

# Create Demo Project-React-Webapp Using Docker

## Step 1: Install Prerequisites

Make sure you have these installed:

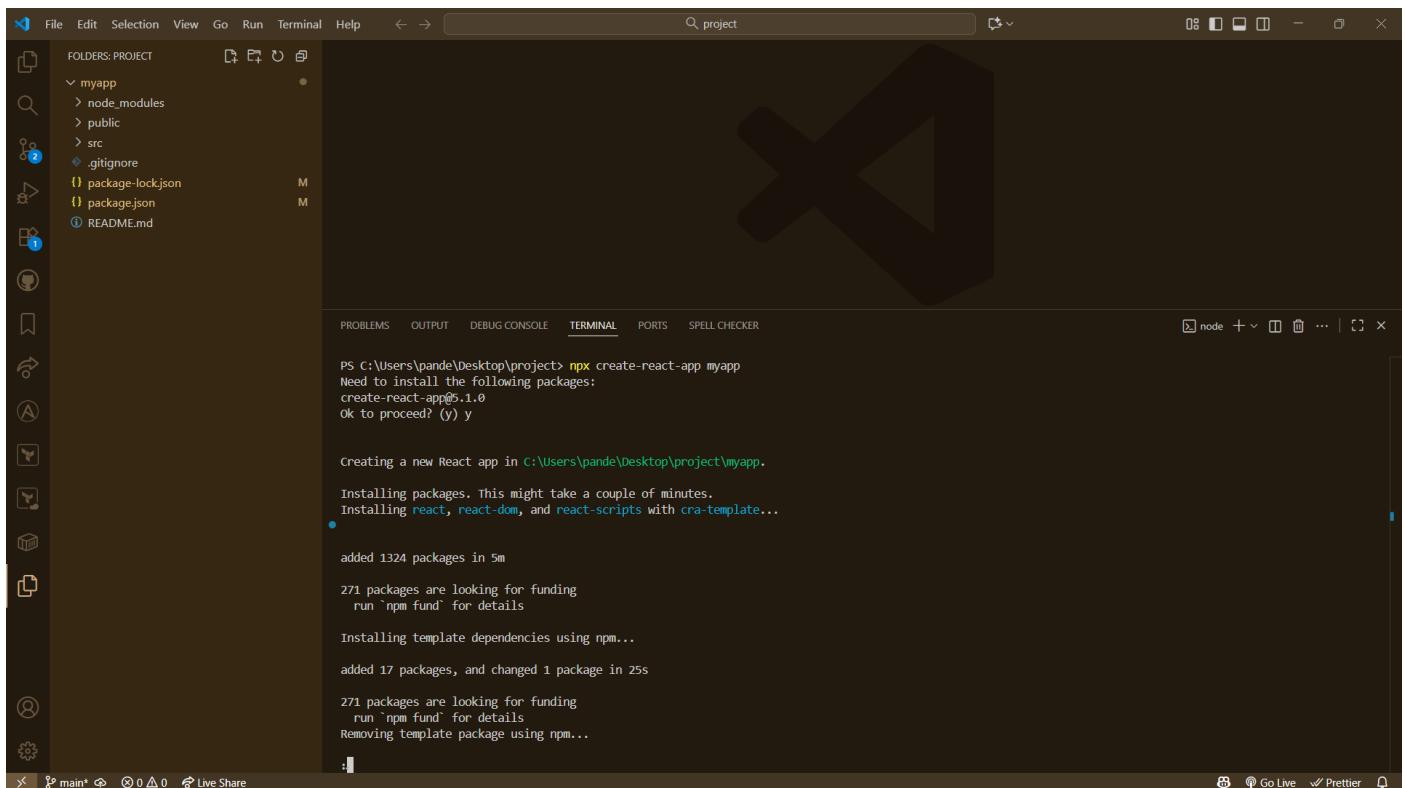
- Node.js (v16 or higher) → <https://nodejs.org>
- npm (comes with Node) or yarn (optional)
- Code Editor → VS Code recommended

Check versions:

```
bash
node -v
npm -v
```

## Step 2: Create a New React App

Using npx: -



The screenshot shows a Windows desktop environment with the Visual Studio Code application open. The code editor window displays a terminal session. The terminal output is as follows:

```
PS C:\Users\pande\Desktop\project> npx create-react-app myapp
Need to install the following packages:
  create-react-app@5.1.0
Ok to proceed? (y) y

Creating a new React app in C:\Users\pande\Desktop\project\myapp.

Installing packages. This might take a couple of minutes.
Installing react, react-dom, and react-scripts with cra-template...
added 1324 packages in 5m
271 packages are looking for funding
  run `npm fund` for details
Installing template dependencies using npm...
added 17 packages, and changed 1 package in 25s
271 packages are looking for funding
  run `npm fund` for details
Removing template package using npm...
```

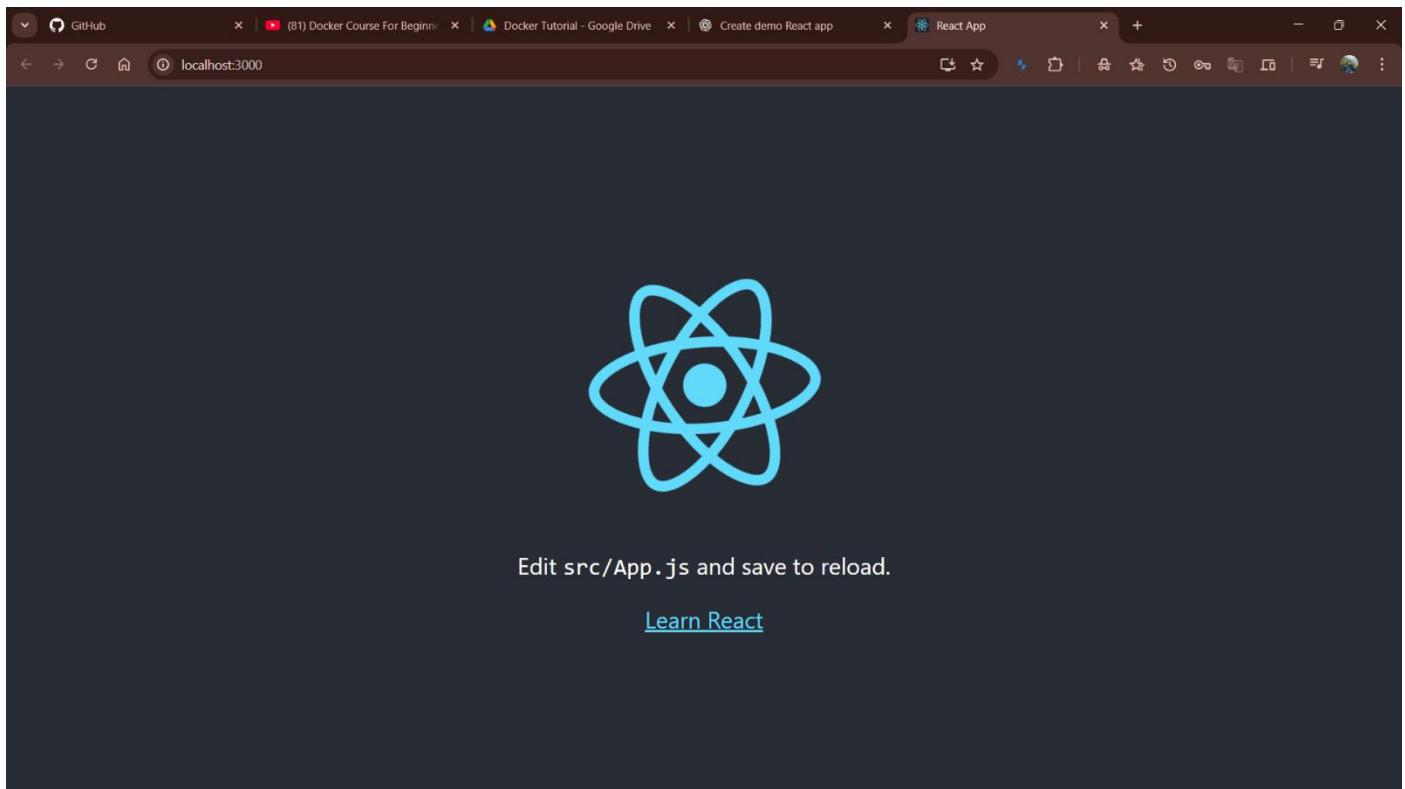
**Then:**

We suggest that you begin by typing:

```
cd myapp
npm start
```

The screenshot shows the Visual Studio Code interface. On the left is the Explorer sidebar with a project structure: FOLDERS: PROJECT, myapp (containing node\_modules, public, src, .gitignore, package-lock.json, package.json, and README.md). The main area is a dark-themed terminal window. At the top of the terminal, it says "We suggest that you begin by typing:" followed by the command "cd myapp\nnpm start". Below this, the terminal shows the command being run in a PowerShell window: "PS C:\Users\pande\Desktop\project> cd myapp" and "PS C:\Users\pande\Desktop\project\myapp> npm start". The output of the command shows several deprecation warnings from Node.js, such as "DeprecationWarning: 'onAfterSetupMiddleware' option is deprecated. Please use the 'setupMiddlewares' option.", and ends with "starting the development server...". The bottom right corner of the terminal shows a status bar with "powershell - myapp" and other icons.

**React Webpage: -**



## How to create Docker file?

### 1 What is a Dockerfile?

A Dockerfile is a script with instructions to build a Docker image — like a recipe for packaging your app (code + dependencies + runtime) into a container.

### 2 Basic Dockerfile for a React App

Create a file named Dockerfile in your project's root (same folder as package.json):

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer (Left):** Shows the project structure: FOLDERS: PROJECT, myapp (expanded), public, src, .gitignore, Dockerfile (selected), package-lock.json, package.json, and README.md.
- Editor (Top Right):** Displays the contents of the Dockerfile. The file starts with a Node.js image, sets the working directory to /app, copies the app files, exposes port 80, and runs npm install. It then starts the server with the command "npm start".

```
1 # Step 1: Use Node image to build the app
2 FROM node:20
3
4 # Set working directory
5 WORKDIR /app
6
7 # Copy the rest of the app
8 COPY . .
9
10 # Build the app for production
11 RUN npm install
12
13 # Expose port 80
14 EXPOSE 3000
15
16 # Start server
17 CMD [ "npm", "start" ]
```
- Terminal (Bottom):** Shows the terminal prompt PS C:\Users\pande\Desktop\project\myapp>.
- Bottom Status Bar:** Shows the current file is main\*, with 0 changes, and other status indicators like Live Share, Go Live, and Prettier.

## **How to Build Docker Image: -**

**Use Command: -**

## Docker Build .

The screenshot shows a VS Code interface with the following details:

- File Explorer:** Shows a project structure with folders `myapp`, `public`, `src`, and files `.gitignore`, `Dockerfile`, `package-lock.json`, `package.json`, and `README.md`.
- Editor:** Displays a `Dockerfile` content:

```
1 # Step 1: Use Node image to build the app
2 FROM node:20
3
4 # Set working directory
5 WORKDIR /app
6
7 # Copy the rest of the app
8 COPY . .
9
10 # Build the app for production
11 RUN npm install
12
13 # Expose port 80
14 EXPOSE 3000
15
16 # Start server
17 CMD [ "npm", "start" ]
```
- Terminal:** Shows the output of a `docker build` command:

```
PS C:\Users\pande\Desktop\project\myapp> docker build .
[+] Building 9.5s (8/9)
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 300B
=> [internal] load metadata for docker.io/library/node:20
=> [auth] library/node:pull token for registry-1.docker.io
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [1/4] FROM docker.io/library/node:20@sha256:47acd49500971c0fbfe02323b2d04f6df40a933b123889636fc1f76bf69f58a
=> => resolve docker.io/library/node:20@sha256:47acd49500971c0fbfe02323b2d04f6df40a933b123889636fc1f76bf69f58a
● => [internal] load build context
=> => transferring context: 699.93kB
=> CACHED [2/4] WORKDIR /app
=> [3/4] COPY . .
=> [4/4] RUN npm install
```

The terminal also shows a tab for `docker - myapp` and a status bar indicating the file is `Dockerfile`.



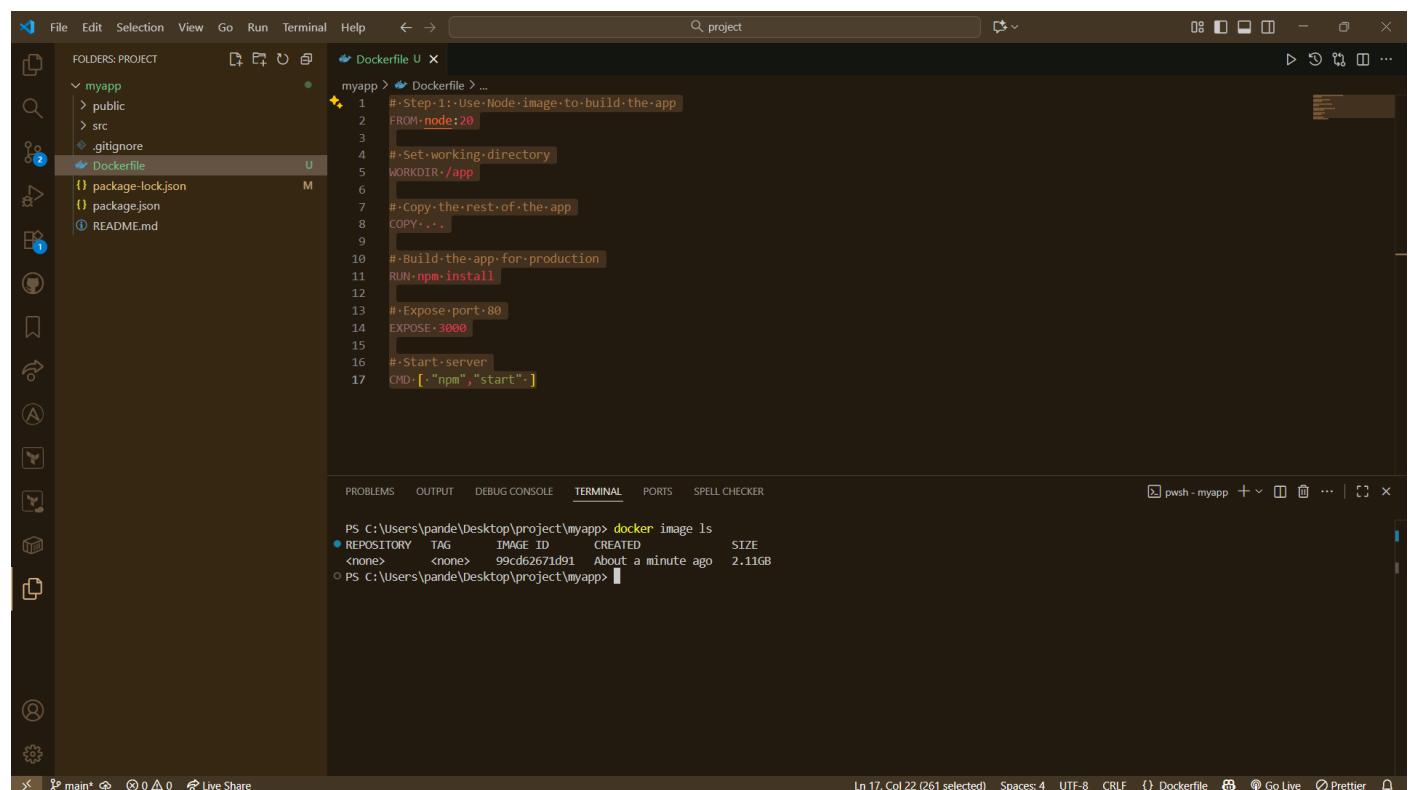
## Step Command

## Description

- |   |                                       |                        |
|---|---------------------------------------|------------------------|
| 1 | docker build -t my-react-app .        | Build image            |
| 2 | docker images                         | List images            |
| 3 | docker run -d -p 8080:80 my-react-app | Run container          |
| 4 | docker ps                             | See running containers |

---

Docker images: -



The screenshot shows a VS Code interface with a dark theme. On the left is a sidebar with icons for file operations like Open, Save, and Find. The main area shows a project structure for 'myapp' containing 'public', 'src', '.gitignore', and a 'Dockerfile'. The 'Dockerfile' is open in the center editor, displaying the following code:

```
myapp > Dockerfile > ...
1 #< Step 1: Use Node image to build the app
2 FROM node:20
3
4 #< Set working directory
5 WORKDIR /app
6
7 #< Copy the rest of the app
8 COPY . .
9
10 #< Build the app for production
11 RUN npm install
12
13 #< Expose port 80
14 EXPOSE 3000
15
16 #< Start server
17 CMD [ "npm", "start" ]
```

Below the editor is a terminal window titled 'pwsh - myapp' showing the command 'docker image ls' and its output:

```
PS C:\Users\pande\Desktop\project\myapp> docker image ls
REPOSITORY TAG IMAGE ID CREATED SIZE
<none> <none> 99cd62671d91 About a minute ago 2.11GB
PS C:\Users\pande\Desktop\project\myapp>
```

At the bottom of the interface, there are status bars for 'Live Share' and 'Prettier'.

---

## Manage And Running Container: -

### Use Command: -

Docker run -image id

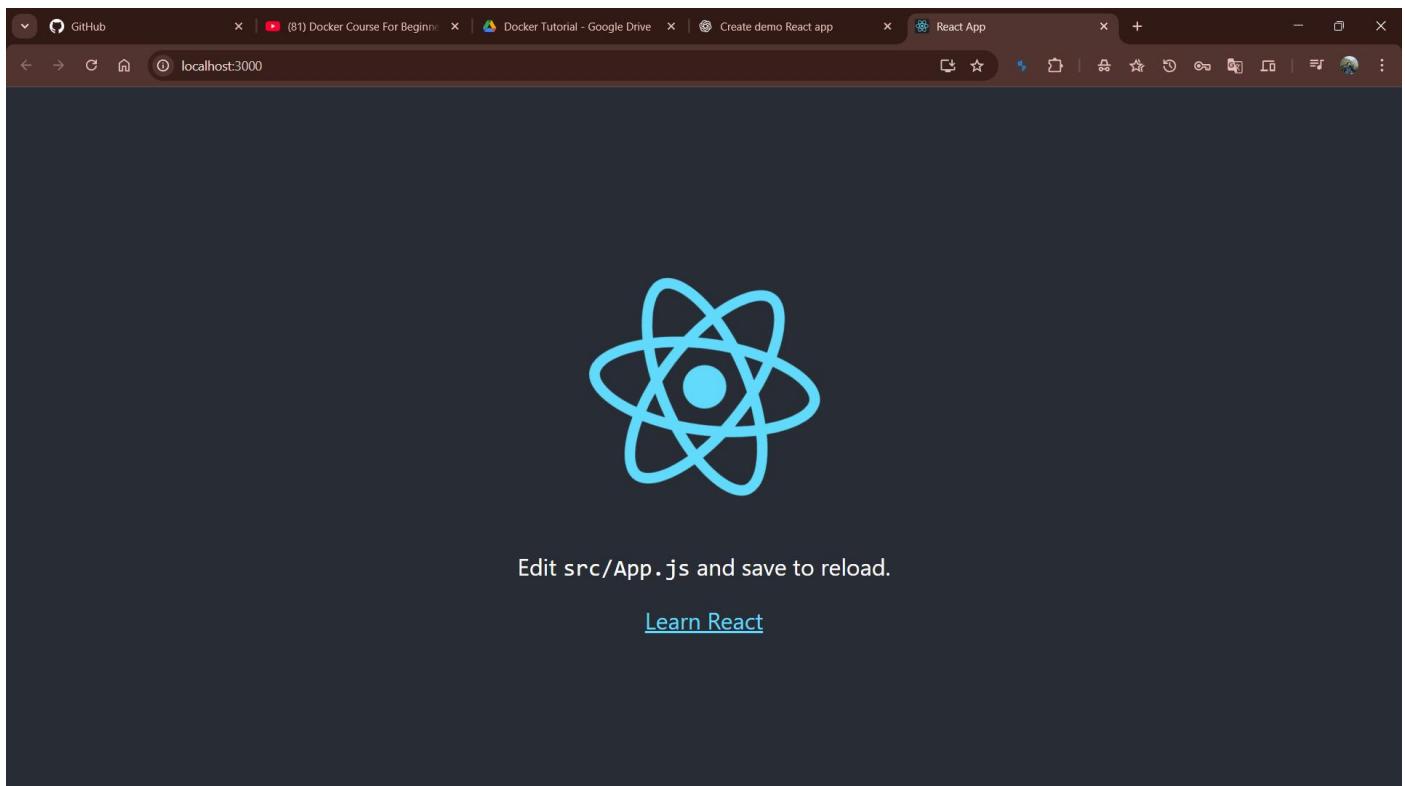
Comm.: - **docker run -p 3000:3000 99cd62671d91**

The screenshot shows the Visual Studio Code interface. On the left is the Explorer sidebar with a 'PROJECT' folder containing 'myapp' (with 'public', 'src', '.gitignore', and 'Dockerfile'), 'package-lock.json', 'package.json', and 'README.md'. The 'Dockerfile' is selected. The main area shows the contents of the Dockerfile:

```
myapp > Dockerfile > ...
1 # Step 1: Use Node image to build the app
2 FROM node:20
3
4 # Set working directory
5 WORKDIR /app
6
7 # Copy the rest of the app
8 COPY . .
9
10 # Build the app for production
```

Below the code editor is a terminal window titled 'docker - myapp' showing the command: `PS C:\Users\pande\Desktop\project\myapp> docker run -p 3000:3000 99cd62671d91`. The terminal output shows the application starting and the React development server running. It also indicates that webpack has compiled successfully.

## Now See Webpage Is Running on the Docker Container



## View Of Docker Desktop:

## Running Container: -

The screenshot shows the Docker Desktop interface with the 'Containers' tab selected. On the left sidebar, 'Containers' is highlighted. At the top, there's a search bar and a status bar showing 'Engine running'. The main area displays two running containers:

Name	Container ID	Image	Port(s)	CPU (%)	Last started	Actions
awesome_davinci	dfad29891147	99cd62671d91		0%	6 minutes ago	[Actions]
vigilant_neumann	8bdb43182ddc	99cd62671d91	3000:3000	0.01%	3 minutes ago	[Actions]

At the bottom, it says 'Showing 2 items'.

## Image: -

The screenshot shows the Docker Desktop interface with the 'Images' tab selected. On the left sidebar, 'Images' is highlighted. At the top, there's a search bar and a status bar showing 'Engine running'. The main area displays one image:

Name	Tag	Image ID	Created	Size	Actions
<none>	<none>	99cd62671d91	12 minutes ago	2.1 GB	[Actions]

At the bottom, it says 'Showing 1 item'.