Streaming-Daten-Lösungen in AWS mit Amazon Kinesis

Juli 2017

Hinweise

Dieses Dokument wird nur zu Informationszwecken zur Verfügung gestellt. Es stellt das aktuelle Produktangebot und die Verfahren von AWS zum Ausstellungsdatum dieses Dokuments dar. Änderungen vorbehalten. Kunden sind für ihre eigene unabhängige Einschätzung der Informationen in diesem Dokument und jedwede Nutzung der AWS-Services verantwortlich. Jeder Service wird „wie besehen“ ohne Gewähr und ohne Garantie jeglicher Art, weder ausdrücklich noch impliziert, bereitgestellt. Dieses Dokument gibt keine Garantien, Gewährleistungen, vertraglichen Verpflichtungen, Bedingungen oder Zusicherungen von AWS, seinen Partnern, Zulieferern oder Lizenzgebern. Die Verantwortung und Haftung von AWS gegenüber seinen Kunden werden durch AWS-Vereinbarungen geregelt. Dieses Dokument ist weder ganz noch teilweise Teil der Vereinbarungen von AWS mit seinen Kunden und ändert diese Vereinbarungen auch nicht.

Inhalt

Kurzbeschreibung

Dateningenieure, Datenanalysten und Entwickler arbeiten daran, ihre Analysen von Stapelanalysen zu Echtzeitsanalysen weiterzuentwickeln, damit ihre Unternehmen erfahren können, was ihre Kunden, Anwendungen und Produkte zum genauen Zeitpunkt tun und so umgehend zu reagieren. In diesem Whitepaper wird die Entwicklung von Stapelanalysen zu Echtzeitanalysen behandelt. Es wird beschrieben, wie Services wie Amazon Kinesis Streams, Amazon Kinesis Firehose und Amazon Kinesis Analytics verwendet werden können, um Echtzeit-Anwendungen zu implementieren, und es werden Ihnen häufig verwendete Vorgehen dieser Dienstleistungen vorgestellt.

Einleitung

Durch das explosive Wachstum von Datenquellen, die kontinuierliche Daten-Streams generieren, empfangen Unternehmen heutzutage riesige Datenmengen in enormer Geschwindigkeit. Ob es sich um Protokolldaten von Anwendungs-Servern, Clickstream-Daten von Websites und mobilen Apps, Daten von IoT-Geräten (Internet der Dinge) oder Telemetrie-Geräten handelt, alle enthalten Informationen, die Ihnen helfen, mehr darüber zu erfahren, was Ihre Kunden, Anwendungen und Produkte jetzt gerade tun. Diese Daten in Echtzeit zu verarbeiten und zu analysieren ist wichtig, um Aufgaben wie die kontinuierliche Überwachung zu erledigen, um sicherzustellen, dass Ihre Anwendungen mit hoher Verfügbarkeit laufen und Service-Angebote und Produktempfehlungen personalisiert sind. Echtzeit-Verarbeitung kann auch in andere gängige Anwendungsfälle eingesetzt werden, z. B. bei Website-Analysen und maschinellem Lernen, und ist genauer und prozessfähiger, indem Daten in Sekunden oder Minuten zur Verfügung stehen und nicht erst nach Stunden oder Tagen.

Echtzeit-Anwendungsszenarien

Es gibt zwei Arten von Anwendungsfall-Szenarien für Streaming-Data-Anwendungen:

- **Weiterentwicklung von Batch-Analysen zu Streaming-Analysen**

Sie können Echtzeit-Analysen für Daten durchführen, die üblicherweise mit der Batch-Verarbeitung in Data Warehouses oder mithilfe von Hadoop-Frameworks durchgeführt werden. Die häufigsten Anwendungsfälle finden sich bei Data Lakes, Data Science und bei maschinellem Lernen. Sie können Streaming-Lösungen verwenden, um Daten in Echtzeit kontinuierlich in Ihre Data Lakes zu laden. Sie können auch Modelle für maschinelles Lernen regelmäßiger aktualisieren, wenn neue Daten häufiger verfügbar sind, wodurch die Genauigkeit und Zuverlässigkeit des Outputs unterstützt werden. Zillow verwendet Amazon Kinesis Streams zum Beispiel zum Sammeln öffentlicher Daten und MLS Listings und bieten so Käufern und Verkäufern aktuelle Wertschätzungen nahezu in Echtzeit. Zillow sendet dieselben Daten an Amazon Simple Storage Service (S3) Data Lakes mit Kinesis Streams, damit alle Anwendungen mit den neuesten Informationen arbeiten können.

- **Die Bereitstellung von Echtzeit-Anwendungen**

Sie können Streaming-Daten-Services für Echtzeit-Anwendungen wie Anwendungsüberwachung, Betrugserkennung und Live-Bestenlisten nutzen. Diese Anwendungsfälle erfordern Millisekunden End-to-End-Latenzen – von der Dateneingabe bis zur Verarbeitung bis hin zur Übertragung der Ergebnisse an den Zielspeicherort und an andere Systeme. Netflix nutzt beispielsweise Kinesis Streams, um die Kommunikation zwischen allen Anwendungen zu überwachen, damit diese Probleme schnell erkannt und behoben werden können, wodurch die hohe Service Uptime und eine hohe Verfügbarkeit für Kunden sichergestellt wird. Während der am häufigsten anwendbare Anwendungsfall die Überwachung der Anwendungsleistung ist, gibt es eine zunehmende Anzahl von Echtzeitanwendungen in den Bereichen Ad-Tech, Spiele und IoT, die in diese Kategorie fallen.

Unterschied zwischen Batch- und Stream-Verarbeitung

Sie benötigen andere Tools zum Sammeln, Vorbereiten und Verarbeiten von Streaming-Daten in Echtzeit als die Tools, die Sie bisher für die Batch-Analyse verwendet haben. Bei herkömmlichen Analysen sammeln Sie die Daten, laden sie regelmäßig in eine Datenbank und analysieren sie Stunden, Tage oder Wochen später. Die Analyse von Echtzeitdaten erfordert einen anderen Ansatz. Statt Datenbankabfragen über gespeicherte Daten auszuführen, verarbeiten Stream-Verarbeitungsanwendungen die Daten kontinuierlich in Echtzeit, sogar noch bevor sie gespeichert werden. Streaming-Daten können in atemberaubendem Tempo ankommen und die Datenmengen können jederzeit nach oben oder unten variieren. Stream-Datenverarbeitungsplattformen müssen in der Lage sein, die Geschwindigkeit und Variabilität eingehender Daten zu bewältigen und sie bei ihrem Eintreffen zu verarbeiten, oft Millionen bis Hunderte von Millionen von Ereignissen pro Stunde.

Herausforderungen der Stream-Verarbeitung

Durch die Verarbeitung von Echtzeitdaten bei ihrer Ankunft können Sie viel schneller Entscheidungen treffen, als dies mit herkömmlichen Datenanalysetechnologien möglich ist. Das Erstellen und Betreiben eigener angepasster Streaming-Daten-Pipelines ist jedoch kompliziert und ressourcenintensiv. Sie müssen ein System aufbauen, mit dem Daten aus

Tausenden von Datenquellen gesammelt, vorbereitet und gleichzeitig übertragen werden können. Sie müssen die Speicher- und Rechenressourcen genau abstimmen, damit die Daten für maximalen Durchsatz und niedrige Latenz effizient gestapelt und übertragen werden. Sie müssen eine regelrechte Server-Flotte bereitstellen und verwalten, um das System zu skalieren, damit Sie mit den unterschiedlichen Datengeschwindigkeiten umgehen können. Nachdem Sie diese Plattform erstellt haben, müssen Sie das System überwachen und nach allen Server- oder Netzwerkfehlern wiederherstellen, indem Sie die Datenverarbeitung ab der entsprechenden Stelle im Stream abholen, ohne doppelte Daten zu erstellen. All das kostet wertvolle Zeit und viel Geld und am Ende des Tages kommen die meisten Unternehmen einfach nicht umhin, sich mit dem Status Quo zufrieden geben zu müssen und ihr Geschäft mit Informationen zu betreiben, die Stunden oder Tage alt sind.

Von Batch zu Echtzeit: Ein Beispiel

Um besser zu verstehen, wie sich Organisationen von der Batch-Verarbeitung zur Stream-Verarbeitung mit AWS bewegen, lassen Sie uns ein Beispiel durchgehen. In diesem Beispiel überprüfen wir ein Szenario und besprechen ausführlich, wie AWS-Services Amazon Kinesis Streams,[1] Amazon Kinesis Firehose[2] und Amazon Kinesis Analytics[3] verwendet werden, um das Problem zu lösen.

Batch-Verarbeitung ist eine gängige Praxis für die Datenverarbeitung. Unternehmen führen häufig reguläre Aufträge durch, um ihre Daten in einer für ihren Anwendungsfall geeigneten Häufigkeit zu analysieren. Beispielsweise könnte eine Organisation am Ende des Monats einen Prozess ausführen, um zu ermitteln, wie viel für jeden ihrer Kunden berechnet werden soll. Oder sie könnten einen stündlichen Auftrag ausführen, um Protokolle von ihren IT-Anwendungen zu analysieren, um festzustellen, welche Fehler innerhalb der letzten Stunde aufgetreten sind. Während diese monatlichen oder stündlichen Prozesse wertvoll sind, was wäre, wenn die gleichen Daten bereits bei ihrer Erstellung analysiert werden könnten? Gibt es zusätzliche Erkenntnisse, die gewonnen werden könnten, oder zusätzliche Nutzen, die geschaffen werden könnten?

Betrachten Sie das monatliche Abrechnungsszenario erneut. Durch die Analyse der Nutzungsdaten eines Kunden, während sie generiert werden, kann ein Unternehmen wertvolle Funktionen nutzen, z. B. die Benachrichtigung von Benutzern, dass sie sich einem vordefinierten Abrechnungslimit nähern. Wenn

die IT-Anwendungsprotokolle in Echtzeit analysiert werden können, kann ein Systemadministrator sofort benachrichtigt werden, um den Fall zu untersuchen und korrigierende Maßnahmen zu ergreifen.

Jetzt kombinieren wir diese beiden zu einem einzelnen Szenario und überprüfen, wie wir eine Lösung erarbeiten können.

Beispielszenario: Mautabrechnung und Benachrichtigung

In diesem vereinfachten Beispiel betreibt eine fiktive Firma, ABC Tolls, Mautautobahnen im ganzen Land. Kunden, die sich bei ABC Tolls registrieren, erhalten einen Transceiver für ihr Auto. Wenn der Kunde durch die Mautkontrolle fährt, empfängt ein Sensor Informationen vom Transceiver und zeichnet die Details der Transaktion in einer relationalen Datenbank auf. ABC Tolls hat eine traditionelle Batch-Architektur. Jeden Tag wird ein geplanter ETL-Prozess (Extract-Transform-Load) ausgeführt, der die täglichen Transaktionen verarbeitet und sie so transformiert, dass sie in ihr jeweiliges Data Warehouse geladen werden können. Am nächsten Tag überprüfen die Analysten von ABC Tolls die Daten mithilfe eines Berichtstools. Darüber hinaus sammelt ein anderer Prozess einmal pro Monat (am Ende des Abrechnungszyklus) alle Transaktionen für jeden Kunden von ABC Tolls, um deren monatliche Zahlung zu berechnen.

ABC Tolls möchte einige Änderungen an diesem System vornehmen. Die erste Anforderung kommt vom Geschäftsanalysten-Team. Diese haben darum gebeten, Berichte aus ihrem Data Warehouse mit Daten zu erstellen, die nicht älter als 30 Minuten sind.

ABC Tolls entwickelt auch eine neue mobile Anwendung für seine Kunden. Bei der Entwicklung der Anwendung wurde entschieden, einige neue Funktionen zu entwickeln. Eine Funktion bietet Kunden die Möglichkeit, einen Ausgabenlimit für ihr Konto festzulegen. Wenn die kumulative Mautgebühr eines Kunden dieses Limit überschreitet, möchte ABC Tolls eine In-Application-Nachricht an den Kunden senden, um ihn innerhalb von 10 Minuten nach Auftreten der Übertretung darüber zu informieren, dass das Limit überschritten wurde.

Zu guter Letzt hat das ABC Tolls Operations-Team einige zusätzliche Anforderungen, die es in das System implementieren möchte. Sie möchten bei

der Überwachung ihrer Mautstationen sofort benachrichtigt werden, wenn der Fahrzeugverkehr einer Mautstation für jeweils 30 Minuten an einem Tag unter eine vordefinierte Schwelle fällt. Zum Beispiel wissen sie aus historischen Daten, dass eine ihrer Mautstationen mittwochs zwischen 14:00 Uhr und 14:30 Uhr etwa 360 Fahrzeuge verzeichnet. In diesem 30-Minuten-Fenster möchte das Operations-Team benachrichtigt werden, wenn die Mautstation weniger als 100 Fahrzeuge verzeichnet. Die Betreiber können dann untersuchen, um zu bestimmen, ob der Verkehr normal ist oder ob ein anderer Faktor zu dem unerwarteten Wert beigetragen hat (z. B. ein defekter Sensor oder ein Autounfall auf der Autobahn).

Das Entwicklerteam von ABC Tolls stellt fest, dass ihre aktuelle Architektur einige Modifikationen benötigt, um diese Anforderungen zu ermöglichen. Sie beschließen, ein Streaming-Datenaufnahme- und Analysesystem zu entwickeln, um diese Anforderungen zu verwirklichen. Sehen wir uns die einzelnen Anforderungen an und werfen wir einen Blick auf die Architekturverbesserungen, die diese Anforderungen ermöglichen können.

Anforderung 1: Weitere aktuelle Daten im Data Warehouse

Derzeit können die Daten im Data Warehouse von ABC Tolls aufgrund ihres täglichen Batch-Prozesses bis zu 24 Stunden alt sein. Ihre derzeitige Data Warehouse-Lösung ist Amazon Redshift. Bei der Überprüfung der Funktionen der Amazon Kinesis-Dienste wurde erkannt, dass Kinesis Firehose eine große Menge an Datensätzen erhalten und in Amazon Redshift einfügen kann. Sie haben einen Kinesis Firehose Delivery Stream erstellt und so konfiguriert, dass alle 15 Minuten Daten in ihre Amazon Redshift-Tabelle kopiert werden. Ihre aktuelle Lösung speichert Datensätze in einem Dateisystem als Teil ihres Batch-Prozesses. Im Rahmen dieser neuen Lösung nutzten sie den Amazon Kinesis Agent auf ihren Servern, um ihre Log-Daten an Kinesis Firehose weiterzuleiten. Da Kinesis Firehose Amazon S3 zum Speichern unbearbeiteter Streaming-Daten verwendet, bevor es in Amazon Redshift kopiert wird, musste ABC Tolls keine andere Lösung zum Archivieren ihrer Rohdaten erstellen.

Abbildung 1 zeigt diese Lösung.

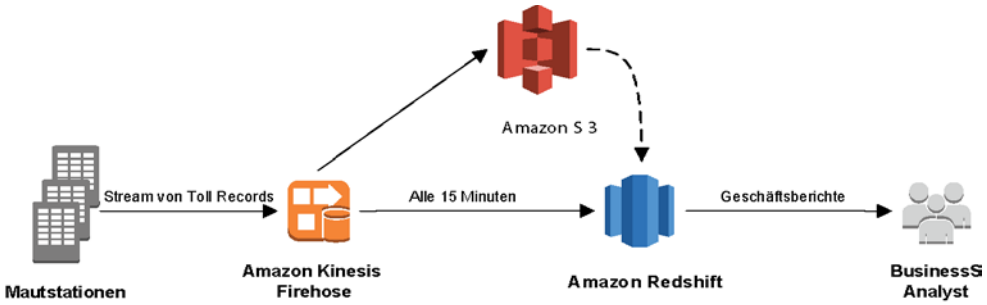

Abbildung 1: Neue Lösung mithilfe von Amazon Kinesis Firehose

Für diesen Teil der Architektur wählte ABC Tolls Kinesis Firehose. Sehen wir uns die Funktionen von Kinesis Firehose im Detail an.

Amazon Kinesis Firehose

Amazon Kinesis Firehose ist der einfachste Weg, Streaming-Daten in AWS zu laden. Er kann Daten erfassen, transformieren und Streamingdaten in Amazon Kinesis Analytics, Amazon S3, Amazon Redshift und Amazon Elasticsearch Service laden. So erhalten Sie Analysedaten für Ihre Business Intelligence-Tools und Dashboards fast in Echtzeit. Es handelt sich um einen vollständig verwalteten Dienst, der automatisch an den Durchsatz Ihrer Daten angepasst wird und keine laufende Verwaltung erfordert. Er kann die Daten vor dem Laden auch in Batches unterteilen, komprimieren und verschlüsseln, um den am Zielort verwendeten Speicherplatz zu minimieren und die Sicherheit zu erhöhen.

Kinesis Firehose ist ein vollständig gemanagter Dienst. Sie müssen keine Anwendungen schreiben oder Ressourcen verwalten. Sie konfigurieren Ihre Datenproduzenten so, dass sie Daten an Kinesis Firehose senden, die die Daten automatisch an das von Ihnen angegebene Ziel übermittelt. Sie können Kinesis Firehose auch so konfigurieren, dass Ihre Daten vor der Datenlieferung transformiert werden.

Daten an Amazon Kinesis Firehose Delivery Stream senden

Um Daten an Ihren Delivery Stream zu senden, gibt es mehrere Optionen. AWS bietet SDKs für viele gängige Programmiersprachen, von denen jede APIs für Kinesis Firehose bereitstellt. AWS hat außerdem ein Dienstprogramm erstellt, mit dem Sie Daten an Ihren Delivery Stream senden können.

Verwenden der API

Die Kinesis Firehose-API bietet zwei Möglichkeiten zum Senden von Daten an Ihren Delivery Stream. `PutRecord` sendet einen Datensatz innerhalb eines Aufrufs. `PutRecordBatch` kann mehrere Datensätze innerhalb eines Aufrufs senden.

Bei jeder Methode müssen Sie bei Verwendung der Methode den Namen des Delivery Stream und des Datensatzes oder Anordnung von Datensätzen angeben. Jeder Datensatz besteht aus einem Datenklumpen mit einer Größe von bis zu 1.000 KB und beliebigen Daten.

Ausführliche Informationen und Beispielcode für die Kinesis-Firehose-API-Vorgänge finden Sie unter Schreiben auf einen Firehose Delivery Stream mithilfe des AWS SDK.[4]

Verwendung des Amazon Kinesis Agent

Der Amazon Kinesis Agent ist eine eigenständige Java-Softwareanwendung, die eine einfache Möglichkeit zum Sammeln und Senden von Daten an Kinesis Streams und Kinesis Firehose bietet. Der Agent überwacht kontinuierlich eine Reihe von Dateien und sendet neue Daten an Ihren Stream. Der Agent übernimmt die Dateirotation, die Überprüfung und das erneute Versuchen beim Auftreten von Fehlern. Er liefert alle Ihre Daten zuverlässig, zeitnah und einfach. Außerdem werden Amazon CloudWatch-Messwerte ausgegeben, um den Streaming-Prozess besser überwachen und Fehler beheben zu können.

Sie können den Agent auf Linux-basierten Serverumgebungen wie Webservern, Protokollservern und Datenbankservern installieren. Konfigurieren Sie nach der Installation des Agenten die zu überwachenden Dateien und den Zieldatenstrom für die Daten. Nachdem der Agent konfiguriert wurde, sammelt er dauerhaft Daten aus den Dateien und sendet sie zuverlässig an den Lieferstream.

Der Agent kann mehrere Dateiverzeichnisse überwachen und in mehrere Streams schreiben. Er kann auch so konfiguriert werden, dass Dateneinträge vorverarbeitet werden, bevor sie an Ihren Stream oder den Delivery Stream gesendet werden.

Wenn Sie eine Migration von einem herkömmlichen Batch-Dateisystem zu Streaming-Daten in Erwägung ziehen, protokollieren Ihre Anwendungen möglicherweise bereits Ereignisse in Dateien auf den Dateisystemen Ihrer

Anwendungsserver. Wenn Ihre Anwendung eine beliebte Protokollierungsbibliothek (z. B. Log4j) verwendet, ist es normalerweise eine einfache Aufgabe, sie so zu konfigurieren, dass sie auf lokale Dateien schreibt. Unabhängig davon, wie die Daten in eine Protokolldatei geschrieben werden, sollten Sie den Agent in diesem Szenario verwenden. Er bietet eine einfache Lösung, die wenige oder keine Änderungen an Ihrem vorhandenen System erfordert. In vielen Fällen kann er gleichzeitig mit Ihrer vorhandenen Batch-Lösung verwendet werden. In diesem Szenario stellt er Kinesis-Streams einen Datenstrom zur Verfügung, wobei die Protokolldateien als Datenquelle für den Stream verwendet werden.

In unserem Beispielszenario entschied sich ABC Tolls dafür, den Agent zum Senden von Streaming-Daten an seinen Lieferstream zu verwenden. Sie erstellten bereits Protokolldateien. Die Weiterleitung der Protokolleinträge an Kinesis Firehose war also eine einfache Installation und Konfiguration des Agenten. Es wurde kein zusätzlicher Code benötigt, um mit dem Streaming ihrer Daten zu beginnen.

Data Transformation

In einigen Szenarien möchten Sie Ihre Streaming-Daten möglicherweise transformieren oder verbessern, bevor sie an ihr Ziel geliefert werden. Beispielsweise können Datenproduzenten in jedem Datensatz unstrukturierten Text senden, den Sie in JSON umwandeln müssen, bevor Sie ihn an Amazon Elasticsearch Service übergeben.

Um Streaming-Datentransformationen zu aktivieren, verwendet Kinesis Firehose eine Funktion, die Sie zur Transformation Ihrer Daten erstellen AWS Lambda.[5]

Data Transformation Flow

Wenn Sie die Kinesis Firehose-Datenumwandlung aktivieren, puffert Kinesis Firehose eingehende Daten bis zu 3 MB oder die Puffergröße, die Sie für den Übermittlungsstream angegeben haben, je nachdem, welcher Wert kleiner ist. Kinesis Firehose ruft dann die angegebene Lambda-Funktion mit jedem gepufferten Batch asynchron auf. Die transformierten Daten werden zur Pufferung von Lambda an Kinesis Firehose gesendet. Transformierte Daten werden an das Ziel übermittelt, wenn die angegebene Puffergröße oder das Pufferintervall erreicht ist, je nachdem, was zuerst eintritt. Abbildung 2 zeigt diesen Prozess für einen Delivery Stream, der Daten an Amazon S3 liefert.

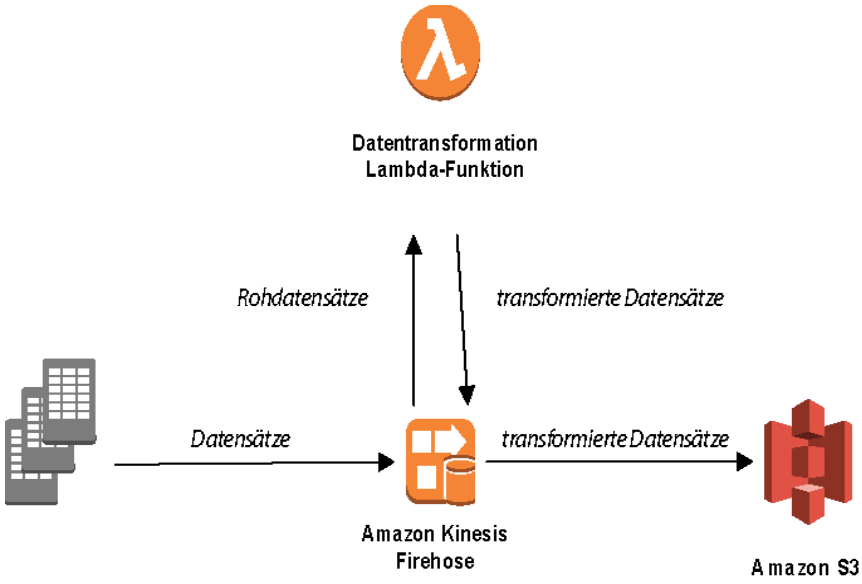

Abbildung 2: Puffern von Daten mit Kinesis Firehose- und Lambda-Funktionen

Datenlieferung

Nachdem die Pufferschwellenwerte für den Delivery Streams erreicht wurden, werden Ihre Daten an das von Ihnen konfigurierte Ziel übermittelt. Es gibt einige Unterschiede in der Art und Weise, wie Kinesis Firehose Daten an jedes Ziel liefert, die wir in den folgenden Abschnitten besprechen werden.

Amazon Simple Storage Service

Amazon Simple Storage Service (Amazon S3) ist ein Objektspeicher mit einer einfachen Web-Service-Schnittstelle zum Speichern und Abrufen einer beliebigen Datenmenge über das Internet.[6] Amazon S3 wurde für eine Datenbereitstellung von 99,999999999 % entwickelt und kann weltweit auf Milliarden Objekte skaliert werden. Sie können Amazon S3 als zentralen Speicher für native Cloud-Anwendungen wie beispielsweise Bulk-Repositories oder als „Data Lake“ für Analysen und als Ziel für Backup und Recovery und Disaster Recovery verwenden.

Datenübermittlungs-Format

Für die Datenlieferung an Amazon S3 verkettet Kinesis Firehose mehrere eingehende Datensätze basierend auf der Pufferkonfiguration Ihres Delivery Stream und liefert sie dann als S3-Objekt an Amazon S3. Möglicherweise möchten Sie am Ende jedes Datensatzes ein Datensatztrennzeichen hinzufügen, bevor Sie es an Kinesis Firehose senden, damit Sie ein geliefertes S3-Objekt in einzelne Datensätze aufteilen können.

Häufigkeit der Datenübermittlung

Die Häufigkeit der Datenübermittlung an Amazon S3 wird durch die Größe des S3-Puffers und den Pufferintervall bestimmt, den Sie für Ihren Delivery Stream konfiguriert haben. Kinesis Firehose puffert die eingehenden Daten, bevor sie an Amazon S3 übermittelt werden. Sie können die Werte für die Amazon S3-Puffergröße (1 MB bis 128 MB) oder das Pufferintervall (60 Sekunden bis 900 Sekunden) konfigurieren. Die zu erst erfüllte Bedingung löst die Datenübermittlung an Amazon S3 aus. Beachten Sie, dass Kinesis Firehose die Puffergröße dynamisch erhöht, um sicherzustellen, dass alle Daten an das Ziel gesendet werden, wenn die Datenlieferung an das Ziel hinter dem Schreiben von Daten in den Übermittlungsstream zurückbleibt.

Datenfluss

Abbildung 3 zeigt den Datenfluss für Amazon S3-Ziele.

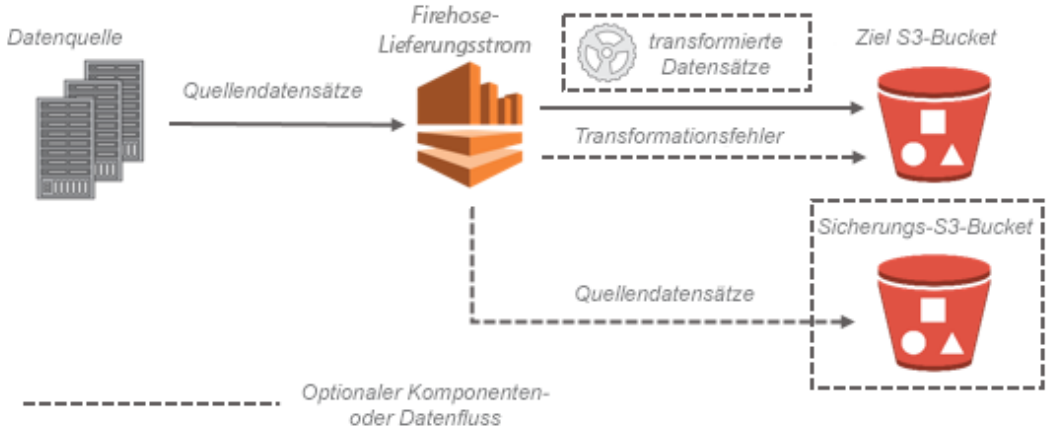

Abbildung 3: Datenfluss von Kinesis Firehose auf S3-Buckets

Amazon Redshift

Amazon Redshift ist ein schnelles, vollständig verwaltetes Data Warehouse, mit dem Sie alle Ihre Daten mithilfe von Standard-SQL und Ihren vorhandenen BI-Tools (Business Intelligence) einfach und kostengünstig analysieren können.[7] Sie können komplexe analytische Abfragen für Petabytes an strukturierten Daten ausführen, indem Sie eine anspruchsvolle Abfrageoptimierung, einen spaltenbasierten Speicher auf lokalen Hochleistungsfestplatten und eine massiv

parallele Abfrageausführung verwenden. Die meisten Ergebnisse stehen bereits innerhalb von Sekunden zur Verfügung.

In unserem Beispiel hat ABC Tolls bereits Amazon Redshift als Data-Warehouse-Lösung verwendet. Bei der Implementierung ihrer Streaming Data-Lösung konfigurierten sie ihren Delivery Stream so, dass sie ihre Streaming-Daten an ihren vorhandenen Amazon Redshift-Cluster weiterleiteten.

Datenübermittlungs-Format

Für die Datenübermittlung an Amazon Redshift liefert Kinesis Firehose eingehende Daten zunächst in dem zuvor beschriebenen Format an Ihren S3-Bucket. Kinesis Firehose gibt dann einen COPY-Befehl von Amazon Redshift aus, um die Daten aus Ihrem S3-Bucket in Ihren Amazon Redshift-Cluster zu laden. Sie müssen sicherstellen, dass das S3-Objekt in Ihren Amazon Redshift-Cluster kopiert werden kann, nachdem Kinesis Firehose mehrere eingehende Datensätze mit einem S3-Objekt verknüpft hat. Weitere Informationen finden Sie unter Amazon Redshift-Befehl COPY Data Format Parameter.

Häufigkeit der Datenübermittlung

Die Häufigkeit der COPY-Vorgänge von Amazon S3 zu Amazon Redshift hängt davon ab, wie schnell Ihr Amazon Redshift-Cluster den COPY-Befehl beenden kann. Wenn noch Daten zum Kopieren vorhanden sind, gibt Kinesis Firehose einen neuen COPY-Befehl aus, sobald der vorherige COPY-Befehl erfolgreich von Amazon Redshift beendet wurde.

Datenfluss

Abbildung 4 zeigt den Datenfluss für Amazon Redshift-Ziele.

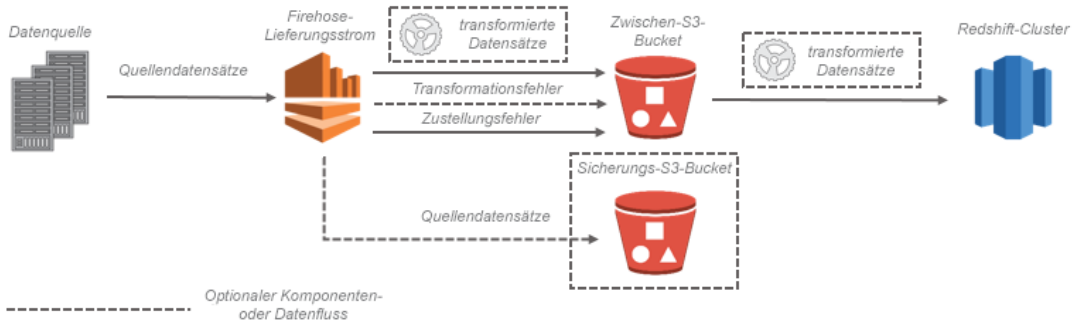

Abbildung 4: Datenfluss von Kinesis Firehose zu Amazon Redshift

Amazon Elasticsearch Service

Amazon Elasticsearch Service (Amazon ES) ist ein vollständig verwalteter Dienst, der die einfach zu verwendenden APIs und Echtzeitfunktionen von Elasticsearch sowie die für Produktionsworkloads erforderliche Verfügbarkeit, Skalierbarkeit und Sicherheit bietet.[8] Mit Amazon wird das Bereitstellen, Ausführen und Skalieren von Elasticsearch für Protokollanalysen, Volltextsuche, Anwendungsüberwachung und mehr ganz einfach.

Datenübermittlungs-Format

Für die Datenbereitstellung an Amazon ES puffert Kinesis Firehose eingehende Datensätze basierend auf der Pufferkonfiguration Ihres Übermittlungsdatenstroms und generiert anschließend eine Elasticsearch-Massenanforderung zum Indizieren mehrerer Datensätze an Ihren Elasticsearch-Cluster. Sie müssen sicherstellen, dass Ihr Datensatz UTF-8-codiert und zu einem einzeiligen JSON-Objekt abgeflacht ist, bevor Sie ihn an Kinesis Firehose senden.

Häufigkeit der Datenübermittlung

Die Häufigkeit der Datenlieferung an Amazon ES wird durch die Puffergröße und Pufferintervallwerte von Elasticsearch bestimmt, die Sie für Ihren Lieferstream konfiguriert haben. Kinesis Firehose puffert die eingehenden Daten, bevor sie an Amazon ES gesendet werden. Sie können die Werte für die Puffergröße von Elasticsearch (1 MB bis 100 MB) oder das Pufferintervall (60 Sekunden bis 900 Sekunden) konfigurieren. Die erfüllte Bedingung löst zuerst die Datenlieferung an Amazon ES aus. Beachten Sie, dass Kinesis Firehose die Puffergröße dynamisch erhöht, um sicherzustellen, dass alle Daten an das Ziel gesendet werden, wenn die Datenlieferung an das Ziel hinter dem Schreiben von Daten in den Übermittlungsstream zurückbleibt.

Datenfluss

Abbildung 5 zeigt den Datenfluss für Amazon ES-Ziele.

Abbildung 5: Bereitstellung von Daten aus Kinesis Firehose auf Amazon ES-Cluster

Zusammenfassung

Kinesis Firehose ist der einfachste Weg, um Ihre Streaming-Daten an ein unterstütztes Ziel zu übertragen. Es handelt sich um eine vollständig verwaltete Lösung, die nur wenig oder keine Entwicklung benötigt. Für ABC Tolls war die Verwendung von Kinesis Firehose eine logische Wahl. Sie nutzten bereits Amazon Redshift als ihre Data-Warehouse-Lösung. Da ihre Datenquellen kontinuierlich in Transaktionsprotokolle geschrieben wurden, konnten sie den Amazon Kinesis Agent nutzen, um diese Daten zu streamen, ohne zusätzliche Codes schreiben zu müssen.

Nachdem ABC Tolls nun einen Strom von Mautdatensätzen erstellt hat und diese Datensätze über Kinesis Firehose erhält, können sie dies als Grundlage für ihre anderen Streaming-Datenanforderungen verwenden.

Anforderung 2: Schwellenwertwarnungen bei Abrechnungen

Um die Funktion zum Senden einer Benachrichtigung zu unterstützen, wenn ein Ausgabenschwellenwert überschritten wird, hat das ABC Tolls-Entwicklungsteam eine mobile Anwendung und eine Tabelle erstellt

Amazon DynamoDB.[9] Die Anwendung ermöglicht Kunden, ihren Schwellenwert festzulegen, und die Tabelle speichert diesen Wert für jeden Kunden. Die Tabelle wird auch verwendet, um den kumulativen Betrag zu speichern, der von jedem Kunden jeden Monat ausgegeben wird. Um rechtzeitige Benachrichtigungen bereitzustellen, muss ABC Tolls den kumulativen Wert in dieser Tabelle zeitnah aktualisieren und diesen Wert mit dem Schwellenwert vergleichen, um festzustellen, ob eine Benachrichtigung an den Kunden gesendet werden soll. Da ihre Maut-Transaktionen bereits über Kinesis Firehose laufen, haben sie sich entschieden, diese Streaming-Daten als Quelle für ihre Aggregation und Alarmierung zu verwenden. Und weil Kinesis Analytics es ihnen ermöglicht hat, SQL zum Aggregieren der Streaming-Daten zu verwenden, ist es eine ideale Lösung für das Problem. In dieser Lösung summiert Kinesis Analytics den Wert der Transaktionen für jeden Kunden über einen Zeitraum von 10 Minuten (Zeitraum). Am Ende des Zeitraums sendet es die Summen an einen Kinesis-Stream. Dieser Stream ist die Ereignisquelle für eine AWS Lambda-Funktion. Die Lambda-Funktion fragt die DynamoDB-Tabelle ab, um die Schwellenwerte und die aktuelle Gesamtsumme aller Kunden abzurufen, die in der Ausgabe von Kinesis Analytics dargestellt werden. Für jeden Kunden aktualisiert die Lambda-Funktion die aktuelle Summe in DynamoDB und vergleicht die Summe mit dem Schwellenwert. Wenn der Schwellenwert überschritten wurde, verwendet das AWS SDK den Amazon Simple Notification Service (SNS), um eine Benachrichtigung an die Kunden zu senden.

In Abbildung 6 ist die Architektur für diese Lösung.

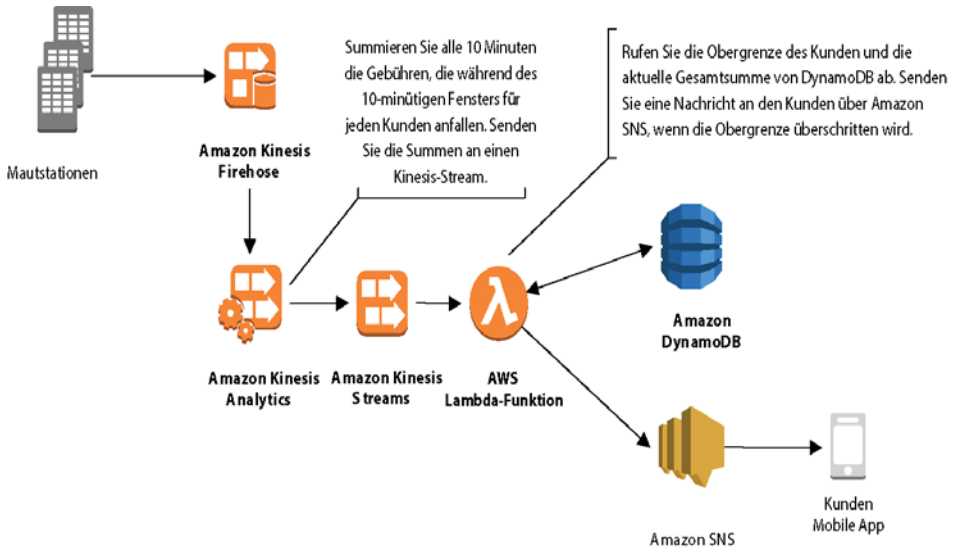

Abbildung 6: Architektur für Abrechnungsalarme und Benachrichtigungen

Mit dieser Lösung bietet ABC Tolls seinen Kunden eine zeitnahe Benachrichtigung bei Ausgabengrenzen.

Um Einblicke in Echtzeit aus ihren Streaming-Daten zu gewinnen, entschied sich ABC Tolls für die Analyse ihrer Streaming-Daten mit Kinesis Analytics. Mit Kinesis Analytics nutzten ABC Tolls die bereits vertraute Sprache SQL, um ihre Daten zu prüfen, während sie durch ihren Lieferungsstrom strömten. Sehen wir uns Kinesis Analytics genauer an.

Amazon Kinesis Analytics

Mit Kinesis Analytics können Sie Streaming-Daten mit SQL verarbeiten und analysieren. Mit diesem Dienst können Sie schnell leistungsstarke SQL-Codes für Streaming-Quellen erstellen und ausführen, um Zeitreihenanalysen durchzuführen, Echtzeit-Dashboards zu speisen und Echtzeitmetriken zu erstellen.

Um mit Kinesis Analytics zu beginnen, erstellen Sie eine Kinesis Analytics-Anwendung, die kontinuierlich Streaming-Daten liest und verarbeitet. Der Dienst unterstützt das Einlesen von Daten aus Kinesis Streams und Kinesis Firehose Streaming-Quellen. Anschließend erstellen Sie Ihren SQL-Code mit dem interaktiven Editor und testen ihn mit Live-Streaming-Daten. Sie können auch Ziele konfigurieren, an denen Kinesis Analytics die Ergebnisse beibehalten soll. Kinesis Analytics unterstützt Kinesis Firehose (Amazon S3, Amazon Redshift und Amazon Elasticsearch Service) und Kinesis Streams als Ziele.

Wichtige Konzepte

Eine *Anwendung* ist die primäre Ressource in Kinesis Analytics, die Sie in Ihrem Konto erstellen können. Kinesis Analytics-Anwendungen lesen und verarbeiten kontinuierlich Streaming-Daten in Echtzeit. Sie schreiben Anwendungscode mit SQL, um die eingehenden Streaming-Daten zu verarbeiten und eine Ausgabe zu erzeugen. Kinesis Analytics schreibt dann die Ausgabe in ein konfiguriertes Ziel. Abbildung 7 zeigt eine typische Anwendungsarchitektur.

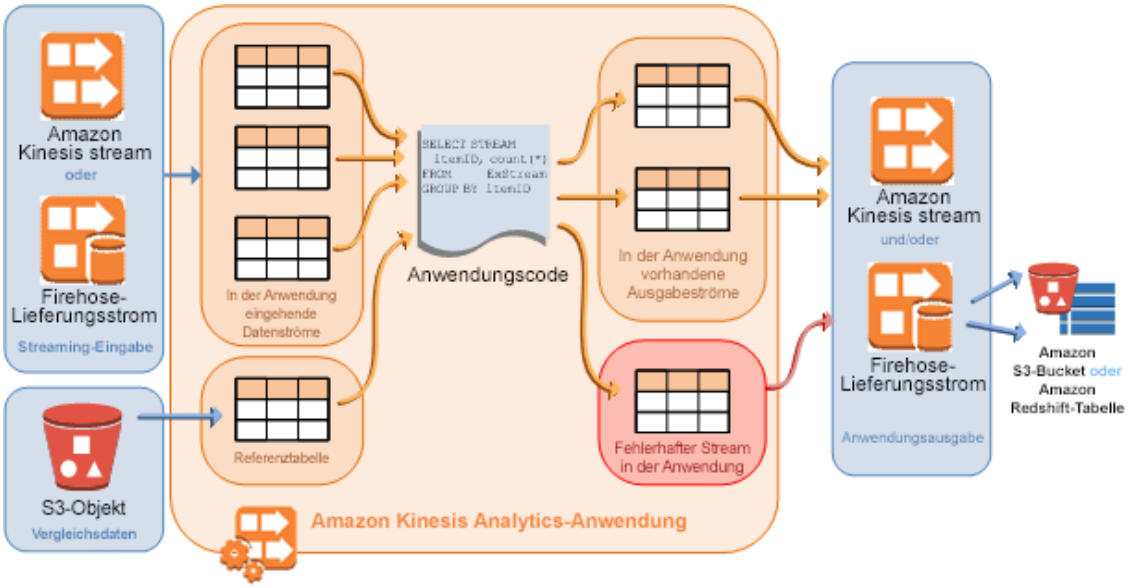

Abbildung 7: Architektur für eine Kinesis Analytics-Anwendung

Jede Anwendung hat einen Namen, eine Beschreibung, eine Versionskennung und einen Status. Beim Erstellen einer Anwendung konfigurieren Sie einfach die Eingabe, erstellen den Anwendungscode und konfigurieren die Ausgabe.

Input

Die Anwendungseingabe ist die Streaming-Quelle für Ihre Anwendung. Sie können entweder einen Kinesis-Stream oder einen Lieferstream als Streaming-Quelle auswählen. Sie können optional eine Referenzdatenquelle konfigurieren, um Ihren Eingabedatenstrom innerhalb der Anwendung anzureichern. Dies führt zu einer anwendungsbezogenen Referenztabelle. Sie müssen Ihre Referenzdaten als Objekt in einem S3-Bucket speichern. Wenn die Anwendung gestartet wird, liest Kinesis Analytics das S3-Objekt und erstellt eine Anwendungstabelle.

ABC Tolls nutzte ihren Lieferungsstrom als Eingabe für ihre Kinesis Analytics-Anwendung.

Anwendungscode

Ihr Anwendungscode besteht aus einer Reihe von SQL-Anweisungen, die Eingaben verarbeiten und Ausgaben erzeugen. Sie können SQL-Anweisungen für In-Application-Streams, Referenztabellen schreiben, und Sie können JOIN-Abfragen schreiben, um Daten aus diesen beiden Quellen zu kombinieren.

In seiner einfachsten Form kann Anwendungscode eine einzelne SQL-Anweisung sein, die aus einer Streaming-Eingabe auswählt und Ergebnisse in eine Streaming-Ausgabe einfügt. Es kann auch eine Reihe von SQL-Anweisungen sein, bei denen die Ausgabe einer Anweisung in die Eingabe der nächsten SQL-Anweisung einfließt. Darüber hinaus können Sie Anwendungscodes schreiben, um einen Eingabestream in mehrere Streams aufzuteilen und dann zusätzliche Abfragen anzuwenden, um diese Streams zu verarbeiten.

Output

In Ihrem Anwendungscode werden Abfrageergebnisse in In-Application-Streams übertragen. In Ihrem Anwendungscode können Sie einen oder mehrere In-Application-Streams für Zwischenergebnisse erstellen. Sie können dann die Anwendungsausgabe optional so konfigurieren, dass Daten in den In-Application-Streams persistent bleiben, die Ihre Anwendungsausgabe (auch als In-Application-Ausgabestreams bezeichnet) für externe Ziele enthalten. Externe Ziele können ein Lieferstream oder ein Kinesis-Stream sein.

ABC-Maut verwendet Kinesis Streams als Ziel für ihre aggregierten Werte.

Zusammenfassung

Mit Kinesis Analytics können Sie SQL verwenden, um Einblicke in Ihre Daten zu erhalten, wenn diese durch das System laufen. ABC Tolls schrieb ihr SQL, um 10-minütige Zusammenfassungen durchzuführen, um die von ihren Kunden verursachten Mautgebühren zu summieren. Die Ausgabewerte dieser 10-Minuten-Aggregationen könnten mit den Schwellenwerten ihrer Kunden verglichen werden.

Wie bereits erwähnt, gibt Kinesis Analytics seine Ergebnisse entweder an Kinesis Streams oder Kinesis Firehose weiter. In diesem Beispiel entschied sich ABC Tolls dafür, die Ausgabe von Kinesis Analytics aufgrund der Integration

von Kinesis Streams mit AWS Lambda an einen Kinesis-Stream zu senden. Lassen Sie uns Kinesis Streams etwas genauer anschauen.

Amazon Kinesis Streams

Mit Amazon Kinesis Streams können Sie benutzerdefinierte Echtzeitanwendungen mithilfe gängiger Streaming-Frameworks erstellen und Streaming-Daten in einen beliebigen Datenspeicher laden. Sie können Hunderttausende Datenproduzenten konfigurieren, um Daten kontinuierlich in einen Kinesis Stream zu übertragen, z. B. Daten von Clickstreams, Anwendungsprotokollen, IoT-Sensoren und Social Media Feeds. In weniger als einer Sekunde stehen die Daten Ihrer Anwendung zum Lesen und Verarbeiten aus dem Stream zur Verfügung.

Wenn Sie eine Lösung mit Kinesis Streams implementieren, erstellen Sie benutzerdefinierte Datenverarbeitungsanwendungen, die als *Kinesis Streams-Anwendungen* bezeichnet werden. Eine typische Kinesis Streams-Anwendung liest Daten aus einem Kinesis Stream als Datensätze.

Während Sie mit Kinesis Streams eine Vielzahl von Streaming-Datenproblemen lösen können, ist die Echtzeit-Aggregation oder -analyse von Daten und das anschließende Laden der Aggregatdaten in ein Data Warehouse- oder Map-Reduce-Cluster eine häufige Anwendung.

Die Daten werden in Kinesis Streams eingegeben, was Haltbarkeit und Elastizität gewährleistet. Die Verzögerung zwischen dem Zeitpunkt, zu dem ein Datensatz in den Stream geladen wird, und der Zeit, zu der er abgerufen werden kann (put-to-get delay), beträgt typischerweise weniger als eine Sekunde - mit anderen Worten, eine Kinesis Streams-Anwendung kann die Daten aus dem Stream konsumieren. Da Kinesis Streams ein verwalteter Service ist, entlastet er Sie von der operativen Belastung beim Erstellen und Ausführen einer Dateneingabepipeline.

Daten an Amazon Kinesis Streams senden

Es gibt mehrere Mechanismen, um Daten an Ihren Stream zu senden. AWS bietet SDKs für viele gängige Programmiersprachen, von denen jede APIs für Kinesis-Streams bereitstellt. AWS hat außerdem mehrere Dienstprogramme erstellt, mit denen Sie Daten an Ihren Stream senden können. Lassen Sie uns einen Überblick über die verschiedenen Ansätze geben, die Sie verwenden können, und warum Sie diese auswählen sollten.

Amazon Kinesis Agent

Der Amazon Kinesis Agent wurde bereits früher als ein Tool beschrieben, mit dem Daten an Kinesis Firehose gesendet werden können. Das gleiche Tool kann verwendet werden, um Daten an Kinesis Streams zu senden. Weitere Informationen zum Installieren und Konfigurieren des Kinesis-Agenten finden Sie unter In Amazon Kinesis Firehose mit Amazon Kinesis Agent schreiben.[10]

Amazon Kinesis Producer Library (KPL)

Die KPL vereinfacht die Entwicklung von Herstelleranwendungen, sodass Entwickler einen hohen Schreibdurchsatz für einen oder mehrere Kinesis Streams erzielen können. Die KPL ist eine benutzerfreundliche, in hohem Maße konfigurierbare Bibliothek, die Sie auf Ihren Hosts installieren, die die Daten generieren, die Sie in Kinesis Streams streamen möchten. Sie fungiert als Vermittler zwischen Ihrem Producer-Anwendungscode und den Kinesis Streams-API-Aktionen. Die KPL führt die folgenden Hauptaufgaben aus:

- Mit einem automatischen und konfigurierbaren Wiederholungsmechanismus in einen oder mehrere Kinesis-Streams schreiben
- Datensätze sammeln und `PutRecords` nutzen, zum Schreiben mehrerer Datensätze in mehrere Shards pro Anforderung
- Benutzerdatensätze aggregieren, um die Nutzlastgröße zu erhöhen und den Durchsatz zu verbessern
- Integriert sich nahtlos in die Amazon Kinesis Client Library (KCL), um Batch-Datensätze auf dem Konsumenten zu aggregieren
- Amazon CloudWatch-Messwerte in Ihrem Namen übermitteln, um die Leistung des Herstellers sichtbar zu machen

Die KPL kann entweder in synchronen oder asynchronen Anwendungsfällen verwendet werden. Wir schlagen vor, die höhere Leistung der asynchronen Schnittstelle zu verwenden, es sei denn, es gibt einen bestimmten Grund, ein synchrones Verhalten zu verwenden. Weitere Informationen zu diesen beiden Anwendungsfällen und Beispielcode finden Sie unter Mit der KPL in den Kinesis Data Stream schreiben.[11]

Die KPL kann helfen, leistungsstarke Produzenten aufzubauen. Stellen Sie sich eine Situation vor, in der Ihre Amazon Elastic Compute Cloud-Instanzen (EC2) als Proxy dienen, um 100-Byte-Ereignisse von Hunderten oder Tausenden von

leistungsschwachen Geräten zu erfassen und Datensätze in einen Kinesis Stream zu schreiben. Diese EC2-Instanzen müssen jeweils Tausende von Ereignissen pro Sekunde in Ihren Kinesis Stream schreiben. Um den erforderlichen Durchsatz zu erreichen, müssen Hersteller zusätzlich zur Wiederholungslogik eine komplizierte Logik wie Batch-Verarbeitung oder Multithreading implementieren und die Verbraucherseite aufzeichnen. Die KPL führt alle diese Aufgaben für Sie durch.

Da die KPL Ihre Datensätze puffert, bevor sie an einen Kinesis Stream gesendet werden, kann die KPL eine zusätzliche Verarbeitungsverzögerung verursachen, je nachdem, wie lange Sie die KPL konfiguriert haben, um Datensätze zu puffern, bevor sie an Kinesis gesendet werden. Größere Pufferzeiten ergeben eine höhere Packungseffizienz und eine bessere Leistung. Anwendungen, die diese zusätzliche Verzögerung nicht tolerieren können, müssen möglicherweise das AWS SDK direkt verwenden.

Wenn Ihre Anwendung keine Datensätze in einer lokalen Datei protokolliert und eine große Anzahl von kleinen Datensätzen pro Sekunde erstellt, sollten Sie die KPL verwenden.

Einzelheiten zur Verwendung der KPL zur Datenproduktion finden Sie unter Entwickeln von Amazon Kinesis Data Streams-Produzenten mit der Kinesis-Producer-Bibliothek.[12]

Amazon Kinesis API

Nachdem ein Stream erstellt wurde, können Sie ihm Datensätze hinzufügen. Ein Datensatz ist eine Datenstruktur, die die zu verarbeitenden Daten in Form eines Datenklumpens enthält. Nachdem Sie die Daten im Datensatz gespeichert haben, prüft, interpretiert oder ändert Kinesis Streams die Daten in keiner Weise.

Es gibt zwei verschiedene Vorgänge in der Kinesis Stream API, die Daten zu einem Stream hinzufügen: `PutRecords` und `PutRecord`. Die `PutRecords` Operation sendet mehrere Datensätze an Ihren Stream per HTTP-Anforderung, und die singuläre `PutRecord` Operation sendet Datensätze nacheinander an Ihren Stream (für jeden Datensatz ist eine separate HTTP-Anforderung erforderlich). Sie sollten `PutRecords` für die meisten Anwendungen verwenden, da dadurch ein höherer Durchsatz pro Datenproduzent erreicht wird.

Da die APIs in allen AWS SDKs verfügbar sind, bietet die Verwendung der API zum Schreiben von Datensätzen die flexibelste Lösung zum Senden von Daten an einen Kinesis Stream. Wenn Sie den Kinesis Agent oder KPL nicht verwenden können (z. B. wenn Sie Nachrichten direkt von einer mobilen Anwendung aus schreiben möchten oder wenn Sie die End-to-End-Latenz der Nachricht so weit wie möglich minimieren möchten), verwenden Sie die APIs, um Aufzeichnungen zu Ihrem Kinesis Stream zu senden.

Weitere Informationen zu diesen APIs finden Sie unter Verwenden der API in der Dokumentation der Kinesis Streams.[13] Die Details für jede API-Operation finden Sie in der Amazon Kinesis Streams API Reference.[14]

Verarbeiten von Daten in Amazon Kinesis Streams

Ein Konsument ist eine Anwendung, die Daten von Kinesis Streams liest und verarbeitet. Sie können Konsumenten für Kinesis-Streams auf verschiedene Arten erstellen. In diesem Abschnitt werden vier der gängigsten Ansätze behandelt: Verwenden von Kinesis Analytics, Verwenden der KCL, Verwenden von Amazon Lambda und direktes Verwenden der Kinesis Streams API.

Verwendung von Amazon Kinesis Analytics

Zuvor haben wir besprochen, wie Kinesis Analytics Streaming-Daten mit Standard-SQL analysieren kann. Kinesis Analytics kann die Daten aus Ihrem Kinesis-Stream lesen und mit der von Ihnen bereitgestellten SQL verarbeiten. Weitere Informationen zum Verarbeiten Ihrer Streamingdaten mit Kinesis Analytics finden Sie unter Konfigurieren der Anwendungseingabe im Kinesis Analytics-Entwicklerhandbuch.

Verwendung der Amazon Kinesis Client Library (KCL)

Sie können mithilfe der KCL eine Konsumentenanwendung für Kinesis Streams entwickeln. Obwohl Sie die Kinesis Streams-API verwenden können, um Daten aus einem Amazon Kinesis Stream abzurufen, empfehlen wir, die Entwurfsmuster und den Code für von der KCL bereitgestellte Konsumentenanwendungen zu verwenden.

Mit der KCL können Sie Daten aus einem Kinesis Stream konsumieren und verarbeiten. Diese Art der Anwendung wird auch als Konsument bezeichnet. Die KCL kümmert sich um viele der komplexen Aufgaben, die mit verteilter Datenverarbeitung verbunden sind, z. B. Lastverteilung über mehrere Instanzen hinweg, Reaktion auf Instanzfehler, Prüfpunktverarbeitung verarbeiteter

Datensätze und Reaktion auf Resharding. Mit der KCL können Sie sich auf das Schreiben von Datensatzverarbeitungslogik konzentrieren.

Die KCL ist eine Java-Bibliothek. Unterstützung für andere Sprachen als Java wird über eine mehrsprachige Schnittstelle bereitgestellt. Zur Laufzeit setzt eine KCL-Anwendung einen Worker mit Konfigurationsinformationen ein und verwendet dann einen Datensatzprozessor, um die von einem Kinesis Stream empfangenen Daten zu verarbeiten. Sie können eine KCL-Anwendung für beliebig viele Instanzen ausführen. Mehrere Instanzen der gleichen Anwendung koordinieren dynamisch Fehler und Lastenausgleich. Sie können auch mehrere KCL-Anwendungen im selben Stream ausführen, abhängig von den Durchsatzbeschränkungen. Die KCL fungiert als Vermittler zwischen Ihrer Aufzeichnungsverarbeitungslogik und Kinesis Streams.

Ausführliche Informationen zum Erstellen einer eigenen KCL-Anwendung finden Sie unter Entwickeln von Amazon Kinesis Data Streams-Konsumenten mit der Kinesis Client Library.[15]

Verwenden von AWS Lambda

AWS Lambda ist ein Rechenservice, mit dem Sie Codes ausführen können, ohne Server bereitzustellen oder zu verwalten.[16] AWS Lambda führt Ihren Code nur bei Bedarf aus und skaliert automatisch. Mit AWS Lambda können Sie Codes ohne Verwaltungsaufwand ausführen. AWS Lambda führt Ihren Code auf einer Hochverfügbarkeits-Computing-Infrastruktur aus und übernimmt die gesamte Verwaltung der Computing-Ressourcen, einschließlich Server- und Betriebssystemwartung, Kapazitätsbereitstellung und automatischer Skalierung sowie Codeüberwachung und -protokollierung. Sie müssen Ihren Code nur in einer der von AWS Lambda unterstützten Sprachen bereitstellen.

Sie können Lambda-Funktionen abonnieren, um Batch-Datensätze automatisch aus Ihrem Kinesis Stream zu lesen und zu verarbeiten, wenn Datensätze im Stream erkannt werden. AWS Lambda fragt dann den Stream regelmäßig (einmal pro Sekunde) nach neuen Datensätzen ab. Wenn neue Datensätze erkannt werden, wird die Lambda-Funktion aufgerufen, indem die neuen Datensätze als Parameter übergeben werden. Wenn keine neuen Datensätze gefunden werden, wird Ihre Lambda-Funktion nicht aufgerufen.

Ausführliche Informationen zur Verwendung von AWS Lambda zum Abrufen von Daten aus Kinesis Streams finden Sie unter Verwenden von AWS Lambda mit Amazon Kinesis.[17]

Verwenden der API

In den meisten Anwendungsfällen sollten Sie KCL oder AWS Lambda verwenden, um Daten aus einem Stream abzurufen und zu verarbeiten. Wenn Sie jedoch Ihre eigene Konsumentenanwendung von Grund auf neu schreiben möchten, gibt es mehrere Methoden, die dies ermöglichen. Die Kinesis Streams-API stellt die Methode `GetShardIterator` zum Abrufen von Daten aus einem Stream bereit. `GetRecords` Dies ist ein Pull-Modell, bei dem Ihr Code Daten direkt aus den Shards des Streams bezieht. Weitere Informationen zum Verfassen einer eigenen Konsumentenanwendung mithilfe der API finden Sie unter Entwickeln von Amazon Kinesis Data Streams-Konsumenten unter Verwendung der Amazon Kinesis Data Streams-API. Details zur API finden Sie in der Amazon Kinesis Streams-API-Referenz.[18]

Auswählen des besten Konsumentenmodells für Ihre Anwendung

Woher wissen Sie, welches Konsumentenmodell für Ihren Anwendungsfall am besten geeignet ist? Jeder Ansatz hat seine eigenen Kompromisse und Sie müssen entscheiden, was für Sie wichtig ist. Hier finden Sie einige allgemeine Hinweise zur Auswahl des richtigen Verbrauchermodells.

In den meisten Fällen sollten Sie mit AWS Lambda beginnen. Durch die Benutzerfreundlichkeit und das einfache Bereitstellungsmodell können Sie schnell einen Datenkonsumenten erstellen. Der Kompromiss zur Verwendung von AWS Lambda besteht darin, dass jeder Aufruf Ihrer Lambda-Funktion als zustandslos betrachtet werden sollte. Das heißt, Sie können Ergebnisse früherer Aufrufe Ihrer Funktion nicht problemlos verwenden (z. B. frühere Datensätze aus Ihrem Stream). Beachten Sie auch, dass die maximale Ausführungszeit für eine einzelne Lambda-Funktion 5 Minuten beträgt. Wenn die Verarbeitung einer einzigen Charge von Datensätzen länger als 5 Minuten dauert, ist AWS Lambda möglicherweise nicht der beste Konsument für Ihren Anwendungsfall.

Wenn Sie entscheiden, dass Sie AWS Lambda nicht verwenden können, sollten Sie eine eigene Verarbeitungsanwendung mit der KCL erstellen. Da Sie KCL-Anwendungen auf EC2-Instanzen in Ihrem AWS-Konto bereitstellen, haben Sie eine große Flexibilität und Kontrolle hinsichtlich der lokalen Datenpersistenz und der Statusanforderungen für Ihre Daten.

Ihre dritte Option besteht darin, Ihre eigene Anwendung mithilfe der APIs direkt zu erstellen. Dies gibt Ihnen die meiste Kontrolle und Flexibilität, aber Sie müssen auch Ihre eigene Logik erstellen, um mit gängigen Consumer-Anwendungsfunktionen wie Checkpointing, Skalierung und Failover umzugehen.

Zusammenfassung

Kinesis Streams ermöglicht den einfachen Empfang von Streaming-Daten. Sie können einen Kinesis-Stream skalieren, um nur ein paar Datensätze pro Sekunde oder Millionen von Datensätzen pro Sekunde zu verarbeiten. Für ABC Tolls war die Datenrate in ihren Stream nicht groß. Sie profitierten jedoch von der direkten Integration mit AWS Lambda, wodurch API-Aufrufe für Benutzerbenachrichtigungen problemlos an Amazon SNS gesendet werden konnten.

Anforderung 3: Andere Schwellenwertwarnungen

Die letztendliche Anforderung ähnelt der vorherigen Anforderung, bringt jedoch ein zusätzliches Problem mit sich. Um diese letzte Anforderung zu wiederholen, möchten die Betreiber von ABC Tolls sofort benachrichtigt werden, wenn der Fahrzeugverkehr für eine Mautstation für jeden 30-Minuten-Zeitraum an einem Tag unter einen vordefinierten Schwellenwert fällt. Zum Beispiel wissen sie aus historischen Daten, dass eine ihrer Mautstationen mittwochs zwischen 14:00 Uhr und 14:30 Uhr etwa 360 Fahrzeuge verzeichnet. Wenn in diesem 30-Minuten-Fenster eine Mautstation weniger als 100 Fahrzeuge sieht, möchte sie benachrichtigt werden.

ABC Tolls möchte die aktuellen Fahrzeuggesamtwerte für jede Station mit einer bekannten Durchschnittsrate für diese Station vergleichen. Um dies zu erreichen, erstellten sie für jede Station eine Datei, die Schwellenwert-Verkehrswerte für jedes 30-Minuten-Fenster enthielt. Wie bereits beschrieben, unterstützt Kinesis Analytics die Verwendung von Referenzdaten. Basierend auf den Daten in einer Datei, die in einem S3-Bucket gespeichert ist, wird ein In-Application-Stream (wie eine Tabelle) erstellt. Mit dieser Funktion konnten ABC Tolls-Entwickler SQL in ihre Kinesis Analytics-Anwendung schreiben, um die Anzahl der Fahrzeuge, die an jeder Station in einem 30-minütigen Fenster gesehen wurden, zu zählen und diese Werte mit den Schwellenwerten in der Datei zu vergleichen. Wenn der Schwellenwert überschritten wurde, gibt Kinesis Analytics einen Datensatz an einen Kinesis-Stream aus. Wenn Datensätze im Stream ankommen, wird eine Lambda-Funktion ausgeführt, die Amazon SNS verwendet, um eine Benachrichtigung an AWS Tolls-Operatoren zu senden. Abbildung 8 zeigt die Architektur für dieses Szenario.

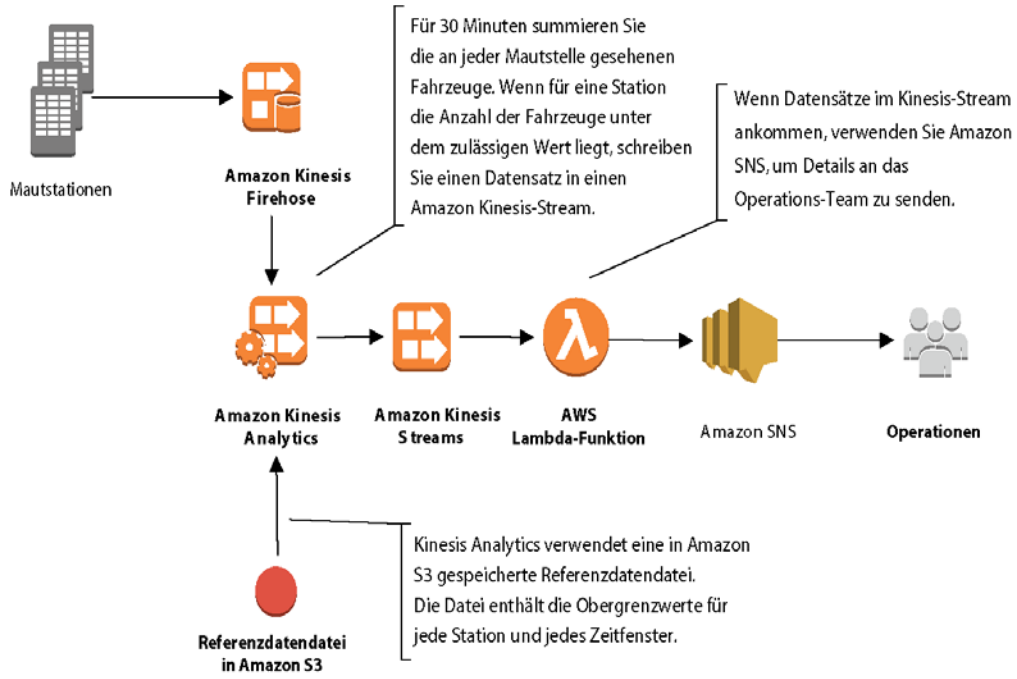

Abbildung 8: Architektur für Warnungen und Benachrichtigungen mit 30-Minuten-Zeitfenstern

Vollständige Architektur

Mit einer Lösung für jede Anforderung haben wir jetzt unsere gesamte Streaming-Lösung, wie in Abbildung 9 gezeigt.

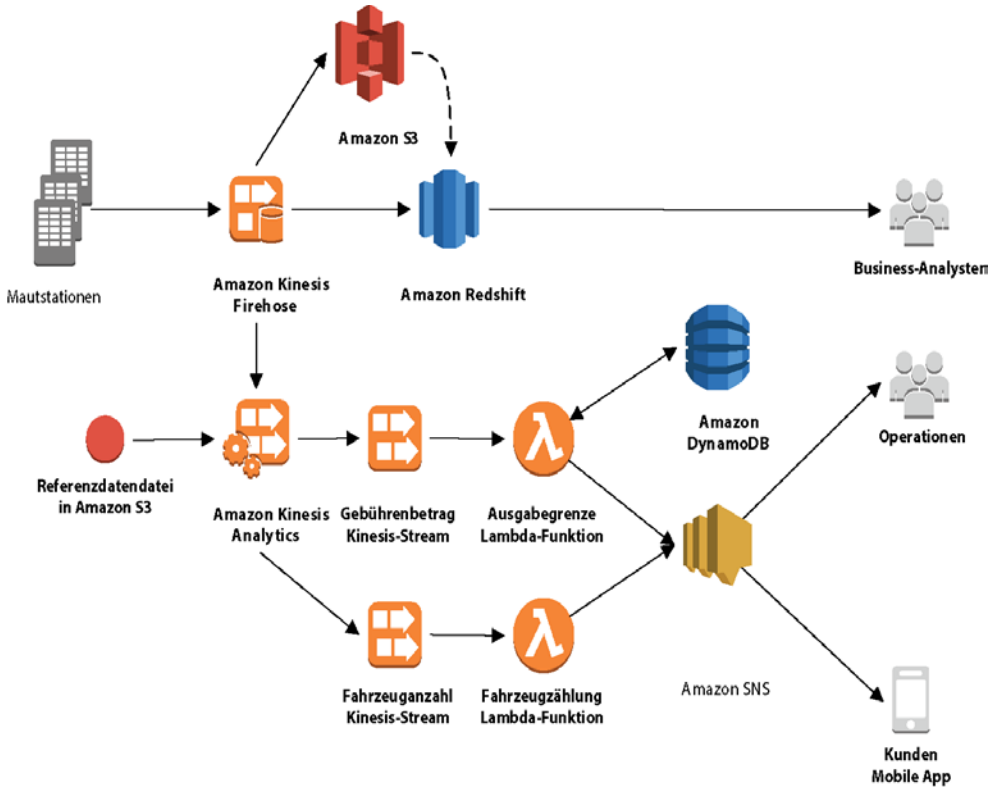

Abbildung 9: Die Architektur der allgemeinen Streaming-Lösung

Dieses Design bietet ABC Tolls eine flexible, reaktive Architektur. Indem sie die Transaktionen ihrer Kunden in Echtzeit streamen, sind sie in der Lage, ihre Anforderungen mit sehr geringem Entwicklungsaufwand und minimaler zu verwaltender Infrastruktur zu realisieren.

Fazit

In diesem Dokument haben wir untersucht, wie das fiktive Unternehmen ABC Tolls mithilfe von Amazon Kinesis-Services einen herkömmlichen Batch-Workflow in einen Streaming-Workflow umsetzte. Diese Migration bot ihnen die Möglichkeit, neue Funktionen und Funktionen hinzuzufügen, die mit ihrer alten Batch-Lösung nicht möglich waren.

Durch die Analyse der Daten bei der Erstellung erhalten Sie Einblicke in das, was Ihr Unternehmen gerade tut. Mithilfe von Amazon Kinesis-Services können Sie sich auf Ihre Anwendung konzentrieren, um zeitkritische

Geschäftsentscheidungen zu treffen, anstatt die Infrastruktur bereitzustellen und zu verwalten.

Mitwirkende

Dieses Dokument ist unter der Mitarbeit folgender Personen und Organisationen entstanden:

- Allan MacInnis, Solutions Architect, AWS
- Chander Matrubhutam, Product Marketing Manager, AWS

Hinweise

[1] https://aws.amazon.com/kinesis/streams/

[2] https://aws.amazon.com/kinesis/firehose/

[3] https://aws.amazon.com/kinesis/analytics/

[4] http://docs.aws.amazon.com/firehose/latest/dev/writing-with-sdk.html

[5] https://aws.amazon.com/lambda/

[6] https://aws.amazon.com/s3/

[7] https://aws.amazon.com/redshift/

[8] https://aws.amazon.com/elasticsearch-service/

[9] https://aws.amazon.com/dynamodb/

[10] http://docs.aws.amazon.com/firehose/latest/dev/writing-with-agents.html

[11] http://docs.aws.amazon.com/streams/latest/dev/kinesis-kpl-writing.html

[12] http://docs.aws.amazon.com/streams/latest/dev/developing-producers-with-kpl.html

[13] http://docs.aws.amazon.com/streams/latest/dev/developing-producers-with-sdk.html

[14] http://docs.aws.amazon.com/kinesis/latest/APIReference/Welcome.html

[15] http://docs.aws.amazon.com/streams/latest/dev/developing-consumers-with-kcl.html

[16] https://aws.amazon.com/lambda/

[17] http://docs.aws.amazon.com/lambda/latest/dg/with-kinesis.html

[18] http://docs.aws.amazon.com/kinesis/latest/APIReference/Welcome.html