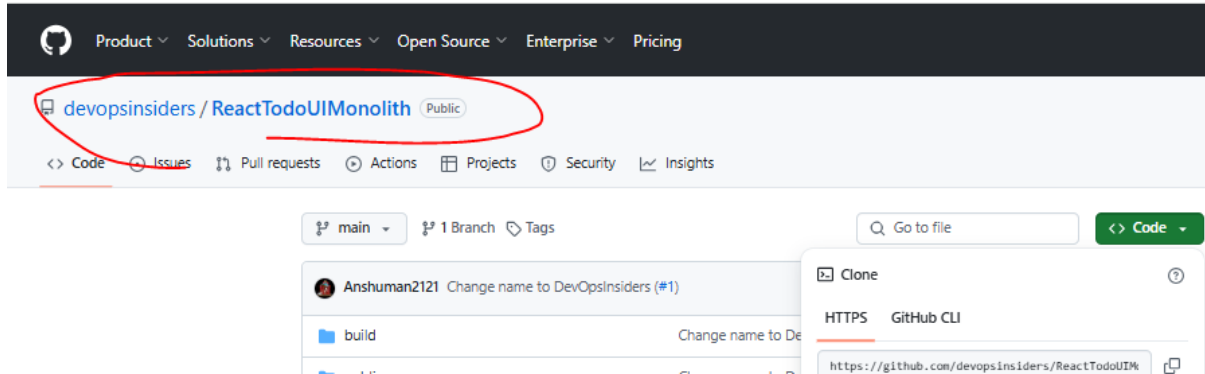


- 1) Since react JS code cannot be put on nginx directly so we have to firstly convert it into html, css, javascript
- 2) Create vm, install docker,
- 3) git clone <https://github.com/devopsinsiders/ReactTodoUIMonolith.git>



```
azureuser@vmdocker:~$ git clone https://github.com/devopsinsiders/ReactTodoUIMonolith.git
Cloning into 'ReactTodoUIMonolith'...
remote: Enumerating objects: 107, done.
remote: Counting objects: 100% (14/14), done.
remote: Compressing objects: 100% (6/6), done.
remote: Total 107 (delta 13), reused 8 (delta 8), pack-reused 93 (from 1)
Receiving objects: 100% (107/107), 1.25 MiB | 4.09 MiB/s, done.
Resolving deltas: 100% (49/49), done.
azureuser@vmdocker:~$
```

- 4) **ls**

cd ReactTodoUIMonolith/

ls

```
azureuser@vmgg:~$ ls
ReactTodoUIMonolith  snap
azureuser@vmgg:~$ cd ReactTodoUIMonolith/
azureuser@vmgg:~/ReactTodoUIMonolith$ ls
README.md  build  package-lock.json  package.json  public  src
```

- 5) **rm -r build** – remove build folder

ls

```
azureuser@vmgg:~/ReactTodoUIMonolith$ rm -r build
azureuser@vmgg:~/ReactTodoUIMonolith$ ls
README.md  package-lock.json  package.json  public  src
azureuser@vmgg:~/ReactTodoUIMonolith$
```

- 6) **touch Dockerfile**

ls

```
azureuser@vmdocker:~/ReactTodoUIMonolith$ touch Dockerfile
azureuser@vmdocker:~/ReactTodoUIMonolith$ ls
Dockerfile  README.md  package-lock.json  package.json  public  src
azureuser@vmdocker:~/ReactTodoUIMonolith$
```

7) Only build folder has to be put on nginx

8)

```
azureuser@vmdocker: ~/ReactTodoUIMonolith
GNU nano 7.2
FROM node:16.20.2
WORKDIR /dhondhu
COPY . .
RUN npm install
RUN npm run build

RUN apt update && apt install -y nginx
RUN cp -r /dhondhu/build/* /var/www/html/
```

FROM node:16.20.2

WORKDIR /dhondhu

COPY . .

RUN npm install

RUN npm run build

RUN apt update && apt install -y nginx

RUN cp -r /dhondhu/build/* /var/www/html/

9) **sudo docker buildx build -t todoui .** = this command builds a Docker image from the current directory, tags it as todoui, and uses Docker's extended build features

10) **docker images**

```
azureuser@vmdocker:~/ReactTodoUIMonolith$ sudo docker images
REPOSITORY    TAG       IMAGE ID       CREATED        Size
todoui        latest   f1ebcad628d3   5 minutes ago  1.34GB
azureuser@vmdocker:~/ReactTodoUIMonolith$
```

11) **docker run -p 8080:80 todoui** = run to create container using image **todoui**

```
root@vmdocker:/home/azureuser# docker run -p 8080:80 todoui
root@vmdocker:/home/azureuser# docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS          NAMES
```

12) **docker ps -a**

```
root@vmdocker:/home/azureuser# docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS          NAMES
ecd8844daf9a   todoui    "docker-entrypoint.s..."  31 seconds ago  Exited (0) 29 seconds ago          tender_davinci
root@vmdocker:/home/azureuser#
```

13) So above issue came that container started and exited or stopped after some time, So let us go inside working directory of container i.e. dhondhu

docker run -it -p 8080:80 todoui bash

ls

```
root@vmdocker:/home/azureuser# docker run -it -p 8080:80 todoui bash
root@a20d8b54e59d:/dhondhu# ls
Dockerfile README.md build node_modules package-lock.json package.json public src
root@a20d8b54e59d:/dhondhu#
```

14) **service nginx status** = command to check status of nginx which shows it is not running

```
root@a20d8b54e59d:/dhondhu/build# service nginx status
[....] nginx is not running ...
```

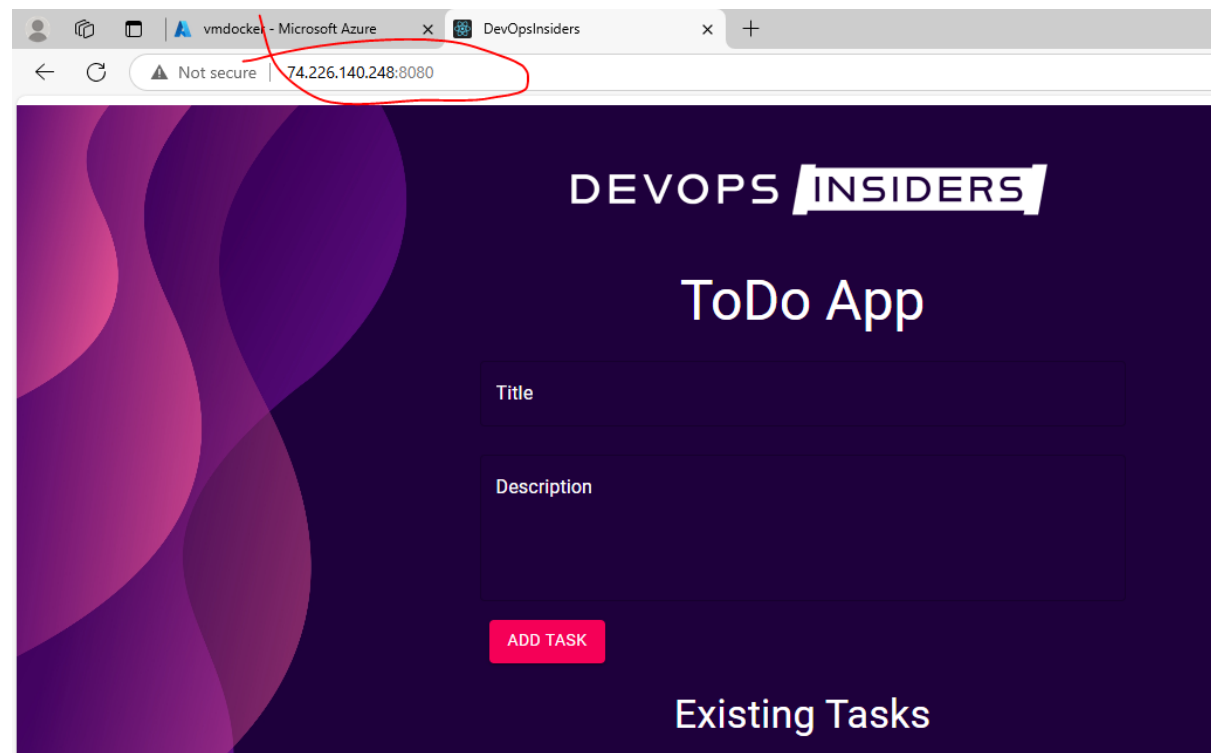
15) **service nginx start** = command to start nginx

```
root@a20d8b54e59d:/dhondhu/build# service nginx start
[ ok ] Starting nginx: nginx.
```

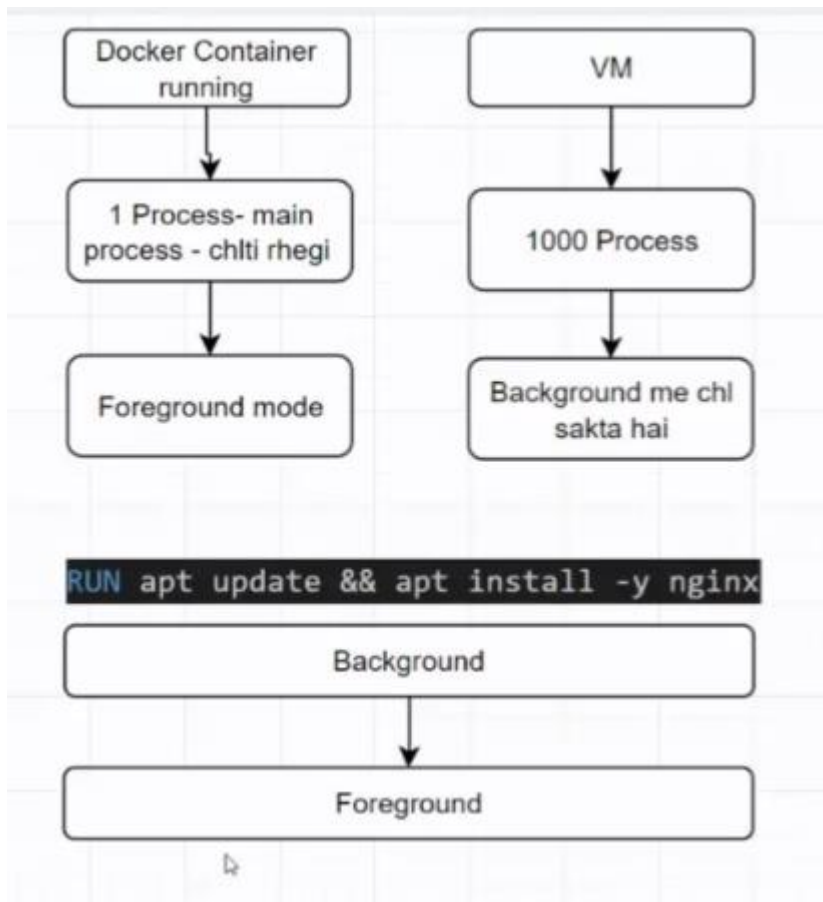
service nginx status

```
root@a20d8b54e59d:/dhondhu/build# service nginx status
[ ok ] nginx is running.
root@a20d8b54e59d:/dhondhu/build#
```

16) 74.226.140.248:8080



17)



Agenda – How nginx can be brought from background to foreground

To run Nginx in the foreground using the command line, use this command:

```
nginx -g 'daemon off;'
```

nginx -g 'daemon off;'

1) This command is used to run our nginx container in foreground.

```

azureuser@vmdocker: ~/ReactTodoUIMonolith
GNU nano 7.2 Dockerfile
FROM node:16.20.2
WORKDIR /dhondhu
COPY . .
RUN npm install
RUN npm run build

RUN apt update && apt install -y nginx #nginx running in background
RUN cp -r /dhondhu/build/* /var/www/html/
RUN nginx -g 'daemon off;' #to run nginx in foreground
  
```

2) **sudo docker build -t todoui .** = command to build a docker image named as todoui

So this time “**RUN**” was used in daemon off command so image was continuously building. So basically using RUN a temporary container gets created.

3) Using “**CMD**” a temporary container do not gets created.

```
azureuser@vmdocker: ~/ReactTodoUIMonolith
GNU nano 7.2
FROM node:16.20.2
WORKDIR /dhondhu
COPY . .
RUN npm install
RUN npm run build

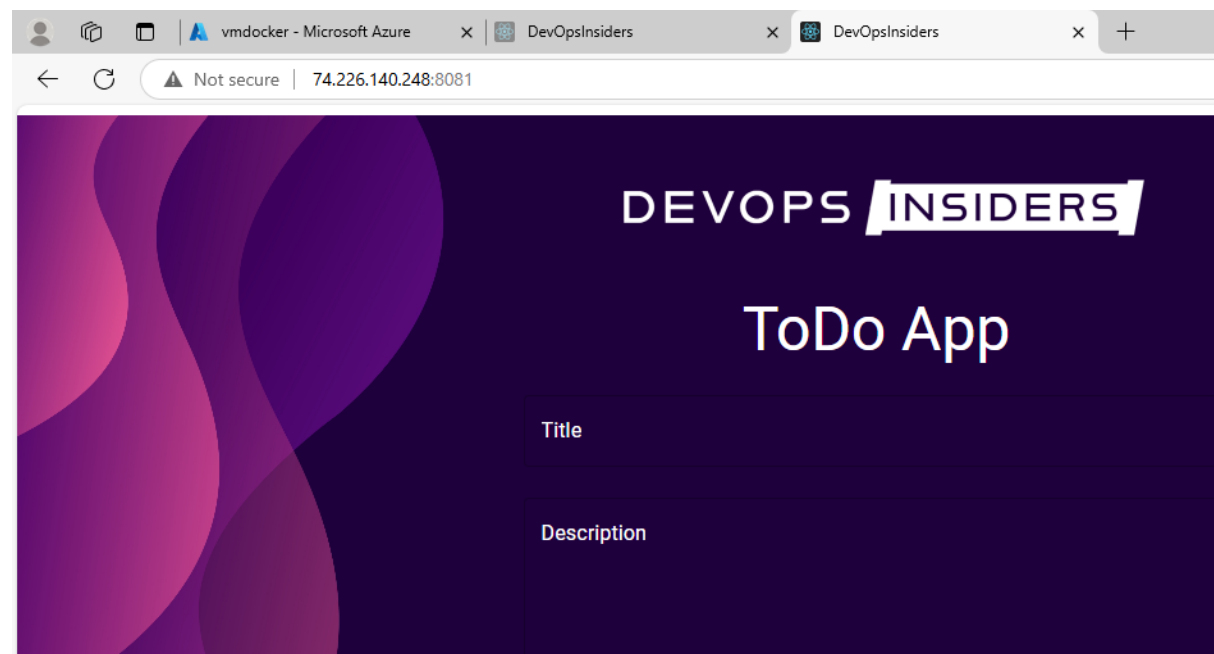
RUN apt update && apt install -y nginx      #nginx running in background
RUN cp -r /dhondhu/build/* /var/www/html/
CMD nginx -g 'daemon off;'                #to run nginx in foreground
```

sudo docker build -t todoui . = command to build a docker image named as todoui

docker images

```
root@vmdocker:/home/azureuser/ReactTodoUIMonolith# docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
todoui        latest    4d09465a1759   3 minutes ago  1.34GB
<none>        <none>    flebcad628d3   29 hours ago   1.34GB
root@vmdocker:/home/azureuser/ReactTodoUIMonolith#
```

4) 74.226.140.248:8081



5) Now to run container in background “-d”. So the container will go to background but processes will be in foreground only.

```
root@vmdocker:/home/azureuser/ReactTodoUIMonolith# docker run -d -p 8082:80 todoui
e4984ca4e839564c0e32c8cf68907f565af1ed347ba0c4544950525d5b5b479a
root@vmdocker:/home/azureuser/ReactTodoUIMonolith#
```

*) **docker container commit** = plumber command

The screenshot shows the Docker documentation page for the 'docker container commit' command. The page is titled 'docker container commit' and includes a description, usage, and aliases. The usage is shown as 'docker container commit [OPTIONS] CONTAINER [REPOSITORY[:TAG]]'. The description states: 'Create a new image from a container's changes'. The aliases section shows 'docker commit'. The page also includes a sidebar with a list of Docker CLI reference topics and a 'Table of contents' on the right.

Agenda – Removing CMD command and running simple to create image

1) Now removing CMD command and running simple to create image

```
root@vmdocker:/home/azureuser/ReactTodoUIMonolith
GNU nano 7.2
FROM node:16.20.2
WORKDIR /dhondhu
COPY . .
RUN npm install
RUN npm run build

RUN apt update && apt install -y nginx #nginx running in background
RUN cp -r /dhondhu/build/* /var/www/html/
```

2) **sudo docker build -t todoui .** = command to build a docker image named as todoui

3) **docker images**

```
root@vmdocker:/home/azureuser/ReactTodoUIMonolith# docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
todoui        latest   feale545debc   3 minutes ago  1.34GB
<none>        <none>    4d09465a1759   19 hours ago  1.34GB
<none>        <none>    f1ebcad628d3   2 days ago    1.34GB
root@vmdocker:/home/azureuser/ReactTodoUIMonolith#
```

4) So again use CMD and after using we will get that while running we can override/ the default command.

Agenda – Using ENTRYPOINT command and running simple to create image

1) Write in Docker file

```
root@vmdocker: /home/azureuser/ReactTodoUIMonolith
GNU nano 7.2
FROM node:16.20.2
WORKDIR /dhondhu
COPY . .
RUN npm install
RUN npm run build

RUN apt update && apt install -y nginx      #nginx running in background
RUN cp -r /dhondhu/build/* /var/www/html/
ENTRYPOINT nginx -g 'daemon off;'
```

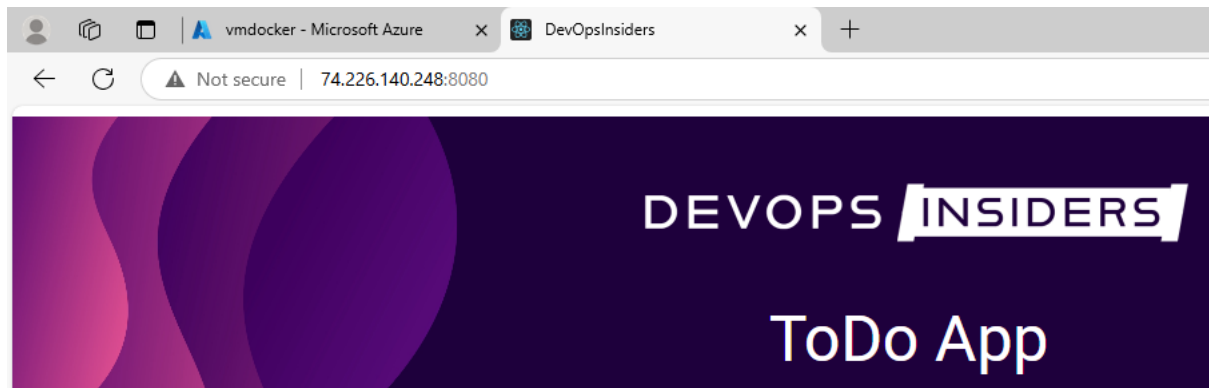
2) **sudo docker build -t todoui .** = command to build a docker image named as todoui

docker images

```
root@vmdocker: /home/azureuser/ReactTodoUIMonolith# docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
todoui        latest    8cfbe43bc4ca   29 seconds ago 1.34GB
<none>        <none>    fea1e545debc   2 hours ago   1.34GB
<none>        <none>    4d09465a1759   21 hours ago   1.34GB
<none>        <none>    f1ebcad628d3    2 days ago    1.34GB
root@vmdocker: /home/azureuser/ReactTodoUIMonolith#
```

3) **docker run -p 8080:80 todoui** = run command to create container

74.226.140.248:8080



4) Now checking by passing command after todoui in case of entry point.

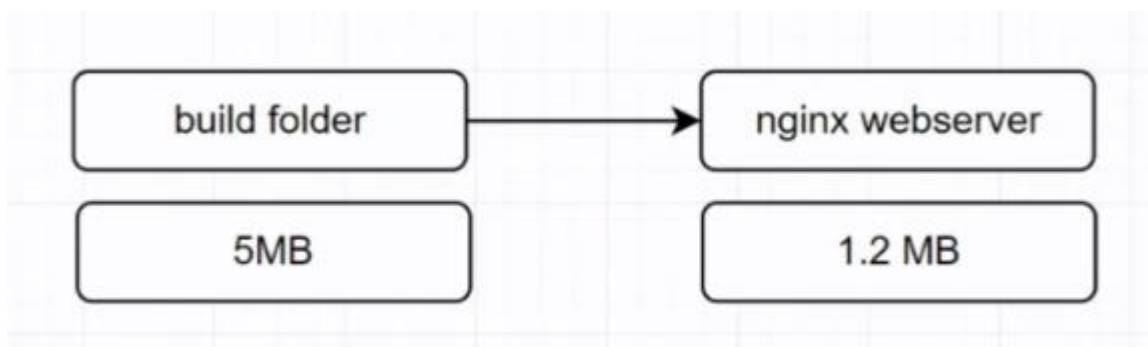
docker run -p 8080:80 todoui ls = run command to create container by passing extra command ls

So now above command will not be executed instead of it Entrypoint command of docker file will be executed.

So entrypoint is like hardcoding of command which will always be running, no one can change it ever. Basically it cannot be overridden or replaced by passing a new command.

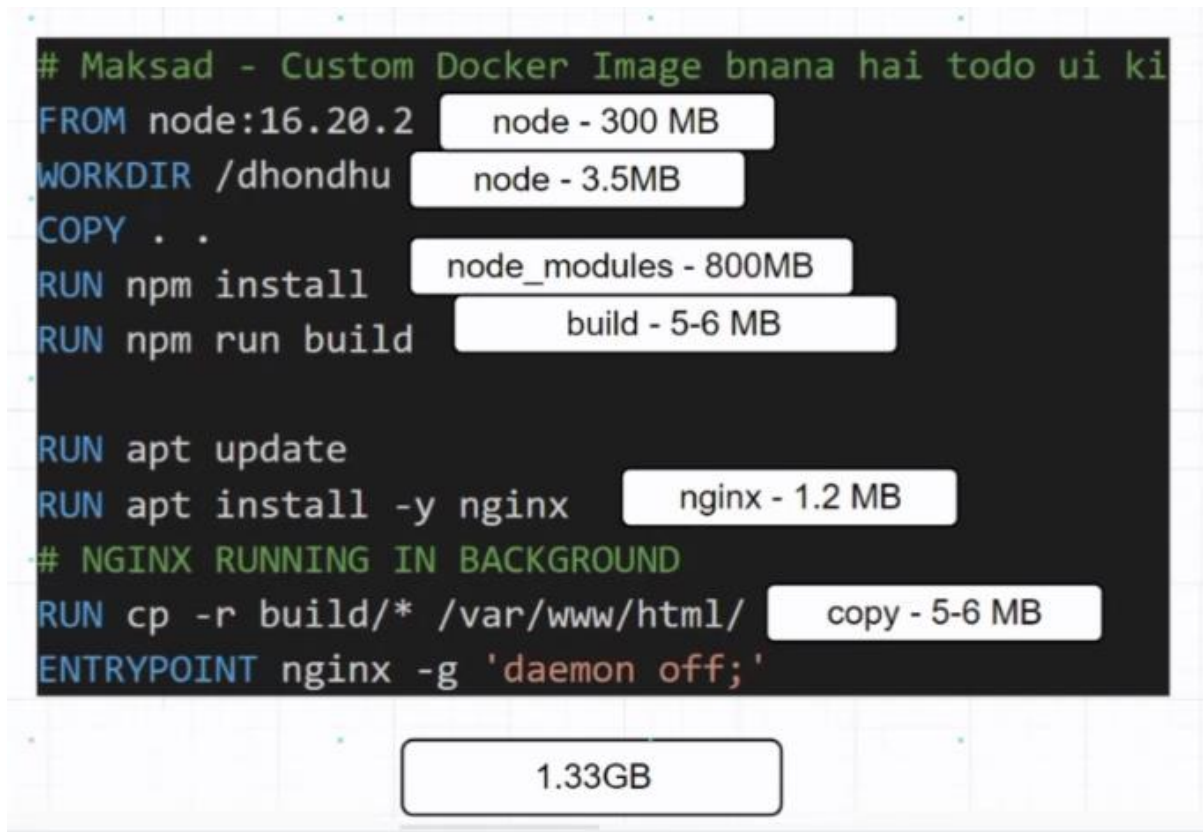
Agenda – problem in image as its of bigger size (#MAKSAD - Custom docker image banana hai todo ui ka)

1) Things required to run todo ui = build folder + nginx webserver

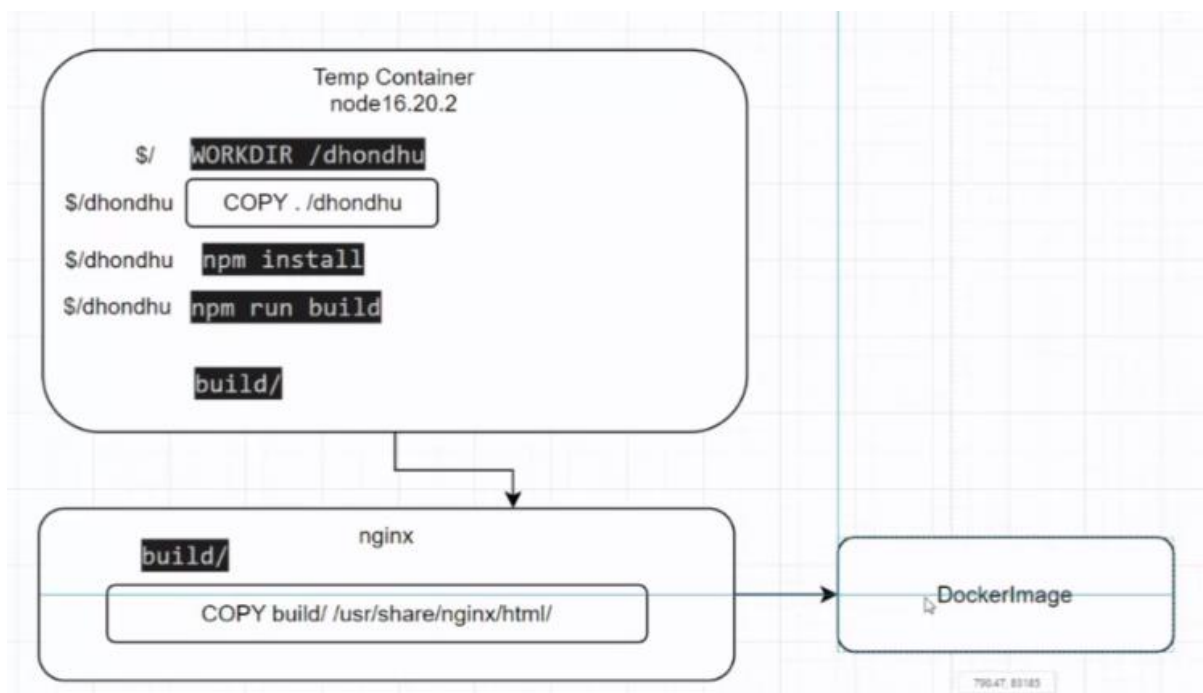


```

root@vmdocker:/home/azureuser/ReactToDoUIMonolith# docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
todoui        latest    8cfbe43bc4ca   29 seconds ago 1.34GB
<none>        <none>    fea1e545debc   2 hours ago    1.34GB
<none>        <none>    4d09465a1759   21 hours ago    1.34GB
<none>        <none>    f1ebcad628d3   2 days ago     1.34GB
root@vmdocker:/home/azureuser/ReactToDoUIMonolith# docker run -p 8080:80 todoui
  
```

2)



3) Now to give name to temporary containers

```

azureuser@vmdocker: ~/ReactTodoUIMonolith
GNU nano 7.2
#MAKSAD - Custom docker image banana hai todo ui ka
FROM node:16.20.2 AS tondumal
WORKDIR /dhondhu
COPY . .
RUN npm install
RUN npm run build

FROM nginx
COPY --from=tondumal /dhondhu/build/ /usr/share/nginx/html

```

4) **docker build -t patliimage .** = command run to build patli image

```

root@vmdocker:/home/azureuser/ReactTodoUIMonolith# docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
patliimage    latest    94df29409897   2 minutes ago  195MB
todoui        latest    8cfbe43bc4ca   2 hours ago    1.34GB
<none>        <none>    fea1e545debc   5 hours ago    1.34GB
<none>        <none>    4d09465a1759   24 hours ago   1.34GB
<none>        <none>    flebcad628d3   2 days ago     1.34GB
root@vmdocker:/home/azureuser/ReactTodoUIMonolith#

```

So now new nginx image that we have created is of 195 mb, which is created after leaving pehle ka kuuda kachra

5) We give as as many as From statements to create multiple containers but it will take of last one in the final as output.

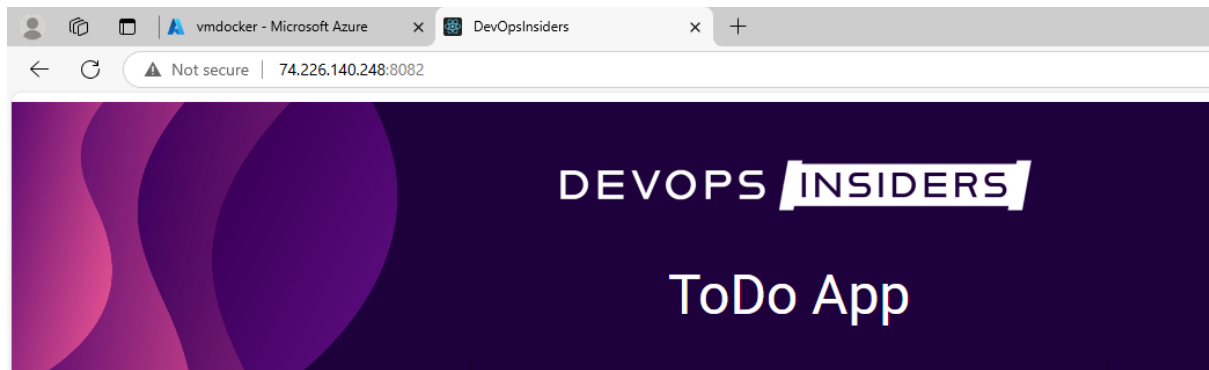
6) **docker run -p 8082:80 patliimage** = command to run and create container

```

root@vmdocker:/home/azureuser/ReactTodoUIMonolith# docker run -p 8082:80 patliimage
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf
/docker-entrypoint.sh: Sourcing /docker-entrypoint.d/15-local-resolvers.envsh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
/docker-entrypoint.sh: Configuration complete; ready for start up
2025/04/11 13:29:14 [notice] 1#1: using the "epoll" event method
2025/04/11 13:29:14 [notice] 1#1: nginx/1.27.4
2025/04/11 13:29:14 [notice] 1#1: built by gcc 12.2.0 (Debian 12.2.0-14)
2025/04/11 13:29:14 [notice] 1#1: OS: Linux 6.8.0-1021-azure
2025/04/11 13:29:14 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1024:524288
2025/04/11 13:29:14 [notice] 1#1: start worker processes
2025/04/11 13:29:14 [notice] 1#1: start worker process 30
2025/04/11 13:29:14 [notice] 1#1: start worker process 31
106.205.85.55 - - [11/Apr/2025:13:30:26 +0000] "GET / HTTP/1.1" 200 649 "-" "Mozilla/5.0 (Windows NT 10

```

74.226.140.248:8082



7) Now go to docker hub find alpine image

```
root@vmdocker: /home/azureuser/ReactToDoUIMonolith
GNU nano 7.2
#MAKSAD - Custom docker image banana hai todo ui ka
FROM node:16.20.2 AS tondumal
WORKDIR /dhondhu
COPY . .
RUN npm install
RUN npm run build

FROM nginx:alpine
COPY --from=tondumal /dhondhu/build/ /usr/share/nginx/html
```

docker build -t bahutpatliimage . = command run to build patli image

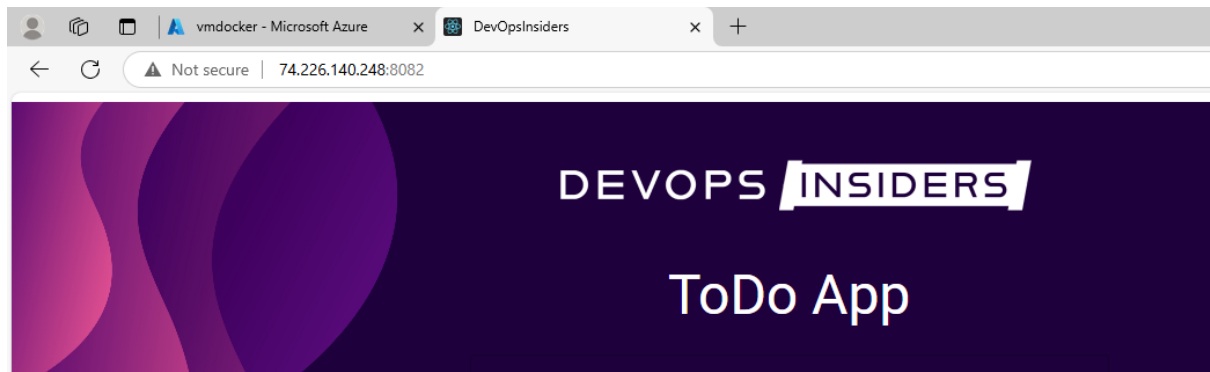
docker images

```
root@vmdocker:/home/azureuser/ReactToDoUIMonolith# docker images
REPOSITORY          TAG             IMAGE ID        CREATED         SIZE
bahutpatliimage    latest         d639e855c309   7 seconds ago  50.4MB
patliimage          latest         94df29409897   28 minutes ago 195MB
todoui             latest         8cfbe43bc4ca   3 hours ago    1.34GB
<none>             <none>         fea1e545debc   5 hours ago    1.34GB
<none>             <none>         4d09465a1759   24 hours ago   1.34GB
<none>             <none>         f1ebcad628d3   2 days ago     1.34GB
root@vmdocker:/home/azureuser/ReactToDoUIMonolith#
```

So now imag has becomemore smaller

docker run -p 8082:80 bahutpatliimage = command to run and create container

74.226.140.248:8082



+++++