

UniFECAF

Análise e Desenvolvimento de Sistemas

Projeto Cloud DevOps

Lucas Ribeiro Soares

RA: 105437

Arquitetura de Microsserviços em Ambiente Cloud-Native

1. Arquitetura de Microserviços e o papel do DevOps

A arquitetura de microserviços consiste na divisão de uma aplicação em serviços pequenos, independentes e especializados, que se comunicam entre si por meio de APIs. Diferente de sistemas monolíticos, onde toda a aplicação é construída em um único bloco, os microserviços permitem maior flexibilidade, escalabilidade e facilidade de manutenção.

Em ambientes modernos, essa arquitetura é amplamente utilizada em sistemas cloud-native, pois facilita o deploy contínuo, a escalabilidade sob demanda e a resiliência. Cada serviço pode ser desenvolvido, testado e implantado de forma independente.

O DevOps surge como uma cultura e conjunto de práticas que integram desenvolvimento (Dev) e operações (Ops). Seu objetivo é acelerar a entrega de software com qualidade, automação e confiabilidade. Em ambientes de microserviços, o DevOps é essencial para automatizar build, testes, deploy e monitoramento, garantindo ciclos de entrega contínua.

2. Containerização: Docker e Kubernetes

A containerização é a técnica de empacotar uma aplicação juntamente com todas as suas dependências em um container, garantindo que ela rode da mesma forma em qualquer ambiente. O Docker é a ferramenta mais popular para criação e execução de containers. Ele permite construir imagens versionadas, leves e reproduzíveis.

Já o Kubernetes é uma plataforma de orquestração de containers. Enquanto o Docker executa containers individualmente, o Kubernetes gerencia múltiplos containers em ambientes distribuídos, oferecendo recursos como balanceamento de carga, autoescala, reinício automático e atualizações sem downtime.

3. Orquestração de Contêineres

A orquestração de contêineres consiste na gestão automatizada do ciclo de vida dos serviços em execução. No projeto desenvolvido, foram utilizados Deployments e Services do Kubernetes, permitindo a criação de réplicas dos serviços e sua exposição interna e externa. Além disso, foi aplicada a estratégia de Rolling Update, que garante atualizações contínuas sem indisponibilidade do sistema.

4. CI/CD em ambientes distribuídos

CI/CD (Continuous Integration / Continuous Delivery) é um conjunto de práticas que automatizam integração, testes e entrega de software.

CI (Integração Contínua): cada alteração no código é automaticamente testada e validada.

CD (Entrega Contínua): após aprovação, a aplicação pode ser automaticamente publicada ou implantada.

Em arquiteturas distribuídas, pipelines CI/CD garantem consistência, reduzem erros humanos e aceleram entregas. Ferramentas como GitHub Actions permitem implementar pipelines automatizados diretamente no repositório.

5. Observabilidade: métricas, logs e traces

CI/CD (Continuous Integration / Continuous Delivery) é um conjunto de práticas que automatizam integração, testes e entrega de software.

CI (Integração Contínua): cada alteração no código é automaticamente testada e validada.

CD (Entrega Contínua): após aprovação, a aplicação pode ser automaticamente publicada ou implantada.

Em arquiteturas distribuídas, pipelines CI/CD garantem consistência, reduzem erros humanos e aceleram entregas. Ferramentas como GitHub Actions permitem implementar pipelines automatizados diretamente no repositório.

6. Justificativa das decisões arquiteturais

Observabilidade é a capacidade de entender o estado interno de um sistema a partir de seus dados externos. Em ambientes distribuídos, ela é essencial.

Métricas: dados numéricos sobre consumo de CPU, memória e latência.

Logs: registros de eventos das aplicações.

Traces: rastreamento de requisições entre serviços.

Ferramentas como Prometheus (métricas), Loki (logs) e Jaeger (traces) são amplamente usadas em ambientes Kubernetes.

Conclusão

A combinação de microserviços, containers, orquestração Kubernetes e práticas DevOps fornece uma base sólida para aplicações escaláveis, resilientes e de rápida entrega. O projeto demonstra, de forma prática, como esses conceitos são aplicados em um ambiente real.