

O que é um cluster Kubernetes?

Um cluster Kubernetes é um conjunto de nodes que trabalham juntos para executar todos os nossos pods. Um cluster Kubernetes é composto por nodes que podem ser tanto control plane quanto workers. O control plane é responsável por gerenciar o cluster, enquanto os workers são responsáveis por executar os pods que são criados no cluster pelos usuários.

Quando estamos pensando em um cluster Kubernetes, precisamos lembrar que a principal função do Kubernetes é orquestrar containers. O Kubernetes é um orquestrador de containers, sendo assim quando estamos falando de um cluster Kubernetes, estamos falando de um cluster de orquestradores de containers. Eu sempre gosto de pensar em um cluster Kubernetes como se fosse uma orquestra, onde temos uma pessoa regendo a orquestra, que é o control plane, e temos as pessoas musicistas, que estão executando os instrumentos, que são os workers.

Sendo assim, o control plane é responsável por gerenciar o cluster, como por exemplo:

- Criar e gerenciar os recursos do cluster, como por exemplo, namespaces, deployments, services, configmaps, secrets, etc.
- Gerenciar os workers do cluster.
- Gerenciar a rede do cluster.
- O etcd desempenha um papel crucial na manutenção da estabilidade e confiabilidade do cluster. Ele armazena as informações de configuração de todos os componentes do control plane, incluindo os detalhes dos serviços, pods e outros recursos do cluster. Graças ao seu design distribuído, o etcd é capaz de tolerar falhas e garantir a continuidade dos dados, mesmo em caso de falha de um ou mais nós. Além disso, ele suporta a comunicação segura entre os componentes do cluster, usando criptografia TLS para proteger os dados.
- O scheduler é o componente responsável por decidir em qual nó os pods serão executados, levando em consideração os requisitos e os recursos disponíveis. O scheduler também monitora constantemente a situação do cluster e, se necessário, ajusta a distribuição dos pods para garantir a melhor utilização dos recursos e manter a harmonia entre os componentes.
- O controller-manager é responsável por gerenciar os diferentes controladores que regulam o estado do cluster e mantêm tudo funcionando. Ele monitora constantemente o estado atual dos recursos e compara-os com o estado desejado, fazendo ajustes conforme necessário.

- Onde está o api-server é o componente central que expõe a API do Kubernetes, permitindo que outros componentes do control plane, como o controller-manager e o scheduler, bem como ferramentas externas, se comuniquem e interajam com o cluster. O api-server é a principal interface de comunicação do Kubernetes, autenticando e autorizando solicitações, processando-as e fornecendo as respostas apropriadas. Ele garante que as informações sejam compartilhadas e acessadas de forma segura e eficiente, possibilitando uma colaboração harmoniosa entre todos os componentes do cluster.

Já no lado dos workers, as coisas são bem mais simples, pois a principal função deles é executar os pods que são criados no cluster pelos usuários. Nos workers nós temos, por padrão, os seguintes componentes do Kubernetes:

- O kubelet é o agente que funciona em cada nó do cluster, garantindo que os containers estejam funcionando conforme o esperado dentro dos pods. Ele assume o controle de cada nó, garantindo que os containers estejam sendo executados conforme as instruções recebidas do control plane. Ele monitora constantemente o estado atual dos pods e compara-os com o estado desejado. Caso haja alguma divergência, o kubelet faz os ajustes necessários para que os containers sigam funcionando perfeitamente.
- O kube-proxy, que é o componente responsável fazer ser possível que os pods e os services se comuniquem entre si e com o mundo externo. Ele observa o control plane para identificar mudanças na configuração dos serviços e, em seguida, atualiza as regras de encaminhamento de tráfego para garantir que tudo continue fluindo conforme o esperado.
- Todos os pods de nossas aplicações.