

## Cloud Deployment — React App with Docker & GitHub Actions

This project demonstrates how to containerize a React application with Docker and configure a CI/CD pipeline using GitHub Actions to automatically test, build, and deploy the app to GitHub Pages.

### Objective:

- Dockerize a production-ready React application using multi-stage builds.
- Automate testing and deployment using GitHub Actions.
- Deploy the final build to GitHub Pages.

### Tech Stack:

Frontend: React

Containerization: Docker (Node + Nginx multi-stage)

CI/CD: GitHub Actions

Deployment: GitHub Pages

### Dockerfile (Multi-Stage Build):

```
FROM node:18-alpine AS build
```

```
WORKDIR /app
```

```
COPY package*.json ./
```

```
RUN npm install
```

```
COPY . .
```

```
RUN npm run build
```

```
FROM nginx:stable-alpine
```

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

```
EXPOSE 80
```

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

### GitHub Actions Workflow (.github/workflows/main.yml):

```
name: CI/CD Pipeline with GitHub Pages Deployment
```

```
on:
```

```
push:
```

```
branches: ["main"]
```

```
jobs:
```

```
deploy:
```

```
runs-on: ubuntu-latest
```

```
steps:
```

```
- uses: actions/checkout@v3
```

```
- uses: actions/setup-node@v3
```

```
with:
```

node-version: "18"

- run: npm install

- run: npm test --if-present

- run: npm run build

- uses: peaceiris/actions-gh-pages@v3

with:

github\_token: \${{ secrets.GITHUB\_TOKEN }}

publish\_dir: ./build

GitHub Pages Setup:

Add to package.json:

"homepage": "https://.github.io/"

Then enable GitHub Pages from repository settings.

Run Locally:

npm install

npm start

Run with Docker:

docker build -t react-app-prod .

docker run -p 80:80 react-app-prod