

Documento de Arquitetura do Sistema iFood

1. Introdução

Este documento descreve a arquitetura do sistema da iFood, destacando sua infraestrutura, componentes principais, integração de APIs, segurança, escalabilidade e estratégias de observabilidade. O objetivo é fornecer uma visão abrangente das práticas tecnológicas que sustentam a plataforma iFood, que oferece serviços de entrega de alimentos.

2. Visão Geral da Arquitetura

2.1 Transição para Microserviços e Arquitetura Orientada a Eventos

- **Arquitetura Microservices:** A iFood migrou de uma arquitetura monolítica para uma arquitetura de microserviços orientada a eventos para melhorar a resiliência, escalabilidade e desempenho【8†source】【10†source】.
- **EventBridge da AWS:** Utilizado como o principal barramento de eventos, o Amazon EventBridge permite comunicação escalável e de alta disponibilidade entre microserviços【8†source】.

2.2 Componentes Principais

- **Gestão de Dados Mestre:** Um dos microserviços responsáveis por gerenciar os dados principais da plataforma【8†source】.
- **Provisão e Requisição:** Microserviços dedicados a gerenciar solicitações de serviços e fornecimento de dados【8†source】.
- **Backend for Frontend (BFF):** Padrão arquitetural usado para integrar sistemas legados, convertendo solicitações síncronas em assíncronas【8†source】.

3. Infraestrutura Tecnológica

3.1 Serviços de Nuvem

- **Plataforma de Nuvem:** A iFood utiliza Amazon Web Services (AWS) para hospedar seus serviços, garantindo alta disponibilidade e escalabilidade【8†source】.
- **Orquestração de Contêineres:** Kubernetes é utilizado para gerenciar contêineres Docker, facilitando a escalabilidade horizontal e a resiliência【8†source】.

3.2 Armazenamento de Dados

- **Data Lake:** Implementado com Databricks, permite a análise integrada e o armazenamento de grandes volumes de dados【10†source】.
- **Bancos de Dados Descentralizados:** Cada microserviço possui seu próprio banco de dados, prevenindo gargalos e permitindo escalabilidade independente【8†source】.

4. Integração de APIs

- **Gateway de API:** AWS API Gateway gerencia e monitora o tráfego das APIs, fornecendo autenticação e segurança【8†source】.
- **APIs de Pagamento e Geolocalização:** Integração com APIs de pagamento (PayPal, Stripe) e geolocalização (Google Maps) para suporte a transações e rastreamento【8†source】【9†source】.

5. Segurança

- **Autenticação e Autorização:** Implementação de OAuth 2.0 e autenticação multifatorial (MFA) para proteger o acesso【8†source】.
- **Criptografia:** Dados são criptografados em trânsito e em repouso utilizando protocolos seguros【9†source】.

6. Escalabilidade e Performance

- **Arquitetura de Microserviços:** Permite que cada componente seja escalado de acordo com suas necessidades de carga

de trabalho【8†source】.

- **Comunicação Assíncrona:** Reduz o acoplamento dinâmico e aumenta a resiliência do sistema【8†source】.

7. Observabilidade e Monitoramento

- **Centralização de Logs:** Utilização de ferramentas como ELK Stack para centralizar e analisar logs em tempo real【8†source】.
- **Monitoramento de Performance:** Ferramentas como Prometheus e Grafana são usadas para monitorar métricas de sistema【8†source】.

8. Conclusão

A arquitetura do sistema da iFood é projetada para ser escalável, resiliente e segura, suportando a rápida expansão e inovação contínua da plataforma. A adoção de microserviços e uma abordagem orientada a eventos melhoraram significativamente a eficiência operacional e a experiência do usuário【10†source】.

Este documento fornece uma visão abrangente dos componentes arquitetônicos do iFood e pode servir como referência para futuras melhorias e expansões do sistema. Se você precisar de mais detalhes ou especificações, sinta-se à vontade para perguntar!