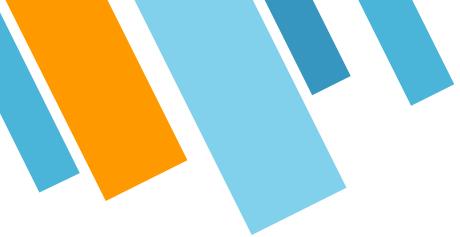




Bem vindo ao mundo dos Contêineres



TONIN R. BOLZAN

Software Engineer ❤️ DevOps

CTO na Metrossoft S/A - Santa Maria/RS

12 anos de experiência

<https://bolzan.io>





SIMUS

O melhor sistema de saúde

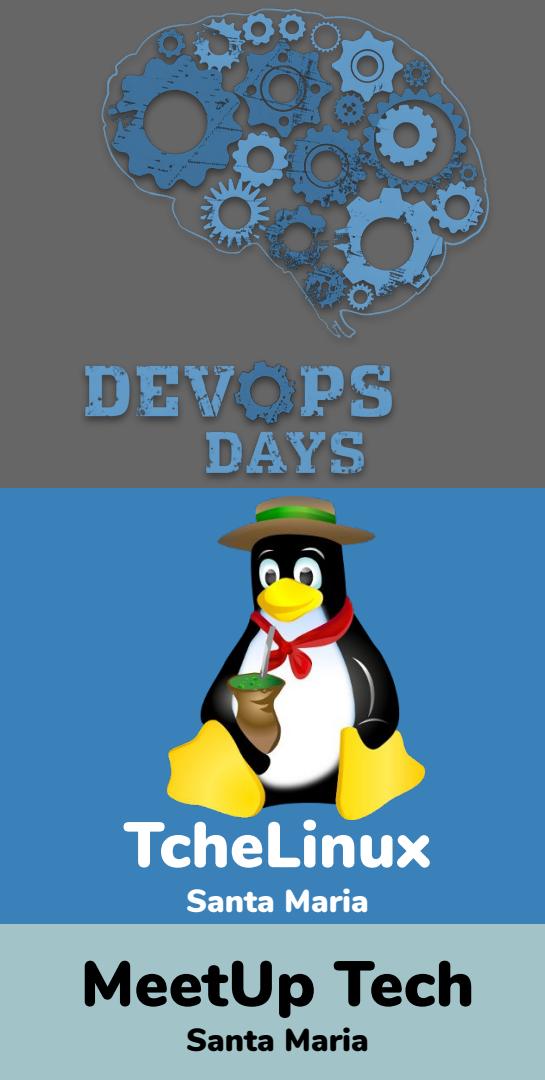
UmanoTec.com

**Respiramos
Software Livre**

Especialista em Software
para Saúde
SUS / Privado

Milhares de cidadãos
atendidos diariamente.

Operacional e Gestão



1.

DevOps

Um universo de possibilidades entre o
Desenvolvimento e as Operações



DevOps Borat
@DEVOPS_BORAT



To make error is human. To propagate error to all server in automatic way is #devops.

16:55 - 26 de fev de 2011

1.643 3.342 pessoas estão falando sobre isso



Cometer erros é humano. Propagar o erro para todos os servidores de forma automática é #devops



“O DevOps é a combinação de filosofias culturais, práticas e ferramentas que aumentam a capacidade de uma empresa de distribuir aplicativos e serviços em alta velocidade”

Amazon AWS



DevOps não é uma atribuição

Em essência, é uma cultura, um movimento, uma filosofia

- » É uma forma de trabalhar, pensar e agir
 - » Promover colaboração entre equipes
 - » Tornar os Devs consciente do Ops e vice-versa
 - » Agilizar o ciclo de lançamentos do produto
 - » Acelerar o tempo para resolver problemas
- 



Nessa idéia entram os containers

2.

Containers são quase VMs

Vamos definir e separar as coisas



Containers não são VMs

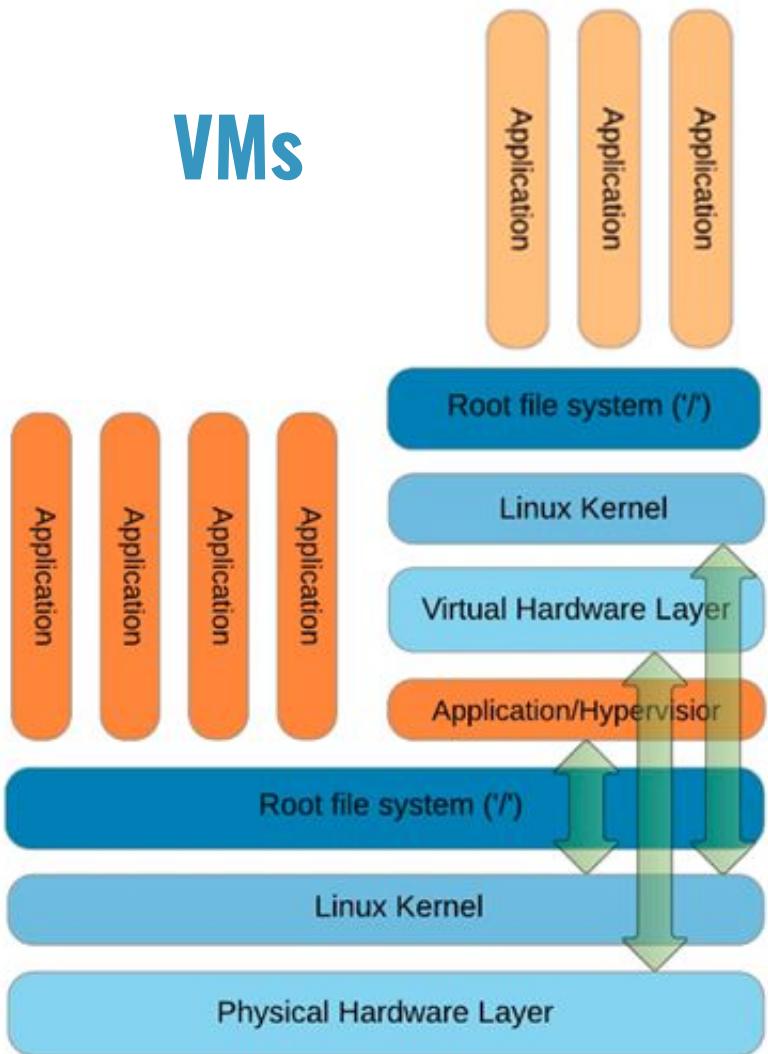
- » Não são virtualizados (hypervisor)
- » São leves e rápidos
- » Normalmente são efêmeros



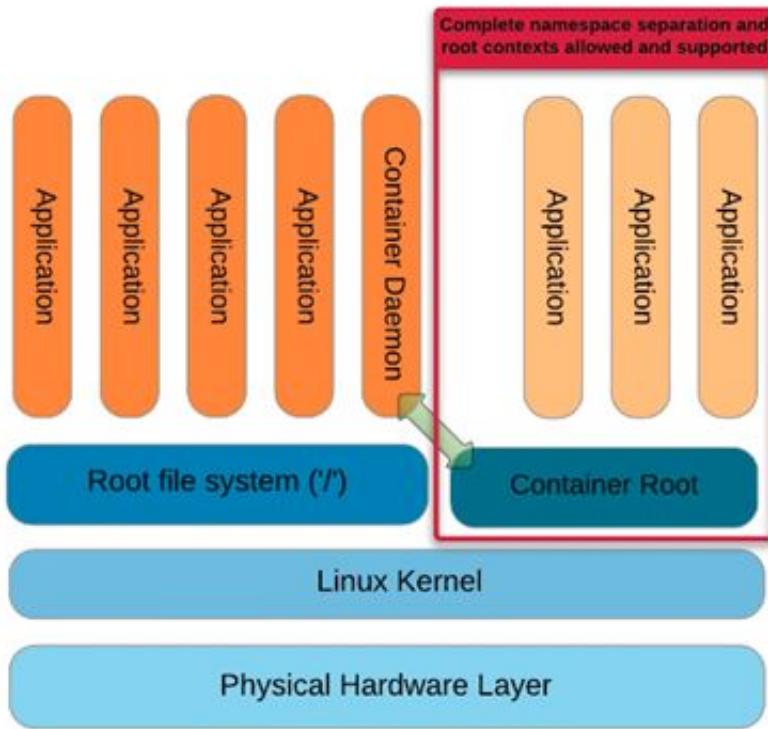
Mas são parecidos

- » Empacotamento e “Isolamento”
- » Um S/O dentro de outro S/O

VMs



Containers





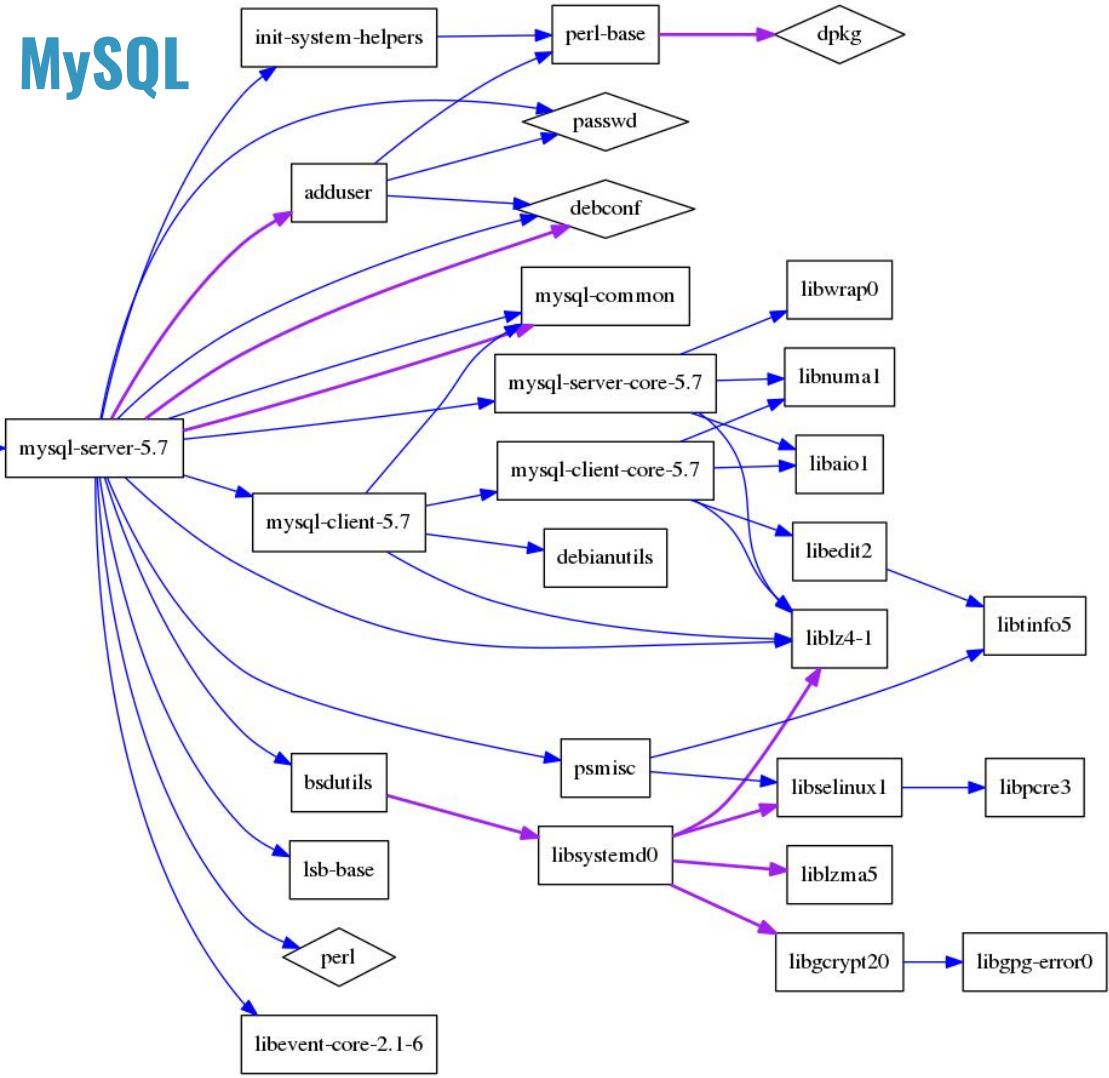
Mas por que eu preciso disso?

Qualquer que seja VMs ou Container

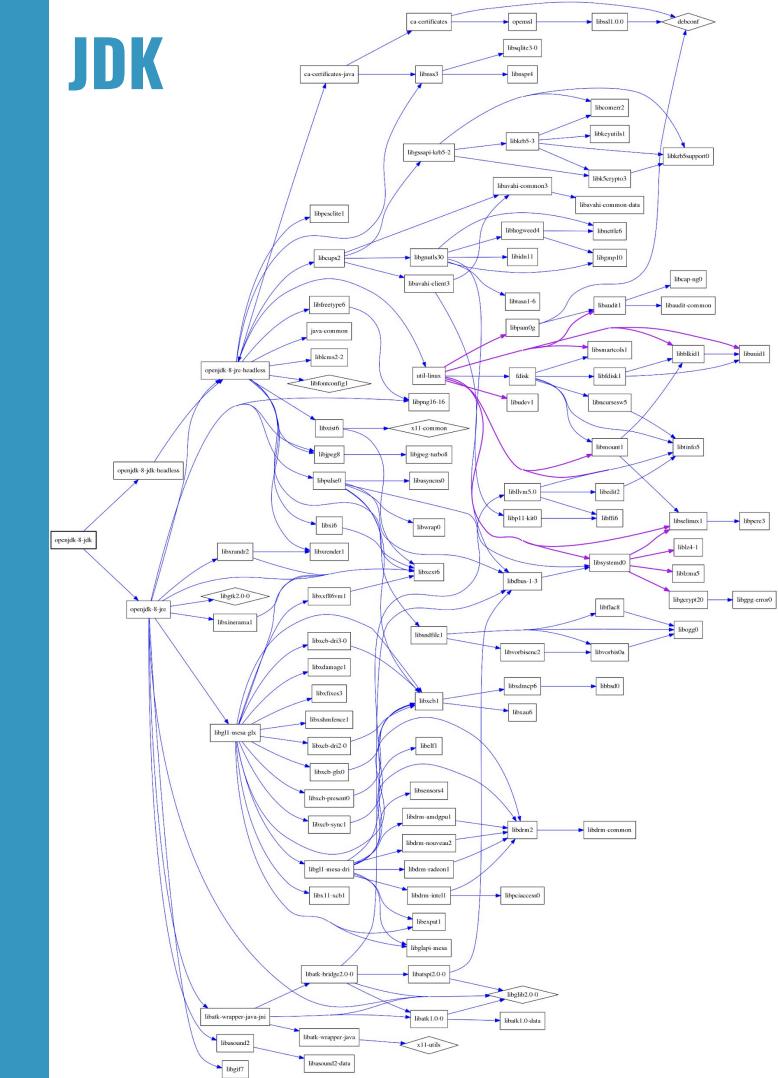
Não é mais fácil “só” instalar o programa

- » WampServer
 - » MAMP
 - » LAMP / LEMP
- 

MySQL



JDK



Não é mais fácil instalar o programa?

NÃO

- » Você “suja” todo o seu sistema
- » Fica limitado a uma versão do software
- » Cada S/O tem sua forma de instalação
- » Difícil de reproduzir



Muitos benefícios

Qualquer que seja VMs ou Container

- » Segurança
 - » Independência
 - » Empacotamento
 - » Isolamento
 - » Limitação de recursos
 - » Reprodutibilidade
 - » Agilidade
- 

Porque usar container e não VM?

- » São leves e rápidos
- » São pequenos

TODOMUNDOUSA

3.

Tools Overview

Vamos as ferramentas

Host

Linux
Windows
Mac OS x

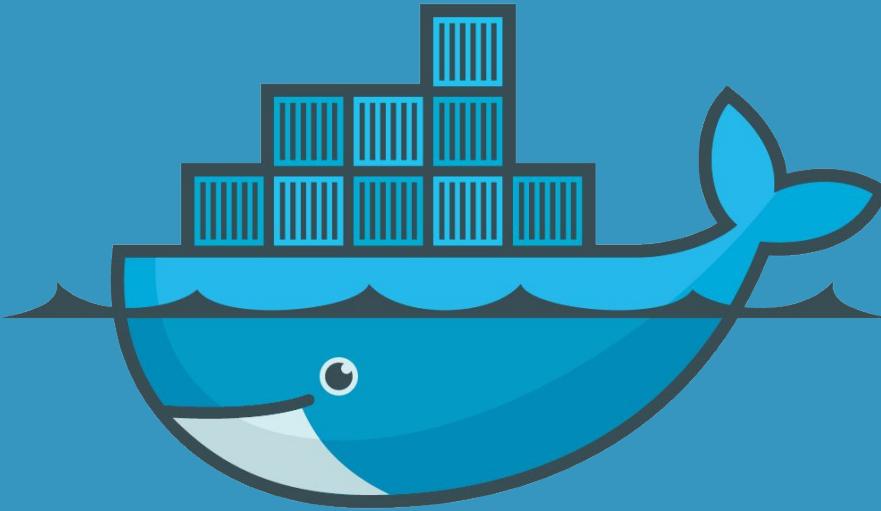
Container

Docker
LXC
Rkt
Snap
Flatpak

Orchestration

Kubernetes
Mesos
Amazon ECS
Azure ACS
Google GCE

DOCKER

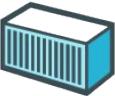


Conceitos



Build

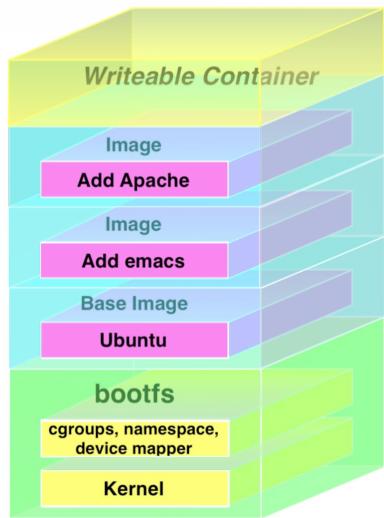
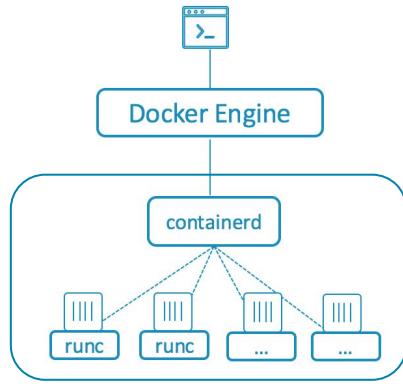
- » **Docker CLI**
 - » Executável de linha de comando
- » **Docker Engine**
 - » Daemon que roda como serviço
- » **Container runtime**
 - » Responsável por “rodar” o container
- » **Container image**
 - » Formato de armazenamento em camadas
- » **Repositório**
 - » <https://hub.docker.com>



Ship



Run



Instalação Ubuntu 18.04

```
# apt install docker.io
```

run - Faz download e roda uma aplicação docker

```
$ docker run --rm -ti gcr.io/google-containers/debian-base-amd64:v1.0.0 sh
```

```
$ docker run --rm -ti metrosoftsa/php-dev:latest php -a
```

```
$ docker run --rm -ti debian bash
```

docker run :> Manda rodar o container

--rm :> Limpa tudo depois de rodar

-t :> tty

-i :> interactive

Container :> <uri>/<nome>/<imagem>:<versao>

Dockerfile

```
FROM metrosoftsa/php-dev:latest
ENV APP_ENV development
COPY . /srv
RUN composer install -d /srv --no-dev --no-ansi --no-progress
CMD ["php", "/srv/index.php"]
```

Esses arquivos são receitas para uma imagem
“**FROM**” estende uma imagem anterior



build - Roda e empacota a partir das definições do Dockerfile

```
$ docker build -t tonin/php:latest -f Dockerfile .
```

save & load - Realiza o dump e restore da imagem OCI

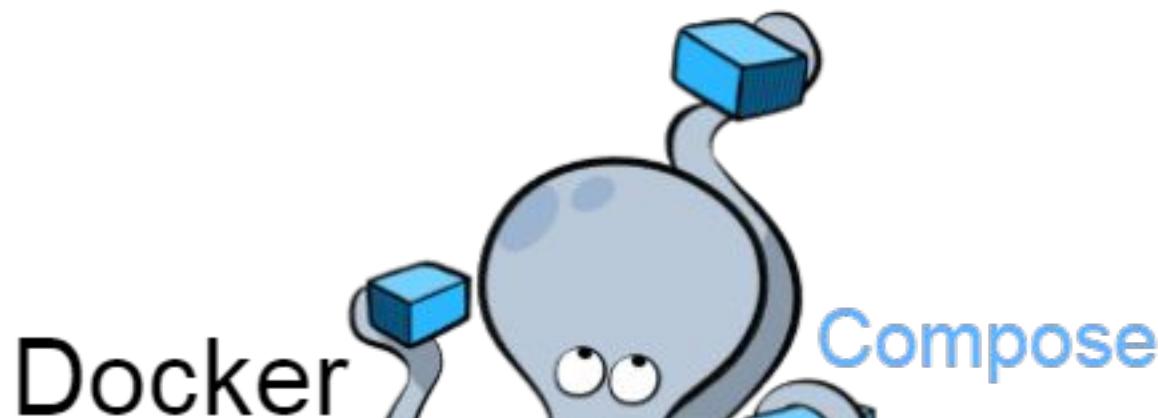
```
$ docker save -o php.tar tonin/php:latest
```

```
$ docker load -i php.tar
```



push - Envia para o repositório docker hub (padrão)

```
$ docker push tonin/php:latest
```



apt install docker-compose

4.

Local vs. Produção

Qual a diferença?



Em produção você tem muitos containers

Orquestração **Kubernetes**





“Put simply, Kubernetes made the simple things hard and the hard things possible”

Tom Wilkie, Grafana/Google

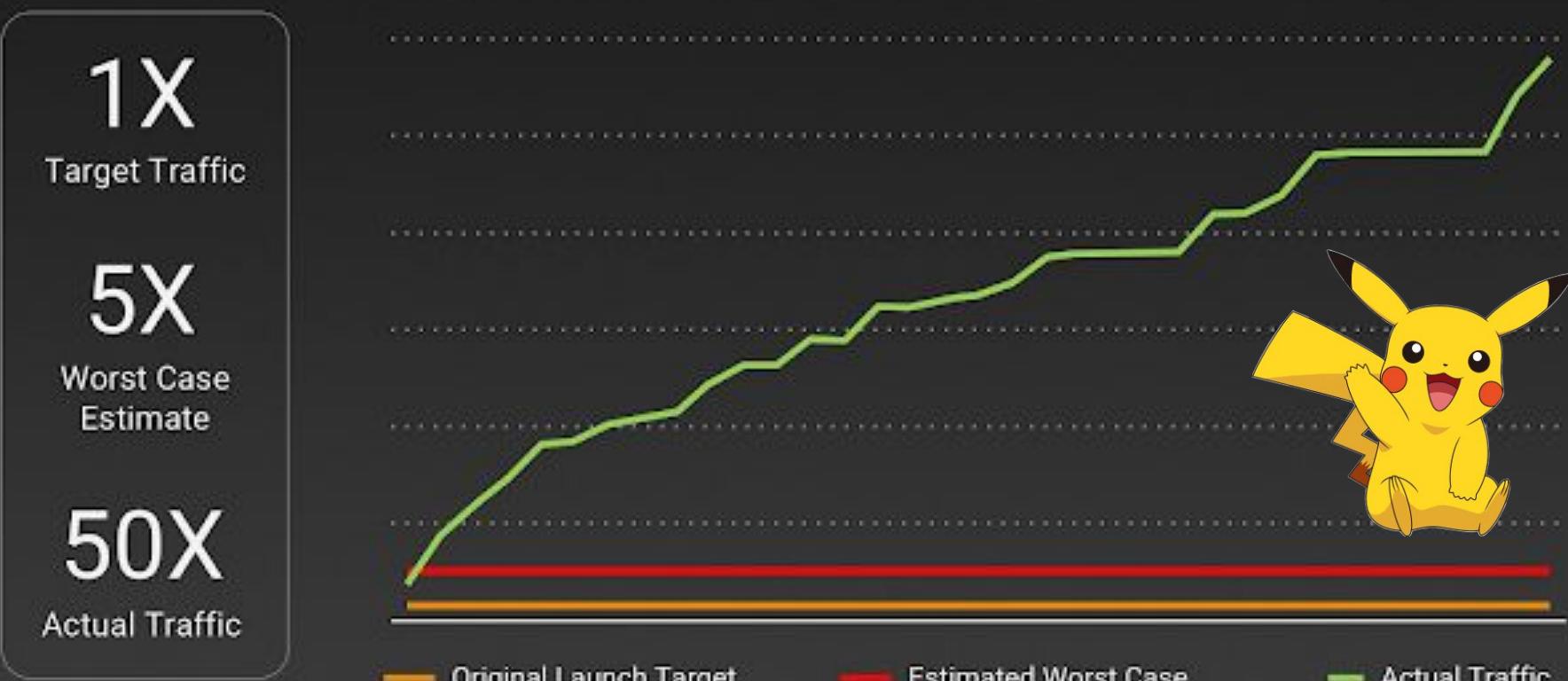
Orquestração

- » Automação
- » Padronização
- » Agilidade
- » Controle
- » Monitoramento
- » Escalabilidade

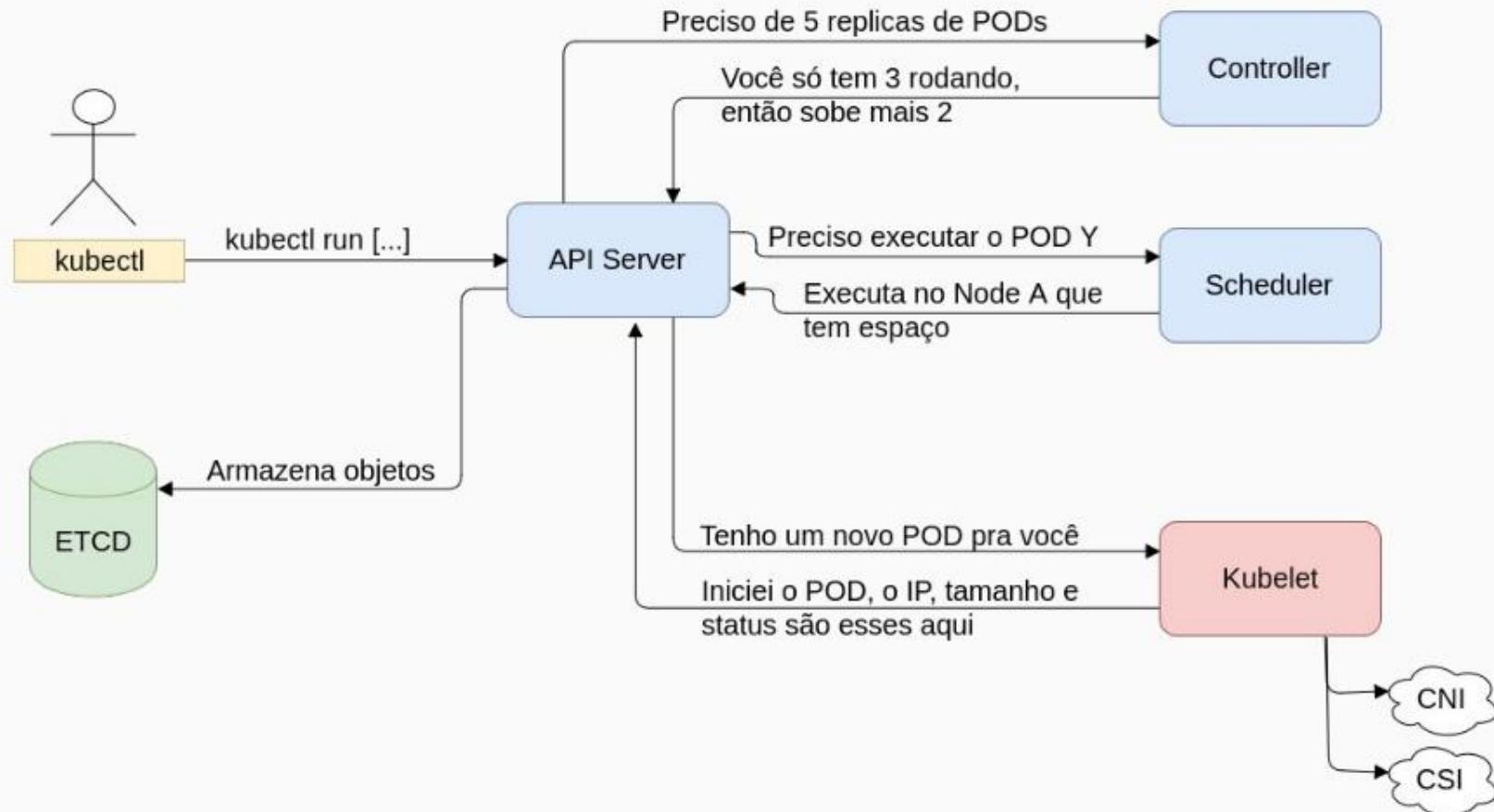


Google Cloud OnBoard Porto Alegre

Escalabilidade Pokémon GO



Ecossistema Kubernetes





HELM

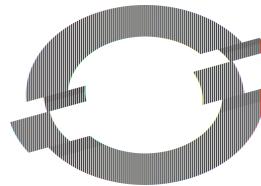
Gerenciador de pacotes para Kubernetes

<https://helm.sh>

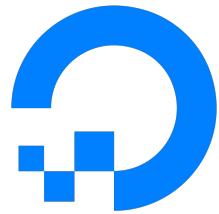
`helm install stable/wordpress`



Google Cloud Platform



RED HAT®
OPENSHIFT
Container Platf



DigitalOcean



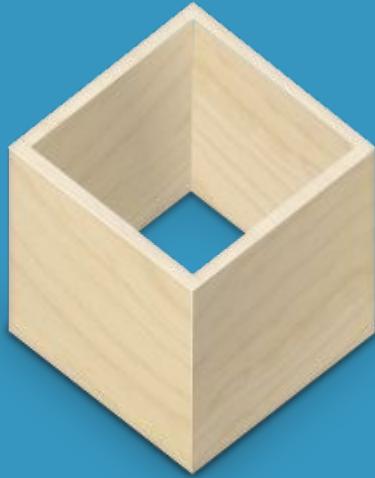
 CLOUD NATIVE
COMPUTING FOUNDATION
<https://cncf.io>



5.

E as aplicações gráficas?

Containers para aplicações gráficas



FLATPAK



snapcraft

SNAP - Padrão no Ubuntu

» `sudo snap install android-studio`

<https://snapcraft.io/store>

FLATPAK - Padrão no Fedora

» `flatpak remote-add --if-not-exists flathub https://flathub.org/repo/flathub.flatpakrepo`

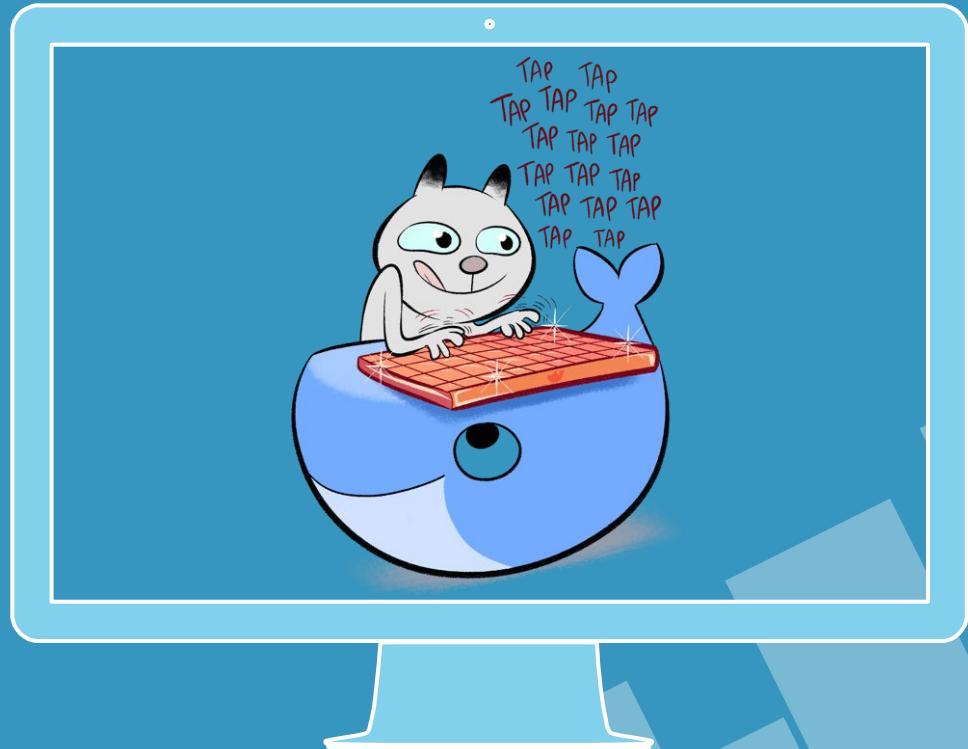
» `flatpak install flathub com.google.AndroidStudio`

<https://flathub.org>

PANORAMA GERAL

- Linux
 - Sistema Operacional
- Docker
 - Container Runner
- Docker Compose
 - Multi-container spec
- Docker HUB
 - Site de hospedagem
- Kubernetes
 - Orquestrador
- HELM
 - Gerenciador de pacotes
- Snap
 - Container para o desktop
- Flatpak
 - Container para o desktop

Docker Hands-on



Perguntas





Obrigado

