

AZURE KUBERNETES SERVICE



Azure



Wojciech Barczyński - [Hypatos.ai](#)
wbarczynski.pro@gmail.com

WOJCIECH BARCZYŃSKI

- Lead Software Engineer & System Engineer
- Golang Warsaw Organizer
- Interests:
working software
- Hobby:
teaching software
engineering



BACKGROUND

- ML FinTech ➔ microservices and k8s
- Before:
 - 1 z 10 Indonesian mobile ecommerce
(Rocket Internet)
- Spent 3.5y with Openstack:
 - small team, 1000+ nodes, 21 data centers
- I do not like INFRA :D

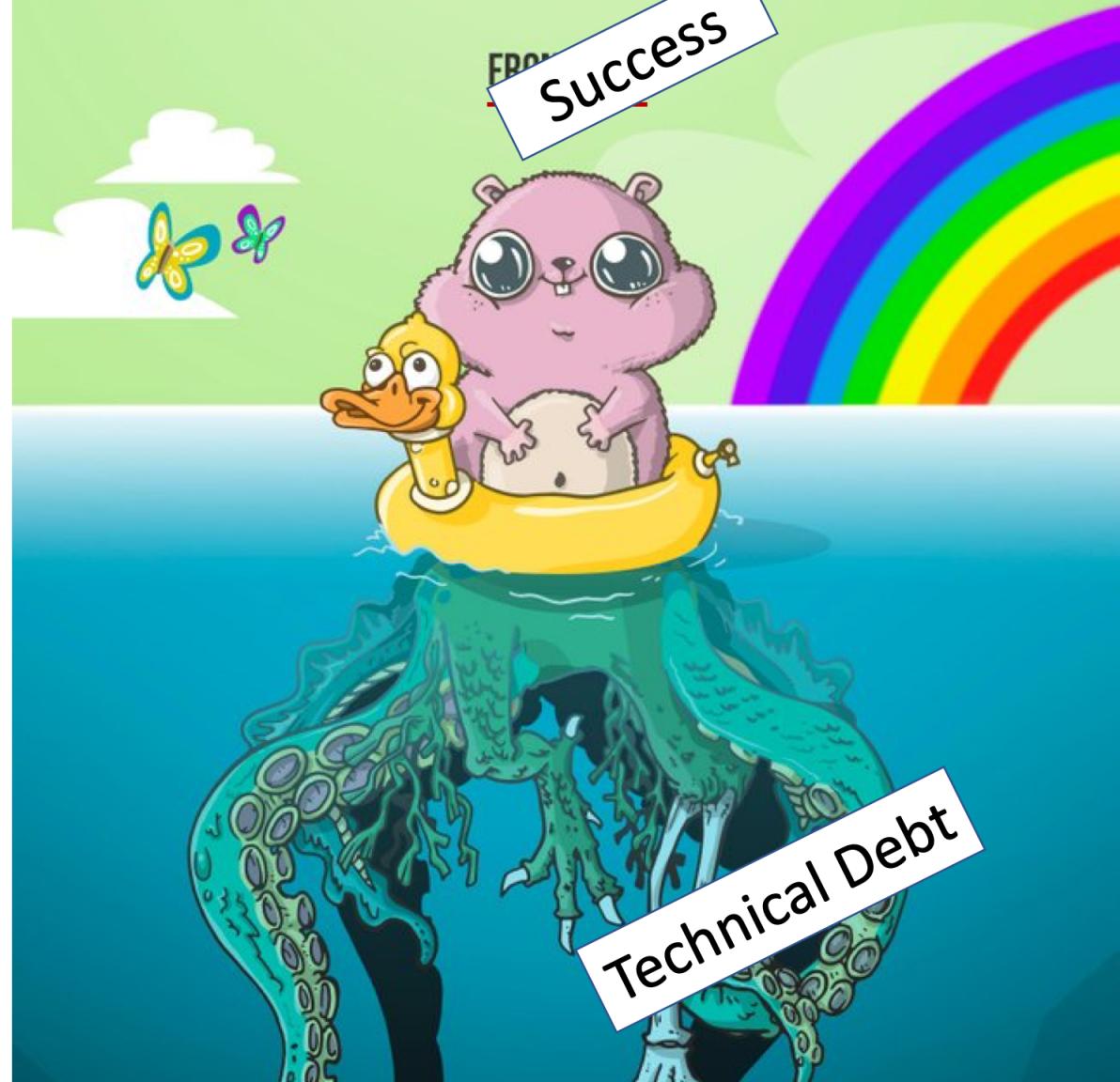
Slow delivery

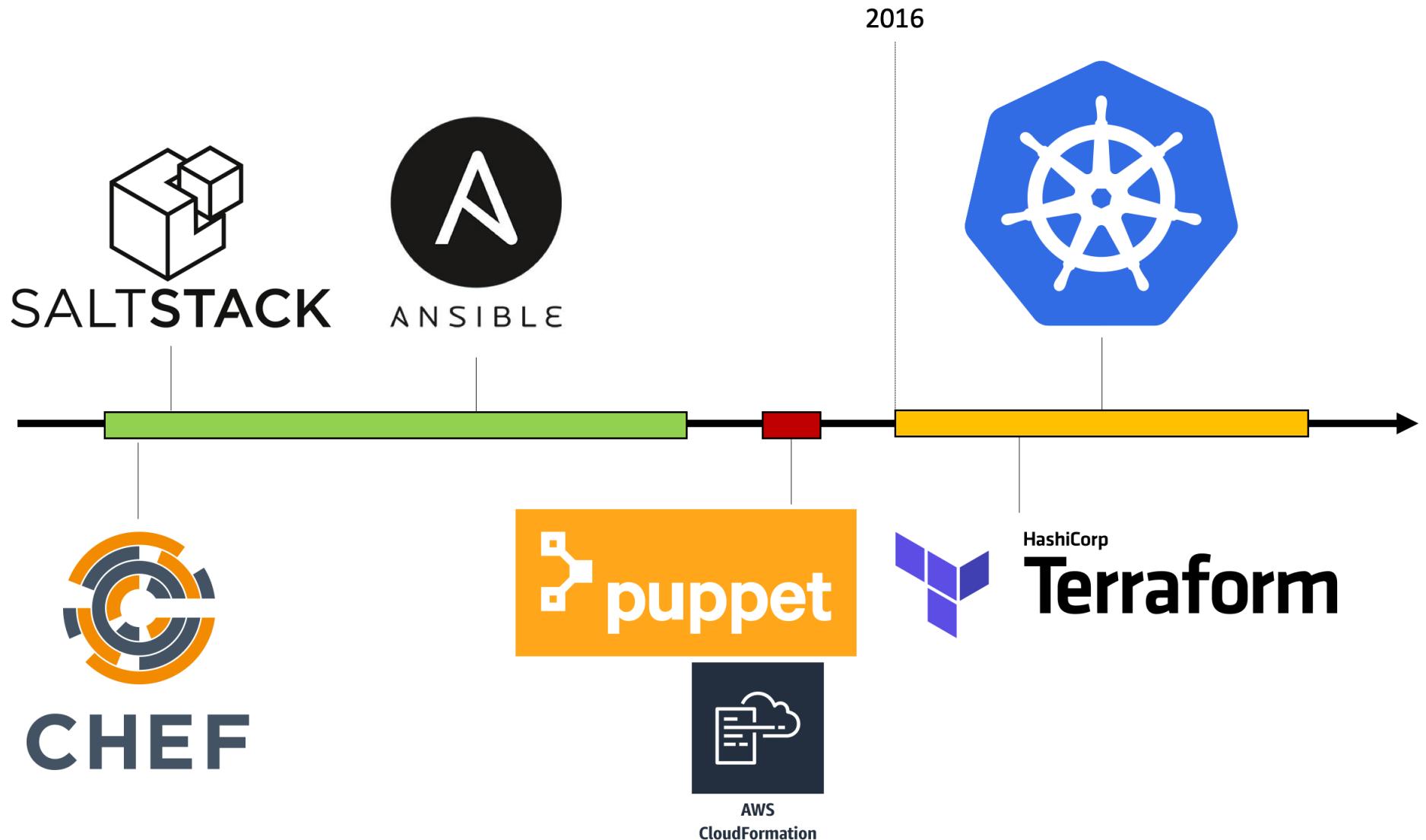
Continuous Deployment?

Fear

Frustration

XX% Idle Machines





Black (Blue) Box

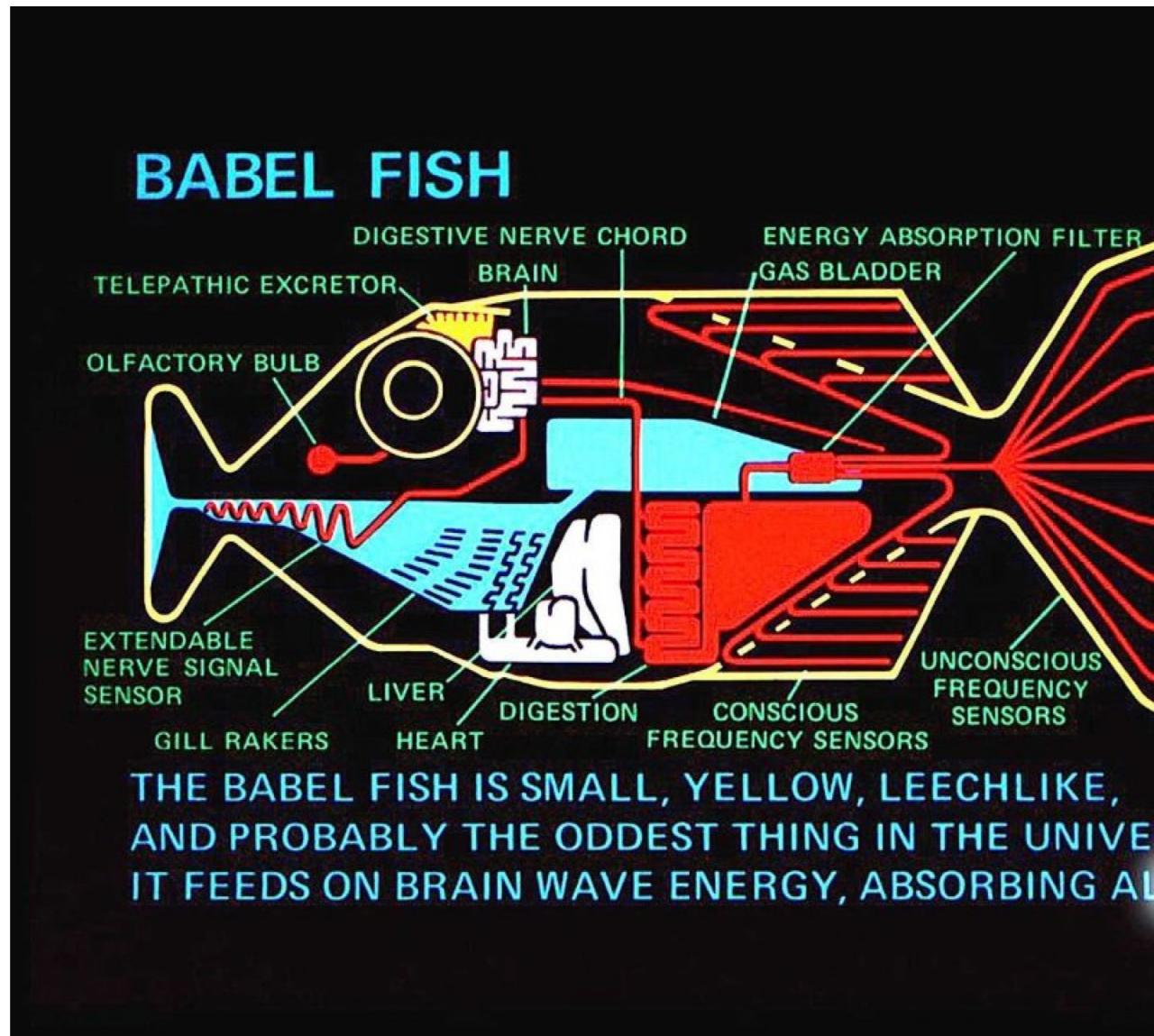
Infrastructure (almost) invisible

Easy* Continuous Deployment



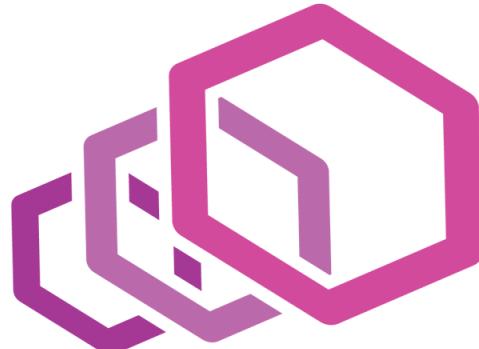
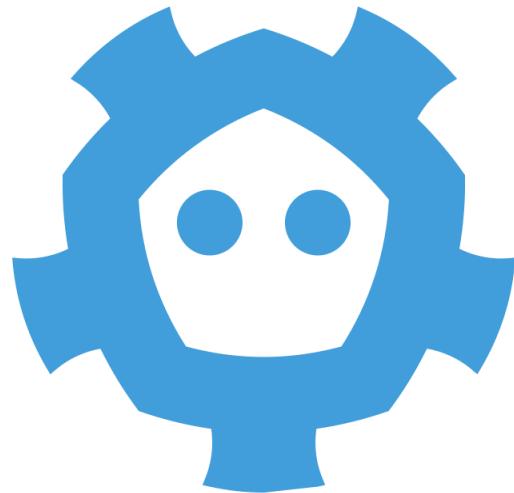
[https://en.wikipedia.org/wiki/File:Dr_Who_\(316350537\).jpg](https://en.wikipedia.org/wiki/File:Dr_Who_(316350537).jpg)

Common
Language
Artifacts
Platform



Learn-as-you-go

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



envoy



No free lunch - application way smarter
(12factorapps, coordination, metrics, ...)

<https://www.flickr.com/photos/160866001@N07/>

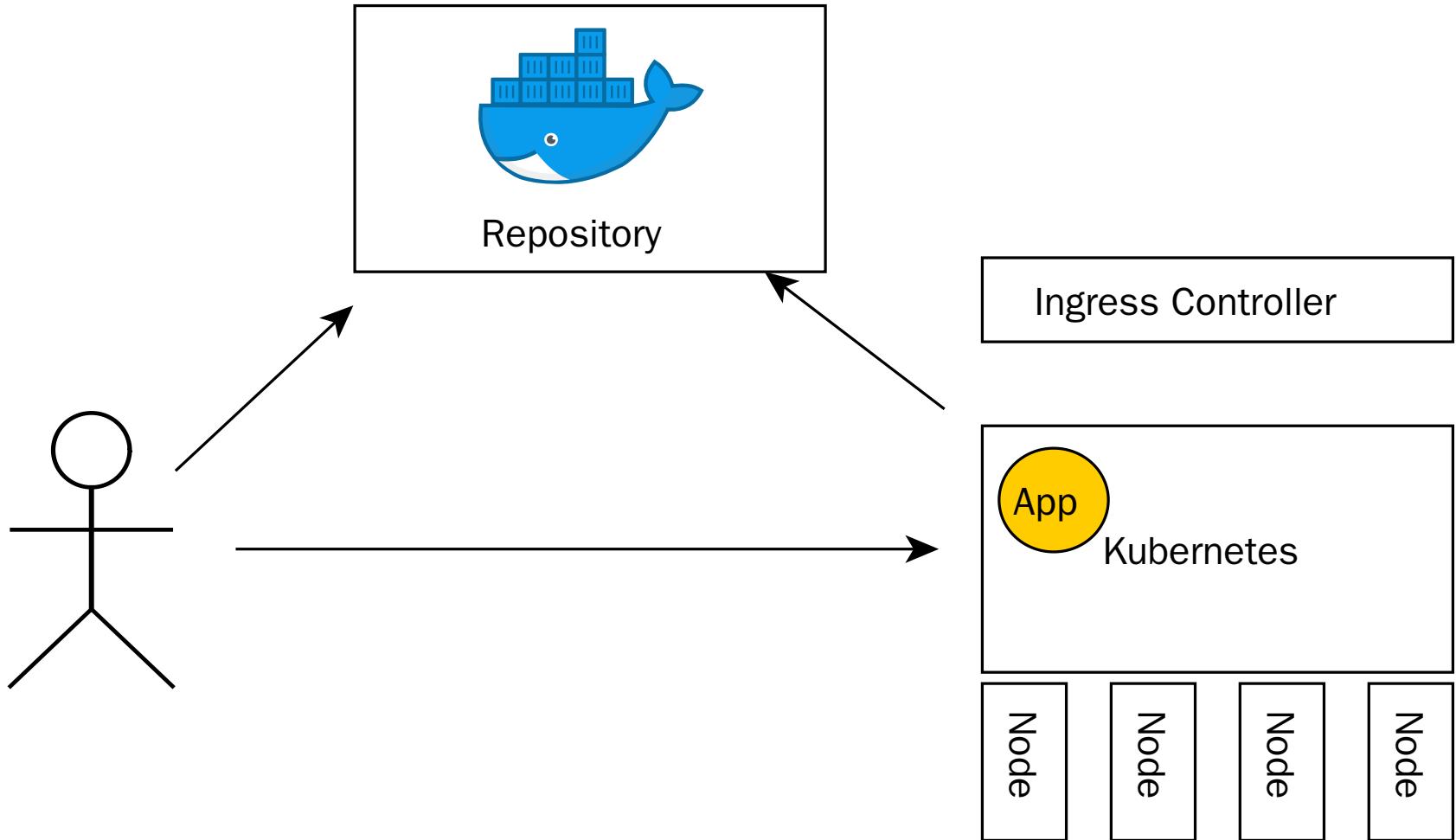
KUBERNETES

KUBERNETES

- Container orchestration
- Batteries for your 12factory apps*
- ...heading to integration platform
- Independent from IaaS provider

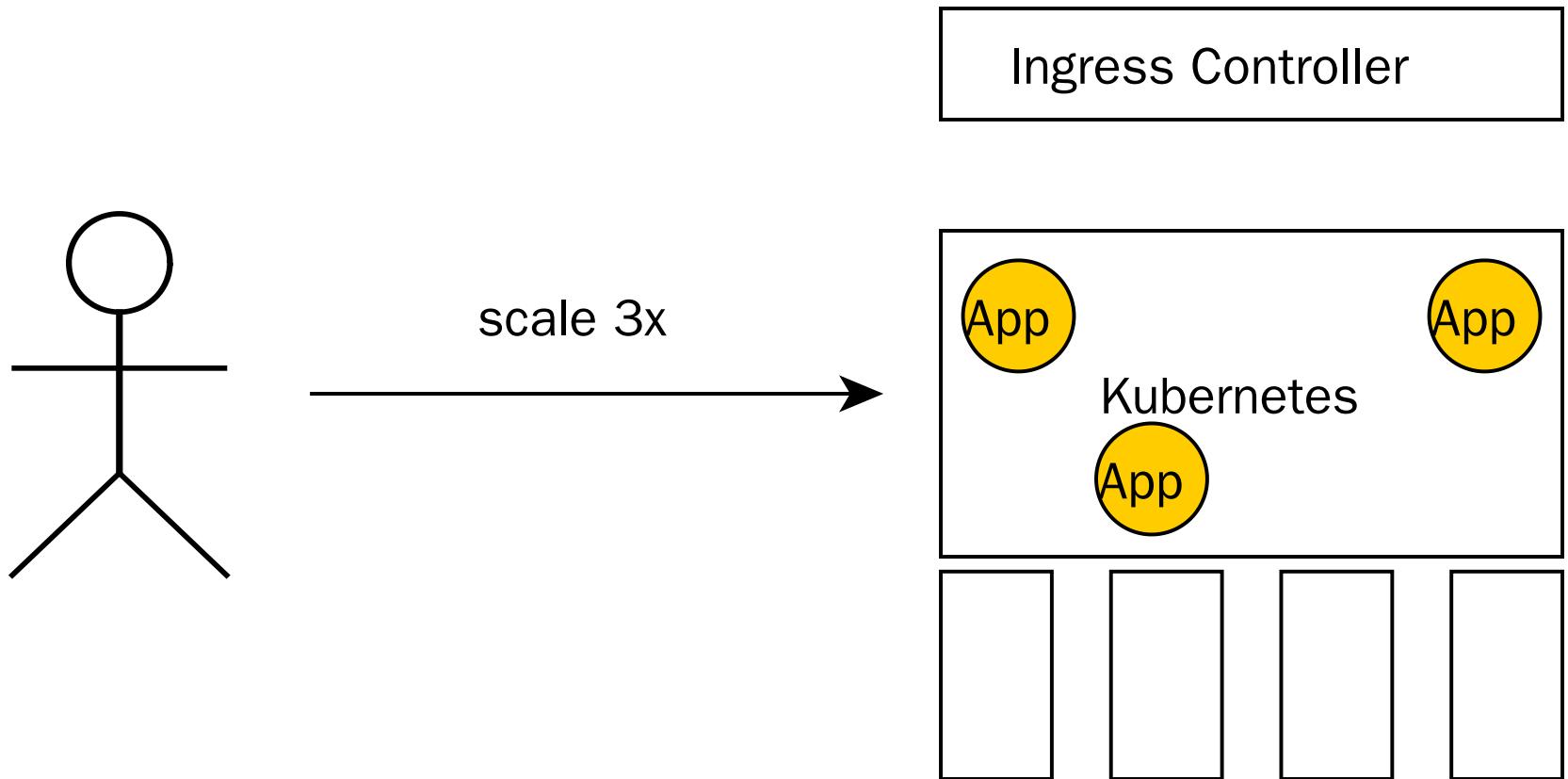
* - not only

KUBERNETES



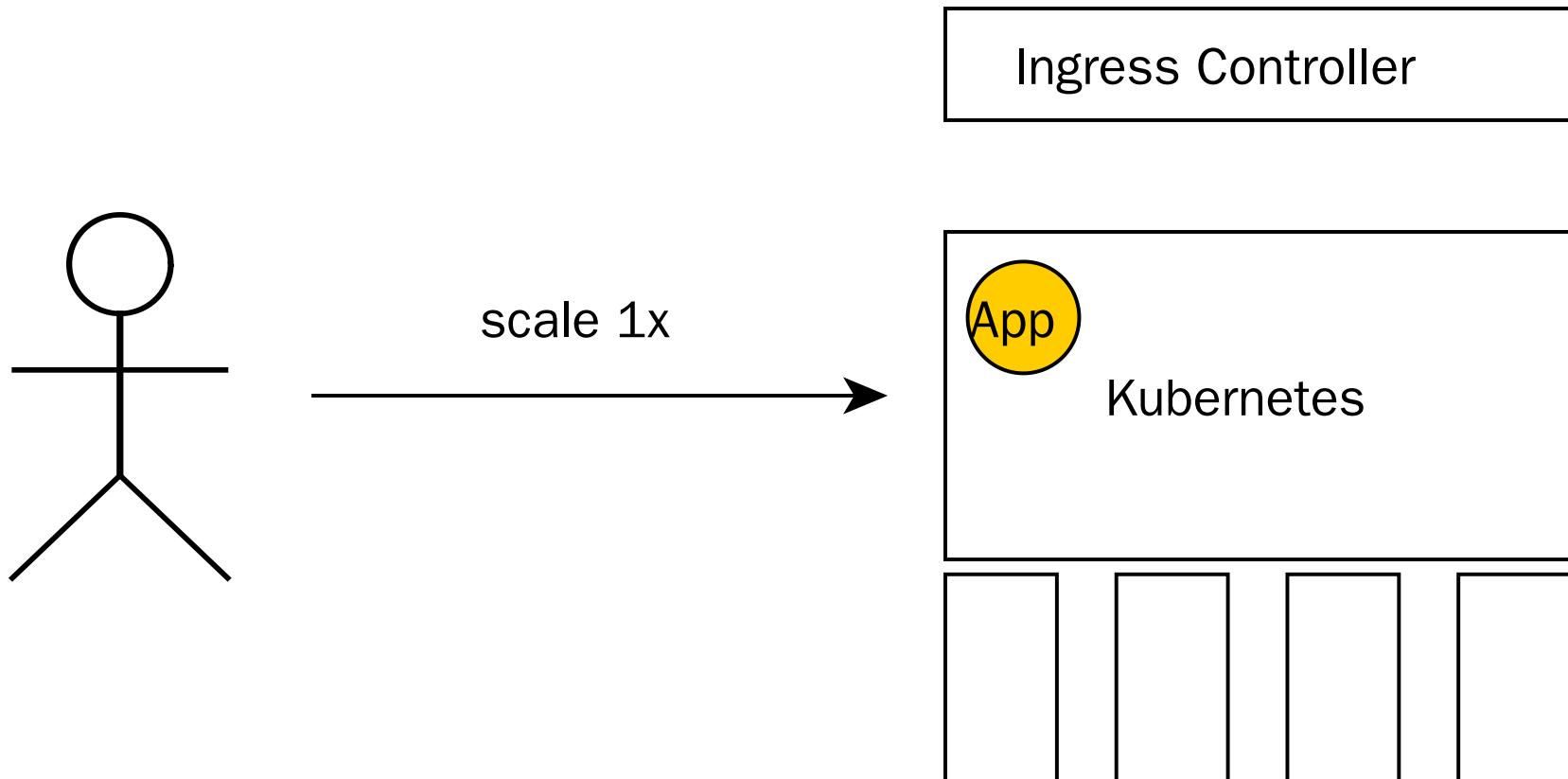
make docker_push; kubectl create -f app-srv-dpl.yaml

SCALE UP! SCALE DOWN!



```
kubectl --replicas=3 -f app-srv-dpl.yaml
```

SCALE UP! SCALE DOWN!

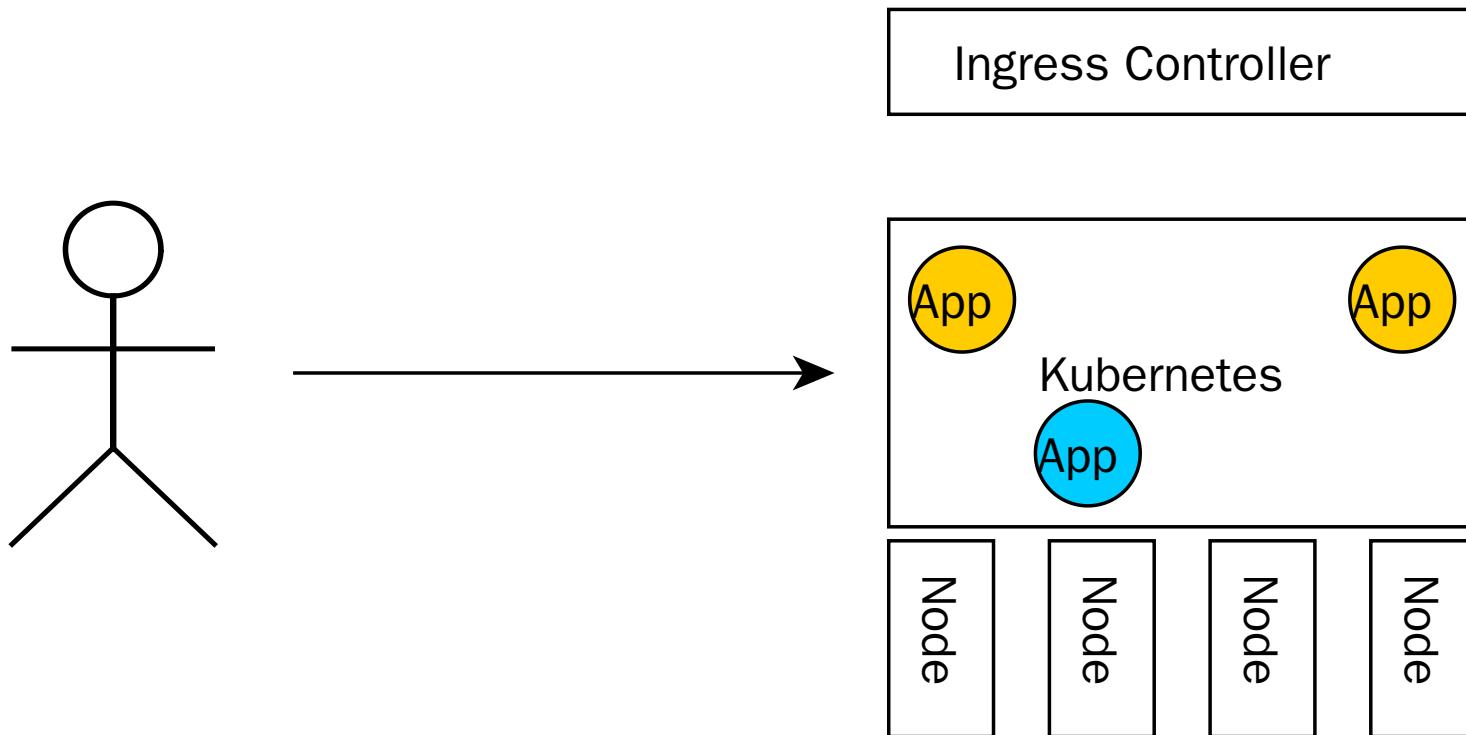


`kubectl --replicas=1 -f app-srv-dpl.yaml`

BASIC CONCEPTS

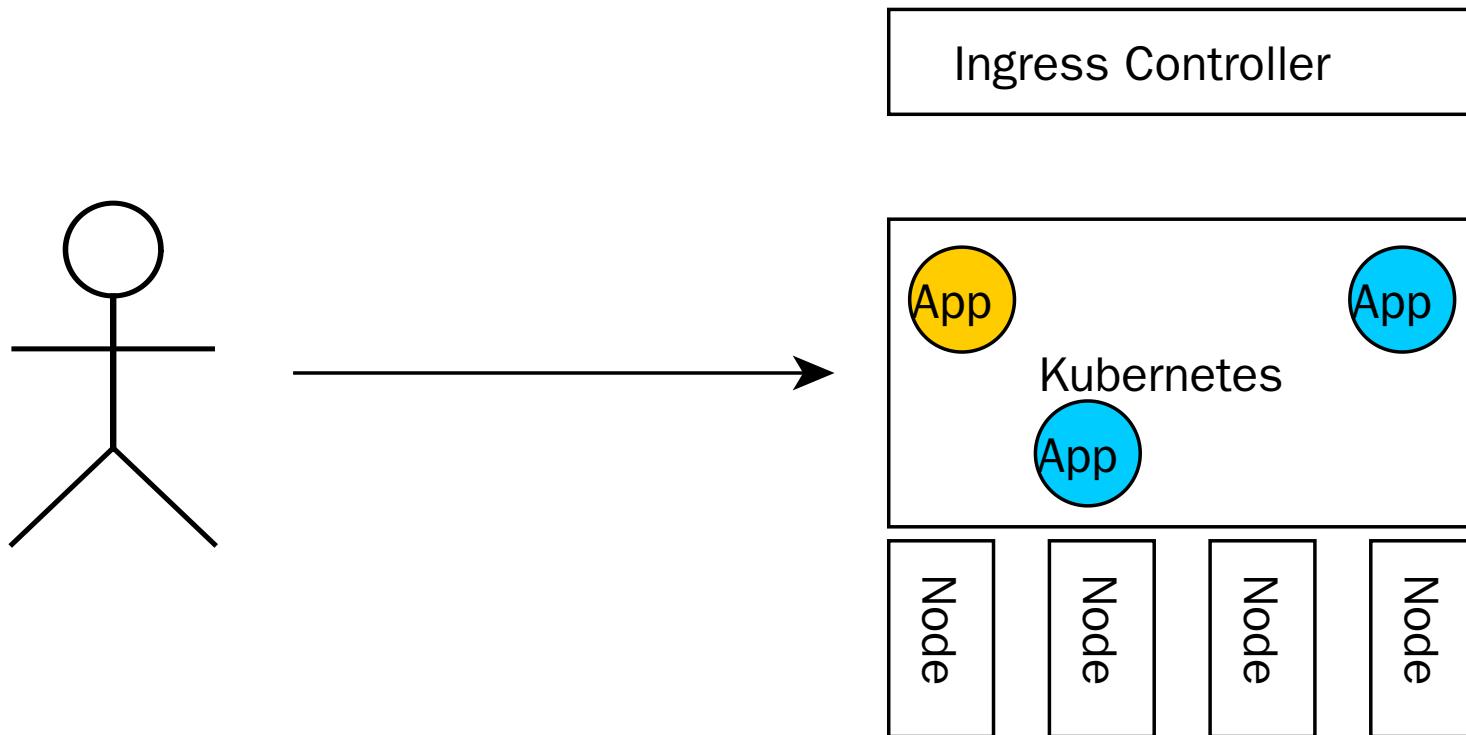
Name	Purpose	
Service	Interface	Entry point (Service Name)
Deployment	Factory	How many pods, which pods
Pod	Implementation	1+ docker running

ROLLING UPDATES!



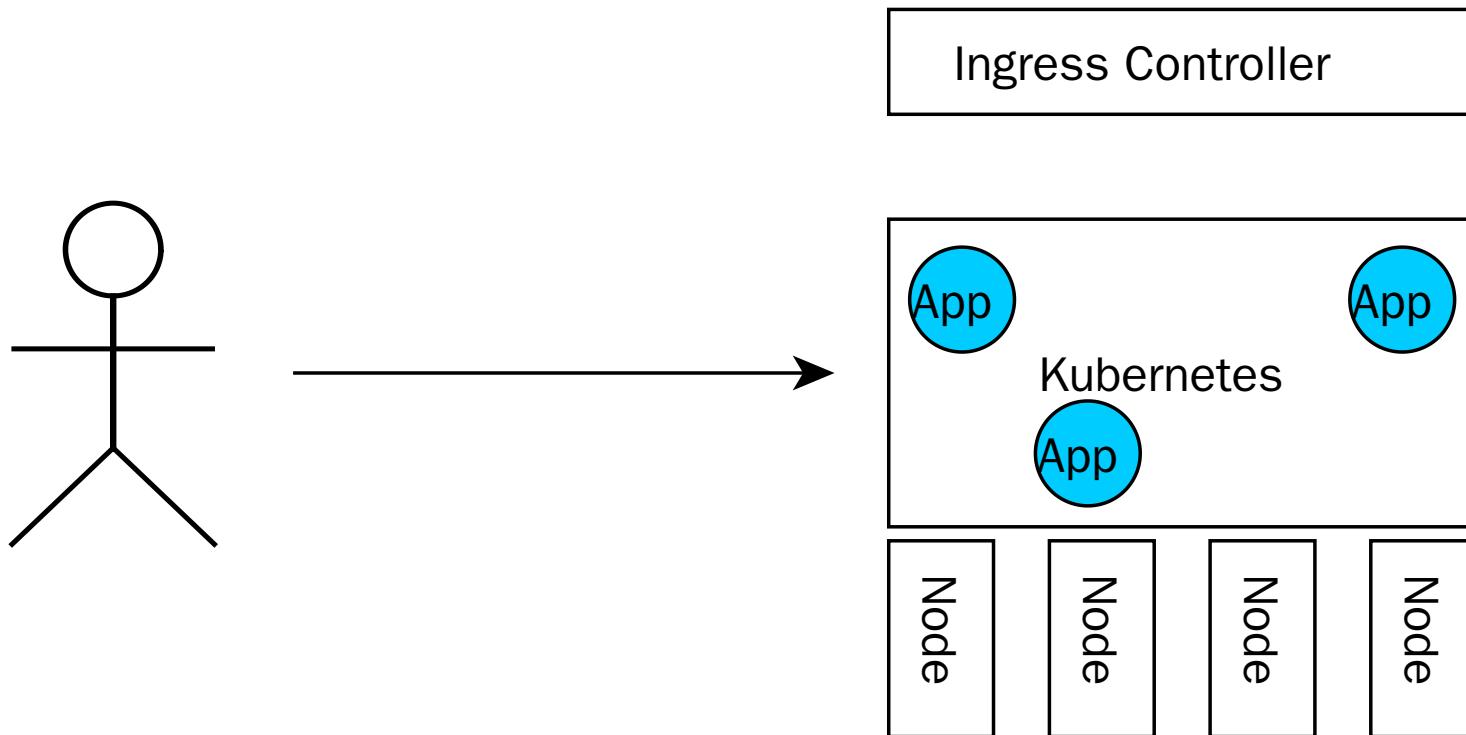
```
kubectl set image deployment/app app=app:v2.0.0
```

ROLLING UPDATES!



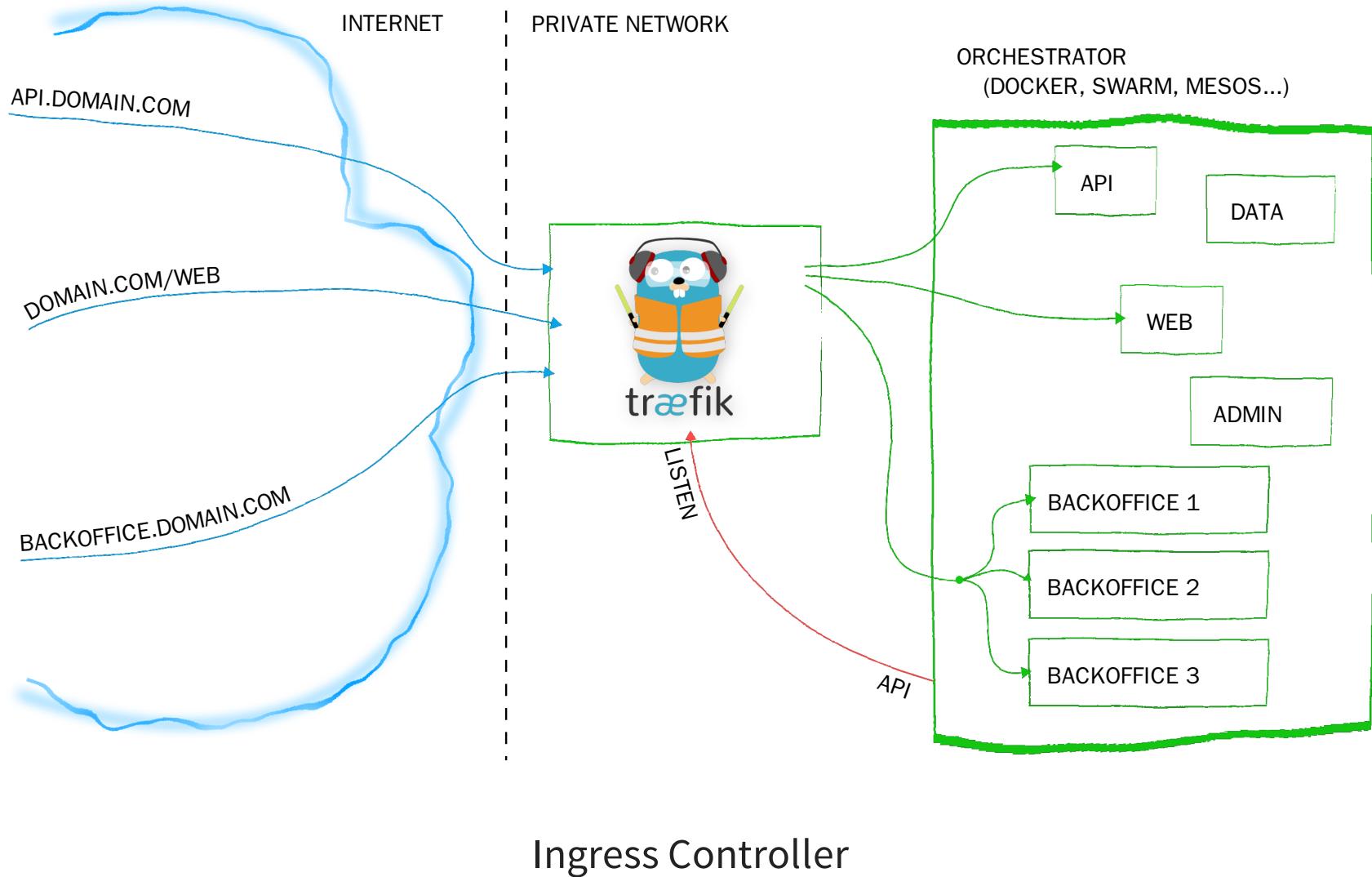
```
kubectl set image deployment/app app=app:v2.0.0
```

ROLLING UPDATES!



```
kubectl set image deployment/app app=app:v2.0.0
```

HOW GET USER REQUESTS?



INGRESS

Pattern

api.smacc.io/v1/users

Target App Service

users-v1

api.smacc.io/v2/users

users-v2

smacc.io

web

SERVICE DISCOVERY

- names in DNS:

```
curl http://users/list
```

- labels:

```
name=value
```

- annotations:

```
prometheus.io/scrape: "true"
```

SERVICE DISCOVERY

- loosely couple components
- auto-wiring with logging and monitoring

service.yaml

```
apiVersion: apps/v1
apiVersion: v1
kind: Service
metadata:
  name: intro-app-svc
spec:
  ports:
  - port: 80
    protocol: TCP
  selector:
    app: intro-app
  type: LoadBalancer
```

https://github.com/wojciech12/workshop_kubernetes_and_cloudnative

deployment.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: demo-api
  labels:
    app: demo-api
spec:
  replicas: 3
  strategy:
    type: Recreate
  selector:
    matchLabels:
      app: demo-api
  template:
    metadata:
```

https://github.com/wojciech12/workshop_kubernetes_and_cloudnative

ALL HAIL THE GIT!

- Yaml
- integration:
monitoring, alarming, ingress-controller
- ...
- Devs can forget about infra... almost
- DevOps Culture Dream!
 - + all tools -> CRD

KUBERNETES @ AZURE

OPTIONS

- AKS - managed
- ~~ACS~~
- Your own installation with installer
- RedHat Openshift

AZURE KUBERNETES SERVICE

- GKE for Google
- EKS or Fargate for Amazon

AKS

- Independent from IaaS
- Our OnPrem = Our OnCloud
- Reuse all my knowhow from OnPrem and GKE
- Plug and play, e.g., monitoring

AKS

- Managed kubernetes
- Integration with Azure Infrastruktur
- Addons: monitoring

See [docs](#)

AZURE K8S INTEGRATION

- Load Balancers
- Persistence Volumes including Resizable Volumes
- Graphic Cards Support
- Authentication and the Azure AD integration
- Monitoring as an addon

MANAGED

- You: k8s workers
- Azure: k8s masters

MANAGED - YOU

- Upgrade your k8s
- Can ssh to your workers
- Can modify and restart kube-system pods
- Node updates when we restart a node

MANAGED - AZURE

- update your kube-system pods
- k8s config
- updates on nodes mem-pres and on-restart

UPDATES FROM AZURE

Bumpy but no time to waste!

- Kubernetes configuration - 💀
- System memory-preserving updates - 💀

LIMITS

- No node-pool support
- Volumes in MC_YOUR_CLUSTER_NAME resource group ☠
- CD: Service Principle for AKS gets always an admin token

ANNOYING

- Limits
- Slow attaching and detaching volumes
- Watch out:
 - limits volumes attached to node (1 core = 1 disc)
- ~~Slow deletes~~
- ~~You are not able to delete a pod without~~ ~~--force~~

LOVE!

- Openness on github: [AKS issues](#)

AKS
@ SMACC.IO / HYPATOS.AI

OUR SETUP AZURE (1)

- az aks CLI for setting k8s
- aks imported to terraform
- Terraform for everything else

TF also sets our AWS

OUR SETUP AZURE (2)

- Create AKS clusters with TF

DATABASES 1: MANAGED

- Postgres
- CosmosDB as Mongodb

DATABASES 2: IN AKS

- Postgres
- Mongodb
- Kafka
- Etc/Vault

Note: you should know what you do.
Replication factor and Backup.

TECH

- Golang for all
- Python for Machine Learning
- JS & Emberjs WebUI
- OpenAPI
- ML Pipeline - evaluate kubeflow i patchyderm

KUBERNETES

- Pure, generated, kubernetes config
- 2x kubernetes operators

CLOUD NATIVE

- Observability: Prometheus stack and EFK
- Traefik for ingress
- etcd + Vault

CONTINUOUS DEPLOYMENT

- Github
- TravisCI
- hub.docker.com
- Kubernetes

In spirit similar to the [Kelsey Hightower approach](#)

CONTINUOUS DEPLOYMENT

- master -> dev
- tag -> staging
- PR accepted -> production

In spirit similar to the [Kelsey Hightower](#) approach

HOW TO START

PREREQUISITES

- az CLI
- kubectl

CREATE

```
az aks create --name my-aks-pet \
--resource-group GROUP \
--node-vm-size 'Standard_D4_v2' \
--node-count 4 \
--service-principal \
--client-secret
```

Always with SP! TF recommended

CREATE

- Demo: [portal](#)

Do not create AKS through portal

READY TO GO!

```
az aks get-credentials \
    -g GROUP \
    -n MyAKSPet
```

READY TO GO!

```
kubectl get pods
```

READY TO GO!

```
$ git clone \
git@github.com:wojciech12/workshop_kubernetes_and_cloudnativ
$ kubectl apply -f introduction/kube-deployment.yml
$ kubectl apply -f introduction/kube-service.yml
```

READY TO GO!

```
kubectl get pods
```

TIPS AND HINTS

COMMON PROBLEMS

- DNS is dead - it is so-k8s
- Worker node lost connectivity
- Memory-preserving update hits hard
- Slow volume detaching/attaching

PROBLEMS

Solving Problems Heuristics

- Check the age of system pods :D
- If sth does work -> resize your cluster
- Deleting system pods might help
- Restarting a node might help

UPDATE

1. Check what GKE supports
2. Check issues on github.com/Azure/AKS
3. Analyze:

```
az aks get-versions -o table -l  
westeurope
```

UPDATE

4. Analyze:

```
az aks get-upgrades -g GROUP --name  
my-aks-pet -o table
```

UPDATE

5. You MUST not
 1.8.0 > 1.9.10, better: 1.8.0 > 1.8.11 first
6. The last step:

```
az aks upgrade --name my-aks-pet \
    --resource-group GROUP \
    --output table \
    --kubernetes-version 1.10.8
```

AZURE UPDATES - NODES

NAME	VERSION	OS-IMAGE	KERN
aks-nodepool1-27173880-0	v1.10.3	Ubuntu 16.04.4 LTS	4.15
aks-nodepool1-27173880-1	v1.10.3	Ubuntu 16.04.4 LTS	4.15
aks-nodepool1-27173880-2	v1.10.3	Ubuntu 16.04.4 LTS	4.15
aks-nodepool1-27173880-3	v1.10.3	Ubuntu 16.04.4 LTS	4.15
aks-nodepool1-27173880-5	v1.10.3	Ubuntu 16.04.4 LTS	4.15

```
kubectl get nodes -o wide
```

UPDATE

Name	MasterVersion	NodePoolVersion	Upgrades
default	1.10.3	1.10.3	1.10.5,

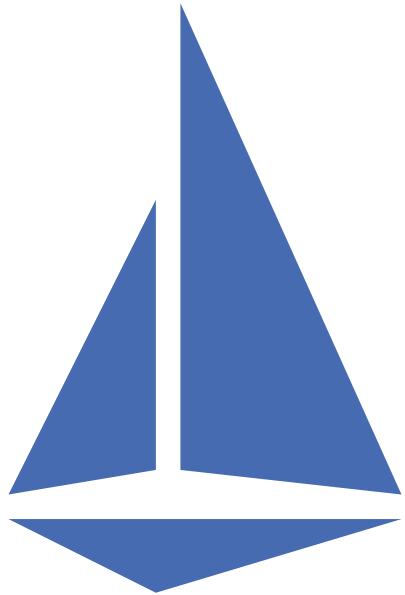
```
az aks get-upgrades -g GROUP \
--name my-aks-pet -o table
```

UPGRADE

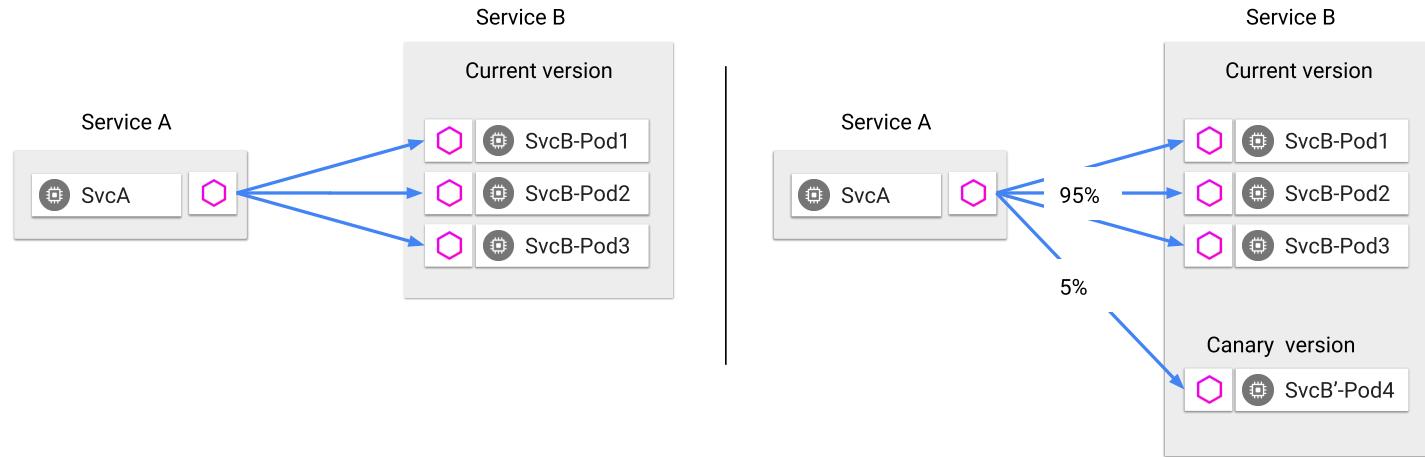
```
az aks upgrade --name my-aks-pet \  
  --resource-group GROUP \  
  --output table \  
  --kubernetes-version 1.10.3
```

Do not rush!

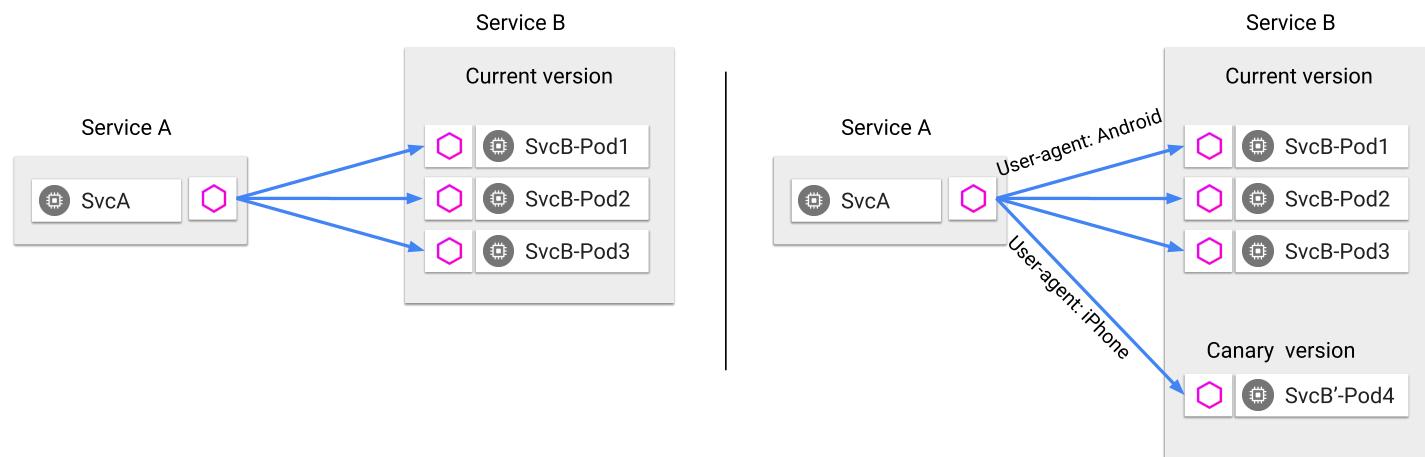
ISTIO. FUTURE?



ISTIO



Traffic splitting decoupled from infrastructure scaling - proportion of traffic routed to a version is independent of number of instances supporting the version



Content-based traffic steering - The content of a request can be used to determine the destination of a request

ISTIO

ISTIO

- Service Mesh
- Networking control given to operator
- Encryption everywhere
- Observability for everything except your code

ISTIO

- Next time, demo

SUMMARY

- K8S not a silver bullet, but damn close
- No time to waste, check K8S
- AKS the easiest way to start with AKS in Azure
- Less bumpy than before - see [github issues](#)

SUMMARY

- Wish for ones who trust only in MS: AKS onPrem
- If not: Vanilla on Ubuntu, CentOS/RH Openshift, SUSE Cloud App Platfrom

`github.com/wojciech12`

`wbarczynski.pl/talks`

`wbarczynski.pro@gmail.com`

DZIĘKUJĘ. PYTANIA?

ps. We are hiring.

```
123 def distance_matrix(regions):  
124     """ Computes a distance matrix against a region list """  
125     tuples = [r.as_tuple() for r in regions]  
126     return cdist(tuples, tuples, region_distance)  
127  
128  
129 def clusterize(words, **kwargs):  
130     # TODO: write a cool docstring here  
131     db = DBSCAN(metric="precomputed", **kwargs)  
132     X = distance_matrix([Region.from_word(w) for w in words])  
133     labels = [int(l) for l in db.fit_predict(X)]
```



BACKUP SLIDES

```
123 def distance_matrix(regions):  
124     """ Computes a distance matrix against a region list """  
125     tuples = [r.as_tuple() for r in regions]  
126     return cdist(tuples, tuples, region_distance)  
127  
128  
129 def clusterize(words, **kwargs):  
130     # TODO: write a cool docstring here  
131     db = DBSCAN(metric="precomputed", **kwargs)  
132     X = distance_matrix([Region.from_word(w) for w in words])  
133     labels = [int(l) for l in db.fit_predict(X)]
```



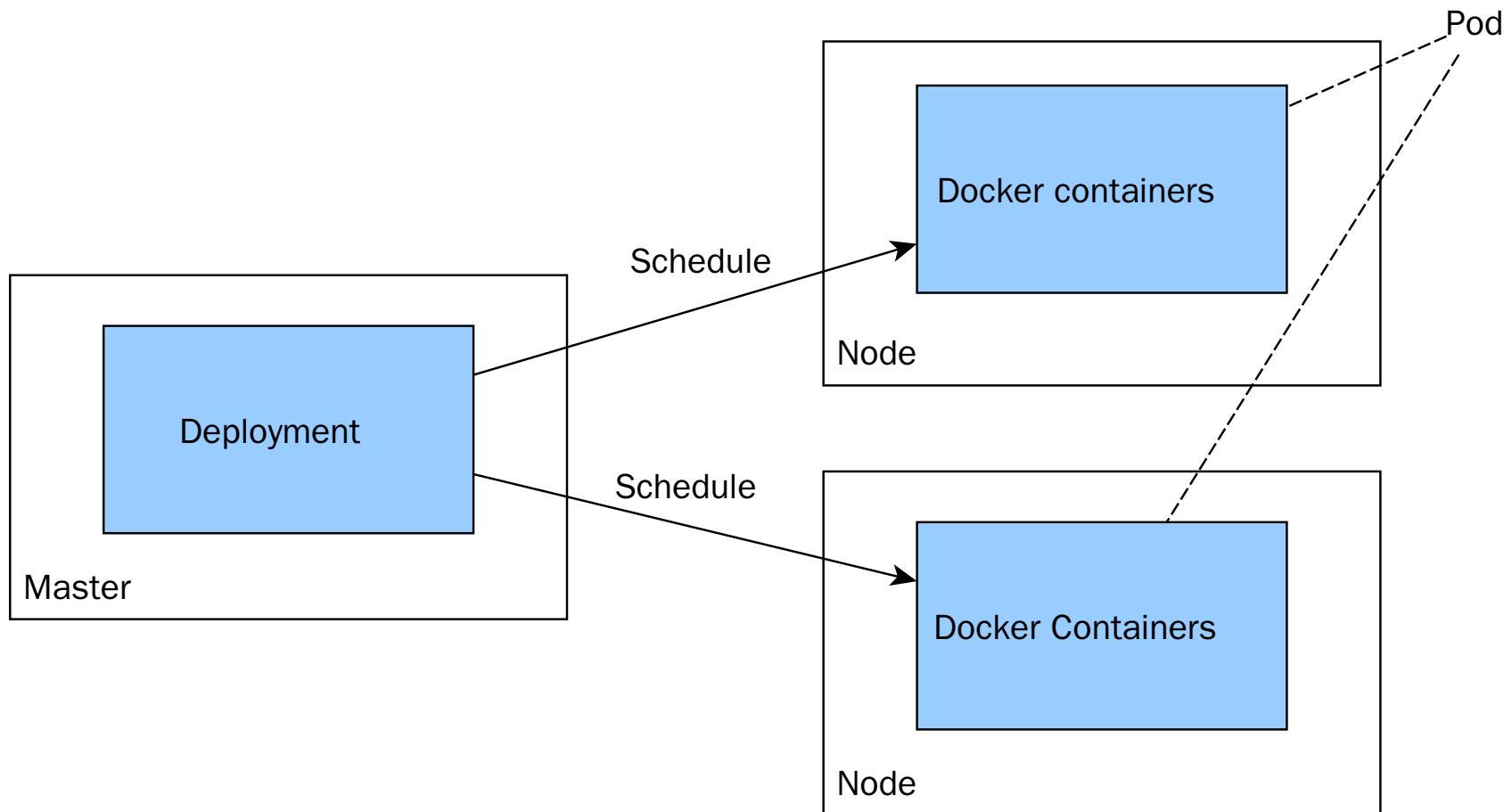
HIRING

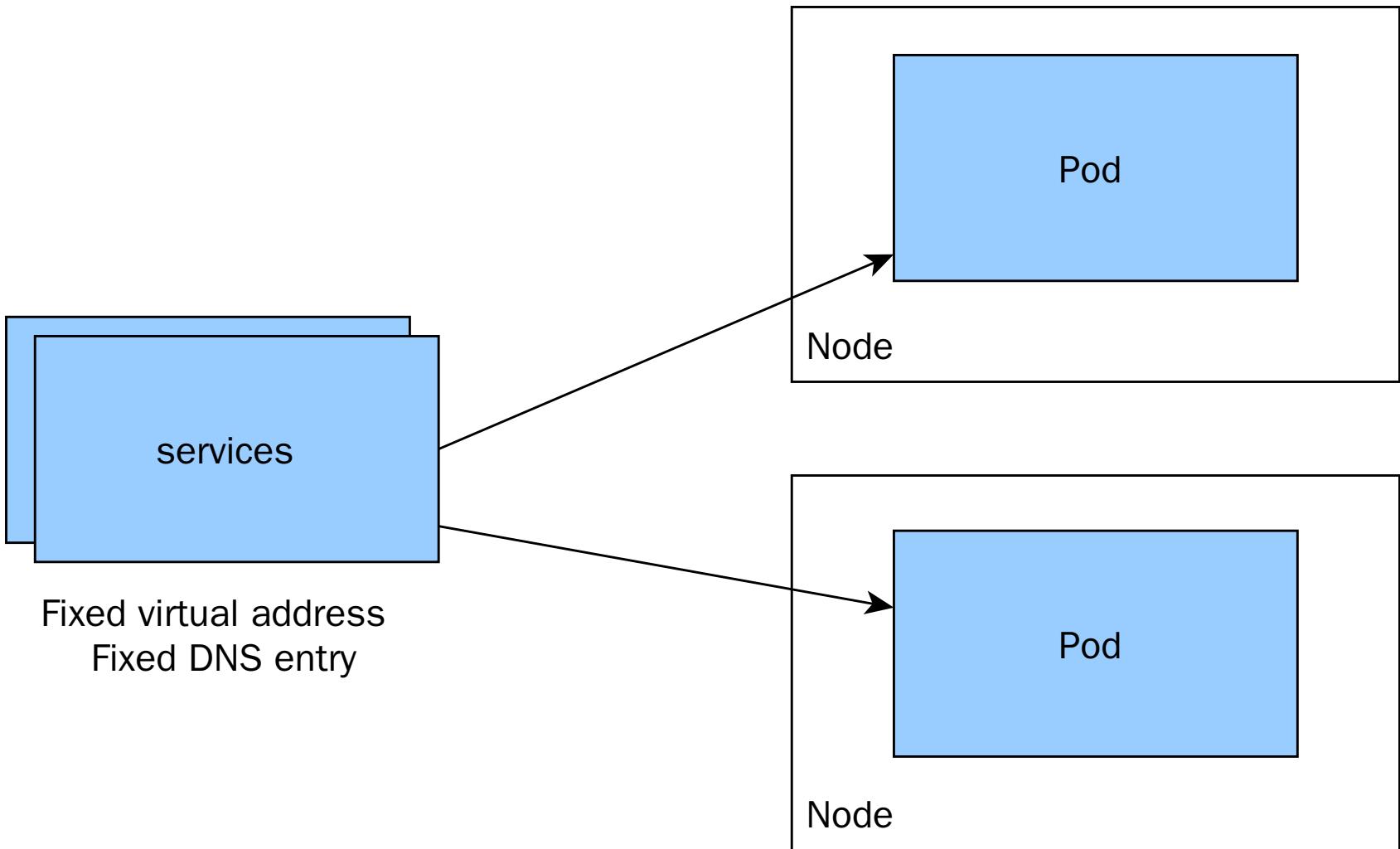
- Senior Polyglot Software Engineers
- Frontend Engineers
- 1 Data-Driven Product Manager

Apply: hello-warsaw@smacc.io,
Questions? wb@hypatos.ai, [FB](#) or [LI](#)

We will teach you Go if needed. No k8s or ML, we will take care of that.

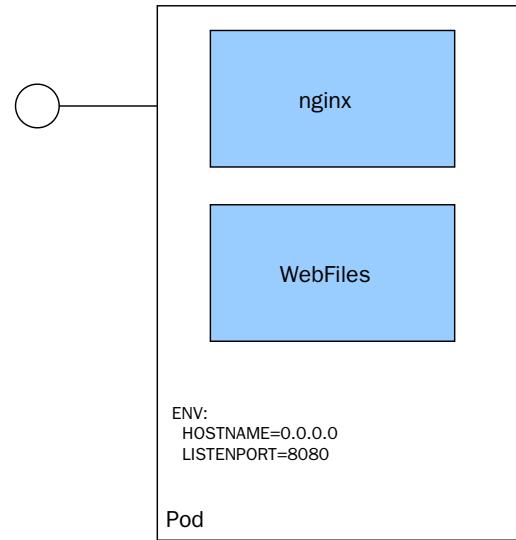
KUBERNETES CONCEPTS



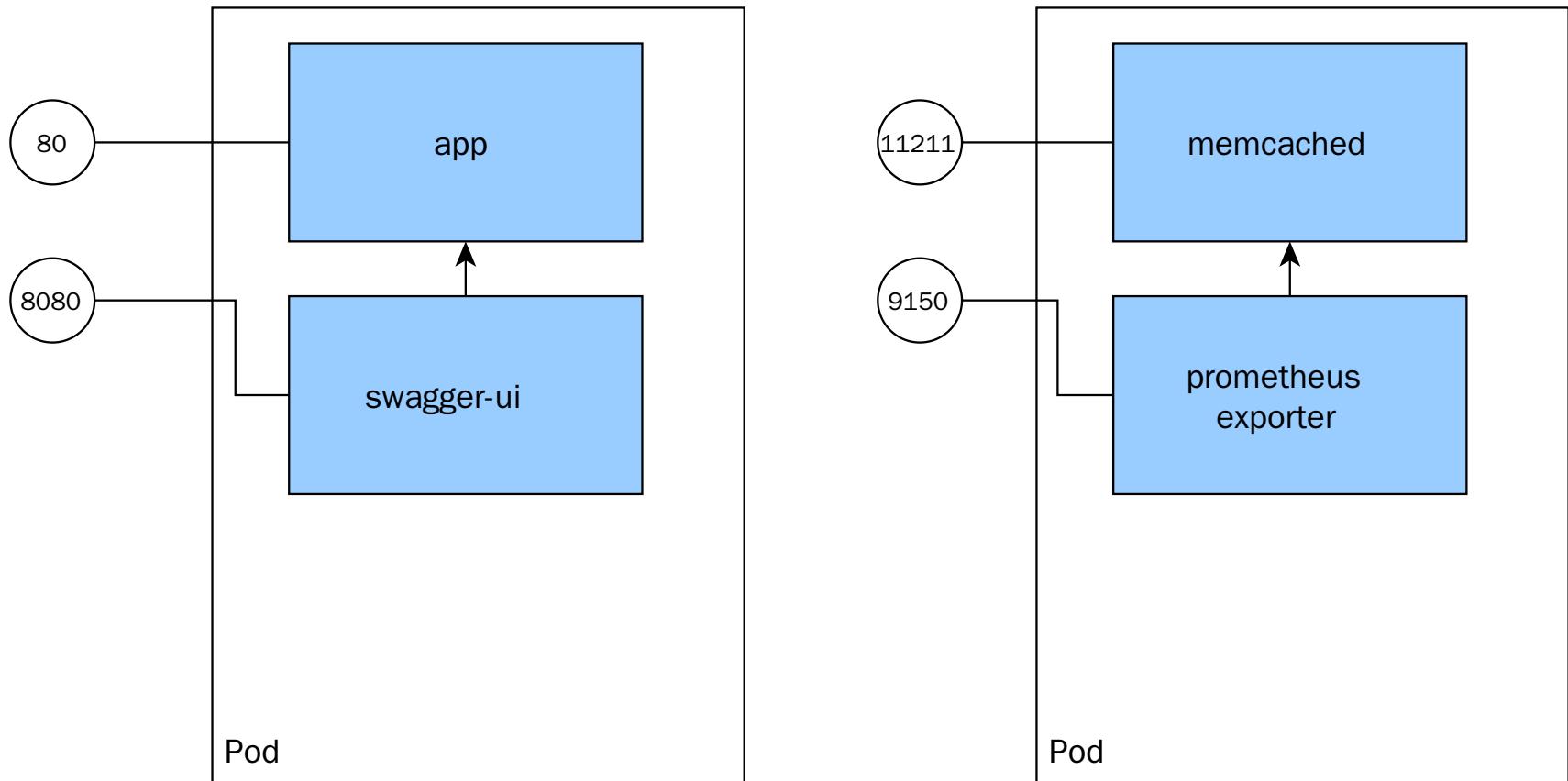


PODS

- See each other on localhost
- Live and die together
- Can expose multiple ports



SIDE-CARS



ROLLING RELEASE WITH DEPLOYMENTS

