Teste Técnico

Arquitetura DSB – Modelo de Classificação

Tiago Assunção Silva

Apresento aqui uma proposta de arquitetura técnica para um sistema de recomendação de matérias, baseada no histórico de navegação nas páginas da Globo.com. Planejo utilizar uma combinação de abordagens de recomendação genérica (por exemplo, as páginas mais visitadas) e específica (por exemplo, filtragem colaborativa). Ressalto que esta é apenas uma solução possível, que pode ser adaptada conforme a necessidade.

1. Coleta de Dados

Implementaremos um rastreador usando JavaScript incorporado em cada página dos portais da Globo (g1, ge, etc). Quando um usuário visita uma página, o navegador do usuário baixa e executa o script. Esse script rastreia as atividades do usuário na página e coleta informações, como o ID do usuário, o URL da página visitada, o tempo gasto na página, se o usuário clicou em algum link na página, a rolagem da página, e o identificador do portal (g1, ge, etc) ao qual a página pertence.

O rastreador então envia esses dados para um único recurso com um método POST no AWS API Gateway. O ponto de entrada principal será uma API REST construída com o AWS API Gateway. Esse recurso da API poderia ser algo como "/pageview", e todos os sinais de pageview seriam enviados para essa única rota, independentemente do portal.

Quando o AWS API Gateway recebe um sinal de pageview, ele aciona uma função AWS Lambda. Essa primeira função Lambda processa o sinal de pageview e, com base no identificador do portal (g1, ge, etc) incluído nos dados, tenta direcionar os dados para o stream de dados correspondente no AWS Kinesis Data Streams.

No caso de falha ao enviar os dados para o AWS Kinesis Data Streams por qualquer motivo, os dados são enviados para uma fila do AWS SQS para serem processados novamente mais tarde. Uma segunda função Lambda é acionada para processar as mensagens da fila do AWS SQS. Ela tenta enviar os dados para o AWS Kinesis Data Streams novamente. Se a tentativa for bem-sucedida, a mensagem é removida da fila do AWS SQS. Se a tentativa falhar, a mensagem não é removida e se torna visível novamente após o período de visibilidade para uma nova tentativa de processamento. Isso garante que nenhuma mensagem seja perdida por causa de erros temporários.

Dessa forma, cada portal terá seu próprio stream de dados no AWS Kinesis Data Streams, permitindo que as recomendações sejam processadas de forma independente para cada portal.

Isso simplifica a arquitetura da API, mantendo a capacidade de processar recomendações de forma independente para cada portal.

Sobre o uso do AWS Kinesis Data Streams em vez do AWS SQS. O AWS SQS e AWS Kinesis Data Streams têm finalidades e casos de uso diferentes, e escolher entre os dois depende do tipo de problema que você está tentando resolver.

- **Volume de Dados e Taxa de Transferência**: AWS Kinesis Data Streams é projetado para coleta, armazenamento e análise de grandes volumes de dados em tempo real. Ele pode consumir e processar dados em terabytes por hora, tornando-o ideal para aplicações de big data como análise de logs em tempo real, coleta de dados de telemetria, processamento de fluxo de eventos etc. Em contrapartida, enquanto o AWS SQS pode lidar com um alto volume de mensagens, ele não é projetado especificamente para tratar grandes volumes de streaming de dados em tempo real.

- **Ordem dos Dados**: Como mencionado anteriormente, AWS Kinesis Data Streams mantém a ordem dos registros dentro de um shard. Isso é útil para análises em tempo real onde a ordem dos eventos é importante. Embora o AWS SQS possa manter a ordem das mensagens com as filas FIFO, isso vem com algumas restrições, como limites na taxa de transferência.

- **Consumo de Dados**: AWS Kinesis Data Streams permite que vários consumidores processem o mesmo fluxo de dados simultaneamente, o que é útil para casos de uso onde o mesmo fluxo de dados precisa ser processado de maneiras diferentes (por exemplo, processamento em tempo real e carregamento em um data warehouse). Com o AWS SQS, uma vez que uma mensagem é consumida e deletada por um consumidor, ela não está mais disponível para outros consumidores.

- **Retenção de Dados**: AWS Kinesis Data Streams pode armazenar dados por até 365 dias, permitindo análise histórica dos dados. O AWS SQS, por outro lado, é um serviço de enfileiramento de mensagens - uma vez que uma mensagem é consumida, ela é removida da fila.

Portanto, a escolha entre AWS SQS e AWS Kinesis Data Streams depende de suas necessidades específicas. Se você precisa apenas de uma solução de enfileiramento de mensagens para desacoplar componentes de sua aplicação, o AWS SQS pode ser uma escolha melhor. Mas se você precisa processar e analisar grandes volumes de dados em tempo real, especialmente onde a ordem dos eventos e o consumo simultâneo por múltiplos consumidores é importante, o AWS Kinesis Data Streams seria mais apropriado.

2. Armazenamento e Processamento de Dados com Auto-Scaling

Criaremos streams de dados do AWS Kinesis Data Streams separados para cada portal (por exemplo, um para G1, outro para GE, etc.). Cada stream será a espinha dorsal do pipeline de dados do portal correspondente, recebendo sinais de pageview em tempo real e distribuindo-os para vários consumidores de dados.

Cada um desses streams terá a capacidade de auto-escala, já que o AWS Kinesis Data Streams lida automaticamente com todas as solicitações de entrada, escalando conforme necessário para acomodar o volume de tráfego, utilizando de parâmetro definidos. Isso significa que o número de shards (ou "linhas de transporte" de dados) dentro de cada stream será ajustado automaticamente com base na demanda. Isso nos permite lidar com picos de tráfego sem a necessidade de intervenção manual, garantindo eficiência de custos e resiliência do sistema.

Um dos consumidores desses streams será o AWS Kinesis Data Firehose, que estará configurado para gravar os dados no AWS S3, na classe de armazenamento S3 Glacier Deep Archive, a classe de armazenamento de menor custo do S3, porém a recuperação de dados pode levar 12 horas ou mais. Escolhi o AWS S3 pela sua escalabilidade, durabilidade e custo-efetividade para armazenamento de grandes volumes de dados. Se precisarmos modificar os dados antes de armazená-los, podemos configurar uma função Lambda no AWS Kinesis Data Firehose para processar os registros conforme eles chegam.

Outro consumidor será o AWS Kinesis Data Analytics, que processará os dados em tempo real conforme eles chegam, transformando-os em um formato adequado para o treinamento do modelo de recomendação. Após o processamento, o AWS Kinesis Data Analytics escreverá os dados transformados de volta para um novo stream do AWS Kinesis Data Streams.

Por fim, teremos o AWS SageMaker ou o AWS Personalize como consumidores desse novo stream do AWS Kinesis Data Streams. Eles serão responsáveis por treinar e servir um modelo de recomendação baseado nesses dados processados.

Essa abordagem de utilizar streams separados para cada portal, cada um com sua própria capacidade de auto-escala, nos permite processar os dados de forma eficiente e isolada para cada portal. Isso é essencial para atender às necessidades específicas de cada portal.

3. Sistema de Recomendação

Para o sistema de recomendação, poderíamos usar o AWS SageMaker ou o AWS Personalize.

Se optarmos pelo AWS SageMaker, o AWS SageMaker é uma plataforma de machine learning totalmente gerenciada que permite aos desenvolvedores e cientistas de dados construir, treinar e implantar modelos de machine learning (ML) rapidamente. AWS SageMaker remove as barreiras que normalmente retardam os desenvolvedores de ML, fornecendo todos os componentes necessários para treinar, ajustar, implementar e gerenciar modelos escalonáveis de ML. Ele suporta uma variedade de algoritmos de aprendizado de máquina e permite que você traga seus próprios algoritmos personalizados ou frameworks de aprendizado de máquina. O AWS SageMaker é uma plataforma de uso geral para machine learning e pode ser usado para uma variedade de aplicações, desde previsão de demanda, detecção de fraude, análise de sentimentos, entre outros.

Por outro lado, se optarmos pelo AWS Personalize, podemos aproveitar os algoritmos integrados de recomendação da plataforma, que se adaptam automaticamente ao comportamento do usuário ao longo do tempo. O AWS Personalize é um serviço de machine learning que facilita o desenvolvimento de recomendações personalizadas para aplicativos. Ele usa algoritmos de machine learning (baseados em anos de experiência da Amazon com recomendações personalizadas) para criar modelos de recomendação que são personalizados para seus usuários

e suas necessidades. Diferentemente do AWS SageMaker, o AWS Personalize é um serviço gerenciado específico para a tarefa de recomendação. Ele cuida de todo o pipeline de machine learning para criar, treinar e implementar um modelo de recomendação. O Personalize é usado quando o objetivo é fornecer recomendações personalizadas para os usuários, como recomendações de produtos em um site de comércio eletrônico ou recomendações de filmes em um serviço de streaming.

Em resumo, enquanto o AWS SageMaker é uma plataforma de uso geral para construir, treinar e implementar modelos de machine learning, o AWS Personalize é um serviço específico para criar sistemas de recomendação personalizados. A escolha entre os dois dependerá do problema específico que você está tentando resolver.

Em ambos os casos, as recomendações podem ser servidas em tempo real e serão expostas por meio de uma API REST, que será consumida pelos portais do grupo Globo. Assim como o AWS SageMaker, a saída do AWS Personalize é normalmente obtida fazendo uma chamada para a API do serviço.

- **Amazon SageMaker**: Quando você implanta um modelo no AWS SageMaker, ele cria um endpoint HTTP que pode ser usado para fazer previsões em tempo real. Quando você envia dados para este endpoint (normalmente em um formato específico, como JSON ou CSV), ele retorna a previsão do modelo. Portanto, a saída do AWS SageMaker é normalmente obtida fazendo uma solicitação HTTP para o endpoint do modelo.

- **Amazon Personalize**: Depois de treinar um modelo no AWS Personalize, você pode obter recomendações usando a API do AWS Personalize. Isso é normalmente feito através de uma chamada para a operação **get_recommendations** ou **get_personalized_ranking**, que retorna uma lista de itens recomendados.

Em ambos os casos, os resultados não são automaticamente escritos em um AWS Kinesis Data Stream ou armazenados no AWS S3. No entanto, você poderia configurar sua aplicação para armazenar os resultados em um AWS Kinesis Data Stream, no AWS S3, ou em outro lugar, se isso for útil para o seu caso de uso.

4. API de Recomendações em Tempo Real

Para expor as recomendações aos portais do Grupo Globo, criaremos uma API REST personalizada. Esta API atuará como uma camada de interface entre os serviços de recomendação (AWS SageMaker ou AWS Personalize) e os portais do Grupo Globo. A API será construída e hospedada usando o AWS API Gateway e AWS Lambda.

- **AWS Lambda**: Implementaremos uma função AWS Lambda que será ativada sempre que a API REST receber uma solicitação. A função conterá a lógica para buscar as recomendações do AWS SageMaker ou AWS Personalize (ou ambos) e retornar os resultados.

- **AWS API Gateway**: Configuraremos o AWS API Gateway para criar a API REST. O API Gateway será responsável por receber as solicitações HTTP, encaminhá-las para a função AWS Lambda e então devolver a resposta da função AWS Lambda ao cliente.

- **Integração entre API Gateway e Lambda**: Configuraremos a integração entre o AWS API Gateway e a função AWS Lambda. Isto é feito através do console do AWS API Gateway, onde podemos especificar qual função AWS Lambda com base no identificador do portal (g1, ge, etc) incluído nos dados, e assim tentar direcionar para o stream de dados correspondente no AWS Kinesis Data Streams.

- **Amazon CloudWatch**: Para monitorar as chamadas à API e a performance da aplicação.

Esta abordagem apresenta as seguintes vantagens:

- **Personalização**: Embora o AWS SageMaker e o AWS Personalize já ofereçam APIs, criar uma API personalizada nos permite maior controle sobre a lógica de negócio e a experiência do usuário. Por exemplo, podemos agregar recomendações de ambos os serviços, implementar lógica de fallback, essa lógica de fallback garante que a aplicação ainda possa fornecer algum valor mesmo quando algumas de suas partes não estão funcionando como esperado, retornando um conjunto padrão de recomendações.

- **Simplificação para os consumidores da API**: Ao expor uma única API para os portais do grupo Globo, simplificamos a integração para os consumidores da API. Eles não precisam se preocupar com os detalhes de como obter as recomendações do AWS SageMaker ou AWS Personalize, pois isso é abstraído pela nossa API.

- **Escalabilidade e desempenho**: Tanto o AWS Lambda quanto o AWS API Gateway podem escalar automaticamente para lidar com o aumento da demanda, garantindo que nossa API possa atender a um grande número de solicitações simultâneas.

- **Princípios semelhantes ao padrão DAO**: Embora a situação seja diferente, a criação desta API personalizada aplica princípios semelhantes ao do padrão DAO (Data Access Object), um padrão de design de software que encapsula o acesso a dados. Assim como o DAO proporciona uma maneira consistente e centralizada de gerenciar o acesso a dados, nossa API oferece uma maneira consistente e centralizada de gerenciar o acesso às recomendações.

O AWS Cognito pode ser usado na API de recomendações para gerenciar a autenticação e autorização de usuários. Veja abaixo como esse processo funcionaria:

- **Autenticação**: Quando um usuário tenta acessar a API de recomendações, o primeiro passo é verificar sua identidade. O AWS Cognito pode fazer isso de várias maneiras, incluindo a verificação de credenciais do usuário (como nome de usuário e senha) ou através de provedores de identidade de terceiros (como Facebook, Google, etc.). Se a autenticação for bem-sucedida, o Cognito gera um conjunto de tokens JWT (JSON Web Tokens), que incluem um ID de token, um token de acesso e um token de atualização. O token de acesso é usado para autorizar solicitações à API de recomendações.

- **Autorização**: Depois que um usuário é autenticado e tem um token de acesso, ele pode fazer solicitações à API de recomendações. Para cada solicitação, a API verifica o token de acesso para garantir que ele é válido e que o usuário tem permissão para acessar o

recurso solicitado. O AWS Cognito facilita esse processo, pois os tokens de acesso que gera incluem informações sobre o usuário e sobre as permissões que ele tem.

- **Gerenciamento de usuários**: Além de autenticação e autorização, o AWS Cognito também fornece recursos para gerenciar usuários. Isso inclui a criação e exclusão de contas de usuário, a alteração de permissões de usuário, o rastreamento de atividades de usuário e muito mais. Esses recursos podem ser úteis para monitorar o uso da API de recomendações e garantir que ela esteja sendo usada de maneira apropriada.

Ao usar o AWS Cognito para autenticação e autorização, é possível garantir que apenas usuários autenticados e autorizados possam acessar a API de recomendações. Isso pode ajudar a proteger os dados dos usuários e garantir que a API seja usada de maneira adequada. Além disso, ao usar o AWS Cognito para gerenciar usuários, é possível ter uma visão clara de quem está usando a API e como eles estão a usando.

5. Balanceamento de Carga e Escalabilidade

Dada a natureza das arquiteturas serverless, AWS Lambda e AWS API Gateway, a escalabilidade e o balanceamento de carga são recursos intrínsecos desses serviços, sem a necessidade de configurações adicionais. No entanto, é importante detalhar como esses recursos são alcançados e os benefícios que proporcionam.

- **AWS API Gateway**: lida automaticamente com todas as solicitações de entrada, escalando conforme necessário para acomodar o volume de tráfego. Isso significa que mesmo durante picos de alta demanda, o API Gateway é capaz de processar todas as solicitações recebidas sem atrasos ou gargalos. Além disso, o AWS API Gateway fornece um endpoint único para nossas APIs, simplificando o gerenciamento e a manutenção.

- **AWS Lambda**: ajusta automaticamente a capacidade para manter o desempenho do código, independentemente da escala da carga de trabalho. Cada função Lambda opera de forma isolada, garantindo que problemas em uma não afetem as outras. O AWS Lambda permite a configuração de políticas de escalonamento, que determinam como o serviço deve se expandir em resposta a mudanças na demanda. Existem duas políticas principais de escalonamento:

    - **Política de escalonamento baseada em destino**: Nessa política, você define um valor de utilização desejado para suas funções. A AWS ajustará a quantidade de execuções simultâneas para manter essa utilização. Se você definir o limite máximo de execuções simultâneas como 100 e a utilização desejada como 70%, a AWS ajustará a quantidade de execuções simultâneas para manter aproximadamente 70 execuções simultâneas em média (70% de 100).

    - **Política de escalonamento passo a passo**: Nessa política, você pode definir um incremento percentual para aumentar a quantidade de execuções simultâneas quando a utilização atual excede um determinado limiar. Por exemplo, você pode definir uma política de escalonamento passo a passo para aumentar a quantidade de execuções simultâneas de uma função AWS Lambda em 10% se

a utilização da função exceder 70%. Aqui, "utilização" refere-se à porcentagem de execuções simultâneas em relação ao limite máximo de execuções simultâneas que você definiu para uma função AWS Lambda específica.

- **AWS Kinesis Data Streams**: escala automaticamente para acomodar os dados que estão sendo enviados para ele, permitindo que o sistema lide com picos de tráfego sem perda de dados. Quando você configura o auto-scaling para um stream do AWS Kinesis Data Streams, você precisa definir três coisas principais:

    - **Métrica de escala**: Esta é a métrica que o auto-scaling usará para determinar se precisa adicionar ou remover shards. Por exemplo, você pode escolher a taxa de entrada de registros se você espera que o número de registros (ou eventos) que você está enviando para o stream varie. Ou você pode escolher a taxa de entrada de bytes se o tamanho dos seus dados for variável e você quiser garantir que o stream tenha capacidade suficiente para lidar com picos de dados de entrada.

    - **Valor do alvo**: Este é o valor que você gostaria que a sua métrica de escala mantivesse. Por exemplo, se você escolheu a taxa de entrada de registros como a sua métrica de escala, você poderia definir o valor do alvo como 1.000 registros por segundo. O auto-scaling então tentará adicionar ou remover shards para manter a taxa de entrada de registros o mais próxima possível de 1.000 registros por segundo.

    - **Limites mínimo e máximo para o número de shards**: Estes são os limites para o número de shards que o seu stream pode ter. O limite mínimo é o número de shards que o seu stream terá, mesmo que a sua métrica de escala seja muito baixa. O limite máximo é o número máximo de shards que o seu stream pode ter, mesmo que a sua métrica de escala seja muito alta.

- **AWS Kinesis Data Analytics**: Suporta auto scaling. Ele faz isso escalonando elasticamente o aplicativo para acomodar a taxa de transferência de dados do seu fluxo de origem e a complexidade da sua consulta na maioria dos cenários. O AWS Kinesis Data Analytics provisiona capacidade na forma de Unidades de Processamento Kinesis (KPUs). Uma única KPU fornece memória (4 GB) e computação e redes correspondentes. O limite padrão para KPUs para o seu aplicativo é de 32, mas você pode solicitar um aumento desse limite se necessário1.

- **AWS SageMaker**: fornece um modelo de hospedagem totalmente gerenciado que suporta o balanceamento de carga e a escalabilidade automática. Ele faz isso através do uso de variantes de modelo. Cada variante é associada a uma instância EC2, e você pode configurar cada variante para usar um certo número de instâncias. O SageMaker automaticamente balanceará o tráfego de inferência entre as instâncias para uma variante. Além disso, o SageMaker também permite a escalabilidade automática, que você pode configurar para ajustar dinamicamente o número de instâncias com base no tráfego de inferência. Isso permite que o SageMaker se adapte a flutuações na demanda.

- **AWS Personalize**: Você não precisa gerenciar a infraestrutura subjacente. O serviço se ajusta automaticamente para lidar com a demanda, seja ela constante ou com picos. Quando você cria uma campanha no Personalize, uma "campanha" é essencialmente uma instância de um modelo de machine learning treinado (também conhecido como "solução") que foi implantado para uso em um aplicativo, você define um valor mínimo de TPS (transações por segundo). Isso é essencialmente uma garantia de capacidade. Se a demanda exceder esse valor, o Personalize escala automaticamente para acomodar o tráfego adicional. Além disso, se o tráfego cair, o Personalize automaticamente reduz a capacidade para economizar custos.

Em resumo, a escalabilidade e o balanceamento de carga são gerenciados de forma eficiente e automática pelos serviços AWS em nossa arquitetura, permitindo que o sistema se adapte às necessidades de tráfego e demanda em tempo real. Isso fornece um uso eficiente dos recursos, garantindo uma alta disponibilidade e desempenho mesmo sob condições de carga elevada.

6. Logging, Monitoramento e Custos

Para garantir a eficiência, segurança e confiabilidade do sistema de recomendação em tempo real, é essencial que tenhamos um sistema robusto de logging e monitoramento em funcionamento. Os serviços da AWS fornecem várias ferramentas que nos ajudarão a alcançar esse objetivo.

- **AWS CloudWatch**: Esta será a nossa principal ferramenta de monitoramento. Com o Amazon CloudWatch, podemos coletar e acompanhar métricas, coletar e monitorar arquivos de log, definir alarmes e reagir automaticamente às mudanças no status de nossos recursos da AWS. O CloudWatch nos oferece visibilidade em tempo real de nosso aplicativo, permitindo que monitoramos os recursos utilizados, a performance do aplicativo e a saúde operacional do mesmo.

- **AWS Budgets**: O AWS CloudWatch também pode ser utilizado para monitorar os gastos na AWS, em combinação com o AWS Budgets. O AWS Budgets permite definir limites de custos personalizados que se alinham com o seu orçamento. Quando esses limites são excedidos, o AWS Budgets pode enviar alertas, que também podem ser encaminhados para o AWS CloudWatch. Assim, você pode criar alarmes do CloudWatch para monitorar o uso estimado e os custos cobrados em sua conta da AWS. Isso lhe permite receber notificações quando o uso ou os custos excedem os valores que você definiu.

- **AWS Kinesis Data Firehose**: A integração do AWS Kinesis Data Firehose com o AWS CloudWatch Logs nos permitirá monitorar e solucionar problemas de maneira eficiente. O AWS Kinesis Data Firehose facilita a coleta, entrega e o carregamento dos dados de streaming para o AWS S3, o AWS Redshift, o AWS Elasticsearch Service e o Splunk. O AWS Kinesis Data Firehose irá capturar, transformar e carregar automaticamente os logs de streaming no destino escolhido.

- **AWS CloudWatch Logs**: Nos permite centralizar os logs de todos os nossos sistemas, aplicativos e serviços da AWS. Podemos visualizar todos os logs em uma única linha do tempo e identificar padrões, detectar anomalias e solucionar problemas.

- **AWS CloudWatch Alarms**: Podemos configurar alarmes e notificações com base em métricas específicas. Isso nos permitirá receber notificações em tempo real sobre qualquer problema potencial que possa afetar a performance do nosso sistema.

- **AWS CloudWatch Events**: CloudWatch Events nos ajudará a responder a mudanças no estado dos nossos recursos da AWS. Com isso, podemos criar regras que correspondem a eventos específicos e encaminhar automaticamente esses eventos para funções AWS Lambda, AWS Kinesis Data Streams ou outros targets para lidar com eles.

No geral, a combinação dessas ferramentas e serviços nos ajudará a criar um sistema de recomendação em tempo real altamente monitorado e confiável.

Abaixo listo alguns exemplos de integrações entre o AWS CloudWatch e algumas ferramentas utilizadas:

AWS API Gateway:

- **Monitoramento de métricas**: O AWS CloudWatch coleta automaticamente métricas do AWS API Gateway, como o número de chamadas de API bem-sucedidas, o número de erros de servidor, a latência do API Gateway e a latência de integração. Você pode visualizar essas métricas no console do AWS CloudWatch e criar painéis personalizados para acompanhar o desempenho do seu AWS API Gateway.

- **Registro de logs detalhados**: Você pode configurar o AWS API Gateway para registrar solicitações e respostas de API detalhadas no AWS CloudWatch. Isso inclui detalhes do corpo da solicitação, cabeçalhos, caminhos de URL e parâmetros de consulta. Esses logs podem ser úteis para depuração e rastreamento de problemas.

- **Criação de alarmes**: O AWS CloudWatch permite criar alarmes baseados em métricas específicas. Por exemplo, você pode criar um alarme que envia uma notificação sempre que a latência do API Gateway excede um determinado limite. Isso pode ajudá-lo a identificar e responder rapidamente a problemas de desempenho.

- Para configurar o AWS CloudWatch para o AWS API Gateway, você precisa habilitar o registro de logs e escolher o nível de detalhe do log no console do AWS API Gateway. Você também precisa garantir que o AWS API Gateway tenha as permissões necessárias para gravar logs no AWS CloudWatch.

AWS Lambda:

- No contexto do AWS Lambda, o AWS CloudWatch fornece métricas integradas, como o número de invocações de função, a duração das invocações, erros, etc. Você também pode criar alarmes baseados nessas métricas para notificá-lo ou tomar ações automáticas quando, por exemplo, o número de erros excede um determinado limite.

- Além disso, o AWS CloudWatch é uma ferramenta poderosa para trabalhar com logs. Quando você habilita o registro de logs para a sua função Lambda, os logs são enviados

para o AWS CloudWatch Logs. Isso permite que você visualize, pesquise e faça o download dos logs para análise posterior.

AWS Kinesis Data Streams:

- O AWS CloudWatch pode monitorar a quantidade de dados que estão sendo enviados para o AWS Kinesis Data Streams, bem como a latência de leitura dos dados. Isso pode nos ajudar a identificar quaisquer gargalos de desempenho.

Dessa forma, o AWS CloudWatch nos proporciona uma visibilidade abrangente do sistema, incluindo a utilização de recursos, o desempenho da aplicação e a saúde operacional. Além disso, com o AWS CloudWatch, podemos reagir rapidamente a alterações no ambiente da AWS, o que pode ser útil para identificar e reagir a problemas de desempenho ou gargalos antes que eles afetem o serviço ao usuário final.

7. Segurança e Redes

A segurança é uma prioridade máxima em qualquer aplicação, especialmente em sistemas que gerenciam dados sensíveis dos usuários. Aqui estão algumas das práticas recomendadas e serviços AWS que podem ser usados para garantir a segurança no seu sistema de recomendação.

7.1. Gerenciamento de identidade e acesso

Use o AWS Identity and Access Management (IAM) para controlar o acesso aos seus recursos da AWS. Você pode criar usuários IAM com permissões específicas, garantindo que cada entidade (seja uma pessoa ou um serviço AWS) tenha as permissões mínimas necessárias para realizar seu trabalho. Isso é conhecido como o princípio do menor privilégio.

- **Usuários IAM**: Em IAM, um usuário é uma entidade que você cria na AWS. O usuário representa a pessoa ou serviço que usa o usuário para interagir com a AWS. Um usuário em AWS consiste em um nome e credenciais.

- **Grupos IAM**: Um grupo é uma coleção de usuários do AWS IAM. Você pode usar grupos para especificar permissões para um conjunto de usuários, o que pode tornar mais fácil gerenciar as permissões para esses usuários.

- **Políticas e permissões**: Você gerencia o acesso em AWS ao criar políticas e anexá-las a usuários, grupos ou funções do AWS IAM. Uma política é um objeto no AWS que, quando associado a uma identidade ou recurso, define suas permissões.

- **Funções IAM**: Você pode criar uma função do AWS IAM e definir as permissões para a função para permitir que serviços e usuários assumam essa função. Ao assumir uma função, os serviços e usuários obtêm temporariamente as permissões para os recursos da AWS que você permitiu.

- **Autenticação Multi-Fator (MFA)**: Para fornecer um nível adicional de segurança, você pode ativar a autenticação AWS MFA para a sua conta AWS e para usuários individuais do AWS IAM. A AWS MFA exige que o usuário apresente duas formas de identificação independentes: algo que o usuário conhece (senha) e algo que o usuário tem (dispositivo de autenticação).

- **Rotação de chaves de acesso**: Você pode (e deve) girar as chaves de acesso regularmente. O IAM facilita a rotação de suas chaves de acesso sem interrupção para suas aplicações.

Usar o IAM, você pode estabelecer uma estratégia de segurança robusta para sua organização, garantindo que apenas usuários e serviços autorizados tenham acesso aos seus recursos AWS.

7.2. Criptografia

Use a criptografia para proteger seus dados em trânsito e em repouso. Os serviços AWS, como o AWS Kinesis Data Streams e o AWS S3, oferecem a opção de criptografar os dados armazenados. Da mesma forma, você deve garantir que todos os dados enviados por redes não seguras sejam criptografados usando SSL/TLS.

7.3. Proteção de rede

Use grupos de segurança e listas de controle de acesso à rede (NACLs) para proteger sua rede. Os grupos de segurança atuam como um firewall de nível de instância que controla o tráfego de entrada e saída. As NACLs, por outro lado, atuam como um firewall no nível do VPC e controlam o tráfego de entrada e saída para todas as sub-redes dentro do VPC.

- **NACLs (Network Access Control Lists)**: As NACLs são um recurso opcional que você pode usar para adicionar uma camada adicional de segurança ao seu VPC. Elas atuam como um firewall para controlar o tráfego de entrada e saída em um nível associado a sub-redes dentro de um VPC. Elas permitem que você crie regras de permissões de entrada e saída que são aplicadas ao tráfego de IP de entrada e saída que atravessa a sub-rede associada.

- **VPC (Virtual Private Cloud)**: O Amazon Virtual Private Cloud (Amazon VPC) permite que você provisione uma seção isolada logicamente da Nuvem AWS onde você pode lançar recursos da AWS em uma rede virtual que você define. Você tem controle completo sobre seu ambiente de rede virtual, incluindo seleção de intervalo de endereços IP próprios, criação de sub-redes e configuração de tabelas de rotas e gateways de rede.

Em termos mais simples, um VPC pode ser pensado como sua própria rede privada na nuvem. Dentro de um VPC, você pode definir suas próprias sub-redes, controlar como as sub-redes são roteadas, configurar gateways de internet e muito mais.

7.4. Gerenciamento de chaves

O AWS Key Management Service (KMS) pode ser usado para criar e gerenciar chaves criptográficas usadas para criptografar os dados. A AWS KMS é integrada a outros serviços da AWS para ajudar a proteger os dados que você armazena nesses serviços e as chaves que você usa para criptografar os dados.

7.5. Autenticação e autorização

Para API de recomendações em tempo real, é importante garantir que apenas clientes autenticados possam acessar as recomendações. O AWS Cognito é um serviço que facilita a

adição de autenticação, autorização e gerenciamento de usuários para suas aplicações web e móveis.

7.6. Monitoramento e auditoria

Use o AWS CloudTrail para rastrear, auditar e monitorar as ações feitas na sua conta da AWS. O CloudTrail registra todas as ações feitas na AWS Management Console, AWS SDKs, linhas de comando e outros serviços da AWS. Isso pode ser útil para a detecção de comportamento suspeito ou não autorizado.

7.7. Proteção DDoS

Proteja suas aplicações contra os ataques DDoS usando o AWS Shield, um serviço gerenciado que fornece proteção contra os tipos mais comuns de ataques DDoS.

Desenho da Arquitetura