Exercise 3: Building new images

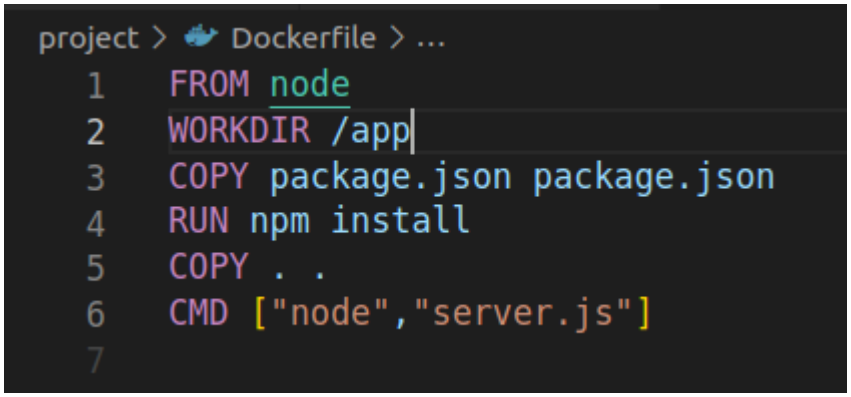
In this exercise we will do how to create a new Docker image. In this exercise i have already a simple react project in my system so let's build the image for this project.

Creating a Docker-file.
for building a image we must have to make a dockerfile and run that dockerfile to build a image. The Dockerfile is a "recipe" of sorts, that contains a list of instructions on how to build a new image. let's create a dockerfile.

$ touch Dockerfile

inside the dockerfile we have to provide this.

```
FROM node                 # Use the official Node.js image as the base image
WORKDIR /app/src          # Set the working directory to /app/src
COPY package.json .       # Copy package.json from the current directory to /app/src
COPY package-lock.json .  # Copy package-lock.json from the current directory to /app/src
RUN npm install           # Install dependencies defined in package.json
COPY . .                  # Copy the current directory (all files) to /app/src
EXPOSE 3000               # Expose port 3000 to allow connections to the container
CMD ["npm", "start"]      # start the application with npm
```



finally, we create our dockerfile.

Building a dockerfile.

to build docker images from dockerfile we use **docker build** command. The docker build command reads a Dockerfile, and runs its instructions to create a new image.

$ docker build -t dabhidhruvraj/react .

here dabhidhruvraj/react is our image name and . pointed that we want docker image in this directory. running docker images we can see our new image.

```
dhruvrajsinh@sf-cpu-371:~/Node.js-Expess-MongoDB-CRUD$ docker build -t dabhidhruvraj/reactapp .
[+] Building 0.0s (2/2) FINISHED
 => [internal] load build definition from Dockerfile
 => => transferring dockerfile: 2B
 => [internal] load .dockerignore
 => => transferring context: 2B
```

Running a container:

```
Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them
dhruvrajsinh@sf-cpu-371:~/Node.js-Expess-MongoDB-CRUD/project$ docker run -d -p 3001:3000 --name reactcontainer dabhidhruvraj/reactapp
7e4d36b00ab5aca60f3130ce52d6513a67311bb04bb0e4508d2c9746b0923922
dhruvrajsinh@sf-cpu-371:~/Node.js-Expess-MongoDB-CRUD/project$
```

Let's run this app in our browser at localhost:3001