

DevOps

Gledson Scotti

Introdução ao Kubernetes

Objetivo da Aula:

- Entender o que é Kubernetes e sua importância na orquestração de contêineres.
- Conhecer os principais componentes e arquitetura do Kubernetes.
- Explorar casos de uso reais e realizar exercícios práticos.

O que é ?

- Kubernetes (K8s) é uma plataforma de código aberto para orquestração de contêineres.
- Automatiza implantação, escalonamento e gerenciamento de aplicações containerizadas.
- Criado pelo Google em 2014, agora mantido pela Cloud Native Computing Foundation (CNCF).

Por que usar?

- Escalabilidade horizontal.
- Alta disponibilidade.
- Gerenciamento eficiente de recursos.

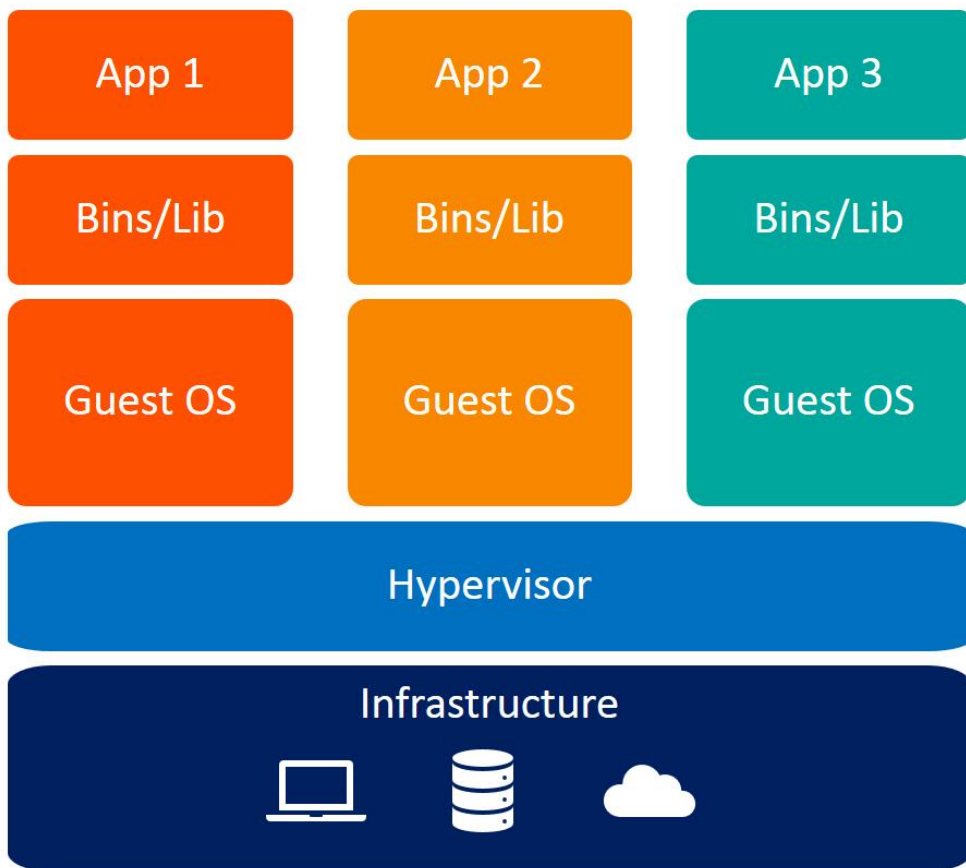
Por que Contêineres?

- Contêineres empacotam aplicações com suas dependências.
- São leves, portáteis e consistentes entre ambientes (desenvolvimento, teste, produção).

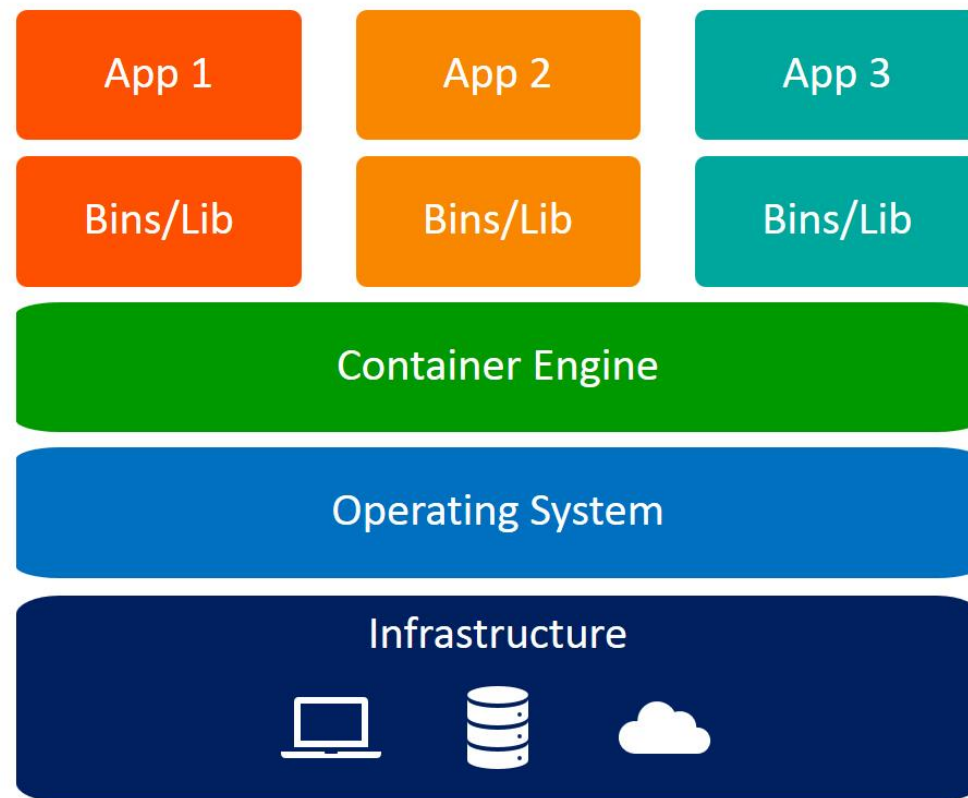
Comparação com VMs:

- Contêineres compartilham o kernel do SO, enquanto VMs têm SO completo.
- Mais rápidos e eficientes em termos de recursos.

Introdução ao Kubernetes



Virtual Machines



Containers

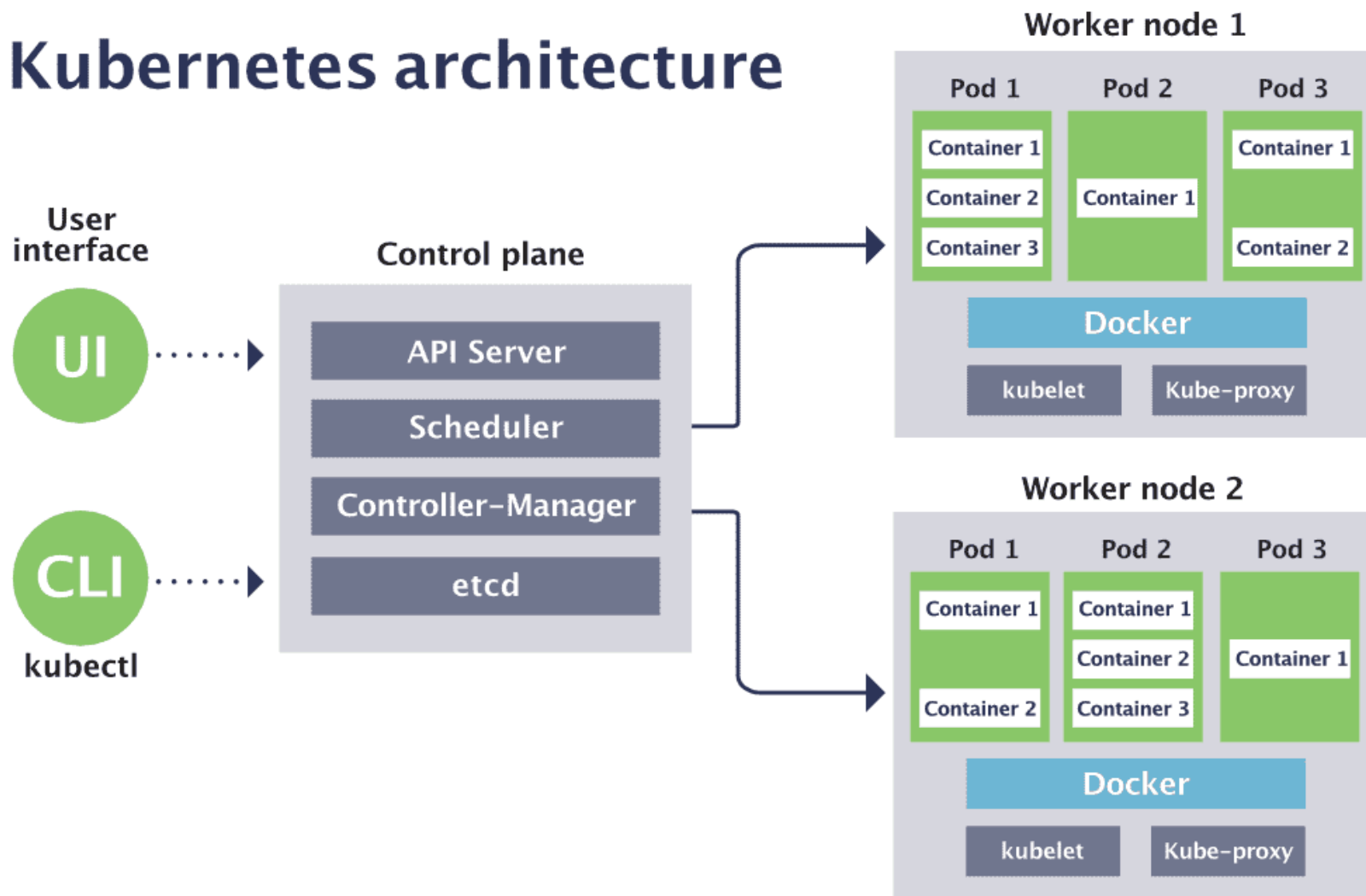
Master

- Máquina que controla os Nodes (Nós);
 - API Server: Interface de comunicação;
 - etcd: Banco de dados chave-valor para estado do cluster;
 - Controller Manager: Garante o estado desejado;
 - Scheduler: Aloca pods nos nós.
- Responsável pelas atribuições de tarefas aos Nodes.

Nodes

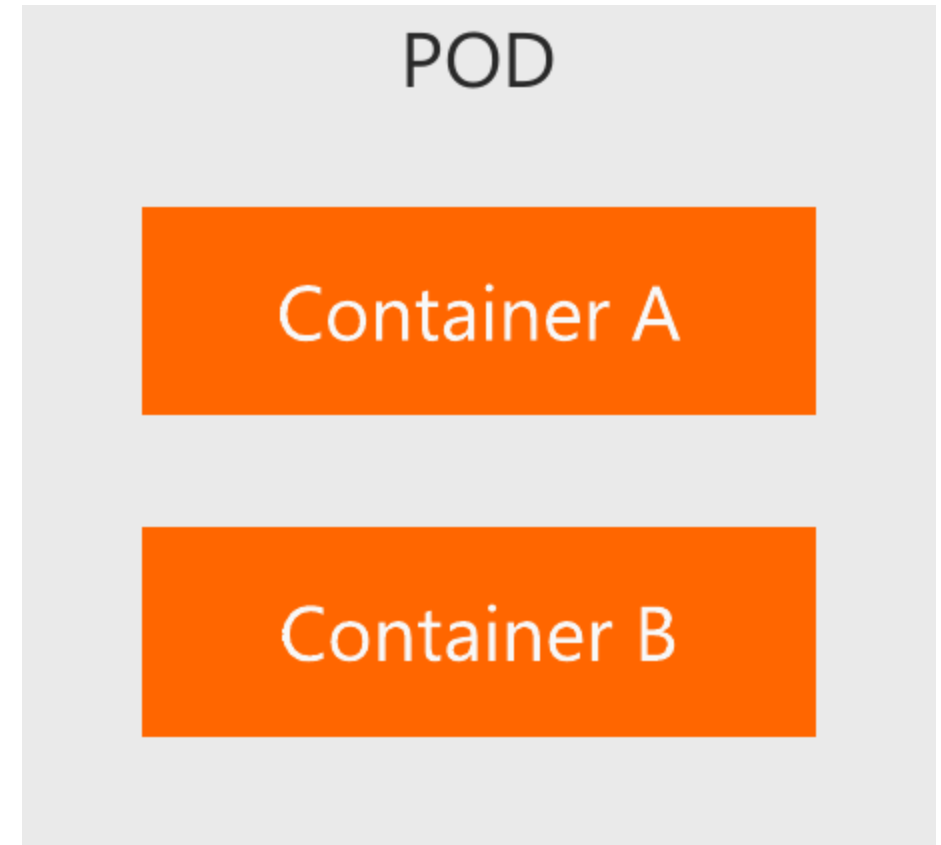
- Máquina que realizam as tarefas atribuídas pelo Master
 - Kubelet: Agente que gerencia contêineres no nó;
 - Kube-proxy: Gerencia rede e balanceamento de carga;
 - Container Runtime (ex.: Docker, containerd): Executa contêineres.

Kubernetes architecture



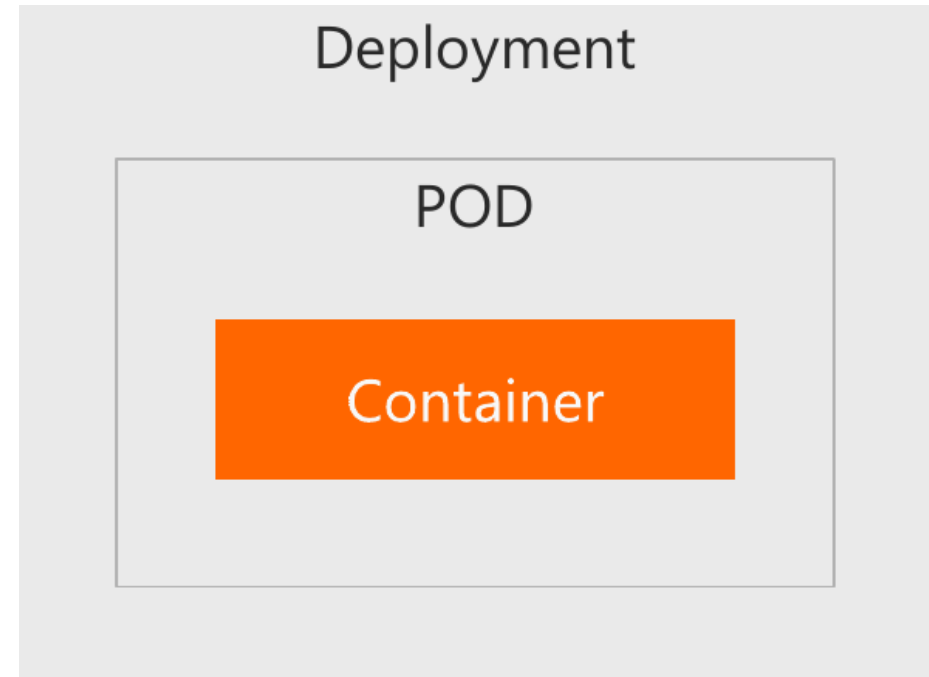
Pod

- Grupo de um ou mais containers implantados em um Node (Nó)
- Compartilham o mesmo endereço IP, IPC, nome do host e outros recursos



Deployment

- Abstração de um Pod com recursos adicionais;
- Conta com gerenciamento de estados.





Kubernetes Objetos Principais

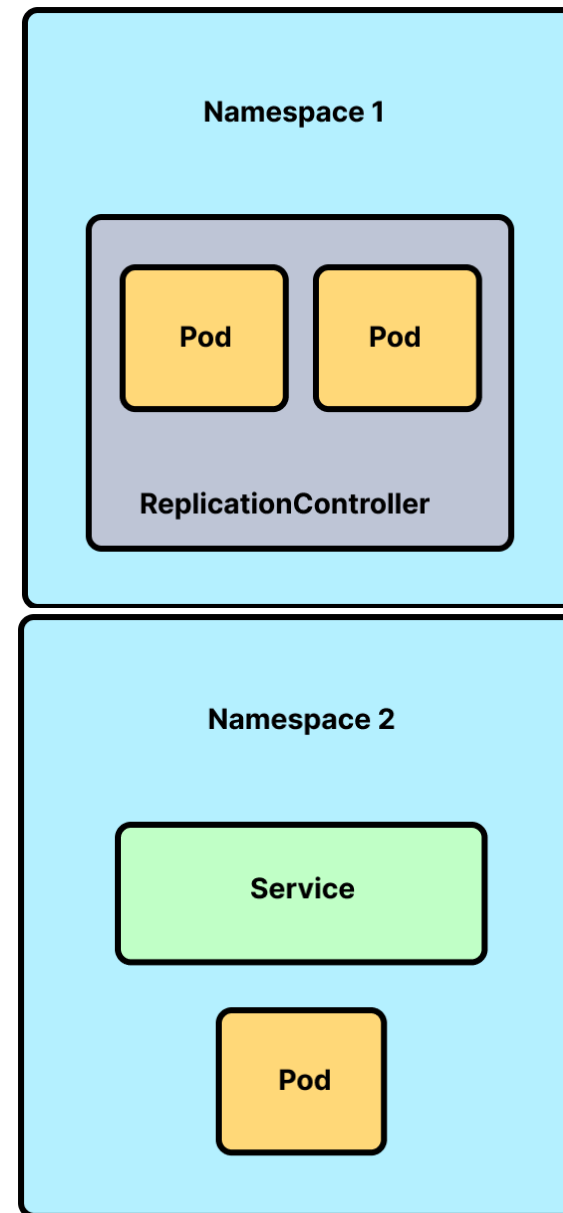
Service

- Objeto mais estável (Pods são criados ou removidos continuamente)
- Cuidará do acesso aos Pods, funcionando como um Load Balancer



Namespace

- Divide logicamente o cluster.
- Útil para:
 - Separar projetos/equipes.
 - Ambientes (dev, staging, prod).





Kubernetes Fluxo de Trabalho

1. **Criação do Manifesto:** O desenvolvedor cria um manifesto **YAML/JSON** que define o estado desejado do aplicativo ou recurso.
2. **Envio ao API Server:** O manifesto é enviado ao API Server do Kubernetes utilizando o comando `kubectl`.
3. **Alocação de Pods:** O Scheduler analisa os requisitos e aloca os pods aos nós disponíveis no cluster.
4. **Manutenção do Estado:** O Controller Manager monitora continuamente o cluster para garantir que o estado atual corresponda ao estado desejado.
5. **Gerenciamento de Contêineres:** O Kubelet, presente em cada nó, gerencia a execução e o estado dos contêineres conforme definido nos pods.



Kubernetes Casos de Uso

Microserviços: O que é? Em vez de construir uma aplicação gigante e monolítica, dividimos ela em pequenos serviços independentes (ex.: um serviço para login, outro para carrinho de compras, outro para pagamentos). Cada serviço roda em seu próprio contêiner.

Como o Kubernetes ajuda? Ele gerencia esses serviços, garantindo que cada um esteja funcionando, escalando (adicionando mais cópias) ou lidando com falhas automaticamente.

Exemplo prático: Um site de e-commerce onde o serviço de catálogo de produtos é separado do serviço de checkout. O Kubernetes coordena a comunicação entre eles.

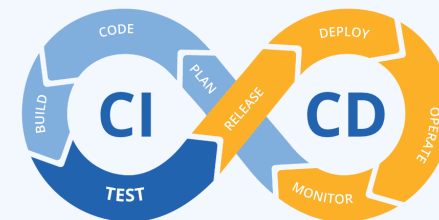


Kubernetes Casos de Uso

CI/CD (Integração Contínua/Entrega Contínua): O que é? CI/CD é um processo para automatizar o desenvolvimento e a entrega de software. Por exemplo, sempre que um desenvolvedor atualiza o código, o sistema testa e implanta automaticamente a nova versão.

Como o Kubernetes ajuda? Ele suporta pipelines de CI/CD, permitindo que novas versões de aplicações sejam implantadas rapidamente em contêineres, sem tempo de inatividade.

Exemplo prático: Usando ferramentas como Jenkins ou GitLab CI, você atualiza uma API e o Kubernetes implanta a nova versão em segundos.



Aplicações Stateless e Stateful:

Stateless (sem estado): O que é? São aplicações que não armazenam dados entre requisições. Cada pedido é independente (ex.: uma API que retorna dados do clima).

Exemplo no Kubernetes: APIs RESTful, como uma aplicação Flask (Python) ou um servidor web NGINX, são gerenciadas facilmente com o Kubernetes, já que não precisam de armazenamento persistente.

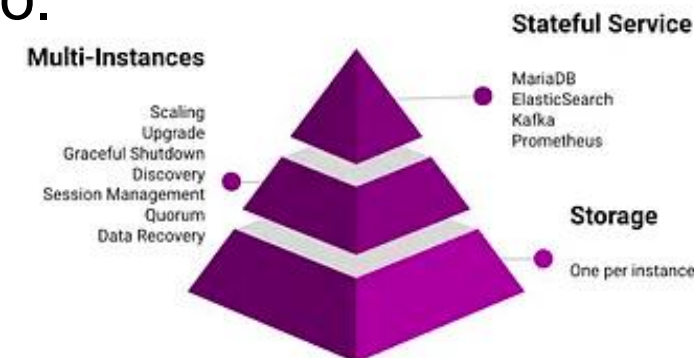


Aplicações Stateless e Stateful:

Stateful (com estado): O que é? São aplicações que precisam guardar dados, como bancos de dados (ex.: MySQL, MongoDB).

Como o Kubernetes ajuda? Usa StatefulSets, um recurso que garante que cada instância do banco de dados tenha uma identidade única e armazenamento persistente.

Exemplo prático: Um banco MySQL rodando em um cluster Kubernetes, onde os dados são salvos mesmo se o contêiner for reiniciado.



Machine Learning:

O que é? Treinar modelos de inteligência artificial (ex.: redes neurais) ou fazer inferências (usar o modelo treinado para previsões) exige muitos recursos computacionais.

Como o Kubernetes ajuda? Ele gerencia clusters escaláveis, distribuindo tarefas de treinamento ou inferência entre vários computadores (nós).

Exemplo prático: Treinar um modelo de reconhecimento de imagens em um cluster Kubernetes, onde cada nó processa uma parte dos dados.

kubectl: É a ferramenta de linha de comando (CLI) para interagir com o Kubernetes. Pense nela como um "controle remoto" do cluster. Comandos como *kubectl get pods* mostram os contêineres rodando, enquanto *kubectl apply* implanta novas aplicações.

Minikube: ambiente Kubernetes que roda localmente no seu computador, sem precisar de um cluster na nuvem, ideal para aprendizado ou testes.

Helm: Gerenciador de pacotes para K8s. Facilita a instalação de aplicações complexas. Instalar o WordPress no Kubernetes com um comando simples: *helm install meu-blog wordpress*.

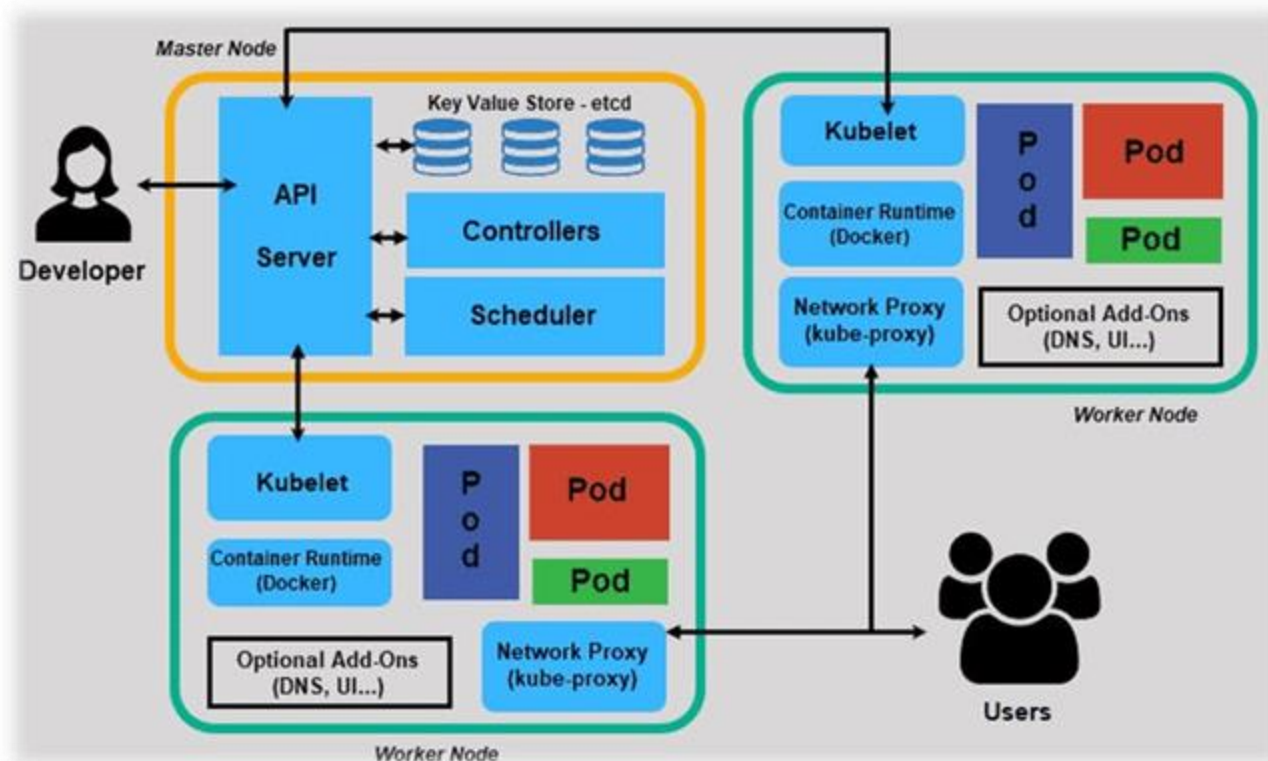
Prometheus + Grafana: São ferramentas de monitoramento. O Prometheus coleta métricas (ex.: uso de CPU, memória) do cluster, e o Grafana exibe essas métricas em gráficos bonitos e fáceis de entender.

Istio: Uma ferramenta para gerenciar o tráfego de rede entre microserviços, além de adicionar segurança e observabilidade. Controla como os serviços se comunicam (ex.: balanceamento de carga, retries automáticos) e protege com políticas de segurança (ex.: criptografia). Istio para redirecionar 10% do tráfego de um site para uma nova versão da aplicação, testando-a antes de liberar para todos.



Kubernetes Conclusão

É uma ferramenta poderosa e indispensável para o gerenciamento de aplicações modernas, oferecendo uma plataforma robusta para orquestrar contêineres em ambientes escaláveis e distribuídos. Mais do que uma tecnologia, é um padrão que capacita inovação e transforma a maneira como construímos e gerenciamos aplicações.



ATIVIDADES NO AVA.