

DevOps Módulo 14 - IT Talent

Escrito por: [Gabriel Oliveira dos Santos](#)

Github: <https://github.com/Hypothesis>

Kubernetes

O que é Kubernetes?

- Kubernetes, abreviado como K8s, é uma ferramenta de orquestração de contêineres de código aberto.
- Automação de implantação, escalonamento e operação de aplicativos containerizados.



O Docker e o Kubernetes são duas tecnologias diferentes com casos de uso diferentes. Você usa o Docker Desktop para executar, editar e gerenciar o desenvolvimento de contêineres. Você usa o Kubernetes para executar aplicações de nível de produção em grande escala.

Estrutura de Kubernetes

Estrutura do Kubernetes

- Sistema Distribuído:
 - Várias máquinas formam um cluster.
 - Pode incluir máquinas físicas e virtuais, on-premise ou na nuvem.

Podem rodar os containers on-premise (em locais físicos) e na nuvem também, e formam como um cluster (que a definição se resume a conjunto de servidores interconectados que atuam como um único sistema trabalhando juntos).

Algoritmos de Agendamento

Algoritmos de Agendamento

- Agendamento Inteligente:
 - Considera recursos disponíveis, prioridade e outras restrições.
 - Move contêineres conforme necessário.



Considera recursos disponíveis, com critérios, a serem executados, como se fossem um escalador de um SO, que escolhem o processo com vários tipos de prioridades.

Compatibilidade com Runtimes

Compatibilidade com Runtimes

- Agnóstico em relação ao Runtime:
 - Suporta diferentes runtimes de contêineres (Docker, Rocket).

O kubernetes podem ser executados com o docker, ou com outra aplicação que roda um container.

Características

Adaptação e Adoção

- Design Modular:
 - Facilitou a adoção ampla.
 - Configuração declarativa para deploy rápido e fácil.



Fácil e amplo, fez com que o Kubernetes ganha-se espaço no mundo tech de forma massiva, mesmo sendo um open-source.

Recursos Destacados do Kubernetes

- Principais Recursos:
 - Rollout e rollback automáticos.
 - Escalonamento horizontal sem interrupções.
 - Gerenciamento de segredos.
 - Descoberta de serviços e balanceamento de carga.
 - Suporte para contêineres Linux e Windows.
 - Coleta de logs.
 - Suporte a aplicativos stateful.
 - Gerenciamento de volumes persistentes.
 - Cotas de CPU e memória.
 - Processamento de jobs em batch.
 - Controle de acesso baseado em funções.
-

Recursos para Equipes de Operação

Recursos para Equipes de Operação

- Autocura e Manutenção:
 - Move contêineres de máquinas com falhas.
 - Recursos integrados para manutenção.
-

O Kubernetes podem mover contêineres de máquinas com falhas, para recursos de manutenção, assim automatizando o processo para gerenciar os contêineres.

Federação de Clusters

Federação de Clusters

- Redundância:
 - Múltiplos clusters podem se unir.
 - Contêineres se movem automaticamente entre clusters.

Quando um cluster morre, ou tem algum erro nos servidores, o próprio Kubernetes faz com que essa aplicação seja direcionada a outro cluster.

Cluster Kubernetes c/ um nó

Cluster de Nó Único

- Implantação em Nó Único:
 - Ideal para desenvolvimento e teste.
 - Docker para Mac e Docker para Windows suportam Kubernetes localmente.
 - Certifique-se de que o Kubernetes está habilitado nas configurações.

Minikube e Kubeadm

- Outras Opções:
 - Minikube: Suporta Linux, Mac e Windows.
 - Kubeadm: Configura clusters de nó único em sistemas Linux, instala o Kubernetes no sistema.

Uso em Integração Contínua

- Clusters de Nó Único em CI:
 - Criar clusters efêmeros rapidamente.
 - Kubernetes em Docker (K-in-D ou kind) para pipelines de integração contínua.



Um nó é uma máquina, quando falamos de um único nó, é a nossa aplicação que roda inteiramente em um único servidor, uma única máquina.

Cluster de Múltiplos Nós

Cluster de Múltiplos Nós

- Implantação em Múltiplos Nós:
 - Ideal para cargas de trabalho de produção.
 - Aproveita o escalonamento horizontal e a tolerância a falhas.

A configuração múltiplos nós é quando vários computadores (nós) rodam a nossa aplicação, assim aumentando a escalabilidade horizontal.

Controle vs. Manutenção

- Questões Importantes:
 - Quanto controle você quer sobre o cluster?
 - Quanto esforço está disposto a investir na manutenção?

Saber qual configuração usar, é necessário ficar ciente que podem ser mais caros, por conta dos provedores de servidores.

Arquitetura do Kubernetes

Visão Geral do Kubernetes

- Kubernetes:
 - Kubernetes é um sistema distribuído.
 - Introduz seu próprio dialeto no espaço de orquestração.
 - Vocabulário:
 - Internalizar o vocabulário é crucial para o sucesso.
 - Glossário disponível.
-

Entendendo a Arquitetura

- Importância da Arquitetura:
 - Necessário para compreender como os recursos funcionam.
 - O cluster Kubernetes é o nível mais alto de abstração.
-

É necessário saber o vocabulário certo do kubernetes para usar-lo certo, assim para ter uma compreensão básica da arquitetura kubernestes.

Estrutura do Cluster

- Cluster Kubernetes:
 - Composto por nós (nodes).
 - Nós podem ser VMs ou máquinas físicas.
 - Nós são categorizados como nós de trabalho (worker nodes) e nós mestres (master nodes).
-

Nós são categorizados como nós de trabalhos e nós mestres, que são gerenciados pelo plano de controle, esses nós podem ser maquinas virtuais ou maquinas físicas.

Plano de controle

Plano de Controle do Kubernetes

- Plano de Controle:
 - Executado nos nós mestres.
 - Conjunto de APIs e softwares (componentes mestres).
 - Interação dos usuários do Kubernetes.
-

Plano de controle, são conjuntos de APIs e softwares que participam de um plano de controle, e esses conjuntos são chamados de componentes mestres.

Agendamento de Contêineres

- Agendamento:
 - O plano de controle agenda contêineres nos nós.
 - Semelhante ao agendamento de processos pelo Kernel na CPU.
 - Decisão de colocar contêineres nos nós conforme requisitos de computação.

O plano de controle agenda os contêineres dos nós, e esse agendamento não está ligado ao tempo, porém é o agendamento de acesso ao kernel, muito semelhante ao escalonador de um SO.

Pods

Pods no Kubernetes

- Pods:
 - Contêineres são agrupados em Pods.
 - Pod é o menor bloco de construção no Kubernetes.
 - Todos os contêineres em um Pod executam no mesmo nó.

Pods são contêineres, que são o menor bloco de construção no Kubernetes.

Serviços e Implantações

- Serviços e Implantações:
 - Serviços definem regras de rede para expor Pods.
 - Kubernetes usa implantações para gerenciar configuração e escalonamento horizontal dos Pods.
-

Comandos Básicos do Kube

Comandos Básicos do Kubectl

- Comandos Básicos:
 - kubectl create: Cria novos recursos do Kubernetes.
 - kubectl delete: Exclui recursos específicos.
 - kubectl get: Retorna uma lista de recursos de um tipo especificado.
 - kubectl describe: Imprime informações detalhadas sobre recursos.
 - kubectl logs: Imprime logs do contêiner.
-

Alguns comandos que mais utilizados para serem usados no Kubernetes

Painel Web

Painel Web do Kubernetes

- Painel Web:
 - Oferece dashboards e visualizações de recursos do cluster.
 - Abordado no laboratório Deploy Stateful Application in a Kubernetes Cluster.
 - Opcional, nem todos os clusters possuem.



A pagina web, é uma forma mais clara e bonita de vermos como está rodando o nosso cluster.