🕉️

Section 2 : Image Creation, Management and Registry.

Docker can build the images automatically by reading the information from the docker file.

Docker file is a text document which contains all the commands that a user should call on command line to assemble the image.

In order to build a image we need to have a docker file.

Where ever the docker file is located we need to go there and need to run the command **docker build (.)**.

Format for the Docker File:

- Here every docker file must start with a "**FROM**" instruction.
- From instruction tells the docker which is the base image from which the current docker file is being build from.
- We use # for comment.
- Every instruction comes in pair of instruction and argument.

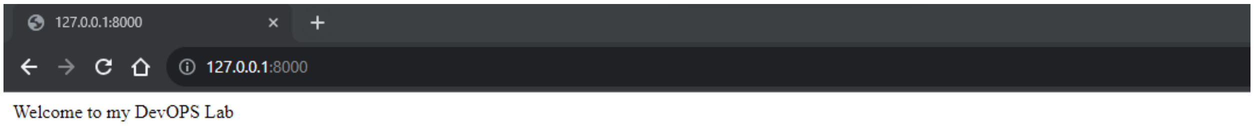
Use Case of Docker File:

▼ Manual approach to create an image:

Create a ubuntu container, Install Nginx on that container. Add a HTML file which says "Welcome to my Docker labs". Commit and save the changes and create a image in this way. This is a manual approach.

```
Commands
# First we will pull a docker image.
sudo docker pull ubuntu
# second step is to run the docker image
sudo docker run -dt -p 8000:80 ubuntu
# Check the container id of the docker image
```

```
sudo docker ps
# Enter into the container using the container id to make some changes into it.
sudo docker exec -dt <container id> bash
# *****inside the container****** run the update command
apt-get update
#Install the nginx on the container
apt-get install -y nginx
# go to the below location and search for index.nginx-debian.html file and overwrite it
cd var/www/html
#overwrite the file with the below message
echo "Welcome to my DevOPs lab" > index.nginx-debian.html
#check the status of nginx service if it is off then start it and exist from the container.
service nginx staus
service nginx start
exit
#Check the contianer ID and commit the conainer with new image name
sudo docker commit <container id> <name>
#RUN THE new image
sudo docker run -dt -p 8000:80 <new image name>
#Enter into the container which was made by new image
sudo docker exec -dt <new image container id> bash
#bring up the nginx service.
service nginx status
service nginx start
#check the ip address at the browser.
```



Welcome to my DevOPS Lab

▼ Use case of docker file:

1. Create a directory. Add a docker file to it. Edit with nano command and add three commands.
2. From command, update command, install nginx command.
3. Come out of this file by first saving it and create a html file with same name as there it is in by default under nginx.
4. Copy that nginx index.html file to the docker file. After copying it to the docker file add last command that runs nginx at the start.
5. Build this docker file.

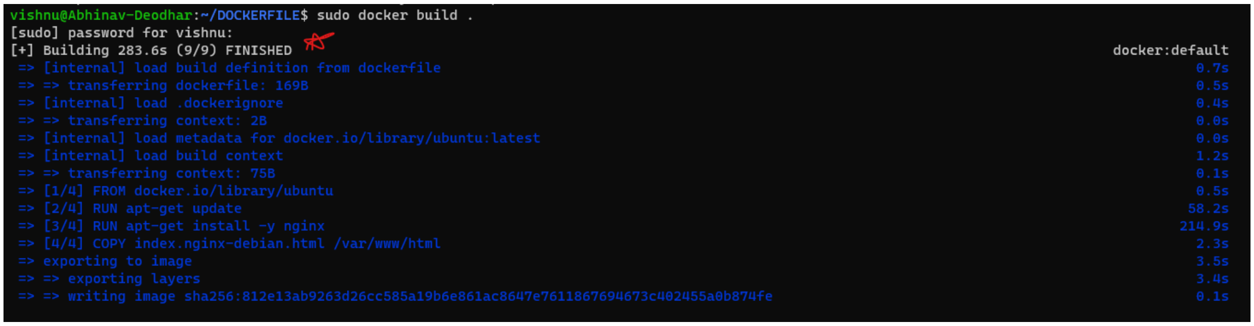
```
#writting from command.
From Ubuntu
RUN apt-get update
RUN apt-get install -y nginx
#save this file and create the html file with default name.
COPY index.nginx-debian.html /var/www/html
CMD nginx -g 'daemon off;'
#We are done with the image build.

#go to the default location and build docker the docker image.
```

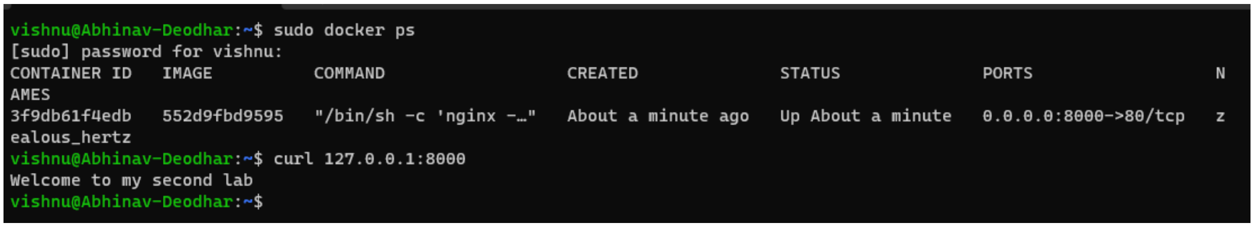
Name of the html file in nginx → index.nginx-debian.html

Location of the file var/www/html under nginx

```
#we will write to index.nginx.html file
"Welcme to my second labs"
```



```
vishnu@Abhinav-Deodhar:~/DOCKERFILE$ sudo docker build .
[sudo] password for vishnu:
[+] Building 283.6s (9/9) FINISHED                                                              docker:default
 => [internal] load build definition from dockerfile                                                      0.7s
 => => transferring dockerfile: 169B                                                                      0.5s
 => [internal] load .dockerignore                                                                         0.4s
 => => transferring context: 2B                                                                           0.0s
 => [internal] load metadata for docker.io/library/ubuntu:latest                                          0.0s
 => [internal] load build context                                                                         1.2s
 => => transferring context: 75B                                                                          0.1s
 => [1/4] FROM docker.io/library/ubuntu                                                                   0.5s
 => [2/4] RUN apt-get update                                                                             58.2s
 => [3/4] RUN apt-get install -y nginx                                                                  214.9s
 => [4/4] COPY index.nginx-debian.html /var/www/html                                                      2.3s
 => exporting to image                                                                                    3.5s
 => => exporting layers                                                                                   3.4s
 => => writing image sha256:812e13ab9263d26cc585a19b6e861ac8647e7611867694673c402455a0b874fe              0.1s
```



```
vishnu@Abhinav-Deodhar:~$ sudo docker ps
[sudo] password for vishnu:
CONTAINER ID   IMAGE          COMMAND                  CREATED              STATUS              PORTS                  N
AMES
3f9db61f4edb   552d9fbd9595   "/bin/sh -c 'nginx -…"   About a minute ago   Up About a minute   0.0.0.0:8000->80/tcp   z
ealous_hertz
vishnu@Abhinav-Deodhar:~$ curl 127.0.0.1:8000
Welcome to my second lab
vishnu@Abhinav-Deodhar:~$
```

ADD and Copy Command:

Both command servers similar purpose in docker. Copy command is used to copy the file from the source to destination.

ADD command also does the same and in addition it decompress the content of the file while copying to the destination location.

Use of ADD is not encouraged by docker as it creates more layers to the docker image. Hence use of curl command is more in practice.

EXPOSE instruction:

This is used as a documentation part to let the person who is running the image know that on which port the container service is running.

We can know it in the port section. If the expose command is absent in the docker file then there is no way to know on which port the docker image is running.

Health-check:

1. We can see the health of the container.
2. if the exist code is 0 then it means that the container is healthy.
3. when the exist code is 1 then the container is unhealthy.
4. We can have a docker file where the time interval of health check is specified and also a ping command is specified which will ping the ip address.