

Docker

Criando um container da aplicação NEXT.js

Dockerfile

Cheque a configuração do arquivo `Dockerfile`, a versão contida no repositório foi gerada a partir do projeto `nextjs-with-docker`

ATENÇÃO As variáveis de ambiente `ENV` definidas na `Dockerfile`, principalmente `PB_URL` essa variável define a URL do backend

Criando a imagem

A imagem oficial desse projeto pode ser encontrada em https://hub.docker.com/r/zrafaf/educa_utf_nextjs

Rode `docker build -t educa_utf_nextjs .` na root do projeto.

Dando *push* em uma imagem para o `docker hub`

1. Faça login no docker usando `docker login`
2. Crie uma nova tag para sua imagem com `docker tag educa_utf_nextjs [username]/educa_utf_nextjs:[version]`
3. Faça o push com `docker push [username]/educa_utf_nextjs:[version]`

Fazendo `push` de uma imagem como `latest`

1. `docker tag educa_utf_nextjs zrafaf/educa_utf_nextjs`
2. `docker push zrafaf/educa_utf_nextjs`

A qualquer momento você pode encontrar suas imagens através de `docker images`

Iniciando os Containers

Variáveis de ambiente

O docker compose faz uso das seguintes variáveis de ambiente:

- `UMAMI_APP_SECRET`
- `UTF_AUTH_TOKEN`

Esses **segredos** devem ser setados dentro de um arquivo `.env` no mesmo diretório do `docker-compose.yml`.

Exemplo de `.env`

Bash

```
UMAMI_APP_SECRET=meusegredoumami
UTF_AUTH_TOKEN=meutokendeautenticacao
```

Docker compose

Para iniciar os containers você pode utilizar o seguinte docker-compose

YAML

```
version: '3.8'

services:
  next:
    depends_on:
      - pocketbase
    image: zrafaf/educa_utf_nextjs:latest
    restart: unless-stopped
    ports:
      - 3000:3000

  pocketbase:
    image: zrafaf/educa_utf_pocketbase:latest
    restart: unless-stopped
    ports:
      - 8090:8090
    volumes:
      - ./pocketbase_data:/pb/pb_data
    environment:
      UTF_AUTH_TOKEN: '${UTF_AUTH_TOKEN}'

  watchtower:
    image: containrrr/watchtower
    restart: unless-stopped
    volumes:
      - /var/run/docker.sock:/var/run/docker.sock
    command: --interval 30
```

```

nginx_proxy:
  image: jc21/nginx-proxy-manager:latest
  restart: unless-stopped
  ports:
    - '80:80'
    - '81:81'
    - '443:443'
  volumes:
    - ./nginx_proxy/data:/data
    - ./nginx_proxy/letsencrypt:/etc/letsencrypt

dozzle:
  container_name: dozzle
  image: amir20/dozzle:latest
  labels:
    - 'com.centurylinklabs.watchtower.enable=false' # Disables watchtower auto
update
  restart: unless-stopped
  volumes:
    - /var/run/docker.sock:/var/run/docker.sock
  ports:
    - 9999:8080

# Runs on port 61208
glances:
  image: nicolargo/glances:latest-full
  pid: host
  network_mode: host
  restart: unless-stopped
  volumes:
    - /var/run/docker.sock:/var/run/docker.sock
    - /run/user/1000/podman/podman.sock:/run/user/1000/podman/podman.sock
  environment:
    - 'GLANCES_OPT=-w'

umami:
  image: zrafa/educa_utf_umami:latest
  ports:
    - '3100:3000'
  environment:
    DATABASE_URL: postgresql://umami:umami@umami_db:5432/umami
    DATABASE_TYPE: postgresql
    APP_SECRET: '${UMAMI_APP_SECRET}'
  depends_on:
    umami_db:
      condition: service_healthy
  restart: unless-stopped

umami_db:
  image: postgres:15-alpine
  environment:
    POSTGRES_DB: umami
    POSTGRES_USER: umami
    POSTGRES_PASSWORD: umami
  volumes:
    - ./umami-db-data:/var/lib/postgresql/data
  restart: unless-stopped

```

```
healthcheck:
  test:
    [
      'CMD-SHELL',
      'pg_isready -U ${POSTGRES_USER} -d ${POSTGRES_DB}',
    ]
  interval: 5s
  timeout: 5s
  retries: 5
```

Você pode iniciar os containers com `docker compose up` .

Info

Se algo der errado você pode tentar criar o volume manualmente com:

Bash

```
mkdir -p ~/pocketbase-data
chown -R $USER:$USER ~/pocketbase-data
```

Você também pode checar se o usuário tem permissão de leitura e escrita com

Bash

```
chmod -R 755 ~/pocketbase-data
```

Atualizando a imagem `educa_utf_nextjs`

Para atualizar para a ultima versão disponivel será necessário **apagar** a imagem defasada.

Para ver as imagens atuais execute `docker images` . Você deve ver algo parecido com:

Bash

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
zrafaf/educa_utf_nextjs	latest	ffb4489371c3	15 minutes ago	818MB

Para deletar a imagem defasada **FORÇADO** você pode executar: `docker rmi [id] -f` .

No caso da imagem a cima seria necessário executar `docker rmi ffb4489371c3 -f`

Última atualização: 4 de março de 2024

Criado em: 4 de março de 2024