|  |  |  |  |
| --- | --- | --- | --- |
| **Wydział Elektrotechniki i Informatyki Politechniki Lubelskiej** | **Programowanie Aplikacji w Chmurze Obliczeniowej** | | |
| Autor sprawozdania:  Agnieszka Marzęda | Kierunek studiów:  I1S | Grupa  6.9 | Rok akademicki  2024/2025 |
| Temat:  Tworzenie plików Dockerfile wykorzystujących metodę wieloetapowego budowania obrazów | | Data wykonania  31.03.2025 r. | Ocena |

1. **Treść utworzonego pliku Dockerfile**

**Etap 1**

FROM golang:1.21-alpine AS builder

WORKDIR /app

COPY . .

ARG VERSION="1.0.0"

RUN go mod init myapp && go mod tidy && \

    go build -ldflags "-X main.version=$VERSION" -o myapp

FROM alpine:latest

WORKDIR /root/

COPY --from=builder /app/myapp .

EXPOSE 8080

CMD ["./myapp"]

**Etap 2**

FROM golang:1.21-alpine AS builder

WORKDIR /app

COPY . .

ARG VERSION="1.0.0"

RUN go mod init myapp && go mod tidy && \

    go build -ldflags "-X main.version=$VERSION" -o myapp

FROM alpine:latest AS runtime

WORKDIR /root/

COPY --from=builder /app/myapp .

EXPOSE 8080

CMD ["./myapp"]

1. **Użyte polecenie do budowy obrazu i wynik jego działania**

docker build --build-arg VERSION=1.2.3 -t myapp .

Obraz zawierający tekst, zrzut ekranu, Czcionka

Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.

1. **Polecenie uruchamiające serwer**

docker run -p 8080:8080 myapp



1. **Polecenie potwierdzające działanie kontenera i poprawne funkcjonowanie opracowanej aplikacji.**

docker ps

1. **Zrzut ekranu (przeglądarka lub curl), potwierdzające że aplikacja realizuje wymaganą funkcjonalność**

**Obraz zawierający tekst, zrzut ekranu, Czcionka

Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.**

**Obraz zawierający tekst, zrzut ekranu, Czcionka

Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.**

**Wioski:**

To ćwiczenie pokazało, jak efektywnie budować aplikację w Dockerze, używając wieloetapowego procesu. Dzięki temu obraz jest lżejszy i bardziej uporządkowany. Rozdzielenie aplikacji Go i Nginx na osobne etapy sprawiło, że każda część działa niezależnie i można ją łatwiej zarządzać. Dodatkowo HEALTHCHECK pozwala sprawdzić, czy wszystko działa poprawnie. Całość można łatwo uruchomić i testować za pomocą Docker Compose.