# Lab 1

## Creating a Kubernetes cluster on OCI (OKE)

<https://www.oracle.com/webfolder/technetwork/tutorials/obe/oci/oke-full/index.html>

* Follow the instructions for the lab with the following exceptions:
  + Login with your OCI username / password. Everyone has their own credentials.
  + We will be doing a quick create.
  + Kubernetes version 1.11.5
  + Create Node Pool use VM.Standard2.2
  + Quantity per subnet 1
  + Additional Add-ons
    - Kubernetes dashboard enabled (selected)
    - Tiller (helm) enabled (selected)
  + We will be using the Phoenix Region
  + You can name your cluster anything you like. It does not have to be tutorial
  + Only go through step 5

## Working with Kubernetes and the dashboard

# Scaling applications

In this lab, you will learn how to scale out a deployed application on Kubernetes.

## Scale a simple application using the dashboard

1. Login to the dashboard
2. Locate the hello-k8s-deployment pod under **Pods**. You should see that the number of deployed pods is 1 (1/1)
3. Locate the 'hello-k8s-deployment' under Deployments in the dashboard
4. Click on the **…​** to the right
5. Select **Scale**
6. Increase the desired number of pods to 3 and click **OK**
7. Watch 2 more pods being created
8. Under Pods, notice how each pod has been deployed to a different Node (with different IP addresses)
9. Click on the **…​** next to the pod and select btn[Delete]. What do you observe and what do you think happened?

## Scale back the application

1. You currently have 3 pods running for this application
2. Locate the 'hello-k8s-deployment' under Deployments in the dashboard
3. Click on the **…​** to the right
4. Select btn[Scale]
5. Decrease the desired number of pods to 2 and click **OK**
6. What do you observe and what do you think happened?

## Additional Experiments with scaling

1. Scale the number of pods to a number greater than the number of worker nodes you have. What do you observe and what do you think happened?
2. Scale the number of pods back to a number smaller than the number of workers nodes you have. What do you observe and what do you think happened?

# Destroy applications

In this lab, you will learn how to delete a deployed application on Kubernetes.

## Destroy a simple application using the dashboard

1. Login to the dashboard as described in [Access the Kubernetes Dashboard](http://au1-tkadm01.au.oracle.com/cloudnativedeepdive/1.0/k8s/basic/dashboard.html)
2. Locate the 'hello-k8s-deployment' under Deployments in the dashboard
3. Click on the **…​** to the right
4. Select **Delete**
5. Locate the 'hello-k8s-service' under services
6. Click on the **…​** to the right
7. Select **Delete**

## Destroy a simple application using kubectl

1. Redeploy the helloworld application
2. List the deployments

$ kubectl get deployments

1. Delete the deployment and service
2. $ kubectl delete deployments hello-k8s-deployment

$ kubectl delete services hello-k8s-service

# Create namespaces to represent dev, test, prod environments

In this lab, you will create 3 namespaces to represent dev, test and production environments.

## Listing namespaces

1. Type the following command
2. $ kubectl get ns
3. Create 3 namespaces (dev, test, prod)
4. $ kubectl create namespace <namespace name>
5. List the namespaces again. You should see the 3 additional namespaces
6. Deploy the helloworld application again
7. Login to the dashboard
8. Locate the **namespaces** pulldown menu on the left.
9. Select the **dev** namespace.
10. What do you observe? Why do you think this happened?

# Set quotas to namespaces

In this lab, you will set different quotas to the dev, test and prod namespaces.

## Create and set quotas on dev namespace

1. Create a file compute-resources.yaml
2. Add the following:

apiVersion: v1

kind: ResourceQuota

metadata:

name: compute-resources

namespace: dev

spec:

hard:

count/pods: "1"

count/replicasets.extensions: "1"

count/secrets: "1"

count/persistentvolumeclaims: "1"

count/services: "1"

Execute:

kubectl apply -f compute-resources.yaml

1. Create another file object-counts.yaml
2. Add the following:

apiVersion: v1

kind: ResourceQuota

metadata:

name: object-counts

namespace:dev

spec:

hard:

configmaps: "1"

persistentvolumeclaims: "1"

replicationcontrollers: "1"

secrets: "1"

services: "1"

services.loadbalancers: "1"

Execute:

kubectl apply -f object-counts.yaml

1. Deploy the helloworld application again but to the dev namespace

kubectl apply -f https://raw.githubusercontent.com/karthequian/kubernetesHelloworld/master/hello.yaml -n=dev

1. Login to the dashboard
2. Change to the dev namespace.
3. Try to scale the number of pods to 2 for 'hello-k8s-deployment'
4. What do you observe? Why do you think this happened?
5. Repeat the above but double the number of pods to 2 and 4 respectively in test and prod namespaces in the ResourceQuota

## Clean up

kubectl delete resourcequota compute-resources -n dev (do this for each of the namespaces created)

kubectl delete resourcequota object-counts -n dev (do this for each of the namespaces created)

kubectl delete -f https://raw.githubusercontent.com/karthequian/kubernetesHelloworld/master/hello.yaml

kubectl delete namespace dev

kubectl delete namespace test

kubectl delete namespace prod