Lab: Horizontal Pod Autoscaler

Introduction:

The Horizontal Pod Autoscaler automatically scales the number of pods in a replication controller, deployment, replica set or stateful set based on Observed CPU utilization. The Horizontal Pod Autoscaler is implemented as a Kubernetes API resource and a controller.

The resource determines the behaviour of the controller. The controller periodically adjusts the number of replicas in a replication controller or deployment to match the observed average CPU utilization to the target specified by user.

Objectives:

- **Create A Nginx Deployment**
- **Create HorizontalPodAutoscalers**
- **Install siege -http Load Simulator**
- **Watch Auto Scale Up & Watch Auto Scale Down**
- **Cleanup**

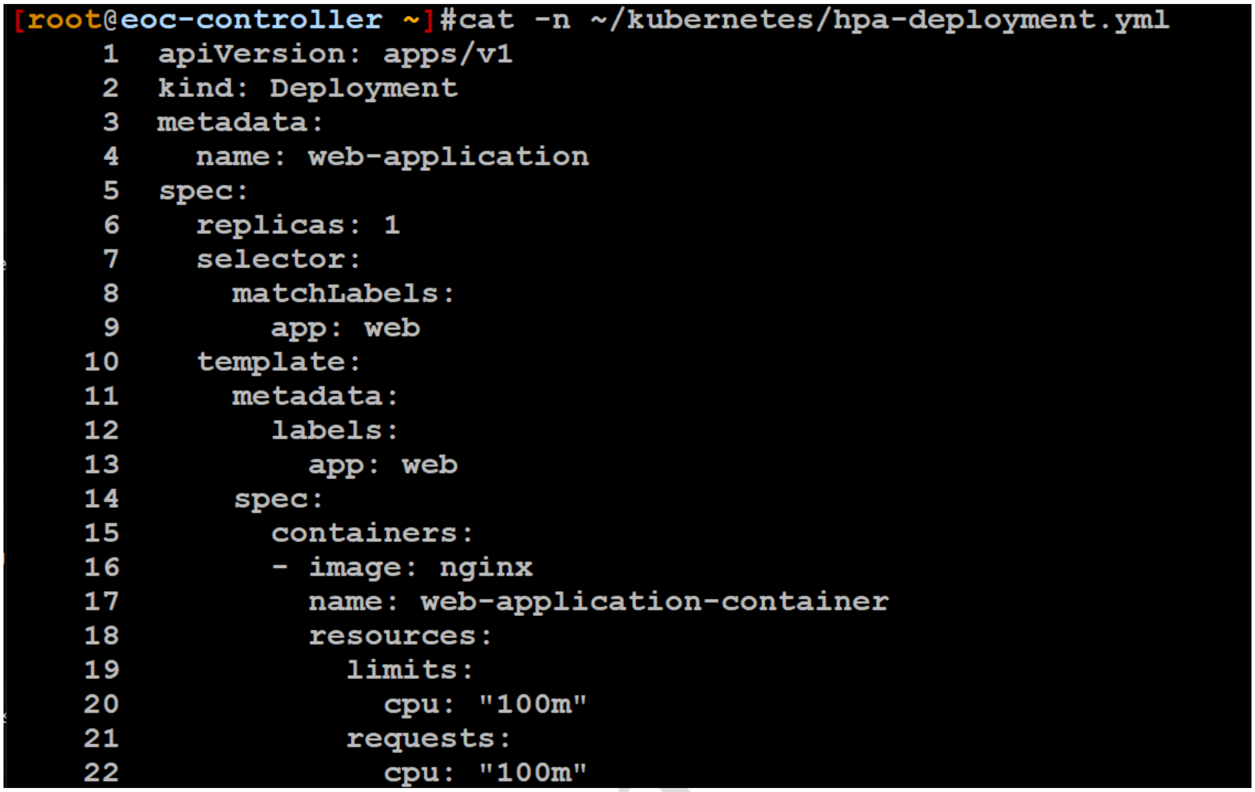
Ensure that you have logged-in as **root** user on **eoc-controller** node.

1. Create A Nginx Deployment

1.1 Let's **view** the manifest to create a deployment for nginx by executing the below command.

```
# cat -n ~/kubernetes/hpa-deployment.yml
```

Output:



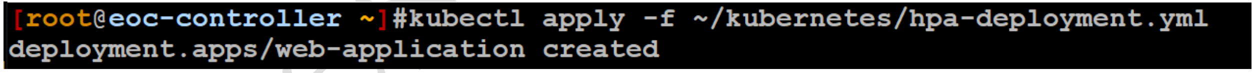
```
[root@eoc-controller ~]#cat -n ~/kubernetes/hpa-deployment.yml
     1  apiVersion: apps/v1
     2  kind: Deployment
     3  metadata:
     4    name: web-application
     5  spec:
     6    replicas: 1
     7    selector:
     8      matchLabels:
     9        app: web
    10    template:
    11      metadata:
    12        labels:
    13          app: web
    14      spec:
    15        containers:
    16        - image: nginx
    17          name: web-application-container
    18          resources:
    19            limits:
    20              cpu: "100m"
    21            requests:
    22              cpu: "100m"
```

Note: The cpu resource is limited to 100m this will help us to scale based on cpu utilization.

1.2 Let's **create** the deployment by executing the below command.

```
# kubectl create -f ~/kubernetes/hpa-deployment.yml
```

Output:

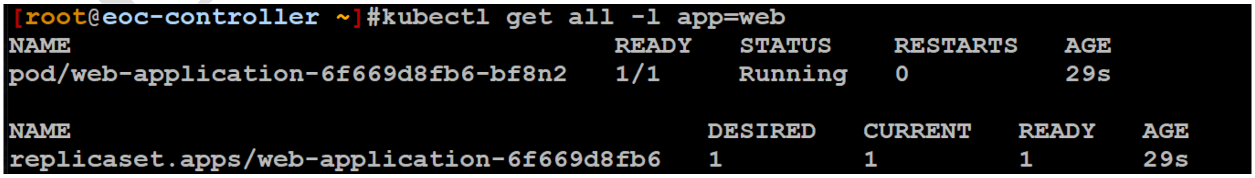


```
[root@eoc-controller ~]#kubectl apply -f ~/kubernetes/hpa-deployment.yml
deployment.apps/web-application created
```

1.3 Let's **verify** the deployment by executing the below command.

```
# kubectl get all -l app=web
```

Output:

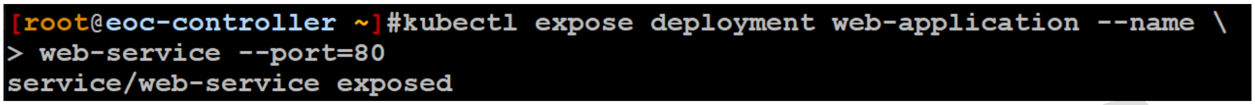


```
[root@eoc-controller ~]#kubectl get all -l app=web
NAME                                   READY   STATUS    RESTARTS   AGE
pod/web-application-6f669d8fb6-bf8n2   1/1     Running   0          29s

NAME                                         DESIRED   CURRENT   READY   AGE
replicaset.apps/web-application-6f669d8fb6   1         1         1       29s
```

1.4 Let’s **create** and expose a service of ClusterIP type to access our pod.

```
# kubectl expose deployment web-application --name \
 web-service --port=80
```

Output:

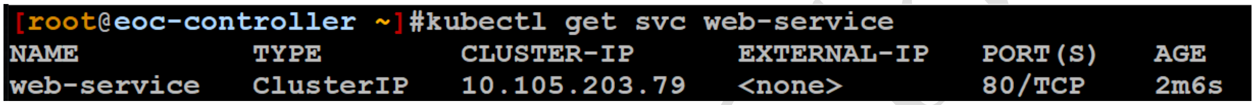


```
[root@eoc-controller ~]#kubectl expose deployment web-application --name \
> web-service --port=80
service/web-service exposed
```

1.5 Let’s **verify** the service by executing the below command.

```
# kubectl get svc web-service
```

Output:



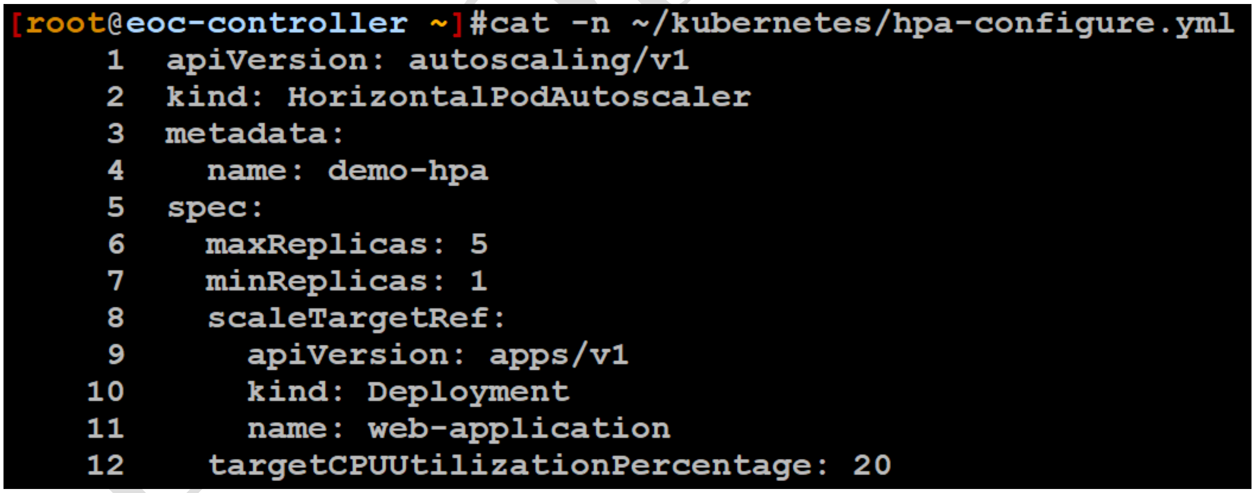
```
[root@eoc-controller ~]#kubectl get svc web-service
NAME            TYPE         CLUSTER-IP      EXTERNAL-IP    PORT(S)    AGE
web-service     ClusterIP    10.105.203.79   <none>         80/TCP     2m6s
```

2 Create HorizontalPodAutoscalers

2.1 Let’s **view** the manifest for creating hpa by executing the below command.

```
# cat -n ~/kubernetes/hpa-configure.yml
```

Output:

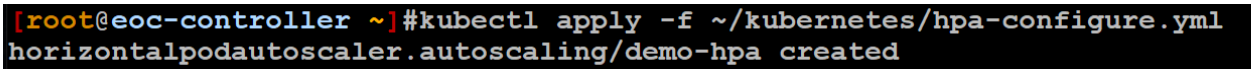


```
[root@eoc-controller ~]#cat -n ~/kubernetes/hpa-configure.yml
     1  apiVersion: autoscaling/v1
     2  kind: HorizontalPodAutoscaler
     3  metadata:
     4    name: demo-hpa
     5  spec:
     6    maxReplicas: 5
     7    minReplicas: 1
     8    scaleTargetRef:
     9      apiVersion: apps/v1
    10      kind: Deployment
    11      name: web-application
    12    targetCPUUtilizationPercentage: 20
```

2.2 Let’s **create** the hpa by executing the below command.

```
# kubectl apply -f ~/kubernetes/hpa-configure.yml
```

Output:

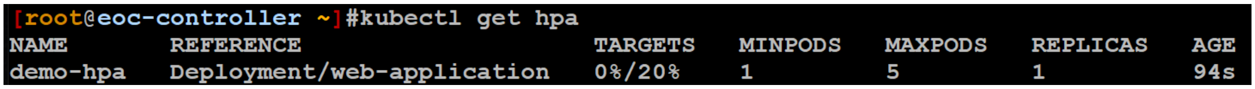


```
[root@eoc-controller ~]#kubectl apply -f ~/kubernetes/hpa-configure.yml
horizontalpodautoscaler.autoscaling/demo-hpa created
```

2.3 Let's **verify** hpa details by executing the below command.

```
# kubectl get hpa
```

Output:



```
[root@eoc-controller ~]#kubectl get hpa
NAME        REFERENCE                       TARGETS    MINPODS    MAXPODS    REPLICAS    AGE
demo-hpa    Deployment/web-application      0%/20%     1          5          1           94s
```

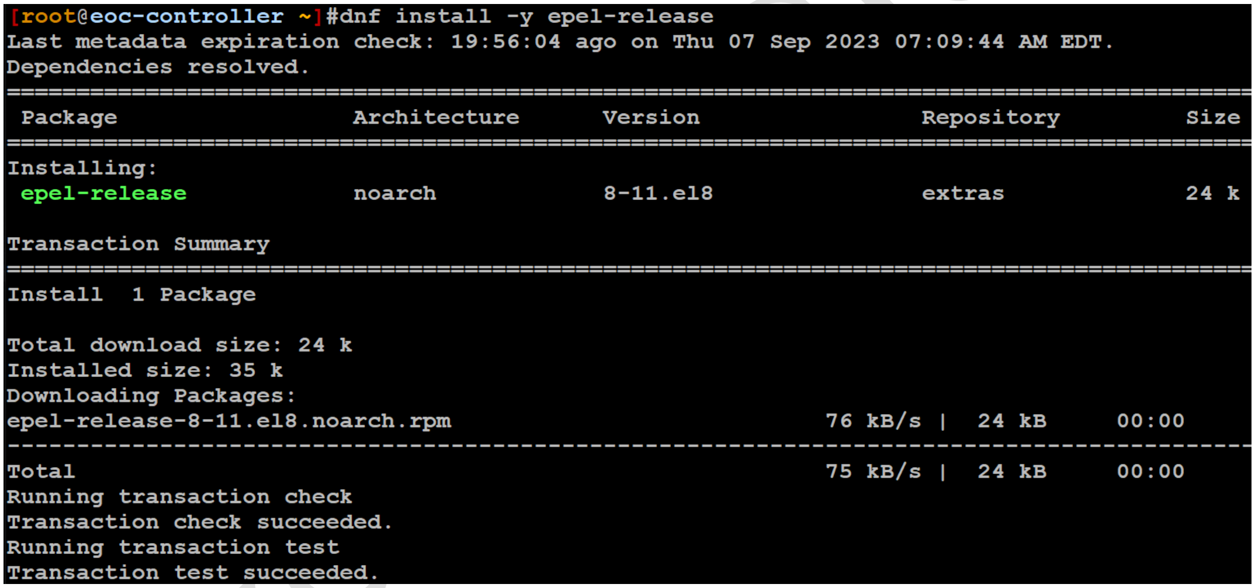
Info:Lets now simiulate the load to increase the cpu utlization and watch the hpa work. We will use siege – which is an powerful HTTP load testing and benchmarking utility.

3 Install siege -http Load Simulator

3.1 Let's **install** the **epel repository** by executing below commands.

```
# dnf install -y epel-release
```

Output:

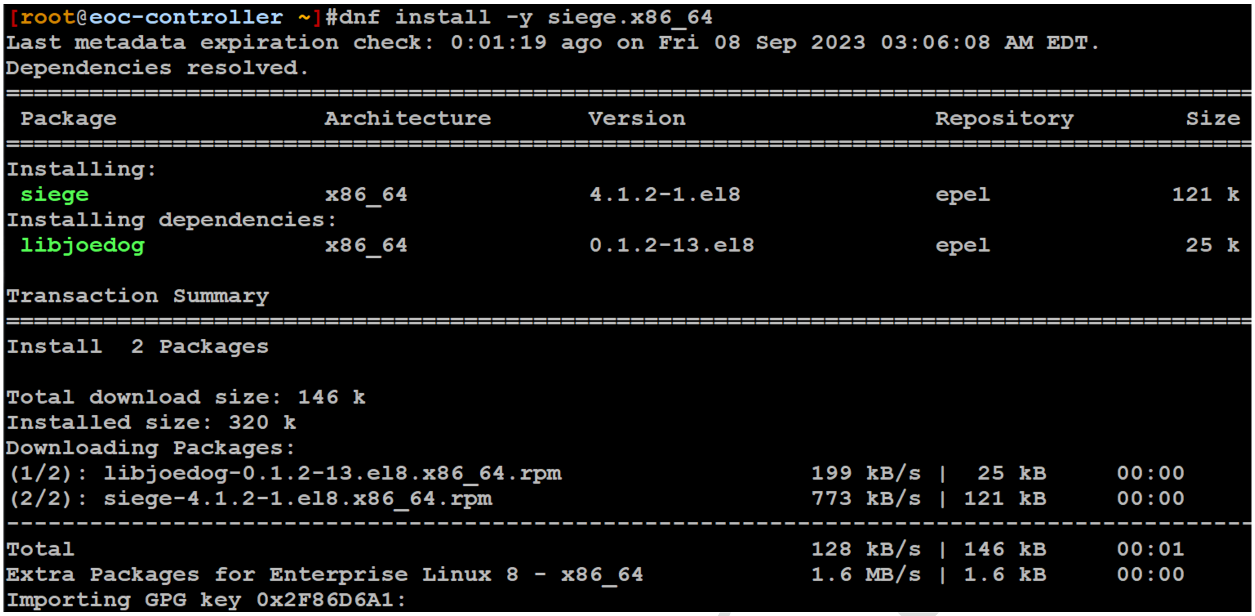


```
[root@eoc-controller ~]#dnf install -y epel-release
Last metadata expiration check: 19:56:04 ago on Thu 07 Sep 2023 07:09:44 AM EDT.
Dependencies resolved.
==========================================================================================
 Package                 Architecture      Version                 Repository        Size
==========================================================================================
Installing:
 epel-release            noarch            8-11.el8                extras            24 k

Transaction Summary
==========================================================================================
Install  1 Package

Total download size: 24 k
Installed size: 35 k
Downloading Packages:
epel-release-8-11.el8.noarch.rpm                          76 kB/s |  24 kB     00:00
------------------------------------------------------------------------------------------
Total                                                     75 kB/s |  24 kB     00:00
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
```

3.2 Let’s **install** the **siege** utility by executing below commands.

```
# dnf install -y siege.x86_64
```

Output:

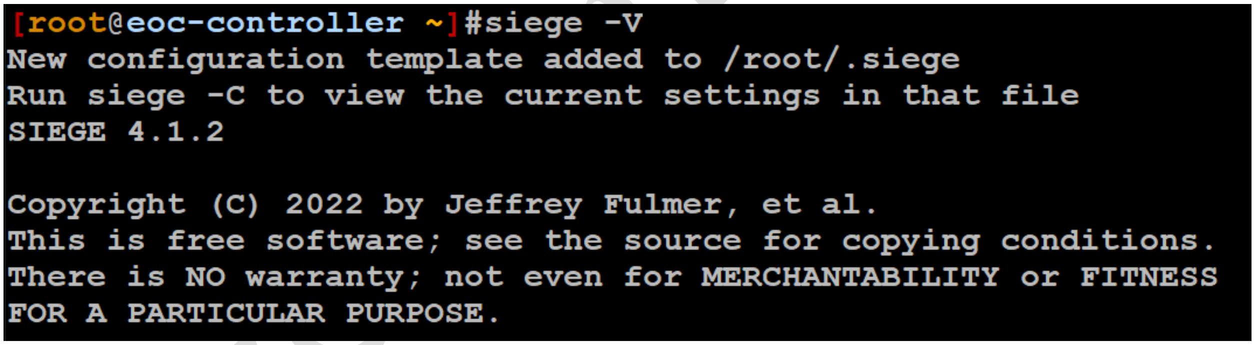


```
[root@eoc-controller ~]#dnf install -y siege.x86_64
Last metadata expiration check: 0:01:19 ago on Fri 08 Sep 2023 03:06:08 AM EDT.
Dependencies resolved.
==========================================================================================
 Package                 Architecture        Version                  Repository     Size
==========================================================================================
Installing:
 siege                   x86_64              4.1.2-1.el8              epel          121 k
Installing dependencies:
 libjoedog               x86_64              0.1.2-13.el8             epel           25 k

Transaction Summary
==========================================================================================
Install  2 Packages

Total download size: 146 k
Installed size: 320 k
Downloading Packages:
(1/2): libjoedog-0.1.2-13.el8.x86_64.rpm                 199 kB/s |  25 kB     00:00
(2/2): siege-4.1.2-1.el8.x86_64.rpm                      773 kB/s | 121 kB     00:00
------------------------------------------------------------------------------------------
Total                                                    128 kB/s | 146 kB     00:01
Extra Packages for Enterprise Linux 8 - x86_64           1.6 MB/s | 1.6 kB     00:00
Importing GPG key 0x2F86D6A1:
```

3.3 Let's **verify** that **siege** is installed correctly by executing the below command.

```
# siege -V
```

Output:



```
[root@eoc-controller ~]#siege -V
New configuration template added to /root/.siege
Run siege -C to view the current settings in that file
SIEGE 4.1.2

Copyright (C) 2022 by Jeffrey Fulmer, et al.
This is free software; see the source for copying conditions.
There is NO warranty; not even for MERCHANTABILITY or FITNESS
FOR A PARTICULAR PURPOSE.
```

Note: Open another terminal and run the below command.

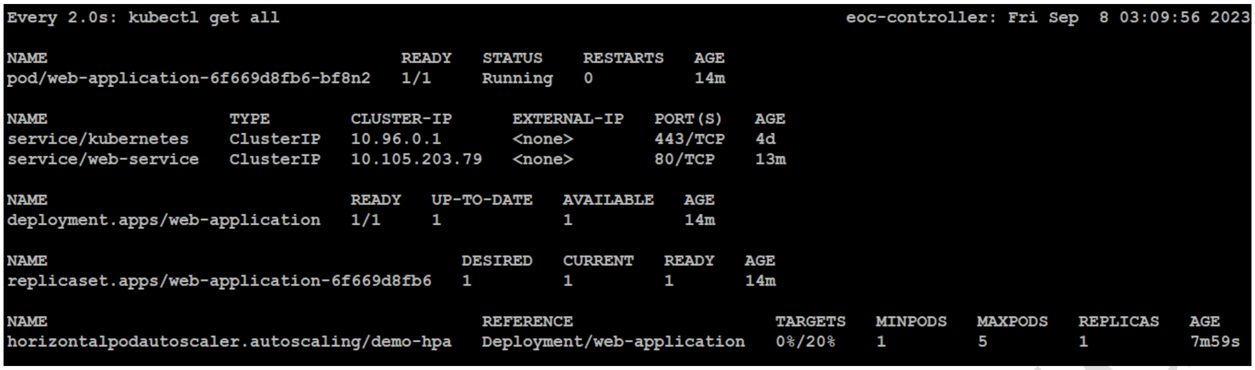
4 Watch Auto Scale Up & Watch Auto Scale Down

4.1 Let's **watch** the deployment by executing the below command.

Terminal 2:

```
# watch kubectl get all
```

Output:



```
Every 2.0s: kubectl get all                                                        eoc-controller: Fri Sep  8 03:09:56 2023

NAME                                       READY   STATUS    RESTARTS   AGE
pod/web-application-6f669d8fb6-bf8n2       1/1     Running   0          14m

NAME                      TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)   AGE
service/kubernetes        ClusterIP   10.96.0.1       <none>        443/TCP   4d
service/web-service       ClusterIP   10.105.203.79   <none>        80/TCP    13m

NAME                                  READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/web-application       1/1     1            1           14m

NAME                                             DESIRED   CURRENT   READY   AGE
replicaset.apps/web-application-6f669d8fb6       1         1         1       14m

NAME                                            REFERENCE                     TARGETS   MINPODS   MAXPODS   REPLICAS   AGE
horizontalpodautoscaler.autoscaling/demo-hpa    Deployment/web-application    0%/20%    1         5         1          7m59s
```

4.2 Let’s **increase** the load by running the siege utility.

Terminal 1:

```
# siege -q -c 5 -t 2m http://10.105.203.79
```

Note: -q = quiet mode, -c = concurrent users (we are setting it to 5), -t = time (we are setting it to 2 minutes) and accessing one of the nodes and using the NodePort of nginx app.

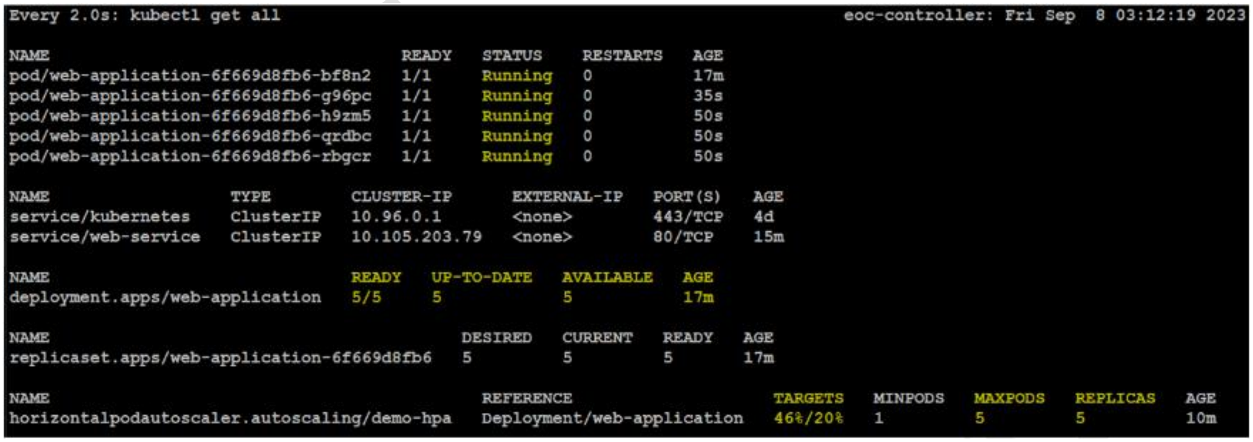
Info: The controller checks the metrics every 15 seconds, as the load gradually increases, the pod will begin to autoscale-up. Continue to watch the Terminal-2 to watch the pods autoscale

4.3 Let’s continue to **watch** the pods by executing the below command.

Terminal 2:

```
# watch kubectl get all
```

Output:



```
Every 2.0s: kubectl get all                                                        eoc-controller: Fri Sep  8 03:12:19 2023

NAME                                       READY   STATUS    RESTARTS   AGE
pod/web-application-6f669d8fb6-bf8n2       1/1     Running   0          17m
pod/web-application-6f669d8fb6-g96pc       1/1     Running   0          35s
pod/web-application-6f669d8fb6-h9zm5       1/1     Running   0          50s
pod/web-application-6f669d8fb6-qrdbc       1/1     Running   0          50s
pod/web-application-6f669d8fb6-rbgcr       1/1     Running   0          50s

NAME                      TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)   AGE
service/kubernetes        ClusterIP   10.96.0.1       <none>        443/TCP   4d
service/web-service       ClusterIP   10.105.203.79   <none>        80/TCP    15m

NAME                                  READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/web-application       5/5     5            5           17m

NAME                                             DESIRED   CURRENT   READY   AGE
replicaset.apps/web-application-6f669d8fb6       5         5         5       17m

NAME                                            REFERENCE                     TARGETS   MINPODS   MAXPODS   REPLICAS   AGE
horizontalpodautoscaler.autoscaling/demo-hpa    Deployment/web-application    46%/20%   1         5         5          10m
```

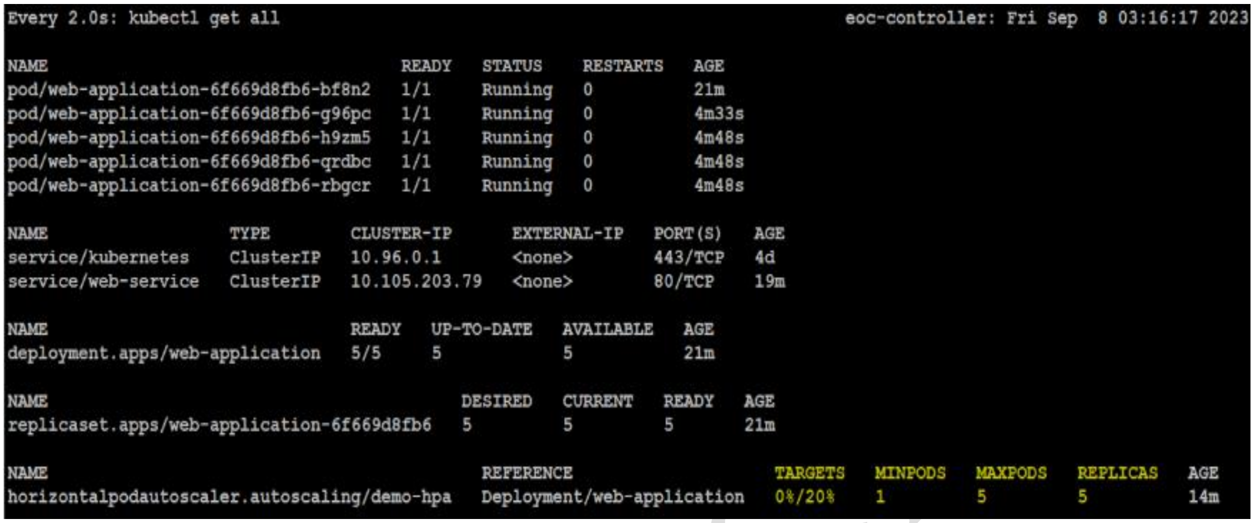
Note: The siege command will complete after 2 minutes and then load will be reduce.

4.4 Let’s continue to watch the pods by executing the below command.

Terminal 2:

```
# watch kubectl get all
```

Output:

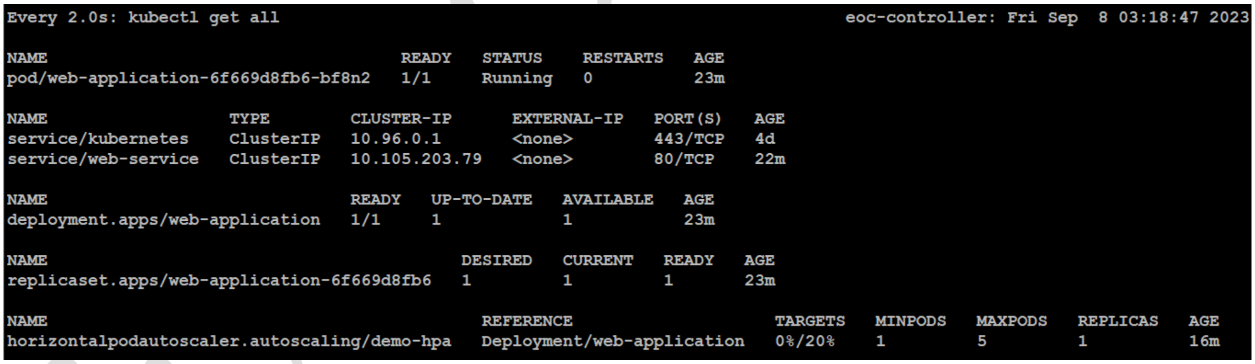


```
Every 2.0s: kubectl get all                                                    eoc-controller: Fri Sep  8 03:16:17 2023

NAME                                  READY   STATUS    RESTARTS   AGE
pod/web-application-6f669d8fb6-bf8n2   1/1     Running   0          21m
pod/web-application-6f669d8fb6-g96pc   1/1     Running   0          4m33s
pod/web-application-6f669d8fb6-h9zm5   1/1     Running   0          4m48s
pod/web-application-6f669d8fb6-qrdbc   1/1     Running   0          4m48s
pod/web-application-6f669d8fb6-rbgcr   1/1     Running   0          4m48s

NAME                  TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)   AGE
service/kubernetes    ClusterIP   10.96.0.1       <none>        443/TCP   4d
service/web-service   ClusterIP   10.105.203.79   <none>        80/TCP    19m

NAME                              READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/web-application   5/5     5            5           21m

NAME                                         DESIRED   CURRENT   READY   AGE
replicaset.apps/web-application-6f669d8fb6   5         5         5       21m

NAME                                           REFERENCE                    TARGETS   MINPODS   MAXPODS   REPLICAS   AGE
horizontalpodautoscaler.autoscaling/demo-hpa   Deployment/web-application   0%/20%    1         5         5          14m
```

Note: As soon as the load is reduced, the cpu % is back to zero, but the pods will no autoscale-down immediately. Instead, it will wait for **stabilizationWindowSeconds:** which by default is 300 seconds and then scale-down.

After 5-7 minutes:

Output:



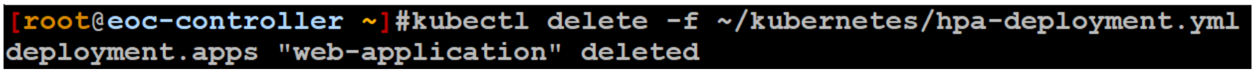
```
Every 2.0s: kubectl get all                                                    eoc-controller: Fri Sep  8 03:18:47 2023

NAME                                  READY   STATUS    RESTARTS   AGE
pod/web-application-6f669d8fb6-bf8n2   1/1     Running   0          23m

NAME                  TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)   AGE
service/kubernetes    ClusterIP   10.96.0.1       <none>        443/TCP   4d
service/web-service   ClusterIP   10.105.203.79   <none>        80/TCP    22m

NAME                              READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/web-application   1/1     1            1           23m

NAME                                         DESIRED   CURRENT   READY   AGE
replicaset.apps/web-application-6f669d8fb6   1         1         1       23m

NAME                                           REFERENCE                    TARGETS   MINPODS   MAXPODS   REPLICAS   AGE
horizontalpodautoscaler.autoscaling/demo-hpa   Deployment/web-application   0%/20%    1         5         1          16m
```

5 Cleanup.

5.1 Let's **delete** the **hpa** deployment app by executing the below commands.

```
# kubectl delete -f ~/kubernetes/hpa-deployment.yml
```

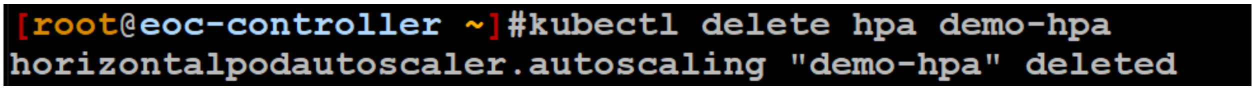
Output:



```
[root@eoc-controller ~]#kubectl delete -f ~/kubernetes/hpa-deployment.yml
deployment.apps "web-application" deleted
```

```
# kubectl delete hpa demo-hpa
```

Output:

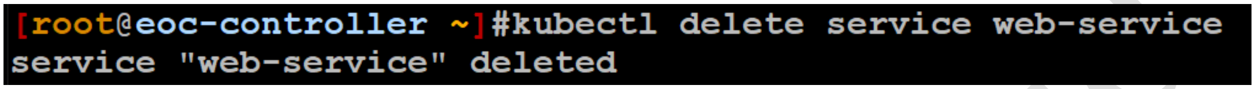


```
[root@eoc-controller ~]#kubectl delete hpa demo-hpa
horizontalpodautoscaler.autoscaling "demo-hpa" deleted
```

5.2 Let’s **delete** the **service** by executing the below commands.

```
# kubectl delete service web-service
```

Output:



```
[root@eoc-controller ~]#kubectl delete service web-service
service "web-service" deleted
```