

Consumer em golang

Motivação do project:

Atualmente com o alto fluxo de dados circulando em cadastros, inscrições e compras online, os dados dos clientes estão cada vez mais expostos à fraude. Pensando nisso, surgiram leis como a LGPD, que vem com o intuito de proteger esses dados, porém, sempre podemos pensar em novas soluções.

Uma empresa de venda de produtos variados, que capta informações dos clientes que efetuam compras, por campanhas de descontos na internet e por cadastros de contas no site de vendas, e os armazena em um banco de dados SQL (Banco 1). Sabendo da sensibilidade dos dados, a empresa precisa de uma solução para o time de televendas ter acesso somente aos dados necessários e em tempo real, de maneira que tenha acesso apenas ao nome, telefone e região do cliente, para que possa ligar e trabalhar a venda.

Dado o cenário, o time de TI montou uma estrutura de API's (producer e consumer) que garante que os dados sejam salvos em um outro banco SQL (Banco 2), em tempo real, e de maneira que apenas os dados necessários sejam gravados. O projeto consiste em uma estrutura (figura 1) composta por 2 bancos de dados SQL, 1 producer e um consumer.

Nessa etapa, será construído apenas o consumer do projeto.

Arquitetura:

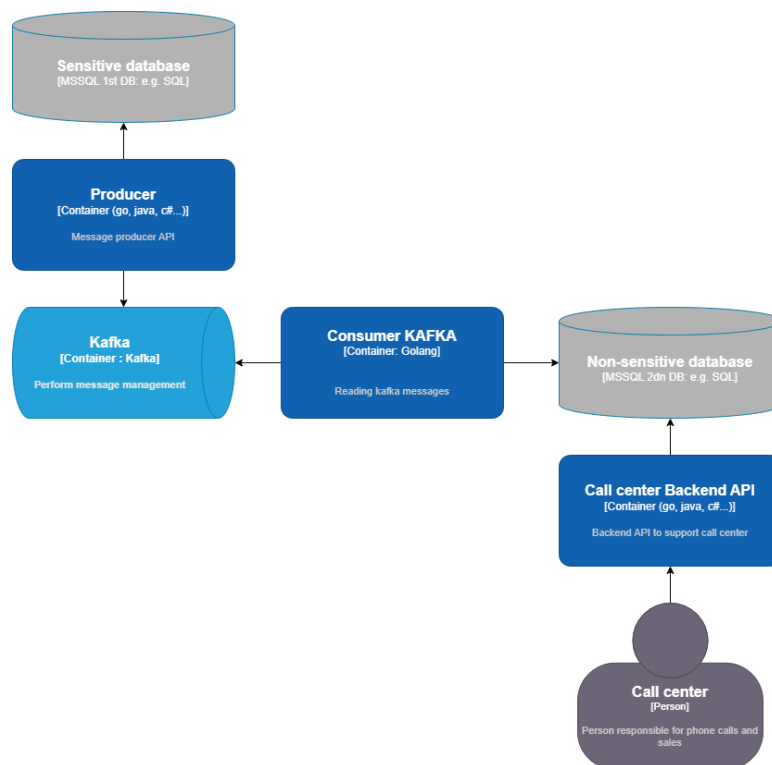


Figura 1: Diagrama geral

Dados do projeto:

O apache kafka é uma plataforma de streaming baseada em sistema distribuído de publish/subscribe, onde há um processo chamado producer, que envia mensagens para um tópico, que é gerenciado e armazenado por um cluster de brokers, e um processo chamado consumer que subscreve ao tópico para realizar a leitura e processamento das mensagens.

A linguagem escolhida foi Golang devido ao baixo consumo de memória nas máquinas de cloud (pensando em azure), que resulta em economia e facilidade de escalar novas instâncias da API em caso de picos de consumo (black friday ou outras campanhas de vendas).

A construção será utilizando a arquitetura de Domain Driven Design (DDD), visando deixar o código o mais genérico possível, para em caso de haver a necessidade de construir novas API's de consumer, seja necessário modificar o código o mínimo possível.