

Projekt DevOps

Ten dokument zawiera dokumentację, która opiera się na tworzeniu obrazu dockera i uruchamiania go w formacie github.dev. Zawierać ona będzie drobne opisy działania oraz zrzuty ekranu.

1. Budowa Dockera

Aplikacja została skonfigurowana w pliku Dockerfile. Proces budowy obrazu został zrealizowany za pomocą polecenia:

```
docker build -t flask-app .
```

Poniżej znajduje się zrzut ekranu z budowy obrazu:

```
total reclaimed space: 11.85MB
@MaxFreezmaker → /workspaces/ProjektDevops (main) $ docker build -t flask-app .
[+] Building 7.9s (10/10) FINISHED
-> [internal] load build definition from Dockerfile
-> => transferring Dockerfile: 186B
-> [internal] load metadata for docker.io/library/python:3.9-slim
-> [internal] load .dockerignore
-> => transferring context: 2B
-> [1/5] FROM docker.io/library/python:3.9-slim@sha256:caaf1af9e23adc6149e5d20662b267ead9505868ff07c7673dc4a7166951cfea
-> => resolve docker.io/library/python:3.9-slim@sha256:caaf1af9e23adc6149e5d20662b267ead9505868ff07c7673dc4a7166951cfea
-> => sha256:caaf1af9e23adc6149e5d20662b267ead9505868ff07c7673dc4a7166951cfea 10.41kB / 10.41kB
-> => sha256:467c454a5863379d6dc810bdfbc963e877c1a887a11ddddd2fca9702d2dfdf27fa 1.75kB / 1.75kB
-> => sha256:47e5116b5ec1de854ad39b1868d3ea37555ff2ed7edb8dccc618db2427d490ae 5.28kB / 5.28kB
-> => sha256:fd674058ff8f8cfa7fb8a20c006fc0128541cbbad7f7f7f28df570d08f9e4d92 28.23MB / 28.23MB
-> => sha256:96df0e5e81799ba220e250fc3d2c1da017b41302df5954c705bec1407dcab03 3.32MB / 3.32MB
-> => sha256:75a2bc32319e3d6a5bc12888b074f86338a9e3a4304d99d1dddc07155b8ba76e 14.93MB / 14.93MB
-> => sha256:d381bf0bf6de4697fb23da0e7d50d080411236ffcd2853432e12fd0f6b372778 248B / 248B
-> => extracting sha256:fd674058ff8f8cfa7fb8a20c006fc0128541cbbad7f7f7f28df570d08f9e4d92
-> => extracting sha256:96df0e5e81799ba220e250fc3d2c1da017b41302df5954c705bec1407dcab03
-> => extracting sha256:75a2bc32319e3d6a5bc12888b074f86338a9e3a4304d99d1dddc07155b8ba76e
-> => extracting sha256:d381bf0bf6de4697fb23da0e7d50d080411236ffcd2853432e12fd0f6b372778
-> [internal] load build context
-> => transferring context: 38.16kB
-> [2/5] WORKDIR /app
-> [3/5] COPY requirements.txt requirements.txt
-> [4/5] RUN pip install -r requirements.txt
-> [5/5] COPY . .
-> exporting to image
-> => exporting layers
-> => writing image sha256:1a99aed6eb229e65da2b945590ff6a054d3d8988ec0c9cf290f7950f7a63dcdd
-> => naming to docker.io/library/flask-app
@MaxFreezmaker → /workspaces/ProjektDevops (main) $ docker run -p 5000:5000 flask-app
* Serving Flask app 'app' (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: off
* Running on all addresses.
  WARNING: This is a development server. Do not use it in a production deployment.
* Running on http://172.17.0.2:5000/ (Press CTRL+C to quit)
172.17.0.1 - - [10/Jan/2025 17:30:27] "GET / HTTP/1.1" 200 -
172.17.0.1 - - [10/Jan/2025 17:30:28] "GET /favicon.ico HTTP/1.1" 404 -
```

2. Uruchomienie aplikacji webowej.

Aplikacja została uruchomiona w kontenerze Dockera za pomocą polecenia:

```
docker run -p 5000:5000 flask-app
```

Poniżej znajduje się zrzut ekranu działającej aplikacji w przeglądarce:



3. Wykonanie Pull Requesta wraz ze scaleniem plików.

Gałąź feature/change-content została utworzona w celu wprowadzenia zmian, w tym dodania pliku ci.yml. Po zakończeniu prac gałąź została scalona z main. Poniżej znajduje się zrzut ekranu przedstawiający Pull Request:

Wprowadzenie Ci.yml i wykonanie testu kompilacji #1 Edit <> Code

Merged MaxAfreetmaker merged 2 commits into `main` from `feature/change-content` now

Conversation 0 Commits 2 Checks 0 Files changed 2 +31 -1

MaxAfreetmaker commented now Owner ...

kompiowanie

MaxAfreetmaker added 2 commits 9 minutes ago

- Dodanie Werkzeug 25245fd
- Wprowadzenie ci.yml e2c1ade

MaxAfreetmaker merged commit `bd2edde` into `main` now 1 check was pending View details Revert

Pull request successfully merged and closed Delete branch

You're all set—the `feature/change-con...` branch can be safely deleted.

Try the new merge experience

Add a comment

Write Preview H B I ≡ <> 🔗 📌 📋 📄 📝 📎 📧 📧 📧 📧

Add your comment here...

Markdown is supported Paste, drop, or click to add files

Comment

Remember, contributions to this repository should follow our [GitHub Community Guidelines](#).

ProTip! Add `.patch` or `.diff` to the end of URLs for Git's plaintext views.

© 2025 GitHub, Inc. Terms Privacy Security Status Docs Contact Manage cookies Do not share my personal information

Reviewers No reviews

Assignees No one—assign yourself

Labels None yet

Projects None yet

Milestone No milestone

Development Successfully merging this pull request may close these issues.

Notifications Customize Unsubscribe

You're receiving notifications because you're watching this repository.

1 participant

Lock conversation

4. Python i systemy wsparcia.

Python to uniwersalny język programowania wysokiego poziomu, który jest szeroko stosowany w różnych dziedzinach, takich jak tworzenie aplikacji webowych, analiza danych, czy automatyzacja procesów. W naszym projekcie wykorzystaliśmy framework Flask, który umożliwia szybkie i łatwe tworzenie aplikacji webowych.

Plik requirements.txt został użyty do zdefiniowania zależności wymaganych do działania aplikacji. W naszym przypadku zawiera następujące biblioteki:

Flask==2.0.3

Werkzeug==2.0.3

Wyjaśnienie zawartości:

Flask: Główny framework używany do stworzenia aplikacji webowej.

Werkzeug: Narzędzie wspierające Flask, odpowiedzialne za routing i debugowanie.

Instalacja zależności:

Aby zainstalować wymagane biblioteki w środowisku Python, użyto polecenia:

pip install -r requirements.txt

To polecenie instaluje wszystkie pakiety wskazane w pliku requirements.txt, zapewniając zgodność wersji używanych bibliotek z konfiguracją projektu.

5. Repozytorium

Twórca : Maksymilian Adamski

Kod studenta : 14405

Git Repo : <https://github.com/MaxAfreezmaker/ProjektDevops>