

# React Multi-Stage Dockerfile Reference Points

## Stage 1: Base Stage

- **Install System Dependencies**
  - Essential system packages (git, curl, bash, ca-certificates)
  - Alpine package manager cleanup
- **Build Dependencies Contains**
  - Node.js base image selection
  - Working directory setup
  - Non-root user creation for security

## Stage 2: Dependencies Stage

- **Copy Dependency Files First (for better caching)**
  - package.json, package-lock.json, yarn.lock, pnpm-lock.yaml
  - Leverage Docker layer caching
- **Install Dependencies with Caching**
  - Configure npm settings for better performance
  - Support multiple package managers (npm/yarn/pnpm)
  - Use frozen lockfiles for reproducible builds
- **Install Development Dependencies**
  - Separate production vs development dependency installation
  - Create separate stage for prod-only dependencies

## Stage 3: Build Stage

- **Copy Source Code**
  - Copy all application source files
  - Set build environment variables
- **Code Quality Checks**
  - ESLint validation
  - Prettier formatting checks
  - TypeScript type checking (if applicable)
- **Run Tests**

- Execute test suites with CI configuration
- Generate code coverage reports
- Non-blocking test execution
- **Build Application**
  - Run production build command
  - Optimize for production deployment
- **Verify Build Artifacts**
  - Check build directory existence
  - Validate essential files (index.html)
  - Monitor build size
- **Security Audit**
  - npm audit execution
  - Moderate/high level vulnerability checks
- **Vulnerability Scanning**
  - Additional security validations
  - Production-specific audit checks
- **Build Stage Cleanup**
  - Remove source code and dev dependencies
  - Clear npm cache
  - Minimize layer size

## Stage 4: Production Stage

- **Install Runtime Dependencies**
  - Minimal runtime system packages
  - Timezone data and certificates
- **Copy Build Artifacts from Build Stage**
  - Transfer built application files
  - Optimize for serving static content
- **Copy Production Dependencies**
  - Include only necessary runtime dependencies
  - Exclude development packages
- **Copy Configuration Files (if they exist)**

- Custom nginx configuration
- Environment-specific settings
- Docker entrypoint scripts
- Production environment files