PROGRAMAÇÃO CONCORRENTE E DISTRIBUÍDA

Orquestração de Containeres

Aula 14 — Kubernetes: conceitos e passo a passo prático

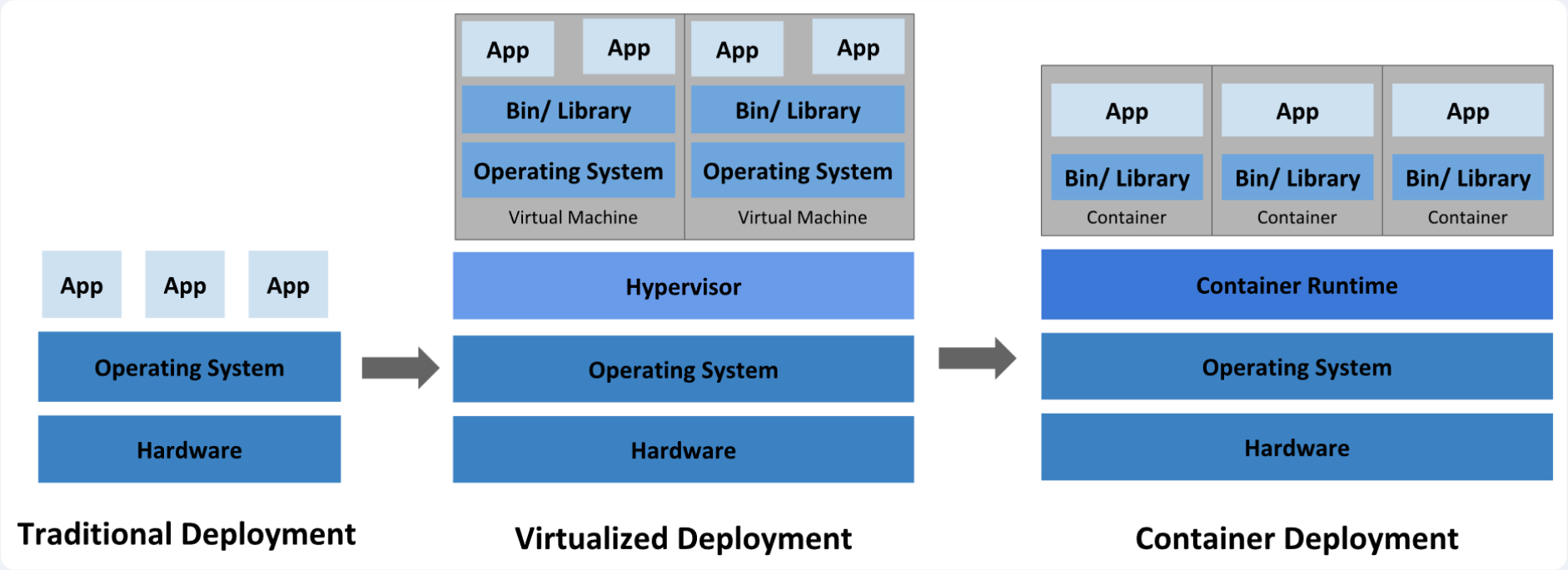
Prof. Newton S. B. Miyoshi
newton.miyoshi@baraodemaua.br
Centro Universitário Barão de Mauá — Ribeirão Preto

PARTE 1

Kubernetes

Por que orquestrar containers, e com o quê.

Revisão


App
App
App
Operating System
Hardware
Traditional Deployment
App
App
App
App
Bin/ Library
Bin/ Library
Operating System
Operating System
Virtual Machine
Virtual Machine
Hypervisor
Operating System
Hardware
Virtualized Deployment
App
App
App
Bin/ Library
Bin/ Library
Bin/ Library
Container
Container
Container
Container Runtime
Operating System
Hardware
Container Deployment

PARA DISCUTIR

Supondo que as aplicações rodam em containers com Docker e Docker Compose, quais problemas podemos ter?

Orquestração de Containers

- Configuração de réplicas.
- Load balancing.
- Rollout e rollback automatizados (CI/CD).
- Self-healing.
- Gerenciamento de chaves (secrets) e configurações.

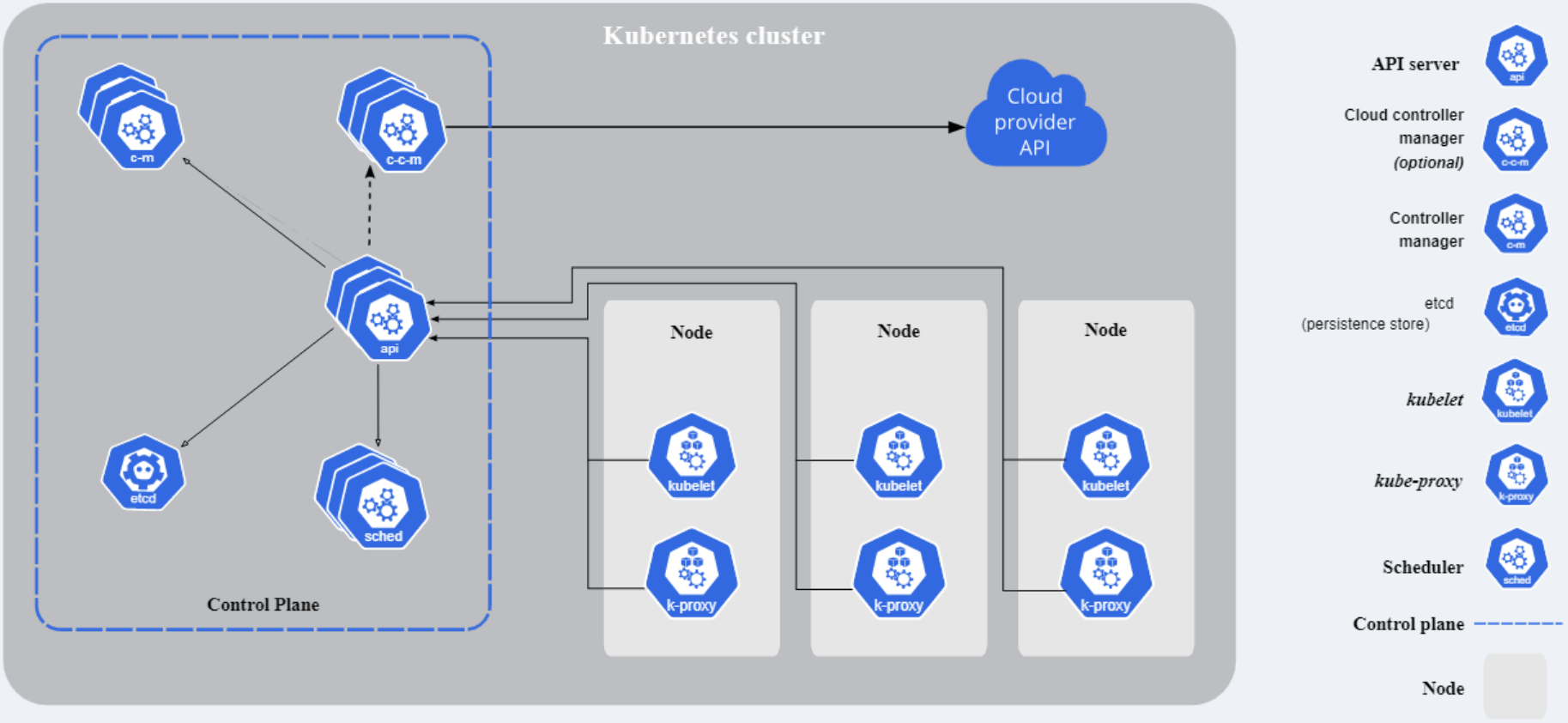
Vantagens

- Infraestrutura declarativa.
- Facilidade de implantação e atualização de aplicativos.
- Isolamento e segurança.
- Gerenciamento de recursos e otimização de custo.
- Escalabilidade.
- Alta disponibilidade.

Desvantagens

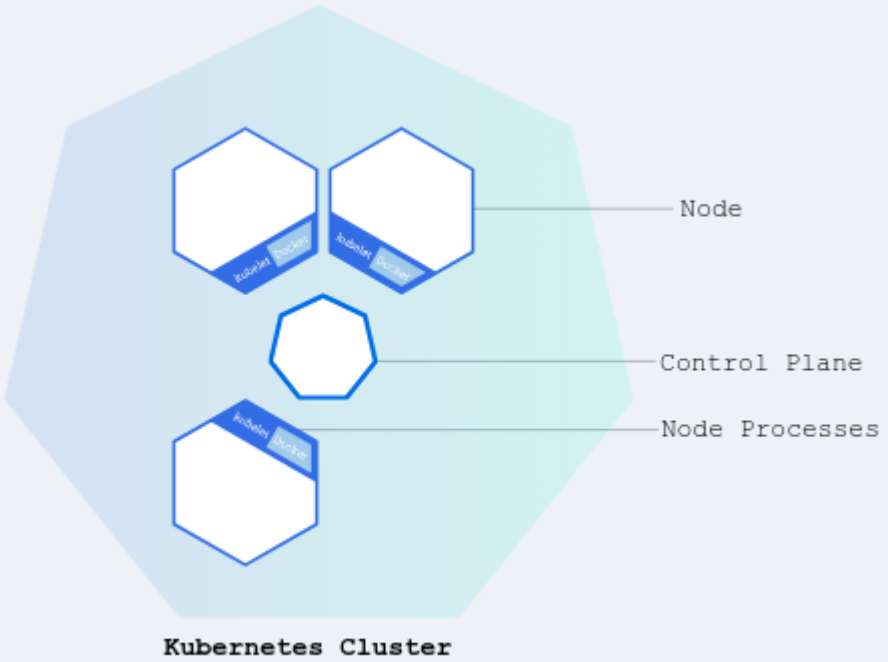
- Complexidade.
- Curva de aprendizado.
- Customização complexa.
- Overhead de rede.
- Recursos mínimos maiores.

Componentes K8s



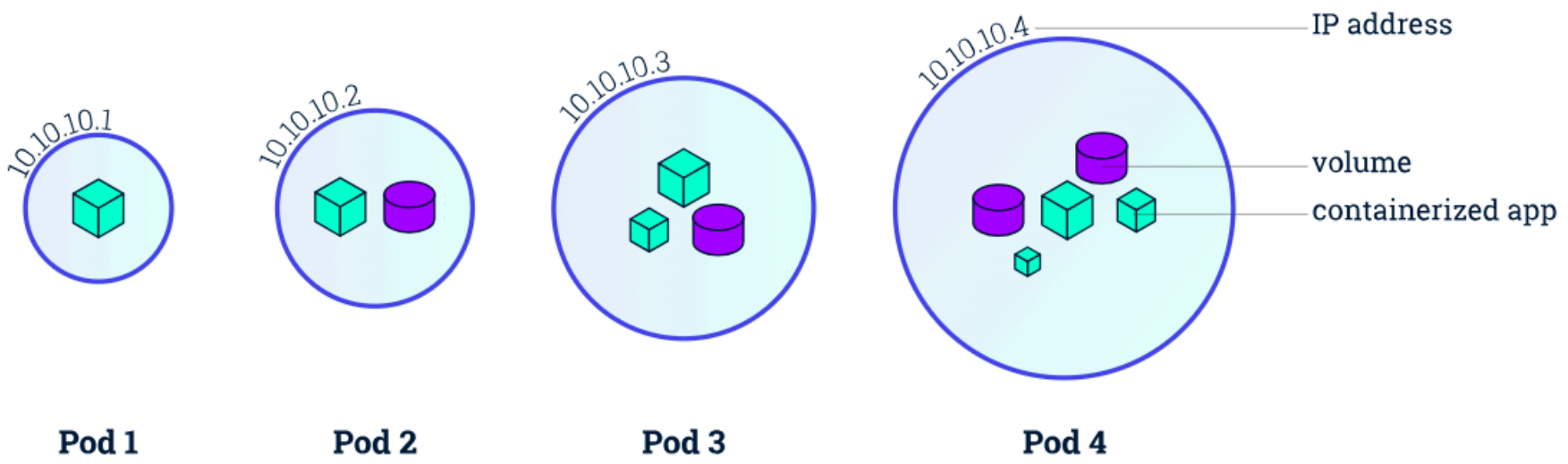
Cluster

- Control plane — coordenação.
- Nodes (nós) — workers que executam as aplicações; podem ser um servidor ou uma VM.

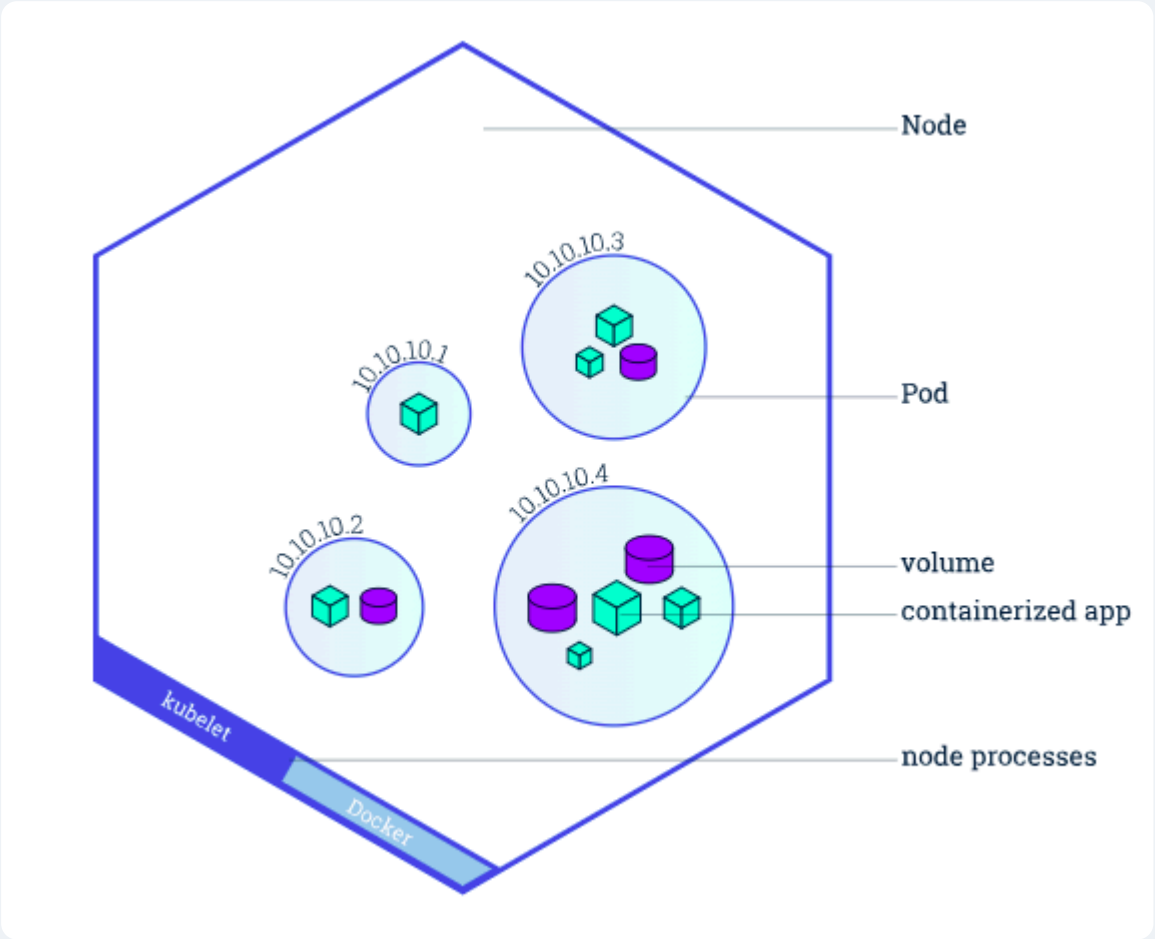


Pods

- Pod é uma abstração que representa 1 ou mais containers (apps).
  - Compartilham recursos: storage, volumes e rede.
- Pods rodam em um Node.



Nodes, Pods e Containers

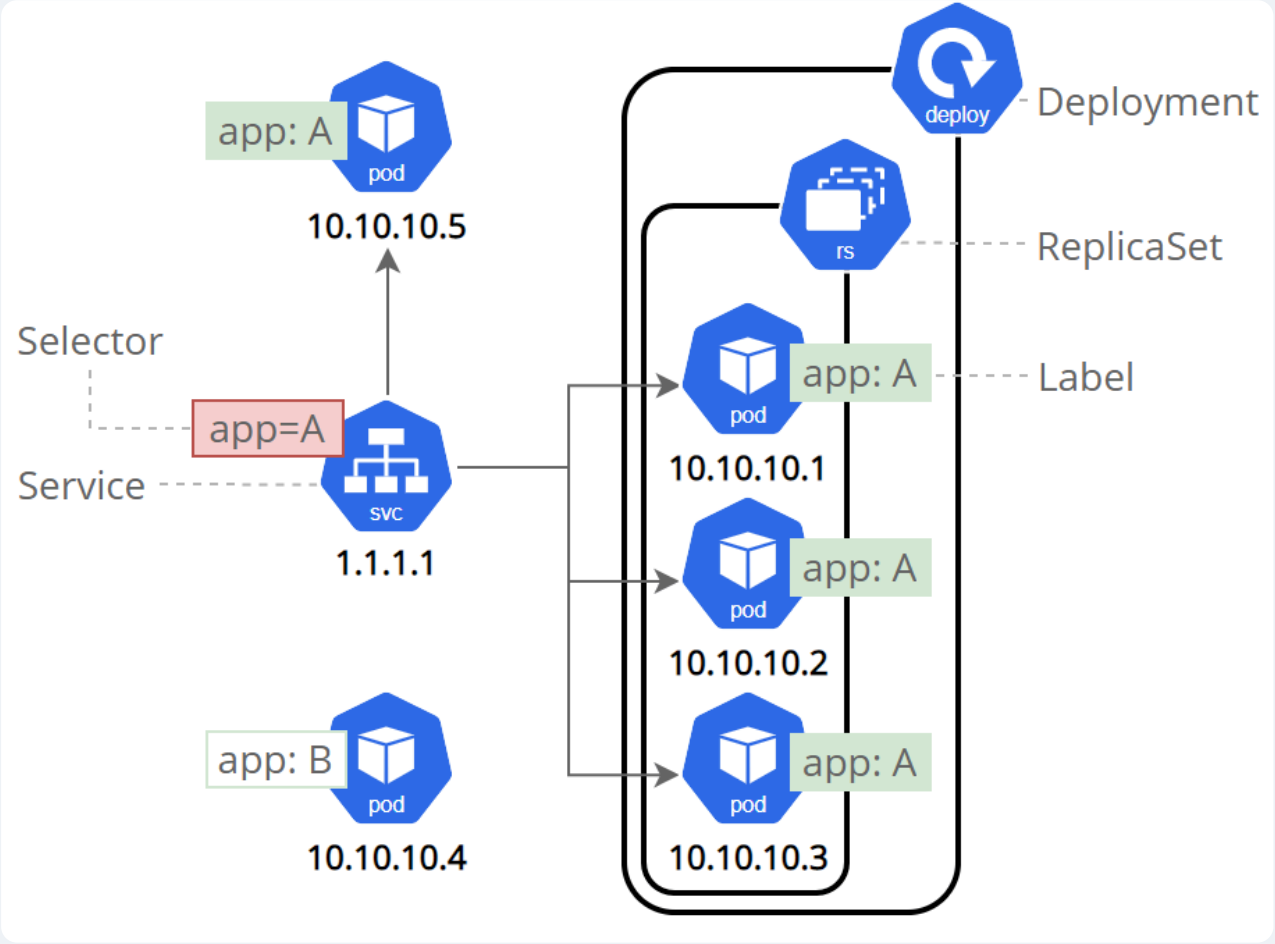


ReplicaSet

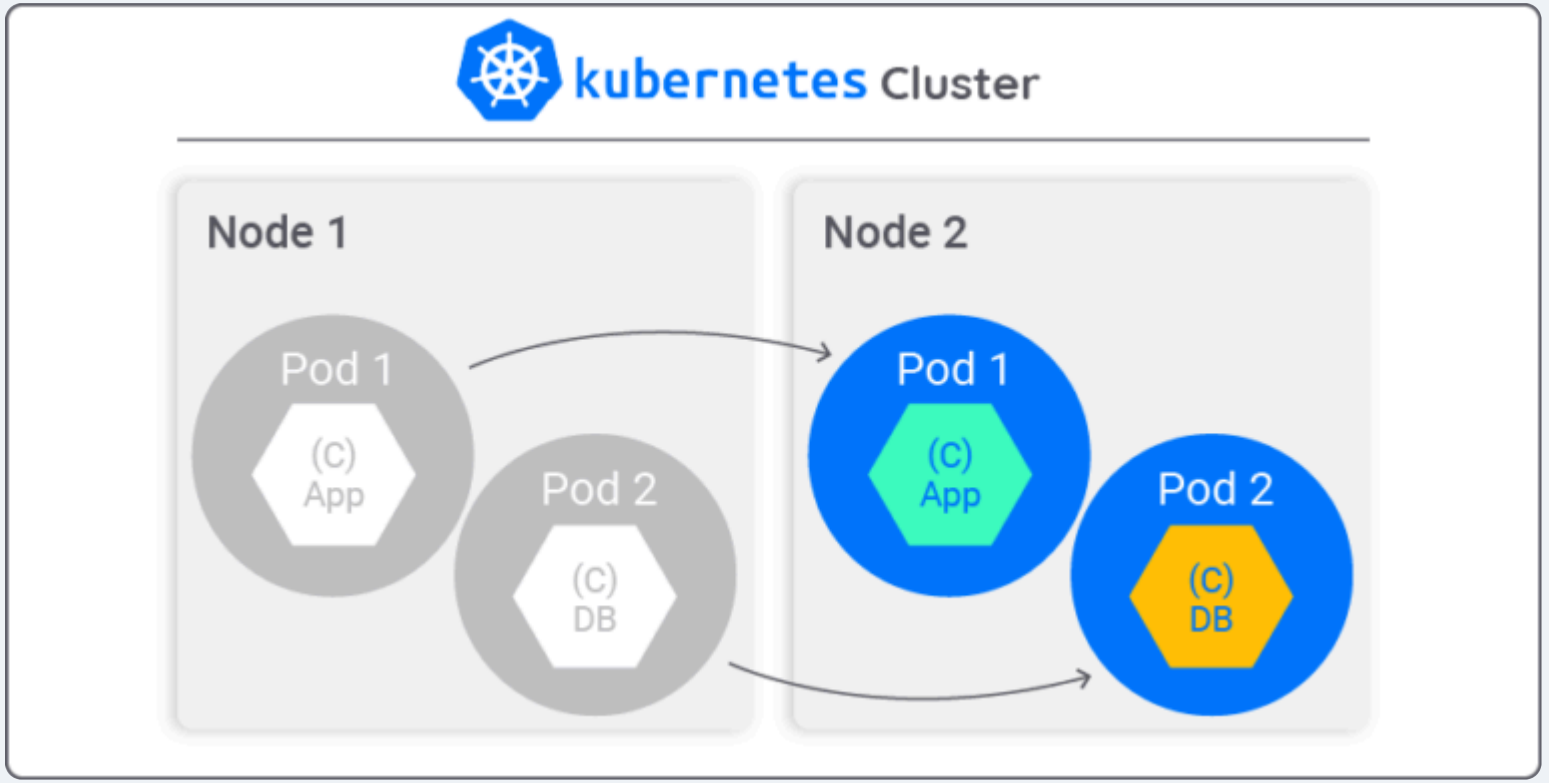
- Conjunto de réplicas de Pods.
- Responsável por manter um conjunto definido de Pods idênticos rodando.
- Deployment: abstração que define as políticas gerais (quantas réplicas, estratégia de atualização...).

Services

- Service é um conjunto de Pods com uma política de acesso.
- Expõe Pods para fora do cluster (internet) — Load Balancer, IP externo.



Componentes K8s — auto-cura



PARTE 2

Kubernetes na prática

Subindo um cluster local com `kind`.

PARA DISCUTIR

Qual a vantagem de usar Kubernetes para o Café com Pão? E qual o passo a passo para subir a aplicação nele?

Passo a passo — instalar o kind

```
# Windows, no PowerShell
curl.exe -Lo kind-windows-amd64.exe `
  https://kind.sigs.k8s.io/dl/v0.17.0/kind-windows-amd64
Move-Item .\kind-windows-amd64.exe c:\some-dir-in-your-PATH\kind.exe
```

Instruções para todos os sistemas: kind.sigs.k8s.io/docs/user/quick-start

Passo a passo — Windows com WSL2

Criar o cluster no WSL2, mapeando uma porta do container para o host:

```yaml
# cluster-config.yml
kind: Cluster
apiVersion: kind.x-k8s.io/v1alpha4
nodes:
- role: control-plane
  extraPortMappings:
  - containerPort: 30000
    hostPort: 30000
    protocol: TCP
```

kind.sigs.k8s.io/docs/user/using-wsl2

Passo a passo — Windows com WSL2

Criar um service nginx e acessar:

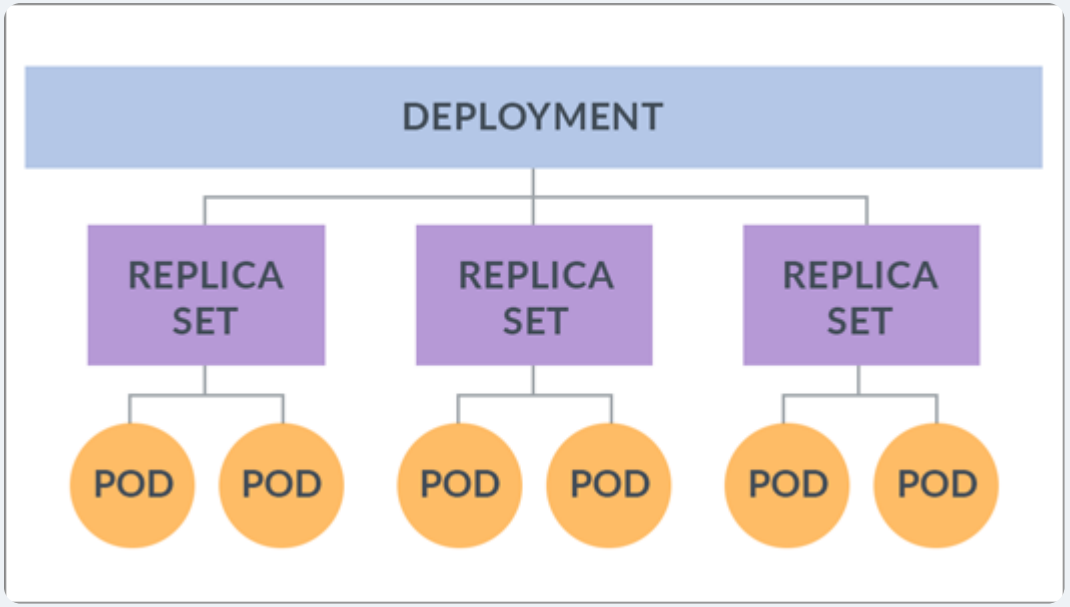
```
$ kind create cluster --config cluster-config.yml
$ kubectl create deployment nginx --image=nginx --port=80
deployment.apps/nginx created
$ kubectl create service nodeport nginx --tcp=80:80 --node-port=30000
service/nginx created
$ curl localhost:30000
```

Passo a passo — criar cluster com `kind`

```
$ kind create cluster
Creating cluster "kind" ...
 ✓ Ensuring node image (kindest/node:v1.25.3)
 ✓ Preparing nodes
 ✓ Writing configuration
 ✓ Starting control-plane
 ✓ Installing CNI
 ✓ Installing StorageClass
Set kubectl context to "kind-kind"

$ kubectl cluster-info
Kubernetes control plane is running at https://127.0.0.1:36771
```

O nome padrão do cluster é `kind`.


DEPLOYMENT
REPLICA SET
REPLICA SET
REPLICA SET
POD
POD
POD
POD
POD
POD

Passo a passo — `hello-node`

```
$ kubectl create deployment hello-node --image=registry.k8s.io/echoserver:1.4
$ kubectl get deployments
$ kubectl get pods
$ kubectl get events
$ kubectl expose deployment hello-node --type=LoadBalancer --port=8080
$ kubectl port-forward deployment/hello-node 8080:8080
$ curl http://localhost:8080
```

Configurar Load Balancer

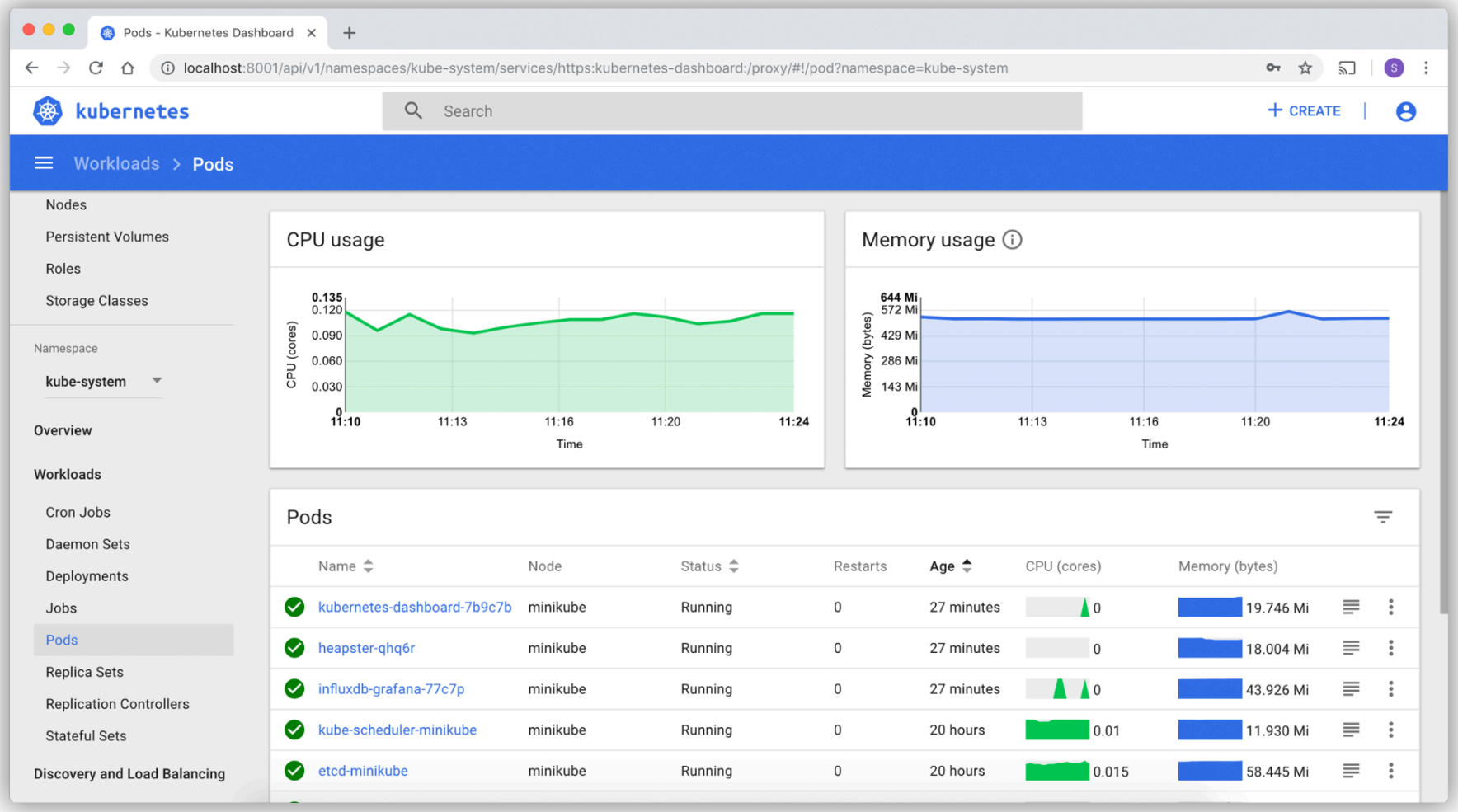
O `kind` não provisiona um LoadBalancer real — para isso, usa-se o MetalLB.

kind.sigs.k8s.io/docs/user/loadbalancer

Passo a passo — limpando os recursos

```
$ kubectl delete service hello-node
$ kubectl delete deployment hello-node
$ kubectl get pods
$ kubectl delete pod pod_name
```

Kubernetes Dashboard



WordPress + MySQL + Secrets

```
$ curl -LO https://k8s.io/examples/application/wordpress/mysql-deployment.yaml
$ curl -LO https://k8s.io/examples/application/wordpress/wordpress-deployment.yaml
```

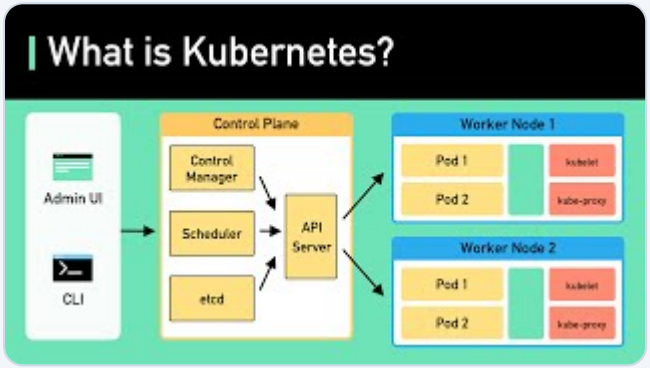
```
secretGenerator:
- name: mysql-pass
  literals:
  - password=s&nh@123
resources:
- mysql-deployment.yaml
- wordpress-deployment.yaml
```

Depois de ajustar o `nodePort` no service do WordPress: `kubectl apply -k ./` — e para desfazer, `kubectl delete -k ./`. Tutorial completo ↗

MATERIAL EM VÍDEO

YOUTUBE **100 Seconds of Kubernetes — Fireship** youtube.com/watch?v=PziYflu8cB8

MATERIAL EM VÍDEO



YOUTUBE **What is Kubernetes? — ByteByteGo** youtube.com/watch?v=TlHvYWVUZyc

MATERIAL EM VÍDEO

YOUTUBE **Iniciando com Kubernetes — Tutorial Passo a Passo — Full Cycle** youtube.com/watch?v=IeB9_PSGp6k

Referências bibliográficas

- TANENBAUM, A. S.; VAN STEEN, M. Distributed Systems. 3. ed. 2018. Cap. 7. ISBN 978-90-815406-2-9. distributed-systems.net
- KUBERNETES. Kubernetes Documentation. kubernetes.io/docs