ARQUITETURA DE MICROSSERVIÇOS EM JAVA COM DOCKER E AWS VANTAGENS E DESAFIOS

Autor(res)

Nicolas Vogiantzis
Paulo Izidoro Da Silva Barros

Categoria do Trabalho

Trabalho Acadêmico

Instituição

CENTRO UNIVERSITÁRIO ANHANGUERA DE SÃO PAULO

Introdução

A arquitetura de microsserviços se apresenta como uma alternativa aos sistemas monolíticos convencionais. Com ela, é possível dividir um aplicativo em partes autônomas, cada qual executando uma tarefa bem definida. Essa divisão diminui as chances de paralisação, já que, em momentos de grande procura, apenas o componente sobrecarregado exige ampliação, sem que seja preciso expandir o aplicativo inteiro. Ao usar tecnologias como Java, Docker e AWS, é viável criar aplicativos descentralizados, ajustáveis e estáveis, cumprindo as metas de performance, adaptabilidade e manutenção constante do mercado de hoje. Contudo, além dos pontos positivos, usar microsserviços também traz problemas importantes, como uma operação complicada, coordenação de serviços e precisão de uma base estrutural sólida. Este estudo investiga os principais benefícios e dificuldades dessa tática, dando uma análise aprofundada para especialistas e estudiosos da área de tecnologia da informação.

Objetivo

Analisar as vantagens e os desafios da implementação de uma arquitetura de microsserviços utilizando Java, Docker e Aws, destacando os aspectos técnicos, operacionais e estratégicos.

Material e Métodos

Para a análise da proposta do artigo, foi necessário conduzir um levantamento de publicações e estudos científicos atuais sobre o modelo de microsserviços, priorizando aplicações que usam Java, Docker e AWS. Além disso, analisamos exemplos de uso reais e testes de performance que demonstram o uso dessas tecnologias em situações práticas. A metodologia adotada envolveu a comparação entre distintas estruturas, levando em conta pontos como performance, capacidade de expansão, segurança e dificuldade de operação.

Resultados e Discussão

Vantagens:
Escalabilidade de Desempenho
A utilização de microsserviços permite a escalabilidade independente de componentes, otimizando o uso de recursos e melhorando o desempenho geral da aplicação. Estudos demonstram que aplicações baseadas em

microsserviços podem lidar com um número significativamente maior de usuários simultâneos em comparação com sistemas monolíticos.

Flexibilidade Tecnológica

Cada microsserviço pode ser desenvolvido utilizando a tecnologia mais adequada para sua função específica, promovendo inovação e agilidade no desenvolvimento.

Resiliência

A falha de um micro serviço não compromete toda a aplicação, facilitando a manutenção e a evolução contínua do sistema.

Desafios:

Complexidade Operacional

A gestão de múltiplos microsserviços exige ferramentas avançadas de orquestração e monitoramento, como kubernetes e Aws CloudWatch, para garantir a operação eficiente do sistema.

Comunicação entre serviços

A comunicação entre microsserviços pode introduzir latência e complexidade, especialmente em ambientes distribuídos.

Segurança e Governança

Garantir a segurança e a conformidade de uma arquitetura distribuída requer políticas rigorosas de autenticação, autorização e criptografia.

Conclusão

A implementação de uma arquitetura de microsserviços utilizando Java, Docker e AWS traz vantagens notáveis em expansão, adaptabilidade e robustez. Porém, é crucial que as empresas estejam prontas para lidar com as dificuldades ligadas à gestão complexa, interação entre os serviços e proteção dos dados. Ao empregar métodos corretos e instrumentos específicos, consegue-se tirar o máximo proveito dessa arquitetura.

Referências

FOWLER, M. Microservices: A definition of this new architectural term. MartinFowler.com, 2014. Disponível em: https://martinfowler.com/articles/microservices.html. Acesso em: 23 set. 2025.

KOTENKO, I. et al. Navigating the challenges and best practices in securing microservices systems. In: CEUR Workshop Proceedings, 2024. Disponível em: https://ceur-ws.org/Vol-3826/paper1.pdf. Acesso em: 19 set. 2025.

VILLAMIZAR, M. et al. Designing Scalable Java Architectures on AWS: Strategies and Best Practices. International Journal for Research in Applied Science and Engineering Technology, v. 12, n. 4, p. 1–10, 2024. Disponível em: https://www.ijraset.com/research-paper/designing-scalable-java-architectures-on-aws. Acesso em: 20 set. 2025.

Documentos Eletrônicos (Whitepapers e Blogs)

AWS. Let's Architect! Projetando arquiteturas de microsserviços. AWS Blog, [s.d.]. Disponível em: https://aws.amazon.com/pt/blogs/aws-brasil/lets-architect-projetando-arquiteturas-de-microsservicos/. Acesso em: 07 set. 2025.

AWS. Implementando microsserviços em AWS. AWS Whitepaper, [s.d.]. Disponível em: https://docs.aws.amazon.com/pt_br/whitepapers/latest/microservices-on-aws/microservices-on-aws.pdf. Acesso em: 14 set. 2025.

DOCKER. Docker Documentation. [s.d.]. Disponível em: https://docs.docker.com/. Acesso em: 26 set. 2025.

SCHABERT, P. Docker: Up & Running: Shipping Reliable Systems in Containers. O'Reilly Media, 2015.

SPRING. Spring Documentation. [s.d.]. Disponível em: https://docs.spring.io/spring-

boot/docs/current/reference/htmlsingle/. Acesso em: 26 set. 2025.