Multi-stage Build

Create a Dockerfile that:

Uses node image to build a React application

Then copies just the built static files to an nginx image

```
ubuntu:~$ mkdir web_app
ubuntu:~$ cd web_app/
ubuntu:~/web_app$ mkdir src
ubuntu:~/web_app$ mkdir public
ubuntu:~/web_app$ touch Dockerfile
ubuntu:~/web_app$ touch package.json
ubuntu:~/web_app$ touch src/App.js
ubuntu:~/web_app$ touch src/index.js
ubuntu:~/web_app$ touch public/index.html
ubuntu:~/web_app$ vi package.json
ubuntu:~/web_app$ vi src/App.js
ubuntu:~/web_app$ vi src/index.js
ubuntu:~/web_app$ vi public/index.html
ubuntu:~/web_app$ vi src/index.js
ubuntu:~/web_app$ vi public/index.html
ubuntu:~/web_app$ vi Dockerfile
ubuntu:~/web_app$ vi Dockerfile
ubuntu:~/web_app$ docker build -t react-app .
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
           Install the buildx component to build images with BuildKit:
           https://docs.docker.com/go/buildx/

Sending build context to Docker daemon  7.168kB
Step 1/10 : FROM node:18 as build
18: Pulling from library/node
23b7d26ef1d2: Pull complete
07d1b5af933d: Pull complete
1eb98adba0eb: Pull complete
b617a119f8a2: Pull complete
ee496386c5de: Pull complete
058db40e5342: Pull complete
04deb1529fda: Pull complete
3b3ca5178f3e: Pull complete
Digest: sha256:df9fa4e0e39c9b97e30240b5bb1d99bdb861573a82002b2c52ac7d6b8d6d773e
Status: Downloaded newer image for node:18
```

Editor Tab 1 +

```jsx
import React from 'react';

function App() {
  return (
    <div>
      <h1>React App in Docker</h1>
      <p>Built with multi-stage Dockerfile</p>
    </div>
  );
}

export default App;
```

Editor Tab 1 +

```json
{
  "name": "react-docker-app",
  "version": "1.0.0",
  "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"
  }
}
```

Editor   Tab 1   +

```jsx
import React from 'react';
import ReactDOM from 'react-dom/client';
import App from './App';

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(<App />);
~
~
~
~
~
```
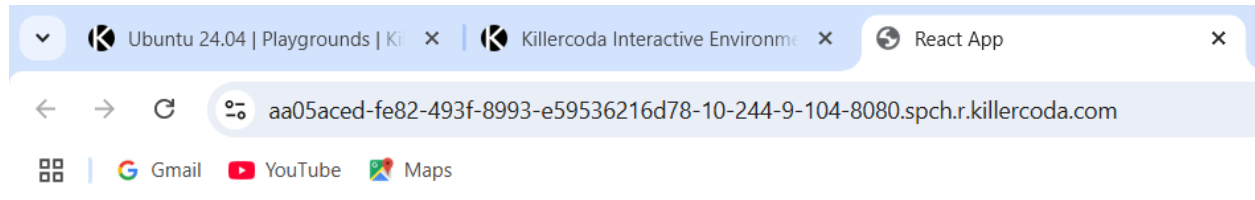
Editor   Tab 1   +

```html
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>React App</title>
  </head>
  <body>
    <div id="root"></div>
  </body>
</html>
~
```

Editor   Tab 1   +

```dockerfile
FROM node:18 as build
WORKDIR /app
COPY package.json ./
RUN npm install
COPY . ./
RUN npm run build

FROM nginx:alpine
COPY --from=build /app/build /usr/share/nginx/html
EXPOSE 80
CMD ["nginx", "-g", "daemon off;"]

~
~
~
```

```
 ---> Removed intermediate container 72ca7c0cd44f
 ---> 53652d2e96d4
Successfully built 53652d2e96d4
Successfully tagged react-app:latest
ubuntu:~/web_app$ docker run -d -p 8080:80 react-app
7af5145905a736146f8d3bee209235a7dc509344db3a1d4f0b988eddcf4e5576
ubuntu:~/web_app$ █
```

React App in Docker

Built with multi-stage Dockerfile