

Nome: Henrico Nardelli Bela

RM: 95985

Turma: 2TIAR

---

## Atividade 1 - Computação Neuromórfica - 2o semestre - 2023

### 1- No seu critério em que cenários a implementação de cluster de kubernetes melhora o desempenho da execução e deploy das aplicações?

Resposta: A implementação de Kubernetes melhora o desempenho da execução e deploy das aplicações em vários âmbitos, alguns deles mais importantes são:

- **Escalabilidade:** Pode escalar aplicativos horizontalmente, adicionando ou removendo nós conforme necessário, para atender a mudanças na demanda. Isso pode ajudar a garantir que seus aplicativos estejam sempre disponíveis e funcionando de forma eficiente.
- **Resiliência:** Lida com falhas de nós automaticamente, redistribuindo os contêineres para nós ativos. Isso pode ajudar a garantir que seus aplicativos permaneçam disponíveis mesmo em caso de falhas de hardware ou software.
- **Manutenção:** Capaz de automatizar muitas tarefas de gerenciamento de contêineres, como implantação, escala e atualização. Isso pode ajudar a reduzir o tempo e o esforço necessários para gerenciar seus aplicativos.
- **Visibilidade:** Apresenta uma interface de usuário e uma API que permitem monitorar e gerenciar seus aplicativos de forma centralizada. Isso pode ajudar a garantir que seus aplicativos estejam funcionando como esperado e que você possa identificar e resolver problemas rapidamente.

### 2- Quais são as diferenças entre docker-compose, docker e kubernetes? Liste exemplos.

Resposta: O Docker Compose, o Docker e o Kubernetes são todas ferramentas que podem ser usadas para gerenciar contêineres. Porém cada ferramenta tem seu propósito definido.

- **Docker Compose:** uma ferramenta de linha de comando que é usada para definir e gerenciar um conjunto de containers Docker. O Docker Compose é uma ferramenta útil para desenvolvedores que desejam criar e testar aplicativos que usam containers Docker.
- **Docker:** uma plataforma que permite criar, executar e gerenciar contêineres. O Docker é uma ferramenta poderosa que pode ser usada para criar aplicativos de alto desempenho e escaláveis.
- **Kubernetes:** uma plataforma de orquestração de contêineres que automatiza o gerenciamento de contêineres em um cluster de nós. O Kubernetes é uma

ferramenta poderosa que pode ser usada para gerenciar aplicativos de grande escala.

Exemplos de aplicações de docker, docker compose e kubernetes:

- O Docker Compose pode ser usado para criar um conjunto de contêineres que executam um servidor web, um banco de dados e um aplicativo.
- O Docker pode ser usado para criar um contêiner que executa um servidor web que pode lidar com um grande volume de tráfego.
- O Kubernetes pode ser usado para gerenciar um cluster de nós que executam contêineres que executam uma aplicação de grande escala.

### **3- O que são os PODs dentro do contexto de kubernetes ? Quem é o responsável pelo seu gerenciamento ? e como isso interfere na resiliência das aplicações?**

Resposta: Um POD é a unidade de menor nível de gerenciamento no Kubernetes. Um POD é um grupo de um ou mais contêineres que compartilham recursos comuns, como um IP, um nome de domínio e um espaço de nomes de rede. Os PODs são projetados para serem altamente disponíveis e resilientes a falhas.

O responsável pelo gerenciamento dos PODs é o Kubernetes Master. O Kubernetes Master é um conjunto de componentes que gerenciam o cluster Kubernetes. O Kubernetes Master é responsável por agendar os PODs nos nós, monitorar os PODs e reiniciar os PODs que falham.

Os PODs são uma ferramenta importante para melhorar a resiliência das aplicações, são altamente disponíveis porque podem ser executados em vários nós, também são resilientes a falhas porque podem ser reiniciados automaticamente se falharem.

4- A empresa COMNEURO , desenvolve soluções de IoT para os principais portos da Europa. A solução core consiste numa rede de sensores LIVOX que capturam dados (topológicos e de reflexividades) de resíduos marítimos. Os dados são capturados em tempo real e enviados para a nuvem pelos serviços de data stream da AWS Kinesis, ativando uma função AWS Lambda que executa um algoritmo de classificação (tendo como base uma regressão logística), categorizando os resíduos em orgânicos, plásticos e inorgânicos. Posteriormente utilizando os serviços de AWS IoT Core a categorização é enviada via protocolo mqtt para a database do cliente.

Considerando o case apresentado, quantos serviços de software consegue identificar? , seria uma boa prática a containerização de toda a solução num único container ou é melhor a separação em “n” serviços(containers)?, aplicaria uma infraestrutura de deploy com tecnologia de Kubernetes?

Resposta:

1. Foi possível identificar quatro serviços de software:

Sensores LIVOX, AWS Kinesis, AWS Lambda, AWS IoT Core

2. Seria uma má prática a containerização de toda a solução num único container. A melhor abordagem seria separar a solução em "n" serviços (containers). Isso tornaria a solução mais escalável, resiliente e fácil de gerenciar.
3. Eu aplicaria uma infraestrutura de deploy com tecnologia de Kubernetes. Pois é uma plataforma de orquestração de contêineres que permite gerenciar um cluster de nós que executam contêineres, além de ser uma ferramenta poderosa que pode ser usada para gerenciar aplicativos de grande escala.

Alguns motivos pelos quais eu recomendaria separar a solução em n containers:

- **Escalabilidade:** Será possível escalá-la horizontalmente adicionando ou removendo nós do cluster Kubernetes. Isso permitirá que a solução atenda a mudanças na demanda de tráfego sem precisar ser recriada.
- **Resiliência:** Será possível torná-la mais resiliente a falhas. Isso pode ser feito executando cada serviço em um contêiner separado e, em seguida, distribuindo os contêineres por vários nós. Se um nó falhar, os outros nós ainda estarão disponíveis para servir a demanda de tráfego.
- **Gerenciabilidade:** Será mais fácil de gerenciar. Isso pode ser feito usando ferramentas como o Kubernetes Dashboard, que permite visualizar o estado dos contêineres e fazer alterações nos contêineres.

No geral, eu acredito que separar a solução em "n" serviços (containers) e implantá-la usando a tecnologia Kubernetes seria a melhor abordagem para essa solução.