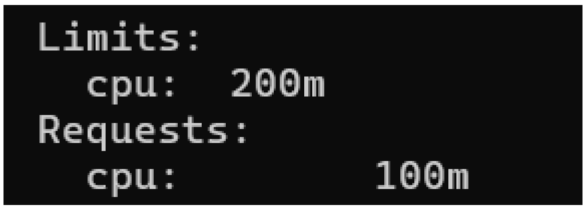
```
kubectl apply -f nginx-deployment.yaml
kubectl apply -f nginx-Service .yaml
```

Verify;



```
root@DESKTOP-C6P8EQS:~/kubernetes/11)HPA_VPA$ kubectl get all
NAME                               READY   STATUS    RESTARTS   AGE
pod/nginx-deploy-bb9f8c596-86hwq   1/1     Running   0          43s
pod/nginx-deploy-bb9f8c596-nlccm   1/1     Running   0          43s

NAME                 TYPE        CLUSTER-IP     EXTERNAL-IP   PORT(S)   AGE
service/kubernetes   ClusterIP   10.96.0.1      <none>        443/TCP   91m
service/nginx-svc    ClusterIP   10.96.104.63   <none>        80/TCP    43s

NAME                           READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/nginx-deploy   2/2     2            2           44s

NAME                                     DESIRED   CURRENT   READY   AGE
replicaset.apps/nginx-deploy-bb9f8c596   2         2         2       43s
```

We have set this CPU request and limit in yaml manifest:

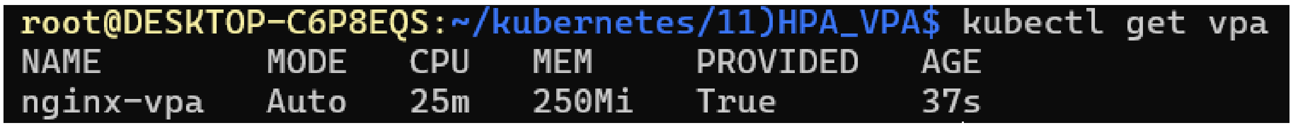


```
Limits:
  cpu:  200m
Requests:
  cpu:        100m
```

Lets create VPA and observe its behaviour;

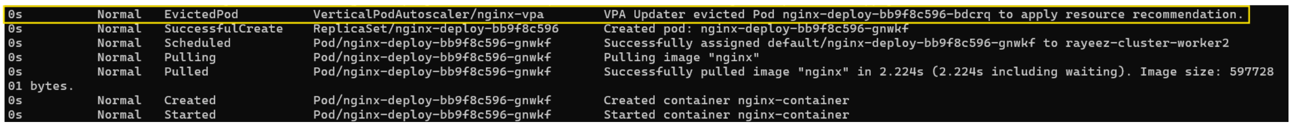
```
kubectl apply -f vpa.yaml
```

Verify;



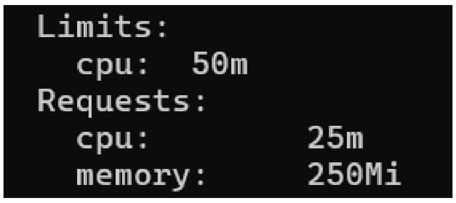
```
root@DESKTOP-C6P8EQS:~/kubernetes/11)HPA_VPA$ kubectl get vpa
NAME        MODE   CPU   MEM     PROVIDED   AGE
nginx-vpa   Auto   25m   250Mi   True       37s
```

The moment we have created VPA; This event took place:



```
0s          Normal   EvictedPod          VerticalPodAutoscaler/nginx-vpa        VPA Updater evicted Pod nginx-deploy-bb9f8c596-bdcrq to apply resource recommendation.
0s          Normal   SuccessfulCreate    ReplicaSet/nginx-deploy-bb9f8c596      Created pod: nginx-deploy-bb9f8c596-gnwkf
0s          Normal   Scheduled           Pod/nginx-deploy-bb9f8c596-gnwkf       Successfully assigned default/nginx-deploy-bb9f8c596-gnwkf to rayeez-cluster-worker2
0s          Normal   Pulling             Pod/nginx-deploy-bb9f8c596-gnwkf       Pulling image "nginx"
0s          Normal   Pulled              Pod/nginx-deploy-bb9f8c596-gnwkf       Successfully pulled image "nginx" in 2.224s (2.224s including waiting). Image size: 597728
01 bytes.
0s          Normal   Created             Pod/nginx-deploy-bb9f8c596-gnwkf       Created container nginx-container
0s          Normal   Started             Pod/nginx-deploy-bb9f8c596-gnwkf       Started container nginx-container
```

Now the CPU requests and limits are adjusted to following values by VPA.

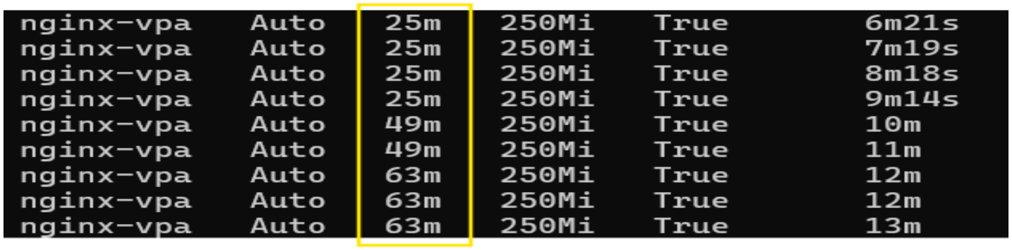


```
Limits:
  cpu:  50m
Requests:
  cpu:        25m
  memory:     250Mi
```

Lets apply some CPU load on the pods using below command from 2 different tabs:

```
kubectl run -it --rm load-generator-1 --image=busybox -- /bin/sh -c "while true; do wget -q -O- http://nginx-svc; done"
```

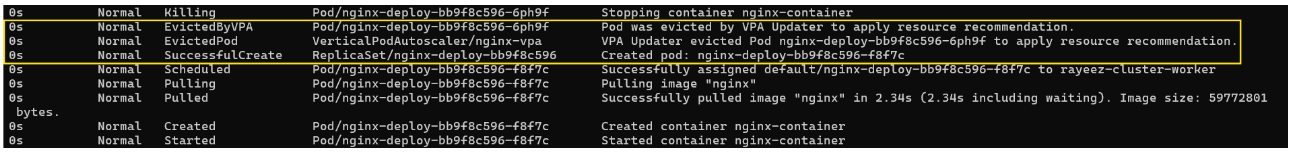
Now Because of increasing CPU load on pods, VPA has set new CPU request, limits values to pods:



```
nginx-vpa   Auto   25m   250Mi   True       6m21s
nginx-vpa   Auto   25m   250Mi   True       7m19s
nginx-vpa   Auto   25m   250Mi   True       8m18s
nginx-vpa   Auto   25m   250Mi   True       9m14s
nginx-vpa   Auto   49m   250Mi   True       10m
nginx-vpa   Auto   49m   250Mi   True       11m
nginx-vpa   Auto   63m   250Mi   True       12m
nginx-vpa   Auto   63m   250Mi   True       12m
nginx-vpa   Auto   63m   250Mi   True       13m
```

Haseeb Ullah

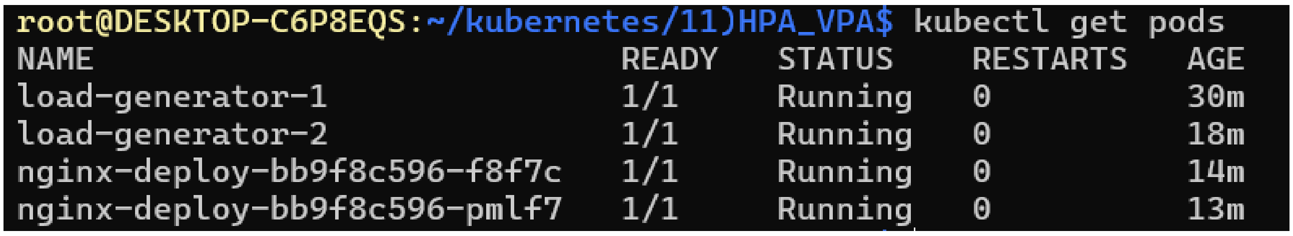
Kubectl Events:



```
0s          Normal   Killing              Pod/nginx-deploy-bb9f8c596-6ph9f       Stopping container nginx-container
0s          Normal   EvictedByVPA         Pod/nginx-deploy-bb9f8c596-6ph9f       Pod was evicted by VPA Updater to apply resource recommendation.
0s          Normal   EvictedPod           VerticalPodAutoscaler/nginx-vpa        VPA Updater evicted Pod nginx-deploy-bb9f8c596-6ph9f to apply resource recommendation.
0s          Normal   SuccessfulCreate     ReplicaSet/nginx-deploy-bb9f8c596      Created pod: nginx-deploy-bb9f8c596-f8f7c
0s          Normal   Scheduled            Pod/nginx-deploy-bb9f8c596-f8f7c       Successfully assigned default/nginx-deploy-bb9f8c596-f8f7c to rayeez-cluster-worker
0s          Normal   Pulling              Pod/nginx-deploy-bb9f8c596-f8f7c       Pulling image "nginx"
0s          Normal   Pulled               Pod/nginx-deploy-bb9f8c596-f8f7c       Successfully pulled image "nginx" in 2.34s (2.34s including waiting). Image size: 59772801
 bytes.
0s          Normal   Created              Pod/nginx-deploy-bb9f8c596-f8f7c       Created container nginx-container
0s          Normal   Started              Pod/nginx-deploy-bb9f8c596-f8f7c       Started container nginx-container
```

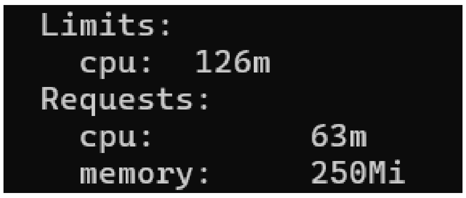
The VPA evicted one of the running pods, updated the CPU requests and limits, and then the ReplicaSet controller created new pods with these updated resource values.

```
Kubecetl get pods
```



```
root@DESKTOP-C6P8EQS:~/kubernetes/11)HPA_VPA$ kubectl get pods
NAME                           READY   STATUS    RESTARTS   AGE
load-generator-1               1/1     Running   0          30m
load-generator-2               1/1     Running   0          18m
nginx-deploy-bb9f8c596-f8f7c   1/1     Running   0          14m
nginx-deploy-bb9f8c596-pmlf7   1/1     Running   0          13m
```

Now the CPU request, Limits are set as below in each pods:



```
Limits:
  cpu:   126m
Requests:
  cpu:        63m
  memory:     250Mi
```

Conclusion: VPA dynamically adjusts the pod's resource requests and limits based on the application's varying load, while keeping the number of pods unchanged.