

# CI/CD Pipeline for Dockerized React App

# CI/CD Pipeline Setup for Dockerized React App (GitHub Actions)

This document contains a complete setup for building and deploying a Dockerized React app using GitHub Actions.

## Folder Structure

```
my-react-app/  
├── .github/  
│   └── workflows/  
│       └── ci-cd.yml  
├── Dockerfile  
├── .dockerignore  
├── package.json  
├── public/  
│   └── index.html  
└── src/  
    └── index.js
```

## Dockerfile

```
FROM node:18-alpine AS builder  
WORKDIR /app  
COPY package*.json ./  
RUN npm install --frozen-lockfile  
COPY . .  
RUN npm run build  
  
FROM nginx:alpine  
COPY --from=builder /app/build /usr/share/nginx/html  
EXPOSE 80  
CMD ["nginx", "-g", "daemon off;"]
```

## .dockerignore

```
node_modules  
build  
.dockerignore  
Dockerfile  
.git  
.gitignore  
README.md
```

## package.json

```
{  
  "name": "my-react-app",  
  "version": "1.0.0",  
  "private": true,  
  "dependencies": {  
    "react": "^18.3.1",  
    "react-dom": "^18.3.1",  
    "react-scripts": "5.0.1"  
  },  
  "scripts": {  
    "start": "react-scripts start",  
    "build": "react-scripts build",  
    "test": "react-scripts test",  
    "eject": "react-scripts eject"  
  }  
}
```

## public/index.html  
<!DOCTYPE html>

```

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

-----

## src/index.js

import React from "react";
import ReactDOM from "react-dom/client";

const App = () => {
  return (
    <div style={{ textAlign: "center", marginTop: "60px" }}>
      <h1>■ CI/CD Enabled Dockerized React App</h1>
      <p>This app is automatically built and deployed using GitHub Actions!</p>
    </div>
  );
};

const root = ReactDOM.createRoot(document.getElementById("root"));
root.render(<App />);

-----

## GitHub Actions Workflow (.github/workflows/ci-cd.yml)

name: CI/CD Pipeline for React App

on:
  push:
    branches:
      - main

jobs:
  build:
    runs-on: ubuntu-latest

    steps:
      - name: ■ Checkout repository
        uses: actions/checkout@v4

      - name: ■■ Set up Node.js
        uses: actions/setup-node@v4
        with:
          node-version: 18

      - name: ■ Install dependencies
        run: npm install --frozen-lockfile

      - name: ■ Build React app
        run: npm run build

      - name: ■ Run tests
        run: npm test -- --watchAll=false

  docker:
    runs-on: ubuntu-latest
    needs: build

    steps:
      - name: ■ Checkout repository
        uses: actions/checkout@v4

      - name: ■ Log in to Docker Hub
        uses: docker/login-action@v3
        with:
          username: ${ secrets.DOCKERHUB_USERNAME }
          password: ${ secrets.DOCKERHUB_TOKEN }

      - name: ■ Build and push Docker image

```

```
uses: docker/build-push-action@v6
with:
  context: .
  push: true
  tags: ${{ secrets.DOCKERHUB_USERNAME }}/my-react-app:latest
```

-----

### ## GitHub Secrets Setup

In your repository → Settings → Secrets and variables → Actions:

- DOCKERHUB\_USERNAME → your Docker Hub username
- DOCKERHUB\_TOKEN → a Docker Hub access token

-----

### ## Summary

- Automatically builds your React app
- Runs tests before deployment
- Builds and pushes Docker image to Docker Hub
- Fully automated CI/CD pipeline using GitHub Actions