

```
eksctl create cluster --name demo-k8s-autoscaling --region ap-southeast-1 --nodegroup-name worker-nodes --node-type t3.small --nodes 2 --nodes-min 1 --nodes-max 4 --managed --spot
```

```
2023-11-19 21:51:00 [✓] EKS cluster 'demo-k8s-autoscaling' in 'ap-southeast-1' region is ready
(base) PS C:\Users\elect> kubectl get nodes
NAME                               STATUS  ROLES   AGE     VERSION
ip-192-168-15-18.ap-southeast-1.compute.internal  Ready   <none>  2m45s  v1.32.9-eks-c39b1d0
ip-192-168-41-176.ap-southeast-1.compute.internal  Ready   <none>  2m45s  v1.32.9-eks-c39b1d0
(base) PS C:\Users\elect> |
```

Tạo cluster

Deploy nginx



```
>Welcome  x  demo.yaml x
Y demo.yaml
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: php-apache
5  spec:
6    selector:
7      matchLabels:
8        run: php-apache
9    replicas: 1
10   template:
11     metadata:
12       labels:
13         run: php-apache
14     spec:
15       containers:
16         - name: php-apache
17           image: registry.k8s.io/hpa-example
18           ports:
19             - containerPort: 80
20           resources:
21             limits:
22               cpu: 500m
23             requests:
24               cpu: 200m
25 ---
26 apiVersion: v1
27 kind: Service
28 metadata:
29   name: php-apache
30   labels:
31     run: php-apache
32 spec:
33   ports:
34     - port: 80
35   selector:
36     run: php-apache
```

```
(base) PS E:\AWS-demo> kubectl apply -f demo.yaml
deployment.apps/php-apache created
service/php-apache created
(base) PS E:\AWS-demo> kubectl autoscale deployment php-apache --cpu-percent=50 --min=1 --max=10
Flag --cpu-percent has been deprecated. Use --cpu with percentage or resource quantity format (e.g., '70%' for utilization or '500m' for milliCPU).
horizontalpodautoscaler.autoscaling/php-apache autoscaled
```

Dùng 1 pod khác để bắn tải vào pod nginx

Kết quả pod nginx tăng replica từ 1 lên 6

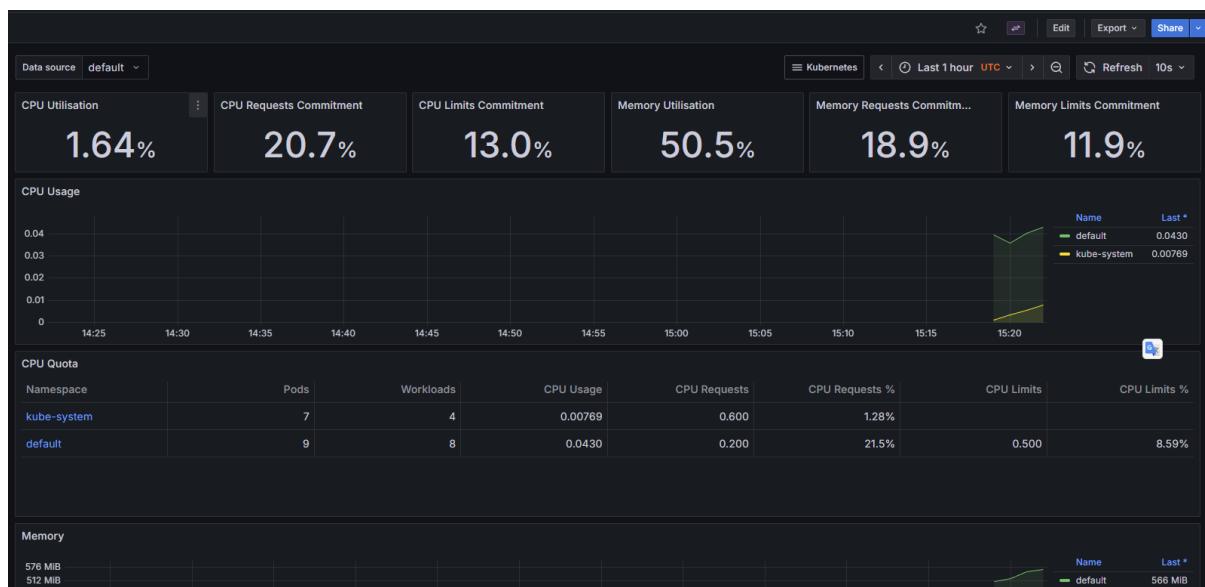
| (base) PS E:\AWS-demo> kubectl get hpa -w | | | | | | |
|---|-----------------------|--------------------|---------|---------|----------|-------|
| NAME | REFERENCE | TARGETS | MINPODS | MAXPODS | REPLICAS | AGE |
| php-apache | Deployment/php-apache | cpu: <unknown>/50% | 1 | 10 | 1 | 20s |
| php-apache | Deployment/php-apache | cpu: 0%/50% | 1 | 10 | 1 | 31s |
| php-apache | Deployment/php-apache | cpu: 238%/50% | 1 | 10 | 1 | 76s |
| php-apache | Deployment/php-apache | cpu: 250%/50% | 1 | 10 | 4 | 91s |
| php-apache | Deployment/php-apache | cpu: 96%/50% | 1 | 10 | 5 | 106s |
| php-apache | Deployment/php-apache | cpu: 64%/50% | 1 | 10 | 6 | 2m1s |
| php-apache | Deployment/php-apache | cpu: 54%/50% | 1 | 10 | 6 | 2m16s |

```
helm repo add prometheus-community https://prometheus-community.github.io/helm-charts
```

```
helm repo update
```

```
helm install monitor prometheus-community/kube-prometheus-stack --set grafana.adminPassword='admin'
```

```
kubectl port-forward svc/monitor-grafana 3000:80 (forward ra local)
```



Dung Jmeter de test tai

The screenshot shows the JMeter interface with an 'HTTP Request' test element selected. The 'Basic' tab is active. In the 'Web Server' section, the protocol is set to 'http://', the server name or IP is 'a47490c2a2ec34b418fe9782de95490e-509466473.ap-southeast-1.elb.amazonaws.com', and the port number is '80'. The 'HTTP Request' method is 'GET' and the path is '/'. The 'Content encoding' dropdown is set to 'Content encoding:'. Below these settings, there are tabs for 'Parameters', 'Body Data', and 'Files Upload'. A 'Send Parameters With the Request:' section is present with columns for 'Name', 'Value', 'URL Encode?', 'Content-Type', and 'Include Eq...'. The log window at the bottom shows 125 log entries, all of which are INFO messages from 'o.a.j.t.JMeterThread' indicating 'Thread started' for Thread Group 1 across various thread IDs (109 to 125).

CPU nhảy lên gần 100% và đã scale đủ 10 replica

The screenshot displays the Kubernetes Metrics API dashboard. At the top, it shows CPU Utilisation, CPU Requests Commitment, CPU Limits Commitment, Memory Utilisation, Memory Requests Commitment, and Memory Limits Commitment for the 'default' namespace. The values are: CPU Utilisation 34.9%, CPU Requests Commitment 67.4%, CPU Limits Commitment 130%, Memory Utilisation 58.3%, Memory Requests Commitment 18.9%, and Memory Limits Commitment 11.9%. Below this, the 'CPU Usage' section shows a line chart of CPU usage over time (from 14:35 to 15:30) for two namespaces: 'default' (green line) and 'kube-system' (yellow line). The 'default' namespace usage is low until 15:20, then spikes sharply to nearly 1.0. The 'kube-system' namespace usage is very low throughout. The 'CPU Quota' section shows a table of CPU quotas for the 'default' and 'kube-system' namespaces.

| Namespace | Quota |
|-------------|--------|
| default | 0.993 |
| kube-system | 0.0109 |

At the bottom, a terminal window shows the output of the command 'kubectl get deploy php-apache -n default', listing the deployment details including targets, min/max pods, replicas, and age.

```
NAME      REFERENCE          TARGETS    MINPODS   MAXPODS   REPLICAS   AGE
php-apache Deployment/php-apache  cpu: 240%/50%  1          10         10          33m
(base) PS C:\Users\elect>
```