

# Effective Kubernetes

Keep it Simple

Wojciech Barczyński - Lead Developer | System Engineer

SMACC.io

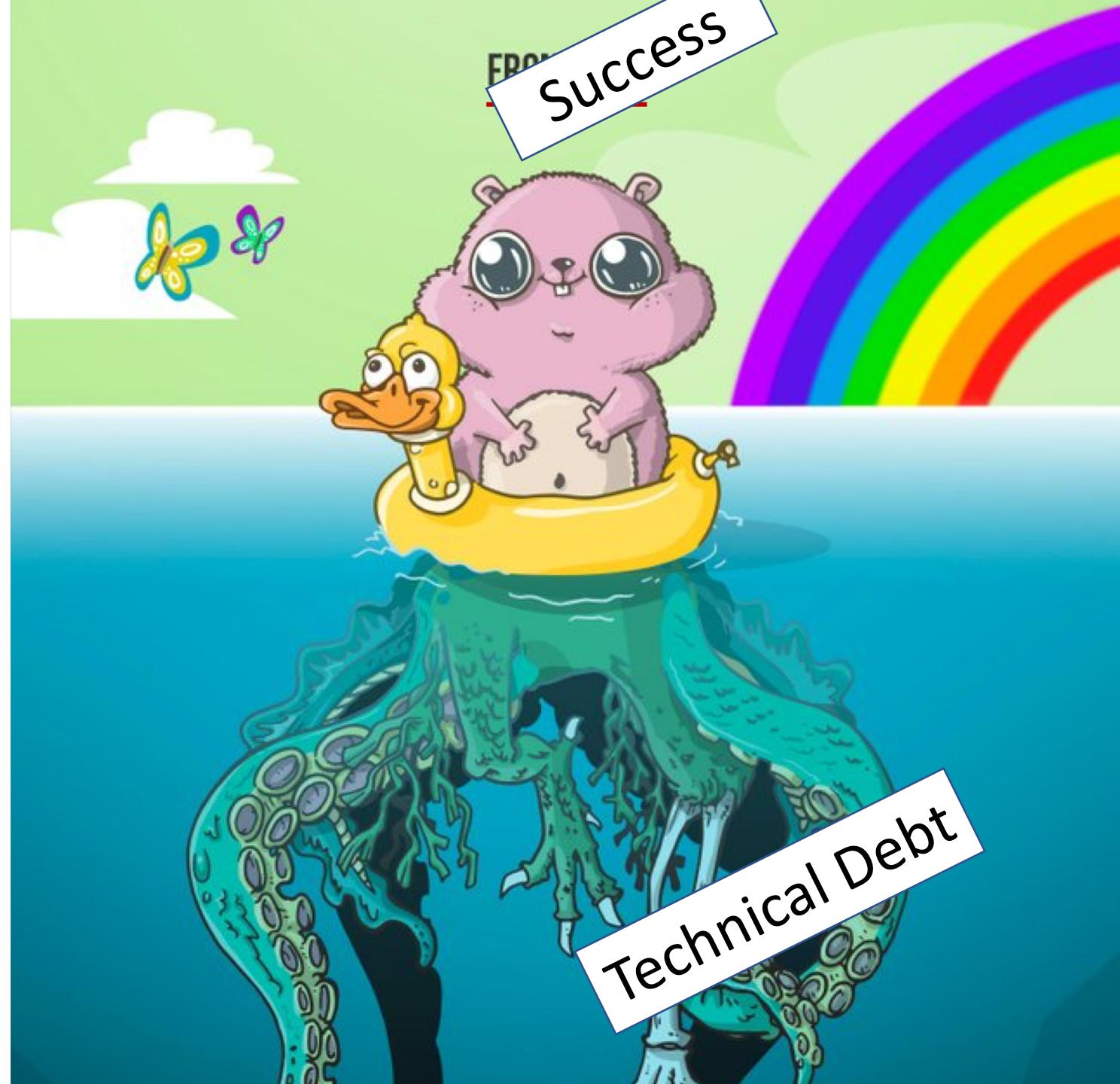
Slow delivery

Continuous Deployment?

Fear

Frustration

XX% Idle Machines





2016



# Black (Blue) Box

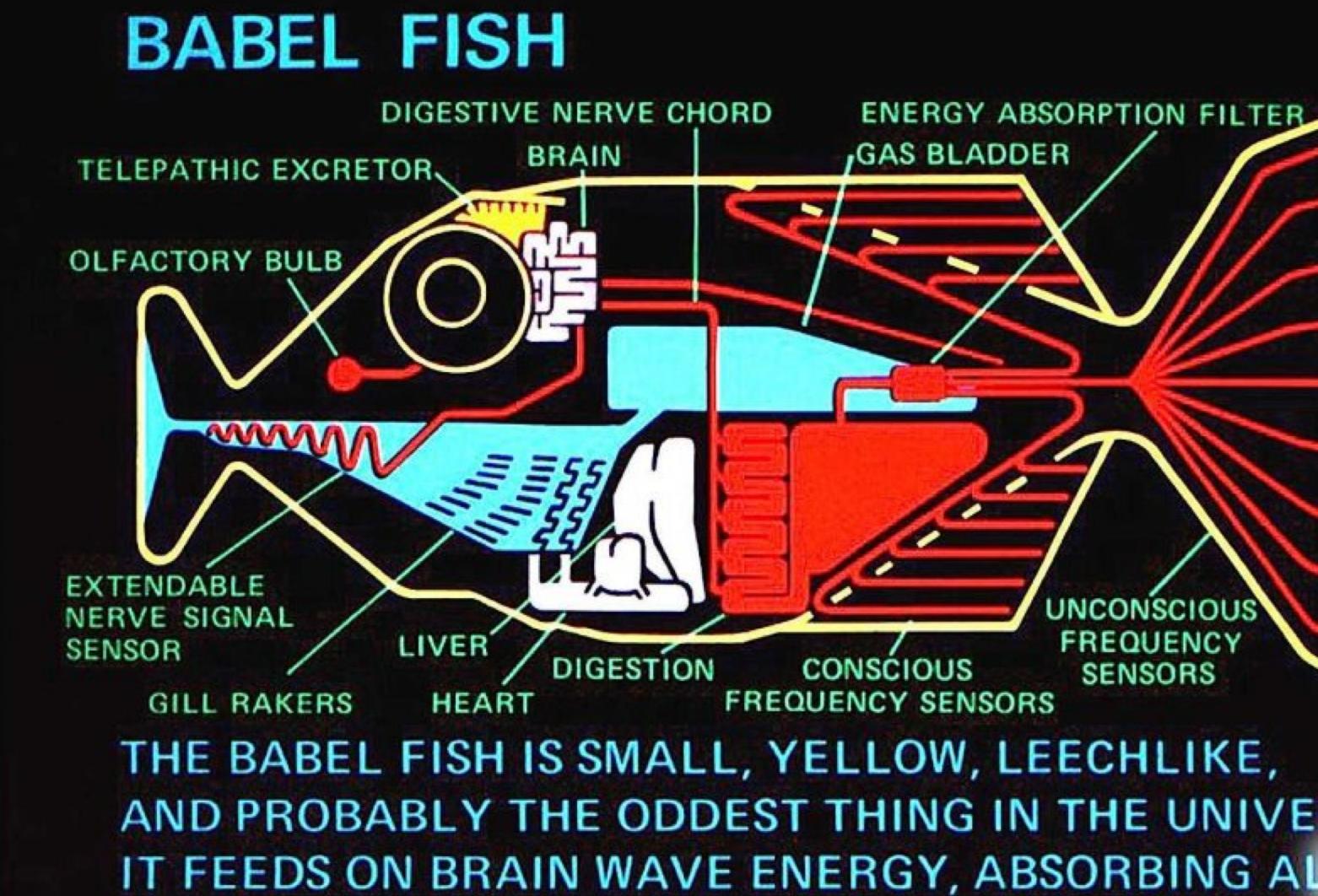
Infrastructure  
(almost) invisible

Easy\* Continuous  
Deployment



[https://en.wikipedia.org/wiki/File:Dr\\_Who\\_\(316350537\).jpg](https://en.wikipedia.org/wiki/File:Dr_Who_(316350537).jpg)

# Common Language Artifacts Platform

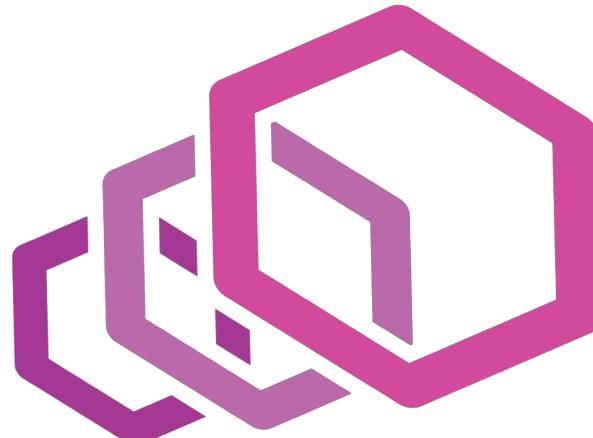
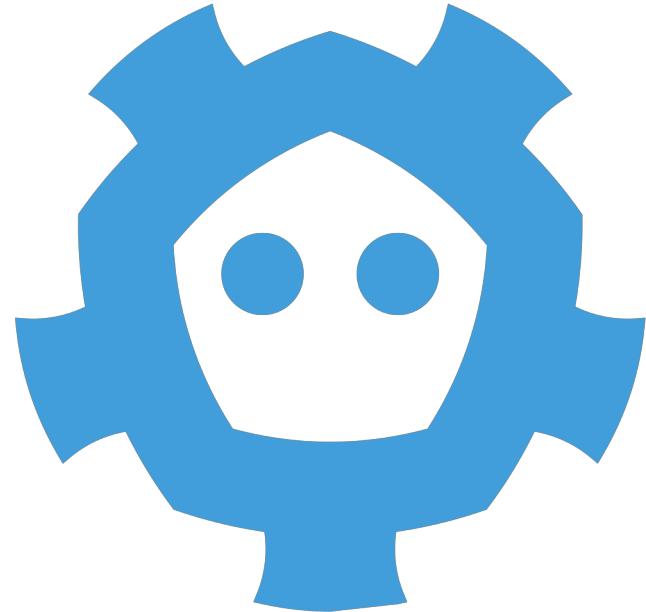




# Learn-as-you-go

# Learn-as-you-go

1. Deploy Cloud-Native app
2. Make a Hell of Mistakes
3. Get it right or Postpone



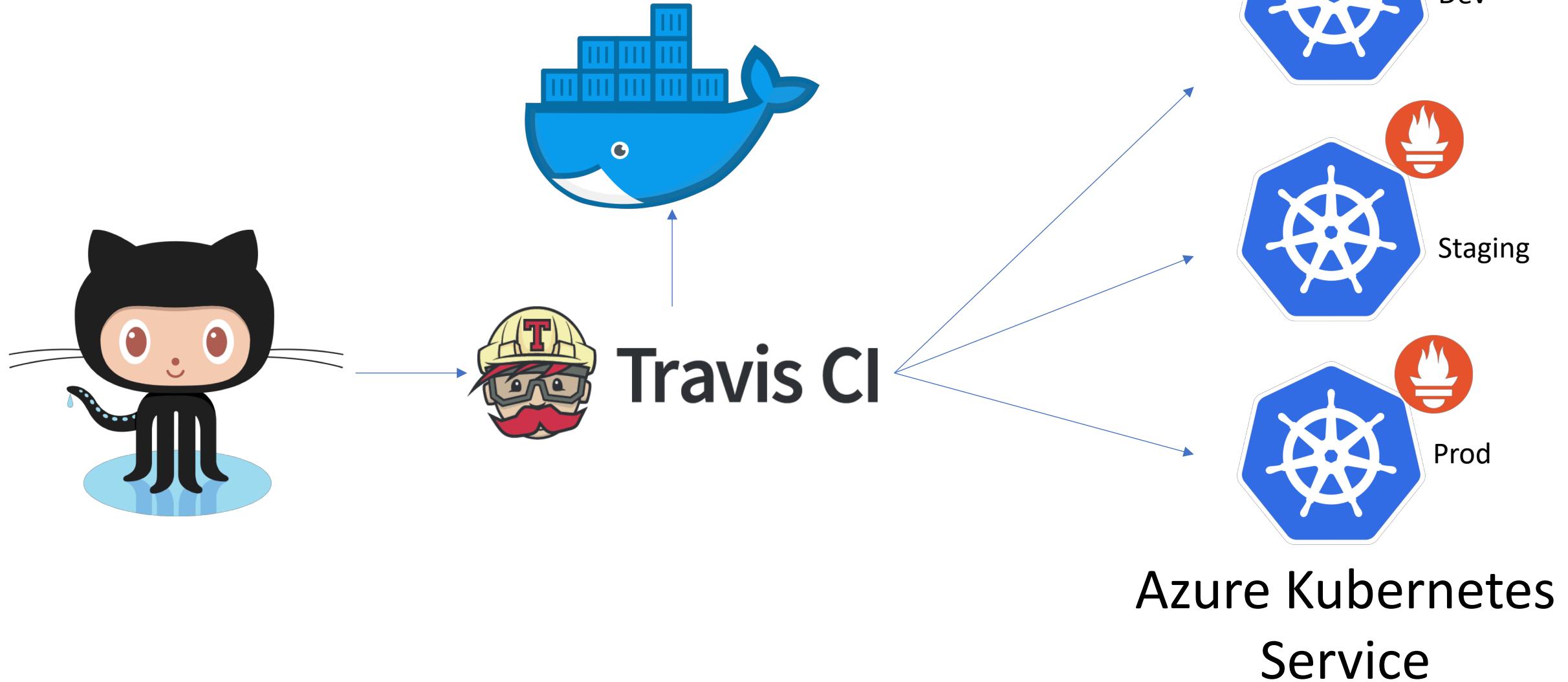
envoy



Keep it simple

BRUNO  
BRUJAH  
Photographer

# Continuous Deployment



# Continuous Deployment

1. make run\_local
2. Code on master → develop env



Development

# Continuous Deployment

- 3. Git tag → staging env
- 4. PR accepted → production env



Staging



Production

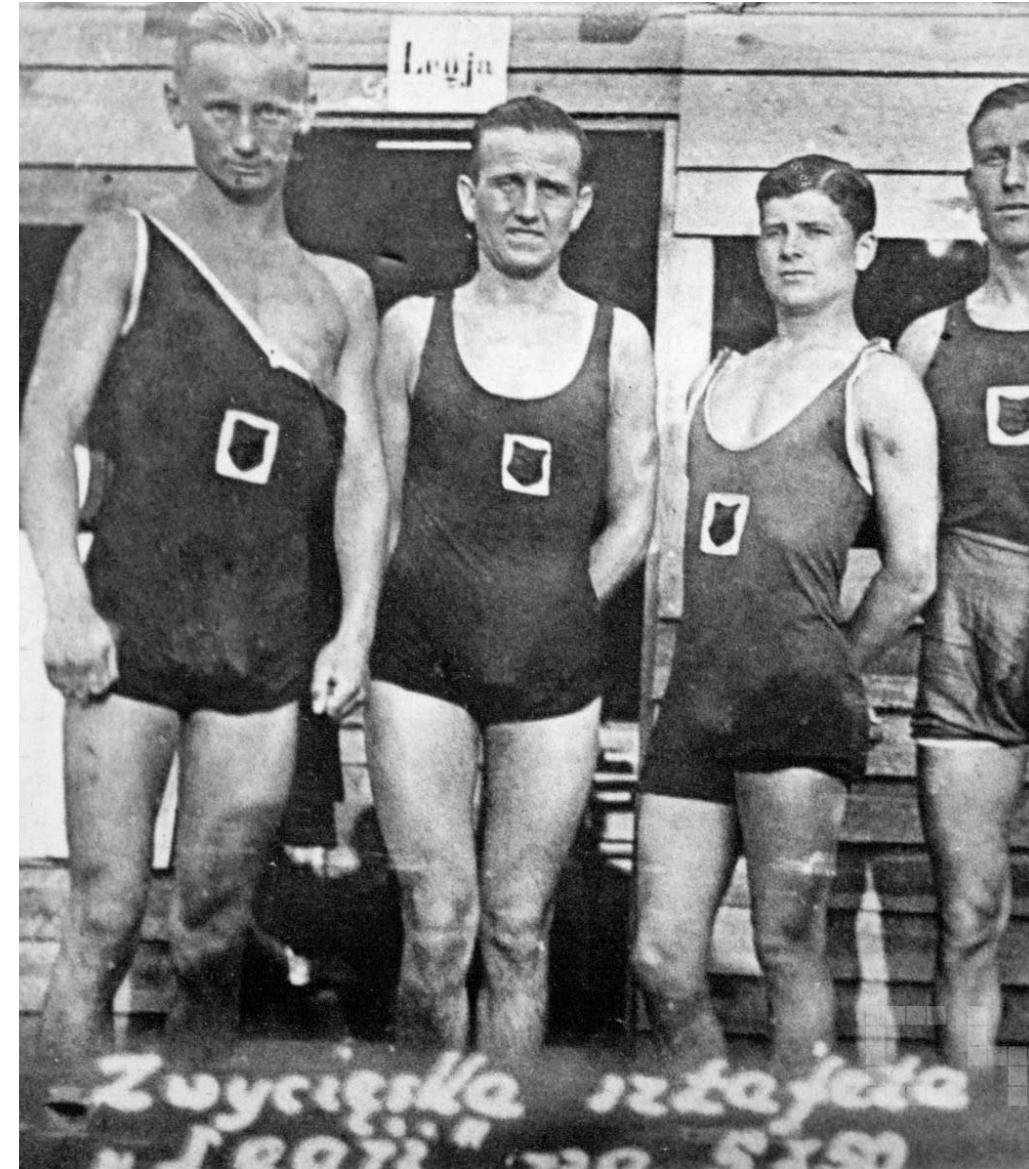
# Keep everybody in the process

1. Teach the team  
Kubernetes definitions
2. Keep the process  
understandable



# Keep everybody in the process

3. Keep K8S files, ...  
easy to read
4. Do not terrorize with  
how-amazing-  
Kubernetes-is :D



# Keep everybody in the process

Copy & Paste:

1. Makefile
2. Kubernetes files
3. TravisCI

```
curl https://github.com/smacc-ci/deploy.sh | bash
```

# Keep everybody in the process

Copy & Paste:

1. Makefile
2. Kubernetes files
3. TravisCI

```
curl https://github.com/smacc-ci/deploy.sh | bash
```

# Conventions over tools!

- Common conventions for repos
- No a single deploying tool
- No encrypted data in repo

ps. Only when you are really really ready.



Application must be way smarter  
(12factorapps, coordination, metrics, ...)

# Effective Kubernetes



[https://www.flickr.com/photos/bruno\\_brujah/](https://www.flickr.com/photos/bruno_brujah/)

- Start with small iterations
- Learn-as-you-go

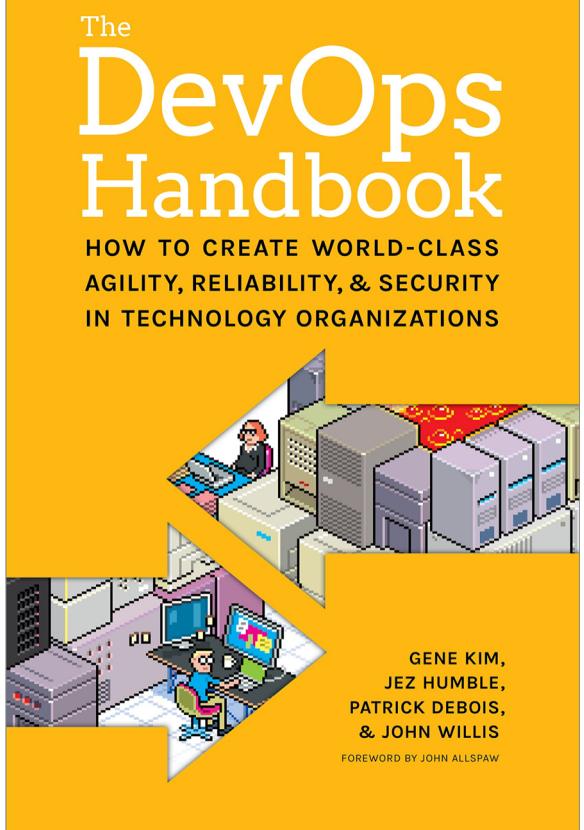
# Effective Kubernetes



[https://www.flickr.com/photos/bruno\\_brujah/](https://www.flickr.com/photos/bruno_brujah/)

- Simplicity
- Keep your Kubernetes understandable

*Hope k8s team keep it  
this way*



Three ways:

- Flow
- Feedback
- Continuous Experimentation and Learning

