

# Programowanie aplikacji w chmurze obliczeniowe

## **LABORATORIUM 5**

### **WYKORZYSTANIE MAGAZYNÓW PRZECHOWYWANIA DANYCH W ŚRODOWISKU DOCKER.**

**Rafał Maciąg**

Numer albumu: 92935

Grupa Laboratoryjna: 6.5

Prowadzący: mgr. D.Głuchowski

## Zadanie 5.1. Podstawowa obsługa wolumenów

```
rafal@rafal-VirtualBox:~$ sudo docker volume create RedishRob
[sudo] password for rafal:
RedishRob
rafal@rafal-VirtualBox:~$ sudo docker volume ls
DRIVER      VOLUME NAME
local       RedishRob
```

```
rafal@rafal-VirtualBox:~$ sudo docker run -it --rm --name lab5_zad1 --mount source=RedishRob,target=/data redis
1:C 13 Apr 2022 18:47:32.120 # 000000000000 Redis is starting 000000000000
1:C 13 Apr 2022 18:47:32.120 # Redis version 6.2.6, bits=64, commit=00000000, modified=0, pid=1, just started
1:C 13 Apr 2022 18:47:32.120 # Warning: no config file specified, using the default config. In order to specify a config file use redis-server /path/to/redis.conf
1:M 13 Apr 2022 18:47:32.121 * Increased maximum number of open files to 10032 (it was originally set to 1024).
1:M 13 Apr 2022 18:47:32.122 * monotonic clock: POSIX clock_gettime

Redis 6.2.6 (00000000/0) 64 bit
Running in standalone mode
Port: 6379
PID: 1
https://redis.io

1:M 13 Apr 2022 18:47:32.123 # Server initialized
1:M 13 Apr 2022 18:47:32.123 # WARNING overcommit_memory is set to 0! Background save may fail under low memory condition. To fix this issue add 'vm.overcommit_memory = 1' to /etc/sysctl.conf and then
boot or run the command 'sysctl vm.overcommit_memory=1' for this to take effect.
1:M 13 Apr 2022 18:47:32.123 * Ready to accept connections
```

```
{
  "Cmd": [
    "redis-server"
  ],
  "Image": "redis",
  "Volumes": {
    "/data": {}
  },
  "WorkingDir": "/data",
  "Entrypoint": [
    "docker-entrypoint.sh"
  ],
  "OnBuild": null,
  "Labels": {}
},
"NetworkSettings": {
  "Bridge": "",
  "SandboxID": "1965da768d6ae82505c208bb4658b6172ce5ee37037fad05b20896105906569b",
  "HairpinMode": false,
  "LinkLocalIPv6Address": "",
  "LinkLocalIPv6PrefixLen": 0,
  "Ports": {
    "6379/tcp": null
  },
  "SandboxKey": "/run/snap.docker/netns/1965da768d6a",
  "SecondaryIPAddresses": null,
  "SecondaryIPv6Addresses": null,
  "EndpointID": "5ac15793fba9f01a8b56c368cdb638d90999bec81e4e412f785758e43ec8c426",
  "Gateway": "172.17.0.1",
  "GlobalIPv6Address": "",
  "GlobalIPv6PrefixLen": 0,
  "IPAddress": "172.17.0.2",
  "IPPrefixLen": 16,
  "IPv6Gateway": "",
  "MacAddress": "02:42:ac:11:00:02",
  "Networks": {
    "bridge": {
      "IPAMConfig": null,
      "Links": null,
      "Aliases": null,
      "NetworkID": "1828208e9b7c511a3523ec42b1c3bba84fad90234549379ff358a76f9ca9af7",
      "EndpointID": "5ac15793fba9f01a8b56c368cdb638d90999bec81e4e412f785758e43ec8c426",
      "Gateway": "172.17.0.1",
      "IPAddress": "172.17.0.2",
      "IPPrefixLen": 16,
      "IPv6Gateway": "",
      "GlobalIPv6Address": "",
      "GlobalIPv6PrefixLen": 0
    }
  }
}
```

## Zadanie 5.2. Zaawansowana obsługa wolumenów

```
rafal@rafal-VirtualBox:~$ cd lab5
rafal@rafal-VirtualBox:~/lab5$ cat dockerfile
From httpd
rafal@rafal-VirtualBox:~/lab5$ cat html_dir/index.html
<DOCTYPE html>
<html>
<head></head>
<body>

    <h1> Rafal Maciag </h1>
</body>

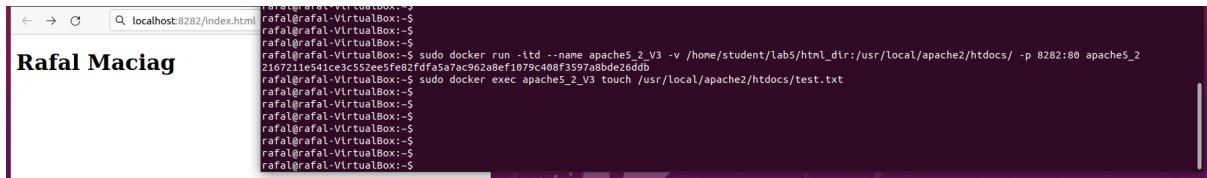
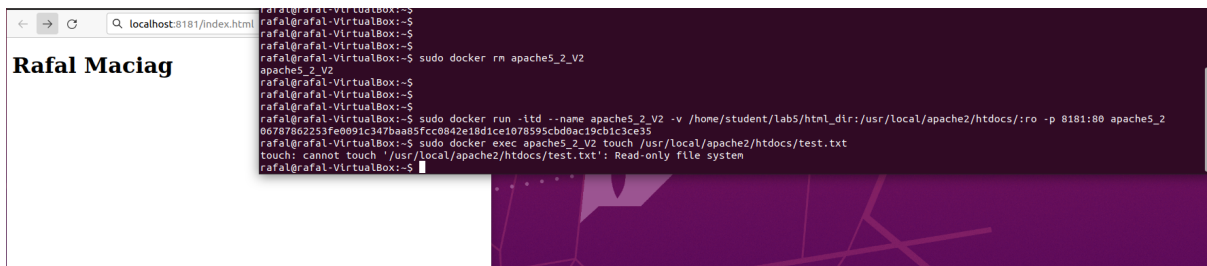
</html>
rafal@rafal-VirtualBox:~/lab5$
```

```
Usage:  docker build [OPTIONS] PATH | URL | -

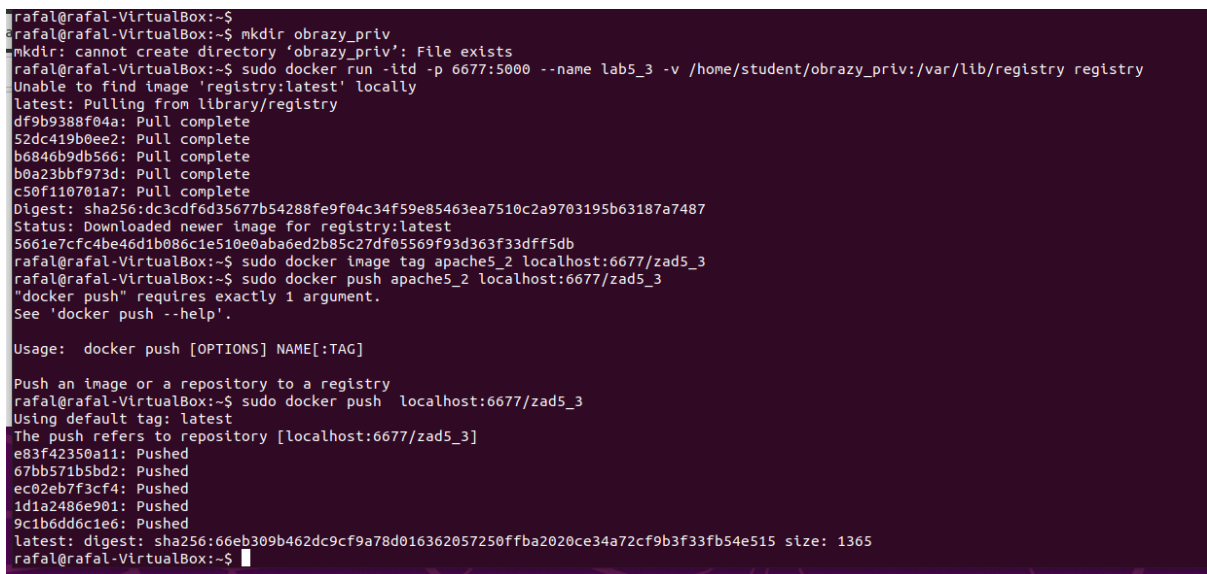
Build an image from a Dockerfile
rafal@rafal-VirtualBox:~/lab5$ sudo docker build -t apache5_2 .

Sending build context to Docker daemon  8.192kB
Step 1/1 : From httpd
latest: Pulling from library/httpd
1fe172e4850f: Extracting 29.82MB/31.38MB
e2fa1fe9b1ec: Download complete
60dd7398e74e: Download complete
ea2ca81c6d4c: Download complete
f646c69a26ec: Download complete
```





## Zadanie 5.3. Dedykowane repozytorium obrazów



```

zad5_3
├── layers
│   └── sha256
│       ├── 1fe172e4850f03bb45d41a20174112bc119fbfec42a650edbbd8491aee32e3c3
│       │   └── link
│       ├── 60dd7398e74ea818d77c54d4d8616797fde2b96e4c9b459fd9db0d3496224504
│       │   └── link
│       ├── c30a467716957ab3adabf990785d220056949d80b3aa46d90de3ea44d532b03e
│       │   └── link
│       ├── e2fa1fe9b1ec33e77081649241dfe4315bb829f497dc224c8841763ed518a061
│       │   └── link
│       ├── ea2ca81c6d4cc8f4cd4ff8335974362ee9115256cbdcde052ad639fddbbcbe04
│       │   └── link
│       └── f646c69a26ec7fbfbb6e84d0f321b84bd4b97b35ac1a3b24533a65496491b256
│           └── link
├── manifests
│   ├── revisions
│   │   └── sha256
│   │       └── 66eb309b462dc9cf9a78d016362057250ffba2020ce34a72cf9b3f33fb54e515
│   │           └── link
│   └── tags
│       ├── latest
│       │   ├── current
│       │   │   └── link
│       │   └── index
│       │       └── sha256
│       │           └── 66eb309b462dc9cf9a78d016362057250ffba2020ce34a72cf9b3f33fb54e515
│       │               └── link
└── _uploads

```

## Zadanie 5.4. Ograniczenie w dostępie do zasobów systemu macierzystego

```

rafal@rafal-VirtualBox:~/0018zy_privs$ cd
rafal@rafal-VirtualBox:~$ sudo docker run -itd --name limiter --cpus="1.0" --memory="512m" alpine
Unable to find image 'alpine:latest' locally
latest: Pulling from library/alpine
df9b9388f04a: Already exists
Digest: sha256:4edbd2beb5f78b1014028f4fbb99f3237d9561100b6881aabbf5acce2c4f9454
Status: Downloaded newer image for alpine:latest
9b0fb32a9fa1e42576a30adeac642cb154ce8269785244003ebab0fbaba98e2e
rafal@rafal-VirtualBox:~$ sudo docker stats --no-stream limiter
CONTAINER ID   NAME      CPU %     MEM USAGE / LIMIT   MEM %     NET I/O       BLOCK I/O    PIDS
9b0fb32a9fa1   limiter   0.00%     376KiB / 512MiB     0.07%     2.69kB / 0B    0B / 0B      1
rafal@rafal-VirtualBox:~$

```

## WNIOSKI

Możliwość zapisu pliku na wolumenie podpiętym do drugiego kontenera (pracującego z tym wolumenem w trybie pozwalającym tylko na odczyt) została przetestowana - próba utworzenia nowego pliku na tym wolumenie zakończyła się niepowodzeniem - wyświetlając stosowny komunikat. Podobna próba w przypadku kontenera trzeciego - pracującego w trybie zapisu i odczytu zakończyła się sukcesem. W przypadku ustalania, że kontener

będzie pracował z danym wolumenem w trybie tylko do odczytu, należy dodać opcję `:ro` dodając wolumen, w sytuacji kiedy kontener ma pracować w trybie odczytu i zapisu nie trzeba dodawać żadnych opcji, ponieważ jest to domyślny tryb pracy. Wszystkie 3 kontenery poprawnie wyświetlały przykładową stronę internetową zapisaną na wolumenie.

Celem zadania 4 było utworzenie kontenera na bazie obrazu `alpine`, którego zasoby zostaną ograniczone do 512MB pamięci RAM oraz jednego rdzenia procesora. Modyfikacje zostały zaaplikowane przy pomocy opcji `--memory` oraz `--cpus`. Następnie statystyki kontenera została sprawdzona przy pomocy polecenia `docker stats`.