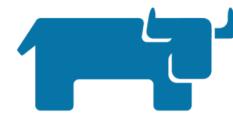




kubernetes

by Google



RANCHER



kubernetes

by Google

jonathanbaraldi@gmail.com

slideshare.net/jonathanbaraldi

Agenda

Containers Docker

Registro

Rancher

Kubernetes

Kubernetes - Arquitetura

Kubernetes - O que é possível

Kubernetes - Soluções



Agenda

Exercício 1 - Docker

Exercício 2 - Build

Exercício 3 - Rancher Single Node

Exercício 4 - Cluster Kubernetes

Exercício 5 - Kubectl

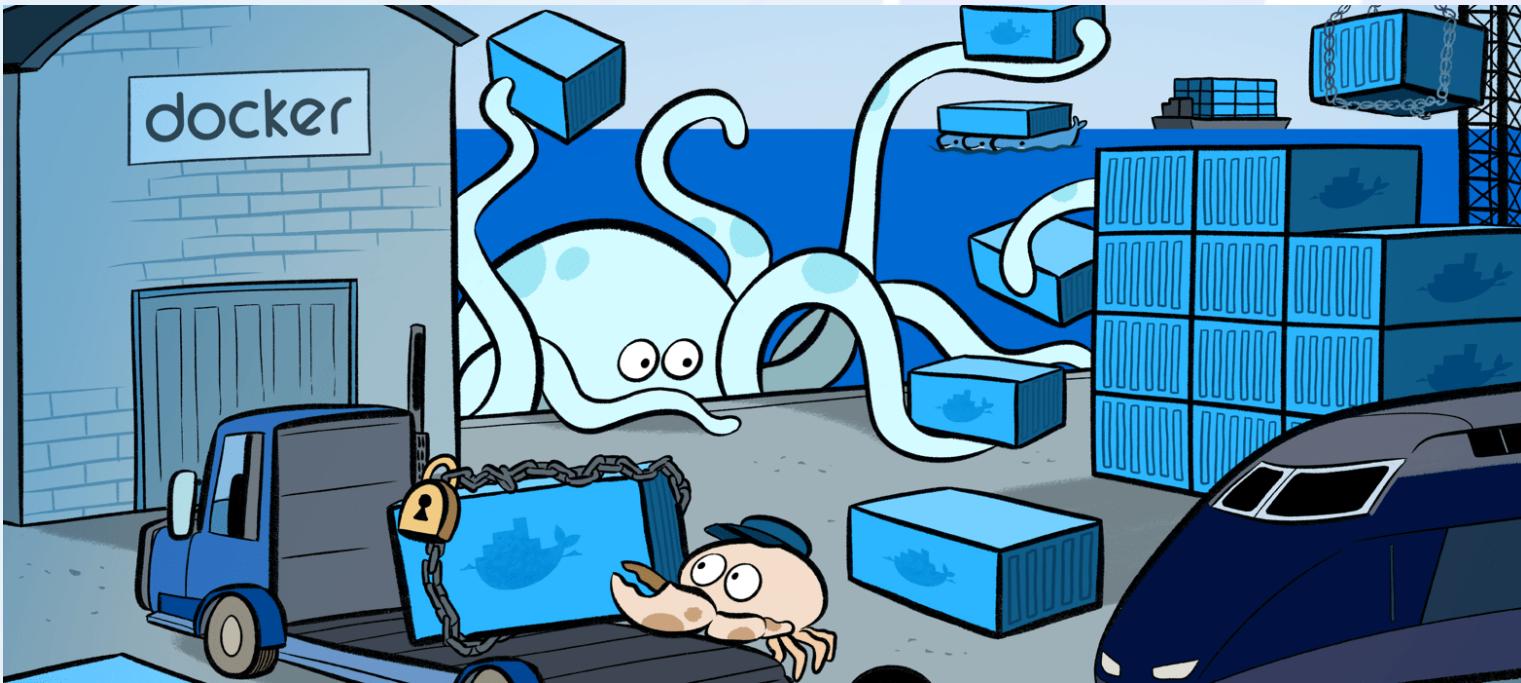


Containers Docker



Containers Docker

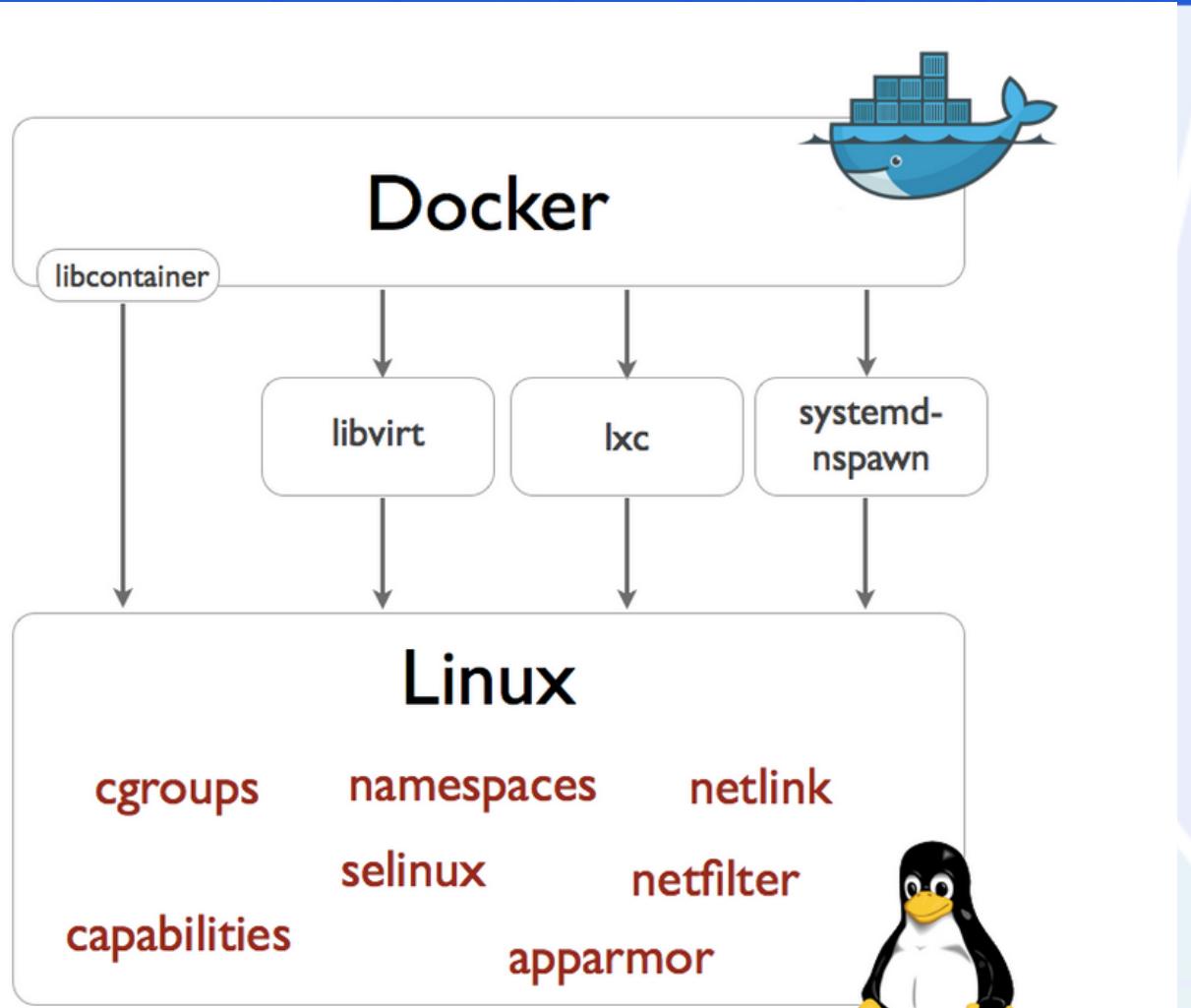
Docker é um projeto de código aberto que automatiza a implantação de aplicativos dentro de contêineres e é mantido pela Docker, Inc.



Containers Docker

O Docker fornece uma camada adicional de abstração e automação da virtualização em nível operacional, utilizando recursos de isolamento do kernel do Linux, como **cgroups** e **namespaces**, e um sistema de arquivos **union mounting** como padrão o **OverlayFS** dando possibilidade ao uso de outros sistema de arquivos.

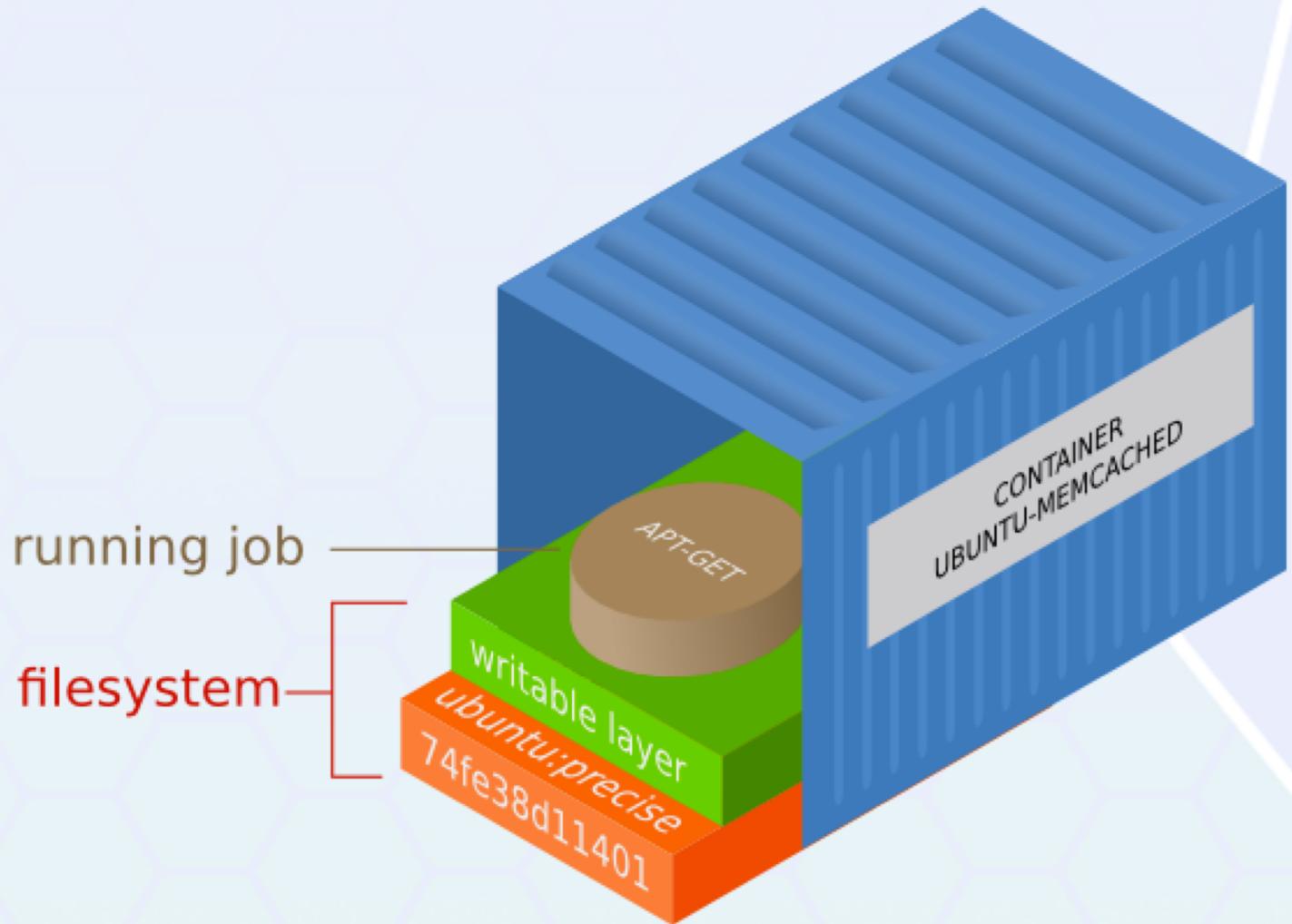
Containers Docker



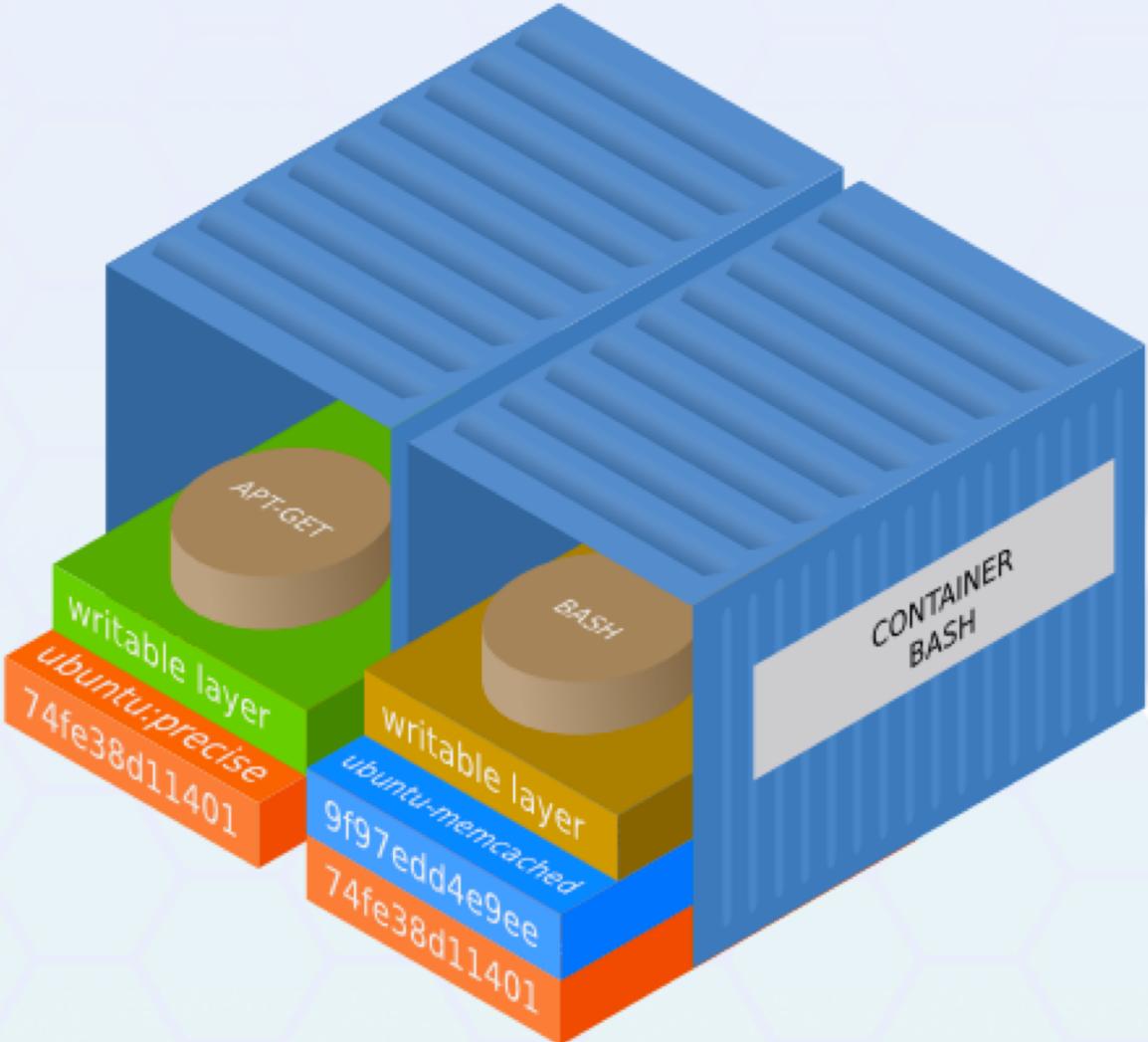
Containers Docker

A idéia do Docker é que cada contêiner rode apenas uma aplicação ou serviço e que seus dados importantes fiquem persistidos fora do container, facultando assim a possibilidade do contêiner ser descartável (destruir o container e recriar sem comprometer a integridade da instalação da aplicação ou serviço).

Containers Docker



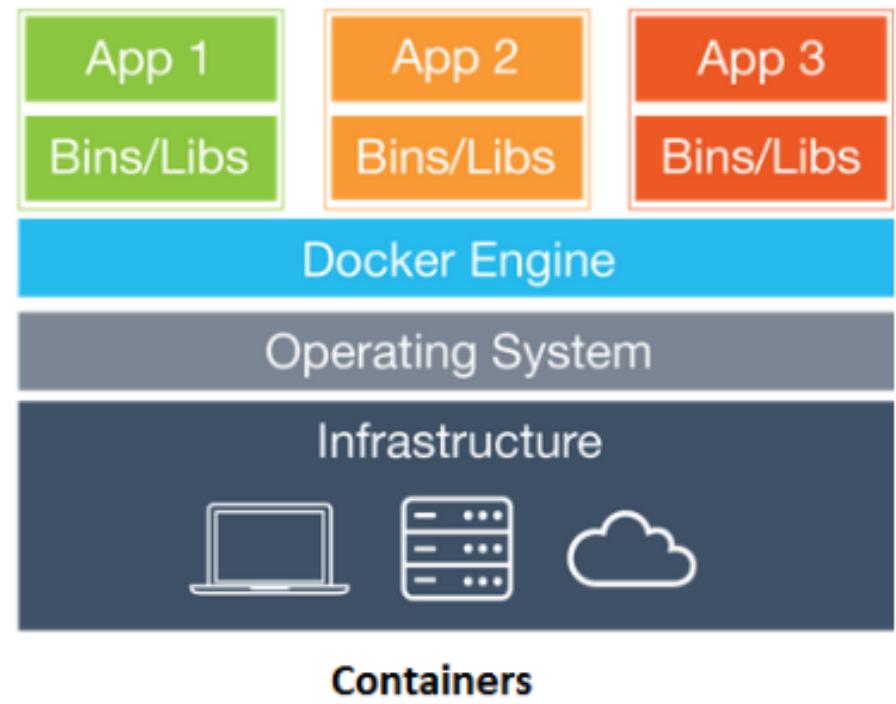
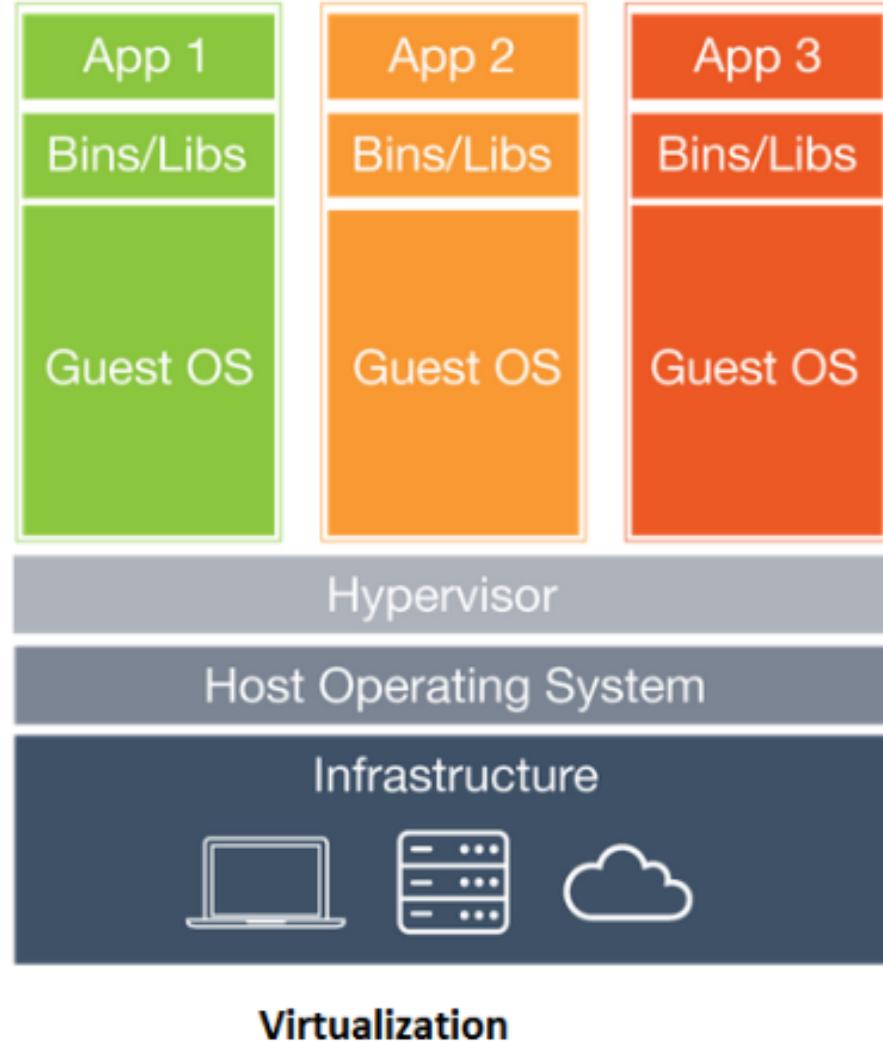
Containers Docker



Containers Docker

Os contêineres e as máquinas virtuais possuem recursos de isolamento e alocação similares, mas funcionam de forma diferente, pois os contêineres virtualizam o sistema operacional e as máquinas virtuais virtualizam um hardware.

Containers Docker



Containers Docker

POR QUÊ USAR?

- **Implementação rápida de aplicativos** - os contêineres incluem os requisitos mínimos de execução do aplicativo, reduzindo seu tamanho e permitindo que sejam implantados rapidamente.
- **Portabilidade entre máquinas** - um aplicativo e todas as suas dependências podem ser empacotadas em um único contêiner independente da versão host do kernel Linux, distribuição ou modelo de implantação.

Containers Docker

POR QUÊ USAR?

- **Controle de versão e reutilização de componentes** -

Você pode controlar versões sucessivas de um container, inspecionar diferenças ou reverter para versões anteriores.

- **Imagens otimizadas** - As imagens do Docker são

tipicamente muito pequenas, o que facilita a entrega rápida e reduz o tempo de implantação de novos contêineres de aplicativos.

Containers Docker

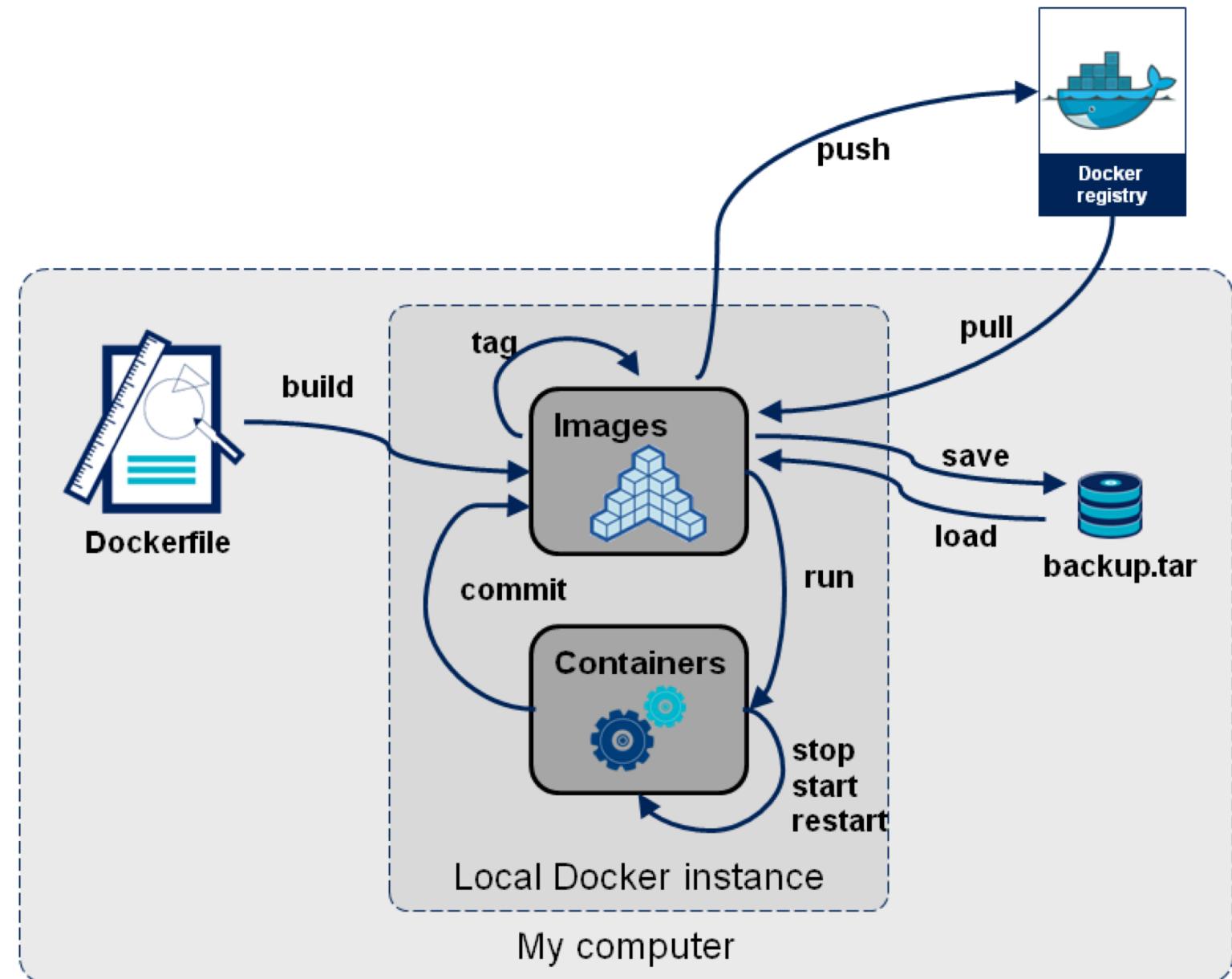
POR QUÊ USAR?

- **Compartilhamento** - você pode usar um repositório remoto ou local para compartilhar suas imagens.
- **Manutenção simplificada** - Docker reduz o esforço e o risco de problemas com dependências de aplicativos.

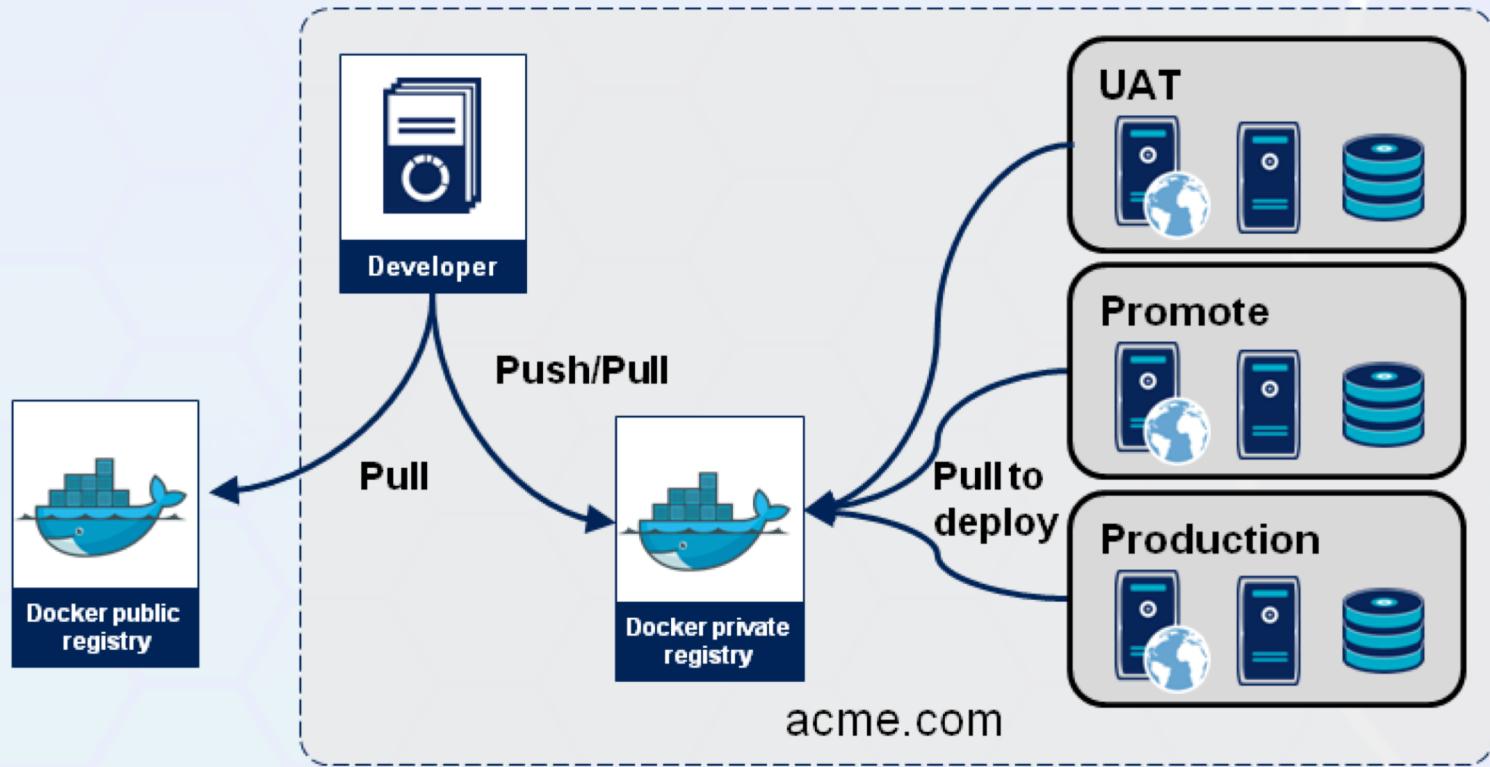
Registro



Registro



Registro



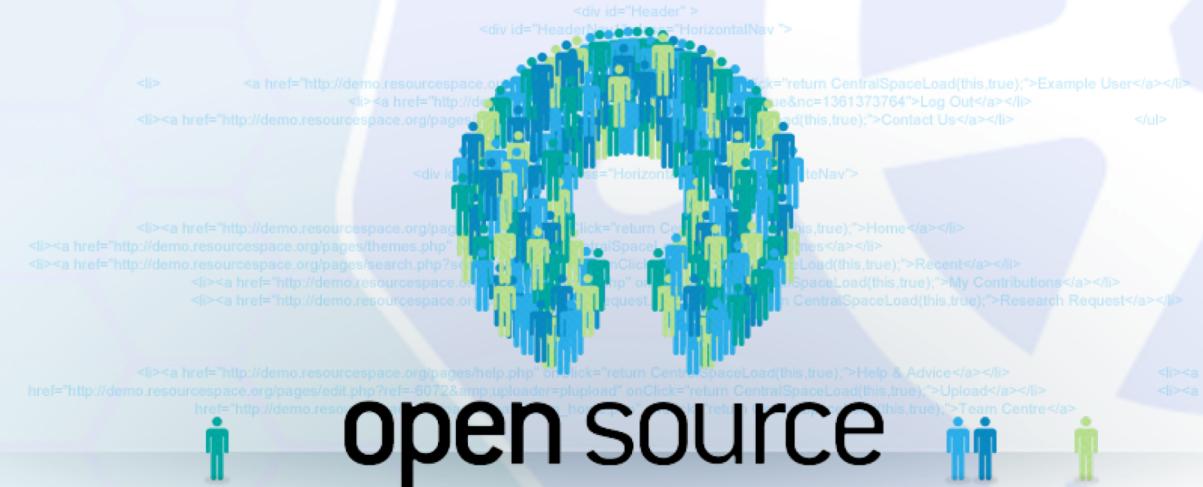
<http://registry.rancher.signallink.us>

Rancher



Rancher

O Rancher é um software de código aberto, que contém um conjunto de softwares necessários para o gerenciamento e orquestração de containers via web.



Rancher

O Rancher aceita tanto hosts Linux de nuvem pública, nuvem privada, servidores físicos ou máquinas virtuais, o Rancher não faz distinção em seu uso, implementando uma camada de serviços de infra-estrutura projetados especificamente para facilitar o gerenciamento de aplicações em contêineres.

Os serviços de infraestrutura do Rancher incluem **rede, armazenamento, balanceador de carga, DNS, service discovery, cronjobs, etc.**

Rancher - Construído em Kubernetes



User Experience

User Interface • App Catalog • Docker run • kubectl • API • CLI

Multi-cluster Management

Provisioning • Upgrades • RBAC • Monitoring • Health Checks • Backup

Kubernetes Cluster



Kubernetes Cluster



GKE Cluster



Imported Clusters

GKE • ACS •
Build Your Own

Multi-cloud Management



**amazon
web services**



**Microsoft
Azure**



**Bare
metal**



**Google
Cloud Platform**



**vmware
vSphere**



DigitalOcean

Rancher - Construído em Kubernetes

The screenshot shows the Rancher web interface. At the top, there is a dark header bar with the Rancher logo, navigation links for Containers, Hosts, Apps, Resources (with a dropdown), Admin (with a red notification dot), and account settings. Below the header is a toolbar with tabs for Containers, Balancers, DNS, and Volumes, along with icons for creating a new container, importing from YAML, and an "Add Container" button.

The main area features a large, stylized illustration of a house with a blue roof and white walls, surrounded by a fence, set against a light gray background. Below this illustration is a blue "Add Container" button with a plus sign.

At the bottom of the interface, there is a footer bar with links for v2.0.0-alpha10, Help & Docs, File an Issue, Forums, Slack, English (with a dropdown), and Download CLI (with a dropdown).

Rancher - Construído em Kubernetes

The screenshot shows the Rancher web interface for managing a Kubernetes cluster named "mykubernetescluster".

Header: mykubernetescluster, Cluster (selected), Nodes, Storage, Projects/Namespaces, Members, Tools, Launch kubectl, Kubeconfig File, More options.

Dashboard Information: Provider Custom, Version v1.10.1, Nodes 1.

Metric Dials:

- CPU:** 54% (0.5 of 1 used)
- Memory:** 4% (0.1 of 3.8 GiB used)
- Pods:** 6% (7 of 110 used)

Status Indicators: etcd (green checkmark), Controller Manager (green checkmark), Scheduler (green checkmark), Nodes (green checkmark).

Footer: v2.0.0, Help & Docs, Forums, Slack, File an Issue, English, Download CLI.

Kubernetes



Kubernetes

Kubernetes é uma plataforma de open source de gerenciamento de containers de nível enterprise, baseado em 15 anos de experiência do Google e pronto para o mundo multi-cloud: nuvem pública, nuvem privada ou híbrida. Possui desenho modular e pode ser rodado em quase qualquer lugar.



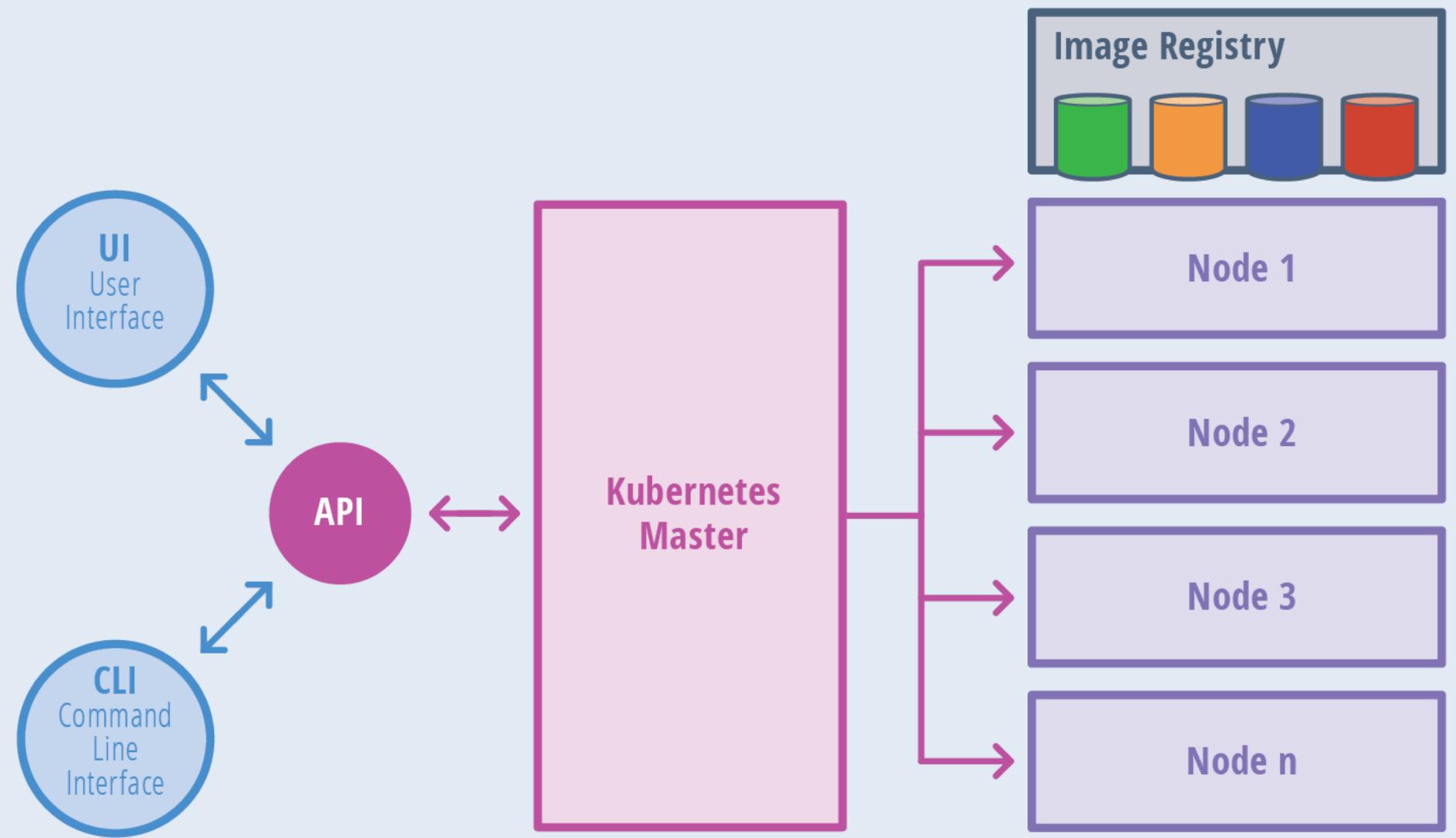
Kubernetes - Arquitetura

O Kuberentes provê uma arquitetura flexível, e com mecanismo desacoplado de service discovery. Como a maioria de plataformas computacionais distribuídas, um cluster consiste:

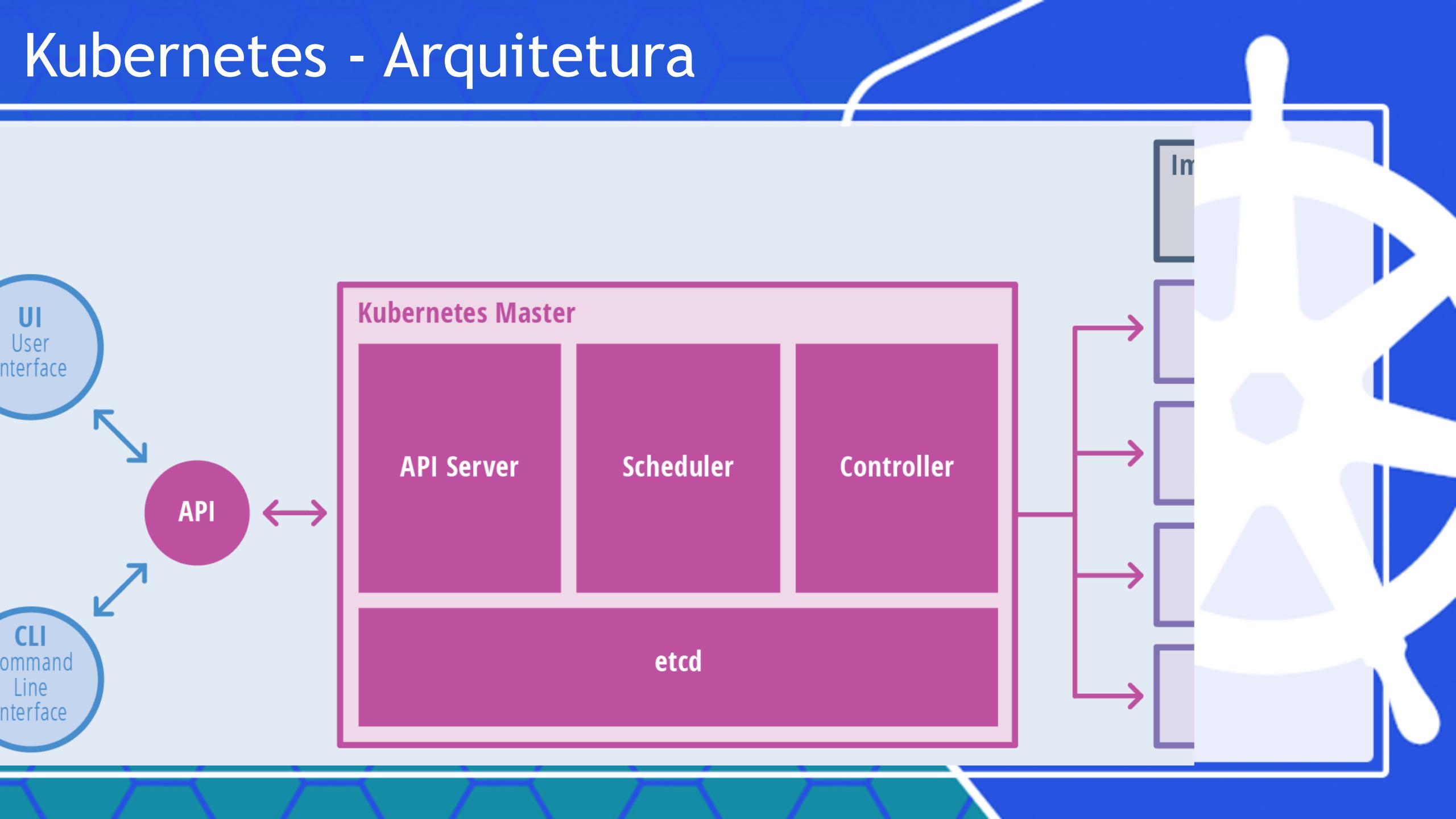
- Pelo menos um master e múltiplos nodes de computação.

O master é responsável pela API, agendamento dos deployments e gerenciamento total do cluster.

Kubernetes - Arquitetura



Kubernetes - Arquitetura



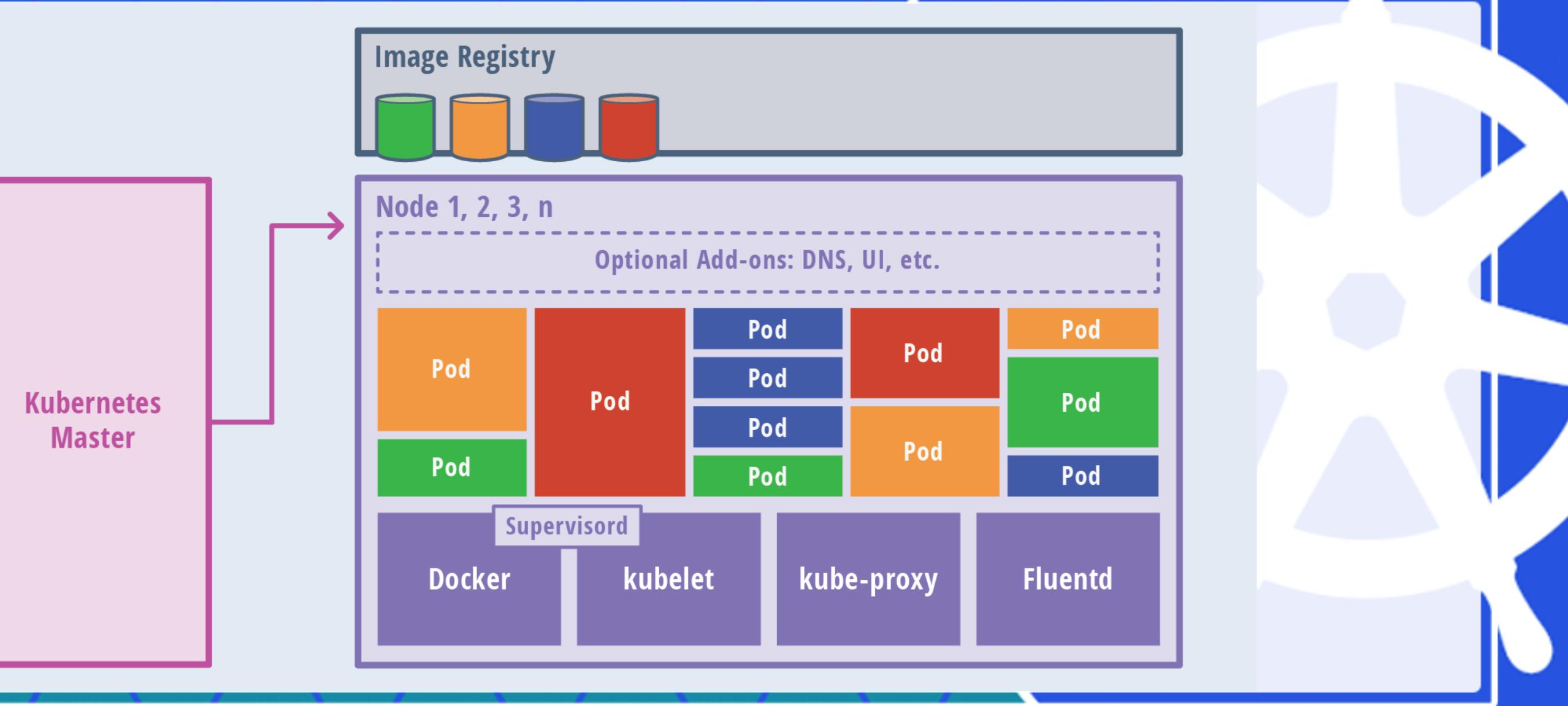
Kubernetes - Arquitetura

Cada nó do cluster roda:

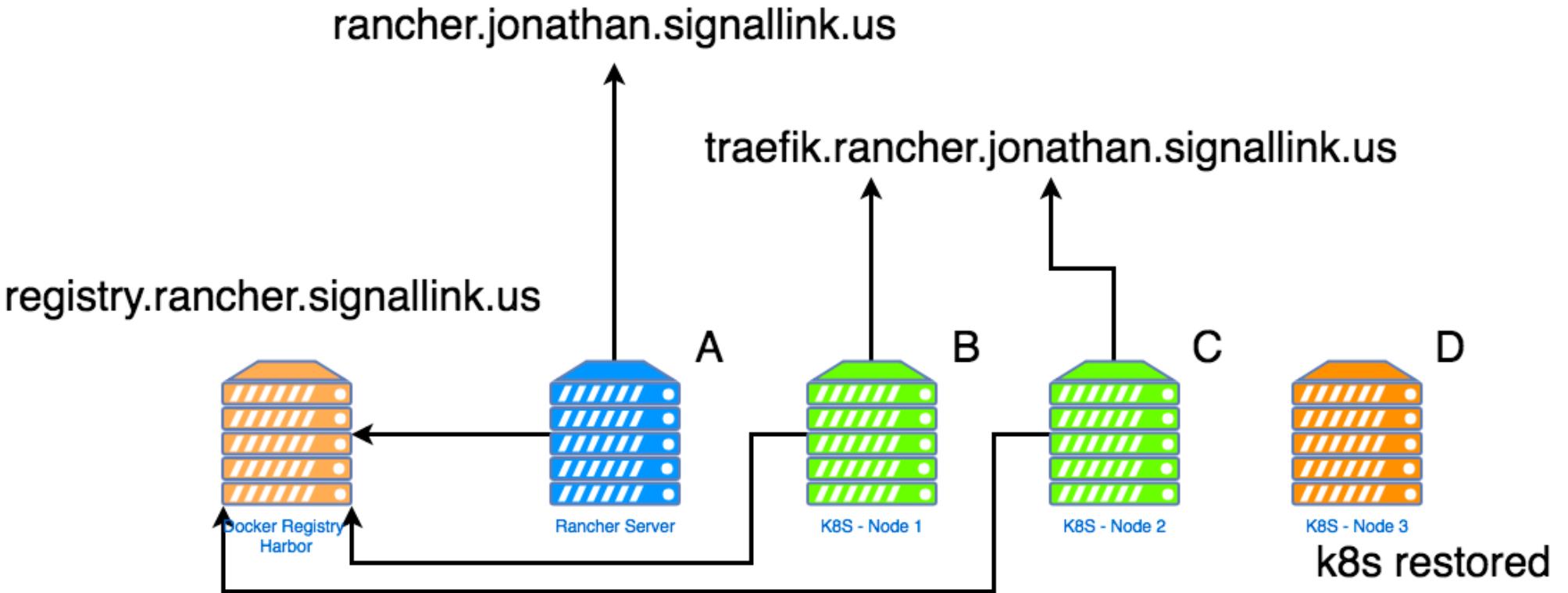
- Container runtime, Docker ou Rocket, com os agentes para se comunicar com o master. (Kubelet, Kube-proxy)
- Componentes adicionais para logs, monitoramento, service discovery e add-ons opcionais.

Os nós, são os trabalhadores do Cluster. Eles expõem os serviços de computação, rede e armazenamento para as aplicações.

Kubernetes - Arquitetura



Kubernetes - Arquitetura do treinamento



Para cada usuário:
rancher.jonathan.signallink.us
.rancher.jonathan.signallink.us

Kubernetes - Componentes

Componentes do Kuberentes para usarmos em nossas aplicações containerizadas:

Pods: Containers que trabalham em conjunto.

Services: Pods que trabalham em conjunto

Deployments: Provê um única declaração para Pods e ReplicaSets.

Labels: Usado para organizar serviços

Kubernetes - Componentes

Daemonsets: Rodar sempre um ou mais pods por nó.

Secrets: Salvar dados sensativos como senhas de bancos de dados.

ConfigMaps: Arquivo de configurações que suas aplicações irão usar.

Cron Jobs: Executar tarefas temporais, uma vez ou repetidas vezes.

Kubernetes - O que podemos fazer

- Deployment de containers e controle de Rollout.

Descreva seus containers e quantos você quer com o "Deployment". Kuberentes irá manter esses containers rodando e irá gerenciar mudanças no deployment (como atualizar imagens ou mudar variáveis de ambiente) com um "rollout". Você pode pausar, resumir, e fazer rollback das alterações como você preferir.

Kubernetes - O que podemos fazer

- Empacotamento de recursos

Você pode declarar um mínimo e máximo de recursos computacionais (CPU & Memória) para seus containers. Kuberentes irá alocar seus containers onde eles puderem caber. Isto incrementa sua eficiência computacional e ultimamente ajuda a baixar seus custos.

Kubernetes - O que podemos fazer

- Service Discovery e Autoscaling.

Kubernetes pode automaticamente expor seus containers para a internet e outros containers no cluster. Ele automaticamente balanceia o tráfego através dos containers. Kubernetes suporta service discovery através de variáveis de ambiente e DNS. Você pode também configurar autoscaling baseado em CPU, Memória e rede para os containers aumentarem seus recursos de utilização.

Kubernetes - O que podemos fazer

- Cluster Heterogêneo

Kubernetes roda em qualquer lugar. Você pode construir seu cluster para um conjunto de VM's rodando na nuvem, on-premises, ou bare metal dentro do seu data center.

Kubernetes - O que podemos fazer

- Armazenamento persistente

Kubernetes inclui suporte para armazenamento persistente conectado as aplicações stateless. Existe suporte para AWS EBS, GCP, e muitos, muitos, muitos outros.

Kubernetes - O que podemos fazer

- HA

Kubernetes é escalável para o planeta Terra. Isto requer atenção especial para funcionalidades de HA como multi-master ou cluster federation. Cluster Federation permite conectar clusters em conjunto, para permitir que se um cluster caia, os containers podem automaticamente serem movidos para o outro cluster.

Kubernetes - O que podemos fazer

Estes itens fazem o Kubernetes perfeito para rodar diferentes arquiteturas de aplicações, desde as monolíticas aplicações web, até aplicações de microserviços altamente distribuídos, e até mesmo aplicações batch.

Kubernetes - Soluções

Alguns dos problemas que você irá enfrentar quando rodar aplicações containerizadas em um ambiente clusterizado, e como o Kubernetes resolve elas.



Kubernetes - Soluções

Scheduling

Você construiu sua grande aplicação baseada em containers? Seeeeensacional!! Agora você precisa se certificar que ela irá rodar onde e quando ela deve. É importante para sua aplicação rodar nas máquinas corretas dentro do seu cluster.

Kubernetes - Soluções

Load balancing

Sua aplicação está up e rodando. Perfeito! agora você precisa se certificar que o tráfego do cliente é distribuído até mesmo através dos nós do seu cluster. É importante que sua aplicação esteja fazendo um bom uso dos recursos em cada host para aguentar a carga dos clientes. Você não quer que alguns containers trabalhem em carga máxima enquanto outros estão sem trabalhar.

Kubernetes - Soluções

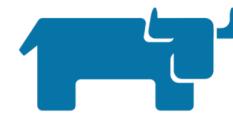
Application scaling

Você tem seus containers rodando e o load balancer está distribuindo a carga muito bem entre eles. Muito, muito bom!! Agora você precisa trazer mais containers online para aguentar o tráfego crescente, e mesmo picos de demanda, e depois terminar os containers quando não são mais necessários. É importante poder suportar picos nas requisições do seu serviço.

Kubernetes - Soluções

Cluster management and monitoring

Agora que você tem sua aplicação rodando eficientemente em um cluster gigantesco, você tem que gerenciar ela. Você precisa definir, lançar, escalar, balancear o tráfego, e monitorar a saúde dos containers que estão rodando. Não é uma tarefa fácil.



RANCHER



kubernetes

by Google

Obrigado!!!

jonathanbaraldi@gmail.com

slideshare.net/jonathanbaraldi

Obrigado!!!