

**Sprawozdanie**  
**Laboratorium nr. 5**  
**Programowanie aplikacji w chmurze obliczeniowej**  
**Dawid Rutkowski**

1. Treść utworzonego pliku Dockerfile.

```
1  FROM scratch AS builder
2  ADD alpine-minirootfs-3.19.1-aarch64.tar /
3  ARG VERSION
4  WORKDIR /usr/app
5
6  RUN apk update && apk add nodejs npm
7
8  COPY ./package.json ./
9  RUN npm install
10 COPY ./index.js ./
11
12 EXPOSE 4000
13
14 FROM nginx:alpine
15
16 ARG VERSION
17 ENV APP_VERSION=${VERSION:-1.0.0}
18
19 RUN apk update && apk add nodejs
20 COPY --from=builder /usr/app /usr/share/nginx/html
21 COPY ./default.conf /etc/nginx/conf.d
22 WORKDIR /usr/share/nginx/html
23 EXPOSE 80
24
25 HEALTHCHECK --interval=15s --timeout=5s --start-period=5s --retries=5 \
26 |   CMD curl -f http://localhost:80 || exit 1
27
28 CMD nginx -g "daemon off;" & node index.js
```

## 2. Użyte polecenia do budowy obrazu, jego uruchomienia i wyniki działania.

```
docker build -f Dockerfile -t domowa:v1 --rm .
```

```
docker run -d -p 81:80 --name testDomowa domowa:v1
```

```
docker ps | grep dom
```

```
cz1 docker build -f Dockerfile -t domowa:v1 --rm .

[+] Building 2.7s (16/16) FINISHED                                docker:desktop-linux
=> [internal] load build definition from Dockerfile                0.0s
=> => transferring dockerfile: 675B                                0.0s
=> [internal] load metadata for docker.io/library/nginx:alpine    0.6s
=> [internal] load .dockerignore                                  0.0s
=> => transferring context: 2B                                       0.0s
=> [internal] load build context                                  0.0s
=> => transferring context: 627B                                      0.0s
=> CACHED [stage-1 1/5] FROM docker.io/library/nginx:alpine@sha256:31bad00311cb5eeb8a6648beadcf67277a175da89989f14727420a80e2e767 0.0s
=> [stage-1 2/5] RUN apk update && apk add nodejs                 1.9s
=> CACHED [builder 1/6] ADD alpine-minirootfs-3.19.1-aarch64.tar / 0.0s
=> CACHED [builder 2/6] WORKDIR /usr/app                          0.0s
=> CACHED [builder 3/6] RUN apk update && apk add nodejs npm       0.0s
=> CACHED [builder 4/6] COPY ./package.json ./                    0.0s
=> CACHED [builder 5/6] RUN npm install                           0.0s
=> CACHED [builder 6/6] COPY ./index.js ./                        0.0s
=> [stage-1 3/5] COPY --from=builder /usr/app /usr/share/nginx/html 0.0s
=> [stage-1 4/5] COPY ./default.conf /etc/nginx/conf.d            0.0s
=> [stage-1 5/5] WORKDIR /usr/share/nginx/html                    0.0s
=> exporting to image                                             0.1s
=> => exporting layers                                             0.1s
=> writing image sha256:801f901dd1fb53b63c46d2f0a5bd1daef7d99bc19ff7b2b81fce8cdd44fe961c 0.0s
=> naming to docker.io/library/domowa:v1                          0.0s

View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/3757pdfcpkdji4i91iu2kx6ju

What's Next?
View a summary of image vulnerabilities and recommendations + docker scout quickview

cz1 docker run -d -p 81:80 --name testDomowa domowa:v1

5e7aa7ecbbf74c6e30d4df3f4b9cfd2ec46a70ad328a7feb09f8db6fb07e87ab
```

```
cz1 docker ps | grep dom
5e7aa7ecbbf7 domowa:v1 "/docker-entrypoint..." About a minute ago Up About a minute (healthy) 0.0.0.0:81->80/tcp testDomowa
```

## 3. Potwierdzenie realizacji požądanej funkcjonalności.

```
cz1 curl http://localhost:81

Adres IP serwera: 172.17.0.4
Nazwa serwera (hostname): 5e7aa7ecbbf7
Wersja aplikacji: 1.0.0
```

```
localhost

Adres IP serwera: 172.17.0.4
Nazwa serwera (hostname): 5684f402cf35
Wersja aplikacji: 1.0.0
```