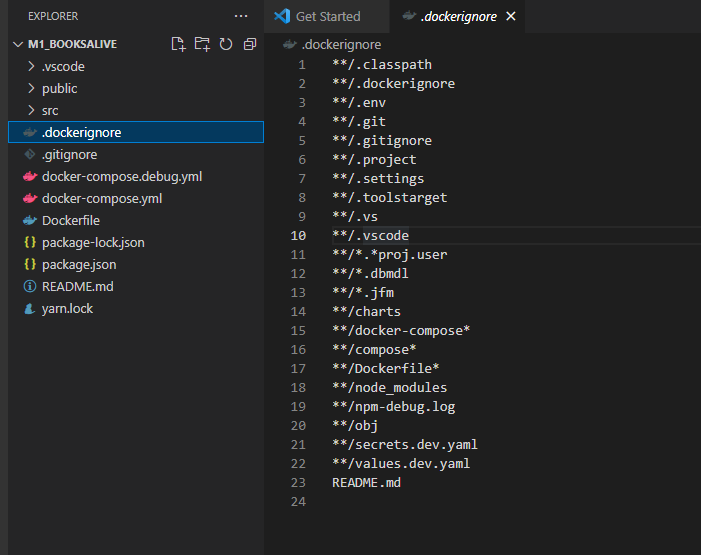
**Instructions:**   
1. Install docker on your PC.   
2. All the files are added to the attached project just run the 4th step command.  
3. Details about what to add in [docker-compose, dockerfile] files is in the end.  
  
How to Dockerize a React Application.

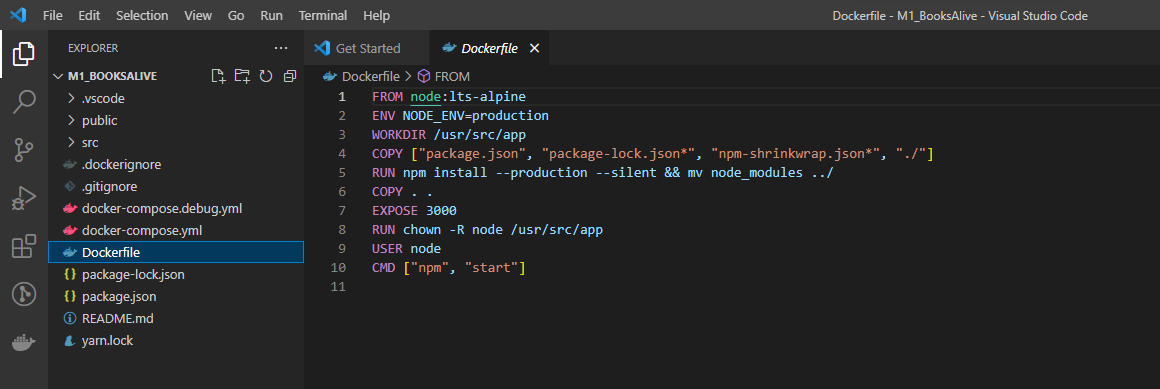
### **Add a .dockerignore File**

Just like the .gitignore file, we need a .dockerignore file as well so that docker knows what to ignore in the build process. There are two ways of creating this file. You can create it manually. Or you can use the touch .dockerignore command in your terminal. You should get your .dockerignore file like this.



### **2. Add a Dockerfile**

To give the build instructions, we need to create a Dockerfile that contains those instructions. Here you can create the file in two ways: manually or using the touch Dockerfile command. Note that, the file name should start with capital D.

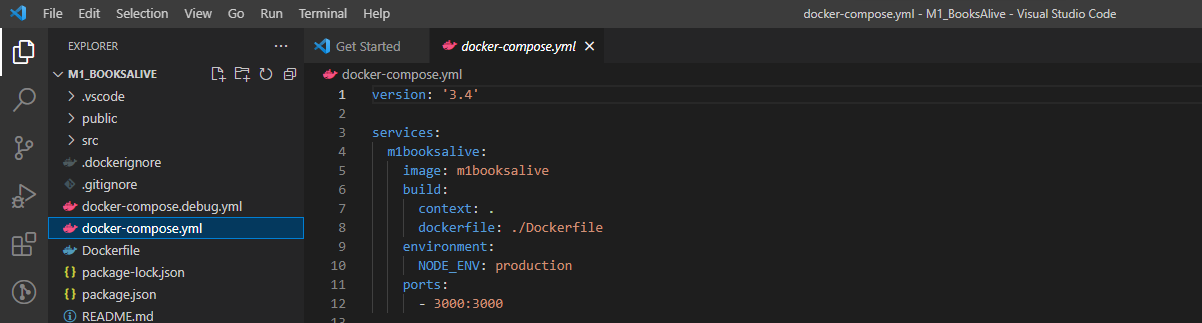


### **3. Creating the Docker Image**

It’s time to create the docker image. For that, we need a docker-compose file. Same as above files. You can either create the file manually. Or you can use the below command for that.

**touch docker-compose.yml**

Let’s create the docker-compose.yml file.



### **4. Running the Docker Image**

Next, to run the docker image, you can use the below command

**docker compose up**

If you are not in the root folder, specify the docker compose file name. Use the below command in a situation like that.

**docker-compose -f docker-compose.yml up**

Go ahead and execute one of these commands in your terminal. You should see the building process. Once the building process is completed, you should be able to navigate to [http://localhost:3000/](http://localhost:8080/)

-------------------------------------------------------------------------------------------------------------  
  
**docker-compose.yml:**

* Here we have to define a version in the root of the file. This is the docker-compose version currently being used. According to the docker documentation, it’s 3.8.
* Then we need to configure our application inside the compose file, as a service.
* **Note** that we need to keep the indentations correctly when defining properties in this file.
* The first property is the container name. This will create the container with the given name if the container does not exist.
* Then, the image name. This will create the image with the given name if the image does not exist.
* For the ports, I have defined 8080 as the hosting port, and 80 as the container port

**DockerFile:**

* Although react is not a node application, it needs to use node to build the application. So as the first step, we must import node.
* The next step is to define a working directory. Usually, we use /app for that.
* As the next step, we need to copy the package.json and the .lock file. If you are using npm, this will be package-lock.json. If you are using yarn this will be yarn.lock**.** Otherwise, we won’t be able to use yarn/npm commands in the next layer. These files should be copied to the working directory which we defined above. For that, we add a **.** after the copy command. But copying the yarn.lock file is not required.