9. Host a html page of ubuntu container

Step 1 :- Installation in main machine

```
# sudo apt update -y
# sudo apt upgrade -y
# sudo apt-get install apache2 -y
# sudo apt-get install systemctl -y
# sudo systemctl start apache2
# sudo systemctl enable apache2
# sudo apt-get install curl
# sudo apt-get install docker.io
```

Step 2 : Pull ubuntu image

```
# sudo docker pull ubuntu
```

step 3 : Creating container of ubuntu image

```
# docker run --name V1 -it ubuntu:latest
    ↳ it will create ubuntu container with name V1
```

Step 4 : In V1 container

```
# sudo apt update -y
# apt upgrade -y
# apt-get install apache2
# apt-get install systemctl
# apt-get install vim -y
# sudo systemctl enable apache2
# sudo systemctl start apache2
```

```
# cd /var/www/html/
# rm -rf index.html  → remove present index.html file
# vim index.html
   ↳ welcome to cdac-acts pune
# exit
```

step 5 : In main machine

```
# sudo docker start v1
# sudo docker exec -it v1 systemctl start apache2
# sudo docker inspect v1  → it gives network info of v1 container.
```

then copy the ip address of v1 container & paste the ip address in the web browser it will show the text in the index.html file which is present in the v1 container.

Q. Host custom webpage using nginx image

step 1 : Installation in main machine

```
# sudo apt update -y
# sudo apt upgrade -y
# sudo apt-get install systemctl -y
# sudo apt-get install apache2 -y
# sudo systemctl enable apache2
# system ctl start apache 2
# sudo apt -get install curl
# sudo apt-get install docker.io
```

step2: Pull nginx image

```
# sudo docker pull nginx
```

step3: creating container of nginx image

```
# sudo docker run -d --name n1 -it nginx:latest
```

↳ it will create nginx containe with name n1

step 4: Getting shell of that container.

```
# sudo docker exec -it n1 /bin/bash
```

step 5: In n1 container.

```
# apt update -y
# apt upgrade -y
# apt-get install vim
# cd /usr/share/nginx/html
# rm -rf index.html
```

# vim index.html
↳ welcome to nginx
# exit

Step 6 : In main machine

# sudo docker inspect n1

Now copy the ip address of n1 container & paste it in the web browser, It will show the text in the index.html file which is present in the n1 container.

Q. to Host custom webpage on port 8100 using nginx image

step 1: Installation in main machine

```
# sudo apt update -y
# sudo apt upgrade -y
# sudo apt-get install systemctl -y
# sudo apt-get install apache2 -y
# sudo apt-get install curl
# sudo apt-get install docker.io
# sudo systemctl enable apache2
# sudo systemctl start apache2
```

step 2: pull nginx image

```
# sudo docker pull nginx
```

step 3: create nginx container which will run on 8100 port

```
# sudo docker run -d --name n1 -p 8100:80 nginx
  ↳ it will create container name n1
```

step 4: Getting shell of n1 container

```
# sudo docker exec -it n1 /bin/bash
```

step 5: In n1 container

```
# apt update -y
# apt upgrade -y
# apt-get install vim
```

```
# cd /usr/share/nginx/html/
# rm -rf index.html
# vim index.html
   ↳ welcome to nginx container 8100
# exit
```

step 6 : In main machine.

```
# ip a
```

then copy the ip address of VM machine & paste it in the web browser & then type :8100 after ip address.

example :- 192.168.20.157 : 8100

ip address of VM machine — port

this will print text in index.html file which is present in n1 container.

Q. Host custom web page by creating Ubuntu image using dockerfile

step 1: In main machine

# sudo apt update -y
# sudo apt upgrade -y
# sudo apt install systemctl -y
# sudo apt install ~~systemctl~~ apache2 -y
# sudo systemctl enable apache2
# sudo systemctl start apache2
# sudo apt install vim -y
# sudo apt install curl

step 2: create new directory

# sudo mkdir app1
# cd app1
# vim index.html
↳ welcome to core-acts Pune

step 3: create Dockerfile in app1 directory

# vim Dockerfile
↳ FROM ubuntu:latest
~~RUN apt upgrade -y.~~
RUN apt update -y
RUN apt upgrade -y
RUN apt install systemctl -y
RUN apt install apache2 -y

```
RUN systemctl enable apache2
RUN systemctl start apache2
RUN rm -rf /var/www/html/index.html
COPY . /var/www/html/
```

/app1 # sudo docker build -t newimage:1 .
↳ this it will create image by name newimage

step 4: create container of image (newimage)
# sudo docker run --name v1 -it newimage:1
↳ this will create container with ~~ma~~ name v1

step 5: In main machine
# sudo docker inspect v1
↳ this will provide detail information of v1.
now copy the ip address & paste it in web browser. it will show the text message in the index.html which is "welcome to CDAC-acts pune"

Q. Host custom index.html web page in nginx container using Dockerfile

step 1: In main machine

```
# sudo apt update -y
# sudo apt upgrade -y
# sudo apt install systemctl -y
# sudo apt install apache2 -y
# sudo systemctl start apache2
# sudo systemctl enable apache2
# sudo apt install vim -y
# sudo apt install curl -y
```

step 2: create new directory

```
# mkdir app2
# cd app2
# vim index.html
  ↳ welcome to nginx image
```

step3: create Dockerfile in app2 directory

```
# vim Dockerfile
  ↳ FROM nginx:latest
    RUN apt update -y
    RUN apt upgrade -y
    RUN rm -rf /usr/share/nginx/html/index.html
    COPY . /usr/share/nginx/html/
```

# sudo docker build -t nginximage:1 .
↳ this will create image by name nginximage

Step 4: create container of image nginximage.
# sudo docker run -d --name n1 -it nginx image:1
↳ this will create container with name n1

Step 5: Hosting web page
# sudo docker inspect n1
↳ this will provide detail info. of n1 container
Now copy the ip address of n1 container & paste it in the web browser, it will show the text message in the index.html file which is " welcome to nginx image "

Q. Host custom index.html webpage in httpd container using Dockerfile

Step 1 : installations in main machine

```
# sudo apt update -y
# sudo apt upgrade -y
# sudo apt install vim -y
# sudo apt install curl -y
# sudo apt install systemctl -y
# sudo apt install apache2 -y
# sudo apt systemctl start apache2
# sudo systemctl enable apache2
```

step 2 : create new directory

```
# mkdir app3
# cd app3
# vim index.html
  ↳ welcome to httpd image
```

step 3 : create Dockerfile in app3 directory

```
# vim Dockerfile
  ↳ FROM httpd : latest
    RUN apt update -y
    RUN apt upgrade -y
    RUN rm -rf /usr/local/apache2/htdocs/index.html
    COPY . /usr/local/apache2/htdocs/
```

#Sudo docker build -t httpdimage:1 .
↳ this will create image by name httpdimage

Step4: create container of image httpd image.
# sudo docker run -d --name h1 -it httpdimage:1
↳ this will create containe by name h1

Step5: Hosting web page.
# sudo docker inspect h1
↳ this will provide detail into of h1 container.
now copy the ip address of h1 container & paste it ~~on~~ in the web browser, it will show the text message in the index.html file which is "welcome to httpd image"

# 8. Push image to public Hub

step 1 : Login to to Docker hub in web browser then create repository of any name.

step 2 : Login docker ~~into~~ in VM terminal

```
# sudo docker login
```

↳ enter id & password of your docker hub account

step 3 : If you have made custom image using Dockerfile, you have to change its name to your docker repository name.

```
# sudo docker tag presentimage:tag newimage:tag
```

newimage should be your repository name
eg :- username / repository name

↳ this will create new image by name newimage with tag

Step 4 : Push image to public hub

```
# sudo docker push newimage:tag
```

this will push your image to your docker hub repo.

Q. transfer image by converting it into .tar file From machine 1 to machine 2.

& docker

Step 1 : install ssh into both machines.

# sudo apt install ssh -y
# sudo apt install docker.io -y

step 2 : in machine 1

create new image or pull any image From docker hub

# sudo docker pull ubuntu : latest

step 3 : In machine 1

create tar file of ubuntu image

# sudo docker save ubuntu : latest > ubuntu.tar

↳ this will create tar file of ubuntu image.

step 4 : Now in machine 2 Find its ip address

# sudo ip a

↳ Now copy the ip address of machine 2

step 5 : Now In machine 1

/home/machi

# scp ubuntu.tar machine 2@machin 2 ip addr.

↳ this will share .tar file from machine 1 to machine 2

step 6 : Extract .tar file in machine 2

# sudo docker load < ubuntu.tar

↳ this will extract .tar file in image