Livre blanc AWS

Solutions de données de streaming sur AWS avec Amazon Kinesis

Copyright © Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Solutions de données de streaming sur AWS avec Amazon Kinesis: Livre blanc AWS

Copyright © Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Les marques et l'habillage commerciaux d'Amazon ne peuvent pas être utilisés en connexion avec un produit ou un service qui n'est pas celui d'Amazon, d'une manière susceptible de causer de la confusion chez les clients ou d'une manière qui dénigre ou discrédite Amazon. Toutes les autres marques commerciales qui ne sont pas la propriété d'Amazon sont la propriété de leurs propriétaires respectifs, qui peuvent ou non être affiliés ou connectés à Amazon, ou sponsorisés par Amazon.

Table of Contents

Résumé 1
Résumé 1
Introduction 2
Scénarios d'application en temps réel et quasi réel 2
Différence entre traitement par lots et en streaming 3
Défis liés au traitement des flux 3
Exemples de solution de données de streaming 5
Scénario 1 : offre Internet basée sur la localisation 5
Amazon Kinesis Data Streams 5
Traitement de flux de données avec AWS Lambda 8
Récapitulatif 8
Scénario 2 : données en temps quasi réel pour les équipes en charge de la sécurité 9
Amazon Kinesis Data Firehose 10
Récapitulatif 15
Scénario 3 : préparation des données de parcours de navigation pour les processus d'informations sur les données 16
AWS Glue et AWS Glue streaming 17
Amazon DynamoDB 18
Amazon SageMaker et points de terminaison de service Amazon SageMaker 19
Déduction d'informations sur les données en temps réel 20
Récapitulatif 20
Scénario 4 : détection et notification en temps réel des anomalies de capteurs 21
Amazon Kinesis Data Analytics 22
Amazon Kinesis Data Analytics pour applications Apache Flink 22
Scénario 5 : surveillance des données de télémétrie en temps réel avec Apache Kafka 25
Amazon Managed Streaming pour Apache Kafka (Amazon MSK) 26
Migration vers Amazon MSK 28
Conclusion et contributeurs 32
Conclusion 32
Participants 32
Révisions du document 33

Solutions de données de streaming sur AWS

Date de publication : 1er septembre 2021 (Révisions du document)

Résumé

Les ingénieurs de données, les analystes de données et les développeurs big data cherchent à faire évoluer leurs analyses du traitement par lots vers le traitement en temps réel afin que leurs entreprises puissent en savoir plus sur ce que font leurs clients, leurs applications et leurs produits en ce moment et réagir rapidement. Ce livre blanc traite de l'évolution de l'analytique du traitement par lot vers le traitement en temps réel. Il décrit comment des services tels qu'Amazon Kinesis Data Streams, Amazon Kinesis Data Firehose, Amazon EMR, Amazon Kinesis Data Analytics, Amazon Managed Streaming for Apache Kafka (Amazon MSK) et d'autres services peuvent être utilisés pour mettre en œuvre des applications en temps réel. Il propose également des modèles de conception courants utilisant ces services.

Introduction

Aujourd'hui, les entreprises reçoivent des données à une échelle et à une vitesse considérables en raison de la croissance explosive des sources de données qui génèrent des flux continus. Qu'il s'agisse de données de journaux provenant de serveurs d'applications, de données de parcours de navigation provenant de sites Web et d'applications mobiles ou de données de télémétrie provenant d'appareils connectés à l'Internet des objets (IoT), elles contiennent toutes des informations qui peuvent vous aider à savoir ce que font vos clients, vos applications et vos produits en ce moment même.

Avoir la capacité de traiter et d'analyser ces données en temps réel est essentiel pour effectuer des tâches telles que la surveillance continue de vos applications. Cela vous permet d'assurer une disponibilité élevée du service et de personnaliser les offres promotionnelles et les recommandations de produits. Le traitement en temps réel et quasi réel peut également rendre d'autres cas d'utilisation courants (tels que l'analyse de sites Web et le Machine Learning) plus précis et exploitables en mettant les données à la disposition de ces applications en quelques secondes ou quelques minutes au lieu de plusieurs heures ou plusieurs jours.

Scénarios d'application en temps réel et quasi réel

Vous pouvez utiliser les services de données de streaming pour des applications en temps réel et quasi réel, notamment la surveillance des applications, la détection des fraudes et les classements dynamiques. Les cas d'utilisation en temps réel nécessitent des temps de latence de bout en bout de l'ordre de quelques millisecondes, depuis l'ingestion jusqu'au traitement en passant par l'émission des résultats vers les magasins de données cibles et d'autres systèmes. Par exemple, Netflix utilise Amazon Kinesis Data Streams pour surveiller les communications entre chacune de ses applications, afin de détecter et de corriger rapidement les problèmes et d'offrir une disponibilité d'exception à ses clients. Bien que le cas d'utilisation le plus courant soit la surveillance des performances des applications, un nombre croissant d'applications en temps réel dans les domaines de la technologie publicitaire, des jeux et de l'IoT entrent dans cette catégorie.

Les cas d'utilisation courants en temps quasi réel comprennent l'analyse des magasins de données pour la science des données et le Machine Learning (ML). Vous pouvez utiliser des solutions de données de streaming pour charger en continu des données en temps réel dans vos lacs de données. Vous pouvez ensuite mettre à jour plus fréquemment les modèles de Machine Learning à mesure que de nouvelles données sont disponibles, pour des résultats plus fiables et plus précis. Par exemple, Zillow utilise Kinesis Data Streams pour collecter les données d'enregistrement public

et de multiples listes de services d'annonces (MLS), puis fournir aux acheteurs et aux vendeurs de maisons les estimations les plus récentes de la valeur des maisons en temps quasi réel. ZipRecruiter utilise Amazon MSK pour ses pipelines de consignations d'événements. Ce sont des composants essentiels de l'infrastructure qui collectent, stockent et traitent en permanence plus de 6 milliards d'événements par jour depuis le site d'emplois ZipRecruiter.

Différence entre traitement par lots et en streaming

Pour collecter, préparer et traiter des données de streaming en temps réel, vous avez besoin d'un ensemble d'outils différents de ceux que vous utilisez traditionnellement pour l'analyse par lots. Avec l'analytique traditionnelle, vous collectez les données, les chargez périodiquement dans une base de données pour les analyser des heures, des jours ou des semaines plus tard. L'analyse des données en temps réel nécessite une approche différente. Les applications de traitement en streaming traitent les données en continu et en temps réel, avant même qu'elles ne soient stockées. Les données de streaming peuvent arriver à un rythme effréné et les volumes de données peuvent varier à la hausse ou à la baisse à tout moment. Les plateformes de traitement des données de streaming doivent être capables de gérer la vitesse et la variabilité des données entrantes et de les traiter au fur et à mesure qu'elles arrivent, souvent par millions voire centaines de millions d'événements par heure.

Défis liés au traitement des flux

Le traitement des données en temps réel au fur et à mesure de leur arrivée peut vous permettre de prendre des décisions beaucoup plus rapidement qu'avec les technologies d'analytique de données traditionnelles. Cependant, la création et l'exploitation de vos propres pipelines de données de streaming personnalisés sont complexes et nécessitent beaucoup de ressources :

- Vous devez créer un système capable de collecter, de préparer et de transmettre de manière rentable des données provenant simultanément de milliers de sources de données.
- Vous devez affiner les ressources de stockage et de calcul afin que les données soient mises en lots et transmises efficacement pour un débit maximal et une faible latence.
- Vous devez déployer et gérer une flotte de serveurs pour mettre le système à l'échelle afin de pouvoir gérer les différentes vitesses de données que vous allez lui envoyer.

La mise à niveau de version est un processus complexe et coûteux. Après avoir créé cette plateforme, vous devez surveiller le système et compenser toute panne de serveur ou de réseau en reprenant le traitement des données au point approprié du flux, sans créer de données en double.

Vous avez également besoin d'une équipe dédiée à la gestion de l'infrastructure. Tout cela demande du temps et de l'argent. En fin de compte, la plupart des entreprises n'y arrivent jamais et doivent se contenter du statu quo : faire fonctionner leur entreprise avec des informations qui datent de quelques heures ou de quelques jours.

Exemples de solution de données de streaming

Scénario 1 : offre Internet basée sur la localisation

La société InternetProvider fournit des services Internet avec une variété d'options de bande passante à des utilisateurs du monde entier. Lorsqu'un utilisateur s'inscrit à Internet, la société InternetProvider lui propose différentes options de bande passante en fonction de sa localisation géographique. Compte tenu de ces exigences, la société InternetProvider a mis en œuvre un service Amazon Kinesis Data Streams pour exploiter les détails et la localisation des utilisateurs. Les détails et la localisation des utilisateurs sont enrichis par différentes options de bande passante avant la publication dans l'application. AWS Lambda permet cet enrichissement en temps réel.

Traitement de flux de données avec AWS Lambda

Amazon Kinesis Data Streams

Amazon Kinesis Data Streams vous permet de créer des applications personnalisées en temps réel grâce à des cadres de traitement de flux populaires et de charger des données de streaming dans de nombreux magasins de données différents. Un flux Kinesis peut être configuré pour recevoir en continu des événements de centaines de milliers de producteurs de données provenant de sources telles que des flux de clics sur un site Web, des capteurs IoT, des flux de réseaux sociaux et des journaux d'applications. En quelques millisecondes, les données sont disponibles pour lecture et traitement par votre application.

Lors de la mise en œuvre d'une solution avec Kinesis Data Streams, vous créez des applications de traitement de données personnalisées appelées applications Kinesis Data Streams. Une application Kinesis Data Streams classique lit les données d'un flux Kinesis en tant qu'enregistrements de données.

Les données intégrées à Kinesis Data Streams sont garanties pour être hautement disponibles et élastiques, et sont disponibles en quelques millisecondes. Vous pouvez ajouter en permanence différents types de données à un flux Kinesis, notamment des parcours de navigation, des journaux d'application et des données de réseaux sociaux, et ce, depuis plusieurs centaines de milliers de sources. En quelques secondes, ces données sont accessibles à vos applications Kinesis qui peuvent les lire et les traiter depuis le flux.

Amazon Kinesis Data Streams est un service de données de streaming entièrement géré. Il gère l'infrastructure, le stockage, la mise en réseau et la configuration nécessaires à la diffusion en continu de vos données, en s'adaptant au débit de celles-ci.

Envoi de données à Amazon Kinesis Data Streams

Il existe plusieurs manières d'envoyer des données à Kinesis Data Streams, ce qui vous permet de concevoir vos solutions de manière flexible.

- Vous pouvez écrire du code en utilisant un des kits SDK AWS pris en charge par plusieurs langages populaires.
- Vous pouvez utiliser Amazon Kinesis Agent, un outil permettant d'envoyer des données à Kinesis Data Streams.

La bibliothèque Amazon Kinesis Producer Library (KPL) simplifie le développement d'applications producteur en permettant aux développeurs d'atteindre un débit d'écriture élevé pour un ou plusieurs flux de données Kinesis.

Facile à utiliser et hautement configurable, la bibliothèque KPL peut être installée sur vos hôtes. Elle sert d'intermédiaire entre le code de votre application producteur et les actions de l'API Kinesis Streams. Pour plus d'informations sur la bibliothèque KPL et sa capacité à produire des événements de manière synchrone et asynchrone à l'aide d'exemples de code, reportez-vous à Écriture dans le flux de données Kinesis à l'aide de KPL.

Dans l'API Kinesis Data Streams, deux opérations différentes permettent d'ajouter des données à un flux : `PutRecords` et `PutRecord`. L'opération `PutRecords` envoie plusieurs enregistrements à votre flux par demande HTTP tandis que `PutRecord` soumet un enregistrement par demande HTTP. Pour obtenir un débit plus élevé pour la plupart des applications, utilisez `PutRecords`.

Pour de plus amples informations sur ces API, consultez Ajout de données à un flux. Vous trouverez les détails de chaque opération d'API dans le document Amazon Kinesis Data Streams API Reference.

Traitement des données dans Amazon Kinesis Data Streams

Pour lire et traiter les données des flux Kinesis, vous devez créer une application consommateur. Il existe différentes manières de créer des consommateurs pour Kinesis Data Streams. Certaines de ces approches incluent l'utilisation d'Amazon Kinesis Data Analytics pour analyser les données de streaming à l'aide de la KCL, l'utilisation de AWS Lambda, l'utilisation de tâches ETL en streaming AWS Glue et l'utilisation directe de l'API Kinesis Data Streams.

Les applications consommateur pour Kinesis Data Streams peuvent être développées à l'aide de la KCL, qui vous permet de consommer et de traiter les données à partir de Kinesis Data Streams. La KCL prend en charge de nombreuses tâches complexes associées à l'informatique distribuée, telles que l'équilibrage de charge sur plusieurs instances, la réponse aux défaillances d'instance, la vérification des enregistrements traités et la réaction au repartitionnement. La KCL vous permet de vous concentrer sur l'écriture de la logique de traitement des enregistrements. Pour plus d'informations sur la façon de créer votre propre application KCL, veuillez consulter Utilisation de la bibliothèque client Kinesis.

Vous pouvez vous abonner à des fonctions Lambda pour lire automatiquement des lots d'enregistrements de votre flux Kinesis et les traiter si des enregistrements sont détectés dans le flux. AWS Lambda interroge périodiquement le flux (une fois par seconde) et, lorsqu'il détecte de nouveaux enregistrements, appelle la fonction Lambda en transmettant ces nouveaux enregistrements en tant que paramètres. La fonction Lambda n'est exécutée que lorsque de nouveaux enregistrements sont détectés. Vous pouvez mapper une fonction Lambda à un consommateur à débit partagé (itérateur standard).

Vous pouvez créer un consommateur qui utilise une fonction appelée sortance améliorée lorsque vous avez besoin d'un débit dédié que vous ne voulez pas opposer aux autres consommateurs qui reçoivent des données du flux. Cette fonction permet aux applications consommateur de recevoir des enregistrements provenant d'un flux avec un débit pouvant atteindre 2 Mo de données par seconde par partition.

Dans la plupart des cas, avec Kinesis Data Analytics, la bibliothèque client Kinesis, AWS Glue ou AWS Lambda doit être utilisé pour traiter les données d'un flux. Toutefois, si vous préférez, vous pouvez créer une application consommateur à l'aide de l'API Kinesis Data Streams. L'API Kinesis Data Streams fournit les méthodes `GetShardIterator` et `GetRecords` pour extraire les données d'un flux.

Dans ce modèle d'extraction, votre code extrait les données directement depuis les partitions du flux. Pour de plus amples informations sur l'écriture de votre propre application consommateur à l'aide de

l'API, consultez Développement d'applications consommateur personnalisées avec un débit partagé avec AWS SDK for Java. Vous trouverez des détails sur l'API dans le document Amazon Kinesis Data Streams API Reference.

Traitement de flux de données avec AWS Lambda

AWS Lambda vous permet d'exécuter du code sans avoir à mettre en service ou gérer des serveurs. Avec Lambda, vous pouvez exécuter le code pour quasiment n'importe quel type d'application ou service principal, sans avoir à vous préoccuper de leur administration. Il vous suffit de télécharger votre code et Lambda s'occupe de tout ce qui est nécessaire à l'exécution de votre code et à son évolution en garantissant une haute disponibilité. Vous pouvez configurer votre code de sorte qu'il se déclenche automatiquement depuis d'autres services AWS, ou l'appeler directement à partir de n'importe quelle application Web ou mobile.

AWS Lambda s'intègre en mode natif à Amazon Kinesis Data Streams. Les complexités liées à l'interrogation, à la création de points de contrôle et à la gestion des erreurs sont supprimées lorsque vous utilisez cette intégration native. Cela permet au code de fonction Lambda de se concentrer sur le traitement de la logique métier.

Vous pouvez mapper une fonction Lambda à un débit partagé (itérateur standard) ou à un consommateur à débit dédié avec diffusion améliorée. Avec un itérateur standard, Lambda interroge chaque partition de votre flux Kinesis pour des enregistrements qui utilisent le protocole HTTP. Pour réduire la latence et optimiser le débit en lecture, vous pouvez créer un consommateur de flux de données avec diffusion améliorée. Les consommateurs de flux de cette architecture obtiennent une connexion dédiée à chaque partition sans entrer en concurrence avec les autres applications lisant le même flux. Amazon Kinesis Data Streams pousse les enregistrements vers Lambda via HTTP/2.

Par défaut, AWS Lambda appelle votre fonction dès que des enregistrements sont disponibles dans le flux. Pour mettre en mémoire tampon les enregistrements des scénarios de traitement par lots, vous pouvez mettre en œuvre une fenêtre de traitement par lots pendant cinq minutes maximum à la source de l'événement. Si votre fonction renvoie une erreur, Lambda procède à des tentatives de traitement du lot jusqu'à ce qu'il y parvienne ou que les données expirent.

Récapitulatif

La société InternetProvider a exploité Amazon Kinesis Data Streams pour diffuser les détails et la localisation des utilisateurs. Le flux d'enregistrement a été utilisé par AWS Lambda pour enrichir les données avec des options de bande passante stockées dans la bibliothèque de la fonction.

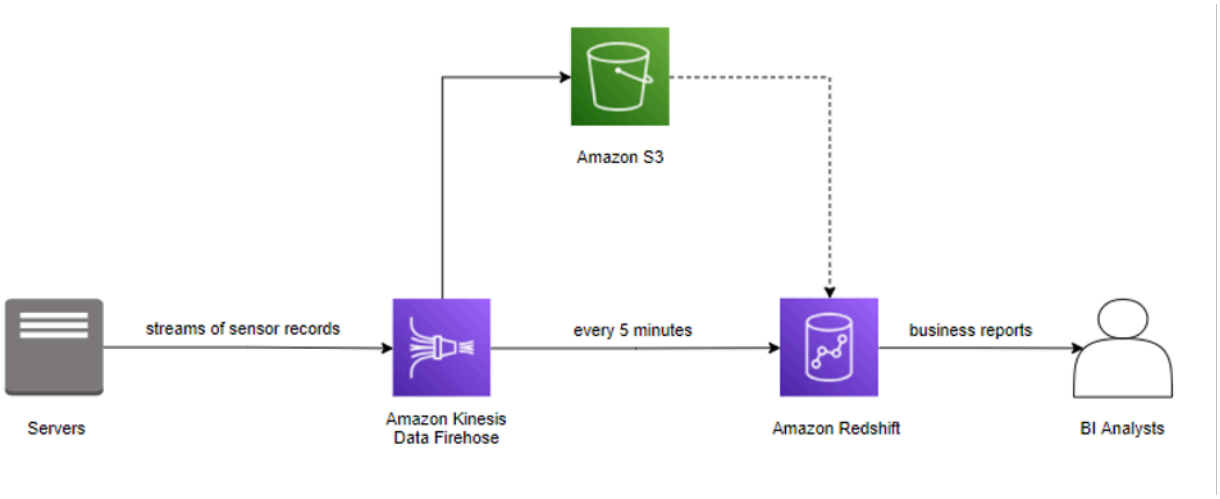
Après l'enrichissement, AWS Lambda a de nouveau publié les options de bande passante dans l'application. Amazon Kinesis Data Streams et AWS Lambda ont géré l'approvisionnement et la gestion des serveurs, permettant ainsi à la société InternetProvider de se concentrer davantage sur le développement d'applications métier.

Scénario 2 : données en temps quasi réel pour les équipes en charge de la sécurité

La société ABC2Badge fournit des capteurs et des badges pour les événements d'entreprise ou de grande envergure tels que l'AWS re:Invent. Les utilisateurs s'inscrivent à l'événement et reçoivent des badges uniques que les capteurs détectent sur l'ensemble du campus. Lorsque les utilisateurs passent devant un capteur, leurs informations anonymisées sont enregistrées dans une base de données relationnelle.

Dans le cadre d'un futur événement et en raison du nombre élevé de participants, l'équipe en charge de la sécurité a demandé à ABC2Badge de collecter des données toutes les 15 minutes pour les zones les plus concentrées du campus. Cela donnera à l'équipe en charge de la sécurité suffisamment de temps pour réagir et répartir le personnel de sécurité proportionnellement aux zones concentrées. Compte tenu de cette nouvelle exigence de l'équipe en charge de la sécurité et du manque d'expérience en matière de création d'une solution de streaming, ABC2Badge recherche une solution simple, mais évolutive et fiable, pour traiter les données en temps quasi réel.

Leur solution d'entrepôt de données actuelle est Amazon Redshift. Lors de l'examen des fonctions des services Amazon Kinesis, ils ont constaté qu'Amazon Kinesis Data Firehose pouvait recevoir un flux d'enregistrements de données, regrouper les enregistrements en fonction de la taille de la mémoire tampon et/ou de l'intervalle de temps, et les insérer dans Amazon Redshift. Ils ont créé un flux de diffusion Kinesis Data Firehose et l'ont configuré de sorte qu'il copie les données vers leurs tables Amazon Redshift toutes les cinq minutes. Dans le cadre de cette nouvelle solution, ils ont utilisé l'agent Amazon Kinesis sur leurs serveurs. Toutes les cinq minutes, Kinesis Data Firehose charge les données dans Amazon Redshift. L'équipe Business Intelligence (BI) peut alors les analyser et envoyer les données à l'équipe en charge de la sécurité toutes les 15 minutes.



Nouvelle solution avec Amazon Kinesis Data Firehose

Amazon Kinesis Data Firehose

Amazon Kinesis Data Firehose est la solution la plus simple pour charger des données de streaming dans AWS. Elle peut capturer, transformer et charger des données de streaming dans Amazon Kinesis Data Analytics, Amazon Simple Storage Service (Amazon S3), Amazon Redshift, Amazon OpenSearch Service (OpenSearch Service) et Splunk. En outre, Kinesis Data Firehose peut charger des données de streaming dans n'importe quel point de terminaison HTTP personnalisé ou dans des points de terminaison HTTP appartenant à des fournisseurs de services tiers pris en charge.

Kinesis Data Firehose permet d'effectuer des analyses en temps quasi réel grâce aux outils de Business Intelligence et aux tableaux de bord que vous utilisez déjà aujourd'hui. Ce service sans serveur entièrement géré s'adapte automatiquement à votre débit de données et ne nécessite aucune administration continue. Kinesis Data Firehose peut regrouper, compresser et chiffrer les données avant de les charger, optimisant ainsi la sécurité et limitant le volume de stockage utilisé à l'emplacement de destination. Il peut également transformer les données sources à l'aide d'AWS Lambda et fournir les données transformées aux destinations. Vous pouvez configurer vos producteurs de données pour envoyer des données vers Kinesis Data Firehose, qui diffuse automatiquement les données vers la destination que vous spécifiez.

Envoi de données vers un flux de diffusion Firehose

Pour envoyer des données à votre flux de diffusion, vous disposez de plusieurs options. AWS propose des kits SDK pour de nombreux langages de programmation populaires, chacun d'entre eux fournissant des API pour Amazon Kinesis Data Firehose. AWS dispose d'un utilitaire pour vous aider

à envoyer des données vers votre flux de diffusion. Kinesis Data Firehose a été intégré à d'autres services AWS pour envoyer des données directement depuis ces services vers votre flux de diffusion.

Utilisation d'Amazon Kinesis Agent

Amazon Kinesis Agent est une application logicielle autonome qui surveille en permanence un ensemble de fichiers journaux à la recherche de nouvelles données à envoyer au flux de diffusion. L'agent gère automatiquement la rotation des fichiers, les points de contrôle et les nouvelles tentatives en cas d'échec. Il émet également des métriques Amazon CloudWatch pour la surveillance et le dépannage du flux de diffusion. Des configurations supplémentaires, telles que le prétraitement des données, la surveillance de plusieurs répertoires de fichiers et l'écriture dans plusieurs flux de diffusion, peuvent être appliquées à l'agent.

L'agent peut être installé sur des serveurs Linux ou Windows tels que des serveurs Web, des serveurs de journaux et des serveurs de base de données. Une fois l'agent installé, il vous suffit de spécifier les fichiers journaux qu'il surveillera et le flux de diffusion vers lequel il sera envoyé. L'agent enverra de manière durable et fiable de nouvelles données au flux de diffusion.

Utilisation de l'API avec le kit SDK AWS et les services AWS en tant que source

L'API Kinesis Data Firehose propose deux opérations pour envoyer des données à votre flux de diffusion. `PutRecord` envoie un enregistrement de données par appel. `PutRecordBatch` peut envoyer plusieurs enregistrements de données par appel et atteindre un débit plus élevé par producteur. Dans chaque méthode, vous devez spécifier le nom du flux de diffusion et l'enregistrement de données, ou le tableau d'enregistrements de données, lorsque vous utilisez cette méthode. Pour de plus amples informations et un exemple de code pour les opérations d'API Kinesis Data Firehose, veuillez consulter Écriture dans un flux de diffusion Firehose à l'aide du kit SDK AWS.

Kinesis Data Firehose fonctionne également avec Kinesis Data Firehose, CloudWatch Logs, CloudWatch Events, Amazon Simple Notification Service (Amazon SNS), Amazon API Gateway et AWS IoT. Vous pouvez envoyer de manière évolutive et fiable vos flux de données, vos journaux, vos événements et vos données d'IoT directement dans une destination Kinesis Data Firehose.

Traitement des données avant livraison à leur destination

Dans certains scénarios, vous pouvez souhaiter transformer ou améliorer vos données de streaming avant que celles-ci ne soient livrées à leur destination. Par exemple, des producteurs de données peuvent envoyer du texte non structuré dans chaque enregistrement de données, celui-ci devant être transformé en JSON avant transmission à OpenSearch Service. Vous pouvez également souhaiter

convertir les données JSON dans un format de fichier en colonnes tel qu'Apache Parquet ou Apache ORC avant de stocker les données dans Amazon S3.

Kinesis Data Firehose dispose d'une capacité de conversion de format de données intégrée. Cela vous permet de convertir facilement vos flux de données JSON au format de fichier Apache Parquet ou Apache ORC.

Flux de transformation de données

Pour activer les transformations de données de streaming, Kinesis Data Firehose utilise une fonction Lambda que vous créez pour transformer vos données. Kinesis Data Firehose met en mémoire tampon les données entrantes jusqu'à une taille spécifiée pour la fonction, puis appelle la fonction Lambda spécifiée de manière asynchrone. Les données transformées sont envoyées depuis Lambda vers Kinesis Data Firehose, et Kinesis Data Firehose transmet les données à la destination.

Conversion de format de données

Vous pouvez également activer la conversion de format de données Kinesis Data Firehose, qui convertira votre flux de données JSON au format Apache Parquet ou Apache ORC. Cette fonction ne peut convertir le format JSON qu'en Apache Parquet ou Apache ORC. Si vous avez des données au format CSV, vous pouvez les transformer en données JSON via une fonction Lambda, puis appliquer la conversion de format de données.

Diffusion de données

En tant que flux de diffusion en temps quasi réel, Kinesis Data Firehose met en mémoire tampon les données entrantes. Une fois les seuils de mise en mémoire tampon de votre flux de diffusion atteints, vos données sont transmises à la destination que vous avez configurée. Il existe certaines différences dans la manière dont Kinesis Data Firehose transmet les données à chaque destination. Ces différences sont détaillées dans les sections suivantes.

Amazon S3

Amazon S3 est un service de stockage d'objets doté d'une interface de services Web simple pour stocker et récupérer n'importe quelle quantité de données depuis n'importe où sur le Web. Il est conçu pour être durable à 99,999999999 % et peut traiter plusieurs billions de données dans le monde entier.

Diffusion de données vers Amazon S3

Pour la diffusion de données vers Amazon S3, Kinesis Data Firehose concatène plusieurs enregistrements entrants en fonction de la configuration de mise en mémoire tampon de votre flux de diffusion, puis les transmet à Amazon S3 en tant qu'objet S3. La fréquence de diffusion de données vers S3 est déterminée par la taille du tampon S3 (de 1 Mo à 128 Mo) ou par l'intervalle de mémoire tampon (de 60 secondes à 900 secondes), selon la première éventualité.

La diffusion de données à votre compartiment S3 peut échouer pour plusieurs raisons. Par exemple, le compartiment peut ne plus exister ou le rôle AWS Identity and Access Management (IAM) que Kinesis Data Firehose assume peut ne pas avoir accès au compartiment. Dans ces conditions, Kinesis Data Firehose effectue de nouvelles tentatives pendant 24 heures au maximum, jusqu'à ce que la diffusion réussisse. La durée maximale de stockage des données de Kinesis Data Firehose est de 24 heures. En cas de défaillance de leur diffusion pendant plus de 24 heures, vos données sont perdues.

Amazon Redshift

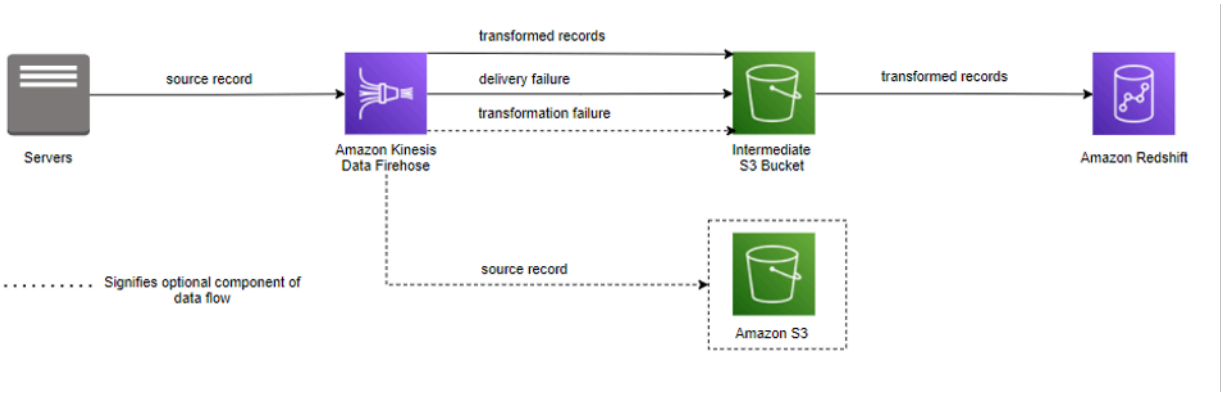
Amazon Redshift est un service d'entrepôt de données rapide et entièrement géré. Il permet d'analyser de manière simple et économique toutes vos données grâce à une syntaxe SQL standard et à vos outils de business intelligence existants. Vous pouvez ainsi exécuter des requêtes analytiques complexes sur plusieurs pétaoctets de données structurées en utilisant l'optimisation de requêtes sophistiquée, le stockage en colonnes sur des disques locaux hautes performances et l'exécution de requêtes massivement parallèle.

Diffusion de données vers Amazon Redshift

Pour la diffusion de données vers Amazon Redshift, Kinesis Data Firehose diffuse d'abord les données entrantes vers votre compartiment S3 dans le format décrit précédemment. Kinesis Data Firehose émet ensuite une commande COPY Amazon Redshift pour charger les données depuis votre compartiment S3 vers votre cluster Amazon Redshift.

La fréquence des opérations `COPY` des données depuis S3 vers Amazon Redshift est déterminée par la rapidité avec laquelle votre cluster Amazon Redshift peut traiter la commande `COPY`. Pour une destination Amazon Redshift, vous pouvez spécifier une durée de nouvelle tentative (de 0 à 7 200 secondes) lors de la création d'un flux de diffusion pour gérer les échecs de diffusion de données. Kinesis Data Firehose procède à de nouvelles tentatives pendant la durée spécifiée et ignore ce lot particulier d'objets S3 en cas d'échec. Les informations concernant les documents ignorés sont transmises à votre compartiment S3 en tant que fichier manifeste dans le dossier errors/, que vous pouvez utiliser pour le renvoi manuel.

Voici un schéma d'architecture du flux de données depuis Kinesis Data Firehose vers Amazon Redshift. Bien que ce flux de données soit propre à Amazon Redshift, Kinesis Data Firehose suit des modèles similaires pour les autres cibles de destination.



Flux de données depuis Kinesis Data Firehose vers Amazon Redshift

Amazon OpenSearch Service (OpenSearch Service)

OpenSearch Service est un service entièrement géré qui fournit des API faciles à utiliser et des capacités en temps réel OpenSearch, ainsi que la disponibilité, la capacité de mise à l'échelle et la sécurité requises par les charges de travail de production. OpenSearch Service facilite le déploiement, l'exploitation et la mise à l'échelle d'OpenSearch pour l'analyse des journaux, la recherche en texte intégral et la surveillance des applications.

Diffusion de données vers OpenSearch Service

Pour la diffusion des données vers OpenSearch Service, Kinesis Data Firehose met en mémoire tampon les enregistrements entrants basés sur la configuration de mise en mémoire tampon de votre flux de diffusion, puis génère une demande groupée OpenSearch pour indexer plusieurs enregistrements dans votre cluster OpenSearch. La fréquence de diffusion des données vers OpenSearch Service est déterminée par la taille de la mémoire tampon OpenSearch (de 1 Mo à 100 Mo) et les valeurs de l'intervalle de mémoire tampon (de 60 secondes à 900 secondes), selon la première éventualité.

Pour la destination OpenSearch Service, vous pouvez spécifier une période de tentative d'envoi (de 0 à 7 200 secondes) lorsque vous créez un flux de diffusion. Kinesis Data Firehose procède à de nouvelles tentatives pendant la durée spécifiée, puis ignore cette demande d'index particulière. Les documents ignorés sont transmis à votre compartiment S3 dans le dossier `elasticsearch_failed/` que vous pouvez utiliser pour le renvoi manuel.

Amazon Kinesis Data Firehose peut procéder à la rotation de votre index OpenSearch Service en fonction d'une certaine durée. Selon l'option de rotation que vous choisissez (`NoRotation`, `OneHour`, `OneDay`, `OneWeek` ou `OneMonth`), Kinesis Data Firehose ajoute une partie de l'horodatage d'arrivée en temps universel coordonné (UTC) au nom d'index que vous avez spécifié.

Point de terminaison HTTP personnalisé ou fournisseur de services tiers pris en charge

Kinesis Data Firehose peut envoyer des données à des points de terminaison HTTP personnalisés ou à des fournisseurs tiers pris en charge tels que Datadog, Dynatrace, LogicMonitor, MongoDB, New Relic, Splunk et Sumo Logic.

Point de terminaison HTTP personnalisé ou fournisseur de services tiers pris en charge

Pour que Kinesis Data Firehose puisse fournir des données à des points de terminaison HTTP personnalisés, ces derniers doivent accepter les demandes et envoyer des réponses en utilisant certains formats de demande et de réponse Kinesis Data Firehose.

Lors de la diffusion de données à un point de terminaison HTTP appartenant à un fournisseur de services tiers pris en charge, vous pouvez utiliser le service AWS Lambda intégré pour créer une fonction permettant de transformer le ou les enregistrements entrants au format correspondant au format attendu par l'intégration du fournisseur de services.

Pour la fréquence de diffusion des données, chaque fournisseur de services a une taille de mémoire tampon recommandée. Contactez votre fournisseur de services pour plus d'informations sur la taille de mémoire tampon recommandée. Pour la gestion des échecs de diffusion de données, Kinesis Data Firehose établit une connexion avec le point de terminaison HTTP en attendant une réponse de la destination. Kinesis Data Firehose continue d'établir la connexion jusqu'à l'expiration de la durée des nouvelles tentatives. Ensuite, Kinesis Data Firehose considère qu'il s'agit d'un échec de diffusion et sauvegarde les données dans votre compartiment S3.

Récapitulatif

Kinesis Data Firehose peut diffuser en permanence vos données de streaming vers une destination prise en charge. Il s'agit d'une solution entièrement gérée qui ne nécessite que peu ou pas de développement. Pour la société ABC2Badge, l'utilisation de Kinesis Data Firehose était un choix naturel. Ils utilisaient déjà Amazon Redshift comme solution d'entrepôt de données. Comme leurs sources de données écrivaient en permanence dans les journaux de transactions, ils ont pu tirer

parti d'Amazon Kinesis Agent pour diffuser ces données sans écrire de code supplémentaire. Maintenant que la société ABC2Badge a créé un flux d'enregistrements de capteurs et reçoit ces enregistrements via Kinesis Data Firehose, elle peut l'utiliser comme base pour le cas d'utilisation de l'équipe en charge de la sécurité.

Scénario 3 : préparation des données de parcours de navigation pour les processus d'informations sur les données

Fast Sneakers est une boutique de mode spécialisée dans la vente de baskets tendance. Le prix d'une paire de chaussures donnée peut augmenter ou diminuer en fonction des stocks et des tendances, comme par exemple le fait qu'une vedette ou une star du sport portant des baskets d'une marque donnée a été vue à la télévision hier soir. Pour Fast Sneakers, il est important de suivre et d'analyser ces tendances afin d'optimiser ses revenus.

Fast Sneakers ne souhaite pas ajouter de frais généraux supplémentaires au projet avec de nouvelles infrastructures à gérer. Elle veut pouvoir répartir le développement entre les parties concernées, afin que les ingénieurs de données se concentrent sur la transformation des données et que les scientifiques des données travaillent sur leur fonctionnalité ML de manière indépendante.

Pour réagir rapidement et ajuster automatiquement les prix en fonction de la demande, Fast Sneakers diffuse des événements importants (comme les données liées à l'intérêt selon le nombre de clics ou les données d'achat) en transformant et en augmentant les données d'événement et en les transmettant à un modèle de ML. Leur modèle de ML est en mesure de déterminer si un ajustement de prix est nécessaire. Cela permet à Fast Sneakers de modifier automatiquement ses prix afin d'optimiser les profits sur ses produits.

Ajustements de prix en temps réel chez Fast Sneakers

Ce diagramme d'architecture montre la solution de streaming en temps réel que Fast Sneakers a créée à l'aide de Kinesis Data Streams, de AWS Glue et de DynamoDB Streams. En tirant parti de ces services, ils disposent d'une solution élastique et fiable qui ne nécessite aucune mise en place ni maintenance de l'infrastructure de support. Ils peuvent consacrer du temps à ce qui apporte de la valeur à leur entreprise en se concentrant sur les tâches d'extraction, de transformation, de chargement (ETL) en streaming et sur leur modèle de Machine Learning.

Pour mieux comprendre l'architecture et les technologies utilisées dans leur charge de travail, voici quelques détails des services utilisés.

AWS Glue et AWS Glue streaming

AWS Glue est un service ETL entièrement géré que vous pouvez utiliser pour cataloguer vos données, les nettoyer, les enrichir et les déplacer de manière fiable entre des magasins de données. Avec AWS Glue, vous pouvez réduire considérablement le coût, la complexité et le temps passé à créer des tâches ETL. AWS Glue est sans serveur ; il ne nécessite donc aucune infrastructure à configurer ou à gérer. Vous payez uniquement pour les ressources consommées pendant l'exécution de vos tâches.

En utilisant AWS Glue, vous pouvez créer une application consommateur avec une tâche ETL AWS Glue en streaming. Cela vous permet d'utiliser Apache Spark et d'autres modules d'écriture basés sur Spark pour consommer et traiter vos données d'événement. La section suivante de ce document traite plus en détail de ce scénario.

AWS Glue Data Catalog

Le AWS Glue Data Catalog contient les références aux données utilisées en tant que sources et cibles de vos tâches ETL dans AWS Glue. Le AWS Glue Data Catalog est un index de la localisation, du schéma et des métriques d'exécution de vos données. Les informations du catalogue de données vous permettent de créer et de surveiller vos tâches ETL. Les informations contenues dans le catalogue de données sont stockées en tant que tables de métadonnées, chaque table spécifiant un magasin de données unique. En configurant un analyseur, vous pouvez évaluer automatiquement de nombreux types de magasin de données, notamment les magasins connectés DynamoDB, S3 et Java Database Connectivity (JDBC), extraire des métadonnées et des schémas, puis créer des définitions de table dans le AWS Glue Data Catalog.

Pour utiliser Amazon Kinesis Data Streams dans le cadre de tâches ETL AWS Glue en streaming, il est recommandé de définir votre flux dans une table au sein d'une base de données AWS Glue Data

Catalog. Vous définissez une table basée sur un flux avec le flux Kinesis, un des nombreux formats pris en charge (CSV, JSON, ORC, Parquet, Avro ou un format client avec Grok). Vous pouvez entrer manuellement un schéma ou ignorer cette étape et laisser votre tâche AWS Glue le déterminer pendant l'exécution du travail.

Tâche ETL en streaming dans AWS Glue

AWS Glue exécute vos tâches ETL dans un environnement sans serveur Apache Spark. AWS Glue exécute ces tâches sur des ressources virtuelles qu'il alloue et gère dans son propre compte de service. En plus de pouvoir exécuter des tâches basées sur Apache Spark, AWS Glue offre un niveau de fonctionnalité supplémentaire en plus de Spark avec les DynamicFrames.

Les DynamicFrames sont des tables distribuées qui prennent en charge les données imbriquées telles que les structures et les tableaux. Chaque enregistrement est auto-descriptif, conçu pour une flexibilité de schéma avec des données semi-structurées. Un enregistrement dans une `DynamicFrame` contient à la fois des données et le schéma décrivant les données. Apache Spark `DataFrames` et `DynamicFrames` sont tous deux pris en charge dans vos scripts ETL et peuvent être convertis dans les deux sens. Les `DynamicFrames` fournissent un ensemble de transformations avancées pour le nettoyage des données et les tâches ETL.

En utilisant Spark Streaming dans votre tâche AWS Glue, vous pouvez créer des tâches ETL en streaming qui s'exécutent en continu et consommer des données provenant de sources de streaming telles qu'Amazon Kinesis Data Streams, Apache Kafka et Amazon MSK. Les tâches peuvent nettoyer, fusionner et transformer les données, puis charger les résultats dans des magasins, notamment des magasins de données Amazon S3, Amazon DynamoDB ou JDBC.

Par défaut, AWS Glue traite et écrit les données dans des fenêtres de 100 secondes. Cela permet de traiter les données efficacement et d'effectuer des agrégations sur les données qui arrivent plus tard que prévu. Vous pouvez configurer la taille de la fenêtre en l'ajustant pour tenir compte de la vitesse de réponse par rapport à la précision de votre agrégation. Les tâches de streaming AWS Glue utilisent des points de contrôle pour suivre les données qui ont été lues à partir du Kinesis Data Stream. Pour obtenir une procédure pas à pas de création d'une tâche ETL en streaming dans AWS Glue, vous pouvez vous référer à Ajout de tâches ETL en streaming dans AWS Glue

Amazon DynamoDB

Amazon DynamoDB est une base de données clé-valeur et de documents offrant des performances de latence de l'ordre de quelques millisecondes, quelle que soit l'ordre de grandeur. Il s'agit d'une

base de données multi-région, multi-active et durable entièrement gérée, avec des systèmes intégrés de sécurité, de sauvegarde, de restauration et de mise en cache en mémoire pour les applications à l'échelle d'Internet. DynamoDB peut traiter plus de dix mille milliards de demandes par jour et supporte des pics de 20 millions de demandes par seconde.

Modifier la capture des données pour DynamoDB

Un flux DynamoDB est un flux ordonné d'informations sur les modifications apportées aux éléments dans une table DynamoDB. Lorsque vous activez un flux sur une table, DynamoDB capture des informations sur chaque modification apportée à des éléments de données dans la table. DynamoDB s'exécute sur AWS Lambda afin que vous puissiez créer des déclencheurs, éléments de code qui répondent automatiquement à des événements dans DynamoDB Streams. Les déclencheurs vous permettent de créer des applications qui réagissent aux modifications de données dans les tables DynamoDB.

Lorsqu'un flux est activé sur une table, vous pouvez associer le flux Amazon Resource Name (ARN) à une fonction Lambda que vous écrivez. Immédiatement après la modification d'un élément dans la table, un nouvel enregistrement apparaît dans le flux de la table. AWS Lambda interroge le flux et appelle votre fonction Lambda quand il détecte de nouveaux enregistrements de flux.

Amazon SageMaker et points de terminaison de service Amazon SageMaker

Amazon SageMaker est une plateforme entièrement gérée qui permet aux développeurs et aux scientifiques des données de créer, d'entraîner et de déployer des modèles de ML rapidement et à n'importe quelle échelle. SageMaker incluent des modules qui peuvent être utilisés conjointement ou indépendamment pour créer, entraîner et déployer vos modèles de ML. Avec les points de terminaison de service Amazon SageMaker, vous pouvez créer un point de terminaison hébergé géré pour une inférence en temps réel avec un modèle déployé que vous avez développé au sein ou en dehors d'Amazon SageMaker.

Avec le kit SDK AWS, vous pouvez invoquer un point de terminaison SageMaker en transmettant des informations sur le type de contenu avec celui-ci, puis recevoir des prédictions en temps réel basées sur les données transmises. Cela vous permet de séparer la conception et le développement de vos modèles de ML du code qui exécute des actions sur les résultats déduits.

Cela permet à vos scientifiques des données de se concentrer sur le ML et aux développeurs qui utilisent le modèle de ML de se concentrer sur la façon dont ils l'utilisent dans leur code. Pour de

plus amples informations sur la façon d'appeler un point de terminaison dans SageMaker, veuillez consulter InvokeEnpoint dans le document Amazon SageMaker API Reference.

Déduction d'informations sur les données en temps réel

Le schéma d'architecture précédent montre que l'application Web existante de Fast Sneakers a ajouté un flux de données Kinesis contenant des événements de flux de clics, qui fournit des données sur le trafic et les événements du site Web. Le catalogue des produits, qui contient des informations telles que la catégorisation, les attributs et le prix de chaque produit, et le tableau des commandes, qui contient des données telles que les articles commandés, la facturation, l'expédition, etc., sont des tables DynamoDB distinctes. La source de flux de données et les tables DynamoDB appropriées ont leurs métadonnées et leurs schémas définis dans le AWS Glue Data Catalog pour être utilisés par la tâche ETL en streaming AWS Glue.

En utilisant Apache Spark, le streaming Spark et `DynamicFrames` dans leur tâche ETL en streaming AWS Glue, Fast Sneakers est capable d'extraire des données d'un des flux de données et de les transformer, en fusionnant les données des tables des produits et des commandes. Avec les données hydratées de la transformation, les jeux de données permettant d'obtenir des résultats d'inférence sont soumis à une table DynamoDB.

Le flux DynamoDB de la table déclenche une fonction Lambda pour chaque nouvel enregistrement écrit. La fonction Lambda envoie les enregistrements précédemment transformés à un point de terminaison SageMaker avec le kit SDK AWS de manière à déduire, le cas échéant, les ajustements de prix nécessaires pour un produit. Si le modèle de ML identifie qu'un ajustement de prix est requis, la fonction Lambda écrit la modification de prix sur le produit dans la table DynamoDB du catalogue.

Récapitulatif

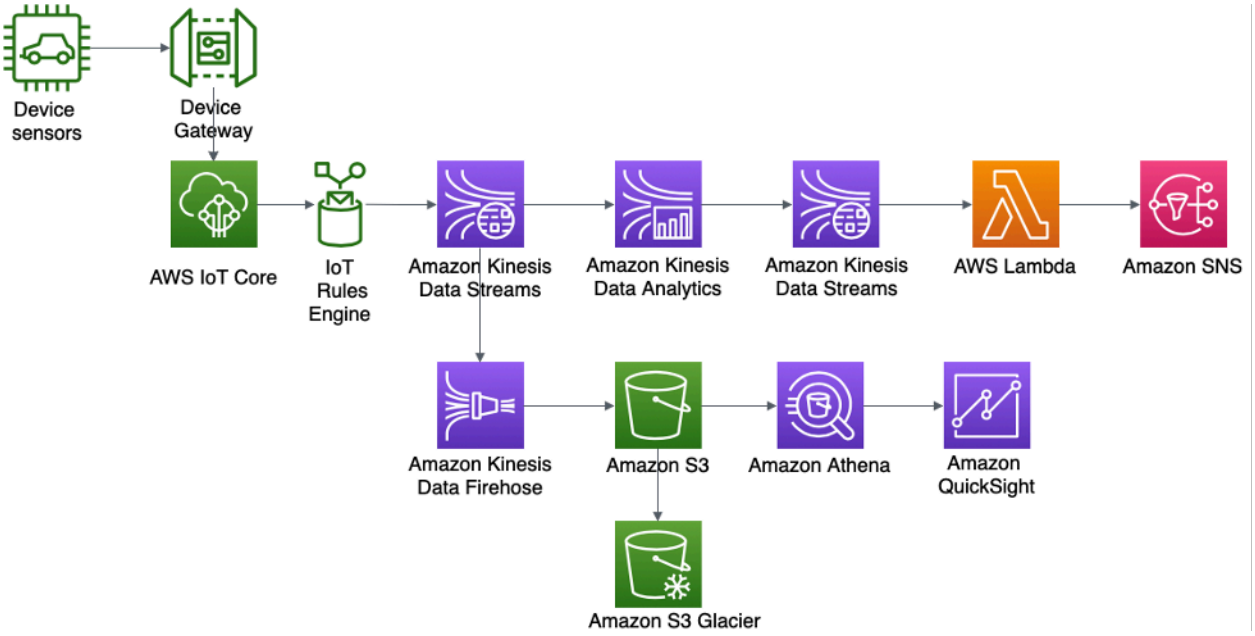
Amazon Kinesis Data Streams facilite la collecte, le traitement et l'analyse de données de streaming en temps réel, afin d'obtenir rapidement des informations stratégiques et de réagir rapidement. En combinaison avec le service d'intégration de données AWS Glue sans serveur, vous pouvez créer des applications de streaming d'événements en temps réel qui préparent et combinent des données pour le ML.

Étant donné que Kinesis Data Streams et les services AWS Glue sont entièrement gérés, AWS supprime la lourde charge de travail indifférenciée liée à la gestion de l'infrastructure de votre plateforme big data. Cela vous permet de vous concentrer sur la génération d'informations sur les données basées sur vos données.

Fast Sneakers peut utiliser le traitement des événements en temps réel et le ML pour permettre à son site Web d'effectuer des ajustements de prix en temps réel entièrement automatisés et ainsi, optimiser leur stock de produits. Cela apporte une valeur importante à leur entreprise tout en évitant d'avoir à créer et à maintenir une plateforme de big data.

Scénario 4 : détection et notification en temps réel des anomalies de capteurs

La société ABC4Logistics transporte des produits pétroliers hautement inflammables tels que de l'essence, du propane liquide (GPL) et du naphte depuis le port vers différentes villes. Des centaines de véhicules sont équipés de plusieurs capteurs pour surveiller des éléments tels que la localisation, la température du moteur, la température à l'intérieur du conteneur, la vitesse de conduite, la localisation de stationnement, l'état des routes, etc. Une des exigences d'ABC4Logistics consiste à surveiller les températures du moteur et du conteneur en temps réel et à alerter le conducteur et l'équipe de surveillance de la flotte en cas d'anomalie. Pour détecter de telles conditions et générer des alertes en temps réel, ABC4Logistics a mis en œuvre l'architecture suivante sur AWS.



Architecture de détection des anomalies de capteur et de notifications en temps réel pour ABC4Logistics

Les données des capteurs sont ingérées par AWS IoT Gateway, où le moteur de règles AWS IoT rend les données en streaming disponibles dans Amazon Kinesis Data Streams. Avec Kinesis Data Analytics, ABC4Logistics peut effectuer des analyses en temps réel sur les données en streaming dans Kinesis Data Streams.

Avec Kinesis Data Analytics, ABC4Logistics peut détecter si les relevés de température des capteurs s'écartent des relevés normaux sur une période de dix secondes, et ingérer l'enregistrement sur une autre instance Kinesis Data Streams, en identifiant les enregistrements anormaux. Amazon Kinesis Data Streams appelle ensuite des fonctions Lambda, qui peuvent envoyer les alertes au conducteur et à l'équipe de surveillance de la flotte via Amazon SNS.

Les données de Kinesis Data Streams sont également transférées vers Amazon Kinesis Data Firehose. Amazon Kinesis Data Firehose conserve ces données dans Amazon S3, ce qui permet à ABC4Logistics d'effectuer des analyses par lots ou en temps quasi réel sur les données des capteurs. ABC4Logistics utilise Amazon Athena pour interroger les données dans S3 et Amazon QuickSight pour les visualisations. Pour la conservation des données à long terme, la stratégie de cycle de vie S3 est utilisée pour archiver les données dans Amazon S3 Glacier.

Les composants importants de cette architecture sont décrits en détail ci-après.

Amazon Kinesis Data Analytics

Amazon Kinesis Data Analytics vous permet de transformer et d'analyser les données de streaming et de répondre aux anomalies en temps réel. Il s'agit d'un service sans serveur sur AWS, ce qui signifie que Kinesis Data Analytics se charge de l'approvisionnement et met à l'échelle l'infrastructure de manière élastique pour gérer tout débit de données. Cela vous débarrasse de la lourde charge de travail indifférenciée de mise en place et de gestion de l'infrastructure de streaming, vous permettant ainsi de consacrer plus de temps à l'écriture d'applications de streaming.

Avec Amazon Kinesis Data Analytics, vous pouvez interroger de manière interactive des données de streaming à l'aide de plusieurs options, notamment le langage SQL standard, des applications Apache Flink en Java, Python et Scala, et créer des applications Apache Beam en Java pour analyser les flux de données.

Ces options vous offrent la flexibilité d'utiliser une approche spécifique en fonction du niveau de complexité de l'application de streaming et de la prise en charge source/cible. La section suivante traite de l'option Kinesis Data Analytics pour les applications Flink.

Amazon Kinesis Data Analytics pour applications Apache Flink

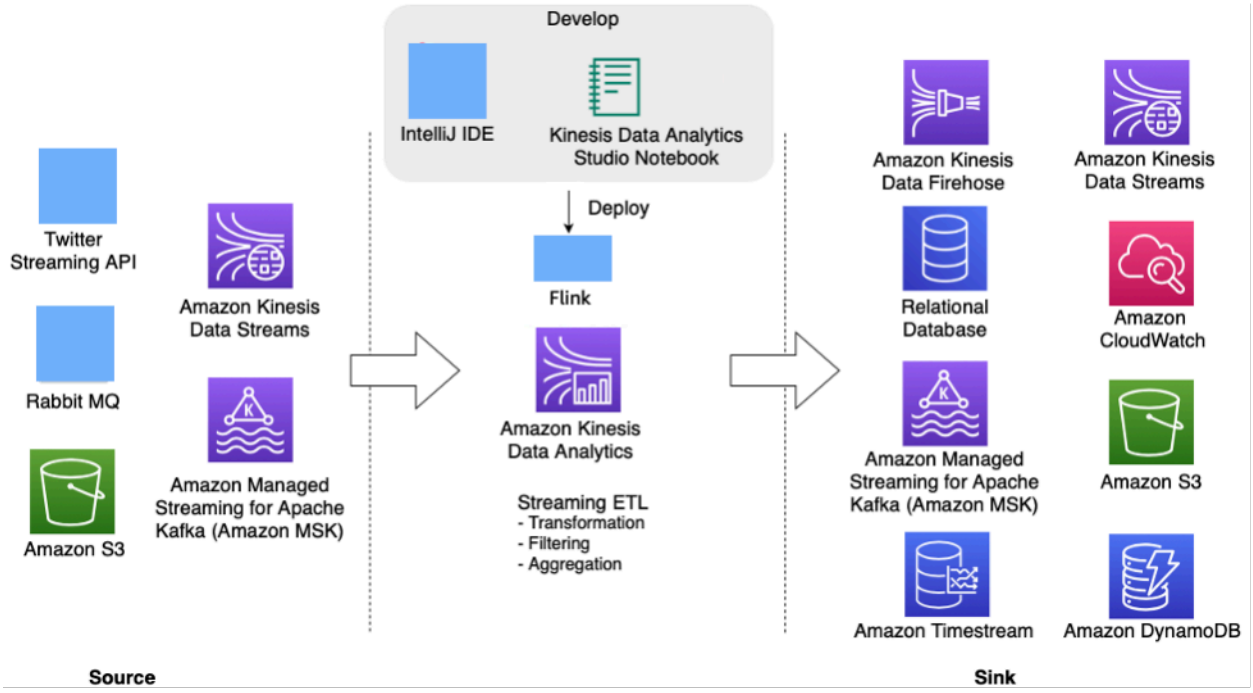
Apache Flink est un cadre open source populaire et un moteur de traitement distribué pour les calculs avec état sur des flux de données avec et sans limite. Apache Flink est conçu pour effectuer des calculs à la vitesse de la mémoire interne et à l'échelle avec le support de la sémantique

« exactement une fois ». Les applications basées sur Apache Flink permettent d'obtenir une faible latence et un débit élevé tout en étant tolérant aux pannes.

Avec Amazon Kinesis Data Analytics pour Apache Flink, vous pouvez créer et exécuter du code contre des sources de streaming pour effectuer des analyses de séries chronologiques, alimenter des tableaux de bord en temps réel et créer des métriques en temps réel sans avoir à gérer l'environnement distribué complexe d'Apache Flink. Vous pouvez utiliser les fonctions de programmation de haut niveau Flink de la même manière que vous les utilisez lorsque vous hébergez vous-même l'infrastructure Flink.

Kinesis Data Analytics pour Apache Flink vous permet de créer des applications en Java, Scala, Python ou SQL afin de traiter et d'analyser des données de streaming. Une application Flink classique lit les données à partir du flux d'entrée, de l'emplacement ou de la source des données, transforme, filtre ou joint les données à l'aide d'opérateurs ou de fonctions, et stocke les données sur un flux de sortie ou un emplacement de données, ou récepteur.

Le diagramme d'architecture suivant montre certaines des sources et des récepteurs pris en charge pour l'application Flink sur Kinesis Data Analytics. Outre les connecteurs prégroupés pour source/récepteur, vous pouvez également intégrer des connecteurs personnalisés à une variété d'autres sources/récepteurs pour les applications Flink sur Kinesis Data Analytics.



Application Apache Flink sur Kinesis Data Analytics pour le traitement des flux en temps réel

Les développeurs peuvent utiliser leur IDE préféré pour développer des applications Flink et les déployer sur Kinesis Data Analytics à partir d'AWS Management Console ou d'outils DevOps.

Amazon Kinesis Data Analytics Studio

Dans le cadre du service Kinesis Data Analytics, Kinesis Data Analytics Studio permet aux clients d'interroger de manière interactive des flux de données en temps réel, mais aussi de créer et d'exécuter facilement des applications de traitement de flux à l'aide de SQL, Python et Scala. Les notebooks Studio sont optimisés par Apache Zeppelin.

Le notebook Studio vous permet de développer votre code d'application Flink dans un environnement de notebook, d'afficher les résultats de votre code en temps réel et de le visualiser. Vous pouvez créer un notebook Studio optimisé par Apache Zeppelin et Apache Flink en un seul clic depuis Kinesis Data Streams et la console Amazon MSK, ou le lancer à partir de Kinesis Data Analytics Console.

Après avoir développé le code de manière itérative dans le cadre de Kinesis Data Analytics Studio, vous pouvez déployer un notebook en tant qu'application d'analyse de données Kinesis pour l'exécuter en mode streaming, lire les données de vos sources, écrire sur vos destinations, gérer l'état de l'application à long terme et dimensionner automatiquement en fonction du débit de vos flux sources. Auparavant, les clients utilisaient Kinesis Data Analytics pour applications SQL pour de telles analyses interactives de données de streaming en temps réel sur AWS.

Kinesis Data Analytics pour applications SQL est toujours disponible, mais pour les nouveaux projets, AWS recommande d'utiliser le nouveau Kinesis Data Analytics Studio. Kinesis Data Analytics Studio combine la facilité d'utilisation avec des capacités analytiques avancées, ce qui vous permet de concevoir des applications sophistiquées de traitement de flux en quelques minutes.

Pour rendre l'application Kinesis Data Analytics Flink tolérante aux pannes, vous pouvez utiliser des points de contrôle et des instantanés, comme décrit dans Implémentation de la tolérance aux pannes dans Kinesis Data Analytics pour Apache Flink.

Les applications Flink Kinesis Data Analytics sont utiles pour écrire des applications d'analyse de streaming complexes, telles que des applications avec une sémantique « exactement une fois » de traitement des données, des capacités de point de contrôle et le traitement de données à partir de sources de données telles que Kinesis Data Streams, Kinesis Data Firehose, Amazon MSK, Rabbit MQ et Apache Cassandra, y compris les connecteurs personnalisés.

Après avoir traité les données de streaming dans l'application Flink, vous pouvez conserver les données dans divers récepteurs ou destinations comme Amazon Kinesis Data Streams, Amazon

Kinesis Data Firehose, Amazon DynamoDB, Amazon OpenSearch Service, Amazon Timestream, Amazon S3, etc. L'application Flink Kinesis Data Analytics fournit également des garanties de performances inférieures à la seconde.

Applications Apache Beam pour Kinesis Data Analytics

Apache Beam est un modèle de programmation pour le traitement des données de streaming. Apache Beam fournit une couche d'API portable pour créer des pipelines sophistiqués de traitement parallèle des données qui peuvent être exécutés sur divers moteurs ou exécuteurs tels que Flink, Spark Streaming, Apache Samza, etc.

Vous pouvez utiliser le cadre Apache Beam avec votre application d'analyse de données Kinesis pour traiter les données de streaming. Les applications d'analyse de données Kinesis qui utilisent Apache Beam font appel à l'exécuteur Apache Flink pour exécuter des pipelines Beam.

Récapitulatif

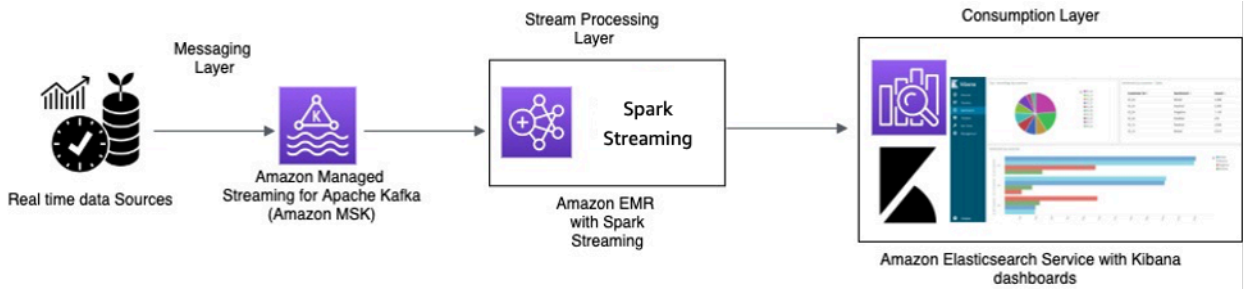
En utilisant les services de streaming AWS Amazon Kinesis Data Streams, Amazon Kinesis Data Analytics et Amazon Kinesis Data Firehose,

ABC4Logistics peut détecter des schémas anormaux dans les relevés de température et informer le conducteur et l'équipe de gestion de la flotte en temps réel, évitant ainsi des accidents majeurs tels qu'une panne complète du véhicule ou un incendie.

Scénario 5 : surveillance des données de télémétrie en temps réel avec Apache Kafka

ABC1cabs est une société de services de réservation de taxi en ligne. Tous les taxis sont équipés d'appareils IoT qui recueillent les données de télémétrie des véhicules. Actuellement, ABC1Cabs exécute des clusters Apache Kafka conçus pour la consommation d'événements en temps réel, la collecte de métriques sur l'état du système, le suivi de l'activité et l'alimentation en données de la plateforme Apache Spark Streaming basée sur un cluster Hadoop sur site.

ABC1Cabs utilise OpenSearch Dashboards pour les métriques métier, le débogage, les alertes et la création d'autres tableaux de bord. Cette société est intéressée par Amazon MSK, par Amazon EMR avec Spark Streaming et par OpenSearch Service avec OpenSearch Dashboards. Leur exigence consiste à réduire les frais administratifs liés à la maintenance des clusters Apache Kafka et Hadoop, tout en utilisant des API et des logiciels open source familiers pour orchestrer leur pipeline de données. Le diagramme d'architecture suivant présente leur solution sur AWS.



Traitement en temps réel avec Amazon MSK et traitement Stream à l'aide d'Apache Spark Streaming sur Amazon EMR et Amazon OpenSearch Service avec OpenSearch Dashboards

Les appareils IoT des taxis collectent des données de télémétrie et les envoient à un hub source. Le hub source est configuré pour envoyer des données en temps réel à Amazon MSK. À l'aide des API de la bibliothèque du producteur Apache Kafka, Amazon MSK est configuré pour diffuser les données dans un cluster Amazon EMR. Le cluster Amazon EMR possède un client Kafka et Spark Streaming installé pour pouvoir consommer et traiter les flux de données.

Spark Streaming possède des connecteurs de récepteur qui peuvent écrire des données directement dans des index définis d'Elasticsearch. Les clusters Elasticsearch OpenSearch Dashboards peuvent être utilisés pour les métriques et les tableaux de bord. Amazon MSK, Amazon EMR avec Spark Streaming et OpenSearch Service avec OpenSearch Dashboards sont tous des services gérés dans lesquels AWS s'occupe de la lourde charge de travail indifférenciée de la gestion de l'infrastructure des différents clusters, ce qui vous permet de créer votre application à l'aide de logiciels open source familiers en quelques clics. La section suivante présente ces services plus en détail.

Amazon Managed Streaming pour Apache Kafka (Amazon MSK)

Apache Kafka est une plateforme open source qui permet aux clients de capturer des données de streaming telles que des événements de flux de clics, des transactions, des événements IoT et des journaux d'application et de machine. Avec ces informations, vous pouvez développer des applications qui effectuent des analyses en temps réel, exécutent des transformations en continu et distribuent ces données aux lacs de données et aux bases de données en temps réel.

Vous pouvez utiliser Kafka en tant que magasin de données de streaming pour découpler les applications du producteur et des consommateurs et permettre un transfert de données fiable entre les deux composants. Bien que Kafka soit une plateforme de flux de données et de messagerie d'entreprise populaire, elle peut être difficile à configurer, à mettre à l'échelle et à gérer en production.

Amazon MSK se charge de ces tâches de gestion et facilite la configuration et l'exécution de Kafka, avec Apache Zookeeper, dans un environnement respectant les bonnes pratiques en matière de

haute disponibilité et de sécurité. Vous pouvez toujours utiliser les opérations de plan de contrôle et les opérations de plan de données de Kafka pour gérer la production et la consommation de données.

Étant donné qu'Amazon MSK exécute et gère Apache Kafka open source, les clients peuvent facilement migrer et exécuter des applications Apache Kafka existantes sur AWS sans avoir à modifier leur code d'application.

Mise à l'échelle

Amazon MSK propose des opérations de mise à l'échelle afin que l'utilisateur puisse dimensionner activement le cluster pendant son exécution. Lors de la création d'un cluster Amazon MSK, vous pouvez spécifier le type d'instance des agents lors du lancement du cluster. Vous pouvez commencer avec quelques agents au sein d'un cluster Amazon MSK. Ensuite, à l'aide d'AWS Management Console ou de l'AWS CLI, vous pouvez augmenter le nombre d'agents (jusqu'à plusieurs centaines) par cluster.

Vous pouvez également mettre à l'échelle vos clusters en modifiant la taille ou la famille de vos agents Apache Kafka. La modification de la taille ou de la famille de vos agents vous donne la possibilité d'ajuster la capacité de calcul de votre cluster Amazon MSK en fonction de l'évolution de vos charges de travail. Utilisez la feuille de calcul Amazon MSK Sizing and Pricing (téléchargement du fichier) pour déterminer le nombre correct d'agents pour votre cluster Amazon MSK. Cette feuille de calcul fournit une estimation du dimensionnement d'un cluster Amazon MSK et des coûts associés liés à Amazon MSK par rapport à un cluster Apache Kafka basé sur EC2, auto-géré et similaire.

Après avoir créé le cluster Amazon MSK, vous pouvez augmenter la quantité de stockage EBS par agent, à l'exception de la diminution du stockage. Les volumes de stockage restent disponibles pendant cette opération de mise à l'échelle. Deux types d'opération de dimensionnement sont proposés : mise à l'échelle automatique et mise à l'échelle manuelle.

Amazon MSK prend en charge l'extension automatique du stockage de votre cluster en réponse à une augmentation de l'utilisation à l'aide des stratégies de mise à l'échelle automatique de l'application. Votre stratégie de mise à l'échelle automatique définit l'utilisation du disque cible et la capacité de mise à l'échelle maximale.

Le seuil d'utilisation du stockage permet à Amazon MSK de déclencher une opération de mise à l'échelle automatique. Pour augmenter le stockage à l'aide de la mise à l'échelle manuelle, attendez que le cluster soit à l'état `ACTIVE`. La mise à l'échelle du stockage requiert un temps de stabilisation d'au moins six heures entre les événements. Même si l'opération met immédiatement à disposition du

stockage supplémentaire, le service effectue des optimisations sur votre cluster qui peuvent prendre jusqu'à 24 heures ou plus.

La durée de ces optimisations est proportionnelle à la taille de votre stockage. En outre, la réplication de plusieurs zones de disponibilité au sein d'une région AWS vous est proposée de manière à fournir une haute disponibilité.

Configuration

Amazon MSK fournit une configuration par défaut pour les agents, les rubriques et les nœuds Apache ZooKeeper. Vous pouvez également créer des configurations personnalisées et utiliser celles-ci pour créer de nouveaux clusters Amazon MSK ou pour mettre à jour des clusters existants. Lorsque vous créez un cluster MSK sans spécifier de configuration Amazon MSK personnalisée, Amazon MSK crée et utilise une configuration par défaut. Pour obtenir la liste des valeurs par défaut, consultez cette configuration Apache Kafka.

À des fins de surveillance, Amazon MSK collecte les métriques Apache Kafka et les envoie à Amazon CloudWatch, où vous pouvez les consulter. Les métriques que vous configurez pour votre cluster MSK sont automatiquement collectées et envoyées à CloudWatch. La surveillance du décalage des consommateurs vous permet d'identifier les consommateurs lents ou bloqués qui ne suivent pas les dernières données disponibles dans une rubrique. Si nécessaire, vous pouvez ensuite prendre des mesures correctives, telles que la mise à l'échelle ou le redémarrage de ces consommateurs.

Migration vers Amazon MSK

La migration depuis un site vers Amazon MSK peut être réalisée par l'intermédiaire d'une des méthodes suivantes.

- MirrorMaker2.0 : MirrorMaker2.0 (MM2) MM2 est un moteur de réplication de données multi-cluster basé sur le cadre Apache Kafka Connect. MM2 est la combinaison d'un connecteur source Apache Kafka et d'un connecteur récepteur. Vous pouvez utiliser un cluster MM2 unique pour migrer des données entre plusieurs clusters. MM2 détecte automatiquement les nouvelles rubriques et partitions, tout en veillant à ce que les configurations de rubrique soient synchronisées entre les clusters. MM2 prend en charge les listes de contrôle d'accès de migration, les configurations de rubrique et la traduction de décalage. Pour de plus amples détails sur la migration, veuillez consulter Migration de clusters à l'aide de MirrorMaker d'Apache Kafka. MM2 est utilisé pour les cas d'utilisation liés à la réplication des configurations de rubrique et à la traduction de décalage.

- Apache Flink : MM2 prend en charge la sémantique « au moins une fois ». Les enregistrements peuvent être dupliqués vers la destination et les consommateurs sont censés être idempotents pour traiter les enregistrements dupliqués. Dans les scénarios de type « exactement une fois », la sémantique est requise, les clients peuvent utiliser Apache Flink. Il fournit une alternative pour obtenir une sémantique de type « exactement une fois ».

  Apache Flink peut également être utilisé pour les scénarios où les données nécessitent des actions de mappage ou de transformation avant d'être soumises au cluster de destination. Apache Flink fournit des connecteurs pour Apache Kafka avec des sources et des récepteurs capables de lire les données d'un cluster Apache Kafka et d'écrire sur un autre. Apache Flink peut être exécuté sur AWS en lançant un cluster Amazon EMR ou en exécutant Apache Flink en tant qu'application à l'aide d'Amazon Kinesis Data Analytics.
- AWS Lambda : grâce au support d'Apache Kafka en tant que source d'événements pour AWS Lambda, les clients peuvent désormais consommer les messages d'une rubrique via une fonction Lambda. Le service AWS Lambda interroge en interne les nouveaux enregistrements ou messages provenant de la source de l'événement, puis appelle de manière synchrone la fonction Lambda cible pour consommer ces messages. Lambda lit les messages par lots et fournit les lots de messages à votre fonction dans la charge utile de l'événement pour traitement. Les messages consommés peuvent ensuite être transformés et/ou écrits directement dans votre cluster Amazon MSK de destination.

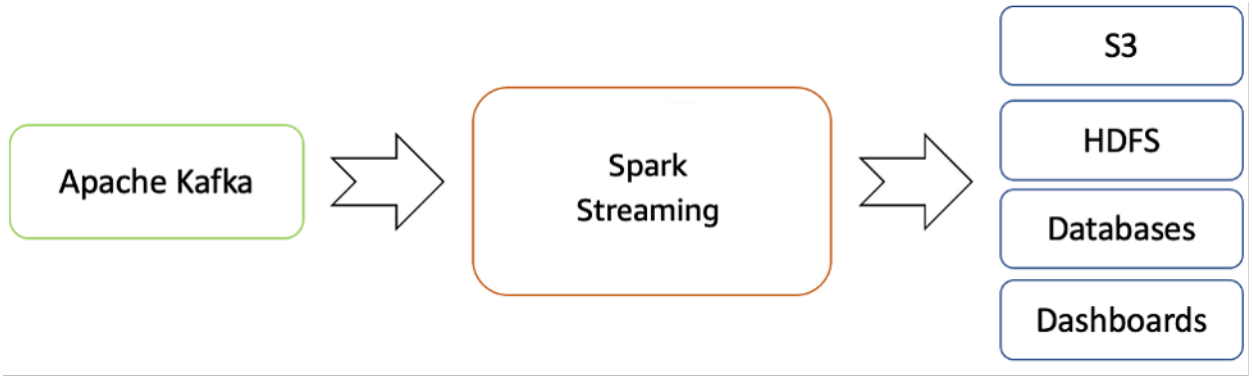
Amazon EMR avec Spark Streaming

Amazon EMR est une plateforme de cluster gérée qui simplifie l'exécution des infrastructures de données massives, telles qu'Apache Hadoop et Apache Spark sur AWS, pour traiter et analyser de grandes quantités de données.

Amazon EMR fournit les capacités de Spark et peut être utilisé pour démarrer Spark Streaming afin de consommer les données de Kafka. Spark Streaming est une extension de l'API Spark principale qui permet le traitement évolutif, à haut débit et tolérant aux pannes de flux de données en direct.

Vous pouvez créer un cluster Amazon EMR à l'aide d'AWS Command Line Interface (AWS CLI) ou sur AWS Management Console et sélectionner Spark et Zeppelin dans les configurations avancées lors de la création du cluster. Comme le montre le diagramme d'architecture suivant, les données peuvent être ingérées à partir de nombreuses sources telles qu'Apache Kafka et Kinesis Data Streams, et peuvent être traitées à l'aide d'algorithmes complexes exprimés avec des fonctions de haut niveau telles que map, reduce, join et window. Pour de plus amples informations, veuillez consulter Transformations on DStreams (Transformations sur DStreams).

Les données traitées peuvent être transférées vers des systèmes de fichiers, des bases de données et des tableaux de bord en direct.



Flux de streaming en temps réel d'Apache Kafka vers l'écosystème Hadoop

Par défaut, Apache Spark Streaming possède un modèle d'exécution par micro-lots. Cependant, depuis la sortie de Spark 2.3, Apache a introduit un nouveau mode de traitement à faible latence appelé traitement continu, qui peut atteindre des latences de bout en bout d'une milliseconde avec des garanties de type « au moins une fois ».

Sans modifier les opérations Dataset/DataFrames dans vos requêtes, vous pouvez choisir le mode en fonction des exigences de votre application. Voici quelques avantages de Spark Streaming :

- Il apporte l'API intégrée au langage d'Apache Spark pour le traitement des flux, ce qui vous permet d'écrire des tâches de streaming de la même manière que vous écrivez des tâches par lots.
- Il prend en charge Java, Scala et Python.
- Il peut récupérer à la fois le travail perdu et l'état de l'opérateur (comme les fenêtres coulissantes) dès le départ, sans aucun code supplémentaire de votre part.
- En s'exécutant sur Spark, Spark Streaming vous permet de réutiliser le même code pour le traitement par lots, de joindre des flux par rapport à des données historiques ou d'exécuter des requêtes ad hoc sur l'état du flux et de créer de puissantes applications interactives, pas seulement des analyses.
- Une fois le flux de données traité avec Spark Streaming, OpenSearch Sink Connector peut être utilisé pour écrire des données dans le cluster OpenSearch Service et, à son tour, OpenSearch Service avec OpenSearch Dashboards peut être utilisé en tant que couche de consommation.

Amazon OpenSearch Service avec OpenSearch Dashboards

OpenSearch Service est un service géré qui facilite le déploiement, l'utilisation et la mise à l'échelle des clusters OpenSearch dans le cloud AWS. OpenSearch est un moteur de recherche et d'analyse à code source libre très populaire, utilisé notamment pour l'analyse des fichiers journaux, la surveillance d'applications en temps réel et l'analyse des parcours de navigation.

OpenSearch Dashboards est un outil à code source libre de visualisation et d'exploration des données, utilisé pour l'analytique des journaux et des séries chronologiques, la surveillance des applications et l'intelligence opérationnelle. Il offre des fonctionnalités puissantes et faciles à utiliser telles que des histogrammes, des graphiques linéaires, des camemberts, des cartes thermiques et un support géospatial intégré.

OpenSearch Dashboards propose une intégration étroite avec OpenSearch, moteur d'analyse et de recherche populaire. Cela fait d'OpenSearch Dashboards le choix par défaut pour la visualisation des données stockées dans OpenSearch. OpenSearch Service fournit une installation d'OpenSearch Dashboards pour chaque domaine OpenSearch Service. Vous trouverez un lien vers OpenSearch Dashboards sur le tableau de bord de votre domaine sur la console OpenSearch Service.

Récapitulatif

Avec Apache Kafka proposé en tant que service géré sur AWS, vous pouvez vous concentrer sur la consommation plutôt que sur la gestion de la coordination entre les agents, ce qui nécessite généralement une compréhension détaillée d'Apache Kafka. Des fonctions telles que la haute disponibilité, la capacité de mise à l'échelle des agents et le contrôle d'accès détaillé sont gérées par la plateforme Amazon MSK.

ABC1Cabs a utilisé ces services pour créer des applications de production sans avoir besoin d'une expertise en gestion d'infrastructure. Ils ont pu se concentrer sur la couche de traitement pour consommer les données d'Amazon MSK et les propager davantage vers la couche de visualisation.

Spark Streaming sur Amazon EMR peut faciliter l'analyse en temps réel des données de streaming et la publication sur OpenSearch Dashboards sur Amazon OpenSearch Service pour la couche de visualisation.

Conclusion et contributeurs

Conclusion

Ce document passe en revue plusieurs scénarios de flux de streaming. Dans ces scénarios, le traitement des données de streaming a permis aux entreprises présentées en exemple d'ajouter de nouvelles fonctionnalités.

En analysant les données au fur et à mesure de leur création, vous obtiendrez des informations sur ce que fait actuellement votre entreprise. Les services de streaming AWS vous permettent de vous concentrer sur votre application pour prendre des décisions commerciales urgentes, plutôt que de déployer et de gérer l'infrastructure

Participants

- Amalia Rabinovitch, architecte de solutions senior, AWS
- Priyanka Chaudhary, Data Lake, architecte de données, AWS
- Zohair Nasimi, architecte de solutions, AWS
- Rob Kuhr, architecte de solutions, AWS
- Ejaz Sayyed, architecte de solutions de partenaires senior, AWS
- Allan MacInnis, architecte de solutions, AWS
- Chander Matrubhutam, responsable marketing produit, AWS

Révisions du document

Pour être informé des mises à jour de ce livre blanc, abonnez-vous au flux RSS.

update-history-change	update-history-description	update-history-date
Mise à jour	Mise à jour pour exactitude technique	1er septembre 2021
Publication initiale	Première publication du livre blanc	1er juillet 2017