APPLICATION CONTAINERIZATION

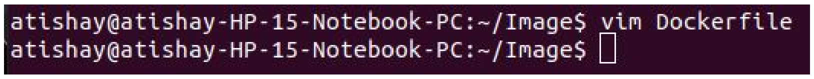
LAB EXPERIMENT 4

Creating a Docker image and pushing it to Docker Hub

A Dockerfile is nothing but a text document that contains all the commands a user requires to run an image.

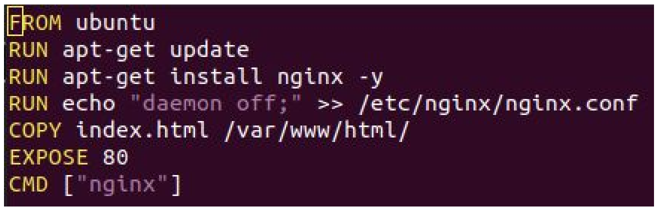
Now let us see steps to create a Dockerfile and commands to run an image using that file.

1. Create a text document and name it “Dockerfile”



```
atishay@atishay-HP-15-Notebook-PC:~/Image$ vim Dockerfile
atishay@atishay-HP-15-Notebook-PC:~/Image$
```

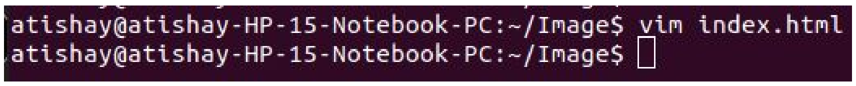
2. Open that document in any editor of your choice, write the following commands and save the document.



```
FROM ubuntu
RUN apt-get update
RUN apt-get install nginx -y
RUN echo "daemon off;" >> /etc/nginx/nginx.conf
COPY index.html /var/www/html/
EXPOSE 80
CMD ["nginx"]
```

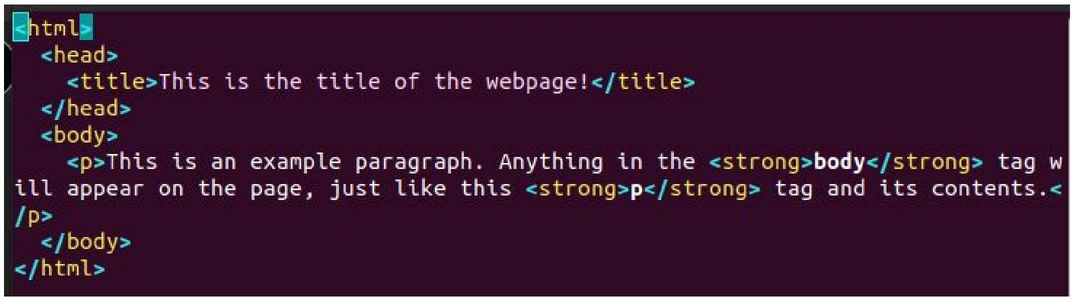
This code will execute in layers as follows:
Layer 1: Ubuntu image will be used as base OS image.
Layer 2: The image will be updated.
Layer 3: Nginx will be installed on the Ubuntu Image.
Layer 4: Configuration file of Nginx will be altered to set daemon off.
Layer 5: A file named index.html will be copied from current directory to /var/www/html.
Layer 6: Port 80 of the container will be exposed.
Layer 7: Nginx will be started.

3. Now create another text document and name it “index.html”



```
atishay@atishay-HP-15-Notebook-PC:~/Image$ vim index.html
atishay@atishay-HP-15-Notebook-PC:~/Image$
```

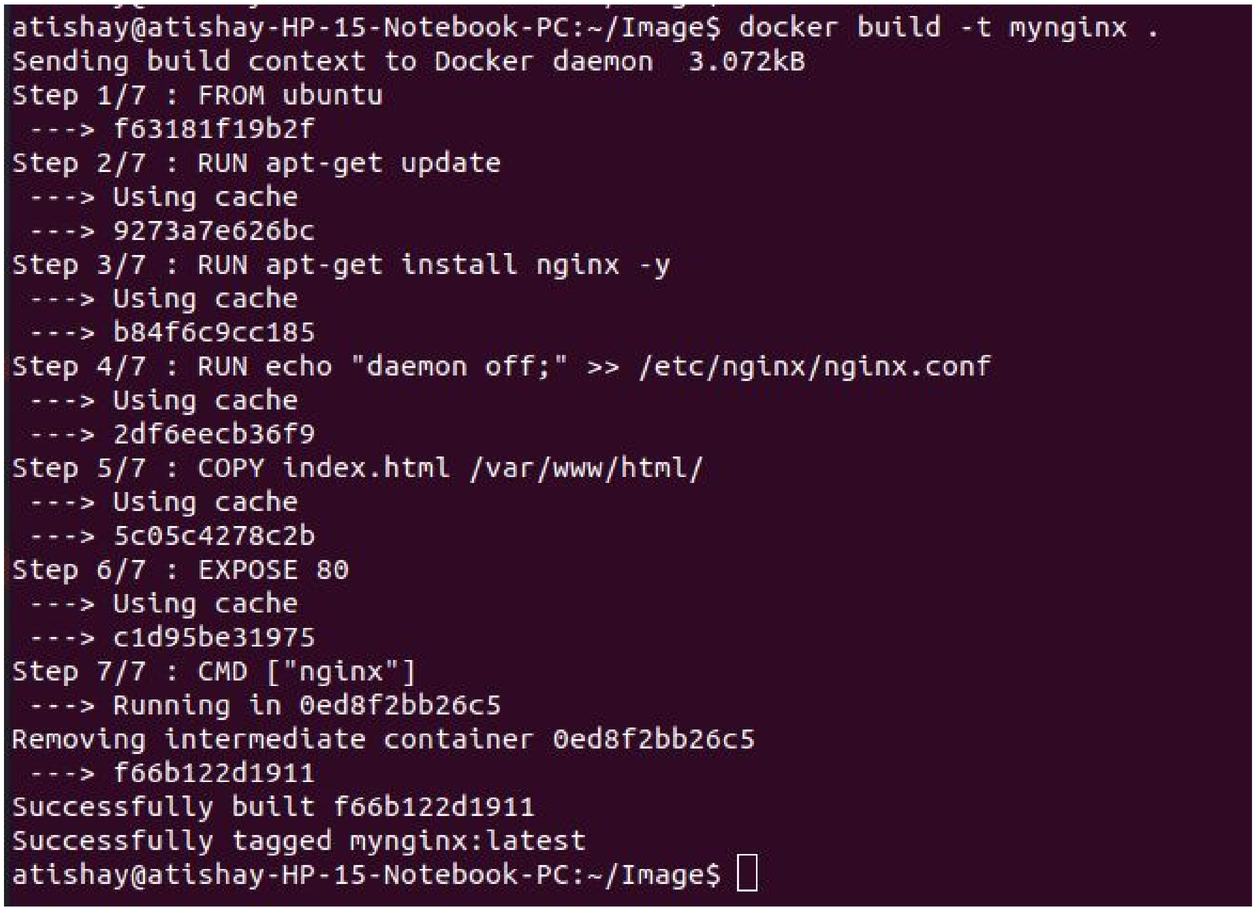
4. Open that document, write the following commands and save the document.



```
<html>
  <head>
    <title>This is the title of the webpage!</title>
  </head>
  <body>
    <p>This is an example paragraph. Anything in the <strong>body</strong> tag w
ill appear on the page, just like this <strong>p</strong> tag and its contents.<
/p>
  </body>
</html>
```

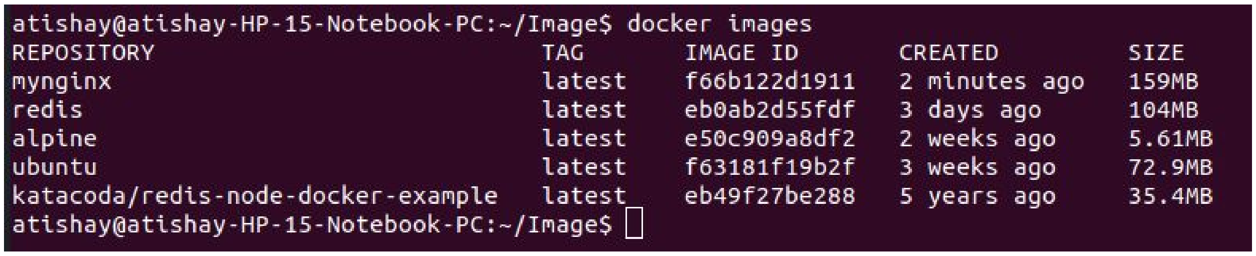
5. Now run this Dockerfile to build a Docker Image.

Command: docker build .
(Here, "." specifies the current directory)
(Here, "option "-t" is used to tag the image with a name and tag optionally. "mynginx" is the name and tag is not provided)



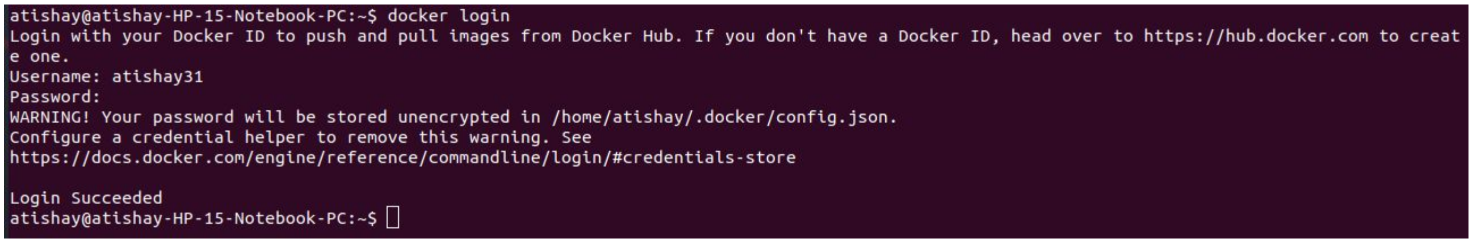
```
atishay@atishay-HP-15-Notebook-PC:~/Image$ docker build -t mynginx .
Sending build context to Docker daemon  3.072kB
Step 1/7 : FROM ubuntu
 ---> f63181f19b2f
Step 2/7 : RUN apt-get update
 ---> Using cache
 ---> 9273a7e626bc
Step 3/7 : RUN apt-get install nginx -y
 ---> Using cache
 ---> b84f6c9cc185
Step 4/7 : RUN echo "daemon off;" >> /etc/nginx/nginx.conf
 ---> Using cache
 ---> 2df6eecb36f9
Step 5/7 : COPY index.html /var/www/html/
 ---> Using cache
 ---> 5c05c4278c2b
Step 6/7 : EXPOSE 80
 ---> Using cache
 ---> c1d95be31975
Step 7/7 : CMD ["nginx"]
 ---> Running in 0ed8f2bb26c5
Removing intermediate container 0ed8f2bb26c5
 ---> f66b122d1911
Successfully built f66b122d1911
Successfully tagged mynginx:latest
atishay@atishay-HP-15-Notebook-PC:~/Image$ 
```

The newly created image can be viewed by executing the following command.
Command: docker images



```
atishay@atishay-HP-15-Notebook-PC:~/Image$ docker images
REPOSITORY                            TAG       IMAGE ID       CREATED         SIZE
mynginx                               latest    f66b122d1911   2 minutes ago   159MB
redis                                 latest    eb0ab2d55fdf   3 days ago      104MB
alpine                                latest    e50c909a8df2   2 weeks ago     5.61MB
ubuntu                                latest    f63181f19b2f   3 weeks ago     72.9MB
katacoda/redis-node-docker-example    latest    eb49f27be288   5 years ago     35.4MB
atishay@atishay-HP-15-Notebook-PC:~/Image$ 
```

6. Now to push it to Docker Hub, first we have to login to Docker Hub.

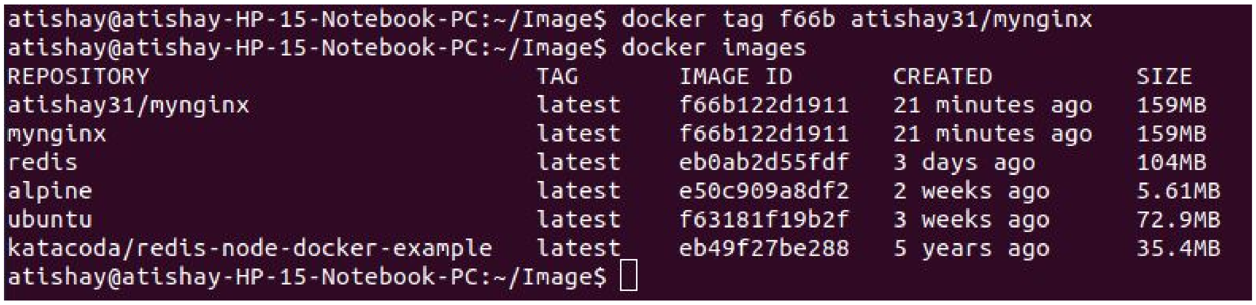


```
atishay@atishay-HP-15-Notebook-PC:~$ docker login
Login with your Docker ID to push and pull images from Docker Hub. If you don't have a Docker ID, head over to https://hub.docker.com to creat
e one.
Username: atishay31
Password:
WARNING! Your password will be stored unencrypted in /home/atishay/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
atishay@atishay-HP-15-Notebook-PC:~$ 
```

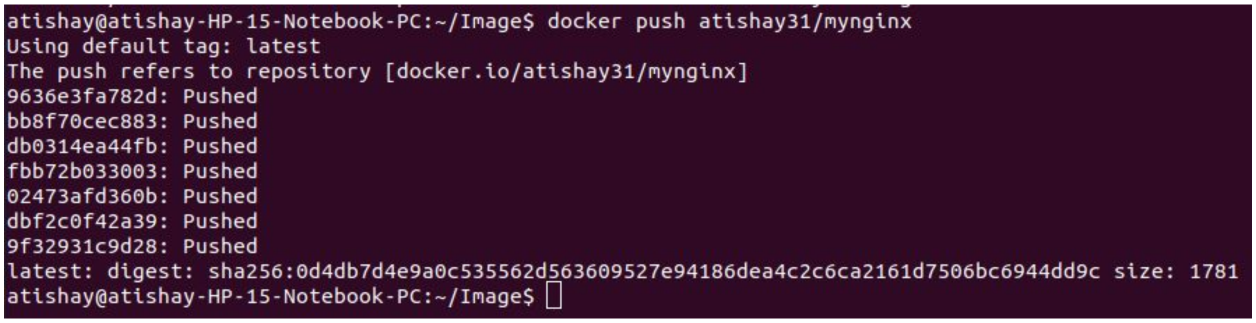
7. Now we will provide a tag to our image and the image can be listed with "docker images" command.

Command syntax: docker tag <image-id> <repository-with-tag>

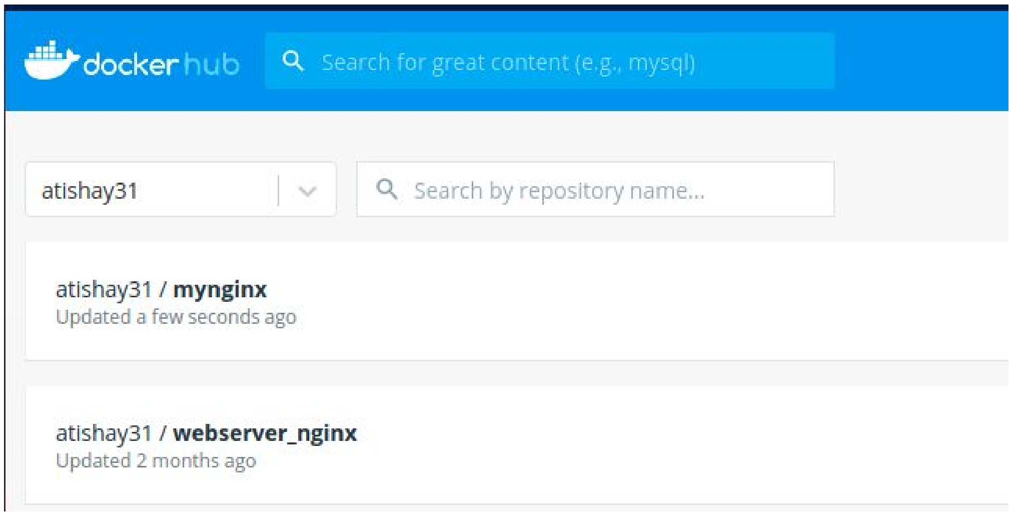


```
atishay@atishay-HP-15-Notebook-PC:~/Image$ docker tag f66b atishay31/mynginx
atishay@atishay-HP-15-Notebook-PC:~/Image$ docker images
REPOSITORY                            TAG       IMAGE ID       CREATED          SIZE
atishay31/mynginx                     latest    f66b122d1911   21 minutes ago   159MB
mynginx                               latest    f66b122d1911   21 minutes ago   159MB
redis                                 latest    eb0ab2d55fdf   3 days ago       104MB
alpine                                latest    e50c909a8df2   2 weeks ago      5.61MB
ubuntu                                latest    f63181f19b2f   3 weeks ago      72.9MB
katacoda/redis-node-docker-example    latest    eb49f27be288   5 years ago      35.4MB
atishay@atishay-HP-15-Notebook-PC:~/Image$ 
```

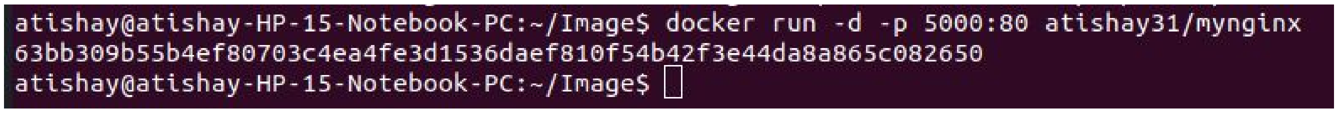
8. Push the image to Docker Hub.

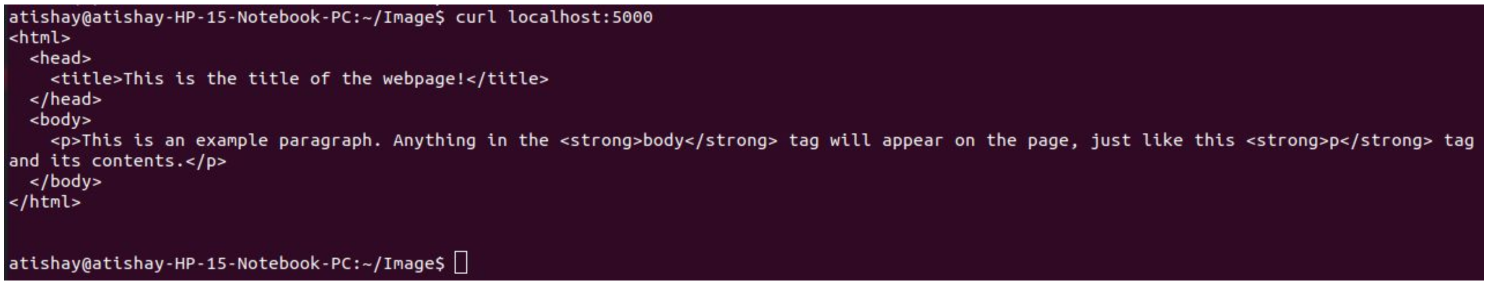

```
atishay@atishay-HP-15-Notebook-PC:~/Image$ docker push atishay31/mynginx
Using default tag: latest
The push refers to repository [docker.io/atishay31/mynginx]
9636e3fa782d: Pushed
bb8f70cec883: Pushed
db0314ea44fb: Pushed
fbb72b033003: Pushed
02473afd360b: Pushed
dbf2c0f42a39: Pushed
9f32931c9d28: Pushed
latest: digest: sha256:0d4db7d4e9a0c535562d563609527e94186dea4c2c6ca2161d7506bc6944dd9c size: 1781
atishay@atishay-HP-15-Notebook-PC:~/Image$
```

The image will be visible on our Docker Hub Repository.


docker hub
Search for great content (e.g., mysql)
atishay31
Search by repository name...
atishay31 / mynginx
Updated a few seconds ago
atishay31 / webserver_nginx
Updated 2 months ago

9. We can verify whether the image is working correctly. To do so, first bind the container's post to the host and then use curl command to check whether the webpage appears or not.


```
atishay@atishay-HP-15-Notebook-PC:~/Image$ docker run -d -p 5000:80 atishay31/mynginx
63bb309b55b4ef80703c4ea4fe3d1536daef810f54b42f3e44da8a865c082650
atishay@atishay-HP-15-Notebook-PC:~/Image$
```


```
atishay@atishay-HP-15-Notebook-PC:~/Image$ curl localhost:5000
<html>
  <head>
    <title>This is the title of the webpage!</title>
  </head>
  <body>
    <p>This is an example paragraph. Anything in the <strong>body</strong> tag will appear on the page, just like this <strong>p</strong> tag
and its contents.</p>
  </body>
</html>


atishay@atishay-HP-15-Notebook-PC:~/Image$
```