# Arquitetura do Kubernetes

## Master

**API** – Responsável por receber e executar os novos comandos, manter a comunicação entre todos os componentes, gerenciar os recursos do cluster podendo através dela criar um pod, editar um replica set, ler os dados de um deployment, deletar um volume, tudo isso sempre através da API.

Quem expõem a API do Kubernetes é o **kube-apiserver**, ele é o front-end para a camada de gerenciamento do Kubernetes.

A forma de se comunicar com a API é através da ferramenta **kubectl**, é através dela que pode-se criar, ler, atualizar e remover recursos do cluster, isso de duas maneiras.

1 – declarativa (arquivos yaml), 2 – imperativa (executar comandos).

**Controller Manager** – Responsável por manter e atualizar o estado desejado.

**kube-scheduler** – É um componente da camada de gerenciamento que observa os pods recém criados sem nenhum nó atribuído e seleciona o nó para executá-los, para isso diversos fatores são levados em consideração desde disponibilidade de hardware dos nós até políticas e restrições.

**ETCD** – Responsável por armazenar todos os dados do cluster K8S em um banco de dados chave: valor.

## Node

**Kubelet –** Responsável pela execução dos **PODs** dentro dos **NODEs**.

**Kube-proxy –** Responsável pela comunicação dos **NODEs** dentro do **Cluster**.

