Docker e Kubernetes

# Microservices

## Artigos

* Microservices (Martin Fowler)

<https://martinfowler.com/articles/microservices.html>

## Vídeos

* GOTO 2014 - Microservices (Martin Fowler)

<https://www.youtube.com/watch?v=wgdBVIX9ifA>

# Gestão de Microservices

Principais formas de gestão de containers levando em conta a necessidade dos processos se comunicarem.

## Orchestration

O

## Choreography

C

# Container

* Sandbox de um processo

## Artigos

* Kubernetes e containers: entenda esses conceitos (Sonda)

<https://blog.sonda.com/kubernetes-e-containers/>

* What is a Container (Docker)

<https://www.docker.com/resources/what-container>

# Docker

* Images
* Docker Files
* Orquestração de containers nativo - Swarm

## Site Oficial

<https://www.docker.com/>

# Kubernetes

* Orquestração de containers
* Automatização
* Balanceamento de carga
* Pods

## Artigos

* KUBERNETES: AN OVERVIEW (Janakiram MSV)

<https://thenewstack.io/kubernetes-an-overview/>

* Kubernetes

<https://pt.slideshare.net/erialc_w/kubernetes-50626679>

# Kubernetes Tutorial

## Create a Kubernetes cluster

### Kubernet Custers

Kubernetes é capaz de coordenar um cluster de computadores conectados para trabalharem como se fossem um único computador.

As aplicações a serem implantadas devem estar desacopladas de hosts individuais. Elas devem estar dentro de um conteiner.

Kubernetes automatiza a distribuição e o agendamento das aplicações em conteiners de maneira eficiente.

Um cluster do kubernet possui dois tipos de recursos:

* Master – coordena o cluster
* Nodes – *workers* que executam as aplicações

O master coordena todas as atividades do cluster, como: agendamento das aplicações, manutenção do estado desejado das aplicações, escalonamento das aplicações.

Os nodes podem ser VMs ou computadores físicos. Ele servirá como um “worker” no cluster do kubernetes. Cada node possui um kubelet, que gerencia o node e faz a comunicação entre node e o master. O node deve possuir uma forma de manipular os conteiners, como o Docker.

Em tráfego de produção, um cluster do kubernetes deve possuir pelo menos 3 nodes.

Quando ocorre implantações de aplicações, o Kubernetes age da seguinte forma:

1. O master é informado que deve iniciar os application containers.
2. O master agenda os nodes que devem executar os containers.
3. Os nodes comunicam com o master por meio do Kubernetes API.

Kubernetes API é disponibilizada pelo master. O usuário final pode utilizar esta API para realizar a comunicação com o cluster.

O cluster do Kubernetes pode ser implantado em máquinas físicas ou virtuais.

É possível utilizar o Minikube para desenvolvimento com Kubernetes. Ele cria uma VM e implanta um cluster simplificado, contendo apenas um node.

## Deploy an App

Uma vez em possessão de um cluster do Kubernetes em execução é possível realizar a implantação de um conteiner com aplicação. Para que isto seja possível, deve ser criada uma configuração de implantação (Deployment). O Deployment instrui como o Kurbenetes deve criar e atualizar as instâncias da aplicação.

Após a criação do Deployment, o master distribui as instâncias da aplicação aos nodes. O Kubernetes Deployment Controller irá monitorar continuamente essas instâncias. E irá repor a instância da aplicação caso um node com uma das instâncias da aplicação, cair ou for deletado.

O controller é um mechanismo que permite recuperação automática na ocorrência de falha da máquina ou de manutenção.

É possível criar e gerir um Deployment por meio do Kubectl. O Kubectl utiliza o Kubernetes API para interagir com o cluster. Ao criar um Deployment, é necessário especificar o image container para a aplicação e o número de réplicas que devem ser executadas. Essas informações poderão ser alteradas depois, ao atualizar o Deployment.

## Explore your App

### Pods

Um pod no Kubernetes representa a abstração de um grupo com um ou mais aplicações containerizadas, e de recursos compartilhados entre esses containers. Os recursos são:

* Volumes: armazenamento compartilhado
* Conexão em rede: um IP único dentro do cluster
* Informações dos containers: como executar cada containers, como qual versão da image ou as portas que deve utilizar

# Extras

## Arquitetura Monolítica x Micro Serviços

A

## Container x VM

C