



# Azure ❤️ Kubernetes

Sebastian Fyda  
sebastian@spiderweb.pl



# Dwa słowa od Ojca Prowadzącego

- 12 lat administruję Linuxami
- 6 lat zajmuję się architekturą rozproszonych systemów IT
- Od 4 lat pracuję z Azure
- Pracowałem/pracuję dla:



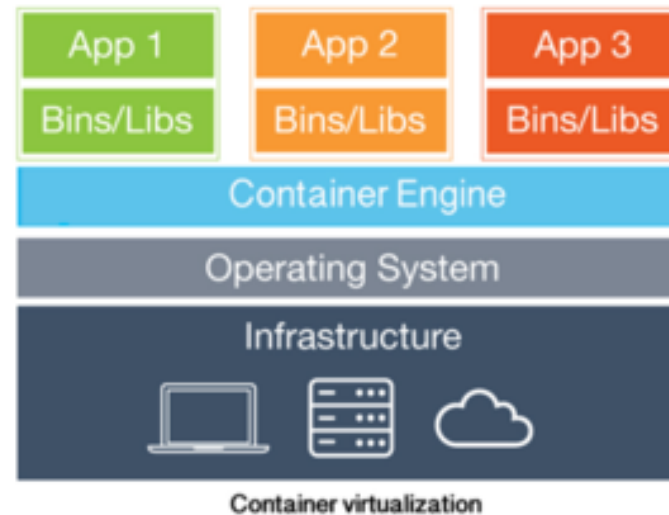
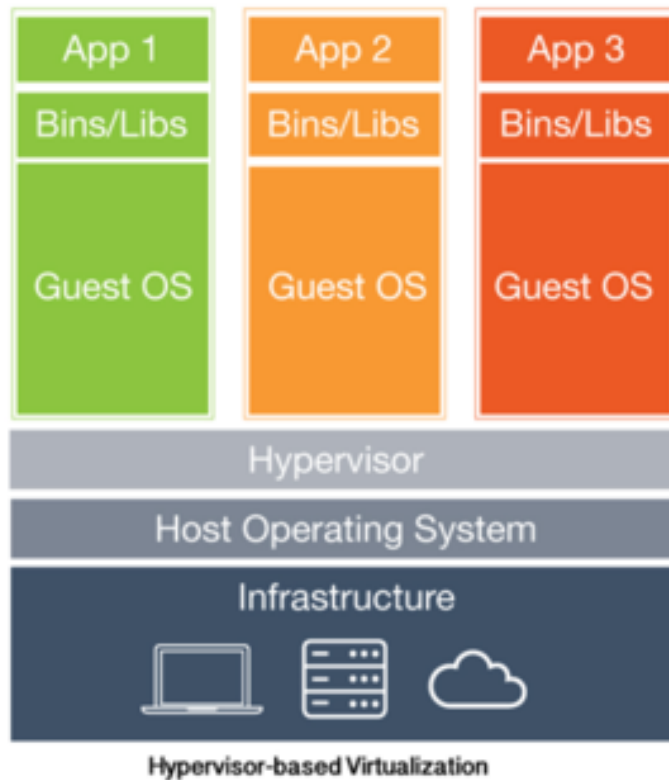


# Agenda

- kontenery – szybkie intro
- orkestracja – jeszcze szybsze intro
- kubernetes – wizja świata
- ks8 – zasoby i jak zbudować prostą aplikację
- k8s w Azure
- k8s w środowisku DEV
- pytania/odpowiedzi/gwizdy/brawa



# Kontenery - 101





# Kontenery - 101

- Zalety
  - Izolacja procesu biznesowego
  - Niewielki footprint
  - Jednorodność środowisk pomiędzy devami
  - Jednorodność środowisk test/dev/staging/prod
- Wady:
  - Stabilność (docker-compose, docker-engine)
  - Interoperacyjność jest mitem (docker+win, docker+mac)
  - Niski poziom adopcji (rkt)
  - Idiotyczny proces wydawniczy (docker-engine, docker-compose)
  - Performance (overlay-fs, php+docker+mac)



# Orchestrator - 101

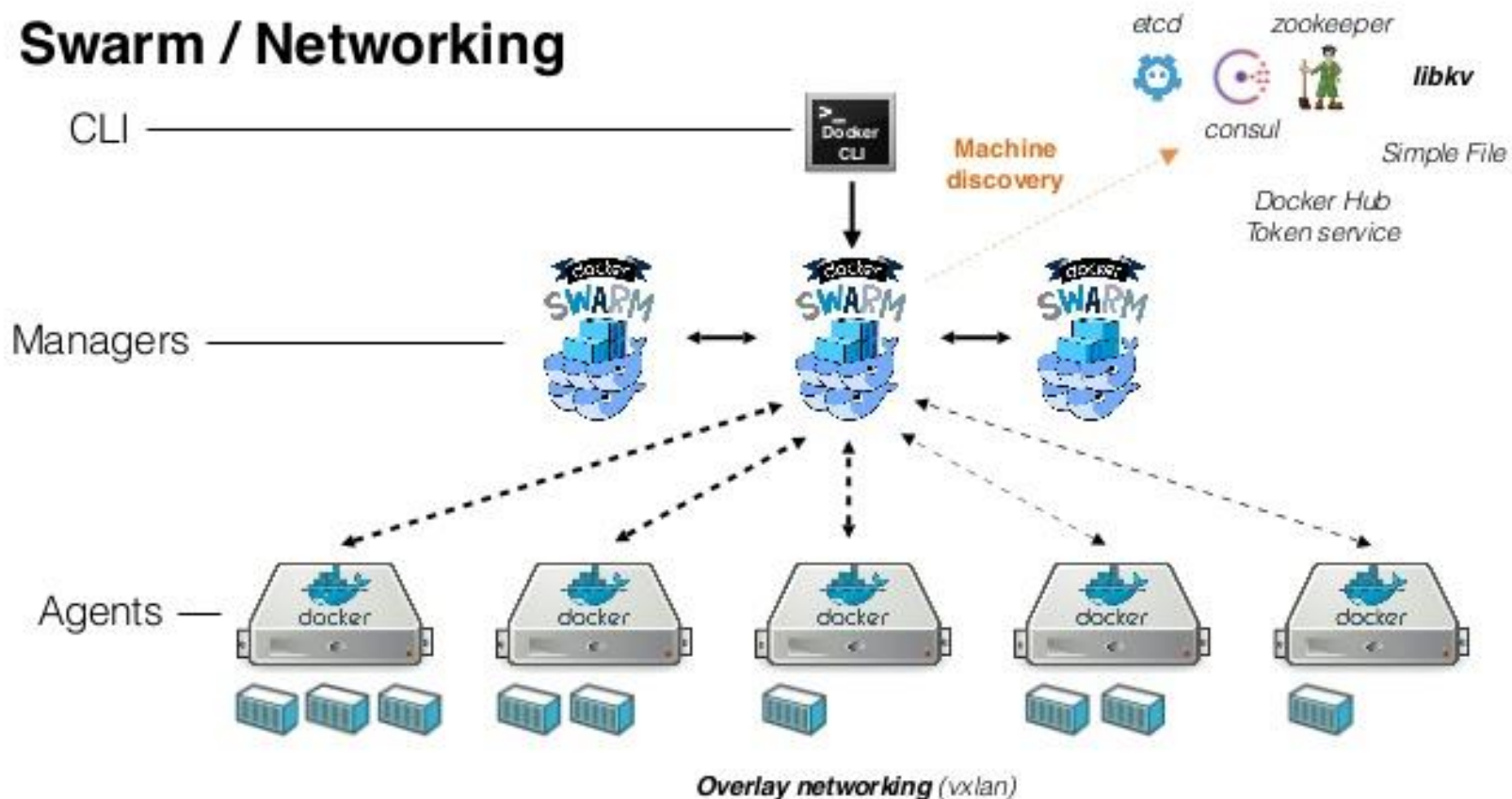
Do wyboru do koloru:

- Kubernetes
- Swarm
- Rancher
- DC/OS – dawniej Mesos



# Orchestration - 101

## Swarm / Networking





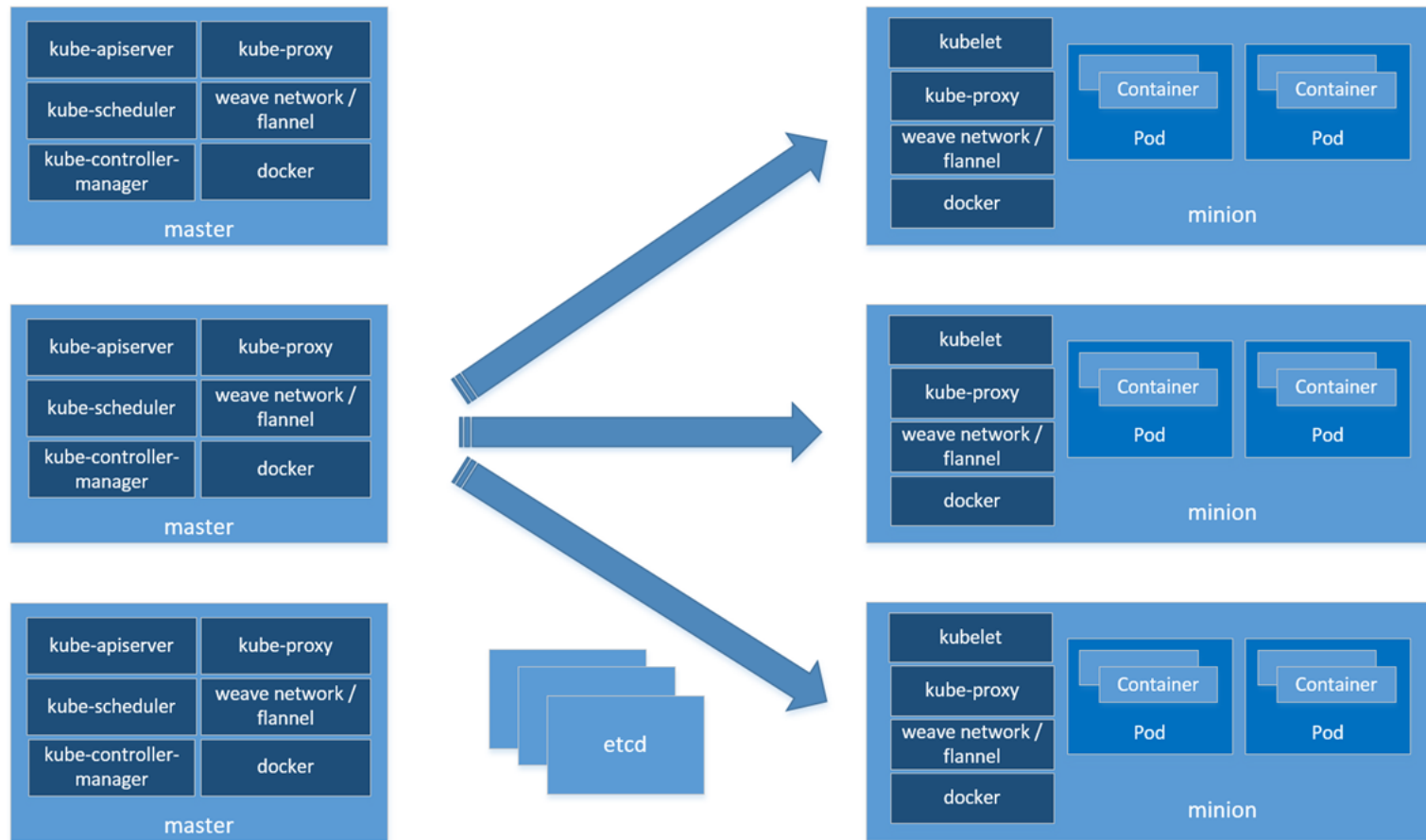
# Kubernetes – wizja świata

- Budujemy z klocków
- Automatyzacja typowych elementów
  - Discovery / dns
  - Load balancing
  - Healt check
- Operujemy na wysokim poziomie abstracji
- Zachowujemy szczegółową kontrolę

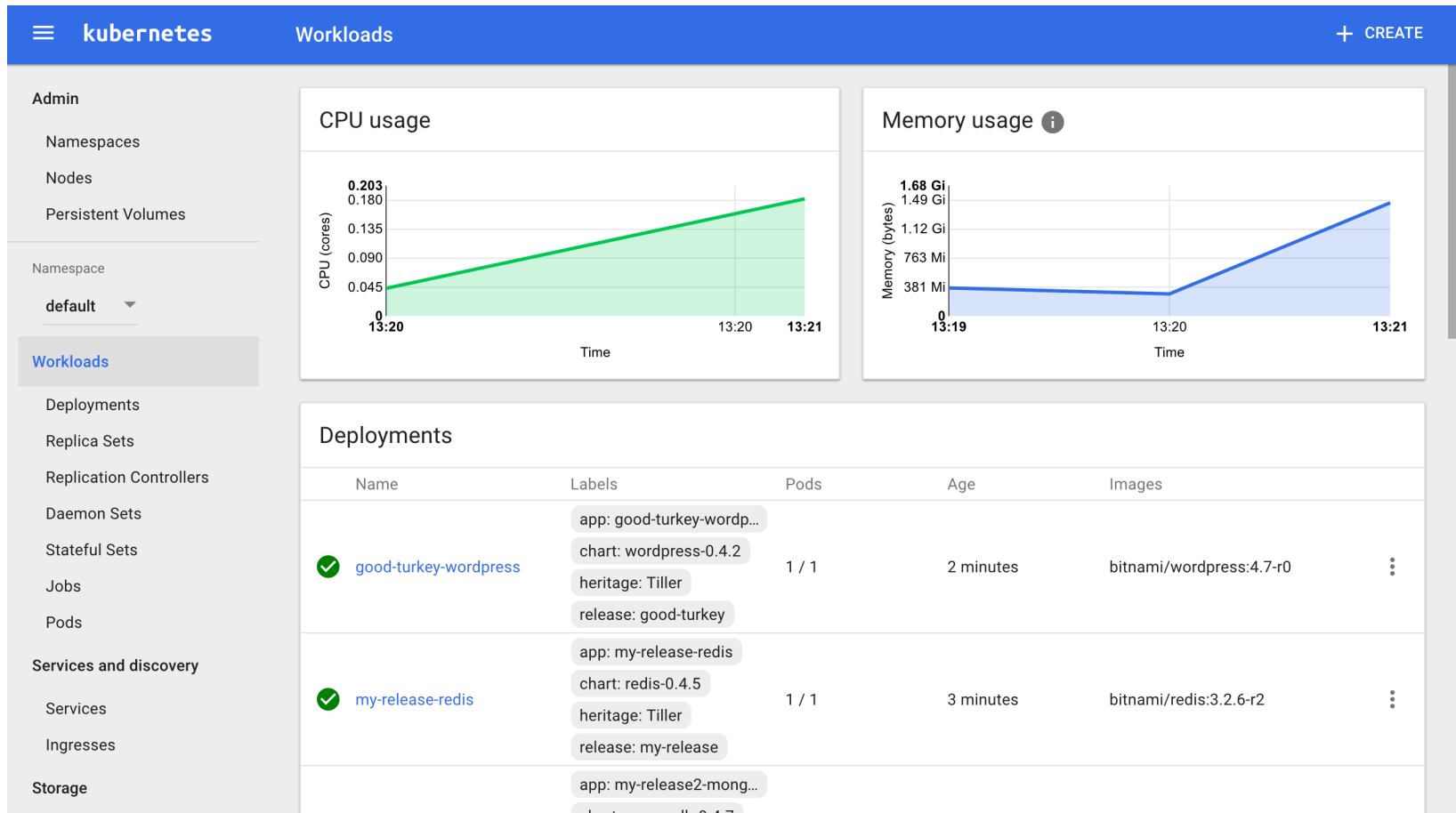




# Kubernetes - architektura



# Kubernetes - GUI





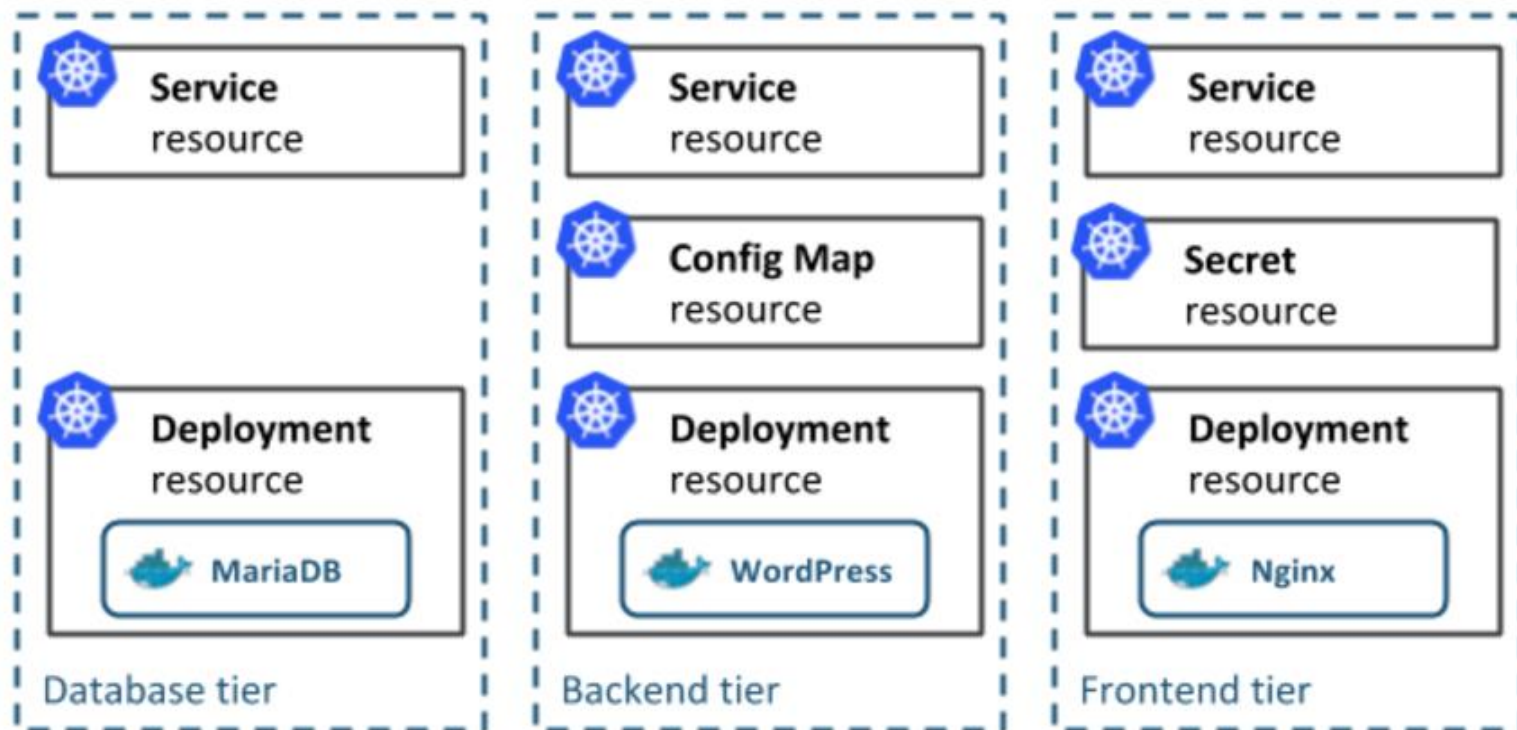
## K8s – zasoby (te najważniejsze)

- namespaces
- deployments (!!!)
- services
- configmaps
- cronjobs
- secrets

Dlaczego nie „pods” ?



# K8s – prosta aplikacja





# K8s – deployment

```
apiVersion: extensions/v1beta1
```

```
kind: Deployment
```

```
metadata:
```

```
  name: backend
```

```
spec:
```

```
  strategy:
```

```
    type: Recreate
```

```
  replicas: 2
```

```
  template:
```

```
    metadata:
```

```
      labels:
```

```
        app: backend
```

```
        color: blue
```

```
        tier: backend
```

```
        track: stable
```

```
spec:
```

```
  volumes:
```

```
    - name: cache-volume
```

```
      emptyDir: {}
```

```
    - name: logs-volume
```

```
      emptyDir: {}
```

```
    - name: backend-app-config-volume
```

```
      configMap:
```

```
        name: backend-app-config
```

```
    - name: nginx-ssl
```

```
      secret:
```

```
        secretName: nginx-ssl
```

```
    - name: nginx-vhosts
```

```
      configMap:
```

```
        name: backend-app-config
```

```
        items:
```

```
          - key: backend.conf
```

```
            path: backend.conf
```



# K8s – deployment

containers:

- image: `${CR}/backend-nginx:${BACKEND_NGINX_TAG}`  
name: backend-nginx  
ports:
  - containerPort: 80imagePullPolicy: IfNotPresent  
volumeMounts:
  - name: nginx-ssl  
mountPath: `/etc/nginx/ssl`  
readOnly: true
  - name: nginx-vhosts  
mountPath: `/etc/nginx/sites-enabled`



# K8s – service

```
apiVersion: v1
kind: Service
metadata:
  name: backend
spec:
  ports:
    - name: http
      port: 80
      targetPort: 80
    - name: https
      port: 443
      targetPort: 443
  selector:
    app: backend
    color: blue
    tier: backend
    track: stable
  type: LoadBalancer
```



# K8s – secret + configmap

```
apiVersion: v1
data:
  tls.crt: <SSL CERT>
  tls.key: <SSL KEY>
kind: Secret
metadata:
  creationTimestamp: 2017-04-21T09:17:13Z
  name: nginx-ssl
  resourceVersion: "9256362"
  selfLink: /api/v1/namespaces/default/secrets/nginx-ssl
  uid: 4e8e6837-2673-11e7-a2c4-000d3ab45d2f
type: kubernetes.io/tls
```

```
kubectl create configmap app-config --from-file=/here/are/my/configs/app
```



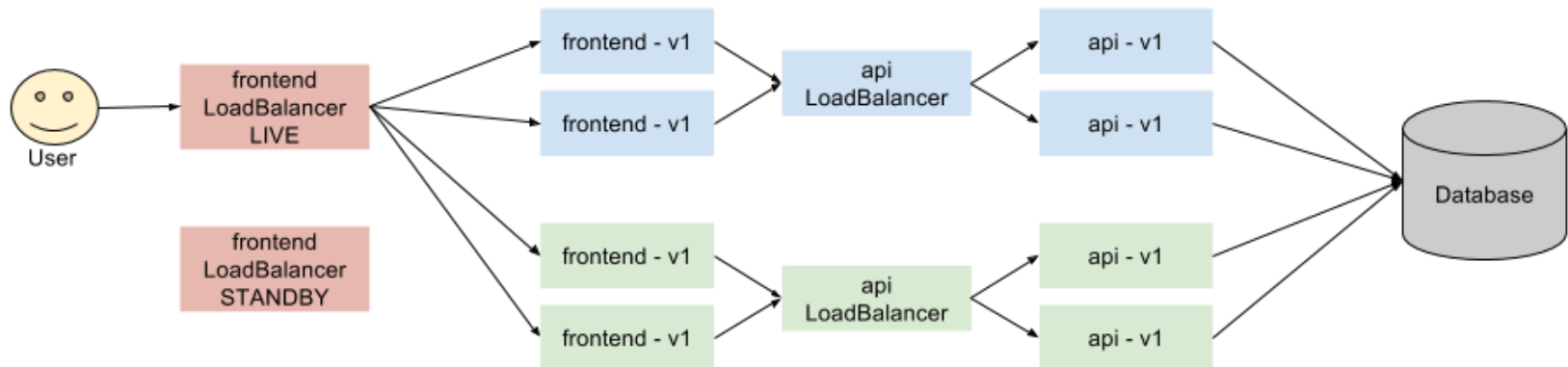


# K8s – kctl

```
tawnos@wulkan:/srv/develop/rebug/gce-rebug-alpha1$ kctl-rebug get pods
NAME                                READY    STATUS    RESTARTS   AGE
backend-2913609657-5dvtc            2/2      Running   0           18h
backend-2913609657-sw7nw            2/2      Running   0           18h
frontend-1212532879-kdw0s          1/1      Running   0           18h
frontend-1212532879-qdvng          1/1      Running   0           18h
memcached-daemon-1723771783-6pn6p  1/1      Running   0           18h
mysql-server-3550276845-1p0x9       1/1      Running   0           18h
redis-1021628553-m6dcv             1/1      Running   0           18h
socket-680967188-z0g66             1/1      Running   0           18h
writer-1905178153-d3dsz            2/2      Running   0           7h
tawnos@wulkan:/srv/develop/rebug/gce-rebug-alpha1$ kctl-rebug get deployments
NAME            DESIRED    CURRENT    UP-TO-DATE    AVAILABLE    AGE
backend         2           2           2              2            26d
frontend        2           2           2              2            26d
memcached-daemon 1           1           1              1            25d
mysql-server    1           1           1              1            16d
redis           1           1           1              1            15d
socket          1           1           1              1            26d
writer          1           1           1              1            26d
2017-05-24 17:14:30 kctl-rebug get deployments
tawnos@wulkan:/srv/develop/rebug/gce-rebug-alpha1$ kctl-rebug get services
NAME            CLUSTER-IP      EXTERNAL-IP      PORT(S)              AGE
backend         10.0.10.91      13.95.17.11      80:30341/TCP,443:30685/TCP 26d
database        10.0.134.220    137.116.202.33   3306:30033/TCP        16d
frontend        10.0.141.225    13.73.107.112    80:30968/TCP,443:31764/TCP 26d
memcached       10.0.215.159    <none>           11211/TCP             25d
redis           10.0.42.23      <nodes>          6379:30369/TCP        15d
socket          10.0.55.11      <none>           8000/TCP              26d
writer          10.0.176.240    52.166.7.107     443:31814/TCP         26d
writer-server-rest 10.0.196.101    <none>           32702/TCP             26d
2017-05-24 17:14:34 kctl-rebug get services
tawnos@wulkan:/srv/develop/rebug/gce-rebug-alpha1$
```



# K8s – labels + blue-green deployment





# Kubernetes w Azure

- Instalacja

- Portal – jednocześnie najprostsza i najbardziej skomplikowana
- CLI

```
az group create -n my_acs_group -l westeuropeaz
acs create -n my_shiny_acs -g my_acs_group -d my-domain \
  --ssh-key-value ~/.ssh/mykey_acs_id_rsa.pub \
  --orchestrator-type kubernetes --agent-vm-size Standard_D2_v2 \
  --agent-count 5
az acs kubernetes install-cli
az acs kubernetes get-credentials -g my_acs_group \
  --ssh-key-file ~/.ssh/mykey_acs_id_rsa --name my_shiny_acs
```

- ARM template – dodatkowe możliwości, np. custom VNET



# Kubernetes w Azure

spw-kubernetes

Resource group

Search (Ctrl+F)

Overview

Activity log

Access control (IAM)

Tags

SETTINGS

Quickstart

Resource costs

Deployments

Policies

Properties

Locks

Automation script

MONITORING

Metrics

Alert rules

Diagnostics logs

Application insights

Log analytics (OM3)

Log search

SUPPORT + TROUBLESHOOTING

New support request

+

 Add

Columns

Delete

Refresh

Move

Essentials

Subscription name (change)

Microsoft Azure Sponsorship

Deployments

1 Failed, 2 Succeeded

Subscription ID

fd1de6b5-eb3a-4a7f-b16d-ea78de53ded1

Filter by name...

All types

All locations

No grouping

25 items

NAME	TYPE	LOCATION
00ghfkkt2zn3oagm0	Storage account	North Europe
agent-availabilitySet-DED208A2	Availability set	North Europe
ghfkkt2zn3omst0	Storage account	North Europe
k8s-agent-DED208A2-0	Virtual machine	North Europe
k8s-agent-DED208A2-1	Virtual machine	North Europe
k8s-agent-DED208A2-2	Virtual machine	North Europe
k8s-agent-DED208A2-nic-0	Network interface	North Europe
k8s-agent-DED208A2-nic-1	Network interface	North Europe
k8s-agent-DED208A2-nic-2	Network interface	North Europe
k8s-master-DED208A2-0	Virtual machine	North Europe
k8s-master-DED208A2-nic-0	Network interface	North Europe
k8s-master-DED208A2-nsg	Network security group	North Europe
k8s-master-DED208A2-routetable	Route table	North Europe
k8s-master-ip-publickubel1-spw-kubernetes-fd1de6-DED208A2	Public IP address	North Europe
k8s-master-lb-DED208A2	Load balancer	North Europe
k8s-vnet-DED208A2	Virtual network	North Europe
master-availabilityset	Availability set	North Europe
publickubel1	Container service	North Europe
publickubel1-spw-kubernetes-fd1de6	Load balancer	North Europe
publickubel1-spw-kubernetes-fd1de6-a37192043fd1f11e6a2c4000d3ab45d2	Public IP address	North Europe
publickubel1-spw-kubernetes-fd1de6-a38884a24fd1f11e6a2c4000d3ab45d2	Public IP address	North Europe
publickubel1-spw-kubernetes-fd1de6-a3bad39500d7d1e7a2c4000d3ab45d2	Public IP address	North Europe
publickubel1-spw-kubernetes-fd1de6-a4f9843820071e7a2c4000d3ab45d2	Public IP address	North Europe
publickubel1-spw-kubernetes-fd1de6-a4f9843820071e7a2c4000d3ab45d2	Public IP address	North Europe

www.spiderweb.pl



# Kubernetes w Azure

- Zalety
  - Automatyczny provisioning klastra
  - Integracja z usługami Azure
    - Automatyczne tworzenie publicznych IP
    - Automatyczne zarządzanie loadbalancerami
    - Automatyczne tworzenie zasobów w Storage dla PersistentVolumes
    - Automatyczne zwalnianie zasobów usuniętych z K8s
- Wady
  - Ograniczenie do konkretnej wersji (brak niektórych alpha-features)
  - Brak możliwości zmiany rozmiaru VMek
  - Nieznana jeszcze ścieżka aktualizacji klastrów



# Kubernetes w środowisku DEV



## minikube

### macOS

```
brew cask install minikube
```

### Linux

```
curl -Lo minikube https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64 && chmod +x minikube && sudo mv minikube /usr/local/bin/
```



# Pytania? Gwizdy?