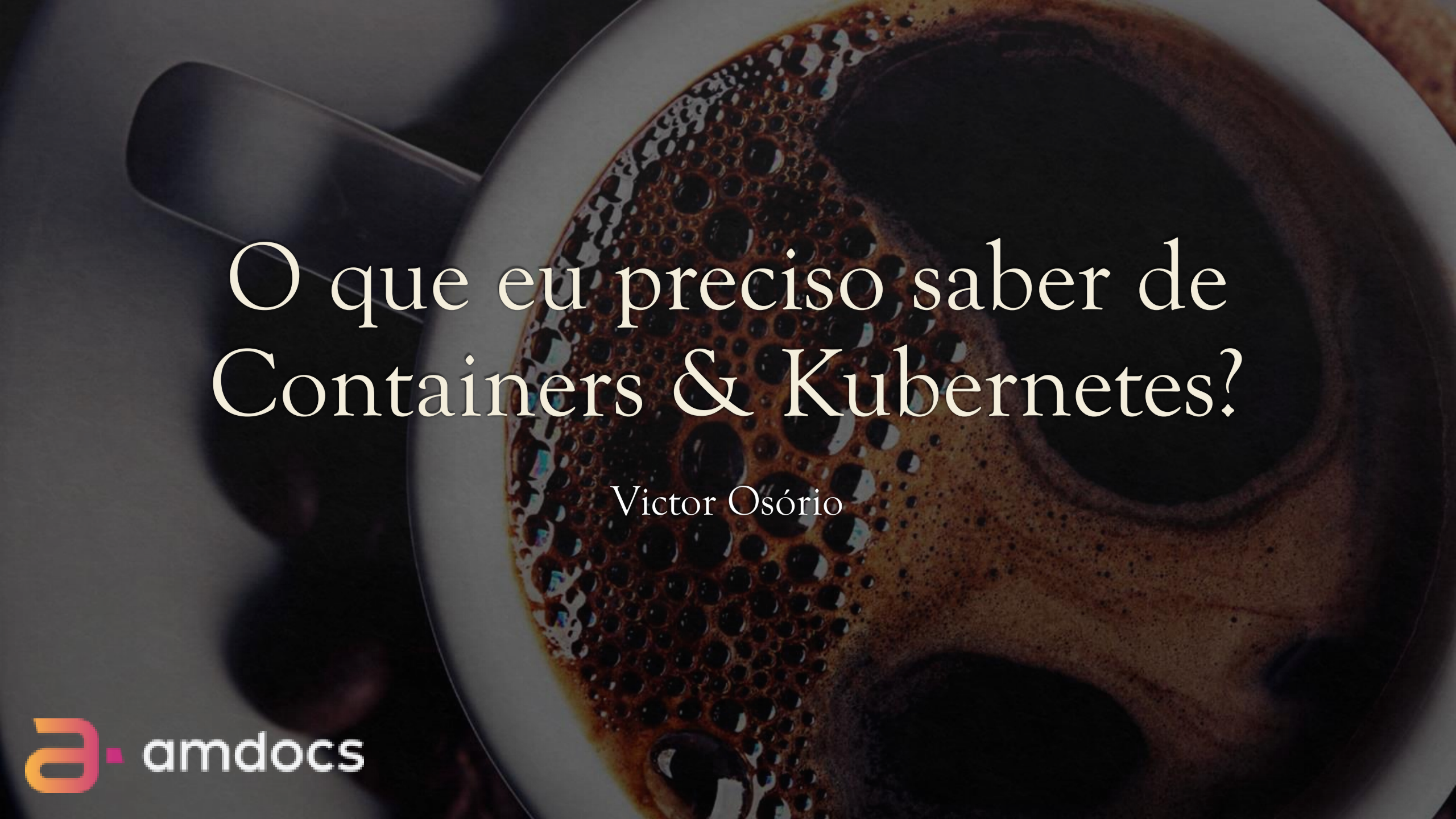


O que eu preciso saber de Containers & Kubernetes?

Victor Osório

amdocs

Quem sou eu?

Victor Osório

- Software Development Specialists
- Product Development @ Amdocs
- EC 02 @ Unicamp
- Java & Apache Kafka
- Twitter: @vepo

amdocs

Agenda

- Porque containers?
- Docker: The Killing Feature
- CNCF: Linux rules the world
- Kubernetes
- Novos padrões arquiteturais

Porque containers?

01

SERVIDORES BARE METAL

02

APPLICATION SERVER

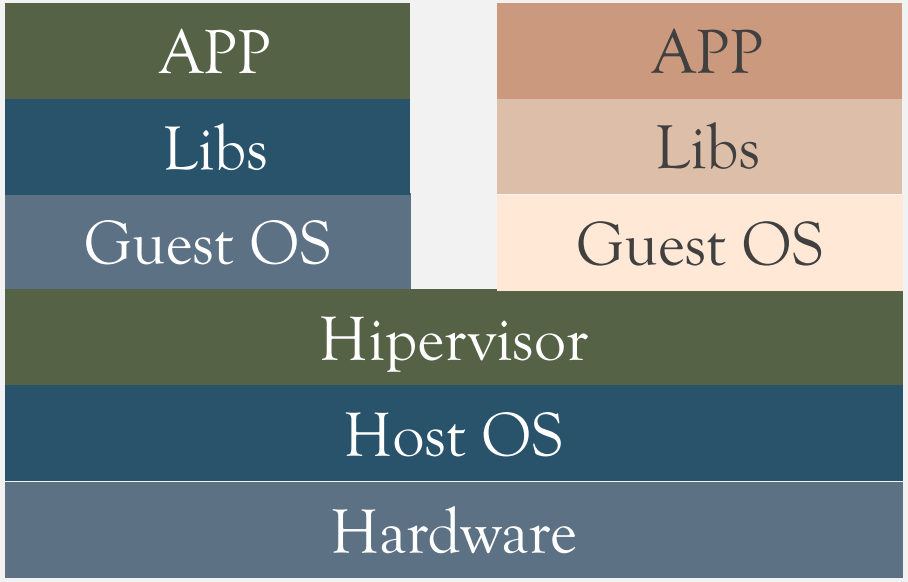
03

RECURSOS COMPARTILHADOS

04

“NA MINHA MÁQUINA FUNCIONA!”

Seria a virtualização uma solução?



- Provê isolamento
- Provê portabilidade
- Desperdício de recursos

Virtualização

Problemas com a virtualização

- Desperdícios de memória
- Tempo de inicialização
- Imagens excessivamente grandes
- Dificuldade para definir imagens

Produtos de mercado

Vagrant | VirtualBox | VMware | QEMU

Uma possível solução no Kernel do Linux?

Funcionalidades

cgroups (2006)

- Limitar uso de memória, CPU, I/O

LXC (2008)

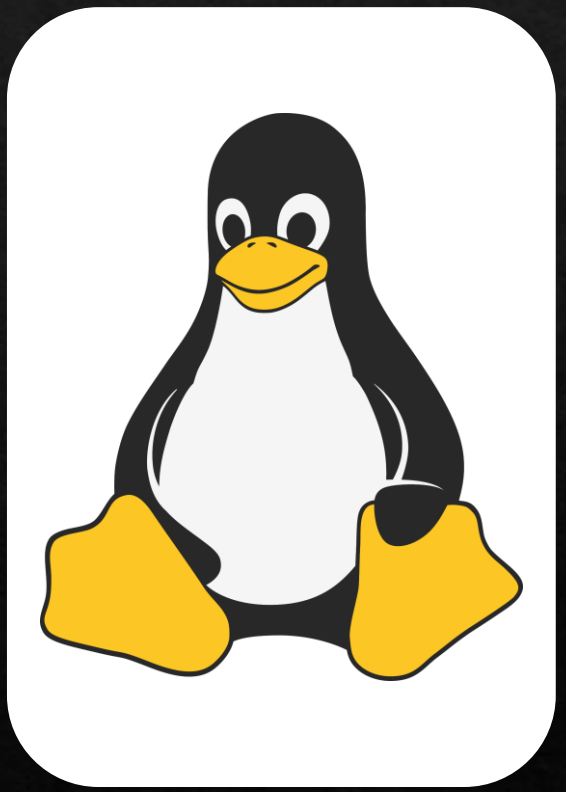
- Isolamento de processo, networking, mounted file system

iptables (1998)

- Roteamento de interface de rede

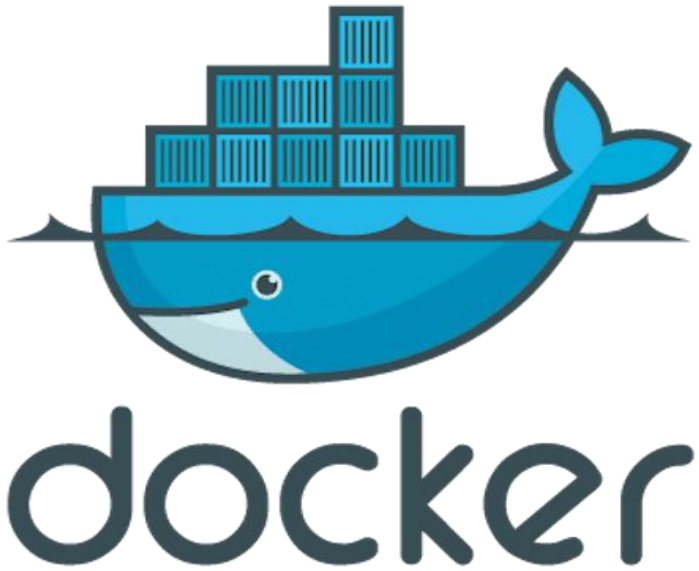
OverlayFS (2010)

- Combidar múltiplos FS



Existia solução, só não era fácil!

2013 



Como Docker facilitou a conteinerização?

Dockerfile

- Conjunto de passos que irá construir uma imagem única porém compartilhadaq

Docker Image

- Contém toda informação necessária para executar um processo

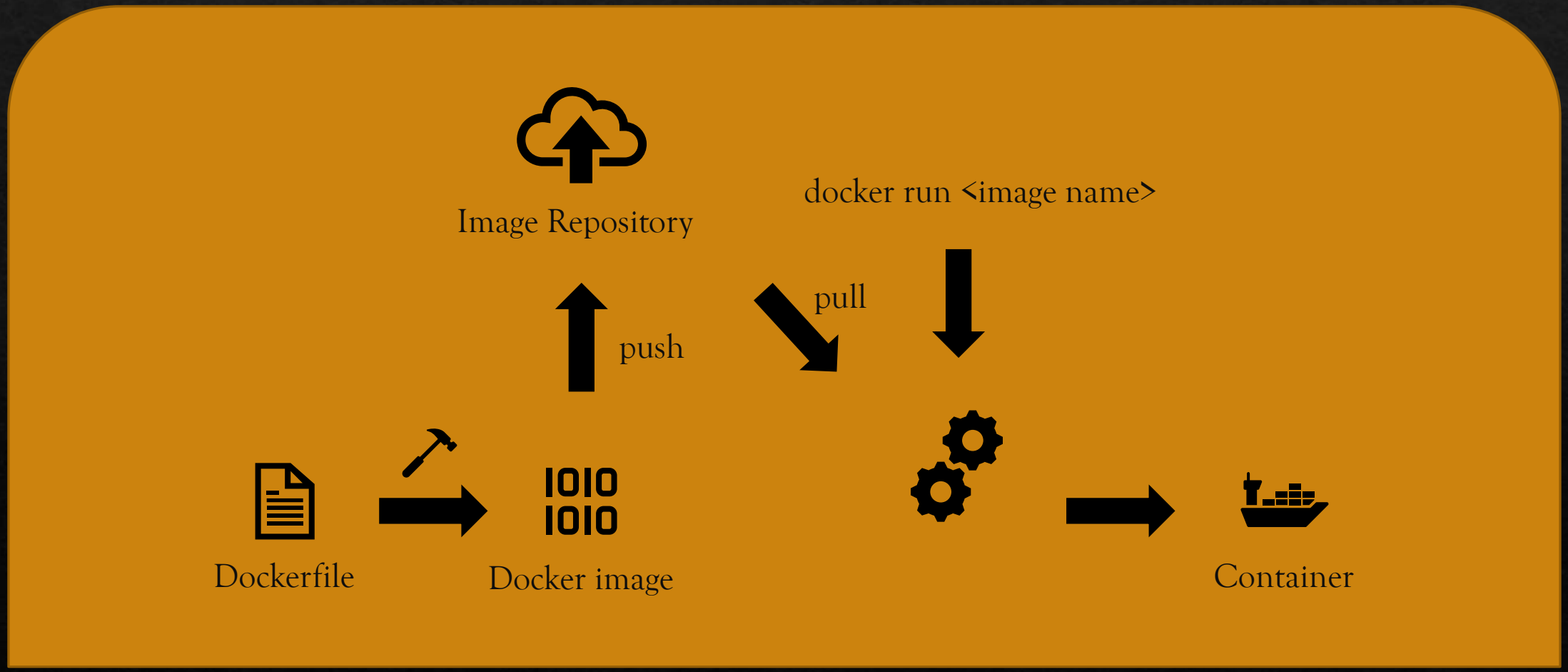
Docker Engine

- Controla ciclo de vida do processo

Docker Hub

- Facilita o compartilhamento de imagens

Ciclo de vida de uma aplicação docker



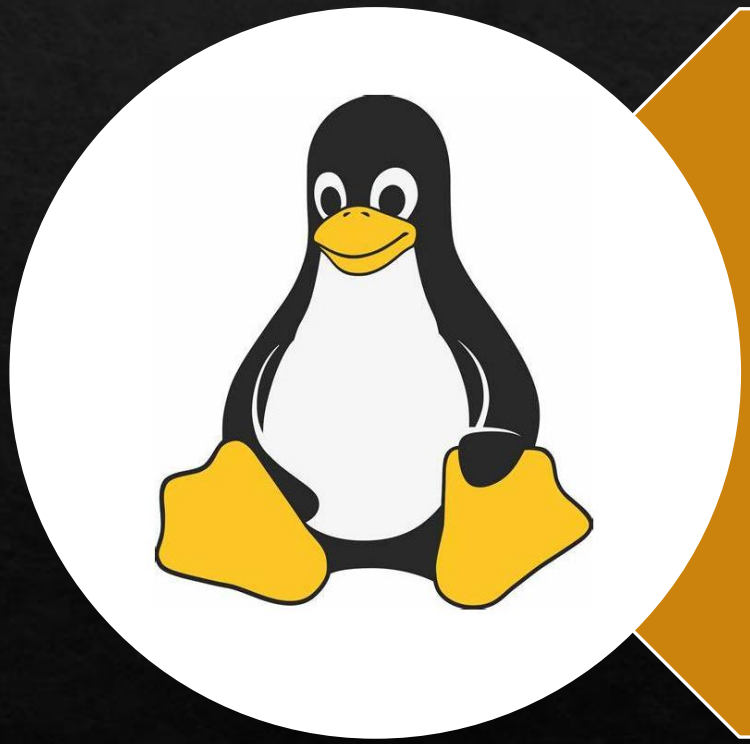
Exemplo de Dockerfile

```
# syntax=docker/dockerfile:1
FROM ubuntu:18.04
COPY . /app
RUN make /app
CMD python /app/app.py
```

Docker ou Containerd.io?

- Docker abriu seu padrão para o CNCF criando o containerd.io
- Existem outros padrões de containers que são compatíveis
- É bem provável que apareçam novas implementações

CNCF: Linux rules the world



Efeitos do Docker

- CI/CD
- Linux como SO padrão para servidores
- Linux Foundation se torna Cloud Native Computing Foundation
  - Definir padrões
  - Landscape
- Kubernetes
- Microservices

Um novo padrão

- Cloud Native
- 12 Factor Application
  - `https://12factor.net/pt_br/`

Kubernetes

Orquestração de Containers

- Plataforma
- Distribuído
- Auto gerenciado

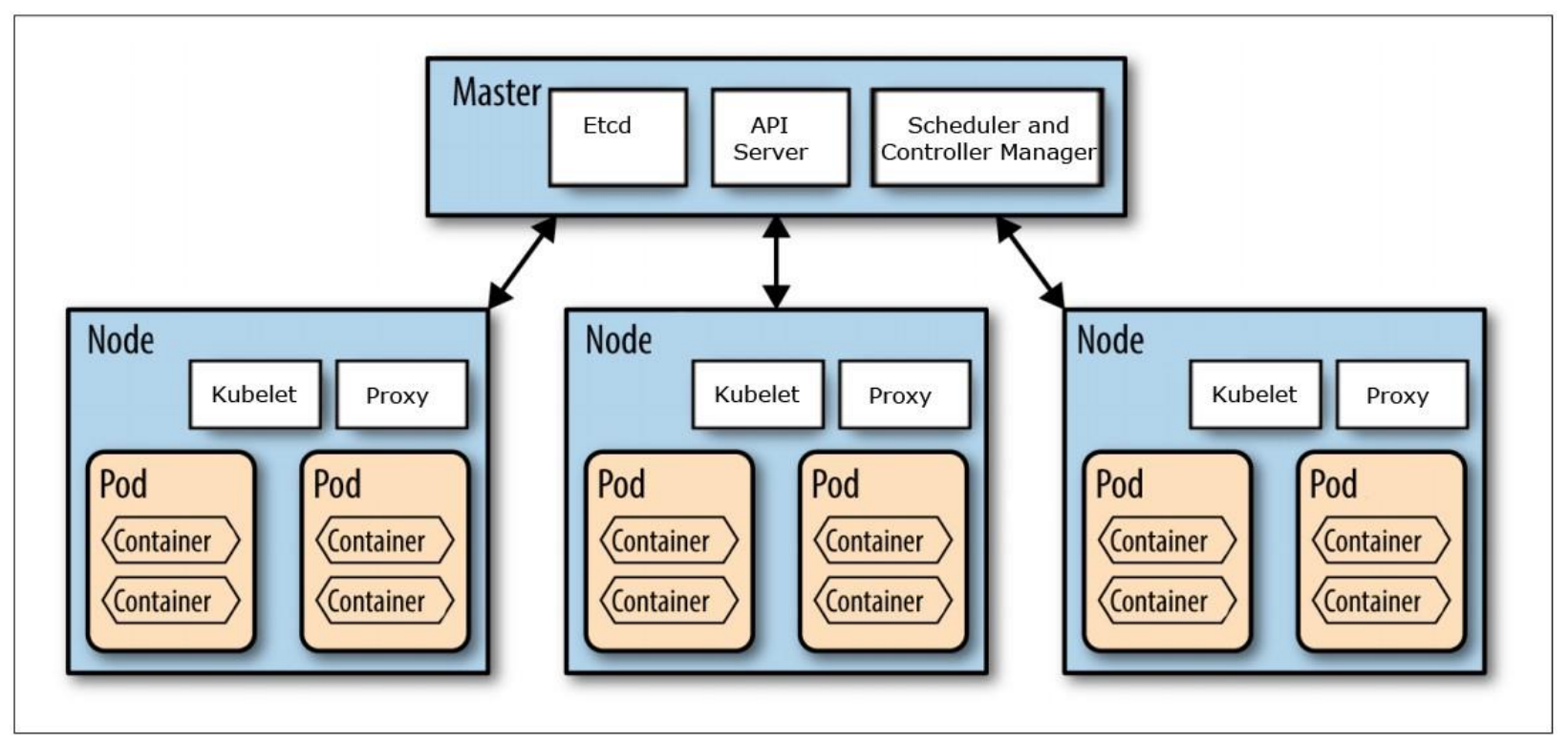
Building Blocks

- POD
- Service
- Deployment
- ReplicaSet
- Ingress
- DaemonSet
- Job
- Volumes

API

- Extensível
- Operators

Kubernetes


Master
Etcd
API Server
Scheduler and Controller Manager
Node
Kubelet
Proxy
Pod
Container
Container
Pod
Container
Container
Node
Kubelet
Proxy
Pod
Container
Container
Pod
Container
Container
Node
Kubelet
Proxy
Pod
Container
Container
Pod
Container
Container

Tipos de PODs

- Control Plane → Controla o estado do Cluster
- Application Plane → Implementa a lógica da aplicação

kubernetes

Tutoriais

- Docker Tutorial
  - https://github.com/vepo/docker-tutorial
- Kubernetes Tutorial
  - https://github.com/vepo/k8s-tutorial