

## docker-compose.yml

```
1 version: '3.7'
2 services:
3   db:
4     image: postgres:12.2
5     env_file:
6       - ./backend/src/.env.prod.db
7     volumes:
8       - postgres_data:/var/lib/postgresql/data/
9   backend:
10    build:
11      context: ./backend/src
12      dockerfile: Dockerfile.prod
13    command: gunicorn backend.wsgi:application --bind 0.0.0.0:8000
14    volumes:
15      - static_volume:/home/app/web/static
16    expose:
17      - 8000
18    depends_on:
19      - db
20    env_file:
21      - ./backend/src/.env.prod
22  nginx:
23    build:
24      context: ./frontend
25      dockerfile: Dockerfile
26    ports:
27      - '8082:80'
28    depends_on:
29      - backend
30  volumes:
31    postgres_data:
32    static_volume:
```

## Dockerfile

```
1 # build stage
2 FROM node:13.7-alpine as build
3 WORKDIR /app
4 COPY package.json /app/package.json
5 RUN yarn global add @quasar/cli
6 RUN apk --no-cache add pkgconfig autoconf automake libtool nasm build-base
7 COPY . /app
8 RUN yarn
9 RUN quasar build
10
11 # production environment
12 FROM nginx:1.17
13 COPY --from-build /app/dist/spa /usr/share/nginx/html
14 RUN rm /etc/nginx/conf.d/default.conf
15 COPY nginx/nginx.conf /etc/nginx/conf.d
16 EXPOSE 80
17 CMD ["nginx", "-g", "daemon off;"]
```

## Dockerfile.prod

```
1 # build
2 FROM python:3.8 as builder
3
4 WORKDIR /usr/src/app
5
6 ENV PYTHONUNBUFFERED 1
7 ENV PYTHONDONTWRITEBYTECODE 1
8
9 RUN apt-get update && apt-get -y install netcat
10
11 COPY . /usr/src/app/
12 RUN pip wheel --no-cache-dir --no-deps --wheel-dir /usr/src/app/wheels -r /usr/src/app/requirements/production.txt
13
14 # final
15 FROM python:3.8
16 RUN mkdir -p /home/app
17 RUN groupadd -g 1000 appuser && useradd -u 1000 -g appuser -s /bin/sh appuser
18 RUN mkdir /home/app
19 RUN mkdir /home/app/web/static
20 WORKDIR /home/app/web
21
22 RUN apt-get update && apt-get -y install netcat
23 COPY --from=builder /usr/src/app/wheels /wheels
24 COPY --from=builder /usr/src/app/requirements/production.txt .
25 RUN pip install --upgrade pip
26 RUN pip install --no-cache /wheels/*
27 COPY ./entrypoint.prod.sh /home/app/web
28 COPY . /home/app/web
29 RUN chown -R appuser:appuser /home/app/web
30 USER appuser
31 RUN ["chmod", "+x", "/home/app/web/entrypoint.prod.sh"]
```

## nginx.conf

```
1 upstream web {
2   server backend:8000;
3 }
4
5 server {
6   listen 80;
7
8   root /usr/share/nginx/html;
9   index index.html index.htm;
10
11   # frontend
12   location / {
13     try_files $uri $uri/ /index.html;
14   }
15   location @rewrites {
16     rewrite ^(.+)$ /index.html last;
17   }
18
19   # backend
20   location ~ ^/(admin|api) {
21     proxy_pass http://web;
22     proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
23     proxy_set_header Host $host;
24     proxy_redirect off;
25   }
26   location /static/ {
27     alias /home/app/web/static/;
28   }
29   error_page 500 502 503 504 /50x.html;
30   location = /50x.html {
31     root /usr/share/nginx/html;
32   }
33
34 }
```