

ORQUESTRAÇÃO COM K3S



O QUE É KUBERNETES E QUAL SEU PAPEL?

Kubernetes é uma plataforma de orquestração de containers usada para:

- Executar aplicações distribuídas
- Gerenciar deploys
- Reiniciar serviços automaticamente
- Escalar aplicações conforme demanda

Ele funciona como um "sistema operacional para clusters", distribuindo carga entre máquinas e mantendo tudo funcionando mesmo quando um nó cai

DEPLOY, ESCALABILIDADE E GERENCIALMENTO

DEPLOY

- Descrição do app em YAML
- Replicação e atualização automaticamente
- Atualizações feitas sem derrubar o sistema

ESCALABILIDADE

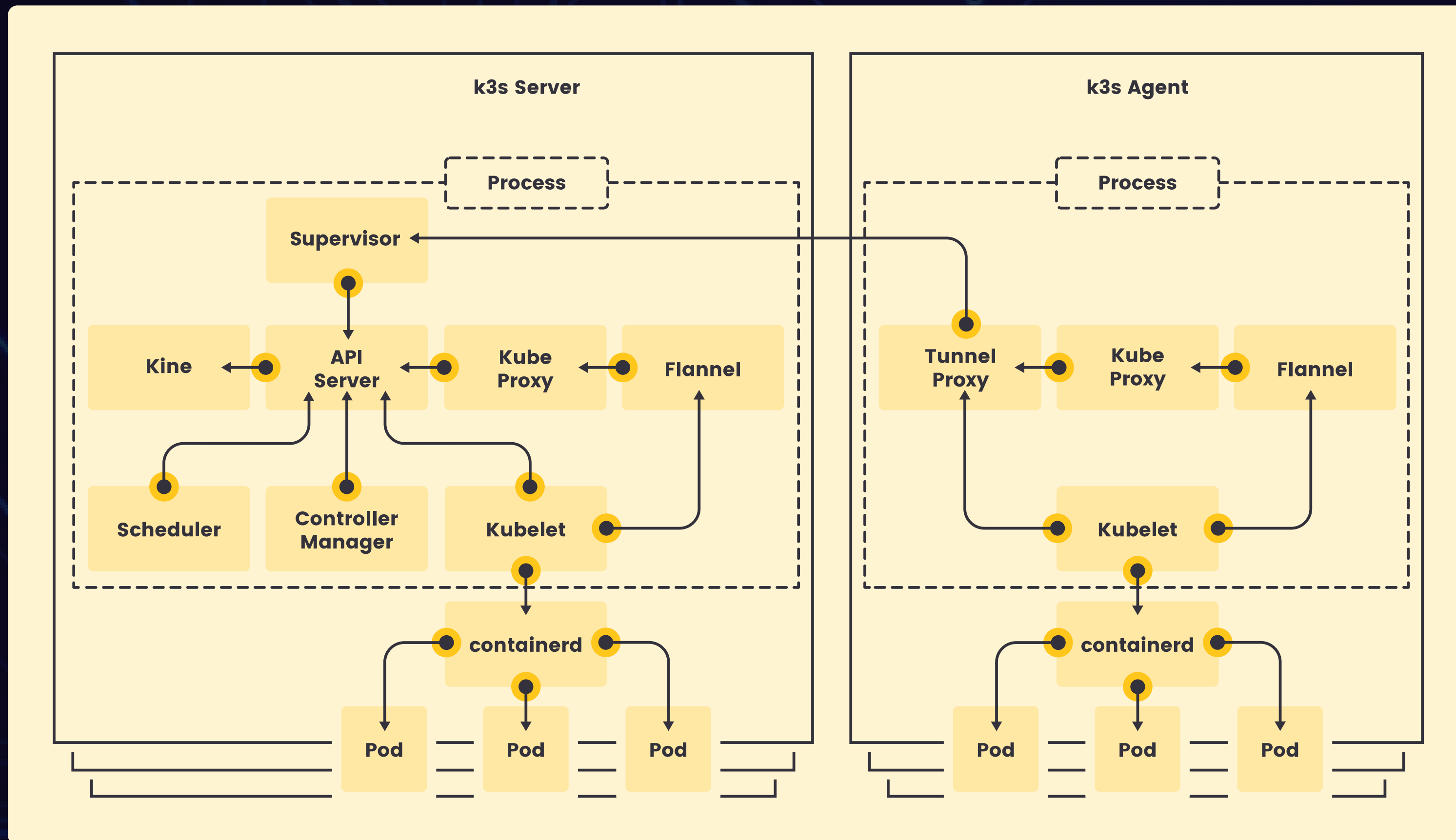
- replicas: $N \rightarrow$ Sobre N instâncias da aplicação
- Autoscaler pode aumentar/diminuir pods conforme uso de CPU ou RAM

ALTA DISPONIBILIDADE

- Se um pod morrer, é recriado
- Se um node cair, realoca pods automaticamente

GERENCIAMENTO

- Logs centralizados
- Service discovery automático
- Rede interna única entre serviços
- Load balancing nativo



ARQUITETURA K3S - CONCEITOS

POD

- Menor unidade executável
- Contém 1 ou mais containers

SERVICE

- Cria um IP estável para acessar pods
- K3s também possui balanceamento de carga interno

DEPLOYMENT

- Blueprint da aplicação
- Garante número de réplicas
- Atualiza e replica updates automaticamente

NODEPORT

- Expõe um serviço do cluster em uma porta do node (30000-32767)

ARQUITETURA K3S - SERVER

API SERVER

- Porta de entrada do cluster
- Recebe comandos via kubectl e gerencia objetos kubernetes

SCHEDULER

- Decide em qual node cada pod vai rodar
- Mesmo em um único server, gerencia os pods

CONTROLLER MANAGER

- Mantem o estado desejado do cluster
- Garante réplicas
- Reinicia pods
- Atualiza deployments

DATASTORE

- Armazena informações sobre o cluster
- Padrão: SQLite
- High Availability: MySQL, Postgres ou etcd externo

ARQUITETURA K3S - AGENT

KUBELET

- Agente que executa os pods em um node

CONTAINERD

- Substitui o Docker
- Mais leve e otimizado para clusters pequenos

KUBE-PROXY

- Regras de rede e roteamento
- Conecta Services e Pods

FLANNEL + VXLAN

- Rede interna dos pods
- VXLAN sobre UDP 8472
- Cada node recebe um range da rede do cluster



FERRAMENTAS DE MONITORAMENTO

01

K9s

Terminal para navegar no cluster

02

Lens/OpenLens

Dashboard com Visualização completa de pods, serviços e logs

03

Prometheus + Grafana

Coleta métricas do cluster e fornece dashboards visuais com CPU, memória e erros

04

Kube-state-metrics

Informações sobre estado de deployments, nodes e pods

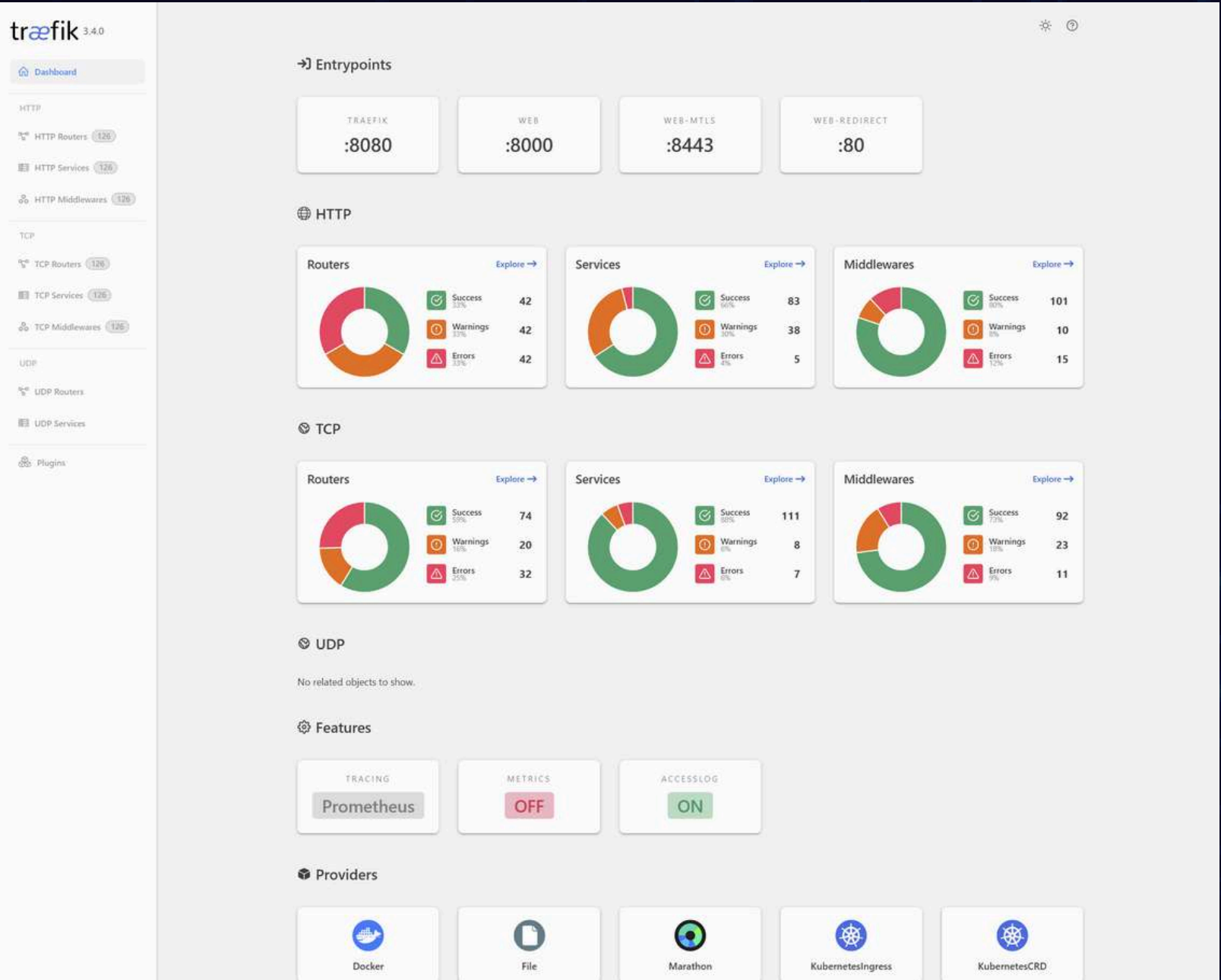
05

Traefik

Ingress controller nativo do k3s, inclui métricas e dashboards



FERRAMENTAS DE MONITORAMENTO



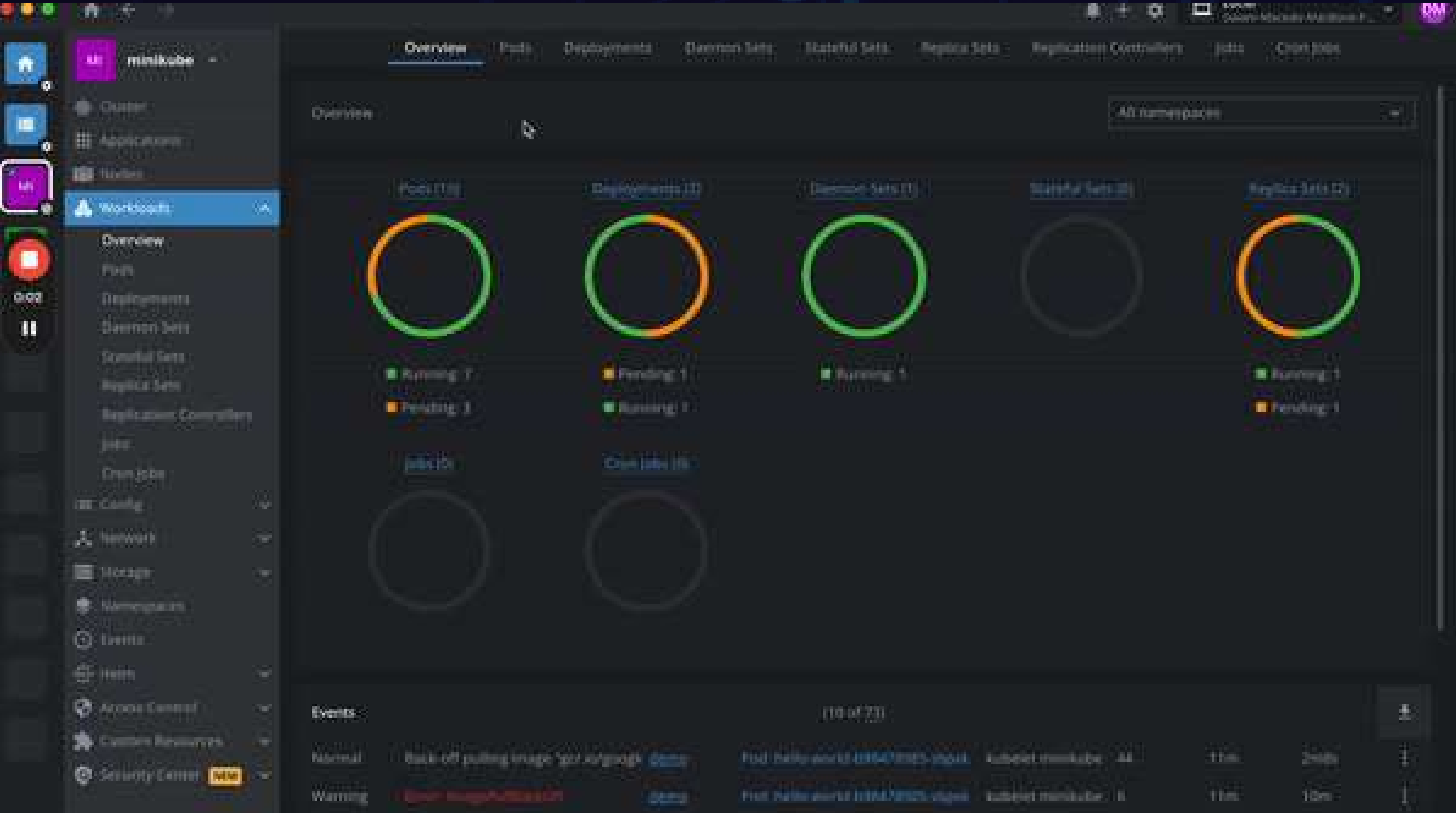
Traefik



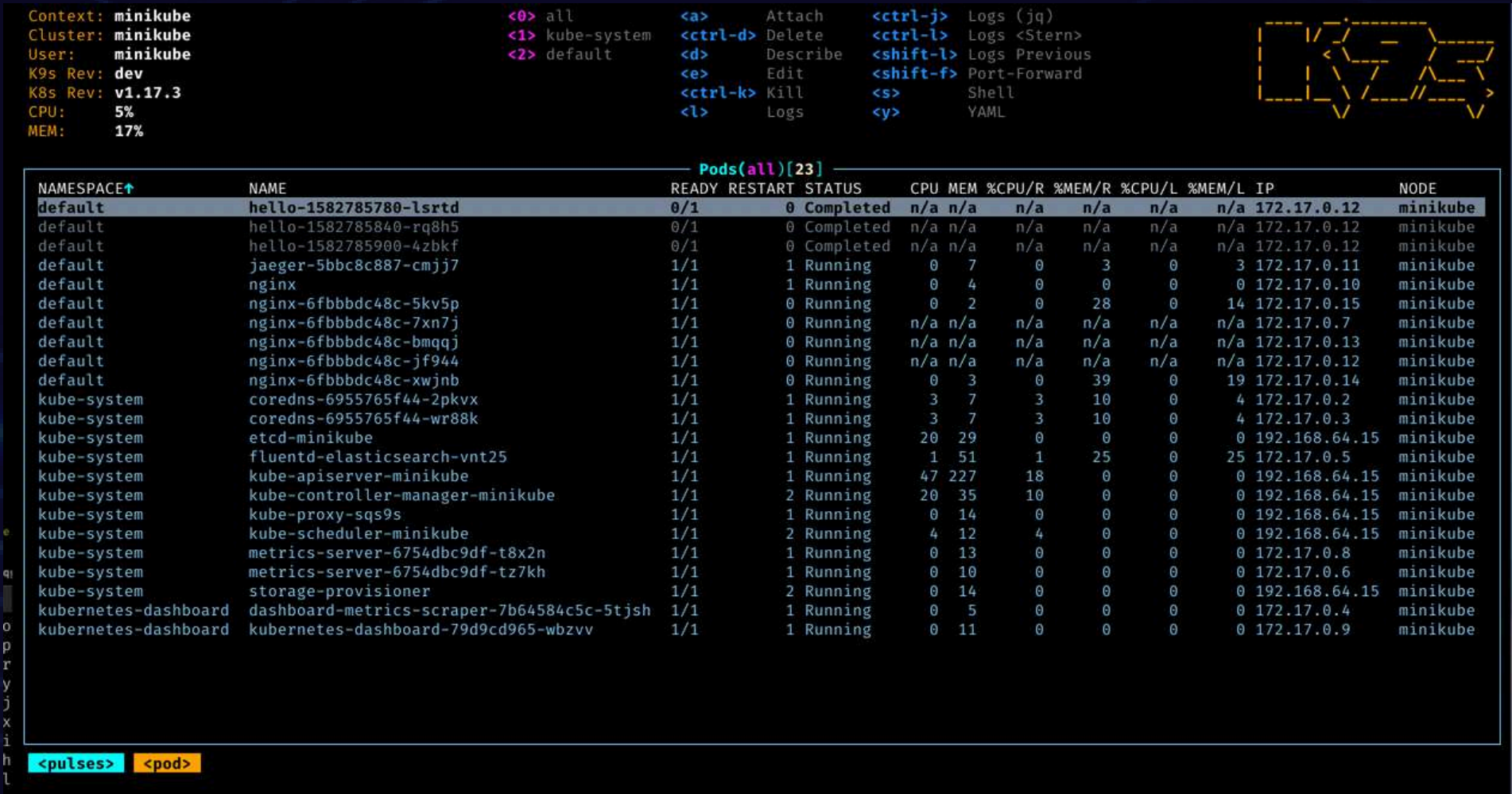
Prometheus



FERRAMENTAS DE MONITORAMENTO



Lens



K9s

K8S VS K3S



K8s

- Completo, pesado, mais componentes
- etcd como datastore
- exige 2-3 control planes para alta disponibilidade
- Maior custo de RAM e CPU
- Foco em datacenters e nuvens grandes



K3s

- Kubernetes mais leve
- SQLite como datastore
- Binário único
- Ideal para clusters pequenos
- Alta compatibilidade com kubernetes
- Menor consumo de RAM e CPU



VALEU!!



Referências:

- [youtube.com](https://www.youtube.com)
- [Rancher](https://rancher.io)
- k3s.io