Arquitetura Serverless para o Chatbot "Dra Jô"

Equipe:

Fernando Henrique Rodrigues da Silva - **RM559660**
Gabriela da Cunha Rocha - **RM561041**
Gustavo Segantini Rossignolli - **RM560111**
Vitor Lopes Romão - **RM559858**

Introdução

Este documento detalha a arquitetura serverless do chatbot "Dra Jô", desenvolvido para atender clientes Solubio em diversas áreas, incluindo serviços financeiros, eficiência agronômica e suporte comercial. A arquitetura utiliza serviços da AWS, garantindo escalabilidade, alta disponibilidade e baixo custo operacional.

Visão Geral da Arquitetura

A arquitetura é composta por diversos serviços AWS integrados para proporcionar uma experiência fluida e eficiente aos usuários que interagem com o chatbot via WhatsApp. Os principais componentes incluem:

- **Interface de Usuário**: WhatsApp
- **Gerenciamento de DNS e Roteamento**: AWS Route 53
- **Segurança de Rede, WAF**: AWS Firewall
- **Ponto de Entrada**: Amazon API Gateway
- **Processamento**: AWS Lambda
- **Fila de Mensagens**: Amazon SQS
- **Chatbot**: AWS Lex
- **Banco de Dados**: Amazon DynamoDB
- **Indexação e Consulta de Dados**: AWS Athena e Amazon S3
- **Monitoramento e Custos**: Serviços de monitoramento integrados com Lambda e SQS

Descrição dos Componentes

Integração com WhatsApp

A integração com o WhatsApp é realizada através de uma API que permite o envio e recebimento de mensagens. Essa interface é a principal forma de interação dos usuários com o chatbot "Dra Jô".

AWS Route 53

O AWS Route 53 gerencia o DNS e roteamento, direcionando o tráfego de usuários para o Amazon API Gateway. Ele garante que as solicitações sejam encaminhadas corretamente, mantendo a disponibilidade do serviço.

AWS Firewall

O AWS Firewall fornece uma camada adicional de segurança, protegendo a arquitetura contra ameaças e acessos não autorizados.

Amazon API Gateway

O Amazon API Gateway atua como ponto de entrada para todas as solicitações provenientes do WhatsApp. Ele gerencia as APIs RESTful que interagem com os serviços backend, garantindo escalabilidade e gerenciamento eficiente de chamadas.

AWS Lambda

As funções Lambda são utilizadas para processar as mensagens recebidas, executar lógica de negócio e interagir com outros serviços, como enfileirar mensagens no Amazon SQS e consultar dados no DynamoDB.

Amazon SQS

O Amazon SQS é utilizado para enfileirar mensagens, garantindo que todas as solicitações sejam processadas de forma assíncrona e escalável, sem perda de dados.

AWS Lex

O AWS Lex é o serviço de chatbot que compreende a linguagem natural dos usuários. Ele é responsável por interpretar as mensagens e gerar respostas adequadas.

Amazon DynamoDB

O DynamoDB armazena informações de sessão, preferências dos usuários e outros dados necessários para personalizar as interações. A interação entre o AWS Lex e o DynamoDB é detalhada a seguir.

Detalhamento da Interação entre AWS Lex e DynamoDB

- **Armazenamento de Contexto**: O Lex utiliza o DynamoDB para armazenar o contexto das conversas, permitindo que o chatbot mantenha uma conversa coerente e contextualizada.
- **Consulta de Dados**: Durante a interação, o Lex pode consultar o DynamoDB para obter informações específicas do usuário, como histórico de compras, dados financeiros ou etapas do cultivo agrícola.
- **Atualização de Dados**: O chatbot pode atualizar informações no DynamoDB, como registrar solicitações de suporte ou agendar visitas de especialistas.

AWS Athena e Amazon S3

O AWS Athena é utilizado para consultar e indexar dados armazenados em buckets S3. Isso permite que o chatbot acesse rapidamente documentos e materiais de apoio para fornecer aos usuários.

Monitoramento de Performance e Custos

A arquitetura inclui componentes de monitoramento que utilizam Lambda e SQS para coletar métricas de performance e custos, garantindo a otimização contínua dos recursos.

Funcionalidades do Chatbot "Dra Jô"

Atendimento ao Cliente em Serviços Financeiros

- **Consulta e Geração de Boletos**: Permite que os clientes consultem faturas pendentes e gerem novos boletos para pagamento.
- **Faturamento e Notas Fiscais Eletrônicas**: Fornece acesso a informações de faturamento e emissão de NF-e.
- **Renegociações**: Auxilia os clientes na renegociação de dívidas e condições de pagamento.

Suporte em Eficiência Agronômica

- **Feedback sobre Tratamentos**: Oferece recomendações sobre tratamentos agrícolas com base nas fases da cultura.
- **Indicação de Procedimentos**: Sugere procedimentos agrícolas adequados, ajudando os clientes a otimizar suas operações.

Envolvimento de Especialistas em Engenharia

O chatbot identifica quando é necessário envolver um especialista humano, encaminhando a conversa ou agendando uma consulta com um engenheiro.

Suporte Comercial

- **Materiais de Apoio e Portfólios de Produtos**: Fornece acesso a catálogos, fichas técnicas e materiais promocionais.
- **Abertura de Novas Negociações**: Facilita o contato com o departamento comercial para iniciar novas negociações.

Fluxo de Dados na Arquitetura

1. **Usuário envia mensagem via WhatsApp**.
2. **A mensagem é roteada pelo AWS Route 53 e passa pelo AWS Firewall**.
3. **O Amazon API Gateway recebe a solicitação e invoca uma função Lambda**.
4. **A função Lambda processa a mensagem e a enfileira no Amazon SQS**.
5. **Outra função Lambda consome a mensagem do SQS e interage com o AWS Lex**.
6. **O AWS Lex interpreta a mensagem e, se necessário, consulta o DynamoDB ou o AWS Athena**.
7. **O Lex gera uma resposta, que é enviada de volta ao usuário através do fluxo inverso**.

8. **Monitoramento contínuo ocorre em paralelo, coletando métricas de performance e custos**.

Integração com SAP e ERP do Cliente

Os dados utilizados pelo chatbot são fornecidos principalmente pelo SAP e outros sistemas ERP do cliente. A integração é feita através de APIs ou serviços de extração de dados, permitindo que o Lex acesse informações atualizadas para interagir com os usuários.

Estimativa de Custos de Infraestrutura para o Chatbot "Dra Jô"

Segue uma estimativa de custos de infraestrutura para a arquitetura serverless do chatbot "Dra Jô", considerando dois cenários de uso:

- **Cenário 1**: 1.000 atendimentos por mês
- **Cenário 2**: 100.000 atendimentos por mês

As estimativas levam em conta os principais serviços AWS utilizados na arquitetura e os custos associados a cada um, com base nos preços públicos da AWS até a data de conhecimento (outubro de 2023).

Premissas

Para calcular os custos, assumimos as seguintes premissas:

- **Interação Média**: Cada atendimento envolve uma média de **10 mensagens** (5 do usuário e 5 do chatbot).
- **Mensagens Totais por Mês**:
    - Cenário 1: 1.000 atendimentos x 10 mensagens = **10.000 mensagens/mês**
    - Cenário 2: 100.000 atendimentos x 10 mensagens = **1.000.000 mensagens/mês**
- **Serviços Considerados**: Incluímos os principais serviços AWS e os custos de integração com o WhatsApp Business API.

Estimativa de Custos por Serviço

1. Amazon API Gateway

Preço:

- $3,50 por milhão de chamadas (REST API)

Cálculo:

- **Cenário 1**:
    - Chamadas: 10.000
    - Custo: (10.000 / 1.000.000) x $3,50 = **$0,035**
- **Cenário 2**:
    - Chamadas: 1.000.000
    - Custo: (1.000.000 / 1.000.000) x $3,50 = **$3,50**

2. AWS Lambda

Preço:

- $0,20 por milhão de solicitações
- $0,0000166667 por GB-segundo de tempo de computação (128 MB de memória)

Cálculo:

- **Cenário 1**:
    - Solicitações: 10.000
    - Tempo total: 10.000 x 100ms x 128MB = 125 GB-segundos
    - Custo de solicitações: (10.000 / 1.000.000) x $0,20 = **$0,002**
    - Custo de computação: 125 x $0,0000166667 = **$0,00208**
    - **Total Lambda**: **$0,00408**
- **Cenário 2**:
    - Solicitações: 1.000.000
    - Tempo total: 1.000.000 x 100ms x 128MB = 12.500 GB-segundos
    - Custo de solicitações: (1.000.000 / 1.000.000) x $0,20 = **$0,20**
    - Custo de computação: 12.500 x $0,0000166667 = **$0,20833**
    - **Total Lambda**: **$0,40833**

3. Amazon SQS

Preço:

- $0,40 por milhão de solicitações

Cálculo:

- **Cenário 1**:
    - Solicitações: 20.000 (envio e recebimento)
    - Custo: (20.000 / 1.000.000) x $0,40 = **$0,008**
- **Cenário 2**:
    - Solicitações: 2.000.000
    - Custo: (2.000.000 / 1.000.000) x $0,40 = **$0,80**

4. AWS Lex

Preço:

- $0,004 por solicitação de texto

Cálculo:

- **Cenário 1**:
  - Solicitações: 10.000
  - Custo: 10.000 x $0,004 = **$40,00**
- **Cenário 2**:
  - Solicitações: 1.000.000
  - Custo: 1.000.000 x $0,004 = **$4.000,00**

5. Amazon DynamoDB

Preço:

- $1,25 por milhão de unidades de gravação
- $0,25 por milhão de unidades de leitura
- Armazenamento: $0,25 por GB/mês (negligenciável para pequenos volumes)

Cálculo:

- **Cenário 1**:
  - Gravações: 10.000
  - Leituras: 10.000
  - Custo de gravação: (10.000 / 1.000.000) x $1,25 = **$0,0125**
  - Custo de leitura: (10.000 / 1.000.000) x $0,25 = **$0,0025**
  - **Total DynamoDB**: **$0,015**
- **Cenário 2**:
  - Gravações: 1.000.000
  - Leituras: 1.000.000
  - Custo de gravação: (1.000.000 / 1.000.000) x $1,25 = **$1,25**
  - Custo de leitura: (1.000.000 / 1.000.000) x $0,25 = **$0,25**
  - **Total DynamoDB**: **$1,50**

6. AWS Athena

Preço:

- $5,00 por TB de dados escaneados

Cálculo:

- **Cenário 1**:
  - Dados escaneados: 100 GB (0,1 TB)
  - Custo: 0,1 x $5,00 = **$0,50**
- **Cenário 2**:
  - Dados escaneados: 10 TB

    - Custo: 10 x $5,00 = **$50,00**

7. Amazon S3

Preço:

- Armazenamento: $0,023 por GB/mês (até 50 TB)

Cálculo:

- **Armazenamento Estimado**: 10 GB
- Custo: 10 x $0,023 = **$0,23** (mesmo para ambos os cenários)

8. AWS Route 53

Preço:

- $0,50 por zona hospedada/mês

Cálculo:

- **Total Route 53**: **$0,50** (mesmo para ambos os cenários)

9. AWS Firewall (AWS WAF)

Preço:

- $5,00 por ACL web/mês
- $1,00 por regra/mês (assumindo 5 regras)
- $0,60 por milhão de solicitações

Cálculo:

- **Cenário 1**:
    - Solicitações: 10.000
    - Custo de solicitações: (10.000 / 1.000.000) x $0,60 = **$0,006**
    - **Total WAF**: $5,00 (ACL) + $5,00 (regras) + $0,006 = **$10,006**
- **Cenário 2**:
    - Solicitações: 1.000.000
    - Custo de solicitações: (1.000.000 / 1.000.000) x $0,60 = **$0,60**
    - **Total WAF**: $5,00 + $5,00 + $0,60 = **$10,60**

10. Monitoramento (AWS CloudWatch)

Preço:

- $0,30 por métrica personalizada/mês
- $0,01 por 1.000 pontos de dados

Cálculo:

- **Métricas**: 10
- **Cenário 1**:
    - Pontos de dados: 10 x 10.000 = 100.000
    - Custo de métricas: 10 x $0,30 = **$3,00**
    - Custo de pontos de dados: (100.000 / 1.000) x $0,01 = **$1,00**
    - **Total CloudWatch**: **$4,00**
- **Cenário 2**:
    - Pontos de dados: 10 x 1.000.000 = 10.000.000
    - Custo de métricas: 10 x $0,30 = **$3,00**
    - Custo de pontos de dados: (10.000.000 / 1.000) x $0,01 = **$100,00**
    - **Total CloudWatch**: **$103,00**

11. Integração com WhatsApp Business API

O custo da integração com o WhatsApp varia de acordo com o provedor (por exemplo, Twilio, MessageBird) e a política de preços do WhatsApp, que é baseada em conversas iniciadas pelo usuário ou pela empresa.

Estimativa:

- **Custo Médio por Conversa**: $0,05 (valor aproximado)
- **Cenário 1**:
    - Conversas: 1.000
    - Custo: 1.000 x $0,05 = **$50,00**
- **Cenário 2**:
    - Conversas: 100.000
    - Custo: 100.000 x $0,05 = **$5.000,00**

Nota: Recomenda-se verificar o provedor escolhido para obter os preços exatos, pois podem haver taxas adicionais.

Resumo dos Custos

Cenário 1: 1.000 Atendimentos por Mês

Serviço	Custo (USD)
Amazon API Gateway	$0,035
AWS Lambda	$0,00408
Amazon SQS	$0,008
AWS Lex	$40,00
Amazon DynamoDB	$0,015
AWS Athena	$0,50
Amazon S3	$0,23
AWS Route 53	$0,50
AWS WAF	$10,006
AWS CloudWatch	$4,00
WhatsApp API	**$50,00**
Total Estimado	**$105,30**

Cenário 2: 100.000 Atendimentos por Mês

Serviço	Custo (USD)
Amazon API Gateway	$3,50
AWS Lambda	$0,40833
Amazon SQS	$0,80
AWS Lex	$4.000,00
Amazon DynamoDB	$1,50
AWS Athena	$50,00
Amazon S3	$0,23
AWS Route 53	$0,50
AWS WAF	$10,60
AWS CloudWatch	$103,00
WhatsApp API	**$5.000,00**
Total Estimado	**$9.170,60**

Análise dos Custos

- **Principais Custos**:
    - **AWS Lex**: Representa a maior parte dos custos nos dois cenários devido ao preço por solicitação.
    - **WhatsApp Business API**: Também é um custo significativo, especialmente no cenário de alto volume.

- **Custos Escaláveis**: Serviços como AWS Lambda, API Gateway e SQS têm custos que escalam linearmente com o número de solicitações, mas permanecem baixos em comparação com Lex e WhatsApp API.
- **Custos Fixos**: Serviços como AWS Route 53, AWS WAF (parcialmente) e Amazon S3 têm custos que não variam significativamente entre os cenários.

Considerações Finais

- **Otimização de Custos**:
    - **AWS Lex**: Avaliar se é possível otimizar o uso do Lex, talvez reduzindo o número de solicitações ou utilizando alternativas quando aplicável.
    - **WhatsApp API**: Negociar tarifas com o provedor ou considerar estratégias para reduzir o número de conversas cobradas.
- **Monitoramento Contínuo**: Implementar práticas de monitoramento para acompanhar os custos em tempo real e identificar oportunidades de otimização.
- **Validação dos Preços**: Os preços podem variar com o tempo ou de acordo com a região. Recomenda-se consultar a calculadora oficial da AWS e os provedores de API do WhatsApp para obter estimativas atualizadas.

Conclusão

A arquitetura serverless proposta para o chatbot **"Dra Jô"** aproveita de forma eficaz os serviços da AWS para criar uma solução escalável, segura e eficiente. Integrando componentes como AWS Route 53, API Gateway, Lambda, SQS, Lex e DynamoDB, o chatbot é capaz de oferecer um atendimento abrangente aos clientes, atendendo a necessidades financeiras, agrícolas e comerciais, além de envolver especialistas humanos quando necessário.

A estimativa de custos detalhada para diferentes volumes de atendimento demonstra que a solução é economicamente viável e escalável. Os custos operacionais ajustam-se proporcionalmente ao número de atendimentos, com os principais gastos associados ao processamento de linguagem natural (AWS Lex) e à integração com o WhatsApp. Por exemplo, para 1.000 atendimentos mensais, o custo estimado é de aproximadamente **$105,30**, enquanto para 100.000 atendimentos mensais, o custo estimado é de cerca de **$9.170,60**.

Com base nessa análise, a empresa pode planejar seu orçamento de forma eficaz e explorar estratégias para otimizar os custos à medida que a demanda cresce. A combinação de uma arquitetura bem estruturada e uma compreensão clara dos custos operacionais garante a sustentabilidade financeira e operacional do chatbot **"Dra Jô"**, permitindo que ele continue a fornecer um serviço de alta qualidade aos clientes.