

# Technologie cloudowe w praktyce

Jakub Kabza Tomasz Smelcerz, SAP  
20.10.2018

PUBLIC



# Agenda



## Java – Introduction to Spring framework

23.10.2018

Tomasz Miler

Bartosz Niesobski

## Technologie Cloudowe w praktyce

20.11.2018

Jakub Kabza

Tomasz Smelcerz

## Wieczorek pod Chmurą

04.12.2018

Mateusz Szostok

Tomasz Pietrek

Tomasz Heflik

## Java – Microservices with Spring Boot

06.11.2018

Tomasz Miler

Bartosz Niesobski

## Scrum Master – The journey towards mastery

26.11.2018

Michał Drzewiecki  
Mariusz Jasiński

## Scripting vs. Compiled, Dynamic vs. Static – in JVM world

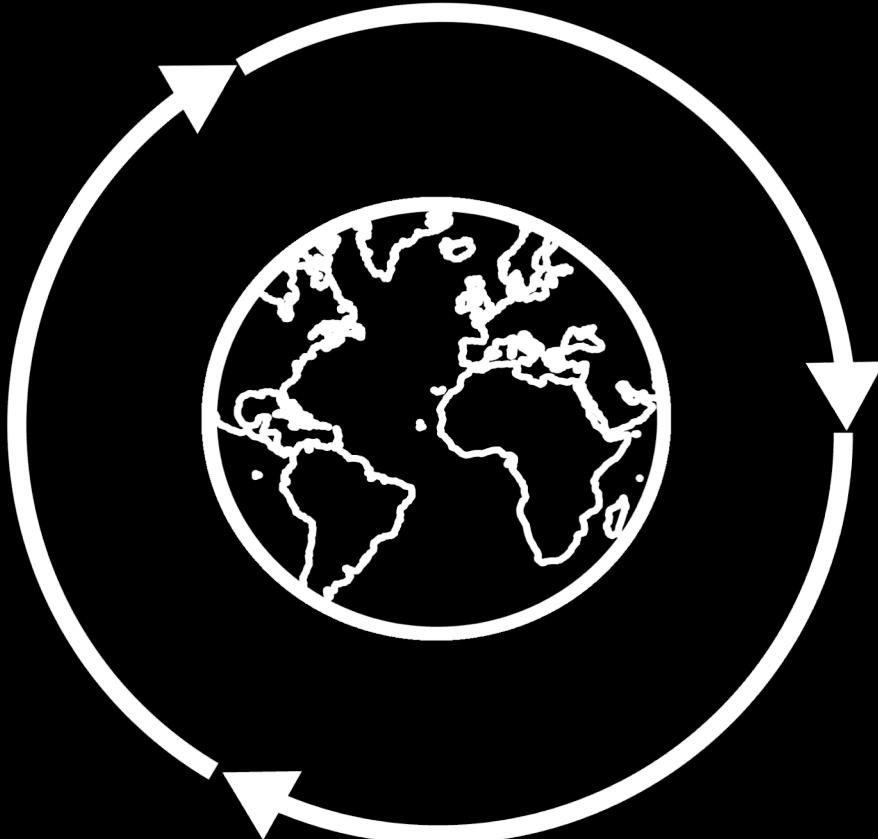
11.12.2018

Karol Grzyb  
Tomasz Miler

# SAP Labs Poland

**Top ecommerce,  
marketing, billing**

**Development:** Go, Java,  
Cloud Native solutions



**> 400 pracowników**

**Najlepszy Pracodawca  
2017 w rankingu AON**

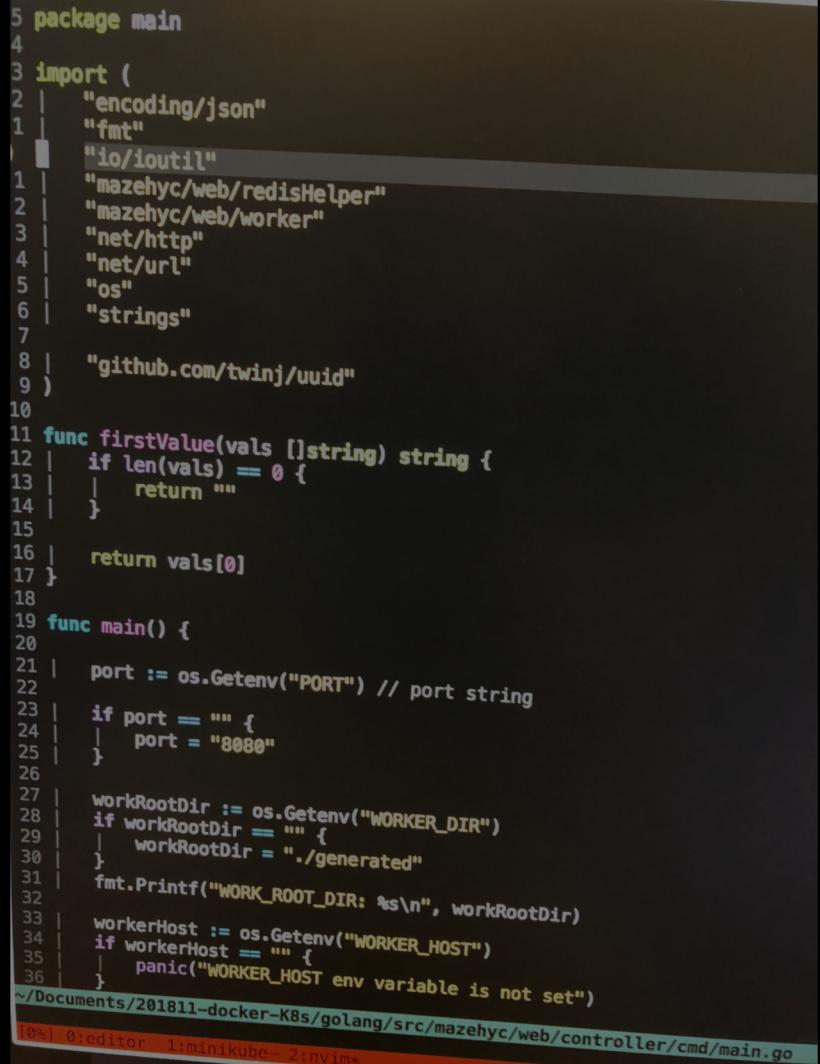
**Jedno z 20 centrów  
SAP's Labs Network**

# O nas...



- **Programiści: Golang, Node.js, Java**
- **DevOps (utrzymanie, CI)**
- **Rozwiązania oparte na technologiach cloudowych (Docker, K8s)**
- **SAP Polska Gliwice**

# Problem: Aplikacje które nie działają poprawnie



```
5 package main
4
3 import (
2 |     "encoding/json"
1 |     "fmt"
1 |     "io/ioutil"
1 |     "mazehyc/web/redisHelper"
2 |     "mazehyc/web/worker"
3 |     "net/http"
4 |     "net/url"
5 |     "os"
6 |     "strings"
7 |
8 |     "github.com/twinj/uuid"
9 )
10
11 func firstValue(vals []string) string {
12     if len(vals) == 0 {
13         return ""
14     }
15
16     return vals[0]
17 }
18
19 func main() {
20
21     port := os.Getenv("PORT") // port string
22     if port == "" {
23         port = "8080"
24     }
25
26
27     workRootDir := os.Getenv("WORKER_DIR")
28     if workRootDir == "" {
29         workRootDir = "./generated"
30     }
31     fmt.Printf("WORK_ROOT_DIR: %s\n", workRootDir)
32
33     workerHost := os.Getenv("WORKER_HOST")
34     if workerHost == "" {
35         panic("WORKER_HOST env variable is not set")
36     }
~/Documents/201811-docker-K8s/golang/src/mazehyc/web/controller/cmd/main.go
[0%] 0:editor 1:minikube 2:nvim+
```

# Problem: Stworzyć aplikację która działa poprawnie\*

Lokalny development vs produkcja

- "u mnie działa" ☺
- Przyczyny: Błędne założenia (lub ich brak) co do środowiska pracy procesu
- Przykłady: Certyfikaty, biblioteki, narzędzia



\* nie tylko na komputerze programisty ☺

# Aplikacja, która ma szansę działać poprawnie

## Podejścia:

- **Zwiększenie odporności aplikacji na możliwe niezgodności?**
- **Eliminacja niezgodności poprzez definiowanie środowiska**
  - **Maszyny wirtualne**
  - **Kontenery aplikacji**

# Kontenery Aplikacji

- OS-Level virtualization dla grupy procesów
- Wirtualizacja środowiska (**sandbox**) a nie sprzętu (**VM**)
- To samo jądro dla wszystkich kontenerów i "Host OS"
- Mechanizmy kernela: **cgroups, namespaces (Linuks)**
- Dużo "lżejsze" niż VM przy zbliżonym poziomie separacji procesów

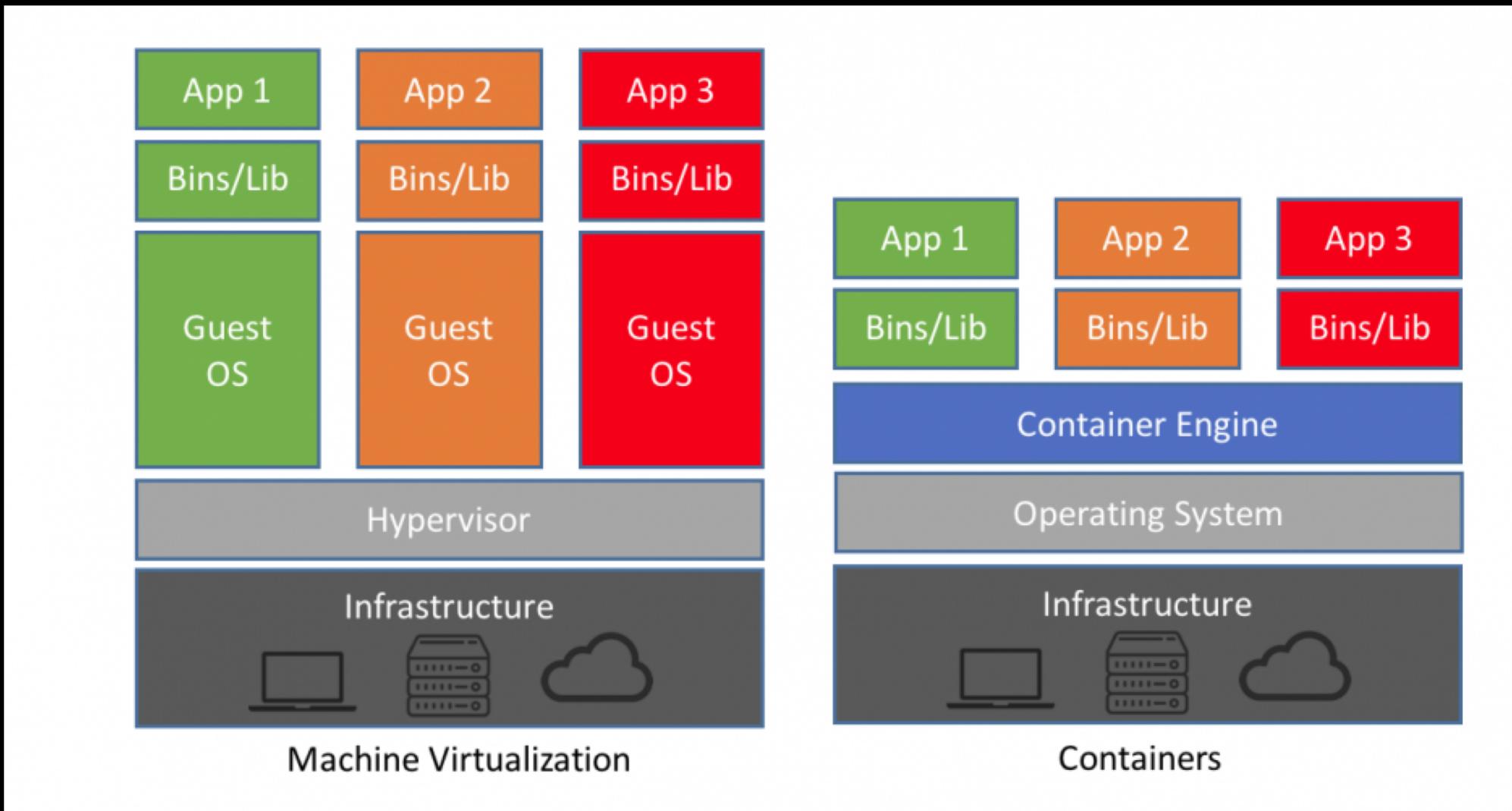
# Kontenery Aplikacji, punkt widzenia aplikacji

- Proces nie jest świadomy że działa w kontenerze
- Kontener "wygląda" jak normalny system Linuksowy
- Poza pewnymi "dziwnymi" zjawiskami (mój PID to... 1?)
- Wszyscy żyjemy w Matriksie ☺ (domyślne namespace'y)

# Kontenery Aplikacji - Historia

- 1979, Unix - chroot
- 2000s, FreeBSD - Jails
- 2006 – 2013, Linux: cgroups, namespaces
- 2008, Linux – LXC
- 2013, Docker (używał wówczas LXC)
- 2014, Rkt (CoreOS)

# Kontenery Aplikacji vs VM



# Czym jest Docker?

- **Technologia i ekosystem gotowych rozwiązań**
- **Definiowanie środowiska pracy aplikacji**
- **Niezależne od faktycznej platformy, na której jest uruchamiana**
- **Linux / Windows / macOS, fizyczny serwer / chmura**

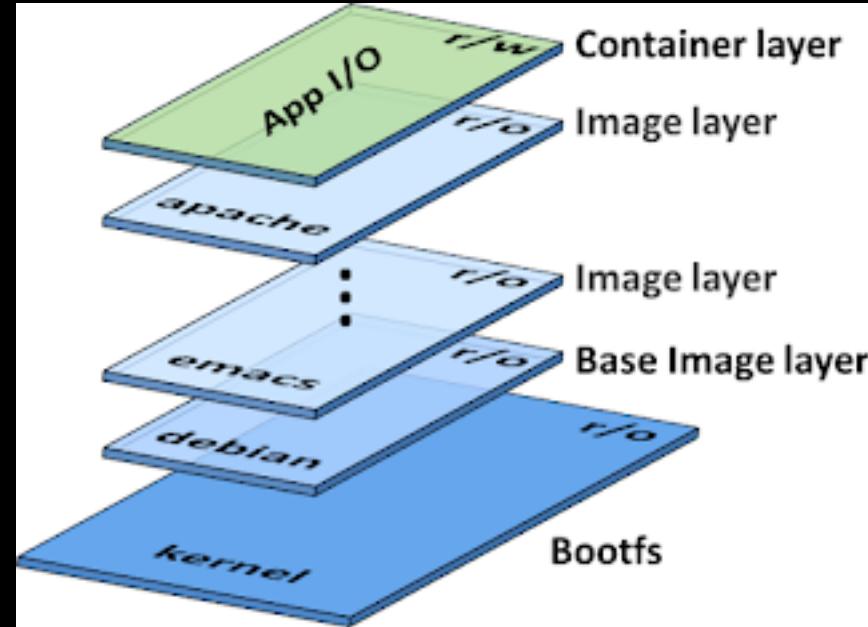
# Kluczowe technologie

- **Docker image** - artefakt
- **Docker registry** - ekosystem
- **Docker engine** - runtime
- **Docker client: docker – cli**

\* *Docker jest napisany w języku Go*

# Docker image

- Layers, CoW, Caching...
- Budowane z warstw (danych)
- Tylko do odczytu (immutable), oprócz ostatniej
- Można je cache'ować (wiele kontenerów na jednym hoście)
- Dystrybucja tylko różnic
- "Pra-przodek": *scratch*

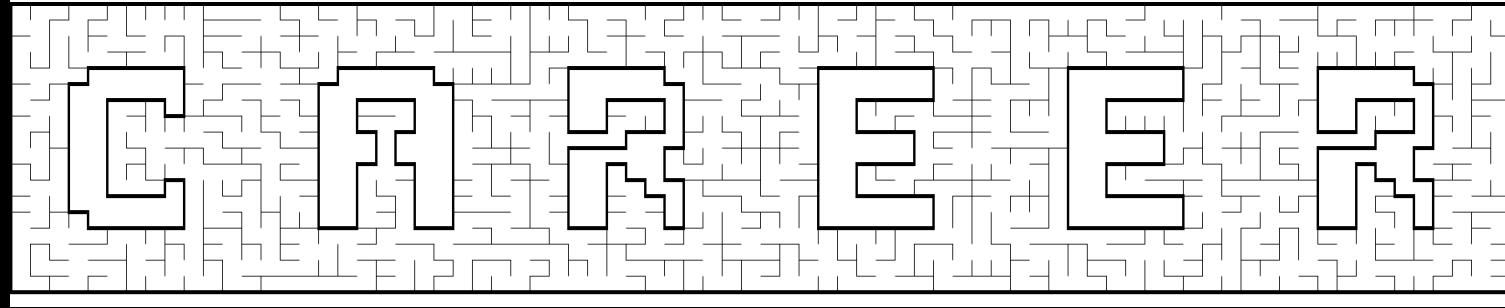
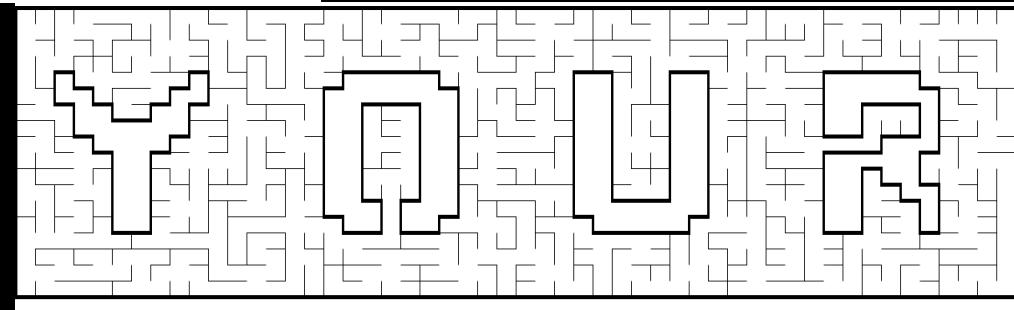
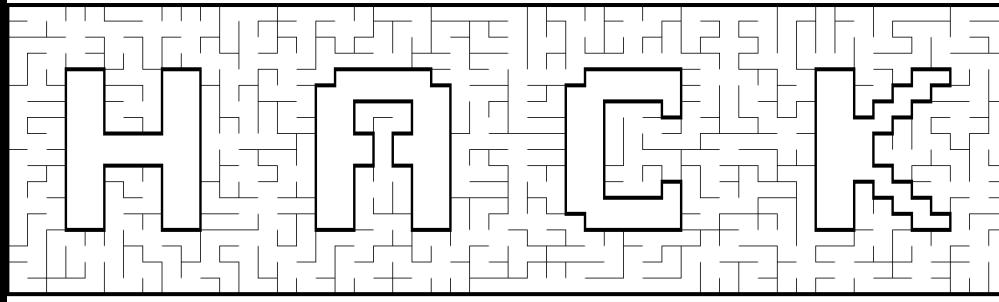


# Docker registry

- <https://hub.docker.com/>
- Inne... (np. eu.gcr.io)
- Mogą być prywatne bądź publiczne

# Docker engine

- **Lokalny silnik menedżera kontenerów**
- **Tworzy kontenery z obrazów**
- **Obsługuje środowisko procesu, mapowanie portów, wolumenów, limity, itd..**
- **Wykorzystuje funkcjonalność jądra Linuks**
- **Nie zmienia kluczowych mechanizmów, ale wprowadza swoją konwencję ich wykorzystania**
- **Jeden ze "sposobów na kontenery". Inne: Rkt, LXC, manualnie**



# Docker i co dalej

- Łatwa "dokeryzacja" pojedynczego procesu
- Łatwa dystrybucja obrazów
- Co z wdrożeniem całego systemu?
- Networking, Service Discovery, Scalability, Monitoring...
- Ze "stajni" Dockera: Docker Compose, Swarm
- Nasza propozycja: Kubernetes

# Kubernetes podstawy

Jakub Kabza, SAP  
November 20, 2018

PUBLIC



# Agenda - Kubernetes

- Czym jest?
- Czym różni się od Dockera?
- Po co mi to?
- Z czego się składa?
- Jak działa?



# Kubernetes

- Projekt typu Open-source rozwijany przez Google od 2014 r. (Go)
- System zarządzania skonteneryzowanymi aplikacjami
- Niskopoziomowa platforma oferująca ogromne możliwości rozszerzania i konfiguracji



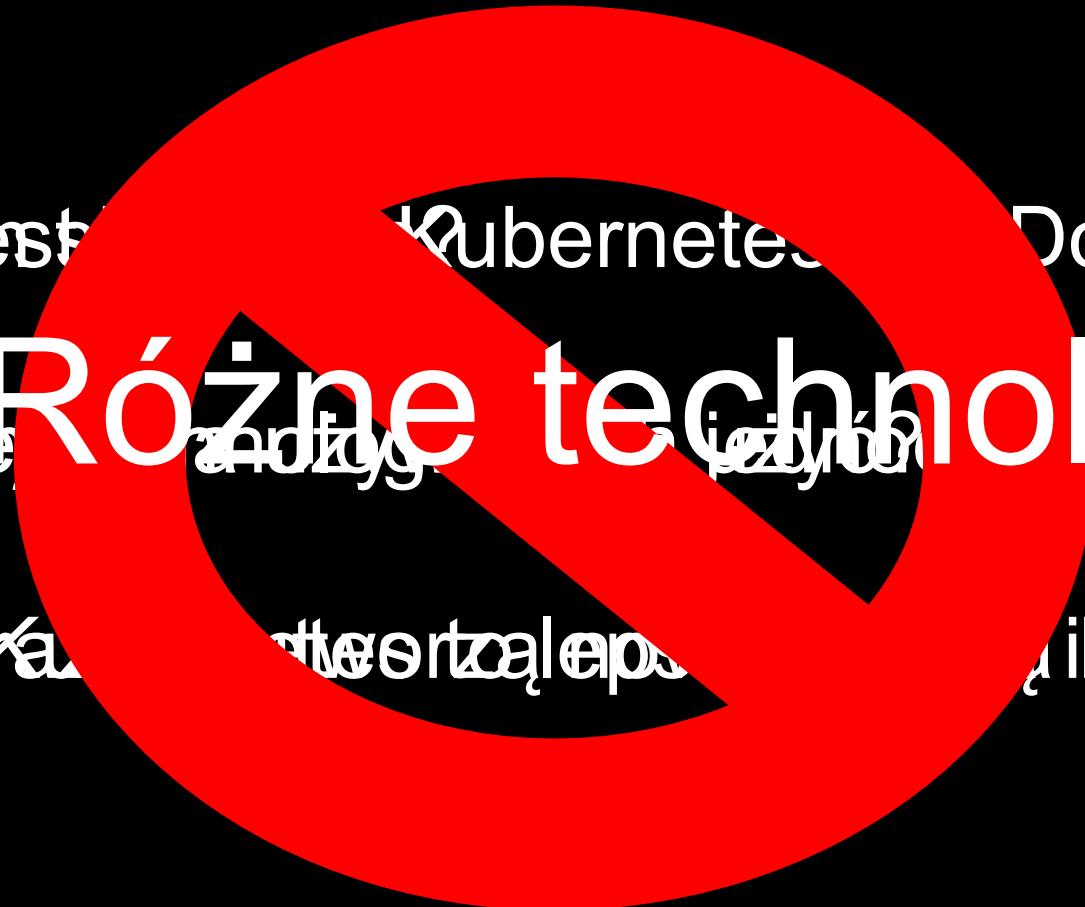
Κυβερνήτης

IPA: /ky.be.rnɛ:.tɛ:s/

1. gubernator, rzadca
2. (na statku) kapitan
3. (w samolocie) pilot

# Kubernetes vs Docker

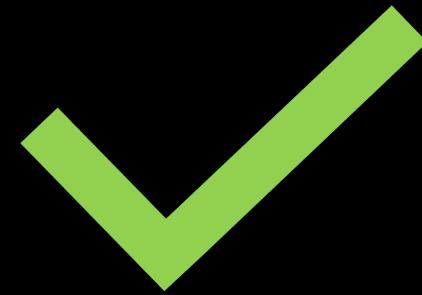
- Czy jestem zmuszony do wyboru Kubernetes i Docker?
- Które technologie są analogiczne do Kubernetes i Docker?
- Czy kiedykolwiek słyszałem o Kubernetes i Docker?



Docker?

Różne technologie?

i Docker?



# Po co mi Kubernetes?

- Docker jest super, stworzyłem świetną aplikację i chcę się nią pochwalić!
- Wszyscy kochają moją apkę! Pojedyncza instancja to za mało. Chciałbym zintegrować moją aplikację z innym mikroserwisem.
  - Uruchamianie
- Jak wyskalować moją aplikację Zarządzanie
- Jak moje kontenery skądś się rozmawiać? Skalowanie
- Co się stanie, kiedy jeden z moich kontenerów padnie?



# Deployment

Nodes  
HAYO  
CKUR  
CAREER

kubectl scale deployment.v1

Ma

Pod 1  
Replica 1

```
You, a few seconds ago | 2 authors (Kyma Bot and others)
apiVersion: apps/v1beta1
kind: Deployment
metadata:
  name: redis-c
spec:
  replicas: 4
  selector:
    matchLabels:
      app: redis-client
  template:
    metadata:
      labels:
        app: redis-client
    spec:
      containers:
        - name: redis-client
          image: "appregistry/redis:3.2.9-r2"
```

-replica  
Pod 1  
Replica 2

Pod 1  
Replica 3

Pod 1  
Replica 4

Manager



API Server

Scheduler

Worker

Worker

Worker

Worker

Worker

Worker

Worker

# Źródła

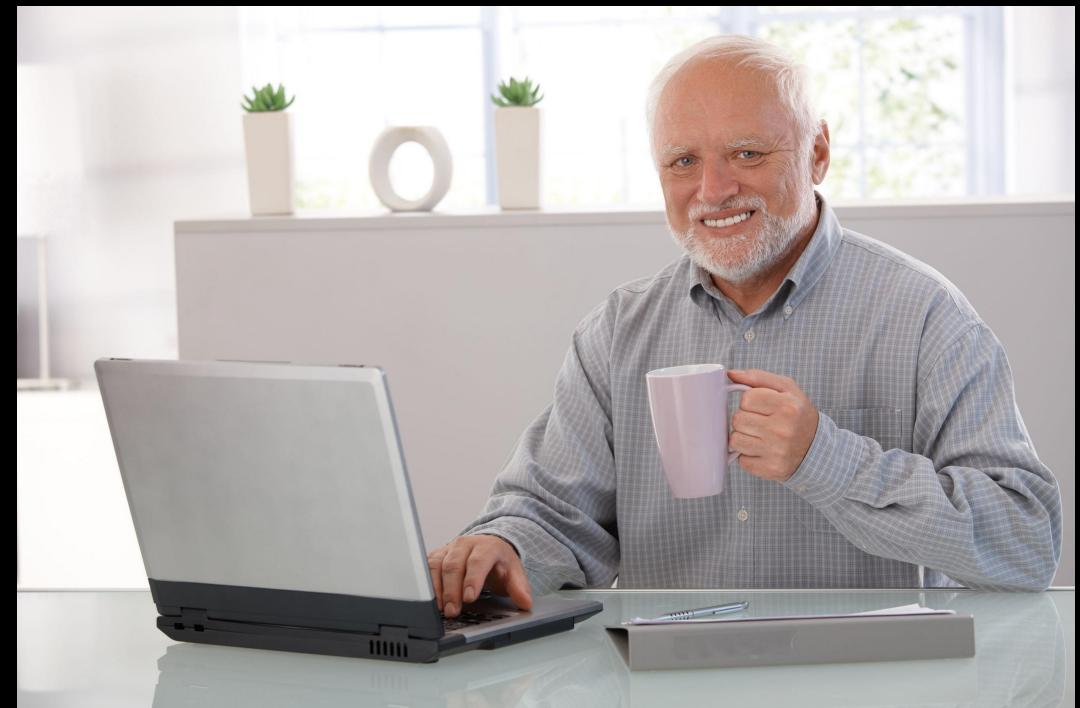
<https://kubernetes.io/>

<https://www.opencontainers.org/>

**IBM FSS FCI and Counter Fraud Management**

[VMware Cloud-Native](#)

<https://en.wiktionary.org/wiki/%CE%BA%CF%85%CE%B2%CE%B5%CF%81%CE%BD%CE%AE%CF%84%CE%B7%CF%82>

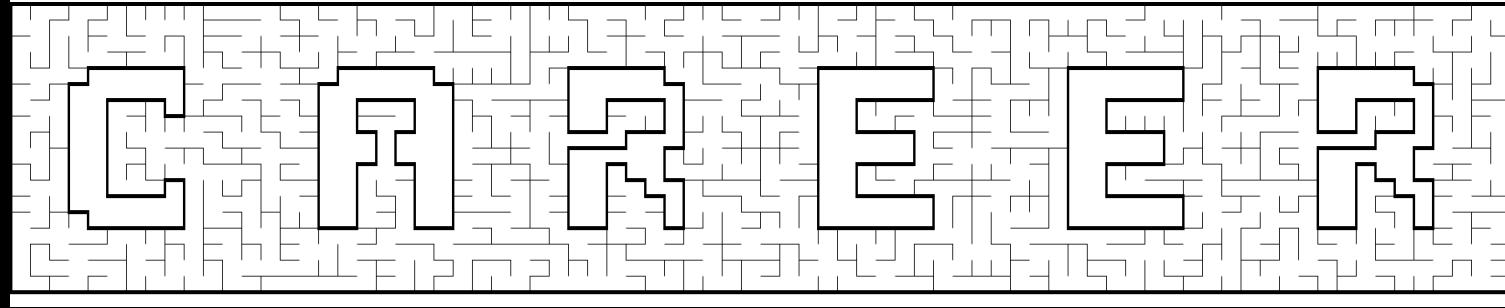
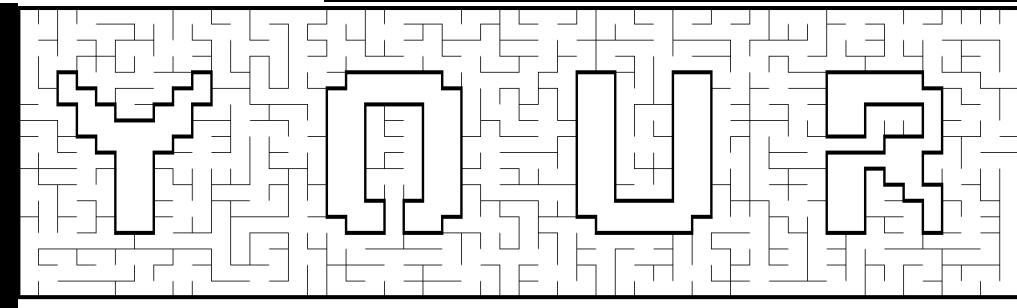
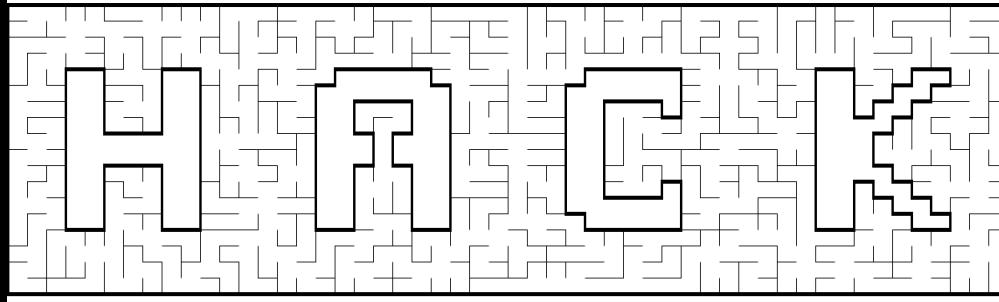


# Dziękuję

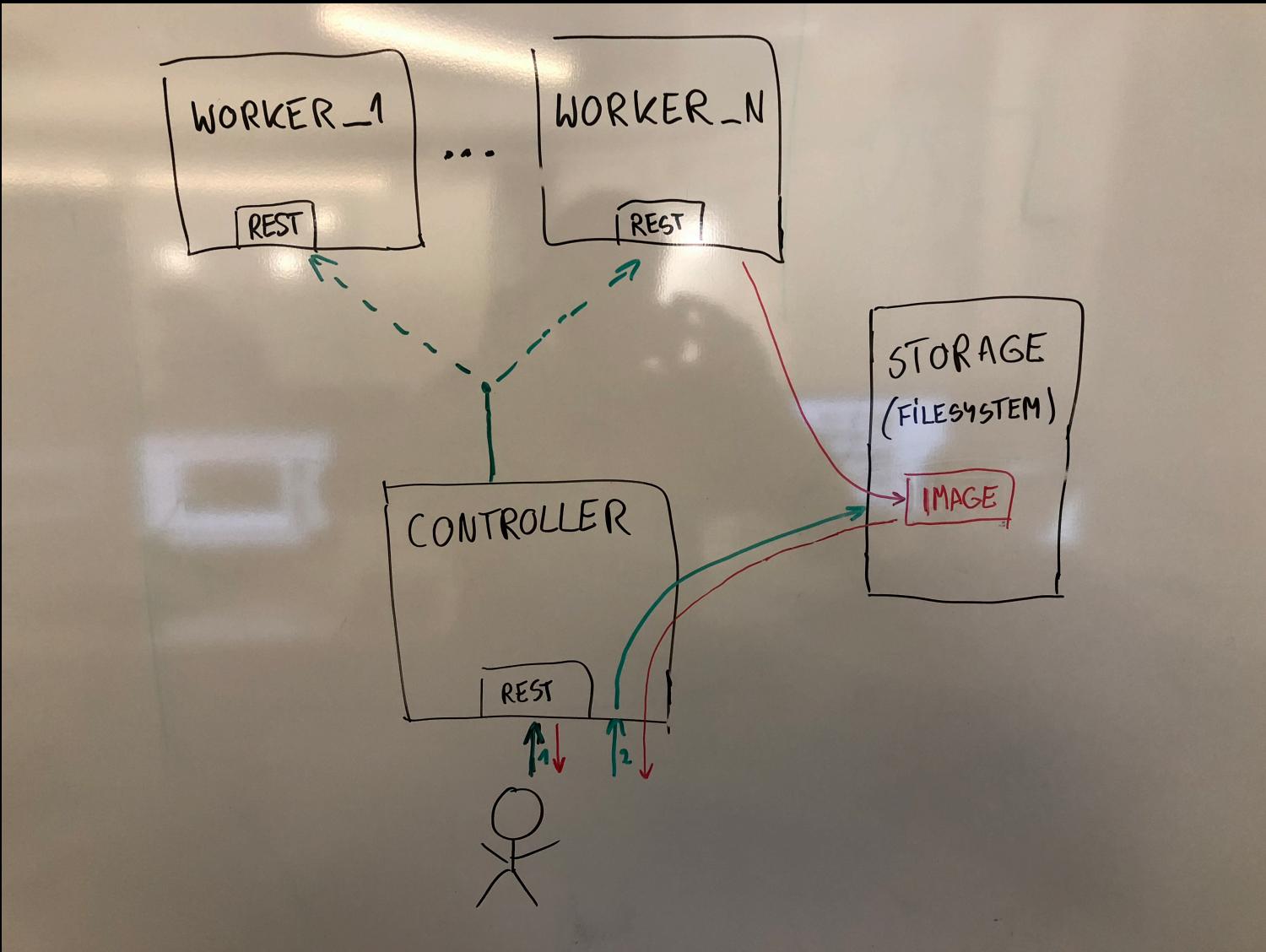
Jakub Kabza

jakub.kabza@sap.com

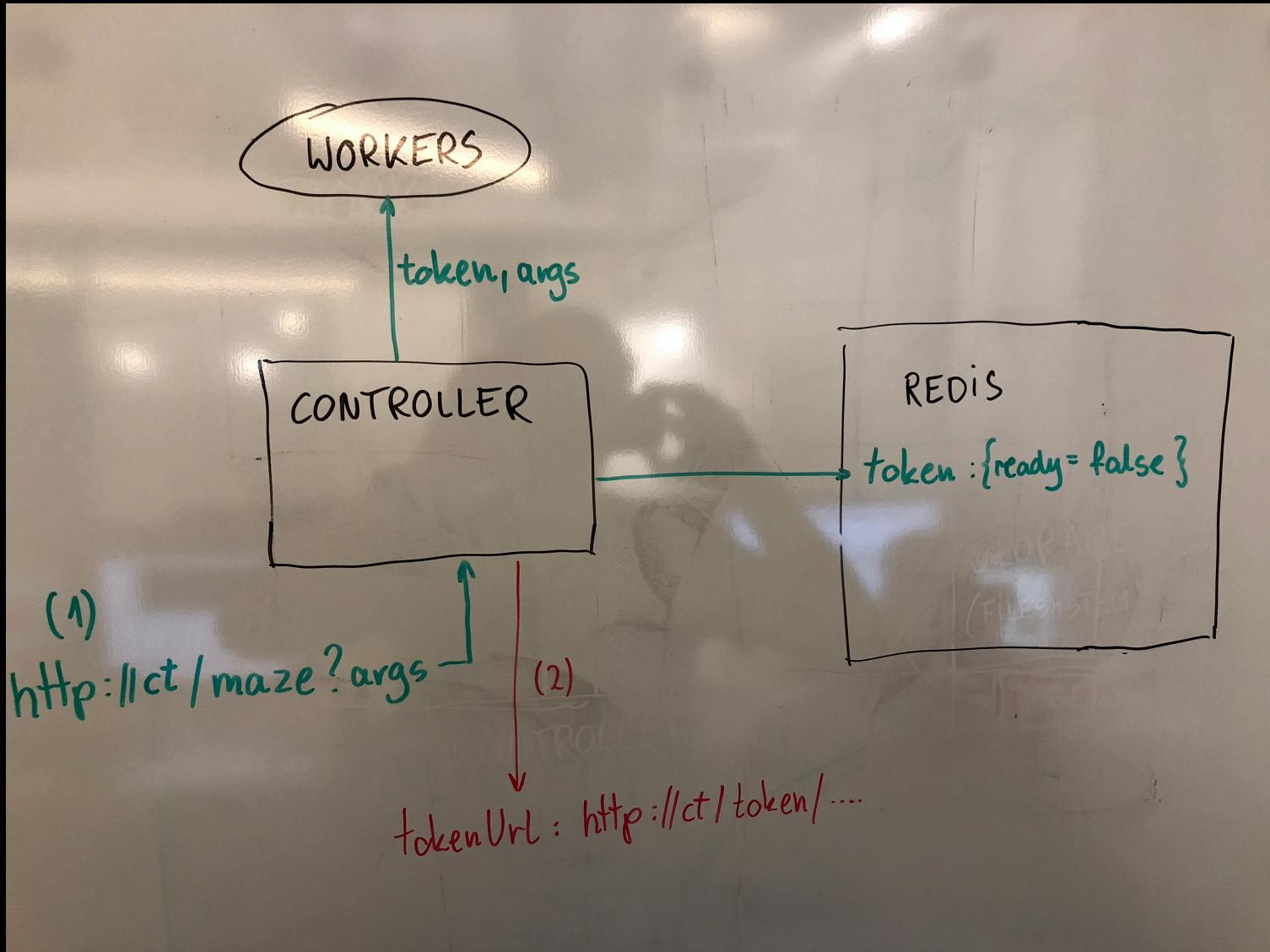




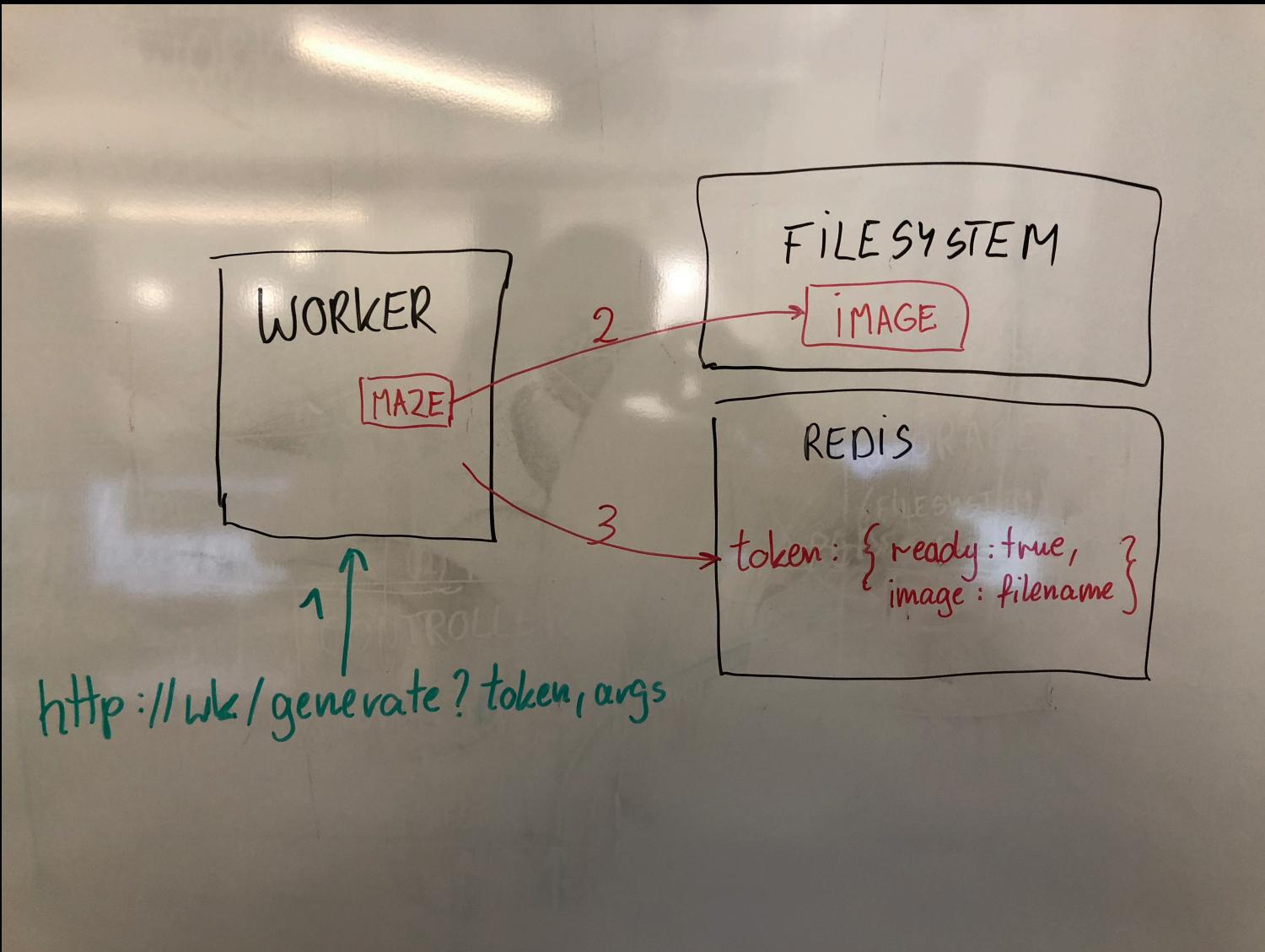
# Aplikacja: prototyp



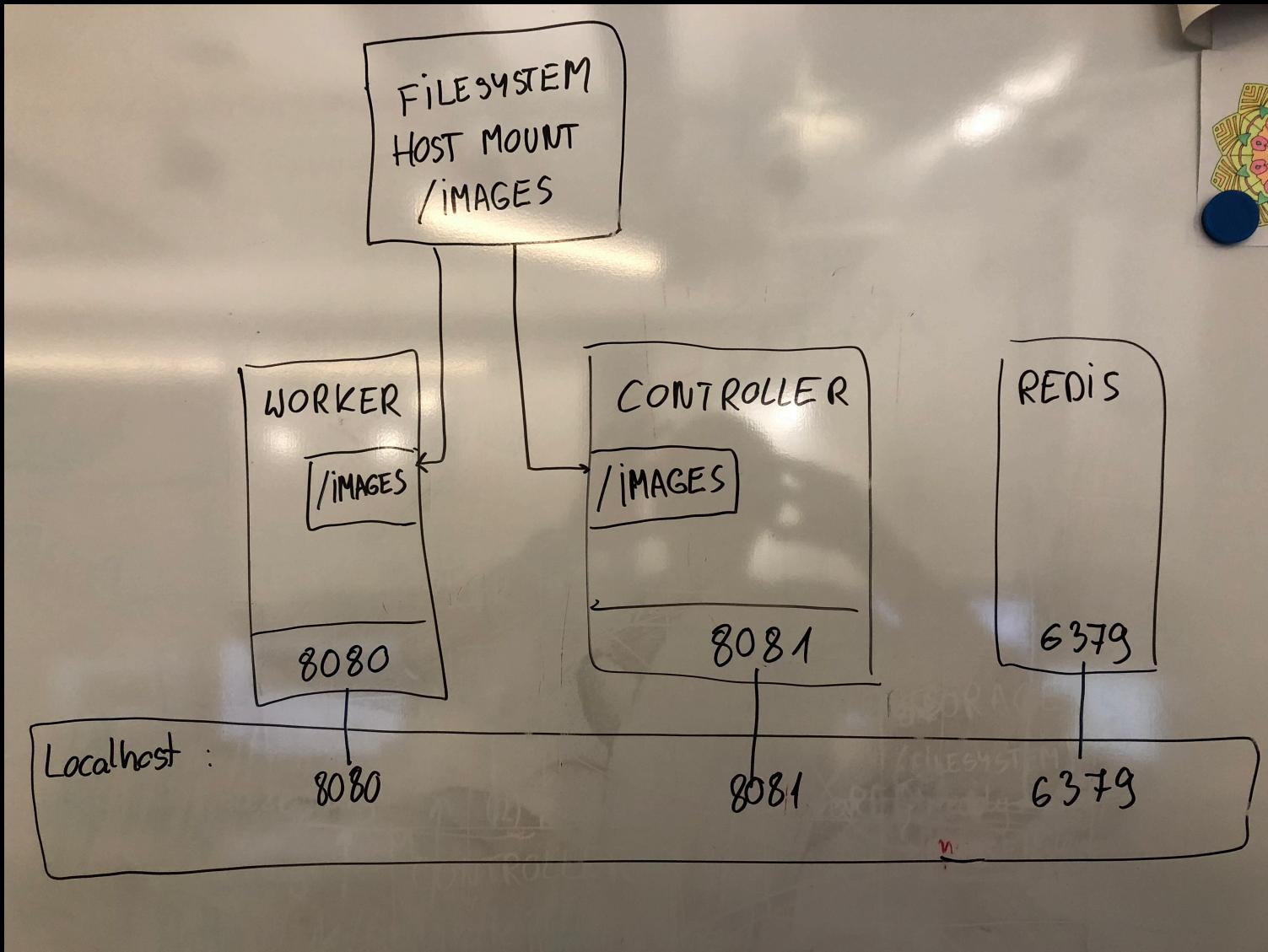
# Aplikacja: controller

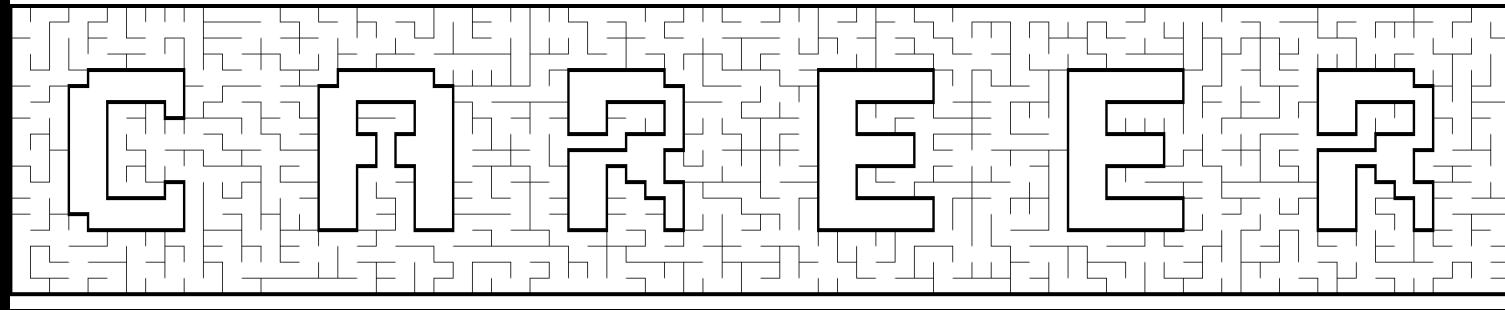
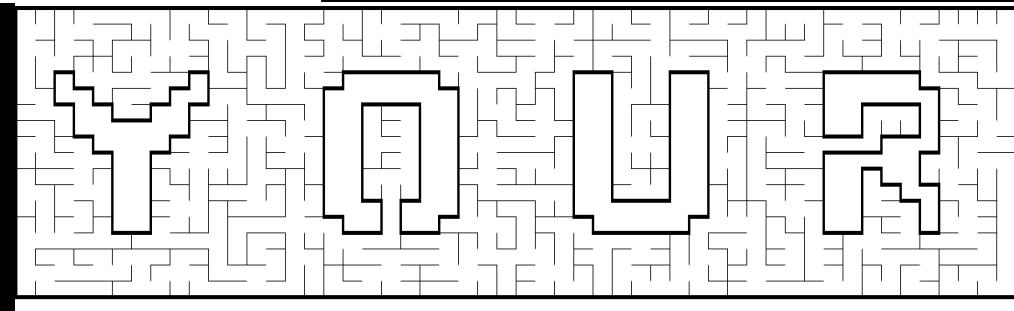
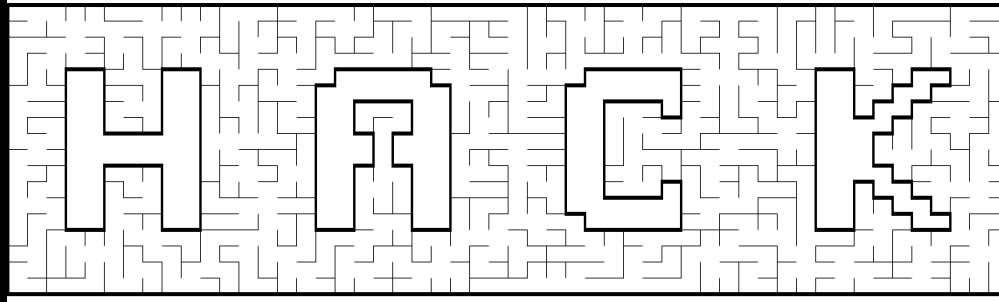


# Aplikacja: worker

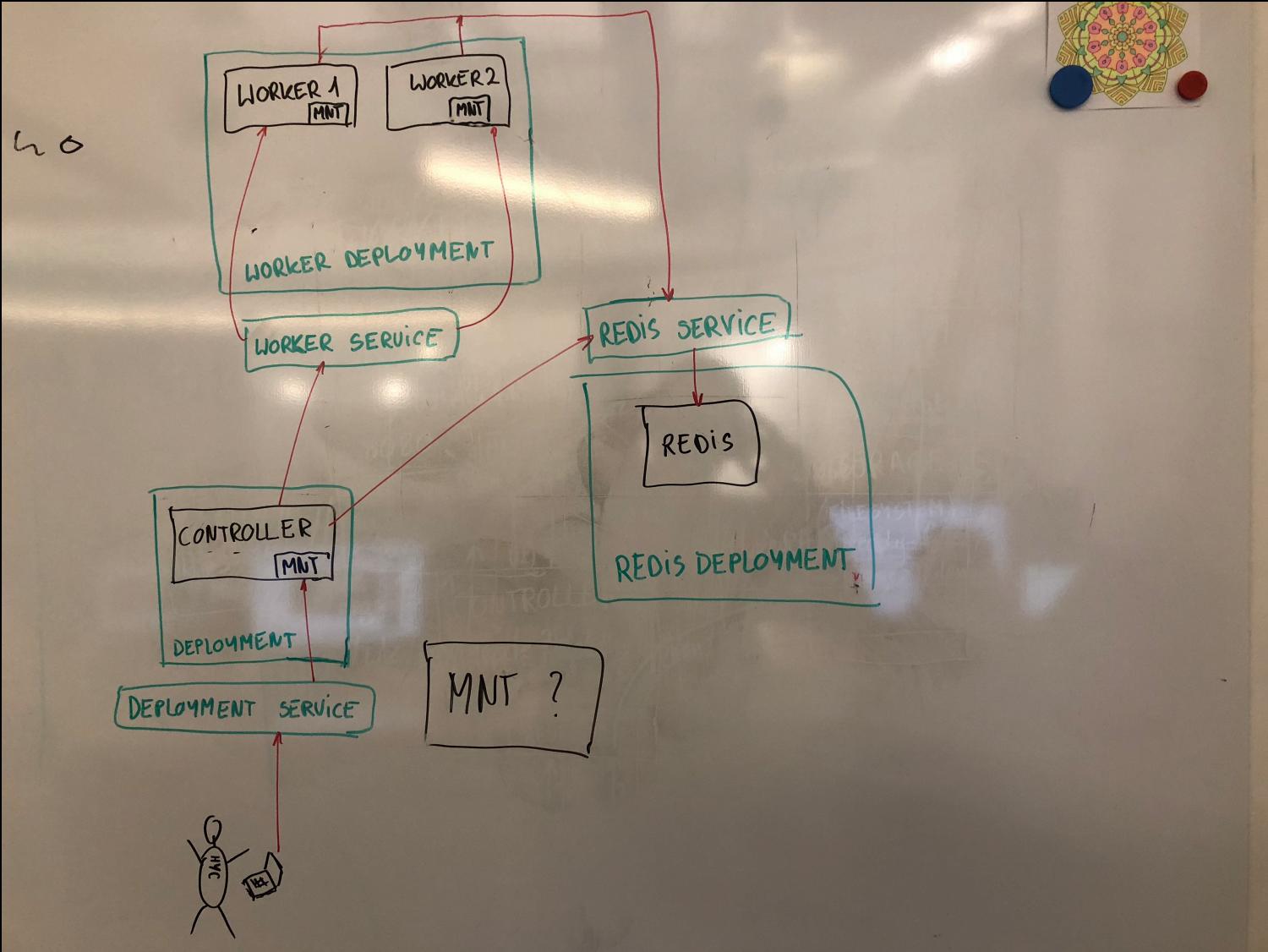


# Aplikacja: Lokalny development





# Aplikacja: Kubernetes



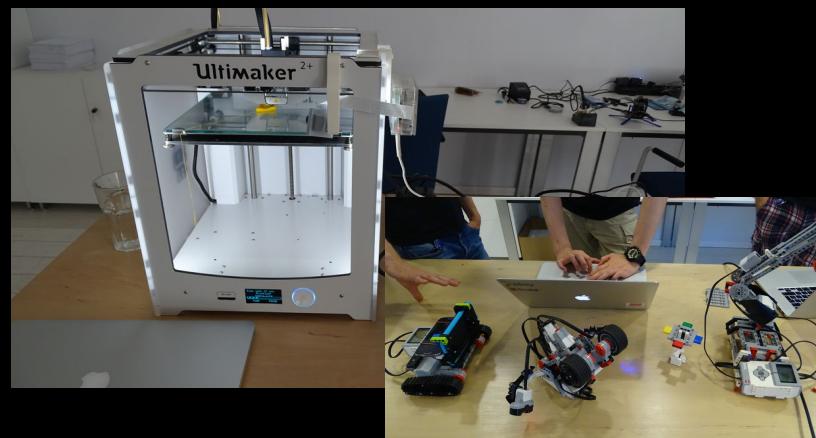
# Kogo szukamy?

## Studentów

- **Students in Development Teams**  
*(Support engineers, Product Development, Open Source)*
- **Hack Team (recruitment starts in April)**  
*(3 months internship, 2 teams realizing dedicated projects)*

## Profesjonalistów

- **Software Developers**  
*(Cloud Native Solutions, Go, Java)*
- **DevOps Engineers**  
*(Kubernetes, Cloud Foundry)*
- **Support Engineers**  
*(Java Spring)*



# Networking session ☺

## Opcja 1

- Zapraszamy na zwiedzanie biura



## Opcja 2

- Q&A z naszymi prowadzącymi

## Opcja 3

- Speed Interviews



Thank you.

