

EXP 5.1- Dockerize a React Application with Multi-Stage Build

```
import React from 'react';
```

```
function App() {
```

```
  return (
```

```
    <div style={{ display: 'flex', justifyContent: 'center',  
alignItems: 'center', height: '100vh', backgroundColor:  
'#282c34' }}>
```

```
      <header style={{ textAlign: 'center' }}>
```

```
        
```

```
      <p style={{ color: 'white', fontSize: 'calc(10px + 2vmin)'  
}}>
```

```
        Hello World from React in a Docker Container!
```

```
      </p>
```

```
    <a
```

```
    style={{ color: '#61dafb' }}
    href="https://reactjs.org"
    target="_blank"
    rel="noopener noreferrer"
  >
    Learn React
  </a>
</header>
<style>
{
  @keyframes spin {
    from {
      transform: rotate(0deg);
    }
    to {
      transform: rotate(360deg);
    }
  }
}
</style>
</div>
```

```
);  
}
```

```
export default App;
```

```
{  
  "name": "react-docker-app",  
  "version": "0.1.0",  
  "private": true,  
  "dependencies": {  
    "react": "^18.2.0",  
    "react-dom": "^18.2.0",  
    "react-scripts": "5.0.1"  
  },  
  "scripts": {  
    "start": "react-scripts start",  
    "build": "react-scripts build",  
    "test": "react-scripts test",  
    "eject": "react-scripts eject"  
  },  
  "eslintConfig": {
```

```
"extends": [  
  "react-app",  
  "react-app/jest"  
]  
,  
"browserslist": {  
  "production": [  
    ">0.2%",  
    "not dead",  
    "not op_mini all"  
  ],  
  "development": [  
    "last 1 chrome version",  
    "last 1 firefox version",  
    "last 1 safari version"  
  ]  
}  
}
```

Stage 1: Build the React application

FROM node:18-alpine AS build

WORKDIR /app

Copy package.json and package-lock.json

COPY package.json ./

COPY package-lock.json ./

Install dependencies

RUN npm install

Copy the rest of the application's source code

COPY . .

Build the application

RUN npm run build

Stage 2: Serve the application using Nginx

FROM nginx:stable-alpine

Copy the built assets from the 'build' stage

COPY --from=build /app/build /usr/share/nginx/html

Expose port 80 to the outside world

EXPOSE 80

Command to run Nginx in the foreground

CMD ["nginx", "-g", "daemon off;"]

OUTPUT-

