Data 13.05.2020r.	*Grupa 05*
Imię i nazwisko Radosław Niestrój	

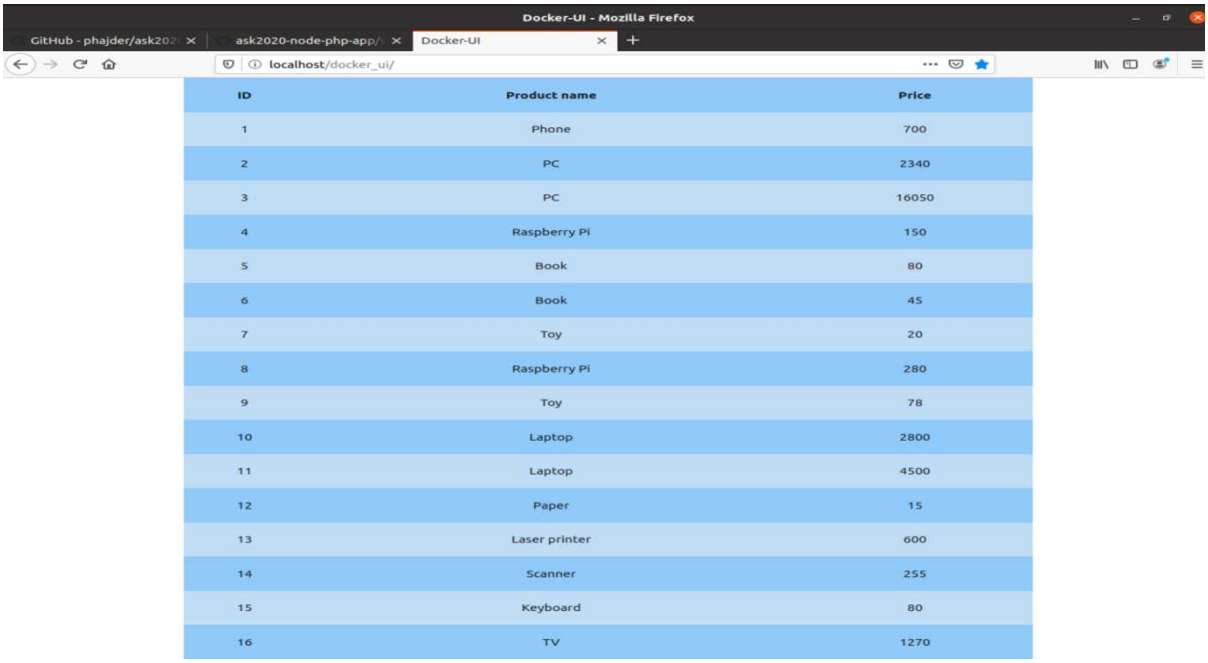
Opis ćwiczenia

Celem ćwiczenia było zapoznanie się z oprogramowaniem Docker i z jego wszechstronnymi zastosowaniami. Zadanie polegało na skonfigurowaniu trzech aplikacji: Mysql jako bazy danych, Node.js jako serwera i Apache jako klienta lokalnego. Pierwszą częścią zadania było skonfigurowanie aplikacji kontrolnej (bez użycia dockera) i wyświetlenie tabeli w postaci strony w przeglądarce. W następnym kroku przechodziło się do głównej części zadania. Jej celem było stworzenie tego samego efektu co w części kontrolnej, tylko przy użyciu dockera. Kolejnym elementem ćwiczenia było połączenie kontenerów i zawarcie ich w sieciach aby móc korzystać z ich nazw, zamiast adresów IP. Ostatnim elementem było utworzenie konfiguracji.

Przebieg ćwiczenia

I część – strona kontrolna, bez użycia dockera

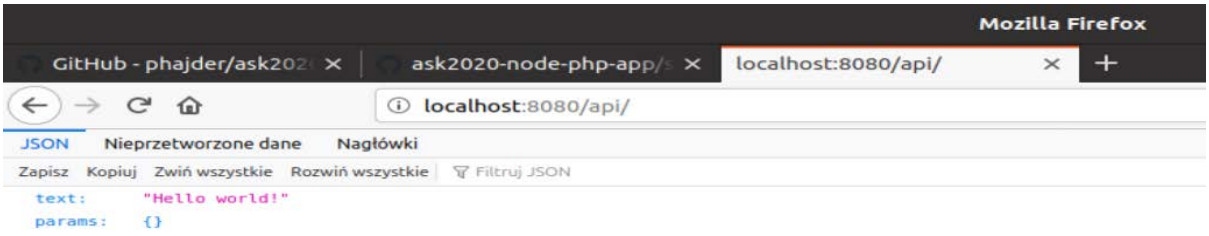
Sprawdzenie działania próbnej wersji programu na serwerze lokalnym Apache i bazie Mysql.



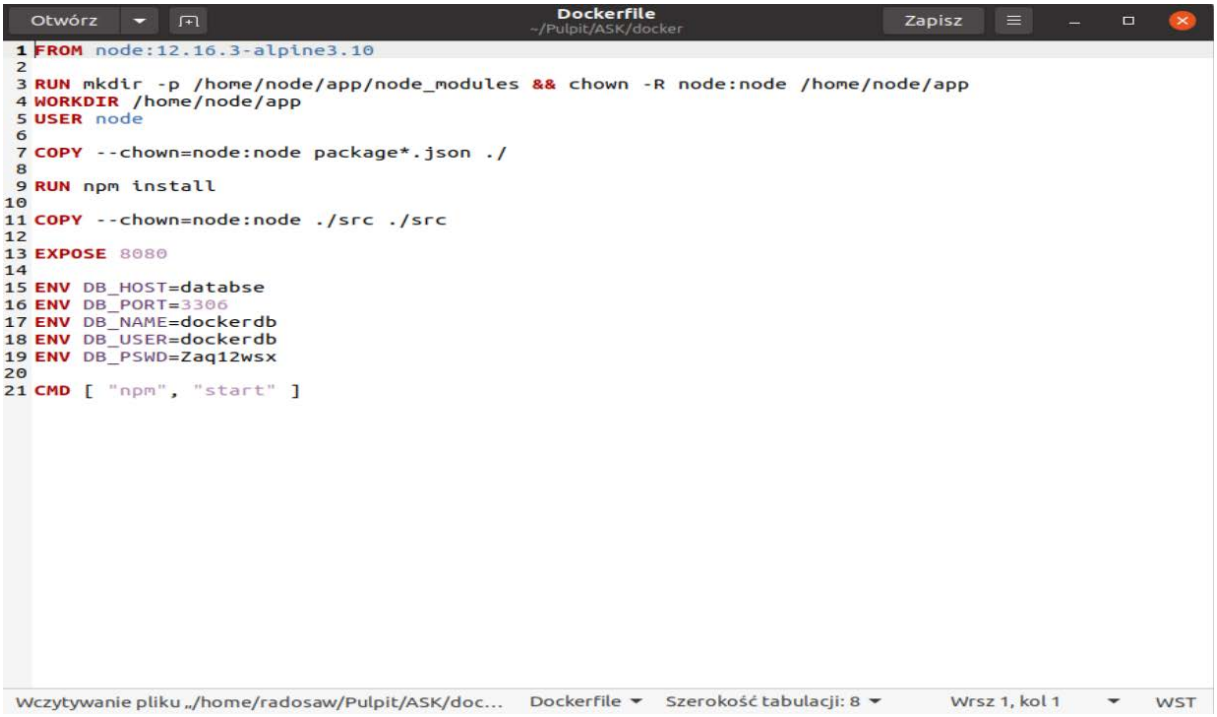
ID	Product name	Price
1	Phone	700
2	PC	2340
3	PC	16050
4	Raspberry Pi	150
5	Book	80
6	Book	45
7	Toy	20
8	Raspberry Pi	280
9	Toy	78
10	Laptop	2800
11	Laptop	4500
12	Paper	15
13	Laser printer	600
14	Scanner	255
15	Keyboard	80
16	TV	1270

II część – strona postawiona przy użyciu dockera

Instalacja npm install oraz sprawdzenie działania serwera po komendzie npm start.



Stworzenie pliku Dockerfile.



Instalacja oraz sprawdzenie poprawnego działania komendy docker.

```
radosaw@radosaw-VirtualBox:~/Pulpit/ASK/docker$ sudo docker run hello-world
[sudo] hasło użytkownika radosaw:

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/
```

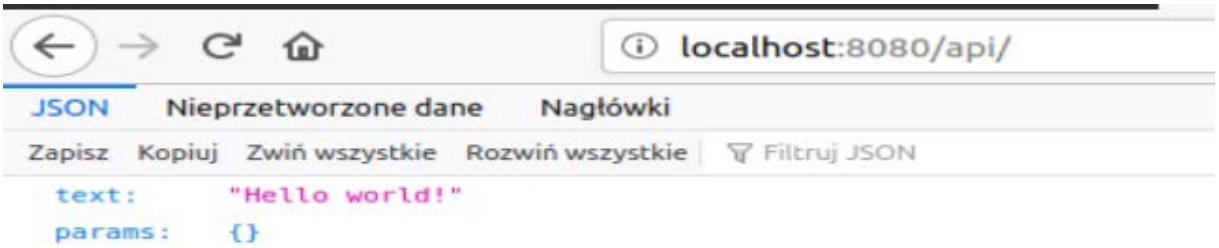
Zbudowanie obrazu na podstawie Dockerfile.

```
radosaw@radosaw-VirtualBox: ~/
radosaw@radosaw-VirtualBox:~/Pulpit/ASK/docker$ sudo docker build . -t docker-lecture-node
Sending build context to Docker daemon  17.54MB
Step 1/14 : FROM node:12.16.3-alpine3.10
 ---> 079fef2db96d
Step 2/14 : RUN mkdir -p /home/node/app/node_modules && chown -R node:node /home/node/app
 ---> Using cache
 ---> a9ed615bf034
Step 3/14 : WORKDIR /home/node/app
 ---> Using cache
 ---> 1d35e37a8cc3
Step 4/14 : USER node
 ---> Using cache
 ---> 89ea086db221
Step 5/14 : COPY --chown=node:node package*.json ./
 ---> Using cache
 ---> 77162807ad8d
Step 6/14 : RUN npm install
 ---> Using cache
 ---> c43883360502
Step 7/14 : COPY --chown=node:node ./src ./src
 ---> Using cache
 ---> 7139b352ea8d
Step 8/14 : EXPOSE 8080
 ---> Using cache
 ---> e74f342324bc
Step 9/14 : ENV DB_HOST=databse
 ---> Using cache
 ---> 20cdc4d92b68
Step 10/14 : ENV DB_PORT=3306
 ---> Using cache
 ---> 0f65f198ef8d
Step 11/14 : ENV DB_NAME=dockerdb
 ---> Using cache
 ---> d572fd5b341f
Step 12/14 : ENV DB_USER=dockerdb
 ---> Using cache
 ---> 71dc494d7a20
Step 13/14 : ENV DB_PSWD=Zaq12wsx
 ---> Running in 9d29bc99b1d1
Removing intermediate container 9d29bc99b1d1
 ---> 33f2fab88338
Step 14/14 : CMD [ "npm", "start" ]
 ---> Running in 4dcd60b013a7
Removing intermediate container 4dcd60b013a7
 ---> 04ca2a6f070f
Successfully built 04ca2a6f070f
Successfully tagged docker-lecture-node:latest
radosaw@radosaw-VirtualBox:~/Pulpit/ASK/docker$
```

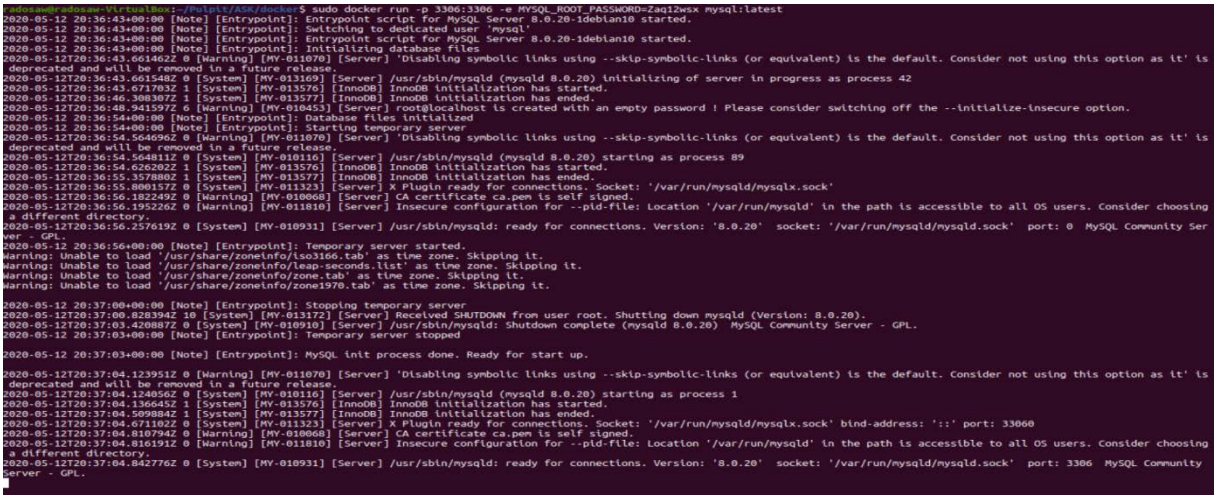
Zbudowanie i włączenie kontenera.

```
radosaw@radosaw-VirtualBox:~/Pulpit/ASK/docker$ sudo docker run -p 8080:8080 docker-lecture-node

> ask2020-node-server@1.0.0 start /home/node/app
> node .

App listening at http://undefined:8080 in development environment...
```

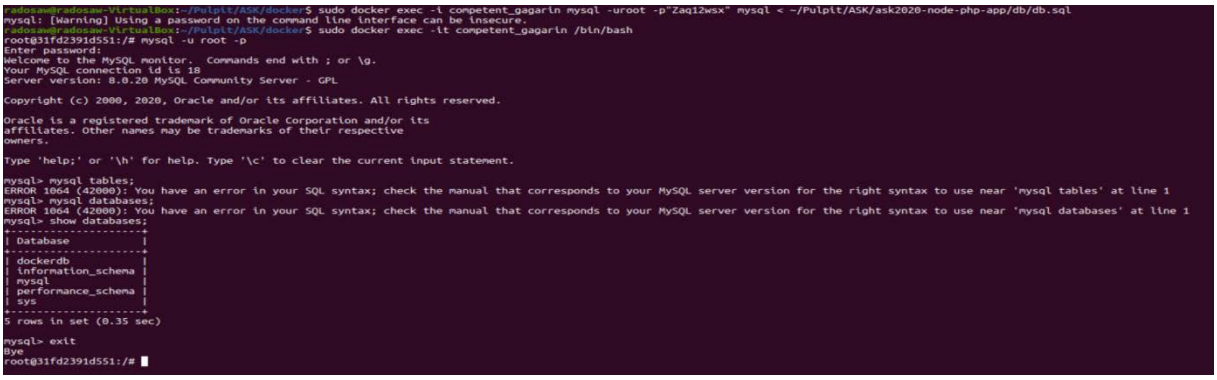
Sprawdzenie działania .



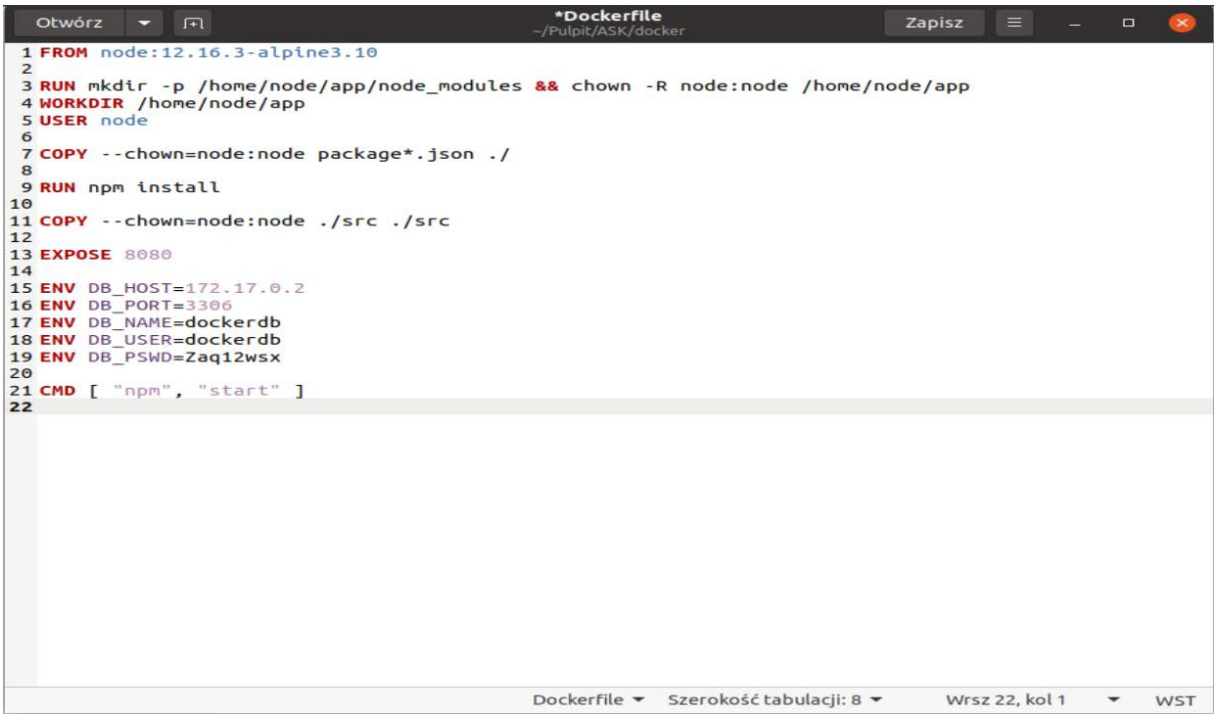
Stworzenie kontenera odpowiedzialnego za bazę danych.



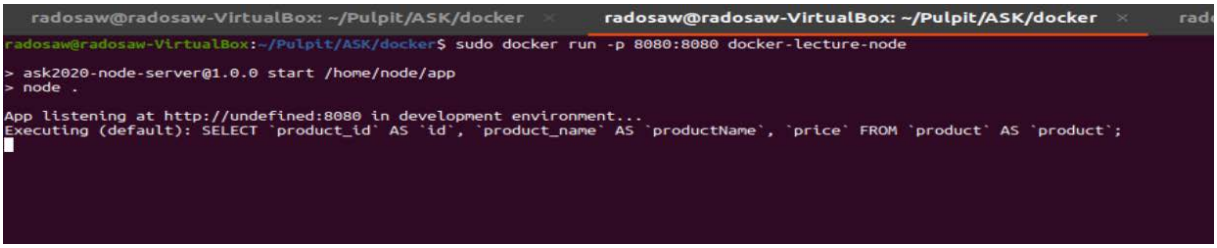
Import bazy do kontenera.



Poprawa pliku Dockerfile na odpowiednie IP kontenera z MySQL.



Ponowne stworzenie obrazu i kontenera.



Działanie obydwu kontenerów.

Sprawdzenie działania.

ID	Product name	Price
1	Phone	700
2	PC	2340
3	PC	16050
4	Raspberry Pi	150
5	Book	80
6	Book	45
7	Toy	20
8	Raspberry Pi	280
9	Toy	78
10	Laptop	2800
11	Laptop	4500
12	Paper	15
13	Laser printer	600
14	Scanner	255
15	Keyboard	80
16	TV	1270

III część – ulepszenie działania systemu poprzez dodanie kontenerów do sieci

Utworzenie sieci.

```
radosaw@radosaw-VirtualBox:~/Pulpit/ASK/docker$ sudo docker network create ask
e36224d6061aaa7c6b9de1898b8bb3022993a2852502b2ee3752afafbff25d1a
```

Z powodu problemów z siecią usunięto wszystkie stare kontenery i utworzono nowe z własnymi, nie automatycznie generowanymi nazwami.

Kontener z bazą danych.

```
radosaw@radosaw-VirtualBox:~$ sudo docker run --name=mysqldb --env="MYSQL_ROOT_PASSWORD=root" --publish 3306:3306 mysql
2020-05-14 09:55:37+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 8.0.20-1debian10 started.
2020-05-14 09:55:37+00:00 [Note] [Entrypoint]: Switching to dedicated user 'mysql'
2020-05-14 09:55:37+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 8.0.20-1debian10 started
```

Import do kontenera bazy danych

```
radosaw@radosaw-VirtualBox:~$ sudo docker exec -i mysqldb mysql -uroot -proot < /home/radosaw/Pulpit/ASK/ask2020-node-php-app/db/db.sql
mysql: [Warning] Using a password on the command line interface can be insecure.
radosaw@radosaw-VirtualBox:~$ sudo docker exec -it mysqldb /bin/bash
root@7b5b44d677d7:/# mysql
ERROR 1045 (28000): Access denied for user 'root'@'localhost' (using password: NO)
root@7b5b44d677d7:/# mysql -uroot -proot
mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 12
Server version: 8.0.20 MySQL Community Server - GPL

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases
    -> ;
+--------------------+
| Database           |
+--------------------+
| dockerdb           |
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
5 rows in set (0.06 sec)

mysql>
```

Dokonanie zmian w pliku Dockerfile, zmiana DB_HOST z IP kontenera z bazą MySQL na jego nazwę.

Otwórz — Dockerfile ~/Pulpit/ASK/docker — Zapisz

```
1  FROM node:12.16.3-alpine3.10
2
3  RUN mkdir -p /home/node/app/node_modules && chown -R node:node /home/node/app
4  WORKDIR /home/node/app
5  USER node
6
7  COPY --chown=node:node package*.json ./
8
9  RUN npm install
10
11 COPY --chown=node:node ./src ./src
12
13 ENV DB_HOST=mysqldb
14 ENV DB_PORT=3306
15 ENV DB_NAME=dockerdb
16 ENV DB_USER=dockerdb
17 ENV DB_PSWD=Zaq12wsx
18
19 CMD [ "npm", "start" ]
```

Wczytywanie pliku „/home/radosaw/Pulpit/ASK/doc... Dockerfile Szerokość tabulacji: 8 Wrsz 13, kol 20 WST

Zbudowanie obrazu i utworzenie na jego podstawie kontenera.

```
radosaw@radosaw-VirtualBox:~/Pulpit/ASK/docker$ sudo docker build . -t app
Sending build context to Docker daemon  17.54MB
Step 1/13 : FROM node:12.16.3-alpine3.10
 ---> 079fef2db96d
Step 2/13 : RUN mkdir -p /home/node/app/node_modules && chown -R node:node /home/node/app
 ---> Using cache
 ---> a9ed615bf034
Step 3/13 : WORKDIR /home/node/app
 ---> Using cache
 ---> 1d35e37a8cc3
Step 4/13 : USER node
 ---> Using cache
 ---> 89ea086db221
Step 5/13 : COPY --chown=node:node package*.json ./
 ---> Using cache
 ---> 77162807ad8d
Step 6/13 : RUN npm install
 ---> Using cache
 ---> c43883360502
Step 7/13 : COPY --chown=node:node ./src ./src
 ---> Using cache
 ---> 7139b352ea8d
Step 8/13 : ENV DB_HOST=mysqldb
 ---> Running in 22ed37fd837b
Removing intermediate container 22ed37fd837b
 ---> 564ef98303de
Step 9/13 : ENV DB_PORT=3306
 ---> Running in 083c0e465520
Removing intermediate container 083c0e465520
 ---> 5aaf90a30298
Step 10/13 : ENV DB_NAME=dockerdb
 ---> Running in 56df28e98867
Removing intermediate container 56df28e98867
 ---> 518dff4a773a
Step 11/13 : ENV DB_USER=dockerdb
 ---> Running in f56739e1f038
Removing intermediate container f56739e1f038
 ---> 32017c40de66
Step 12/13 : ENV DB_PSWD=Zaq12wsx
 ---> Running in 155f4cdf43ed
Removing intermediate container 155f4cdf43ed
 ---> dd428124b606
Step 13/13 : CMD [ "npm", "start" ]
 ---> Running in 4da8f88eb3aa
Removing intermediate container 4da8f88eb3aa
 ---> bb570081dde9
Successfully built bb570081dde9
Successfully tagged app:latest
radosaw@radosaw-VirtualBox:~/Pulpit/ASK/docker$ sudo docker images
REPOSITORY            TAG                 IMAGE ID            CREATED             SIZE
app                   latest              bb570081dde9        7 seconds ago       107MB
docker-lecture-node   latest              c090ffbb4453        36 hours ago        107MB
node                  12.16.3-alpine3.10  079fef2db96d        2 weeks ago         88.4MB
mysql                 latest              a7a67c95e831        2 weeks ago         541MB
hello-world           latest              bf756fb1ae65        4 months ago        13.3kB
radosaw@radosaw-VirtualBox:~/Pulpit/ASK/docker$
```

```
radosaw@radosaw-VirtualBox:~/Pulpit/ASK/docker$ sudo docker run --name=appcon -p 8080:8080 app

> ask2020-node-server@1.0.0 start /home/node/app
> node .

App listening at http://undefined:8080 in development environment...
```

Dodanie kontenerów do sieci.

```
radosaw@radosaw-VirtualBox:~/Pulpit/ASK/docker$ sudo docker network create ask
33122c0d93760dca42f0cb15ec55a2acdafb7c237b6bfdad6977a96c10acbf71
radosaw@radosaw-VirtualBox:~/Pulpit/ASK/docker$ sudo docker network connect ask mysqldb
radosaw@radosaw-VirtualBox:~/Pulpit/ASK/docker$ sudo docker network connect ask appcon
```

```
radosaw@radosaw-VirtualBox:~/Pulpit/ASK/docker$ sudo docker network inspect ask
[
    {
        "Name": "ask",
        "Id": "33122c0d93760dca42f0cb15ec55a2acdafb7c237b6bfdad6977a96c10acbf71",
        "Created": "2020-05-14T12:09:58.263113051+02:00",
        "Scope": "local",
        "Driver": "bridge",
        "EnableIPv6": false,
        "IPAM": {
            "Driver": "default",
            "Options": {},
            "Config": [
                {
                    "Subnet": "172.18.0.0/16",
                    "Gateway": "172.18.0.1"
                }
            ]
        },
        "Internal": false,
        "Attachable": false,
        "Ingress": false,
        "ConfigFrom": {
            "Network": ""
        },
        "ConfigOnly": false,
        "Containers": {
            "7b5b44d677d7f6f7c9ec49cb2fdb0aab5f5f50d3939169b041c89cd95bff0766": {
                "Name": "mysqldb",
                "EndpointID": "de062b6eb8b1bbcdbd26eed5e8ba7f9f9bfa895aead12b60edff7e00ba6b6ca2",
                "MacAddress": "02:42:ac:12:00:02",
                "IPv4Address": "172.18.0.2/16",
                "IPv6Address": ""
            },
            "9b4372d6ead5740b56255b114f17e6e202d5e1a08e387fab92646d4872fdf2e2": {
                "Name": "appcon",
                "EndpointID": "9079ac495adc1cd11f79d7122e4e407081dc323c9fdc14e200083b6b4b49113a",
                "MacAddress": "02:42:ac:12:00:03",
                "IPv4Address": "172.18.0.3/16",
                "IPv6Address": ""
            }
        },
        "Options": {},
        "Labels": {}
    }
]
```

Wynik działania w postaci wyświetlonej tabeli.

ID	Product name	Price
1	Phone	700
2	PC	2340
3	PC	16050
4	Raspberry Pi	150
5	Book	80
6	Book	45
7	Toy	20
8	Raspberry Pi	280
9	Toy	78
10	Laptop	2800
11	Laptop	4500
12	Paper	15
13	Laser printer	600
14	Scanner	255
15	Keyboard	80
16	TV	1270

IV część – uruchomienie kontenerów za pomocą docker-compose up

Utworzenie i uzupełnienie pliku docker-compose.yml.

Uruchomienie komendy docker-compose up.

Wynik działania.

Podsumowanie

Docker jest wszechstronnym i uniwersalnym narzędziem często stosowanym w praktyce. Jest często używany w firmach zajmujących się developmentem. Pozwala na między innymi utrzymanie na jednej maszynie kilku projektów działających na podstawie różnych wersji języka. Bez użycia Dockera konieczne byłoby zmienianie wersji języka na komputerze przy przechodzeniu na inny projekt.

Pracuję jako Junior PHP Developer w firmie Gallinago i mam przyjemność używać Dockera na codzień.