

Soluções de dados de streaming na AWS com o Amazon Kinesis

Julho de 2017



Avisos

Este documento é fornecido apenas para fins informativos. Ele relaciona as atuais ofertas de produtos e práticas da AWS na data de emissão deste documento, que estão sujeitas a alterações sem aviso prévio. Os clientes são responsáveis por fazer sua própria avaliação independente das informações neste documento e de qualquer uso dos produtos ou serviços da AWS, cada um dos quais é fornecido “como está”, sem garantia de qualquer tipo, expressa ou implícita. Este documento não cria quaisquer garantias, representações, compromissos contratuais, condições ou promessas da AWS, suas afiliadas, fornecedores ou licenciadores. As responsabilidades e obrigações da AWS para com seus clientes são controladas por contratos da AWS, e este documento não modifica nem faz parte de qualquer contrato entre a AWS e seus clientes.

Sumário

Introdução	1
Cenários de aplicativos em tempo real	1
Diferença entre processamento de streams e em lote	2
Desafios do processamento de streams	2
Do lote ao tempo real: um exemplo	3
Cenário de exemplo: notificação de faturamento e cobrança de pedágio	4
Requisito 1: mais dados recentes no data warehouse	5
Amazon Kinesis Firehose	6
Requisito 2: alertas de limite de faturamento	13
Amazon Kinesis Analytics	14
Amazon Kinesis Streams	17
Requisito 3: outros alertas de limite	22
Arquitetura completa	24
Conclusão	24
Contribuidores	25

Resumo

Os engenheiros de dados, os analistas de dados e os desenvolvedores de big data estão procurando evoluir as análises de lote em tempo real para que as empresas possam saber mais sobre o que seus clientes, aplicativos e produtos estão fazendo e reagir prontamente. Este whitepaper discute a evolução das análises de lote em tempo real. Ele descreve como serviços como o Amazon Kinesis Streams, o Amazon Kinesis Firehose e o Amazon Kinesis Analytics podem ser usados para implementar aplicativos em tempo real e fornece os padrões de design comuns usando esses serviços.

Introdução

Atualmente, as empresas recebem dados em altíssima escala e velocidade devido ao crescimento explosivo das fontes de dados que geram fluxos de dados continuamente. Sejam dados de log de servidores de aplicativos, dados de clickstream de sites e de aplicativos móveis ou dados de telemetria de Internet das Coisas (IoT), todos eles contêm informações que podem ajudar você a saber mais sobre o que seus clientes, aplicativos e produtos estão fazendo agora. Com a capacidade de processar e analisar esses dados em tempo real, é essencial realizar ações como monitorar continuamente seus aplicativos para garantir a alta disponibilidade do serviço e personalizar ofertas promocionais e recomendações de produtos. O processamento em tempo real também pode tornar outros casos de uso comuns, como análise de sites e machine learning, mais precisos e acionáveis por meio de dados disponíveis para esses aplicativos em segundos ou minutos, em vez de horas ou dias.

Cenários de aplicativos em tempo real

Há dois tipos de cenários de caso de uso para aplicativos de dados de streaming:

- **Evolução de lote para análise de streaming**

Você pode executar análises em tempo real com dados que foram tradicionalmente analisados usando o processamento em lote nos data warehouses ou usando estruturas do Hadoop. Os casos de uso mais comuns nessa categoria incluem data lakes, ciência de dados e machine learning. Você pode usar soluções de dados de streaming para carregar continuamente dados em tempo real em seus data lakes. Você também pode atualizar modelos de machine learning com mais frequência à medida que novos dados ficam disponíveis, garantindo a precisão e a confiabilidade das saídas. Por exemplo, a Zillow usa o Amazon Kinesis Streams para coletar dados de registro públicos e anúncios da MLS e, em seguida, fornece a compradores e vendedores de imóveis as estimativas de valor mais recentes praticamente em tempo real. A Zillow também envia os mesmos dados para o data lake do Amazon Simple Storage Service (S3) usando o Kinesis Streams para que todos os aplicativos trabalhem com as informações mais recentes.

- **Criação de aplicativos em tempo real**

Você pode usar os serviços de dados de streaming em aplicativos em tempo real, como monitoramento de aplicativos, detecção de fraudes e placares ao vivo. Esses casos de uso exigem latências de ponta a ponta de milissegundos, desde a ingestão até o processamento, bem como durante a transmissão dos resultados para armazenamentos de dados de destino e outros sistemas. Por exemplo, a Netflix usa o Kinesis Streams para monitorar as comunicações entre todos os seus aplicativos para detectar e corrigir problemas rapidamente, garantindo alta disponibilidade do serviço para seus clientes. Embora o caso de uso mais comumente aplicável seja monitorar a performance do aplicativo, há um número crescente de aplicativos em tempo real, como jogos, tecnologia de anúncios e IoT, que estão sob esta categoria.

Diferença entre processamento de streams e em lote

Você precisa de um conjunto diferente de ferramentas para coletar, preparar e processar dados de streaming em tempo real do que as ferramentas usadas tradicionalmente para análises em lote. Com a análise tradicional, você reúne dados, os carrega periodicamente em um banco de dados e os analisa horas, dias ou semanas depois. A análise de dados em tempo real requer uma outra abordagem. Em vez de executar consultas de banco de dados em dados armazenados, os aplicativos de processamento de stream processam dados continuamente em tempo real, mesmo antes de serem armazenados. Os dados de streaming podem entrar rapidamente, e os volumes de dados podem variar para cima e para baixo a qualquer momento. As plataformas de processamento de dados de streaming precisam ser capazes de lidar com a velocidade e variabilidade dos dados recebidos e processá-los à medida que chegam, muitas vezes entre milhões e centenas de milhões de eventos por hora.

Desafios do processamento de streams

O processamento de dados em tempo real assim que eles chegam permite tomar decisões com muito mais rapidez do que é possível com as tecnologias de análise de dados tradicionais. No entanto, criar e operar seus próprios pipelines personalizados de dados de streaming é complicado e usa muitos recursos. Você precisa criar um sistema que possa coletar, preparar e transmitir dados que chegam simultaneamente de milhares de fontes de dados com economia. Você precisa ajustar o armazenamento e os recursos de computação para que os dados sejam transformados em lote e transmitidos com eficiência para obter

throughput máximo e baixa latência. Você precisa implantar e gerenciar uma frota de servidores para dimensionar o sistema para que seja possível lidar com as diferentes velocidades de dados que você adicionará. Após ter criado essa plataforma, você precisa monitorar o sistema e recuperar a partir de qualquer servidor ou falhas de rede ao acompanhar o processamento de dados no momento adequado no stream, sem criar dados duplicados. Tudo isso custa tempo e dinheiro e, no final, a maioria das empresas nunca consegue concluir o processo e acabam realizando ações regulares, operando seus negócios com informações ultrapassadas.

Do lote ao tempo real: um exemplo

Para compreender melhor como as organizações estão evoluindo de lote para processamento de stream com a AWS, vamos analisar um exemplo. Nele, vamos analisar um cenário e discutir em detalhes como serviços da AWS ([Amazon Kinesis Streams](#),¹ [Amazon Kinesis Firehose](#)² e [Amazon Kinesis Analytics](#)³) são usados para resolver o problema.

Processamento em lote é uma prática comum para o processamento de dados. As organizações normalmente executam trabalhos regulares para analisar seus dados com uma frequência relevante para seu caso de uso. Por exemplo, uma organização pode executar um processo no final do mês para determinar a quantidade de faturamento para cada um de seus clientes. Ou pode executar uma tarefa por hora para analisar os logs de seus aplicativos de TI para determinar os erros ocorridos na última hora. Embora esses processos mensais ou por hora sejam valiosos, o que aconteceria se os mesmos dados pudessem ser analisados assim que fossem criados? Há informações adicionais que podem ser obtidas ou valor adicional que pode ser criado?

Pense novamente no cenário de faturamento mensal. Ao analisar os dados de uso de um cliente à medida que são gerados, uma organização pode habilitar recursos importantes, como notificar os usuários que eles estão se aproximando de um limite de faturamento predefinido. Se os logs de aplicativos de TI puderem ser analisados em tempo real, um administrador do sistema pode ser notificado imediatamente para investigar e executar uma ação corretiva.

Agora vamos combinar esses dois em um único cenário e analisar como criar uma solução.

Cenário de exemplo: notificação de faturamento e cobrança de pedágio

Neste exemplo simplificado, uma empresa fictícia, a ABC Pedágios, opera pedágios em estradas em todo o país. Os clientes registrados na ABC Pedágios recebem um transceptor para seus carros. Quando o cliente dirige por uma área pedagiada, um sensor recebe informações do transceptor e registra os detalhes da transação em um banco de dados relacional. A ABC Pedágios tem uma arquitetura de lote tradicional. Todo dia, um processo programado de extração, transformação e carregamento (ETL) é executado. Ele processa as transações diárias e as transforma para que possam ser carregadas em seu data warehouse. No próximo dia, os analistas de negócios da ABC Pedágios analisam os dados usando uma ferramenta de relatórios. Além disso, uma vez por mês (no final do ciclo de faturamento), outro processo agrega todas as transações para cada um dos clientes da ABC Pedágios para calcular o pagamento mensal.

A ABC Pedágios gostaria de fazer algumas modificações em seu sistema. O primeiro requisito vem de sua equipe de analistas de negócios. Eles solicitaram a capacidade de executar relatórios de seu data warehouse com dados de no máximo 30 minutos atrás.

A ABC Pedágios também está desenvolvendo um novo aplicativo móvel para seus clientes. Durante o desenvolvimento do aplicativo, ela decidiu criar alguns novos recursos. Um recurso fornece aos clientes a capacidade de definir um limite de gastos para sua conta. Se a fatura acumulada de pedágio do cliente passar desse limite, a ABC Pedágios quer enviar uma mensagem no aplicativo ao cliente para notificá-lo que o limite foi ultrapassado em até 10 minutos após a cobrança.

Por fim, a equipe de operações da ABC Pedágios tem alguns requisitos adicionais que gostariam de introduzir ao sistema. Durante o monitoramento das cabines de pedágio, ela deseja ser notificada imediatamente quando o tráfego de veículos de uma determinada cabine cair abaixo de um limite predefinido para cada período de 30 minutos em um dia. Por exemplo, ela já sabem devido ao histórico de dados que aproximadamente 360 veículos passam por aquela cabine às quartas-feiras entre 14h e 14h30. Nesse janela de 30 minutos, a equipe de operações deseja ser notificada se menos de 100 veículos passarem pela cabine de pedágio. Os operadores podem então investigar para determinar se o tráfego está normal, ou se algum outro fator contribuiu para o valor inesperado (por exemplo, um sensor defeituoso ou um acidente na estrada).

A equipe de engenharia da ABC Pedágios determina que sua arquitetura atual precisa de algumas modificações para oferecer suporte a esses requisitos. Ela decidiu criar um sistema de ingestão e análise de dados de streaming para oferecer suporte aos requisitos. Vamos analisar cada requisito e analisar os aprimoramentos da arquitetura para cada requisito.

Requisito 1: mais dados recentes no data warehouse

Atualmente, os dados no data warehouse da ABC Pedágios podem ser de até 24 horas atrás por causa de seu processo diário de lote. A solução de data warehouse atual é o Amazon Redshift. Ao analisar os recursos de serviços do Amazon Kinesis, a empresa reconheceu que o Kinesis Firehose pode receber um stream de registros de dados e inseri-los no Amazon Redshift. Ela criou um fluxo de entrega do Kinesis Firehose e o configurou para que ele possa copiar dados para a tabela do Amazon Redshift a cada 15 minutos. A solução atual armazena registros para um sistema de arquivos como parte do processo de lote. Como parte dessa nova solução, a empresa usou o Amazon Kinesis Agent em seus servidores para encaminhar os dados de log para o Kinesis Firehose. Como o Kinesis Firehose usa o Amazon S3 para armazenar dados de streaming brutos antes de serem copiados para o Amazon Redshift, a ABC Pedágios não precisa criar outra solução para arquivar seus dados brutos.

A Figura 1 mostra essa solução.

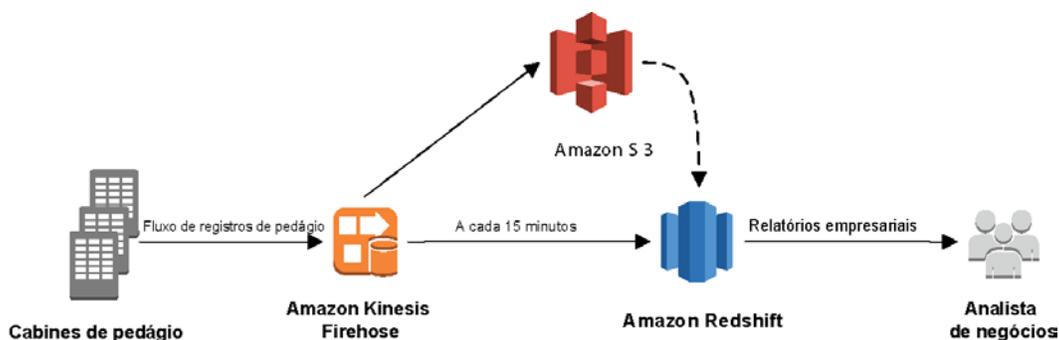


Figura 1: Nova solução usando o Amazon Kinesis Firehose

Para esta parte da arquitetura, a ABC Pedágios escolheu o Kinesis Firehose. Vamos analisar os recursos do Kinesis Firehose em detalhes.

Amazon Kinesis Firehose

O Amazon Kinesis Firehose é a forma mais fácil de carregar dados de streaming no AWS. Ele pode capturar, transformar e carregar os dados de streaming no Amazon Kinesis Analytics, no Amazon S3, no Amazon Redshift e no Amazon Elasticsearch Service, permitindo análise quase em tempo real com os painéis e as ferramentas de inteligência de negócios existentes que você já usa atualmente. É um serviço totalmente gerenciado que escala automaticamente para atender ao throughput dos seus dados e exige uma administração contínua. Ele pode separar em lotes, compactar e criptografar os dados antes de carregá-los, o que minimiza o volume de armazenamento usado no destino e aumenta a segurança.

O Kinesis Firehose é um serviço totalmente gerenciado. Você não precisa escrever aplicativos ou gerenciar recursos. Você configura os produtores de dados para enviar dados ao Kinesis Firehose, que entregará automaticamente os dados ao destino especificado. Você também pode configurar o Kinesis Firehose para transformar os dados antes da entrega dos mesmos.

Envio de dados a um fluxo de entrega do Amazon Kinesis Firehose

Para enviar dados ao fluxo de entrega, há várias opções. A AWS oferece SDKs para várias linguagens de programação populares, sendo que cada um deles fornece APIs para o Kinesis Firehose. A AWS também criou um utilitário para ajudar a enviar dados para o fluxo de entrega.

Uso da API

O Kinesis Firehose API oferece duas operações para envio dos dados ao fluxo de entrega. `PutRecord` envia um registro de dados em uma chamada.

`PutRecordBatch` pode enviar vários registros de dados em uma chamada.

Em cada método, você deve especificar o nome do fluxo de entrega e o registro de dados ou matriz de registros de dados, ao usar o método. Cada registro de dados consiste em um blob de dados que pode ter até 1.000 KB de tamanho e qualquer tipo de dados.

Para obter informações detalhadas e códigos de exemplo para as operações de API do Kinesis Firehose, consulte [Writing to a Firehose Delivery Stream Using the AWS SDK](#)⁴.

Uso do Amazon Kinesis Agent

O Amazon Kinesis Agent é um aplicativo de software independente de Java que oferece uma maneira fácil de coletar e enviar dados ao Kinesis Streams e ao Kinesis Firehose. O agente monitora continuamente um conjunto de arquivos e envia novos dados ao stream. Ele administra o rodízio de arquivos, os pontos de verificação e as novas tentativas após falhas. Seus dados são entregues de maneira confiável, imediata e simples. Ele também emite métricas do Amazon CloudWatch para facilitar o monitoramento e a solução de problemas no processo de streaming.

Você pode instalar o agente em ambientes de servidor baseados em Linux, como servidores web, servidores de log e servidores de banco de dados. Após instalar o agente, configure-o especificando os arquivos a serem monitorados e o fluxo de destino para os dados. Depois que o agente é configurado, ele coleta dados dos arquivos de forma durável e os envia de forma confiável ao fluxo de entrega.

O agente pode monitorar vários diretórios de arquivos e gravar em vários fluxos. Ele também pode ser configurado para pré-processar registros de dados antes que eles sejam enviados para seu stream ou fluxo de entrega.

Se você estiver considerando a migração de um sistema de arquivos em lote tradicional para streaming de dados, é possível que os seus aplicativos já estejam registrando eventos para arquivos nos sistemas de arquivos de seus servidores de aplicativos. Ou, se o seu aplicativo usa uma biblioteca de logs popular (como o Log4j), configurá-lo para gravar arquivos locais é geralmente uma tarefa simples. Não importa como os dados são gravados em um arquivo de log; você deve considerar usar o agente neste cenário. Ele fornece uma solução simples que requer pouca ou nenhuma alteração em seu sistema existente. Em muitos casos, ele pode ser usado simultaneamente com a sua solução de lote existente. Nesse cenário, ele fornece um fluxo de dados no Kinesis Streams, usando os arquivos de log como fonte de dados para o stream.

Em nosso cenário de exemplo, a ABC Pedágios escolheu usar o agente para enviar dados de streaming para o fluxo de entrega. Ela já estava criando arquivos de log. Por isso, o encaminhamento das entradas de log para o Kinesis Firehose exige apenas a instalação e a configuração do agente. Nenhum outro código era necessário para iniciar o streaming de seus dados.

Transformação de dados

Em alguns cenários, você pode transformar ou aprimorar o streaming de dados antes de ser entregue ao seu destino. Por exemplo, produtores de dados podem enviar texto não estruturado em cada registro de dados, e você precisa transformá-lo em JSON antes de encaminhá-lo para o Amazon Elasticsearch Service.

Para habilitar a transformações de dados de streaming, o Kinesis Firehose usa uma função do [AWS Lambda](#) que você cria para transformar seus dados.⁵

Fluxo de transformação de dados

Quando você habilita a transformação de dados do Kinesis Firehose, o Kinesis Firehose armazena em buffer os dados de entrada de até 3 MB ou até o tamanho de buffer especificado para o fluxo de entrega, o que for menor. O Kinesis Firehose chama a função Lambda especificada com cada lote armazenado em buffer de forma assíncrona. Os dados transformados são enviados do Lambda para o Kinesis Firehose para armazenamento em buffer. Eles são entregues no destino quando o tamanho ou o intervalo de buffer especificado é atingido, o que acontecer primeiro. A Figura 2 ilustra esse processo para um fluxo de entrega que fornece dados para o Amazon S3.



Figura 2: Buffering de dados usando o Kinesis Firehose e funções Lambda

Entrega de dados

Após os limites de armazenamento em buffer do fluxo de entrega terem sido atingidos, os dados são entregues no destino configurado. Há algumas diferenças na forma como o Kinesis Firehose fornece dados para cada destino, que analisaremos nas seções a seguir.

Amazon Simple Storage Service

O [Amazon S3](#) armazena objetos com uma interface simples de serviço web que pode ser usada para armazenar e recuperar qualquer quantidade de dados de qualquer lugar na web.⁶ Ele foi projetado para fornecer 99,999999999% de durabilidade e ajusta a escala de trilhões de objetos pelo mundo. Use o Amazon S3 como armazenamento principal de aplicativos nativos da nuvem; como repositório em massa, ou “data lake”, para análise; e como destino de backup, recuperação e recuperação de desastres.

Formato de entrega de dados

Para a entrega de dados para o Amazon S3, o Kinesis Firehose concatena vários registros de entrada com base na configuração de armazenamento em buffer do fluxo de entrega e, em seguida, os entrega ao Amazon S3 como um objeto do S3. Adicione um separador de registro ao final de cada registro antes de enviá-lo ao Kinesis Firehose, para que seja possível dividir um objeto do S3 entregue em registros individuais.

Frequência de entrega de dados

A frequência da entrega de dados para o Amazon S3 é determinada pelos valores de tamanho e intervalo do buffer do S3 configurados para o fluxo de entrega. O Kinesis Firehose armazena em buffer os dados recebidos antes de encaminhá-los para o Amazon S3. Você pode configurar os valores de tamanho de buffer do Amazon S3 (1 MB a 128 MB) ou de intervalo de buffer (60 a 900 segundos). A condição satisfeita primeiro acionará a entrega de dados para o Amazon S3. Observe que, quando a entrega de dados no destino ficar atrás da gravação de dados no fluxo de entrega, o Kinesis Firehose aumentará o tamanho do buffer dinamicamente para alcançar a gravação e garantir que todos os dados serão entregues no destino.

Fluxo de dados

A Figura 3 mostra o fluxo de dados para os destinos do Amazon S3.



Figura 3: Fluxo de dados do Kinesis Firehose para buckets do S3

Amazon Redshift

O [Amazon Redshift](#) é um data warehouse rápido e totalmente gerenciado que torna mais simples e econômica a análise de todos os seus dados usando SQL padrão e as ferramentas de inteligência de negócios de que você já dispõe.⁷ Ele permite que você execute consultas analíticas complexas contra petabytes de dados estruturados usando a otimização de consulta sofisticada, armazenamento em colunas em discos locais, de alta performance e execução massiva da consulta paralela. A maioria dos resultados retorna em segundos.

Em nosso exemplo, a ABC Pedágios já estava usando o Amazon Redshift como sua solução de data warehouse. Quando ela implementou a solução de streaming de dados, já configurou seu fluxo de entrega para entregar os dados de streaming para o seu cluster do Amazon Redshift existente.

Formato de entrega de dados

Para a entrega de dados ao Amazon Redshift, o Kinesis Firehose primeiro entrega os dados de entrada ao bucket do S3 no formato descrito anteriormente. Depois, o Kinesis Firehose emite um comando COPY do Amazon Redshift para carregar os dados do bucket do S3 para o cluster do Amazon Redshift. É necessário verificar se, depois que o Kinesis Firehose concatena vários registros de entrada em um objeto do S3, esse objeto pode ser copiado para o cluster do Amazon Redshift. Para obter mais informações, consulte [Amazon Redshift COPY Command Data Format Parameters](#).

Frequência de entrega de dados

A frequência das operações do comando COPY de dados do Amazon S3 para o Amazon Redshift é determinada pela rapidez com que o cluster do Amazon Redshift pode concluir o comando COPY. Se ainda houver dados a serem copiados, o Kinesis Firehose emitirá um novo comando COPY assim que o comando COPY anterior for concluído com êxito pelo Amazon Redshift.

Fluxo de dados

A Figura 4 mostra o fluxo de dados para destinos do Amazon Redshift.

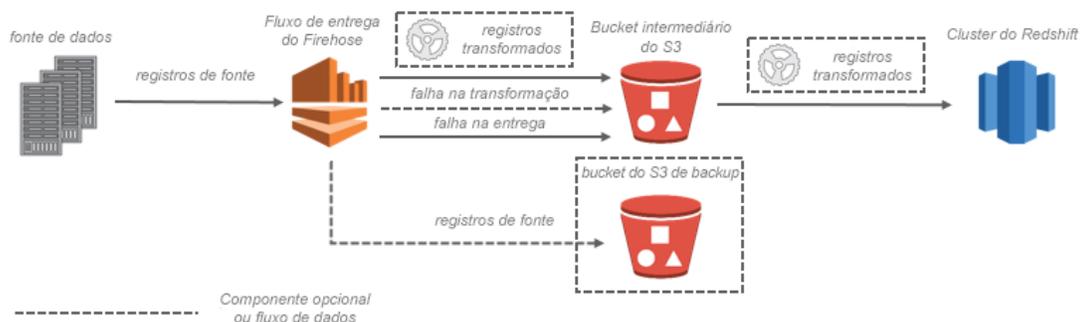


Figura 4: Fluxo de dados do Kinesis Firehose para o Amazon Redshift

Amazon Elasticsearch Service

O [Amazon Elasticsearch Service](#) (Amazon ES) é um serviço totalmente gerenciado que fornece APIs e recursos em tempo real do Elasticsearch fáceis de usar, além da disponibilidade, da escalabilidade e da segurança exigidas pelas cargas de trabalho de produção.⁸ O Amazon ES facilita a implantação, a operação e o dimensionamento do Elasticsearch para análise de log, pesquisa por texto completo, monitoramento de aplicativos, entre outros.

Formato de entrega de dados

Para entrega de dados para o Amazon ES, o Kinesis Firehose armazena em buffer os registros de entrada com base na configuração de armazenamento em buffer do fluxo de entrega e, em seguida, gera uma solicitação em massa do Elasticsearch para indexar vários registros no cluster do Elasticsearch. É necessário verificar se o registro apresenta codificação UTF-8 e está nivelado para um objeto JSON de linha única antes de enviá-lo ao Kinesis Firehose.

Frequência de entrega de dados

A frequência da entrega de dados no Amazon ES é determinada pelos valores de tamanho e intervalo do buffer do Elasticsearch configurados para o fluxo de entrega. O Kinesis Firehose armazena em buffer os dados recebidos antes de encaminhá-los para o Amazon ES. Você pode configurar os valores de tamanho de buffer do Elasticsearch (1 MB a 100 MB) ou de intervalo de buffer (60 a 900 segundos). A condição satisfeita primeiro acionará a entrega de dados para o Amazon ES. Observe que, quando a entrega de dados no destino ficar atrás da gravação de dados no fluxo de entrega, o Kinesis Firehose aumentará o tamanho do buffer dinamicamente para alcançar a gravação e garantir que todos os dados serão entregues no destino.

Fluxo de dados

A Figura 5 mostra o fluxo de dados para o Amazon ES destinos.



Figura 5: A entrega de dados do Kinesis Firehose para o Amazon ES cluster

Resumo

O Kinesis Firehose é a maneira mais fácil de manter seus dados de streaming para um destino compatível. É uma solução totalmente gerenciada, que requer pouco ou nenhum desenvolvimento para usar. Para a ABC Pedágios, usar o Kinesis Firehose foi uma escolha natural. Ela já estava usando o Amazon Redshift como solução de data warehouse. E como suas fontes de dados

estavam continuamente gravando em logs de transação, elas foram capazes de otimizar o Amazon Kinesis Agent para fazer streaming de dados sem escrever qualquer código adicional.

Agora que a ABC Pedágios criou um fluxo de registros de pedágio e está recebendo esses registros por meio do Kinesis Firehose, pode usá-los como base para seus outros requisitos de dados de streaming.

Requisito 2: alertas de limite de faturamento

Para desenvolver o recurso de envio de uma notificação quando um limite de gastos é alcançado, a equipe de desenvolvimento da ABC Pedágios criou um aplicativo móvel e uma tabela do [Amazon DynamoDB](#)⁹. O aplicativo permite que os clientes definam o limite, e a tabela armazena esse valor para cada cliente. A tabela também é usada para armazenar o valor cumulativo gasto por cada cliente, a cada mês. Para oferecer notificações oportunas, a ABC Pedágios precisa atualizar o valor acumulado nessa tabela em tempo hábil e compará-lo com o limite para determinar se uma notificação deve ser enviada para o cliente. Como suas transações de pedágio já estão sendo transmitidas por meio do Kinesis Firehose, ela decidiu usar esses dados de streaming como origem para sua agregação e alertas. E como o Kinesis Analytics permite usar o SQL para agregar dados de streaming, ele é a solução ideal para o problema. Nesta solução, o Kinesis Analytics totaliza o valor das transações para cada cliente ao longo de um período (janela) de 10 minutos. No final da janela, ele envia os totais para um fluxo do Kinesis. Esse stream é a origem dos eventos para uma função do AWS Lambda. A função Lambda consulta a tabela do DynamoDB para recuperar os limites e total gasto atual para cada cliente representado na saída do Kinesis Analytics. Para cada cliente, a função Lambda atualiza o total atual no DynamoDB e também compara o total com o limite. Se o limite for excedido, ele usa o AWS SDK para instruir o Amazon Simple Notification Service (SNS) para enviar uma notificação para os clientes.

A Figura 6 mostra a arquitetura para esta solução.

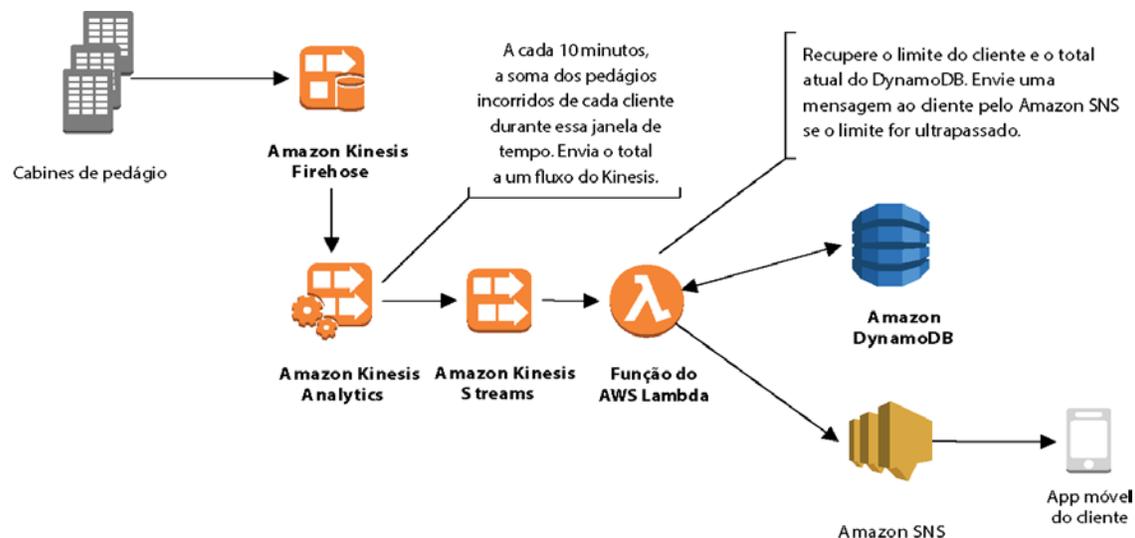


Figura 6: Arquitetura para alertas e notificações de limite de faturamento

Com essa solução, a ABC Pedágios fornece aos clientes uma notificação em tempo hábil quando os limites de gastos são alcançados.

Para obter informações em tempo real de seus dados de streaming, a ABC Pedágios escolheu usar o Kinesis Analytics para analisar seus dados de streaming. Com o Kinesis Analytics, a ABC Pedágios usou SQL, uma linguagem que já conhecia, para inspecionar seus dados transmitidos por meio do fluxo de entrega. Vamos analisar o Kinesis Analytics em mais detalhes.

Amazon Kinesis Analytics

Com o Kinesis Analytics, você pode processar e analisar dados de streaming usando o SQL. O serviço permite que você crie e execute rapidamente um código SQL sofisticado com base nas origens de streaming para executar análises temporais, alimentar painéis em tempo real e criar métricas em tempo real.

Para começar a usar o Kinesis Analytics, você cria um aplicativo do Kinesis Analytics que lê e processa continuamente dados de streaming. O serviço é compatível com a ingestão de dados de origens de streaming do Kinesis Streams e do Kinesis Firehose. Em seguida, você cria o código SQL usando o editor interativo e o testa com dados de streaming ao vivo. Você também pode configurar destinos nos quais o Kinesis Analytics para manter os resultados. O Kinesis Analytics oferece suporte ao Kinesis Firehose (Amazon S3, Amazon Redshift e Amazon Elasticsearch Service) e o Kinesis Streams como destinos.

Principais conceitos

Um *aplicativo* é o recurso principal no Kinesis Analytics que você pode criar em sua conta. Os aplicativos do Kinesis Analytics leem e processam continuamente dados de streaming em tempo real. Escreva o código do aplicativo usando o SQL para processar os dados de streaming de entrada e produzir a saída. Em seguida, o Kinesis Analytics grava a saída em um destino configurado. A Figura 7 ilustra uma arquitetura típica de aplicativo.

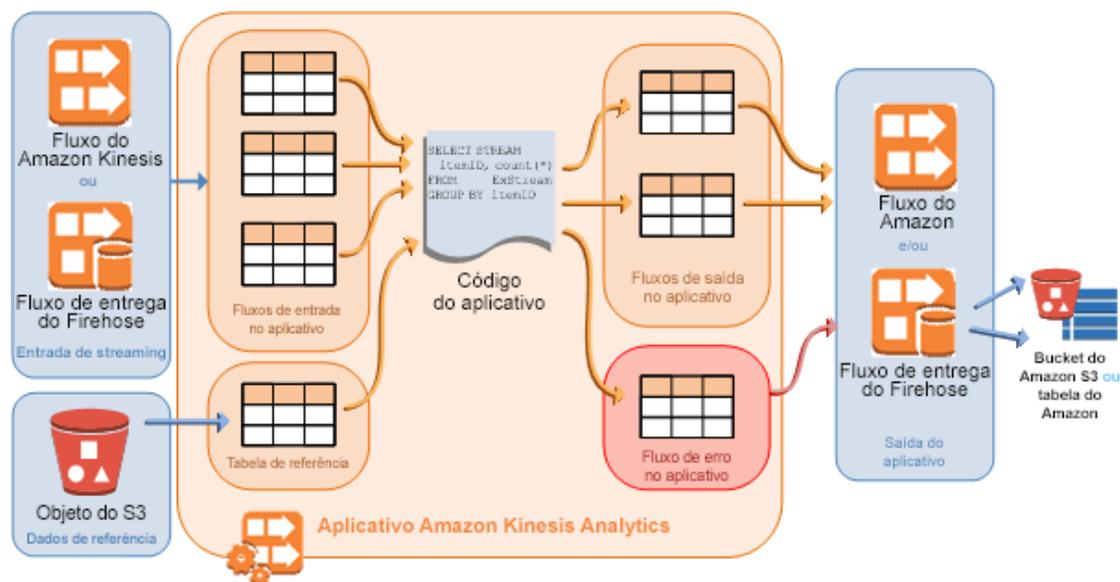


Figura 7: Arquitetura para um aplicativo do Kinesis Analytics

Cada aplicativo tem um nome, uma descrição, o ID da versão e o status. Ao criar um aplicativo, basta configurar a entrada, criar o código do aplicativo e configurar a saída.

Entrada

A entrada do aplicativo é a origem de streaming do aplicativo. Você pode selecionar um fluxo do Kinesis ou um fluxo de entrega como origem de streaming. Se desejar, pode configurar uma fonte de dados de referência para enriquecer o fluxo de dados de entrada no aplicativo. Isso resultará em uma tabela de referência no aplicativo. Você deve armazenar os dados de referência como um objeto em um bucket do S3. Quando o aplicativo é iniciado, o Kinesis Analytics lê o objeto do S3 e cria uma tabela de aplicativo.

A ABC Pedágios usava seu fluxo de entrega como entrada para o aplicativo do Kinesis Analytics.

Código do aplicativo

O código de aplicativo consiste em uma série de instruções SQL que processam entrada e produzem saída. É possível escrever instruções SQL com base em fluxos no aplicativo e tabelas de referência e escrever consultas JOIN para combinar dados dessas duas origens.

Em sua forma mais simples, o código do aplicativo pode ser uma instrução SQL única que faz a seleção de uma entrada de streaming e insere os resultados em uma saída de streaming. Ele também pode ser uma série de instruções SQL em que a saída de uma instrução contribui para a entrada da próxima instrução SQL. Além disso, você pode escrever o código do aplicativo para dividir um fluxo de entrada em vários fluxos e, então, aplicar consultas adicionais para processar esses fluxos.

Saída

No código do aplicativo, os resultados da consulta são passados para os fluxos no aplicativo. No código do aplicativo, é possível criar um ou mais fluxos no aplicativo para manter os resultados intermediários. Depois, se desejar, você pode configurar a saída do aplicativo para manter os dados nos fluxos no aplicativo, que retêm a saída do aplicativo (conhecida também como fluxos de saída no aplicativo), para os destinos externos. Os destinos externos podem ser um fluxo de entrega ou um fluxo do Kinesis.

A ABC Pedágios usava o Kinesis Streams como destino para seus valores agregados.

Resumo

O Kinesis Analytics permite que você use o SQL para obter informações sobre seus dados conforme ele faz streaming por meio do sistema. A ABC Pedágios desenvolveu seu SQL para executar agregações ao longo de 10 minutos para o total de pedágios incorridos por seus clientes. Os valores de saída dessas agregações de 10 minutos poderiam ser comparados com os limites de seus clientes.

Como mencionado anteriormente, o Kinesis Analytics gera seus resultados para o Kinesis Streams ou o Kinesis Firehose. Neste exemplo, a ABC Pedágios optou por enviar a saída do Kinesis Analytics para um fluxo do Kinesis por causa da integração do Kinesis Streams com o AWS Lambda. Vamos saber mais sobre o Kinesis Streams.

Amazon Kinesis Streams

O Amazon Kinesis Streams permite que você crie aplicativos personalizados em tempo real usando estruturas populares de processamento de streaming e carregar dados de streaming em qualquer armazenamento de dados. Você pode configurar centenas de milhares de produtores de dados para colocar dados continuamente em um fluxo do Kinesis, por exemplo, dados de clickstreams de sites, logs de aplicativo, sensores de IoT e feeds de mídia social. Em menos de um segundo, os dados estarão disponíveis para o seu aplicativo ler e processar do stream.

Ao implementar uma solução com o Kinesis Streams, você criará aplicativos personalizados de processamento de dados conhecidos como *aplicativos do Kinesis Streams*. Um aplicativo típico do Kinesis Streams lê dados de um fluxo do Kinesis como registros de dados.

Embora você possa usar o Kinesis Streams para resolver diversos problemas de dados de streaming, um uso comum é a agregação em tempo real ou a análise de dados seguida do carregamento de dados agregados para um data warehouse ou cluster de redução de mapa.

Os dados são colocados no Kinesis Streams, o que garante a durabilidade e a elasticidade. O atraso entre o momento em que um registro é colocado no stream e o momento em que pode ser recuperado (atraso entre put e get) normalmente é de menos de 1 segundo. Em outras palavras, um aplicativo do Kinesis Streams pode começar consumindo os dados do stream quase que imediatamente após a adição deles. Como o Kinesis Streams é um serviço gerenciado, ele assume o peso operacional de criação e da execução de um pipeline de entrada de dados.

Envio de dados ao Amazon Kinesis Streams

Há vários mecanismos para enviar dados ao seu stream. A AWS oferece SDKs para várias linguagens de programação populares, sendo que cada um deles fornece APIs para o Kinesis Streams. A AWS também criou vários utilitários para ajudar a enviar dados para o stream. Vamos analisar cada uma das abordagens que você pode usar e por que escolher uma delas.

Amazon Kinesis Agent

O Amazon Kinesis Agent foi discutido anteriormente como uma ferramenta que pode ser usada para enviar dados para o Kinesis Firehose. A mesma

ferramenta pode ser usada para enviar dados para o Kinesis Streams. Para obter detalhes sobre como instalar e configurar o agente do Kinesis, consulte [Writing to Amazon Kinesis Firehose Using Amazon Kinesis Agent](#).¹⁰

Amazon Kinesis Producer Library (KPL)

A KPL simplifica o desenvolvimento de aplicativos de produtor, permitindo que os desenvolvedores atinjam altos níveis de throughput para um ou mais streams do Kinesis. A KPL é uma biblioteca fácil de usar e altamente configurável que você instala em seus hosts que geram os dados a serem transmitidos para o Kinesis Streams. Ela atua como um intermediário entre o código do seu aplicativo de produtor e as ações da API do Kinesis Streams. A KPL executa as seguintes tarefas principais:

- Grava para um ou mais streams do Kinesis com um mecanismo de nova tentativa automático e configurável
- Coleta registros e usa `PutRecords` para gravar vários registros em vários estilos por solicitação
- Agrega registros de usuário para aumentar o tamanho da carga útil e melhorar o throughput
- Integra-se perfeitamente à Amazon Kinesis Client Library (KCL) para desagregar registros em lote no consumidor
- Envia métricas do Amazon CloudWatch em seu nome para proporcionar visibilidade sobre a performance do produtor

A KPL pode ser usada em casos de uso síncrono ou assíncrono. Sugerimos usar a performance mais alta da interface assíncrona, a menos que haja um motivo específico para usar o comportamento síncrono. Para obter mais informações sobre esses dois casos de uso e o código de exemplo, consulte [Writing to your Streams Stream Using the KPL](#).¹¹

A KPL pode ajudar a criar produtores de alta performance. Considere uma situação em que suas instâncias do Amazon Elastic Compute Cloud (EC2) atuam como proxy para coletar eventos de 100 bytes de centenas ou milhares de dispositivos de baixa potência e gravar registros em um fluxo do Kinesis. Cada instância do EC2 precisa gravar milhares de eventos por segundo no seu stream do Kinesis. Para alcançar o throughput necessário, os produtores precisam implementar uma lógica complexa, como agrupamento em lotes ou multithreading, lógica de nova tentativa e desagregação de registros do consumidor. A KPL executa todas essas tarefas para você.

Como a KPL armazena em buffer seus registros antes que eles sejam enviados para um fluxo do Kinesis, a KPL pode incorrer em mais atrasos de processamento, de acordo com o período de tempo que você tiver configurado para a KPL armazenar os registros em buffer antes de enviá-los ao Kinesis. Tempos de buffer maiores resultam em mais eficiência de empacotamento e melhor performance. Os aplicativos que não toleram esse atraso extra podem precisar usar o AWS SDK diretamente.

Se o seu aplicativo não registra os registros em um arquivo local e cria um grande número de pequenos registros por segundo, considere o uso da KPL.

Para obter detalhes sobre como usar a KPL para produzir dados, consulte [Developing Amazon Kinesis Streams Producers Using the Amazon Kinesis Producer Library](#).¹²

API do Amazon Kinesis

Depois que um stream é criado, você pode adicionar os registros de dados a ele. Um registro é uma estrutura de dados que contém os dados a serem processados na forma de um blob de dados. Depois que você armazena os dados no registro, o Kinesis Streams não inspeciona, interpreta ou altera dados de forma alguma.

Há duas operações diferentes na API do Kinesis Streams que adicionam dados a um stream: `PutRecords` e `PutRecord`. A operação `PutRecords` envia vários registros ao stream por solicitação HTTP, e a operação `PutRecord` singular envia registros ao stream um por vez (uma solicitação HTTP diferente é necessária para cada registro). Talvez convenha usar `PutRecords` para a maioria dos aplicativos, pois ela atingirá uma taxa de transferência mais alta por produtor de dados.

Como as APIs são expostas em todos os SDKs da AWS, usar a API para gravar registros fornece a solução mais flexível para enviar dados para um fluxo do Kinesis. Se você não conseguir usar o Kinesis Agent ou KPL (por exemplo, você deseja gravar mensagens diretamente de um aplicativo móvel ou deseja minimizar a latência completa da mensagem o máximo possível) use as APIs para gravar registros em seu stream do Kinesis.

Para obter detalhes sobre essas APIs, consulte [Using the API](#) na documentação do Kinesis Streams.¹³ Os detalhes de cada operação de API podem ser encontrados na [Amazon Kinesis Streams API Reference](#).¹⁴

Processamento de dados no Amazon Kinesis Streams

Um consumidor é um aplicativo que lê e processa dados do Kinesis Streams. Você pode criar consumidores para o Kinesis Streams de várias maneiras. Nesta seção, discutiremos quatro das abordagens mais comuns: usando o Kinesis Analytics, a KCL, o Amazon Lambda e a API do Kinesis Streams diretamente.

Uso do Amazon Kinesis Analytics

Anteriormente, discutimos como o Kinesis Analytics pode ser usado para analisar dados de streaming usando o SQL padrão. O Kinesis Analytics pode ler os dados de seu fluxo do Kinesis e processá-los usando o SQL que você fornece. Para saber mais sobre o processamento dos dados de streaming usando o Kinesis Analytics, consulte [Configuring Application Input](#) no Guia de Desenvolvedor do Kinesis Analytics.

Uso da Amazon Kinesis Client Library (KCL)

Você pode desenvolver um aplicativo de consumidor para o Kinesis Streams usando a KCL. Embora você possa usar a API do Kinesis Streams para obter dados de um stream do Amazon Kinesis, recomendamos usar os padrões de projeto e código para aplicativos de consumidor fornecidos pela KCL.

A KCL ajuda a consumir e processar dados de um stream do Kinesis. Esse tipo de aplicativo também é conhecido como consumidor. A KCL se encarrega de muitas tarefas complexas associadas à computação distribuída, como balanceamento de carga entre várias instâncias, resposta a falhas de instância, definição de pontos de verificação de registros processados e reação a reestilhaçamento. Ela permite que você se concentre em escrever a lógica de processamento de registro.

A KCL é uma biblioteca Java, que oferece suporte a linguagens diferentes de Java usando uma interface de várias languages. No momento de execução, um aplicativo da KCL instancia um operador com informações de configuração e, em seguida, usa um processador de registro para processar os dados recebidos de um stream do Kinesis. Você pode executar um aplicativo da KCL em qualquer número de instâncias. Várias instâncias do mesmo aplicativo se coordenam em falhas e balanceamento de carga dinamicamente. Você também pode ter vários aplicativos da KCL trabalhando no mesmo stream, sujeitos aos limites de throughput. A KCL atua como um intermediário entre a lógica de processamento de registros e o Kinesis Streams.

Para obter informações detalhadas sobre como criar seu próprio aplicativo KCL, consulte [Developing Amazon Kinesis Streams Consumers Using the Amazon Kinesis Client Library](#).¹⁵

Uso do AWS Lambda

O [AWS Lambda](#)¹⁶ é um serviço de computação que permite executar código sem o provisionamento ou o gerenciamento de servidores. O AWS Lambda executa seu código somente quando necessário e dimensiona automaticamente. Com ele, você pode executar códigos sem administração. O AWS Lambda executa seu código em uma infraestrutura de computação de alta disponibilidade e executa toda a administração de recursos de computação, incluindo manutenção do servidor e do sistema operacional, provisionamento da capacidade e escalabilidade automática e monitoramento do código e registro em log. Tudo o que você precisa fazer é fornecer seu código em uma das linguagens às quais o AWS Lambda oferece suporte.

Você pode assinar as funções do Lambda para ler automaticamente lotes de registros fora do fluxo do Kinesis e processá-los se os registros forem detectados no fluxo. Em seguida, o AWS Lambda pesquisa o fluxo periodicamente (uma vez por segundo) em busca de novos registros. Ao detectar novos registros, ele invoca sua função Lambda passando os novos registros como um parâmetro. Se novos registros não forem detectados, sua função Lambda não é invocada.

Para obter informações detalhadas sobre como usar o AWS Lambda para consumir dados do Kinesis Streams, consulte [Using AWS Lambda with Amazon Kinesis](#).¹⁷

Uso da API

Para a maioria dos casos de uso, você deve usar a KCL ou o AWS Lambda para recuperar e processar dados de um stream. No entanto, se você preferir escrever seu próprio aplicativo de consumidor do zero, há vários métodos que ativam esse recurso. A API do Kinesis Streams fornece os métodos `GetShardIterator` e `GetRecords` para recuperar dados de um stream. Esse é um modelo de extração, em que seu código extrai dados diretamente dos estilhaços do stream. Para obter mais informações sobre como escrever seu próprio aplicativo de consumidor usando a API, consulte [Developing Amazon Kinesis Streams Consumers Using the Amazon Kinesis Streams API](#). Detalhes sobre a API podem ser encontrados na [Amazon Kinesis Streams API Reference](#).¹⁸

Como escolher o melhor modelo de consumidor para seu aplicativo

Como saber qual é o melhor modelo de consumidor para seu caso de uso? Cada abordagem tem suas desvantagens e será necessário decidir que é importante. Veja algumas orientações gerais para ajudá-lo a escolher o modelo de consumidor correto.

Na maioria dos casos, considere começar com o AWS Lambda. Sua facilidade de uso e o modelo de implantação simples permitirá que você crie rapidamente um consumidor de dados. A desvantagem de usar o AWS Lambda é que cada invocação da função Lambda deve ser considerada stateless, ou seja, você não pode usar os resultados de invocações anteriores de sua função (por exemplo, lotes de registros anteriores do seu stream). Além disso, considere que o tempo máximo de execução para uma única função Lambda é 5 minutos. Se um único lote de registros leva mais tempo do que 5 minutos para processar, o AWS Lambda pode não ser a melhor consumidor para seu caso de uso.

Se você decidir que não é possível usar o AWS Lambda, considere a criação de seu próprio aplicativo de processamento com a KCL. Como você implanta aplicativos da KCL para instâncias do EC2 pela sua conta da AWS, há bastante flexibilidade e controle na persistência local de dados e requisitos de estado para seus dados.

A terceira opção é criar seu próprio aplicativo usando as APIs diretamente. Assim, você tem mais controle e flexibilidade, mas também precisa criar sua própria lógica para lidar com os recursos comuns do aplicativo de consumidor, como pontos de verificação, escalabilidade e failover.

Resumo

O Kinesis Streams facilita o recebimento de dados de streaming. Você pode dimensionar um fluxo do Kinesis para lidar com apenas alguns registros por segundo ou milhões de registros por segundo. Para a ABC Pedágios, o streaming da taxa de dados para seu stream de streaming não foi grande. No entanto, um dos benefícios foi a integração direta com o AWS Lambda, que permitiu fazer chamadas de API facilmente para o Amazon SNS para notificações do usuário.

Requisito 3: outros alertas de limite

O requisito final é semelhante ao anterior, mas apresenta um problema adicional. Para recapitular esse último requisito, os operadores da ABC Pedágios queriam ser notificados imediatamente quando o tráfego de veículos em uma cabine de

pedágio ficasse abaixo de um limite predefinido para cada período de 30 minutos em um dia. Por exemplo, ela já sabem devido ao histórico de dados que aproximadamente 360 veículos passam por aquela cabine às quartas-feiras entre 14h e 14h30. Nessa janela de 30 minutos, se menos de 100 veículos passassem por uma cabine de pedágio, eles seriam notificados.

A ABC Pedágios quer para comparar os totais atuais de veículos para cada cabine com uma taxa média conhecida para essa cabine. Para fazer isso, ela criou um arquivo que contém os valores de limite de tráfego para cada janela de 30 minutos, para cada cabine. Conforme descrito anteriormente, o Kinesis Analytics é compatível com o uso de dados de referência. Ele cria um stream no aplicativo (como uma tabela) com base nos dados em um arquivo armazenado em um bucket do S3. Com isso, os desenvolvedores da ABC Pedágios foram capazes de gravar o SQL em seu aplicativo do Kinesis Analytics para contar o número de veículos que passaram por cada cabine em uma janela de 30 minutos e comparar esses valores com os limites no arquivo. Se o limite for excedido, o Kinesis Analytics gera um registro para um fluxo do Kinesis. Quando os registros chegam no stream, uma função Lambda é executada, que usa o Amazon SNS para enviar uma notificação para os operadores da ABC Pedágios. A Figura 8 ilustra a arquitetura para esse cenário.



Figura 8: Arquitetura para alertas e notificações usando períodos de 30 minutos

Arquitetura completa

Com uma solução para cada requisito, agora temos a nossa solução de streaming geral, conforme mostrado na Figura 9.

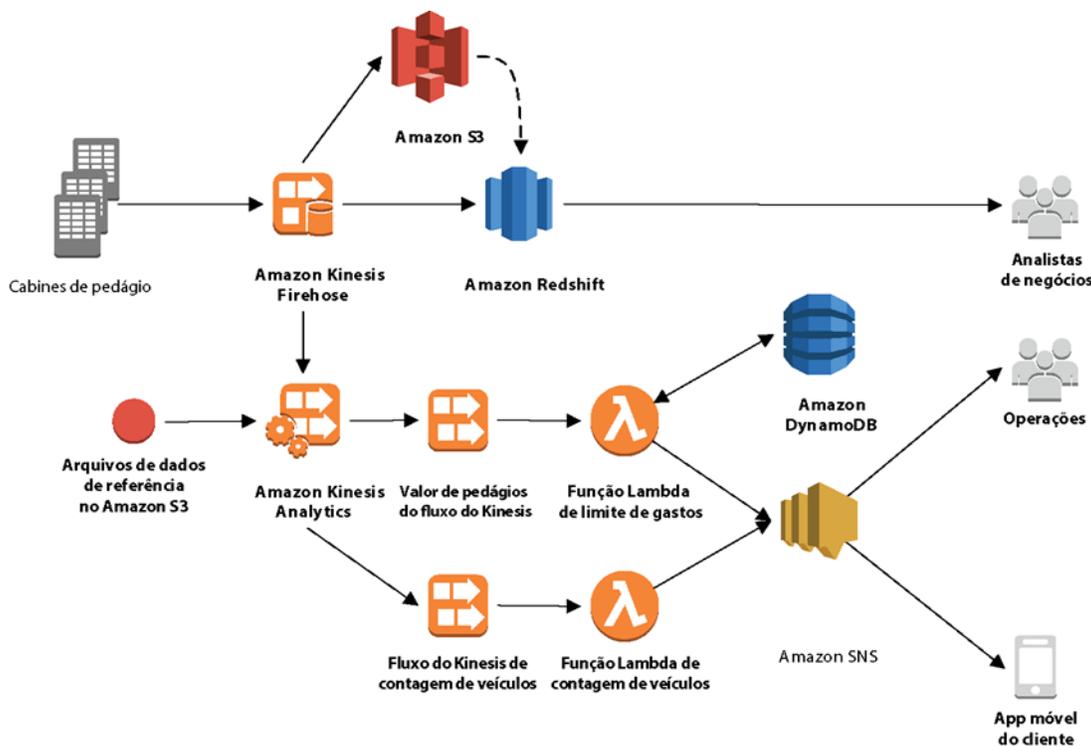


Figura 9: A arquitetura da solução de streaming geral

Este projeto oferece à ABC Pedágios uma arquitetura flexível e reativa. Com o streaming de transações de seus clientes em tempo real, ela pode alcançar seus requisitos com muito pouco esforço de desenvolvimento e a infraestrutura mínima para gerenciar.

Conclusão

Neste documento, analisamos como uma empresa fictícia, a ABC Pedágios, usou os serviços do Amazon Kinesis para mover um fluxo de trabalho de lote tradicional para um fluxo de trabalho de streaming. Essa migração forneceu a capacidade de adicionar novos recursos e funcionalidades que não eram possíveis com sua solução de lote herdada.

Ao analisar os dados à medida que são criados, você poderá obter informações sobre o que sua empresa está fazendo agora. Os serviços do Amazon Kinesis

permitem que você se concentre em seu aplicativo para tomar decisões de negócios urgentes, em vez de implantar e gerenciar a infraestrutura.

Contribuidores

As pessoas e organizações a seguir contribuíram com este documento:

- Allan MacInnis, arquiteto de soluções, AWS
- Chander Matrubhutam, gerente de marketing do produto, AWS

Observações

- ¹ <https://aws.amazon.com/kinesis/streams/>
- ² <https://aws.amazon.com/kinesis/firehose/>
- ³ <https://aws.amazon.com/kinesis/analytics/>
- ⁴ <http://docs.aws.amazon.com/firehose/latest/dev/writing-with-sdk.html>
- ⁵ <https://aws.amazon.com/lambda/>
- ⁶ <https://aws.amazon.com/s3/>
- ⁷ <https://aws.amazon.com/redshift/>
- ⁸ <https://aws.amazon.com/elasticsearch-service/>
- ⁹ <https://aws.amazon.com/dynamodb/>
- ¹⁰ <http://docs.aws.amazon.com/firehose/latest/dev/writing-with-agents.html>
- ¹¹ <http://docs.aws.amazon.com/streams/latest/dev/kinesis-kpl-writing.html>
- ¹² <http://docs.aws.amazon.com/streams/latest/dev/developing-producers-with-kpl.html>
- ¹³ <http://docs.aws.amazon.com/streams/latest/dev/developing-producers-with-sdk.html>
- ¹⁴ <http://docs.aws.amazon.com/kinesis/latest/APIReference/Welcome.html>
- ¹⁵ <http://docs.aws.amazon.com/streams/latest/dev/developing-consumers-with-kcl.html>
- ¹⁶ <https://aws.amazon.com/lambda/>
- ¹⁷ <http://docs.aws.amazon.com/lambda/latest/dg/with-kinesis.html>
- ¹⁸ <http://docs.aws.amazon.com/kinesis/latest/APIReference/Welcome.html>