

Solução

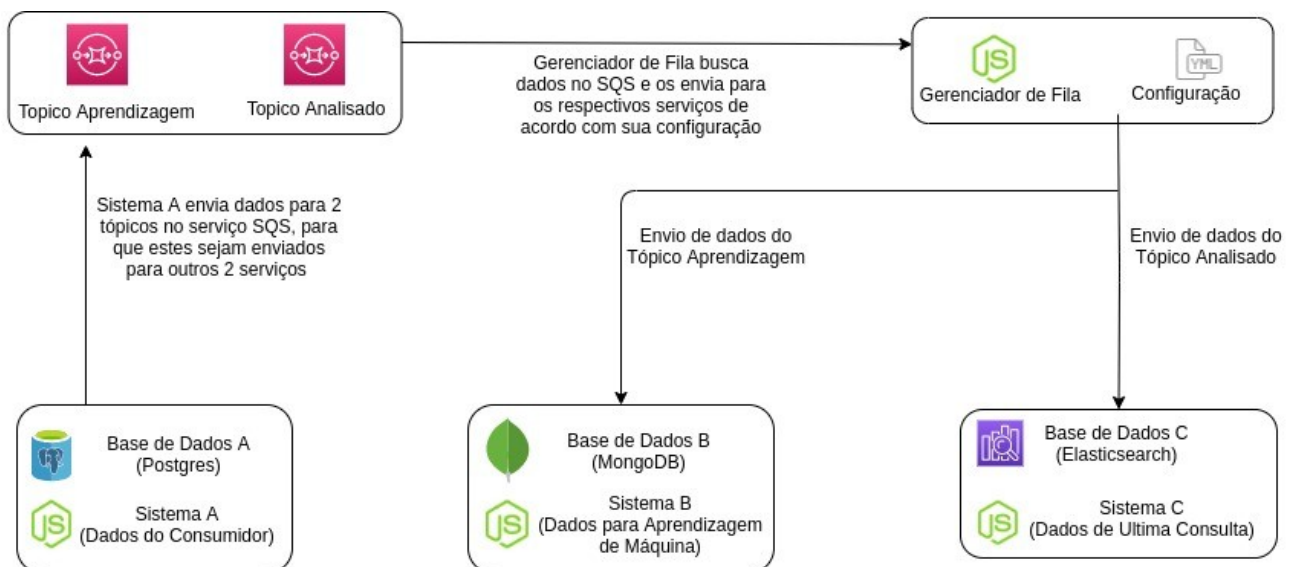
A solução para o problema exposto consiste em criar três bancos de dados em diferentes tecnologias com sistemas disponibilizados de forma que estes sejam independentes, ou seja, possam ser utilizados de forma independente.

Cada sistema terá a sua responsabilidade, o primeiro sistema irá registrar todos os dados do consumidor e enviar para o serviço SQS dados referentes aos demais serviços (aprendizagem de máquina e dados já analisados para rápidas consultas).

O segundo serviço será responsável por disponibilizar dados para fluxos de aprendizagem de máquina.

O terceiro serviço terá como responsabilidade disponibilizar dados referentes à última consulta e o quarto serviço realizará a conexão com o serviço de SQS para consumir uma fila e redirecionar os dados para seus respectivos serviços.

A imagem abaixo representa o fluxo de dados entre os serviços acima mencionados.



Base de dados

Base de dados A

O primeiro banco de dados, seria estruturado em Postgres, devido a possibilidade de criar constraints em campos, relacionamentos em nível de Primary key/Foreign key, assim, a confiabilidade dos dados seria maior que em bancos de dados não relacionais.

Para agilizar consultar e evitar exclusão de dados, como por exemplo, executar uma query sem a condição 'where', podem ser criadas views com as devidas permissões.

O sistema responsável por inserir e buscar dados deste banco, não utilizaria o 'owner' do banco, evitando problemas relacionados a permissão de acesso aos dados.

Caso existam dados muito sensíveis que não possam ser visualizados, pode ser configurado o pgcrypto para evitar o conhecimento do conteúdo registrado no banco.

Este banco de dados deve ser disponibilizado no serviço RDS da Amazon devido à mecanismos de redundância de dados que podem ser utilizados para se precaver de possíveis problemas que podem ocorrer, evitando assim a perda dos dados.

Base de dados B

O segundo banco de dados, seria estruturado em MongoDB, devido a necessidade de se obter um acesso mais rápido às informações.

Como parte das informações seriam trazidas da base de dados A, já teríamos uma base confiável e assim apenas os dados necessários para aprendizagem de máquina e cálculo de raiting seriam enviados.

Base de dados C

O terceiro banco de dados, seria estruturado em Elasticsearch, devido a necessidade de se obter um acesso extremamente rápido às informações.

Neste banco de dados seriam registradas as movimentações financeiras, data da ultima compra com cartão de crédito e demais eventos.

Sistemas

Sistema de acesso a base de dados A

O sistema responsável pelo acesso à base de dados A, seria um micro-serviço desenvolvido em Node.js e sua conexão com o banco de dados através da biblioteca sequelize.

Neste sistema seriam disponibilizadas rotas para inserir e buscar as informações no banco de dados e para enviar informações para uma fila que seria consumida por outros dois sistemas.

Seu acesso deve ser realizado através de autenticação via token jwt e todas requisições realizadas para este serviço devem gerar log para que sejam analisados caso seja necessário.

Sempre que novos dados são enviados para este serviço, outros dados baseados neste, são enviados para uma fila para que sejam consumidos por outros dois serviços.

Sistema de acesso a base de dados B

Este sistema seria o responsável por inserir as informações na base de dados B, calcular o raiting e extrair os dados para aprendizagem de máquina.

Sistema desenvolvido em Node.js utilizando a biblioteca Mongoose para conexão ao banco de dados.

Neste sistema, para agilizar o processamento dos dados disponibilizados na fila, poderia ser utilizada a biblioteca 'cluster', permitindo assim processamentos de dados paralelos de acordo com a capacidade do processadores da maquina onde esta sendo executado.

Sistema de acesso a base de dados C

Sistema desenvolvido em Node.js na arquitetura Rest, responsável por inserir e buscar informações na base de dados C.

Sistema Gerenciador de Fila

Este sistema seria responsável por consumir a fila e enviar os dados para os sistema da base de dados B e para o sistema da base de dados C.

Como este sistema tem como função apenas o envio de dados contidos na fila SQS, não será necessário a comunicação com um banco de dados.

O arquivo de configuração existente neste sistema no formato yaml, será o responsável por relacionar para qual api os dados deverão ser enviados.

Serviço de fila SQS

Serviço de fila disponibilizado pela Amazon que permite que dados sejam expostos para consumo por outras api's, de forma escalável.

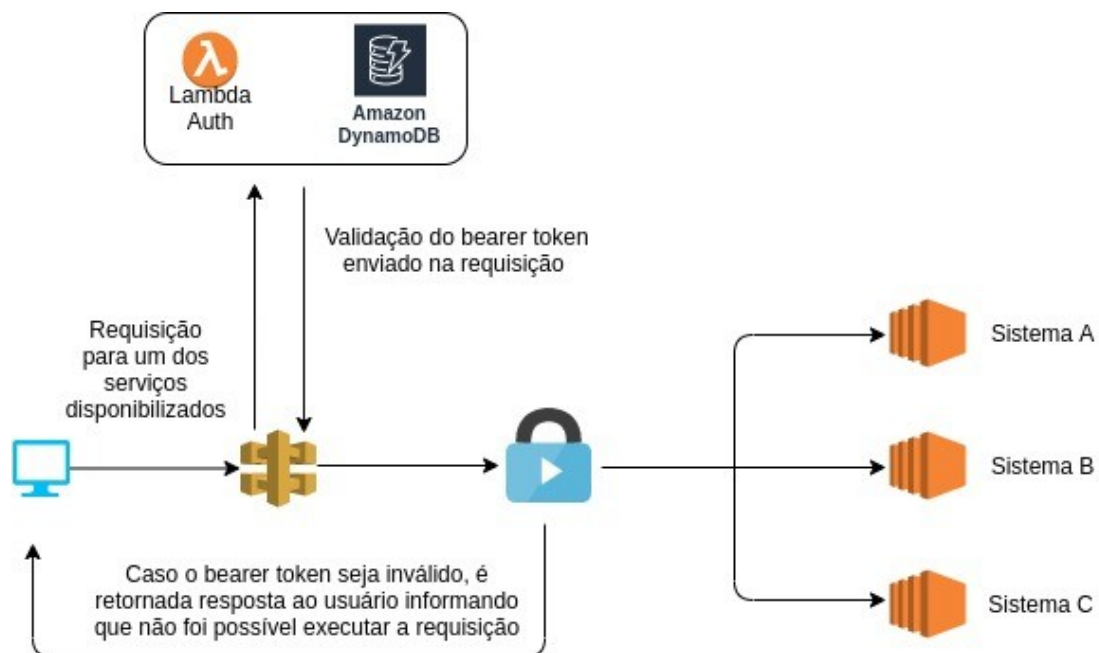
Neste serviço seriam criados dois tópicos, um para o sistema de acesso a base de dados B e outro para consumo do sistema de dados da base C.

O único sistema responsável por consumir os dados contidos no SQS é o Gerenciador de Fila.

Acesso externo aos serviços

Os sistemas desenvolvidos poderão ser consumidos apenas com a utilização de token do tipo jwt. Ao realizar a chamada a um dos serviços disponibilizados, o token será validado através de um serviço lambda.

Após a validação do token, o serviço poderá ser consumido.



Deploy dos sistemas

Todos os sistemas devem ser desenvolvidos para serem utilizados via docker-compose. Os serviços da Amazon utilizados para deploy seriam: code commit, ecr, ecs, codebuild e codepipeline.

Projeto docker

Este projeto deve ser desenvolvido para que seja utilizado no momento do build do sistema. O aws codebuild utiliza este projeto para enviar a