

# Kubernetes

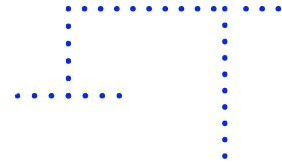
DEVinHouse

Parcerias para desenvolver a sua carreira

**SENAI**

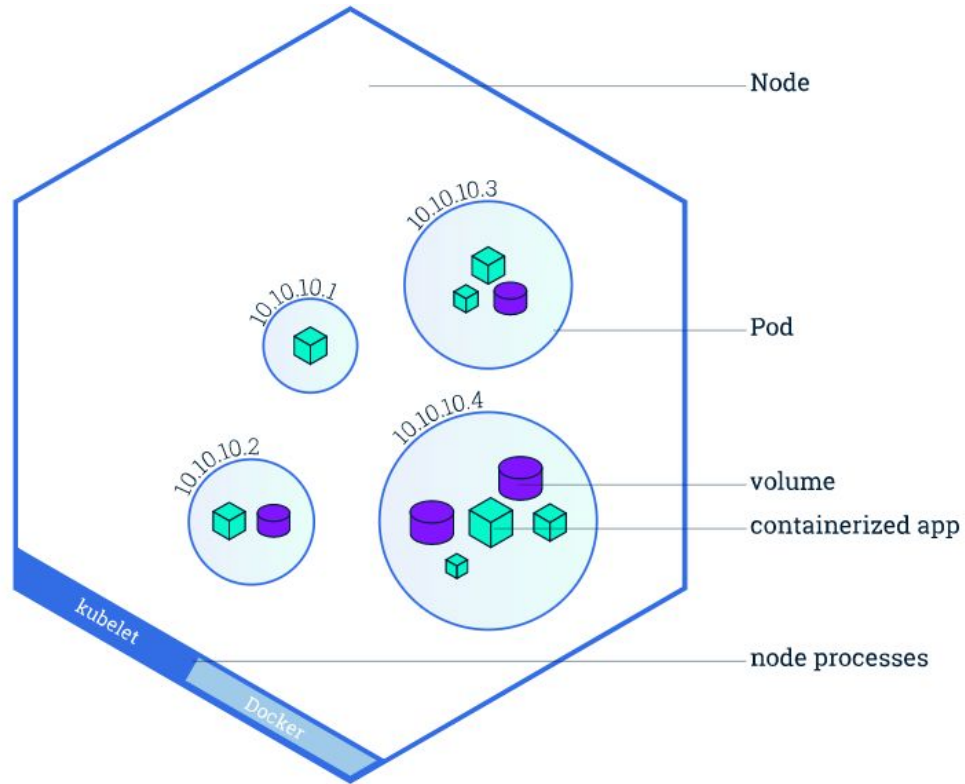
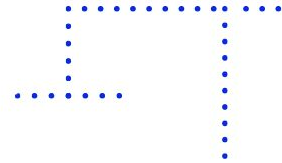
<LAB365>

# Kubernetes

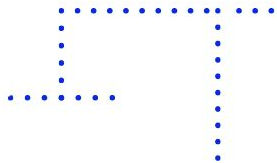


- software de código aberto, produzido pela google para gerenciamento de microsserviços (containers)
- conceito de pequenas aplicações independentes cresceu e assim surgiu demanda de gerenciar todos esses containers
- volumes de container e diferentes ambientes podem ser muito grandes e complexa a orquestração

# Kubernetes

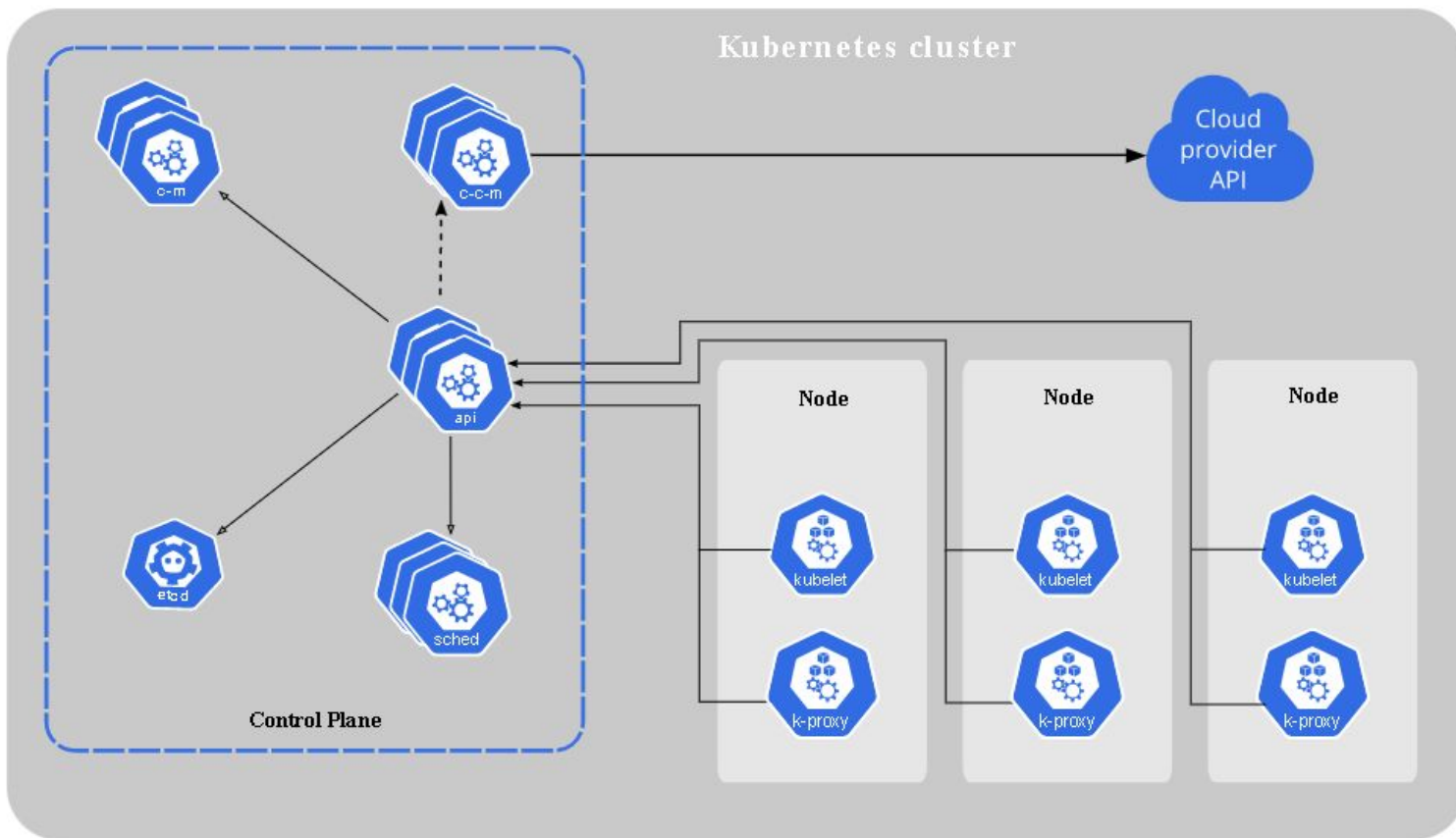


# Kubernetes

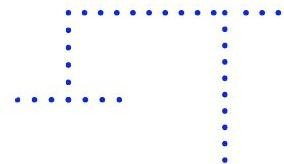


- 3 processos básicos do node kubernetes
  - container runtime (docker)
  - kubelet - responsável pela comunicação entre node e pods (alocar recursos do node para imagem)
  - kube proxy - responsável por distribuir as demandas do serviço com os pods
- 4 processos básicos do master kubernetes
  - api server - primeira validação ao criar um deploy/service no cluster
  - scheduler - analisa qual node está melhor para receber um pod, por exemplo
  - controler manager - analisa a integridade do cluster como pods que encerraram
  - etcd - guarda os históricos de atualização de serviços, réplicas, e alterações em geral no cluster

# Kubernetes



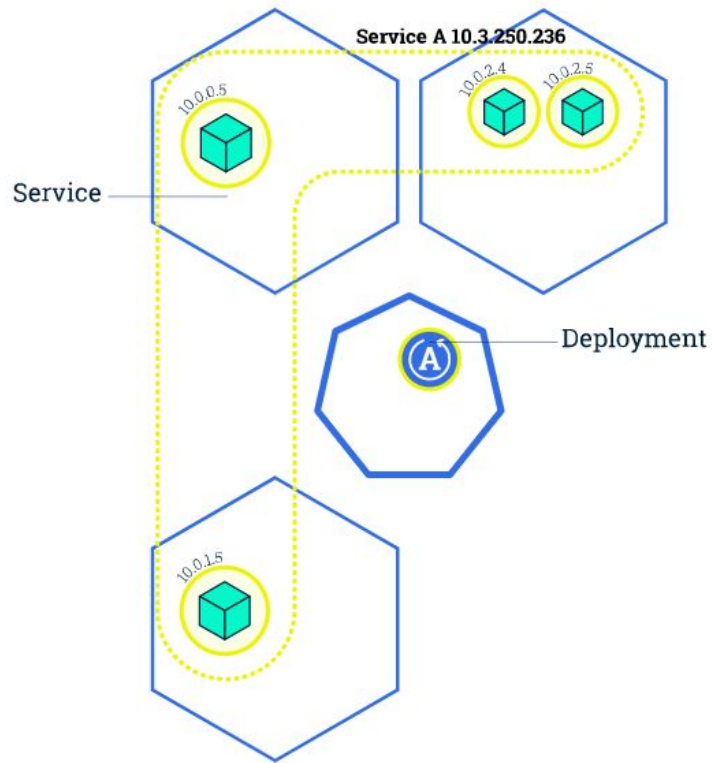
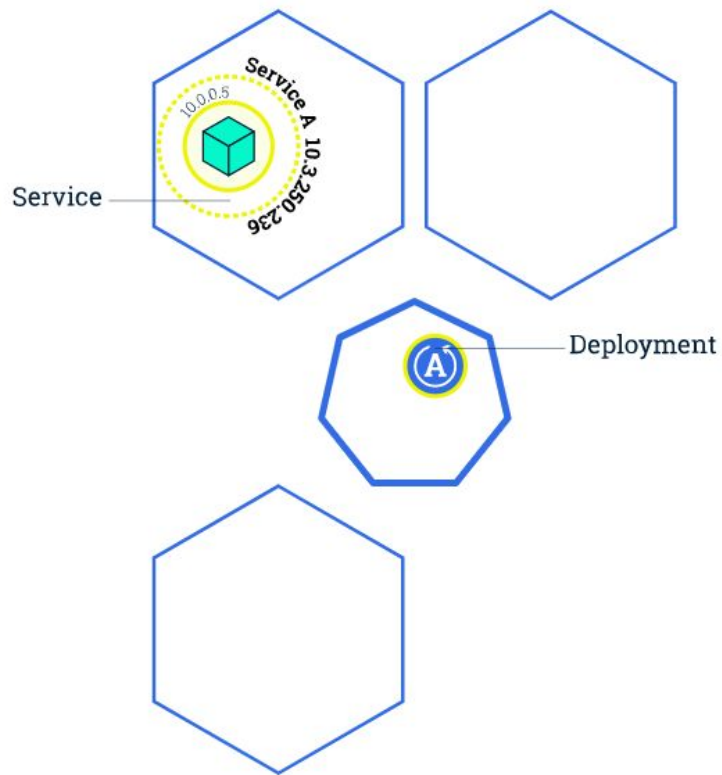
# Kubernetes - arquivo yaml



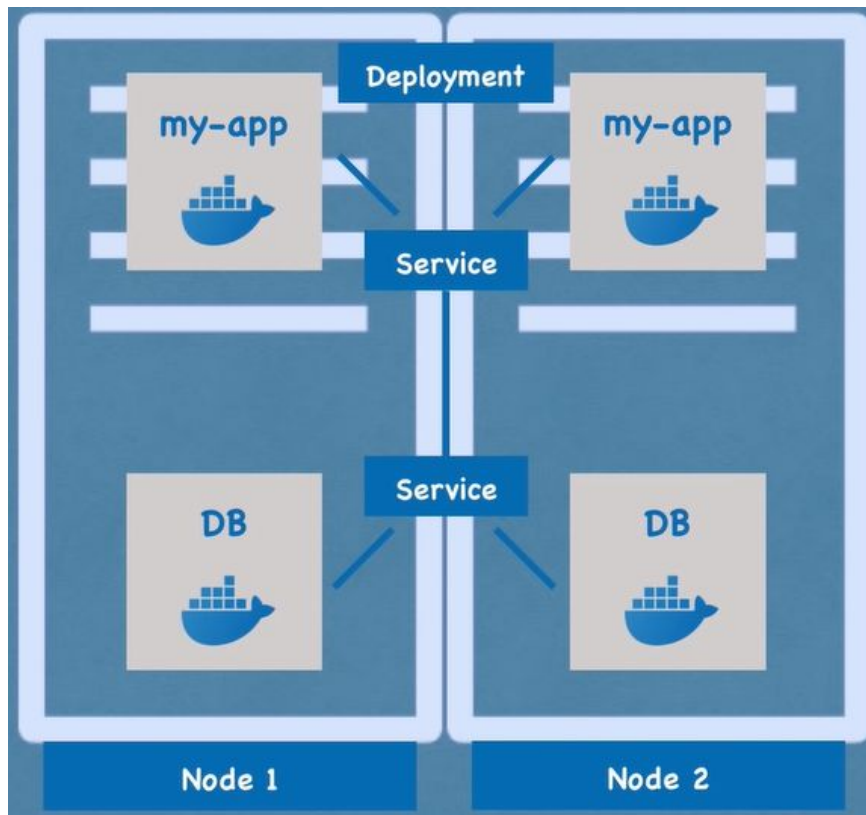
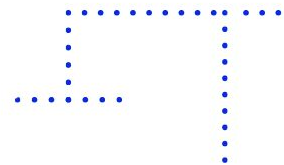
```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: mongodb-deployment
  labels:
    app: mongodb
spec:
  replicas: 1
  selector:
    matchLabels:
      app: mongodb
  template:
    metadata:
      labels:
        app: mongodb
    spec:
      containers:
        - name: mongodb
          image: mongo
          ports:
            - containerPort: 27017
          env:
            - name: MONGO_INITDB_ROOT_USERNAME
              valueFrom:
                secretKeyRef:
                  name: mongodb-secret
                  key: mongo-root-username
            - name: MONGO_INITDB_ROOT_PASSWORD
              valueFrom:
```

```
apiVersion: v1
kind: Service
metadata:
  name: mongodb-service
spec:
  selector:
    app: mongodb
  ports:
    - protocol: TCP
      port: 27017
      targetPort: 27017
```

# Kubernetes



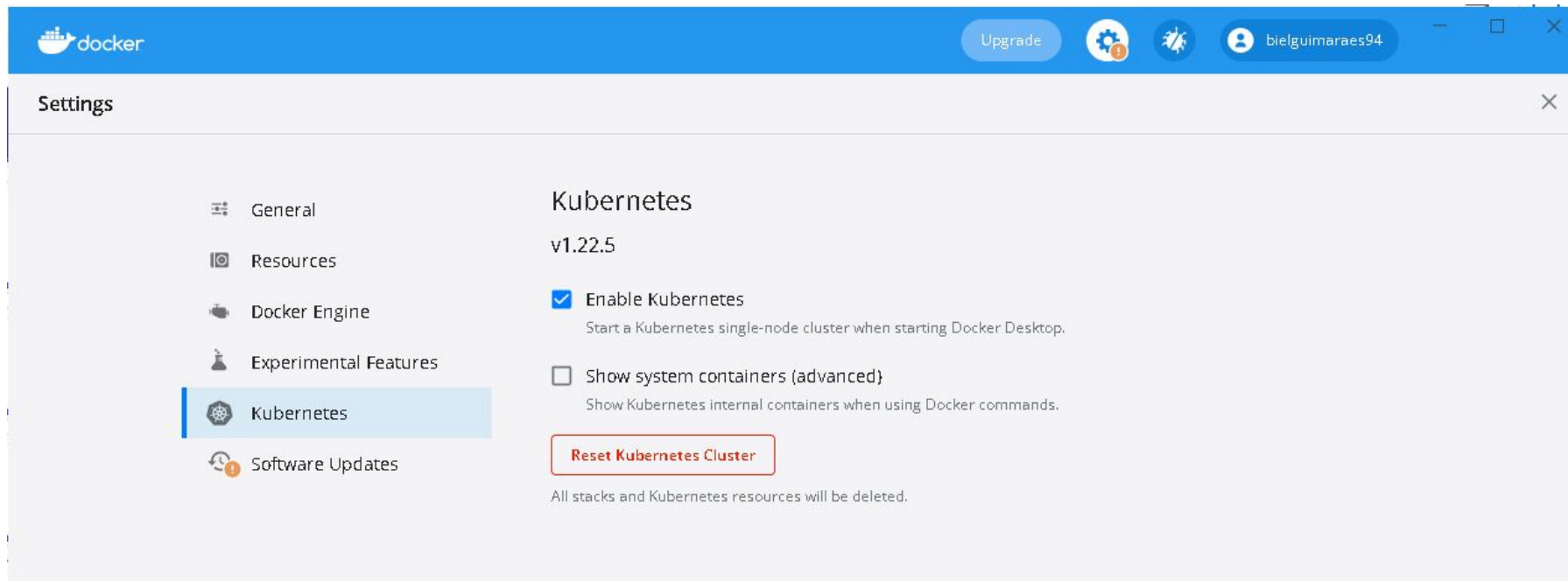
# Kubernetes



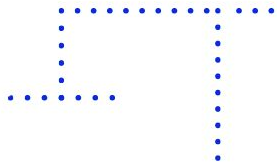


# Kubernetes - instalação

Podemos habilitar o Kubernetes a partir do Docker desktop:



# Kubernetes - instalação



1º Instalando em Ubuntu temos primeiramente instalar o docker:

```
curl -fsSL https://get.docker.com | bash
```

2º vamos importar o certificado:

```
curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | apt-key add -
```

3º adicionar o repositório:

```
echo "deb http://apt.kubernetes.io/ kubernetes-xenial main" >  
/etc/apt/sources.list.d/kubernetes.list
```

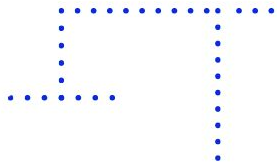
4º atualizar os pacotes com apt-get update e instalar os componentes:

```
apt-get install kubeadm kubectl kubelet -y
```

Obs: desabilitar o swap - "swapoff -a"

ir até vim /etc/fstab e comentar linha responsável pelo swap

# Kubernetes - instalação



Observação: as versões mais recentes do Docker usam o cgroup diferente do Kubernetes.

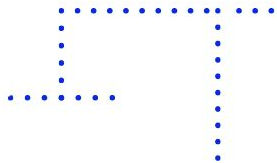
Para evitar erros na criação do cluster é recomendado rodar esses comandos antes de prosseguir:

1° `cat <<EOF | sudo tee /etc/docker/daemon.json`

```
{  
  "exec-opts": ["native.cgroupdriver=systemd"],  
  "log-driver": "json-file",  
  "log-opts": {  
    "max-size": "100m"  
  },  
  "storage-driver": "overlay2"  
}  
EOF
```

2° `sudo systemctl restart docker`

# .....> Kubernetes - Minikube

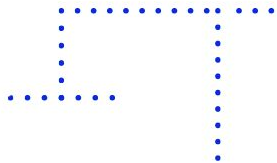


Minikube provê um ambiente de simulação de cluster localmente. Para instalar em VM com Ubuntu basta executar os seguintes comandos:

- 1° `sudo apt install -y curl wget apt-transport-https`
- 2° `wget https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64`
- 3° `sudo cp minikube-linux-amd64 /usr/local/bin/minikube`
- 4° `sudo chmod +x /usr/local/bin/minikube`
- 5° `minikube start --driver=none`

O comando `minikube status` informa se o minikube está em execução.

# Kubernetes - Cluster



Para criar o cluster sem minikube, na forma convencional, executamos o comando no node master:

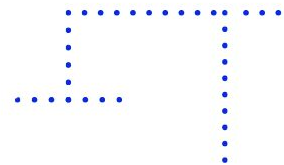
```
kubeadm init --apiserver-advertise-address ip-node-master
```

O próximo passo é instalar o Weave Net para os nodes ficarem em status ready:

```
kubectl apply -f "https://cloud.weave.works/k8s/net?k8s-version=$(kubectl version |  
base64 | tr -d '\n')"
```

Agora é colar os comandos joins retornados para dentro dos workers (semelhante ao Docker Swarm)

# Kubernetes



Acessando o terminal do pod com MySQL:

```
root@k8smaster:/home/ubuntu# kubectl exec -ti mysql-795c9d77dc-bgpmx -- bash
root@mysql-795c9d77dc-bgpmx:/# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 70
Server version: 8.0.28 MySQL Community Server - GPL

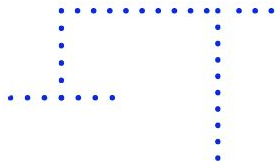
Copyright (c) 2000, 2022, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

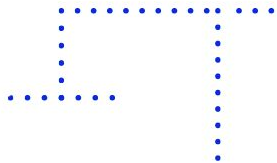
mysql>
```

# Kubernetes



```
root@k8smaster:/home/ubuntu# kubectl delete pod mysql-795c9d77dc-bgpmx
pod "mysql-795c9d77dc-bgpmx" deleted
^C
root@k8smaster:/home/ubuntu# ^C
root@k8smaster:/home/ubuntu# ^C
root@k8smaster:/home/ubuntu# kubectl get pods
NAME                                READY   STATUS             RESTARTS   AGE
mysql-795c9d77dc-9w4mz             1/1    Running            0           9m29s
mysql-795c9d77dc-bgpmx             0/1    Terminating      0           130m
phpmyadmin-deployment-c7d57844d-6hdhw 1/1    Running            0           113m
root@k8smaster:/home/ubuntu#
```

# Orquestração - Rancher



Rancher é uma ferramenta para auxílio na implantação de serviços e no gerenciamento do cluster Kubernetes via interface gráfica.

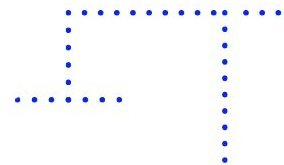
Para instalar temos que primeiramente subir o container:

```
docker run -d -p 8080:8080 rancher/server:stable
```

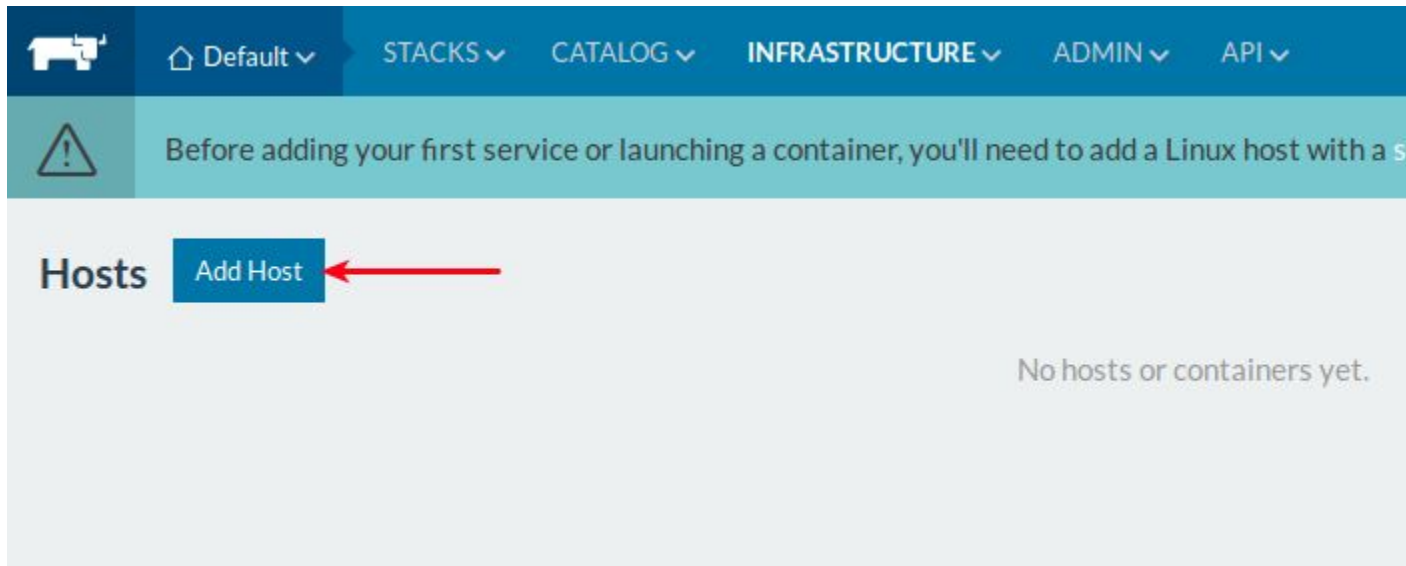
Para abrir a aplicação colocamos no navegador o ip do host gerado o container e porta 8080



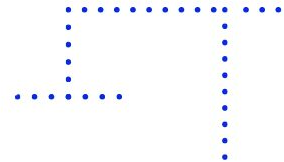
# Orquestração - Rancher



Agora é só adicionar os hosts que queremos gerenciar com a ferramenta. (Atenção as versões Docker suportadas)



# Orquestração - Rancher



## Add Host

PROVIDER

AMAZON EC2

DigitalOcean

exoscale

packet

rackspace

Other

Custom

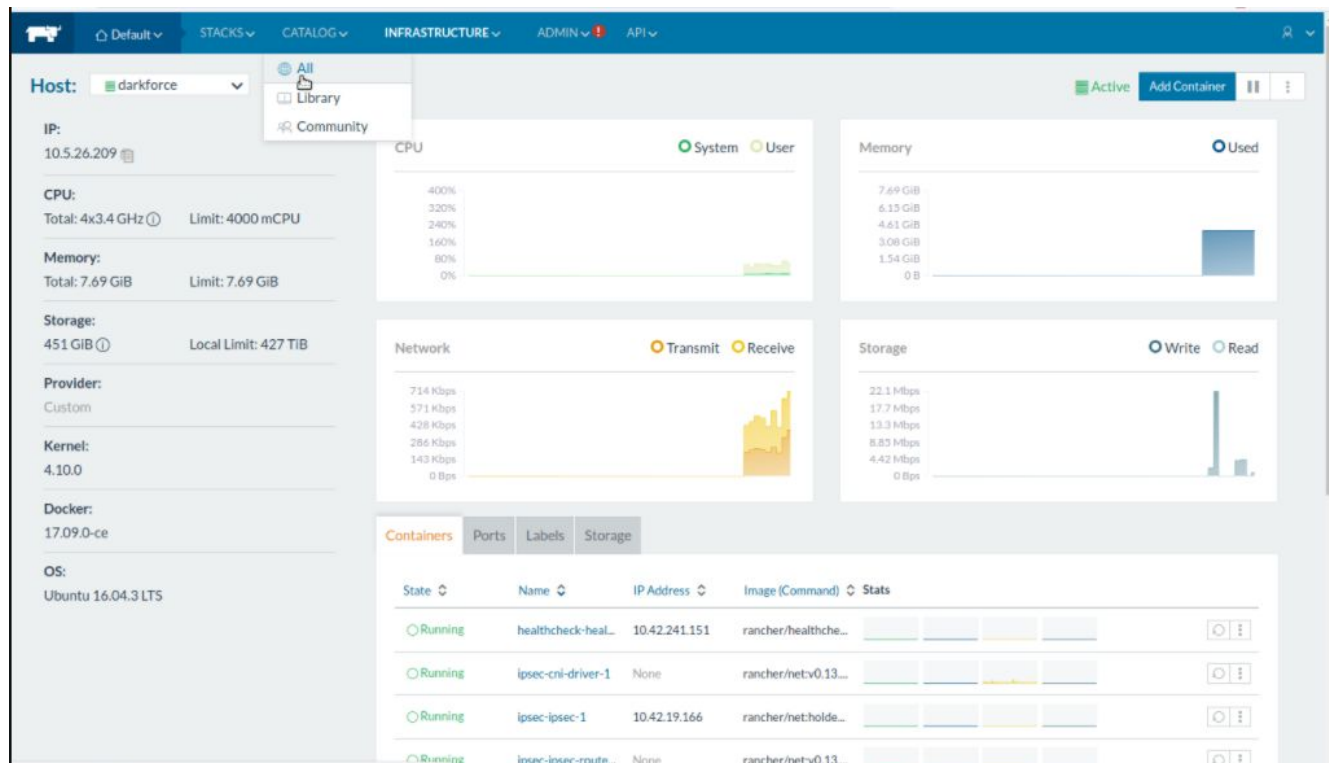
- 1 Start up a Linux machine somewhere and install the latest version of [Docker](#) on it.
- 2 Make sure any security groups or firewalls allow traffic:
  - From and To all other hosts on **UDP** ports **500** and **4500** (for IPsec networking)
- 3 Optional: Add labels to be applied to the host.

[+ Add Label](#)
- 4 Copy, paste, and run the command below to register the host with Rancher:

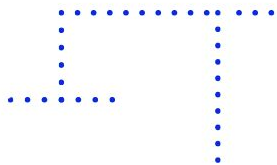
```
sudo docker run -d --privileged -v /var/run/docker.sock:/var/run/docker.sock rancher/agent:v0.8.2 http://rancherdemo.my-cyclingstore.com:8080/v1/scripts/3F167DFA31E3EE
```
- 5 Click close below. The new host should pop up on the [Hosts](#) screen within a minute.



Close


# Orquestração - Rancher




# Orquestração - Rancher





 Predefinição ▾ PILHAS ▾ CATÁLOGO ▾ A INFRAESTRUTURA ▾ ADMINISTRADOR ▾ INCÊNDIO ▾ 

 Antes de adicionar seu primeiro serviço ou iniciar um contêiner, você precisa adicionar um anfitrião com uma versão compatível do Docker.

## Contas

Estado ▾	identificação ▾	Amável ▾	Identidade
 Ativo	1a1	Administrador	Nenhum



Registro de auditoria

Contas

Processos

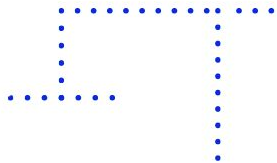
Alta disponibilidade


Controle de acesso


Drivers de máquina

Configurações

# Orquestração - Rancher



 Predefinição ▾ **PILHAS ▾** CATÁLOGO ▾ A INFRAESTRUTURA ▾ ADMINISTRADOR ▾ ! INCÊNDIO ▾

 Antes de adicionar seu primeiro serviço ou iniciar um contêiner, você precisará adicionar um host Linux com uma versão compatível do Docker. Adicionar um anfitrião

**Pilhas de usuários** Adicionar pilha

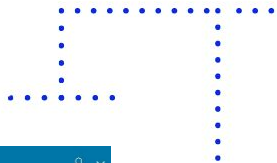
## Adicionando sua primeira pilha



Um serviço é simplesmente um grupo de contêineres criados a partir da mesma imagem do Docker, mas estende o conceito de 'link' do Docker para aproveitar o serviço DNS distribuído leve do Rancher para descoberta de serviços ou implantando um item do Catálogo.


Um serviço também é capaz de aproveitar outros serviços internos do Rancher, como balanceadores de carga, monitoramento de integridade, suporte de atualização e alta disponibilidade. [Saber mais](#)

Definir um serviço Procurar Catálogo

# Orquestração - Rancher




 Default ▾ STACKS ▾ CATALOG ▾ INFRASTRUCTURE ▾ ADMIN ▾ API ▾ 


 Before adding your first service or launching a container, you'll need to add a Linux host with a supported version of Docker. Add a host

### Add Service

**Scale**

☒ Run 1 container 

☐ Always run one instance of this container on every host

mysql-rancher 

**Name**

**Description**

**Select Image\***

☒ Always pull image before creating


Public Host Port

Private Container Port

Protocol

TCP ▾

Show Host IP field



# Orquestração - Rancher



Command

Volumes

Networking

Security/Host

Secrets

Health Check

Labels

Scheduling

Command

e.g. /usr/sbin/httpd -f httpd.conf

Entry Point

e.g. /bin/sh

Working Dir

e.g. /myapp

Console

☒ Interactive & TTY (-i -t)

☐ TTY (-t)

User

e.g. apache

☐ Interactive (-i)

☐ None

Auto Restart

☒ Always

☐ Never (Start Once)

Drain Timeout

0

ms

Environment

 Add Environment Variable

Create

Cancel





# DEVinHouse

Parcerias para desenvolver a sua carreira

**OBRIGADO!**



<LAB365>