



# Wprowadzenie do technologii Docker

Mateusz Rogowski

# Maszyny Wirtualne

Maszyna wirtualna kontroluje wszystkie odwołania uruchamianego programu bezpośrednio do sprzętu lub systemu operacyjnego i zapewnia ich obsługę.

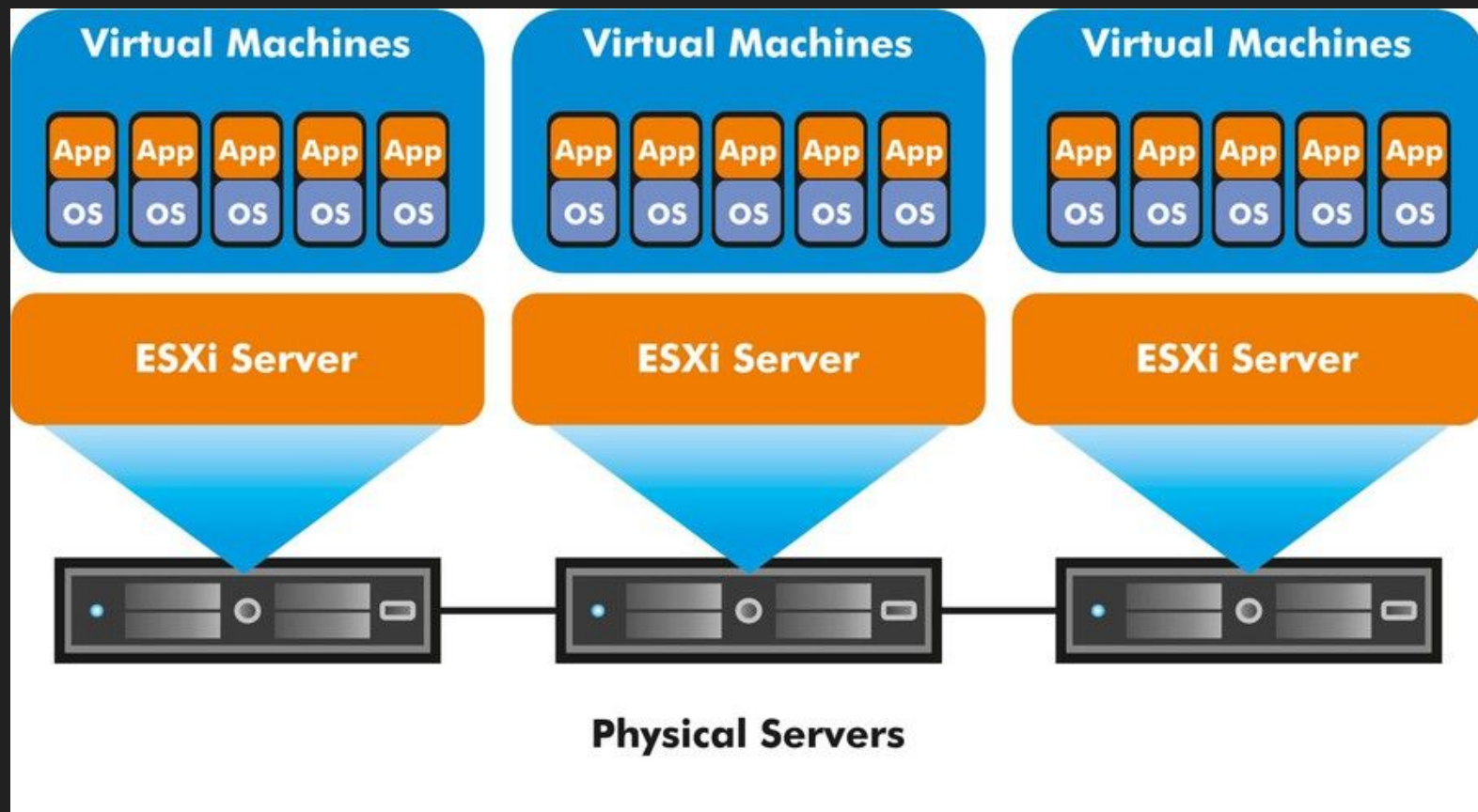
**Dzięki temu program uruchomiony na maszynie wirtualnej „myśli”, że działa na rzeczywistym sprzęcie, podczas gdy w istocie pracuje na sprzęcie wirtualnym, „udawanym” przez odpowiednie oprogramowanie (maszynę wirtualną).**

# Maszyny Wirtualne

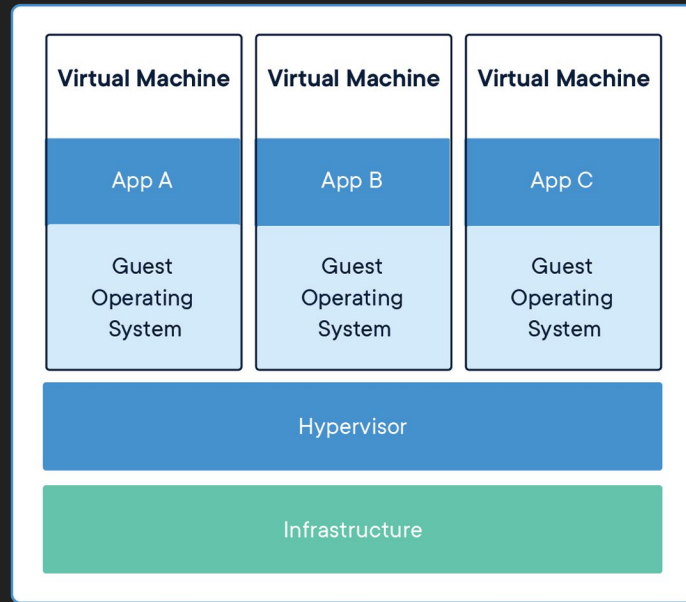
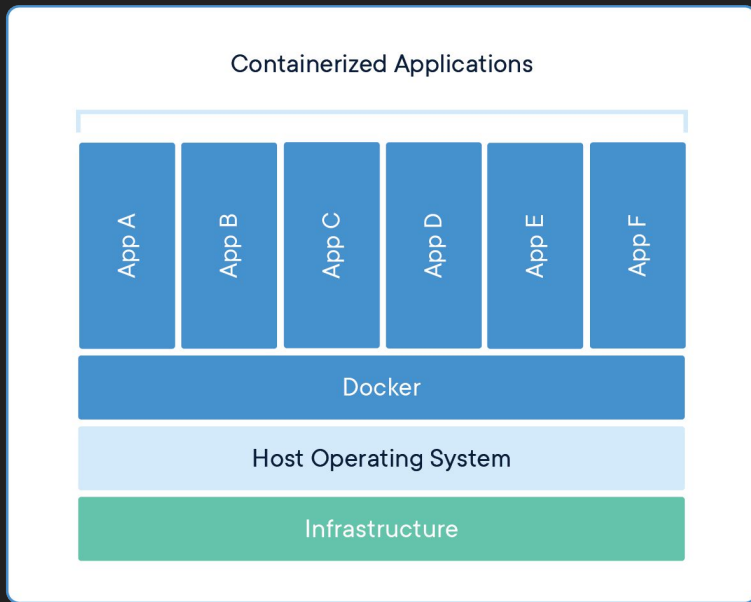
Maszyna wirtualna kontroluje wszystkie odwołania uruchamianego programu bezpośrednio do sprzętu lub systemu operacyjnego i zapewnia ich obsługę.

**Dzięki temu program uruchomiony na maszynie wirtualnej „myśli”, że działa na rzeczywistym sprzęcie, podczas gdy w istocie pracuje na sprzęcie wirtualnym, „udawanym” przez odpowiednie oprogramowanie (maszynę wirtualną).**

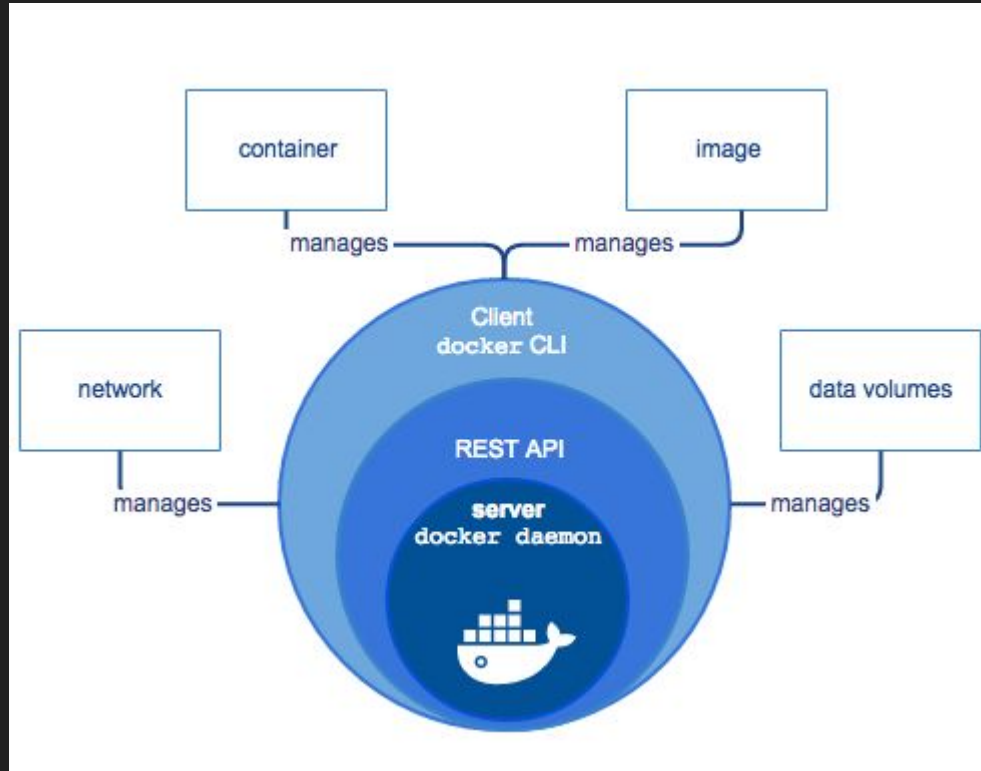
- VMware
- Hyper-V
- VirtualBox



# Containers vs. VMs



# Docker



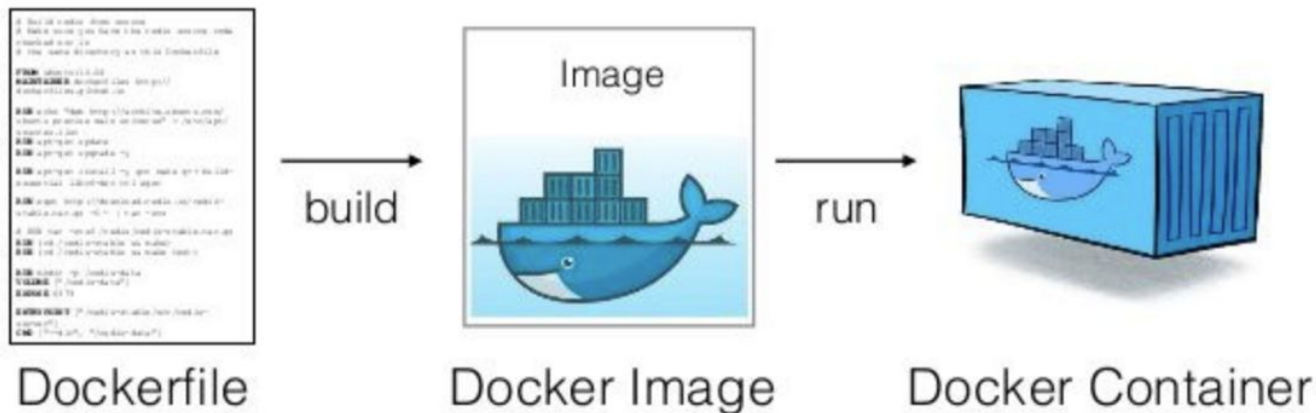
# Podstawowe elementy

**DOCKERFILE** - Docker can build images automatically by reading the instructions from a Dockerfile. A Dockerfile is a text document that contains all the commands a user could call on the command line to assemble an image.

**IMAGE** - An image is a read-only template with instructions for creating a Docker container. Often, an image is based on another image, with some additional customization.

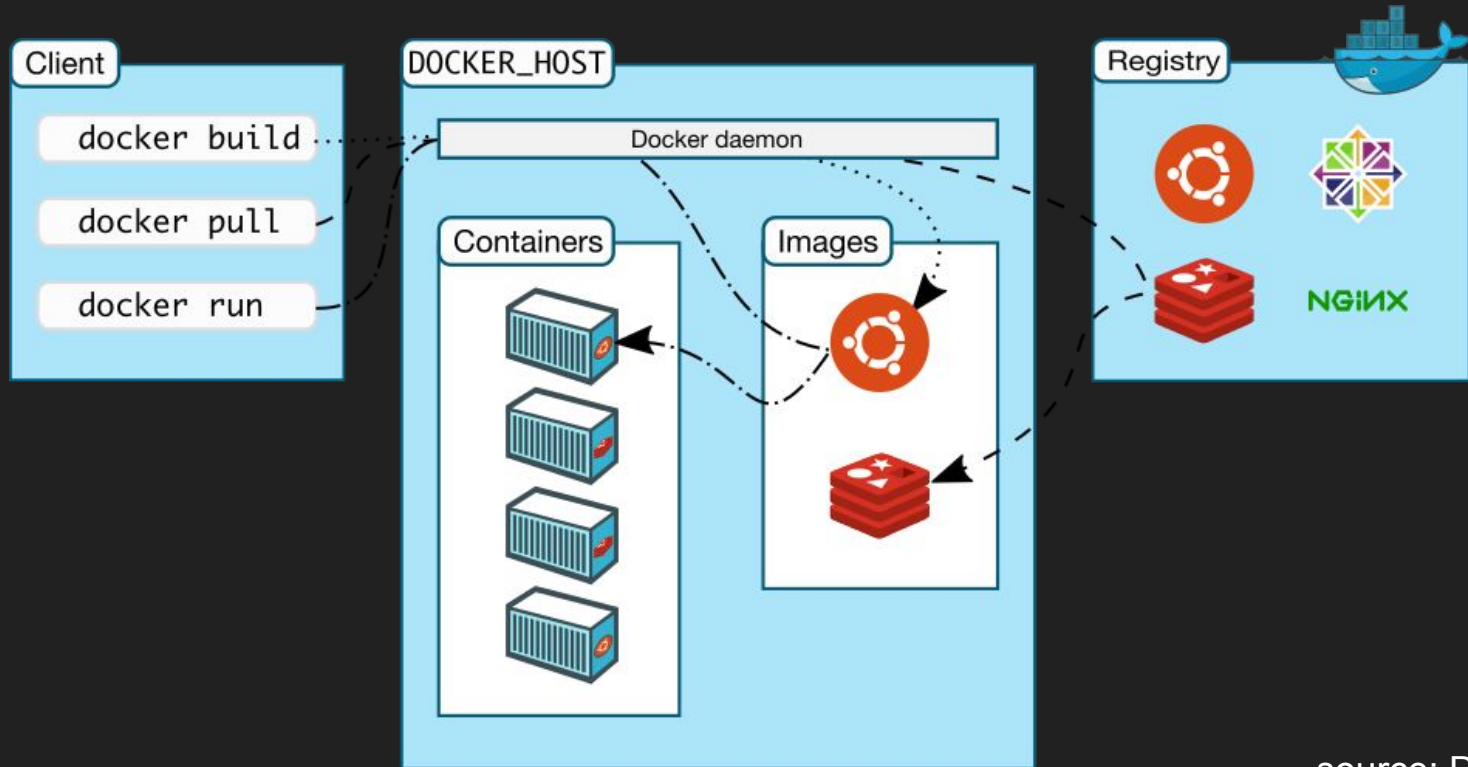
**CONTAINER** - A container is a runnable instance of an image. You can create, start, stop, move, or delete a container using the Docker API or CLI.

# Docker - flow





# Architektura



source: Docker

# Docker - Komendy

**ps** - List containers

**run** - Run a command in a new container

**exec {container\_id}** - Run a command in a running container

**exec -it {container\_id} bash** - Run bash in a running container

**logs {container\_id}** - Fetch the logs of a container

**build** - Build an image from a Dockerfile

**kill {container\_id}** - Kill one or more running containers

# Docker

Live demo - <https://hub.docker.com/>

# Docker

Live demo - <https://hub.docker.com/>

```
docker run --name some-drupal -p 8080:80 -d drupal
```

```
http://localhost:8080
```

```
docker ps
```

```
docker logs {container-id}
```

```
docker logs -f {container-id}
```

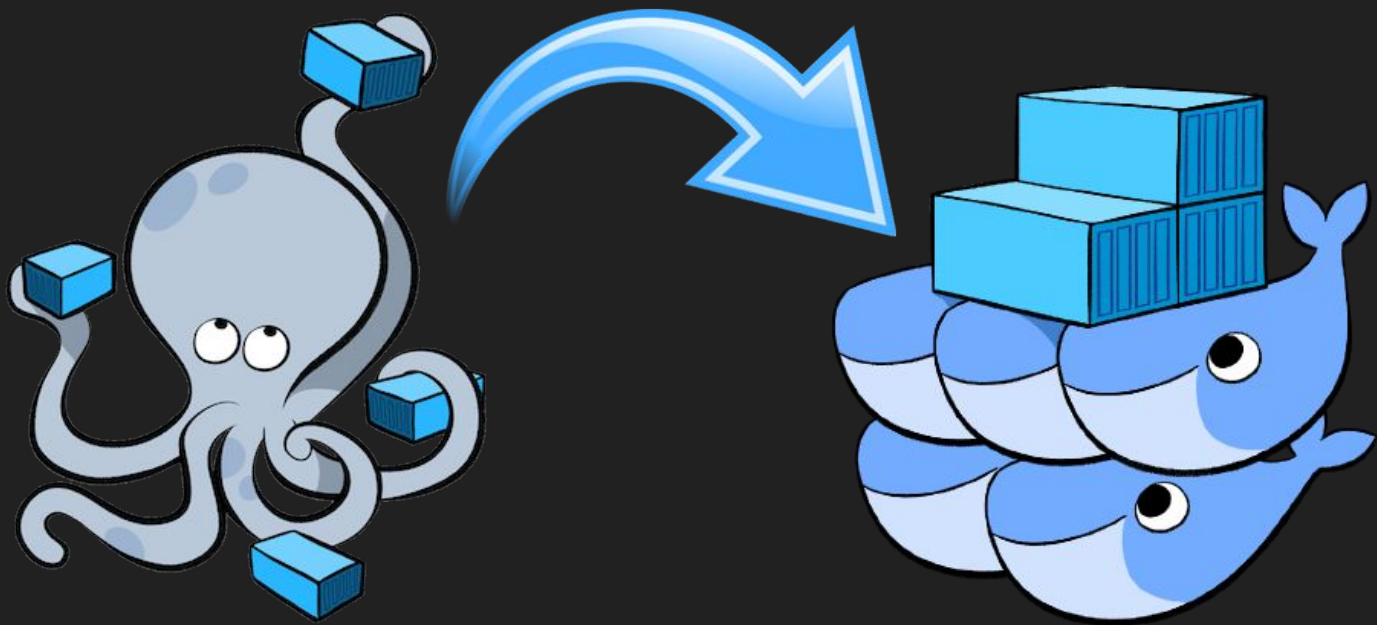
```
docker exec -it {container-id} bash
```

```
docker kill {container-id}
```

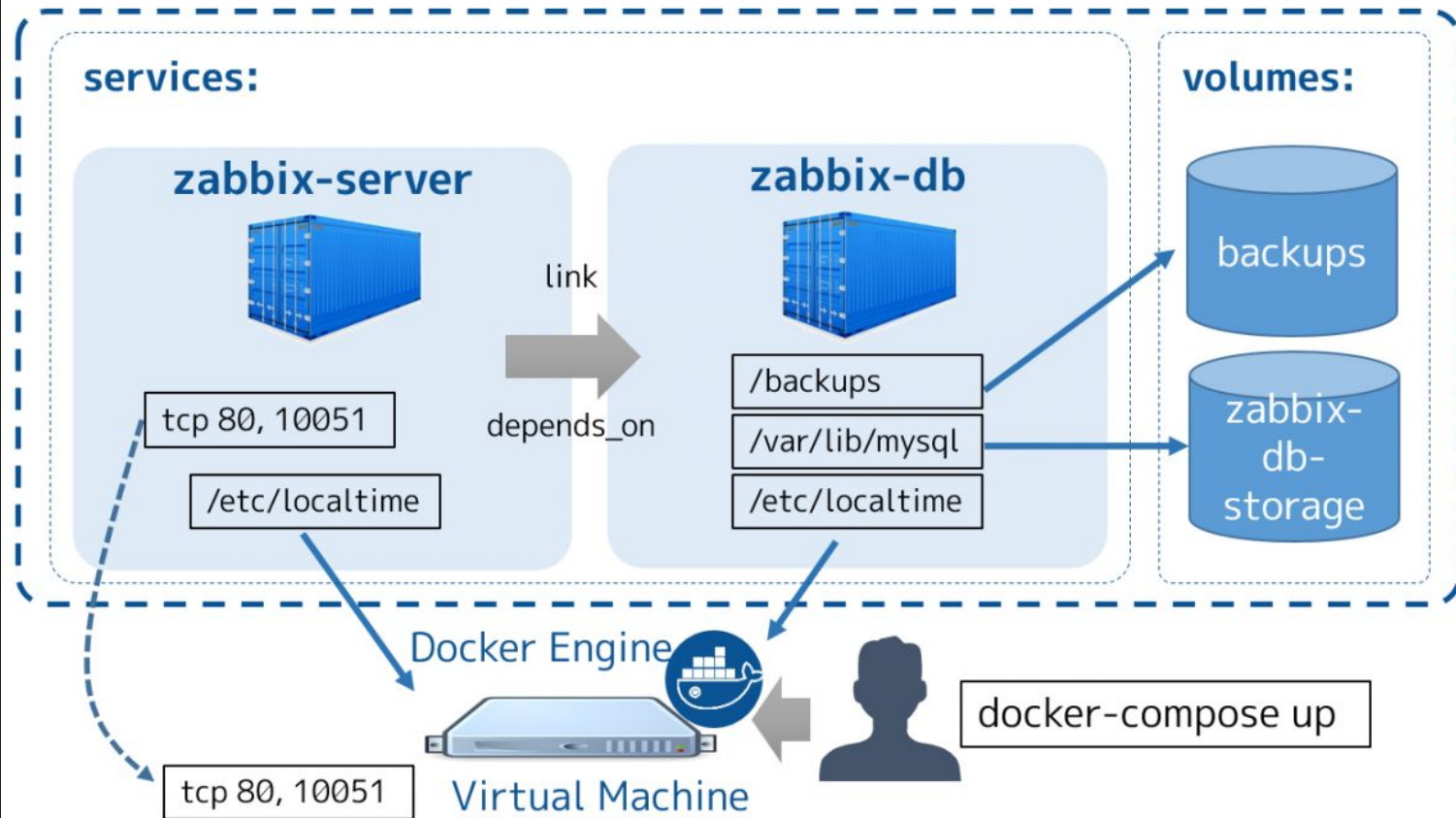
Docker



Compose



## docker-compose.yml (v2 format)



# docker-compose - Komendy

**build** - Build docker compose stack

**up** - Start docker compose stack

**stop** - Stop docker compose stack



# Django project - przykłady

Live demo