

OpenProject

Port 8080 hatte ich probiert zu ändern, hat aber beim ersten Versuch nicht geklappt, daher ist es jetzt dabei geblieben. Wenn man einen Unifi Controller in Docker hat muss man allerdings zwangsweise auf einen anderen Port ausweichen, weil Unifi den 8080er braucht.

OPENPROJECT_HOST_NAME und ggf. Postgres-Passwort abändern!

```
version: "3.7"

networks:
  frontend:
  backend:

volumes:
  pgdata:
  opdata:

x-op-restart-policy: &restart_policy
  restart: unless-stopped
x-op-image: &image
  image: openproject/community:${TAG:-13}
x-op-app: &app
  <<: [*image, *restart_policy]
environment:
  OPENPROJECT_HTTPS: "${OPENPROJECT_HTTPS:-true}"
  OPENPROJECT_HOST_NAME: "${OPENPROJECT_HOST_NAME:-openproject.MEINEDOMAIN.DE}"
  OPENPROJECT_HSTS: "${OPENPROJECT_HSTS:-true}"
  RAILS_CACHE_STORE: "memcache"
  OPENPROJECT_CACHE__MEMCACHE__SERVER: "cache:11211"
  OPENPROJECT_RAILS__RELATIVE__URL__ROOT: "${OPENPROJECT_RAILS__RELATIVE__URL__ROOT:-}"
  DATABASE_URL: "${DATABASE_URL:-
postgres://postgres:p4ssw0rd@db/openproject?pool=20&encoding=unicode&reconnect=true}"
  RAILS_MIN_THREADS: ${RAILS_MIN_THREADS:-4}
  RAILS_MAX_THREADS: ${RAILS_MAX_THREADS:-16}
  # set to true to enable the email receiving feature. See ./docker/cron for more options
  IMAP_ENABLED: "${IMAP_ENABLED:-false}"
```

volumes:

- "\${OPDATA:-opdata}:/var/openproject/assets"

services:

db:

image: postgres:13

<<: *restart_policy

stop_grace_period: "3s"

volumes:

- "\${PGDATA:-pgdata}:/var/lib/postgresql/data"

environment:

POSTGRES_PASSWORD: \${POSTGRES_PASSWORD:-p4ssw0rd}

POSTGRES_DB: openproject

networks:

- backend

cache:

image: memcached

<<: *restart_policy

networks:

- backend

proxy:

<<: [*image, *restart_policy]

command: "./docker/prod/proxy"

ports:

- "\${PORT:-8080}:80"

environment:

APP_HOST: web

OPENPROJECT_RAILS__RELATIVE__URL__ROOT: "\${OPENPROJECT_RAILS__RELATIVE__URL__ROOT:-}"

depends_on:

- web

networks:

- frontend

web:

<<: *app

command: "./docker/prod/web"

networks:

- frontend

```
- backend
depends_on:
  - db
  - cache
  - seeder
labels:
  - autoheal=true
healthcheck:
  test: ["CMD", "curl", "-f",
"http://localhost:8080${OPENPROJECT_RAILS_RELATIVE_URL_ROOT:-}/health_checks/default"]
  interval: 10s
  timeout: 3s
  retries: 3
  start_period: 30s

autoheal:
  image: willfarrell/autoheal:1.2.0
  volumes:
    - "/var/run/docker.sock:/var/run/docker.sock"
  environment:
    AUTOHEAL_CONTAINER_LABEL: autoheal
    AUTOHEAL_START_PERIOD: 600
    AUTOHEAL_INTERVAL: 30

worker:
  <<: *app
  command: "./docker/prod/worker"
  networks:
    - backend
  depends_on:
    - db
    - cache
    - seeder

cron:
  <<: *app
  command: "./docker/prod/cron"
  networks:
    - backend
  depends_on:
```

- db
- cache
- seeder

seeder:

<<: *app

command: "./docker/prod/seeder"

restart: on-failure

networks:

- backend

danach den NGINX einfach auf <https://docker-host-ip:8080> umleiten:

Edit Proxy Host

Details Custom locations SSL Advanced

Domain Names *

openproject.MEINEDOMAIN.DE

| Scheme * | Forward Hostname / IP * | Forward Port * |
|----------|-------------------------|----------------|
| http | 188.....118 | 8080 |

Cache Assets Block Common Exploits

Websockets Support

Access List

Publicly Accessible

Cancel Save

man muss ein bisschen rumspielen mit der Zeile bei den environment-variablen

```
OPENPROJECT_HTTPS: "${OPENPROJECT_HTTPS:-true}"
```

Ich hatte es zuerst auf false, dann den nginx eingerichtet, dann kam auf der OpenProject GUI die Meldung, dass die Variable wieder auf true gesetzt werden muss. Danach klappt es, obwohl NGINX noch auf http verweist. Keine Ahnung was da los ist. Vielleicht klappt es auch direkt wenn die

Variable auf true ist und der HOST_NAME in der nächsten Zeile direkt richtig ist. Das hatte ich zu Beginn vergessen.

Revision #1

Created 2023-12-06 08:15:26 UTC

Updated 2024-06-15 15:07:40 UTC