



IMPACQ

kubectl scale --help

Scale out -

Arkadiusz Borek

[git@github.com:arekborek/scale-out-ljug.git](https://github.com/arekborek/scale-out-ljug.git)

version 0.0.1-SNAPSHOT

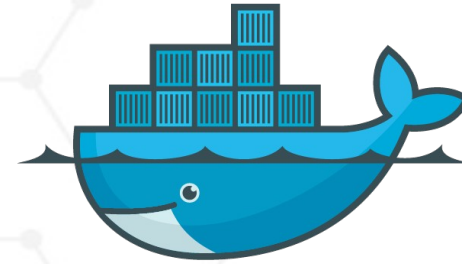
Dziś pomówimy o ...

- 'Hello world' – dla docker.com
- 'Hello world' – dla kubernetes.io
- Scale-out – spring-boot application
 - prosty endpoint
 - spring-security – konfiguracja i klasy
 - ustawienia datasource – specjalnie dla kubernetes, lokalne
 - część docker'a – Dockerfile i com.spotify:docker-maven-plugin
- docker run – lokalna zabawa obrazami
- kubectl run i expos – jak uruchomic swojego pierwszego pod'a
- kubectl scale – dodajemy nowe pod'y

docker.com

Hello world

<https://docs.docker.com/linux/>



docker

Dlaczego ja używam docker'a

- niezależność wersji i konfiguracji
- ogromna łatwość instalacji nowe wersji oprogramowania
- pełna integracja z linux
- profilowanie procesora i pamięci
- profilowanie ruchu sieciowego, oraz analizowanie 'network troubleshooting'
- wymuszenie pracy z konsolą
- w developmencie mamy to samo co na produkcji !!!

kubernetes.io

Hello world

<http://kubernetes.io/docs/getting-started-guides/docker/>

Dlaczego ja używam kubernetes'a

- pod, replica set, replica controller
- profilowanie procesora i pamięci
- kubernetes service
- rolling update, blue-green deployment
- logging, resource monitoring, health checking



kubernetes

Aplikacja spring-boot

endpoint

`pl.ljug.scaleout.request.web.RestController`

spring-security – konfiguracja i klasy

`pl.ljug.scaleout.security.SecurityConfig`

ustawienia datasource – specjalnie dla kubernetes, lokalne

`application.properties`

część docker'a – Dockerfile i `com.spotify:docker-maven-plugin`

`src/main/docker/Dockerfile`

`pom.xml`



Pierwsze uruchomienie

run

```
mvn clean spring-boot:run
```

```
curl -H "Authorization: Basic $(echo -n 'admin:admin' | base64)" localhost:8088/ping2
```

instalacja PostgreSQL

```
docker run -p 5432:5432 --name postgres-scale-out -e POSTGRES_PASSWORD=pass -e  
POSTGRES_DB=scale_out -d postgres
```

Dockerizing

run

mvn clean

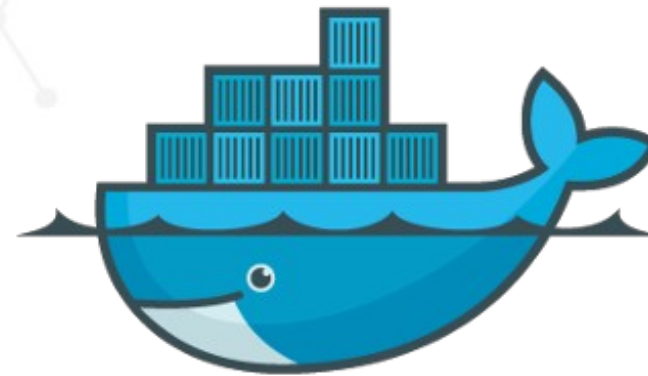
mvn package

mvn docker:build

docker push aborek/docker-scale-out



+



docker

Instalacja kubernetes.io ...

run

K8S_VERSION=1.2.1

```
docker run --volume=/:/rootfs:ro --volume=/sys:/sys:ro --volume=/var/lib/docker/./var/lib/docker:rw  
--volume=/var/lib/kubelet/./var/lib/kubelet:rw --volume=/var/run:/var/run:rw --net=host --pid=host  
--privileged=true --name=kubelet -d gcr.io/google_containers/hyperkube-amd64:v${K8S_VERSION}  
/hyperkube kubelet --containerized --hostname-override="127.0.0.1"  
--address="0.0.0.0" --api-servers=http://localhost:8080 --config=/etc/kubernetes/manifests --allow-privileged=true --v=2
```

instalacja PostgreSQL

```
docker run -p 5432:5432 --name postgres-scale-out -e POSTGRES_PASSWORD=pass -e  
POSTGRES_DB=scale_out -d postgres
```


cd. kubernetes.io ...

pod

```
kubectl run scaleout --image=aborek/docker-scale-out --port=8088
```

```
kubectl expose deployment scaleout --port=8088
```

test

```
curl -H "Authorization: Basic $(echo -n 'admin:admin' | base64)" {IP}:8088/ping2
```

scale

```
kubectl scale deployment scaleout --replicas=3
```

test

```
curl -H "Authorization: Basic $(echo -n 'admin:admin' | base64)" {IP}:8088/ping2
```

```
ab -c 1 -t 5 -H "Authorization: Basic $(echo -n 'admin:admin' | base64)" 10.0.0.253:8088/ping2
```

```
ab -k -c 1 -t 5 -H "Authorization: Basic $(echo -n 'admin:admin' | base64)" 10.0.0.253:8088/ping2
```

cd. kubernetes.io ...

stopowanie kubernetes'a

```
docker stop $(docker ps -q)
```

```
docker rm $(docker ps -aq --since={CONTAINER_NAME})
```

```
kubectl delete deployment,service my-nginx
```

usuwanie aplikacji z kubernetes'a

```
kubectl delete deployment,service scaleout
```

Chciałbym:

- zapoczątkować używanie konsoli
- zapoczątkować używanie docker
- małe wprowadzenie do ab - Apache Benchmark
- czas rozpocząć swoją przygodę z kubernetes

THX !!