Universidade Federal de Uberlândia
Faculdade de Engenharia de Engenharia Elétrica - FEELT
Engenharia de Computação

Orquestração de Containers

Prof. Márcio José da Cunha

16/09/2021

Section 1

Orquestração de Containers - Introdução

Orquestração de Containers

- Orquestração de Containers

A orquestração de containers é um processo que automatiza a implantação, o gerenciamento, dimensionamento e rede de um conjunto de containers

Seu foco é o gerenciamento do ciclo de vida dos containers

As empresas precisam implantar e gerenciar centenas ou milhares de containers e hosts

Orquestração de Containers

- Principais tarefas
- # Configuração e programação de containers
- # Provisionamento e implantação de containers
- # Redundância e disponibilidade
- # Ampliação e remoção de containers para distribuição de carga
- # Infraestrutura
- # Movimentação de containers entre hosts
- # Alocação de recursos
- # Balanceamento de carga

Section 2

Tipos de ferramentas de Orquestração de Containers

Docker Swarm

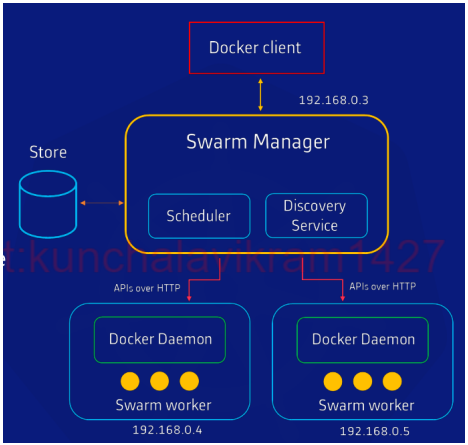
- Docker Swarm

Ferramenta de orquestração de containers open-source, nativo para a engine Docker

Qualquer ferramenta, serviço ou software que roda na engine Docker, roda igualmente no Swarm

Docker cli gerencia a criação, constrói aplicações e serviços, e gerencia o comportamento dentro de um Swarm

- Docker Swarm



Kubernets

- Kubernetes

É uma plataforma open-source, portável e extensiva para a gerência de cargas de trabalho e serviços distribuídos em containers

O K8s facilita tanto a funcionalidades declarativas quanto as de automação

Possui um ecossistema grande e de rápido crescimento

Serviços, suporte e ferramentas são amplamente disponíveis

O K8s é o resultado de um projeto da Google de mais de 15 anos, o projeto Borg.

Section 3

Necessidade de utilização do K8s

Necessidades

- O que o K8s pode fazer

Descoberta de serviço e balanceamento de carga

Orquestração de armazenamento

Lançamento e reversões automatizadas

Empacotamento binário automático

Autocorreção

Gerenciamento de configurações e chaves

Necessidades

- O que o K8s pode fazer

Descoberta de serviço e balanceamento de carga

  - Expõe um container utilizando o nome (DNS) ou pelo próprio endereço IP. Se o tráfego para o container for alto, o K8s faz o balanceamento de carga e distribui o tráfego da rede para que a implantação seja estável

Orquestração de armazenamento

  - Permite que seja montado automaticamente um sistema de armazenamento de sua escolha, como armazenamentos locais, provedores de nuvem, datacenters, dentre outros.

Necessidades

- O que o K8s pode fazer

Autocorreção

  - Caso algum container falhe, o K8s substitui os containers, elimina os que não responderam a verificação de integridade definida pelo usuário e avisa as aplicações

Gerenciamento de configurações e chaves

  - Permite armazenar e gerenciar informações confidenciais, como senhas, chaves, tokens, chaves. É possível implantar e atualizar segredos e configurações de aplicações sem reconstruir suas imagens de container e sem expor os segredos em sua pilha de configuração.

O que o K8s não faz

- O que o K8s não faz

Não limita os tipos de aplicações suportadas.

Não implanta código-fonte e não constrói sua aplicação.

Não fornece serviços em nível de aplicação, tais como middleware (por exemplo, barramentos de mensagem), estruturas de processamento de dados (por exemplo, Spark), bancos de dados (por exemplo, MySQL), caches, nem sistemas de armazenamento em cluster (por exemplo, Ceph), como serviços integrados.

Não dita soluções de logging, monitoramento ou alerta.

Section 4

Componentes do K8s

K8s Cluster

- K8s Cluster

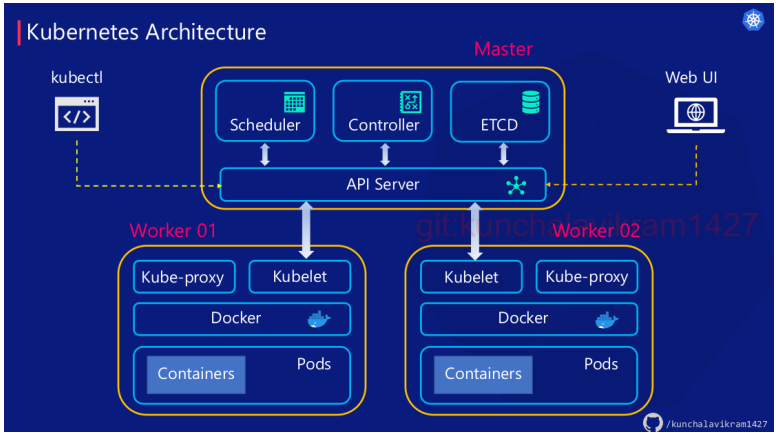
Um cluster consiste em um conjunto de servidores de processamento (máquinas físicas ou virtuais) necessários para executar os seus aplicativos no container. Cada máquina no cluster é chamado de nó

Um nó é a menor unidade de hardware existente no K8s, assim como uma máquina é em um datacenter.

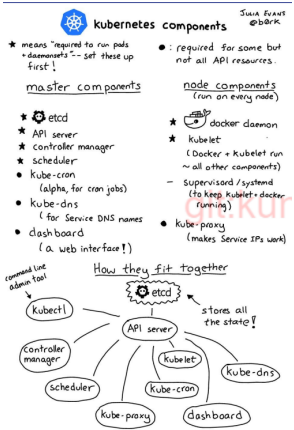
Existem dois tipos de nós:

- **Master node:** gerencia o cluster e hospeda os componentes do plano de controle do K8s
- **Worker node:** executa as suas aplicações containizadas

- K8s Cluster


Kubernetes Architecture
Master
kubectl
Web UI
Scheduler
Controller
ETCD
API Server
Worker 01
Worker 02
Kube-proxy
Kubelet
Docker
Containers
Pods
Kubelet
Kube-proxy
Docker
Containers
Pods
/kunchalavikram1427

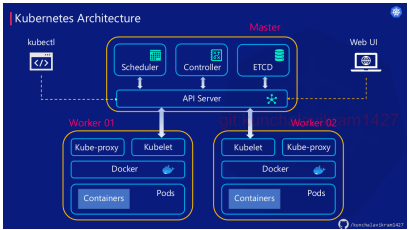
- K8s Cluster


kubernetes components
Julia Evans
@b0rk
★ means "required to run pods + daemonsets" -- set these up first!
● : required for some but not all API resources.
master components
★ etcd
★ API server
★ controller manager
★ scheduler
● kube-cron
(alpha, for cron jobs)
● kube-dns
(for Service DNS names
● dashboard
(a web interface!)
node components
(run on every node)
★ docker daemon
★ kubelet
(Docker + kubelet run ~ all other components)
– supervisord / systemd
(to keep kubelet + docker running)
● kube-proxy
(makes Service IPs work)
How they fit together
command line admin tool
Kubectl
etcd
stores all the state!
API server
Controller manager
Kubelet
kube-dns
scheduler
Kube-cron
Kube-proxy
dashboard

Master Node Components

- Master Node

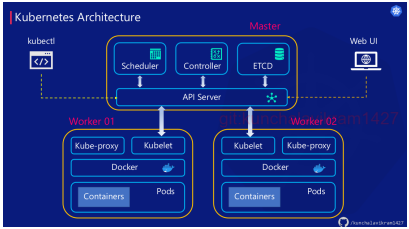
Responsável pelo gerenciamento completo do cluster
Acessível via CLI, GUI ou API
O Master supervisiona os nós no cluster e é responsável pela orquestração real de containers nos Workers
Para alcançar tolerância a falhas, pode haver mais de um nó mestre no cluster - configuração de cluster de alta disponibilidade
Possui 4 componentes: ETCD, scheduler, controller e servidor de API (formam o **Control Plane**)



Master Node Components

- API Server

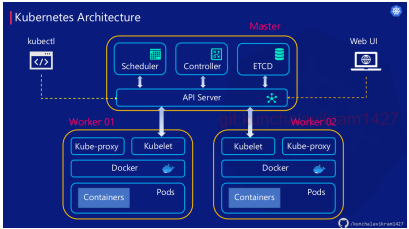
APIs para serem acessadas
O Master se comunica com os outros clusters por meio desse servidor
Aqui são validados os comandos REST executados pelo usuário



Master Node Components

- Controller Manager

Os controladores são os cérebros da orquestração
Eles são responsáveis avisar e responder quando os nós, containers ou endpoints são desligados. O controlador tomam decisões para entregar novos containers em alguns casos
Executa loops de controle que gerencia o estado do cluster, verificando se as implantações, réplicas e nós estão em execução no cluster



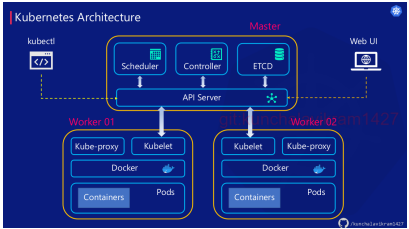
Master Node Components

- ETCD

Banco de dados do K8s, do tipo chave-valor, usado para armazenar todos os dados utilizados para gerenciar o cluster

Quando se tem vários nós e vários masters em um cluster, o ETCD armazena todas informações em todos os nós do cluster de forma distribuída

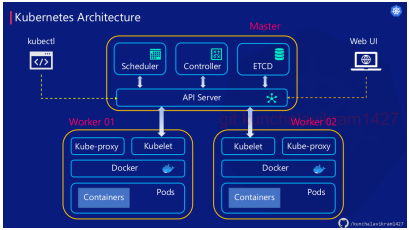
O ETCD é responsável por implementar locks dentro do cluster, garantindo que não existam conflitos entre os Masters



Master Node Components

- Scheduler

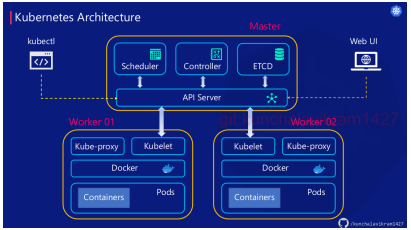
É responsável por distribuir o trabalho ou containers para vários nós
Verifica novos nós e os coloca no mesmo escalonamento das tarefas dos outros nós
Possui decisões de escalonamento, como, requisitos dos pods, restrições/políticas sobre utilização de hardware e software, regras, tolerâncias, etc...



Workers Node Components

- Kubelet

Os Workers possuem os agentes, os Kubelet, que são responsáveis pela interação com o mestre, fornecendo informações sobre os pods, health information de cada nó
Executa as ações solicitadas pelos masters para os workers
O Kubelet não gerencia containers que não foram criados pelo K8s

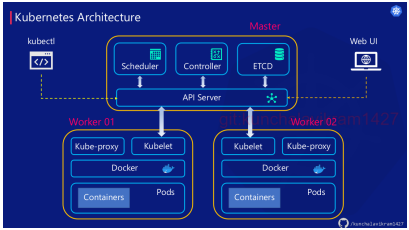


Workers Node Components

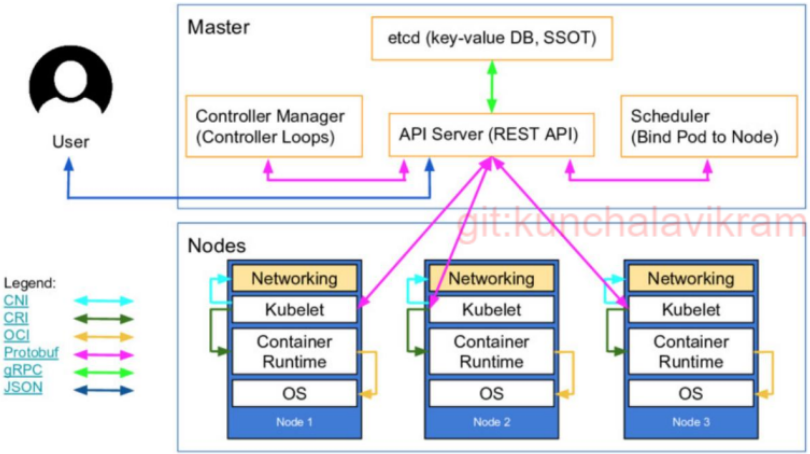
- Kube proxy

É um proxy de rede que executa em cada nó do cluster, implementando parte do conceito de **Kubernetes Service**

É responsável por garantir o tráfego que é roteado para os serviços internos e externos, conforme requeridos pelas regras e políticas de rede



K8s Arquitetura



Section 5

Mão na massa!!!

Exemplo com K8s

- Nesse exemplo, teremos:

Aplicação em python com Flask

Criação de uma imagem com a aplicação

Executar aplicação no K8s

Verificar aplicação em execução