Whitepaper AWS

Soluzioni per i dati di streaming di AWS per Amazon Kinesis

Soluzioni per i dati di streaming di AWS per Amazon Kinesis: Whitepaper AWS

Table of Contents

Soluzioni per i dati di streaming di AWS

Data di pubblicazione: 1 settembre 2021 (Revisioni del documento)

Riassunto

I data engineer, gli analisti di dati e gli sviluppatori di Big Data stanno cercando di sviluppare le loro analisi dei dati dall'analisi in batch a quella in tempo reale in modo che le aziende possano sapere ciò che clienti, applicazioni e prodotti stanno facendo in questo momento e reagire prontamente. Questo whitepaper illustra l'evoluzione dell'analisi dei dati dall'analisi in batch a quella in tempo reale. Descrive come servizi come Amazon Kinesis Data Streams, Amazon Kinesis Data Firehose, Amazon EMR, Amazon Kinesis Data Analytics, Amazon Managed Streaming for Apache Kafka (Amazon MSK) e altri servizi possono essere utilizzati per implementare applicazioni in tempo reale e fornisce modelli di progettazione comuni che utilizzano questi servizi.

Introduzione

A causa della crescita esplosiva delle origini dati che generano continuamente flussi di dati, oggigiorno le aziende ricevono enormi quantità di dati a grande velocità. Sia che si tratti di dati di registro da server di applicazioni, dati di clickstream da siti Web e applicazioni per dispositivi mobili o dati di telemetria da dispositivi Internet of Things (IoT), tutto questo contiene informazioni che possono aiutare a conoscere ciò che clienti, applicazioni e prodotti stanno facendo in questo momento.

Avere la capacità di elaborare e analizzare questi dati in tempo reale è essenziale per attività come il monitoraggio continuo delle applicazioni per garantire tempo di funzionamento elevato del servizio e personalizzare offerte promozionali e suggerimenti sui prodotti. L'elaborazione in tempo reale e quasi in tempo reale può anche rendere più accurati e utilizzabili altri casi d'uso comuni, come l'analisi dei dati dei siti Web e il machine learning, rendendo i dati disponibili a queste applicazioni in pochi secondi o minuti anziché in ore o giorni.

Scenari applicativi in tempo reale e quasi in tempo reale

È possibile utilizzare i servizi di dati di streaming per applicazioni in tempo reale e quasi in tempo reale come il monitoraggio delle applicazioni, il rilevamento frodi e le classifiche in tempo reale. I casi d'uso in tempo reale richiedono latenze end-to-end di millisecondi, dall'importazione dati, all'elaborazione, fino all'emissione dei risultati nei data store di destinazione e in altri sistemi. Ad esempio, Netflix usa Amazon Kinesis Data Streams per monitorare le comunicazioni tra tutte le sue applicazioni, mettendo l'azienda in condizione di rilevare e risolvere i problemi in modo rapido e limitando i tempi di inattività per i clienti. Sebbene il caso d'uso più comunemente applicabile sia il monitoraggio delle prestazioni delle applicazioni, vi è un numero crescente di applicazioni in tempo reale nei settori della tecnologia pubblicitaria, dei giochi e dell'IoT che rientrano in questa categoria.

I casi d'uso comuni quasi in tempo reale includono l'analisi dei dati sui data store per il Data Science e il machine learning (ML). È possibile utilizzare soluzioni per i dati di streaming per caricare continuamente dati in tempo reale nei data lake. È quindi possibile aggiornare i modelli di machine learning con maggiore frequenza non appena sono disponibili nuovi dati, migliorando così accuratezza e affidabilità dei risultati. Ad esempio, Zillow usa Kinesis Data Streams per raccogliere dati di record pubblici e inserzioni con più servizi di quotazione (MLS, multiple listing service), aggiornando le stime degli immobili quasi in tempo reale e garantendo ad acquirenti e venditori l'utilizzo di valori sempre aggiornati. ZipRecruiter, utilizza Amazon MSK per le pipeline di

registrazione degli eventi, che sono componenti infrastrutturali fondamentali in quanto raccolgono, archiviano ed elaborano continuamente oltre sei miliardi di eventi al giorno provenienti dal marketplace delle offerte di lavoro dell'azienda.

Differenza tra elaborazione batch ed elaborazione di flussi

Occorre un set di strumenti diverso per raccogliere, preparare ed elaborare i dati di streaming in tempo reale rispetto agli strumenti tradizionalmente utilizzati per l'analisi dei dati in batch. Con l'analisi dei dati tradizionale, essi vengono raccolti e caricati periodicamente in un database e l'analisi avviene ore, giorni o settimane dopo. L'analisi dei dati in tempo reale richiede un approccio diverso. Le applicazioni di elaborazione in streaming elaborano i dati continuamente in tempo reale, anche prima che vengano archiviati. I dati di streaming possono arrivare a un ritmo incredibile e i volumi di dati possono aumentare o diminuire in qualsiasi momento. Le piattaforme di elaborazione dei dati in streaming devono essere in grado di gestire la velocità e la variabilità dei dati in arrivo ed elaborarli man mano che arrivano, spesso da milioni a centinaia di milioni di eventi all'ora.

Sfide relative all'elaborazione dei flussi

L'elaborazione dei dati in tempo reale man mano che arrivano può consentire di prendere decisioni molto più velocemente di quanto sia possibile con le tradizionali tecnologie di analisi dei dati. Tuttavia, la costruzione e l'utilizzo di pipeline di dati di streaming personalizzate è complicato e richiede molte risorse:

- È necessario costruire un sistema in grado di raccogliere, preparare e trasmettere i dati provenienti simultaneamente da migliaia di origini dati a costi ridotti.
- È necessario ottimizzare le risorse di archiviazione e di calcolo in modo che i dati vengano raggruppati in batch e trasmessi in modo efficiente per la massima velocità effettiva e la bassa latenza.
- È necessario implementare e gestire un parco istanze di server per dimensionare il sistema in modo da poter gestire le diverse velocità dei dati che si intende inviare ad esso.

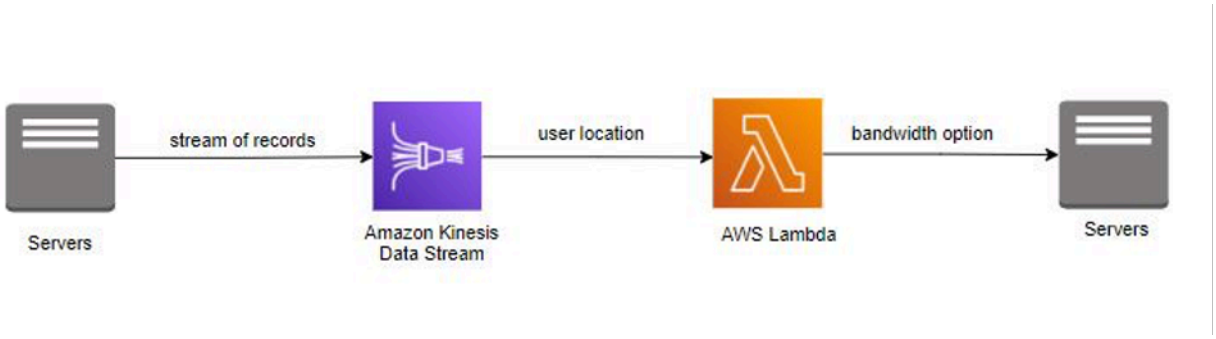
L'aggiornamento delle versioni è un processo complesso e costoso. Dopo aver costruito questa piattaforma, è necessario monitorare il sistema ed effettuare il ripristino da eventuali errori del server o della rete recuperando l'elaborazione dei dati dal punto appropriato nel flusso, senza creare dati duplicati. È inoltre necessario un team dedicato per la gestione dell'infrastruttura. Tutto ciò richiede tempo prezioso e denaro e, in fin dei conti, la maggior parte delle aziende ha difficoltà su questo

aspetto e deve accontentarsi dello status quo e gestire la propria attività con informazioni vecchie di ore o giorni.

Soluzioni per i dati di streaming: esempi

Scenario 1: offerta Internet basata sulla posizione

L'azienda InternetProvider fornisce servizi Internet con una varietà di opzioni di larghezza di banda agli utenti di tutto il mondo. Quando un utente si iscrive a Internet, l'azienda InternetProvider fornisce all'utente diverse opzioni di larghezza di banda in base alla sua posizione geografica. Alla luce di questi requisiti, l'azienda InternetProvider ha implementato un servizio Amazon Kinesis Data Streams per utilizzare i dettagli e la posizione dell'utente I dettagli e la posizione dell'utente vengono arricchiti con diverse opzioni di larghezza di banda prima della pubblicazione nell'applicazione. AWS Lambdaconsente questo arricchimento in tempo reale.



Elaborazione di flussi di dati con AWS Lambda

Amazon Kinesis Data Streams

Amazon Kinesis Data Streams consente di costruire applicazioni personalizzate in tempo reale utilizzando i più diffusi framework di elaborazione dei flussi e di caricare i dati di streaming in diversi data store. Un flusso Kinesis può essere configurato per ricevere continuamente eventi da centinaia di migliaia di produttori di dati forniti da fonti come clickstream di siti Web, sensori IoT, feed di social media e registri delle applicazioni. In pochi millisecondi, i dati sono disponibili per essere letti ed elaborati dall'applicazione.

Quando si implementa una soluzione con Kinesis Data Streams, si costruiscono applicazioni di elaborazione dati personalizzate note come applicazioni Kinesis Data Streams. Un'applicazione Kinesis Data Streams legge i dati da un flusso Kinesis come record di dati.

I dati inseriti in Kinesis Data Streams sono garantiti per essere ad alta disponibilità ed elastici e sono disponibili in millisecondi. È possibile aggiungere in modo continuo diversi tipi di dati provenienti

da centinaia di migliaia di origini a un flusso di Amazon Kinesis, ad esempio clickstream, registri di applicazioni e social media. In pochi secondi, i dati del flusso potranno essere letti ed elaborati dalle applicazioni Kinesis.

Amazon Kinesis Data Streams è un servizio di dati di streaming completamente gestito. Esso gestisce infrastruttura, archiviazione, reti e configurazione necessari per trasmettere i dati al tuo livello di velocità effettiva dei dati.

Invio di dati su Amazon Kinesis Data Streams

Esistono diversi modi per inviare dati a Kinesis Data Streams, garantendo flessibilità nella progettazione delle soluzioni.

- È possibile scrivere codice utilizzando uno degli AWS SDK supportati da più linguaggi comuni.
- È possibile utilizzare l'agente Amazon Kinesis, uno strumento per l'invio di dati a Kinesis Data Streams.

Amazon Kinesis Producer Library (KPL) semplifica lo sviluppo delle applicazioni producer consentendo agli sviluppatori di ottenere una velocità effettiva di scrittura elevata su uno o più flussi dei dati Kinesis.

KPL è una libreria facile da usare e altamente configurabile che si installa sugli host. Funge da intermediario tra il codice dell'applicazione producer e le operazioni dell'API di Kinesis Streams. Per ulteriori informazioni KPL e sulla sua capacità di produrre eventi in modo sincrono e asincrono con esempi di codice, consulta l'argomento relativo alla scrittura in Kinesis Data Streams utilizzando KPL

Ci sono due diverse operazioni nell'API di Kinesis Data Streams che aggiungono dati a un flusso: `PutRecords` e `PutRecord`. L'operazione `PutRecords` invia più record al flusso per richiesta HTTP mentre `PutRecord` invia un record per richiesta HTTP. Per ottenere una velocità effettiva più elevata per la maggior parte delle applicazioni, utilizzare `PutRecords`.

Per ulteriori informazioni su queste API, consulta Aggiunta di dati a un flusso. I dettagli per ciascuna operazione API sono disponibili nella Documentazione di riferimento delle API di Amazon Kinesis Data Streams.

Elaborazione di dati in Amazon Kinesis Data Streams

Per leggere ed elaborare i dati dai flussi Kinesis, è necessario creare un'applicazione consumer. Esistono diversi modi per creare consumer per Kinesis Data Streams. Alcuni di questi approcci

includono l'utilizzo di Amazon Kinesis Data Analytics per analizzare i dati di streaming utilizzando KCL, AWS Lambda, operazioni Streaming ETL AWS Glue e l'utilizzo diretto dell'API di Kinesis Data Streams.

Le applicazioni consumer per Kinesis Data Streams possono essere sviluppate utilizzando KCL, che aiuta a consumare ed elaborare i dati da Kinesis Data Streams. KCL si occupa di molte delle attività complesse associate al calcolo distribuito, come il bilanciamento del carico su più istanze, la risposta ai guasti delle istanze, il checkpoint dei record elaborati e la reazione al resharding. KCL consente di concentrarsi sulla scrittura della logica di elaborazione dei record. Per ulteriori informazioni su come costruire la propria applicazione KCL, consulta Utilizzo della libreria client Kinesis.

È possibile sottoscrivere le funzioni Lambda per leggere automaticamente batch di record dal flusso Kinesis ed elaborarli se vengono rilevati record nel flusso. AWS Lambda esegue periodicamente il polling del flusso (una volta al secondo) per nuovi record e quando li rileva, richiama la funzione Lambda passando i nuovi record come parametri. La funzione Lambda viene eseguita solo quando vengono rilevati nuovi record. È possibile mappare una funzione Lambda a un consumer con velocità effettiva condivisa (iteratore standard)

È possibile costruire un consumer che utilizzi una caratteristica chiamata fan-out avanzato quando è necessaria una velocità effettiva dedicata che non si desidera contendere con altri consumer che ricevono dati dal flusso. Questa caratteristica consente alle applicazioni consumer di ricevere record da un flusso con una velocità effettiva fino a 2 MiB di dati al secondo per ogni partizione.

Nella maggior parte dei casi, l'utilizzo di Kinesis Data Analytics, KCL, AWS Glue o AWS Lambda dovrebbe essere allo scopo di elaborare i dati da un flusso. Tuttavia, se si preferisce, è possibile creare un'applicazione consumer da zero utilizzando l'API Kinesis Data Streams. L'API Kinesis Data Streams fornisce i metodi `GetShardIterator` e `GetRecords` per recuperare i dati da un flusso.

In questo modello pull, il codice estrae i dati direttamente dalle partizioni del flusso. Per ulteriori informazioni sulla scrittura di applicazioni consumer utilizzando l'API, consulta Sviluppo di consumer personalizzati con velocità effettiva condivisa utilizzando AWS SDK per Java. I dettagli relativi all'API sono disponibili nella Documentazione di riferimento delle API di Amazon Kinesis Data Streams.

Elaborazione di flussi di dati con AWS Lambda

AWS Lambda consente di eseguire il codice senza dover effettuare il provisioning né gestire server. Grazie a Lambda, puoi eseguire codici per qualsiasi tipo di applicazione o servizio di back-end senza alcuna amministrazione. È sufficiente caricare il proprio codice e Lambda si occuperà di tutto il necessario per eseguire e dimensionare il codice con disponibilità elevata. Puoi configurare il codice

in modo che venga attivato automaticamente da altri servizi AWS oppure che venga richiamato direttamente da un qualsiasi applicazione Web o per dispositivi mobili.

AWS Lambda si integra in modo nativo con Amazon Kinesis Data Streams. Le complessità di polling, checkpoint e gestione degli errori vengono astratte quando si utilizza questa integrazione nativa. Ciò permette al codice della funzione Lambda di concentrarsi sull'elaborazione della logica di business.

È possibile mappare una funzione Lambda a un consumer a velocità effettiva condivise (iteratore standard) o a un consumer a velocità effettiva dedicata con fan-out avanzato. Per gli iteratori standard, Lambda esegue il polling di ogni partizione nel flusso Kinesis per i record utilizzando il protocollo HTTP. Per ridurre al minimo la latenza e massimizzare la velocità effettiva di lettura, è possibile creare un consumer di flusso dei dati con fan-out avanzato. I consumer di flussi in questa architettura ottengono una connessione dedicata a ogni partizione senza competere con altre applicazioni che leggono dallo stesso flusso. Amazon Kinesis Data Streams invia i record a Lambda tramite HTTP/2.

Di default, AWS Lambda richiama la funzione appena i record sono disponibili nel flusso. Per eseguire il buffer dei record per gli scenari batch, è possibile implementare una finestra batch per un massimo di cinque minuti nell'origine dell'evento. Se la funzione restituisce un errore, Lambda effettua nuovi tentativi sui batch finché l'elaborazione non va a buon fine o fino alla scadenza dei dati.

Riepilogo

L'azienda InternetProvider ha sfruttato Amazon Kinesis Data Streams per trasmettere dettagli e posizione degli utenti. Il flusso di record è stato utilizzato da AWS Lambda per arricchire i dati con opzioni di larghezza di banda archiviate nella libreria della funzione. Dopo l'arricchimento, AWS Lambda ha pubblicato le opzioni di larghezza di banda nell'applicazione. Amazon Kinesis Data Streams e AWS Lambda hanno gestito il provisioning e la gestione dei server, consentendo all'azienda InternetProvider di concentrarsi maggiormente sullo sviluppo di applicazioni aziendali.

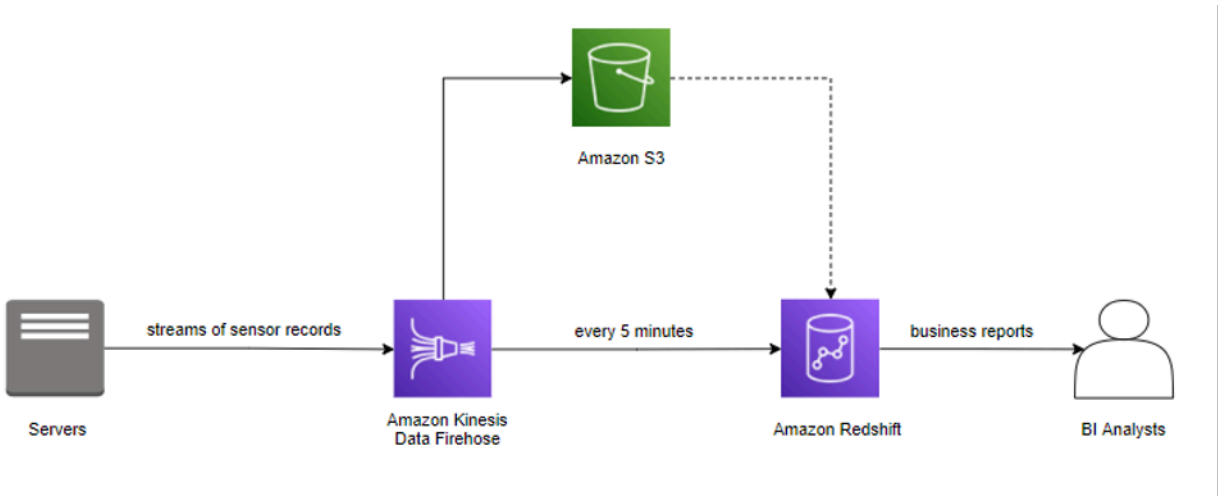
Scenario 2: dati quasi in tempo reale per i team di sicurezza

L'azienda ABC2badge fornisce sensori e badge per eventi aziendali o su larga scala come AWS re:Invent. Gli utenti si iscrivono all'evento e ricevono badge univoci che i sensori rilevano in tutto il campus. Quando gli utenti passano davanti a un sensore, le loro informazioni anonime vengono registrate in un database relazionale.

In un evento imminente, a causa dell'elevato volume di partecipanti, il team responsabile della sicurezza dell'evento ha richiesto ad ABC2badge di acquisire i dati delle aree più concentrate del

campus ogni 15 minuti. Ciò darà al team di sicurezza tempo sufficiente per reagire e distribuire il personale addetto alla sicurezza proporzionalmente nelle aree di concentrazione. Data questa nuova esigenza da parte del team di sicurezza e l'inesperienza nella costruzione di una soluzione per lo streaming, per elaborare i dati quasi in tempo reale, ABC2badge è alla ricerca di una soluzione semplice ma scalabile e affidabile.

L'attuale soluzione di data warehouse è Amazon Redshift. Durante l'esame delle caratteristiche dei servizi Amazon Kinesis, l'azienda ha realizzato che Amazon Kinesis Data Firehose può ricevere un flusso di record di dati, raggruppare i record in base alle dimensioni del buffer e/o all'intervallo di tempo e inserirli in Amazon Redshift. È stato creato un flusso di consegna di Kinesis Data Firehose ed è stato configurato in modo da copiare i dati nelle tabelle di Amazon Redshift ogni cinque minuti. Come parte di questa nuova soluzione, sui server è stato utilizzato l'agente Amazon Kinesis. Ogni cinque minuti, Kinesis Data Firehose carica i dati in Amazon Redshift, dove il team di business intelligence (BI) è abilitato a eseguire l'analisi dei dati e a inviare i dati al team di sicurezza ogni 15 minuti.



Nuova soluzione con Amazon Kinesis Data Firehose

Amazon Kinesis Data Firehose

Amazon Kinesis Data Firehose è il mezzo più semplice per caricare i dati di streaming in AWS. È in grado di acquisire, trasformare e caricare dati di streaming in Amazon Kinesis Data Analytics, Amazon Simple Storage Service (Amazon S3), Amazon Redshift, Amazon OpenSearch Service (OpenSearch Service) e Splunk. Inoltre, Kinesis Data Firehose può caricare i dati di streaming in qualsiasi endpoint HTTP personalizzato o in endpoint HTTP di proprietà di provider di servizi di terze parti supportati.

Kinesis Data Firehose consente l'analisi dei dati quasi in tempo reale con gli strumenti e i pannelli di controllo di business intelligence esistenti già in uso oggi. Si tratta di un servizio serverless completamente gestito che dimensiona automaticamente le risorse in base alla velocità effettiva dei dati e non richiede alcuna attività di amministrazione continua. Kinesis Data Firehose può elaborare in batch, comprimere e crittografare i dati prima del caricamento, riducendo al minimo l'archiviazione utilizzata e migliorando il livello di sicurezza. Utilizzando AWS Lambda, può anche trasformare i dati di origine e inviare i dati trasformati alle destinazioni. Sarà sufficiente configurare le applicazioni producer di dati perché inviino i dati a Kinesis Data Firehose, che a sua volta li distribuirà automaticamente alla destinazione specificata.

Invio di dati a un flusso di consegna Firehose

Per inviare dati al flusso di consegna, sono disponibili diverse opzioni. AWS offre SDK per molti linguaggi di programmazione comuni, ognuno dei quali fornisce API per Amazon Kinesis Data Firehose. AWS dispone di una utility per l'invio di dati al flusso di consegna. Kinesis Data Firehose è stato integrato con altri servizi AWS per inviare dati direttamente da tali servizi nel flusso di consegna.

Utilizzo dell'Agente di Amazon Kinesis

L'agente Amazon Kinesis è un'applicazione software autonoma che monitora continuamente un set di file di log alla ricerca di nuovi dati da inviare al flusso di consegna. L'agente gestisce automaticamente la rotazione dei file, il checkpoint, i nuovi tentativi in caso di errori e invia parametri di Amazon CloudWatch per il monitoraggio e la risoluzione dei problemi del flusso di consegna. All'agente è possibile applicare configurazioni aggiuntive, come la pre-elaborazione dei dati, il monitoraggio di più directory di file e la scrittura su più flussi di consegna.

L'agente può essere installato su server basati su Linux o Windows come server Web, server di registro e server di database. Una volta installato l'agente, è sufficiente specificare i file di log che monitorerà e il flusso di consegna a cui effettuerà gli invii. L'agente invierà in modo duraturo e affidabile nuovi dati al flusso di consegna.

Utilizzo di API con fonti quali AWS SDK e servizi AWS

L'API Kinesis Data Firehose offre due operazioni per l'invio di dati al flusso di consegna. `PutRecord` invia un record di dati all'interno di una chiamata. `PutRecordBatch` può inviare più record di dati all'interno di una chiamata e può raggiungere una velocità effettiva più elevata per produttore. In ogni metodo, è necessario specificare il nome del flusso di consegna e il record di dati, o array di record di dati, quando si utilizza questo metodo. Per ulteriori informazioni e il codice di esempio per le

operazioni delle API di Kinesis Data Firehose, consulta l'argomento relativo alla scrittura in un flusso di consegna di Firehose mediante l'AWS SDK.

Kinesis Data Firehose funziona anche con Kinesis Data Firehose, CloudWatch Logs, CloudWatch Events, Amazon Simple Notification Service (Amazon SNS), Amazon API Gateway e AWS IoT. Puoi inviare in modo scalabile e affidabile flussi di dati, registri, eventi e dati IoT direttamente in una destinazione Kinesis Data Firehose.

Elaborazione dei dati prima della consegna alla destinazione

In alcuni scenari, potresti voler trasformare o migliorare i dati di streaming prima che vengano consegnati a destinazione. Ad esempio, i produttori di dati potrebbero inviare testo non strutturato nei record di dati ed è necessario trasformarli in JSON prima di inviarli a OpenSearch Service. Oppure potresti voler convertire i dati JSON in un formato di file colonnare come Apache Parquet o Apache ORC prima di archiviare i dati in Simple Storage Service (Amazon S3).

Kinesis Data Firehose è dotato di funzionalità incorporate di conversione del formato dati. Con questo strumento è possibile convertire facilmente i flussi di dati JSON in formati di file Apache Parquet o Apache ORC.

Flusso di trasformazione dei dati

Per abilitare le trasformazioni dei dati di streaming, Kinesis Data Firehose utilizza una funzione Lambda creata per trasformare i dati. Kinesis Data Firehose memorizza i dati in entrata in una dimensione di buffer specificata per la funzione e quindi richiama la funzione Lambda specificata in modo asincrono. I dati trasformati vengono inviati da Lambda a Kinesis Data Firehose e Kinesis Data Firehose consegna i dati alla destinazione.

Conversione del formato dei dati

È anche possibile abilitare la conversione del formato dei dati di Kinesis Data Firehose, che convertirà il flusso di dati JSON in Apache Parquet o Apache ORC. Questa caratteristica può convertire solo JSON in Apache Parquet o Apache ORC. Se si dispone di dati in formato CSV, è possibile trasformarli tramite una funzione Lambda in JSON, quindi applicare la conversione del formato dei dati.

Distribuzione di dati

Come un flusso di consegna quasi in tempo reale, Kinesis Data Firehose effettua il buffering dei dati in entrata. Dopo aver raggiunto le soglie di buffering del flusso di consegna, i dati vengono recapitati alla destinazione che hai configurato. Esistono alcune differenze nel modo in cui Kinesis

Data Firehose consegna i dati a ciascuna destinazione, che questo documento esamina nelle sezioni seguenti.

Simple Storage Service (Amazon S3)

Simple Storage Service (Amazon S3) è una soluzione di archiviazione di oggetti con una semplice interfaccia di servizio Web, che consente di archiviare e recuperare qualsiasi quantità di dati ovunque sul Web. È stato progettato per offrire una durabilità del 99,999999999% ed essere in grado di gestire migliaia di miliardi di oggetti in tutto il mondo.

Distribuzione dei dati su Simple Storage Service (Amazon S3)

Per la distribuzione dei dati su Simple Storage Service (Amazon S3), Kinesis Data Firehose concatena più record in entrata in base alla configurazione di buffering del flusso di consegna, quindi li distribuisce a Simple Storage Service (Amazon S3) come oggetto S3. La frequenza di distribuzione dei dati a S3 è determinata dalla dimensione del buffer S3 (da 1 MB a 128 MB) o dall'intervallo del buffer (da 60 secondi a 900 secondi), a seconda dell'evento che si verifica per primo.

La distribuzione dei dati sul bucket S3 potrebbe non riuscire per diversi motivi. Ad esempio, il bucket potrebbe non esistere più oppure il ruolo AWS Identity and Access Management (IAM) assunto da Kinesis Data Firehose potrebbe non avere accesso al bucket. In queste condizioni, Kinesis Data Firehose continua a riprovare per un massimo di 24 ore fino a quando la distribuzione va a buon fine. Il tempo massimo di archiviazione dati di Kinesis Data Firehose è di 24 ore. Se la distribuzione dei dati non va a buon fine per più di 24 ore, i dati vengono persi.

Amazon Redshift

Amazon Redshift è una soluzione di data warehouse performante e completamente gestita che semplifica e riduce i costi dell'analisi dei dati utilizzando SQL standard e gli strumenti BI già esistenti. Questo servizio consente di eseguire query di analisi complesse su petabyte di dati strutturati utilizzando sofisticati sistemi di ottimizzazione delle query, archiviazione a colonne su dischi locali ad alte prestazioni ed esecuzione query in parallelo di grandi volumi di dati.

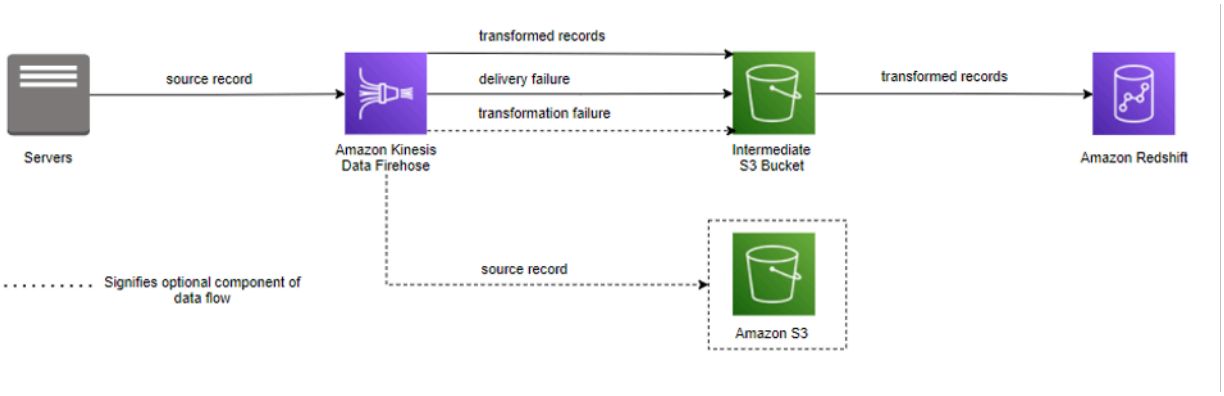
Distribuzione dei dati ad Amazon Redshift

Per la distribuzione dei dati ad Amazon Redshift, Kinesis Data Firehose consegna prima i dati in arrivo nel bucket S3 nel formato descritto in precedenza. Kinesis Data Firehose invia quindi un comando COPY di Amazon Redshift per caricare i dati dal bucket S3 al cluster di Amazon Redshift.

La frequenza delle operazioni `COPY` dei dati da Amazon S3 in Amazon Redshift è determinata dalla rapidità di esecuzione del comando `COPY` da parte del cluster Redshift. Per una destinazione Amazon

Redshift, puoi specificare una durata dei nuovi tentativi (0-7200 secondi) quando crei un flusso di consegna per gestire gli errori di consegna dei dati. Kinesis Data Firehose tenta nuovamente per il tempo specificato e ignora quel particolare batch di oggetti S3 se non riesce. Le informazioni relative agli oggetti non elaborati vengono inviate al bucket S3 sotto forma di file manifesto nella cartella errors/, che potrai utilizzare per recuperare le informazioni manualmente.

Di seguito è riportato un diagramma di architettura del flusso di dati da Kinesis Data Firehose ad Amazon Redshift. Sebbene questo flusso di dati sia esclusivo di Amazon Redshift, Kinesis Data Firehose segue schemi simili per gli altri target di destinazione.



Flusso di dati da Kinesis Data Firehose ad Amazon Redshift

Amazon OpenSearch Service (OpenSearch Service)

OpenSearch Service è un servizio completamente gestito che fornisce le API OpenSearch di facile utilizzo e le funzionalità in tempo reale insieme alla disponibilità, alla scalabilità e alla sicurezza richieste dai carichi di lavoro di produzione. OpenSearch Service semplifica l'implementazione, il funzionamento e il dimensionamento OpenSearch per l'analisi dei dati dei registri, la ricerca di testi completi e il monitoraggio delle applicazioni.

Distribuzione di dati a OpenSearch Service

Per la distribuzione dei dati a OpenSearch Service, Kinesis Data Firehose esegue il buffering dei record in entrata in base alla configurazione del buffering del flusso di consegna, quindi genera una richiesta in blocco OpenSearch per indicizzare più record nel cluster OpenSearch. La frequenza di distribuzione dei dati a OpenSearch Service è determinata dai valori della dimensione del buffer OpenSearch (da 1 MB a 100 MB) e dell'intervallo del buffer (da 60 secondi a 900 secondi), a seconda dell'evento che si verifica per primo.

Per la destinazione OpenSearch Service, puoi specificare una durata di ripetizione (da 0 a 7200 secondi) durante la creazione di un flusso di consegna. Kinesis Data Firehose riprova per il tempo

specificato e poi ignora quella particolare richiesta di indice. I documenti non elaborati vengono distribuiti sul bucket S3 nella cartella `elasticsearch_failed/`, che potrai utilizzare per recuperare le informazioni manualmente.

Amazon Kinesis Data Firehose può ruotare l'indice OpenSearch Service in base a una durata. A seconda dell'opzione di rotazione scelta (`NoRotation`, `OneHour`, `OneDay` `OneWeek`, o `OneMonth`), Kinesis Data Firehose aggiunge una parte del timestamp degli arrivi UTC (Coordinated Universal Time) al nome dell'indice specificato.

Endpoint HTTP personalizzato o provider di servizi di terzi supportato

Kinesis Data Firehose può inviare dati a endpoint HTTP personalizzati o a provider di terze parti supportati come Datadog, Dynatrace, LogicMonitor, MongoDB, New Relic, Splunk e Sumo Logic.

Endpoint HTTP personalizzato o provider di servizi di terzi supportato

Affinché Kinesis Data Firehose possa distribuire correttamente i dati agli endpoint HTTP personalizzati, questi endpoint devono accettare richieste e inviare risposte utilizzando determinati formati di richiesta e risposta di Kinesis Data Firehose.

Quando si forniscono dati a un endpoint HTTP di proprietà di un provider di servizi di terze parti supportato, è possibile utilizzare il servizio AWS Lambda integrato per creare una funzione per trasformare i record in entrata nel formato corrispondente al formato previsto dall'integrazione del provider di servizi.

Per la frequenza di distribuzione dei dati, ogni fornitore di servizi ha una dimensione del buffer consigliata. Collabora con il tuo fornitore di servizi per ulteriori informazioni sulla dimensione del buffer consigliata. Per la gestione degli errori di distribuzione dei dati, Kinesis Data Firehose stabilisce prima una connessione con l'endpoint HTTP in attesa di una risposta dalla destinazione. Kinesis Data Firehose continua a stabilire la connessione fino alla scadenza della durata dei tentativi. Successivamente, Kinesis Data Firehose lo considera un errore di distribuzione dei dati ed esegue il backup dei dati sul bucket S3.

Riepilogo

Kinesis Data Firehose è in grado di distribuire in modo persistente i dati di streaming verso una destinazione supportata. È una soluzione completamente gestita, che richiede uno sviluppo minimo o nullo. Per l'azienda ABC2badge, l'utilizzo di Kinesis Data Firehose è stata una scelta naturale. Stava già utilizzando Amazon Redshift come soluzione per il data warehouse. Poiché le origini dati scrivevano continuamente nei registri delle transazioni, è stata in grado di sfruttare l'agente Amazon

Kinesis per la trasmissione dei dati senza scrivere codice aggiuntivo. Ora che ABC2badge ha creato un flusso di record di sensori e sta ricevendo questi record tramite Kinesis Data Firehose, può usarlo come base per il caso d'uso del team di sicurezza.

Scenario 3: Preparazione dei dati clickstream per i processi di rilevamento delle informazioni dettagliate sui dati

Fast Sneakers è una boutique di moda con particolare attenzione alle sneakers di tendenza. Il prezzo di un determinato paio di scarpe può aumentare o diminuire a seconda dell'inventario e delle tendenze, ad esempio se una celebrità o un campione sportivo hanno indossato una determinata marca di sneaker in TV la sera prima. È importante che Fast Sneakers tenga traccia e analizzi tali tendenze per massimizzare il fatturato.

Fast Sneakers non vuole introdurre ulteriori costi nel progetto con altre infrastrutture da gestire. Vuole riuscire a dividere lo sviluppo in modo appropriato tra le parti, dove i data engineer possono concentrarsi sulla trasformazione dei dati e i Data Scientist possono lavorare sulla funzionalità ML in modo indipendente.

Per reagire rapidamente e adeguare automaticamente i prezzi in base alla domanda, Fast Sneakers trasmette eventi significativi (come i dati relativi ai clic e agli acquisti), trasformando e aumentando i dati degli eventi e inviandoli a un modello di ML. Il suo modello di ML è in grado di determinare se è necessario un adeguamento del prezzo. Ciò permette a Fast Sneakers di modificare automaticamente i propri prezzi per massimizzare il profitto sui propri prodotti.

Adeguamenti dei prezzi in tempo reale

Questo diagramma dell'architettura mostra la soluzione di streaming in tempo reale creata da Fast Sneakers utilizzando Kinesis Data Streams, AWS Glue e DynamoDB Streams. Sfruttando questi servizi, è disponibile una soluzione elastica e affidabile senza dover dedicare tempo alla configurazione e alla gestione dell'infrastruttura di supporto. L'azienda può dedicare il suo tempo a ciò che apporta valore all'azienda concentrandosi sulle operazioni ETL (extract, transform, load) e sul modello di machine learning.

Per comprendere meglio l'architettura e le tecnologie utilizzate nel carico di lavoro, di seguito sono riportati alcuni dettagli dei servizi utilizzati.

Streaming AWS Glue e AWS Glue

AWS Glue è un servizio ETL completamente gestito che puoi utilizzare per catalogare i dati, pulirli, arricchirli e spostarli in modo affidabile tra i data store. Con AWS Glue, è possibile ridurre in modo significativo i costi, la complessità e il tempo dedicato alla creazione di operazioni ETL. AWS Glue è serverless, quindi non vi è alcuna infrastruttura da configurare o gestire. I prezzi sono calcolati in base alle risorse impiegate per l'esecuzione dei processi.

Utilizzando AWS Glue, è possibile creare un'applicazione consumer con un'operazione streaming ETL AWS Glue. Ciò consente di utilizzare la scrittura su Apache Spark e altri moduli basati su Spark per utilizzare ed elaborare i dati degli eventi. La sezione successiva di questo documento approfondisce questo scenario.

AWS Glue Data Catalog

AWS Glue Data Catalog contiene riferimenti a dati che vengono utilizzati come origini e destinazioni delle operazioni ETL in AWS Glue. Il AWS Glue Data Catalog è un indice per i parametri di posizione, schema e tempo di esecuzione dei dati. È possibile utilizzare le informazioni presenti nel catalogo dati per creare e monitorare le operazioni ETL. Le informazioni nel catalogo dati vengono archiviate come tabelle di metadati, dove ogni tabella specifica un singolo datastore. Configurando un crawler, è possibile valutare automaticamente numerosi tipi di archivi dati, inclusi gli archivi connessi a DynamoDB, S3 e Java Database Connectivity (JDBC), estrarre metadati e schemi e quindi creare definizioni di tabelle in AWS Glue Data Catalog.

Per utilizzare Amazon Kinesis Data Streams nelle operazioni Streaming ETL AWS Glue, è preferibile definire il flusso in una tabella di un database AWS Glue Data Catalog. Definire una tabella di origine del flusso con il flusso Kinesis, uno dei tanti formati supportati (CSV, JSON, ORC, Parquet, Avro o un formato cliente con Grok). È possibile inserire manualmente uno schema oppure lasciare questa fase al processo AWS Glue per determinarla durante il tempo di esecuzione del processo.

Operazione Streaming ETL in AWS Glue

AWS Glue esegue le operazioni ETL in un ambiente Apache Spark serverless. AWS Glue esegue questi processi su risorse virtuali di cui effettua il provisioning e che gestisce nel proprio account di servizio. Oltre a essere in grado di eseguire processi basati su Apache Spark, AWS Glue fornisce un ulteriore livello di funzionalità su Spark con DynamicFrames.

DynamicFrames sono tabelle distribuite che supportano dati annidati come strutture e array. Ogni record è auto descrittivo, progettato per la flessibilità dello schema con dati semi-strutturati. Un record in un `DynamicFrame` contiene sia i dati che lo schema che descrive i dati. Sia `DataFrames` che `DynamicFrames` di Apache Spark sono supportati negli script ETL ed è possibile alternarli. `DynamicFrames` forniscono una serie di trasformazioni avanzate per la pulizia dei dati e per ETL.

Utilizzando Spark Streaming nel processo AWS Glue, puoi creare operazioni Streaming ETL che vengono eseguite continuamente e utilizzare dati da fonti di streaming come Amazon Kinesis Data Streams, Apache Kafka e Amazon MSK. I processi possono pulire, unire e trasformare i dati, quindi caricare i risultati in archivi dati tra cui Simple Storage Service (Amazon S3), Amazon DynamoDB o JDBC.

Di default, AWS Glue elabora e scrive i dati in finestre di 100 secondi. Ciò permette di elaborare i dati in modo efficiente e di eseguire aggregazioni su dati che arrivano più tardi del previsto. È possibile configurare la dimensione della finestra regolandola per adattarsi alla velocità di risposta rispetto all'accuratezza dell'aggregazione. I processi di streaming AWS Glue utilizzano i checkpoint per tenere traccia dei dati letti da Kinesis Data Stream. Per una procedura dettagliata sulla creazione di un'operazione Streaming ETL in AWS Glue puoi consultare Aggiunta di operazioni Streaming ETL in AWS Glue

Amazon DynamoDB

Amazon DynamoDB è un database chiave-valore e di documenti in grado di garantire prestazioni in millisecondi a una cifra, indipendentemente dalle dimensioni. Si tratta di un database durevole, multiregione, multiattivo e completamente gestito che offre sicurezza, backup e ripristino integrati e memorizzazione nella cache in memoria per applicazioni Internet. DynamoDB può gestire oltre dieci trilioni di richieste al giorno e supporta picchi di oltre 20 milioni di richieste al secondo.

Acquisizione dei dati di modifica per DynamoDB Streams

Un flusso DynamoDB è un flusso ordinato di informazioni sulle modifiche apportate agli elementi in una tabella DynamoDB. Quando abiliti un flusso in una tabella, DynamoDB acquisisce informazioni su ogni modifica apportata agli elementi di dati nella tabella. Amazon DynamoDB viene eseguito su

AWS Lambda, in modo da poter creare delle attivazioni, vale a dire parti di codice che rispondono automaticamente agli eventi in DynamoDB Streams. Con le attivazioni è possibile costruire applicazioni che rispondono alle modifiche di dati nelle tabelle Dynamo DB.

Quando su una tabella è abilitato un flusso, puoi associare l'Amazon Resource Name (ARN) del flusso a una funzione Lambda che scrivi. Immediatamente dopo che un elemento nella tabella viene modificato, nel flusso della tabella viene visualizzato un nuovo record. AWS Lambda esegue il polling del flusso e, quando rileva nuovi record nel flusso, richiama in modo sincrono la funzione Lambda.

Endpoint del servizio Amazon SageMaker

Amazon SageMaker è una piattaforma completamente gestita che consente a sviluppatori e Data Scientist di costruire, addestrare e implementare modelli ML in modo rapido e in qualsiasi dimensione. SageMaker include moduli che possono essere utilizzati congiuntamente ad altri o in modo indipendente per costruire, addestrare e implementare i modelli di ML. Con gli endpoint del servizio di Amazon SageMaker, puoi creare endpoint ospitati gestiti per l'inferenza in tempo reale con un modello implementato sviluppato all'interno o all'esterno di Amazon SageMaker.

Utilizzando l'AWS SDK, è possibile richiamare un endpoint SageMaker che trasmette le informazioni sul tipo di contenuti insieme ai contenuti e quindi ricevere previsioni in tempo reale basate sui dati trasferiti. Ciò consente di mantenere la progettazione e lo sviluppo dei modelli ML separati dal codice che esegue operazioni sui risultati dedotti.

Ciò consente ai Data Scientist di concentrarsi sull'ML e agli sviluppatori che utilizzano il modello ML di concentrarsi su come lo utilizzano nel codice. Per ulteriori informazioni su come richiamare un endpoint in SageMaker, consulta InvokeEnpoint nella Documentazione di riferimento dell'API di Amazon SageMaker.

Deduzione delle informazioni dettagliate sui dati in tempo reale

Il diagramma dell'architettura precedente indica che l'applicazione Web esistente di Fast Sneakers ha aggiunto un flusso dei dati Kinesis contenente eventi di clickstream, che fornisce dati sul traffico e sugli eventi dal sito Web. Il catalogo prodotti, che contiene informazioni quali la categorizzazione, gli attributi dei prodotti e i prezzi e la tabella degli ordini, che contiene dati quali articoli ordinati, fatturazione, spedizione e così via, sono tabelle DynamoDB separate. L'origine del flusso dei dati e le tabelle DynamoDB appropriate hanno i metadati e gli schemi definiti in AWS Glue Data Catalog modo da essere utilizzati dall'operazione Streaming ETL AWS Glue.

Utilizzando Apache Spark, Spark Streaming e `DynamicFrames` nelle operazioni Streaming ETL AWS Glue, Fast Sneakers è in grado di estrarre i dati dal flusso dei dati e trasformarli, unendo i dati

delle tabelle di prodotti e degli ordini. Con i dati dopo la trasformazione, i set di dati da cui ottenere i risultati dell'inferenza vengono inviati a una tabella DynamoDB.

Il flusso DynamoDB per la tabella attiva una funzione Lambda per ogni nuovo record scritto. La funzione Lambda invia i record precedentemente trasformati a un endpoint SageMaker con l'AWS SDK per dedurre quali eventuali adeguamenti di prezzo sono necessari per un prodotto. Se il modello ML identifica un adeguamento al prezzo richiesto, la funzione Lambda scrive la variazione di prezzo sul prodotto nella tabella DynamoDB del catalogo.

Riepilogo

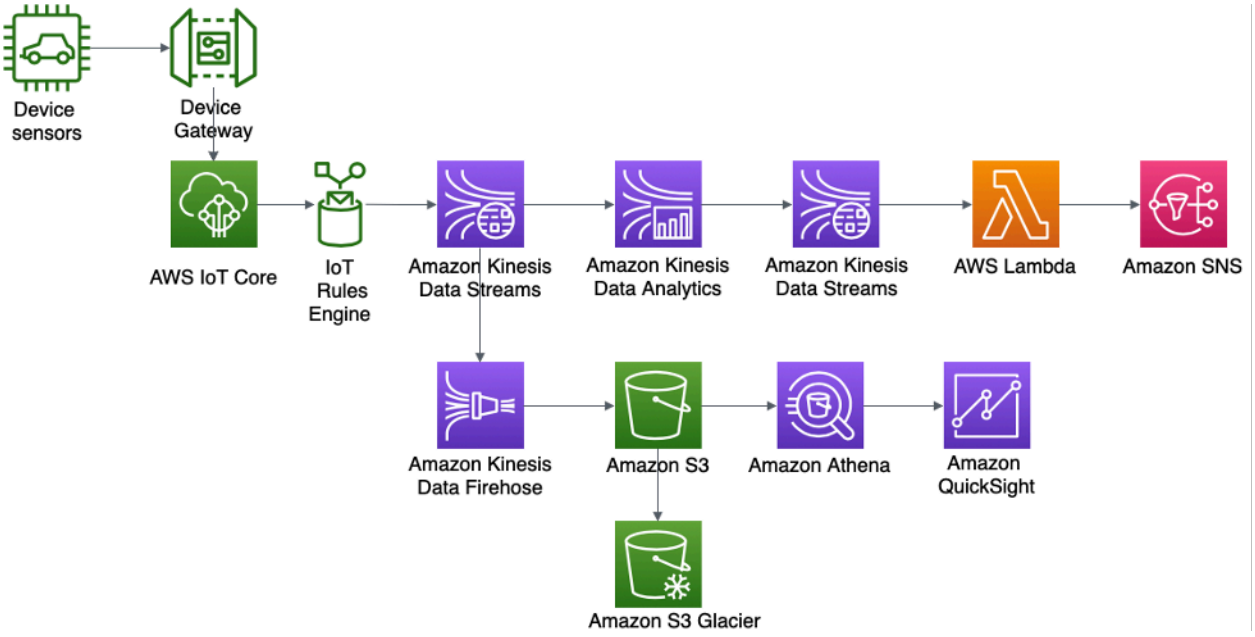
Amazon Kinesis Data Streams semplifica raccolta, elaborazione e analisi dei dati di streaming in tempo reale, per ottenere informazioni dettagliate tempestive e reagire rapidamente alle nuove informazioni. In combinazione con il servizio serverless di integrazione dei dati AWS Glue, puoi creare applicazioni di streaming di eventi in tempo reale che preparano e combinano i dati per ML.

Poiché sia Kinesis Data Streams che i servizi AWS Glue sono completamente gestiti, AWS elimina il pesante e indifferenziato carico di gestione dell'infrastruttura per la tua piattaforma di Big Data, permettendoti di concentrarti sulla generazione di informazioni dettagliate sui dati in base ai tuoi dati.

Fast Sneakers può utilizzare l'elaborazione di eventi in tempo reale e l'ML per consentire al proprio sito Web di apportare adeguamenti dei prezzi in tempo reale completamente automatici, per massimizzare le scorte di prodotti Ciò apporta il massimo valore alla sua attività evitando la necessità di creare e gestire una piattaforma di Big Data.

Scenario 4: rilevamento delle anomalie e notifiche in tempo reale dei sensori del dispositivo

L'azienda ABC4logistics trasporta prodotti petroliferi altamente infiammabili come benzina, propano liquido (GPL) e nafta dal porto in varie città. Ci sono centinaia di veicoli che hanno più sensori installati su di essi per monitorare aspetti quali posizione, temperatura del motore, temperatura all'interno del container, velocità di guida, posizione di parcheggio, condizioni stradali e così via. Uno dei requisiti di ABC4Logistics è quello di monitorare le temperature del motore e del container in tempo reale e avvisare il conducente e il team di monitoraggio della flotta in caso di anomalie. Per rilevare tali condizioni e generare avvisi in tempo reale, ABC4Logistics ha implementato la seguente architettura su AWS.



Il dispositivo di ABC4Logistics rileva le anomalie in tempo reale e l'architettura delle notifiche

I dati provenienti dai sensori dei dispositivi vengono acquisiti da AWS IoT Gateway, dove il motore delle regole AWS IoT renderà disponibili i dati di streaming in Amazon Kinesis Data Streams. Utilizzando Kinesis Data Analytics, ABC4Logistics può eseguire analisi dei dati in tempo reale sui dati di streaming in Kinesis Data Streams.

Utilizzando Kinesis Data Analytics, ABC4Logistics è in grado di rilevare se le letture della temperatura dei sensori si discostano dalle normali letture per un periodo di dieci secondi e di inserire il record su un'altra istanza di Kinesis Data Streams, identificando i record anomali. Amazon Kinesis Data Streams richiama quindi le funzioni Lambda, che possono inviare gli avvisi al conducente e al team di monitoraggio della flotta tramite Amazon SNS.

Anche i dati in Kinesis Data Streams vengono trasferiti ad Amazon Kinesis Data Firehose. Amazon Kinesis Data Firehose conserva questi dati in Simple Storage Service (Amazon S3), consentendo ad ABC4Logistics di eseguire analisi dei dati in batch o quasi in tempo reale sui dati dei sensori. ABC4Logistics utilizza Amazon Athena per eseguire query sui dati in S3 e Amazon QuickSight per le visualizzazioni. Per la conservazione dei dati a lungo termine, la policy del Ciclo di vita S3 viene utilizzata per archiviare i dati in Amazon S3 Glacier.

Di seguito vengono descritti in dettaglio i componenti importanti di questa architettura.

Amazon Kinesis Data Analytics

Amazon Kinesis Data Analytics consente di trasformare e analizzare i dati di streaming e rispondere alle anomalie in tempo reale. È un servizio serverless su AWS, il che significa che Kinesis Data Analytics si occupa di effettuare il provisioning e dimensiona in modo elastico l'infrastruttura per gestire qualsiasi velocità effettiva dei dati. Ciò elimina tutto il pesante lavoro indifferenziato della configurazione e della gestione dell'infrastruttura di streaming e consente di dedicare più tempo alla scrittura di applicazioni di streaming.

Con Amazon Kinesis Data Analytics, puoi eseguire query interattive sui dati di streaming utilizzando diverse opzioni, tra cui SQL standard, applicazioni Apache Flink in Java, Python e Scala e costruire applicazioni Apache Beam utilizzando Java per analizzare i flussi dei dati.

Queste opzioni offrono la flessibilità di utilizzare un approccio specifico a seconda del livello di complessità dell'applicazione di streaming e del supporto di origine/destinazione. Nella sezione seguente viene illustrata l'opzione Kinesis Data Analytics per applicazioni Flink.

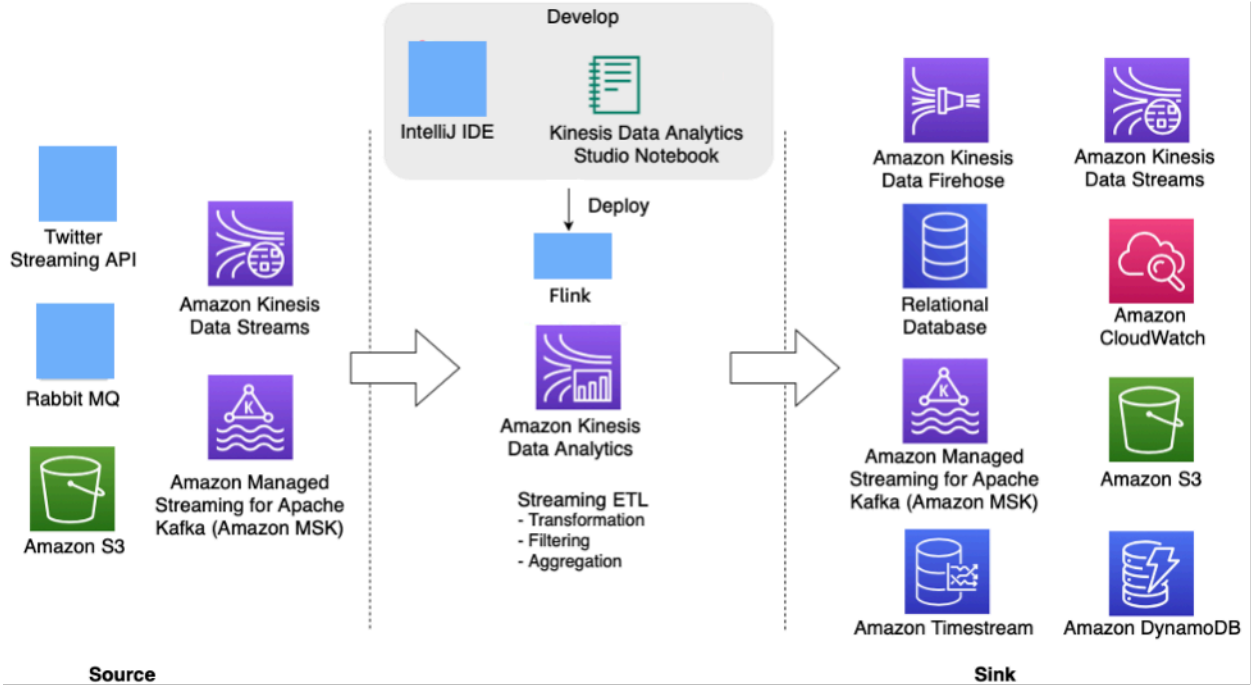
Amazon Kinesis Data Analytics per applicazioni Apache Flink

Apache Flink è un popolare framework open source e un motore di elaborazione distribuito per calcoli con stato su flussi dei dati illimitati e limitati. Apache Flink è progettato per eseguire calcoli alla velocità in memoria e su larga scala con il supporto per la semantica "exacly-once". Le applicazioni basate su Apache Flink aiutano a raggiungere una bassa latenza con una velocità effettiva elevata con tolleranza ai guasti.

Con Amazon Kinesis Data Analytics per Apache Flink, puoi creare ed eseguire codice su fonti di streaming per eseguire analisi dei dati di serie temporali, alimentare pannelli di controllo in tempo reale e creare parametri in tempo reale senza gestire il complesso ambiente distribuito di Apache Flink. È possibile utilizzare le caratteristiche di programmazione Flink di elevato livello nello stesso modo in cui le si utilizza quando si ospita personalmente l'infrastruttura Flink.

Kinesis Data Analytics per Apache Flink consente di creare applicazioni in Java, Scala, Python o SQL per elaborare e analizzare i dati di streaming. Una tipica applicazione Flink legge i dati dal flusso di input o dalla posizione o dall'origine dei dati, trasforma/filtra o unisce i dati utilizzando operatori o funzioni e archivia i dati nel flusso di output o nella posizione dei dati o nel sink.

Il seguente diagramma dell'architettura mostra alcune delle origini e dei sink supportati per l'applicazione Flink di Kinesis Data Analytics. Oltre ai connettori preassemblati per origine/sink, puoi anche inserire connettori personalizzati in una varietà di altre origini/sink per le applicazioni Flink su Kinesis Data Analytics.



Applicazione Apache Flink su Kinesis Data Analytics per l'elaborazione di flussi in tempo reale

Gli sviluppatori possono utilizzare l'IDE preferito per sviluppare applicazioni Flink e implementarle su Kinesis Data Analytics da AWS Management Console o strumenti DevOps.

Amazon Kinesis Data Analytics Studio

Come parte del servizio Kinesis Data Analytics,Kinesis Data Analytics Studio è disponibile per i clienti per query interattive sui flussi di dati in tempo reale e per costruire ed eseguire facilmente applicazioni di elaborazione dei flussi utilizzando SQL, Python e Scala. I notebook Studio utilizzano la tecnologia Apache Zeppelin.

Utilizzando il notebook Studio, è possibile sviluppare il codice dell'applicazione Flink in un ambiente notebook, visualizzare i risultati del codice in tempo reale e visualizzarli nel notebook. Puoi creare un notebook Studio basato su Apache Zeppelin e Apache Flink con un solo clic da Kinesis Data Streams e dalla console Amazon MSK oppure avviarlo dalla console di Kinesis Data Analytics.

Una volta sviluppato il codice in modo iterativo come parte di Kinesis Data Analytics Studio, è possibile implementare un notebook come applicazione di analisi dei dati Kinesis, per l'esecuzione continua in modalità di streaming, la lettura dei dati dalle origini, la scrittura nelle destinazioni, la gestione dello stato dell'applicazione con esecuzione prolungata e il dimensionamento automatico in base alla velocità effettiva dei flussi di origine. In precedenza, i clienti utilizzavano Kinesis Data Analytics for SQL Applications per l'analisi interattiva dei dati di streaming in tempo reale su AWS.

Kinesis Data Analytics per le applicazioni SQL è ancora disponibile, ma per i nuovi progetti, AWS consiglia di utilizzare il nuovo Kinesis Data Analytics Studio. Kinesis Data Analytics Studio combina facilità d'uso e capacità analitiche avanzate, permettendoti di costruire sofisticate applicazioni di elaborazione di flussi in pochi minuti.

Per rendere l'applicazione Flink di Kinesis Data Analytics tollerante ai guasti, è possibile utilizzare checkpoint e snapshot, come descritto in nella sezione relativa all'implementazione della tolleranza ai guasti in Kinesis Data Analytics per Apache Flink.

Le applicazioni Flink di Kinesis Data Analytics sono utili per la scrittura di applicazioni di analisi dei dati di flusso complesse con semantica "exactly-once" di elaborazione dei dati, funzionalità di checkpoint ed elaborazione di dati da origini dati come Kinesis Data Streams, Kinesis Data Firehose, Amazon MSK, Rabbit MQ e Apache Cassandra inclusi i connettori personalizzati.

Dopo aver elaborato i dati di streaming nell'applicazione Flink, è possibile mantenere i dati in vari sink o destinazioni come Amazon Kinesis Data Streams, Amazon Kinesis Data Firehose, Amazon DynamoDB, Amazon OpenSearch Service, Amazon Timestream, Simple Storage Service (Amazon S3) e così via. L'applicazione Kinesis Data Analytics Flink fornisce anche garanzie di prestazioni inferiori al secondo.

Applicazioni Apache Beam per Kinesis Data Analytics

Apache Beam è un modello di programmazione per l'elaborazione di dati di streaming Apache Beam fornisce un livello API trasferibile per la costruzione di sofisticate pipeline di elaborazione parallela di dati che possono essere eseguite su una varietà di motori o di runner come Flink, Spark Streaming, Apache Samza e così via.

È possibile utilizzare il framework Apache Beam con l'applicazione di analisi dei dati Kinesis per elaborare i dati di streaming. Le applicazioni di analisi dei dati Kinesis che utilizzano Apache Beam utilizzano il runner Apache Flink per eseguire le pipeline Beam.

Riepilogo

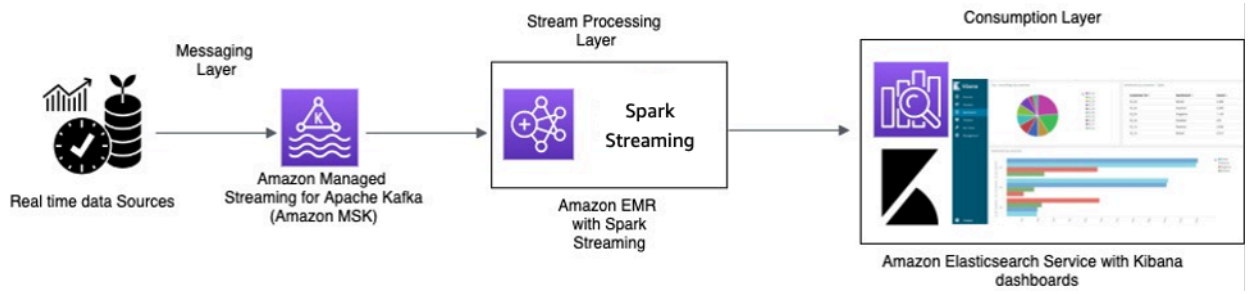
Utilizzando i servizi di streaming di AWS Amazon Kinesis Data Streams, Amazon Kinesis Data Analytics e Amazon Kinesis Data Firehose,

ABC4logistics è in grado di rilevare modelli anomali nelle letture della temperatura e avvisare il conducente e il team di gestione della flotta in tempo reale, prevenendo incidenti gravi come guasti totali del veicolo o incendi.

Scenario 5: monitoraggio dei dati di telemetria in tempo reale con Apache Kafka

ABC1cabs è una società di servizi di prenotazione di taxi online. Tutti taxi dispongono di dispositivi IoT che raccolgono dati di telemetria dai veicoli. Attualmente, ABC1Cabs esegue cluster Apache Kafka progettati per il consumo di eventi in tempo reale, la raccolta di parametri di integrità del sistema, il monitoraggio delle attività e l'inserimento dei dati nella piattaforma Apache Spark Streaming basata su un cluster Hadoop On-Premise.

ABC1Cabs utilizza OpenSearch Dashboards per parametri aziendali, debug, avvisi e creazione di altri pannelli di controllo. Sono interessati ad Amazon MSK, Amazon EMR con Spark Streaming e a OpenSearch Service con OpenSearch Dashboards. Il requisito è ridurre il sovraccarico amministrativo per la gestione dei cluster Apache Kafka e Hadoop, utilizzando al contempo software open source e API note per orchestrare la pipeline dei dati. Il seguente diagramma di architettura mostra la soluzione su AWS.



Elaborazione in tempo reale con Amazon MSK ed elaborazione Stream mediante Apache Spark Streaming su Amazon EMR e Amazon OpenSearch Service con OpenSearch Dashboards

I dispositivi IoT dei taxi raccolgono dati di telemetria e li inviano a un hub di origine. L'hub di origine è configurato per inviare dati in tempo reale ad Amazon MSK. Utilizzando le API della libreria producer Apache Kafka, Amazon MSK è configurato per la trasmissione dei dati in un cluster Amazon EMR. Il cluster Amazon EMR ha un client Kafka e Spark Streaming installati per poter utilizzare ed elaborare i flussi di dati.

Spark Streaming ha connettori sink che possono scrivere dati direttamente su indici definiti di Elasticsearch. I cluster Elasticsearch con OpenSearch Dashboards possono essere utilizzati per parametri e pannelli di controllo. Amazon MSK, Amazon EMR con Spark Streaming e OpenSearch Service con OpenSearch Dashboards sono tutti servizi gestiti, in cui AWS gestisce il pesante carico indifferenziato della gestione dell'infrastruttura di diversi cluster, che ti consente di costruire la tua

applicazione utilizzando un software open source noto con pochi clic. La sezione successiva esamina più da vicino questi servizi.

Amazon Managed Streaming for Apache Kafka (Amazon MSK)

Apache Kafka è una piattaforma open source che consente ai clienti di acquisire dati di streaming come eventi di flussi di clic, transazioni, eventi IoT e registri di applicazioni e computer. Con queste informazioni, è possibile sviluppare applicazioni che eseguono analisi dei dati in tempo reale, eseguono trasformazioni continue e distribuiscono questi dati a data lake e database in tempo reale.

È possibile utilizzare Kafka come archivio dati di streaming per disaccoppiare le applicazioni producer e consumer e consentire un trasferimento affidabile dei dati tra i due componenti. Sebbene Kafka sia una popolare piattaforma aziendale di streaming dei dati e messaggistica, può essere difficile da configurare, dimensionare e gestire in produzione.

Amazon MSK si occupa di queste attività di gestione e semplifica l'impostazione, la configurazione e l'esecuzione di Kafka, insieme ad Apache Zookeeper, in un ambiente che segue le best practice per la disponibilità e la sicurezza elevate. È ancora possibile utilizzare le operazioni del piano di controllo e le operazioni del piano dati di Kafka per gestire la produzione e il consumo di dati.

Poiché Amazon MSK esegue e gestisce Apache Kafka open source, semplifica la migrazione e l'esecuzione delle applicazioni Apache Kafka esistenti in AWS senza dover apportare modifiche al codice dell'applicazione.

Dimensionamento

Amazon MSK offre operazioni di dimensionamento in modo che l'utente possa dimensionare attivamente il cluster durante l'esecuzione. Quando crei un cluster Amazon MSK, puoi specificare il tipo di istanza dei broker al momento dell'avvio del cluster. Puoi iniziare con alcuni broker all'interno di un cluster Amazon MSK. Quindi, utilizzando la AWS Management Console o AWS CLI, puoi aumentare fino a centinaia di broker per cluster.

In alternativa, puoi dimensionare i tuoi cluster cambiando le dimensioni o la famiglia dei tuoi broker Apache Kafka. Cambiando le dimensioni o la famiglia dei tuoi broker, otterrai la flessibilità necessaria per adattare la capacità di elaborazione nel cluster Amazon MSK ed effettuare delle modifiche ai carichi di lavoro. Utilizza il foglio di calcolo Dimensionamento e prezzi di Amazon MSK (download del file) per determinare il numero corretto di broker per il tuo cluster Amazon MSK. Questo foglio di calcolo fornisce una stima del ridimensionamento di un cluster Amazon MSK e dei costi associati di Amazon MSK rispetto a un cluster Apache Kafka simile basato su EC2 e autogestito.

Dopo aver creato il cluster Amazon MSK, puoi aumentare la quantità di archiviazione EBS per broker, ad eccezione della riduzione dello spazio di archiviazione. I volumi di archiviazione rimangono disponibili durante questa operazione di dimensionamento. Offre due tipi di operazioni di dimensionamento: scalabilità automatica e manuale.

Amazon MSK supporta l'espansione automatica dell'archiviazione del cluster in risposta a un maggiore utilizzo utilizzando le policy di scalabilità automatica delle applicazioni. La policy di scalabilità automatica imposta l'utilizzo del disco di destinazione e la capacità di dimensionamento massimo.

La soglia di utilizzo dell'archiviazione aiuta Amazon MSK ad attivare un'operazione di scalabilità automatica. Per aumentare lo spazio di archiviazione utilizzando il dimensionamento manuale, attendi che il cluster si trovi nello stato `ACTIVE`. Il dimensionamento dell'archiviazione ha un tempo di raffreddamento di almeno sei ore tra un evento e l'altro. Anche se l'operazione rende immediatamente disponibile archiviazione aggiuntiva, il servizio esegue ottimizzazioni sul cluster che possono richiedere fino a 24 ore o più.

La durata di queste ottimizzazioni è proporzionale alle dimensioni dell'archiviazione. Inoltre, offre anche la replica di zone di disponibilità multipla all'interno di una regione AWS per fornire elevata disponibilità.

Configurazione

Amazon MSK fornisce una configurazione di default per broker, argomenti e nodi Apache ZooKeeper. Puoi inoltre creare configurazioni personalizzate e utilizzarle per creare nuovi cluster Amazon MSK o per aggiornare cluster esistenti. Quando crei un cluster MSK senza specificare una configurazione Amazon MSK personalizzata, Amazon MSK crea e utilizza una configurazione di default. Per un elenco dei valori di default, consulta la pagina relativa alla configurazione di Apache Kafka.

Per scopi di monitoraggio, Amazon MSK raccoglie i parametri di Apache Kafka e li invia ad Amazon CloudWatch, dove è possibile visualizzarli. I parametri configurati per i cluster MSK sono automaticamente raccolti e inviati a CloudWatch. Il monitoraggio del ritardo dei consumer consente di identificare quelli lenti o bloccati che non sono al passo con i dati più recenti disponibili in un argomento. Se necessario, è quindi possibile intraprendere operazioni correttive, come il dimensionamento o il riavvio di tali consumer.

Migrazione ad Amazon MSK

La migrazione dall'ambiente on-premise ad Amazon MSK può essere effettuata con uno dei seguenti metodi.

- MirrorMaker2.0: MirrorMaker2.0 (MM2) MM2 è un motore di replica di dati multi-cluster basato sul framework Apache Kafka Connect. MM2 è una combinazione di un connettore origine Apache Kafka e un connettore sink. È possibile utilizzare un singolo cluster MM2 per migrare i dati tra più cluster. MM2 rileva automaticamente nuovi argomenti e partizioni, garantendo allo stesso tempo che le configurazioni degli argomenti siano sincronizzate tra i cluster. MM2 supporta ACL di migrazioni, configurazioni di argomenti e conversione di offset. Per ulteriori dettagli sulla migrazione, consulta Migrazione di cluster utilizzando MirrorMaker di Apache Kafka. MM2 viene utilizzato per casi d'uso relativi alla replica di configurazioni di argomenti e conversione automatica di offset.
- Apache Flink: MM2 supporta la semantica "exactly-once". I record possono essere duplicati nella destinazione ed è previsto che i consumer siano idempotenti per gestire i record duplicati. In scenari "exactly-once", la semantica è richiesta ai clienti che possono utilizzare Apache Flink. Fornisce un'alternativa per la semantica "exactly-once".

  Apache Flink può essere utilizzato anche per scenari in cui i dati richiedono operazioni di mappatura o trasformazione prima dell'invio al cluster di destinazione. Apache Flink fornisce connettori per Apache Kafka con origini e sink in grado di leggere i dati da un cluster Apache Kafka e scrivere su un altro. Apache Flink può essere eseguito su AWS avviando un cluster Amazon EMR o eseguendo Apache Flink come applicazione utilizzando Amazon Kinesis Data Analytics.
- AWS Lambda: con il supporto per Apache Kafka come origine di eventi per AWS Lambda, i clienti possono ora utilizzare messaggi da un argomento tramite una funzione Lambda. Il servizio AWS Lambda esegue internamente il polling per nuovi record o messaggi dall'origine eventi, quindi richiama in modo sincrono la funzione Lambda di destinazione per consumare i messaggi. Lambda legge i messaggi in batch e fornisce i batch di messaggi alla funzione nel payload dell'evento per l'elaborazione. I messaggi consumati possono quindi essere trasformati e/o scritti direttamente nel cluster Amazon MSK di destinazione.

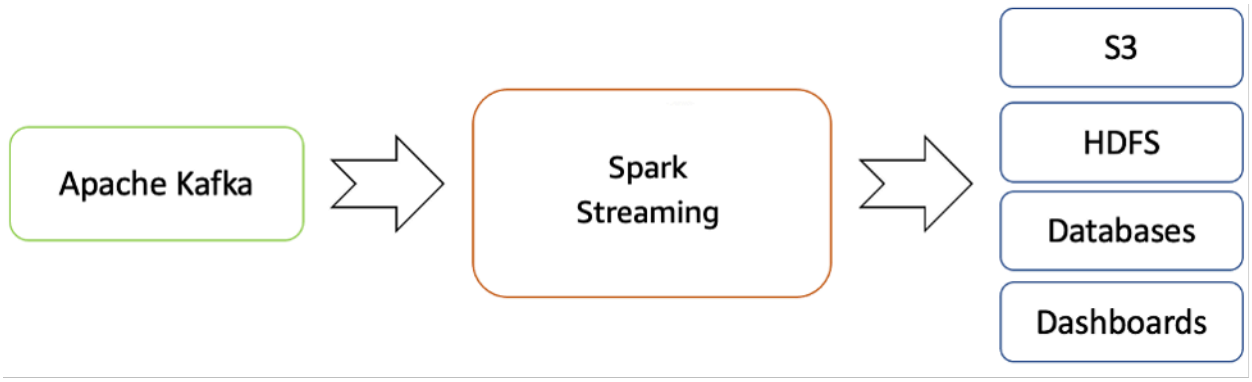
Amazon EMR con Spark Streaming

Amazon EMR è una piattaforma di cluster gestita che semplifica l'esecuzione di grandi framework di dati, come ad esempio Apache Hadoop e Apache Spark, su AWS per elaborare e analizzare grandi quantità di dati.

Amazon EMR offre le funzionalità di Spark e può essere utilizzato per avviare Spark Streaming Spark per consumare dati da Kafka. Spark Streaming è un'estensione dell'API Spark principale che consente l'elaborazione di flussi dei dati in tempo reale scalabile, a velocità effettiva elevata e tollerante ai guasti.

Puoi creare un cluster Amazon EMR utilizzando AWS Command Line Interface (AWS CLI) o sulla AWS Management Console e selezionare Spark e Zeppelin nelle configurazioni avanzate durante la creazione del cluster. Come mostrato nel seguente diagramma di architettura, i dati possono essere acquisiti da molte fonti come Apache Kafka e Kinesis Data Streams e possono essere elaborati utilizzando algoritmi complessi espressi con funzioni di alto livello come map, reduce, join e window. Per ulteriori informazioni, consulta Transformations on DStreams.

I dati elaborati possono essere inviati a file system, database e pannelli di controllo.



Flusso di streaming in tempo reale da Apache Kafka all'ecosistema Hadoop

Di default, Apache Spark Streaming ha un modello di esecuzione in micro-batch. Tuttavia, a partire da Spark 2.3, Apache ha introdotto una nuova modalità di elaborazione a bassa latenza chiamata Continuous Processing, che può raggiungere latenze end-to-end fino a un millisecondo con garanzie "at-least-once".

Senza modificare le operazioni DataSet/DataFrames nelle query, è possibile scegliere la modalità in base ai requisiti dell'applicazione. Alcuni dei vantaggi di Spark Streaming sono:

- Utilizza l'API integrata nel linguaggio di Apache Spark per l'elaborazione dello streaming, consentendoti di scrivere processi di streaming nello stesso modo in cui scrivi processi batch.
- Supporta Java, Scala e Python.
- Può recuperare sia il lavoro perso che lo stato dell'operatore (come finestre scorrevoli) immediatamente, senza alcun codice aggiuntivo da parte dell'utente.
- In esecuzione su Spark, Spark Streaming consente di riutilizzare lo stesso codice per l'elaborazione batch, unire i flussi con dati cronologici o eseguire query ad hoc sullo stato del flusso e costruire potenti applicazioni interattive, non solamente l'analisi dei dati.
- Dopo che il flusso di dati è stato elaborato con Spark Streaming, OpenSearch Sink Connector può essere utilizzato per scrivere dati nel cluster OpenSearch Service e, a sua volta, OpenSearch Service con OpenSearch Dashboards può essere utilizzato come livello di consumo.

Amazon OpenSearch Service con OpenSearch Dashboards

OpenSearch Service è un servizio gestito che semplifica implementazione, funzionamento e dimensionamento di cluster OpenSearch in AWS Cloud. OpenSearch è un diffuso motore di ricerca e analisi dei dati open source per casi d'uso come analisi dei dati dei registri, monitoraggio delle applicazioni in tempo reale e analisi dei clickstream.

OpenSearch Dashboards sono uno strumento di visualizzazione ed esplorazione dei dati utilizzato per analisi dei dati dei registri e serie temporali, monitoraggio delle applicazioni e casi d'uso di intelligenza operativa. Offre caratteristiche potenti e di facile utilizzo come istogrammi, grafici a linee, grafici a torta, mappe di calore e supporto geospaziale integrato.

OpenSearch Dashboards fornisce una perfetta integrazione con OpenSearch, un noto motore di analisi dei dati e di ricerca, che rende OpenSearch Dashboards la scelta di default per la visualizzazione dei dati archiviati inOpenSearch. OpenSearch Service fornisce un'installazione di OpenSearch Dashboards con ogni dominio OpenSearch Service. Puoi trovare un link a OpenSearch Dashboards nel pannello di controllo del tuo dominio sulla console OpenSearch Service.

Riepilogo

Con Apache Kafka offerto come servizio gestito su AWS, puoi concentrarti sul consumo piuttosto che sulla gestione del coordinamento tra i broker, che di solito richiede una conoscenza approfondita di Apache Kafka. Caratteristiche come l'elevata disponibilità, la scalabilità dei broker e il controllo granulare degli accessi sono gestite dalla piattaforma Amazon MSK.

ABC1CAB ha utilizzato questi servizi per costruire applicazioni di produzione senza bisogno di competenze nella gestione dell'infrastruttura. È stato possibile concentrarsi sul livello di elaborazione per consumare dati da Amazon MSK e propagare ulteriormente al livello di visualizzazione.

Spark Streaming su Amazon EMR può aiutare l'analisi dei dati di streaming in tempo reale e la pubblicazione su OpenSearch Dashboards su Amazon OpenSearch Service per il livello di visualizzazione.

Conclusione e collaboratori

Conclusione

Questo documento ha esaminato diversi scenari per i flussi di lavoro in streaming. In questi scenari, l'elaborazione dei dati di streaming ha fornito alle aziende presentate come esempi la possibilità di aggiungere nuove caratteristiche e funzionalità.

Analizzando i dati man mano che vengono creati, si otterranno informazioni dettagliate su come sta operando l'azienda. I servizi di streaming di AWS consentono di concentrarsi sulle proprie applicazioni per prendere decisioni aziendali urgenti, piuttosto che implementare e gestire l'infrastruttura

Collaboratori

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

Revisioni del documento

Per ricevere una notifica sugli aggiornamenti di questo whitepaper, iscriviti al feed RSS.

update-history-change	update-history-description	update-history-date
Aggiornato	Aggiornato per l'accuratezza tecnica	1 settembre 2021
Pubblicazione originale	Prima pubblicazione del whitepaper	1 luglio 2017