**Prerequisites:**

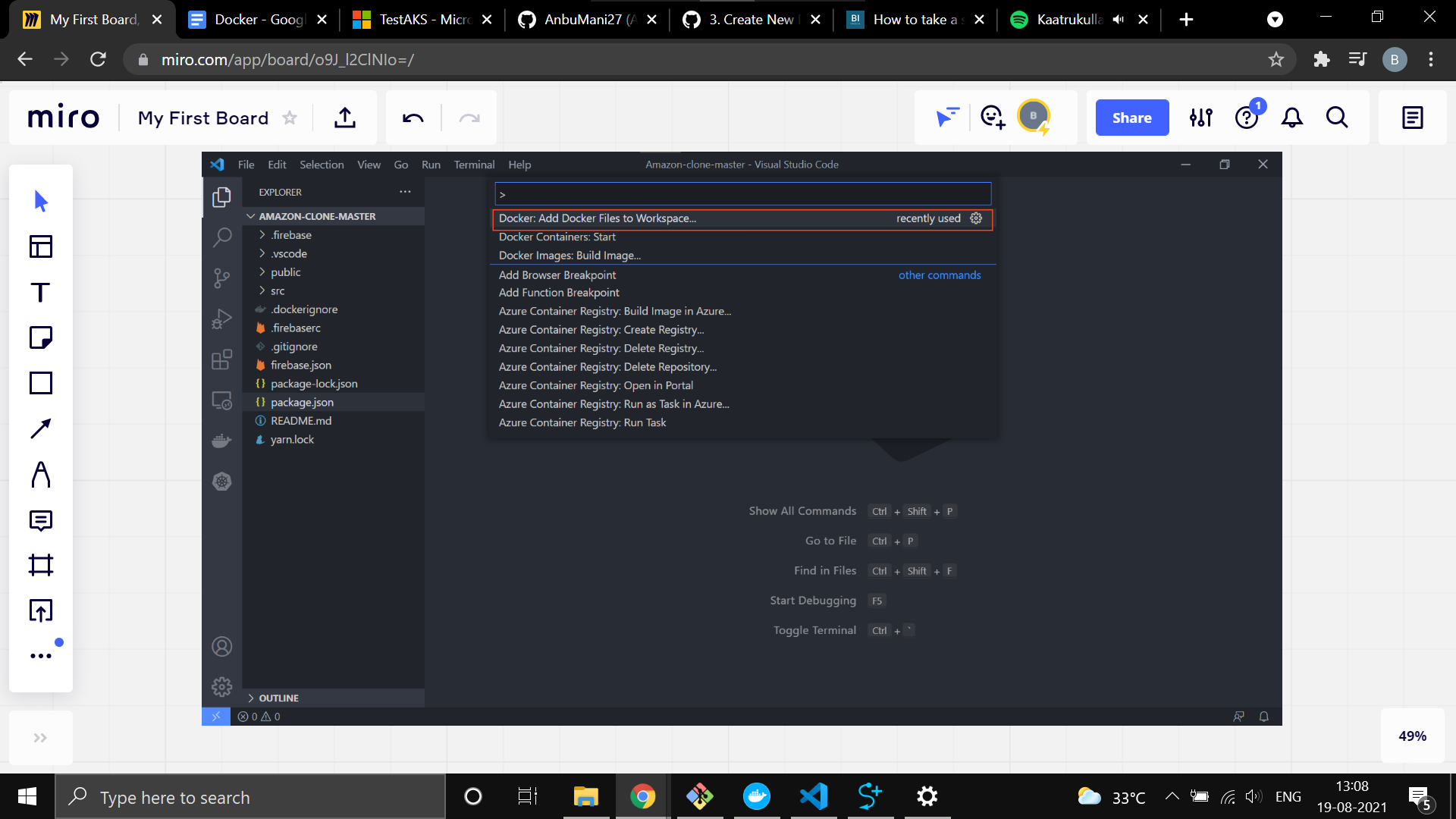
1. Docker installed on your machine
2. Docker extension installed on your vs code

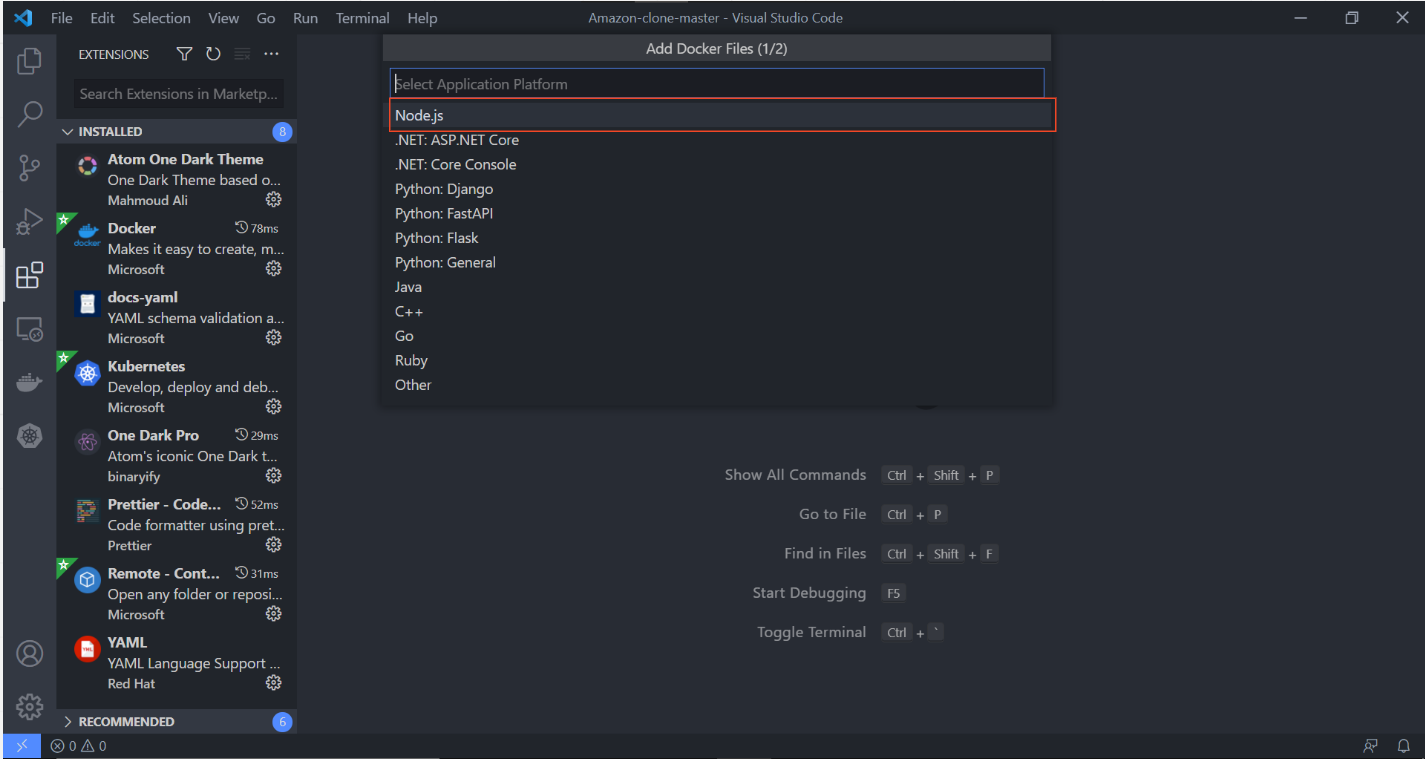
**To know more about Docker:** [**https://docker-curriculum.com/**](https://docker-curriculum.com/)

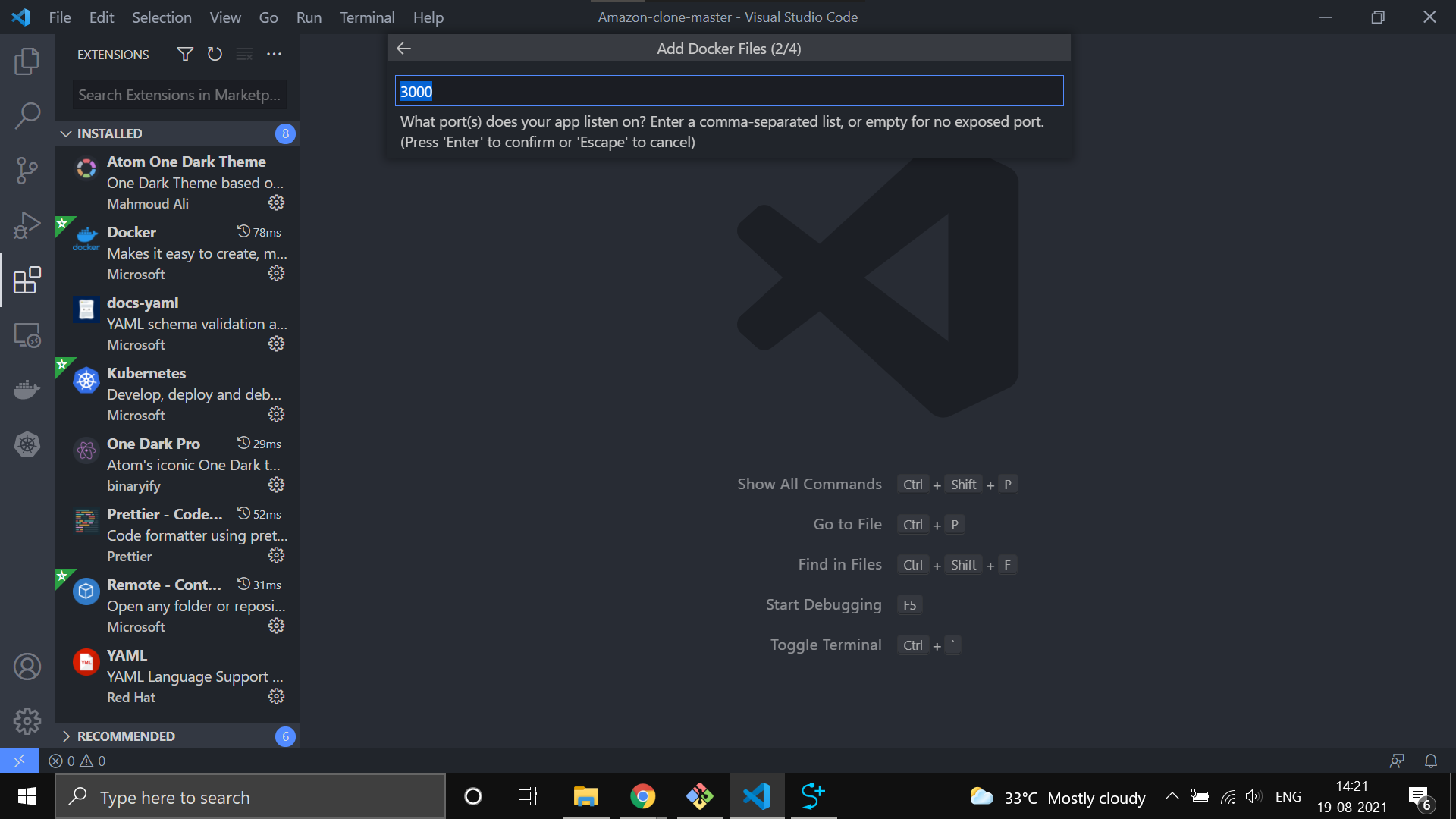
**React (Dockerfile):**

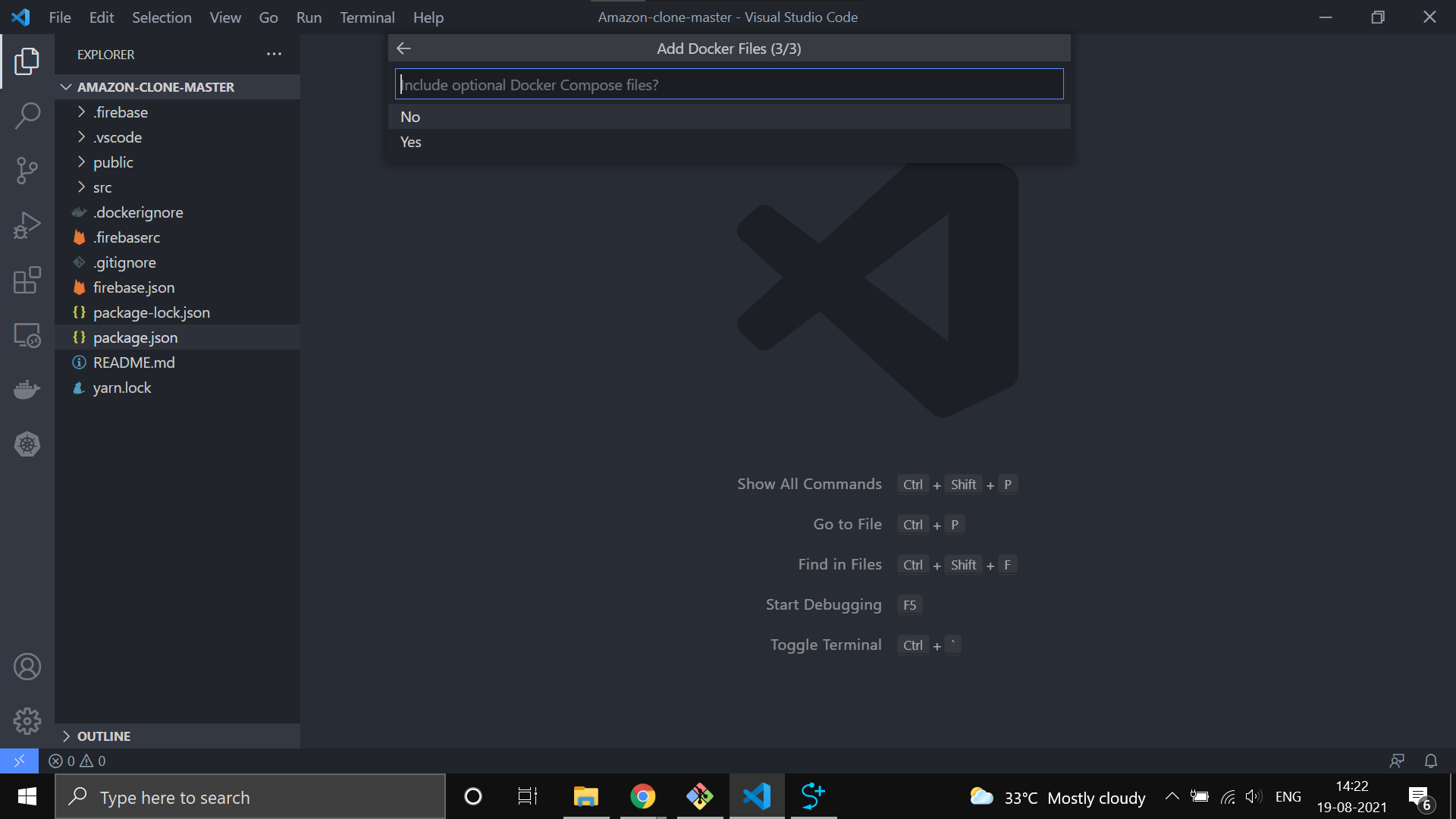
1.Create a docker file for your project using vs code

* Press ctrl + shift + p
* Search add ‘docker files to workspace’ and click that
* Select ‘Application option’ that suits your project
* Provide port in which the project listens
* Select compose file if you need for your project









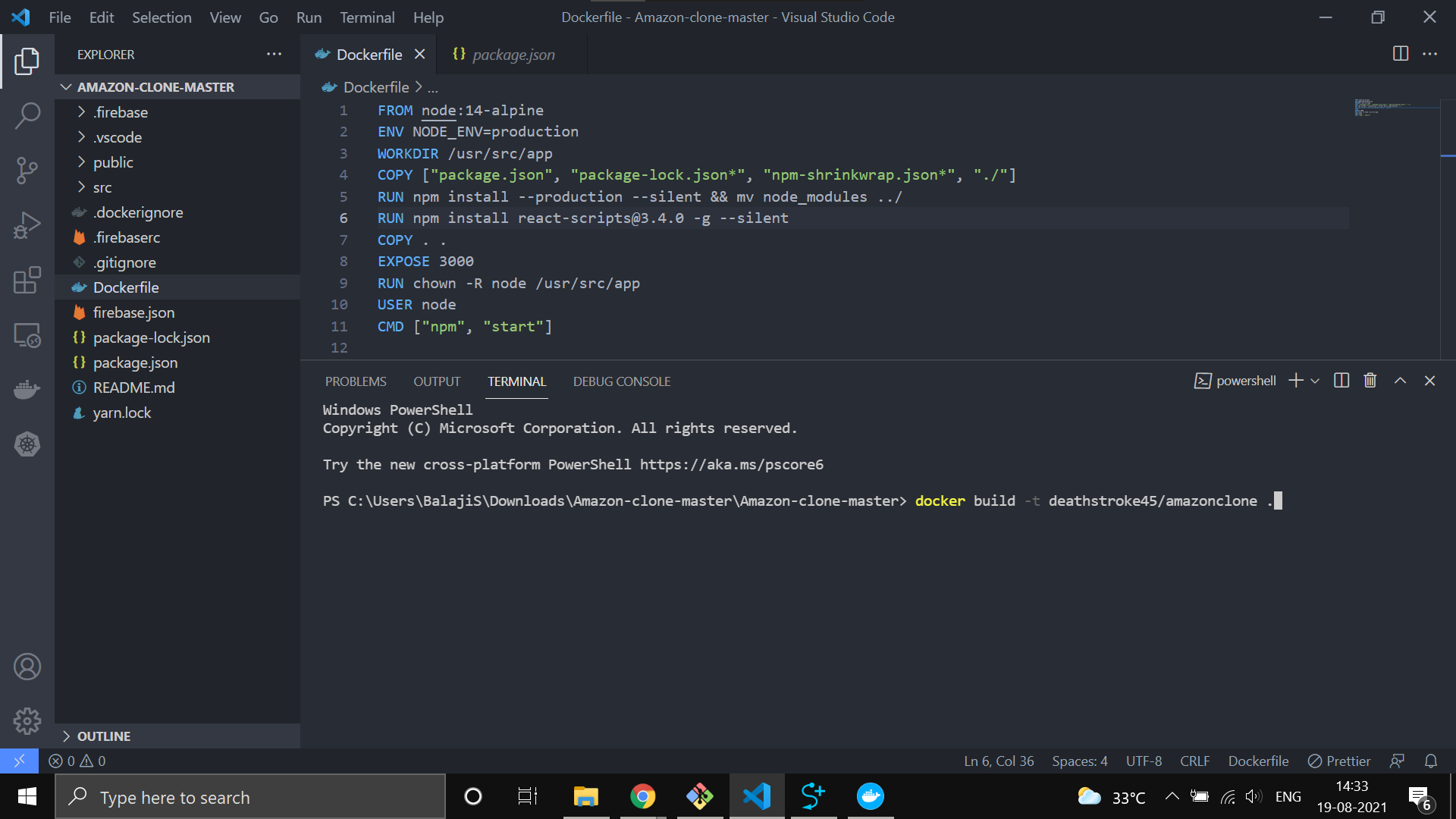
2.After docker file generated use this command to build the image

**docker build <image name> .**

(dot is important because it refers the docker file in the current folder)

-t is flag variable for tag

With out -t docker will put some random names to the images



I had two issue with react scripts

1. React scripts are not found even though it is available in the node modules .

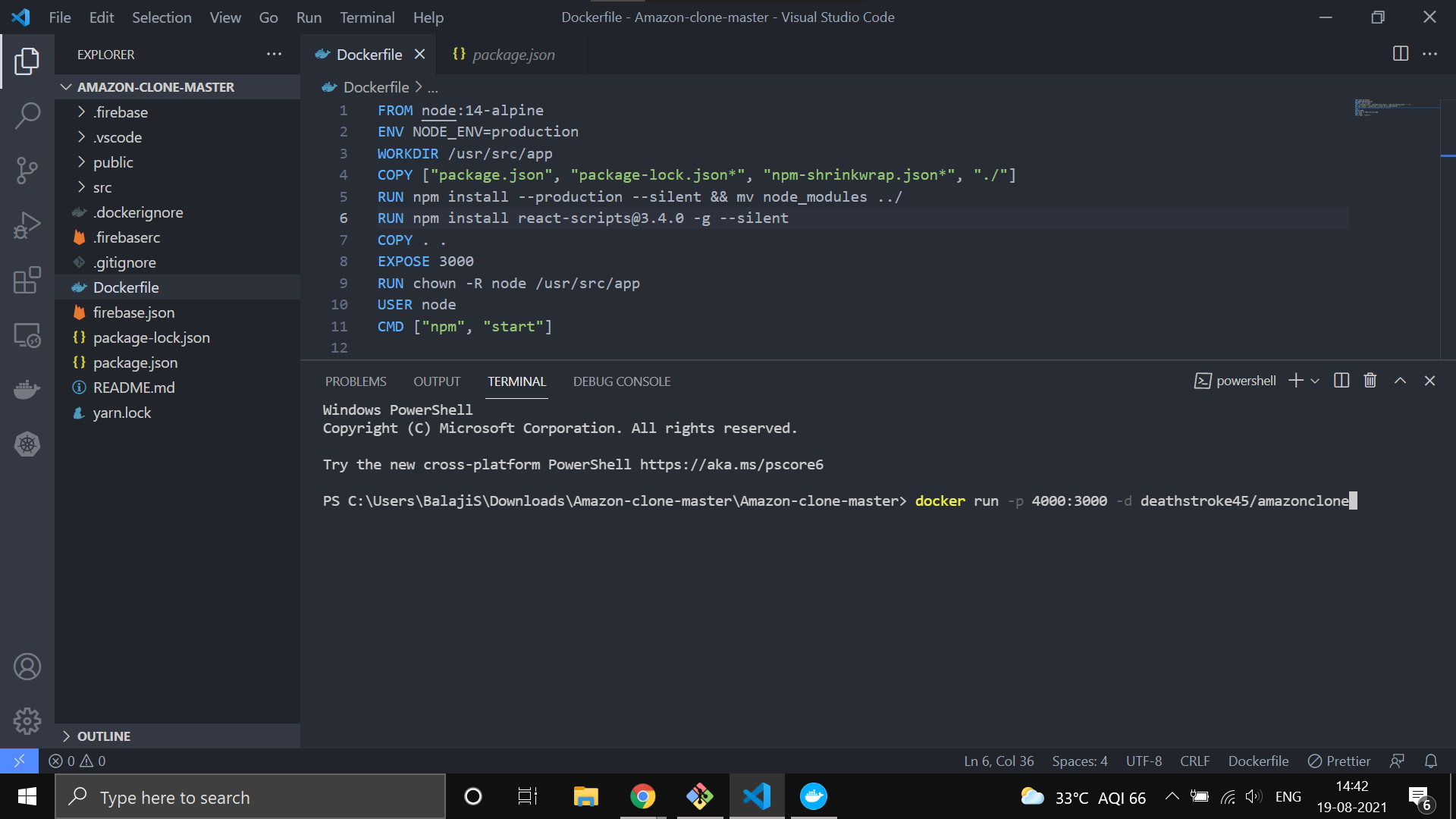
**Solution:** Install react scripts separately

1. I installed react scripts version 3.4.1 in that version container exited after it started .

**Solution**:Move to react scripts v3.4.0

Run the image using

**docker run -d -p <host port>:<expose port> <image name>**



Check whether the container is running are not by going to **localhost:<host port>**

**Angular (**Dockerfile)

FROM node:12.7-alpine AS build

WORKDIR /usr/src/app

COPY . .

ARG build\_mode=production\_build

EXPOSE 8080

RUN npm cache clean --force \

&& npm i \

&& npm install --global http-server \

&& npm run ${build\_mode} \

&& find \* -maxdepth 0 -name 'dist' -prune -o -exec rm -rf '{}' ';'

CMD http-server /usr/src/app/dist --proxy http://localhost:8080?

# commands to run

# docker build -t <tagname> --build-arg build\_mode=stage\_build .

# docker run -d -p 8080:8080 <tagname>

For angular there is an issue with generated docker file by the vs code.So this template is useful.

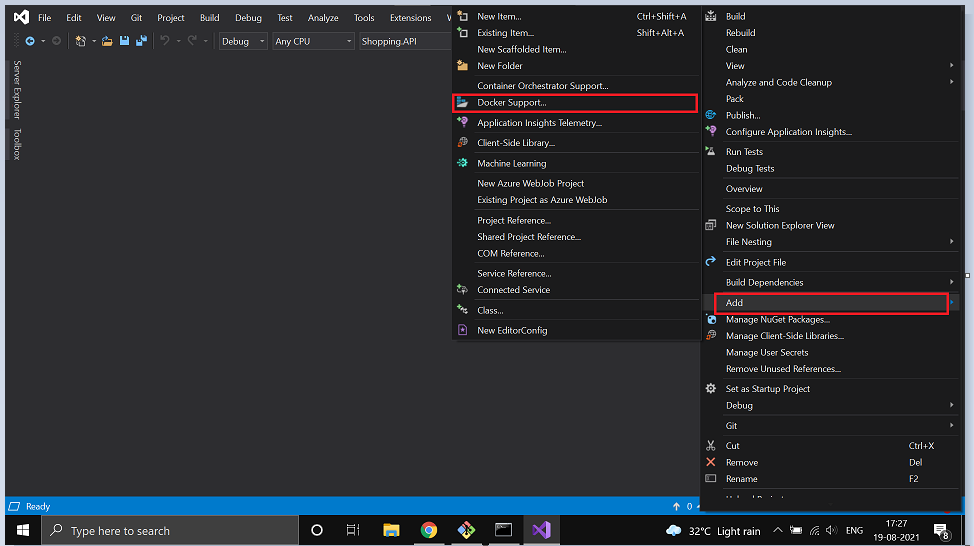
docker build -t <image name> . (dot is important)

docker run -d -p <host port>:<expose port> <image name>

.**Netcore(Dockerfile):**

1. Create a docker file for your project using Visual Studio

* Right click on API project and navigate to add and then docker support
* Select what kind of os need for your project (windows or linux)



docker build -t <image name> -f “<path to the docker file>”

docker run -d -p <host port>:<expose port> <image name>

-d is for detach mode (your terminal will get closed after api get started)

**Note:**

Run the docker build where all the file is scoped to the Dockerfile.

**Python (Dockerfile);**

FROM python:3.8-slim-buster

WORKDIR /app

COPY requirements.txt requirements.txt

RUN pip3 install -r requirements.txt

COPY . .

CMD [ "python3", "-m" , "flask", "run", "--host=0.0.0.0"]

**Docker Compose file: (Template)**

version: "3.2"

services:

testDb:

container\_name: testDb

image: mcr.microsoft.com/mssql/server:2019-latest

ports:

- "1433:1433"

environment:

SA\_PASSWORD: "balaji55!@"

ACCEPT\_EULA: "Y"

shop:

container\_name: testDb

build:

context: .

dockerfile: ShoppingAPI/Shopping.API/Dockerfile

image: deathstroke45/shoppingapi:latest

depends\_on: testDb

ports:

- "8080:80"

To run this **docker compose up**