# Kubernetes - Autoscaling

**Autoscaling** is one of the key features in Kubernetes cluster. It is a feature in which the cluster is capable of increasing the number of nodes as the demand for service response increases and decrease the number of nodes as the requirement decreases. This feature of auto scaling is currently supported in Google Cloud Engine (GCE) and Google Container Engine (GKE) and will start with AWS pretty soon.

In order to set up scalable infrastructure in GCE, we need to first have an active GCE project with features of Google cloud monitoring, google cloud logging, and stackdriver enabled.

First, we will set up the cluster with few nodes running in it. Once done, we need to set up the following environment variable.

## Environment Variable

export NUM\_NODES = 2

export KUBE\_AUTOSCALER\_MIN\_NODES = 2

export KUBE\_AUTOSCALER\_MAX\_NODES = 5

export KUBE\_ENABLE\_CLUSTER\_AUTOSCALER = true

Once done, we will start the cluster by running **kube-up.sh**. This will create a cluster together with cluster auto-scalar add on.

./cluster/kube-up.sh

On creation of the cluster, we can check our cluster using the following kubectl command.

$ kubectl get nodes

NAME STATUS AGE

kubernetes-master Ready,SchedulingDisabled 10m

kubernetes-minion-group-de5q Ready 10m

kubernetes-minion-group-yhdx Ready 8m

Now, we can deploy an application on the cluster and then enable the horizontal pod autoscaler. This can be done using the following command.

$ kubectl autoscale deployment <Application Name> --cpu-percent = 50 --min = 1 --

max = 10

The above command shows that we will maintain at least one and maximum 10 replica of the POD as the load on the application increases.

We can check the status of autoscaler by running the **$kubclt get hpa** command. We will increase the load on the pods using the following command.

$ kubectl run -i --tty load-generator --image = busybox /bin/sh

$ while true; do wget -q -O- http://php-apache.default.svc.cluster.local; done

We can check the **hpa** by running **$ kubectl get hpa** command.

$ kubectl get hpa

NAME REFERENCE TARGET CURRENT

php-apache Deployment/php-apache/scale 50% 310%

MINPODS MAXPODS AGE

1 20 2m

$ kubectl get deployment php-apache

NAME DESIRED CURRENT UP-TO-DATE AVAILABLE AGE

php-apache 7 7 7 3 4m

We can check the number of pods running using the following command.

jsz@jsz-desk2:~/k8s-src$ kubectl get pods

php-apache-2046965998-3ewo6 0/1 Pending 0 1m

php-apache-2046965998-8m03k 1/1 Running 0 1m

php-apache-2046965998-ddpgp 1/1 Running 0 5m

php-apache-2046965998-lrik6 1/1 Running 0 1m

php-apache-2046965998-nj465 0/1 Pending 0 1m

php-apache-2046965998-tmwg1 1/1 Running 0 1m

php-apache-2046965998-xkbw1 0/1 Pending 0 1m

And finally, we can get the node status.

$ kubectl get nodes

NAME STATUS AGE

kubernetes-master Ready,SchedulingDisabled 9m

kubernetes-minion-group-6z5i Ready 43s

kubernetes-minion-group-de5q Ready 9m

kubernetes-minion-group-yhdx Ready 9m