

# Sprawozdanie

Programowanie aplikacji w chmurze obliczeniowej

## Laboratorium 8

### INSTALACJA SYSTEMU HADOOP W KONTENERZE DOCKER

Łukasz Oleksiuk

Grupa: 6.6

Index: 097690

**Zadanie 8.1.** Porównanie obrazów Hadoop:

**Podejście z laboratorium:**

Plik Dockerfile:

---

```
1 # Bazowy obraz Ubuntu 20.04
2 FROM ubuntu:20.04
3
4 # Instalacja niezbędnych pakietów
5 RUN apt-get update && \
6     apt-get install -y openjdk-11-jdk wget ssh rsync && \
7     apt-get clean
8
9 # Pobranie i rozpakowanie Hadoop
10 RUN wget https://archive.apache.org/dist/hadoop/common/hadoop-3.3.1/hadoop-3.3.1.tar.gz && \
11     tar -xvzf hadoop-3.3.1.tar.gz && \
12     mv hadoop-3.3.1 /usr/local/hadoop && \
13     rm hadoop-3.3.1.tar.gz
14
15 # Ustawienie zmiennych środowiskowych
16 ENV JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64
17 ENV HADOOP_HOME=/usr/local/hadoop
18 ENV PATH=$PATH:$HADOOP_HOME/bin:$HADOOP_HOME/sbin
19
20 # Utworzenie katalogów dla HDFS
21 RUN mkdir -p /usr/local/hadoop/hdfs/namenode && \
22     mkdir -p /usr/local/hadoop/hdfs/datanode
23
24 # Skopiowanie pliku konfiguracyjnego hadoop-env.sh
25 COPY hadoop-env.sh /usr/local/hadoop/etc/hadoop/hadoop-env.sh
26
27 # Konfiguracja SSH dla Hadoop
28 RUN ssh-keygen -t rsa -P "" -f ~/.ssh/id_rsa && \
29     cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
30
31 # Skrypt startowy Hadoop
32 COPY start-hadoop.sh /usr/local/bin/start-hadoop.sh
33 RUN chmod +x /usr/local/bin/start-hadoop.sh
34
35 # Eksponowanie portów
36 EXPOSE 50070 8088
37
38 # Ustawienie komendy startowej
39 CMD ["/usr/local/bin/start-hadoop.sh"]
```

---

Plik hadoop-env.sh:

```
Dockerfile x hadoop-env.sh x start-hadoop.sh x
1 # The java implementation to use.
2 export JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64
3 export HADOOP_HOME=/usr/local/hadoop
4
```

Skrypt start-hadoop.sh:

```
Dockerfile x hadoop-env.sh x start-hadoop.sh x
1 #!/bin/bash
2
3 # Start Hadoop services
4 $HADOOP_HOME/sbin/start-dfs.sh
5 $HADOOP_HOME/sbin/start-yarn.sh
6
7 # Keep the container running
8 tail -f /dev/null
9
```

Uruchomienie kontenera:

```
student@vhost1:~/docker_lab8$ sudo docker run -d --name hadoop_custom -p 50071:50070 -p 8089:8088 custom-hadoop
a88726f145ff2fdf6d78509787c7ab2f23d779cc139f87d9040a4114af654dcf
```

```
student@vhost1:~/docker_lab8$ sudo docker inspect hadoop_custom
[
  {
    "Id": "a88726f145ff2fdf6d78509787c7ab2f23d779cc139f87d9040a4114af654dcf",
    "Created": "2024-05-21T11:41:25.356138807Z",
    "Path": "/bin/bash",
    "Args": [],
    "State": {
      "Status": "exited",
      "Running": false,
      "Paused": false,
      "Restarting": false,
      "OOMKilled": false,
      "Dead": false,
      "Pid": 0,
      "ExitCode": 0,
      "Error": "",
      "StartedAt": "2024-05-21T11:41:25.817125763Z",
      "FinishedAt": "2024-05-21T11:41:25.834634131Z"
    }
  },
]
```

## Podejście z dockerhub:

Pobranie obrazu:

```
student@vhost1:~/docker_lab8$ sudo docker pull sequenceiq/hadoop-docker:2.7.1
2.7.1: Pulling from sequenceiq/hadoop-docker
Image docker.io/sequenceiq/hadoop-docker:2.7.1 uses outdated schema1 manifest format. Please upgrade to a schema2 image
for better future compatibility. More information at https://docs.docker.com/registry/spec/deprecated-schema-v1/
b253335dcf03: Pulling fs layer
a3ed95caeb02: Pulling fs layer
69623ef05416: Pulling fs layer
8d2023764774: Waiting
0c3c0ff61963: Waiting
ff0696749bf6: Waiting
72accdc282f3: Waiting
5298ddb3b339: Waiting
f252bbba6bda: Waiting
3984257f0553: Waiting
26343a20fa29: Waiting
f3e272e0e801: Waiting
ad78a593ca62: Waiting
673712aa7667: Waiting
aaf06cd0aa6e: Waiting
fed9c9377250: Waiting
d4385c519f63: Waiting
49ca93868354: Waiting
98e62c38a660: Waiting
3679d1cf91a0: Waiting
31ae294be02b: Waiting
13605254d8c3: Waiting
a54805751dfa: Waiting
38537e9c387f: Waiting
dc639853e053: Pull complete
e267620cd7fd: Pull complete
93990a6b26ca: Pull complete
11ffe2baf32d: Pull complete
c91b10bf3a44: Pull complete
adede6edfea0: Pull complete
4afb2f219fa7: Pull complete
0335bc4000de: Pull complete
e6c5265506dc: Pull complete
3bb2b06400be: Pull complete
d9665143ac9a: Pull complete
2a1a28b12647: Pull complete
5c175609cbf3: Pull complete
e2a7d6798159: Pull complete
88d87e462c71: Pull complete
3a404fc6437e: Pull complete
5517052ef612: Pull complete
fa61c616ddd1: Pull complete
d4ab0c19cb91: Pull complete
9aa826a9ca93: Pull complete
b2ecd44f6d78: Pull complete
824658b0b14c: Pull complete
e5c31d8cbbce: Pull complete
Digest: sha256:2da37e4e4eaa57bc99dd64987391ce9e1384c63b4fa56b7525a60849a758fb950
Status: Downloaded newer image for sequenceiq/hadoop-docker:2.7.1
docker.io/sequenceiq/hadoop-docker:2.7.1
```

Uruchomienie kontenera:

```
student@vhost1:~/docker_lab8$ sudo docker run -d --name hadoop_hub -p 50070:50070 -p 8088:8088 sequenceiq/hadoop-docker:2.7.1 /etc/bootstrap.sh -bash
[sudo] password for student:
692f3b08a60e5abad830403041a1df02b725d39ca0a652b6643aa8d0890a0e7d
```

```
student@vhost1:~/docker_lab8$ sudo docker inspect hadoop_hub
[
  {
    "Id": "692f3b08a60e5abad830403041a1df02b725d39ca0a652b6643aa8d0890a0e7d",
    "Created": "2024-05-21T11:34:16.02963944Z",
    "Path": "/etc/bootstrap.sh",
    "Args": [
      "-bash"
    ],
    "State": {
      "Status": "exited",
      "Running": false,
      "Paused": false,
      "Restarting": false,
      "OOMKilled": false,
      "Dead": false,
      "Pid": 0,
      "ExitCode": 0,
      "Error": "",
      "StartedAt": "2024-05-21T11:34:16.300411194Z",
      "FinishedAt": "2024-05-21T11:34:35.644841025Z"
    }
  }
]
```

Sprawdzenie stanu kontenera:

```
student@vhost1:~/docker_lab8$ sudo docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS
692f3b08a60e   sequenceiq/hadoop-docker:2.7.1     "/etc/bootstrap.sh -..." 8 seconds ago  Up 8 seconds  2122/tcp, 8030-8033/tcp, 8040/tcp, 8042/tc
p, 19888/tcp, 49707/tcp, 50010/tcp, 0.0.0.0:8088->8088/tcp, :::8088->8088/tcp, 50020/tcp, 50075/tcp, 50090/tcp, 0.0.0.0:50070->50070/tcp, ::50070
->50070/tcp
hadoop_hub
edfbd1bb30d1   f629ad336abd                       "/bin/sh -c 'wget ht..." 2 minutes ago  Up 2 minutes
```

## Wnioski:

Po wykonaniu tego laboratorium można zauważyć, że gotowy obraz z Docker Hub umożliwia łatwe i szybkie uruchomienie środowiska Hadoop bez konieczności konfiguracji od podstaw. Obrazy te są standardowe, więc mogą być kompatybilne z większą ilością tworzonych projektów. Z drugiej strony własny obraz umożliwia pełną konfigurację środowiska, jednak jest to bardziej czasochłonne i wymaga większej wiedzy, przy własnym obrazie można popełnić więcej błędów przez co utworzony kontenerem może zachowywać się niezgodnie z oczekiwaniami