

Module-8: Container Orchestration using Kubernetes Part - I

Demo Document - 2

edureka!

edureka!

© Brain4ce Education Solutions Pvt. Ltd.

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  
9a3a7b9effcfefb238789dbb4355d0e4a9b9cba2383375d3209a2659f719e6675
```

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

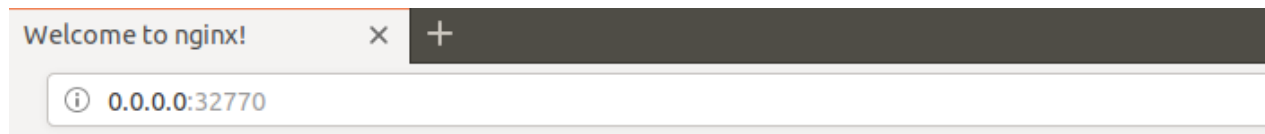
```
$sudo docker port webserver
```

```
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>



Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

b. Put a custom file for a static webpage

- Create a directory webserver using these commands

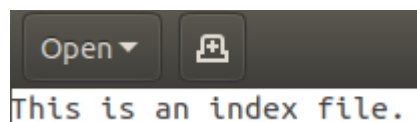
```
$mkdir webserver  
$cd webserver
```

- 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
```



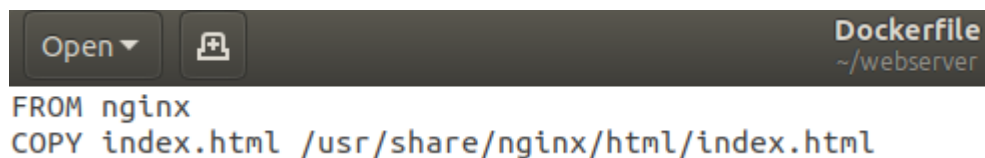
- Under this, create a file 'Dockerfile' using the command:

\$gedit Dockerfile

Enter the following inside it:

```
FROM nginx
```

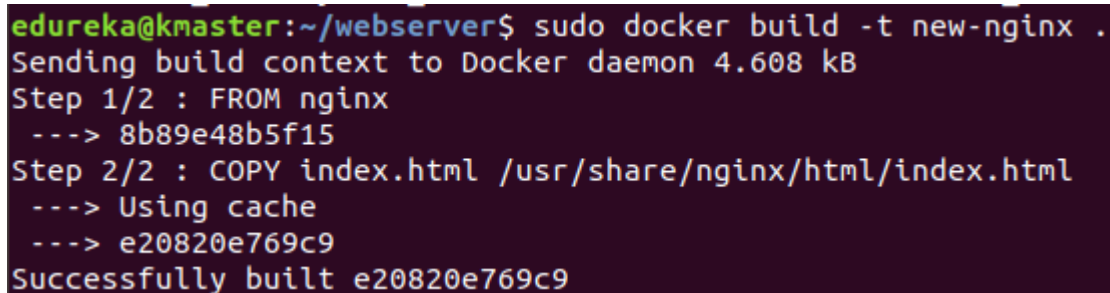
```
COPY index.html /usr/share/nginx/html/index.html
```

A screenshot of a code editor window. The title bar shows 'Dockerfile' and the path '~/webserver'. The editor contains two lines of text: 'FROM nginx' and 'COPY index.html /usr/share/nginx/html/index.html'.

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

c. Create custom container image

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

A screenshot of a terminal window showing the output of the 'docker build' command. The prompt is 'edureka@kmaster:~/webserver\$'. The output shows the build context being sent, two steps (FROM nginx and COPY index.html), and the final successful build of the 'new-nginx' image.

```
edureka@kmaster:~/webserver$ sudo docker build -t new-nginx .
Sending build context to Docker daemon 4.608 kB
Step 1/2 : FROM nginx
--> 8b89e48b5f15
Step 2/2 : COPY index.html /usr/share/nginx/html/index.html
--> Using cache
--> e20820e769c9
Successfully built e20820e769c9
```

- Verify the image:

```
$sudo docker images
```

A screenshot of a terminal window showing the output of the 'docker images' command. The prompt is 'edureka@kmaster:~/webserver\$'. The output is a table with columns: REPOSITORY, TAG, IMAGE ID, CREATED, and SIZE.

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
devopsedu/new-nginx1	latest	e20820e769c9	3 hours ago	109 MB
new-nginx1	latest	e20820e769c9	3 hours ago	109 MB

d. Push the image to the docker hub

- First login to it:

```
$sudo docker login
```

```
edureka@kmaster:~/webserver$ sudo docker login
[sudo] password for edureka:
Login with your Docker ID to push and pull images from a registry.
Username (devopsedu): devopsedu
Password:
Login Succeeded
```

- Then push the image:

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

```
edureka@kmaster:~/webserver$ sudo docker push devopsedu/new-nginx2
The push refers to a repository [docker.io/devopsedu/new-nginx2]
3d0d2c283b92: Mounted from devopsedu/new-nginx1
d1bade4185fe: Mounted from devopsedu/new-nginx1
190f3188c8aa: Mounted from devopsedu/new-nginx1
cdb3f9544e4c: Mounted from devopsedu/new-nginx1
latest: digest: sha256:153860112cd834054d1cf17112dc31e9efd73d4068536662be92506622c555dc size: 1155
```

- 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