

<https://www.youtube.com/watch?v=yFPYGeKwmpk> - czy mamy 3kę

w openshift mam projekt - jak database

- w nim aplikacje, one postawione jako docker+ kubernetes
- moge zwiekszyc ich ilosc
- czyli dla mnie jakis image to apka, moge kilka dawac
- ale u mnie w konfigu musi byc zmiana node name :/
- jak stawiam 5 takich samych mam od kopa load balancing
- budowa to wziecie kodu i docker _ kubernetas w locie
- fajna apka zarzadzajaca openshift, czy mamy cos takiego ? (openshift origin console , web console)
- czy wszystko z command line oc
- pod projektem w browse duzo opcji
- mozna widziec buildy, logi,
- w deployment 1, ale 5 replica
- mozna persistant storage outside containers
- mamy dostep do wszystkich wykonanych operacji - events
- w images mamy docekr registry
- na pads mamy aktywne docker containcers
- mamy dostep ile pamiecie zjada pad, jego logi
 - mozna klikanc w konsole i z aplikacji polecenia na kontenerze
 - - routs mowi jak exponowany i na jakim porcie na zewnatrz
 - mozna pod github podpiac, ze jak kod sie zmieni to automat przebudowa
 - mozna to z konsoli, np. Oc get pods
 - oc get dc (deployment config)
 - oc scale dc frontend --replica=2
 - mozemy widzec zmienne srodowiskowe kontenera
 - mozna logi full screan, dopisuje je
 - do oc najpierw sie loguje z konsoli i okreslam projekt
 - mozna oc new project nazwa
 - w nim moge nowy proj latwo

- czy uzywamy red hut developer studio ?

redis :

```
FROM          ubuntu:14.04
RUN           apt-get install -y redis-server
EXPOSE        6379
ENTRYPOINT    ["/usr/bin/redis-server"]
```

- poczytac o obrazach elk i o budowi elk na dockerach

OPEN SHIFT

- szybki deploy i skalowanie aplikacji
- do budowania i odpalania aplikacji
- aplikacja – to zwykła web aplikacja
- gear – to kontener z zasobami do uruchomienia apki
- cardridge – to pluginy, frameworki, komponenty do uruchomienia apki
- czy używamy open shift online ? - można tu łatwo dodać aplikację, wybieram jej typ i nazwę
- mogę przez gui dodać bazy tam
- wchodzę do katalogu i wrzucam moją apkę, zmieniam zamiar do bazy
- mogę dostarczyć hooks do budowy i deploy skryptów naszych
- po zmianach push do openshift i odpala

KUBERNETIS

- pod to najmniejsza deployowana część
- w pod mam 1 lub więcej kontener images
- w App1.yml mam dane na temat klastra k8s
- - pod (i jego obrazy), replica – ile tych podów będzie, potem kolejne pody z obrazami i replica
- podaje tę konfigurację yml do K8S cluster service API
- w klastrze kubernetes są workery, które mają api (K), do komunikacji z klastrem serwisem
- przy wczytaniu konfiguracji pody są rozkładane na workery (ich repliki na osobne workery)
- jak jakiś worker pada, to jego pody są przenoszone na inny
- daje deployment, skalowanie i monitorowanie
- w moim deployment obrazie mam wszystkie potrzebne dane i apkę
- robi recover jak padnie kontener
- daje load balancer

OPEN SHIFT



- każdy pod ma ip, nad nimi serwis – daje do nich dostęp, a route – dostęp publiczny i load balancing

- pądsy są dostępne przez serwisy, po nazwie, są rozmieszczone na węzłach, współdzieli zasoby węzła
- do pąds jest replication controller, nowe replica tworzy, jak pada to odtwarza
- build konfiguracji, mówi jak deploy image, image może być z source image (dostępne dla podstawowych jak java itd), albo z Dockerfile
- można web hook, by się odpalił po zmianie kodu

D+K8+OS

- obraz docker pliki do środka ładuje z tego samego katalogu
- można startować z różnym portem
- kubectl version – kub control
- **open shift enterprise console**
- w kube4docker mamy skrypty tworzące namespace, pąds
- można zapytania z kubectl jak z oc, pobierz pąds, ustaw replica

```
[vagrant@rhel-cdk kube4docker]$ kubectl --namespace=kubedemo get pods
NAME          READY   STATUS    RESTARTS   AGE
mynode-3focn   1/1     Running   0           30s
[vagrant@rhel-cdk kube4docker]$ kubectl --namespace=kubedemo get services
NAME          CLUSTER_IP      EXTERNAL_IP    PORT(S)      SELECTOR      AGE
mynode        172.30.59.211    8000/TCP       run=mynode    17s
[vagrant@rhel-cdk kube4docker]$ curl 172.30.59.211:8000
HELLO Host/Pod: mynode-3focn
[vagrant@rhel-cdk kube4docker]$ curl 172.30.59.211:8000
HELLO Host/Pod: mynode-3focn
[vagrant@rhel-cdk kube4docker]$ ./4_scale_replicas.sh
kubectl --namespace=kubedemo scale rc mynode --replicas=3
replicationcontroller "mynode" scaled
```

- teraz jak strzelam curl'em to mam load balance jak mam więcej instancji replica