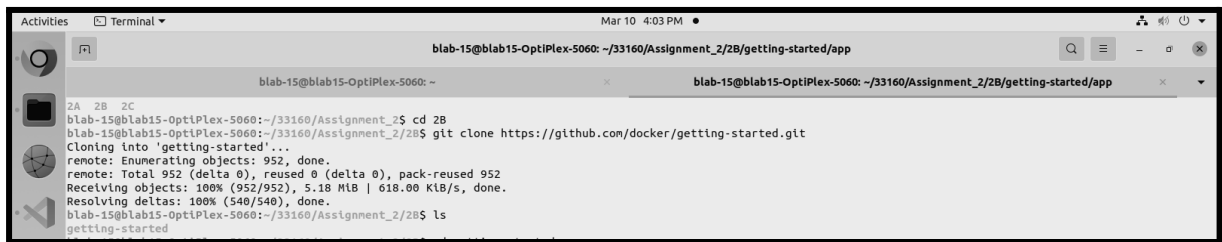


6. After installation of docker run following commands to setup first project:

Clone the project from github in order to run first project:

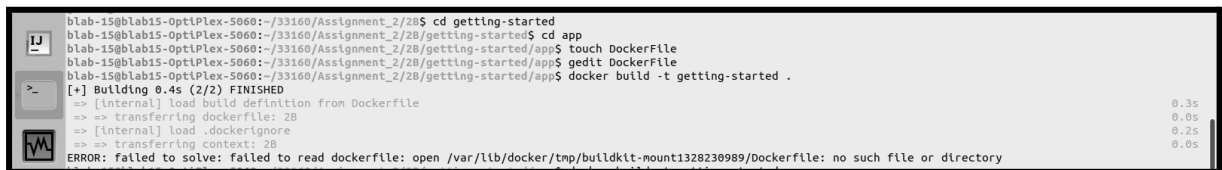
```
git clone https://github.com/docker/getting-started.git
```



```
blab-15@blab15-OptiPlex-5060: ~/33160/Assignment_2/2B/getting-started/app
blab-15@blab15-OptiPlex-5060: ~$ cd 2B
blab-15@blab15-OptiPlex-5060: ~/33160/Assignment_2/2B$ git clone https://github.com/docker/getting-started.git
Cloning into 'getting-started'...
remote: Enumerating objects: 952, done.
remote: Total 952 (delta 0), reused 0 (delta 0), pack-reused 952
Receiving objects: 100% (952/952), 5.18 MiB | 618.00 KiB/s, done.
Resolving deltas: 100% (540/540), done.
blab-15@blab15-OptiPlex-5060: ~/33160/Assignment_2/2B$ ls
getting-started
```

7. Head to the app directory in getting-started folder and create a file named Dockerfile and add the following content in it:

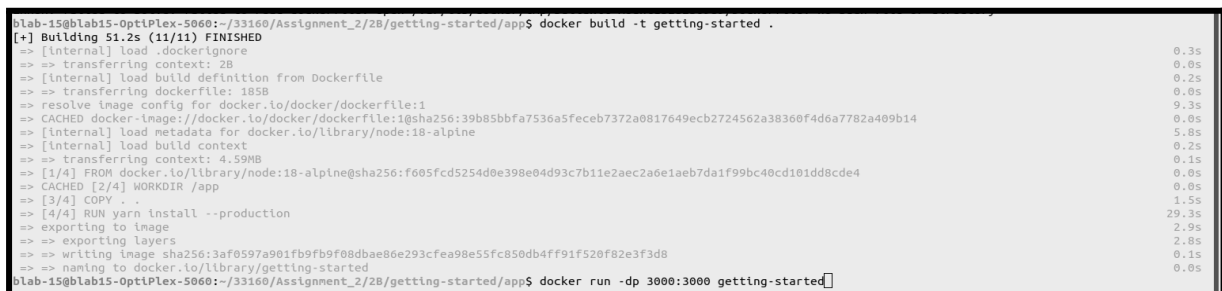
```
FROM node:18-alpine
WORKDIR /app
COPY . .
RUN yarn install --production
CMD ["node", "src/index.js"]
EXPOSE 3000
```



```
blab-15@blab15-OptiPlex-5060: ~/33160/Assignment_2/2B$ cd getting-started
blab-15@blab15-OptiPlex-5060: ~/33160/Assignment_2/2B/getting-started$ cd app
blab-15@blab15-OptiPlex-5060: ~/33160/Assignment_2/2B/getting-started/app$ touch Dockerfile
blab-15@blab15-OptiPlex-5060: ~/33160/Assignment_2/2B/getting-started/app$ gedit Dockerfile
blab-15@blab15-OptiPlex-5060: ~/33160/Assignment_2/2B/getting-started/app$ docker build -t getting-started .
[+] Building 0.4s (2/2) FINISHED
=> [internal] load build definition from Dockerfile 0.3s
=> transferring dockerfile: 2B 0.0s
=> [internal] load .dockerignore 0.2s
=> transferring context: 2B 0.0s
ERROR: failed to solve: failed to read dockerfile: open /var/lib/docker/tmp/buildkit-mount132830989/Dockerfile: no such file or directory
```

8. Build the project using 1st command and then use the 2nd command to launch the project:

1. `docker build -t getting-started .`
2. `docker run -dp 3000:3000 getting-started`



```
blab-15@blab15-OptiPlex-5060: ~/33160/Assignment_2/2B/getting-started/app$ docker build -t getting-started .
[+] Building 51.2s (11/11) FINISHED
=> [internal] load .dockerignore 0.3s
=> transferring context: 2B 0.0s
=> [internal] load build definition from Dockerfile 0.2s
=> transferring dockerfile: 185B 0.0s
=> resolve image config for docker.io/docker/dockerfile:1 9.3s
=> CACHED docker-image://docker.io/docker/dockerfile:1@sha256:39b85bbfa7536a5f6eb7372a0817649ecb2724562a38360f4d6a7782a409b14 0.0s
=> [internal] load metadata for docker.io/library/node:18-alpine 5.8s
=> [internal] load build context 0.2s
=> transferring context: 4.59MB 0.1s
=> [1/4] FROM docker.io/library/node:18-alpine@sha256:f605fcd5254d0e398e04d93c7b11e2aec2a6e1aeb7da1f99bc40cd101dd8cde4 0.0s
=> CACHED [2/4] WORKDIR /app 0.0s
=> [3/4] COPY . . 1.5s
=> [4/4] RUN yarn install --production 29.3s
=> exporting to image 2.9s
=> writing image sha256:3af0597a901fb9fb9f08dbae86e293cfea98e55fc850db4ff91f520f82e3f3d8 0.1s
=> naming to docker.io/library/getting-started 0.0s
blab-15@blab15-OptiPlex-5060: ~/33160/Assignment_2/2B/getting-started/app$ docker run -dp 3000:3000 getting-started
```

9. To know the status of the project run following command:

```
docker ps
```

```
Activities Terminal Mar 10 4:05 PM blab-15@blab15-OptiPlex-5060: ~/33160/Assignment_2/2B/getting-started/app

blab-15@blab15-OptiPlex-5060: ~
history Show the history of an image
import Import the contents from a tarball to create a filesystem image
inspect Return low-level information on Docker objects
kill Kill one or more running containers
load Load an image from a tar archive or STDIN
logs Fetch the logs of a container
pause Pause all processes within one or more containers
port List port mappings or a specific mapping for the container
rename Rename a container
restart Restart one or more containers
rm Remove one or more containers
rmi Remove one or more images
save Save one or more images to a tar archive (streamed to STDOUT by default)
start Start one or more stopped containers
stats Display a live stream of container(s) resource usage statistics
stop Stop one or more running containers
tag Create a tag TARGET_IMAGE that refers to SOURCE_IMAGE
top Display the running processes of a container
unpause Unpause all processes within one or more containers
update Update configuration of one or more containers
wait Block until one or more containers stop, then print their exit codes

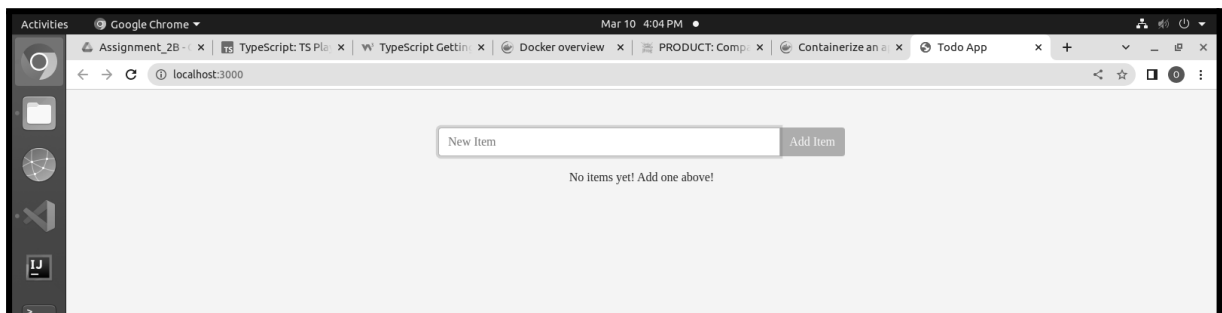
Global Options:
-c, --config string      Location of client config files (default "/home/blab-15/.docker")
-D, --debug              Enable debug mode
-H, --host list          Daemon socket(s) to connect to
-l, --log-level string   Set the logging level ("debug", "info", "warn", "error", "fatal") (default "info")
--tls                   Use TLS; implied by --tlsverify
--tlscacert string      Trust certs signed only by this CA (default "/home/blab-15/.docker/ca.pem")
--tlscert string        Path to TLS certificate file (default "/home/blab-15/.docker/cert.pem")
--tlskey string         Path to TLS key file (default "/home/blab-15/.docker/key.pem")
--tlsverify             Use TLS and verify the remote
-v, --version            Print version information and quit

Run 'docker COMMAND --help' for more information on a command.

For more help on how to use Docker, head to https://docs.docker.com/go/guides/

blab-15@blab15-OptiPlex-5060:~/33160/Assignment_2/2B/getting-started/app$ docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS                    NAMES
46fe2583e9cf   getting-started   "docker-entrypoint.s..."   About a minute ago   Up About a minute   0.0.0.0:3000->3000/tcp, :::3000->3000/tcp   busy_thompson
```

10. The launched project will look like this:



11. After adding data it will look like this:

