Whitepaper zu AWS

Streamen von Datenlösungen in AWS mit Amazon Kinesis

Streamen von Datenlösungen in AWS mit Amazon Kinesis: Whitepaper zu AWS

Table of Contents

Lösungen für Streaming-Daten in AWS

Datum der Veröffentlichung: 1. September 2021 (Am Dokument vorgenommene Änderungen)

Überblick

Dateningenieure, Datenanalysten und Big-Data-Entwickler wollen ihre Analysen von einer Batch-Verarbeitung zu einer Verarbeitung in Echtzeit weiterentwickeln, damit ihre Unternehmen erfahren können, welche Aktivitäten ihre Kunden, Anwendungen und Produkte aktuell vornehmen, und umgehend reagieren können. In diesem Whitepaper wird die Entwicklung der Analytik von Batch zu Echtzeit erörtert. Es wird beschrieben, wie Services wie Amazon Kinesis Data Streams, Amazon Kinesis Data Firehose, Amazon EMR, Amazon Kinesis Data Analytics, Amazon Managed Streaming for Apache Kafka (Amazon MSK) und andere für die Implementierung von Echtzeitanwendungen verwendet werden können, und es werden allgemeine Gestaltungsmuster für diese Services bereitgestellt.

Einführung

Aufgrund der explosionsartigen Zunahme von Datenquellen, die kontinuierlich Datenströme erzeugen, erhalten Unternehmen heute Daten in großem Umfang und mit hoher Geschwindigkeit. Ob Protokolldaten von Anwendungsservern, Klickstrom-Daten von Websites und mobile Apps oder Telemetriedaten von Internet-of-Things-(IoT)-Geräten – sie alle enthalten Informationen, die Sie dabei unterstützen zu erfahren, welche Aktivitäten Ihre Kunden, Anwendungen und Produkte aktuell vornehmen.

Die Fähigkeit, diese Daten in Echtzeit zu verarbeiten und zu analysieren, ist unerlässlich, um z. B. Ihre Anwendungen kontinuierlich zu überwachen, eine hohe Betriebszeit zu gewährleisten und Werbeangebote und Produktempfehlungen zu personalisieren. Die Echtzeit- und echtzeitnahe Verarbeitung kann auch andere gängige Anwendungsfälle wie Website-Analysen und Machine Learning präziser und besser umsetzbar machen, da die Daten für diese Anwendungen in Sekunden oder Minuten statt in Stunden oder Tagen zur Verfügung stehen.

Echtzeit- und echtzeitnahe Anwendungsszenarien

Sie können Streaming-Datendienste für Echtzeit- und echtzeitnahe Anwendungen wie Anwendungsüberwachung, Betrugserkennung und Live-Ranglisten nutzen. Echtzeit-Anwendungsfälle erfordern End-to-End-Latenzen von Millisekunden – von der Erfassung über die Verarbeitung bis hin zur Übermittlung der Ergebnisse an Zieldatenspeicher und andere Systeme. Netflix beispielsweise verwendet Amazon Kinesis Data Streams, um die Kommunikation zwischen all seinen Anwendungen zu überwachen, damit Probleme schnell erkannt und behoben werden und so den Kunden hohe Servicebetriebszeiten sowie eine hohe Verfügbarkeit geboten werden kann. Der am häufigsten anzutreffende Anwendungsfall ist die Überwachung der Anwendungsleistung, aber auch immer mehr Echtzeitanwendungen in den Bereichen Werbetechnik (Ad Tech), Gaming und IoT fallen in diese Kategorie.

Zu den gängigen echtzeitnahen Anwendungsfällen gehören Analysen von Datenspeichern für die Datenwissenschaft und das Machine Learning (ML). Sie können Streaming-Datenlösungen verwenden, um kontinuierlich Echtzeitdaten in Ihre Data Lakes zu laden. Sie können ML-Modelle dann häufiger aktualisieren, wenn neue Daten zur Verfügung stehen, und so die Genauigkeit und Zuverlässigkeit der Ausgaben sicherstellen. Zillow beispielsweise nutzt Kinesis Data Streams, um Daten aus öffentlichen Registern und MLS-Listen (Multiple Listing Service) zu sammeln und dann Käufern und Verkäufern von Häusern und Wohnungen die aktuellsten Schätzungen des Immobilienwerts nahezu in Echtzeit bereitzustellen. Das Unternehmen ZipRecruiter nutzt Amazon

MSK für seine Ereignisprotokollierungspipelines. Diese sind wichtige Infrastrukturkomponenten, die täglich über sechs Milliarden Ereignisse von der ZipRecruiter-Stellenbörse erfassen, speichern und kontinuierlich verarbeiten.

Unterschied zwischen Batch- und Stream-Verarbeitung

Für das Sammeln, Aufbereiten und Verarbeiten von Echtzeit-Streaming-Daten benötigen Sie andere Tools als die Tools, die Sie bislang für die Batch-Analyse verwendet haben. Bei der herkömmlichen Analyse sammeln Sie die Daten, laden sie regelmäßig in eine Datenbank und analysieren diese Stunden, Tage oder Wochen später. Die Analyse von Echtzeitdaten erfordert einen anderen Ansatz. Stream-Verarbeitungsanwendungen verarbeiten Daten kontinuierlich in Echtzeit, und zwar noch bevor sie gespeichert werden. Streaming-Daten können in rasantem Tempo eingehen, und das Datenvolumen kann jederzeit nach oben und unten schwanken. Plattformen für die Verarbeitung von Datenströmen müssen in der Lage sein, die Geschwindigkeit und Variabilität der eingehenden Daten zu bewältigen und sie zu verarbeiten, sobald sie eintreffen – oft sind dies Millionen bis Hunderte Millionen Ereignisse pro Stunde.

Herausforderungen bei der Streamverarbeitung

Durch die Verarbeitung von Echtzeitdaten bereits bei deren Eingang können Sie Entscheidungen viel schneller treffen, als dies mit herkömmlichen Technologien der Datenanalytik möglich ist. Der Aufbau und Betrieb eigener benutzerdefinierter Pipelines für Streaming-Daten ist jedoch kompliziert und ressourcenintensiv:

- Sie müssen ein System aufbauen, das Daten, die gleichzeitig aus Tausenden von Datenquellen stammen, kostengünstig erfassen, aufbereiten und übertragen kann.
- Für maximalen Durchsatz und niedrige Latenzzeiten müssen Sie die Speicher- und Rechenressourcen so abstimmen, dass die Daten effizient gebündelt und übertragen werden.
- Sie müssen eine Flotte von Servern bereitstellen und verwalten, um das System so zu skalieren, dass Sie die unterschiedlichen Datengeschwindigkeiten bewältigen können, die Sie ihm übergeben werden.

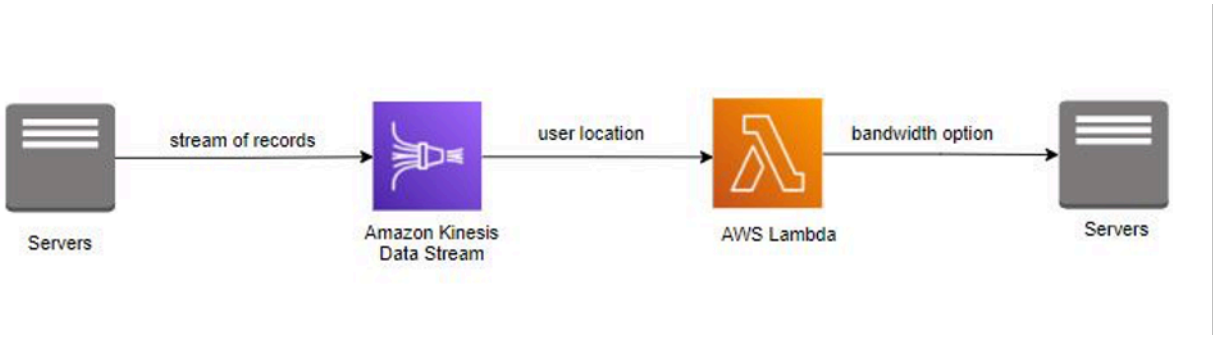
Das Versionsupgrade stellt einen komplexen und kostspieligen Prozess dar. Nachdem Sie diese Plattform aufgebaut haben, müssen Sie das System überwachen und nach Server- oder Netzwerkausfällen wiederherstellen, indem Sie die Datenverarbeitung an der richtigen Stelle im Sream nachholen, ohne dabei doppelte Daten zu erzeugen. Darüber hinaus benötigen Sie ein

spezielles Team für die Verwaltung der Infrastruktur. All dies kostet wertvolle Zeit und Geld. Letzten Endes gelangen die meisten Unternehmen nie ans Ziel, müssen sich mit dem Status Quo zufrieden geben und ihr Geschäft mit Informationen betreiben, die Stunden oder Tage alt sind.

Lösungen für Streaming-Daten: Beispiele

Szenario 1: Internetangebot basierend auf dem Standort

Das Unternehmen InternetProvider bietet Internetdienste mit einer Vielzahl von Bandbreitenoptionen für Benutzer auf der ganzen Welt. Wenn sich ein Benutzer für das Internet anmeldet, bietet das Unternehmen InternetProvider dem Benutzer je nach geografischem Standort verschiedene Bandbreitenoptionen an. Vor dem Hintergrund dieser Anforderungen implementierte das Unternehmen InternetProvider Amazon Kinesis Data Stream, um die Benutzer- und Standortdaten zu nutzen. Die Benutzer- und Standortdaten werden mit verschiedenen Bandbreitenoptionen angereichert, bevor sie an die Anwendung zurückgegeben werden. AWS Lambda ermöglicht diese Anreicherung in Echtzeit.



Verarbeitung von Datenströmen mit AWS Lambda

Amazon Kinesis Data Streams

Mit Amazon Kinesis Data Streams können Sie benutzerdefinierte Echtzeitanwendungen mit gängigen Frameworks für die Datenstromverarbeitung erstellen und Streaming-Daten in viele verschiedene Datenspeicher laden. Ein Kinesis-Stream kann so konfiguriert werden, dass er kontinuierlich Ereignisse von Hunderttausenden von Datenproduzenten empfängt, die aus Quellen wie Website-Clickstreams, IoT-Sensoren, Feeds in sozialen Medien und Anwendungsprotokollen stammen. Innerhalb von Millisekunden stehen die Daten zum Lesen und Verarbeiten durch Ihre Anwendung zur Verfügung.

Bei der Implementierung einer Lösung mit Kinesis Data Streams erstellen Sie benutzerdefinierte Datenverarbeitungsanwendungen, die als Kinesis-Data-Streams-Anwendungen bezeichnet werden. Eine typische Kinesis-Data-Streams-Anwendung liest Daten in Form von Datensätzen aus einem Kinesis-Stream.

Die in Kinesis-Data-Streams übertragenen Daten sind hochverfügbar und elastisch und stehen innerhalb von Millisekunden zur Verfügung. Sie können kontinuierlich verschiedene Datentypen wie Clickstreams, Anwendungsprotokolle und soziale Medien aus Hunderttausenden von Quellen einem Kinesis-Stream hinzufügen. Die Daten sind dann innerhalb von Sekunden für Ihre Kinesis-Anwendungen verfügbar und können aus dem Stream gelesen und dann verarbeitet werden.

Amazon Kinesis Data Streams ist ein vollständig verwalteter Daten-Streaming-Service. Er verwaltet Infrastruktur, Speicher, Vernetzung und Konfiguration, die zum Streamen Ihrer Daten mit dem von Ihrer Anwendung benötigten Durchsatzgrad erforderlich sind.

Senden von Daten an Amazon Kinesis Data Streams

Es gibt mehrere Möglichkeiten, Daten an Kinesis Data Streams zu senden, wodurch Sie bei der Gestaltung Ihrer Lösungen flexibel sind.

- Sie können Code mit einem der AWS SDKs schreiben, die von mehreren gängigen Sprachen unterstützt werden.
- Sie können den Amazon Kinesis Agent verwenden, ein Tool zum Senden von Daten an Kinesis Data Streams.

Die Amazon Kinesis Producer Library (KPL) vereinfacht die Entwicklung von Produzentenanwendungen, wodurch Entwickler einen hohen Schreibdurchsatz auf einen oder mehreren Kinesis-Datenströmen erzielen.

Die KPL ist eine einfach zu verwendende, hochgradig konfigurierbare Bibliothek, die Sie auf Ihren Hosts installieren. Sie dient als Vermittler zwischen dem Code Ihrer Produzentenanwendung und den Kinesis-Streams-API-Aktionen. Weitere Informationen über die KPL und ihre Fähigkeit, Ereignisse synchron und asynchron zu erzeugen, sowie Codebeispiele finden Sie unter Schreiben in Kinesis-Datenströme mithilfe der KPL

In der API von Kinesis Data Streams gibt es zwei verschiedene Vorgänge, mit denen Daten zu einem Datenstrom hinzugefügt werden: `PutRecords` und `PutRecord`. Mit dem `PutRecords`-Vorgang werden mehrere Datensätze pro HTTP-Anfrage an Ihren Stream gesendet, während mit `PutRecord` einen Datensatz pro HTTP-Anfrage übermittelt wird. Um einen höheren Durchsatz für die meisten Anwendungen zu erreichen, verwenden Sie `PutRecords`.

Weitere Informationen zu diesen APIs finden Sie unter Hinzufügen von Daten zu einem Stream. Die Details für jeden API-Vorgang finden Sie in der Referenz zur API von Amazon Kinesis Data Streams.

Verarbeitung von Daten in Amazon Kinesis Data Streams

Um Daten aus Kinesis-Streams zu lesen und zu verarbeiten, müssen Sie eine Verbraucheranwendung erstellen. Es gibt verschiedene Möglichkeiten, Verbraucher für Kinesis Data Streams zu erstellen. Einige dieser Ansätze umfassen die Verwendung von Amazon Kinesis Data Analytics zur Analyse von Streaming-Daten mit KCL, die Verwendung von AWS Lambda, -Streaming-ETL-AufträgenAWS Glue und die direkte Verwendung der API von Kinesis Data Streams.

Verbraucheranwendungen für Kinesis Data Streams können mit der KCL entwickelt werden, die Sie bei der Nutzung und Verarbeitung von Daten aus Kinesis Data Streams unterstützt. Die KCL übernimmt viele der komplexen Aufgaben im Zusammenhang mit verteilter Datenverarbeitung, wie beispielsweise die Vornahme der Lastenverteilung zwischen mehreren Instances, das Reagieren auf Instance-Fehler, das Einrichten von Prüfpunkten für verarbeitete Datensätze und das Reagieren auf Resharding. Die KCL ermöglicht Ihnen, sich auf das Schreiben der Datensatzverarbeitungslogik zu konzentrieren. Weitere Informationen zum Erstellen einer eigenen KCL-Anwendung finden Sie unter Verwenden der Kinesis Client Library.

Sie können Lambda-Funktionen abonnieren, um automatisch Datensatz-Batches aus Ihrem Kinesis-Stream zu lesen und sie zu verarbeiten, wenn Datensätze im Stream erkannt werden. AWS Lambda fragt den Stream regelmäßig (einmal pro Sekunde) nach neuen Datensätzen ab. Werden neue Datensätze erkannt, ruft der Service die Lambda-Funktion auf und übergibt die neuen Datensätze als Parameter. Die Lambda-Funktion wird nur ausgeführt, wenn neue Datensätze erkannt werden. Sie können eine Lambda-Funktion einem Verbraucher mit gemeinsamem Durchsatz (Standard-Iterator) zuordnen.

Sie können einen Verbraucher erstellen, der eine Funktion namens Enhanced Fan-Out (Erweitertes Rundsenden) verwendet, wenn Sie einen speziellen Durchsatz benötigen, der nicht mit anderen Verbrauchern, die Daten aus dem Stream empfangen, konkurrieren soll. Mit dieser Funktion können Verbraucher Datensätze aus einem Stream mit einem Durchsatz von bis zu zwei MB Daten pro Sekunde und Shard empfangen.

In den meisten Fällen sollte Kinesis Data Analytics, KCL, AWS Glue oder AWS Lambda verwendet werden, um Daten aus einem Stream zu verarbeiten. Wenn Sie möchten können Sie jedoch mit der API von Kinesis Data Streams eine Verbraucheranwendung von Grund auf neu erstellen. Die API von Kinesis Data Streams bietet die Methoden `GetShardIterator` und `GetRecords` zum Abrufen von Daten aus einem Stream.

Bei diesem Pull-Modell extrahiert Ihr Code Daten direkt aus den Shards des Streams. Weitere Informationen zum Schreiben Ihrer eigenen Verbraucheranwendung mithilfe der API finden Sie unter

Entwickeln benutzerdefinierter Verbraucher mit gemeinsam genutztem Durchsatz mithilfe des AWS SDK für Java. Details zur API finden Sie in der Referenz zur API von Amazon Kinesis Data Streams.

Verarbeitung von Datenströmen mit AWS Lambda

Mit AWS Lambda können Sie Code ausführen, ohne Server bereitstellen und verwalten zu müssen. Mit Lambda können Sie Code für praktisch jeden Anwendungstyp oder Backend-Service ohne den damit verbundenen Verwaltungsaufwand ausführen. Sie laden einfach Ihren Code hoch und Lambda kümmert sich darum, dass Ihr Code mit hoher Verfügbarkeit ausgeführt und skaliert wird. Sie können Ihren Code so einrichten, dass er automatisch von anderen AWS-Services ausgelöst wird, oder ihn indirekt von einer beliebigen Web- oder Mobil-App aufrufen.

AWS Lambda lässt sich nativ in Amazon Kinesis Data Streams integrieren. Die Komplexität von Abfragen, Prüfpunkten und Fehlerbehandlung wird durch die Verwendung dieser nativen Integration reduziert. Dadurch kann sich der Lambda-Funktionscode auf die Verarbeitung der Geschäftslogik konzentrieren.

Sie können eine Lambda-Funktion einem Verbraucher mit gemeinsam genutzten Durchsatz (Standard-Iterator) oder einem Verbraucher mit dediziertem Durchsatz mit erweitertem Rundsenden zuordnen. Bei einem Standard-Iterator fragt Lambda jeden Shard in Ihrem Kinesis-Stream nach Datensätzen ab, die das HTTP-Protokoll verwenden. Um die Latenz zu minimieren und den Lesedurchsatz zu maximieren, können Sie einen Datenstromverbraucher mit erweitertem Rundsenden erstellen. Stream-Verbraucher in dieser Architektur erhalten eine dedizierte Verbindung zu jedem Shard, ohne mit anderen Anwendungen zu konkurrieren, die aus demselben Stream lesen. Amazon Kinesis Data Streams überträgt Datensätze über HTTP/2 an Lambda.

Standardmäßig ruft AWS Lambda Ihre Funktion auf, sobald Datensätze im Stream verfügbar sind. Um die Datensätze für Batch-Szenarien zu puffern, können Sie an der Ereignisquelle ein Batch-Fenster von bis zu fünf Minuten implementieren. Wenn Ihre Funktion einen Fehler zurückgibt, wiederholt Lambda die Batch-Verarbeitung, bis diese erfolgreich ist oder die Daten ablaufen.

Übersicht

Das Unternehmen InternetProvider nutzte Amazon Kinesis Data Streams zum Streamen von Benutzer- und Standortdaten. Der Datenstrom wurde von AWS Lambda genutzt, um die Daten mit Bandbreitenoptionen anzureichern, die in der Bibliothek der Funktion gespeichert sind. Nach der Anreicherung übermittelte AWS Lambda die Bandbreitenoptionen zurück an die Anwendung. Amazon Kinesis Data Streams und AWS Lambda übernahmen die Bereitstellung und Verwaltung

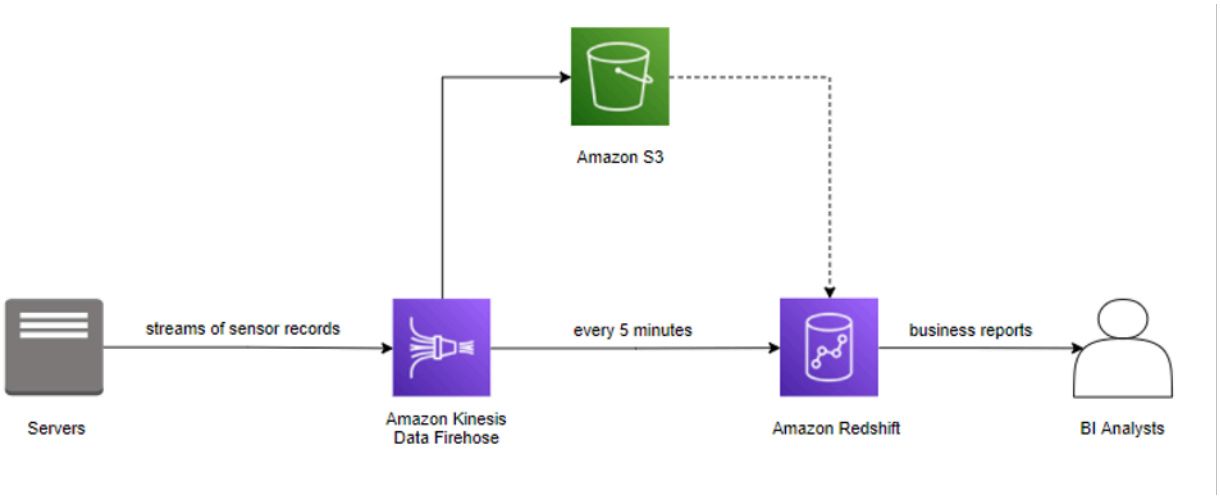
der Server, sodass sich das Unternehmen InternetProvider stärker auf die Entwicklung von Geschäftsanwendungen konzentrieren konnte.

Szenario 2: Echtzeitnahe Daten für Sicherheitsteams

Das Unternehmen ABC2Badge liefert Sensoren und digitale Ausweise für Firmen- oder Großveranstaltungen wie AWS re:Invent. Benutzer melden sich für die Veranstaltung an und erhalten eigene digitale Ausweise, die von den Sensoren auf dem gesamten Campus erfasst werden. Sobald Benutzer an einem Sensor vorbeigehen, werden ihre anonymisierten Informationen in einer relationalen Datenbank gespeichert.

Bei einer bevorstehenden Veranstaltung wurde ABC2Badge aufgrund der hohen Teilnehmerzahl vom Sicherheitsteam der Veranstaltung gebeten, alle 15 Minuten Daten für die Bereiche mit der höchsten Konzentration auf dem Campus zu sammeln. Dadurch hat das Sicherheitsteam genügend Zeit, um zu reagieren und das Sicherheitspersonal proportional zu den konzentrierten Bereichen zu verteilen. Vor dem Hintergrund dieser neuen Anforderung für das Sicherheitsteam und der Unerfahrenheit beim Aufbau einer Streaming-Lösung, die Daten nahezu in Echtzeit verarbeiten soll, sucht ABC2Badge nach einer einfachen, aber skalierbaren und zuverlässigen Lösung.

Die derzeitige Data-Warehouse-Lösung ist Amazon Redshift. Bei der Überprüfung der Funktionen der Amazon-Kinesis-Services erkannten sie, dass Amazon Kinesis Data Firehose einen Stream von Datensätzen empfangen, die Datensätze basierend auf Puffergröße und/oder Zeitintervall stapeln und in Amazon Redshift einfügen kann. Sie erstellten einen Kinesis-Data-Firehose-Bereitstellungsdatenstrom und konfigurierten ihn so, dass er alle fünf Minuten Daten in ihre Amazon-Redshift-Tabellen kopiert. Als Teil dieser neuen Lösung verwendeten sie den Amazon Kinesis Agent auf ihren Servern. Alle fünf Minuten lädt Kinesis Data Firehose Daten in Amazon Redshift, wo das Business-Intelligence-(BI)-Team seine Analysen durchführen und die Daten alle 15 Minuten an das Sicherheitsteam senden kann.



Neue Lösung mit Amazon Kinesis Data Firehose

Amazon Kinesis Data Firehose

Die einfachste Methode, um Streaming-Daten in AWS zu laden, bietet Amazon Kinesis Data Firehose. Amazon Kinesis Data Firehose kann Streaming-Daten erfassen, umwandeln und in Amazon Kinesis Data Analytics, Amazon Simple Storage Service (Amazon S3), Amazon Redshift, Amazon OpenSearch Service (OpenSearch Service) und Splunk laden. Darüber hinaus kann Kinesis Data Firehose Streaming-Daten in jeden benutzerdefinierten HTTP-Endpunkt oder HTTP-Endpunkte von unterstützten Drittanbietern laden.

Kinesis Data Firehose ermöglicht echtzeitnahe Analysen mit vorhandenen Business-Intelligence-Tools und Dashboards, die Sie bereits heute verwenden. Es handelt sich hier um einen vollständig verwalteten Serverless-Service, der automatisch dem Durchsatz Ihrer Daten entsprechend skaliert wird und keine weitere Verwaltung erfordert. Kinesis Data Firehose kann die Daten vor dem Laden in Batches unterteilen, komprimieren und verschlüsseln, um den am Zielort verwendeten Speicherplatz zu minimieren und die Sicherheit zu erhöhen. Es kann auch die Quelldaten mithilfe von AWS Lambda umwandeln und die umgewandelten Daten an Ziele liefern. Sie konfigurieren Ihre Datenproduzenten so, dass die Daten an Amazon Kinesis Data Firehose gesendet werden, das die Daten automatisch an dem von Ihnen angegebenen Ziel bereitstellt.

Senden von Daten an einen Firehose-Bereitstellungsdatenstrom

Um Daten an Ihren Bereitstellungsdatenstrom zu senden, gibt es mehrere Möglichkeiten. AWS bietet SDKs für viele gängige Programmiersprachen, von denen jede APIs für Amazon Kinesis Data Firehose bereitstellt. AWS verfügt über ein Dienstprogramm, mit dem Sie Daten an Ihren

Bereitstellungsdatenstrom senden können. Kinesis Data Firehose wurde in andere AWS-Services integriert, um Daten direkt von diesen Services in Ihren Bereitstellungsdatenstrom zu senden.

Verwenden des Amazon Kinesis Agent

Amazon Kinesis Agent ist eine eigenständige Softwareanwendung, die kontinuierlich eine Reihe von Protokolldateien auf neue Daten überwacht, die an den Bereitstellungsdatenstrom gesendet werden sollen. Der Agent kümmert sich automatisch um die Dateirotation, Prüfpunkte, Wiederholungsversuche bei Fehlern und gibt Amazon CloudWatch-Metriken zur Überwachung und Fehlerbehebung des Bereitstellungsdatenstroms aus. Zusätzliche Konfigurationen, wie z. B. die Vorverarbeitung von Daten, die Überwachung mehrerer Dateiverzeichnisse und das Schreiben in mehrere Bereitstellungsdatenströme, können auf den Agent angewendet werden.

Der Agent kann auf Linux- oder Windows-basierten Servern wie Webservern, Protokollservern und Datenbankservern installiert werden. Nach der Installation des Agent geben Sie einfach an, welche Protokolldateien er überwachen und an welchen Bereitstellungsdatenstrom er senden soll. Der Agent sendet dauerhaft und zuverlässig neue Daten an den Bereitstellungsdatenstrom.

Verwenden der API mit AWS SDK und AWS-Services als Quelle

Die Kinesis-Data-Firehose-API bietet zwei Operationen zum Senden von Daten an Ihren Bereitstellungsdatenstrom. `PutRecord` sendet einen einzelnen Datensatz innerhalb eines Aufrufs. `PutRecordBatch` kann mehrere Datensätze innerhalb eines Aufrufs senden und einen höheren Durchsatz pro Produzent erreichen. Bei jeder Methode müssen Sie den Namen des Bereitstellungsdatenstroms und den Datensatz oder das Array von Datensätzen angeben, wenn Sie diese Methode verwenden. Weitere Informationen und Beispielcode für die Kinesis-Data Firehose-API-Operationen finden Sie unter Schreiben in einen Firehose-Bereitstellungsdatenstrom mit dem AWS SDK.

Kinesis Data Firehose läuft auch mit Kinesis Data Firehose, CloudWatch Logs, CloudWatch Events, Amazon Simple Notification Service (Amazon SNS), Amazon API Gateway und AWS IoT. Sie können Ihre Datenströme, Protokolle, Ereignisse und IoT-Daten skalierbar und zuverlässig direkt an ein Kinesis-Data-Firehose-Ziel senden.

Verarbeiten von Daten vor der Bereitstellung am Ziel

In manchen Szenarien möchten Sie möglicherweise Ihre Streaming-Daten umwandeln oder optimieren, bevor sie an ihrem Ziel bereitgestellt werden. Datenproduzenten könnten zum Beispiel unstrukturierten Text in jedem Datensatz senden, und Sie müssen ihn in JSON umwandeln,

bevor Sie ihn an OpenSearch Service übermitteln. Oder Sie möchten die JSON-Daten in ein spaltenförmiges Dateiformat wie Apache Parquet oder Apache ORC konvertieren, bevor Sie die Daten in Amazon S3 speichern.

Kinesis Data Firehose verfügt über eine integrierte Konvertierungsfunktion für Datenformate. Damit können Sie Ihre JSON-Datenströme problemlos in Apache-Parquet- oder Apache-ORC-Dateiformate konvertieren.

Datentransformationsfluss

Um Streaming-Datentransformationen zu ermöglichen, verwendet Kinesis Data Firehose eine Lambda-Funktion, die Sie zur Transformation Ihrer Daten erstellen. Kinesis Data Firehose puffert eingehende Daten auf eine für die Funktion angegebene Puffergröße und ruft dann die angegebene Lambda-Funktion asynchron auf. Die transformierten Daten werden von Lambda an Kinesis Data Firehose gesendet, und Kinesis Data Firehose liefert die Daten an das Ziel.

Konvertierung von Datenformaten

Sie können auch die Datenformatkonvertierung von Kinesis Data Firehose aktivieren, die den JSON-Datenstrom in das Apache-Parquet- oder Apache-ORC-Datenformat konvertiert. Diese Funktion kann nur JSON in das Apache-Parquet- oder Apache-ORC-Datenformat konvertieren. Wenn Ihre Daten im CSV-Format vorliegen, können Sie diese Daten über eine Lambda-Funktion in JSON umwandeln und dann die Datenformatkonvertierung anwenden.

Datenbereitstellung

Kinesis Data Firehose stellt eingehende Daten nahezu in Echtzeit bereit und puffert diese. Nachdem die Schwellenwerte für die Pufferung Ihres Datenstroms erreicht wurden, werden Ihre Daten an das von Ihnen konfigurierte Ziel übermittelt. Es gibt einige Unterschiede in der Art und Weise, wie Kinesis Data Firehose Daten an die einzelnen Ziele liefert, die in diesem Dokument in den folgenden Abschnitten erläutert werden.

Amazon S3

Amazon S3 ist ein Objektspeicher mit einer einfachen Webservice-Schnittstelle. Sie können darüber von überall im Internet beliebige Datenmengen abrufen und speichern. Er wurde für eine Datenbeständigkeit von 99,999999999 % entwickelt und kann weltweit auf Billionen von Objekten skaliert werden.

Datenbereitstellung in Amazon S3

Für die Datenbereitstellung in Amazon S3 verkettet Kinesis Data Firehose mehrere eingehende Datensätze basierend auf der Pufferkonfiguration Ihres Bereitstellungsdatenstroms und stellt sie dann als S3-Objekt in Amazon S3 bereit. Die Häufigkeit der Datenbereitstellung in S3 wird durch die S3-Puffergröße (1 MB bis 128 MB) oder das Pufferintervall (60 Sekunden bis 900 Sekunden) bestimmt, je nachdem, was zuerst eintritt.

Die Datenbereitstellung in Ihrem S3-Bucket kann aus verschiedenen Gründen fehlschlagen. Zum Beispiel könnte der Bucket nicht mehr existieren, oder die AWS Identity and Access Management-(IAM-)Rolle, die Kinesis Data Firehose übernimmt, könnte keinen Zugriff auf den Bucket haben. Unter diesen Umständen wiederholt Kinesis Data Firehose den Vorgang bis zu 24 Stunden, bis die Bereitstellung erfolgreich ist. Die maximale Datenspeicherzeit von Kinesis Data Firehose beträgt 24 Stunden. Falls die Datenbereitstellung länger als 24 Stunden fehlschlägt, gehen die Daten verloren.

Amazon Redshift

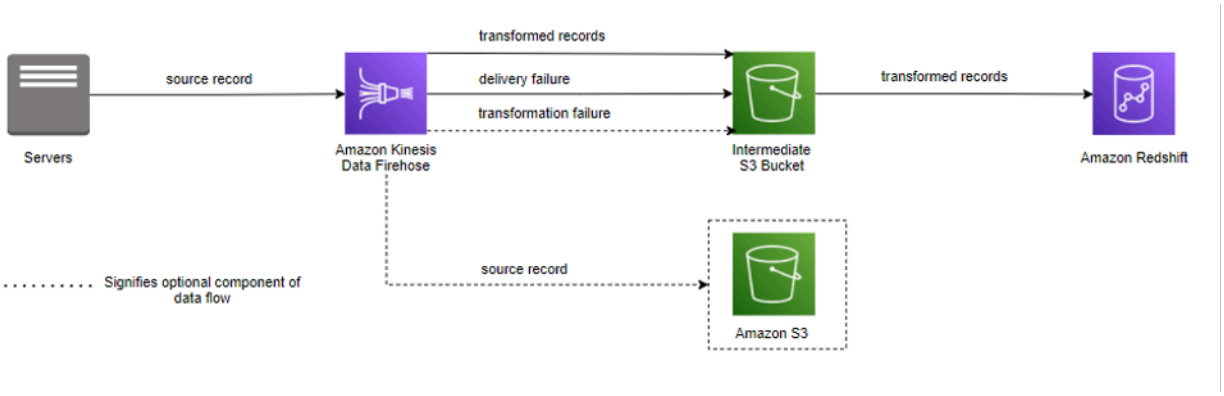
Amazon Redshift ist ein schnelles, vollständig verwaltetes Data Warehouse, mit dem Sie im Zusammenspiel mit Ihren vorhandenen BI-Tools und mithilfe von Standard-SQL alle Ihre Daten einfach und kostengünstig analysieren können. Es ermöglicht Ihnen die Ausführung komplexer Analyseabfragen für mehrere Petabyte strukturierter Daten mithilfe einer durchdachten Abfrageoptimierung, Spaltenspeicherung auf lokalen Hochleistungsdatenträgern und einer umfangreichen parallel laufenden Abfrage.

Datenbereitstellung in Amazon Redshift

Für die Datenbereitstellung in Amazon Redshift stellt Kinesis Data Firehose zunächst die eingehenden Daten in Ihrem S3-Bucket in dem zuvor beschriebenen Format bereit. Kinesis Data Firehose gibt dann einen Amazon-Redshift-COPY-Befehl aus, um die Daten aus Ihrem S3-Bucket in Ihren Amazon-Redshift-Cluster zu laden.

Die Häufigkeit der `COPY`-Vorgänge für Daten von S3 nach Amazon Redshift wird davon bestimmt, wie schnell Ihr Amazon-Redshift-Cluster den `COPY`-Befehl abschließen kann. Für ein Amazon-Redshift-Ziel können Sie beim Erstellen eines Bereitstellungsdatenstroms eine Wiederholungsdauer (0–7200 Sekunden) angeben, um Fehler bei der Datenbereitstellung zu beheben. Bei nicht erfolgreicher Ausführung wiederholt Kinesis Data Firehose den Vorgang für den festgelegten Zeitraum und überspringt den jeweiligen Batch von S3-Objekten. Die Informationen zu den übersprungenen Objekten werden Ihrem S3-Bucket als Manifestdatei im Ordner „Errors/“ (Fehler) bereitgestellt. Sie können diesen für manuelle Backfill-Vorgänge verwenden.

Nachstehend finden Sie ein Architekturdiagramm des Datenflusses von Kinesis Data Firehose zu Amazon Redshift. Obwohl dieser Datenfluss nur für Amazon Redshift gilt, folgt Kinesis Data Firehose ähnlichen Mustern für die anderen Zielspeicherorte.



Datenfluss von Kinesis Data Firehose zu Amazon Redshift

Amazon OpenSearch Service (OpenSearch Service)

OpenSearch Service ist ein vollständig verwalteter Service, der die benutzerfreundlichen OpenSearch-APIs und Echtzeitfunktionen zusammen mit der Verfügbarkeit, Skalierbarkeit und Sicherheit bereitstellt, die für Produktionsworkloads erforderlich sind. OpenSearch Service erleichtert die Bereitstellung, den Betrieb und die Skalierung von OpenSearch für Protokollanalysen, Volltextsuche und Anwendungsüberwachung.

Datenbereitstellung in OpenSearch Service

Für die Datenbereitstellung in OpenSearch Service puffert Kinesis Data Firehose eingehende Datensätze basierend auf der Pufferkonfiguration Ihres Bereitstellungsdatenstroms und generiert dann eine OpenSearch-Massenanfrage, um mehrere Datensätze in Ihrem OpenSearch-Cluster zu indizieren. Die Häufigkeit der Datenbereitstellung in OpenSearch Service wird durch die Werte für die OpenSearch-Puffergröße (1 MB bis 100 MB) und das Pufferintervall (60 Sekunden bis 900 Sekunden) bestimmt, je nachdem, was zuerst eintritt.

Sie können beim Erstellen eines Bereitstellungsdatenstroms für das OpenSearch Service-Ziel eine Wiederholungsdauer angeben (0–7200 Sekunden). Kinesis Data Firehose wiederholt den Vorgang für den festgelegten Zeitraum und überspringt anschließend die jeweilige Indexanfrage. Die übersprungenen Dokumente werden Ihrem S3-Bucket im Ordner `elasticsearch_failed/` bereitgestellt. Sie können diesen für manuelle Backfill-Vorgänge verwenden.

Amazon Kinesis Data Firehose kann Ihren OpenSearch Service-Index basierend auf einer Zeitdauer rotieren. Je nach gewählter Rotationsoption (`NoRotation`, `OneHour`, `OneDay`, `OneWeek` oder

`OneMonth`) fügt Kinesis Data Firehose einen Abschnitt des UTC-Ankunftszeitstempels (Coordinated Universal Time, UTC) an den angegebenen Indexnamen an.

Benutzerdefinierter HTTP-Endpunkt oder unterstützter externer Dienstanbieter

Kinesis Data Firehose kann Daten entweder an benutzerdefinierte HTTP-Endpunkte oder an unterstützte Drittanbieter wie Datadog, Dynatrace, LogicMonitor, MongoDB, New Relic, Splunk und Sumo Logic senden.

Benutzerdefinierter HTTP-Endpunkt oder unterstützter externer Dienstanbieter

Damit Kinesis Data Firehose erfolgreich Daten an benutzerdefinierte HTTP-Endpunkte liefern kann, müssen diese Endpunkte Anfragen akzeptieren und Antworten unter Verwendung bestimmter Kinesis-Data-Firehose-Anfrage- und Antwortformate senden.

Wenn Sie Daten an einem HTTP-Endpunkt bereitstellen, der zu einem unterstützten externen Dienstanbieter gehört, können Sie den integrierten AWS Lambda-Service verwenden, um eine Funktion zu erstellen, die den eingehenden Datensatz/die eingehenden Datensätze in das Format transformiert, das dem Format entspricht, das die Integration des Dienstanbieters erwartet.

Für die Häufigkeit der Datenbereitstellung gibt jeder Dienstanbieter eine empfohlene Puffergröße an. Wenden Sie sich an Ihren Dienstanbieter, um weitere Informationen über die empfohlene Puffergröße zu erhalten. Für die Behandlung von Datenbereitstellungsfehlern stellt Kinesis Data Firehose zunächst eine Verbindung mit dem HTTP-Endpunkt her, indem es auf eine Antwort vom Ziel wartet. Kinesis Data Firehose versucht so lange eine Verbindung herzustellen, bis die Wiederholungsdauer abgelaufen ist. Danach betrachtet Kinesis Data Firehose den Vorgang als Datenbereitstellungsfehler und sichert die Daten in Ihrem S3 Bucket.

Übersicht

Kinesis Data Firehose kann Ihre Streaming-Daten dauerhaft an ein unterstütztes Ziel liefern. Es handelt sich um eine vollständig verwaltete Lösung, die wenig oder gar keine Entwicklung erfordert. Für das Unternehmen ABC2Badge war die Verwendung von Kinesis Data Firehose eine logische Wahl. Es verwendete bereits Amazon Redshift als Data-Warehouse-Lösung. Da seine Datenquellen kontinuierlich in Transaktionsprotokolle schrieben, konnte das Unternehmen den Amazon Kinesis Agent nutzen, um diese Daten zu streamen, ohne zusätzlichen Code schreiben zu müssen. Nachdem das Unternehmen ABC2Badge einen Stream von Sensoraufzeichnungen erstellt hat und diese Aufzeichnungen über Kinesis Data Firehose empfängt, kann es diese als Grundlage für den Anwendungsfall des Sicherheitsteams verwenden.

Szenario 3: Aufbereitung von Clickstream-Daten für Erkenntnisse aus Daten

Fast Sneakers ist eine Modeboutique mit Schwerpunkt auf im Trend liegenden Sneakers. Der Preis für ein bestimmtes Paar Schuhe kann steigen oder fallen, je nach Bestand und Trends, z. B. welcher Prominente oder Sportstar gestern Abend im Fernsehen mit Markensneakern gesehen wurde. Für das Unternehmen Fast Sneakers ist es wichtig, diese Trends zu verfolgen und zu analysieren, um seinen Umsatz zu maximieren.

Fast Sneakers möchte zusätzlichen Projektaufwand für die Wartung einer neuen Infrastruktur vermeiden. Es möchten in der Lage sein, die Entwicklung auf die am Projekt beteiligten Parteien so aufzuteilen, dass sich die Dateningenieure auf die Datentransformation konzentrieren können und die Datenwissenschaftler separat an ihren ML-Funktionen arbeiten können.

Um schnell zu reagieren und die Preise automatisch an die Nachfrage anzupassen, streamt Fast Sneakers wichtige Ereignisse (wie Klick- und Kaufdaten), transformiert und ergänzt die Ereignisdaten und speist sie in ein ML-Modell ein. Sein ML-Modell kann feststellen, ob eine Preisanpassung erforderlich ist. Auf diese Weise kann Fast Sneakers seine Preise automatisch anpassen, um den Gewinn für seine Produkte zu maximieren.

Fast Sneakers – Preisanpassungen in Echtzeit

Dieses Architekturdiagramm zeigt die Echtzeit-Streaming-Lösung, die Fast Sneakers mithilfe von Kinesis Data Streams, AWS Glue und DynamoDB Streams erstellt hat. Durch die Nutzung dieser Services steht dem Unternehmen eine elastische und zuverlässige Lösung zur Verfügung, ohne

dass es Zeit für die Einrichtung und Wartung der unterstützenden Infrastruktur aufwenden muss. Das Unternehmen kann seine Zeit auf das verwenden, was ihm einen Mehrwert bringt, indem es sich auf einen ETL-Streaming-Auftrag (Extract, Transform, Load) und sein Machine-Learning-Modell konzentriert.

Zum besseren Verständnis der Architektur und der Technologien, die in seinem Workload verwendet werden, werden im Folgenden einige Einzelheiten zu den verwendeten Services aufgeführt.

AWS Glue- und AWS Glue-Streaming

AWS Glue ist ein vollständig verwalteter ETL-Service, mit dem Sie Ihre Daten katalogisieren, bereinigen, anreichern und zuverlässig zwischen Datenspeichern verschieben können. Mit AWS Glue können Sie die Kosten, die Komplexität und den Zeitaufwand für die Erstellung von ETL-Aufträgen erheblich reduzieren. AWS Glue ist Serverless, d. h. es muss keine Infrastruktur eingerichtet oder verwaltet werden. Sie zahlen nur für die Ressourcen, die verbraucht werden, während Ihre Aufträge ausgeführt werden.

Mithilfe von AWS Glue können Sie eine Verbraucheranwendung mit einem AWS Glue-Streaming-ETL-Auftrag erstellen. So können Sie Apache Spark und andere Spark-basierte Module verwenden, um Ihre Ereignisdaten zu nutzen und zu verarbeiten. Der nächste Abschnitt dieses Dokuments geht näher auf dieses Szenario ein.

AWS Glue Data Catalog

Der AWS Glue Data Catalog enthält Verweise auf Daten, die als Quellen und Ziele für Ihre ETL-Aufträge in AWS Glue verwendet werden. Der AWS Glue Data Catalog ist ein Index für die Speicherort-, Schema- und Laufzeitmetriken Ihrer Daten. Sie können die Informationen im Datenkatalog verwenden, um Ihre ETL-Aufträge zu erstellen und zu überwachen. Informationen im Datenkatalog werden als Metadatentabellen gespeichert, wobei jede Tabelle einen einzigen Datenspeicher angibt. Durch die Einrichtung eines Crawlers können Sie zahlreiche Arten von Datenspeichern, einschließlich DynamoDB, S3 und mit Java Database Connectivity (JDBC) verbundene Speicher, automatisch bewerten, Metadaten und Schemata extrahieren und anschließend Tabellendefinitionen im AWS Glue Data Catalog erstellen.

Um mit Amazon Kinesis Data Streams in AWS Glue-Streaming-ETL-Aufträgen zu arbeiten, empfiehlt es sich, den Stream in einer Tabelle in einer AWS Glue Data Catalog-Datenbank zu definieren. Sie definieren eine Stream-bezogene Tabelle mit dem Kinesis-Stream, einem der vielen unterstützten Formate (CSV, JSON, ORC, Parquet, Avro oder ein Kundenformat mit Grok). Sie können ein Schema

manuell eingeben oder diesen Schritt Ihrem AWS Glue-Auftrag überlassen, um ihn während der Laufzeit des Auftrags festzulegen.

AWS Glue-Streaming-ETL-Auftrag

AWS Glue führt Ihre ETL-Aufträge in einer Apache-Spark-Serverless-Umgebung aus. AWS Glue führt diese Aufträge auf virtuellen Ressourcen aus, die es in seinem eigenen Servicekonto bereitstellt und verwaltet. Neben der Möglichkeit, Apache-Spark-basierte Aufträge auszuführen, bietet AWS Glue mit DynamicFrames zusätzlich zu Spark eine weitere Funktionsebene.

DynamicFrames sind verteilte Tabellen, die verschachtelte Daten wie Strukturen und Arrays unterstützen. Jeder Datensatz ist selbstbeschreibend und wurde auf Schema-Flexibilität mit halbstrukturierten Daten ausgelegt. Ein Datensatz in einem `DynamicFrame` enthält sowohl Daten als auch das Schema, das die Daten beschreibt. Sowohl Apache-Spark-`DataFrames` als auch `DynamicFrames` werden in Ihren ETL-Skripten unterstützt, und Sie können sie hin und her konvertieren. `DynamicFrames` bieten eine Reihe von erweiterten Transformationen für die Datenbereinigung und ETL.

Durch die Verwendung von Spark Streaming in Ihrem AWS Glue-Auftrag können Sie Streaming-ETL-Aufträge erstellen, die kontinuierlich ausgeführt werden, und Daten aus Streaming-Quellen wie Amazon Kinesis Data Streams, Apache Kafka und Amazon MSK nutzen. Die Aufträge können die Daten bereinigen, zusammenführen und transformieren und dann die Ergebnisse in Speicher wie Amazon-S3-, Amazon-DynamoDB- oder JDBC-Datenspeicher laden.

AWS Glue verarbeitet und schreibt standardmäßig Daten in 100-Sekunden-Fenstern. Dadurch können Daten effizient verarbeitet und Aggregationen für Daten ausgeführt werden, die später als erwartet eintreffen. Sie können die Fenstergröße konfigurieren, indem Sie sie an die Reaktionsgeschwindigkeit im Vergleich zur Genauigkeit Ihrer Aggregation anpassen. AWS Glue-Streaming-Aufträge verwenden Prüfpunkte, um die Daten zu verfolgen, die aus dem Kinesis-Datenstrom gelesen wurden. Eine Anleitung zum Erstellen eines Streaming-ETL-Jobs in AWS Glue finden Sie unter Hinzufügen von Streaming-ETL-Aufträgen in AWS Glue.

Amazon DynamoDB

Amazon DynamoDB ist eine Schlüssel-Werte- und Dokumentdatenbank, die für beliebig große Datenmengen eine Leistung im einstelligen Millisekundenbereich bereitstellt. Es handelt sich um eine vollständig verwaltete, multiregionale, multiaktivfähige, dauerhafte Datenbank mit integrierter Sicherheit, Sicherung und Wiederherstellung sowie In-Memory-Caching für Anwendungen im

Internetmaßstab. DynamoDB kann mehr als zehn Billionen Anforderungen pro Tag bearbeiten und Spitzen von mehr als 20 Millionen Anforderungen pro Sekunde unterstützen.

Erfassung von Datenänderungen für DynamoDB-Streams

Ein DynamoDB-Stream ist ein strukturierter Informationsfluss zu Elementänderungen in einer DynamoDB-Tabelle. Wenn Sie den Stream für eine Tabelle aktivieren, werden von DynamoDB Informationen über jede Änderung an den Datenelementen in der Tabelle erfasst. DynamoDB wird auf AWS Lambda ausgeführt, so dass Sie Auslöser erstellen können – Codeelemente, die auf Ereignisse in DynamoDB-Streams automatisch reagieren. Mit Auslösern können Sie Anwendungen entwickeln, die auf Datenänderungen in DynamoDB-Tabellen reagieren.

Wenn ein Stream für eine Tabelle aktiviert ist, können Sie den Stream-Amazon-Ressourcennamen (ARN) mit einer Lambda-Funktion verknüpfen, die Sie schreiben. Sobald ein Element in der Tabelle geändert wurde, erscheint ein neuer Datensatz im Stream der Tabelle. AWS Lambda fragt den Stream ab und ruft die Lambda-Funktion bei Erkennung neuer Stream-Datensätze synchron auf.

Service-Endpunkte von Amazon SageMaker und Amazon SageMaker

Amazon SageMaker ist eine vollständig verwaltete Plattform, die Entwicklern und Datenwissenschaftlern die Möglichkeit bietet, ML-Modelle schnell und in beliebigem Umfang zu erstellen, zu trainieren und bereitzustellen. SageMaker besteht aus Modulen, die unabhängig voneinander oder gemeinsam für das Entwickeln, Trainieren und Bereitstellen Ihrer ML-Modelle verwendet werden können. Mit Amazon-SageMaker-Service-Endpunkten können Sie einen verwalteten, bereitgestellten Endpunkt für Echtzeit-Inferenz mit einem bereitgestellten Modell erstellen, das Sie innerhalb oder außerhalb von Amazon SageMaker entwickelt haben.

Mithilfe des AWS SDK können Sie einen SageMaker-Endpunkt aufrufen, der Informationen zum Inhaltstyp zusammen mit dem Inhalt übergibt, und dann Echtzeit-Vorhersagen basierend auf den übergebenen Daten erhalten. Auf diese Weise können Sie die Gestaltung und Entwicklung Ihrer ML-Modelle von dem Code trennen, mit dem Aktionen für die abgeleiteten Ergebnisse ausgeführt werden.

So können Ihre Datenwissenschaftler ihr Hauptaugenmerk auf ML richten, während sich die Entwickler, die das ML-Modell verwenden, darauf konzentrieren können, wie sie es in ihrem Code verwenden. Weitere Informationen über das Aufrufen eines Endpunkts in SageMaker finden Sie in der Referenz zum Aufrufen eines Endpunkts in der Amazon-SageMaker-API.

Ableiten von Datenerkenntnissen in Echtzeit

Das vorherige Architekturdiagramm zeigt, dass die bestehende Webanwendung von Fast Sneakers einen Kinesis-Datenstrom mit Clickstream-Ereignissen hinzugefügt hat, der Daten zum Datenverkehr und zu den Ereignissen von der Website bereitstellt. Der Produktkatalog, der Informationen wie Kategorisierung, Produktattribute und Preise enthält, und die Bestelltabelle, die Daten wie bestellte Artikel, Fakturierung, Versand usw. enthält, sind separate DynamoDB-Tabellen. Die Metadaten und Schemata der Datenstromquelle und der entsprechenden DynamoDB-Tabellen sind in AWS Glue Data Catalog definiert und werden vom AWS Glue-Streaming-ETL-Auftrag verwendet.

Durch die Verwendung von Apache Spark, Spark Streaming und `DynamicFrames` in seinem AWS Glue-Streaming-ETL-Auftrag ist das Unternehmen Fast Sneakers in der Lage, Daten aus beiden Datenströmen zu extrahieren und zu transformieren und Daten aus den Produkt- und Auftragstabellen zusammenzuführen. Mit den hydratisierten Daten aus der Transformation werden die Datensätze, aus denen Inferenzergebnisse abgerufen werden sollen, an eine DynamoDB-Tabelle übermittelt.

Der DynamoDB-Stream für die Tabelle löst für jeden neu geschriebenen Datensatz eine Lambda-Funktion aus. Die Lambda-Funktion übermittelt die zuvor umgewandelten Datensätze an einen SageMaker-Endpunkt mit dem AWS SDK, um abzuleiten, ob und welche Preisanpassungen für ein Produkt erforderlich sind. Wenn das ML-Modell feststellt, dass eine Anpassung des Preises erforderlich ist, schreibt die Lambda-Funktion die Preisänderung für das Produkt in die DynamoDB-Tabelle des Katalogs.

Übersicht

Amazon Kinesis Data Streams vereinfacht das Erfassen, Verarbeiten und Analysieren von Echtzeit-Streaming-Daten, damit Sie zeitnahe Einblicke erhalten und schnell auf neue Informationen reagieren können. In Kombination mit dem Serverless-Datenintegrationsdienst von AWS Glue können Sie Echtzeit-Ereignis-Streaming-Anwendungen erstellen, die Daten für ML vorbereiten und kombinieren.

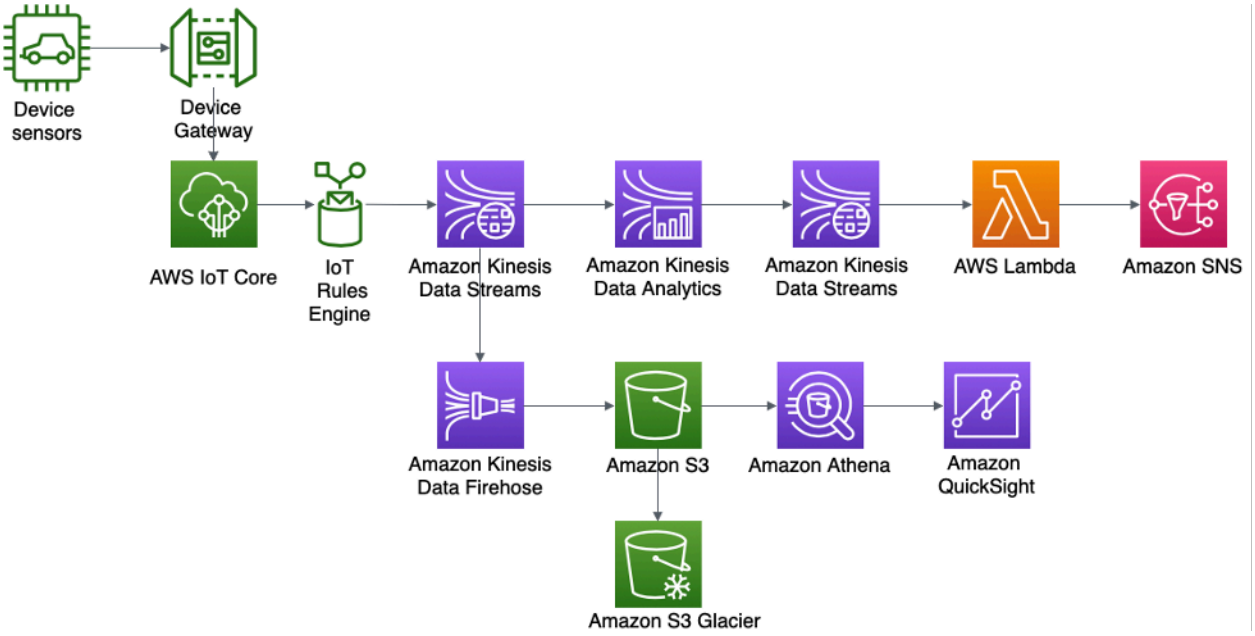
Da sowohl Kinesis Data Streams als auch die AWS Glue-Services vollständig verwaltet werden, nimmt AWS Ihnen den undifferenzierten Arbeitsaufwand der Verwaltung der Infrastruktur für Ihre Big-Data-Plattform ab, sodass Sie sich auf das Gewinnen von Erkenntnissen auf der Grundlage Ihrer Daten konzentrieren können.

Das Unternehmen Fast Sneakers kann die Echtzeit-Ereignisverarbeitung und ML nutzen, um seine Website in die Lage zu versetzen, vollautomatische Preisanpassungen in Echtzeit vorzunehmen,

um seinen Produktbestand zu maximieren. Dies bringt dem Unternehmen den größten Nutzen und vermeidet gleichzeitig die Notwendigkeit, eine Big-Data-Plattform zu erstellen und zu pflegen.

Szenario 4: Erkennung von Unregelmäßigkeiten durch Gerätesensoren in Echtzeit und Benachrichtigung

Das Unternehmen ABC4Logistics transportiert leicht entzündliche Mineralölprodukte wie Benzin, Flüssigpropan (LPG) und Naphtha vom Hafen in verschiedene Städte. Es besitzt eine Flotte mit Hunderten Fahrzeugen, die mit mehreren Sensoren ausgerüstet sind, um z. B. den Standort, die Motortemperatur, die Temperatur im Inneren des Containers, die Fahrgeschwindigkeit, den Standort des Fahrzeugs, den Straßenzustand und so weiter zu überwachen. Eine der Anforderungen von ABC4Logistics besteht darin, die Temperaturen des Motors und des Containers in Echtzeit zu überwachen und den Fahrer und das Flottenüberwachungsteam im Falle einer Unregelmäßigkeit zu alarmieren. Um solche Bedingungen zu erkennen und Warnungen in Echtzeit zu generieren, implementierte ABC4Logistics die folgende Architektur in AWS.



Geräte-Sensoren für die Echtzeit-Erkennung von Unregelmäßigkeiten und Benachrichtigungsarchitektur von ABC4Logistics

Die Daten von Gerätesensoren werden vom AWS IoT-Gateway aufgenommen, in dem die -RegelAWS IoT-Engine die Streaming-Daten in Amazon Kinesis Data Streams zur Verfügung stellt. Mit Kinesis Data Analytics kann ABC4Logistics die Echtzeitanalyse von Streaming-Daten in Kinesis Data Streams durchführen.

Mithilfe von Kinesis Data Analytics kann ABC4Logistics erkennen, wenn die Temperaturmesswerte der Sensoren über einen Zeitraum von zehn Sekunden von den normalen Messwerten abweichen, und den Datensatz in eine andere Kinesis-Data-Streams-Instance einlesen, um die anomalen Datensätze zu identifizieren. Amazon Kinesis Data Streams ruft dann Lambda-Funktionen auf, die die Warnungen über Amazon SNS an den Fahrer und das Flottenüberwachungsteam senden können.

Daten in Kinesis Data Streams werden auch an Amazon Kinesis Data Firehose weitergeleitet. Amazon Kinesis Data Firehose persistiert diese Daten in Amazon S3 und ermöglicht ABC4Logistics die Durchführung von Batch- oder echtzeitnahen Analysen der Sensordaten. ABC4Logistics verwendet Amazon Athena zur Abfrage von Daten in S3 und Amazon QuickSight für Visualisierungen. Für die langfristige Datenaufbewahrung wird die S3-Lebenszyklus-Richtlinie zur Archivierung von Daten in Amazon S3 Glacier verwendet.

Wichtige Komponenten dieser Architektur werden im Folgenden erläutert.

Amazon Kinesis Data Analytics

Amazon Kinesis Data Analytics ermöglicht es Ihnen, Streaming-Daten zu transformieren und zu analysieren und auf Unregelmäßigkeiten in Echtzeit zu reagieren. Es handelt sich um einen Serverless-Service in AWS, d. h., Kinesis Data Analytics übernimmt die Bereitstellung und elastische Skalierung der Infrastruktur für die Verarbeitung eines beliebigen Datendurchsatzes. Dadurch entfällt der undifferenzierte Arbeitsaufwand beim Einrichten und Verwalten der Streaming-Infrastruktur und Sie können mehr Zeit für das Schreiben von Steaming-Anwendungen aufwenden.

Mit Amazon Kinesis Data Analytics können Sie interaktiv Streaming-Daten mit mehreren Optionen abfragen, einschließlich Standard-SQL-, Apache-Flink-Anwendungen in Java, Python und Scala, und Apache-Beam-Anwendungen mit Java zur Analyse von Datenströmen erstellen.

Diese Optionen bieten Ihnen die Flexibilität, je nach Komplexität der Streaming-Anwendung und der Quell-/Zielunterstützung einen bestimmten Ansatz zu wählen. Im folgenden Abschnitt wird die Option Kinesis Data Analytics für Flink-Anwendungen beschrieben.

Kinesis Data Analytics für Apache-Flink-Anwendungen

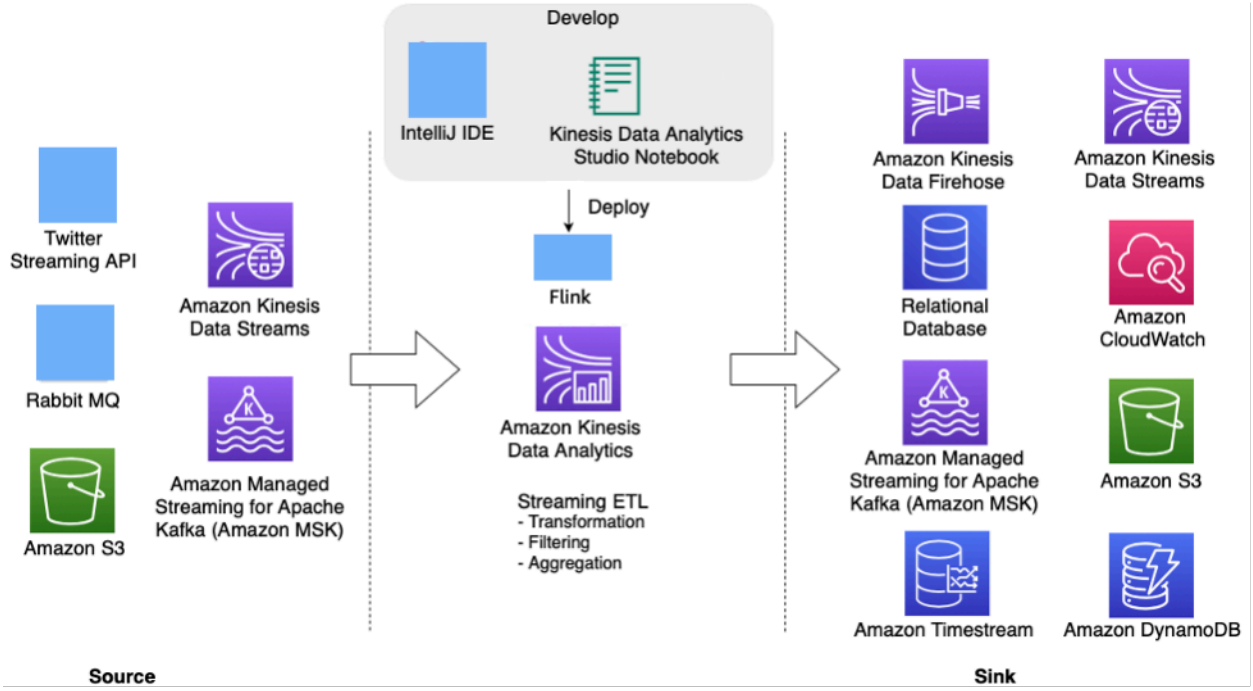
Apache Flink ist ein gängiges Open-Source-Framework und eine Engine für verteilte Verarbeitung für zustandsbehaftete Berechnungen über unbegrenzte und begrenzte Datenströme. Apache Flink wurde entwickelt, um Berechnungen mit In-Memory-Geschwindigkeit und in großem Umfang mit Unterstützung für „Genau Einmal“-Semantik durchzuführen. Mit Apache-Flink-basierten

Anwendungen lassen sich eine niedrige Latenz bei hohem Durchsatz auf fehlertolerante Weise erreichen.

Mit Amazon Kinesis Data Analytics für Apache Flink können Sie Code für Streaming-Quellen erstellen und ausführen, um Zeitreihenanalysen durchzuführen, Echtzeit-Dashboards zu speisen und Echtzeit-Metriken zu erstellen, ohne die komplexe verteilte Apache-Flink-Umgebung verwalten zu müssen. Sie können die allgemeinen Funktionen der Flink-Programmierung auf die gleiche Weise nutzen, wie Sie sie nutzen, wenn Sie die Flink-Infrastruktur selbst hosten.

Mit Kinesis Data Analytics für Apache Flink können Sie Anwendungen in Java, Scala, Python oder SQL zur Verarbeitung und Analyse von Streaming-Daten erstellen. Eine typische Flink-Anwendung liest die Daten aus dem Eingabe-Stream oder dem Datenspeicherort oder der Quelle, transformiert/filtert oder verbindet die Daten mithilfe von Operatoren oder Funktionen und speichert die Daten im Ausgabe-Stream oder Datenspeicherort oder in der Senke.

Das folgende Architekturdiagramm zeigt einige der unterstützten Quellen und Senken für die Kinesis-Data-Analytics-Flink-Anwendung. Zusätzlich zu den vorgebündelten Konnektoren für Quelle/Senke können Sie auch benutzerdefinierte Konnektoren zu einer Vielzahl anderer Quellen/Senken für Flink-Anwendungen in Kinesis Data Analytics einbringen.



Apache-Flink-Anwendung in Kinesis Data Analytics für Datenstromverarbeitung in Echtzeit

Entwickler können ihre bevorzugte IDE verwenden, um Flink-Anwendungen zu entwickeln und sie über die AWS Management Console oder DevOps-Tools in Kinesis Data Analytics bereitzustellen.

Amazon Kinesis Data Analytics Studio

Als Teil des Kinesis Data Analytics Service steht Kunden Kinesis Data Analytics Studio zur Verfügung, um Datenströme interaktiv und in Echtzeit abzufragen und Datenstromverarbeitungsanwendungen mit SQL, Python und Scala einfach zu erstellen und auszuführen. Studio-Notebooks werden von Apache Zeppelin unterstützt.

Mit Studio-Notebook haben Sie die Möglichkeit, Ihren Flink-Anwendungscode in einer Notebook-Umgebung zu entwickeln, sich die Ergebnisse Ihres Codes in Echtzeit anzeigen zu lassen und in Ihrem Notebook zu visualisieren. Sie können ein von Apache Zeppelin und Apache Flink unterstütztes Studio-Notebook mit einem einzigen Klick von Kinesis Data Streams und der Amazon MSK-Konsole aus erstellen oder es von der Kinesis-Data-Analytics-Konsole aus starten.

Sobald Sie den Code iterativ als Teil des Kinesis Data Analytics Studio entwickelt haben, können Sie ein Notebook als Kinesis-Datenanalytik-Anwendung bereitstellen, um es kontinuierlich im Streaming-Modus auszuführen, Daten aus Ihren Quellen zu lesen, in Ihre Ziele zu schreiben, einen Anwendungsstatus langfristig beizubehalten und automatisch basierend auf dem Durchsatz Ihrer Quell-Streams zu skalieren. Bislang verwendeten Kunden für solche interaktiven Analysen von Echtzeit-Streaming-Daten in AWS Kinesis Data Analytics für SQL-Anwendungen.

Kinesis Data Analytics für SQL-Anwendungen ist zwar weiterhin verfügbar, für neue Projekte empfiehlt AWS jedoch die Verwendung des neuen Kinesis Data Analytics Studio. Kinesis Data Analytics Studio kombiniert Benutzerfreundlichkeit mit fortschrittlichen Analysefunktionen und bietet die Möglichkeit, in wenigen Minuten anspruchsvolle Stream-Verarbeitungsanwendungen zu erstellen.

Um die Kinesis-Data-Analytics-Flink-Anwendung fehlertolerant zu gestalten, können Sie die Prüfpunkterstellung und Snapshots verwenden, wie im Abschnitt Implementieren der Fehlertoleranz in Kinesis Data Analytics für Apache Flink beschrieben.

Kinesis-Data-Analytics-Flink-Anwendungen sind nützlich für das Schreiben komplexer Streaming-Analytik-Anwendungen, wie z. B. Anwendungen mit „Genau Einmal“-Semantik der Datenverarbeitung, Prüfpunkterstellungsfunktionen und die Verarbeitung von Daten aus Datenquellen wie Kinesis Data Streams, Kinesis Data Firehose, Amazon MSK, Rabbit MQ und Apache Cassandra einschließlich benutzerdefinierter Konnektoren.

Nach der Verarbeitung von Streaming-Daten in der Flink-Anwendung können Sie die Daten in verschiedenen Senken oder Zielen wie Amazon Kinesis Data Streams, Amazon Kinesis Data Firehose, Amazon DynamoDB, Amazon OpenSearch Service, Amazon Timestream, Amazon S3 usw. persistieren. Die Kinesis-Data-Analytics-Flink-Anwendung bietet außerdem Leistungsgarantien in Sekundenbruchteilen.

Apache-Beam-Anwendungen für Kinesis Data Analytics

Apache Beam ist ein Programmiermodell für die Verarbeitung von Streaming-Daten. Apache Beam bietet eine portable API-Schicht für das Erstellen anspruchsvoller datenparalleler Verarbeitungspipelines, die über eine Vielzahl von Engines oder Runnern wie Flink, Spark Streaming, Apache Samza usw. ausgeführt werden können.

Sie können das Apache-Beam-Framework mit Ihrer Kinesis-Datenanalytik-Anwendung verwenden, um Streaming-Daten zu verarbeiten. Kinesis-Datenanalytik-Anwendungen, die Apache Beam verwenden, nutzen den Apache-Flink-Runner, um Beam-Pipelines auszuführen.

Übersicht

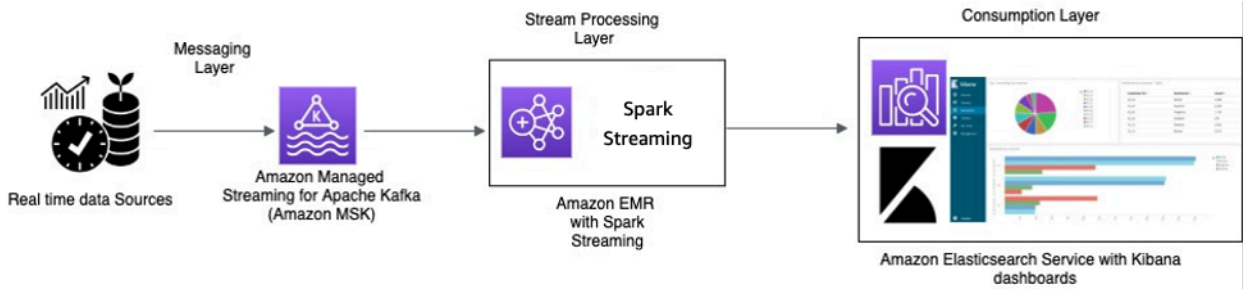
Durch die Nutzung der AWS-Streaming-Services Amazon Kinesis Data Streams, Amazon Kinesis Data Analytics und Amazon Kinesis Data Firehose

kann ABC4Logistics anomale Muster in den Temperaturmesswerten erkennen und den Fahrer und das Flottenmanagementteam in Echtzeit benachrichtigen, um schwere Unfälle wie einen kompletten Fahrzeugausfall oder einen Brand zu verhindern.

Szenario 5: Telemetriedatenüberwachung in Echtzeit mit Apache Kafka

ABC1Cabs ist ein Online-Taxibuchungsunternehmen. Alle Taxis sind mit IoT-Geräten ausgestattet, die Telemetriedaten von den Fahrzeugen sammeln. Derzeit betreibt ABC1Cabs Apache-Kafka-Cluster, die für die Nutzung von Echtzeit-Ereignissen, die Erfassung von Systemzustandsmetriken und die Verfolgung von Aktivitäten ausgelegt sind und die Daten in die auf einem On-Premises-Hadoop-Cluster aufgebaute Apache-Spark-Streaming-Plattform einspeisen.

ABC1Cabs verwendet OpenSearch Dashboards für Geschäftsmetriken, Debugging, Warnungen und das Erstellen anderer Dashboards. Das Unternehmen interessiert sich für den Einsatz von Amazon MSK, Amazon EMR mit Spark Streaming und OpenSearch Service mit OpenSearch Dashboards. Der Verwaltungsaufwand für die Wartung von Apache-Kafka- und Hadoop-Clustern soll reduziert und gleichzeitig vertraute Open-Source-Software und APIs für die Orchestrierung seiner Datenpipeline genutzt werden. Das folgende Architekturdiagramm zeigt seine Lösung in AWS.



Echtzeitverarbeitung mit Amazon MSK und Datenstromverarbeitung mit Apache Spark Streaming in Amazon EMR und Amazon OpenSearch Service mit OpenSearch Dashboards

Die IoT-Geräte im Taxi sammeln Telemetriedaten und senden sie an einen Quell-Hub. Der Quell-Hub ist so konfiguriert, dass er Daten in Echtzeit an Amazon MSK sendet. Mithilfe der APIs der Apache-Kafka-Produzentenbibliothek wird Amazon MSK so konfiguriert, dass die Daten in einen Amazon-EMR-Cluster gestreamt werden. Auf dem Amazon EMR-Cluster sind ein Kafka-Client und Spark Streaming installiert, um die Datenströme nutzen und verarbeiten zu können.

Spark Streaming verfügt über Senken-Konnektoren, die Daten direkt in definierte Indizes von Elasticsearch schreiben können. Elasticsearch-Cluster mit OpenSearch Dashboards können für Metriken und Dashboards verwendet werden. Amazon MSK, Amazon EMR mit Spark Streaming und OpenSearch Service mit OpenSearch Dashboards sind allesamt verwaltete Services, bei denen AWS den undifferenzierten Aufwand der Infrastrukturverwaltung verschiedener Cluster übernimmt, sodass Sie Ihre Anwendung mit vertrauter Open-Source-Software mit wenigen Klicks erstellen können. Im nächsten Abschnitt werden diese Services näher beleuchtet.

Amazon Managed Streaming for Apache Kafka (Amazon MSK)

Apache Kafka ist eine Open-Source-Plattform, mit der Kunden Streaming-Daten wie Clickstream-Ereignisse, Transaktionen, IoT-Ereignisse sowie Anwendungs- und Maschinenprotokolle erfassen können. Mit diesen Informationen können Sie Anwendungen entwickeln, die Echtzeit-Analysen durchführen, kontinuierliche Transformationen ausführen und diese Daten in Echtzeit an Data Lakes und Datenbanken verteilen.

Sie können Kafka als Streaming-Datenspeicher verwenden, um Anwendungen von Produzenten und Verbrauchern zu entkoppeln und eine zuverlässige Datenübertragung zwischen den beiden Komponenten zu ermöglichen. Zwar ist Kafka eine gängige Daten-Streaming- und Messaging-Plattform für Unternehmen, es kann sich jedoch schwierig gestalten, sie in der Produktion einzurichten, zu skalieren und zu verwalten.

Amazon MSK übernimmt diese Verwaltungsaufgaben und erleichtert das Einrichten, Konfigurieren und Ausführen von Kafka zusammen mit Apache Zookeeper in einer Umgebung, die den bewährten Methoden für hohe Verfügbarkeit und Sicherheit entspricht. Sie können weiterhin Kafkas Operationen auf der Steuer- und Datenebene verwenden, um die Erzeugung und den Verbrauch von Daten zu verwalten.

Da Amazon MSK Open-Source-Apache Kafka ausführt und verwaltet, können Kunden bestehende Apache-Kafka-Anwendungen einfach auf AWS migrieren und ausführen, ohne Änderungen am Anwendungscode vornehmen zu müssen.

Skalierung

Amazon MSK bietet Skalierungsoperationen an, sodass der Benutzer den Cluster aktiv skalieren kann, während er ausgeführt wird. Während des Erstellens eines Amazon-MSK-Clusters können Sie beim Start des Clusters den Instance-Typ der Broker angeben. Sie können mit wenigen Brokern innerhalb eines Amazon-MSK-Clusters beginnen. Anschließend können Sie mithilfe der AWS Management Console oder AWS CLI bis zu Hunderten von Brokern pro Cluster skalieren.

Alternativ können Sie Ihre Cluster skalieren, indem Sie die Größe oder Familie Ihrer Apache-Kafka-Broker ändern. Das Ändern der Größe oder der Familie Ihrer Broker gibt Ihnen die Flexibilität, die Datenverarbeitungskapazität Ihres Amazon-MSK-Clusters an Änderungen Ihrer Workloads anzupassen. Verwenden Sie die Tabelle Amazon MSK: Dimensionierung und Preise (Dateidownload), um die richtige Anzahl von Brokern für Ihren Amazon-MSK-Cluster zu ermitteln. Diese Tabelle enthält eine Schätzung für die Dimensionierung eines Amazon-MSK-Clusters und die damit verbundenen Kosten von Amazon MSK im Vergleich zu einem ähnlichen, selbstverwalteten, EC2-basierten Apache-Kafka-Cluster.

Nach dem Erstellen des Amazon-MSK-Clusters können Sie den EBS-Speicherplatz pro Broker erhöhen, jedoch nicht verringern. Während dieses Skalierungsvorgangs bleiben Speichervolumen verfügbar. Es werden zwei Arten von Skalierungsoperationen angeboten: Automatische Skalierung (Auto Scaling) und Manuelle Skalierung (Manual Scaling).

Amazon MSK unterstützt die automatische Erweiterung des Clusterspeichers als Reaktion auf eine erhöhte Nutzung mithilfe von Auto-Scaling-Anwendungsrichtlinien. Ihre automatische Skalierungsrichtlinie legt die Auslastung der Zielfestplatte und die maximale Skalierungskapazität fest.

Der Schwellenwert für die Speicherauslastung hilft Amazon MSK dabei, eine automatische Skalierungsoperation auszulösen. Um den Speicherplatz durch manuelle Skalierung zu erhöhen,

warten Sie, bis sich der Cluster im Zustand `ACTIVE` befindet. Die Speicherskalierung weist eine Ruhephase von mindestens sechs Stunden zwischen den Ereignissen auf. Auch wenn durch den Vorgang sofort zusätzlicher Speicherplatz verfügbar gemacht wird, führt der Service Optimierungen an Ihrem Cluster durch, die bis zu 24 Stunden oder länger dauern können.

Die Dauer dieser Optimierungen ist proportional zu Ihrer Speichergröße. Darüber hinaus bietet der Service auch die Replikation in mehreren Availability Zones innerhalb einer AWS-Region an, um die Hochverfügbarkeit zu gewährleisten.

Konfiguration

Amazon MSK bietet eine Standardkonfiguration für Broker, Themen und Apache-ZooKeeper-Knoten. Ebenso können Sie benutzerdefinierte Konfigurationen erstellen und sie verwenden, um neue Amazon-MSK-Cluster zu erstellen oder vorhandene Cluster zu aktualisieren. Wenn Sie einen MSK-Cluster erstellen, ohne eine benutzerdefinierte Amazon-MSK-Konfiguration anzugeben, erstellt und verwendet Amazon MSK eine Standardkonfiguration. Eine Liste dieser Standardwerte finden Sie unter Konfiguration von Apache Kafka.

Zu Überwachungszwecken sammelt Amazon MSK Apache-Kafka-Metriken und sendet sie an Amazon CloudWatch, wo Sie sich diese anzeigen lassen können. Die Metriken, die Sie für Ihren MSK-Cluster konfigurieren, werden automatisch gesammelt und an CloudWatch übergeben. Durch die Überwachung der Verzögerung bei den Verbrauchern können Sie langsame oder hängen gebliebene Verbraucher identifizieren, die bei den neuesten verfügbaren Daten zu einem Thema nicht auf dem aktuellen Stand sind. Bei Bedarf können Sie dann Abhilfemaßnahmen ergreifen, z. B. die Skalierung oder den Neustart dieser Verbraucher vornehmen.

Migration zu Amazon MSK

Die Migration von On-Premises zu Amazon MSK kann mit einer der folgenden Methoden durchgeführt werden.

- MirrorMaker2.0 – MirrorMaker2.0 (MM2) ist eine Multi-Cluster-Datenreplikations-Engine, die auf dem Apache-Kafka-Connect Framework basiert.. MM2 ist eine Kombination aus einem Apache-Kafka-Quellenkonnektor und einem Senkenkonnektor. Sie können einen einzelnen MM2-Cluster verwenden, um Daten zwischen mehreren Clustern zu migrieren. MM2 erkennt automatisch neue Themen und Partitionen und stellt gleichzeitig sicher, dass die Themenkonfigurationen zwischen den Clustern synchronisiert werden. MM2 unterstützt Migrations-ACLs, Themenkonfigurationen und Offset-Übersetzung. Weitere Einzelheiten zur Migration finden Sie unter Migrieren von Clustern

mit Apache Kafkas MirrorMaker. MM2 wird für Anwendungsfälle im Zusammenhang mit der Replikation von Themenkonfigurationen und der automatischen Offset-Übersetzung verwendet.

- Apache Flink – MM2 unterstützt mindestens Semantik, die genau einmal abgeschlossen wird. Datensätze können am Zielort dupliziert werden, und Verbrauchern müssen idempotent sein, um doppelte Datensätze verarbeiten zu können. In Szenarien mit „Genau Einmal“-Semantik müssen Kunden Apache Flink verwenden können. Es bietet eine Alternative, um „Genau Einmal“-Semantik zu erreichen.

  Apache Flink kann auch für Szenarien verwendet werden, in denen Daten vor der Übermittlung an den Zielcluster Mapping- oder Transformationsaktionen erfordern. Apache Flink bietet Konnektoren für Apache Kafka mit Quellen und Senken, die Daten von einem Apache-Kafka-Cluster lesen und in einen anderen schreiben können. Apache Flink kann in AWS ausgeführt werden, indem ein Amazon-EMR-Cluster gestartet wird oder indem Apache Flink als Anwendung mit Amazon Kinesis Data Analytics ausgeführt wird.
- AWS Lambda – Mit der Unterstützung von Apache Kafka als Ereignisquelle für AWS Lambda können Kunden jetzt Nachrichten aus einem Thema über eine Lambda-Funktion verwenden. Der AWS Lambda-Service fragt intern nach neuen Datensätzen oder Nachrichten von der Ereignisquelle ab und ruft dann synchron die Ziel-Lambda-Funktion auf, um diese Nachrichten zu verwenden. Lambda liest die Nachrichten in Batches und stellt die Nachrichten-Batches Ihrer Funktion in der Ereignisnutzlast zur Verarbeitung zur Verfügung. Die verwendeten Nachrichten können dann transformiert und/oder direkt in den Ziel-Amazon-MSK-Cluster geschrieben werden.

Amazon EMR mit Spark Streaming

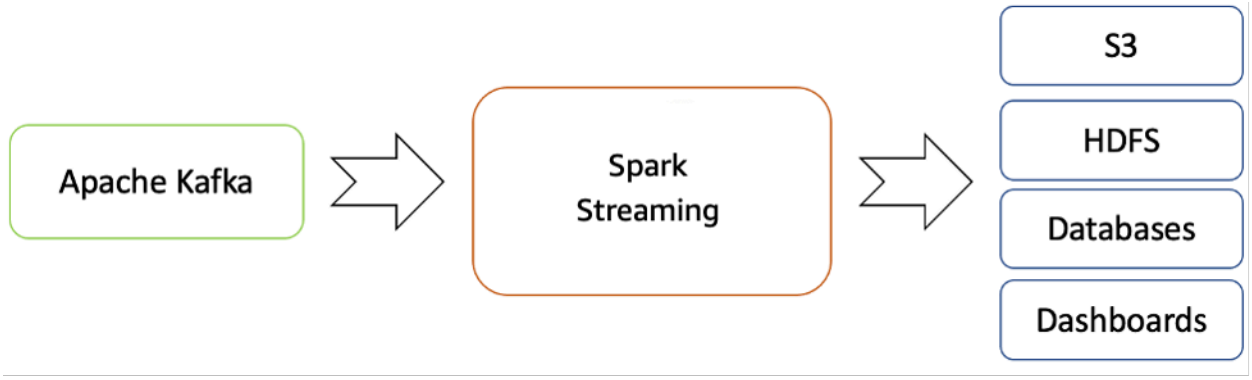
Bei Amazon EMR handelt es sich um eine verwaltete Cluster-Plattform, die die Ausführung von Big-Data-Frameworks in AWS wie Apache Hadoop und Apache Spark vereinfacht, um riesige Datenmengen zu verarbeiten und zu analysieren.

Amazon EMR bietet die Fähigkeiten von Spark und kann zum Starten von Spark Streaming verwendet werden, um Daten aus Kafka zu nutzen. Spark Streaming ist eine Erweiterung der Spark-Kern-API, die eine skalierbare, durchsatzstarke und fehlertolerante Datenstromverarbeitung von Live-Datenströmen ermöglicht.

Sie können einen Amazon-EMR-Cluster mit der AWS Command Line Interface (AWS CLI) oder in der AWS Management Console erstellen und beim Erstellen des Clusters Spark und Zeppelin in den erweiterten Konfigurationen auswählen. Wie im folgenden Architekturdiagramm dargestellt, können Daten aus vielen Quellen wie Apache Kafka und Kinesis Data Streams eingespeist und mit

komplexen Algorithmen verarbeitet werden, die durch High-Level-Funktionen wie Map, Reduce, Join und Window ausgedrückt werden. Weitere Informationen finden Sie unter Transformationen in DStreams.

Die verarbeiteten Daten können an Dateisysteme, Datenbanken und Live-Dashboards weitergeleitet werden.



Echtzeit-Streaming-Fluss von Apache Kafka zur Hadoop-Umgebung

Apache Spark Streaming verfügt standardmäßig über ein Micro-Batch-Ausführungsmodell. Seit dem Erscheinen von Spark 2.3 hat Apache jedoch einen neuen Verarbeitungsmodus mit niedriger Latenz namens Continuous Processing eingeführt und End-to-End-Latenzen von nur einer Millisekunde mit At-Least-Once-Garantien erreichen kann.

Ohne die Dataset/DataFrames-Operationen in Ihren Abfragen zu ändern, können Sie den Modus entsprechend den Anforderungen Ihrer Anwendung wählen. Einige der Vorteile von Spark Streaming sind:

- Es bringt die sprachintegrierte API von Apache Spark in die Datenstromverarbeitung ein, sodass Sie Streaming-Aufträge auf die gleiche Weise schreiben können wie Batch-Aufträge.
- Es unterstützt Java, Scala und Python.
- Es kann sowohl verlorene Arbeit als auch den Operator-Zustand (z. B. Schiebefenster) wiederherstellen, ohne dass Sie einen zusätzlichen Code eingeben müssen.
- Durch das Ausführen auf Spark ermöglicht Spark Streaming die Wiederverwendung desselben Codes für die Batch-Verarbeitung, das Verknüpfen von Streams mit historischen Daten oder das Ausführen von Ad-hoc-Abfragen auf dem Stream-Status und das Erstellen leistungsstarker interaktiver Anwendungen, nicht nur für die Analytik.
- Nachdem der Datenstrom mit Spark Streaming verarbeitet wurde, kann der OpenSearch-Senken-Konnektor verwendet werden, um Daten in den OpenSearch Service-Cluster zu schreiben, und

im Gegenzug kann OpenSearch Service mit OpenSearch Dashboards als Verbrauchsebene verwendet werden.

Amazon OpenSearch Service mit OpenSearch Dashboards

OpenSearch Service ist ein verwalteter Service für das einfache Bereitstellen, Betreiben und Skalieren von OpenSearch-Clustern in der AWS Cloud. OpenSearch ist eine beliebte Such- und Analyse-Engine für Anwendungsfälle wie beispielsweise Analysen, Anwendungsüberwachung in Echtzeit und Clickstream-Analysen.

OpenSearch Dashboards ist ein Open-Source-Datenvisualisierungs- und -explorationstool, das für Protokoll- und Zeitreihenanalysen, Anwendungsüberwachung und Operational-Intelligence-Anwendungsfälle eingesetzt wird. Es bietet leistungsstarke und benutzerfreundliche Funktionen wie Histogramme, Liniendiagramme, Kuchendiagramme, Heatmaps und integrierte Geodatenunterstützung.

OpenSearch Dashboards bietet eine enge Integration in OpenSearch, einer gängigen Analyse- und Suchmaschine, die OpenSearch Dashboards zur Standardwahl für die Visualisierung von in OpenSearch gespeicherten Daten macht. OpenSearch Service bietet eine Installation von OpenSearch Dashboards mit jeder OpenSearch Service-Domäne. Einen Link zu OpenSearch Dashboards finden Sie auf Ihrem Domain-Dashboard in der OpenSearch Service-Konsole.

Übersicht

Mit Apache Kafka, das als verwalteter Service in AWS angeboten wird, können Sie sich auf die Nutzung statt auf die Verwaltung der Koordination zwischen den Brokern konzentrieren, was in der Regel ein umfassendes Verständnis von Apache Kafka erfordert. Funktionen wie Hochverfügbarkeit, Broker-Skalierbarkeit und differenzierte Zugriffskontrolle werden von der Amazon-MSK-Plattform verwaltet.

Das Unternehmen ABC1Cabs nutzte diese Services, um eine Produktionsanwendung zu erstellen, für die kein Fachwissen in Bezug auf die Infrastrukturverwaltung erforderlich ist. Es konnte sich auf die Verarbeitungsebene konzentrieren, um Daten von Amazon MSK zu nutzen und an die Visualisierungsebene weiterzuleiten.

Spark Streaming auf Amazon EMR kann die Echtzeitanalyse von Streaming-Daten und die Veröffentlichung auf OpenSearch-Dashboards in Amazon OpenSearch Service für die Visualisierungsebene unterstützen.

Fazit und Mitwirkende

Fazit

In diesem Dokument wurden mehrere Szenarien für Streaming-Workflows untersucht. In diesen Szenarien bot die Streaming-Datenverarbeitung den Beispielunternehmen die Möglichkeit, neue Merkmale und Funktionen hinzuzufügen.

Durch die Analyse der Daten bei ihrer Erstellung erhalten Sie Erkenntnisse über die aktuellen Aktivitäten Ihres Unternehmens. AWS-Streaming-Services ermöglichen es Ihnen, sich auf Ihre Anwendung zu konzentrieren und zeitkritische Geschäftsentscheidungen zu treffen, anstatt die Infrastruktur bereitzustellen und zu verwalten

Mitwirkende

- Amalia Rabinovitch, Sr. Solutions Architect, AWS
- Priyanka Chaudhary, Data Lake, Datenarchitektin, AWS
- Zohair Nasimi, Lösungsarchitekt, AWS
- Rob Kuhr, Lösungsarchitekt, AWS
- Ejaz Sayyed, Sr. Architekt für Partnerlösungen, AWS
- Allan MacInnis, Lösungsarchitekt, AWS
- Chander Matrubhutam, Product Marketing Manager, AWS

Am Dokument vorgenommene Änderungen

Abonnieren Sie den RSS-Feed, um über Aktualisierungen des Whitepapers benachrichtigt zu werden.

Update-Historie-Änderung	Update-Historie-Beschreibung	Update-Historie-Datum
Aktualisiert	Für technische Genauigkeit aktualisiert	1. September 2021
Erstveröffentlichung	Erstveröffentlichung des Whitepapers	1. Juli 2017