Module-8: Container Orchestration using Kubernetes Part - I

Demo Document - 2

DEMO-2: Deployment in Kubernetes

Step 1:

Before putting the application on Kubernetes cluster, you need to perform the following steps:

a. Deploy a simple Docker container for webserver
b. Put a custom file for a static web-page
c. Create custom container image
d. Push it to Docker hub
e. Using the same image, deploy it on your Kubernetes cluster

a. Deploy a simple docker container for webserver

Note: You can choose either apache or nginx
We will use the latest release of the nginx and download from the default docker registry using the following command:

```
$sudo docker run -d -P --name webserver nginx
```



```
edureka@kmaster:~/webserver$ sudo docker run -d -P --name webserver nginx
9a3a7b9effcfeb238789dbb4355d0e4a9b9cba2383375d3209a2659f719e6675
```

Let's see the default page of nginx. Get the port on which it is exposed using this command:

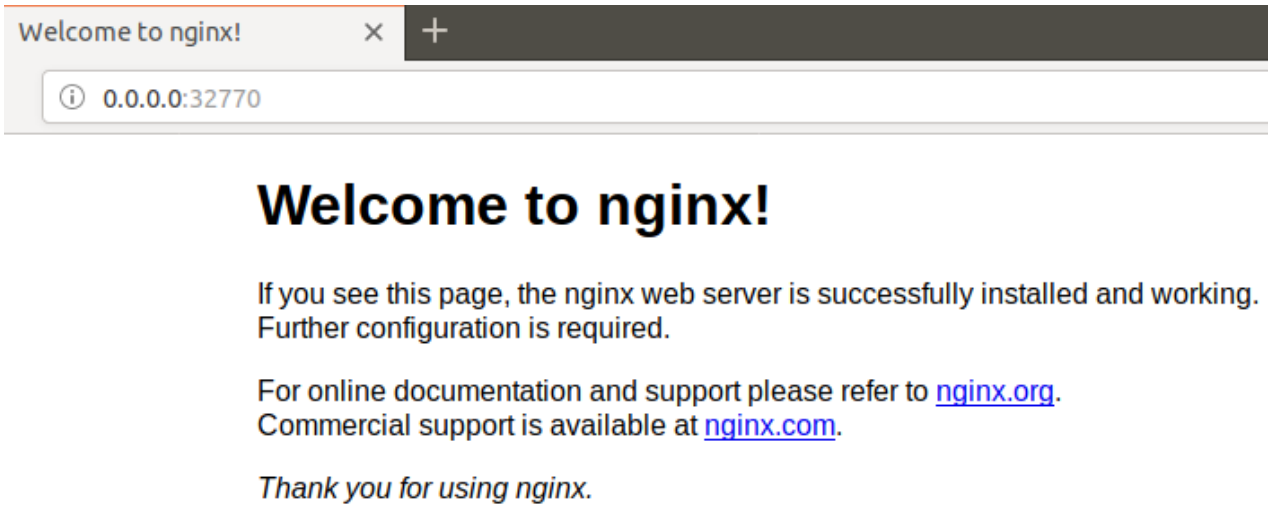
```
$sudo docker port webserver
```

```
edureka@kmaster:~/webserver$ sudo docker port webserver
80/tcp -> 0.0.0.0:32770
```

Note: webserver is the name of the container which we provided

Open the browser and use the routable IP to your host on which container is running. In our case, it is on host 'kmaster':

http://kmaster:32770



b. Put a custom file for a static webpage

- Create a directory webserver using these commands

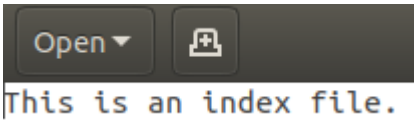
```
$mkdir webserver
cd webserver
```

- Create a file 'index.html' with some content to it using the command:

```
$gedit index.html
```

- View the contents of the file:

```
$cat index.html
```



- Under this, create a file 'Dockerfile' using the command:

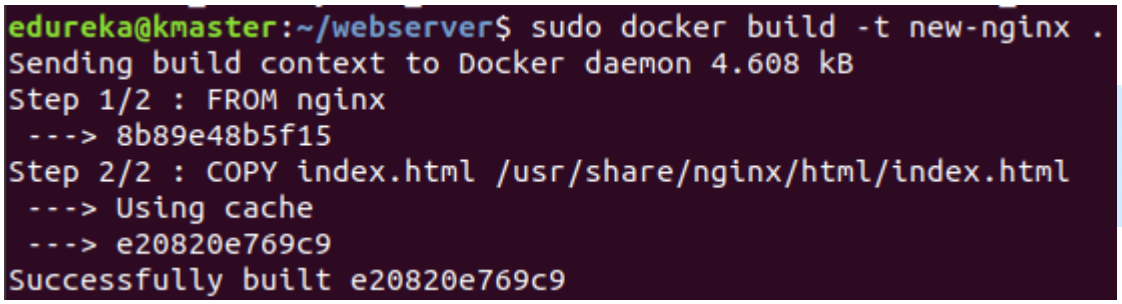
```
$gedit Dockerfile
```

Enter the following inside it:

```
FROM nginx
COPY index.html /usr/share/nginx/html/index.html
```

c. Create custom container image

```
$sudo docker build -t new-nginx .
```



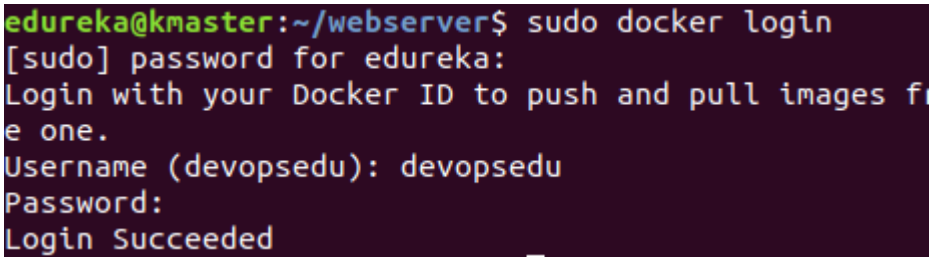
- Verify the image:

```
$sudo docker images
```

d. Push the image to the docker hub

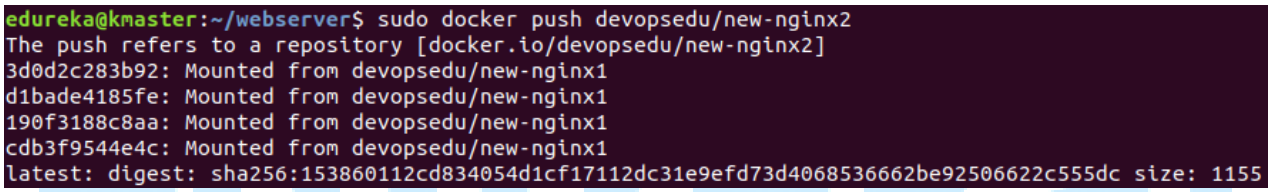
- First login to it:

```
$sudo docker login
```



- Then push the image:

```
$sudo docker push devopsedu/newnginx1
```



- Now, let's create a .yaml file to create Kubernetes deployment with 2 replicaset:

```
apiVersion: apps/v1
kind: Deployment
metadata:
 name: new-nginx-deployment
spec:
 selector:
  matchLabels:
   app: new-nginx1
 replicas: 2
 template:
  metadata:
   labels:
    app: new-nginx1
  spec:
   containers:
   - name: new-nginx
    image: devopsedu/new-nginx1
    ports:
    - containerPort: 80
```


e. Create the Kubernetes deployment

```
$kubectl create -f new-nginx1.yaml
$kubectl get deployments
```

```
edureka@kmaster:~/webserver$ kubectl create -f new-nginx1.yaml
deployment.apps/new-nginx-deployment created
```

```
edureka@kmaster:~/webserver$ kubectl get deployments
NAME                   DESIRED   CURRENT   UP-TO-DATE   AVAILABLE   AGE
new-nginx-deployment   2         2         2            2           2m
```