

React Multi-Stage Dockerfile Reference Points

Stage 1: Base Stage

- **Install System Dependencies**

dockerfile

```
RUN apk add --no-cache git curl bash ca-certificates && rm -rf /var/cache/apk/*
```

- **Build Dependencies Contains**

dockerfile

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

```
WORKDIR /app
```

```
RUN addgroup -g 1001 -S nodejs && adduser -S reactuser -u 1001
```

Stage 2: Dependencies Stage

- **Copy Dependency Files First (for better caching)**

dockerfile

```
COPY package.json package-lock.json* yarn.lock* pnpm-lock.yaml* ./
```

- **Install Dependencies with Caching**

dockerfile

```
RUN npm config set fund false && npm config set audit-level moderate
```

```
RUN npm ci --only=production=false --no-audit --prefer-offline --progress=false
```

- **Install Development Dependencies**

dockerfile

```
# For production dependencies only
```

```
RUN npm ci --only=production --no-audit --prefer-offline --progress=false
```

Stage 3: Build Stage

- **Copy Source Code**

dockerfile

```
COPY . .  
ENV NODE_ENV=production  
ENV GENERATE_SOURCEMAP=false
```

- **Code Quality Checks**

dockerfile

```
RUN npm run lint 2>/dev/null || echo "Lint check completed"  
RUN npm run format:check 2>/dev/null || echo "Format check completed"  
RUN npx tsc --noEmit --skipLibCheck 2>/dev/null || echo "TypeScript check completed"
```

- **Run Tests**

dockerfile

```
RUN CI=true npm run test -- --coverage --watchAll=false --passWithNoTests
```

- **Build Application**

dockerfile

```
RUN npm run build
```

- **Verify Build Artifacts**

dockerfile

```
RUN ls -la build/ && test -f build/index.html && echo "Build verified"  
RUN du -sh build/
```

- **Security Audit**

dockerfile

```
RUN npm audit --audit-level=moderate || echo "Security audit completed"
```

- **Vulnerability Scanning**

dockerfile

```
RUN npm audit --audit-level=high --production || echo "Production audit completed"
```

- **Build Stage Cleanup**

dockerfile

```
RUN rm -rf src/ public/ node_modules/.cache/ && npm cache clean --force
```

Stage 4: Production Stage

- **Install Runtime Dependencies**

dockerfile

```
RUN apk add --no-cache curl ca-certificates tzdata && rm -rf /var/cache/apk/*
```

- **Copy Build Artifacts from Build Stage**

dockerfile

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

- **Copy Production Dependencies**

dockerfile

```
COPY --from=prod-dependencies /app/node_modules ./node_modules
```

- **Copy Configuration Files (if they exist)**

dockerfile

```
COPY nginx.conf* /etc/nginx/conf.d/
```

```
COPY docker-entrypoint.sh* /docker-entrypoint.d/
```

```
COPY .env.production* /usr/share/nginx/html/
```