AWS ホワイトペーパー

Amazon Kinesis を使用した AWS でのストリーミングデータソリューション

Amazon Kinesis を使用した AWS でのストリーミングデータソリューション: AWS ホワイトペーパー

Table of Contents

AWS でのストリーミングデータソリューション

公開日: 2021 年 9 月 1 日 (改訂履歴)

要約

データエンジニア、データアナリスト、ビッグデータデベロッパーは、自社の顧客、アプリケーション、製品が現在行っていることを把握し、迅速に対応できるように、分析をバッチからリアルタイムに進化させようとしています。このホワイトペーパーでは、バッチからリアルタイムへの分析の進化について説明します。Amazon Kinesis Data Streams、Amazon Kinesis Data Firehose、Amazon EMR、Amazon Kinesis Data Analytics、Amazon Managed Streaming for Apache Kafka (Amazon MSK)、およびその他のサービスを使用してリアルタイムアプリケーションを実装する方法について説明し、これらのサービスを使用する一般的な設計パターンを提供します。

はじめに

データストリームを継続的に生成するデータソースの爆発的な増加により、今日の企業は膨大な規模と速度でデータを受信しています。アプリケーションサーバーからのログデータ、ウェブサイトやモバイルアプリケーションからのクリックストリームデータ、IoT (モノのインターネット) デバイスからのテレメトリデータなど、すべてのデータには、顧客、アプリケーション、製品が現在何をしているのかを知るのに役立つ情報が含まれています。

このデータをリアルタイムで処理および分析できることが、アプリケーションを継続的にモニターして高いサービスアップタイムを確保することや、プロモーション特典や製品レコメンデーションのパーソナライズを行うためには不可欠です。リアルタイムおよびほぼリアルタイムの処理により、ウェブサイト分析や機械学習などの他の一般的なユースケースでも、これらのアプリケーションでデータを数時間や数日ではなく数秒または数分で利用できるようになることで、より正確で実用的なものになります。

リアルタイムおよびほぼリアルタイムのアプリケーションシナリオ

ストリーミングデータサービスは、アプリケーションのモニタリング、不正検出、ライブリーダーボードのようなリアルタイムおよびほぼリアルタイムのアプリケーションに使用できます。リアルタイムのユースケースでは、取り込みから処理、ターゲットのデータストアや他のシステムへの結果の送信に至るまで、エンドツーエンドでミリ秒単位のレイテンシーが要求されます。例えば、Netflix は Amazon Kinesis Data Streams を使用してすべてのアプリケーション間のコミュニケーションをモニターすることで、問題をすぐに検出して修正し、アップタイムと利用可能性の高いサービスをお客様に提供しています。最も一般的に適用できるユースケースはアプリケーションパフォーマンスのモニタリングですが、アドテック、ゲーム、IoT の分野では、このカテゴリに分類されるリアルタイムアプリケーションの数が増えています。

一般的なほぼリアルタイムのユースケースには、データサイエンスおよび機械学習のためのデータストアでの分析が含まれます。ストリーミングデータソリューションを使用すると、データレイクにリアルタイムデータを継続的にロードできます。新しいデータが利用可能になり次第、機械学習モデルを更新して、出力の精度と信頼性を高めることができます。例えば、Zillow は Kinesis Data Streams を使用して、公開レコードデータやマルチリスティングサービス (MLS、multiple listing service) リスティングを収集し、住宅購入者と販売者に最新の住宅価格推定をほぼリアルタイムで提供します。ZipRecruiter では、ZipRecruiter 雇用市場から 1 日あたり 60 億のイベントを収集、保存し、継続的に処理するための重要なインフラストラクチャコンポーネントであるイベントのログ記録パイプラインに、Amazon MSK を使用しています。

バッチ処理とストリーム処理の違い

リアルタイムストリーミングデータの収集、準備、処理には、従来バッチ分析に使用していたツールとは異なるツールセットが必要です。従来の分析では、データを収集して定期的にデータベースにロードし、数時間、数日、または数週間後に分析していました。リアルタイムデータの分析には別のアプローチが必要です。ストリーム処理アプリケーションは、データが保存される前であっても、リアルタイムで継続的にデータを処理します。ストリーミングデータは猛烈なペースで流入し、データ量は増減する可能性が常にあります。ストリーミングデータ処理プラットフォームは、受信データの速度と変動性に対処でき、データを受信してすぐに処理できる必要があります。多くの場合、1 時間あたり数百万から数億のイベントが発生します。

ストリーミング処理の課題

届いたデータをリアルタイムで処理することで、従来のデータ分析テクノロジーよりもはるかに迅速に意思決定を下すことができます。ただし、独自のカスタムストリーミングデータパイプラインの構築と運用は複雑で、リソースを大量に消費します。

- 数千ものデータソースから同時に受信するデータをコスト効率よく収集、準備、送信できるシステムを構築する必要があります。
- データを効率的にバッチ処理して送信し、スループットを最大化してレイテンシーを低く抑えるために、ストレージとコンピューティングリソースを細かく調整しなければなりません。
- システムに投入されるさまざまな速度のデータを処理できるようにするには、サーバーのフリートをデプロイして管理し、システムをスケールする必要があります。

バージョンアップは複雑でコストのかかるプロセスです。このプラットフォームを構築した後は、システムをモニターし、重複データを作成せずにストリーミングの適切なポイントからデータ処理をキャッチアップして、サーバーまたはネットワークの障害から復旧する必要があります。また、インフラストラクチャ管理のための専任チームも必要です。これらすべてに貴重な時間と費用がかかり、結局のところ、ほとんどの企業はそこにたどり着くことはなく、現状に甘んじ、数時間または数日前の情報でビジネスを運営しなければなりません。

ストリーミングデータソリューション: 例

シナリオ 1: 場所に基づくインターネットの提供

InternetProvider は、世界中のユーザーにさまざまな帯域幅オプションを備えたインターネットサービスを提供しています。ユーザーがインターネットにサインアップすると、InternetProvider は、ユーザーの地理的位置に基づいて異なる帯域幅オプションをユーザーに提供します。これらの要件を考慮して、InternetProvider は Amazon Kinesis Data Streams を実装し、ユーザーの詳細と場所を処理しました。ユーザーの詳細と場所は、さまざまな帯域幅オプションでエンリッチされてから、アプリケーションにパブリッシュして戻されます。AWS Lambda は、このリアルタイムエンリッチメントを可能にします。

AWS Lambda を使用してデータのストリーミングを処理する

Amazon Kinesis Data Streams

Amazon Kinesis Data Streams を使用すると、一般的なストリーミング処理フレームワークを使用してカスタムのリアルタイムアプリケーションを構築し、ストリーミングデータをさまざまなデータストアにロードできます。Kinesis のストリーミングは、ウェブサイトのクリックストリーム、IoT センサー、ソーシャルメディアフィード、アプリケーションログなどのソースから配信された数十万ものデータプロデューサーからのイベントを、継続的に受信するように設定できます。ミリ秒以内に、アプリケーションでデータを読み取りおよび処理できるようになります。

Kinesis Data Streams でソリューションを実装する場合、Kinesis Data Streams アプリケーションと呼ばれるカスタムデータ処理アプリケーションを作成します。一般的な Kinesis Data Streams アプリケーションは、Kinesis ストリーミングからデータをデータレコードとして読み取ります。

Kinesis Data Streams に格納されるデータは、高い可用性と伸縮自在性が確保され、ミリ秒で利用可能になります。クリックストリーム、アプリケーションログ、ソーシャルメディアなど、さまざまなタイプのデータを、何十万ものソースから Kinesis のストリーミングに継続して追加できます。ほ

んの数秒後には、Kinesis アプリケーションで、ストリーミングからデータを読み取って処理できます。

Amazon Kinesis Data Streams はフルマネージド型のストリーミングデータサービスです。データスループットのレベルでデータをストリーミングするために必要なインフラストラクチャ、ストレージ、ネットワーク、設定を管理します。

Amazon Kinesis Data Streams へのデータの送信

Kinesis Data Streams にデータを送信する方法はいくつかあり、ソリューションを柔軟に設計できます。

- 一般的な複数の言語でサポートされている AWS SDK の 1 つを使用してコードを記述できます。
- Kinesis Data Streams にデータを送信するためのツールである Amazon Kinesis Agent を使用できます。

Amazon Kinesis Producer Library (KPL) は、プロデューサーアプリケーションの開発を簡素化し、デベロッパーが 1 つ以上の Kinesis のデータストリームへの高い書き込みスループットを実現できるようにします。

KPL は使いやすく、高度な構成が可能なライブラリで、お客様のホストにインストールできます。これは、プロデューサーアプリケーションのコードと Kinesis Streams API アクション間の仲介として機能します。KPL およびコード例を使用してイベントを同期および非同期に生成する機能の詳細については、「KPL を使用した Kinesis Data Streams への書き込み」を参照してください。

Kinesis Data Streams API には、ストリームにデータを追加する 2 つの異なるオペレーション (`PutRecords` と `PutRecord`) があります。`PutRecords` オペレーションは HTTP リクエストごとに複数のレコードをストリーミングに送信し、`PutRecord` は HTTP リクエストごとに 1 つのレコードを送信します。ほとんどのアプリケーションで高いスループットを達成するには、`PutRecords` を使用します。

これらの API の詳細については、「ストリーミングへのデータの追加」を参照してください。各 API オペレーションの詳細については、「Amazon Kinesis Data Streams API Reference」を参照してください。

Amazon Kinesis Data Streams でのデータ処理

Kinesis のストリーミングからデータを読み取って処理するには、コンシューマーアプリケーションを作成する必要があります。Kinesis Data Streams のコンシューマーを作成するには、さまざまな方

法があります。KCL を使用したストリーミングデータの分析に Amazon Kinesis Data Analytics を使用する、AWS Lambda を使用する、AWS Glue で ETL ジョブをストリーミングする、Kinesis Data Streams API を直接使用する、などのアプローチがあります。

Kinesis Data Streams のコンシューマーアプリケーションは、Kinesis Data Streams からのデータの消費および処理を助ける KCL を使用して開発できます。KCL は、分散コンピューティングに関連する多くの複雑なタスクを処理します。たとえば、複数のインスタンス間でのロードバランシング、インスタンスの障害に対する応答、処理済みのレコードのチェックポイント作成、リシャーディングへの対応が挙げられます。KCL によって、レコード処理のロジックの記述に集中できます。独自の KCL アプリケーションを構築する方法の詳細については、「Kinesis クライアントライブラリの使用」を参照してください。

Lambda 関数をサブスクライブして、Kinesis のストリーミングからレコードのバッチを自動的に読み取り、ストリーミングでレコードが検出された場合は処理できます。AWS Lambda はストリーミングを定期的 (1 秒に 1 回) にポーリングして新しいレコードを検出し、新しいレコードを検出すると Lambda 関数を呼び出して新しいレコードをパラメータとして渡します。Lambda 関数は、新しいレコードが検出された場合にのみ実行されます。Lambda 関数を共有スループットコンシューマー (標準イテレーター) にマッピングできます。

ストリーミングからデータを受信している他のコンシューマーと競合しない専用スループットが必要な場合は、拡張ファンアウトと呼ばれる機能を使用するコンシューマーを構築できます。この機能により、コンシューマーは、シャードあたり 1 秒間に最大 2 MB のデータのスループットで、ストリーミングからレコードを受け取ることができます。

ほとんどの場合、Kinesis Data Analytics、KCL、AWS Glue、または AWS Lambda を使用して、ストリーミングからのデータを処理する必要があります。ただし、Kinesis Data Streams API を使用してコンシューマーアプリケーションを最初から作成することもできます。Kinesis Data Streams API には、ストリーミングからデータを取得するための `GetShardIterator` および `GetRecords` メソッドが用意されています。

このプルモデルでは、コードはストリーミングのシャードから直接データを抽出します。API を使用して独自のコンシューマーアプリケーションを作成する方法の詳細については、「AWS SDK for Java を使用したスループット共有カスタムコンシューマーの開発」を参照してください。API に関する詳細については、「Amazon Kinesis Data Streams API Reference」を参照してください。

AWS Lambda を使用してデータのストリーミングを処理する

AWS Lambda によって、サーバーのプロビジョニングや管理をすることなく、コードを実行できるようになります。Lambda では、実質的にあらゆるタイプのアプリケーションやバックエンドサービ

スに対して、管理タスクを実行せずにコードを実行できます。コードをアップロードするだけで、コードの実行とスケールに必要な処理はすべて Lambda により自動的に実行され、高い可用性が維持されます。コードは、AWS の他のサービスから自動的にトリガーしたり、ウェブやモバイルアプリケーションから直接呼び出したりするように設定できます。

AWS Lambda は Amazon Kinesis Data Streams とネイティブに統合されます。このネイティブ統合を使用すると、ポーリング、チェックポイント、エラー処理の複雑さが抽象化されます。これにより、Lambda 関数コードはビジネスロジックの処理に集中できます。

Lambda 関数を共有スループット (標準イテレーター) にマップすることも、拡張ファンアウトを使用する専用スループットコンシューマーにマップすることもできます。標準イテレーターの場合、Lambda は HTTP プロトコルを使用して、Kinesis のストリーミングの各シャードにレコードがあるかどうかをポーリングします。レイテンシーを最小限に抑え、読み取りスループットを最大化するために、拡張ファンアウトを使用するデータストリームコンシューマーを作成できます。このアーキテクチャのストリームコンシューマーは、同じストリーミングから読み取る他のアプリケーションと競合することなく、各シャードへの専用接続を取得します。Amazon Kinesis Data Streams は HTTP/2 経由でレコードを Lambda にプッシュします。

デフォルトでは、AWS Lambda はストリーミングのレコードが利用可能になるとすぐに、関数を呼び出します。バッチシナリオでレコードをバッファするには、イベントソースで最大 5 分間のバッチウィンドウを実装できます。関数がエラーを返した場合、処理が成功するか、データの有効期限が切れるまで、Lambda はバッチを再試行します。

概要

InternetProvider 社は、Amazon Kinesis Data Streams を活用して、ユーザーの詳細と場所をストリーミングしました。レコードのストリーミングは AWS Lambda で消費され、関数のライブラリに保存された帯域幅オプションを使用して、データがエンリッチされました。エンリッチメント後、AWS Lambda は帯域幅オプションをアプリケーションにパブリッシュして戻しました。Amazon Kinesis Data Streams と AWS Lambda は、サーバーのプロビジョニングと管理を処理し、InternetProvider 社はビジネスアプリケーション開発により集中できるようになりました。

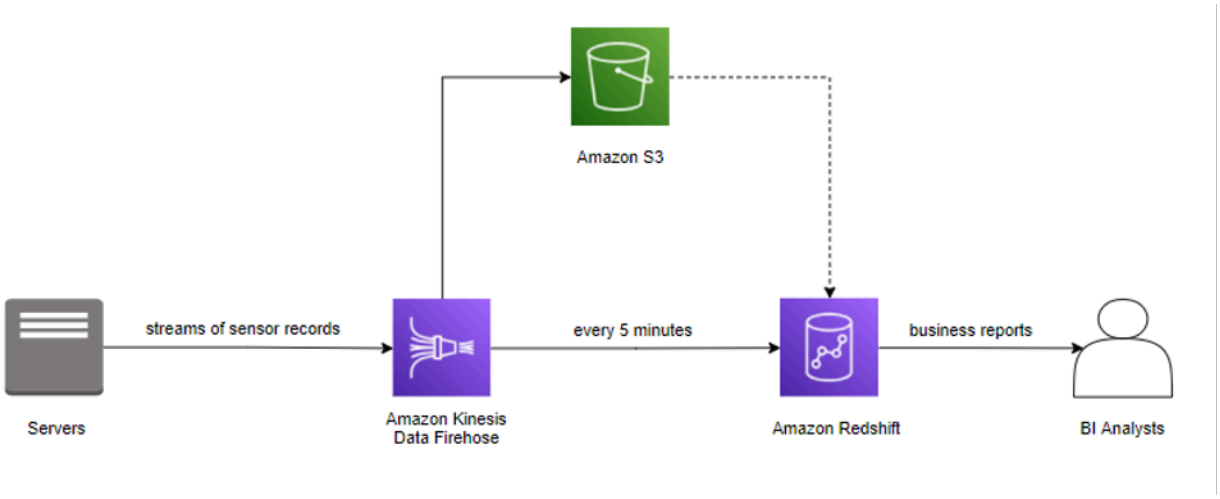
シナリオ 2: セキュリティチームのためのほぼリアルタイムのデータ

ABC2Badge 社は、AWS re:Invent などの企業イベントや大規模なイベント用に、センサーとバッジを提供しています。ユーザーはイベントにサインアップし、キャンパス全体でセンサーが取得するー

意のバッジを受け取ります。ユーザーがセンサーを通過すると、匿名化された情報がリレーショナルデータベースに記録されます。

来るイベントで、参加者数が多いため、ABC2Badge 社はイベントセキュリティチームから、15 分ごとにキャンパスの最も混雑したエリアのデータを収集するよう要請されました。これにより、セキュリティチームは、エリアの混雑に比例して対応しセキュリティ担当者を分散させるのに十分な時間を確保できます。ABC2Badge 社は、セキュリティチームからの新しい要件と、ほぼリアルタイムでデータを処理するためのストリーミングソリューションを構築する経験が浅いことから、シンプルでありながらスケーラブルで信頼性の高いソリューションを求めています。

同社の現在のデータウェアハウスソリューションは Amazon Redshift です。Amazon Kinesis サービスの機能を確認したところ、Amazon Kinesis Data Firehose が、データレコードのストリーミングを受信し、バッファサイズや時間間隔に基づいてレコードをバッチ処理して、Amazon Redshift に挿入できることがわかりました。同社は Kinesis Data Firehose 配信ストリームを作成し、5 分ごとに Amazon Redshift テーブルにデータをコピーするように設定しました。この新しいソリューションの一環として、自社のサーバーで Amazon Kinesis Agent を使用しました。Kinesis Data Firehose は 5 分ごとに Amazon Redshift にデータをロードします。Amazon Redshift では、ビジネスインテリジェンス (BI) チームが分析を実行し、15 分ごとにデータをセキュリティチームに送信できます。



Amazon Kinesis Data Firehose を使用した新しいソリューション

Amazon Kinesis Data Firehose

Amazon Kinesis Data Firehose は、ストリーミングデータを AWS にロードする最も簡単な方法です。ストリーミングデータをキャプチャ、変換し、Amazon Kinesis Data Analytics、Amazon Simple Storage Service (Amazon S3)、Amazon Redshift、Amazon OpenSearch Service (OpenSearch Service)、Splunk にロードできます。さらに、Kinesis Data Firehose は、カスタム HTTP エンドポ

イント、またはサポートされているサードパーティーサービスプロバイダーが所有する HTTP エンドポイントにストリーミングデータをロードできます。

Kinesis Data Firehose は、現在使用している既存のビジネスインテリジェンスツールとダッシュボードを使用して、ほぼリアルタイムの分析を可能にします。データのスループットに合わせて自動的にスケールするフルマネージドサーバーレスサービスであるため、継続的な管理は不要です。Kinesis Data Firehose は、ロード前にデータのバッチ処理、圧縮処理、暗号化が行われるため、送信先でのストレージ量を最小化し、セキュリティを強化できます。また、AWS Lambda を使用してソースデータを変換し、変換されたデータを送信先に配信することもできます。Kinesis Data Firehose にデータを送信するようデータプロデューサーを設定して、指定した送信先にデータが自動配信されるようにできます。

Firehose 配信ストリームへのデータ送信

配信ストリームにデータを送信する方法は、いくつかあります。AWS は多くの一般的なプログラミング言語用の SDK を提供しており、それぞれが Amazon Kinesis Data Firehose 用の API を提供しています。AWS には、配信ストリームにデータを送信するためのユーティリティがあります。Kinesis Data Firehose は AWS の他のサービスと統合され、これらのサービスから配信ストリームにデータを直接送信します。

Amazon Kinesis Agent を使用する

Amazon Kinesis エージェントは、配信ストリームに送信される新しいデータがないか一連のログファイルを継続的にモニターするスタンドアロンのソフトウェアアプリケーションです。エージェントは、ファイルローテーション、チェックポイント、障害発生時の再試行を自動的に処理し、配信ストリームのモニタリングとトラブルシューティングのために Amazon CloudWatch メトリクスを出力します。データの前処理、複数のファイルディレクトリのモニタリング、複数の配信ストリームへの書き込みなど、追加の構成をエージェントに適用できます。

このエージェントは、ウェブサーバー、ログサーバー、データベースサーバーなどの Linux または Windows ベースのサーバーにインストールできます。エージェントをインストールしたら、モニターするログファイルと、送信する配信ストリームを指定するだけです。エージェントは、新しいデータを持続的かつ確実に配信ストリームに送信します。

AWS SDK と AWS のサービスをソースとして API を使用する

Kinesis Data Firehose API には、配信ストリームにデータを送信するための 2 つのオペレーションが用意されています。`PutRecord` は、1 回のコールで 1 つのデータレコードを送信します。`PutRecordBatch` は、1 回のコールで複数のデータレコードを送信でき、プロデューサあたり

のスループットを向上させることができます。どちらのメソッドでも、そのメソッドを使用する際に、配信ストリームの名前とデータレコード、またはデータレコードの配列を指定する必要があります。Kinesis Data Firehose API オペレーションの詳細とサンプルコードについては、「AWS SDK を使用した Firehose 配信ストリームへの書き込み」を参照してください。

Kinesis Data Firehose は、Kinesis Data Firehose、CloudWatch Logs、CloudWatch Events、Amazon Simple Notification Service (Amazon SNS)、Amazon API Gateway、AWS IoT とともに実行することもできます。データ、ログ、イベント、IoT データのストリーミングを Kinesis Data Firehose の送信先に直接、スケーラブルかつ確実に送信できます。

データを送信先に配信する前に処理する

場合によっては、ストリーミングデータを送信先に配信する前に、変換または強化が必要になることがあります。例えば、データプロデューサーが各データレコードの非構造化テキストを送信し、OpenSearch Service に配信する前に JSON に変換する必要がある場合があります。または、Simple Storage Service (Amazon S3) にデータを保存する前に、JSON データを Apache Parquet や Apache ORC などの列指向ファイル形式に変換することもできます。

Kinesis Data Firehose には、データの形式変換機能が組み込まれています。これにより、JSON データのストリーミングを Apache Parquet または Apache ORC ファイル形式に簡単に変換できます。

データ変換フロー

ストリーミングデータ変換を有効にするには、Kinesis Data Firehose はユーザーが作成した Lambda 関数を使用してデータを変換します。Kinesis Data Firehose は、指定されたバッファサイズにで受信データを関数用にバッファリングし、指定された Lambda 関数を非同期に呼び出します。変換されたデータは Lambda から Kinesis Data Firehose に送信され、Kinesis Data Firehose はデータを送信先に配信します。

データ形式の変換

また、Kinesis Data Firehose のデータ形式変換を有効にすることもできます。これにより、JSON データのストリーミングが Apache Parquet または Apache ORC に変換されます。この機能は JSON を Apache Parquet または Apache ORC にのみ変換できます。CSV 形式のデータがある場合は、そのデータを Lambda 関数を介して JSON に変換してから、データ形式の変換を適用できます。

データ配信

Kinesis Data Firehose は、ほぼリアルタイムの配信ストリームとして、受信データをバッファリングします。配信ストリームのバッファリングしきい値に達すると、設定した送信先にデータが配信さ

れます。Kinesis Data Firehose が各送信先にデータを配信する方法には、いくつかの違いがあります。このホワイトペーパーでは、以下のセクションで説明します。

Simple Storage Service (Amazon S3)

Amazon Simple Storage Service (Amazon S3) は、ウェブのどこからでもお好みの量のデータを保存および取得できる、シンプルなウェブサービスインターフェイスを備えたオブジェクトストレージです。これは 99.999999999% の耐久性を提供し、世界中の数兆個を超えるオブジェクトを保管するまでスケールできるように設計されています。

Simple Storage Service (Amazon S3) へのデータ配信

Simple Storage Service (Amazon S3) へのデータ配信では、Kinesis Data Firehose は配信ストリームのバッファリング設定に基づいて複数の受信レコードを連結し、S3 オブジェクトとして Amazon S3 に配信します。S3 へのデータ配信の頻度は、S3 のバッファサイズ (1 MB ~ 128 MB) またはバッファ間隔 (60 秒から 900 秒) のいずれか早い方によって決まります。

S3 バケットへのデータ配信は、さまざまな理由で失敗する場合があります。例えば、バケットがもう存在しない場合や、Kinesis Data Firehose が引き受ける AWS Identity and Access Management (IAM) ロールがバケットにアクセスできない場合などです。そのような状況では、Kinesis Data Firehose は配信が成功するまで最大 24 時間にわたり再試行し続けます。Kinesis Data Firehose の最大データ保存時間は 24 時間です。データ配信が 24 時間を超えて失敗した場合、データは失われます。

Amazon Redshift

Amazon Redshift は高速でフルマネージド型のデータウェアハウスです。標準 SQL および既存の BI ツールを使用して、すべてのデータをシンプルかつコスト効率よく分析できます。洗練されたクエリ最適化、ハイパフォーマンスなローカルディスク上に配置された列指向ストレージ、超並列クエリ実行を使用し、ペタバイト単位の構造化データに対して複雑な分析クエリを実行できます。

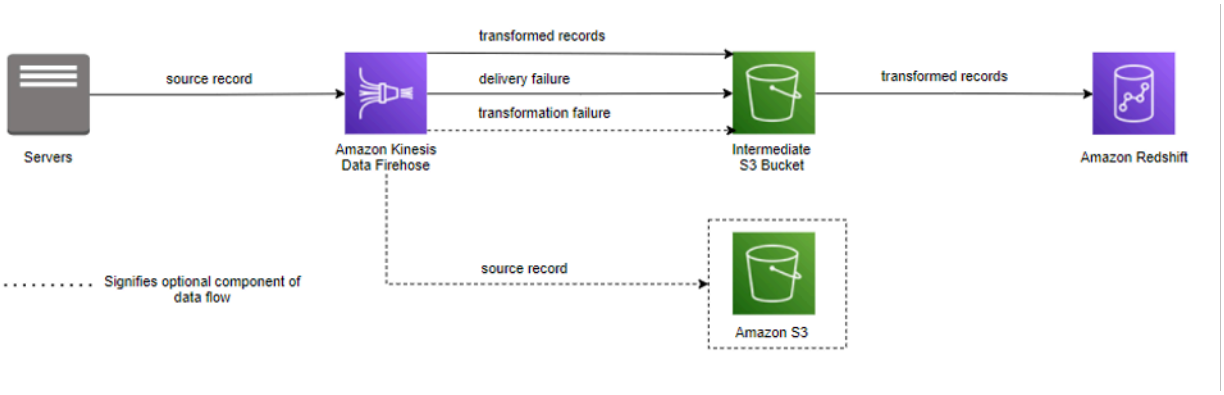
Amazon Redshift へのデータ配信

Amazon Redshift へのデータ配信では、Kinesis Data Firehose は最初に受信データを S3 バケットに前述の形式で配信します。その後、Kinesis Data Firehose は Amazon Redshift COPY コマンドを発行して、S3 バケットから Amazon Redshift クラスターにデータをロードします。

S3 から Amazon Redshift へのデータ COPY オペレーションの頻度は、Amazon Redshift クラスターの COPY コマンド処理速度によって異なります。Amazon Redshift の送信先では、データ配信の失

敗を処理する配信ストリームの作成時に再試行期間 (0 ~ 7,200 秒) を指定できます。Kinesis Data Firehose は指定された期間にわたって再試行し、失敗した場合は S3 オブジェクトの特定のバッチをスキップします。スキップされたオブジェクトの情報は、errors/ フォルダーのマニフェストファイルとして S3 バケットに配信されます。この情報は手動のバックフィルに使用できます。

以下は、Kinesis Data Firehose から Amazon Redshift へのデータフローのアーキテクチャ図表です。このデータフローは Amazon Redshift に固有のものですが、Kinesis Data Firehose は他のターゲットに対しても同様のパターンに従います。



Kinesis Data Firehose から Amazon Redshift へのデータフロー

Amazon OpenSearch Service (OpenSearch Service)

OpenSearch Service は、本番ワークロードに必要な可用性、スケーラビリティ、セキュリティに加えて、OpenSearch の使いやすい API とリアルタイム機能を提供するフルマネージドサービスです。OpenSearch Service を使用すると、ログ分析、全文検索、アプリケーションモニタリングのための OpenSearch のデプロイ、運用、スケーリングが容易になります。

OpenSearch Service へのデータ配信

Kinesis Data Firehose は、OpenSearch Service へのデータ配信のため、配信ストリームのバッファリング設定に基づいて受信レコードをバッファしてから、OpenSearch クラスターに複数のレコードのインデックスを作成する、OpenSearch 一括リクエストを生成します。OpenSearch Service へのデータ配信の頻度は、OpenSearch のバッファサイズ (1 MB ~ 100 MB) とバッファ間隔 (60 秒 ~ 900 秒) の値のいずれか早い方によって決まります。

OpenSearch Service の送信先で、配信ストリームの作成時に、再試行の期間 (0 ~ 7200 秒) を指定できます。Kinesis Data Firehose は指定された期間にわたって再試行してから、その特定のインデックスリクエストをスキップします。スキップされたドキュメントは `elasticsearch_failed/` フォルダーの S3 バケットに配信され、手動のバックフィルに使用できます。

Amazon Kinesis Data Firehose では、期間に基づいて OpenSearch Service インデックスをローテーションできます。選択したローテーションオプション (NoRotation、OneHour、OneDay、OneWeek、OneMonth) に応じて、Kinesis Data Firehose は協定世界時 (UTC) の到着タイムスタンプの一部を指定されたインデックス名に追加します。

カスタム HTTP エンドポイントまたはサポートされているサードパーティーサービスプロバイダー

Kinesis Data Firehose は、カスタム HTTP エンドポイント、または Datadog、Dynatrace、LogicMonitor、MongoDB、New Relic、Splunk、Sumo Logic など、サポートされているサードパーティープロバイダーにデータを送信できます。

カスタム HTTP エンドポイントまたはサポートされているサードパーティーサービスプロバイダー

Kinesis Data Firehose がカスタム HTTP エンドポイントにデータを正常に配信するには、これらのエンドポイントがリクエストを受け入れ、特定の Kinesis Data Firehose リクエスト形式およびレスポンス形式を使用してレスポンスを送信する必要があります。

サポートされているサードパーティーサービスプロバイダーが所有する HTTP エンドポイントにデータを配信する場合、統合されている AWS Lambda サービスを使用して、受信レコードを、サービスプロバイダーに組み込まれているものが対応している形式に変換する関数を作成できます。

データ配信の頻度については、各サービスプロバイダーに推奨バッファーサイズがあります。推奨バッファサイズの詳細については、サービスプロバイダーにお問い合わせください。データ配信の失敗処理では、Kinesis Data Firehose はまず、送信先からの応答を待って HTTP エンドポイントとの接続を確立します。Kinesis Data Firehose は、再試行期間が終了するまで接続を確立し続けます。その後、Kinesis Data Firehose はこれをデータ配信の失敗と見なし、データを S3 バケットにバックアップします。

概要

Kinesis Data Firehose は、サポートされている送信先にストリーミングデータを持続的に配信できます。フルマネージドソリューションであり、開発の必要性はわずか、または皆無です。ABC2Badge 社にとって、Kinesis Data Firehose を使用するのは自然な選択でした。どうしゃは既に Amazon Redshift をデータウェアハウスソリューションとして使用していました。データソースはトランザクションログに継続的に書き込むため、Amazon Kinesis Agent を活用して、追加のコードを記述することなくそのデータをストリーミングできました。ABC2Badge 社はセンサーレコードのストリーミングを作成し、Kinesis Data Firehose 経由でこれらのレコードを受信するようになったため、これをセキュリティチームのユースケースのベースとして使用できます。

シナリオ 3: データインサイトプロセスのためのクリックストリームデータの準備

Fast Sneakers は、トレンディなスニーカーを中心としたファッションブティックです。在庫や、昨夜テレビでそのブランドのスニーカーを身に着けた有名人や有名選手が映ったなどのトレンドに応じて、特定の靴の価格が上下する可能性があります。Fast Sneakers では、収益を最大化するために、このようなトレンドを追跡および分析することが重要です。

Fast Sneakers は、保守が必要な新しいインフラストラクチャを伴う追加のオーバーヘッドをプロジェクトに導入することを望んでいません。同社は、データエンジニアがデータ変換に集中し、データサイエンティストが独自に機械学習機能に取り組めるような、適切なグループに開発を分割したいと考えています。

需要に応じて迅速に対応し、自動的に価格を調整するために、Fast Sneakers は、重要なイベント (クリックインタレストや購入データなど) をストリーミングし、イベントデータを変換および拡張して ML モデルにフィードしています。機械学習モデルは、価格調整が必要かどうかを判断できます。これにより、 Fast Sneakers は、自社製品の利益を最大化するために、価格を自動的に変更できます。

Fast Sneakers のリアルタイムの価格調整

このアーキテクチャ図表は、Fast Sneakers が Kinesis Data Streams、AWS Glue、および DynamoDB Streams を使用して作成したリアルタイムストリーミングソリューションを示しています。これらのサービスを活用することで、同社はサポートインフラストラクチャのセットアップや保守に時間を費やすことなく、伸縮性と信頼性に優れたソリューションを得ることができました。スト

リーミングの抽出、変換、ロード (ETL) ジョブと機械学習モデルに集中することで、会社に価値をもたらすものに時間を費やすことができます。

ワークロードで使用されるアーキテクチャとテクノロジーをよりよく理解するために、使用するサービスの詳細を以下に示します。

AWS Glue および AWS Glue ストリーミング

AWS Glue は、データのカタログ化、クリーニング、エンリッチ化、データストア間での確実な移動に使用できるフルマネージド型 ETL サービスです。AWS Glue を使用すると、ETL ジョブの作成にかかるコスト、複雑さ、時間を大幅に削減できます。AWS Glue はサーバーレスであるため、インフラストラクチャをセットアップまたは管理する必要はありません。ジョブの実行中に使用するリソースに対してのみ支払いが発生します。

AWS Glue を使用すると、AWS Glue ストリーミング ETL ジョブを使用してコンシューマーアプリケーションを作成できます。これにより、Apache Spark やその他の Spark ベースのモジュール書き込みを利用して、イベントデータを消費および処理できます。このドキュメントの次のセクションでは、このシナリオについてさらに詳しく説明します。

AWS Glue Data Catalog

AWS Glue Data Catalog には、AWS Glue で ETL ジョブのソースおよびターゲットとして使用されるデータへの参照が含まれています。AWS Glue Data Catalog は、データの場所、スキーマ、およびランタイムメトリクスへのインデックスです。データカタログ内の情報は、ETL ジョブの作成とモニターに使用できます。Data Catalog の情報はメタデータテーブルとして保存され、各テーブルが 1 つのデータストアを指定します。クローラを設定すると、DynamoDB、S3、Java Database Connectivity (JDBC) 接続ストアなど、さまざまな種類のデータストアを自動的に評価し、メタデータとスキーマを抽出して、AWS Glue Data Catalog でテーブル定義を作成できます。

AWS Glue ストリーミング ETL ジョブで Amazon Kinesis Data Streams を操作するには、AWS Glue Data Catalog データベースのテーブルにストリーミングを定義するのがベストプラクティスです。サポートされている多くの形式 (CSV、JSON、ORC、Parquet、Avro、または Grok を使用したお客様の形式) の 1 つである Kinesis のストリーミングを使用して、ストリームソーステーブルを定義します。スキーマを手動で入力することも、このステップを AWS Glue ジョブに任せてジョブの実行時に決定することもできます。

AWS Glue ストリーミング ETL ジョブ

AWS Glue は、Apache Spark サーバーレス環境で ETL ジョブを実行します。AWS Glue は、独自のサービスアカウントでプロビジョンして管理する仮想リソースでこれらのジョブを実行しま

す。AWS Glue では、Apache Spark ベースのジョブを実行できることに加えて、DynamicFrames を使用した Spark の上に、追加レベルの機能が提供されます。

DynamicFrames は、構造体や配列などのネストされたデータをサポートする分散テーブルです。各レコードは自己記述型であり、半構造化データのスキーマの柔軟性を持つよう設計されています。`DynamicFrame` のレコードには、データと、そのデータを記述するスキーマの両方が含まれます。Apache Spark `DataFrames` と `DynamicFrames` はどちらも ETL スクリプトでサポートされており、前後に変換できます。`DynamicFrames` は、データクリーニングと ETL のための高度な変換のセットを提供します。

AWS Glue ジョブで Spark ストリーミングを使用することで、継続的に実行されるストリーミング ETL ジョブを作成し、Amazon Kinesis Data Streams、Apache Kafka、Amazon MSK などのストリーミングソースからデータを消費できます。ジョブはデータのクリーニング、マージ、変換を行い、その結果を Simple Storage Service (Amazon S3)、Amazon DynamoDB、JDBC データストアなどのストアにロードできます。

デフォルトでは、AWS Glue によるデータ処理と書き出しは 100 秒ウィンドウ単位で行われます。これにより、データを効率的に処理しつつ、想定より遅く到着したデータに対する集計を実行できます。ウィンドウサイズは、応答速度と集計の精度に合わせて調整することで構成できます。AWS Glue ストリーミングジョブは、Kinesis Data Stream から読み取られたデータを追跡するためにチェックポイントを使用します。AWS Glue でのストリーミング ETL ジョブの作成に関するチュートリアルについては、AWS Glue の「ストリーミング ETL ジョブの追加」を参照してください。

Amazon DynamoDB

Amazon DynamoDB は、あらゆる規模で 10 ミリ秒未満のパフォーマンスを実現する key-value データベースおよびドキュメントデータベースです。これはフルマネージド型でマルチリージョン、マルチアクティブで耐久性があるデータベースであり、セキュリティ、バックアップと復元、インターネット規模のアプリケーション用のインメモリキャッシュが組み込まれています。DynamoDB では、1 日あたり 10 兆件を超えるリクエストを処理でき、ピーク時で 1 秒あたり 2,000 万件を超えるリクエストに対応できます。

DynamoDB ストリームのデータキャプチャの変更

DynamoDB ストリームは、DynamoDB テーブル内の項目に加えられた変更に関する情報の順序付けされた情報です。テーブルでストリーミングを有効にすると、DynamoDB はテーブル内のデータ項目に加えられた各変更に関する情報をキャプチャします。DynamoDB は AWS Lambda 上で実行されているため、トリガー (DynamoDB ストリーム内のイベントに自動的に応答するコード) を作成で

きます。トリガーを使用すると、DynamoDB テーブル内のデータ変更に対応するアプリケーションを構築できます。

テーブルでストリーミングを有効にすると、ストリーミングの Amazon リソースネーム (ARN) を、お客様が作成した Lambda 関数に関連付けることができます。テーブルの項目が変更されるとすぐに、新しいレコードがテーブルのストリームに表示されます。AWS Lambda はストリーミングをポーリングし、新しいストリームレコードを検出すると、Lambda 関数を同期的に呼び出します。

Amazon SageMaker および Amazon SageMaker サービスエンドポイント

Amazon SageMaker は、デベロッパーやデータサイエンティストが機械学習モデルを迅速かつあらゆる規模で構築、トレーニング、デプロイできるようにするフルマネージドプラットフォームです。SageMaker には、機械学習モデルを構築、トレーニング、デプロイするために、組み合わせと単体のどちらでも使用可能なモジュールが用意されています。Amazon SageMaker サービスエンドポイントを使用すると、Amazon SageMaker の内部または外部で開発したデプロイ済みモデルを使用して、リアルタイム推論用のマネージド型ホストエンドポイントを作成できます。

AWS SDK を利用することで、コンテンツとともにコンテンツタイプ情報を渡す SageMaker エンドポイントを呼び出し、渡されたデータに基づいてリアルタイムの予測を受け取ることができます。これにより、機械学習モデルの設計と開発を、推測された結果に対してアクションを実行するコードから分離しておくことができます。

これにより、データサイエンティストは機械学習に集中でき、機械学習モデルを使用しているデベロッパーは機械学習モデルをコードでどのように使用するかに集中できます。SageMaker でエンドポイントを呼び出す方法の詳細については、Amazon SageMaker API リファレンスの「InvokeEndpoint」を参照してください。

データインサイトをリアルタイムで推測する

前のアーキテクチャ図表は、Fast Sneakers の既存のウェブアプリケーションが、クリックストリームイベントを含む Kinesis Data Stream を追加したことを示しています。このデータストリームは、ウェブサイトからのトラフィックとイベントデータを提供します。分類、製品属性、価格などの情報を含む製品カタログと、注文された商品、請求、配送などのデータを含む注文テーブルは、別個の DynamoDB テーブルです。データストリームソースと該当する DynamoDB テーブルには、AWS Glue ストリーミング ETL ジョブで使用されるメタデータと AWS Glue Data Catalog で定義されたスキーマがあります。

Fast Sneakers は、Apache Spark、Spark ストリーミング、および AWS Glue ストリーミング ETL ジョブの `DynamicFrames` を利用することで、いずれかのデータストリームからデータを抽出し、

それを変換し、商品テーブルと注文テーブルのデータをマージできます。変換からハイドレートされたデータを使用して、推論結果を取得するデータセットが DynamoDB テーブルに送信されます。

テーブルの DynamoDB ストリームは、新しいレコードが書き込まれるたびに Lambda 関数をトリガーします。Lambda 関数は、以前に変換されたレコードを、AWS SDK を使用して SageMaker エンドポイントに送信し、製品にどのような価格調整が必要かを推測します。機械学習モデルで価格の調整が必要であることが特定された場合、Lambda 関数は価格変更をカタログ DynamoDB テーブルに製品に書き込みます。

概要

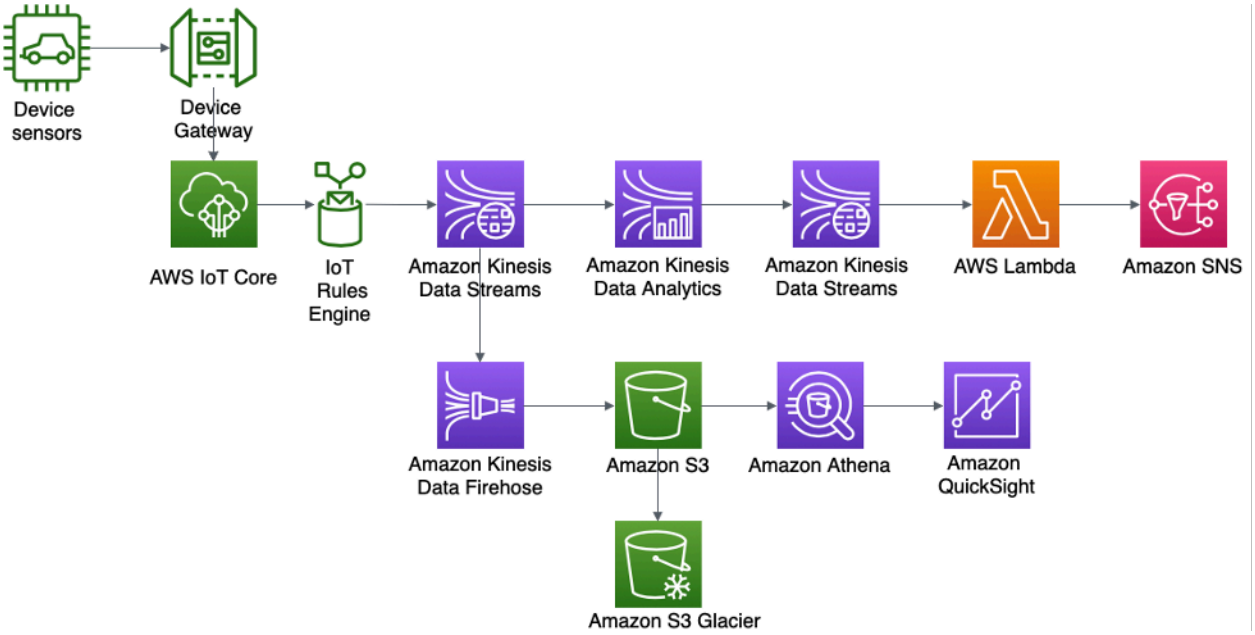
Amazon Kinesis Data Streams を使用すると、リアルタイムのストリーミングデータを簡単に収集、処理、分析できるため、タイムリーなインサイトを得て、新しい情報にすばやく対応できます。AWS Glue サーバーレスデータ統合サービスと組み合わせることで、機械学習用にデータを準備および結合するリアルタイムのイベントストリーミングアプリケーションを作成できます。

Kinesis Data Streams と AWS Glue サービスはどちらもフルマネージド型であるため、AWS はビッグデータプラットフォームのインフラストラクチャ管理という差別化されていない重労働を排除し、お客様はデータに基づいたデータインサイトの生成に集中できます。

Fast Sneakers は、リアルタイムのイベント処理と機械学習を利用して、自社のウェブサイトでリアルタイムの価格調整を完全に自動化し、製品の在庫を最大化できます。これにより、ビッグデータプラットフォームの作成と保守の必要性を回避しながら、ビジネスに最大の価値がもたらされます。

シナリオ 4: デバイスセンサーのリアルタイム異常検出と通知

ABC4Logistics 社は、ガソリン、液体プロパン (LPG)、ナフサなどの可燃性の高い石油製品を港から各都市に輸送しています。位置、エンジン温度、コンテナ内の温度、走行速度、駐車場所、道路状況などをモニタリングするために複数のセンサーが搭載された車両が数百台あります。ABC4Logistics の要件の 1 つは、エンジンとコンテナの温度をリアルタイムでモニタリングし、異常が発生した場合にドライバーとフリートモニタリングチームに警告することです。このような状況を検出し、リアルタイムでアラートを生成するために、ABC4Logistics は次のアーキテクチャを AWS に実装しました。



ABC4Logistics のデバイスセンサーのリアルタイム異常検出および通知アーキテクチャ

デバイスセンサーからのデータは AWS IoT Gateway によって取り込まれ、AWS IoT ルールエンジンによってストリーミングデータが Amazon Kinesis Data Streams で利用可能になります。ABC4Logistics は、Kinesis Data Analytics を使用して、Kinesis Data Streams 内のストリーミングデータに対してリアルタイム分析を実行できます。

ABC4Logistics は、Kinesis Data Analytics を使用して、センサーからの温度測定値が 10 秒間にわたって通常の測定値から逸脱しているかどうかを検出し、そのレコードを別の Kinesis Data Streams インスタンスに取り込み、異常なレコードを特定できます。Amazon Kinesis Data Streams は Lambda 関数を呼び出し、Amazon SNS を介してドライバーとフリートモニタリングチームにアラートを送信できます。

Kinesis Data Streams 内のデータも Amazon Kinesis Data Firehose にプッシュダウンされます。Amazon Kinesis Data Firehose はこのデータを Simple Storage Service (Amazon S3) に保持するため、ABC4Logistics はセンサーデータに対してバッチまたはほぼリアルタイムの分析を実行できます。ABC4Logistics では、Amazon Athena を使用して S3 内のデータをクエリし、Amazon QuickSight を使用して可視化を行います。データを長期間保持するには、S3 ライフサイクルポリシーを使用して Amazon S3 Glacier にデータをアーカイブします。

次に、このアーキテクチャの重要なコンポーネントについて詳しく説明します。

Amazon Kinesis Data Analytics

Amazon Kinesis Data Analytics を使用すると、ストリーミングデータを変換および分析し、異常値にリアルタイムで対応できます。これは AWS 上のサーバーレスサービスです。つまり、Kinesis Data Analytics がプロビジョニングを処理し、あらゆるデータスループットを処理できるようにインフラストラクチャを伸縮自在にスケールします。これにより、ストリーミングインフラストラクチャのセットアップと管理という差別化されていない重労働が排除され、より多くの時間をストリーミングアプリケーションの作成に費やすことができます。

Amazon Kinesis Data Analytics を使用すると、複数のオプション (スタンダード SQL、Java、Python、Scala の Apache Flink アプリケーションを含む) を使用してインタラクティブにストリーミングデータをクエリしたり、Java を使用して Apache Beam アプリケーションを構築してデータストリームを分析したりできます。

これらのオプションにより、ストリーミングアプリケーションとソース/ターゲットのサポートの複雑度に応じて、特定のアプローチを柔軟に使用できます。次のセクションでは、Flink アプリケーション用 Kinesis Data Analytics オプションについて説明します。

Amazon Kinesis Data Analytics for Apache Flink アプリケーション

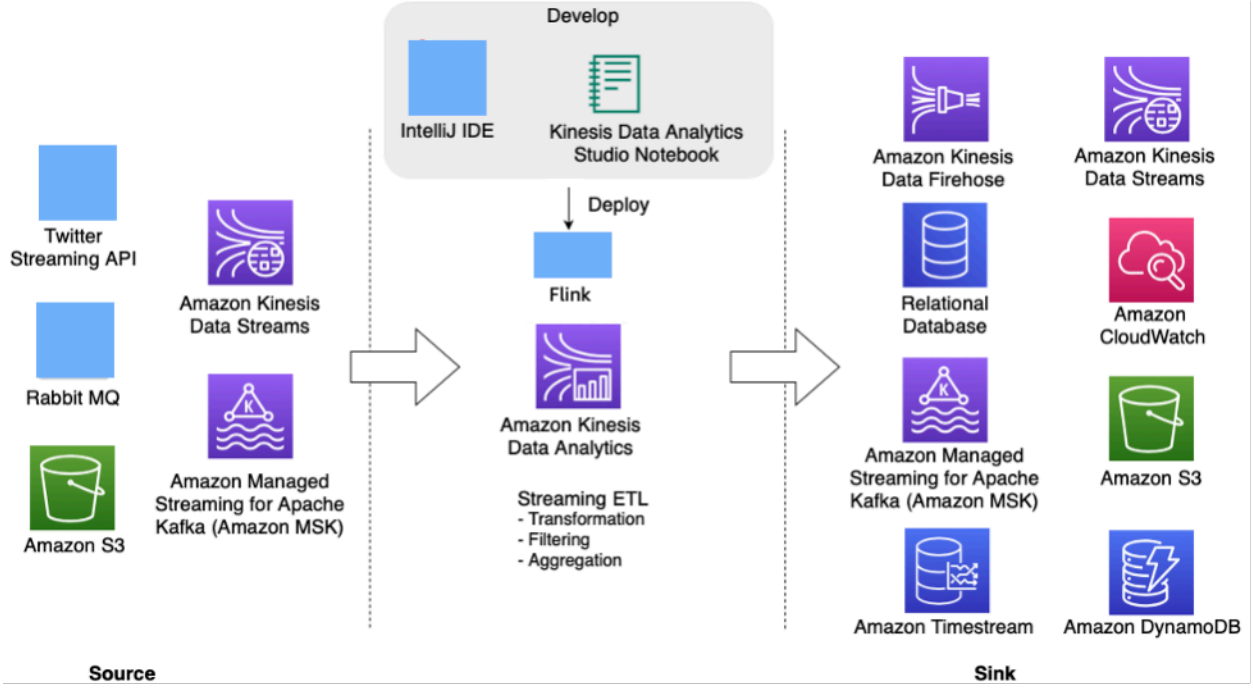
Apache Flink は人気の高いオープンソースフレームワークであり、無制限および制限付きデータストリームに対するステートフルな計算を行う分散処理エンジンです。Apache Flink は、exactly-one セマンティクスをサポートし、インメモリ速度とスケールで計算を実行するように設計されています。Apache Flink ベースのアプリケーションは、耐障害性の高い方法で、高スループットで低レイテンシーを実現するのに役立ちます。

Amazon Kinesis Data Analytics for Apache Flink を使用すると、複雑な分散型 Apache Flink 環境を管理しなくても、ストリーミングソースに対してコードを作成して実行し、時系列分析の実行、リアルタイムダッシュボードのフィード、リアルタイムのメトリクスの作成を行うことができます。高レベルの Flink プログラミング機能は、自分で Flink インフラストラクチャをホストする場合と同じ方法で使用できます。

Apache Flink 用 Kinesis Data Analytics を使用すると、Java、Scala、Python、または SQL でアプリケーションを作成して、ストリーミングデータを処理および分析できます。典型的な Flink アプリケーションは、入力ストリームまたはデータロケーションまたはソースからデータを読み取り、演算子または関数を使用してデータを変換/フィルターまたは結合し、データを出力ストリームまたはデータロケーション、またはシンクに保存します。

以下のアーキテクチャ図表は、Kinesis Data Analytics Flink アプリケーションでサポートされているソースとシンクの一部を示しています。ソース/シンク用にあらかじめバンドルされたコネクタに加

えて、Kinesis Data Analytics の Flink アプリケーション用のさまざまなソース/シンクにカスタムコネクタを取り込むこともできます。



リアルタイムストリーミング処理のための Kinesis Data Analytics 上の Apache Flink アプリケーション

デベロッパーは、お好みの IDE を使用して Flink アプリケーションを開発し、AWS Management Console または DevOps ツールから Kinesis Data Analytics にデプロイできます。

Amazon Kinesis Data Analytics Studio

Kinesis Data Analytics Studio は、Kinesis Data Analytics サービスの一部として、リアルタイムでデータストリームをインタラクティブにクエリしたり、SQL、Python、および Scala を使用してストリーム処理アプリケーションを簡単に構築して実行したりできます。Studio ノートブックは Apache Zeppelin を搭載しています。

Studio ノートブックを使用すると、ノートブック環境で Flink アプリケーションコードを開発し、コードの結果をリアルタイムで表示して、ノートブック内で可視化できます。Apache Zeppelin と Apache Flink を搭載した Studio ノートブックは、Kinesis Data Streams と Amazon MSK コンソールからワンクリックで作成することも、Kinesis Data Analytics コンソールから起動することもできます。

Kinesis Data Analytics Studio の一部としてコードを反復的に開発したら、ノートブックを Kinesis Data Analytics アプリケーションとしてデプロイし、ストリーミングモードでの継続的な実行、ソー

スからのデータの読み取り、送信先への書き込み、長時間実行されるアプリケーションの状態の維持、ソースストリームのスループットに基づいて自動的なスケーリングを行うことができます。以前、お客様は AWS でのリアルタイムストリーミングデータのインタラクティブな分析に Kinesis Data Analytics for SQL アプリケーションを使用していました。

Kinesis Data Analytics for SQL アプリケーションは引き続き使用できますが、新しいプロジェクトでは、新しい Kinesis Data Analytics Studio を使用することをお勧めします。Kinesis Data Analytics Studio は使いやすさと高度な分析機能を組み合わせており、洗練されたストリーム処理アプリケーションでも数分で構築できます。

Kinesis Data Analytics Flink アプリケーションに耐障害性を備えるには、「Kinesis Data Analytics for Apache Flink での耐障害性の実装」で説明されているように、チェックポイントとスナップショットを利用できます。

Kinesis Data Analytics Flink アプリケーションは、データ処理でexactly-one セマンティクスのアプリケーション、チェックポイント機能、Kinesis Data Streams、Kinesis Data Firehose、Amazon MSK、Rabbit MQ、Apache Cassandra (カスタムコネクタを含む) などのデータソースからのデータ処理など、複雑なストリーミング分析アプリケーションを作成するのに便利です。

Flink アプリケーションでストリーミングデータを処理した後、Amazon Kinesis Data Streams、Amazon Kinesis Data Firehose、Amazon DynamoDB、Amazon OpenSearch Service、Amazon Timestream、Simple Storage Service (Amazon S3) など、さまざまなシンクまたは送信先にデータを永続化できます。Kinesis Data Analytics Flink アプリケーションでは、1 秒未満のパフォーマンス保証も提供されます。

Kinesis Data Analytics 向け Apache Beam アプリケーション

Apache Beam は、ストリーミングデータを処理するためのプログラミングモデルです。Apache Beam は、さまざまなエンジン、または Flink、Spark ストリーミング、Apache Samza などのランナーで実行できる高度なデータ並列処理パイプラインを構築するためのポータブル API レイヤーを提供します。

Kinesis Data Analytics アプリケーションで Apache Beam フレームワークを使用して、ストリーミングデータを処理できます。Apache Beam を使用する Kinesis Data Analytics アプリケーションは、Apache Flink ランナーを使用して Beam パイプラインを実行します。

概要

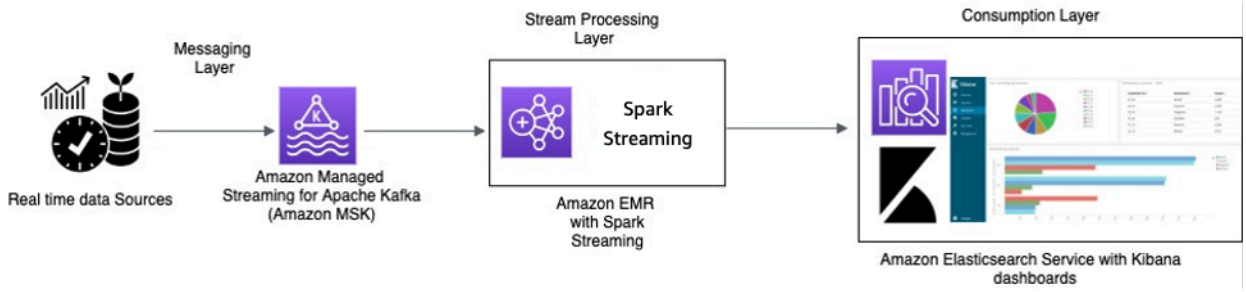
AWS ストリーミングサービスである Amazon Kinesis Data Streams、Amazon Kinesis Data Analytics、および Amazon Kinesis Data Firehose を利用することで、

ABC4Logistics は、温度測定値の異常なパターンを検出し、ドライバーと車両管理チームにリアルタイムで通知して、車両の完全な故障や火災などの重大な事故を防止できます。

シナリオ 5: Apache Kafka を使用したリアルタイムのテレメトリデータのモニタリング

ABC1Cabs はオンラインタクシー予約サービス会社です。すべてのタクシーには、車両からテレメトリデータを収集する IoT デバイスがあります。現在、ABC1Cabs は、リアルタイムのイベント消費、システムヘルスメトリクスの収集、アクティビティの追跡、オンプレミスの Hadoop クラスター上に構築された Apache Spark ストリーミングプラットフォームへのデータのフィードを目的として設計された Apache Kafka クラスターを実行しています。

ABC1Cabs は、ビジネスメトリクス、デバッグ、アラート、その他のダッシュボードの作成に OpenSearch Dashboards を使用しています。同社は、Amazon MSK、Spark ストリーミングを使用した Amazon EMR、および OpenSearch Dashboards を備えた OpenSearch Service に関心を持っています。Apache Kafka および Hadoop クラスターを維持するための管理オーバーヘッドを削減すると同時に、使い慣れたオープンソースソフトウェアと API を使用してデータパイプラインをオーケストレートすることが要件です。次のアーキテクチャ図表は、同社の AWS でのソリューションを示しています。



Amazon MSK によるリアルタイム処理、および Amazon EMR での Apache Spark ストリーミングおよび OpenSearch Dashboards を備えた Amazon OpenSearch Service を使用したストリーム処理

タクシーの IoT デバイスはテレメトリデータを収集し、ソースハブに送信します。ソースハブは Amazon MSK にリアルタイムでデータを送信するように設定されています。Amazon MSK は Apache Kafka プロデューサーライブラリ API を使用して、データを Amazon EMR クラスターにストリーミングするように設定されます。Amazon EMR クラスターには Kafka クライアントと Spark ストリーミングがインストールされており、データストリームを消費して処理できます。

Spark ストリーミングには、Elasticsearch の定義済みインデックスにデータを直接書き込むことができるシンクコネクタがあります。OpenSearch Dashboards を備えた Elasticsearch クラスターは、メトリクスとダッシュボードに使用できます。Amazon MSK、Spark ストリーミングを使用した

Amazon EMR、OpenSearch Dashboards を備えた OpenSearch Service はすべてマネージドサービスであり、AWS がさまざまなクラスターのインフラストラクチャ管理という差別化されていない重労働を管理します。これにより、使い慣れたオープンソースソフトウェアを使用して、数回のクリックでアプリケーションを構築できます。次のセクションでは、これらのサービスについて詳しく説明します。

Amazon Managed Streaming for Apache Kafka (Amazon MSK)

Apache Kafka は、お客様がクリックストリームイベント、トランザクション、IoT イベント、アプリケーションログ、マシンログなどのストリーミングデータをキャプチャできるオープンソースプラットフォームです。この情報を使用して、リアルタイム分析を実行し、継続的な変換を実行し、このデータをデータレイクやデータベースにリアルタイムで配信するアプリケーションを開発できます。

Kafka をストリーミングデータストアとして使用して、アプリケーションをプロデューサーとコンシューマーから切り離し、2 つのコンポーネント間で信頼性の高いデータ転送を実現できます。Kafka は一般的なエンタープライズデータストリーミングおよびメッセージングプラットフォームですが、本番環境ではセットアップ、スケーリング、管理が難しい場合があります。

Amazon MSK はこれらの管理タスクを処理し、高可用性とセキュリティのためのベストプラクティスに従った環境で、Apache Zookeeper とともに Kafka を簡単にセットアップ、設定、実行します。お客様は引き続き Kafka のコントロールプレーンオペレーションとデータプレーンオペレーションを使用して、データの生成と消費を管理できます。

Amazon MSK はオープンソースの Apache Kafka を実行および管理しているため、お客様はアプリケーションコードを変更することなく、既存の Apache Kafka アプリケーションを AWS 上で簡単に移行して実行できます。

スケーリング

Amazon MSK では、ユーザーがクラスターの実行中にアクティブにスケールできるように、スケーリングオペレーションを提供しています。Amazon MSK クラスターを作成する際に、クラスターの起動時のブローカーのインスタンスタイプを指定できます。Amazon MSK クラスター内で少ないブローカーから始めて、その後、AWS Management Console または AWS CLI を使用して、クラスターあたり数百のブローカーまでスケールアップできます。

または、Apache Kafka ブローカーのサイズまたはファミリーを変更して、クラスターをスケールすることもできます。ブローカーのサイズまたはファミリーを変更すると、ワークロードの変化に伴って Amazon MSK クラスターのコンピューティング性能を柔軟に調整できます。Amazon MSK クラスターに適したブローカー数を決定するには、Amazon MSK Sizing and Pricing spreadsheet (ファイ

ルのダウンロード) を使用します。このスプレッドシートでは、Amazon MSK と同様のセルフマネージド型の EC2 ベースの Apache Kafka クラスターと比較した、Amazon MSK の関連コストのサイジングの見積もりを示します。

Amazon MSK クラスターを作成したら、ブローカーごとの EBS ストレージ容量を増やすことができます。ただし、ストレージを減らすことはできません。ストレージボリュームは、このスケールアップオペレーション中も引き続き使用できます。オートスケーリングと手動スケーリングの 2 種類のスケーリングオペレーションが用意されています。

Amazon MSK では、アプリケーションのオートスケーリングポリシーを使用して、使用量の増加に応じてクラスターのストレージを自動的に拡張できます。オートスケーリングポリシーにより、ターゲットディスク使用率と最大スケーリング容量が設定されます。

ストレージ使用率のしきい値は、Amazon MSK がオートスケーリングオペレーションをトリガーするのに役立ちます。手動スケーリングを使用してストレージを増やすには、クラスターが `ACTIVE` 状態になるまで待ちます。ストレージのスケーリングでは、イベント間のクールダウン期間が最低 6 時間となります。このオペレーションによって追加のストレージがすぐに使用可能になりますが、クラスターを最適化するために最大 24 時間以上かかる場合があります。

最適化の所要時間は、ストレージサイズに比例します。さらに、AWS リージョン内でマルチアベイラビリティーゾーンのレプリケーションも提供され、高可用性を提供します。

設定

Amazon MSK は、ブローカー、トピック、および Apache Zookeeper ノードのデフォルト設定を提供します。また、カスタム設定を作成し、それらを使用して新しい Amazon MSK クラスターを作成したり、既存のクラスターを更新することもできます。カスタム Amazon MSK 設定を指定せずに MSK クラスターを作成すると、Amazon MSK はデフォルト設定を作成して使用します。これらのデフォルト値のリストについては、「Apache Kafka の設定」を参照してください。

Amazon MSK は、モニタリング目的で Apache Kafka メトリクスを収集し、Amazon CloudWatch に送信します。このメトリクスは Amazon CloudWatch で確認できます。MSK クラスター用に設定するメトリクスは、自動的に収集され、CloudWatch にプッシュされます。コンシューマーラグをモニタリングすると、トピックで利用可能な最新データに追いついていない、遅いコンシューマーやスタックしたコンシューマーを特定できます。必要に応じて、それらのコンシューマーのスケーリングや再起動などの是正措置を講じることができます。

Amazon MSK への移行

オンプレミスから Amazon MSK への移行は、次のいずれかの方法で実現できます。

- MirrorMaker2.0 - MirrorMaker2.0 (MM2) MM2 は、Apache Kafka Connect フレームワークをベースにしたマルチクラスターのデータレプリケーションエンジンです。MM2 は Apache Kafka ソースコネクタとシンクコネクタの組み合わせです。1 つの MM2 クラスターを使用して、複数のクラスター間でデータを移行できます。MM2 は、新しいトピックとパーティションを自動的に検出すると同時に、トピックの設定がクラスター間で同期されるようにします。MM2 では、移行 ACL、トピック設定、オフセット変換がサポートされています。移行に関する詳細については、「Apache Kafka の MirrorMaker を使用したクラスターの移行」を参照してください。MM2 は、トピックの構成とオフセット変換の自動レプリケーションに関連するユースケースに使用されます。
- Apache Flink - MM2 は少なくとも 1 回のセマンティクスをサポートしています。レコードは送信先に複製することができ、コンシューマーは重複レコードを処理するためにべき等であるべきです。exactly-once シナリオでは、コンシューマーが Apache Flink を使用できるセマンティクスが必要です。これは、厳密に 1 回のセマンティクスを実現するための代替手段を提供します。

  Apache Flink は、データが送信先クラスターに送信される前にマッピングまたは変換アクションを必要とするシナリオにも使用できます。Apache Flink は、ある Apache Kafka クラスターからデータを読み取り、別の Apache Kafka クラスターに書き込むことができるソースとシンクを備えた Apache Kafka 用のコネクターを提供します。Amazon EMR クラスターを起動するか、Amazon Kinesis Data Analytics を使用してアプリケーションとして Apache Flink を実行することにより、Apache Flink を AWS 上で実行できます。
- AWS Lambda - AWS Lambda のイベントソースとして Apache Kafka がサポートされ、お客様は Lambda 関数を介してトピックからのメッセージを使用できるようになりました。AWS Lambda サービスは、イベントソースからの新しいレコードまたはメッセージを内部でポーリングし、ターゲット Lambda 関数を同期的に呼び出してこれらのメッセージを消費します。Lambda はメッセージをバッチで読み取り、処理のためにイベントペイロード内の関数にメッセージバッチを提供します。消費されたメッセージは、変換または送信先の Amazon MSK クラスターに直接書き込みできます。

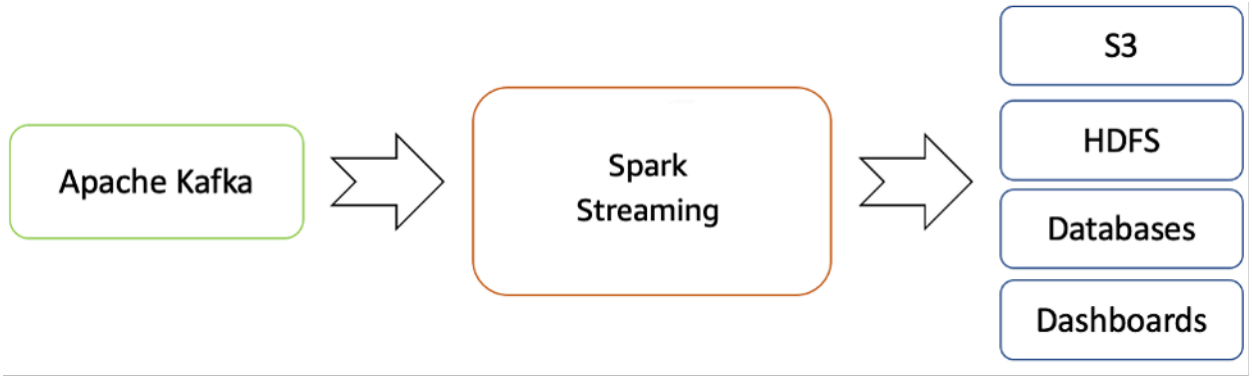
Spark ストリーミングを使用した Amazon EMR

Amazon EMR は、AWS でビッグデータフレームワーク (Apache Hadoop や Apache Spark など) の実行を簡素化して、大量のデータを処理および分析するマネージド型クラスタープラットフォームです。

Amazon EMR は Spark の機能を提供し、Spark ストリーミングを開始して Kafka からデータを消費するために使用できます。Spark ストリーミングは、スケーラブルで高スループット、耐障害性を備えたライブデータストリームのストリーミング処理を可能にするコア Spark API の拡張機能です。

Amazon EMR クラスターは、AWS Command Line Interface (AWS CLI) または AWS Management Console を使用して作成し、クラスターの作成時に詳細設定で Spark および Zeppelin を選択できます。次のアーキテクチャ図表に示すように、データは Apache Kafka や Kinesis Data Streams などの多くのソースから取り込まれ、map、reduce、join、window などの高レベル関数で記述された複雑なアルゴリズムを使用して処理できます。詳細については、「DStreams での変換」を参照してください。

処理されたデータは、ファイルシステム、データベース、ライブダッシュボードにプッシュできます。



Apache Kafka から Hadoop エコシステムへのリアルタイムストリーミングフロー

デフォルトでは、Apache Spark ストリーミングにはマイクロバッチ実行モデルがあります。しかし、Spark 2.3 のリリースから、Apache は Continuous Processing と呼ばれる新しい低レイテンシー処理モードを導入しました。このモードでは、at-least-once 保証で 1 ミリ秒という低いエンドツーエンドのレイテンシーを実現できます。

クエリの Dataset/DataFrames オペレーションを変更しなくても、アプリケーションの要件に基づいてモードを選択できます。Spark ストリーミングの利点には、次のようなものがあります。

- Apache Spark の言語統合 API がストリーミング処理に導入され、バッチジョブを記述するのと同じ方法でストリーミングジョブを記述できます。
- Java、Scala、Python をサポートしています。
- 余分なコードなしで、失われた作業とオペレーターの状態 (スライディングウィンドウなど) の両方を、すぐに回復できます。
- Spark ストリーミングは、Spark 上で実行することで、バッチ処理に同じコードを再利用したり、履歴データに対してストリーミングを結合したり、ストリーミングの状態に対してアドホッククエリを実行したりして、分析だけでなく強力な対話型アプリケーションを構築できます。

- Spark ストリーミングでデータストリームが処理された後、OpenSearch Sink Connector を使用して OpenSearch Service クラスターにデータを書き込むことができます。一方、OpenSearch Dashboards を使用する OpenSearch Service は、消費レイヤーとして使用できます。

OpenSearch Dashboards を備えた Amazon OpenSearch Service

OpenSearch Service は、AWS クラウドで OpenSearch クラスターを簡単にデプロイ、運用、スケールするマネージドサービスです。OpenSearch はログ分析、リアルタイムのアプリケーションモニタリング、クリックストリーム分析などのユースケース向けの、人気の高いオープンソースの検索および分析エンジンです。

OpenSearch Dashboards は、ログと時系列の分析、アプリケーションのモニタリング、オペレーショナルインテリジェンスのユースケースに使われる、オープンソースのデータの可視化および調査ツールです。ヒストグラム、線グラフ、円グラフ、ヒートマップ、組み込み型の地理空間のサポートなど、強力で使いやすい機能を提供します。

OpenSearch Dashboards は、人気のある分析および検索エンジンである OpenSearch との緊密な統合を提供します。OpenSearch Dashboards は、OpenSearch に保存されているデータを可視化するためのデフォルトの選択肢となっています。OpenSearch Service では、すべての OpenSearch Service ドメインに OpenSearch Dashboards がインストールされます。OpenSearch Dashboards へのリンクは、OpenSearch Service コンソールのドメインダッシュボードにあります。

概要

Apache Kafka は AWS でマネージドサービスとして提供されているため、通常は Apache Kafka を詳細に理解する必要があるブローカー間の調整の管理ではなく、消費に集中できます。高可用性、ブローカーのスケーラビリティ、きめ細かいアクセスコントロールなどの機能は Amazon MSK プラットフォームによって管理されます。

ABC1Cabs はこれらのサービスを利用して、インフラストラクチャ管理の専門知識を必要とせずに本番アプリケーションを構築しました。Amazon MSK からデータを消費し、さらに可視化レイヤーに伝播する、処理レイヤーに集中できます。

Amazon EMR の Spark ストリーミングは、ストリーミングデータのリアルタイム分析や、可視化レイヤーでの Amazon OpenSearch Service の OpenSearch Dashboards へのパブリッシュに役立ちます。

結論と寄稿者

まとめ

このドキュメントでは、ストリーミングワークフローのシナリオをいくつか確認しました。これらのシナリオでは、ストリーミングデータ処理により、サンプル企業は新しい機能を追加できるようになりました。

データの作成時にデータを分析することで、ビジネスが今何をしているのかについてのインサイトを得ることができます。AWS のストリーミングサービスを使用すると、インフラストラクチャのデプロイや管理に煩わされることなく、アプリケーションに集中して時間的制約のあるビジネス上の意思決定を行うことができます。

寄稿者

- AWS、シニアソリューションアーキテクト、Amalia Rabinovitch
- AWS、データアーキテクト、データレイク、Priyanka Chaudhary
- AWS、ソリューションアーキテクト、Zohair Nasimi
- AWS、ソリューションアーキテクト、Rob Kuhr
- AWS、シニアパートナーソリューションアーキテクト、Ejaz Sayyed
- AWS、ソリューションアーキテクト、Allan Macinnis
- AWS、プロダクトマーケティングマネージャー、Chander Matrubhutam

改訂履歴

このホワイトペーパーの更新に関する通知を受け取るには、RSS フィードをサブスクライブしてください。

update-history-change	update-history-description	update-history-date
更新	技術的な正確性を図るために更新されました	2021 年 9 月 1 日
初版公開	ホワイトペーパーの初版公開	2017 年 7 月 1 日