120A3051

Shreya Idate

Batch: E3

Experiment No: 9

AIM: To learn Dockerfile instructions, build an image for a sample web application using Dockerfile.

THEORY:

Docker also gives you the capability to create your own Docker images, and it can be done with the help of Docker Files. A Docker File is a simple text file with instructions on how to build your images.

FROM keyword tells Docker, from which base image you want to base your image from. In our example, we are creating an image from the Ubuntu image.

The next command is the person who is going to maintain this image. Here you specify the MAINTAINER keyword and just mention the email ID.

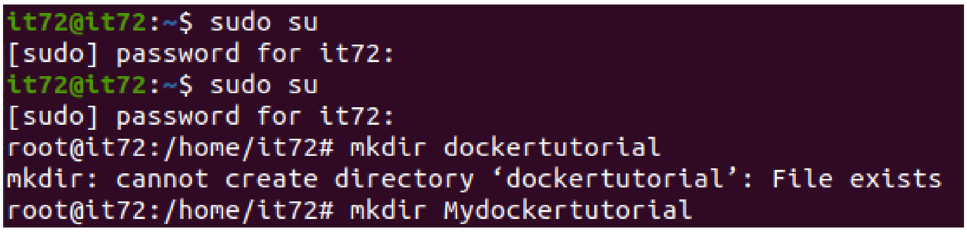
The RUN command is used to run instructions against the image. In our case, we first update our Ubuntu system and then install the nginx server on our Ubuntu image.

The last CMD command is used to display a message to the user.

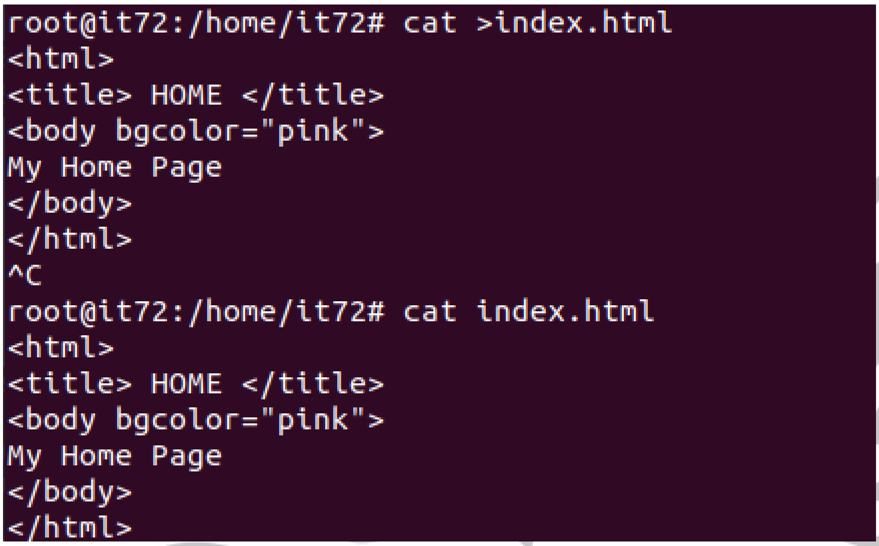
Create a file with a name as Dockerfile:

```
root@it77-OptiPlex-3050:/home/it77/dockertutorial# nano Dockerfile

FROM ubuntu:latest   # Adding a layer of Ubuntu as a base image

MAINTAINER "BUSHRA"  # Adding an author

RUN apt update –y     # Adding a layer of ‘apt-update’ in our image

RUN apt install nginx –y       # Adding a layer of ‘nginx’ webserver in our image EXPOSE 80     # Exposing port 80 of web server

COPY index.html /var/www/html/index.html # Hosting our web page / application CMD ["nginx", "-g", "daemon off;"]           # Starting nginx webserver in foreground

root@it77-OptiPlex-3050:/home/it77/dockertutorial# docker build -t siesnginxservers .

root@it77-OptiPlex-3050:/home/it77/dockertutorial# docker run -itd -p 8989:80 siesnginxservers root@it77-OptiPlex-3050:/home/it77/dockertutorial# ufw allow 8989
```

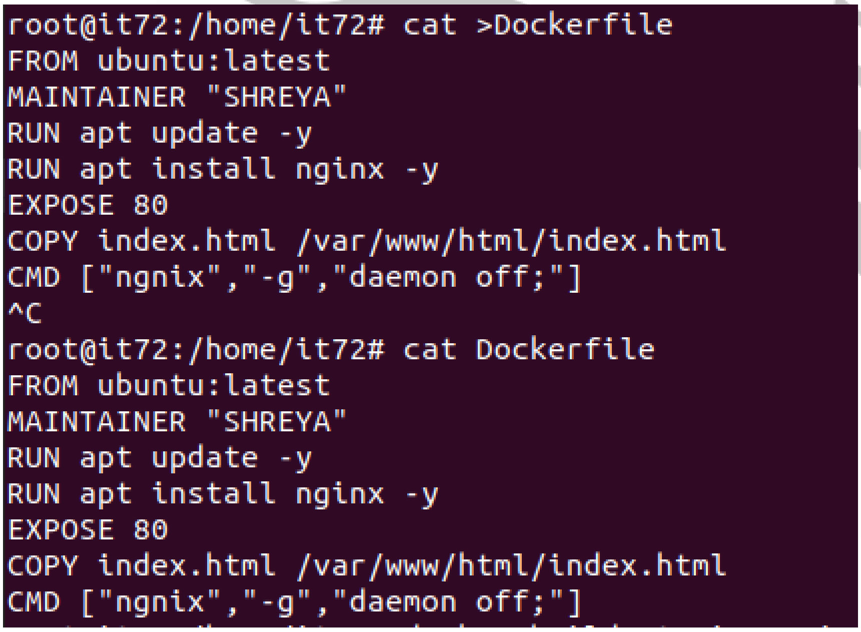
Open browser and put IP/localhost with port number as 8989 and run your web application.



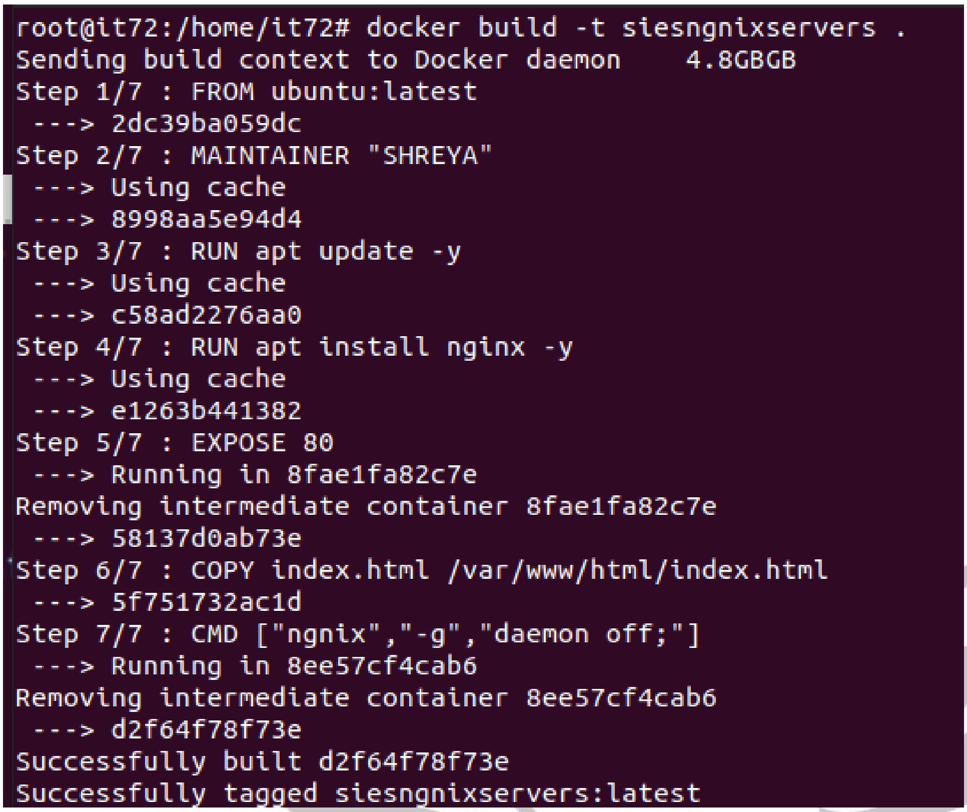
```
it72@it72:~$ sudo su
[sudo] password for it72:
it72@it72:~$ sudo su
[sudo] password for it72:
root@it72:/home/it72# mkdir dockertutorial
mkdir: cannot create directory ‘dockertutorial’: File exists
root@it72:/home/it72# mkdir Mydockertutorial
```



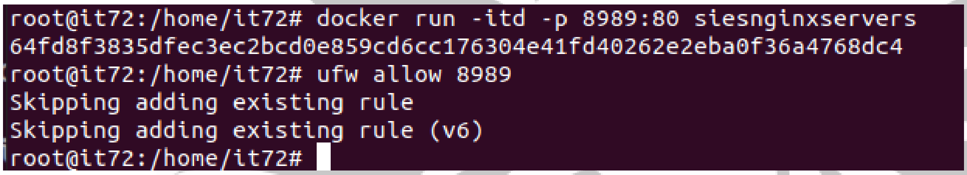
```
root@it72:/home/it72# cat >index.html
<html>
<title> HOME </title>
<body bgcolor="pink">
My Home Page
</body>
</html>
^C
root@it72:/home/it72# cat index.html
<html>
<title> HOME </title>
<body bgcolor="pink">
My Home Page
</body>
</html>
```



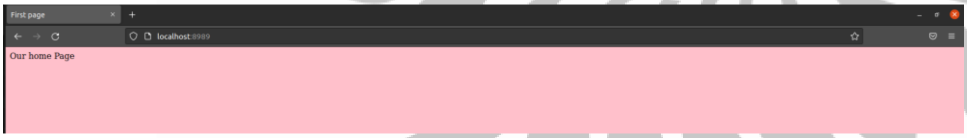
```
root@it72:/home/it72# cat >Dockerfile
FROM ubuntu:latest
MAINTAINER "SHREYA"
RUN apt update -y
RUN apt install nginx -y
EXPOSE 80
COPY index.html /var/www/html/index.html
CMD ["ngnix","-g","daemon off;"]
^C
root@it72:/home/it72# cat Dockerfile
FROM ubuntu:latest
MAINTAINER "SHREYA"
RUN apt update -y
RUN apt install nginx -y
EXPOSE 80
COPY index.html /var/www/html/index.html
CMD ["ngnix","-g","daemon off;"]
```



```
root@it72:/home/it72# docker build -t siesngnixservers .
Sending build context to Docker daemon    4.8GBGB
Step 1/7 : FROM ubuntu:latest
 ---> 2dc39ba059dc
Step 2/7 : MAINTAINER "SHREYA"
 ---> Using cache
 ---> 8998aa5e94d4
Step 3/7 : RUN apt update -y
 ---> Using cache
 ---> c58ad2276aa0
Step 4/7 : RUN apt install nginx -y
 ---> Using cache
 ---> e1263b441382
Step 5/7 : EXPOSE 80
 ---> Running in 8fae1fa82c7e
Removing intermediate container 8fae1fa82c7e
 ---> 58137d0ab73e
Step 6/7 : COPY index.html /var/www/html/index.html
 ---> 5f751732ac1d
Step 7/7 : CMD ["ngnix","-g","daemon off;"]
 ---> Running in 8ee57cf4cab6
Removing intermediate container 8ee57cf4cab6
 ---> d2f64f78f73e
Successfully built d2f64f78f73e
Successfully tagged siesngnixservers:latest
```



```
root@it72:/home/it72# docker run -itd -p 8989:80 siesnginxservers
64fd8f3835dfec3ec2bcd0e859cd6cc176304e41fd40262e2eba0f36a4768dc4
root@it72:/home/it72# ufw allow 8989
Skipping adding existing rule
Skipping adding existing rule (v6)
root@it72:/home/it72#
```



Our home Page

CONCLUSION: Successfully learnt Dockerfile instructions, built an image for a sample web application using Dockerfile.