CENTRO UNIVERSITÁRIO UNI METROCAMP | WYDEN

T6
Eng. da Computação
Sistemas Distribuídos

Nome: Lucas Eduardo Vasconcelos Alves
RA: 1510019454

Turma: 10° semestre, noturno

Professor: Santiago Robles

Data: **06/11/2019**

Microservices

Microservices é uma das sensações do momento, muito se fala sobre este estilo arquitetural e para muitos é a solução de todos os problemas. Microservices é (mais) um novo termo que está sendo bastante utilizado como definição de arquitetura ideal para aplicações corporativas, o termo inclusive é mais novo do que a própria implementação de fato, sendo que o principal objetivo é desenvolver serviços distribuídos e independentes que compõem uma ou mais aplicações.

Microservices (ou arquitetura de microserviços) é um estilo arquitetural que propõe uma abordagem de desenvolver uma aplicação através da construção de pequenos serviços, cada um com sua própria responsabilidade (capacidade de negócios) e comunicando-se através de mecanismos “leves”. Geralmente assumem o formato de API’s conversando através de HTTP.

Por serem independentes e pequenos (micro) eles funcionam através de mecanismos de deploy independentes e totalmente automatizados onde há o mínimo de gerenciamento centralizado sobre como são escritos. Sendo assim podem ser escritos em diferentes linguagens e tecnologias.

Quando sua aplicação realmente faz sucesso, você, muitas vezes, precisa de mais desenvolvedores, tem mais clientes e requests para atender, precisa aumentar a disponibilidade, distribuir a aplicação globalmente com tempo de resposta recorde e baixa latência. Para resolver esses problemas de sucesso, a Netflix utiliza a AWS como plataforma de Cloud Computing. A empresa migrou de uma aplicação monolítica, que era um gigante projeto Java/Web que faziam deploy através de um único pacote WAR, para uma arquitetura com centenas de microservices.

Um dos principais pontos de sucesso que levou a Netflix migrar para microservices foi Decomposição em Microservices, ou seja, permitir que o acesso para usuário final esteja sempre disponível, de forma que se uma parte da aplicação falhar, o restante permaneça em funcionamento, não permitindo que a falha se espalhe no sistema como um todo.

Concluindo, uma arquitetura de microservices deve assumir que tudo que pode falhar cedo ou tarde vai falhar, por isso deve se prevenir para que a aplicação continue disponível mesmo em caso de falhas em alguns dos

componentes da arquitetura, por isso o estado da aplicação deve estar apenas na camada de persistência, seja isso um banco de dados ou cache store.