KUBERNETES HOME EXERCISE

This is the first version of this exercise(only tried by me) so there may be mistakes or problems, if some problem occur please contact me via mail urban.bilstrup@hh.se. When you are finished, submit answers to the exercise on blackboard.

1. Some basics

Go to the walk through

https://kubernetes.io/docs/tutorials/kubernetes-basics/

and try out the emulator and some basic functionality. Each module (not the first, learn Kubernetes basics) have a interactive tutorial, read and run the tutorials and answer the questions below.

Question: Give a short description of Kubernetes and what functionality do it provides?

1 Create a Kubernetes cluster

Questions:

a) Describe differences between master node and worker nodes?
b) What version of minkube is used in the tutorial?

2 Deploy an app

Question:

a) Describe kubectl?
b) What is the output of **kubectl get nodes**
c) What is the ouput of **kubectl get deployments**

3 Explore your app

Question:

a) What is a pod, describe its entities?
b) What information do the **kubectl describe pods** command give

4 Expose your app publicly

Question:

a) Describe what a service is in Kubernetes?
b) What do the **kubectl expose deployment/kubernetes-bootcamp --type="NodePort" --port 8080** command do?

5 Scale up your app

Question:

a) How do labels and Label Selector objects relate to a Service?
b) What is the output **kubectl get rs** command?

6 Update your app

Question:

a) Describe rolling updates in Kubernetes?

b) Whats doe the **kubectl set image deployments/kubernetes-bootcamp kubernetes-bootcamp=jocatalin/kubernetes-bootcamp:v2** command do?

2. Installing MINIKUBE

All instructions for installing minikube on Linux, MAC OS and Windows can be found on

https://kubernetes.io/docs/tasks/tools/install-minikube/

3. Hello World

Now go to

https://kubernetes.io/docs/setup/learning-environment/minikube/

and try out the Quickstart guide for minikube and see that it works ok.

4 Set up a webserver and scale it

Now it is time to try to set up a webserver and expose and scale it.

On the Kubernetes master, we can use kubectl run to create a certain number of containers. The Kubernetes master will then schedule the pods for the nodes to run, with the general command formatting, as follows:

$ kubectl run <replication controller name> --image=<image name> --replicas=<number of replicas> --port=<exposing port>

The suggested HTTP server is nginx and therefore use **image=nginx** , but of course you are free to use any other application(image).

Then scale the app and expose the port for external access

Finally, access the webserver via a web client

Questions:

a) What does the web browser show when you access the web server?
b) What is the output from the **kubectl get pods** command
c) What is the output from the **kubectl get deployment** command
d) What is the output from the **kubectl get services** command