

Dockerfile

FROM node:18-alpine 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;"]

Dockerfile Build & Run:

docker build -t react-app .

docker run -p 8080:80 react-app

Push to Dockerhub:

docker login

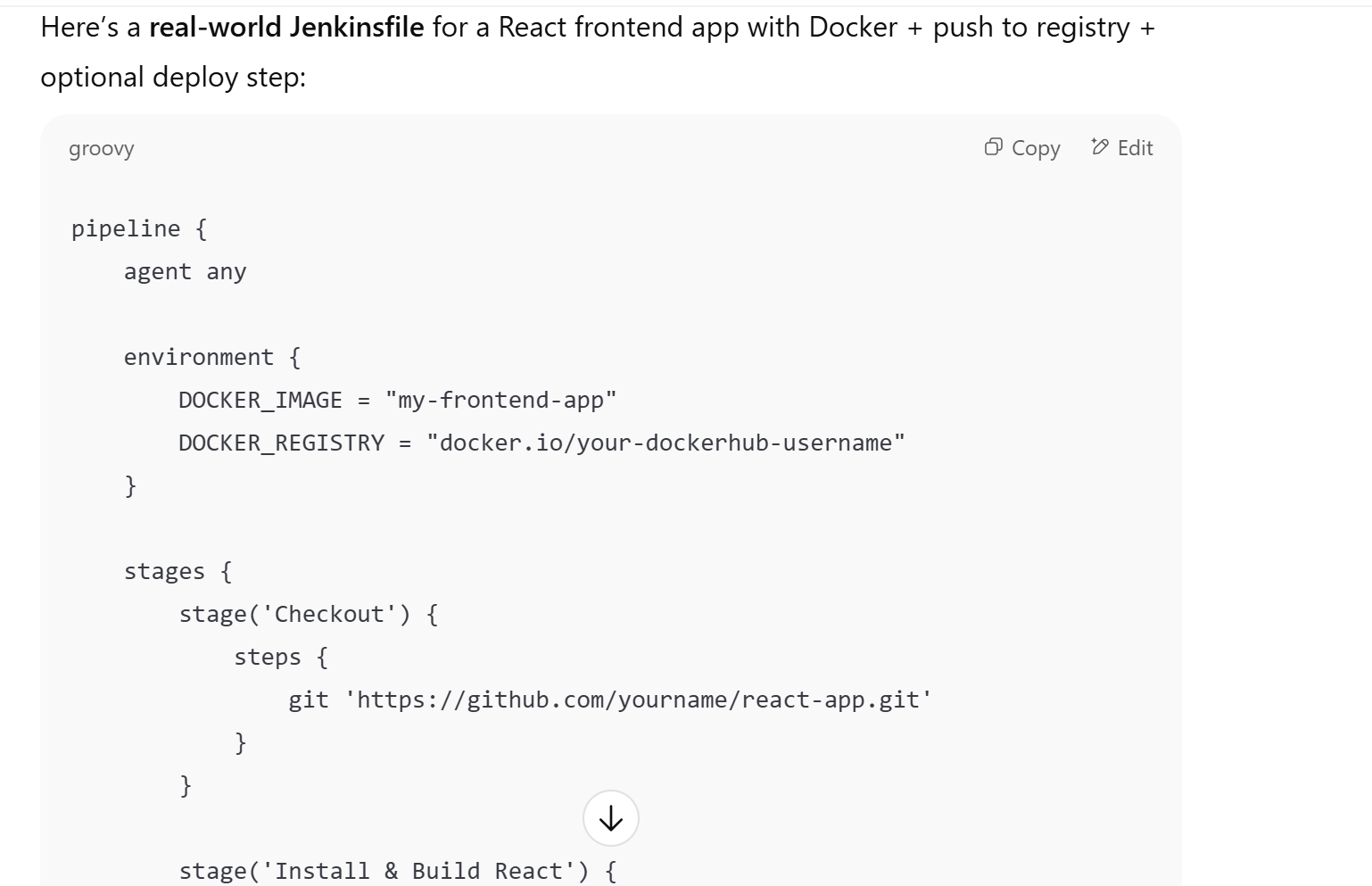
docker tag react-app yourname/react-app:v1

docker push yourname/react-app:v1

👉 **Summary**: In dev, Docker build is optional — mainly used for consistency or testing.

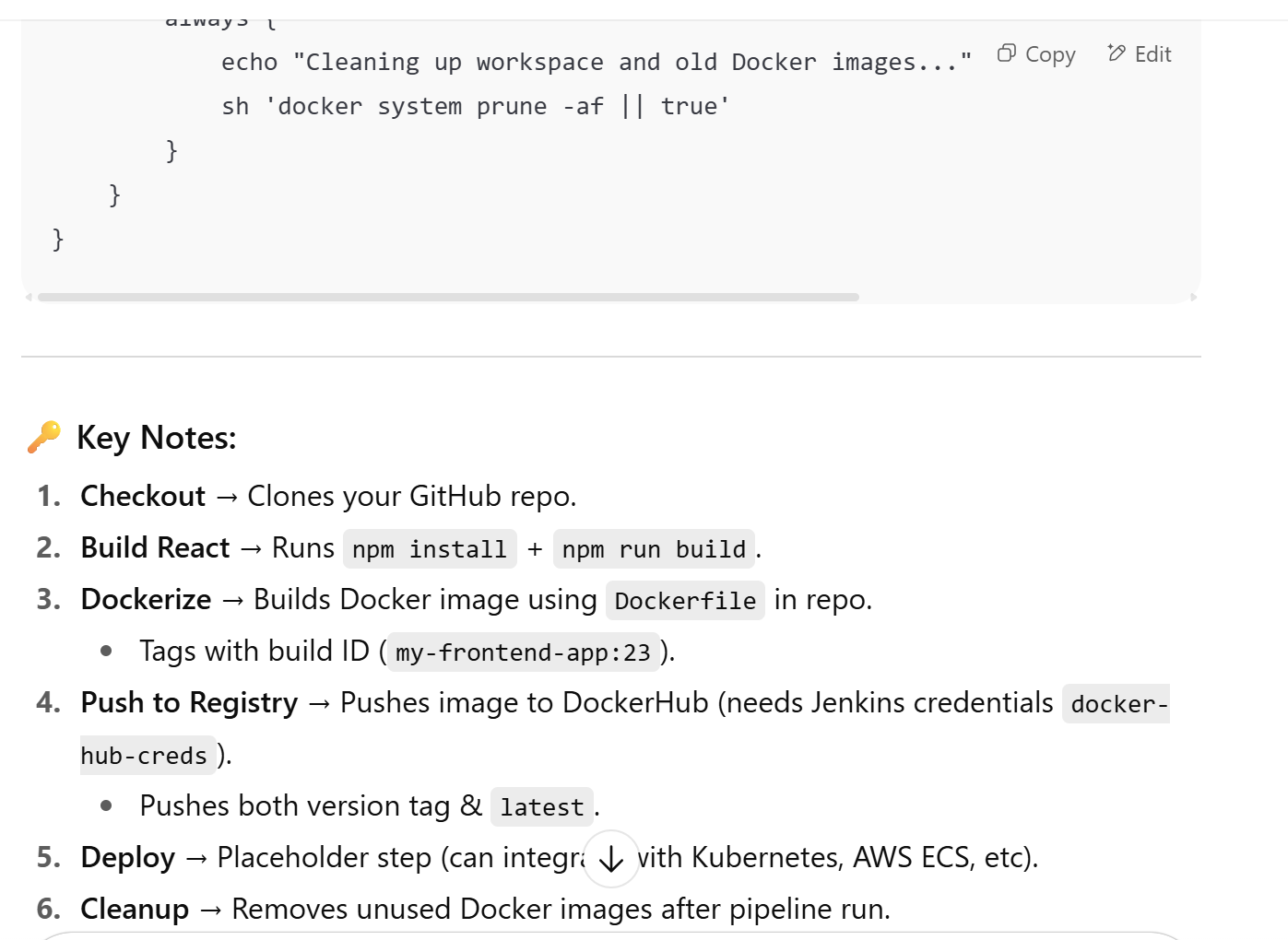
👉 **Summary**: In Jenkins, Docker build is essential for production deployment.







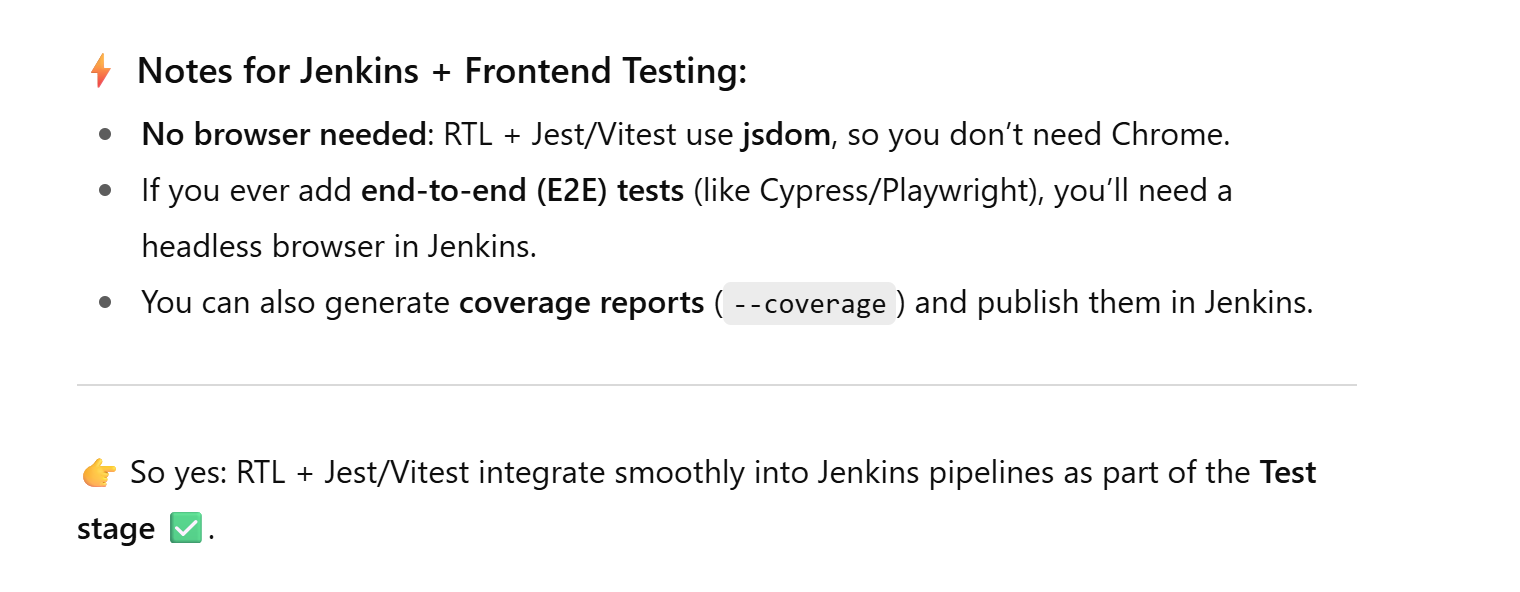




**🔹 So in summary:**

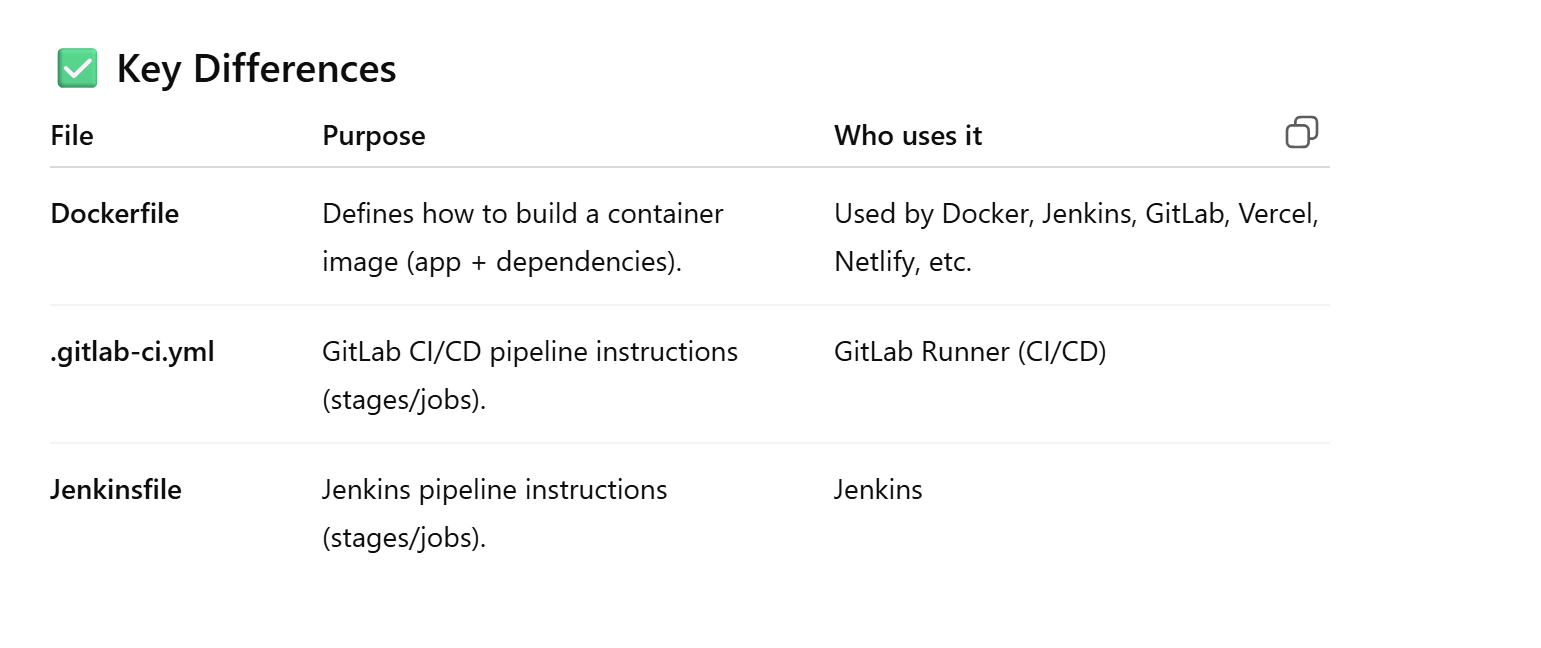
* **Jenkins uses the Dockerfile from your repo.**
* **Jenkins pipeline** just automates: build → tag → push → deploy.
* Jenkins doesn’t create its own Dockerfile — unless your pipeline script explicitly writes one (rare, but possible).

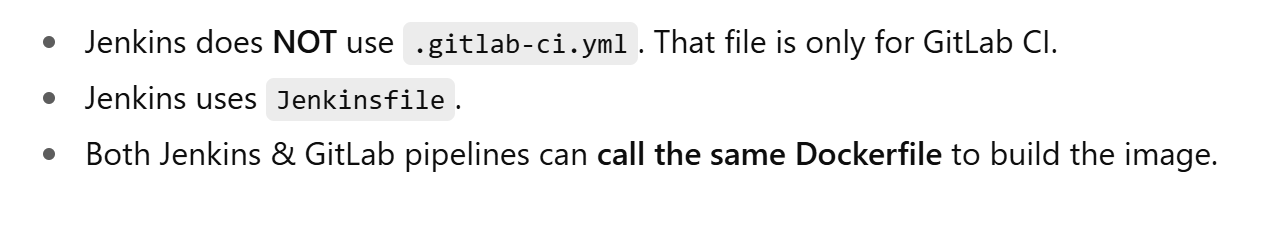
**🔑 When to test in the pipeline**

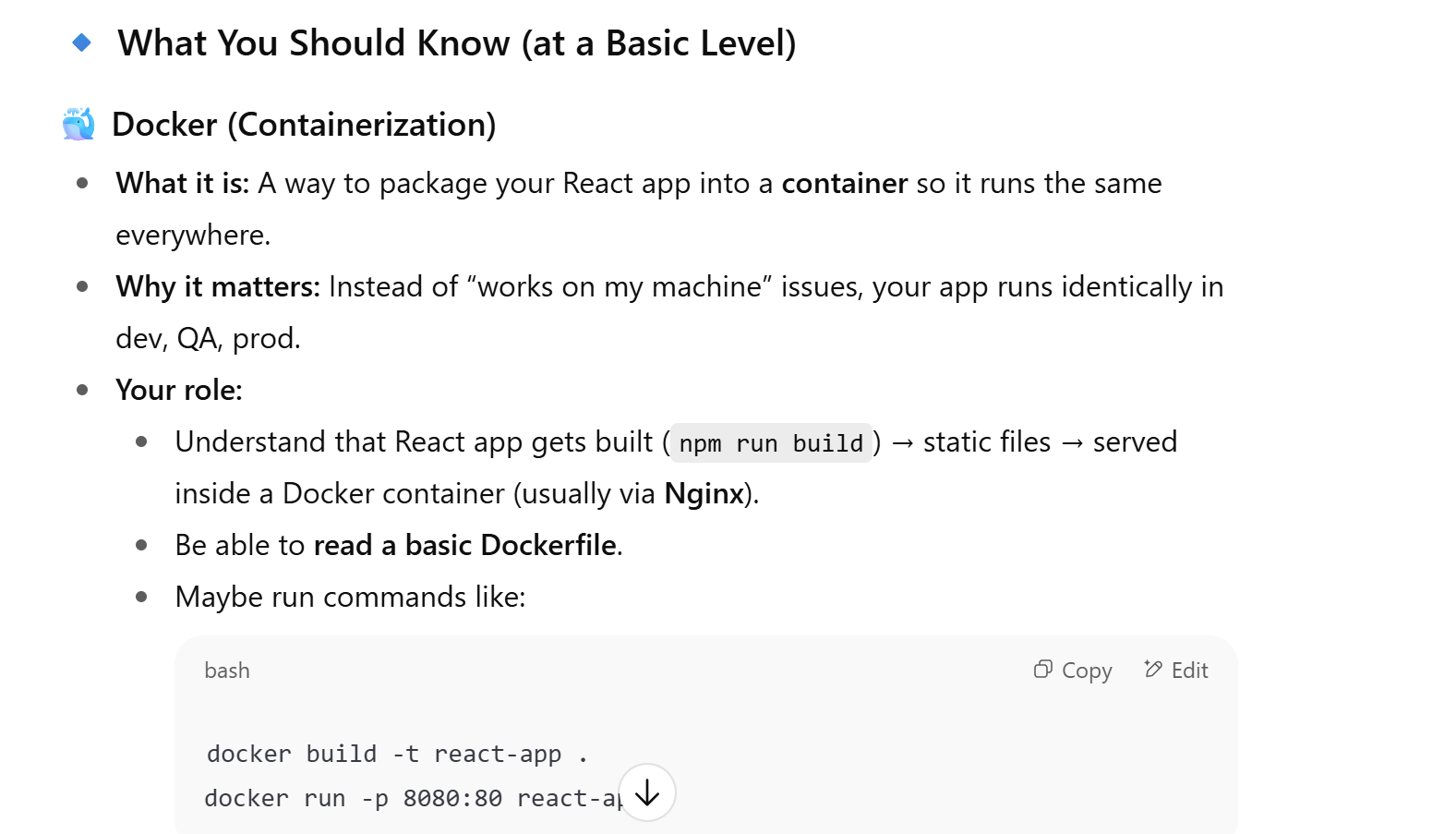
* ✅ **After npm install** → because dependencies are required for tests.
* ✅ **Before npm run build & Docker** → because there’s no point building & pushing a broken app.
* ✅ You can also add **linting (eslint, prettier checks)** before or inside the Test stage.
* 

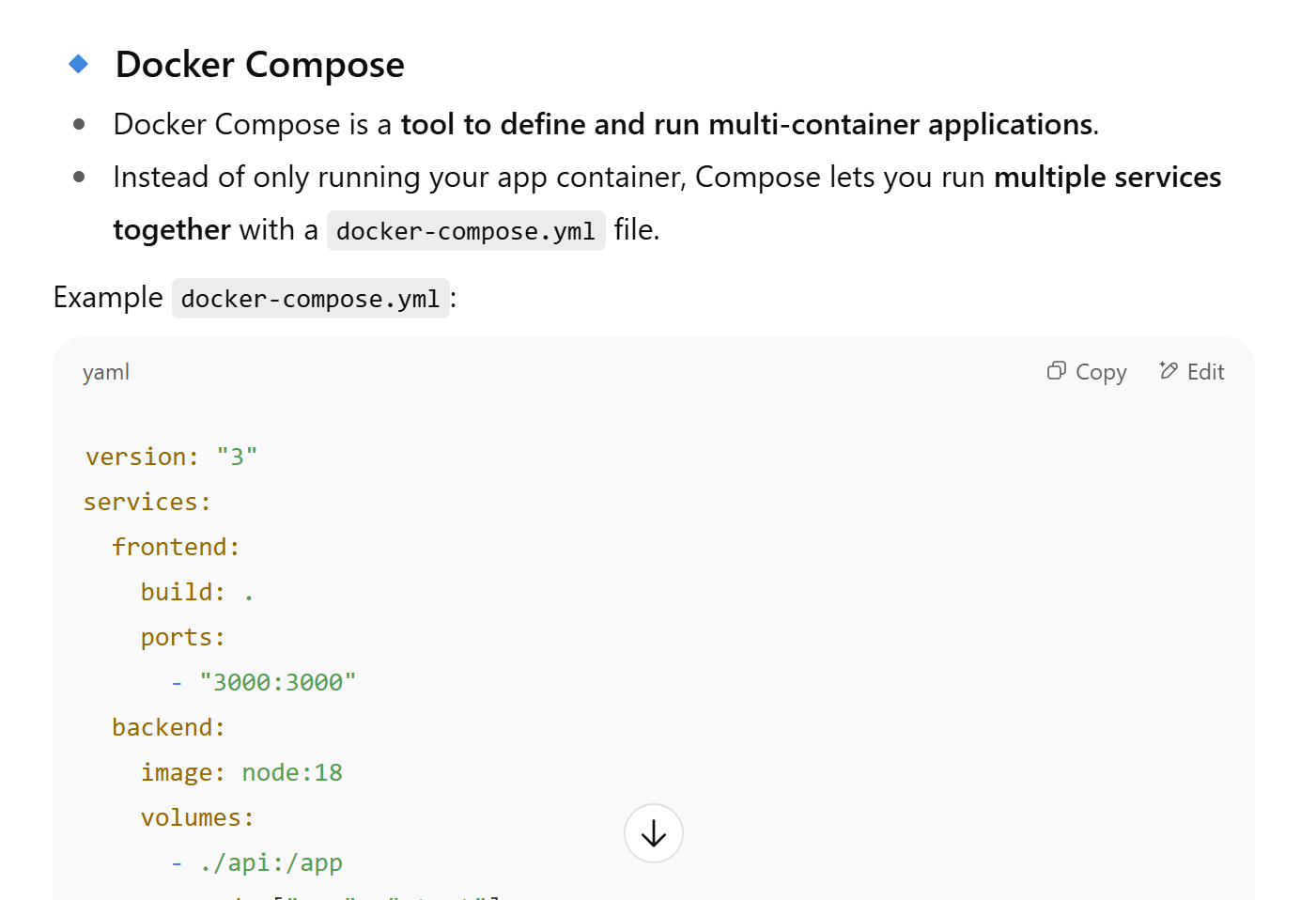
✅ Correct structure:

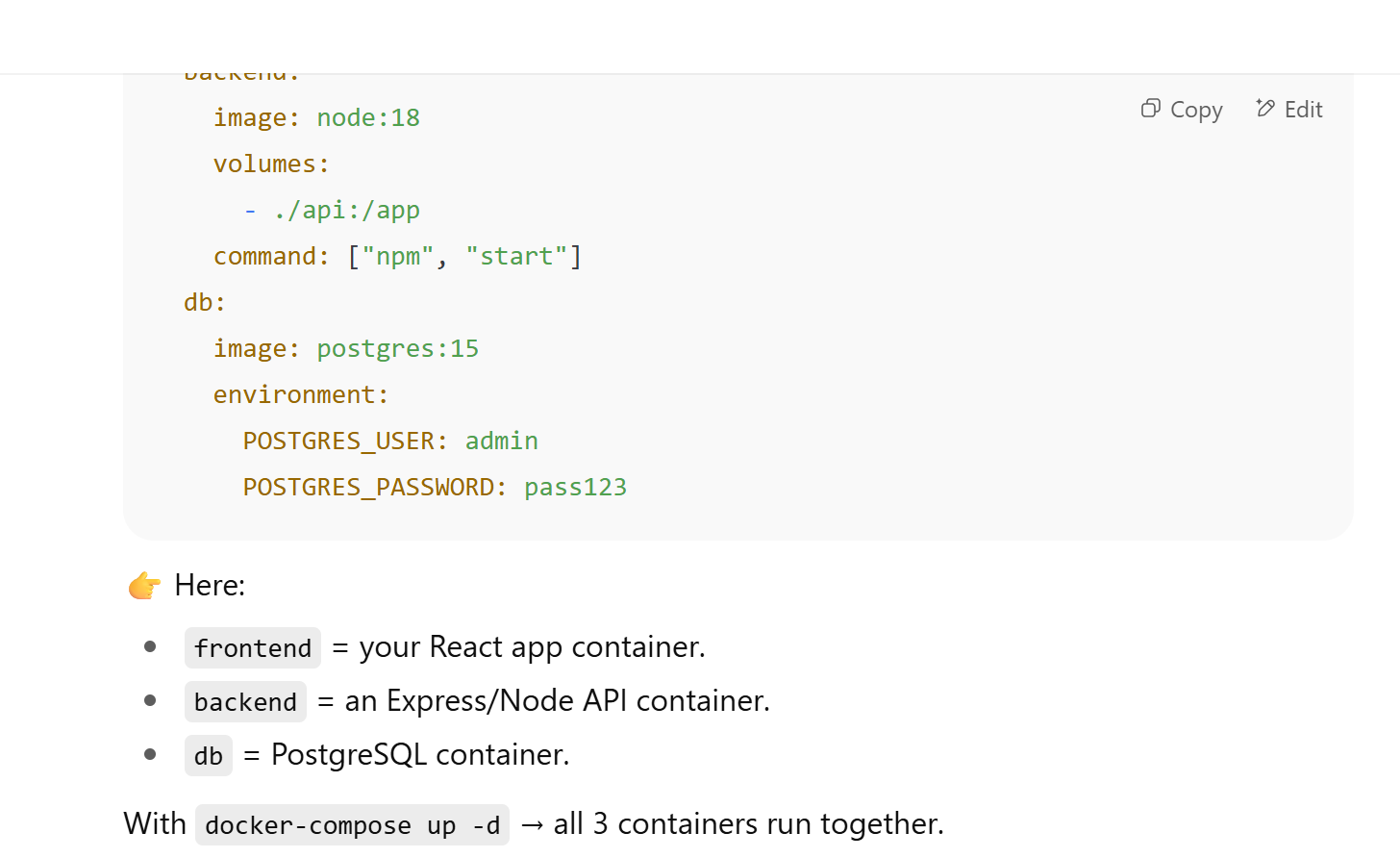
* **Unit Tests, Lint, Static Analysis** → before build.
* **E2E/Integration Tests** → after build.



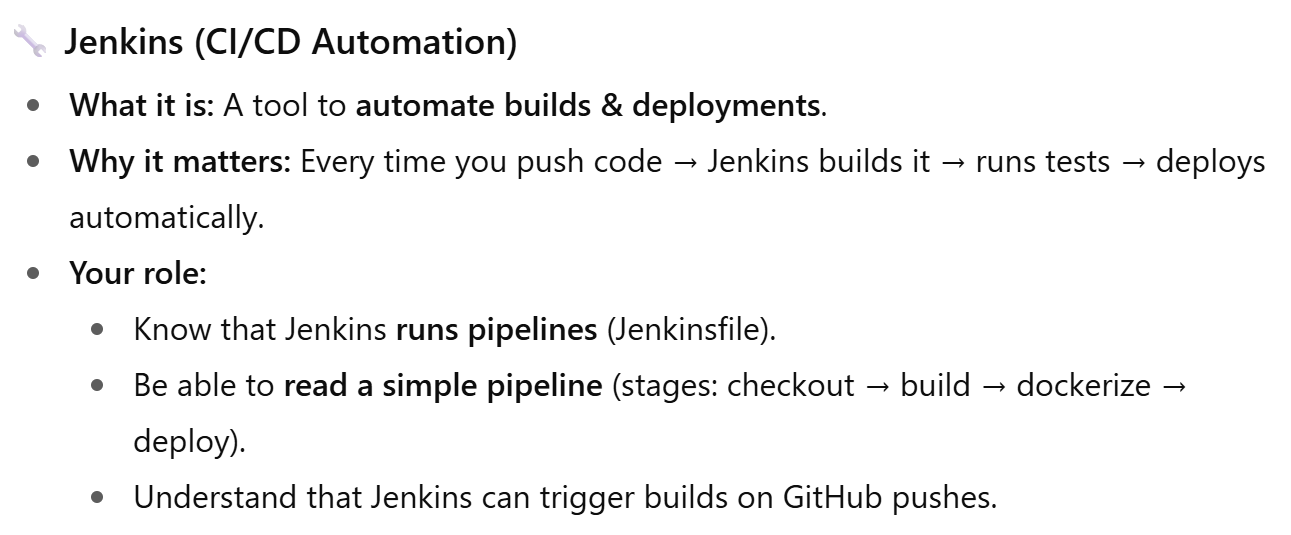


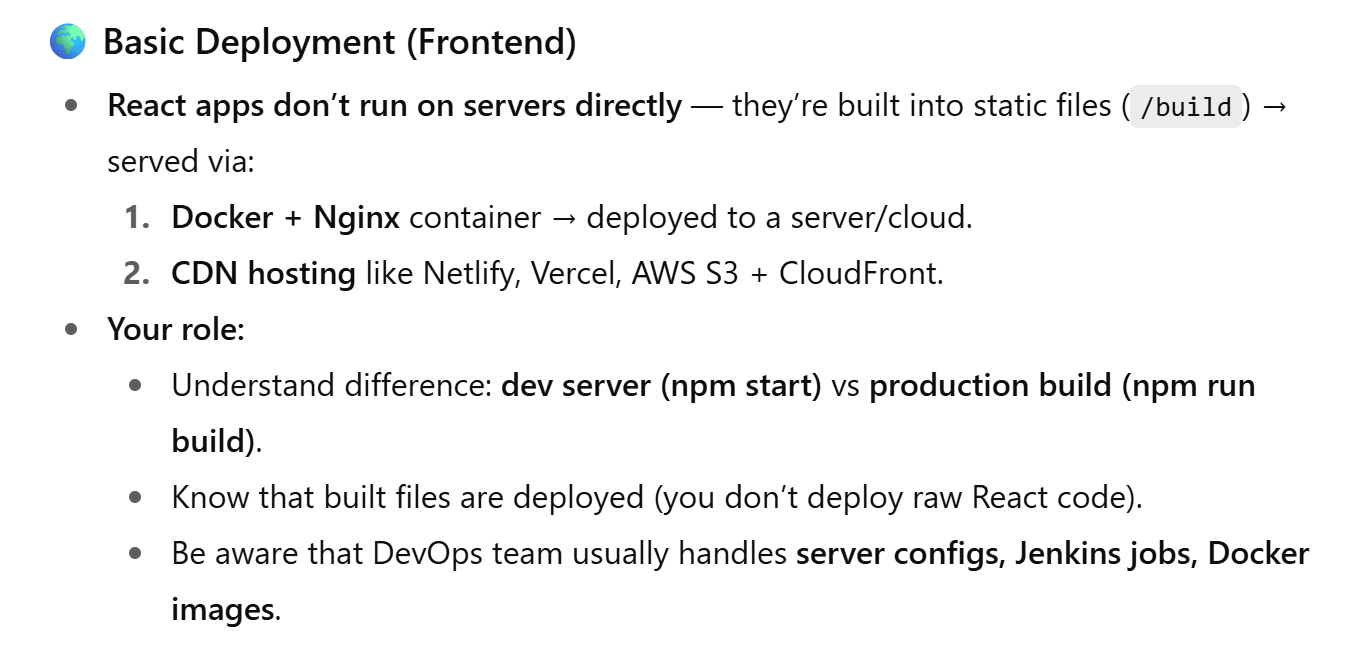


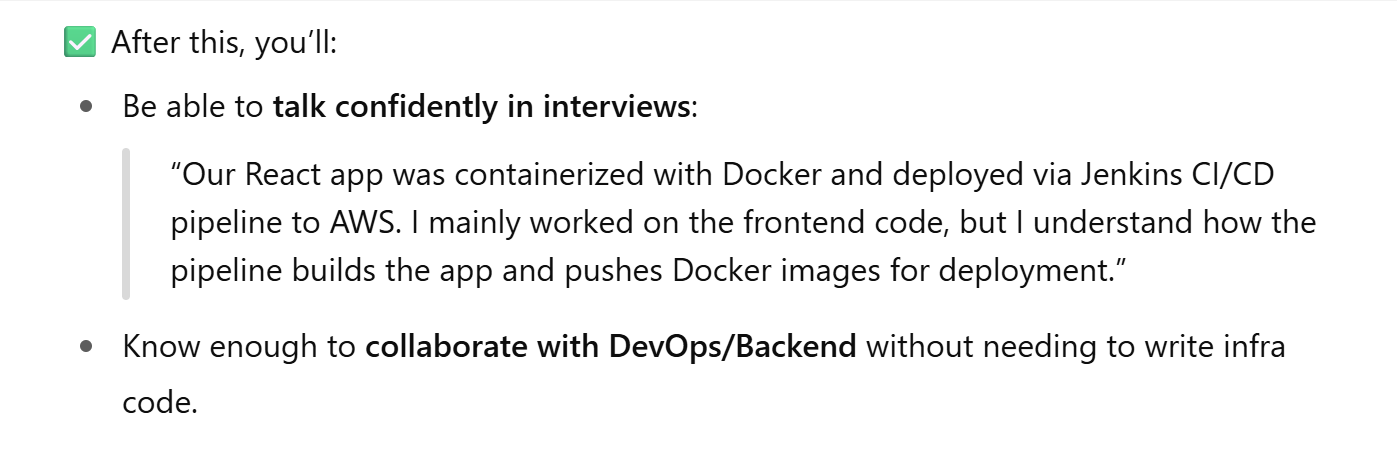


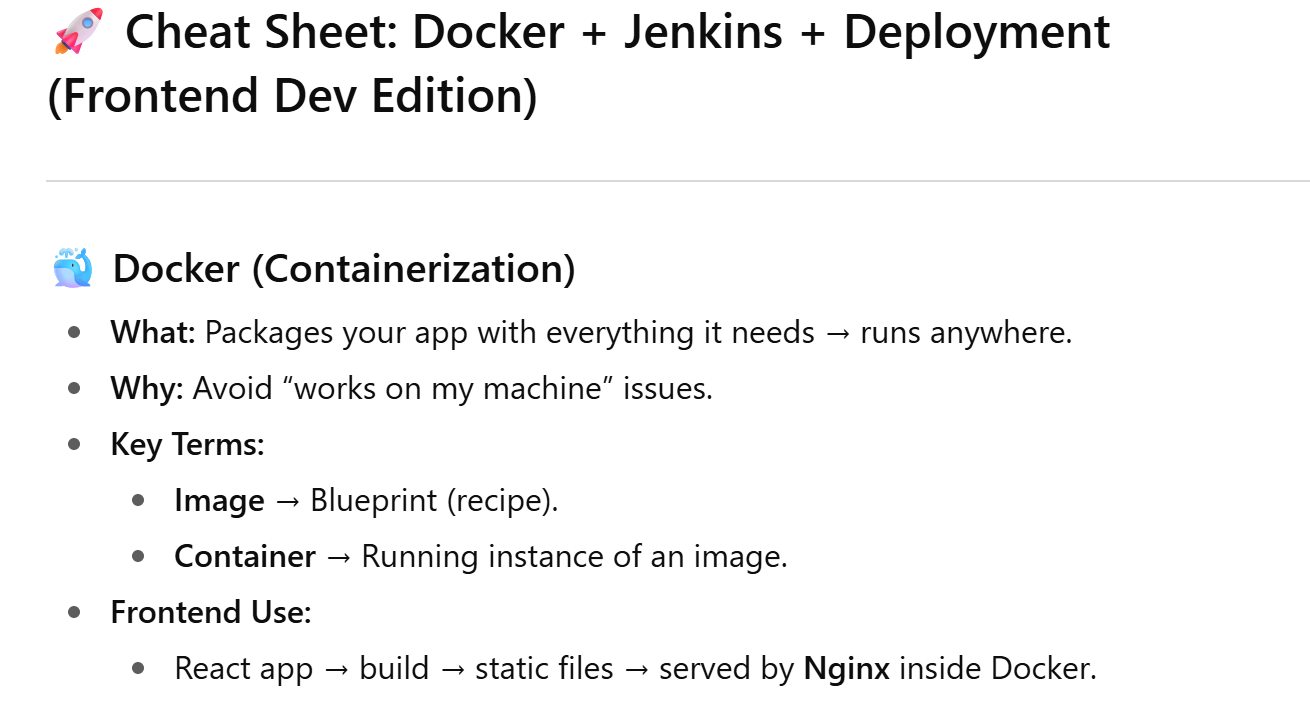


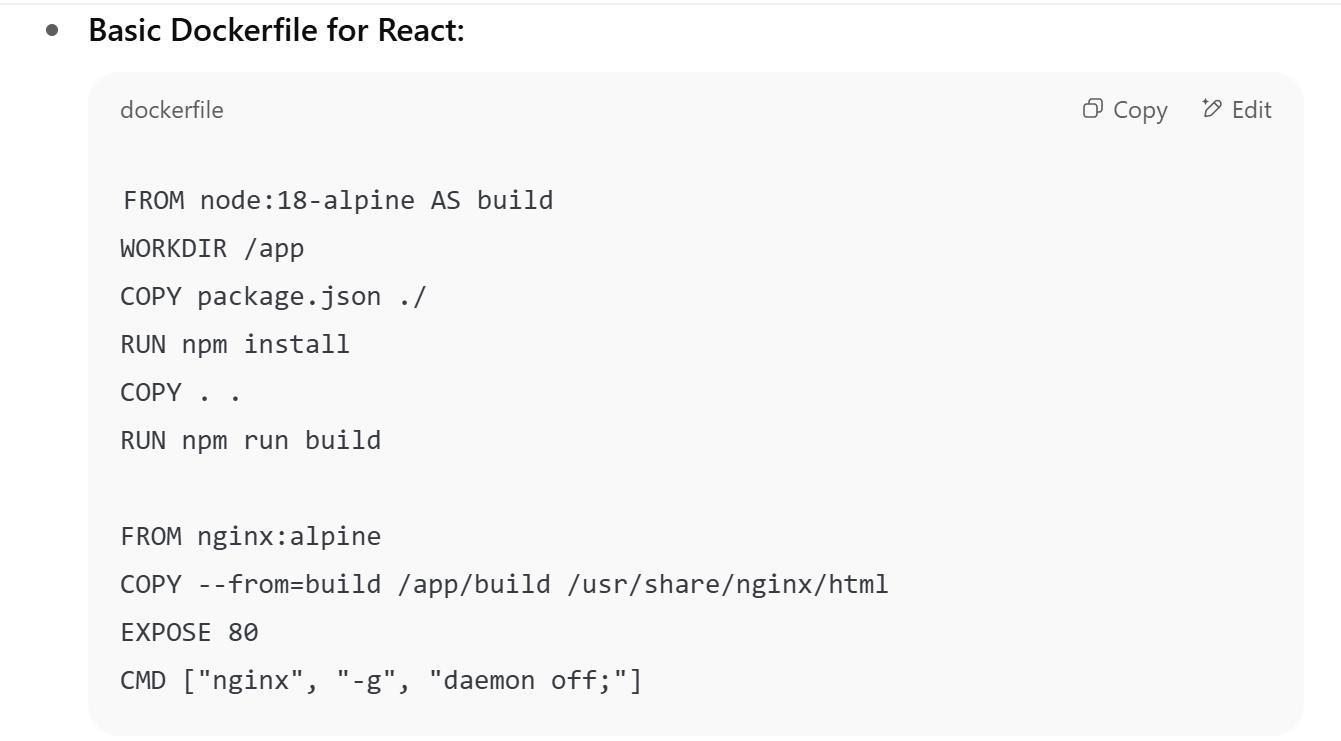
Use **Compose** for local development/testing.

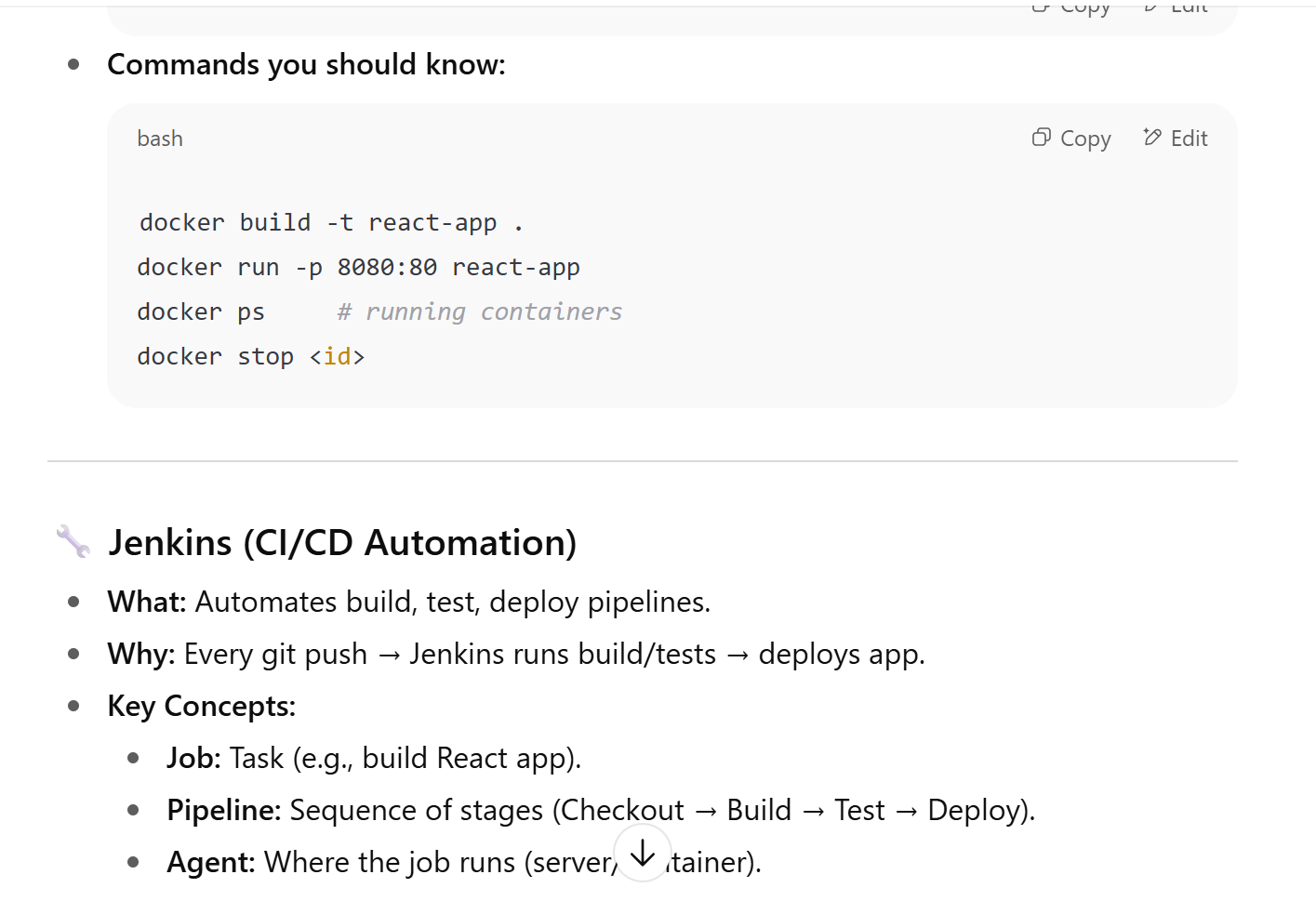




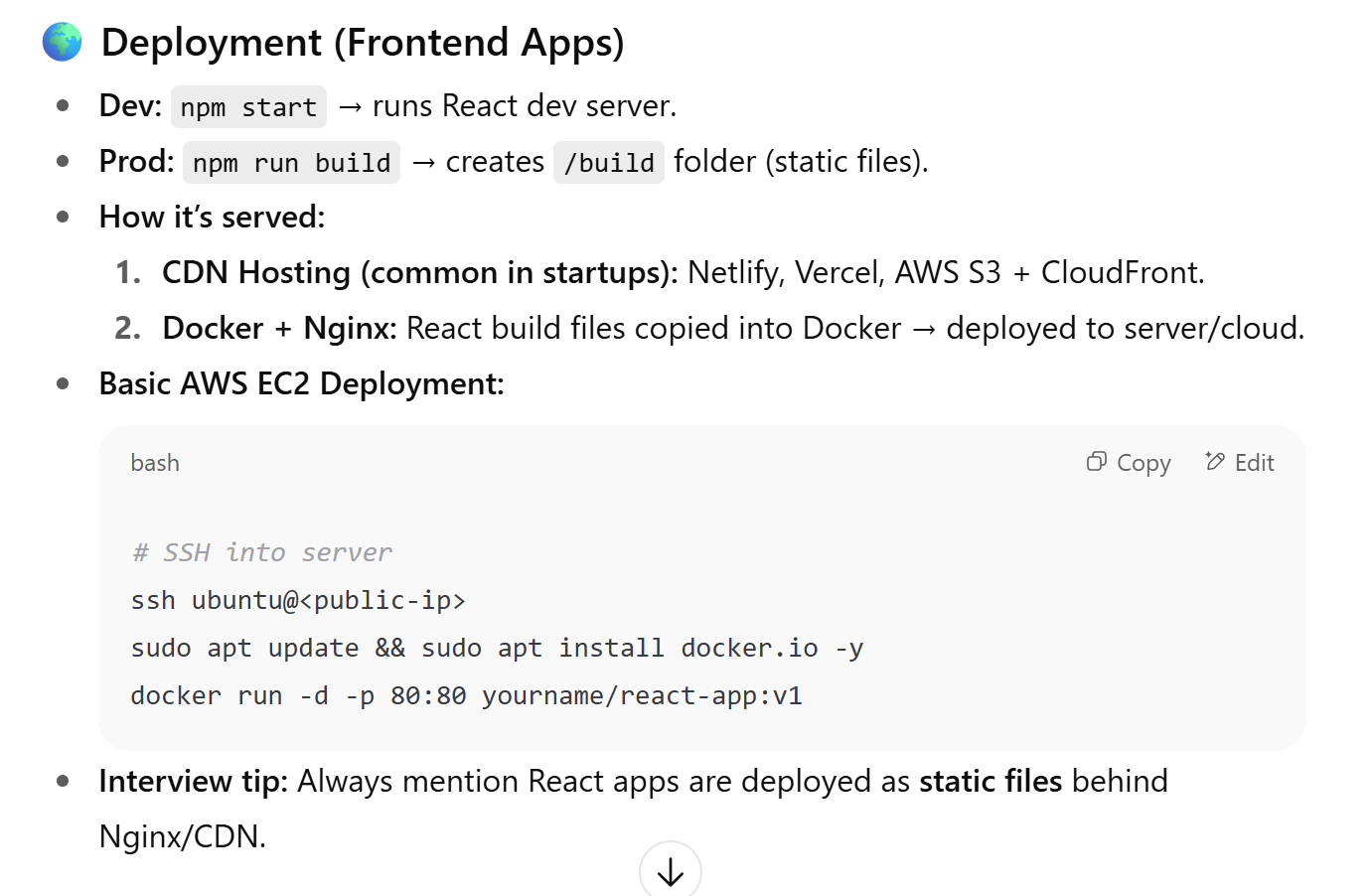


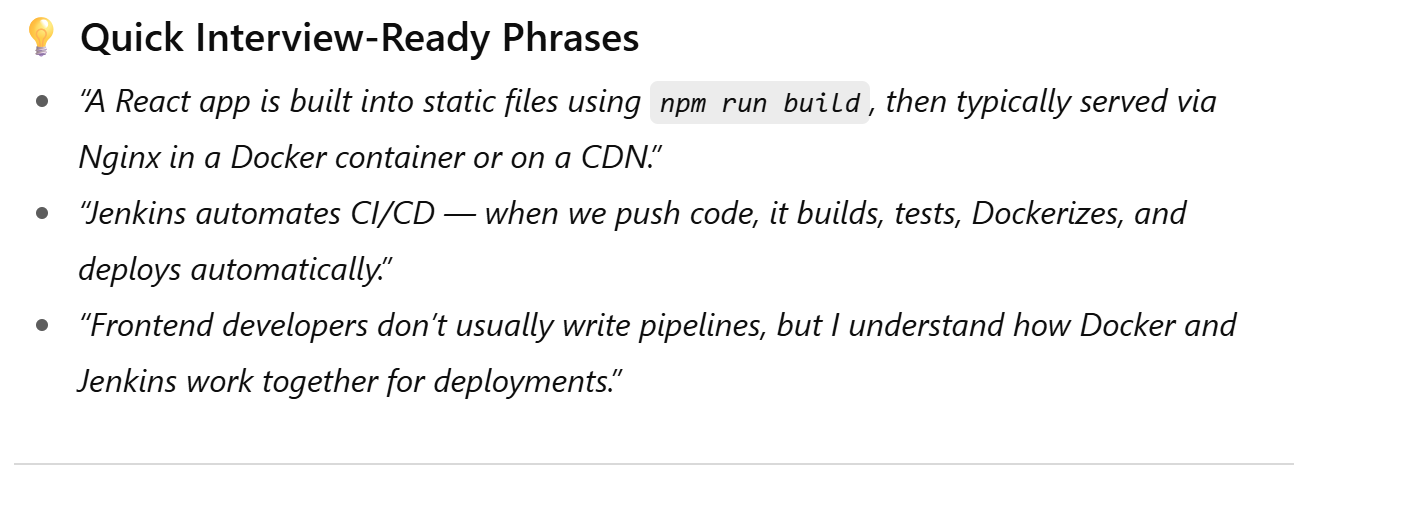


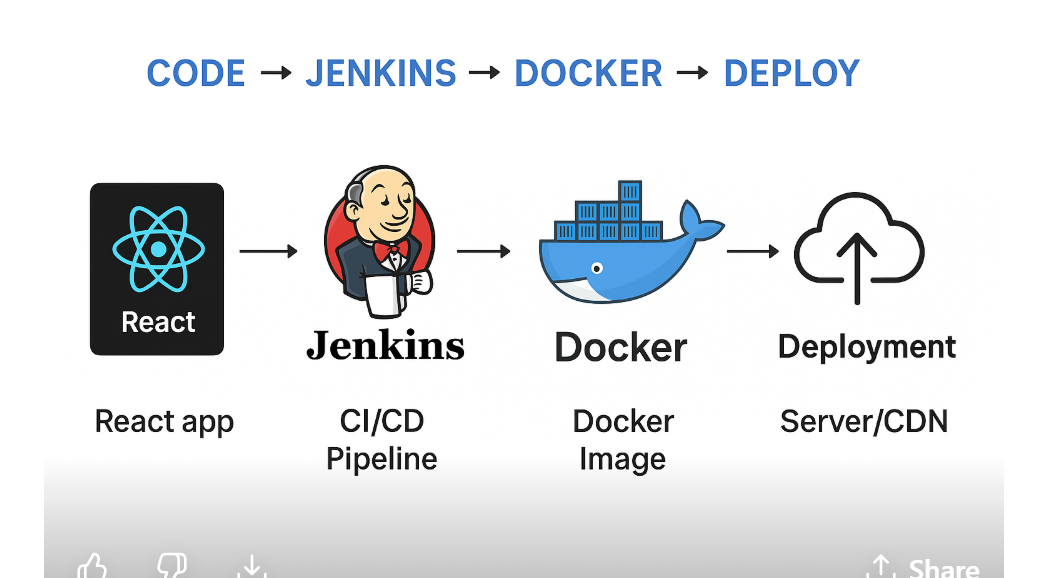


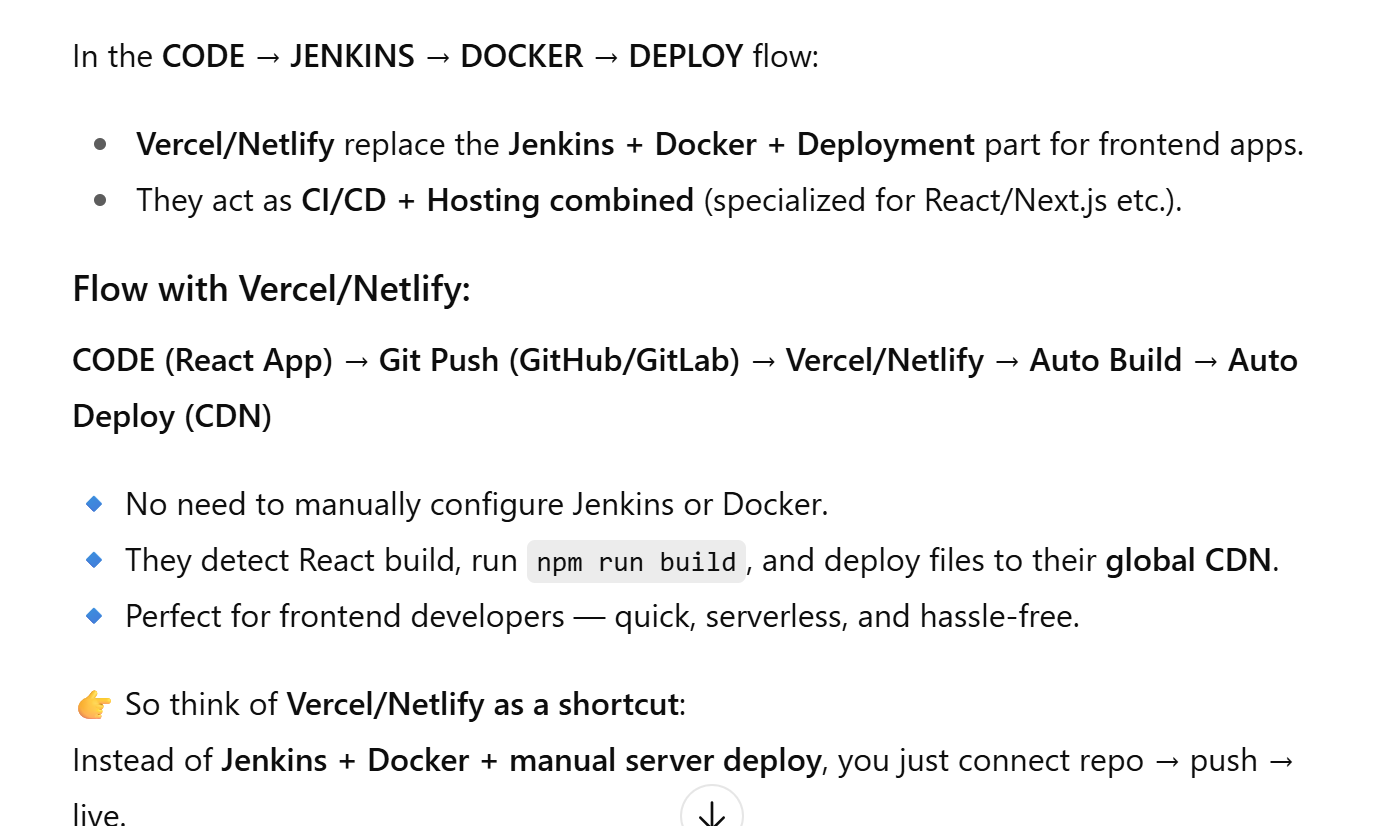




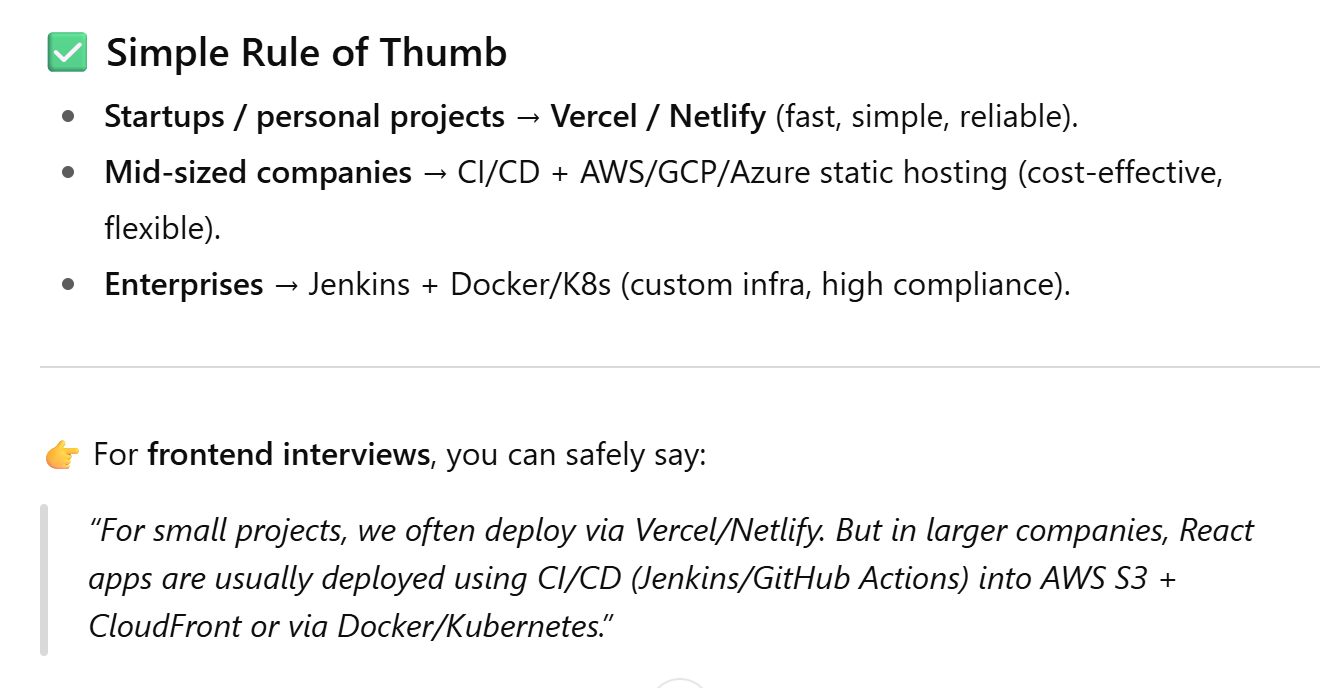












Deployment/Hosting: Vercel, Netlify, GitHub Pages

CI/CD Basics: GitHub Actions, Jenkins (basic understanding)

**Vercel** (very strong in Next.js ecosystem) -> Popular in **startups and Next.js-heavy companies**. -> used by  OpenAI (docs & microsites)

 TikTok (some web properties)

 Uber (developer portals)

Netlify -> Often used for **JAMstack sites, marketing pages, dashboards**.-> used by  Nike (landing pages & microsites)

 Figma (community/microsites)

**🌐 What is Hostinger?**

* **Hostinger** is a **traditional web hosting provider**.
* It provides:
  + Shared Hosting
  + VPS Hosting
  + Cloud Hosting
  + Domain names, emails, WordPress hosting
* Target audience: individuals, small businesses, WordPress sites, blogs, static websites.

