





Introdução a Kubernetes



III Workshop de Tecnologia da Informação
IFCE

Quem Somos

Atlântico

Antecipar o futuro. Esse é o nosso negócio.

Paulo Abner



Jonnys Gomes

AGENDA

Docker

Conhecendo o kubernetes

Kubernetes Básico

Core Concepts

Hands on

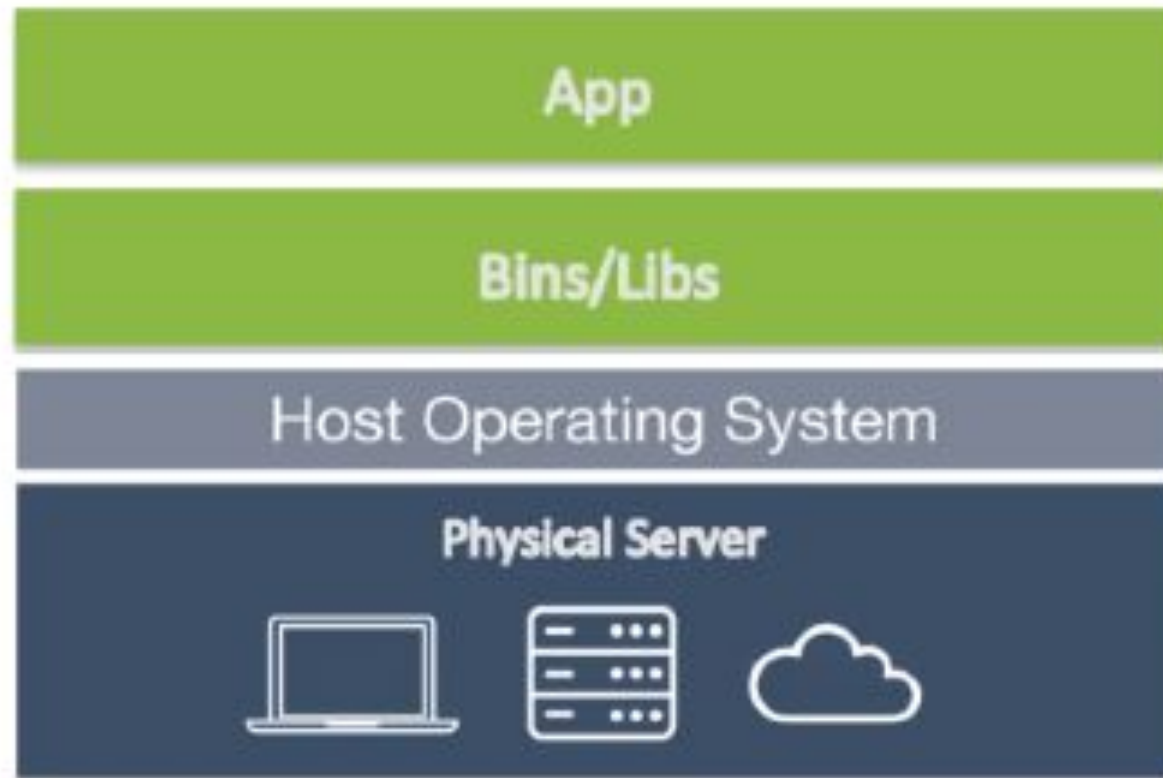




No princípio

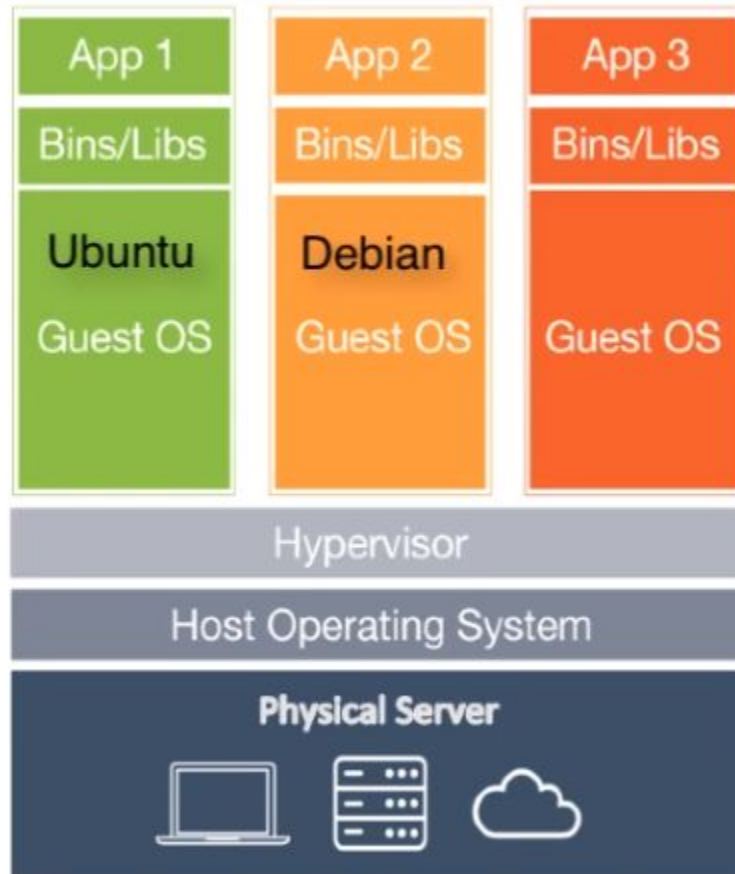


Pré-Virtualização

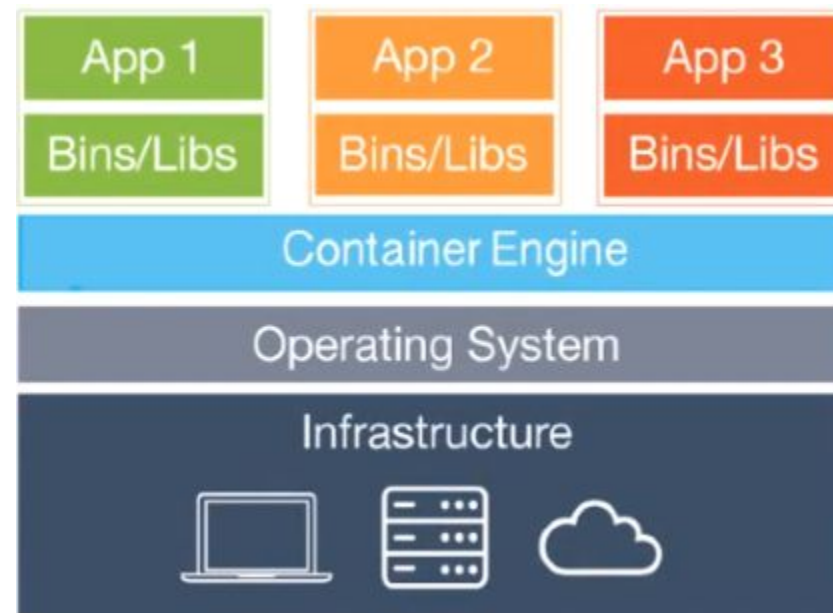


Pre-Virtualization

Virtualização



Virtualização com containers





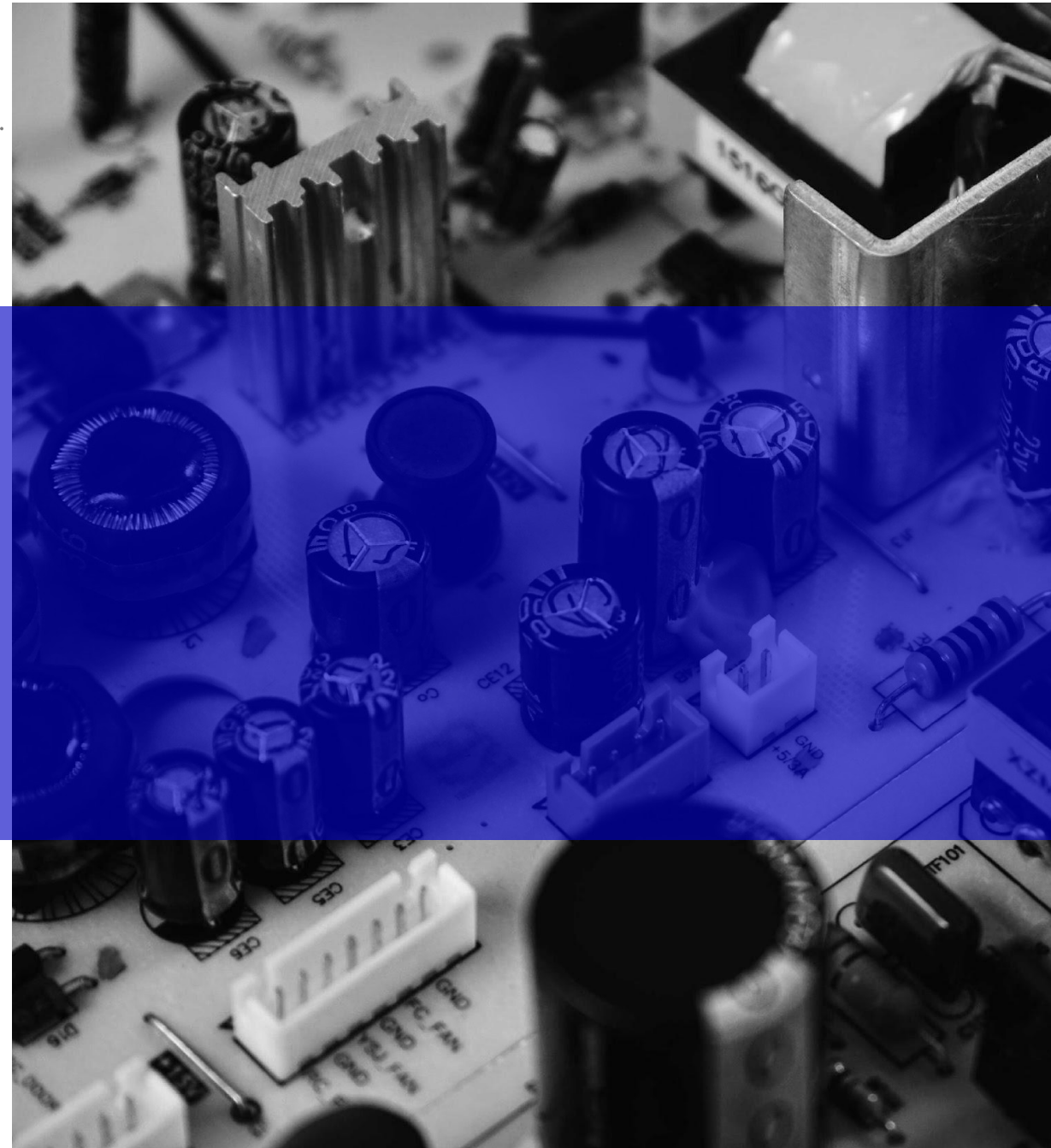
docker

Docker



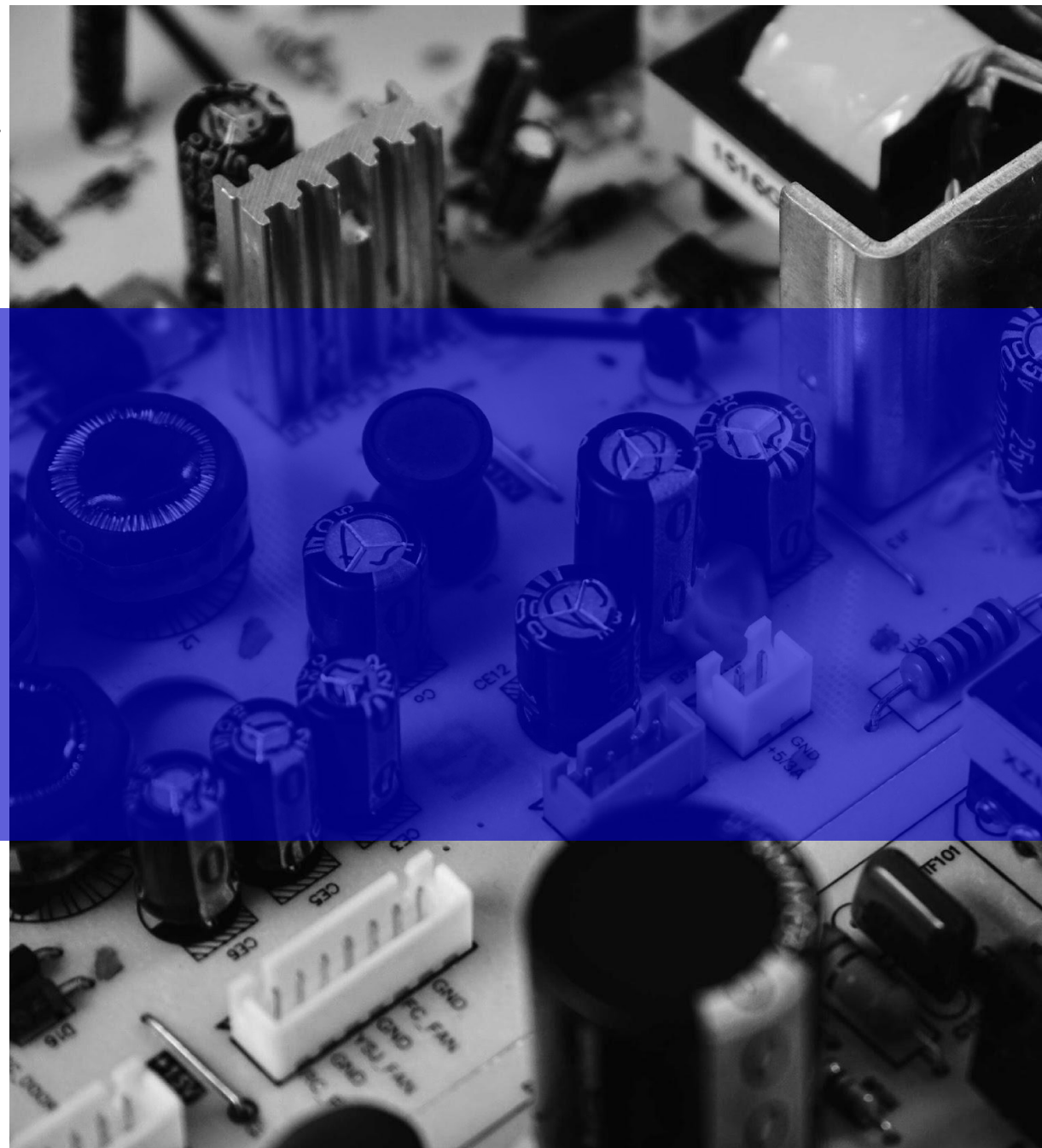
Docker

- Surgiu em uma empresa de hospedagem chamada *dotCloud*
- Lançado como open source em 2013 com o nome de Docker
- Desenvolvido em linguagem Go
- Principais contribuidores: Google, IBM, Cisco, Microsoft, Red Hat

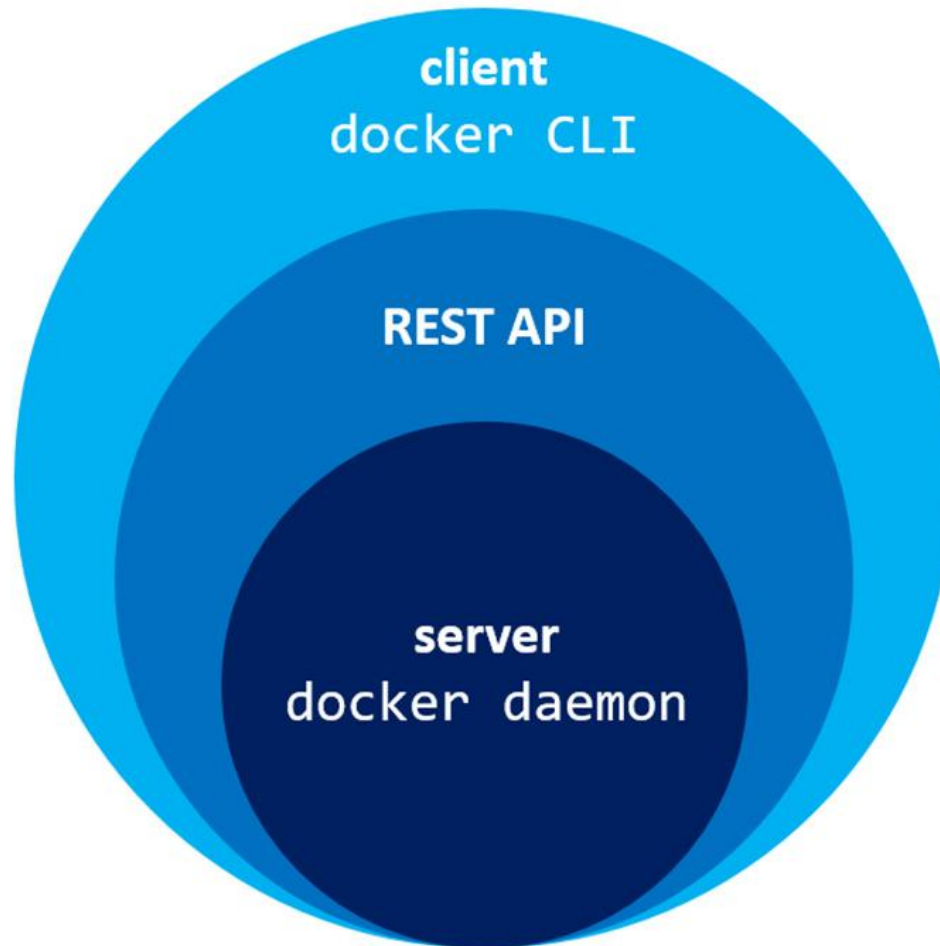


Docker

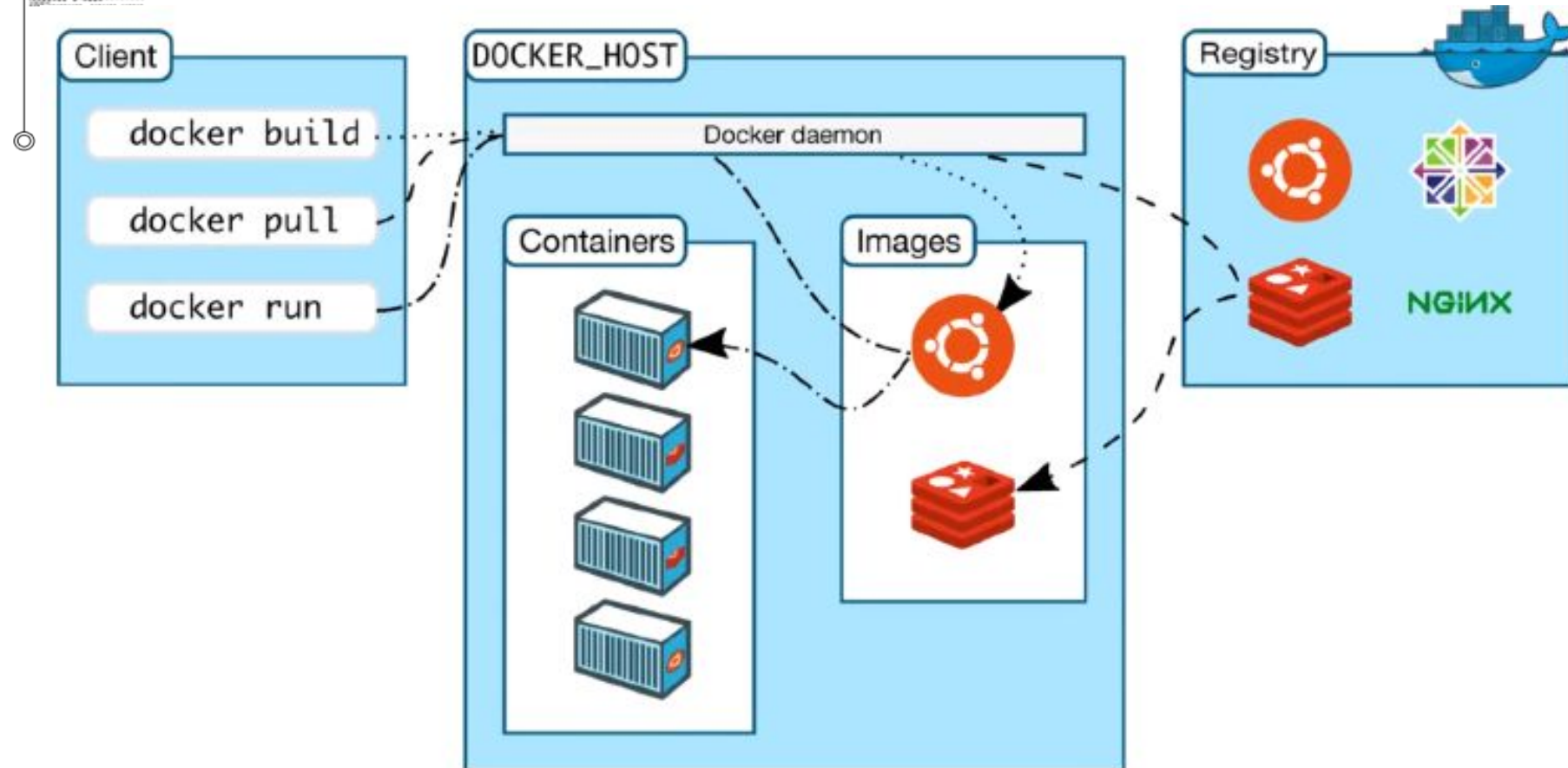
- Plataforma de virtualização a nível de SO, onde o kernel permite que múltiplos processos sejam executados isoladamente no mesmo host
- Cada um desses processos isolados é chamado de container
- Docker utiliza módulos do kernel linux para criar ambientes isolados dentro de um mesmo host



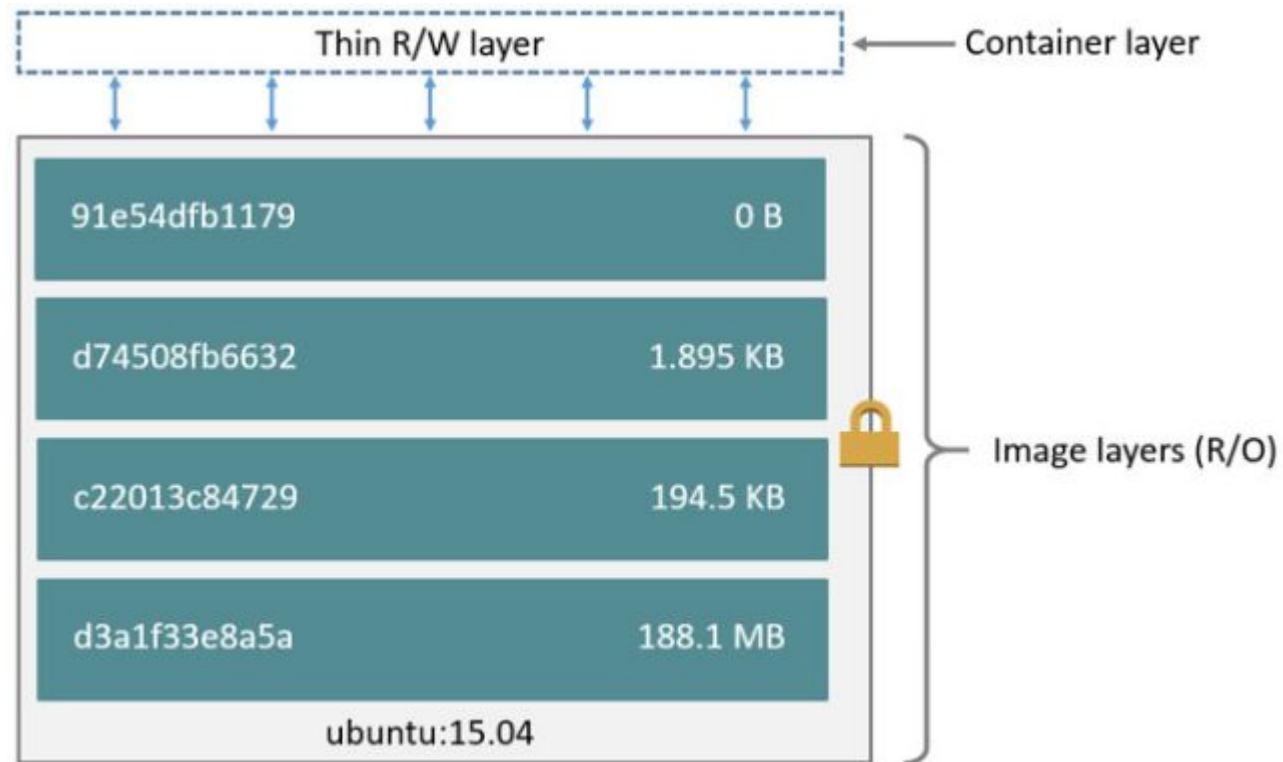
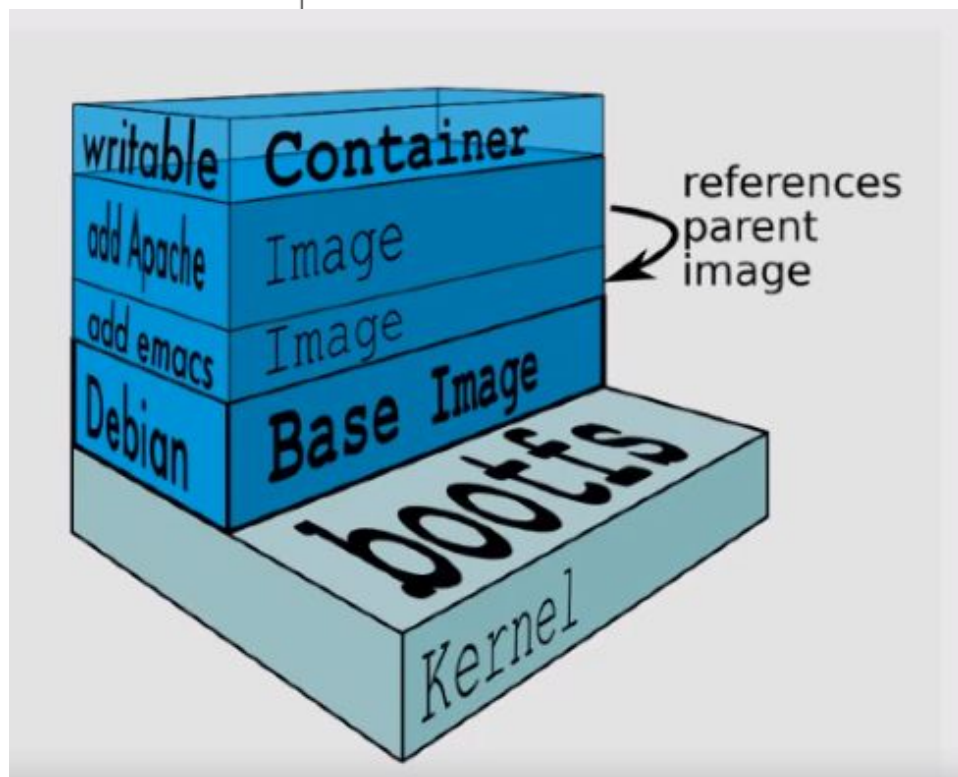
Arquitetura



Arquitetura



Layers





Layers

- Todas as camadas, exceto a última, são read-only
- Quando um novo container é gerado a partir de uma imagem uma nova layer r/w é adicionada na pilha de layers
- Toda mudança em um container é aplicado na camada superior
- Quando um container é deletado a camada superior também é deletada
- Quando um container é deletado, as camadas somente leitura não são deletadas
- Múltiplos containers podem compartilhar as camadas de somente leitura

Imagens

- Imagens são templates para criar containers
- Imagens são criadas com os comandos build ou commit
- Imagens são formadas por várias camadas (layers)
- Imagens são armazenadas em docker registries
 - Docker Hub
 - Docker Trusted Registry
 - AWS Elastic Container Registry

Container

- Se uma imagem é uma classe, um container é uma instância daquela classe
- Containers são leves e portáteis
- Containers são criados a partir de imagens
- Dentro do container devem estar todas os binários e dependências necessárias para executar a aplicação



Registry

- Local onde são armazenadas as imagens
- Público ou privado
- Dentro de um registry as imagens são armazenadas em repositórios
- Repositório é uma coleção de imagens com mesmo nome, diferenciados por tags



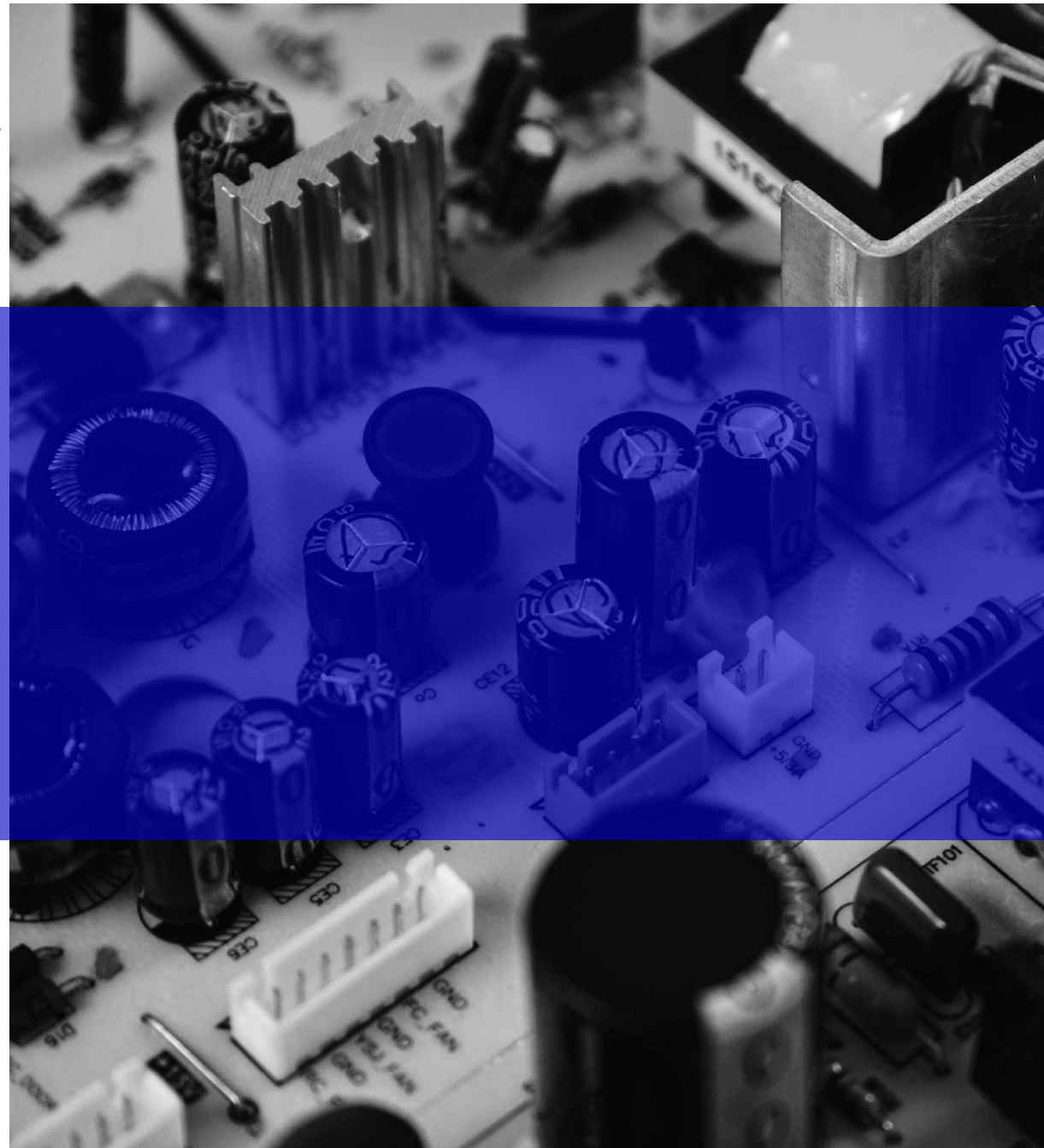
Docker Hub



Docker

Instalação

- `sudo apt-get update`
- `sudo apt-get install docker.io`
- adicionar usuário no grupo docker
- `sudo usermod -aG docker ${USER}`
- logout/logon (reboot now)



Docker - Comandos - Busybox

- docker images
- docker pull
- docker run
 - -i -t
 - --name
- docker ps
- docker commit
- docker inspect
- docker logs
- docker rm
- docker rmi



Hands on





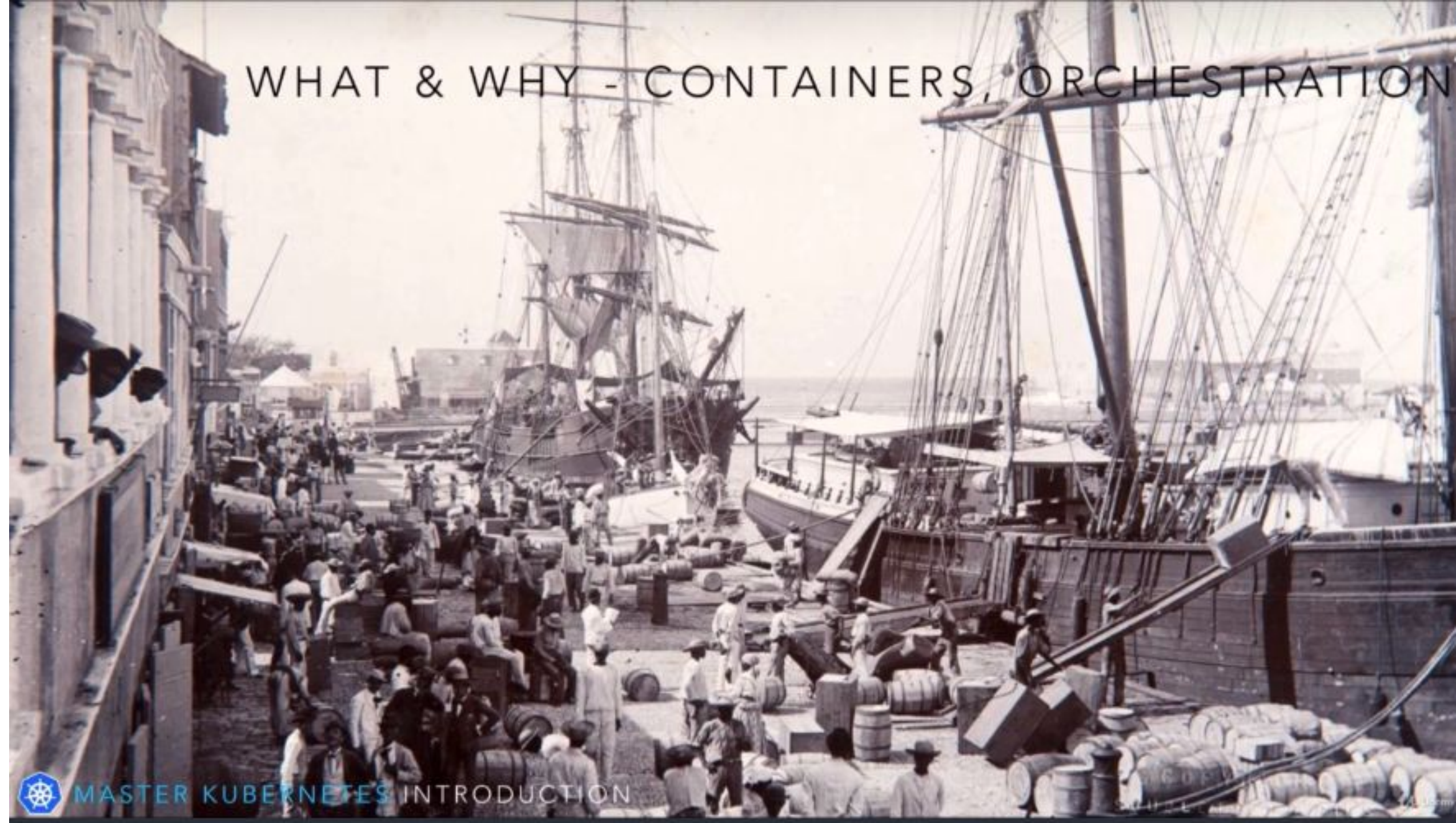
Kubernetes



Kubernetes

	Containers	Orquestrador
Função	Manter softwares separados dentro de um sistema operacional	Define as relações entre containers, de onde são criados, como são escalados e como se conectam com o mundo
Antes	Máquinas Virtuais Instalações diretas	Scripts Configurações manuais
Principais	Docker RKT Garden LXC Mesos	Kubernetes Docker swarm Amazon ECS Mesos

WHAT & WHY - CONTAINERS, ORCHESTRATION



WHAT & WHY - CONTAINERS, ORCHESTRATION





Kubernetes

- *“Kubernetes is an open-source system for automating deployment, scaling, and management of containerized applications.” **
- Originalmente foi projetado e criado pelo Google (Google Borg)
- Versão 1.0 foi lançada em 2015 e no mesmo ano o projeto foi doado para a Cloud Native Computing Foundation
- Principais contribuidores: Google, Red Hat, Microsoft, IBM, ...
- Última versão estável: v1.10.10 **

* <https://kubernetes.io/>

** <https://github.com/kubernetes/kubernetes/releases>



Kubernetes

Features

- Simplifica o deploy de aplicações em containers
- Replicação de containers
- Auto-scaling de containers
- Health check de containers
- Monitoramento de recursos por container
- Gerenciamento de volumes persistentes
- Service discovery
- Load Balancing



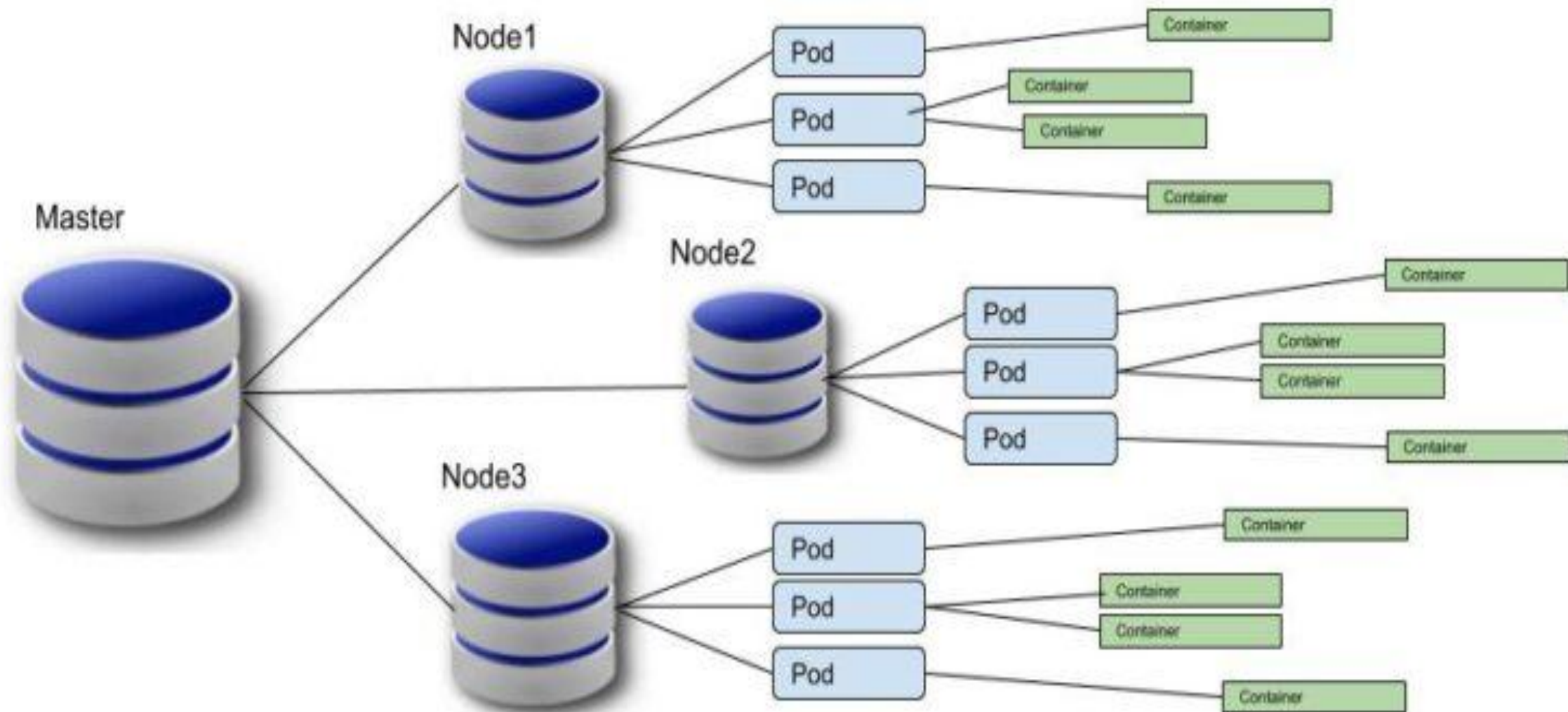
Kubernetes

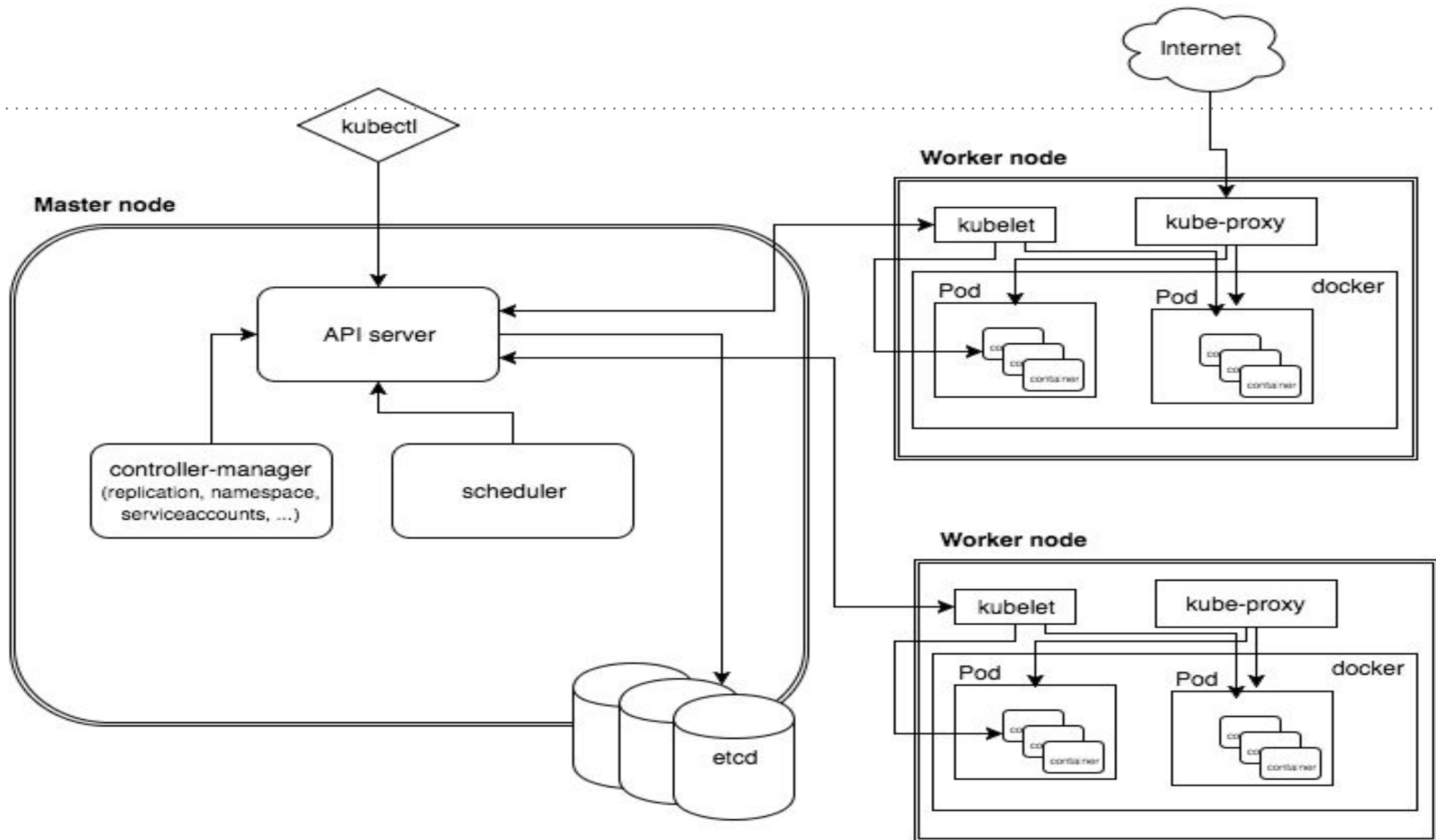
Conceitos

- Master
- Nó ou Nodo
- Pod
- Serviços
- Controller Manager
 - Replication Controller
- kubelet
- kubectl
- kube-proxy

Kubernetes

Conceitos







Kubernetes

Minikube

- Instalação all-in-one do k8s
- Todos os componentes rodam em uma VM
- Funções não são suportadas:
 - Load Balancers
 - Persistent Volumes
 - Ingress
- Usaremos kubectl para se comunicar com minikube
- PS: Minikube requer virtualização



Minikube

Instalação

- Instalar kubectl
- Instalar um hypervisor (minikube requirement)
- Instalar minikube



Instalando kubectl

```
sudo apt-get install curl
```

```
curl -LO https://storage.googleapis.com/kubernetes-release/release/$(curl -s  
https://storage.googleapis.com/kubernetes-release/release/stable.txt)/bin/linux/amd64/kubectl
```

```
chmod +x ./kubectl
```

```
sudo mv ./kubectl /usr/local/bin/kubectl
```

```
kubectl version
```



Instalando Minikube

```
sudo apt-get install virtualbox
```

```
curl -Lo minikube https://storage.googleapis.com/minikube/releases/v0.30.0/minikube-linux-amd64
```

```
chmod +x minikube && sudo mv minikube /usr/local/bin/
```

```
sudo minikube start
```

```
kubectl run hello-minikube --image=gcr.io/google_containers/echoserver:1.4 --port=8080
```

```
kubectl expose deployment hello-minikube --type=NodePort
```

```
kubectl get pod
```

```
curl $(minikube service hello-minikube --url)
```

```
kubectl delete deployment hello-minikube
```

```
minikube stop
```




Kubernetes

auth / auto

- \$USER/.kube/config



Kubernetes

Objects

- “Kubernetes Objects” são entidades persistentes que representam o estado do cluster.
 - Pod
 - Deployment
 - Service
 - Volume
 - Configmap



Primeira App K8S





Kubernetes

App K8S

- “Deployments” são o que podemos considerar como aplicações ou serviços.
- Deployments são descritos como uma coleção de recursos e referências
- Deployments, podem muitas vezes, ser baseados na forma como é deploy do serviço
- Podem ser descritos em formato YAML



Kubernetes

App K8S - Tomcat

- Iremos fazer o deploy the um Tomcat Server
- Usaremos o repositório oficial do docker
- Tarefas
 - Definir o deployment
 - qual container usar
 - qual serviços esse container oferecerá
 - Expor os serviços
 - qual porta será exposta para o mundo
 - Deploy em nosso cluster



Kubernetes

App K8S - Deployments

- Definir o deployment
- O mais simples deployment em kubernetes é termos um único pod
- Nosso deploy terá um único Pod, sem redundância, sem separação de serviços. Será apenas um deploy no K8s
- Vamos criar um arquivo em branco chamado deployments.yaml
- Importante
 - image name
 - número de réplicas
 - container port
 - nome da aplicação

Kubernetes

App K8S - Deployments

- Criar deployment
 - `kubectl apply -f deployments.yaml`
- Expor o Pod
 - `kubectl expose deployment tomcat-deployment --type=NodePort`
- Descobrir a url de acesso
 - `minikube service tomcat-deployment --url`
- Revisar o que foi feito
 - definimos de deployment
 - expomos os serviços

Kubernetes

Basic Commands

- kubectl prover acesso ao kubernetes
- Através de linha de comando
- kubectl get pods
 - lista todos os Pods em todos os namespaces
 - mostra o nome do pod, quantas instancias estão rodando, quantos restart houve e a idade
- kubectl describe pods <pod name>
 - Descreve informações sobre todos os pods
 - podemos passar um argumento para um simples pod
- kubectl expose <type> <identificador> --type=NodePort
 - Expõe uma porta para um determinado deployment, pod ou outro recurso



Kubernetes

Basic Commands

- `kubectl port-forward <pod-name> localport:remoteport`
- `kubectl attach <pod name> -c <container>`
- `kubectl exec -it <podname> comando`
- `kubectl label pods <podname> key=value`
- `kubectl run <name> --image=image`

minikube dashboard



Kubernetes

Auto scaling

- `kubectl port-forward <pod-name> localport:remoteport`
- `kubectl attach <pod name> -c <container>`
- `kubectl exec -it <podname> comando`
- `kubectl label pods <podname> key=value`
- `kubectl run <name> --image=image`

minikube dashboard

Referências

kubernetes.io

<https://www.mirantis.com/blog/introduction-to-yaml-creating-a-kubernetes-deployment/>

<https://www.level-up.one>

github

pauloabner

jonnysgomes

OBRIGADO

Paulo Abner (abner@atlantico.com.br)

Jonny Gomes (jonny_gomes@atlantico.com.br)

spartas@atlantico.com.br