AWS 백서

Amazon Kinesis를 사용한 AWS 기반 스트리밍 데이터 솔루션

Amazon Kinesis를 사용한 AWS 기반 스트리밍 데이터 솔루션: AWS 백서

Table of Contents

AWS의 스트리밍 데이터 솔루션

게시 날짜: 2021년 9월 1일(문서 개정)

요약

데이터 엔지니어, 데이터 분석가 및 빅 데이터 개발자는 기업이 고객, 애플리케이션 및 제품이 현재 수행하는 작업을 파악하고 신속하게 대응할 수 있도록 분석을 배치에서 실시간으로 발전시키려고 합니다. 이 백서는 배치에서 실시간으로 발전한 분석 기능의 변화를 설명합니다. 그리고 Amazon Kinesis Data Streams, Amazon Kinesis Data Firehose, Amazon EMR, Amazon Kinesis Data Analytics, Amazon Managed Streaming for Apache Kafka(Amazon MSK)와 같은 서비스를 사용하여 실시간 애플리케이션을 구현하는 방법을 설명하고 이러한 서비스를 사용하여 일반적인 설계 패턴을 제공합니다.

소개

오늘날 기업은 지속적으로 데이터 스트림을 생성하는 데이터 원본의 폭발적인 증가로 인해 엄청난 규모와 속도로 데이터를 수신합니다. 애플리케이션 서버의 로그 데이터, 웹 사이트 및 모바일 앱의 클릭 스트림 데이터 또는 사물 인터넷(IoT) 디바이스의 원격 분석 데이터 등 모든 데이터에는 고객, 애플리케이션 및 제품이 현재 수행하는 작업을 파악하는 데 도움이 되는 정보가 포함되어 있습니다.

이러한 데이터를 실시간으로 처리하고 분석하는 기능은 애플리케이션을 지속적으로 모니터링하여 높은 서비스 가동 시간을 보장하고 프로모션 제안 및 제품 추천을 개인별 맞춤화하는 등의 작업을 수행하는 데 필수적입니다. 실시간 및 실시간에 가까운 처리는 웹 사이트 분석 및 기계 학습과 같은 다른 일반적인 사용 사례를 이러한 애플리케이션에서 몇 시간 또는 며칠이 아닌 몇 초 또는 몇 분 만에 사용할 수 있도록 하여 더 정확하고 실행 가능하게 만들 수 있습니다.

실시간 및 실시간에 가까운 애플리케이션 시나리오

스트리밍 데이터 서비스를 애플리케이션 모니터링, 사기 탐지, 실시간 순위표 등 실시간에 가까운 실시간 애플리케이션에 사용할 수 있습니다. 실시간 사용 사례에는 수집에서 처리, 대상 데이터 스토어 및 기타 시스템에 결과를 내보내는 데 이르기까지 밀리초 단위의 엔드 투 엔드 대기 시간이 필요합니다. 예를 들어, Netflix는 Amazon Kinesis Data Streams를 사용하여 모든 애플리케이션 간 통신을 모니터링함으로써 문제를 신속하게 탐지하고 해결하여 고객을 위한 높은 서비스 가동 시간과 가용성을 보장합니다. 가장 일반적으로 적용 가능한 사용 사례는 애플리케이션 성능 모니터링이지만 광고 기술, 게임 및 IoT에서 이 범주에 속하는 실시간 애플리케이션의 수가 증가하고 있습니다.

실시간에 가까운 일반적인 사용 사례에는 데이터 과학 및 기계 학습을 위한 데이터 스토어에 대한 분석이 포함됩니다. 스트리밍 데이터 솔루션을 사용하여 실시간 데이터를 데이터 레이크로 지속적으로 로드할 수 있습니다. 그러면 새로운 데이터를 사용할 수 있게 됨에 따라 기계 학습 모델을 좀 더 빈번하게 업데이트하여 결과의 정확성과 신뢰성을 보장할 수 있습니다. 예를 들어 Zillow는 Kinesis Data Streams를 사용하여 공개 기록 데이터 및 MLS(Multiple Listing Service) 목록을 수집한 다음 주택 구매자와 판매자에게 거의 실시간으로 주택의 최신 예상 가격을 제공합니다. ZipRecruiter는 이벤트 로깅 파이프라인을 위해 Amazon MSK를 사용합니다. 이러한 파이프라인은 ZipRecruiter 구인/구직 마켓플레이스에서 매일 발생하는 60억 건이 넘는 이벤트를 수집 및 저장하고 지속적으로 처리하는 핵심적인 인프라 구성 요소입니다.

배치 처리와 스트림 처리의 차이

실시간 스트리밍 데이터를 수집, 준비 및 처리하려면 기존에 배치 분석에 사용했던 도구와는 다른 도구 세트가 필요합니다. 기존 분석을 사용하면 데이터를 수집하여 주기적으로 데이터베이스에 로드하고

몇 시간, 며칠 또는 몇 주 후에 분석할 수 있습니다. 실시간 데이터 분석을 위해서는 다른 접근 방식이 필요합니다. 스트림 처리 애플리케이션은 데이터가 저장되기 전에도 실시간으로 지속적으로 데이터를 처리합니다. 스트리밍 데이터는 빠른 속도로 유입될 수 있으며 데이터 볼륨은 언제든지 증가 및 감소할 수 있습니다. 스트림 데이터 처리 플랫폼은 수신 데이터의 속도와 가변성을 처리하고 데이터가 도착하면 처리할 수 있어야 하며, 대개 시간당 수백만 ~ 수억 개의 이벤트를 처리할 수 있어야 합니다.

스트림 처리를 위한 과제

실시간 데이터가 도착할 때 처리하면 기존 데이터 분석 기술보다 훨씬 빠르게 의사 결정을 내릴 수 있습니다. 하지만 자체 사용자 정의 스트리밍 데이터 파이프라인을 구축하고 운영하는 것은 복잡하고 리소스가 많이 필요한 작업입니다.

- 따라서 수천 개의 데이터 원본에서 동시에 들어오는 데이터를 비용 효율적으로 수집, 준비 및 전송할 수 있는 시스템을 구축해야 합니다.
- 최대 처리량과 짧은 대기 시간을 위해 데이터를 효율적으로 배치 처리하고 전송할 수 있도록 스토리지 및 컴퓨팅 리소스를 미세 조정해야 합니다.
- 다양한 속도의 데이터를 처리할 수 있도록 시스템을 확장하려면 여러 서버를 배포하고 관리해야 합니다.

버전 업그레이드는 복잡하고 비용이 많이 드는 프로세스입니다. 이 플랫폼을 구축한 후에는 중복 데이터를 생성하지 않고 스트림의 적절한 지점에서 데이터 처리를 추적하여 시스템을 모니터링하고 서버 또는 네트워크 장애로부터 복구해야 합니다. 인프라 관리를 위한 전담 팀도 필요합니다. 이 모든 작업에는 귀중한 시간과 비용이 필요하며, 결국 대부분의 기업은 결코 거기에 도달하지 못하고 현상 유지에 안주하며 몇 시간 또는 며칠이 지난 정보로 비즈니스를 운영해야 합니다.

스트리밍 데이터 솔루션: 예

시나리오 1: 위치를 기반으로 한 인터넷 제공

InternetProvider는 전 세계 사용자에게 다양한 대역폭 옵션을 갖춘 인터넷 서비스를 제공하는 회사입니다. 사용자가 인터넷에 가입하면 InternetProvider는 사용자의 지리적 위치에 따라 다양한 대역폭 옵션을 사용자에게 제공합니다. 이러한 요구 사항을 고려하여 InternetProvider는 사용자 세부 정보와 위치를 사용하기 위해 Amazon Kinesis Data Streams를 구현했습니다. 사용자 세부 정보 및 위치는 애플리케이션에 게시되기 전에 다양한 대역폭 옵션으로 보강됩니다. AWS Lambda는 이러한 실시간 보강을 가능하게 합니다.

AWS Lambda를 사용하여 데이터 스트림 처리

Amazon Kinesis Data Streams

Amazon Kinesis Data Streams를 사용하면 널리 사용되는 스트림 처리 프레임워크를 사용하여 실시간 사용자 정의 애플리케이션을 구축하고 스트리밍 데이터를 다양한 데이터 스토어로 로드할 수 있습니다. Kinesis 스트림은 웹 사이트 클릭스트림, IoT 센서, 소셜 미디어 피드 및 애플리케이션 로그와 같은 소스에서 전송된 수십만 개의 데이터 생산자로부터 이벤트를 지속적으로 수신하도록 구성할 수 있습니다. 애플리케이션은 몇 밀리초 내에 데이터를 읽고 처리할 수 있습니다.

Kinesis Data Streams를 사용하여 솔루션을 구현할 때 Kinesis Data Streams 애플리케이션이라는 사용자 정의 데이터 처리 애플리케이션을 생성합니다. 일반적인 Kinesis Data Streams 애플리케이션은 Kinesis 스트림의 데이터를 데이터 레코드로 읽습니다.

Kinesis Data Streams에 저장되는 데이터는 고가용성과 탄력성을 보장하며, 밀리초 단위로 사용할 수 있습니다. 수십만 개의 소스에서 클릭스트림, 애플리케이션 로그, 소셜 미디어와 같은 다양한 유형의 데이터를 Kinesis 스트림에 지속적으로 추가할 수 있습니다. 그러면 몇 초 안에 Kinesis 애플리케이션에서는 스트림의 데이터를 읽고 처리할 수 있습니다.

Amazon Kinesis Data Streams는 완전관리형 스트리밍 데이터 서비스입니다. 이 서비스는 데이터 처리량 수준에서 데이터를 스트리밍하는 데 필요한 인프라, 스토리지, 네트워킹 및 구성을 관리합니다.

Amazon Kinesis Data Streams에 데이터 전송

Kinesis Data Streams에 데이터를 전송하는 방법에는 여러 가지가 있으므로 유연하게 솔루션을 설계할 수 있습니다.

- 널리 사용되는 여러 언어에서 지원하는 AWS SDK 중 하나를 사용하여 코드를 작성할 수 있습니다.
- Kinesis Data Streams로 데이터를 전송하기 위한 도구인 Amazon Kinesis 에이전트를 사용할 수 있습니다.

Amazon Kinesis Producer Library(KPL)는 개발자가 하나 이상의 Kinesis 데이터 스트림에 대한 높은 쓰기 처리량을 달성할 수 있도록 지원함으로써 생산자 애플리케이션 개발을 간소화합니다.

KPL은 호스트에 설치하는 사용하기 쉽고 고도로 구성 가능한 라이브러리로, 생산자 애플리케이션 코드와 Kinesis Streams API 작업 간에 중개자 역할을 합니다. KPL과 코드 예제를 사용하여 동기 및 비동기식으로 이벤트를 생성하는 KPL의 기능에 대한 자세한 내용은 Writing to your Kinesis Data Streams Using the KPL(KPL을 사용하여 Kinesis Data Streams에 쓰기) 단원을 참조하세요.

Kinesis Data Streams API에는 스트림에 데이터를 추가하는 두 가지 작업인 `PutRecords`와 `PutRecord`가 있습니다. `PutRecords` 작업은 HTTP 요청당 스트림에 여러 레코드를 전송하는 반면 `PutRecord`는 HTTP 요청당 하나의 레코드를 제출합니다. 대부분의 애플리케이션에서 `PutRecords`를 사용하면 더 높은 처리량을 달성할 수 있습니다.

이러한 API에 대한 자세한 내용은 Adding Data to a Stream(스트림에 데이터 추가)을 참조하세요. 각 API 작업에 대한 세부 정보는 Amazon Kinesis Data Streams API Reference(Amazon Kinesis Data Streams API 참조)에서 확인할 수 있습니다.

Amazon Kinesis Data Streams에서 데이터 처리

Kinesis 스트림에서 데이터를 읽고 처리하려면 소비자 애플리케이션을 생성해야 합니다. Kinesis Data Streams용 소비자를 생성하는 방법에는 여러 가지가 있습니다. 이러한 접근 방식에는 KCL을 사용한 스트리밍 데이터 분석을 위해 Amazon Kinesis Data Analytics를 사용하는 방법, AWS Lambda, AWS Glue 스트리밍 ETL 작업을 사용하는 방법 및 Kinesis Data Streams API를 직접 사용하는 방법이 포함됩니다.

Kinesis Data Streams용 소비자 애플리케이션은 Kinesis Data Streams의 데이터를 사용하고 처리하는 데 도움이 되는 KCL을 사용하여 개발할 수 있습니다. KCL은 여러 인스턴스 간 로드 밸런싱, 인스턴

스 장애에 대한 대응, 처리된 레코드에 대한 체크포인트, 리샤딩에 대한 대응과 같이 분산 컴퓨팅과 관련된 여러 가지 복잡한 작업을 처리합니다. KCL을 사용하면 레코드 처리 로직을 작성하는 데 집중할 수 있습니다. 자체 KCL 애플리케이션을 구축하는 방법에 대한 자세한 내용은 Using the Kinesis Client Library(Kinesis Client Library 사용)를 참조하세요.

Lambda 함수를 구독하여 Kinesis 스트림에서 레코드의 배치를 자동으로 읽고 스트림에서 레코드가 탐지되면 이를 처리하도록 할 수 있습니다. AWS Lambda는 새 레코드를 찾기 위해 스트림을 주기적으로(초당 한 번) 폴링하고 새 레코드를 탐지하면 새 레코드를 파라미터로 전달하는 Lambda 함수를 호출합니다. Lambda 함수는 새 레코드가 탐지될 때만 실행됩니다. Lambda 함수를 공유 처리량 소비자(표준 반복기)에 매핑할 수 있습니다.

스트림에서 데이터를 수신하는 다른 소비자와 경쟁하지 않는 전용 처리량이 필요한 경우 향상된 팬아웃이라는 기능을 사용하는 소비자를 구축할 수 있습니다. 이 기능을 사용하면 소비자가 샤드당 1초에 최대 2MB의 데이터 처리량으로 스트림에서 레코드를 수신할 수 있습니다.

대부분의 경우 Kinesis Data Analytics, KCL, AWS Glue 또는 AWS Lambda를 사용하여 스트림의 데이터를 처리해야 합니다. 하지만 원하는 경우 Kinesis Data Streams API를 사용하여 처음부터 소비자 애플리케이션을 생성할 수 있습니다. Kinesis Data Streams API는 스트림에서 데이터를 검색할 수 있는 `GetShardIterator` 및 `GetRecords` 메서드를 제공합니다.

이 끌어오기 모델의 경우 코드는 스트림의 샤드에서 직접 데이터를 추출합니다. API를 사용하여 자체 소비자 애플리케이션을 작성하는 방법에 대한 자세한 내용은 Developing Custom Consumers with Shared Throughput Using the AWS SDK for Java(Java용 AWS SDK를 사용하여 공유 처리량으로 사용자 정의 소비자 개발) 단원을 참조하세요. API에 대한 자세한 내용은 Amazon Kinesis Data Streams API Reference(Amazon Kinesis Data StreamsAPI 참조)에서 확인할 수 있습니다.

AWS Lambda를 사용하여 데이터 스트림 처리

AWS Lambda를 사용하면 서버를 프로비저닝하거나 관리할 필요 없이 코드를 실행할 수 있습니다. Lambda를 사용하면 전혀 관리할 필요 없이 사실상 모든 유형의 애플리케이션 또는 백엔드 서비스에 대한 코드를 실행할 수 있습니다. 코드를 업로드하기만 하면 고가용성을 유지한 채로 코드를 실행하고 확장하는 데 필요한 모든 것을 Lambda가 알아서 처리해 줍니다. 코드가 다른 AWS 서비스에서 자동으로 트리거되도록 설정하거나 어떤 웹 또는 모바일 앱에서도 코드를 직접 호출할 수 있습니다.

AWS Lambda는 기본적으로 Amazon Kinesis Data Streams와 통합됩니다. 이 기본 통합을 사용하면 폴링, 체크포인트 및 오류 처리 복잡성이 개념화됩니다. 이를 통해 Lambda 함수 코드는 비즈니스 로직 처리에 집중할 수 있습니다.

Lambda 함수를 공유 처리량(표준 반복기) 또는 향상된 팬아웃 기능이 있는 전용 처리량 소비자에 매핑할 수 있습니다. 표준 반복기를 사용할 경우 Lambda는 HTTP 프로토콜을 사용하여 Kinesis 스트림

의 각 샤드를 폴링합니다. 대기 시간을 최소화하고 읽기 처리량을 최대화하기 위해 향상된 팬아웃으로 데이터 스트림 소비자를 생성할 수 있습니다. 이 아키텍처의 스트림 소비자는 동일한 스트림에서 읽는 다른 애플리케이션과 경쟁하지 않고도 각 샤드에 대한 전용 연결을 얻습니다. Amazon Kinesis Data Streams는 HTTP/2를 통해 레코드를 Lambda로 푸시합니다.

기본적으로 AWS Lambda는 스트림에서 레코드를 사용할 수 있게 되는 즉시 함수를 호출합니다. 배치 시나리오의 경우 레코드를 버퍼링하기 위해 이벤트 소스에서 최대 5분 동안 배치 창을 구현할 수 있습니다. 함수가 오류를 반환하면 Lambda는 처리가 성공할 때까지 또는 데이터가 만료될 때까지 배치를 재시도합니다.

요약

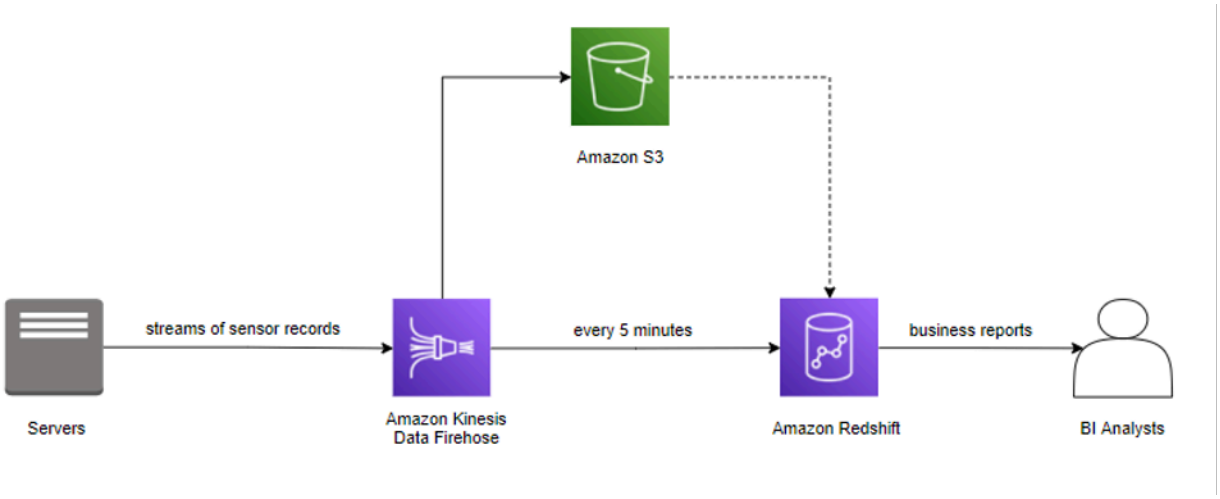
InternetProvider는 Amazon Kinesis Data Streams를 활용하여 사용자 세부 정보와 위치를 스트리밍했습니다. 레코드 스트림은 함수 라이브러리에 저장된 대역폭 옵션으로 데이터를 보강하기 위해 AWS Lambda에서 사용되었습니다. 보강 후 AWS Lambda는 대역폭 옵션을 애플리케이션에 다시 게시했습니다. InternetProvider는 Amazon Kinesis Data Streams와 AWS Lambda로 서버 프로비저닝 및 관리를 처리함으로써 비즈니스 애플리케이션 개발에 더 집중할 수 있었습니다.

시나리오 2: 보안 팀을 위한 실시간에 가까운 데이터

ABC2Badge는 기업이나 AWS re:Invent와 같은 대규모 이벤트를 위해 센서와 배지를 제공하는 회사입니다. 사용자가 이벤트에 등록하면 캠퍼스 전체에서 센서가 감지하는 고유한 배지를 받습니다. 사용자가 센서를 통과하면 익명화된 정보가 관계형 데이터베이스에 기록됩니다.

예정된 이벤트의 참석자 수가 많기 때문에 이벤트 보안 팀은 ABC2Badge에 15분마다 캠퍼스에서 사람이 가장 많이 모이는 영역에 대한 데이터를 수집하도록 요청했습니다. 이를 통해 보안 팀은 상황에 대응하고 밀집도에 따라 보안 인력을 분산시킬 충분한 시간을 확보할 수 있습니다. 보안 팀의 새로운 이 요구 사항과 거의 실시간으로 날짜를 처리하는 스트리밍 솔루션을 구축한 경험이 부족하다는 점을 감안하여 ABC2Badge는 간단하면서도 확장 가능하고 신뢰할 수 있는 솔루션을 찾고 있습니다.

이 회사가 현재 사용하는 데이터 웨어하우스 솔루션은 Amazon Redshift입니다. ABC2Badge는 Amazon Kinesis 서비스의 기능을 검토하면서 Amazon Kinesis Data Firehose가 데이터 레코드 스트림을 수신하고 버퍼 크기 또는 시간 간격에 따라 레코드를 배치 처리한 후 Amazon Redshift에 삽입할 수 있음을 알게 되었습니다. 이에 따라 Kinesis Data Firehose 전송 스트림을 생성하고 5분마다 Amazon Redshift 테이블에 데이터를 복사하도록 구성했습니다. 이 새로운 솔루션의 일환으로 서버에서 Amazon Kinesis 에이전트를 사용했습니다. Kinesis Data Firehose는 5분마다 Amazon Redshift로 데이터를 로드합니다. 그러면 비즈니스 인텔리전스(BI) 팀이 분석을 수행하고 15분마다 보안 팀에 데이터를 전송할 수 있습니다.



Amazon Kinesis Data Firehose를 사용한 새로운 솔루션

Amazon Kinesis Data Firehose

Amazon Kinesis Data Firehose는 스트리밍 데이터를 AWS로 로드하는 가장 쉬운 방법입니다. Amazon Kinesis Data Firehose는 스트리밍 데이터를 캡처 및 변환하고 Amazon Kinesis Data Analytics, Amazon Simple Storage Service(Amazon S3), Amazon Redshift, Amazon OpenSearch Service(OpenSearch Service) 및 Splunk로 로드할 수 있습니다. 또한 Kinesis Data Firehose는 스트리밍 데이터를 사용자 정의 HTTP 엔드포인트 또는 지원되는 서드 파티 서비스 공급자가 소유한 HTTP 엔드포인트로 로드할 수 있습니다.

Kinesis Data Firehose는 이미 사용 중인 기존 비즈니스 인텔리전스 도구 및 대시보드를 통해 실시간에 가까운 분석을 지원합니다. 또한 완전관리형 서버리스 서비스이므로 데이터 처리량에 대응하여 자동으로 확장되며 지속적인 관리가 필요 없습니다. Kinesis Data Firehose는 데이터를 로드하기 전에 배치, 압축 및 암호화하여 대상 스토리지의 사용량을 최소화하고 보안을 강화할 수 있습니다. AWS Lambda를 사용하여 소스 데이터를 변환하고 변환된 데이터를 대상으로 전달할 수도 있습니다. 데이터를 Kinesis Data Firehose에 전송하도록 데이터 생산자를 구성하면 Amazon Kinesis Data Firehose에서 자동으로 데이터를 사용자가 지정한 대상으로 전송합니다.

Firehose 전송 스트림으로 데이터 전송

전송 스트림으로 데이터를 전송하기 위한 몇 가지 옵션이 있습니다. AWS는 널리 사용되는 여러 프로그래밍 언어에 대한 SDK를 제공하며, 각 SDK는 Amazon Kinesis Data Firehose용 API를 제공합니다. AWS에는 전송 스트림으로 데이터를 전송하는 데 사용할 수 있는 유틸리티가 있습니다. Kinesis Data Firehose는 다른 AWS 서비스와 통합되어 해당 서비스에서 전송 스트림으로 직접 데이터를 전송합니다.

Amazon Kinesis 에이전트 사용

Amazon Kinesis 에이전트는 전송 스트림으로 전송될 새 데이터에 대한 로그 파일 세트를 지속적으로 모니터링하는 독립형 소프트웨어 애플리케이션입니다. 에이전트는 파일 교체, 체크포인트 수행, 실패 시 재시도를 자동으로 처리하고 전송 스트림에 대한 모니터링 및 문제 해결을 위해 Amazon CloudWatch 지표를 내보냅니다. 데이터 사전 처리, 여러 파일 디렉터리 모니터링 및 여러 전송 스트림에 쓰기 등의 추가 구성을 에이전트에 적용할 수 있습니다.

에이전트는 Linux 또는 Windows 기반 서버(예: 웹 서버, 로그 서버 및 데이터베이스 서버)에 설치할 수 있습니다. 에이전트를 설치했으면 모니터링할 로그 파일과 에이전트가 전송할 전송 스트림을 지정하기만 하면 됩니다. 에이전트는 새 데이터를 전송 스트림으로 지속적이고 안정적으로 전송합니다.

AWS SDK 및 AWS 서비스와 함께 API를 소스로 사용

Kinesis Data Firehose API는 전송 스트림으로 데이터를 전송하기 위한 두 가지 작업을 제공합니다. `PutRecord`는 호출 한 번으로 데이터 레코드 하나를 전송합니다. `PutRecordBatch`는 호출 한 번으로 여러 데이터 레코드를 전송할 수 있으며 생산자당 더 높은 처리량을 달성할 수 있습니다. 이 메서드를 사용할 때는 각 메서드에서 전송 스트림의 이름과 데이터 레코드 또는 데이터 레코드 배열을 지정해야 합니다. Kinesis Data Firehose API 작업에 대한 자세한 내용과 샘플 코드는 Writing to a Firehose Delivery Stream Using the AWS SDK(AWS SDK를 사용하여 Firehose 전송 스트림에 쓰기) 단원을 참조하세요.

Kinesis Data Firehose는 Kinesis Data Firehose, CloudWatch Logs, CloudWatch Events, Amazon Simple Notification Service(Amazon SNS), Amazon API Gateway 및 AWS IoT와도 함께 실행됩니다. 확장 가능하고 안정적으로 데이터 스트림, 로그, 이벤트 및 IoT 데이터를 Kinesis Data Firehose 대상으로 직접 전송할 수 있습니다.

대상으로 전송되기 전에 데이터 처리

경우에 따라 스트리밍 데이터가 대상으로 전송되기 전에 스트리밍 데이터를 변환하거나 보완할 수 있습니다. 예를 들어 데이터 생산자가 각 데이터 레코드에서 비정형 텍스트를 보낼 수 있으며, 이러한 데이터는 OpenSearch Service로 전송하기 전에 JSON으로 변환해야 합니다. 또는 JSON 데이터를 Amazon S3에 저장하기 전에 Apache Parquet 또는 Apache ORC와 같은 열 형식 파일로 변환할 수도 있습니다.

Kinesis Data Firehose에는 데이터 형식 변환 기능이 내장되어 있습니다. 이를 통해 JSON 데이터 스트림을 Apache Parquet 또는 Apache ORC 파일 형식으로 쉽게 변환할 수 있습니다.

데이터 변환 흐름

스트리밍 데이터 변환을 지원하기 위해 Kinesis Data Firehose는 사용자가 생성한 Lambda 함수를 사용하여 데이터를 변환합니다. Kinesis Data Firehose는 수신 데이터를 함수에 대해 지정된 버퍼 크기로 버퍼링한 다음 지정된 Lambda 함수를 비동기식으로 호출합니다. 변환된 데이터는 Lambda에서 Kinesis Data Firehose로 전송되고, Kinesis Data Firehose는 해당 데이터를 대상으로 전송합니다.

데이터 형식 변환

Kinesis Data Firehose 데이터 형식 변환을 사용하여 JSON 데이터 스트림을 Apache Parquet 또는 Apache ORC로 변환할 수도 있습니다. 이 기능은 JSON만 Apache Parquet 또는 Apache ORC로 변환할 수 있습니다. 데이터가 CSV 형식인 경우 Lambda 함수를 통해 해당 데이터를 JSON으로 변환한 다음 데이터 형식 변환을 적용할 수 있습니다.

데이터 전송

실시간에 가까운 전송 스트림인 Kinesis Data Firehose는 수신 데이터를 버퍼링합니다. 전송 스트림의 버퍼링 임계값에 도달하면 데이터가 구성한 대상으로 전송됩니다. Kinesis Data Firehose가 각 대상으로 데이터를 전송하는 방법에는 몇 가지 차이가 있으며 이 백서의 다음 단원에서 이러한 차이를 검토합니다.

Amazon S3

Amazon S3은 간단한 웹 서비스 인터페이스를 통해 웹 어디서나 원하는 양의 데이터를 저장 및 검색할 수 있는 객체 스토리지입니다. 99.999999999%의 내구성을 제공하며, 전 세계적으로 수조 이상의 객체로 확장할 수 있도록 설계되었습니다.

Amazon S3으로 데이터 전송

Amazon S3으로 데이터를 전송하는 경우 Kinesis Data Firehose는 전송 스트림의 버퍼링 구성에 따라 수신 레코드 여러 개를 연결한 다음 이를 S3 객체로 Amazon S3에 전송합니다. S3으로의 데이터 전송 빈도는 S3 버퍼 크기(1MB~128MB) 또는 버퍼 간격(60초~900초) 중 먼저 도달하는 값에 따라 결정됩니다.

S3 버킷으로의 데이터 전송은 여러 가지 이유로 실패할 수 있습니다. 예를 들어 버킷이 더 이상 존재하지 않거나 Kinesis Data Firehose에서 수임하는 AWS Identity and Access Management(IAM) 역할에 버킷에 대한 액세스 권한이 없을 수 있습니다. 이러한 경우 Kinesis Data Firehose는 전송에 성공할 때까지 최대 24시간 동안 계속 다시 시도합니다. Kinesis Data Firehose의 최대 데이터 저장 시간은 24시간입니다. 24시간 넘게 데이터 전송에 실패할 경우 데이터가 손실됩니다.

Amazon Redshift

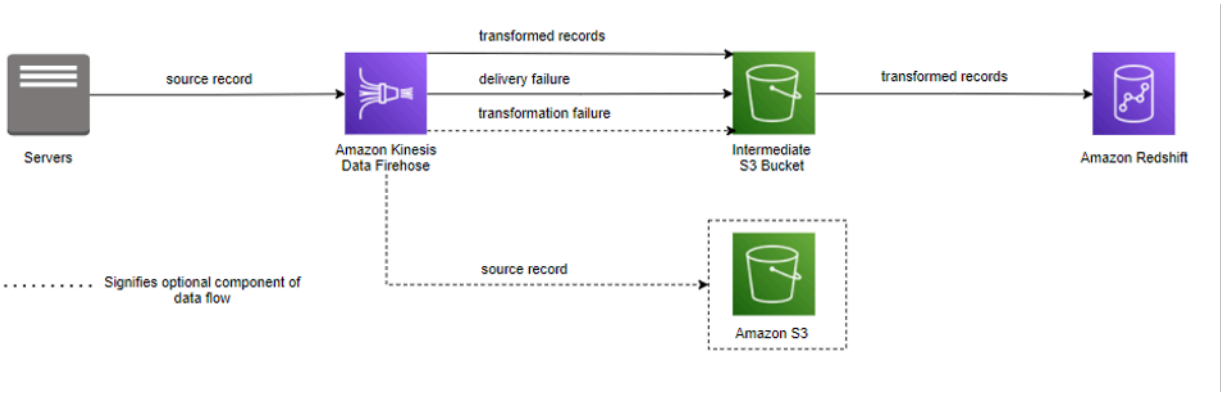
Amazon Redshift는 속도가 빠른 완전관리형 데이터 웨어하우스로, 표준 SQL 및 기존 BI 도구를 사용하여 모든 데이터를 간편하고 비용 효율적으로 분석할 수 있습니다. Amazon Redshift를 사용하면 정교한 쿼리 최적화, 고성능 로컬 디스크의 열 형식 스토리지, 대량 병렬 쿼리 실행 기능을 사용하여 페타바이트 규모의 정형 데이터에 대해 복잡한 분석 쿼리를 실행할 수 있습니다.

Amazon Redshift로 데이터 전송

Amazon Redshift로 데이터를 전송하는 경우 Kinesis Data Firehose는 먼저 앞서 설명한 형식으로 수신 데이터를 S3 버킷으로 전송합니다. 그런 다음 Kinesis Data Firehose는 Amazon Redshift COPY 명령을 실행하여 S3 버킷에서 Amazon Redshift 클러스터로 데이터를 로드합니다.

S3에서 Amazon Redshift로의 데이터 COPY 작업의 빈도는 Amazon Redshift 클러스터가 COPY 명령을 완료하는 속도에 따라 다릅니다. Amazon Redshift 대상의 경우 데이터 전송 실패를 처리하기 위해 전송 스트림을 생성할 때 재시도 시간(0초~7,200초)을 지정할 수 있습니다. Kinesis Data Firehose는 지정된 시간 동안 재시도하고 실패한 경우 특정 S3 객체 배치를 건너뜁니다. 건너뛴 객체의 정보는 errors/ 폴더의 매니페스트 파일로 S3 버킷으로 전송되며, 이를 수동 백필에 사용할 수 있습니다.

다음은 Kinesis Data Firehose에서 Amazon Redshift로의 데이터 흐름에 대한 아키텍처 다이어그램입니다. 이 데이터 흐름은 Amazon Redshift에 고유하지만 Kinesis Data Firehose는 다른 대상의 경우에도 유사한 패턴을 따릅니다.



Kinesis Data Firehose에서 Amazon Redshift로의 데이터 흐름

Amazon OpenSearch Service(OpenSearch Service)

OpenSearch Service는 OpenSearch의 사용하기 쉬운 API 및 실시간 기능과 더불어 프로덕션 워크로드에 필요한 가용성, 확장성 및 보안을 제공하는 완전관리형 서비스입니다. OpenSearch Service는 로그 분석, 전체 텍스트 검색 및 애플리케이션 모니터링을 위해 OpenSearch를 쉽게 배포, 운영 및 확장할 수 있게 해 줍니다.

OpenSearch Service로 데이터 전송

OpenSearch Service로 데이터를 전송하기 위해 Kinesis Data Firehose는 전송 스트림의 버퍼링 구성에 따라 수신 레코드를 버퍼링한 후 OpenSearch 클러스터로 여러 레코드를 인덱싱하기 위한 OpenSearch 대량 요청을 생성합니다. OpenSearch Service로의 데이터 전송 빈도는 OpenSearch 버퍼 크기(1MB~100MB) 또는 버퍼 간격(60초~900초) 중 먼저 도달하는 값에 따라 결정됩니다.

OpenSearch Service 대상에 대해 전송 스트림 생성 시 재시도 시간(0초~7,200초)을 지정할 수 있습니다. Kinesis Data Firehose는 지정된 시간 동안 재시도한 다음 이 특정 인덱스 요청을 건너뜁니다. 건너뛴 문서는 `elasticsearch_failed/` 폴더로 S3 버킷으로 전송되며, 이를 수동 백필에 사용할 수 있습니다.

Amazon Kinesis Data Firehose는 시간을 기준으로 OpenSearch Service 인덱스를 교체할 수 있습니다. 선택한 교체 옵션(`NoRotation`, `OneHour`, `OneDay`, `OneWeek` 또는 `OneMonth`)에 따라 Kinesis Data Firehose는 UTC(협정 세계 표준시) 도착 타임스탬프의 일부를 지정된 인덱스 이름에 추가합니다.

사용자 정의 HTTP 엔드포인트 또는 지원되는 서드 파티 서비스 공급자

Kinesis Data Firehose는 사용자 정의 HTTP 엔드포인트 또는 Datadog, Dynatrace, LogicMonitor, MongoDB, New Relic, Splunk, Sumo Logic과 같은 지원되는 서드 파티 공급자로 데이터를 전송할 수 있습니다.

사용자 정의 HTTP 엔드포인트 또는 지원되는 서드 파티 서비스 공급자

Kinesis Data Firehose가 사용자 정의 HTTP 엔드포인트에 데이터를 성공적으로 전송하려면 이러한 엔드포인트가 요청을 수락하고 특정 Kinesis Data Firehose 요청 및 응답 형식을 사용하여 응답을 전송해야 합니다.

지원되는 서드 파티 서비스 공급자가 소유한 HTTP 엔드포인트에 데이터를 전송할 때 통합 AWS Lambda 서비스를 사용하여 수신 레코드를 서비스 공급자의 통합에서 요구하는 형식과 일치하는 형식으로 변환하는 함수를 생성할 수 있습니다.

데이터 전송 빈도의 경우 서비스 공급자마다 권장 버퍼 크기가 있습니다. 권장 버퍼 크기에 대한 자세한 내용은 서비스 공급자에게 문의하세요. 데이터 전송 실패를 처리하기 위해 Kinesis Data Firehose는 먼저 대상으로부터의 응답을 기다린 후 HTTP 엔드포인트와의 연결을 설정합니다. Kinesis Data Firehose는 재시도 기간이 만료될 때까지 계속 연결을 시도합니다. 그 이후에는 Kinesis Data Firehose가 이를 전송 실패로 간주하고 데이터를 S3 버킷에 백업합니다.

요약

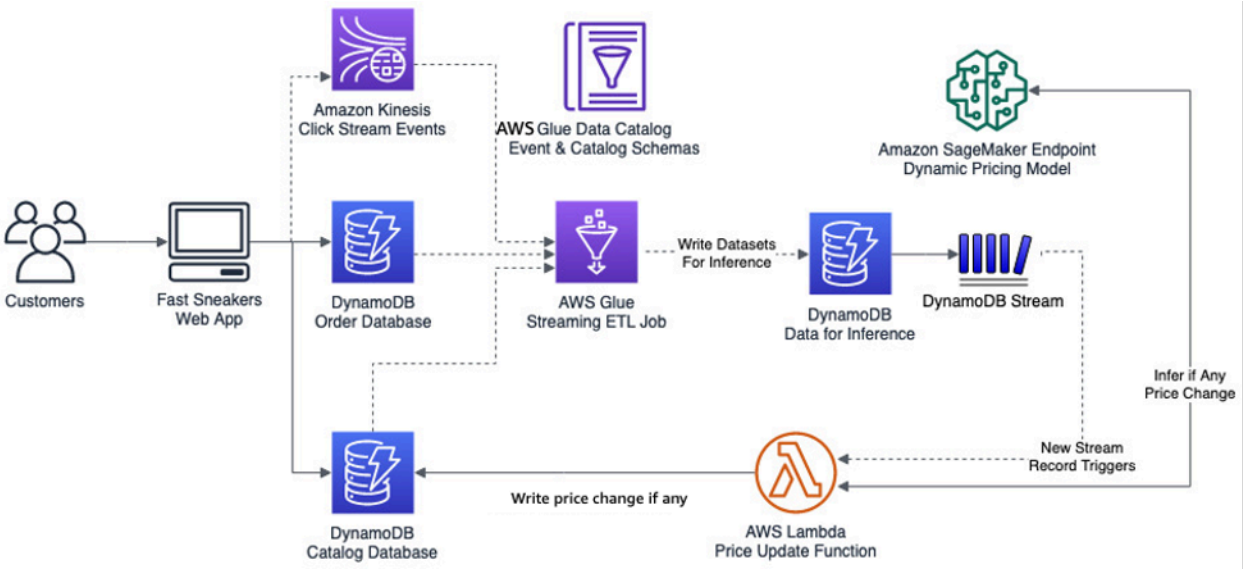
Kinesis Data Firehose는 스트리밍 데이터를 지원되는 대상으로 지속적으로 전송할 수 있습니다. 완전관리형 솔루션이므로 개발이 거의 또는 전혀 필요하지 않습니다. ABC2Badge의 경우 Kinesis Data Firehose가 자연스러운 선택이었습니다. 이 회사는 이미 Amazon Redshift를 데이터 웨어하우스 솔루션으로 사용하고 있었습니다. 데이터 원본이 트랜잭션 로그에 지속적으로 기록되기 때문에 Amazon Kinesis 에이전트를 활용하여 추가 코드를 작성하지 않고도 데이터를 스트리밍할 수 있었습니다. 이제 ABC2Badge는 센서 레코드 스트림을 생성하고 Kinesis Data Firehose를 통해 이러한 레코드를 수신하고 있으므로 이를 보안 팀 사용 사례의 기반으로 사용할 수 있습니다.

시나리오 3: 데이터 인사이트 프로세스를 위한 클릭스트림 데이터 준비

Fast Sneakers는 트렌디한 스니커즈를 주력 상품으로 하는 패션 부티크입니다. 재고나 트렌드(예: 지난 밤 TV에서 유명 브랜드 스니커즈를 착용한 유명인이나 스포츠 스타가 출현한 경우)에 따라 해당 신발 한 켤레의 가격이 오르거나 내릴 수 있습니다. 이러한 트렌드를 추적하고 분석하여 수익을 극대화하는 것이 Fast Sneakers에게는 중요합니다.

Fast Sneakers는 유지 관리가 필요한 새로운 인프라로 인해 프로젝트에 추가 오버헤드가 발생하는 것을 원하지 않습니다. 또한 개발을 적합한 담당자로 분할하여, 데이터 엔지니어는 데이터 변환에 집중하고 데이터 사이언티스트는 기계 학습 기능에 대한 작업을 독립적으로 할 수 있기를 바랍니다.

수요에 따라 신속하게 대응하고 자동으로 가격을 조정하기 위해 Fast Sneakers는 중요한 이벤트(예: 관심 항목 클릭 및 구매 데이터)를 스트리밍하여 이벤트 데이터를 변환 및 보강하고 기계 학습 모델에 제공합니다. 이 업체의 기계 학습 모델은 가격 조정이 필요한지 여부를 결정할 수 있습니다. 이를 통해 Fast Sneakers는 가격을 자동으로 조정하여 제품의 수익을 극대화할 수 있습니다.



Fast Sneakers 실시간 가격 조정

이 아키텍처 다이어그램은 Fast Sneakers가 Kinesis Data Streams, AWS Glue 및 DynamoDB Streams를 사용하여 생성한 실시간 스트리밍 솔루션을 보여 줍니다. Fast Sneakers는 이러한 서비스를 활용하여 지원 인프라를 설정하고 유지 관리하는 데 시간을 소비하지 않고도 탄력적이고 안정적인 솔루션을 구축할 수 있습니다. 스트리밍 추출, 변환 및 로드(ETL) 작업과 기계 학습 모델에 집중하여 가치를 창출하는 데 시간을 할애할 수 있습니다.

다음은 이 업체의 워크로드에 사용되는 아키텍처 및 기술에 대한 이해를 돕는 사용된 서비스에 대한 몇 가지 세부 정보입니다.

AWS Glue 및 AWS Glue 스트리밍

AWS Glue는 데이터를 카탈로그화하고, 정리하고, 보강하고, 데이터 스토어 간에 안정적으로 이동하는 데 사용할 수 있는 완전관리형 ETL 서비스입니다. AWS Glue를 사용하면 ETL 작업 생성에 따르는 비용, 복잡성 및 시간을 크게 줄일 수 있습니다. AWS Glue는 서버리스이므로 설정하거나 관리할 인프라가 없습니다. 작업이 실행되는 동안 사용한 리소스에 대한 비용만 지불하면 됩니다.

AWS Glue를 활용하여 AWS Glue 스트리밍 ETL 작업으로 소비자 애플리케이션을 생성할 수 있습니다. 이를 통해 Apache Spark 및 기타 Spark 기반 모듈 쓰기를 활용하여 이벤트 데이터를 사용하고 처리할 수 있습니다. 이 문서의 다음 단원에서는 이 시나리오에 대해 자세히 설명합니다.

AWS Glue Data Catalog

AWS Glue Data Catalog에는 AWS Glue에서 ETL 작업의 소스 및 대상으로 사용되는 데이터에 대한 참조가 포함되어 있습니다. AWS Glue Data Catalog는 데이터의 위치, 스키마 및 런타임 지표에 대한

인덱스입니다. ETL 작업을 생성하고 모니터링하는 데 데이터 카탈로그의 정보를 사용할 수 있습니다. 데이터 카탈로그의 정보는 메타데이터 테이블로 저장되며 여기서 각 테이블은 단일 데이터 스토어를 지정합니다. 크롤러를 설정하면 DynamoDB, S3 및 JDBC(Java Database Connectivity) 연결 스토어를 비롯한 다양한 유형의 데이터 스토어를 자동으로 평가하고 메타데이터와 스키마를 추출한 다음 AWS Glue Data Catalog에서 테이블 정의를 생성할 수 있습니다.

AWS Glue 스트리밍 ETL 작업에서 Amazon Kinesis Data Streams를 사용하려면 AWS Glue Data Catalog 데이터베이스의 테이블에 스트림을 정의하는 것이 좋습니다. 지원되는 여러 형식(CSV, JSON, ORC, Parquet, Avro 또는 Grok을 사용한 고객 형식) 중 하나인 Kinesis 스트림을 사용하여 스트림 소스 테이블을 정의합니다. 스키마를 직접 입력하거나 이 단계를 AWS Glue 작업에 맡겨 작업 런타임 중에 결정할 수 있습니다.

AWS Glue 스트리밍 ETL 작업

AWS Glue는 Apache Spark 서버리스 환경에서 ETL 작업을 실행합니다. AWS Glue는 이런 작업을 가상 리소스에서 실행하여 자체 서비스 계정을 프로비저닝하고 관리합니다. Apache Spark 기반 작업을 실행할 수 있을 뿐만 아니라 AWS Glue는 DynamicFrames를 사용하여 Spark를 기반으로 더 높은 수준의 기능을 제공합니다.

DynamicFrames는 구조체 및 배열과 같은 중첩 데이터를 지원하는 분산 테이블입니다. 각 레코드는 자기 설명적이며 반정형 데이터가 있는 스키마 유연성을 위해 설계되었습니다. `DynamicFrame`의 레코드에는 데이터와 데이터를 설명하는 스키마가 모두 포함됩니다. Apache Spark `DataFrames`와 `DynamicFrames`는 모두 ETL 스크립트에서 지원되며 서로 변환할 수 있습니다. `DynamicFrames`는 데이터 정리 및 ETL을 위한 일련의 고급 변환을 제공합니다.

AWS Glue 작업에서 Spark Streaming을 사용하면 지속적으로 실행되는 스트리밍 ETL 작업을 생성하고 Amazon Kinesis Data Streams, Apache Kafka 및 Amazon MSK와 같은 스트리밍 소스의 데이터를 사용할 수 있습니다. 이 작업은 데이터를 정리, 병합 및 변환한 다음 Amazon S3, Amazon DynamoDB 또는 JDBC 데이터 스토어를 포함한 스토어로 결과를 로드할 수 있습니다.

기본적으로 AWS Glue는 100초 동안 데이터를 처리하고 작성합니다. 이를 통해 데이터를 효율적으로 처리할 수 있으며 예상보다 늦게 도착하는 데이터에 대해 집계를 수행할 수 있습니다. 응답 속도와 집계의 정확도를 적절하게 조정하여 기간을 구성할 수 있습니다. AWS Glue 스트리밍 작업은 체크포인트를 사용하여 Kinesis Data Streams에서 읽은 데이터를 추적합니다. AWS Glue에서 스트리밍 ETL 작업을 생성하는 방법에 대한 연습은 AWS Glue에서 스트리밍 ETL 작업 추가를 참조하세요.

Amazon DynamoDB

Amazon DynamoDB는 모든 규모에서 10밀리초 미만의 성능을 제공하는 키–값 및 문서 데이터베이스입니다. 완전관리형의 내구성이 뛰어난 다중 리전, 다중 활성 데이터베이스로서, 인터넷 규모 애플리케

이션을 위한 보안, 백업 및 복원, 인 메모리 캐싱 기능을 기본적으로 제공합니다. DynamoDB는 하루에 10조 개가 넘는 요청을 처리할 수 있으며 초당 2천만 개 이상의 요청까지 지원할 수 있습니다.

DynamoDB 스트림에 대한 변경 데이터 캡처

DynamoDB 스트림은 DynamoDB 테이블 항목의 변경에 대한 정렬된 정보 흐름입니다. 테이블에서 스트림을 활성화하면 DynamoDB가 테이블의 데이터 항목에 발생한 모든 변경 정보를 캡처합니다. DynamoDB는 DynamoDB 스트림의 이벤트에 자동으로 응답하는 코드 조각인 트리거를 만들 수 있도록 AWS Lambda에서 실행됩니다. 트리거를 사용하면 DynamoDB 테이블의 데이터 수정에 응답하는 애플리케이션을 구축할 수 있습니다.

테이블에서 스트림을 활성화할 경우 스트림 Amazon 리소스 이름(ARN)을 사용자가 작성한 Lambda 함수와 연결할 수 있습니다. 테이블의 항목이 수정되는 즉시 새로운 레코드가 테이블의 스트림에 표시됩니다. AWS Lambda는 새로운 스트림 레코드가 감지될 때마다 스트림을 폴링하고 Lambda 함수를 동기식으로 호출합니다.

Amazon SageMaker 및 Amazon SageMaker 서비스 엔드포인트

Amazon SageMaker는 개발자와 데이터 사이언티스트가 어떤 규모에서도 기계 학습 모델을 빠르게 구축, 훈련 및 배포할 수 있도록 지원하는 완전관리형 플랫폼입니다. SageMaker에는 함께 사용하거나 개별적으로 사용하여 기계 학습 모델을 구축, 훈련 및 배포할 수 있는 모듈이 포함되어 있습니다. Amazon SageMaker 서비스 엔드포인트를 사용하면 Amazon SageMaker 내부 또는 외부에서 개발한 배포된 모델을 통해 실시간 추론을 위한 관리형 호스팅 엔드포인트를 생성할 수 있습니다.

AWS SDK를 사용하면 콘텐츠와 함께 콘텐츠 유형 정보를 전달하는 SageMaker 엔드포인트를 호출한 다음 전달된 데이터를 기반으로 실시간 예측을 수신할 수 있습니다. 이를 통해 추론된 결과에 대한 작업을 수행하는 코드와 기계 학습 모델의 설계 및 개발을 분리할 수 있습니다.

이렇게 하면 데이터 사이언티스트가 기계 학습에 집중할 수 있고 기계 학습 모델을 사용하는 개발자는 코드에서 기계 학습을 사용하는 방법에 집중할 수 있습니다. SageMaker에서 엔드포인트를 호출하는 방법에 대한 자세한 내용은 InvokeEnpoint in the Amazon SageMaker API Reference(Amazon SageMaker API 참조의 InvokeEnpoint)를 참조하세요.

실시간으로 데이터 인사이트 추론

위의 아키텍처 다이어그램은 Fast Sneakers의 기존 웹 애플리케이션이 웹 사이트의 트래픽 및 이벤트 데이터를 제공하는 클릭스트림 이벤트가 포함된 Kinesis 데이터 스트림을 추가했음을 보여 줍니다. 분류, 제품 속성, 가격 등의 정보가 포함된 제품 카탈로그와 주문 품목, 결제, 배송 등의 데이터가 있는 주문 테이블은 별도의 DynamoDB 테이블입니다. 데이터 스트림 소스와 적절한 DynamoDB 테이블에는

AWS Glue 스트리밍 ETL 작업에서 사용할 수 있도록 AWS Glue Data Catalog에 정의된 메타데이터와 스키마가 있습니다.

Fast Sneakers는 Apache Spark, Spark Streaming 및 `DynamicFrames`를 AWS Glue 스트리밍 ETL 작업에 활용하여 데이터 스트림에서 데이터를 추출하고 변환함으로써 제품 및 주문 테이블의 데이터를 병합할 수 있습니다. 변환에서 하이드레이션된 데이터를 사용하면 추론 결과를 가져올 데이터 집합이 DynamoDB 테이블로 제출됩니다.

테이블에 대한 DynamoDB 스트림은 새로 작성된 각 레코드에 대해 Lambda 함수를 트리거합니다. Lambda 함수는 이전에 변환된 레코드를 AWS SDK와 함께 SageMaker 엔드포인트에 제출하여 제품에 필요한 가격 조정(있는 경우)을 추론합니다. 가격 조정이 필요하다고 기계 학습 모델이 식별하면 Lambda 함수는 카탈로그 DynamoDB 테이블의 제품에 가격 변동을 기록합니다.

요약

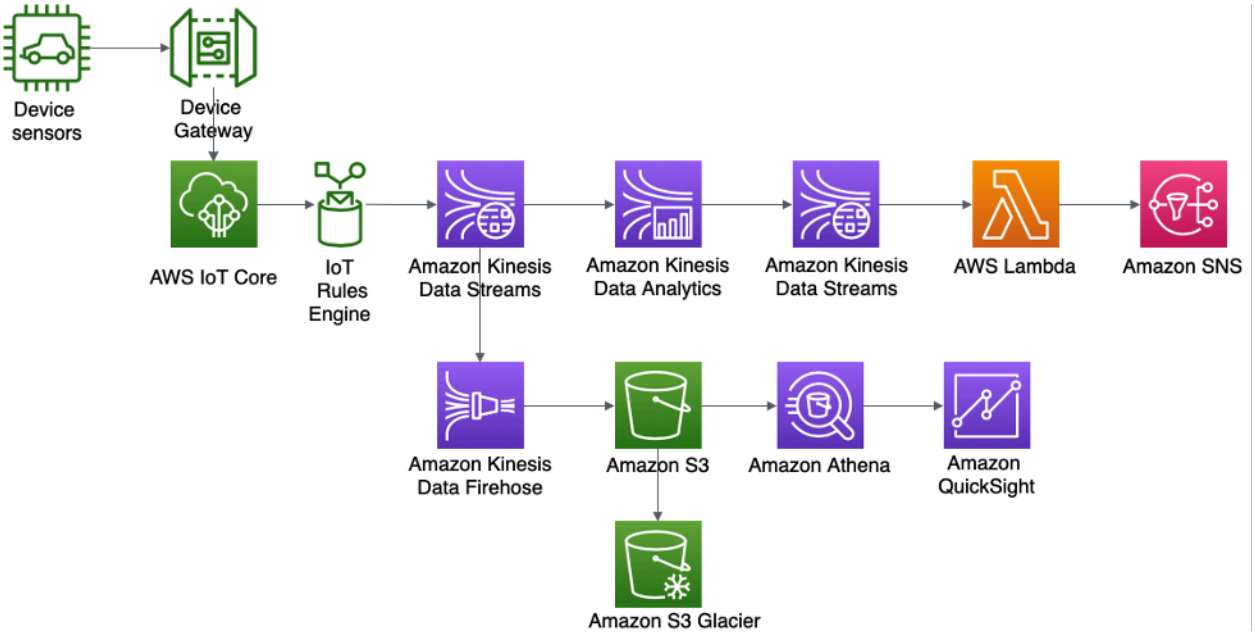
Amazon Kinesis Data Streams를 사용하면 실시간 스트리밍 데이터를 손쉽게 수집, 처리 및 분석할 수 있으므로 적시에 인사이트를 확보하고 새로운 정보에 신속하게 대응할 수 있습니다. AWS Glue 서버리스 데이터 통합 서비스와 함께 사용하면 기계 학습용 데이터를 준비하고 결합하는 실시간 이벤트 스트리밍 애플리케이션을 만들 수 있습니다.

Kinesis Data Streams와 AWS Glue 서비스 모두 완전관리형 서비스이므로, AWS는 빅 데이터 플랫폼을 위한 인프라 관리의 획일적인 부담을 덜어주고 사용자는 데이터를 기반으로 데이터 인사이트를 생성하는 데 집중할 수 있습니다.

Fast Sneakers는 실시간 이벤트 처리 및 기계 학습을 활용하여 웹 사이트에서 완전 자동화된 실시간 가격 조정을 수행하여 제품 재고를 최적화할 수 있습니다. 이를 통해 빅 데이터 플랫폼을 만들고 유지 관리하지 않고도 비즈니스 가치를 극대화할 수 있습니다.

시나리오 4: 디바이스 센서 실시간 이상 탐지 및 알림

ABC4Logistics는 가솔린, 액체 프로판(LPG), 나프타와 같은 인화성이 높은 석유 제품을 항구에서 여러 도시로 운송하는 회사입니다. 이 회사는 위치, 엔진 온도, 컨테이너 내부 온도, 주행 속도, 주차 위치, 도로 상태 등을 모니터링하기 위해 여러 개의 센서가 설치된 수백 대의 차량을 보유하고 있습니다. ABC4Logistics의 요구 사항 중 하나는 엔진과 컨테이너의 온도를 실시간으로 모니터링하고 이상이 발생할 경우 운전자와 차량 모니터링 팀에 경고하는 것입니다. 이러한 상황을 탐지하고 실시간으로 알림을 생성하기 위해 ABC4Logistics는 AWS를 기반으로 다음 아키텍처를 구현했습니다.



ABC4Logistics의 디바이스 센서 실시간 이상 탐지 및 알림 아키텍처

디바이스 센서의 데이터는 AWS IoT 게이트웨이에 의해 수집되며, 여기서AWS IoT 규칙 엔진은 스트리밍 데이터를 Amazon Kinesis Data Streams에서 사용할 수 있도록 합니다. ABC4Logistics는 Kinesis Data Analytics를 사용하여 Kinesis Data Streams의 스트리밍 데이터에 대한 실시간 분석을 수행할 수 있습니다.

ABC4Logistics는 Kinesis Data Analytics를 사용하여 센서의 온도 판독값이 10초 동안 정상 판독값에서 벗어나면 이를 탐지하고 레코드를 다른 Kinesis Data Streams 인스턴스에 수집하여 비정상적인 레코드를 식별할 수 있습니다. 그런 다음 Amazon Kinesis Data Streams가 Lambda 함수를 호출하여 Amazon SNS를 통해 운전자와 차량 모니터링 팀에 알림을 보낼 수 있습니다.

또한 Kinesis Data Streams의 데이터가 Amazon Kinesis Data Firehose로 푸시됩니다. Amazon Kinesis Data Firehose는 이 데이터를 Amazon S3에 보관하여 ABC4Logistics가 센서 데이터에 대해 배치 또는 실시간에 가까운 분석을 수행할 수 있도록 합니다. ABC4Logistics는 Amazon Athena를 사용하여 S3의 데이터를 쿼리하고 시각화를 위해 Amazon QuickSight를 사용합니다. 장기 데이터 보존을 위해 S3 수명 주기 정책은 Amazon S3 Glacier에 데이터를 아카이브하는 데 사용됩니다.

이 아키텍처의 중요한 구성 요소는 아래에 자세히 설명되어 있습니다.

Amazon Kinesis Data Analytics

Amazon Kinesis Data Analytics를 사용하면 스트리밍 데이터를 변환 및 분석하고 실시간으로 이상 현상에 대응할 수 있습니다. AWS 기반 서버리스 서비스인 Kinesis Data Analytics는 프로비저닝을 처리

하고 모든 데이터 처리량을 감당할 수 있도록 인프라를 탄력적으로 조정합니다. 이렇게 하면 스트리밍 인프라를 설정하고 관리하는 획일적인 부담이 모두 사라지고, 스트리밍 애플리케이션을 작성하는 데 더 많은 시간을 할애할 수 있습니다.

Amazon Kinesis Data Analytics를 사용하면 표준 SQL과 Java, Python 및 Scala의 Apache Flink 애플리케이션 등 다양한 옵션을 사용하여 스트리밍 데이터를 대화식으로 쿼리하고 Java로 Apache Beam 애플리케이션을 구축하여 데이터 스트림을 분석할 수 있습니다.

이러한 옵션은 스트리밍 애플리케이션 및 소스/대상 지원의 복잡성 수준에 따라 특정 접근 방식을 유연하게 사용할 수 있도록 합니다. 다음 단원에서는 Flink 애플리케이션용 Kinesis Data Analytics 옵션에 대해 설명합니다.

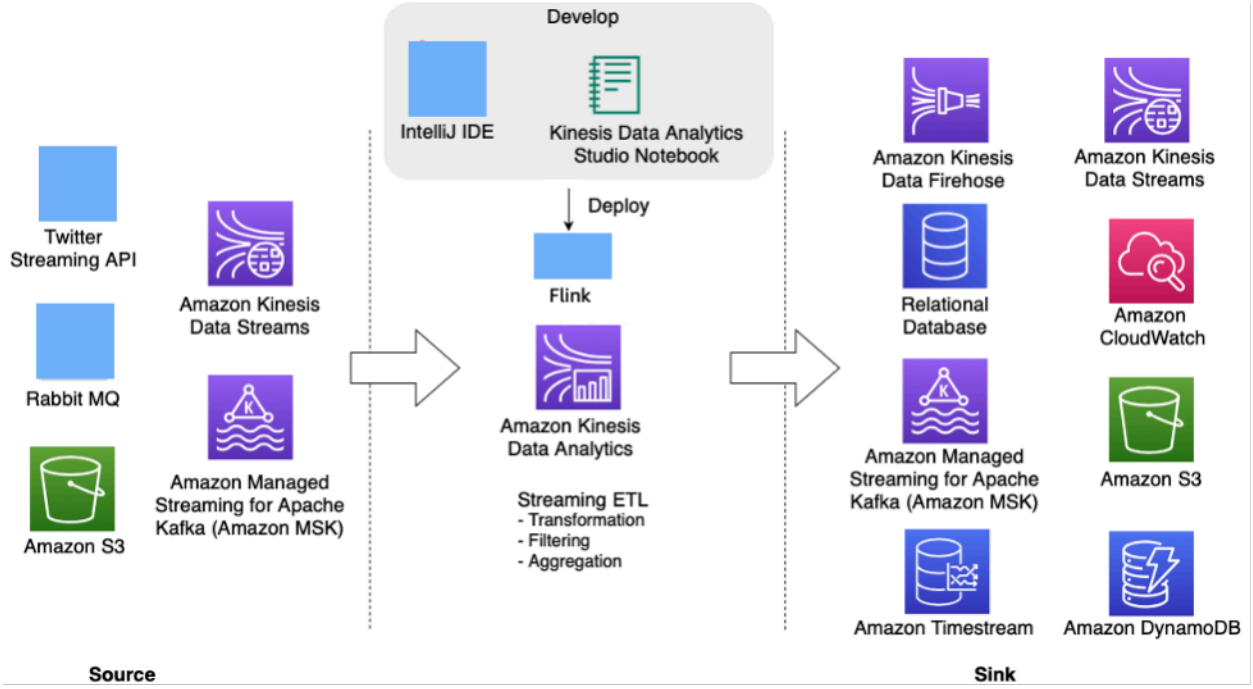
Apache Flink 애플리케이션용 Amazon Kinesis Data Analytics

Apache Flink는 무제한 및 제한된 데이터 스트림의 상태 유지 계산을 위한 인기 있는 오픈 소스 프레임워크 및 분산 처리 엔진입니다. Apache Flink는 정확히 하나의 의미 체계를 지원하여 인 메모리 속도와 대규모로 계산을 수행하도록 설계되었습니다. Apache Flink를 기반으로 하는 애플리케이션은 내결함성 방식으로 높은 처리량으로 짧은 대기 시간을 달성할 수 있도록 지원합니다.

Apache Flink용 Amazon Kinesis Data Analytics를 사용하면 복잡한 분산 Apache Flink 환경을 관리하지 않고도 스트리밍 소스에 대해 코드를 작성 및 실행하여 시계열 분석을 수행하고, 실시간 대시보드를 제공하며, 실시간 지표를 생성할 수 있습니다. Flink 인프라를 직접 호스팅할 때와 동일한 방식으로 높은 수준의 Flink 프로그래밍 기능을 사용할 수 있습니다.

Apache Flink용 Kinesis Data Analytics를 사용하면 Java, Scala, Python 또는 SQL로 애플리케이션을 생성하여 스트리밍 데이터를 처리하고 분석할 수 있습니다. 일반적인 Flink 애플리케이션은 입력 스트림이나 데이터 위치 또는 소스에서 데이터를 읽고, 연산자 또는 함수를 사용하여 데이터를 변환/필터링하거나 결합하며, 출력 스트림이나 데이터 위치 또는 싱크에 데이터를 저장합니다.

다음 아키텍처 다이어그램은 Kinesis Data Analytics Flink 애플리케이션에 지원되는 소스 및 싱크 중 일부를 보여 줍니다. 소스/싱크에 대해 미리 번들로 제공되는 커넥터 외에도 Kinesis Data Analytics의 Flink 애플리케이션을 위한 다양한 기타 소스/싱크로 사용자 정의 커넥터를 가져올 수도 있습니다.



실시간 스트림 처리를 위한 Kinesis Data Analytics의 Apache Flink 애플리케이션

개발자는 선호하는 IDE를 사용하여 Flink 애플리케이션을 개발하고 AWS Management Console 또는 DevOps 도구에서 Kinesis Data Analytics에 배포할 수 있습니다.

Amazon Kinesis Data Analytics Studio

Kinesis Data Analytics 서비스의 일부인 Kinesis Data Analytics Studio를 사용하면 고객이 실시간으로 데이터 스트림을 대화식으로 쿼리하고 SQL, Python 및 Scala를 사용하여 스트림 처리 애플리케이션을 쉽게 구축 및 실행할 수 있습니다. Studio 노트북은 Apache Zeppelin으로 구동됩니다.

Studio 노트북을 사용하면 노트북 환경에서 Flink 애플리케이션 코드를 개발하고, 코드 결과를 실시간으로 확인하며, 노트북 내에서 시각화할 수 있습니다. Kinesis Data Streams 및 Amazon MSK 콘솔에서 클릭 한 번으로 Apache Zeppelin 및 Apache Flink로 구동되는 Studio 노트북을 생성하거나 Kinesis Data Analytics 콘솔에서 실행할 수 있습니다.

Kinesis Data Analytics Studio의 일부로 반복적으로 코드를 개발하면 노트북을 Kinesis Data Analytics 애플리케이션으로 배포하여 스트리밍 모드에서 지속적으로 실행하고, 소스에서 데이터를 읽고, 대상에 쓰고, 장기 실행 애플리케이션의 상태를 유지 관리하며, 소스 스트림의 처리량에 따라 크기 조정이 자동으로 수행됩니다. 이전에는 고객이 AWS에서 실시간 스트리밍 데이터를 대화식으로 분석하기 위해 SQL 애플리케이션용 Kinesis Data Analytics를 사용했습니다.

SQL 애플리케이션용 Kinesis Data Analytics도 계속 사용할 수 있지만 새 프로젝트의 경우 새로운 Kinesis Data Analytics Studio를 사용하는 것이 좋습니다. Kinesis Data Analytics Studio에서는 고급 분석 기능을 손쉽게 사용하여 정교한 스트림 처리 애플리케이션을 몇 분 안에 구축할 수 있습니다.

Kinesis Data Analytics Flink 애플리케이션의 내결함성을 설정하려면 Implementing Fault Tolerance in Kinesis Data Analytics for Apache Flink(Apache Flink용 Kinesis Data Analytics에서 내결함성 구현)에 설명된 대로 체크포인트 및 스냅샷을 사용하면 됩니다.

Kinesis Data Analytics Flink 애플리케이션은 데이터 처리, 체크포인트 기능 및 데이터 원본(예: Kinesis Data Streams, Kinesis Data Firehose, Amazon MSK, Rabbit MQ 및 사용자 정의 커넥터를 포함한 Apache Cassandra)의 데이터 처리에 대해 정확히 하나의 의미 체계를 사용하는 애플리케이션과 같은 복잡한 스트리밍 분석 애플리케이션을 작성하는 데 유용합니다.

Flink 애플리케이션에서 스트리밍 데이터를 처리한 후에는 Amazon Kinesis Data Streams, Amazon Kinesis Data Firehose, Amazon DynamoDB, Amazon OpenSearch Service, Amazon Timestream, Amazon S3 등과 같은 다양한 싱크 또는 대상에 데이터를 유지할 수 있습니다. Kinesis Data Analytics Flink 애플리케이션은 1초 미만의 성능을 보장합니다.

Kinesis Data Analytics용 Apache Beam 애플리케이션

Apache Beam은 스트리밍 데이터를 처리하기 위한 프로그래밍 모델입니다. Apache Beam은 Flink, Spark Streaming, Apache Samza 등의 다양한 엔진 또는 실행기에서 실행될 수 있는 정교한 데이터 병렬 처리 파이프라인을 구축하기 위한 이식 가능한 API 계층을 제공합니다.

Kinesis Data Analytics 애플리케이션과 함께 Apache Beam 프레임워크를 사용하여 스트리밍 데이터를 처리할 수 있습니다. Apache Beam을 사용하는 Kinesis Data Analytics 애플리케이션은 Apache Flink 실행기를 사용하여 Beam 파이프라인을 실행합니다.

요약

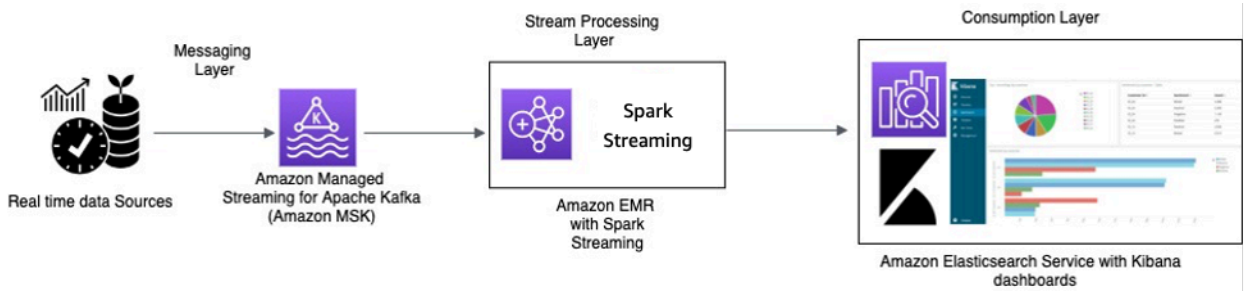
AWS 스트리밍 서비스인 Amazon Kinesis Data Streams, Amazon Kinesis Data Analytics 및 Amazon Kinesis Data Firehose를 사용하여

ABC4Logistics는 온도 판독 값의 이상 패턴을 탐지하고 운전자와 차량 관리 팀에 실시간으로 통보하여 차량 전파나 화재와 같은 주요 사고를 예방할 수 있습니다.

시나리오 5: Apache Kafka를 사용한 실시간 원격 측정 데이터 모니터링

ABC1Cabs는 온라인 택시 예약 서비스 회사입니다. 모든 택시에는 차량에서 원격 측정 데이터를 수집하는 IoT 디바이스가 있습니다. 현재 ABC1Cabs는 실시간 이벤트 사용을 위해 설계된 Apache Kafka 클러스터를 실행하고 있으며 시스템 상태 지표를 수집하고, 활동을 추적하며, 온프레미스 Hadoop 클러스터에 구축된 Apache Spark Streaming 플랫폼에 데이터를 공급합니다.

ABC1Cabs는 비즈니스 지표, 디버깅, 알림 및 기타 대시보드 생성을 위해 OpenSearch Dashboards를 사용합니다. 이 회사는 Amazon MSK, Spark Streaming을 지원하는 Amazon EMR 및 OpenSearch Dashboards가 있는 OpenSearch Service에 관심이 있습니다. 이 회사의 요구 사항은 Apache Kafka 및 Hadoop 클러스터 유지 관리 오버헤드를 줄이는 동시에 익숙한 오픈 소스 소프트웨어와 API를 사용하여 데이터 파이프라인을 오케스트레이션하는 것입니다. 다음 아키텍처 다이어그램은 AWS 기반 솔루션을 보여 줍니다.



Amazon MSK, Amazon EMR 기반 Apache Spark Streaming을 사용한 스트림 처리 및 OpenSearch Dashboards가 있는 Amazon OpenSearch Service를 통한 실시간 처리

택시 IoT 디바이스는 원격 측정 데이터를 수집하여 소스 허브로 전송합니다. 소스 허브는 Amazon MSK에 실시간으로 데이터를 전송하도록 구성되어 있습니다. Amazon MSK는 Apache Kafka 생산자 라이브러리 API를 사용하여 데이터를 Amazon EMR 클러스터로 스트리밍하도록 구성됩니다. Amazon EMR 클러스터에는 데이터 스트림을 사용하고 처리할 수 있도록 Kafka 클라이언트와 Spark Streaming이 설치되어 있습니다.

Spark Streaming에는 Elasticsearch의 정의된 인덱스에 직접 데이터를 쓸 수 있는 싱크 커넥터가 있습니다. OpenSearch Dashboards가 있는 Elasticsearch 클러스터를 지표 및 대시보드에 사용할 수 있습니다. Amazon MSK, Spark Streaming을 지원하는 Amazon EMR 및 OpenSearch Dashboards가 있는 OpenSearch Service는 모두 관리형 서비스로, AWS가 다양한 클러스터의 인프라 관리라는 획일적인 부담을 덜어주므로 몇 번의 클릭만으로 친숙한 오픈 소스 소프트웨어를 사용하여 애플리케이션을 구축할 수 있습니다. 다음 단원에서는 이러한 서비스에 대해 자세히 살펴보겠습니다.

Amazon Managed Streaming for Apache Kafka(Amazon MSK)

Apache Kafka는 고객이 클릭스트림 이벤트, 트랜잭션, IoT 이벤트, 애플리케이션 및 시스템 로그와 같은 스트리밍 데이터를 캡처할 수 있도록 지원하는 오픈 소스 플랫폼입니다. 이 정보를 사용하여 실시간 분석을 수행하고 지속적인 변환을 실행하며 이 데이터를 데이터 레이크 및 데이터베이스에 실시간으로 배포하는 애플리케이션을 개발할 수 있습니다.

Kafka를 스트리밍 데이터 스토어로 사용하여 생산자와 소비자로부터 애플리케이션을 분리하고 두 구성 요소 간에 안정적으로 데이터를 전송할 수 있습니다. Kafka는 널리 사용되는 엔터프라이즈 데이터 스트리밍 및 메시징 플랫폼이지만 프로덕션 환경에서 설정, 크기 조정 및 관리하기가 어려울 수 있습니다.

Amazon MSK는 이러한 관리 작업을 처리하고 고가용성 및 보안을 위한 모범 사례에 따라 Apache Zookeeper와 함께 Kafka를 쉽게 설정, 구성 및 실행할 수 있도록 지원합니다. 여전히 Kafka의 제어 영역 운영 및 데이터 영역 운영을 사용하여 데이터 생산 및 소비를 관리할 수 있습니다.

Amazon MSK는 오픈 소스 Apache Kafka를 실행하고 관리하기 때문에 고객은 애플리케이션 코드를 변경할 필요 없이 AWS에서 기존 Apache Kafka 애플리케이션을 손쉽게 마이그레이션하고 실행할 수 있습니다.

크기 조정

Amazon MSK는 사용자가 클러스터를 실행하는 동안 적극적으로 크기를 조정할 수 있도록 크기 조정 작업을 제공합니다. Amazon MSK 클러스터를 생성하는 경우 클러스터 시작 시 브로커의 인스턴스 유형을 지정할 수 있습니다. Amazon MSK 클러스터 내에서 몇 개의 브로커로 시작할 수 있습니다. 그런 다음 AWS Management Console 또는 AWS CLI를 사용하여 클러스터당 수백 개의 브로커로 확장할 수 있습니다.

Apache Kafka 브로커의 크기 또는 패밀리를 변경하여 클러스터 크기를 조정할 수도 있습니다. 브로커 크기 또는 패밀리를 변경하면 워크로드의 변화에 맞게 Amazon MSK 클러스터의 컴퓨팅 용량을 유연하게 조정할 수 있습니다. Amazon MSK 크기 조정 및 요금 스프레드시트(파일 다운로드)를 사용하여 Amazon MSK 클러스터에 적합한 브로커 수를 결정합니다. 이 스프레드시트는 유사한 자체 관리형 EC2 기반 Apache Kafka 클러스터와 비교한 Amazon MSK 클러스터의 관련 비용과 크기 조정 추정치를 제공합니다.

Amazon MSK 클러스터를 생성한 후 스토리지를 줄이는 경우를 제외하고 브로커당 EBS 스토리지의 양을 늘릴 수 있습니다. 스토리지 볼륨은 이 확장 작업 중에 계속 사용할 수 있습니다. 자동 크기 조정과 수동 크기 조정이라는 두 가지 유형의 크기 조정 작업을 제공합니다.

Amazon MSK는 애플리케이션 자동 크기 조정 정책을 사용하여 사용량 증가에 대응하여 클러스터 스토리지의 자동 확장을 지원합니다. 자동 크기 조정 정책은 대상 디스크 사용률과 최대 크기 조정 용량을 설정합니다.

스토리지 사용률 임계값은 Amazon MSK가 자동 크기 조정 작업을 트리거하는 데 도움이 됩니다. 수동 크기 조정을 사용하여 스토리지를 늘리려면 클러스터가 `ACTIVE` 상태가 될 때까지 기다립니다. 스토리지 크기 조정의 휴지 기간은 이벤트 간에 최소 6시간입니다. 작업을 통해 추가 스토리지를 즉시 사용할 수 있지만 서비스는 클러스터에서 최대 24시간 이상 걸릴 수 있는 최적화를 수행합니다.

이러한 최적화 시간은 스토리지 크기에 비례합니다. 또한 AWS 리전 내에서 다중 가용 영역 복제를 제공하여 고가용성을 제공합니다.

구성

Amazon MSK는 브로커, 주제 및 Apache Zookeeper 노드의 기본 구성을 제공합니다. 또한 사용자 정의 구성을 생성하고, 이를 사용해 새 Amazon MSK 클러스터를 생성하거나 기존 클러스터를 업데이트할 수 있습니다. 사용자 정의 Amazon MSK 구성을 지정하지 않고 MSK 클러스터를 생성하면 Amazon MSK가 기본 구성을 생성하여 사용합니다. 이러한 기본값 목록은 Apache Kafka Configuration(Apache Kafka 구성)을 참조하세요.

모니터링을 위해 Amazon MSK는 Apache Kafka 지표를 수집하여 Amazon CloudWatch로 전송하고 사용자가 확인할 수 있습니다. MSK 클러스터에 대해 구성한 지표는 자동으로 수집되어 CloudWatch에 푸시됩니다. 소비자 지연을 모니터링하면 주제에서 사용 가능한 최신 데이터를 따라가지 못하는 느리거나 멈춘 소비자를 식별할 수 있습니다. 그런 다음 필요한 경우 해당 소비자에 대해 크기 조정 또는 재부팅과 같은 수정 조치를 취할 수 있습니다.

Amazon MSK로 마이그레이션

다음 방법 중 하나를 사용하여 온프레미스에서 Amazon MSK로 마이그레이션할 수 있습니다.

- MirrorMaker2.0 - MirrorMaker2.0(MM2)은 Apache Kafka Connect 프레임워크를 기반으로 하는 다중 클러스터 데이터 복제 엔진입니다. MM2는 Apache Kafka 소스 커넥터와 싱크 커넥터의 조합입니다. MM2 클러스터 하나를 사용하여 여러 클러스터 간에 데이터를 마이그레이션할 수 있습니다. MM2는 새 주제와 파티션을 자동으로 검색하는 동시에 주제 구성이 클러스터 간에 동기화되도록 합니다. MM2는 마이그레이션 ACL, 주제 구성 및 오프셋 변환을 지원합니다. 마이그레이션과 관련된 자세한 내용은 Migrating Clusters Using Apache Kafka's MirrorMaker(Apache Kafka의 MirrorMaker를 사용하여 클러스터 마이그레이션)를 참조하세요. MM2는 주제 구성 복제 및 오프셋 변환과 관련된 사용 사례에 자동으로 사용됩니다.

- Apache Flink - MM2는 최소 한 번 의미 체계를 지원합니다. 레코드는 대상에 복제될 수 있으며 소비자는 중복 레코드를 처리하기 위해 멱등성이 있어야 합니다. 정확히 한 번 시나리오에서는 고객이 Apache Flink를 사용할 수 있는 의미 체계가 필요합니다. MM2는 정확히 한 번 의미 체계를 달성할 수 있는 대안을 제공합니다.

  Apache Flink는 대상 클러스터에 제출하기 전에 데이터에 매핑하거나 변환하는 작업이 필요한 시나리오에도 사용할 수 있습니다. Apache Flink는 하나의 Apache Kafka 클러스터에서 데이터를 읽고 다른 클러스터에 쓸 수 있는 소스 및 싱크와 함께 Apache Kafka용 커넥터를 제공합니다. Apache Flink는 Amazon EMR 클러스터를 시작하거나 Amazon Kinesis Data Analytics를 통해 Apache Flink를 애플리케이션으로 실행하여 AWS에서 실행할 수 있습니다.
- AWS Lambda - AWS Lambda에 대한 이벤트 소스로 Apache Kafka를 지원하므로 이제 고객이 Lambda 함수를 통해 주제의 메시지를 사용할 수 있습니다. AWS Lambda 서비스는 이벤트 소스에서 새 레코드나 메시지를 내부적으로 폴링한 다음 대상 Lambda 함수를 동기적으로 호출하여 이러한 메시지를 사용합니다. Lambda는 배치로 메시지를 읽고 처리를 위해 이벤트 페이로드의 함수에 메시지 배치를 제공합니다. 그런 다음 사용된 메시지를 대상 Amazon MSK 클러스터로 변환하거나 직접 쓸 수 있습니다.

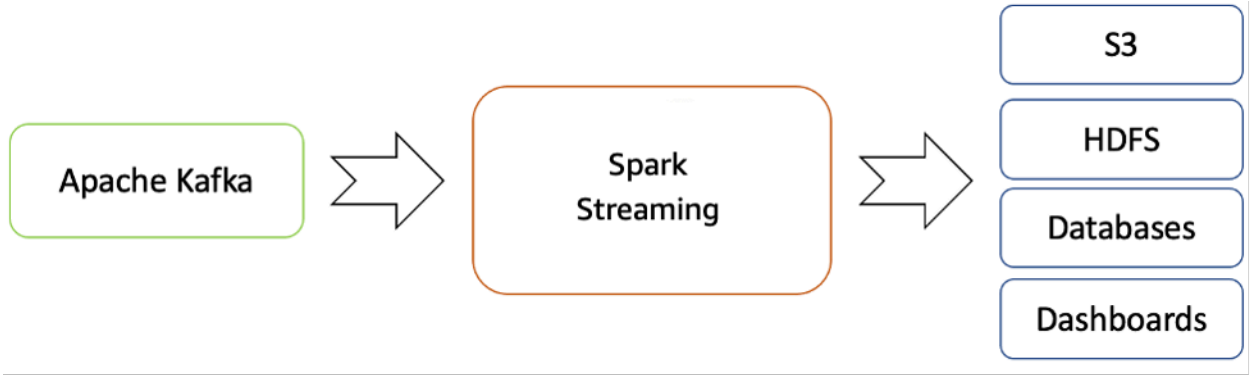
Spark Streaming을 지원하는 Amazon EMR

Amazon EMR은 방대한 양의 데이터를 처리 및 분석하기 위해 Apache Hadoop 및 Apache Spark와 같은 빅 데이터 프레임워크 실행을 간소화하는 관리형 클러스터 플랫폼입니다.

Amazon EMR은 Spark의 기능을 제공하며 Spark Streaming을 시작하여 Kafka의 데이터를 사용할 수 있습니다. Spark Streaming은 라이브 데이터 스트림의 확장 가능하고 처리량이 높으며 내결함성을 갖춘 스트림 처리를 가능하게 하는 핵심 Spark API의 확장형입니다.

AWS Command Line Interface(AWS CLI)를 사용하거나 AWS Management Console에서 Amazon EMR 클러스터를 생성할 수 있으며, 클러스터를 생성하는 동안 고급 구성에서 Spark 및 Zeppelin을 선택할 수 있습니다. 다음 아키텍처 다이어그램에서 볼 수 있듯이 Apache Kafka 및 Kinesis Data Streams와 같은 다양한 소스에서 데이터를 수집할 수 있으며 맵, 축소, 결합 및 창과 같은 상위 수준 함수로 표현된 복잡한 알고리즘을 사용하여 데이터를 처리할 수 있습니다. 자세한 내용은 Transformations on DStreams(DSreams에서의 변환)를 참조하세요.

처리된 데이터는 파일 시스템, 데이터베이스 및 라이브 대시보드로 푸시할 수 있습니다.



Apache Kafka에서 Hadoop 에코시스템으로의 실시간 스트리밍 흐름

기본적으로 Apache Spark Streaming에는 마이크로 배치 실행 모델이 있습니다. 그러나 Spark 2.3이 출시된 이후 Apache는 연속 처리라는 새로운 낮은 대기 시간 처리 모드를 도입했습니다. 이 모드는 최소 한 번 보장으로 1밀리초의 낮은 엔드 투 엔드 대기 시간을 달성할 수 있습니다.

쿼리에서 DataSet/DataFrame 작업을 변경하지 않고도 애플리케이션 요구 사항에 따라 모드를 선택할 수 있습니다. Spark Streaming의 몇 가지 이점은 다음과 같습니다.

- Apache Spark의 언어 통합 API를 스트림 처리에 제공하므로 배치 작업을 작성하는 것과 동일한 방식으로 스트리밍 작업을 작성할 수 있습니다.
- Java, Scala, Python을 지원합니다.
- 사용자가 추가 코드를 작성할 필요 없이 손실된 작업과 작업자 상태(예: 슬라이딩 창)를 모두 복구할 수 있습니다.
- Spark에서 Spark Streaming을 실행하면 배치 처리에 동일한 코드를 재사용하거나, 기록 데이터에 스트림을 결합하거나, 스트림 상태에 대해 임시 쿼리를 실행하고, 분석뿐만 아니라 강력한 대화형 애플리케이션을 구축할 수 있습니다.
- Spark Streaming으로 데이터 스트림을 처리한 후 OpenSearch Sink Connector를 사용하여 OpenSearch Service 클러스터에 데이터를 쓸 수 있으며 OpenSearch Dashboards를 소비 계층으로 사용할 수 있습니다.

OpenSearch Dashboards가 있는 Amazon OpenSearch Service

OpenSearch Service는 AWS 클라우드에서 OpenSearch 클러스터를 쉽게 배포, 운영 및 크기 조정할 수 있는 관리형 서비스입니다. OpenSearch는 로그 분석, 실시간 애플리케이션 모니터링, 클릭스트림 분석 같은 사용 사례를 위한 인기 있는 오픈 소스 검색 및 분석 엔진입니다.

OpenSearch Dashboards는 로그와 시계열 분석, 애플리케이션 모니터링, 운영 인텔리전스 사용 사례에 사용되는 오픈 소스 데이터 시각화 및 탐색 도구입니다. 히스토그램, 선형 그래프, 원형 차트, 열 지도, 내장 지리 공간적 지원과 같은 강력하고 사용하기 쉬운 기능을 제공합니다.

OpenSearch Dashboards는 널리 사용되는 분석 및 검색 엔진인 OpenSearch와의 통합을 제공하므로 OpenSearch Dashboards가 OpenSearch에 저장된 데이터를 시각화하기 위한 기본 선택이 됩니다. OpenSearch Service는 모든 OpenSearch Service 도메인에 OpenSearch Dashboards를 설치할 수 있도록 합니다. OpenSearch Service 콘솔의 도메인 대시보드에서 OpenSearch Dashboards에 대한 링크를 찾을 수 있습니다.

요약

AWS에서 관리형 서비스로 제공되는 Apache Kafka를 사용하면 브로커 간의 조정을 관리하는 대신 사용에 집중할 수 있습니다. 이를 위해서는 일반적으로 Apache Kafka에 대한 깊은 이해가 필요합니다. 고가용성, 브로커 확장성, 세분화된 액세스 제어와 같은 기능은 Amazon MSK 플랫폼에서 관리합니다.

ABC1Cabs는 이러한 서비스를 활용하여 인프라 관리 전문 지식 없이도 프로덕션 애플리케이션을 구축했습니다. Amazon MSK의 데이터를 소비하고 시각화 계층으로 전파하기 위해 처리 계층에 집중할 수 있었습니다.

Amazon EMR의 Spark Streaming은 스트리밍 데이터를 실시간으로 분석하고 시각화 계층에 대한 Amazon OpenSearch Service에 OpenSearch Dashboards를 게시하는 데 도움이 됩니다.

결론 및 기여자

결론

이 문서에서는 스트리밍 워크플로에 대한 몇 가지 시나리오를 검토했습니다. 이러한 시나리오에서 예로 나온 회사는 스트리밍 데이터 처리를 통해 새로운 기능을 추가할 수 있었습니다.

데이터가 생성될 때 데이터를 분석하면 현재 비즈니스가 무엇을 하고 있는지에 대한 인사이트를 얻을 수 있습니다. AWS 스트리밍 서비스를 사용하면 인프라를 배포하고 관리하는 대신 애플리케이션에 집중하여 시간에 민감한 비즈니스 결정을 쉽게 내릴 수 있습니다.

기여자

- Amalia Rabinovitch, AWS 선임 솔루션스 아키텍트
- Priyanka Chaudhary, AWS 데이터 레이크 부문 데이터 아키텍트
- Zohair Nasimi, AWS 솔루션스 아키텍트
- Rob Kuhr, AWS 솔루션스 아키텍트
- Ejaz Sayyed, AWS 선임 파트너 솔루션스 아키텍트
- Allan MacInnis, AWS 솔루션스 아키텍트
- Chander Matrubhutam, AWS 제품 마케팅 관리자

문서 개정

이 백서의 업데이트에 대한 알림을 받으려면 RSS 피드를 구독하세요.

update-history-change	update-history-description	update-history-date
업데이트	기술적 정확성을 위해 업데이트되었습니다.	2021년 9월 1일
최초 게시	처음 게시된 백서	2017년 7월 1일