

## Contents

- HPA를 사용해서 자동 스케일링 해보기

### HPA를 사용해서 자동 스케일링 해보기

Horizontal Pod Autoscaler(HPA)를 이용하여 자동으로 Pod의 개수를 조절하는 실습입니다.

실습 내용은 [HorizontalPodAutoscaler Walkthrough](#) 를 기반으로 하였습니다.

먼저 자원 모니터링을 위한 metrics-server가 필요합니다.

우리는 앞서 Minikube의 Metrics-Server Addon을 Enable시켰기 때문에 따로 해줄일은 없습니다.

정상적으로 Addon이 Enable된 경우 아래와 같이 확인 가능합니다.

```
ubuntu@ip-10-0-10-180:~$ kubectl top node
NAME           CPU(cores)   CPU%    MEMORY(bytes)   MEMORY%
ip-172-31-20-30  187m        9%     2234Mi        57%
```

명령어 : `kubectl top node`

## Docker & Kubernetes - [Hands-on] 14. Kubernetes Horizontal Pod Autoscaler

이제 준비가 됐으면, 다음 명령어를 실행하여 간단한 테스트용 Pod 를 준비합니다.

```
ubuntu@ip-10-0-10-180:~$ kubectl apply -f https://k8s.io/examples/application/php-apache.yaml
deployment.apps/php-apache created
service/php-apache created
```

명령어 : `kubectl apply -f https://k8s.io/examples/application/php-apache.yaml`  
Deployment와 Service가 만들어집니다.

이제 hpa를 생성합니다.

명령어는 다음과 같습니다.

CPU 사용량을 50%로 유지하기 위해서 Pod의 개수를 1에서 10 사이로 조정하라는 의미입니다.

```
ubuntu@ip-10-0-10-180:~$ kubectl autoscale deployment php-apache --cpu-percent=50 --min=1 --max=10
horizontalpodautoscaler.autoscaling/php-apache autoscaled
```

명령어 : `kubectl autoscale deployment php-apache --cpu-percent=50 --min=1 --max=10`

잘 만들어졌나 볼까요?

```
ubuntu@ip-10-0-10-180:~$ kubectl get hpa
NAME      REFERENCE      TARGETS      MINPODS      MAXPODS      REPLICAS      AGE
php-apache  Deployment/php-apache  0%/50%       1            10           1            24s
```

명령어 : `kubectl get hpa`

## Docker & Kubernetes - [Hands-on] 14. Kubernetes Horizontal Pod Autoscaler

다음은 먼저 생성한 Pod에 부하를 줄 도우미 친구입니다.

간단한 sh 명령어를 실행할 pod(load-generator)를 만들어서 반복문을 실행합니다.  
앞에서 만든 Pod에 계속 요청(http request)을 보내서 CPU 사용율을 높이게 됩니다.

시스템에 사용자가 늘어난 상황을 비슷하게 만든거라고 보시면 됩니다.

```
ubuntu@ip-10-0-10-180:~$ kubectl run -i --tty load-generator --rm --image=busybox --restart=Never -- /bin/sh -c "while sleep 0.01; do wget -q -O- http://php-apache; done"
If you don't see a command prompt, try pressing enter.
OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!
OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!
OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!
OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!
```

명령어 :

```
kubectl run -i --tty load-generator --rm --image=busybox --restart=Never -- /bin/sh -c "while sleep 0.01; do wget -q -O- http://php-apache;
done"
```

## Docker & Kubernetes - [Hands-on] 14. Kubernetes Horizontal Pod Autoscaler

이제 터미널을 하나 더 열고 아래 명령어를 실행해서 어떤 변화가 있는지 알아봅니다.

```
ubuntu@ip-10-0-10-180:~$ kubectl get hpa
NAME          REFERENCE      TARGETS      MINPODS      MAXPODS      REPLICAS      AGE
php-apache    Deployment/php-apache  73%/50%     1            10           6            3m52s

ubuntu@ip-10-0-10-180:~$ kubectl get pods
NAME                  READY   STATUS    RESTARTS   AGE
load-generator        1/1     Running   0          99s
php-apache-d4cf67d68-6bq8s  1/1     Running   0          5m6s
php-apache-d4cf67d68-7v7t4  1/1     Running   0          71s
php-apache-d4cf67d68-7vptw  1/1     Running   0          56s
php-apache-d4cf67d68-81bqn  1/1     Running   0          71s
php-apache-d4cf67d68-cfczl  1/1     Running   0          71s
php-apache-d4cf67d68-k58b6  1/1     Running   0          26s
php-apache-d4cf67d68-k1148  1/1     Running   0          11s
php-apache-d4cf67d68-trgtm  1/1     Running   0          11s
```

명령어 : `kubectl get hpa`, `kubectl get pods`

1개에서 시작한 Pod의 개수가 늘어나는 걸 확인할 수 있습니다.

어느정도 시간이 지나서, Pod가 늘어나는걸 보셨으면, 첫 번째 Terminal의 반복문을 중지해주세요.

Ctrl + c로 중지하시면 됩니다.

부하를 중지하면 다시 Pod의 수가 줄어드는것도 확인할 수 있습니다.

끝~~~ \_↗(。'◡'。)