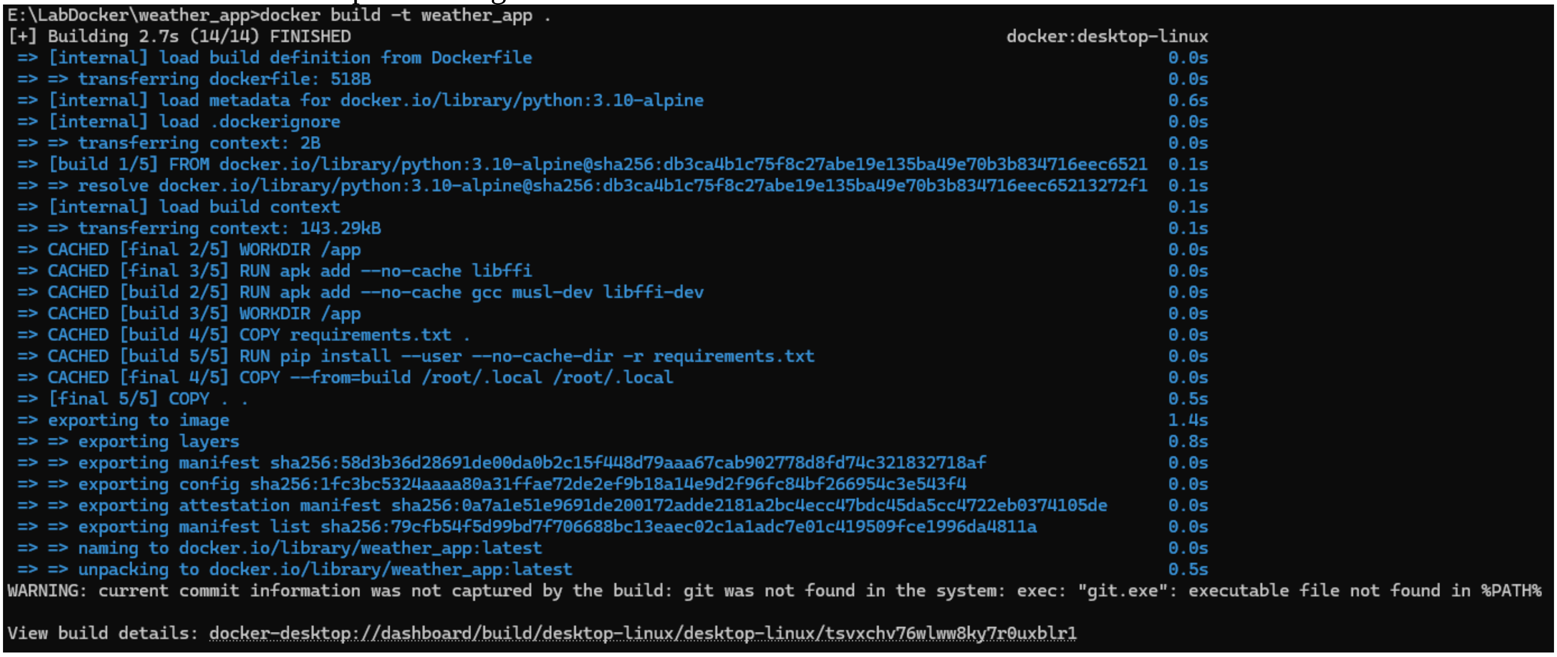
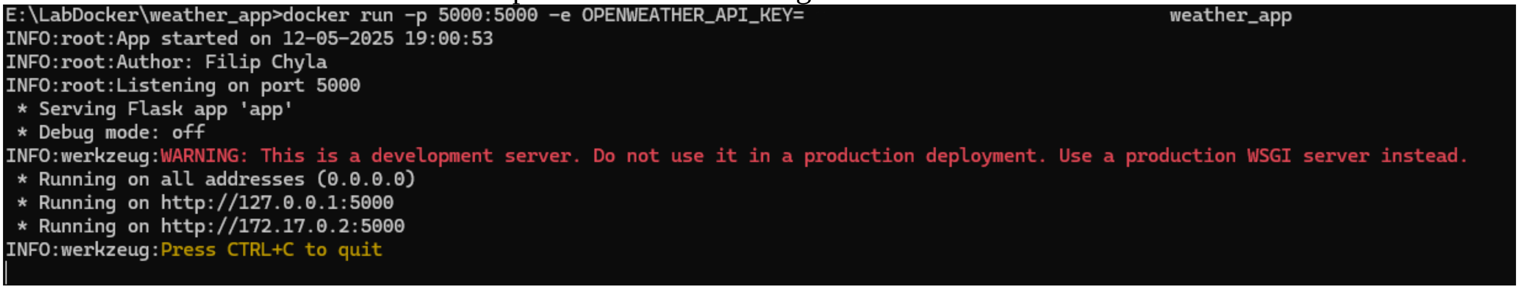
1. Zbudowanie opracowanego obrazu kontenera:



```
E:\LabDocker\weather_app>docker build -t weather_app .
[+] Building 2.7s (14/14) FINISHED                                                                    docker:desktop-linux
 => [internal] load build definition from Dockerfile                                                                 0.0s
 => => transferring dockerfile: 518B                                                                                 0.0s
 => [internal] load metadata for docker.io/library/python:3.10-alpine                                                0.6s
 => [internal] load .dockerignore                                                                                    0.0s
 => => transferring context: 2B                                                                                      0.0s
 => [build 1/5] FROM docker.io/library/python:3.10-alpine@sha256:db3ca4b1c75f8c27abe19e135ba49e70b3b834716eec6521  0.1s
 => => resolve docker.io/library/python:3.10-alpine@sha256:db3ca4b1c75f8c27abe19e135ba49e70b3b834716eec65213272f1  0.1s
 => [internal] load build context                                                                                    0.1s
 => => transferring context: 143.29kB                                                                                0.1s
 => CACHED [final 2/5] WORKDIR /app                                                                                  0.0s
 => CACHED [final 3/5] RUN apk add --no-cache libffi                                                                 0.0s
 => CACHED [build 2/5] RUN apk add --no-cache gcc musl-dev libffi-dev                                                0.0s
 => CACHED [build 3/5] WORKDIR /app                                                                                  0.0s
 => CACHED [build 4/5] COPY requirements.txt .                                                                       0.0s
 => CACHED [build 5/5] RUN pip install --user --no-cache-dir -r requirements.txt                                     0.0s
 => CACHED [final 4/5] COPY --from=build /root/.local /root/.local                                                   0.0s
 => [final 5/5] COPY . .                                                                                             0.5s
 => exporting to image                                                                                               1.4s
 => => exporting layers                                                                                              0.8s
 => => exporting manifest sha256:58d3b36d28691de00da0b2c15f448d79aaa67cab902778d8fd74c321832718af                    0.0s
 => => exporting config sha256:1fc3bc5324aaaa80a31ffae72de2ef9b18a14e9d2f96fc84bf266954c3e543f4                      0.0s
 => => exporting attestation manifest sha256:0a7a1e51e9691de200172adde2181a2bc4ecc47bdc45da5cc4722eb0374105de        0.0s
 => => exporting manifest list sha256:79cfb54f5d99bd7f706688bc13eaec02c1a1adc7e01c419509fce1996da4811a               0.0s
 => => naming to docker.io/library/weather_app:latest                                                                0.0s
 => => unpacking to docker.io/library/weather_app:latest                                                             0.5s
WARNING: current commit information was not captured by the build: git was not found in the system: exec: "git.exe": executable file not found in %PATH%

View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/tsvxchv76wlww8ky7r0uxblr1
```

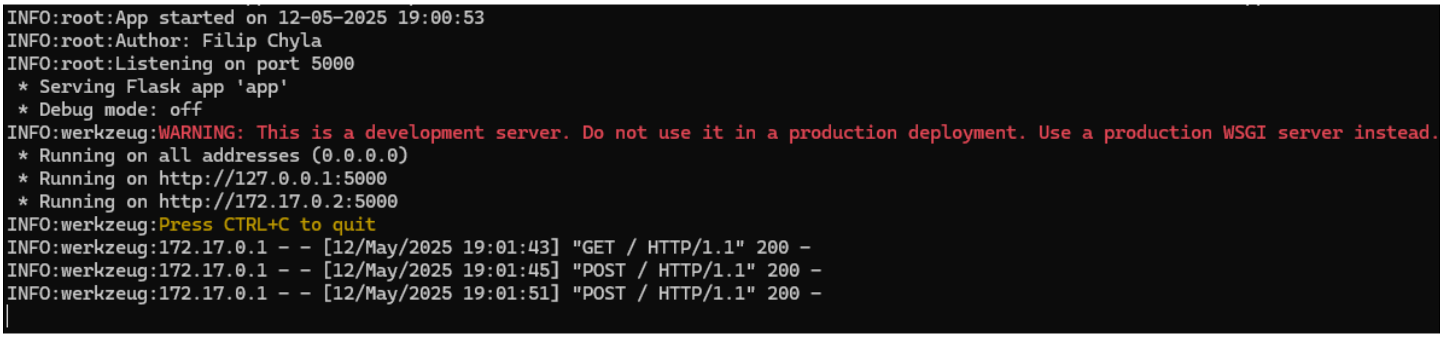
2. Uruchomienie kontenera na podstawie zbudowanego obrazu:



```
E:\LabDocker\weather_app>docker run -p 5000:5000 -e OPENWEATHER_API_KEY=                                  weather_app
INFO:root:App started on 12-05-2025 19:00:53
INFO:root:Author: Filip Chyla
INFO:root:Listening on port 5000
 * Serving Flask app 'app'
 * Debug mode: off
INFO:werkzeug: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://127.0.0.1:5000
 * Running on http://172.17.0.2:5000
INFO:werkzeug:Press CTRL+C to quit
```

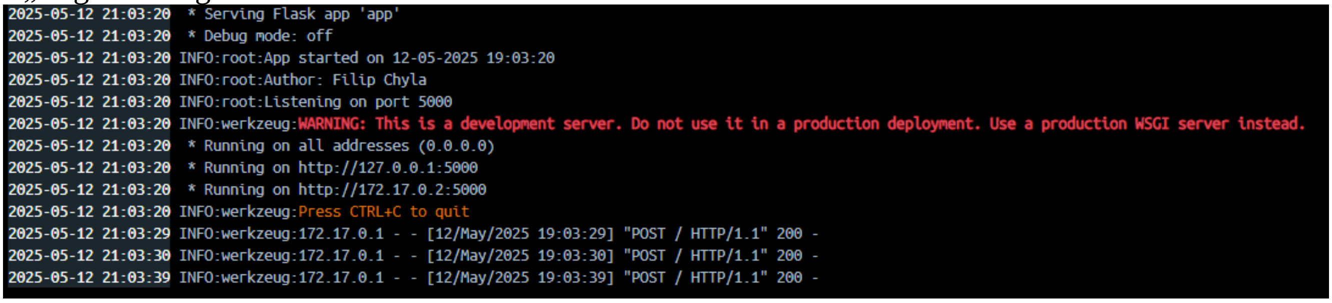
3. Sposób uzyskania informacji z logów, które wygenerowała opracowana aplikacja podczas uruchamiana kontenera

a. Przy użyciu komendy podanej w punkcie 2 widzimy logi w konsoli



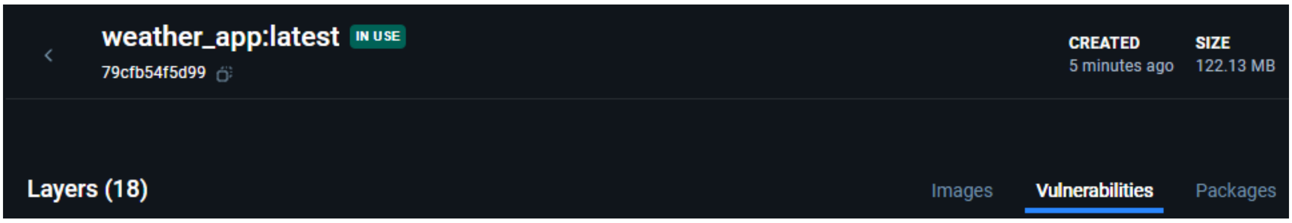
```
INFO:root:App started on 12-05-2025 19:00:53
INFO:root:Author: Filip Chyla
INFO:root:Listening on port 5000
 * Serving Flask app 'app'
 * Debug mode: off
INFO:werkzeug:WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:5000
 * Running on http://172.17.0.2:5000
INFO:werkzeug:Press CTRL+C to quit
INFO:werkzeug:172.17.0.1 - - [12/May/2025 19:01:43] "GET / HTTP/1.1" 200 -
INFO:werkzeug:172.17.0.1 - - [12/May/2025 19:01:45] "POST / HTTP/1.1" 200 -
INFO:werkzeug:172.17.0.1 - - [12/May/2025 19:01:51] "POST / HTTP/1.1" 200 -
```

b. W przypadku użycia opcji -d, możemy śledzić logi w aplikacji docker desktop w zakładce „Logs” naszego kontenera



```
2025-05-12 21:03:20  * Serving Flask app 'app'
2025-05-12 21:03:20  * Debug mode: off
2025-05-12 21:03:20 INFO:root:App started on 12-05-2025 19:03:20
2025-05-12 21:03:20 INFO:root:Author: Filip Chyla
2025-05-12 21:03:20 INFO:root:Listening on port 5000
2025-05-12 21:03:20 INFO:werkzeug:WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
2025-05-12 21:03:20  * Running on all addresses (0.0.0.0)
2025-05-12 21:03:20  * Running on http://127.0.0.1:5000
2025-05-12 21:03:20  * Running on http://172.17.0.2:5000
2025-05-12 21:03:20 INFO:werkzeug:Press CTRL+C to quit
2025-05-12 21:03:29 INFO:werkzeug:172.17.0.1 - - [12/May/2025 19:03:29] "POST / HTTP/1.1" 200 -
2025-05-12 21:03:30 INFO:werkzeug:172.17.0.1 - - [12/May/2025 19:03:30] "POST / HTTP/1.1" 200 -
2025-05-12 21:03:39 INFO:werkzeug:172.17.0.1 - - [12/May/2025 19:03:39] "POST / HTTP/1.1" 200 -
```

d. Sprawdzenie, ile warstw posiada zbudowany obraz oraz jaki ma rozmiar:



weather_app:latest IN USE

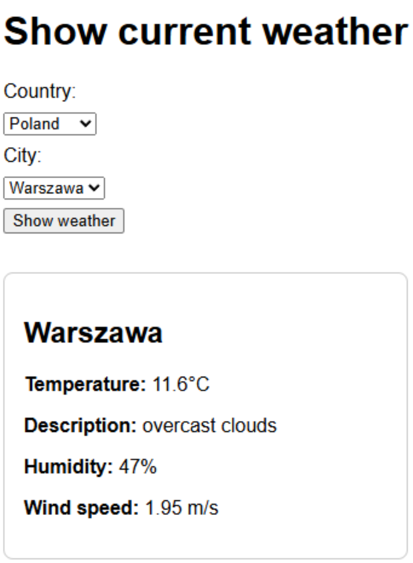
79cfb54f5d99

CREATED 5 minutes ago

SIZE 122.13 MB

Layers (18)

Przykładowe zrzuty ekranu działającej aplikacji:



Show current weather

Country: Poland

City: Warszawa

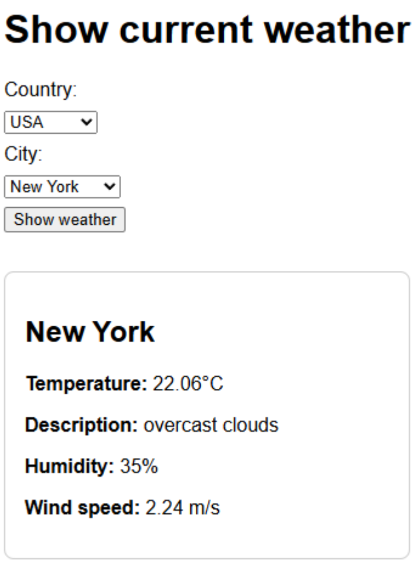
Warszawa

Temperature: 11.6°C

Description: overcast clouds

Humidity: 47%

Wind speed: 1.95 m/s



Show current weather

Country: USA

City: New York

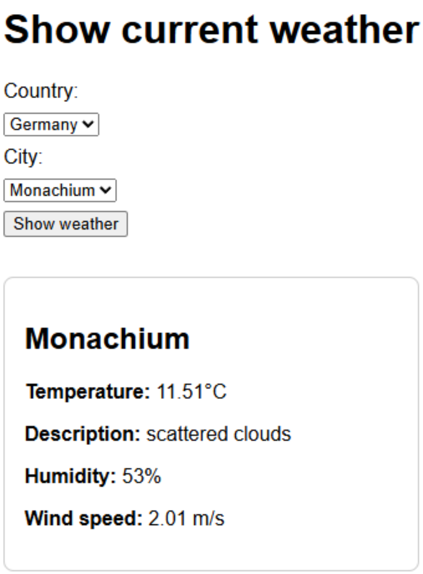
New York

Temperature: 22.06°C

Description: overcast clouds

Humidity: 35%

Wind speed: 2.24 m/s



Show current weather

Country: Germany

City: Monachium

Monachium

Temperature: 11.51°C

Description: scattered clouds

Humidity: 53%

Wind speed: 2.01 m/s

Komentarz:
W razie chęci sprawdzenia działania aplikacji należy stworzyć konto na stronie openweathermap.org – link do rejestracji konta i podczas uruchamiania kontenera(jak w punkcie 2) podać swój klucz API jako zmienną środowiskową OPENWEATHER_API_KEY.