

Notes

Deploying production React app.

Date: 2023-11-10

By: csr13

Download as PDF

React is entrypoint for entry level frontend developers, it's a popular framework.

So, is time to deploy production. You ask frontend team (always arrogant) lead: why deploy instructions dictate only how to run react development build? Where production instructions?

But when I deploy and see a source tree map of the entire application on the browser development settings, with all the source code for the frontend.

Ask frontend for solution to do this, they say "not sure".

When deploying for production react apps, react is nice enough to let you compile the app in a single build/ directory. For not generating source map, a small but *very* important step.

Dockerfile for nginx/frontend

```
#syntax=docker/dockerfile:1
FROM node AS frontend
WORKDIR /frontend
COPY frontend .
RUN yarn install
RUN GENERATE_SOURCEMAP=false npm run build

FROM nginx

RUN rm -rfvd /etc/nginx/conf.d && \
    mkdir -p /usr/share/nginx/html/static && \
    mkdir -p /var/www/html && \
    mkdir -p /etc/nginx/conf.d

COPY --from=frontend /frontend/build /var/www/html
COPY ./docker/nginx/conf.d/default.conf /etc/nginx/conf.d/default.conf
```

The main thing to look at is the RUN step for npm run build; passing environment variable behind it, GENERATE_SOURCEMAP=false does the trick. Multi-stage builds used on the Dockerfile, check docker best practices for advices.

Seems like small step, but makes big difference; see figure 1.2 below

No source map tree on production builds, sysadmin can rest better.



Figure 1.1



Figure 1.2

Last thing to do is modify nginx config file so that it can search for that directory, which was copied to /var/www/html inside nginx container (see nginx build stage on Dockerfile)

```
upstream backend {
    server api:6969 weight=2;
}

server {
    ...
    ...

    location /api/ {
        ...
        ...
        proxy_pass http://backend;
    }

    location / {
        root /var/www/html;
        try_files $uri /index.html;
    }
}
```