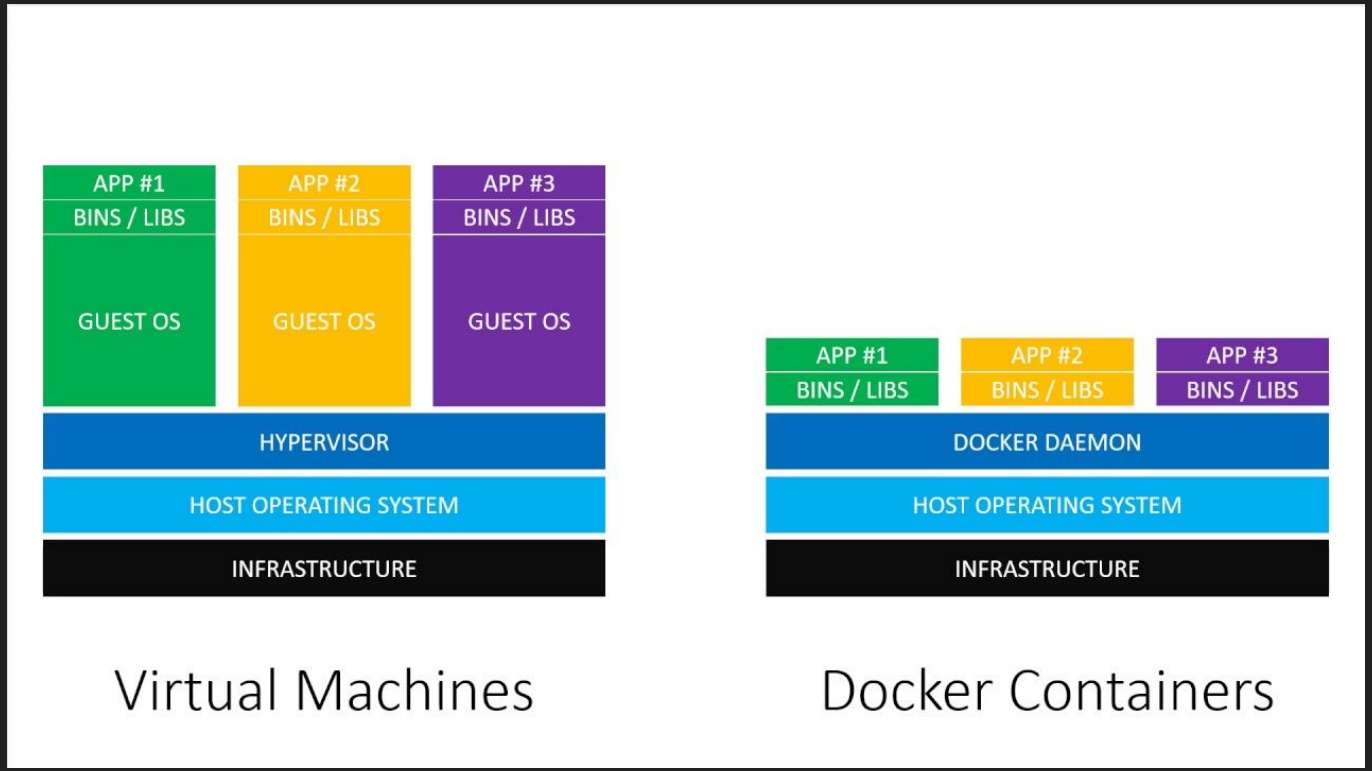
Konteneryzacja i jej wykorzystanie w projekcie z uczeniem maszynowym

Konteneryzacja

- Wykorzystanie kernela hosta
- Ograniczony podzbiór zasobów
- Operating-system-level virtualization
- linux namespaces

Konteneryzacja vs wirtualizacja (pełna)



Konteneryzacja - performance

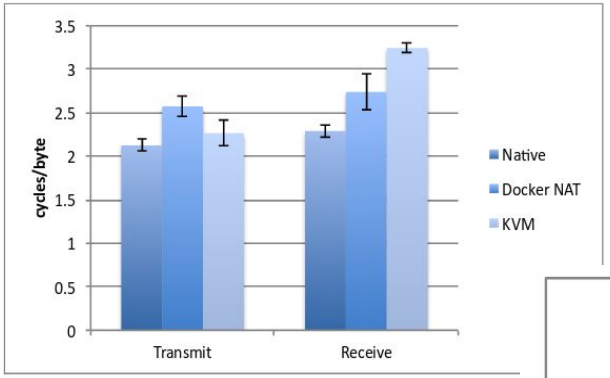


Fig. 2. TCP bulk transfer efficiency (CPU cycles/byte)

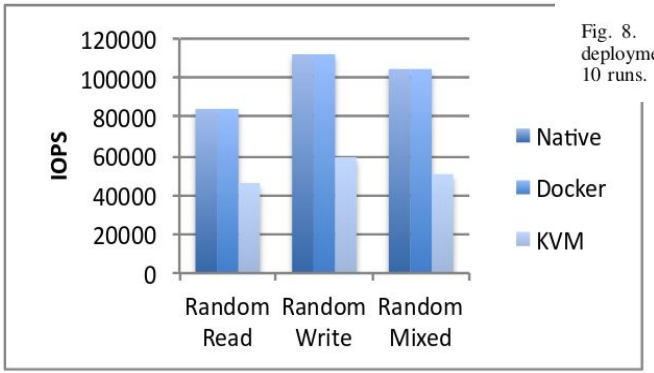


Fig. 6. Random I/O throughput (IOPS).

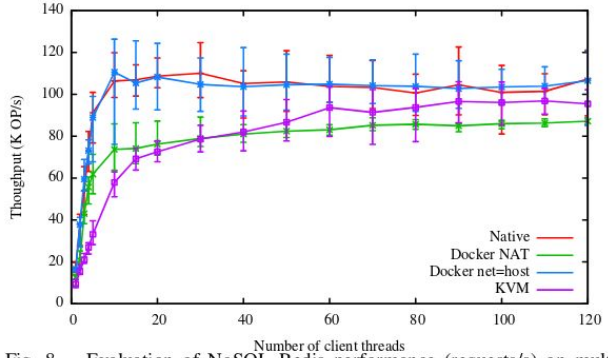


Fig. 8. Evaluation of NoSQL Redis performance (requests/s) on multiple deployment scenarios. Each data point is the arithmetic mean obtained from 10 runs.

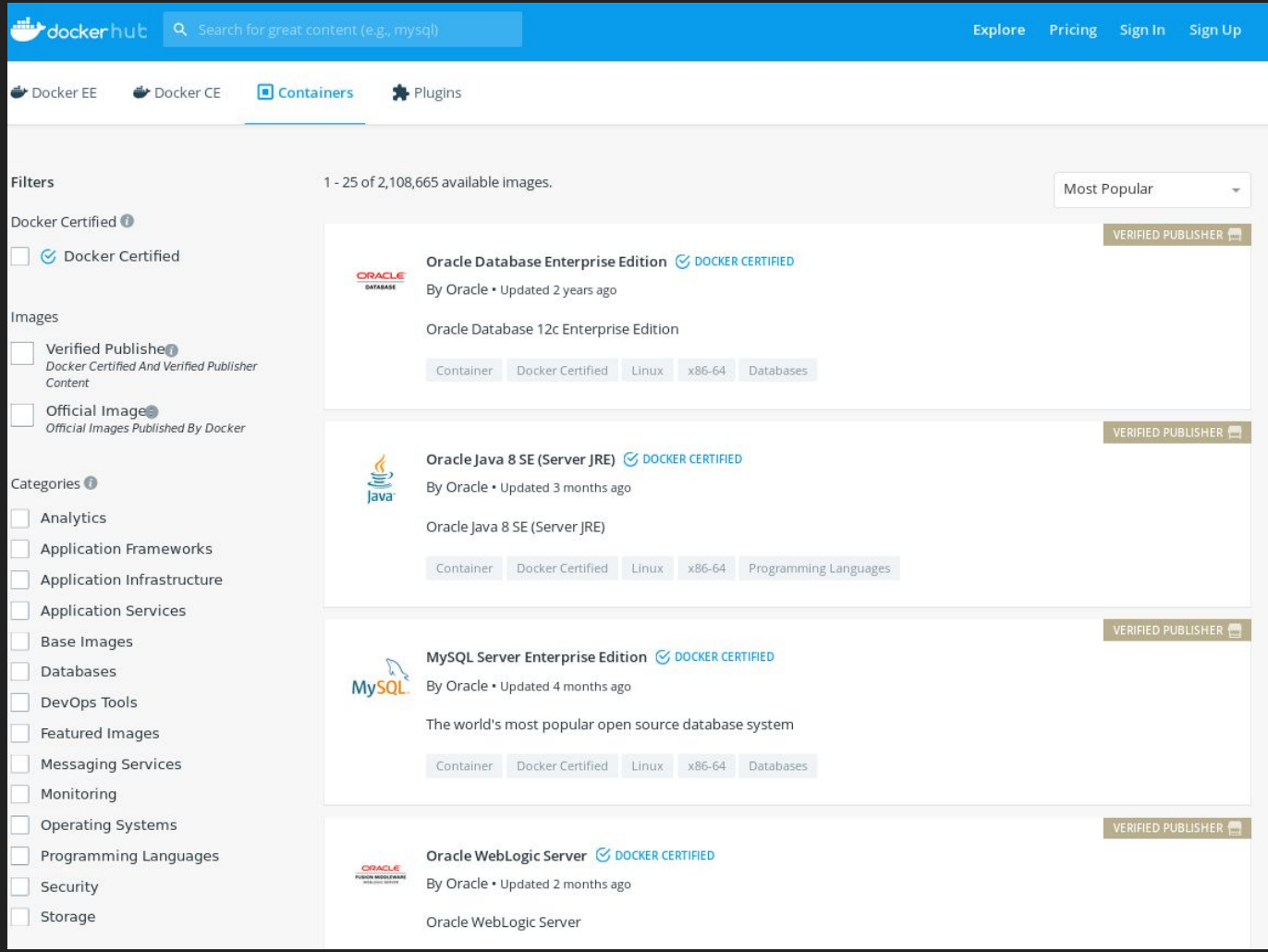
https://domino.research.ibm.com/library/cyberdig.nsf/papers/0929052195DD819C85257D2300681E7B/$File/rc25482.pdf

wirtualizacja	konteneryzacja
wirtualizacja na poziomie sprzętowym	wirtualizacja na poziomie systemu operacyjnego
“ciężkie”	“lekkie”
powolny dostęp do zasobów	dostęp do zasobów w czasie rzeczywistym
limitowana wydajność	natywna wydajność
mniejsza przenośność	przenośność
bezpieczniejszy - w pełni odizolowany od hosta	mniej bezpieczny - izolacja na poziomie procesu (Deamon dockera wymaga uprawnień *root* - Docker daemon attack surface)

Docker hub

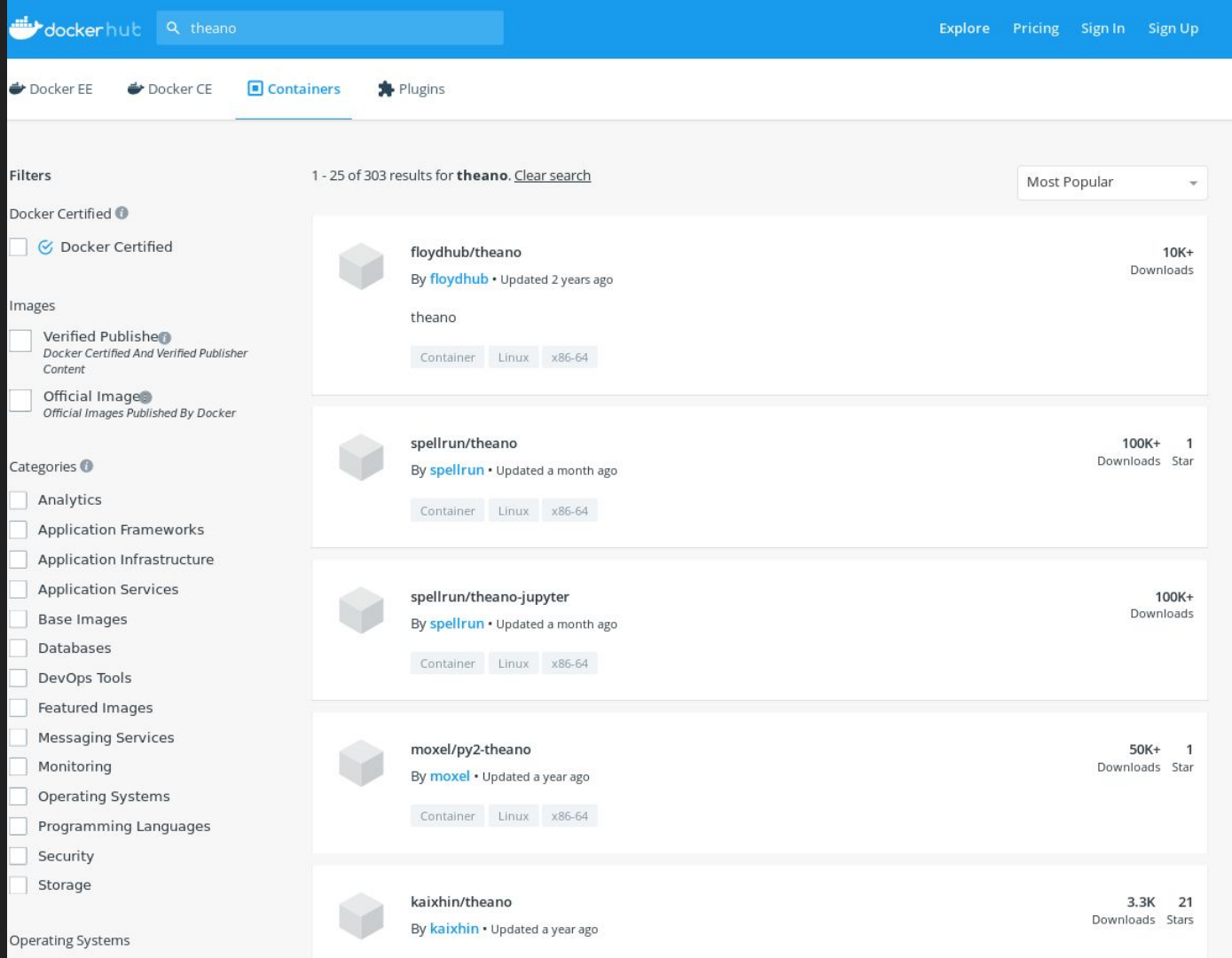
world's largest library and community for container images

~ 2M images



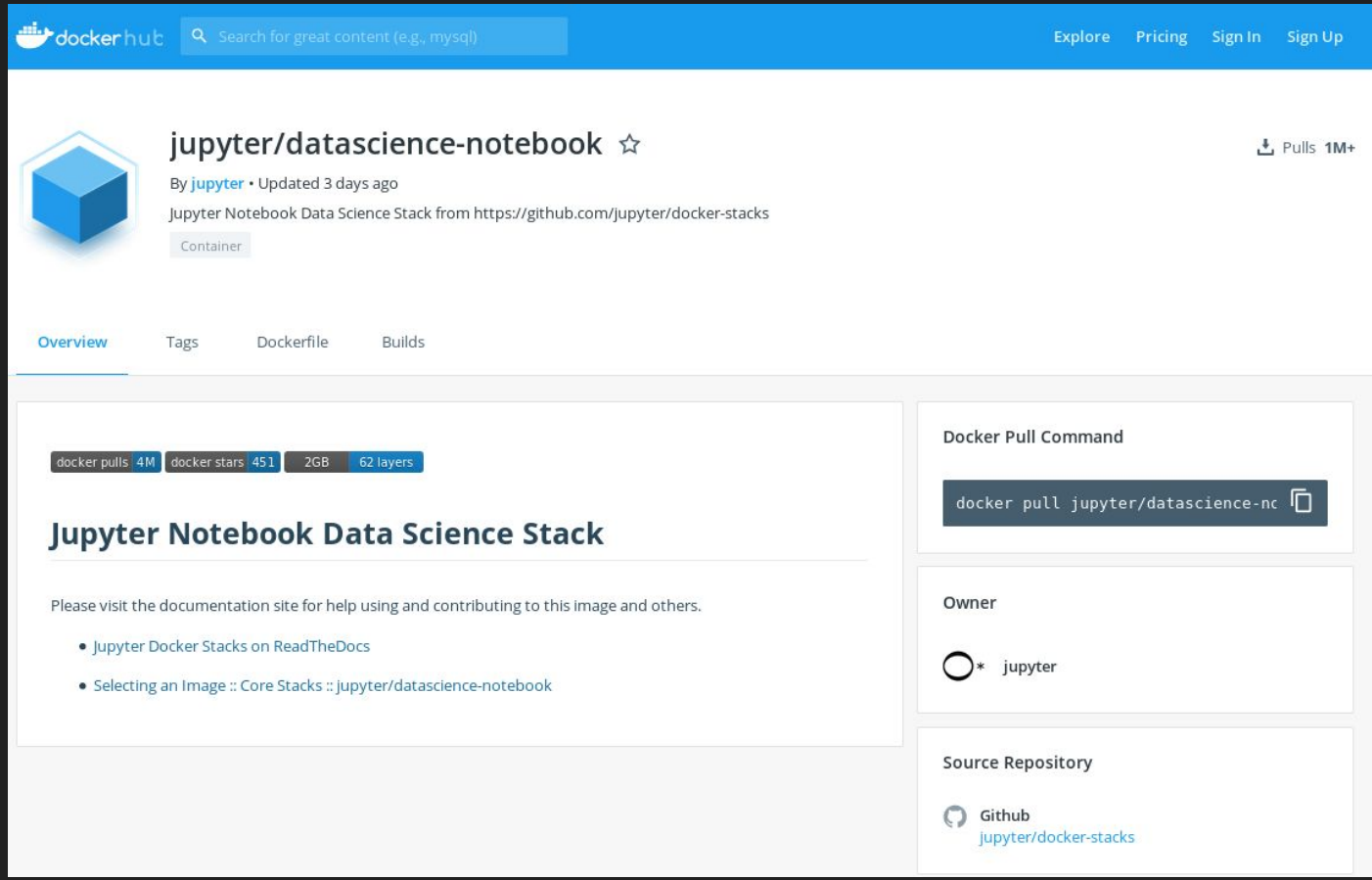
Docker hub

Theano ?



Docker hub

jupyter/datascience-notebook



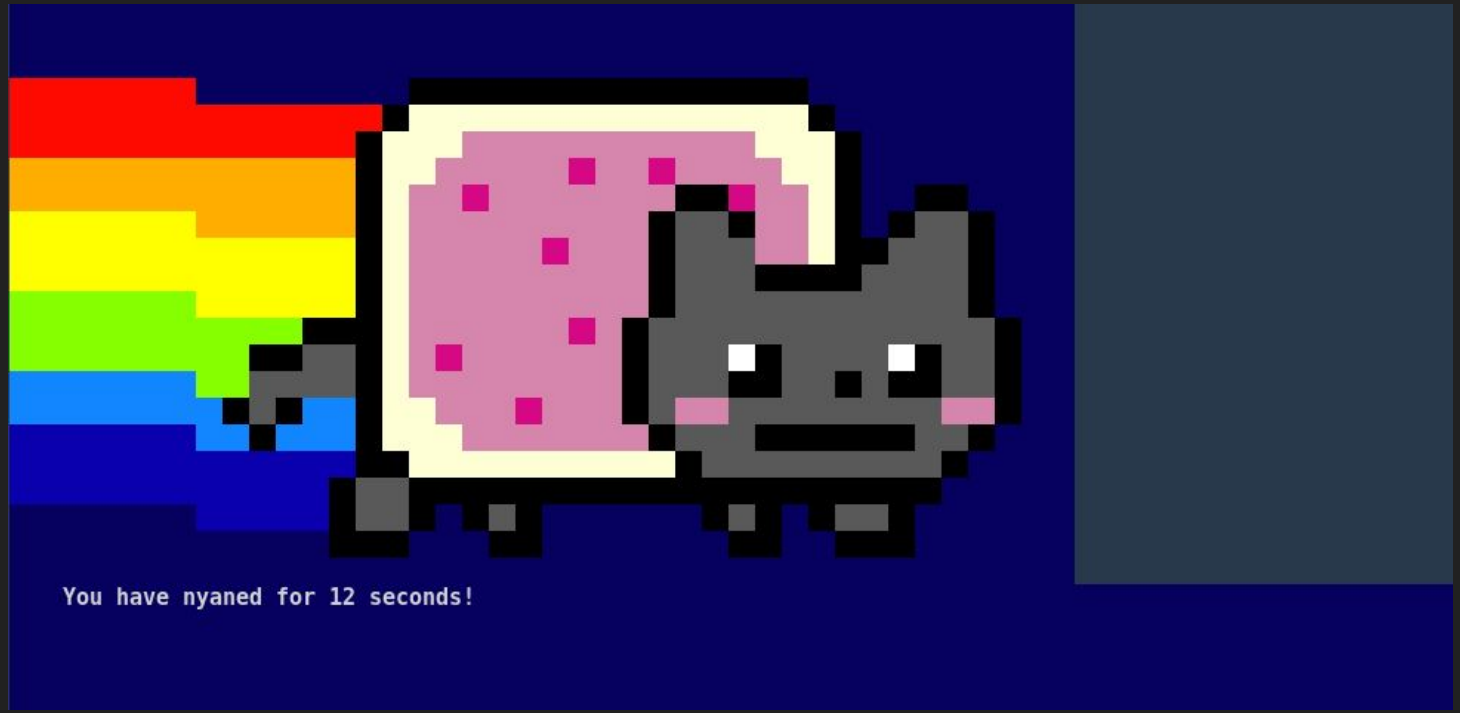
Dockerowe *hello-world*

```
$ docker run hello-world
```

```
jakub@jakub:~/docker$ sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
1b930d010525: Pull complete
Digest: sha256:2557e3c07ed1e38f26e389462d03ed943586f744621577a99efb77324b0fe535
Status: Downloaded newer image for hello-world:latest

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/

jakub@jakub:~/docker$
```

Dockerowe *hello-world*

```
$ docker run -it --rm supertest2014/nyan
```



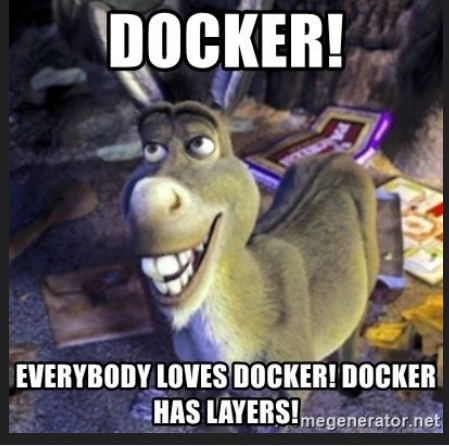
Obraz, a kontener

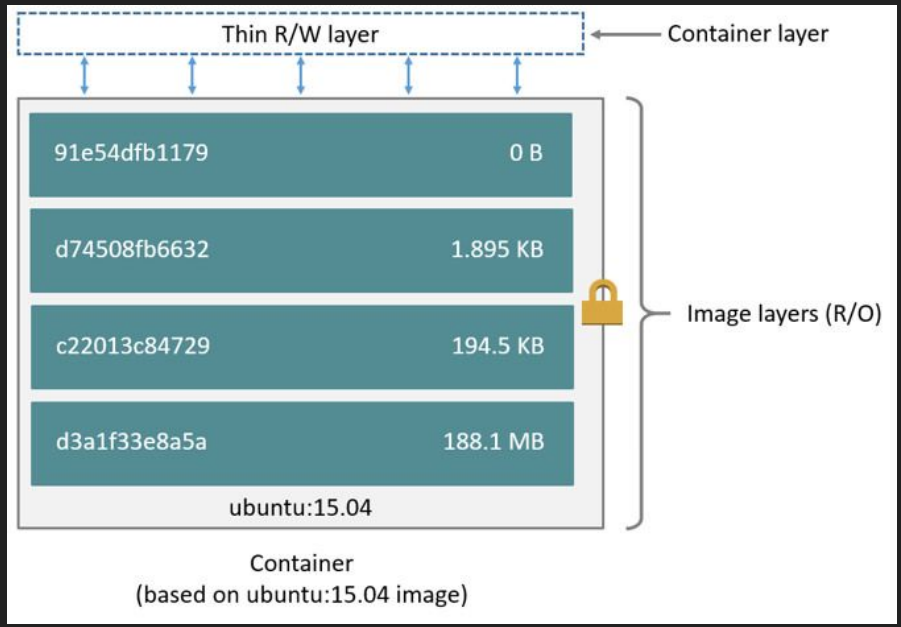
Docker image jest zbudowany z warstw.

Każda z warstw, oprócz ostatniej jest read-only.

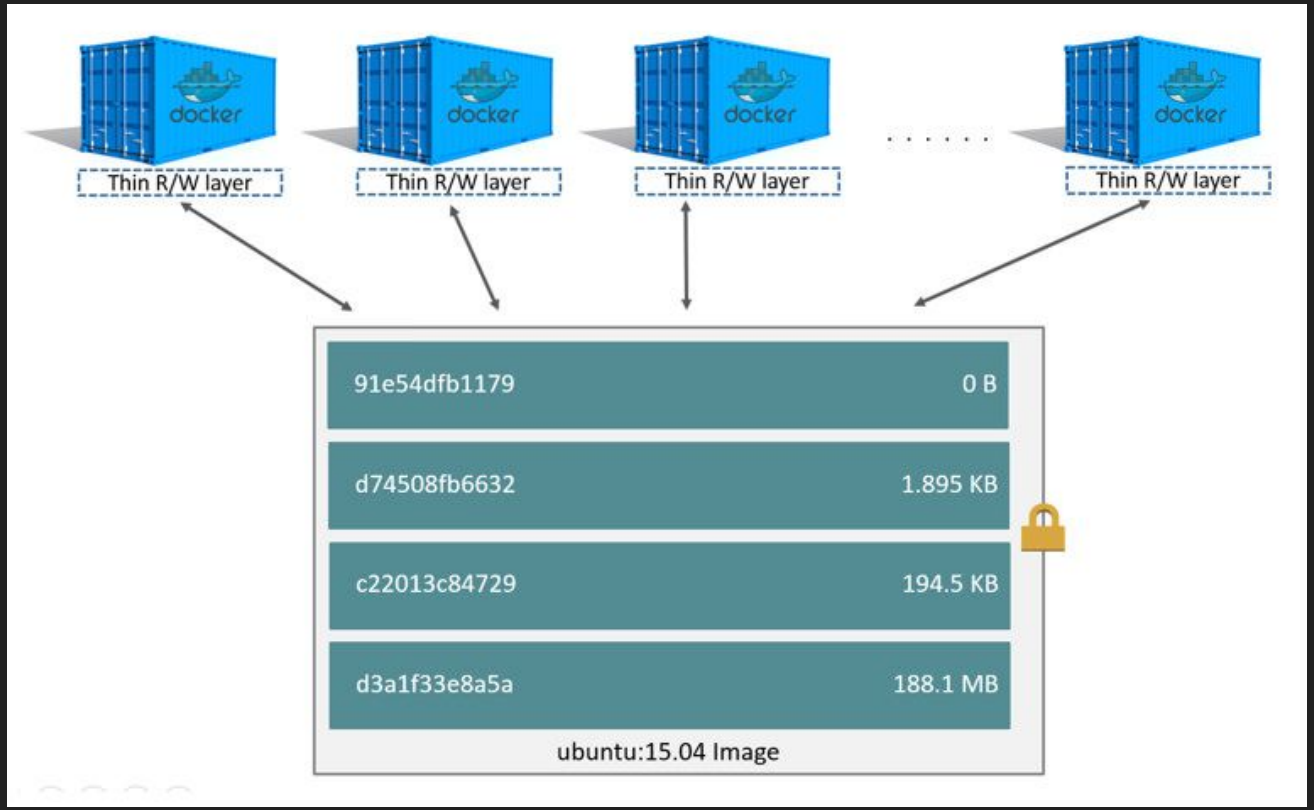
```
FROM ubuntu:15.04
COPY . /app
RUN make /app
CMD python /app/app.py
```

Każda z powyższych komend z *Dockerfile* tworzy nową warstwę.





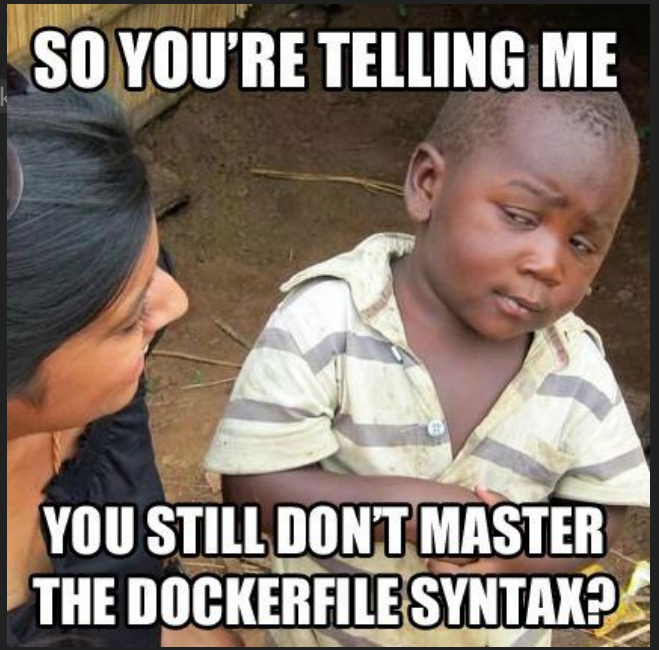
Obraz, a kontener



Dockerfile

Plik zawierający instrukcje wykonywane podczas budowania obrazu.

```
# Use an official Python runtime as a parent image
FROM python:2.7-slim
# The WORKDIR instruction sets the working directory for any
# RUN, CMD, ENTRYPOINT, COPY and ADD instructions that follow it in the Dock
WORKDIR /app
# copy all files from app directory into the container at /app
COPY app/* /app/
# Install any needed packages specified in requirements.txt
RUN pip install --trusted-host pypi.python.org -r requirements.txt
# Make port 80 available to the world outside this container
EXPOSE 80
# Define environment variable
ENV NAME World
# Run app.py when the container launches
CMD ["python", "app.py"]
```



Budowanie obrazu z Dockerfile

```
docker build -t repository:tag .
```

tworzy obraz na podstawie pliku Dockerfile z aktualnego folderu, zapisuje go jako *repository* z tagiem *tag.*

```
jakub@jakub:~/docker/simple_flask_app$ sudo docker build -t python2:simple-flask .
[sudo] password for jakub:
Sending build context to Docker daemon  6.656kB
Step 1/7 : FROM python:2.7-slim
 ---> 8559620b5b0d
Step 2/7 : WORKDIR /app
 ---> Using cache
 ---> 0b5a89d390b1
Step 3/7 : COPY app/* /app/
 ---> Using cache
 ---> f9334269f1f0
Step 4/7 : RUN pip install --trusted-host pypi.python.org -r requirements.txt
 ---> Using cache
 ---> f9ee8a97b0b3
Step 5/7 : EXPOSE 80
 ---> Using cache
 ---> c0245a344b7e
Step 6/7 : ENV NAME World
 ---> Using cache
 ---> f3b5c974a619
Step 7/7 : CMD ["python", "app.py"]
 ---> Using cache
 ---> 11b6a3c24507
Successfully built 11b6a3c24507
Successfully tagged python2:simple-flask
jakub@jakub:~/docker/simple_flask_app$ sudo docker image ls
REPOSITORY                TAG                 IMAGE ID            CREATED             SIZE
python2                   simple-flask        11b6a3c24507        29 minutes ago      131MB
```

Docker podstawowe komendy

`docker pull image_name` pobiera obraz z docker hub

```
jakub@jakub:~$ sudo docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
898c46f3b1a1: Pull complete
63366dfa0a50: Pull complete
041d4cd74a92: Pull complete
6e1bee0f8701: Pull complete
Digest: sha256:017eef0b616011647b269b5c65826e2e2ebddbe5d1f8c1e56b3599fb14fabec8
Status: Downloaded newer image for ubuntu:latest
jakub@jakub:~$ sudo docker image list
REPOSITORY                  TAG                 IMAGE ID            CREATED             SIZE
ubuntu                      latest              94e814e2efa8        12 days ago         88.9MB
<none>                      <none>              51cf1ffae2bf        2 weeks ago         5.53MB
jupyter/scipy-notebook      latest              2568eabb27a6        2 weeks ago         4.31GB
spellrun/theano-jupyter     latest              4679956efb2b        3 weeks ago         3.53GB
hello-world                 latest              fce289e99eb9        2 months ago        1.84kB
supertest2014/nyan          latest              3dd6111154fc        4 years ago         262MB
jakub@jakub:~$
```

Docker podstawowe komendy

`docker run image_name` uruchamia proces w odizolowanym kontenerze

```
jakub@jakub:~$ sudo docker run ubuntu
jakub@jakub:~$ sudo docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS                      PORTS               NAMES
9a6174ee43b2        ubuntu              "/bin/bash"         8 seconds ago       Exited (0) 6 seconds ago                        happy_raman
jakub@jakub:~$
```

`docker run -p 4000:80 image_name` przekierowanie portu 80 w kontenerze na port 4000 na hoście

`docker run -it --rm --name container_name image [startup command]`

- `-i` pozwala korzystać z STDIN
- `-t` uruchamia terminal
- `--rm` usuwa kontener po zakończeniu
- `--name` ustawia nazwę kontenera

Docker podstawowe komendy

`docker run -it --name simple_ubuntu ubuntu /bin/bash`

```
jakub@jakub:~/docker/simple_flask_app$ sudo docker run -it --name simple_ubuntu ubuntu /bin/bash
root@44967f69fc7d:/# exit
exit
jakub@jakub:~/docker/simple_flask_app$ sudo docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS                      PORTS               NAMES
44967f69fc7d        ubuntu              "/bin/bash"         5 seconds ago       Exited (0) 3 seconds ago                        simple_ubuntu
```

`docker run -it --rm --name simple_ubuntu ubuntu /bin/bash`

(utrata możliwości commitowania zmian po zakończeniu działania `$ docker commit container -m “message”`)

```
jakub@jakub:~/docker/simple_flask_app$ sudo docker run -it --rm --name simple_ubuntu ubuntu /bin/bash
root@3f7cff51fe9a:/# exit
exit
jakub@jakub:~/docker/simple_flask_app$ sudo docker ps -a
CONTAINER ID        IMAGE                  COMMAND             CREATED             STATUS                      PORTS               NAMES
de7d32b5ac4a        python2:simple-flask   "/bin/bash"         About an hour ago   Exited (0) 45 minutes ago                       gallant_morse
jakub@jakub:~/docker/simple_flask_app$
```

Docker podstawowe komendy

`docker image ls` wyświetla obrazy lokalne

```
jakub@jakub:~$ sudo docker image ls
REPOSITORY                TAG                 IMAGE ID            CREATED             SIZE
<none>                    <none>              51cf1ffae2bf        2 weeks ago         5.53MB
jupyter/scipy-notebook    latest              2568eabb27a6        2 weeks ago         4.31GB
spellrun/theano-jupyter   latest              4679956efb2b        3 weeks ago         3.53GB
alpine                    latest              caf27325b298        7 weeks ago         5.53MB
hello-world               latest              fce289e99eb9        2 months ago        1.84kB
supertest2014/nyan        latest              3dd6111154fc        4 years ago         262MB
jakub@jakub:~$
```

`docker image ps -a` wyświetla uruchomione i ‘zakończone’ kontenery

```
jakub@jakub:~/docker$ sudo docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS               NAMES
jakub@jakub:~/docker$ sudo docker run ubuntu
jakub@jakub:~/docker$ sudo docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS                      PORTS               NAMES
01c522f97788        ubuntu              "/bin/bash"         5 seconds ago       Exited (0) 4 seconds ago                        stoic_hodgkin
jakub@jakub:~/docker$
```

Docker podstawowe komendy

`docker stop container` wysyła sygnał “SIGTERM” do *container* (po 10 sec. “KILL”)

`docker kill container` wysyła sygnał “SIGKILL” do *container*

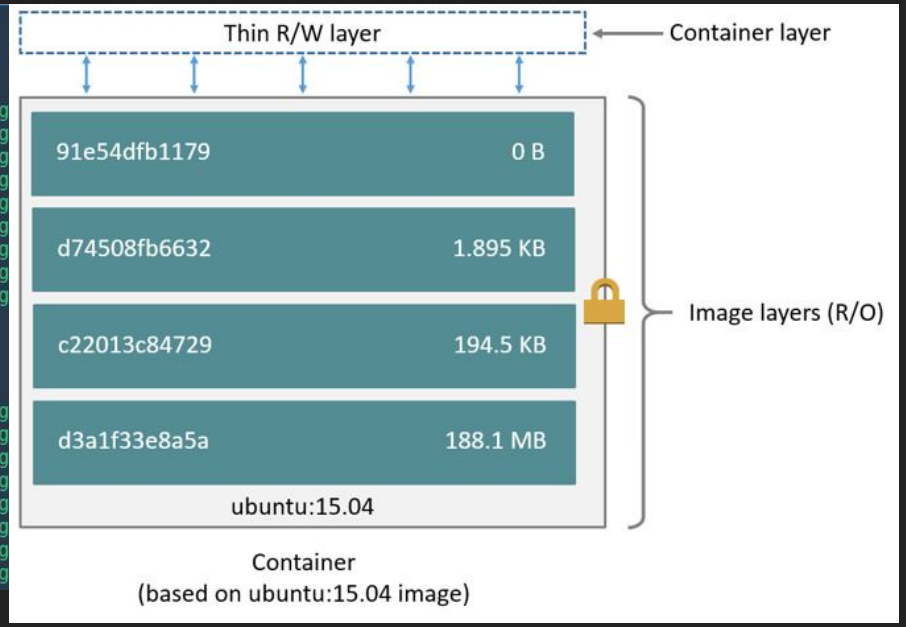
`docker kill -s signal container` wysyła sygnał *signal*

```
jakub@jakub:~$ sudo docker ps
CONTAINER ID        IMAGE               COMMAND               CREATED             STATUS              PORTS               NAMES
1859db13437f        supertest2014/nyan   "/bin/sh -c nyancat"   43 minutes ago      Up 43 minutes                           serene_neumann
jakub@jakub:~$ sudo docker kill serene_neumann
serene_neumann
jakub@jakub:~$ sudo docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS               NAMES
jakub@jakub:~$
```

Docker podstawowe komendy

`docker rm container` kończy działanie kontenera *container* nie pozwalając na zapisanie zmian.

```
jakub@jakub:~$ sudo docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED
jakub@jakub:~$ sudo docker ps -a
CONTAINER ID        IMAGE                     COMMAND                  CREATED
b2a9a12fd302        jupyter/scipy-notebook    "tini -g -- start-no…"   2 weeks ag
34cd245dc57a        jupyter/scipy-notebook    "tini -g -- start-no…"   2 weeks ag
6e3cfbde2bcd        jupyter/scipy-notebook    "tini -g -- start-no…"   2 weeks ag
dc9c12ea7cf2        jupyter/scipy-notebook    "tini -g -- start-no…"   2 weeks ag
177ee02ebdba        jupyter/scipy-notebook    "tini -g -- start-no…"   2 weeks ag
ec592d8a71f5        jupyter/scipy-notebook    "tini -g -- start-no…"   2 weeks ag
a4bec052dfac        jupyter/scipy-notebook    "tini -g -- --networ…"   2 weeks ag
85b092ce6472        jupyter/scipy-notebook    "tini -g -- start-no…"   2 weeks ag
4bd88ff31e15        hello-world               "/hello"                 2 weeks ag
jakub@jakub:~$ sudo docker rm blissful_bose
blissful_bose
jakub@jakub:~$ sudo docker ps -a
CONTAINER ID        IMAGE                     COMMAND                  CREATED
34cd245dc57a        jupyter/scipy-notebook    "tini -g -- start-no…"   2 weeks ag
6e3cfbde2bcd        jupyter/scipy-notebook    "tini -g -- start-no…"   2 weeks ag
dc9c12ea7cf2        jupyter/scipy-notebook    "tini -g -- start-no…"   2 weeks ag
177ee02ebdba        jupyter/scipy-notebook    "tini -g -- start-no…"   2 weeks ag
ec592d8a71f5        jupyter/scipy-notebook    "tini -g -- start-no…"   2 weeks ag
a4bec052dfac        jupyter/scipy-notebook    "tini -g -- --networ…"   2 weeks ag
85b092ce6472        jupyter/scipy-notebook    "tini -g -- start-no…"   2 weeks ag
4bd88ff31e15        hello-world               "/hello"                 2 weeks ag
```



Docker podstawowe komendy

`docker rm -f container` flaga -f musi zostać użyta do zamknięcia działającego kontenera

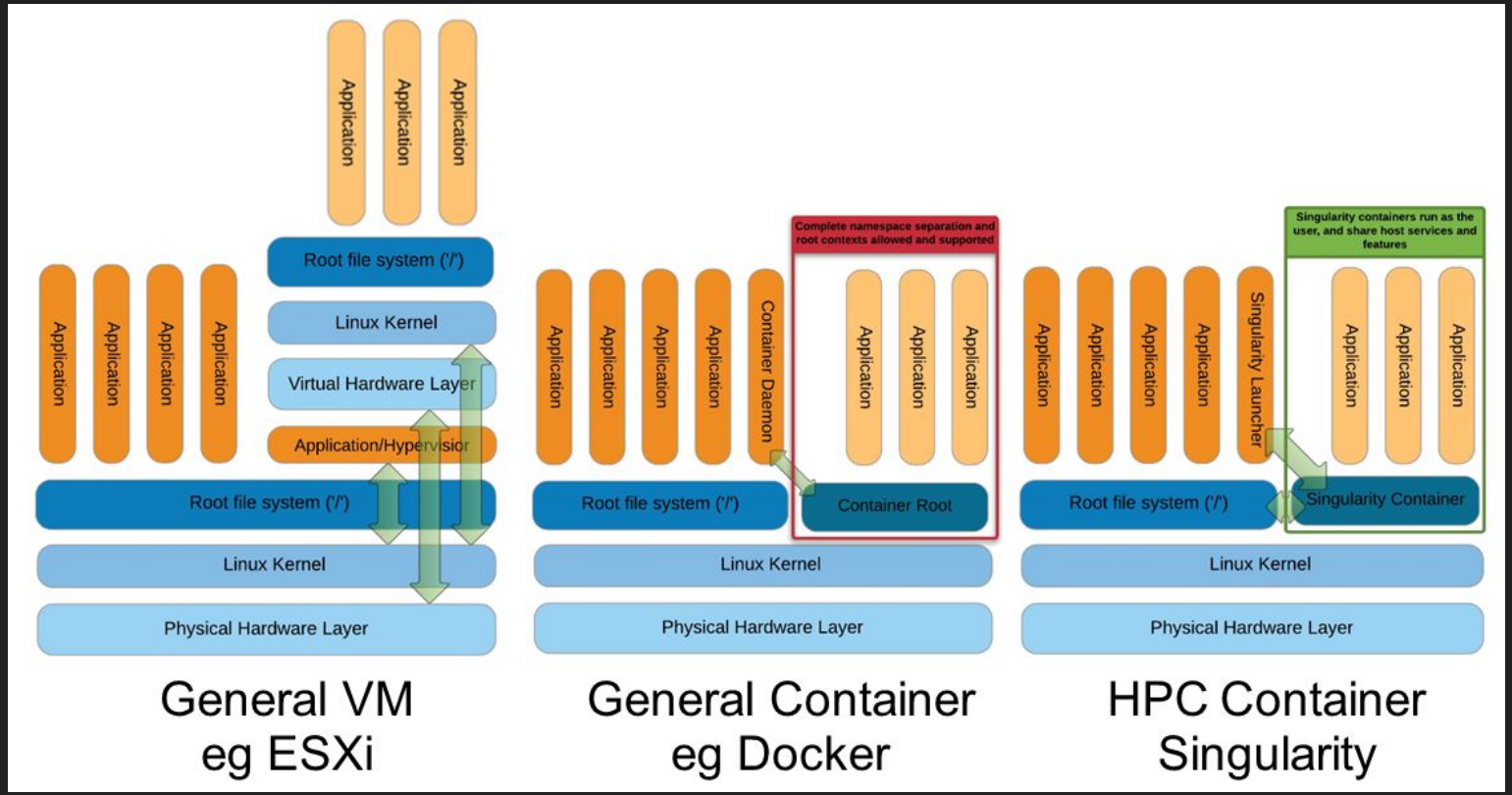
```
jakub@jakub:~$ sudo docker ps
CONTAINER ID        IMAGE                  COMMAND                  CREATED             STATUS              PORTS                    NAMES
c80ab9f43643        jupyter/scipy-notebook   "tini -g -- start-no…"   4 minutes ago       Up About a minute   0.0.0.0:8888->8888/tcp   admiring_chatelet
jakub@jakub:~$ sudo docker rm admiring_chatelet
Error response from daemon: You cannot remove a running container c80ab9f43643263dc9ed69bf5309482a78ab98f637d875a259e4d4707d15234f. Stop the container before attempting removal or force remove
jakub@jakub:~$
jakub@jakub:~$ sudo docker rm -f admiring_chatelet
admiring_chatelet
jakub@jakub:~$ sudo docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS               NAMES
jakub@jakub:~$
```

Docker podstawowe komendy

`docker rmi image` komenda usuwa podany *image*

```
jakub@jakub:~$ sudo docker image list
[sudo] password for jakub:
REPOSITORY                  TAG                 IMAGE ID            CREATED             SIZE
<none>                      <none>              51cf1ffae2bf        2 weeks ago         5.53MB
jupyter/scipy-notebook      latest              2568eabb27a6        2 weeks ago         4.31GB
spellrun/theano-jupyter     latest              4679956efb2b        3 weeks ago         3.53GB
alpine                      latest              caf27325b298        7 weeks ago         5.53MB
hello-world                 latest              fce289e99eb9        2 months ago        1.84kB
supertest2014/nyan          latest              3dd6111154fc        4 years ago         262MB
jakub@jakub:~$ sudo docker rmi alpine
Untagged: alpine:latest
Untagged: alpine@sha256:b3dbf31b77fd99d9c08f780ce6f5282aba076d70a513a8be859d8d3a4d0c92b8
jakub@jakub:~$ sudo docker image list
REPOSITORY                  TAG                 IMAGE ID            CREATED             SIZE
<none>                      <none>              51cf1ffae2bf        2 weeks ago         5.53MB
jupyter/scipy-notebook      latest              2568eabb27a6        2 weeks ago         4.31GB
spellrun/theano-jupyter     latest              4679956efb2b        3 weeks ago         3.53GB
hello-world                 latest              fce289e99eb9        2 months ago        1.84kB
supertest2014/nyan          latest              3dd6111154fc        4 years ago         262MB
jakub@jakub:~$
```

Docker vs Singularity



Shifter vs Charlie Cloud vs Docker vs Singularity

	Shifter	Charlie Cloud	Docker	Singularity
Privilege model	SUID	UserNS	Root Daemon	SUID/UserNS
Support current production Linux distros	Yes	No	No	Yes
Internal image build/boostrap	No*	No*	No**	Yes
No privileged or trusted daemons	Yes	Yes	No	Yes
No additional network configurations	Yes	Yes	No	Yes
No additional hardware	Maybe	Yes	Maybe	Yes
Access to host filesystem	Yes	Yes	Yes***	Yes
Native support for GPU	No	No	No	Yes
Native support for InfiniBand	Yes	Yes	No	Yes
Native support for MPI	Yes	Yes	No	Yes
Works with all schedulers	No	Yes	No	Yes
Designed for general scientific use cases	Yes	No	No	Yes
Contained environment has coorect perms	Yes	No	Yes	Yes
Containers are portable, unmodified by use	No	No	No	Yes
Trivial HPC install (one package, zero conf)	No	Yes	Yes	Yes
Admins can control and limit capabilities	Yes	No	No	Yes

* Relies on Docker
** Depends on upstream
*** With security implications

Docker vs Singularity

Docker	Singularity
Dockerfile	definition_file.def
docker pull	singularity pull
docker build	singularity build
docker run	singularity instance start
docker run image_name command	singularity exec command
docker run -it image_name /bin/bash	singularity shell
docker ps	singularity instance list
	singularity test

Docker vs Singularity

- obraz == jeden plik, brak warstw
- inside user == outside user != root

  „If you want to be root inside the container, you must first be root outside the container“
- Brak elewacji uprawnień umożliwia łatwy deployment na HPC

.def file

- header - zawiera podstawowy os (image) oraz sposób jego pozyskania (docker hub, singularity hub, localimage, …)
- sekcje:
  - %setup - instrukcje wykonywane na hoście po pobraniu obrazu podstawowego
  - %files - kopiowanie plików z hosta do obrazu
  - %environment - zmienne dla runtime
  - %post - instrukcje wykonywane w obrazie po pobraniu obrazu podstawowego
  - %runscript - instrukcje wykonywane podczas wywołania *singularity run image.sif*
  - %startscript - instrukcje wykonywane podczas *instance start*
  - %test
  - %labels
  - %help

```
Bootstrap: library
From: ubuntu:18.04

%setup
    touch /file1
    touch ${SINGULARITY_ROOTFS}/file2

%files
    /file1
    /file1 /opt

%environment
    export LISTEN_PORT=12345
    export LC_ALL=C

%post
    apt-get update && apt-get install -y netcat
    NOW=`date`
    echo "export NOW=\"${NOW}\"" >> $SINGULARITY_ENVIRONMENT

%runscript
    echo "Container was created $NOW"
    echo "Arguments received: $*"
    exec echo "$@"

%startscript
    nc -lp $LISTEN_PORT

%test
    grep -q NAME=\"Ubuntu\" /etc/os-release
    if [ $? -eq 0 ]; then
        echo "Container base is Ubuntu as expected."
    else
        echo "Container base is not Ubuntu."
    fi

%labels
    Author d@sylabs.io
    Version v0.0.1

%help
    This is a demo container used to illustrate a def file that uses all
    supported sections.
```

Singularity build

```
sudo singularity build simple_ubuntu18.04.sif simple_ubuntu18.04.def
```

buduje obraz singularity z pliku .def

```
jakub@jakub:~/singularity/simple_singularity_app$ sudo singularity build simple_ubuntu18.04.sif simple_ubuntu18.04.def
WARNING: Authentication token file not found : Only pulls of public images will succeed
INFO:    Starting build...
INFO:    Downloading library image
 26.68 MiB / 26.68 MiB [==========================================================================================================================================] 100.00% 383.92 KiB/s 1m11s
INFO:    Running setup scriptlet
+ touch /file1
+ touch /tmp/sbuild-096395742/fs/file2
INFO:    Copying /file1 to /tmp/sbuild-096395742/fs/file1
INFO:    Copying /file1 to /tmp/sbuild-096395742/fs/opt
INFO:    Running post scriptlet
+ apt-get update
+ date
+ NOW=Sun Mar 24 16:26:39 UTC 2019
+ echo export NOW="Sun Mar 24 16:26:39 UTC 2019"
INFO:    Running test scriptlet
+ grep -q NAME="Ubuntu" /etc/os-release
+ [ 0 -eq 0 ]
+ echo Container base is Ubuntu as expected.
Container base is Ubuntu as expected.
INFO:    Adding help info
INFO:    Adding labels
INFO:    Adding environment to container
INFO:    Adding startscript
INFO:    Adding runscript
INFO:    Adding testscript
INFO:    Creating SIF file...
INFO:    Build complete: simple_ubuntu18.04.sif
jakub@jakub:~/singularity/simple_singularity_app$ ls -la
total 59280
drwxrwxrwx 1 root root      304 Mar 24 17:26 .
drwxrwxrwx 1 root root      176 Mar 24 17:22 ..
-rwxrwxrwx 1 root root      816 Mar 24 17:18 simple_ubuntu18.04.def
-rwxrwxrwx 1 root root 60698624 Mar 24 17:26 simple_ubuntu18.04.sif
jakub@jakub:~/singularity/simple_singularity_app$
```

Singularity run

```
singularity run simple_ubuntu18.04.sif
```

```
jakub@jakub:~/singularity/simple_singularity_app$ singularity run simple_ubuntu18.04.sif  hello world
Container was created Sun Mar 24 16:26:39 UTC 2019
Arguments received: hello world
hello world
```

```
singularity test simple_ubuntu18.04.sif
```

```
jakub@jakub:~/singularity/simple_singularity_app$ singularity test simple_ubuntu18.04.sif
Container base is Ubuntu as expected.
```

```
singularity instance start simple_ubuntu18.04.sif simple_ubuntu
```

```
jakub@jakub:~/singularity/simple_singularity_app$ singularity instance start simple_ubuntu18.04.sif simple_ubuntu
INFO:    instance started successfully
jakub@jakub:~/singularity/simple_singularity_app$ singularity instance list
INSTANCE NAME    PID      IMAGE
simple_ubuntu    32482    /home/jakub/singularity/simple_singularity_app/simple_ubuntu18.04.sif
jakub@jakub:~/singularity/simple_singularity_app$ singularity instance stop simple_ubuntu
Stopping simple_ubuntu instance of /home/jakub/singularity/simple_singularity_app/simple_ubuntu18.04.sif (PID=32482)
jakub@jakub:~/singularity/simple_singularity_app$
```

Singularity shell

`singularity shell simple_ubuntu18.04.sif`

Ma możliwość modyfikacji danych na hoście - by default!

```
jakub@jakub:~/singularity/simple_singularity_app$ singularity shell simple_ubuntu18.04.sif
Singularity simple_ubuntu18.04.sif:~/singularity/simple_singularity_app> ls -la
total 59280
drwxrwxrwx 1 root  root         0 Mar 24 17:26 .
drwxr-xr-x 3 jakub jakub       60 Mar 24 19:20 ..
-rwxrwxrwx 1 root  root       816 Mar 24 17:18 simple_ubuntu18.04.def
-rwxrwxrwx 1 root  root  60698624 Mar 24 17:26 simple_ubuntu18.04.sif
Singularity simple_ubuntu18.04.sif:~/singularity/simple_singularity_app> echo "hello world" > some_file
Singularity simple_ubuntu18.04.sif:~/singularity/simple_singularity_app> exit
exit
jakub@jakub:~/singularity/simple_singularity_app$ cat some_file
hello world
jakub@jakub:~/singularity/simple_singularity_app$
```

Singularity - cuda

`--nv` enable experimental Nvidia support

Może być stosowane dla *instance start, shell, run.*

Wymaga wcześniejszej instalacji driverów od nvidii.

Docker dla ML

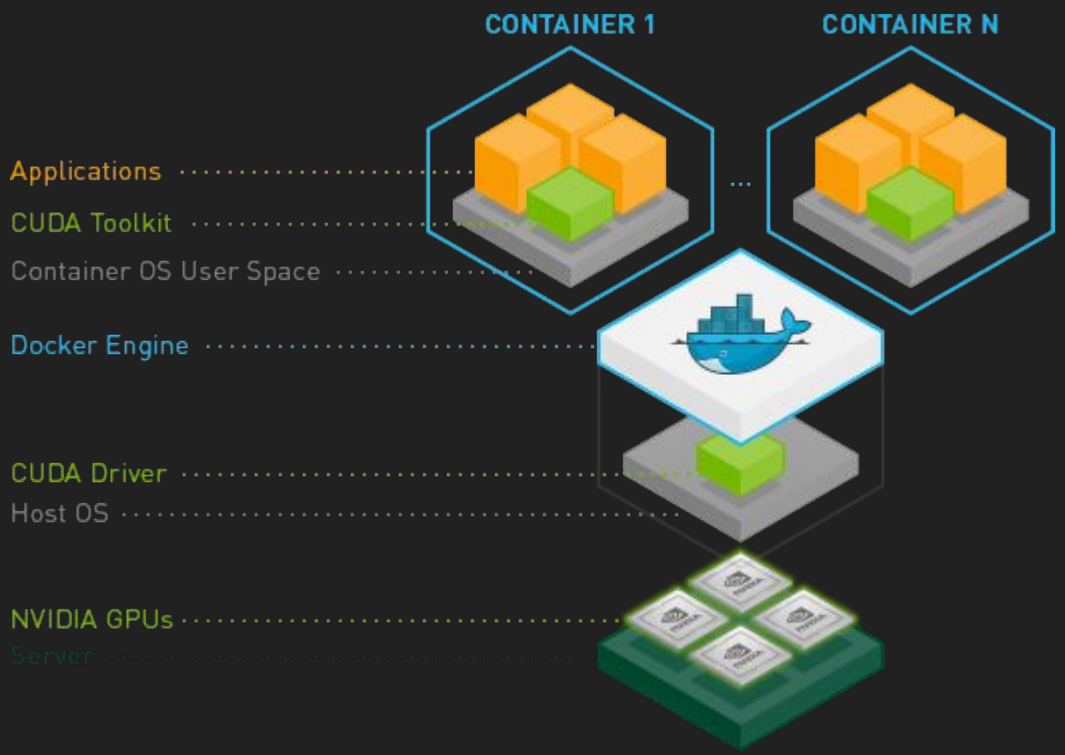
Gotowe obrazy:

```
docker pull tensorflow/tensorflow
```

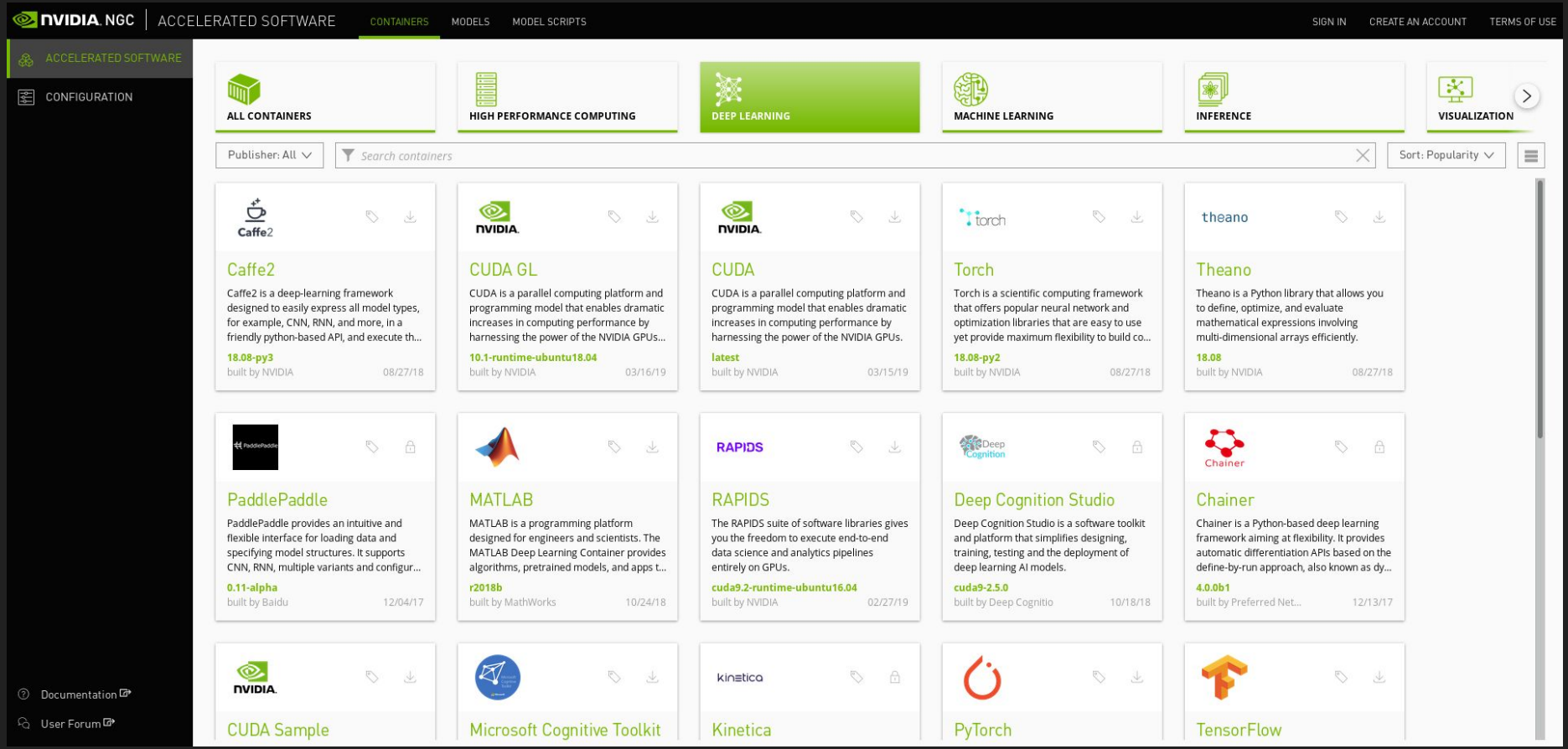
```
docker pull jupyter/datascience-notebook
```

```
docker pull nvidia/cuda
```

https://github.com/floydhub/dl-docker (TensorFlow, Theano, Torch, Caffe, etc.)



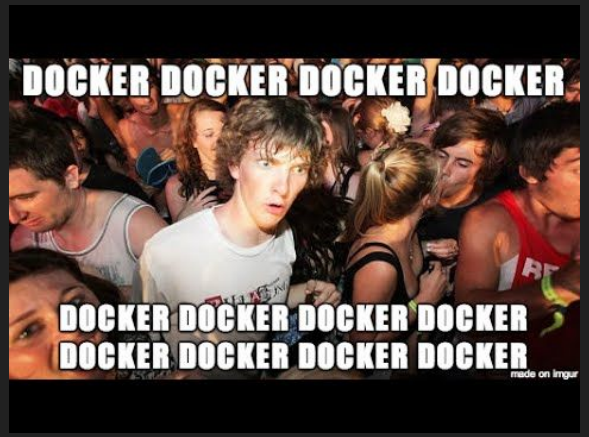
https://github.com/NVIDIA/nvidia-docker


NVIDIA NGC | ACCELERATED SOFTWARE
CONTAINERS
MODELS
MODEL SCRIPTS
SIGN IN
CREATE AN ACCOUNT
TERMS OF USE
ACCELERATED SOFTWARE
CONFIGURATION
ALL CONTAINERS
HIGH PERFORMANCE COMPUTING
DEEP LEARNING
MACHINE LEARNING
INFERENCE
VISUALIZATION
Publisher: All
Search containers
Sort: Popularity
Caffe2
Caffe2 is a deep-learning framework designed to easily express all model types, for example, CNN, RNN, and more, in a friendly python-based API, and execute th...
18.08-py3
built by NVIDIA
08/27/18
CUDA GL
CUDA is a parallel computing platform and programming model that enables dramatic increases in computing performance by harnessing the power of the NVIDIA GPUs...
10.1-runtime-ubuntu18.04
built by NVIDIA
03/16/19
CUDA
CUDA is a parallel computing platform and programming model that enables dramatic increases in computing performance by harnessing the power of the NVIDIA GPUs.
latest
built by NVIDIA
03/15/19
Torch
Torch is a scientific computing framework that offers popular neural network and optimization libraries that are easy to use yet provide maximum flexibility to build co...
18.08-py2
built by NVIDIA
08/27/18
Theano
Theano is a Python library that allows you to define, optimize, and evaluate mathematical expressions involving multi-dimensional arrays efficiently.
18.08
built by NVIDIA
08/27/18
PaddlePaddle
PaddlePaddle provides an intuitive and flexible interface for loading data and specifying model structures. It supports CNN, RNN, multiple variants and configur...
0.11-alpha
built by Baidu
12/04/17
MATLAB
MATLAB is a programming platform designed for engineers and scientists. The MATLAB Deep Learning Container provides algorithms, pretrained models, and apps t...
r2018b
built by MathWorks
10/24/18
RAPIDS
The RAPIDS suite of software libraries gives you the freedom to execute end-to-end data science and analytics pipelines entirely on GPUs.
cuda9.2-runtime-ubuntu16.04
built by NVIDIA
02/27/19
Deep Cognition Studio
Deep Cognition Studio is a software toolkit and platform that simplifies designing, training, testing and the deployment of deep learning AI models.
cuda9-2.5.0
built by Deep Cognitio
10/18/18
Chainer
Chainer is a Python-based deep learning framework aiming at flexibility. It provides automatic differentiation APIs based on the define-by-run approach, also known as dy...
4.0.0b1
built by Preferred Net...
12/13/17
CUDA Sample
Microsoft Cognitive Toolkit
Kinetica
PyTorch
TensorFlow
Documentation
User Forum

TODO

```
$ git clone https://github.com/jakub-tomczak/PatternsImageClassifier.git patterns

$ cd patterns

$ sudo docker build -t python:patterns-recognition .

$ sudo docker run -it python:patterns-recognition /bin/bash

$ python main.py
```

Pytania?



Docker docker docker

- Docker-compose
- Kubernetes, Docker Swarm
- http://www.qnib.org/data/isc2018/ISC2018_Gantikow_ContainerTechnology-EngineArchitecture101.pdf

Dziękuję za uwagę

Źródła

- Singularity - https://www.sylabs.io
- Porównanie performance'u - https://domino.research.ibm.com/library/cyberdig.nsf/papers/0929052195DD819C85257D2300681E7B/$File/rc25482.pdf
- Docker docs - https://docs.docker.com/
- http://gpucomputing.shef.ac.uk/education/creating_gpu_singularity/
- https://hub.docker.com/