



◎ ° +

◊ ◊ ◊

meetup

# VANTAGENS DE UTILIZAR UM SERVICE MESH COM O KUBERNETES PARA GERENCIAR SEUS MICROSERVIÇOS

André Pontes Sampaio

Claudio Eduardo de Oliveira

José Castillo Lema

Junho/2021

...

# APRESENTAÇÃO

## **KUBERNETES & SERVICE MESH**

*EFFECTIVE MICROSERVICE MANAGEMENT*



**ANDRÉ SAMPAIO**

PROFESSOR  
DE MBA  
E SHIFT - FIAP



**CLÁUDIO OLIVEIRA**

SR SOFTWARE  
ENGINEER  
LUIZA LABS



**JOSÉ CASTILLO**

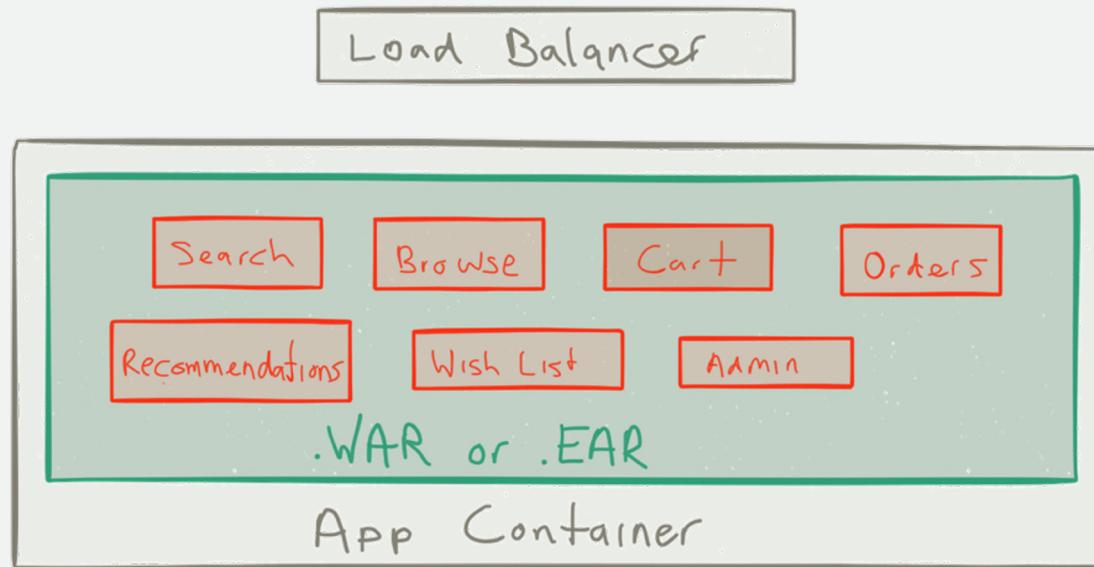
TELCO CLOUD  
CONSULTANT  
RED HAT

# AGENDA

1. Introdução
  - a. Microserviços
  - b. Kubernetes
2. Service mesh
3. GitOps

# PRINCIPAIS DORES

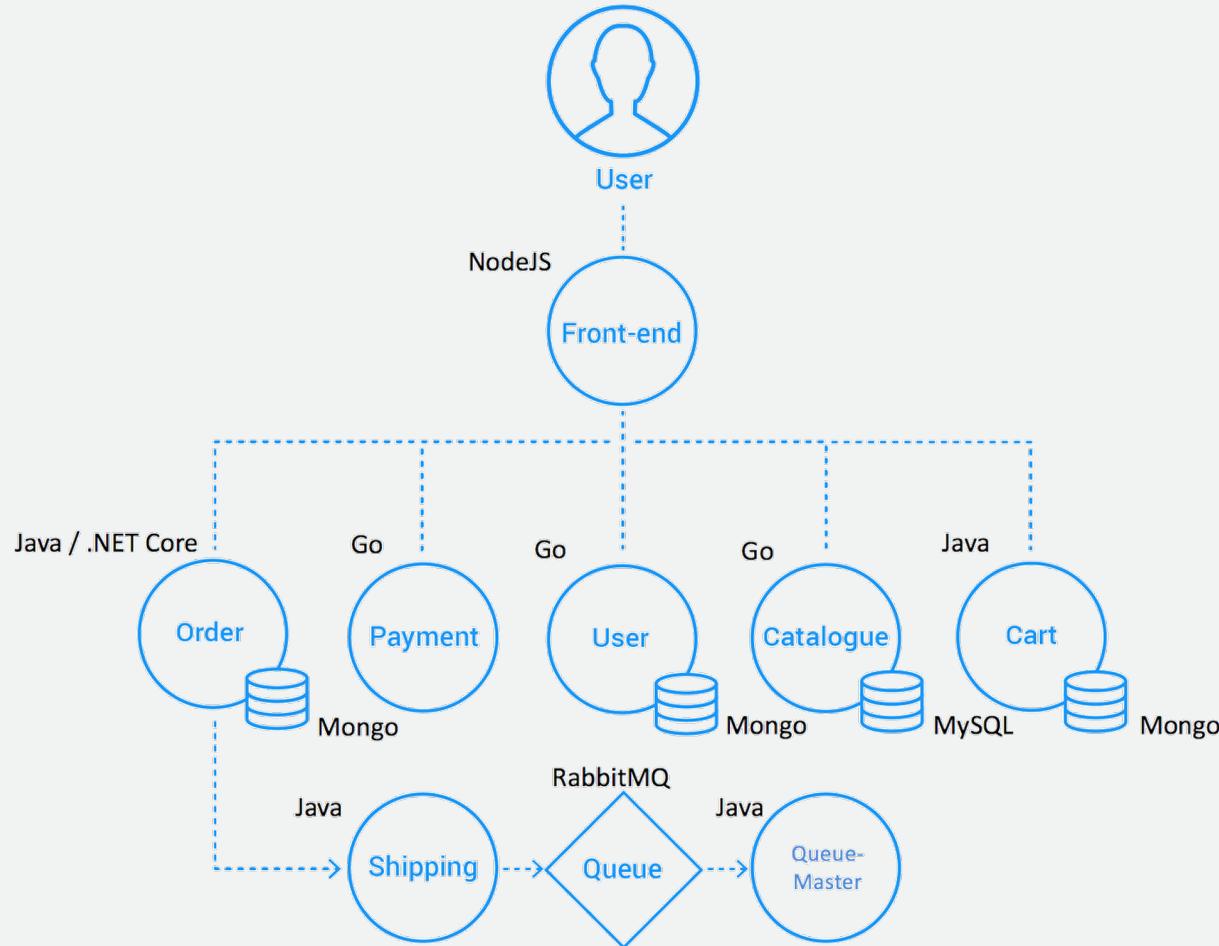
## Monolítico versus Microsserviços



Arquitetura Java EE

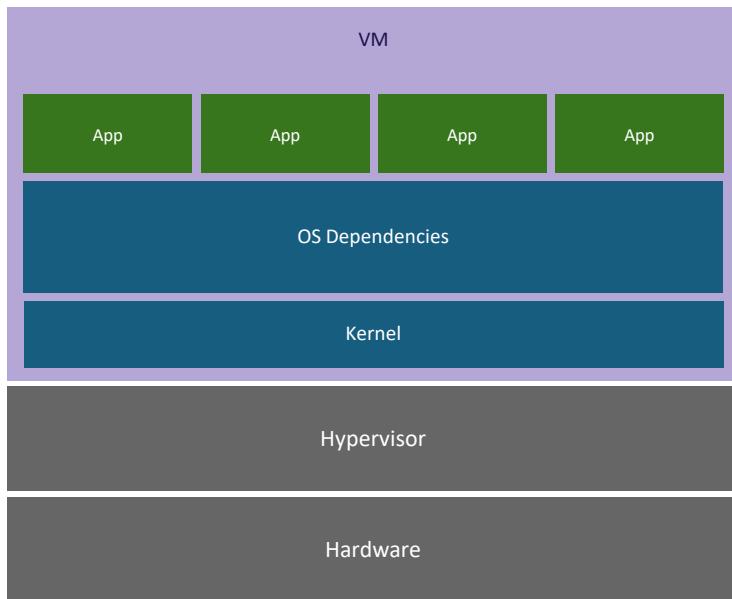
# MICROSERVIÇOS

## Monolítico versus Microsserviços



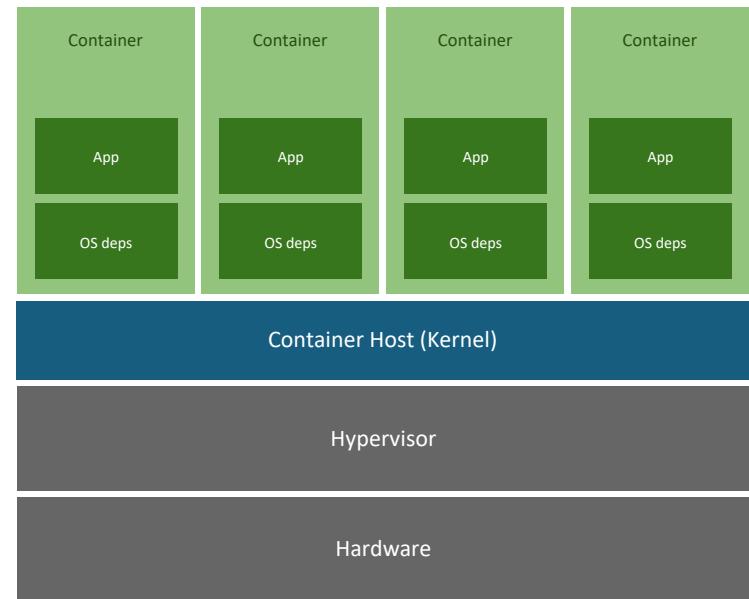
# VMs vs containers

VIRTUAL MACHINES



VM isolates the hardware

CONTAINERS



Container isolates the process

# UTILIZAÇÃO DE CONTÊINERES

Registro de uma imagem no Hub

Desenvolvimento em máquina local

Deployment mais veloz que vms

Implantação e distribuição da imagem

Mais densidade por servidor

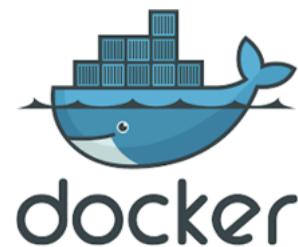
Execução padronizada dos aplicativos

## Características:

Facilita as práticas DevOps

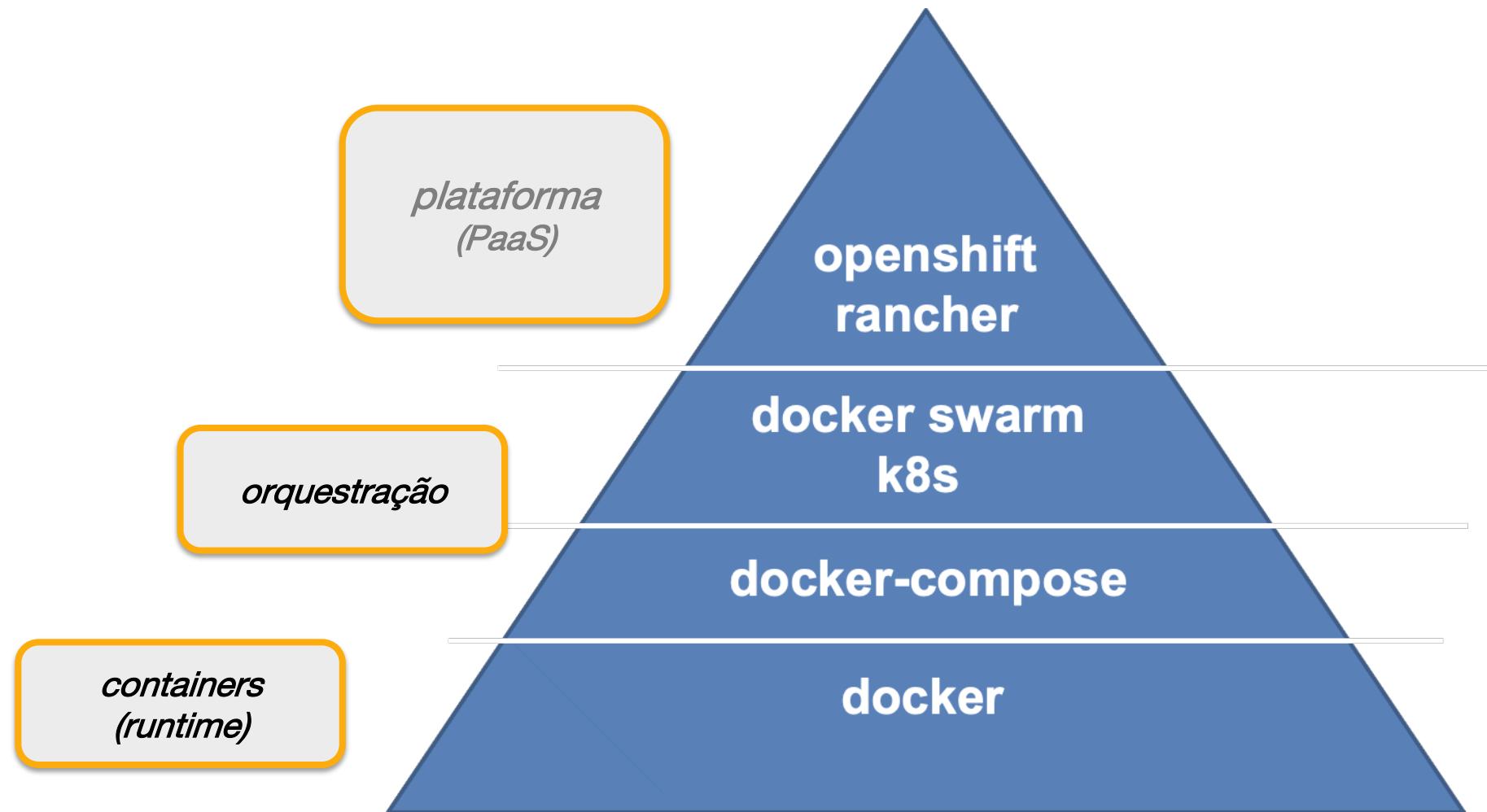
Aplicações não possuem dependências do sistema

# UTILIZAÇÃO DE CONTÊINERES

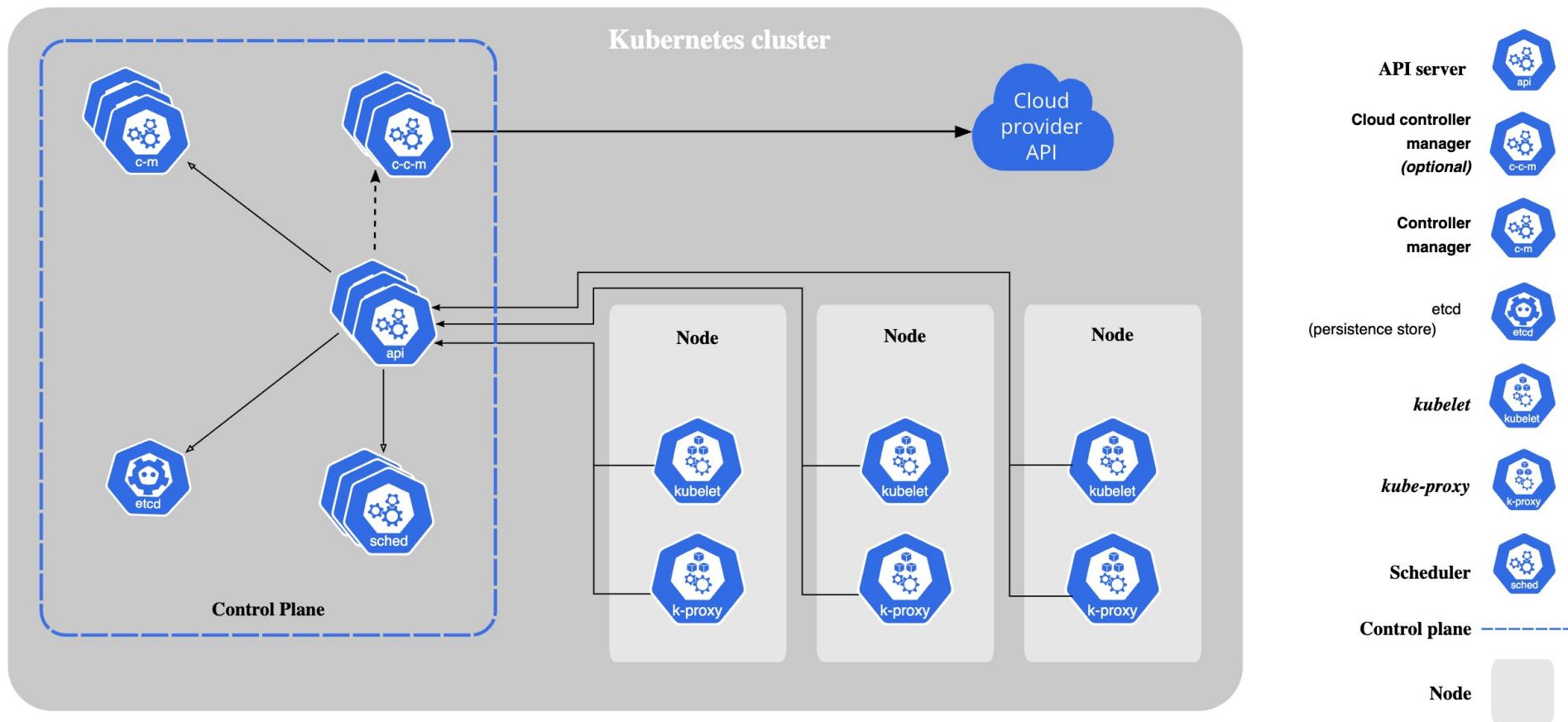


# UTILIZAÇÃO DE CONTÊINERES

Develop once deploy everywhere



# ORQUESTRAÇÃO DE CONTÊINERES



# UTILIZAÇÃO DE CONTÊINERES



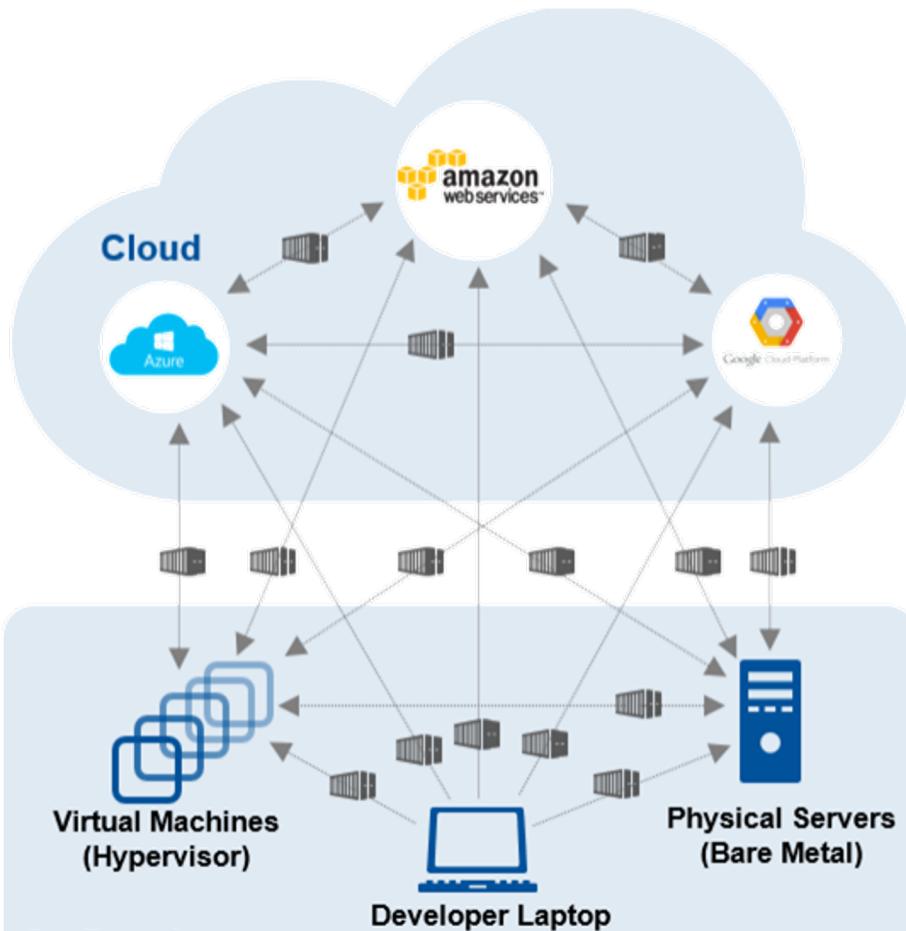
**kubernetes**



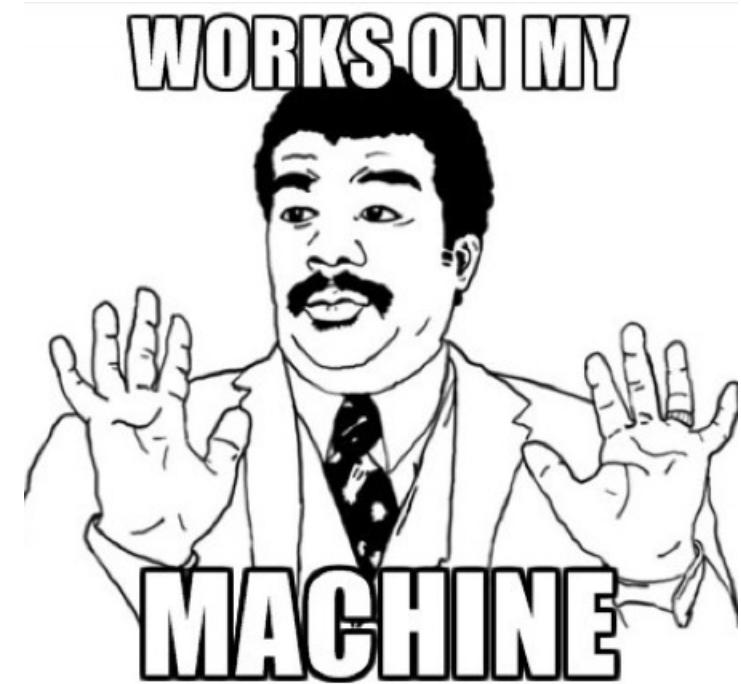
Apache  
**MESOS**<sup>TM</sup>

# UTILIZAÇÃO DE CONTÊINERES

Mobilidade de ambientes com Contêineres

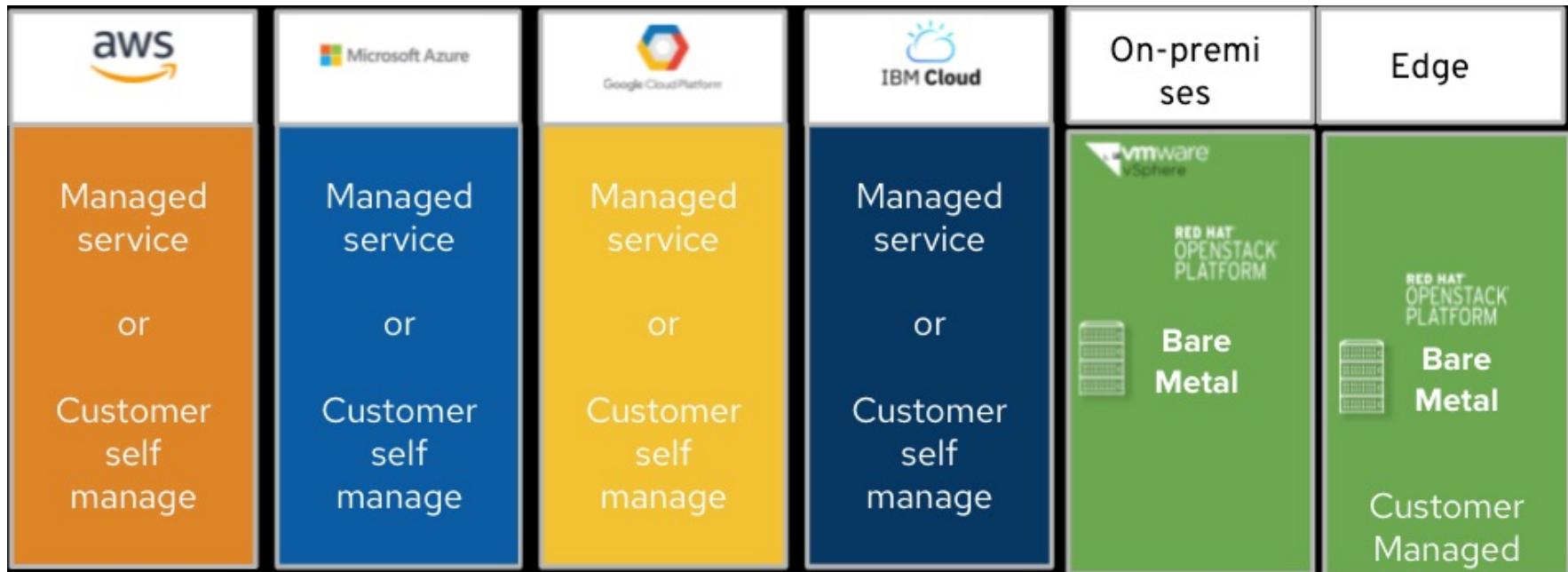


Write once,  
deploy anywhere



# UTILIZAÇÃO DE CONTÊINERES

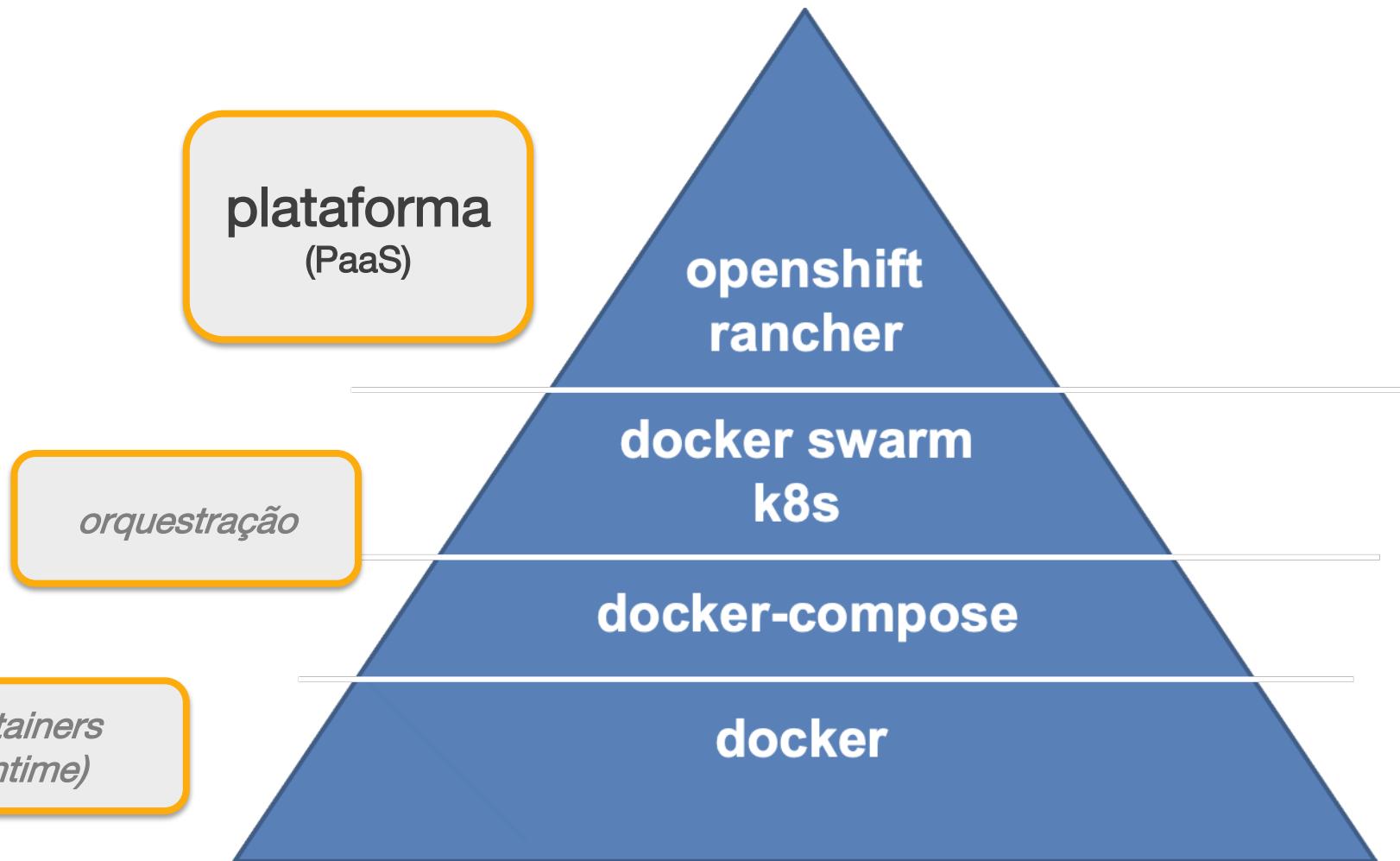
Develop once deploy everywhere



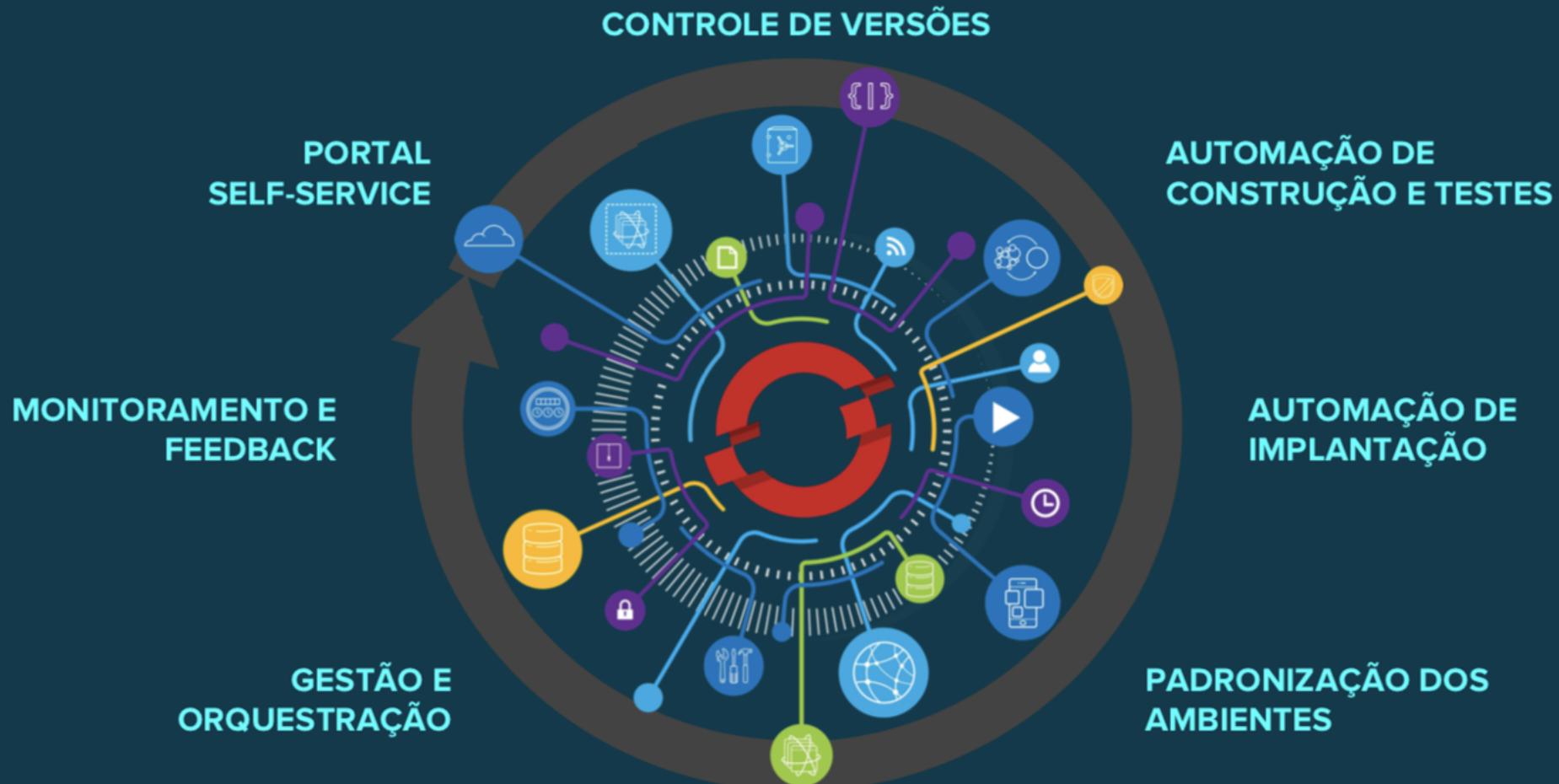
Identical API; consistent developer and operator experience.

# UTILIZAÇÃO DE CONTÊINERES

Develop once deploy everywhere



# UTILIZAÇÃO DE CONTÊINERES



# UTILIZAÇÃO DE CONTÊINERES



## OPENSIFT

# DEMO 1 : KUBERNETES (COMPONENTES)



K3S



docker



Execução:



kubernetes

MicroK8s

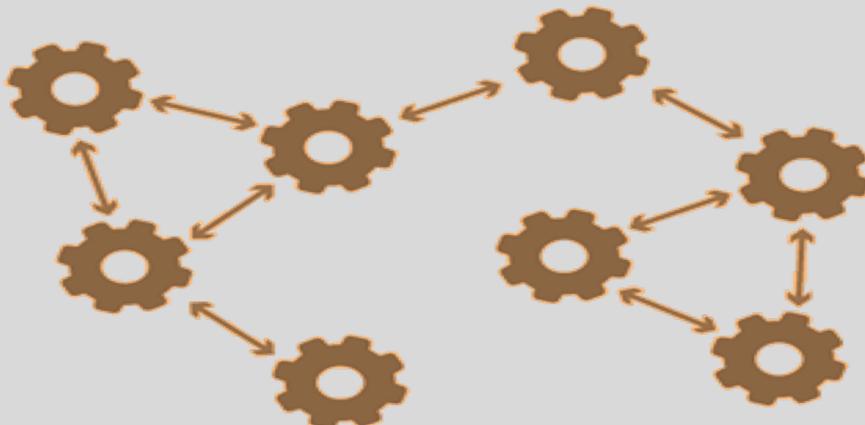


minikube

- > Gerenciamento
- > Facilidade de uso
- > Exemplo simples

# SERVICE MESH

Arquitetura Service Mesh (multi-conectada)  
e Side Car (controle das requisições)



Balanceamento e failover

Monitoramento e Segurança

Resiliência nas comunicações

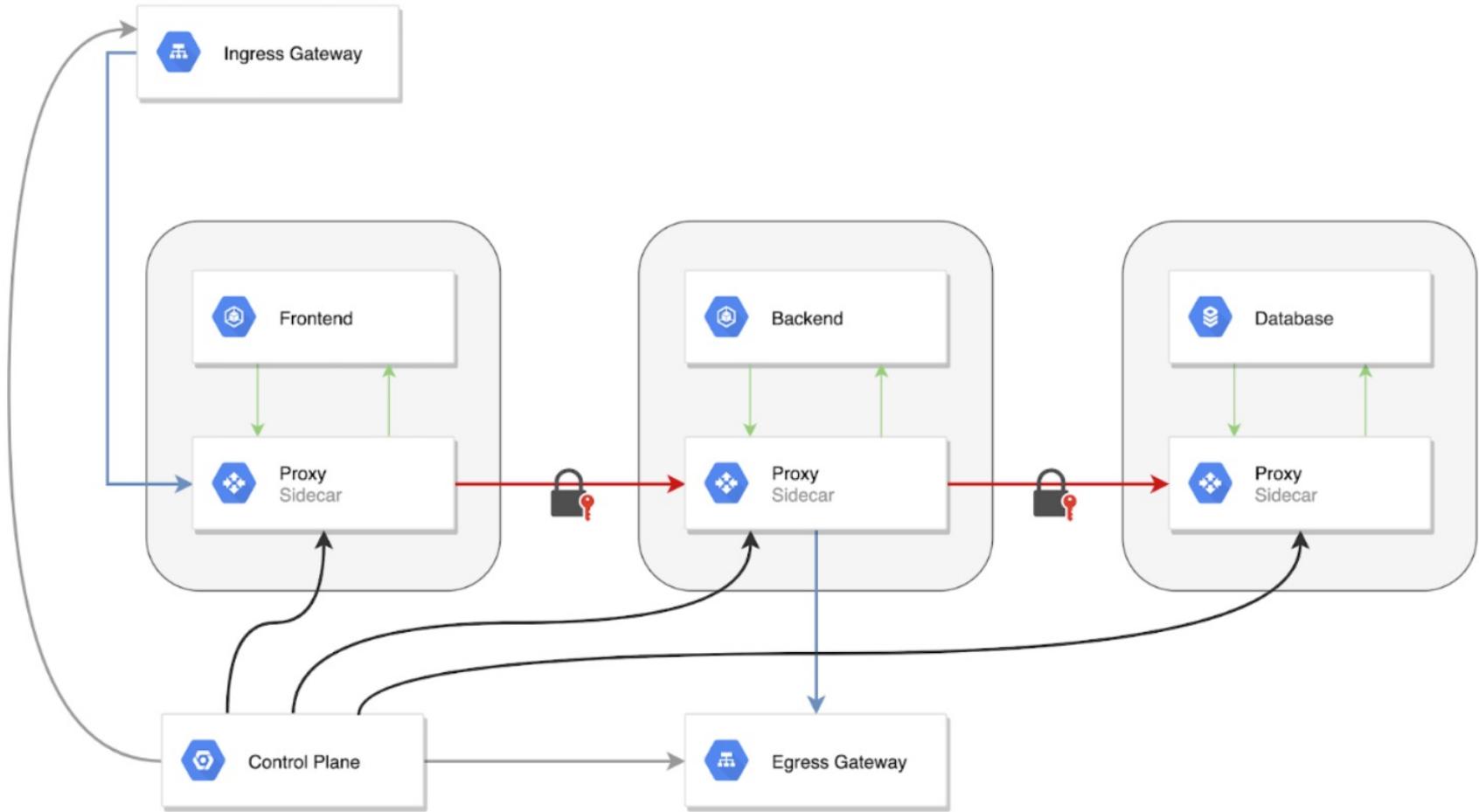
Descoberta de Serviço

## SERVICE MESH

Aplicações se comunicam de forma independente entre si

# SERVICE MESH

Arquitetura Service Mesh (multi-conectada)  
e Side Car (controle das requisições)



# DEMO 2 : SERVICE MESH



Download:

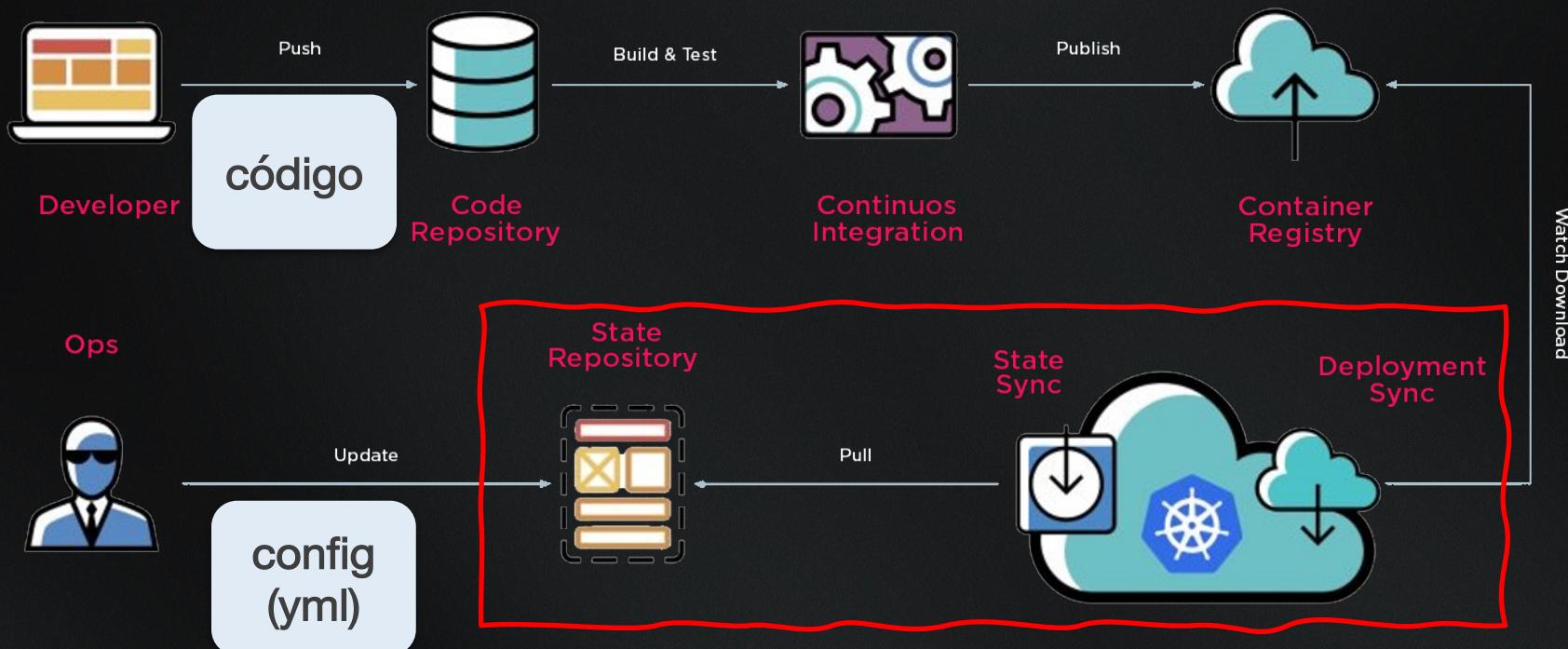
```
> git clone https://github.com/tonanuvem/k8s-exemplos.git
```

Execução:

```
> sh k8s-exemplos/meetup_run.sh
```

## GitOps : Pipeline CD unificado usando GIT

### CI / CD with GitOps



# PROGRESSIVE APP DELIVERY

- lançar features novas gradualmente
- não deve ter downtime
- sua aplicação deve ser “*stateless*”
- novas releases devem ser retro-compatível Blue-Green, Canary e outros

# ARGO CD

É uma ferramenta de *Continuous Delivery* para Kubernetes que emprega a utilização de GitOps



# ARGO ROLLOUT

É uma ferramenta que provê mecanismos de suporte a **Progressive App Delivery**, fornecendo Canary Releases, Rollouts e Experimentações em ambientes Kubernetes



Mas como GitOps, Continuous Delivery e  
Progressive se conectam com **Service Mesh** ???

# Observability

Serviços HTTP estão “automaticamente” incluídos na métricas do Istio

{REST API}

HTTP 400

HTTP 500

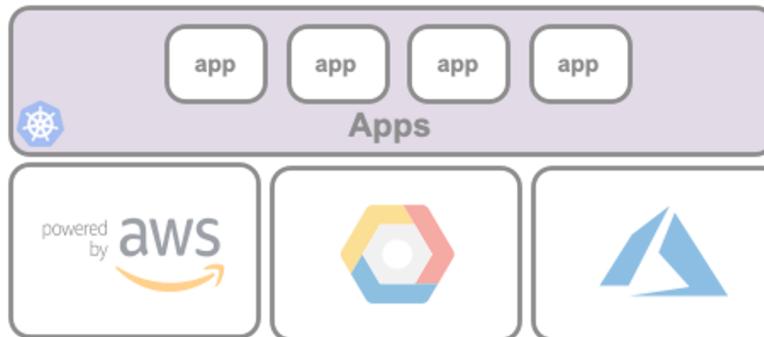
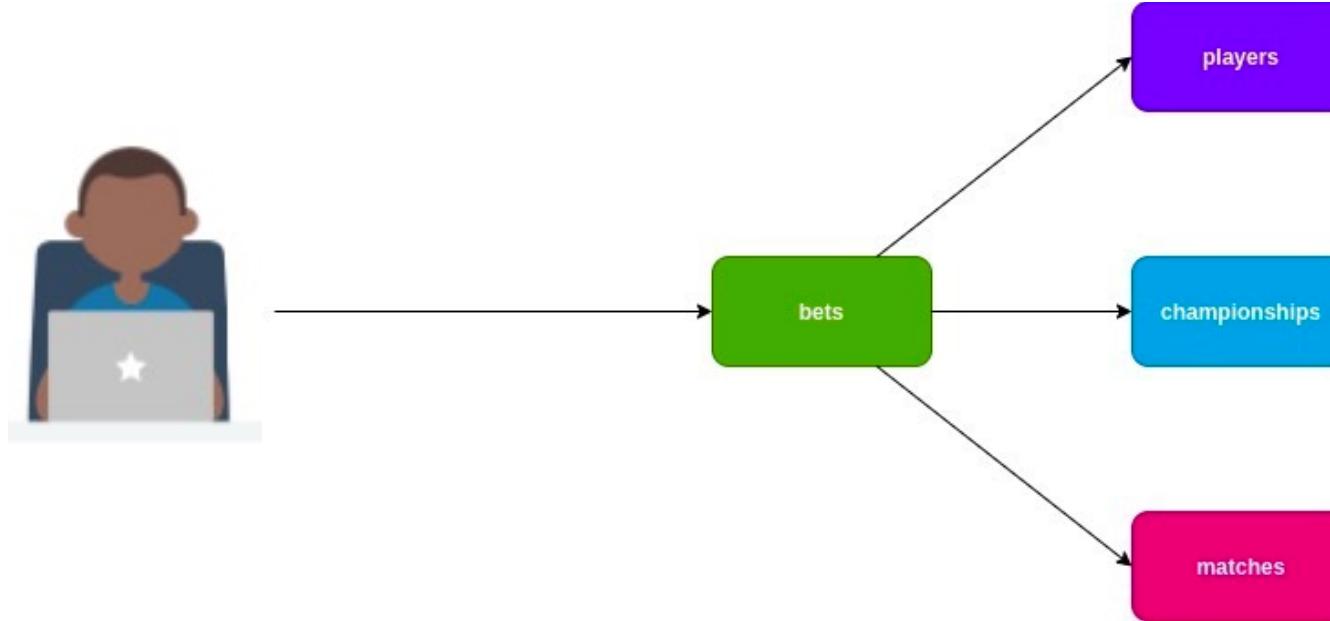
HTTP 200

# CRITÉRIO

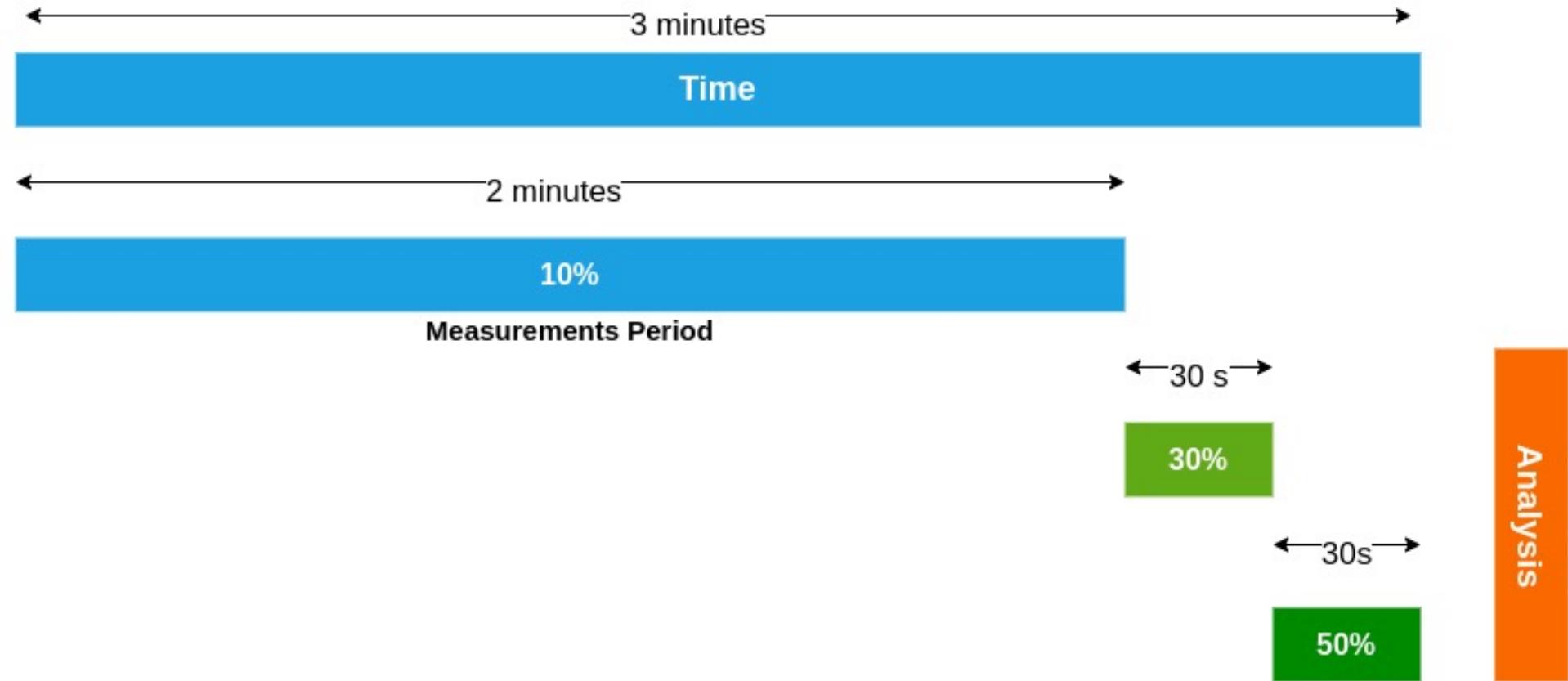
Se as chamadas HTTP com Status Code ~2,\* forem maiores que 95% ... Release executada com sucesso

Senão  
Rollback imediato!!!!

# TIMELINE DA RELEASE



# TIMELINE DA RELEASE



# | DEMO 3 : ARGOCD



Download:

```
> git clone https://github.com/tonanuvem/k8s-exemplos.git
```

Execução:

```
> sh k8s-exemplos/argocd_run.sh
```

OBRIGADO!

**Resumindo o que vimos hoje**

- **Microserviços (conceitos e cenários de uso)**
- **Contêineres em Docker**
- **Orquestradores de Contêineres**
- **Service Mesh**

## Próximos passos

- Fundamentals of Containers, Kubernetes and OpenShift  
<https://www.edx.org/es/course/fundamentals-of-containers-kubernetes-and-red-hat>
- Try OpenShift  
<https://www.openshift.com/try/>
- EKS workshop  
<https://www.eksworkshop.com/>

FIAP