Exercise Two - Converting the Docker-compose flow to Kubernetes (3%)

Before we start exercise 2 first, we delete all previous deployments and make sure that no pods are running on minikube cluster. To do this I use the command:

Kubectl delete deployment hello-minikube

Task 1 (0.5%): Install Kompose. This is the tool that we will use to convert the docker-compose file to Kubernetes.

I installed Kompose by running the command in cmd:

curl -L https://github.com/kubernetes/kompose/releases/download/v1.26.0/kompose-windows-amd64.exe -o kompose.exe

This installed kompose on my laptop.

Task 2 (0.5%): Use the source code from the first Docker tutorial (https://docs.docker.com/compose/gettingstarted/). You have to make some adjustments to the docker-compose file. You should make the following changes:

I followed the docker tutorial and created 4 files called app.py, dockerfile, requirements and docker-compose.yml and put the files into a folder called building_image which was in the ex2 folder.

I then ran the command docker compose up to build the image and run the code:

The docker compose up command aggregates the output of each container

After running this code, it opened a website at http://127.0.0.1:8000/.



I then created a repository in Docker hub https://hub.docker.com/ called waleeducd/getstarted2.0.

I then tagged the image I built in task 2 using the docker tutorial by using the command: docker tag

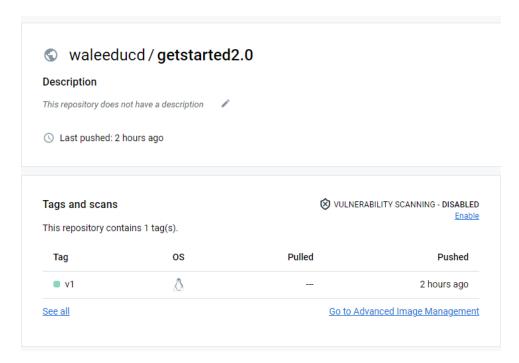
```
C:\Users\waleed\Desktop\PRACTICAL3_waleed_wazir_19396951\ex2>docker tag 92073afcefb9 waleeducd/getstarted2.0:v1
```

This will tag a local image with ID "92073afcefb9" into the "getstarted2.0" repository tagged as "v1" then I logged into my docker account by typing docker login.

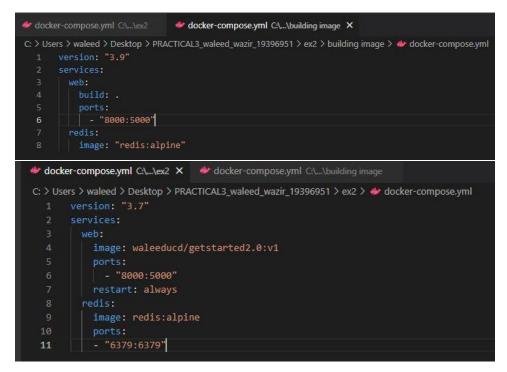
```
C:\Users\waleed\Desktop\PRACTICAL3_waleed_wazir_19396951\ex2>docker login
Authenticating with existing credentials...
Login Succeeded
```

I then pushed/published the image to my dockerhub using the command:

docker push waleeducd/getstarted2.0:v1 - this Pushs an image or a repository to a registry



After publishing the image to my repository, I went to the folder "building image" and copied and pasted the docker-compose.yml file it into "ex2" folder, I then opened up that file up and change the following:



version: "3.9" to "3.7", build: to image: and set image to waleeducd/getstarted and added the restart: policy set as always and I also defined the ports for redis service as you can see from the images.

Task 3 (1%): Convert the docker-compose file to Kubernetes, and create a deployment to your cluster

Now that I have changed the docker compose file I then convert the file to kubernetes and created deployments for our cluster using the command:

Kompose convert

This converts the docker-compose.yml file to files that you can use with kubectl

```
:\Users\waleed\Desktop\PRACTICAL3_waleed_wazir_19396951\ex2>kompose convert
INFO Kubernetes file "redis-service.yaml" created
INFO Kubernetes file "web-service.yaml" created
INFO Kubernetes file "redis-deployment.yaml" created
INFO Kubernetes file "web-deployment.yaml" created
```

I then added the deployments using the command **kubectl apply -f redis-service.yaml,web-service.yaml,redis-deployment.yaml,web-deployment.yaml**

This Applys a configuration to a resource

```
C:\Users\waleed\Desktop\PRACTICAL3_waleed_wazir_19396951\ex2>kubectl apply -f redis-service.yaml,web-service.yaml,redis-deployment.yaml,web-deployment.ya
service/redis configured
service/web configured
deployment.apps/redis configured
deployment.apps/web configured
```

Then using the command **minikube service name** this will automatically open our browser to the nodes proxy address.

I first ran minikube service redis

```
\Users\waleed\Desktop\PRACTICAL3_waleed_wazir_19396951\ex2>minikube service redis
NAMESPACE
           NAME
                    TARGET PORT
                                      URI
default
           redis
                                  No node port
service default/redis has no node port
Starting tunnel for service redis.
NAMESPACE
           NAME
                   TARGET PORT
                                           URI
default
           redis
                                  http://127.0.0.1:52288
Opening service default/redis in default browser...
Because you are using a Docker driver on windows, the terminal needs to be open to run it.
Stopping tunnel for service redis.
```

Then I ran the command **minikube service web** and a browser opened up similar to, the same output with the one you saw when you used the docker-compose.

NAMESPACE	NAME	TARGET PORT	URL
default	web		No node port
		eb has no node or service web	
NAMESPACE	NAME	TARGET PORT	URL
default	web		http://127.0.0.

← → C ① 127.0.0.1:52470

Hello Waleed! You've visited me 3 times.