



POLITECHNIKA
LUBELSKA



POLITECHNIKA
LUBELSKA
WYDZIAŁ ELEKTROTECHNIKI
I INFORMATYKI

Programowanie Full-Stack w Chmurze Obliczeniowej

Sprawozdanie 1

Sebastian Słupny

Prowadzący

Sławomir Przyłucki

1. (max. 20%)

Proszę napisać program serwera (dowolny język programowania), który realizować będzie następującą funkcjonalność:

```
server.py > ...
1  import socket
2  from datetime import datetime
3  from flask import Flask, request
4
5  # Konfiguracja serwera
6  HOST = '0.0.0.0' # Serwer nasłuchuje na wszystkich interfejsach sieciowych
7  PORT = 8080 # Port, na którym serwer nasłuchuje
8
9  app = Flask(__name__)
10
11 def log_info():
12     # Pobranie informacji o dacie i czasie uruchomienia serwera
13     current_time = datetime.now().strftime('%Y-%m-%d %H:%M:%S')
14
15     # Pobranie adresu IP klienta
16     client_address = request.remote_addr
17
18     # Wyświetlanie informacji w logach
19     print(f"Serwer uruchomiony: {current_time}")
20     print(f"Autor serwera: Sebastian Slupny")
21     print(f"Port nasłuchiwania: {PORT}")
22     print(f"Adres IP klienta: {client_address}")
23
24 @app.route('/')
25 def hello():
26     # Logowanie informacji o kliencie
27     log_info()
28
29     # Przygotowanie odpowiedzi dla klienta
30     response = f"Adres IP klienta: {request.remote_addr}\n"
31     response += f>Data i godzina w strefie czasowej : {datetime.now()}\n"
32
33     return response
34
35 if __name__ == '__main__':
36     app.run(host=HOST, port=PORT)
37
```

Kod server.py

2. (max. 50%)

Opracować plik Dockerfile...

```
Dockerfile > ...
1  # Autor: Sebastian Slupny"
2  # Etap budowania
3  FROM python:3.9-slim as builder
4
5  # Skopiowanie plików projektu
6  COPY server.py .
7
8  # Instalacja zależności
9  RUN pip install flask
10
11 # Etap finalny
12 FROM python:3.9-slim
13
14 # Skopiowanie pliku serwera
15 COPY --from=builder server.py /app/server.py
16
17 # Skopiowanie zależności
18 COPY --from=builder /usr/local/lib/python3.9/site-packages /usr/local/lib/python3.9/site-packages
19
20 # Ustalenie zmiennej środowiskowej
21 ENV PORT=8080
22
23 # Uruchomienie serwera
24 CMD ["python", "/app/server.py"]
```

Plik Dockerfile

3. (max. 30%)

Należy podać polecenia niezbędne do:

a. zbudowania opracowanego obrazu kontenera

docker build -t pwcho_zad1:tag -f Dockerfile .

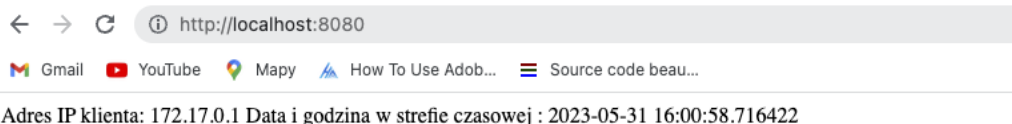
Polecenie docker build buduje obraz na podstawie pliku Dockerfile znajdującego się w bieżącym katalogu

```
sebastian_slupny@macbook-pro-sebastian PWCHO % docker build -t pwcho_zad1:tag -f Dockerfile .
[+] Building 6.4s (11/11) FINISHED
=> [internal] load build definition from Dockerfile                                0.0s
=> => transferring dockerfile: 37B                                              0.0s
=> [internal] load .dockerignore                                                0.0s
=> => transferring context: 2B                                                  0.0s
=> [internal] load metadata for docker.io/library/python:3.9-slim              6.3s
=> [auth] library/python:pull token for registry-1.docker.io                  0.0s
=> [internal] load build context                                              0.0s
=> => transferring context: 31B                                                0.0s
=> [stage-1 1/3] FROM docker.io/library/python:3.9-slim@sha256:5cde4e147c4165ad8dbf8a4df9631863766eeb0b79b890fafe6885b3b127af74 0.0s
=> CACHED [builder 2/3] COPY server.py .                                       0.0s
=> CACHED [builder 3/3] RUN pip install flask                                  0.0s
=> CACHED [stage-1 2/3] COPY --from=builder server.py /app/server.py           0.0s
=> CACHED [stage-1 3/3] COPY --from=builder /usr/local/lib/python3.9/site-packages /usr/local/lib/python3.9/site-packages 0.0s
=> exporting to image                                                         0.0s
=> => exporting layers                                                         0.0s
=> => writing image sha256:ef5a67324a3116481fc5e8f6130a845c3b0ce200e04059c3801a23b21ee01b77 0.0s
=> => naming to docker.io/library/pwcho_zad1:tag                             0.0s
sebastian_slupny@macbook-pro-sebastian PWCHO %
```

b. uruchomienia kontenera na podstawie zbudowanego obrazu,

docker run -p 8080:8080 pwcho_zad1:tag

8080 numer portu, na którym udostępniany jest kontener na hoście. 8080 numer portu, na którym serwer wewnątrz kontenera nasłuchuje. Polecenie docker run uruchamia kontener na podstawie określonego obrazu.



```
sebastian_slupny@macbook-pro-sebastian PWCHO % docker run -p 8080:8080 pwcho_zad1:tag
* Serving Flask app 'server'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:8080
* Running on http://172.17.0.2:8080
Press CTRL+C to quit
172.17.0.1 - - [31/May/2023 16:00:58] "GET / HTTP/1.1" 200 -
172.17.0.1 - - [31/May/2023 16:00:59] "GET /favicon.ico HTTP/1.1" 404 -
172.17.0.1 - - [31/May/2023 16:01:19] "GET / HTTP/1.1" 200 -
172.17.0.1 - - [31/May/2023 16:01:22] "GET / HTTP/1.1" 200 -
172.17.0.1 - - [31/May/2023 16:01:23] "GET / HTTP/1.1" 200 -
172.17.0.1 - - [31/May/2023 16:01:25] "GET / HTTP/1.1" 200 -
^CSerwer uruchomiony: 2023-05-31 16:00:58
Autor serwera: Sebastian Slupny
Port nasłuchiwania: 8080
Adres IP klienta: 172.17.0.1
Serwer uruchomiony: 2023-05-31 16:01:19
Autor serwera: Sebastian Slupny
Port nasłuchiwania: 8080
Adres IP klienta: 172.17.0.1
Serwer uruchomiony: 2023-05-31 16:01:22
Autor serwera: Sebastian Slupny
Port nasłuchiwania: 8080
Adres IP klienta: 172.17.0.1
Serwer uruchomiony: 2023-05-31 16:01:23
Autor serwera: Sebastian Slupny
Port nasłuchiwania: 8080
Adres IP klienta: 172.17.0.1
Serwer uruchomiony: 2023-05-31 16:01:25
Autor serwera: Sebastian Slupny
Port nasłuchiwania: 8080
Adres IP klienta: 172.17.0.1
sebastian_slupny@macbook-pro-sebastian PWCHO %
```

- c. sposobu uzyskania informacji, które wygenerował serwer w trakcie uruchamiania kontenera (patrz: punkt 1a),

Można uzyskać informacje generowane przez serwer w trakcie uruchamiania kontenera, przekierowując logi kontenera na standardowe wyjście. Można to zrobić, dodając flagę `-t` (pseudo-TTY) i `-i` (interaktywność) do polecenia `docker run`

`docker run -t -i -p 8080:8080 pwcho_zad1:tag`

Dzięki temu logi serwera będą widoczne na bieżąco w konsoli, w której uruchomiono kontener.

```
sebastian_slupny@macbook-pro-sebastian PWCHO % docker run -t -i -p 8080:8080 pwcho_zad1:tag
* Serving Flask app 'server'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:8080
* Running on http://172.17.0.2:8080
Press CTRL+C to quit
Serwer uruchomiony: 2023-05-31 16:04:22
Autor serwera: Sebastian Slupny
Port nasłuchiwania: 8080
Adres IP klienta: 172.17.0.1
172.17.0.1 - - [31/May/2023 16:04:22] "GET / HTTP/1.1" 200 -
Serwer uruchomiony: 2023-05-31 16:04:26
Autor serwera: Sebastian Slupny
Port nasłuchiwania: 8080
Adres IP klienta: 172.17.0.1
172.17.0.1 - - [31/May/2023 16:04:26] "GET / HTTP/1.1" 200 -
□
```

- d. sprawdzenia, ile warstw posiada zbudowany obraz.

Polecenie do sprawdzenia, ile warstw posiada zbudowany obraz:

`docker history pwcho_zad1:tag`

```
sebastian_slupny@macbook-pro-sebastian PWCHO % docker history pwcho_zad1:tag
^C
sebastian_slupny@macbook-pro-sebastian PWCHO % docker history pwcho_zad1:tag
IMAGE          CREATED        CREATED BY                                      SIZE      COMMENT
ef5a67324a31   27 minutes ago CMD ["python" "/app/server.py"]              0B         buildkit.dockerfile.v0
<missing>      27 minutes ago ENV PORT=8080                                  0B         buildkit.dockerfile.v0
<missing>      27 minutes ago COPY /usr/local/lib/python3.9/site-packages ... 17.5MB     buildkit.dockerfile.v0
<missing>      27 minutes ago COPY server.py /app/server.py # buildkit      1.04kB     buildkit.dockerfile.v0
<missing>      8 days ago     CMD ["python3"]                               0B         buildkit.dockerfile.v0
<missing>      8 days ago     RUN /bin/sh -c set -eux; savedAptMark="$(a... 11.5MB     buildkit.dockerfile.v0
<missing>      8 days ago     ENV PYTHON_GET_PIP_SHA256=96461deced5c2a487d... 0B         buildkit.dockerfile.v0
<missing>      8 days ago     ENV PYTHON_GET_PIP_URL=https://github.com/py... 0B         buildkit.dockerfile.v0
<missing>      8 days ago     ENV PYTHON_SETUPTOOLS_VERSION=58.1.0          0B         buildkit.dockerfile.v0
<missing>      8 days ago     ENV PYTHON_PIP_VERSION=22.0.4                 0B         buildkit.dockerfile.v0
<missing>      8 days ago     RUN /bin/sh -c set -eux; for src in idle3 p... 32B        buildkit.dockerfile.v0
<missing>      8 days ago     RUN /bin/sh -c set -eux; savedAptMark="$(a... 29MB       buildkit.dockerfile.v0
<missing>      8 days ago     ENV PYTHON_VERSION=3.9.16                     0B         buildkit.dockerfile.v0
<missing>      8 days ago     ENV GPG_KEY=E3FF2839C048B25C084DEBE9B26995E3... 0B         buildkit.dockerfile.v0
<missing>      8 days ago     RUN /bin/sh -c set -eux; apt-get update; a... 3.11MB     buildkit.dockerfile.v0
<missing>      8 days ago     ENV LANG=C.UTF-8                               0B         buildkit.dockerfile.v0
<missing>      8 days ago     ENV PATH=/usr/local/bin:/usr/local/sbin:/usr... 0B         buildkit.dockerfile.v0
<missing>      8 days ago     /bin/sh -c #(nop) CMD ["bash"]                0B         buildkit.dockerfile.v0
<missing>      8 days ago     /bin/sh -c #(nop) ADD file:88252a7f118b4d6f5... 80.5MB     buildkit.dockerfile.v0
sebastian_slupny@macbook-pro-sebastian PWCHO %
```

Repozytorium GitHub: https://github.com/sebastianslupny4/pwcho_zad1/

Repozytorium DockerHub: https://hub.docker.com/r/sebastianslupny/pwcho_zad1