



Introdução a Kubernetes e Rancher

A stylized illustration of a city skyline with various buildings, including skyscrapers and industrial structures, set against a light gray background. In the foreground, there are green hills and small trees.

Flávio Gussi
Sales Engineer

Gabriel Lins
Sales Engineer



FLÁVIO GUSSI
Engenheiro de Vendas



linkedin.com/in/flaviogussi



fgussi



Flávio Gussi



flaviogussi



GABRIEL LINS
Engenheiro de Vendas



linkedin.com/in/gbrlins



glins



Gabriel Lins



gbrlins

Concorra a um kit de brindes da SUSE!

1. Faça uma publicação no **LinkedIn** usando a hashtag **#susebrk8s**
2. Inscreva-se no canal do **Youtube**



1º lugar: adesivos + caderno + caneca + camiseta + caneta

2º lugar: adesivos + caderno + caneta

3º lugar: adesivos + caderno + caneta

*Imagens meramente ilustrativas,
os produtos podem ser diferentes!

Façam perguntas no chat!
Responderemos na medida do possível

Agenda

01 Introdução

02 Componentes Kubernetes

03 Introdução a CLI

04 Introdução e instalação K3s

05 Perguntas e respostas

06 SORTEIO!!

Fique ligado!

O repositório dessa aula está em:

<https://github.com/SUSEBrSalesEngineer/susebrk8s>



Materiais: Docs

<https://rancher.com/docs/>

The screenshot shows a web browser window displaying the Rancher documentation site at <https://rancher.com/docs/rancher/v2.x/en/quick-start-guide/deployment/quickstart-manual-setup/>. The page title is "Manual Quick Start". The header includes the Rancher logo, navigation links for "DOCS", "REQUEST A DEMO", "PRICING", "SUPPORT", "CONTACT", and a search bar. Below the header is a main menu with "GET STARTED" highlighted. On the left, a sidebar titled "RANCHER 2.X" lists several sections: "Overview", "Quick Start Guides" (which is expanded to show "Deploying Rancher Server", "Amazon AWS Quick Start", "DigitalOcean Quick Start", "Vagrant Quick Start", and "Manual Quick Start" which is also expanded to show "Deploying Workloads", "Installation", "Upgrades and Rollbacks", and "Backups and Disaster"). The main content area starts with a heading "Manual Quick Start" followed by a sub-heading "Howdy Partner! This tutorial walks you through:" and a bulleted list: "• Installation of Rancher v2.0", "• Creation of your first cluster", and "• Deployment of an application, Nginx". Below this is a section titled "Quick Start Outline" with the sub-heading "This Quick Start Guide is divided into different tasks for easier consumption." and a numbered list: "1. Provision a Linux Host", "2. Install Rancher", "3. Log In", and "4. Create the Cluster".

Materiais: Faça parte da nossa comunidade

slack.rancher.io/

A screenshot of the Rancher Slack interface. The left sidebar shows channels like #general, #onlinetraining, and #general. The main window shows a conversation in the #general channel. A QR code is overlaid on the bottom right of the screenshot.

community.suse.com/

A screenshot of the SUSE & Rancher Community website. The left sidebar has links to Home, Discovery, Members, and About. The main area shows a welcome message and various community sections like Topics, Events, and Groups. A QR code is overlaid on the bottom right of the screenshot.

Materiais: Faça parte da nossa comunidade

Cowmeleon

Canal para a comunidade no YouTube

The screenshot shows the YouTube channel page for 'Cowmeleon'. The channel has 362 subscribers. The navigation bar includes links for INÍCIO, VÍDEOS, PLAYLISTS, CANAIS, DISCUSSÃO, and SOBRE. Below the navigation, there is a section titled 'Envios' with a link to 'REPRODUZIR TODOS'. Four video thumbnails are displayed:

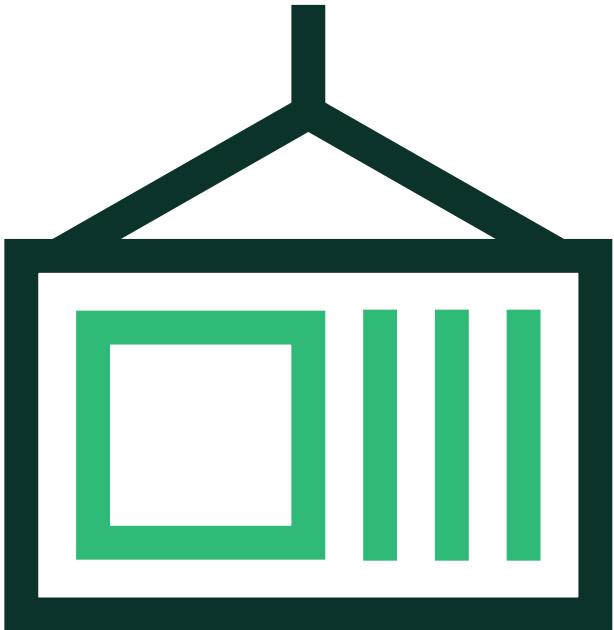
- Como configurar openSUSE Para Jogos** (11:13) - 5 Configurações no OpenSUSE Para Rodar... 79 visualizações • há 2 dias
- Quais as novidades no Rancher 2.6 ?** (24:57) - Quais as novidades no Rancher 2.6 ? 93 visualizações • há 1 semana
- Integrando o Harbor como registry privada no Rancher** (18:43) - Integrando o Harbor como registry privada no Rancher 68 visualizações • há 2 semanas
- K8S SEGUR** (1:00) - SEUS CLUSTERS KUBERNETES ESTÃO... 132 visualizações • há 3 semanas



Porquê containers?

Containers

Porquê usar containers?



Imagens em container sobem mais rápido

Containers podem rodar em qualquer lugar

Containers fazem melhor uso dos recursos

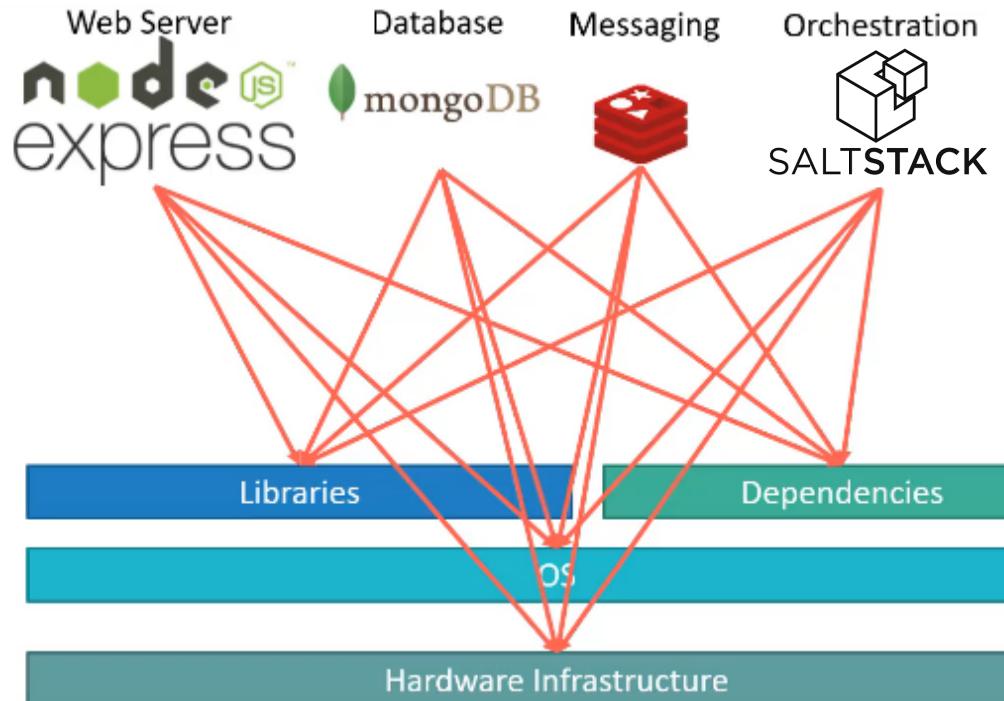
Resolução de problemas em compatibilidade

São mais leves

As imagens são menores

Containers

Porquê usar containers?



- Compatibilidade/Dependência
- Muito tempo para configurar
- Diferentes ambientes (Dev/Test/Prod)

Gerenciar containers

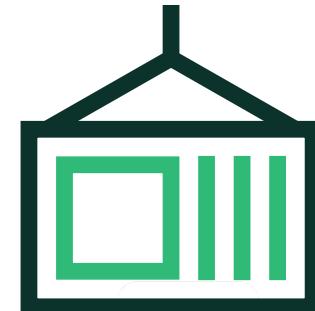
E agora?



Produção
1000 containers

Teste
100 containers

Desenvolvimento
50 containers



- Deploy de imagens e inicialização de containers;
- Integração e orquestrar microserviços;
- Escalar containers e cluster baseado em demanda;
- Balanceamento de carga entre containers;

O que é Kubernetes?

Kubernetes

O que é Kubernetes?



kubernetes

Orquestração de container com K8s

Plataforma
open-source

Construído
para gerenciar
containers

Empresas utilizam Kubernetes



NIANTIC



O que é necessário para liberar o potencial do Kubernetes?

DevOps
Automation

Secure Application Deployment

Shared Tools & Services

Centralized
Management

Security, Policy and User Management

Consistent Cluster Operations

Certified
Kubernetes
Distributions

Run Certified Kubernetes from Dev to Edge



Datacenter



Cloud



Dev



Branch



Edge

O que é necessário para liberar o potencial do Kubernetes?



EPINIO

Secure Application Deployment

Load Balancing

Routing

Metrics

Autoscaling

Canary

Git Deployments

Shared Tooling & Services



Prometheus



Security & Authentication



Active Directory



GitHub



SAML



Ping Identity



okta

Policy management

Configuration enforcement

RBAC policies

CIS benchmark monitoring

Pod & network security policies

Simplified Cluster Operations & Infrastructure Management

Kubernetes version management

Visibility & diagnostics

Monitoring & alerting

Centralized audit

Node pool management

Cluster provisioning



RKE



Datacenter



Amazon EKS



Azure AKS



Google GKE



Cloud



K3S



Dev



Branch



Edge

Estrutura do K8s



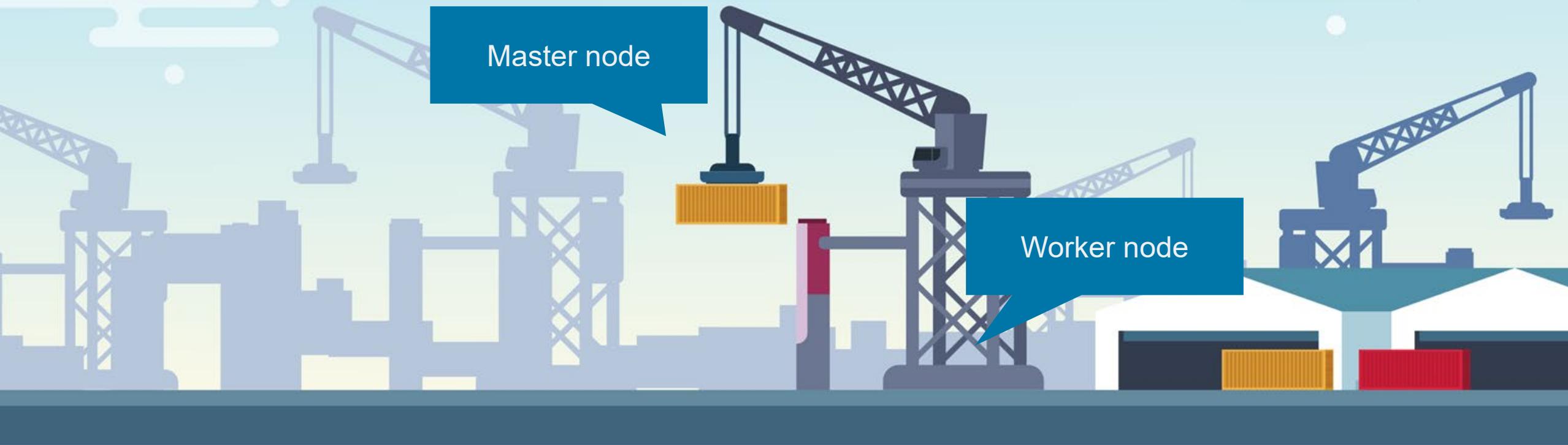
CONCEITOS BÁSICOS



Conceitos básicos

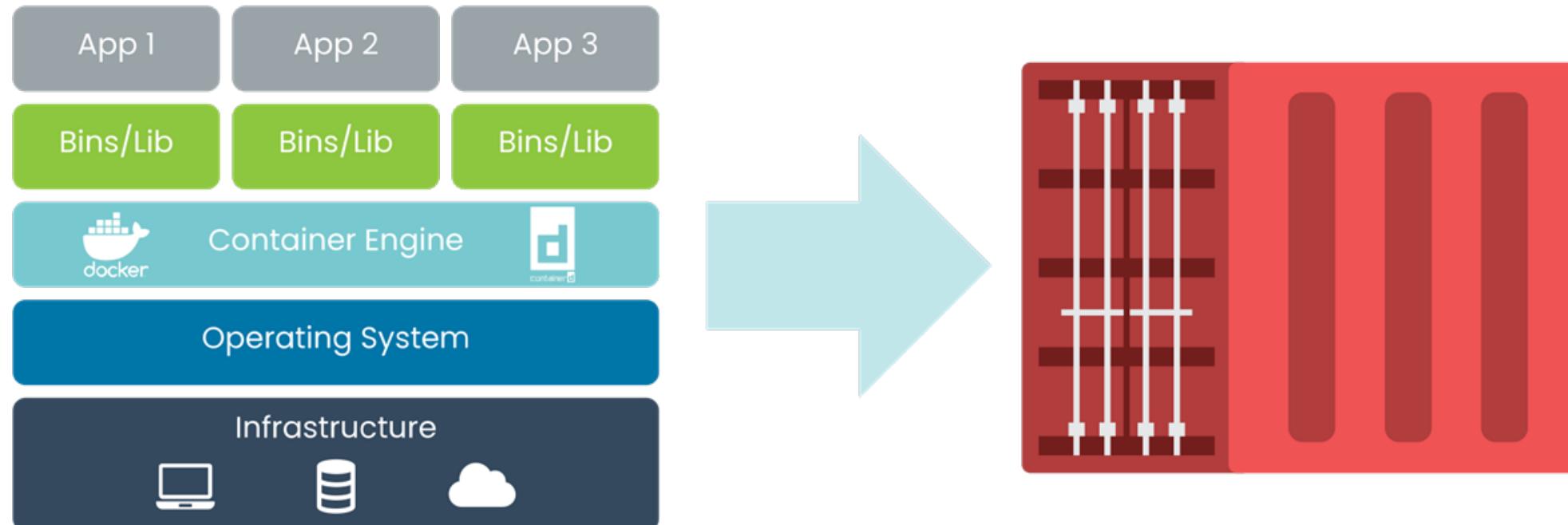


Analogia do navio de carga



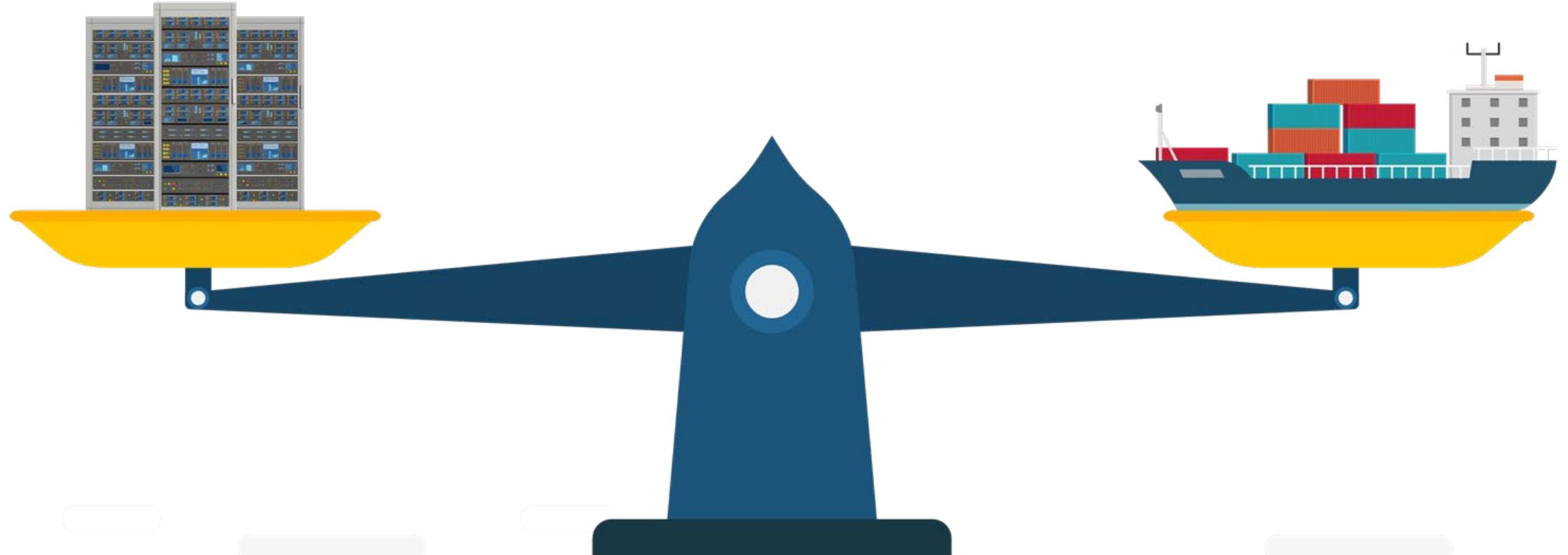
Container Docker

Container Kubernetes = carga



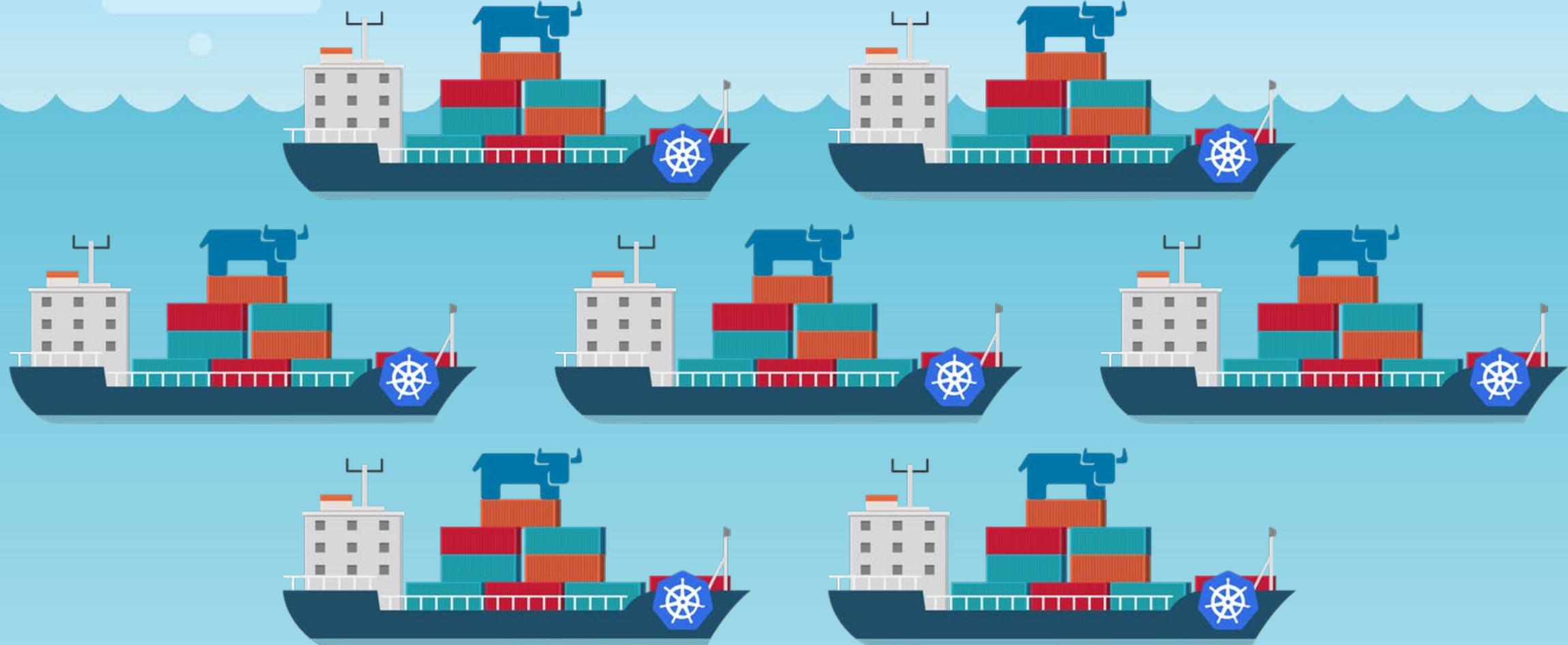
Arquitetura K8s

Nó Worker = Navio de carga



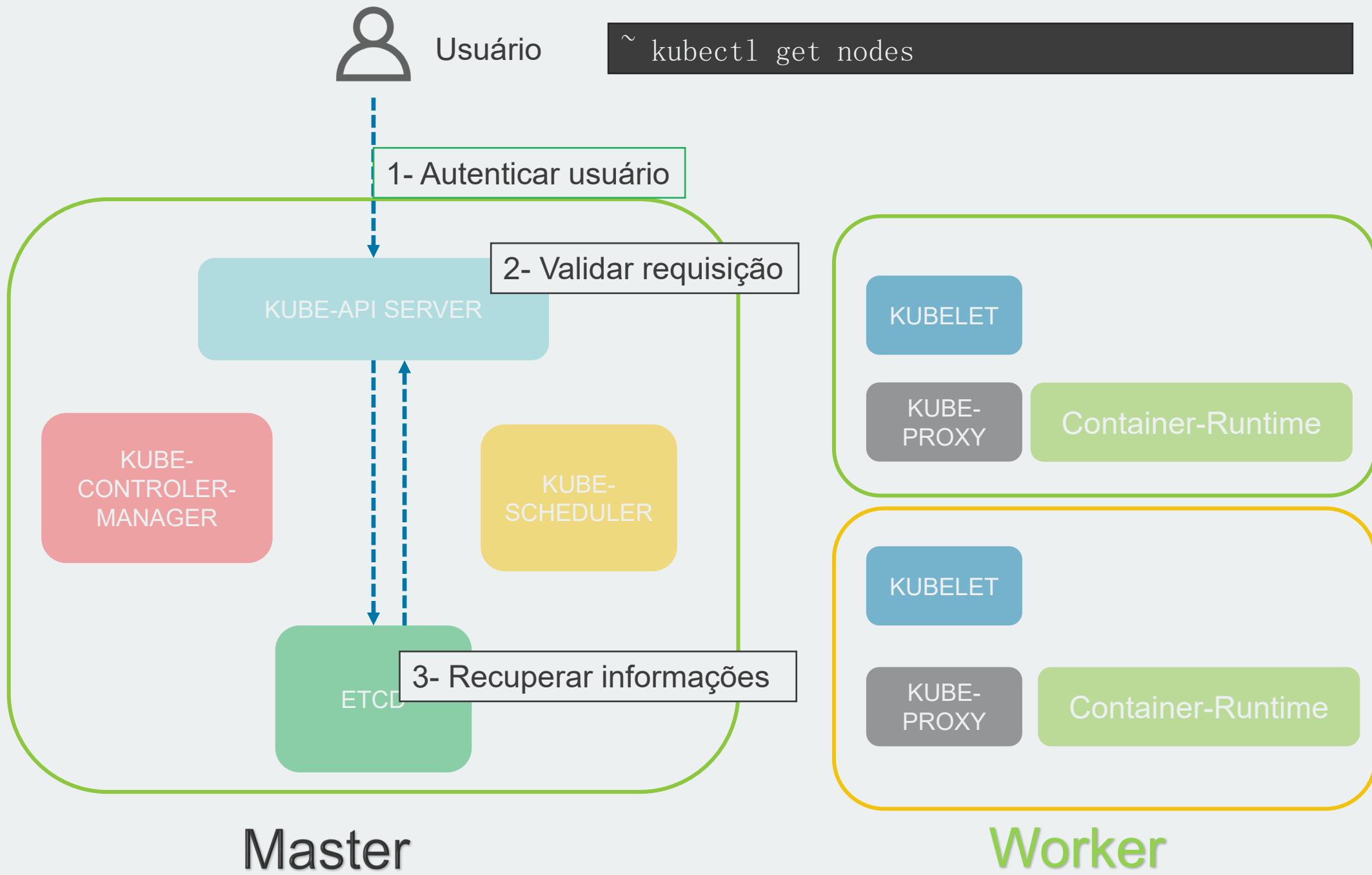
Arquitetura K8s

Frota = Cluster Kubernetes

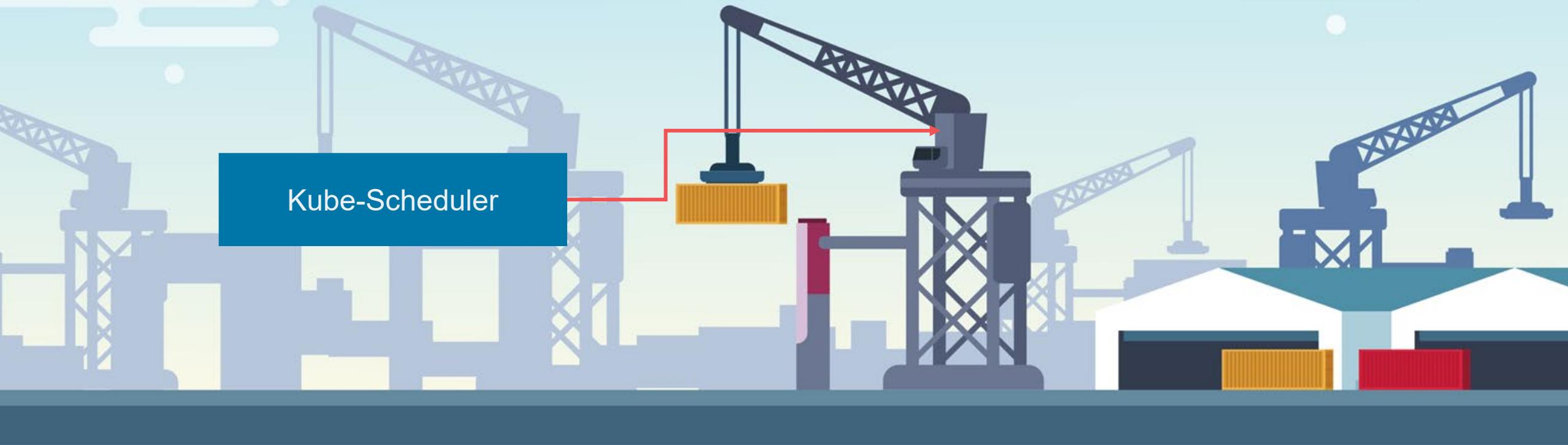


Componentes do Master node

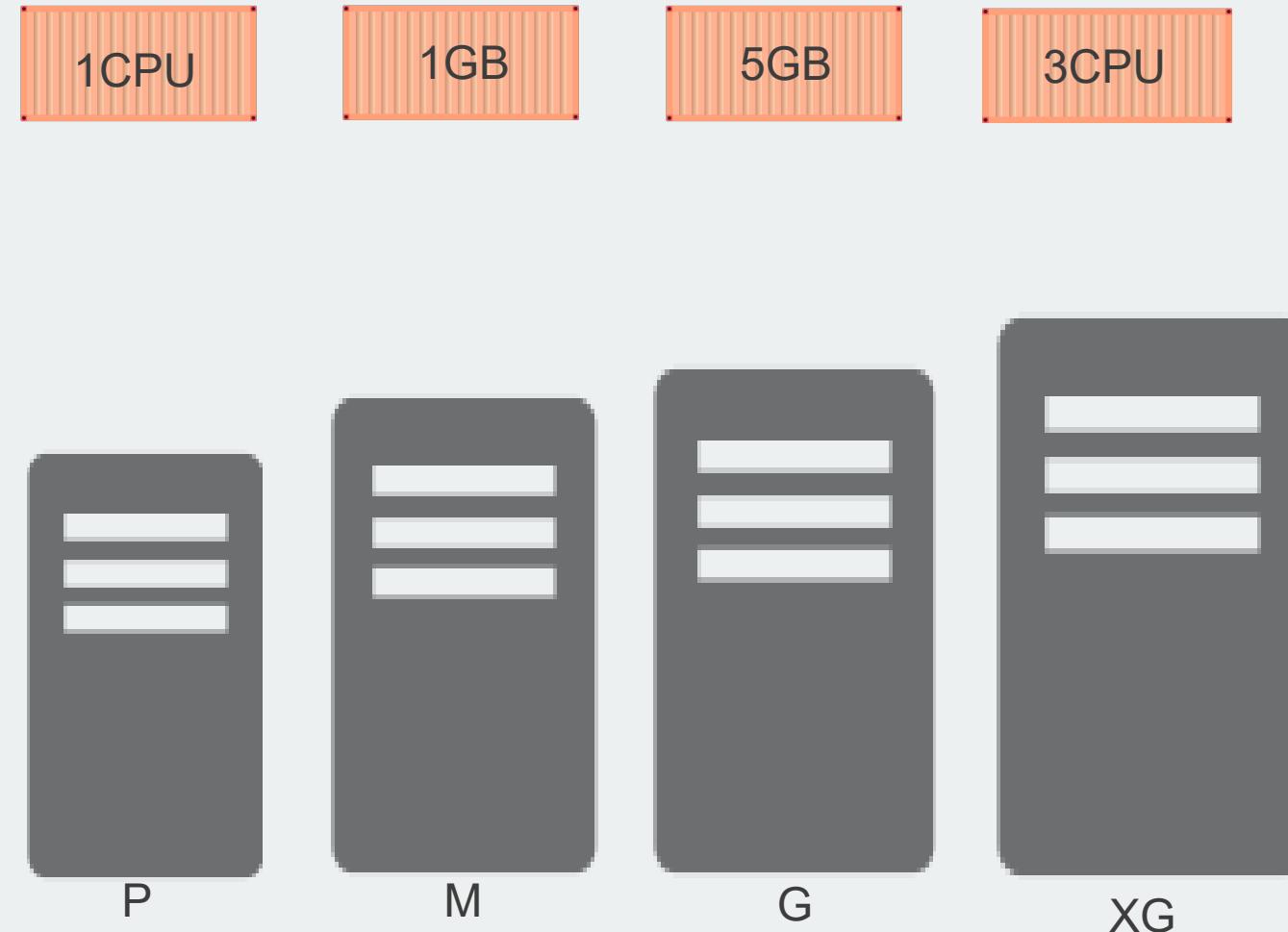




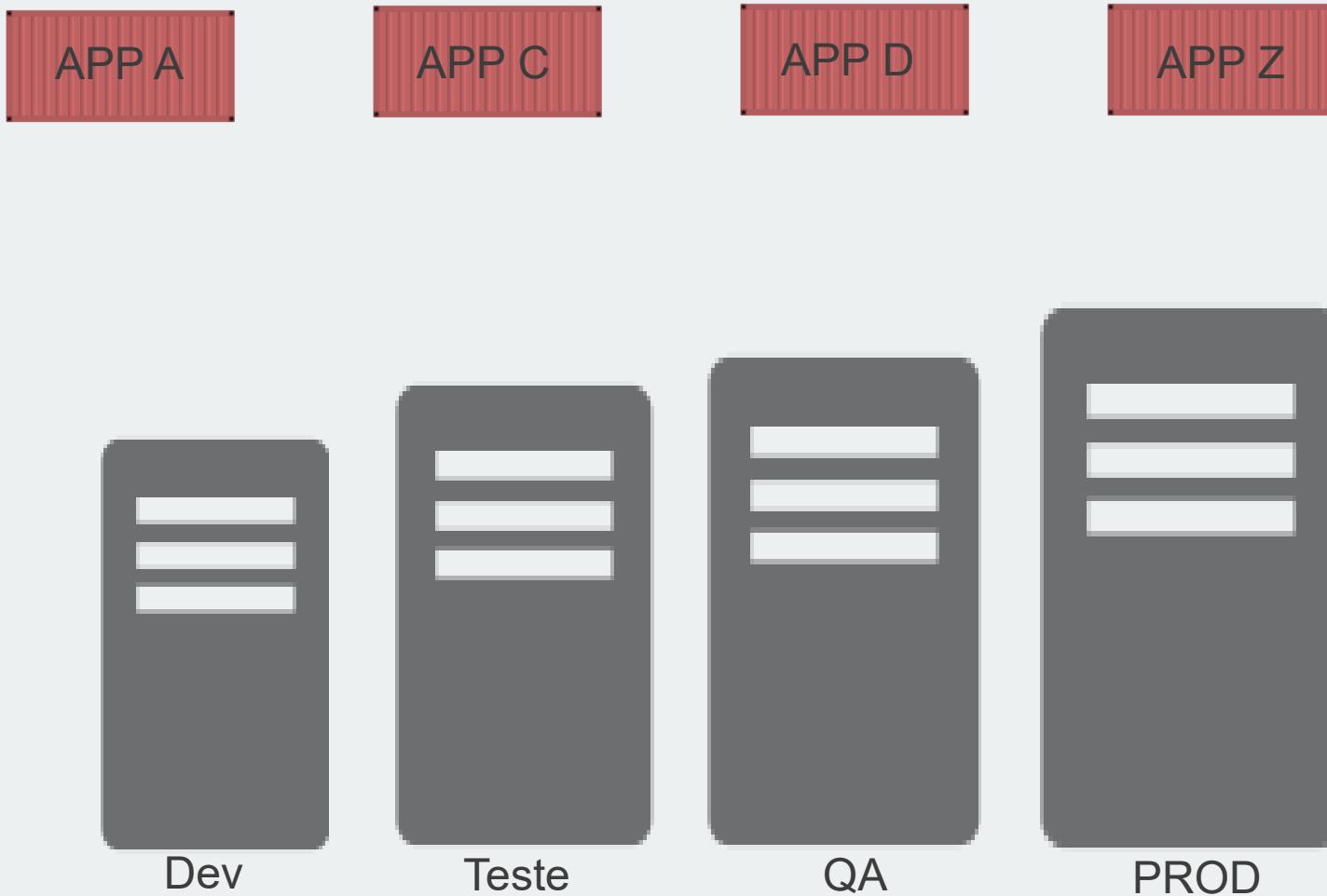
Componentes do Master node



Ex: Direcionar carga apropiada para os nodes



Ex: Direcionar carga aos nós de função



Componentes do Master node



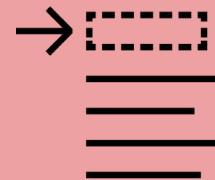
Kube controller manager



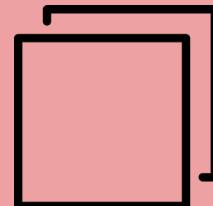
Node controller



Deployment controller



Namespace controller



Replication controller



Endpoint controller



Service account
manager

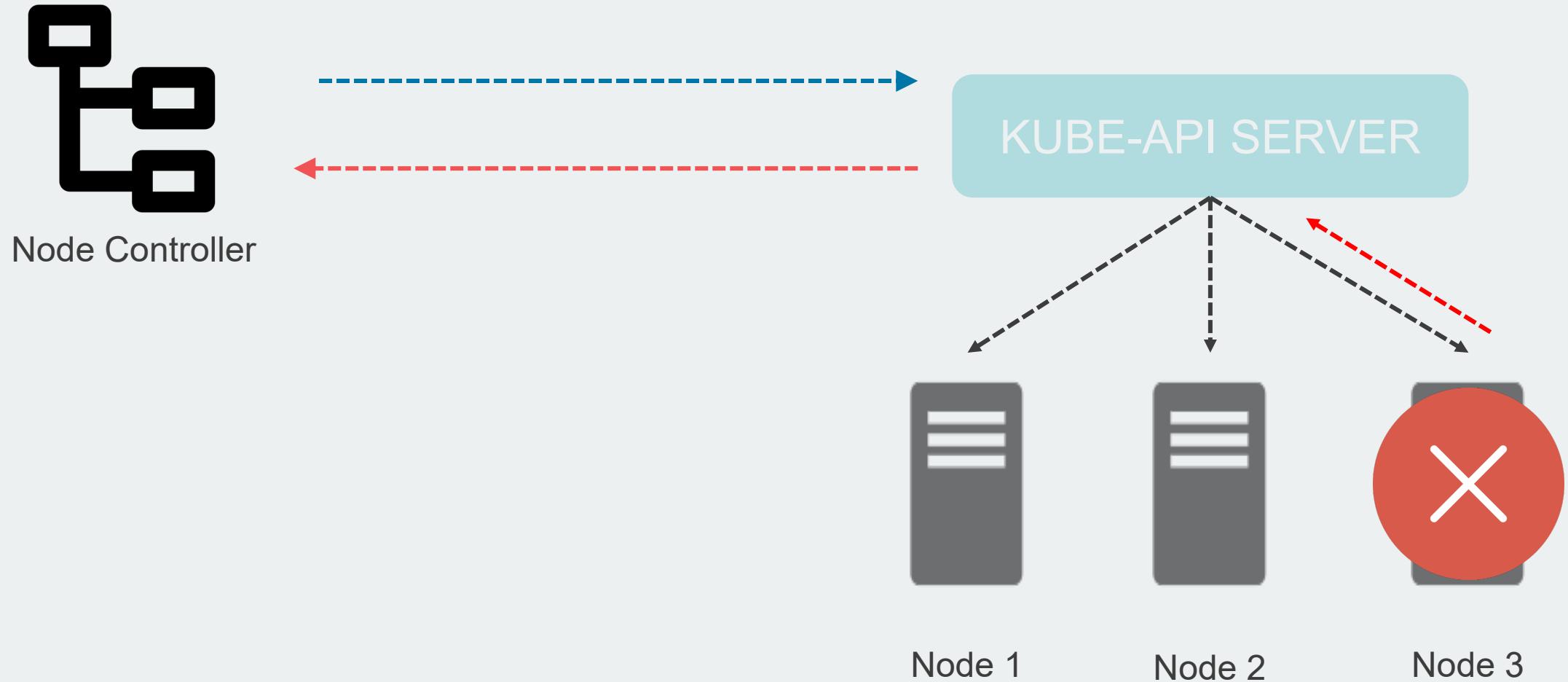


Job Controller



Cron-Job

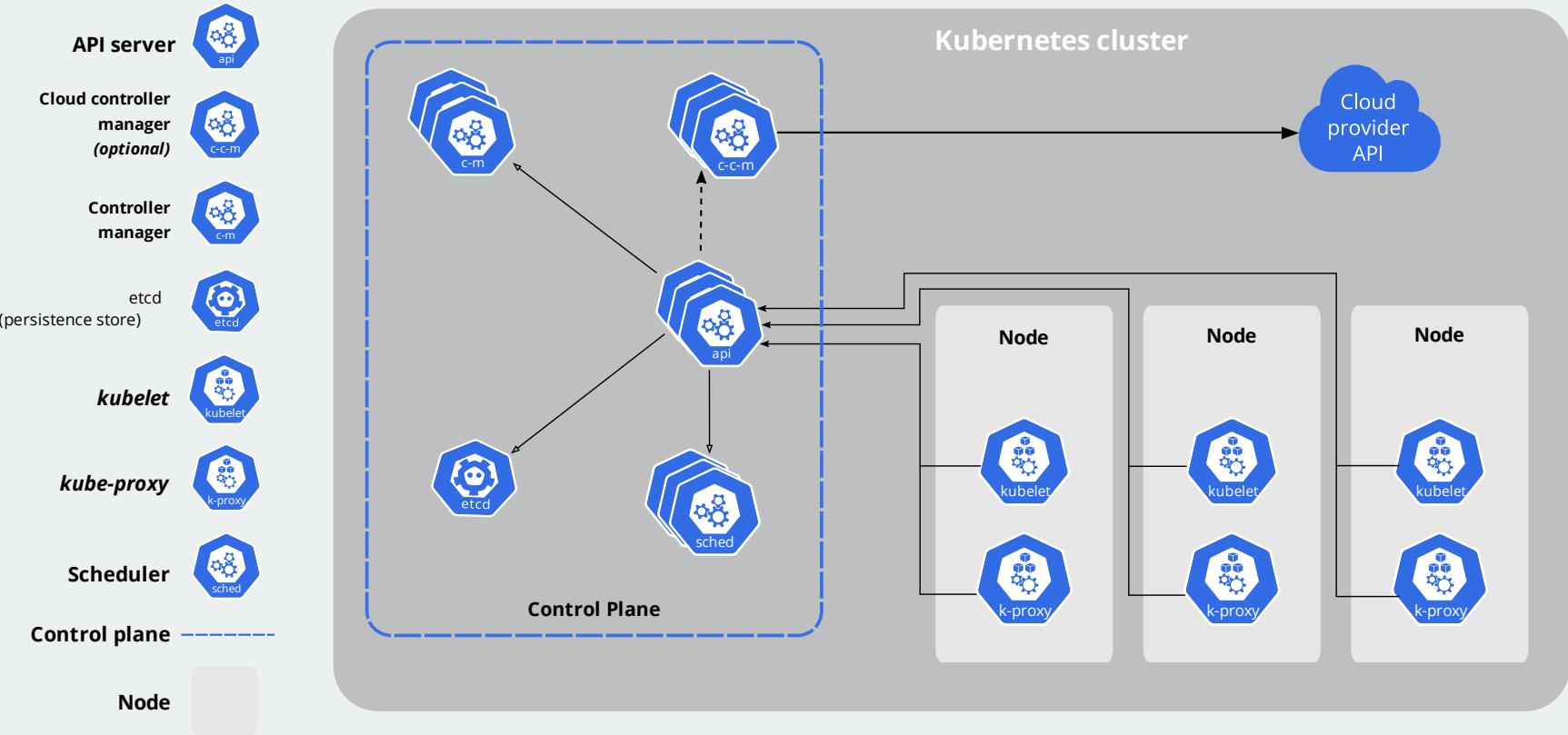
Ex: Node Controller



Componentes do Worker node



Estrutura do Kubernetes



KEY	VALUE
Name	Edson
Age	20
Address	Sergipe

KEY	VALUE
Name	Vinicius
Age	32
Address	Taguatinga

KEY	VALUE
Name	Manolo
Age	50
Address	São Paulo

Vamos falar sobre K3D

- Um jeito de rodar um cluster K3s em Docker
- Pensado para desenvolvimento
- Para fácil estudo em Kubernetes

<https://k3d.io/>

Faça aí você mesmo!

(mas se não quiser tudo bem)

* Deve ser feito em sistemas linux.

- **Instalar docker** ex: zypper in docker
- **Inicie o docker** ex: systemctl start docker
- **Instale o k3d:** curl -s https://raw.githubusercontent.com/rancher/k3d/main/install.sh | bash
- **Crie seu cluster:** k3d cluster create *mycluster*
- **PRONTO!**

Vamos criar alguns clusters!

<https://www.katacoda.com/courses/kubernetes/launch-single-node-cluster>

OU

Learn With K8s Labs

<https://labs.play-with-k8s.com>

Repositório em:

<https://github.com/SUSEBrSalesEngineer/k8s-intro-training>

Hora do café!

Voltaremos em breve



Vamos falar sobre o K3s

K3s ≠ K3D !!!

- Distribuição super leve mantida pela **CNCF**
- Diferente das soluções para **não produção** como minikube,microk8s,kind
- Pequeno binário que utiliza menos de 512Mb de RAM para **TODO** o cluster.
- Fácil instalação utilizando **K3Sup**

<https://k3s.io>

<https://k3sup.dev>

Introdução à arquivos YAML

Todos **objetos no K8s**, como pods por exemplo, no K8s podem ser criados via arquivos .yaml

Objetos – Entidades persistentes em um cluster. Ex: Pods, Deployments e Services

YAML – YAML Ain't Markup Language é um formato de serialização de dados legíveis.

O Kubernetes sempre vai trabalhar para garantir a existência dos objetos que foram definidos via arquivos YAML

Introdução à arquivos YAML

```
apiVersion: v1
```

```
kind: Pod
```

```
metadata:
```

```
  name: static-web
```

```
  labels:
```

```
    role: myrole
```

```
spec:
```

```
  containers:
```

```
    - name: web
```

```
      image: nginx
```

```
      ports:
```

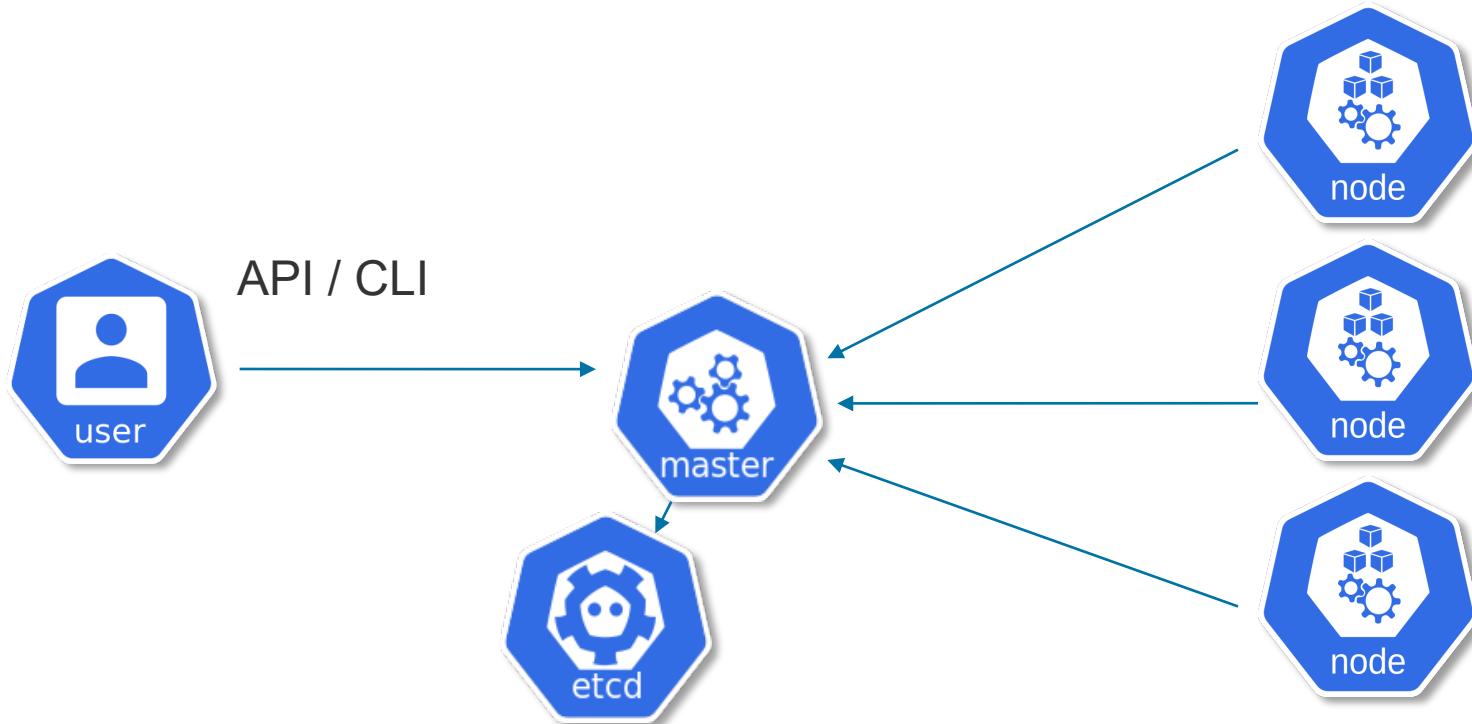
```
        - name: web
```

```
          containerPort: 80
```

Kubernetes

- Kubernetes é uma poderosa plataforma opensource para gerenciar serviços e workloads contêinerizados.
- Kubernetes utiliza sintaxe declarativa para descrever um determinado estado recurso do cluster.

Kubernetes - Arquitetura



- Controlplane (master): Gerencia o cluster e expõe a API para controle.
- Etcd: Guarda todos os dados do estado do cluster kubernetes.
- Worker: Executa os workloads e suporte de componentes.

Objetos no Kubernetes

- ✓ Pods
- ✓ ReplicaSets
- ✓ Deployments
- ✓ Services
- ✓ Ingresses



Pods



Kubernetes - Recursos - Pod



- Menor unidade de um deployment.
 - Similar ao que é uma VM no mundo dos hypervisors.
- Pode conter mais que um container por POD
- Tem seus proprios recursos como:
 - Storage (Ou volumes compartilhados)
 - Rede
- Localizado em um nó de worker.

Pod Spec Básico

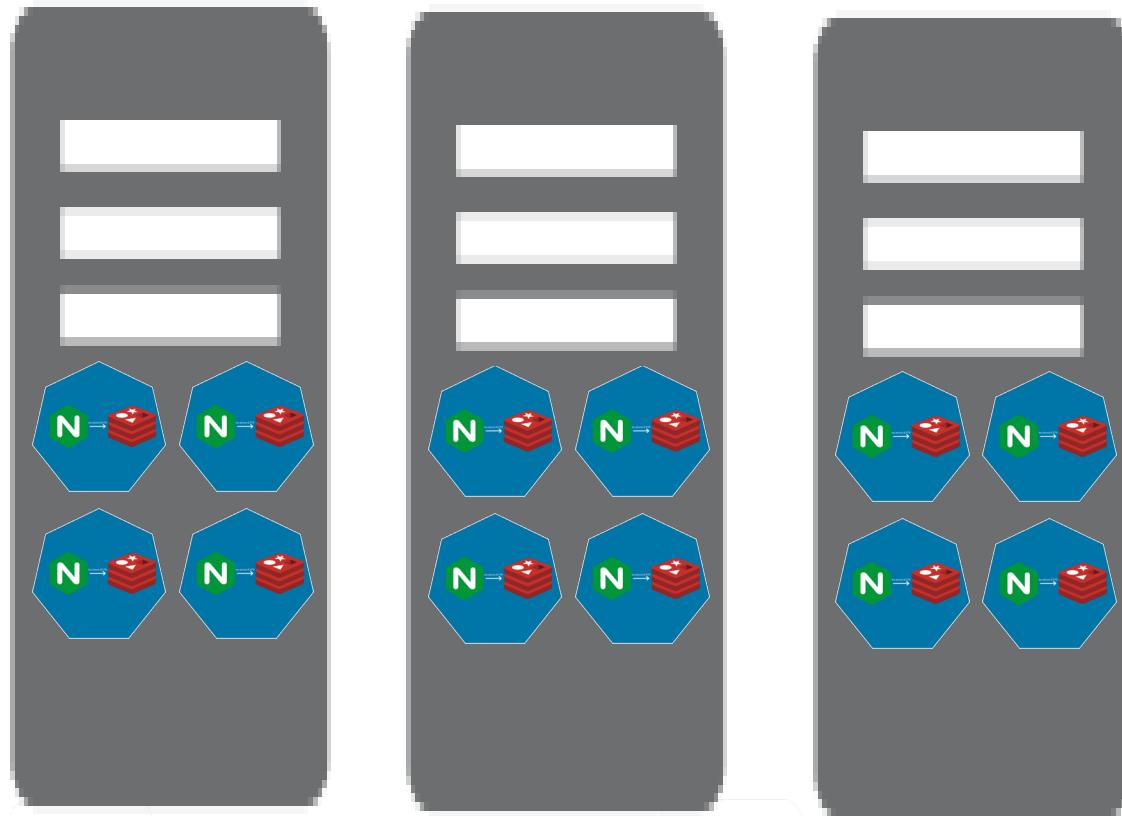
```
apiVersion: v1
kind: Pod
metadata:
  name: myapp-pod
  labels:
    app: myapp
spec:
  containers:
  - name: myapp-container
    image: busybox
    command: ['sh', '-c', 'echo Hello Kubernetes! && sleep 1']
```

ReplicaSets

Kubernetes - Replica Sets

- Define o número desejado para persistir pods.
- Introduz o gerenciamento por estado.

Replica Set



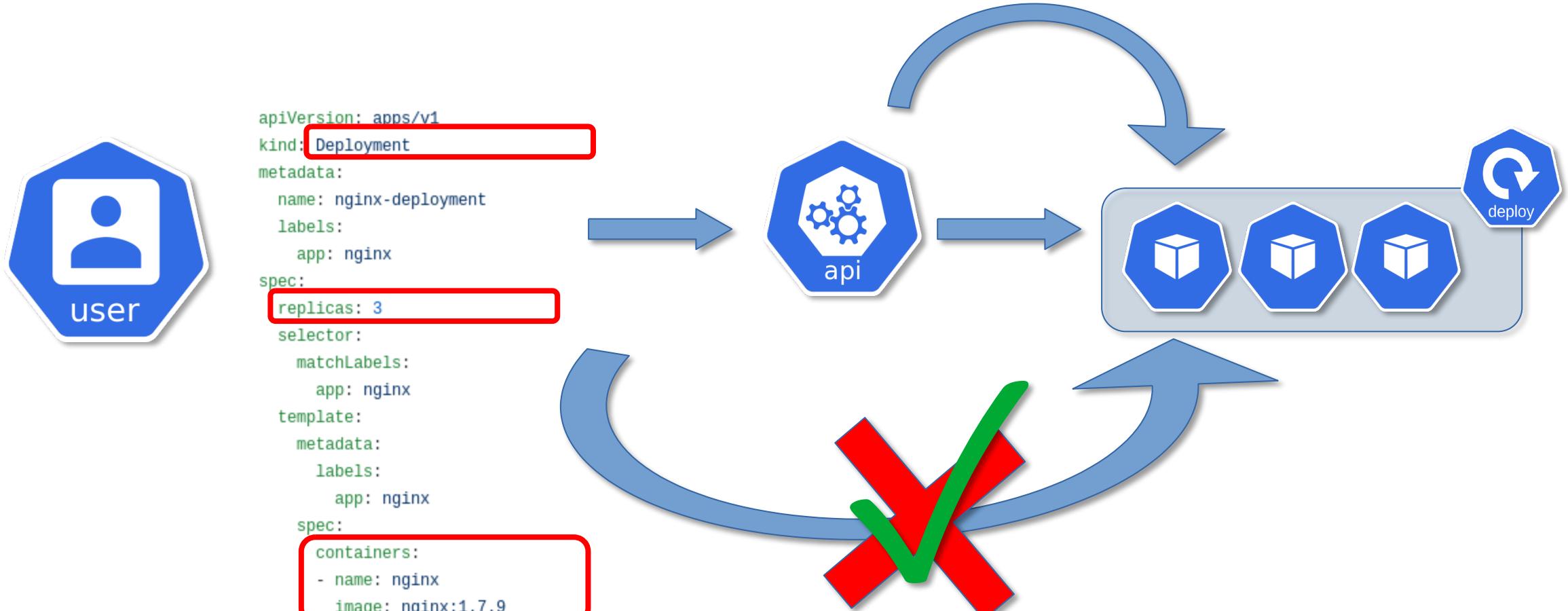
Deployments

Kubernetes - Deployments

- Abstração para gerenciar um numero idêntico de pods
- Gerencia esses pods como um grupo.
 - Atualização
 - Escalabilidade
 - Rollback
- Gerencia os estados desejado.



Kubernetes – Declarative Syntax



Kubernetes – Declarative Syntax

Deployment

"Eu quero 3
Pods Node.js
rodando"

Node.js
app

Node.js
app

Node.js
app

Kubernetes – Declarative Syntax

```
kind: Deployment
apiVersion: extensions/v1beta1
metadata:
  name: hostname-101-deployment
spec:
  replicas: 3
  selector:
    # Like saying "Make sure there are three pods running
    # with the label app = hostname and version = v101"
    matchLabels:
      app: hostname
      version: v101
  template:
    metadata:
      labels:
        # The `app` label is used by both the service
        # and the deployment to select the pods they operate on.
        app: hostname
        # The `version` label is used only by the deployment
        # to control replication.
        version: v101
    spec:
      containers:
        - name: nginx-hostname
          image: kubegoldenguide/nginx-hostname:1.0.1
          ports:
            - containerPort: 80
```

Número de cópias

Devem existir 3 cópias
com os seguintes labels
rodando

Template com as mesmas
especificações de um
Pod!

Kubernetes Deployment utilizando a CLI



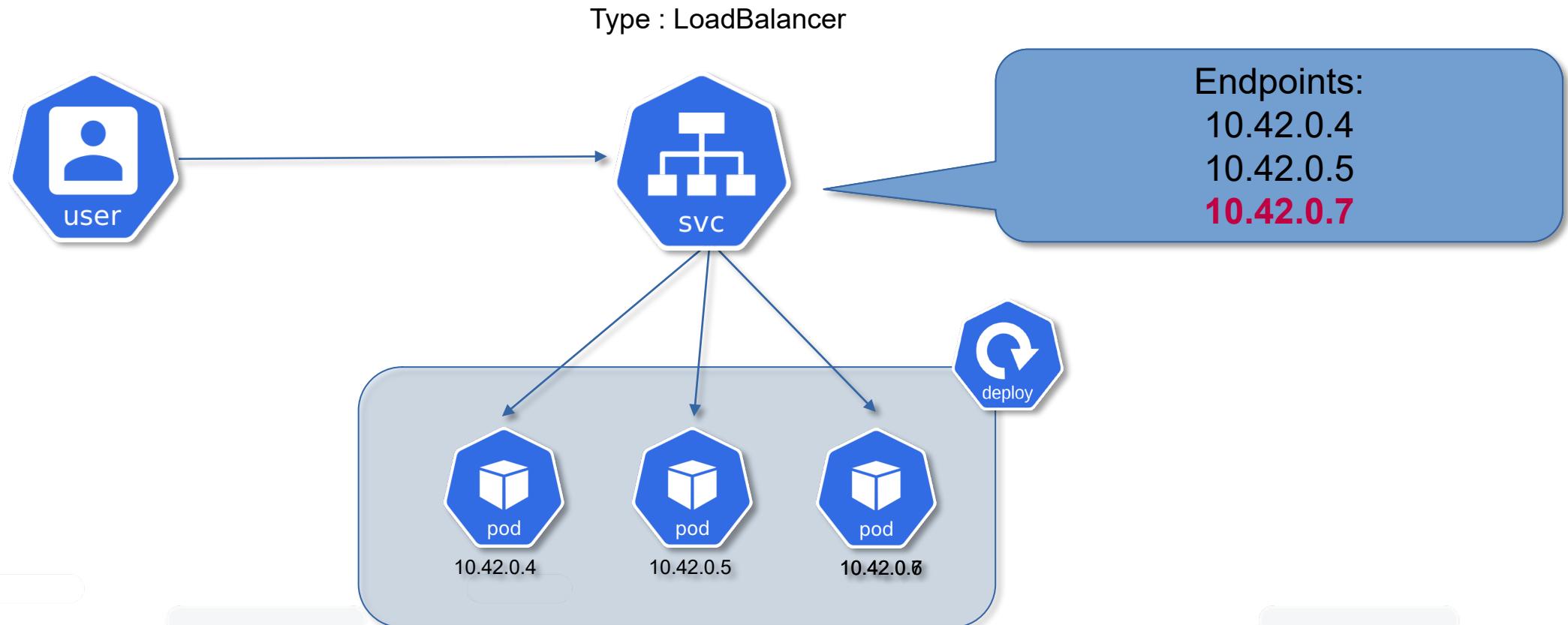
Services

Kubernetes - Recursos - Service



- Expõe a aplicação fornecida pelos pods.
- Confiável com configurações estáticas como deployments.
- Tem seu próprio IP / DNS
- Pods são **mortais** – eles possuem seu próprio ciclo de vida
 - Não adereça a eles diretamente
- Services podem ser internos(ClusterIP) ou Externos (NodePort & Loadbalancer)

Kubernetes - Recursos - Service



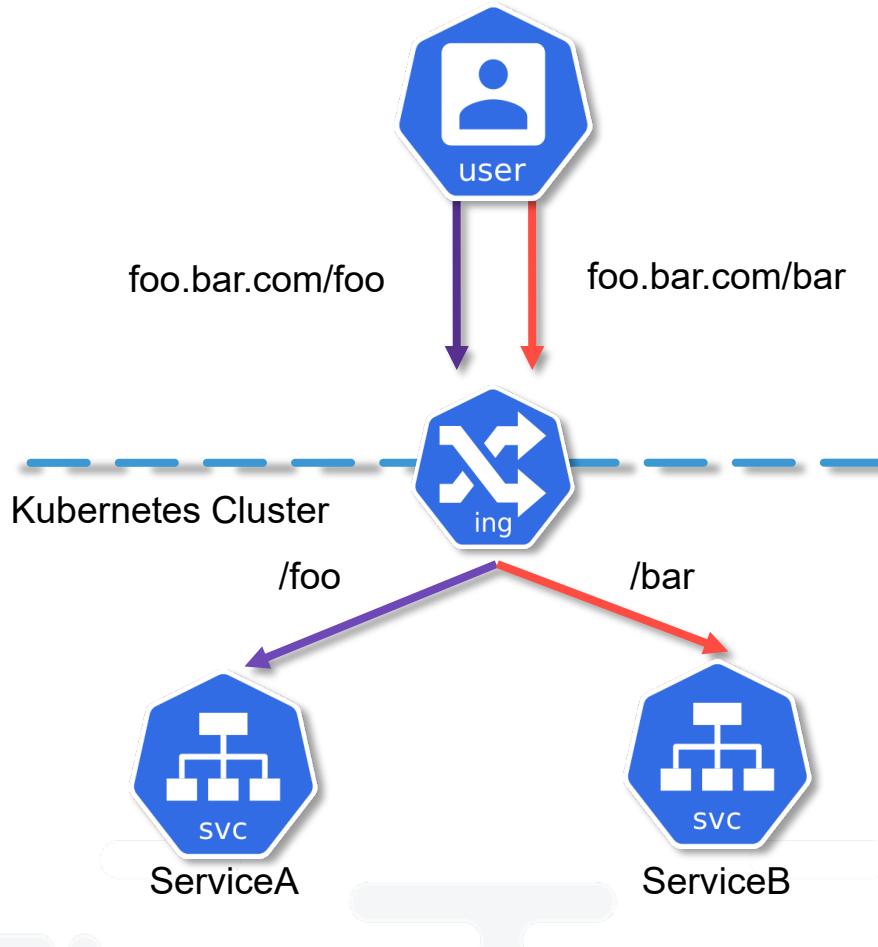
Ingresses

Kubernetes -Recursos - Ingress



- Gerencia um acesso externo a um service no cluster
 - Comum HTTP
- Permite ao service acesso externo por uma URL
- Trafego em Load Balance
- Delimita SSL/TLS

Kubernetes Resources - Ingress



```
1  apiVersion: networking.k8s.io/v1beta1
2  kind: Ingress
3  metadata:
4    name: simple-fanout-example
5  spec:
6    rules:
7      - host: foo.bar.com
8        http:
9          paths:
10         - path: /fooooo
11           backend:
12             serviceName: serviceA
13             servicePort: 4200
14         - path: /bar
15           backend:
16             serviceName: serviceB
17             servicePort: 8080
```

Community SUSE está aqui!

The screenshot shows a course landing page for 'Certified Rancher Operator: Level 1'. At the top, it says 'Run Kubernetes Everywhere'. Below that, there's a summary table:

Start Date Self-Paced	Duration 5 Weeks	Price Free
--------------------------	---------------------	---------------

A large blue button labeled 'Enroll Now' is prominently displayed. To the right of the table, there's a detailed description of the course:

- Organization: RANCHER
- Effort: 3-5 Hours
- Level: Introductory
- Subject: Installing And Using Rancher For Deploying And Managing Kubernetes

Below this, there's a section titled 'What You'll Learn' with a bulleted list:

- Introduction to RKE
- Deploying Rancher Server
- Deploying Kubernetes Clusters
- Working With Kubernetes
- Securing Kubernetes

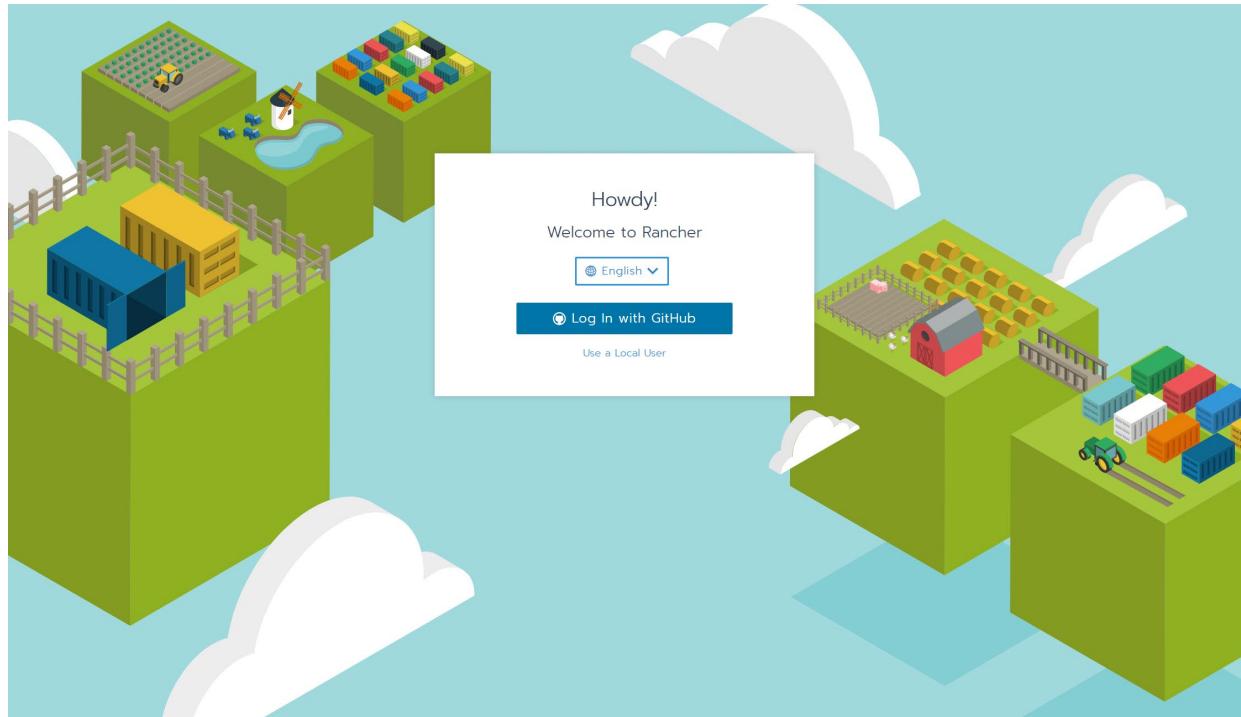
At the bottom left, there's a logo for 'RANCHER ACADEMY' featuring a white bull icon.

- Videos de introdução
- Teoria
- Demonstração
- Laboratorios
- Quiz
- Exame final
- Certificação

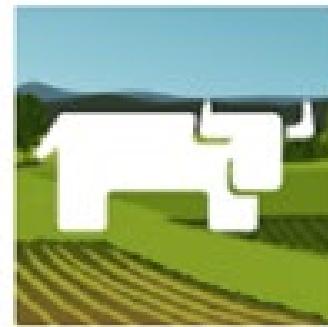
Ganhe conhecimento e aprofundamento.

Vá em:
<https://community.suse.com/>
Sessão "Academy Classes"

Rancher Demo



<https://slack.rancher.io>



#Kubernetes channel
[#portugues](#)

Sorteio



Obrigado por sua participação!

