

# DOCKER

---

**WPROWADZENIE DO KONTENERÓW  
DLA PROGRAMISTÓW**

# MACIEJ BIENIA

- ♦ **maciej.bienia@gmail.com**
- ♦ **slack: @maciej.bienia**
- ♦ **Software Developer**
- ♦ **SysAdmin**
- ♦ **Linux and Open Source enthusiast**

**Dlaczego**

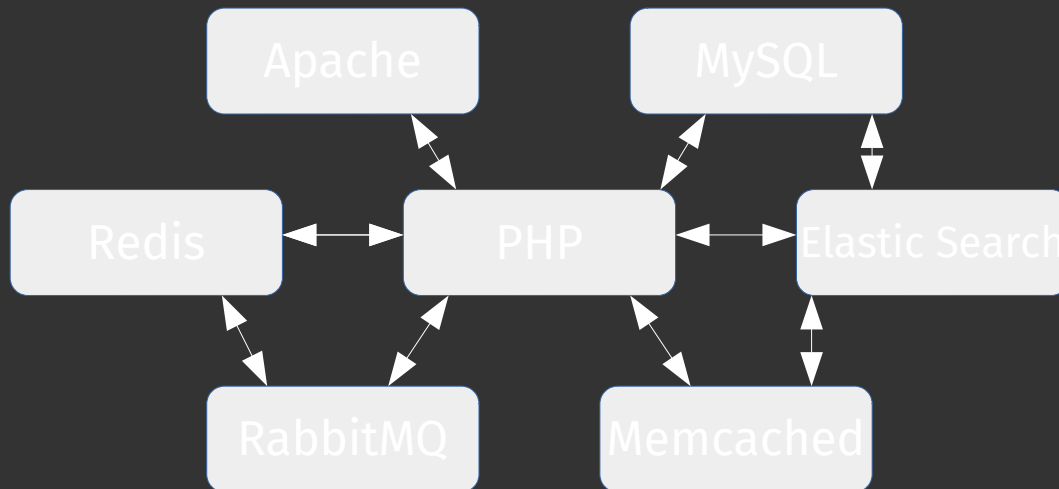
**DOCKER?**

# Dlaczego **DOCKER**?



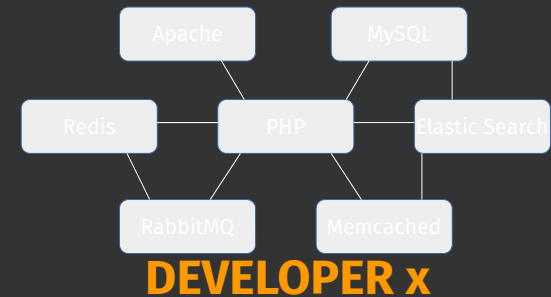
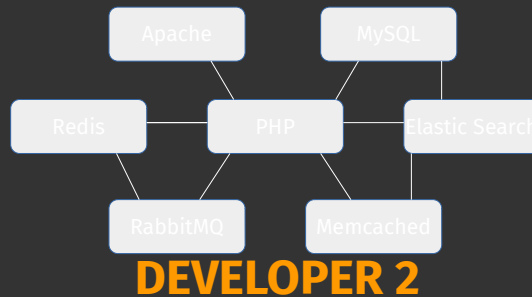
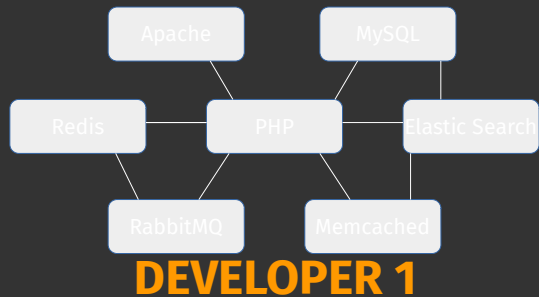
# Dlaczego **DOCKER**?

## ZŁOŻONE ZALEŻNOŚCI W PROJEKCIE

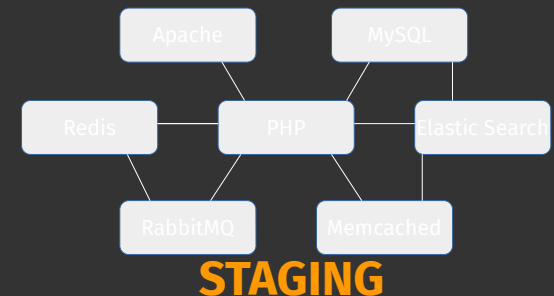
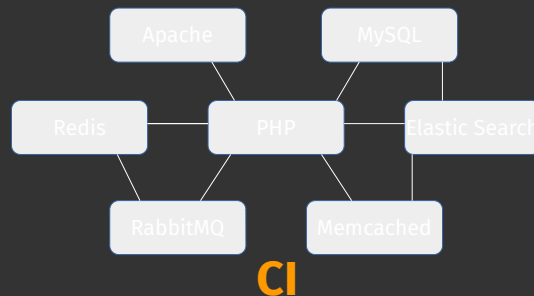
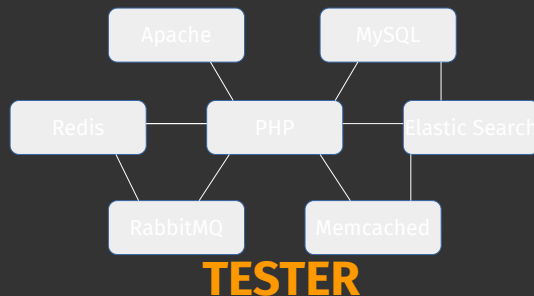


## Dlaczego **DOCKER**?

























### WIELE RÓWNOCZESNYCH PROJEKTÓW



**\*N projektów**



# Dlaczego **DOCKER**?

	Dev	Stage	Prod	CI
nginx				
MySQL				
PHP				
Redis				
Elastic Search				
Rabbit MQ				

# CO ZYSKUJEMY?

- ♦ **budujemy raz,  
używamy gdzie chcemy**



## CO ZYSKUJEMY?

- ♦ **nie występują problemy z zależnościami oraz z niekompatybilnością zależności**

# CO ZYSKUJEMY?

- ♦ **brak narzutów VM**

# CO ZYSKUJEMY?

- ♦ pełna automatyzacja

# CO ZYSKUJEMY?

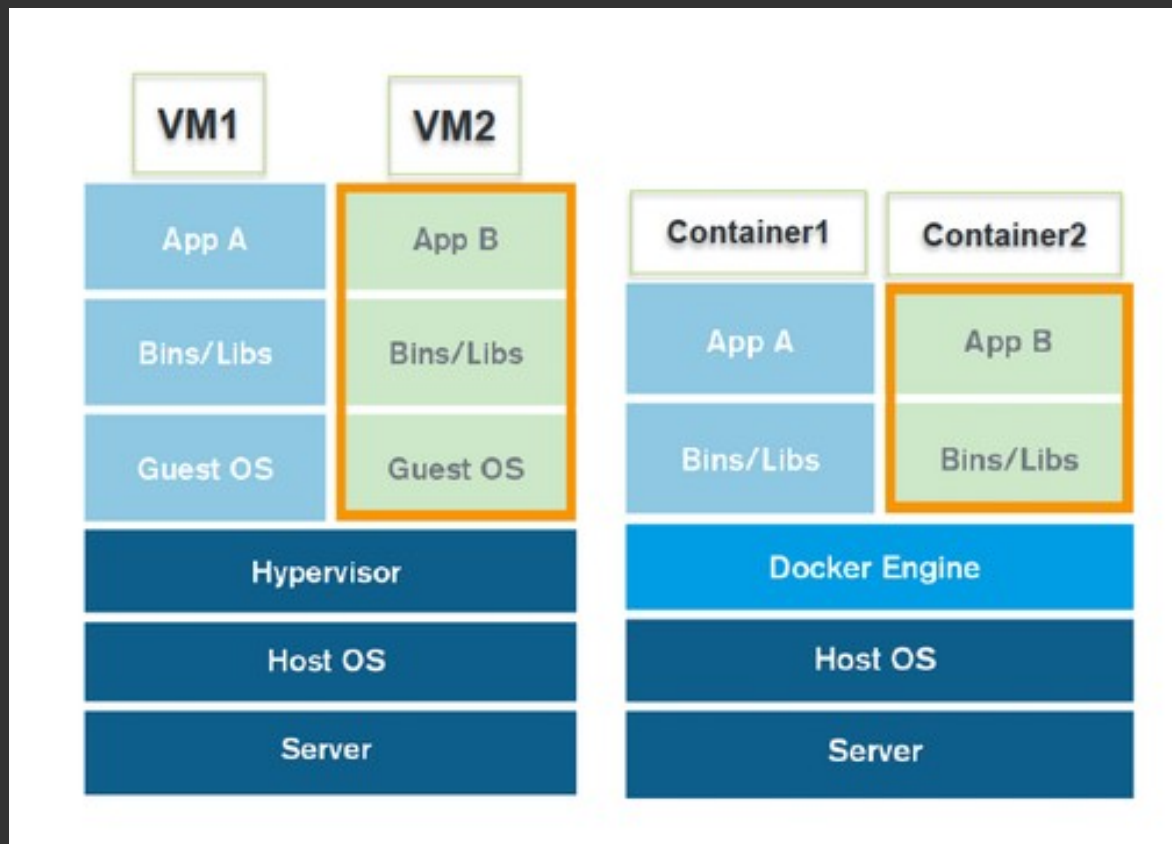
- ♦ łatwe skalowanie

Jak działa

**DOCKER?**

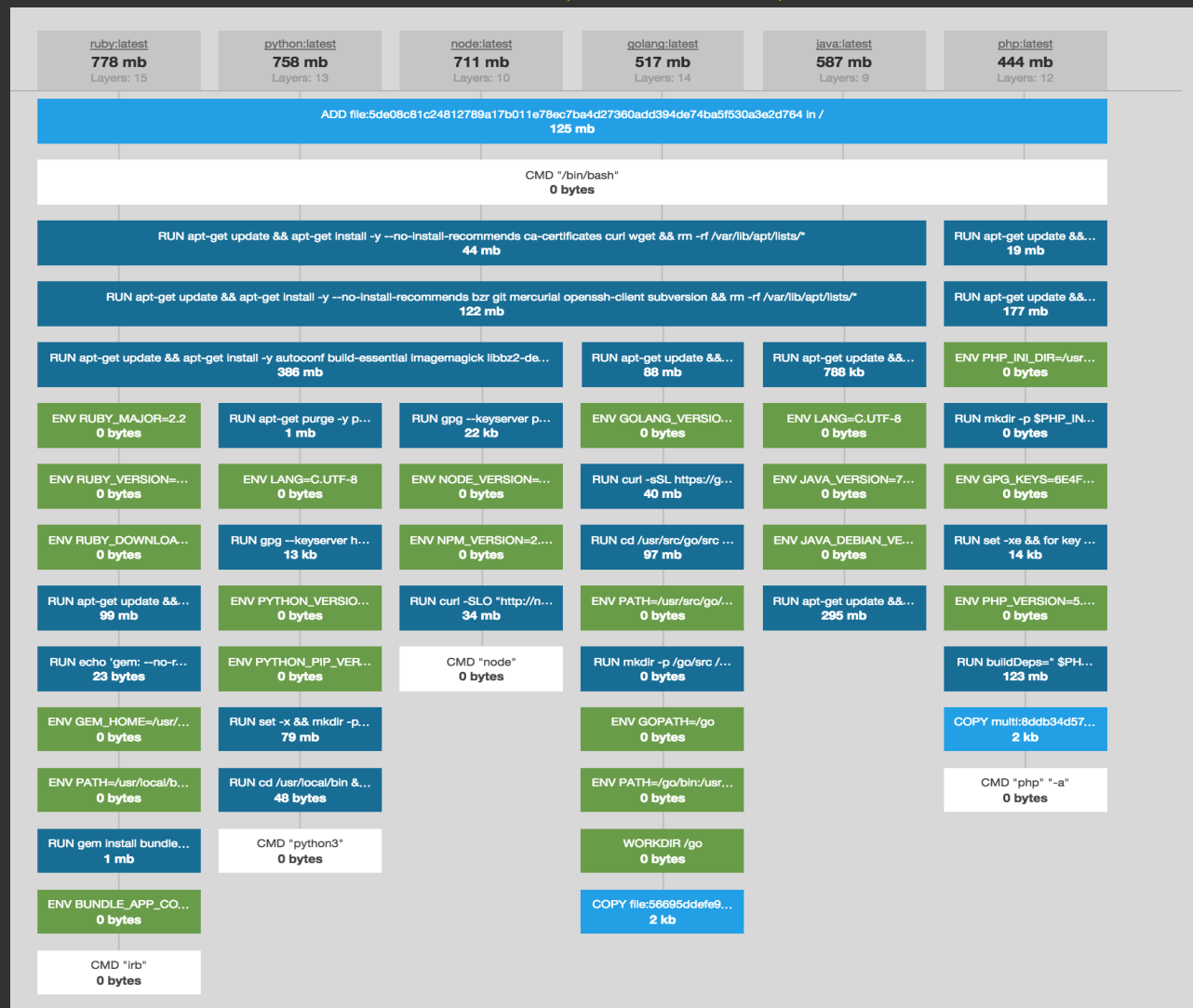
# JAK DZIAŁA DOCKER?

## Maszyna Wirtualna vs. Docker



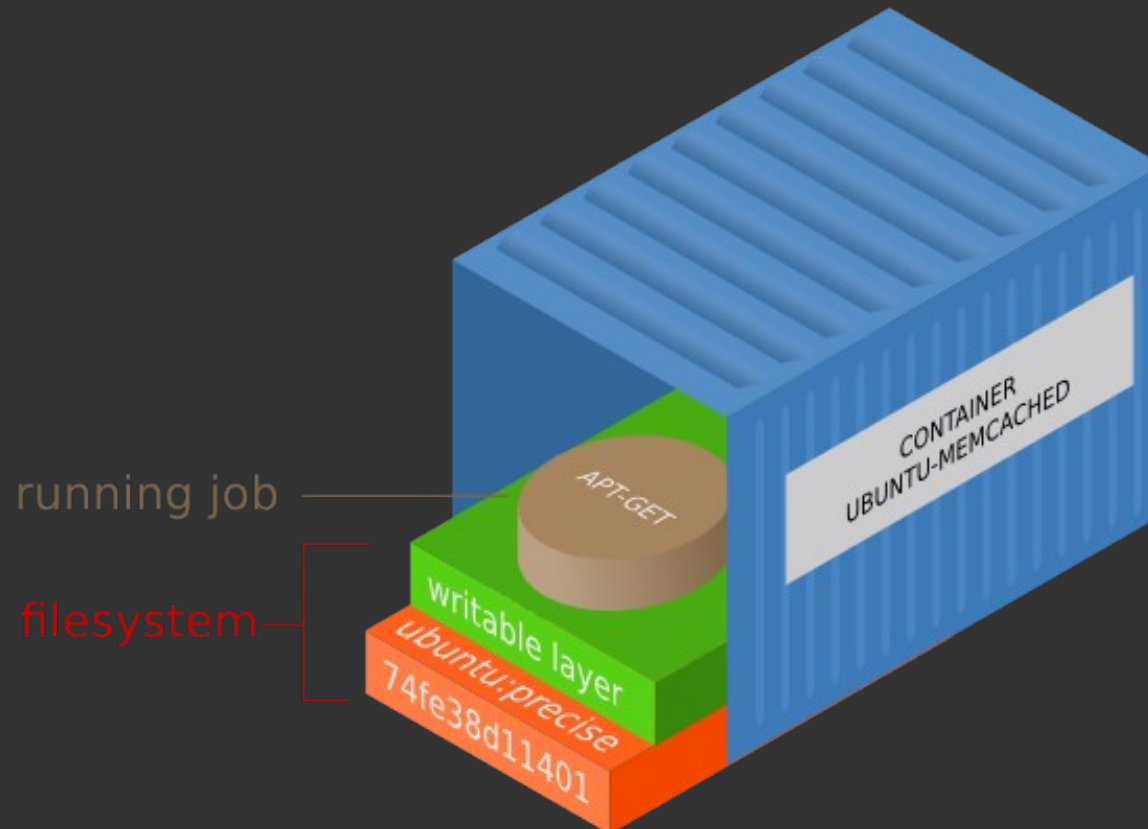
## JAK DZIAŁA DOCKER?

### Obraz (IMAGE)



# JAK DZIAŁA DOCKER?

## Kontener (CONTAINER)





# JAK DZIAŁA DOCKER?

## Repozytorium (REPOSITORY)

Oficjalne

**DOCKER HUB**

<https://hub.docker.com>

Prywatne

**REGISTRY**

registry:latest

# WORKSHOP TIME

# URUCHAMIAMY KONTENER

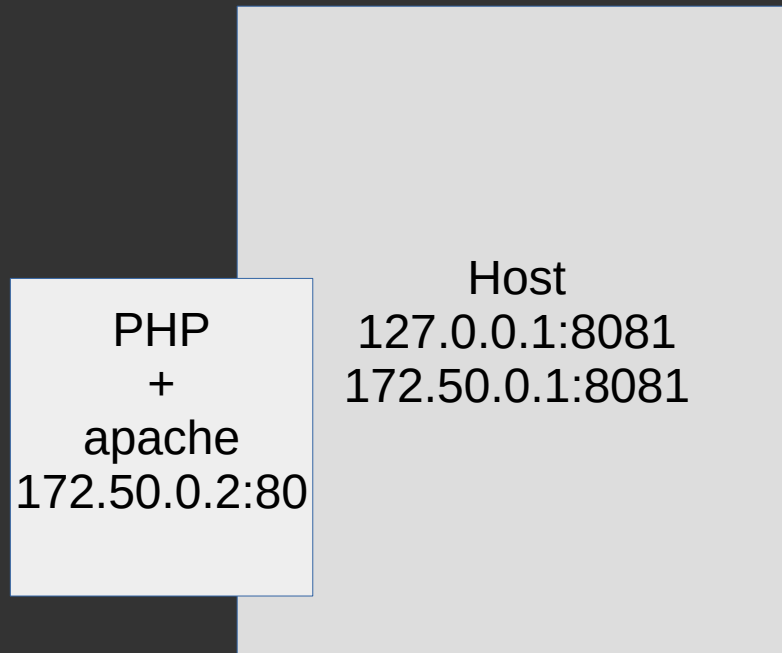
```
$ docker run -v $PWD:/var/www/html -p 8081:80 php:7.4-apache
```

# URUCHAMIAMY KONTENER

```
$ docker run -v $PWD:/var/www/html -p 8081:80 php:7.4-apache
```

# URUCHAMIAMY KONTENER

```
$ docker run -v $PWD:/var/www/html -p 8081:80 php:7.4-apache
```



# URUCHAMIAMY KONTENER

\$ docker ps

CONTAINER ID	IMAGE	COMMAND	CREATED
STATUS	PORTS	NAMES	
019d4c24fbbf	php:7.4-apache	"docker-php-entrypoi.."	19 minutes ago
Up 19 minutes	0.0.0.0:8081->80/tcp	zen_hamilton	

# DOCKER – WPROWADZENIE DO KONTENERÓW

---

```
$ docker run -v $PWD:/var/www/html -p 8081:80 php:7.4-apache  
php:5.6-apache
```

```
$ ls .  
index.php
```

```
$ cat index.php  
<?php  
phpinfo();
```

## Warianty obrazów dla PHP:

- php:<version>-**cli**
- php:<version>-**apache**
- php:<version>-**fpm**
- php:<version>-**alpine**

# Dockerfile

```
$ ls .  
Dockerfile  index.php  php.ini
```

```
$ cat Dockerfile  
FROM php:7.4-apache  
  
RUN apt-get update && apt-get -y install libcurl4-openssl-dev \  
    && docker-php-ext-install pdo \  
    && docker-php-ext-install pdo_mysql.  
  
COPY ./php.ini /usr/local/etc/php
```

```
$ docker build -t myphp .
```

```
$ docker run -v $PWD:/var/www/html -p 8081:80 myphp
```



# DOCKER COMPOSE

```
$ docker run -v $PWD:/var/www/html -p 8081:80 php:7.4-apache
```

# DOCKER COMPOSE

~~\$ docker run -v \$PWD:/var/www/html -p 8081:80 php:7.4-apache~~

# docker-compose.yml

**version: '3.8'**

**Services:**

**www:**

**image: php:7.4-apache**

**volumes:**

**- ./www:/var/www/html**

**ports:**

**- 8080:80**

# docker-compose.yml

**version: '3.8'**

**Services:**

**www:**

**image: php:7.4-apache**

**volumes:**

**- ./www:/var/www/html**

**ports:**

**- 8080:80**

**\$ docker-compose up**

```
Starting demo3_www_1 ... done
```

```
Attaching to demo3_www_1
```

```
www_1 | AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 172.21.0.2. Set the 'ServerName' directive globally to suppress this message
```

```
www_1 | AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 172.21.0.2. Set the 'ServerName' directive globally to suppress this message
```

```
www_1 | [Sun Sep 06 10:48:29.292702 2020] [mpm_prefork:notice] [pid 1] AH00163: Apache/2.4.38 (Debian) PHP/7.4.10 configured -- resuming normal operations
```

```
www_1 | [Sun Sep 06 10:48:29.292741 2020] [core:notice] [pid 1] AH00094: Command line: 'apache2 -D FOREGROUND'
```

# DOCKER COMPOSE

## Łączenie kontenerów



# Docker-compose.yml

```
version: '3.8'
services:
  www:
    image: php:7.4-apache
    volumes:
      - ./www:/var/www/html
    ports:
      - 8080:80
    depends_on:
      - redis
  redis:
    image: redis
```

# www/index.php

```
<?php
    require_once __DIR__.'/vendor/autoload.php';

    $client = new Predis\Client(['host'=>'redis']);

    $counter = $client->incr('counter');

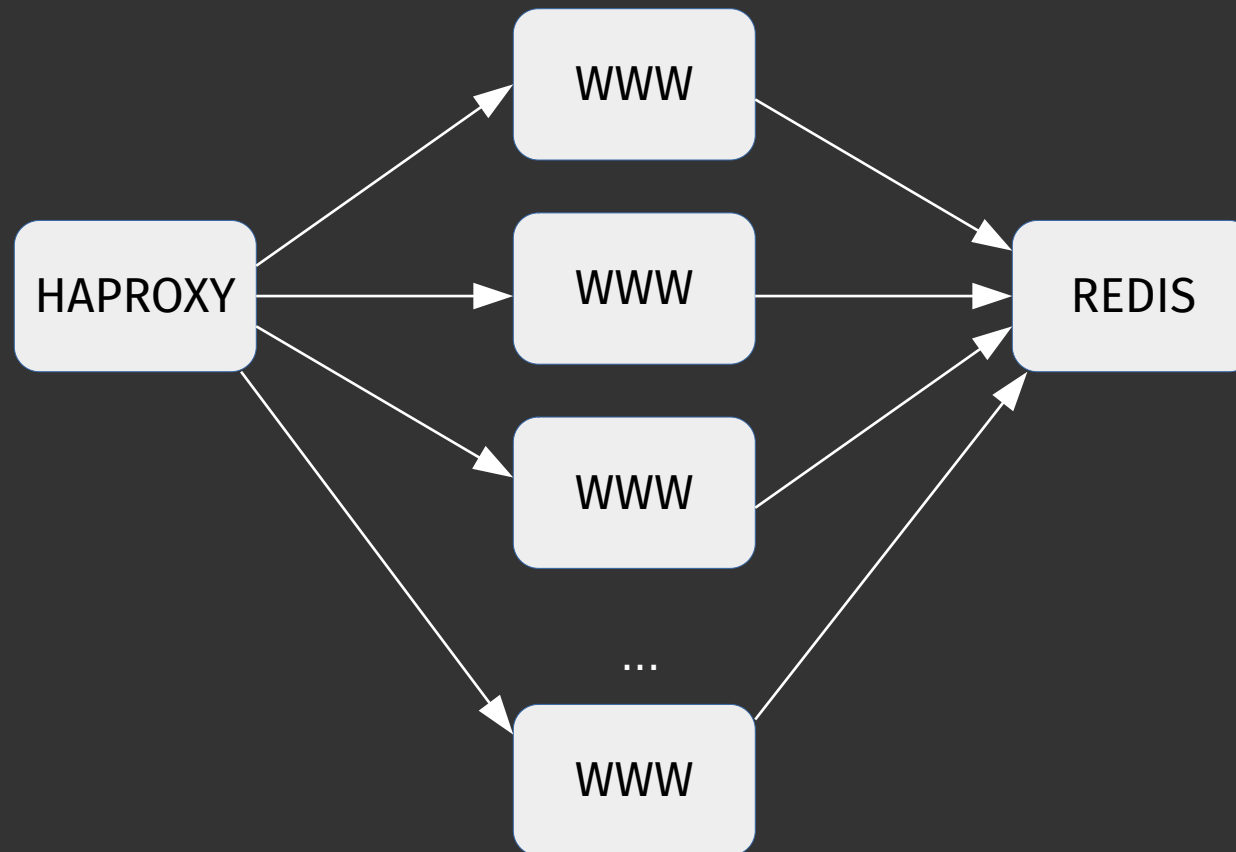
    echo 'Hostname:'.gethostname().'<br/>';
    echo 'Counter:'. $counter.'<br/>';
```

```
$ cd www && composer require predis/predis && cd ..
```

```
$ docker-compose up -d
```

```
$ docker-compose ps
```

# SKALOWANIE APLIKACJI





# Docker-compose.yml

```
version: '3.8'
```

```
services:
```

```
  www:
```

```
    image: php:7.4-apache
```

```
    volumes:
```

```
      - ./www:/var/www/html
```

```
    ports:
```

```
      - 8080:80
```

```
    depends_on:
```

```
      redis
```

```
  redis:
```

```
    image: redis
```

# Docker-compose.yml

```
version: '2'
```

```
services:
```

```
  www: ...
```

```
  redis: ...
```

```
  haproxy:
```

```
    - image: dockercloud/haproxy:latest
```

# Docker-compose.yml

```
version: '2'
services:
  www: ...
  redis: ...
  haproxy:
    - image: dockerccloud/haproxy:latest
  links:
    - www
```

# Docker-compose.yml

```
version: '2'
services:
  www: ...
  redis: ...
  haproxy:
    - image: dockerccloud/haproxy:latest
  links:
    - www
  ports:
    - 80:80
```

# Docker-compose.yml

**version:** '3.8'

**services:**

**www:** ...

**redis:** ...

**haproxy:**

- **image:** dockerccloud/haproxy:latest

**links:**

- **www**

**ports:**

- **80:80**
- **1936:1936**

**volumes:**

- **/var/run/docker.sock:/var/run/docker.sock**

**environment:**

- **STAT\_AUTH='admin:password'**

# Docker-compose.yml

```
$ docker-compose up -d
```

```
$ docker-compose scale www=3
```

```
$ docker-compose ps
```

# PROSTY MIKROSERWIS w PHP

**Docker**

**MySQL + phpmyadmin**

**Fat Free**

**RedBean ORM**

# PROSTY MIKROSERWIS w PHP

Enpointy:

\* /

\* /product

\* /product/<uid>

\* /product/insert



**CO DALEJ?**

**DOCKER SWARM**

**KUBERNETES**

**APACHE MESOS**

# PLIKI Z WARSZTATÓW

<https://github.com/multiart/docker/>

**DZIĘKUJĘ ZA UWAGĘ**  
**PYTANIA?**