



Curso Bônus

Planejando Sua Carreira Para as
Profissões do Futuro

Arquitetura do Kubernetes

Depois de passar pelos 6 Labs deste curso, agora vai ficar bem mais fácil compreender o que é o Kubernetes.

O Kubernetes é um sistema de gerenciamento de containers de código aberto que ajuda a implantar, escalar e gerenciar aplicativos em containers. Ele foi projetado para automatizar muitas das tarefas de gerenciamento de containers, como implantação, escalonamento e recuperação de falhas.

A arquitetura do Kubernetes é baseada em um conjunto de componentes que trabalham juntos para gerenciar um cluster de máquinas virtuais ou físicas e executar aplicativos em containers. Aqui está uma visão geral da arquitetura do Kubernetes:

Máquinas do cluster: um cluster Kubernetes é composto por um conjunto de máquinas virtuais ou físicas que são gerenciadas pelo Kubernetes. Cada máquina no cluster é chamada de nó e pode ser um nó mestre ou um nó de trabalho.

Nó mestre: o nó mestre é responsável por gerenciar o cluster. Ele tem vários componentes que trabalham juntos para controlar os nós de trabalho e garantir que os pods e os containers estejam sendo executados corretamente. Alguns dos componentes do nó mestre incluem:

- **kube-apiserver:** expõe um ponto de extremidade RESTful (endpoint) para que os aplicativos possam interagir com o cluster.
- **etcd:** é um banco de dados distribuído que armazena os estados do cluster.
- **kube-scheduler:** seleciona qual nó de trabalho um pod deve ser executado e atribui o pod ao nó.
- **kube-controller-manager:** gerencia os controladores, que são responsáveis por garantir que o estado desejado do cluster seja alcançado.

Nó de trabalho: os nós de trabalho são responsáveis por executar os pods e os containers. Cada nó de trabalho tem um agente chamado kubelet, que é responsável por gerenciar os pods no nó. Além disso, cada nó de trabalho tem um container runtime, como o Docker, que é responsável por executar os containers em si.

Pods: um pod é a unidade básica de execução no Kubernetes. Ele é composto por um ou mais containers que compartilham um espaço de endereçamento de rede e um sistema de arquivos. Os pods são geralmente usados para executar aplicativos que dependem de uma comunicação muito próxima entre os containers.

Serviços: um serviço é um ponto de extremidade de rede que é responsável por redirecionar o tráfego para um conjunto de pods. Isso permite que os aplicativos se comuniquem com outros aplicativos e com os usuários finais de maneira consistente, independentemente da localização dos pods.

Deployments: embora os pods sejam a unidade básica de computação no Kubernetes, eles normalmente não são lançados diretamente em um cluster. Em vez disso, os pods geralmente são gerenciados por mais uma camada de abstração: o deployment. O objetivo principal de um deployment é declarar quantas réplicas de um pod devem estar em execução por vez. Quando uma implantação é adicionada ao cluster, ela automaticamente aumenta o número solicitado de pods e os monitora. Se um pod morre, o deployment irá recriá-lo automaticamente.

Considere o link abaixo para uma visão geral do Kubernetes e seus principais componentes:

<https://www.cienciaedados.com/kubernetes-pods-nodes-containers-e-clusters/>

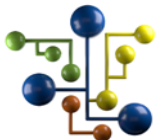
Aqui está um resumo do funcionamento do Kubernetes:

- O desenvolvedor cria um aplicativo baseado em containers e o empacota em uma imagem de container.
- A imagem de container é carregada em um registro de container, como o Docker Hub.
- O desenvolvedor cria um arquivo de configuração do Kubernetes chamado manifesto, que descreve como o aplicativo deve ser implantado e gerenciado no cluster.
- O desenvolvedor envia o manifesto para o cluster do Kubernetes usando o comando `kubectl apply`.

Cursos de Aperfeiçoamento Profissional - Bônus da Formação

- O Kubernetes lê o manifesto e cria os objetos necessários para implantar o aplicativo, como pods e serviços.
- O Kubernetes gerencia o aplicativo, implantando novas réplicas de pods quando necessário e recuperando automaticamente os pods que falharem.
- Os usuários finais acessam o aplicativo através de um serviço do Kubernetes, que redireciona o tráfego para os pods apropriados.

O Kubernetes oferece muitas outras funcionalidades, como balanceamento de carga, rollouts e rollbacks de atualizações, gerenciamento de storage e muito mais. Isso o torna uma ferramenta poderosa para gerenciar aplicativos em containers em uma escala maior.



Equipe DSA

Muito Obrigado!
Continue Trilhando Uma Excelente Jornada de Aprendizagem.