

EXERCÍCIOS DO MÓDULO 07: Contêineres, *builds* e *deployments*

Samuel Ferreira Machado¹

EXERCÍCIO 1 – CONTÊINERES E MICROSERVIÇOS

Explique, com suas palavras, a diferença entre contêineres e microserviços.

Extra: é possível subir microserviços em uma máquina virtual? Justifique sua resposta.

Microserviços são uma abordagem arquitetural para o desenvolvimento de software em que uma aplicação é dividida em serviços independentes, pequenos e autônomos que se comunicam entre si por meio de APIs (Interfaces de Programação de Aplicativos). Cada microserviço é responsável por uma única funcionalidade específica do sistema e pode ser desenvolvido, implantado e dimensionado de forma independente dos outros serviços. Quanto aos contêineres, são um meio para implementar os microserviços. Os contêineres possibilita o empacotamento de uma aplicação com todas suas bibliotecas, códigos, configurações, dependências e tudo o mais que é necessário para rodar em qualquer máquina.

Sobre subir microserviços em uma máquina virtual, é possível. Mas, não vejo como uma boa opção muito por conta do desempenho, escalabilidade e gerenciamento. Os microserviços em contêineres é sem dúvida a melhor alternativa pelo ganho nesses 3 fatores citados.

EXERCÍCIO 2 – INTEGRAÇÃO CONTÍNUA

Explique, com suas palavras, por que os desenvolvedores precisam fazer merge frequente de suas mudanças de código para que a integração contínua funcione adequadamente.

A prática de fazer merges frequentes de mudanças de código é fundamental para garantir a integração contínua bem-sucedida, facilitar a colaboração, reduzir riscos e fornecer *feedback* rápido e contínuo aos desenvolvedores. Isso ajuda a manter a estabilidade e a qualidade do código, além de acelerar o processo de desenvolvimento de software.

¹ Aluno do Curso Especialista Back-End Java da EBAC.

EXERCÍCIO 3 – ENTREGA CONTÍNUA

Explique, com suas palavras, por que é necessário ter uma cobertura abrangente de testes para que a prática de entrega contínua funcione.

Uma cobertura abrangente de testes é fundamental para a prática de entrega contínua, pois garante a detecção precoce de problemas, mantém a confiança no código, facilita a detecção de dependências e impactos e permite a automação dos testes. Esses benefícios são essenciais para a entrega contínua bem-sucedida, pois garantem a qualidade, a estabilidade e a confiabilidade do software que está sendo entregue em um ritmo rápido e contínuo.

EXERCÍCIO 4 – IMPLANTAÇÃO CONTÍNUA

Explique, com suas palavras, as diferenças entre entrega contínua e implantação contínua.

Entrega Contínua (*Continuous Delivery - CD*): é uma abordagem de desenvolvimento de software em que as alterações de código são desenvolvidas, testadas e preparadas para implantação de forma incremental e frequente. No contexto da entrega contínua, o *software* é construído, testado e empacotado automaticamente em um estado pronto para ser implantado em produção. No entanto, a decisão final de implantar a versão em produção é tomada manualmente. A entrega contínua permite que as equipes de desenvolvimento e operações trabalhem em conjunto de forma colaborativa e ágil, garantindo que o software esteja sempre em um estado pronto para ser entregue. Esse processo inclui a execução de testes automatizados e verificações de qualidade antes da implantação.

Implantação Contínua (*Continuous Deployment - CI*): por sua vez, é uma extensão da entrega contínua. Nesse caso, todas as mudanças de código que passaram pelos testes e verificações automatizados são implantadas automaticamente no ambiente de produção, sem intervenção manual adicional. A implantação contínua automatiza o processo de colocar as alterações em produção de forma contínua e incremental. De tal modo, assim que uma alteração é considerada pronta para ser entregue e aprovada nos testes automatizados, ela é implantada automaticamente no ambiente de produção, reduzindo o tempo entre o desenvolvimento e a disponibilidade para os usuários finais. Essa abordagem oferece

maior agilidade e velocidade na entrega de *software*, minimizando a intervenção manual.