

Diffusion des solutions de données sur AWS avec Amazon Kinesis

Juillet 2017



Mentions légales

Ce document est fourni à titre informatif uniquement. Il présente l'offre de produits et les pratiques actuelles d'AWS à la date de publication de ce document, des informations qui sont susceptibles d'être modifiées sans préavis. Il incombe aux clients de procéder à leur propre évaluation indépendante des informations contenues dans ce document et chaque client est responsable de son utilisation des produits ou services AWS, chacun étant fourni « en l'état », sans garantie d'aucune sorte, qu'elle soit explicite ou implicite. Ce document n'offre pas de garantie, représentation, engagement contractuel, condition ou assurance de la part d'AWS, de ses sociétés apparentées, fournisseurs ou concédants de licence. Les responsabilités et obligations d'AWS vis-à-vis de ses clients sont régies par les contrats AWS. Le présent document ne fait partie d'aucun contrat et ne modifie aucun contrat entre AWS et ses clients.

Table des matières

Introduction	1
Scénarios d'application en temps réel	1
Différence entre le traitement par lots et le traitement de flux	2
Défis de traitement des flux	3
De lot à temps réel : Un exemple	3
Exemple de scénario : Facturation de la route à péage et notification	4
Condition 1 : Données plus récentes dans l'entrepôt de données	5
Amazon Kinesis Firehose	6
Condition 2 : Alertes de seuil de facturation	14
Amazon Kinesis Analytics	15
Amazon Kinesis Streams	18
Condition 3 : Autres seuils d'alertes	24
Architecture complète	26
Conclusion	27
Participants	27

Résumé

Les ingénieurs de données, les analystes de données et les développeurs Big Data cherchent à faire évoluer leurs analyses à partir de lot en temps réel afin que leurs entreprises puissent découvrir ce que leurs clients, les applications et les produits font actuellement et qu'ils puissent réagir rapidement. Ce livre blanc décrit l'évolution de l'analyse de lot en temps réel. Il décrit comment les services tels qu'Amazon Kinesis Streams, Amazon Kinesis Firehose et Amazon Kinesis Analytics peuvent être utilisés pour implémenter des applications en temps réel et fournit des modèles de conception courants à l'aide de ces services.

Introduction

Les entreprises aujourd'hui reçoivent des données à une échelle massive en raison de la vitesse de la croissance fulgurante des sources de données qui génèrent continuellement des flux de données. Qu'il s'agisse de données de journaux depuis les serveurs d'application, de données de flux de clics depuis les sites Web et applications mobiles ou de données de télémétrie de l'Internet des objets (IoT), tous les appareils contiennent des informations qui peuvent vous aider à en savoir plus sur ce que vos clients, les applications et les produits font en temps réel. Avoir la possibilité de traiter et d'analyser ces données en temps réel est essentiel pour effectuer des tâches telles que surveiller en permanence vos applications afin de garantir une haute disponibilité du service et personnaliser les offres promotionnelles et les recommandations de produit. Le traitement en temps réel peut également rendre d'autres cas d'utilisation courants, tels que les analyses de site Web et l'apprentissage-machine, plus précis et exploitables en rendant les données accessibles à ces applications en quelques secondes ou minutes au lieu de plusieurs heures ou jours.

Scénarios d'application en temps réel

Il existe deux types de scénarios de cas d'utilisation pour les applications de transmission de données en continu :

- **Evoluer de l'analyse par lots à l'analyse en diffusion**

Vous pouvez effectuer des analyses en temps réel sur les données qui ont été traditionnellement analysées à l'aide de traitement par lots dans des entrepôts de données ou à l'aide des infrastructures Hadoop. Les cas d'utilisation les plus fréquents dans cette catégorie incluent les lacs de données, la science des données et l'apprentissage-machine. Vous pouvez utiliser des solutions de flux de données en continu pour charger continuellement des données en temps réel dans vos lacs de données. Vous pouvez également mettre à jour des modèles d'apprentissage-machine plus fréquemment au fur et à mesure que de nouvelles données deviennent disponibles, en garantissant la précision et la fiabilité des sorties. Par exemple, Zillow utilise Amazon Kinesis Streams pour collecter les données d'enregistrement public et des listes MLS, puis fournit aux acheteurs et vendeurs de domiciles les estimations les plus à jour de la valeur du domicile en temps quasi réel. Zillow envoie également les mêmes données à son lac de données Amazon Simple Storage Service (S3) à l'aide de Kinesis Streams de

sorte que toutes les applications fonctionnent avec les informations les plus récentes.

- **Construction des applications en temps réel**

Vous pouvez utiliser des services de transmission de données en continu pour des applications en temps réel telles que la surveillance des applications, la détection des fraudes, et les classements en direct. Ces cas d'utilisation nécessitent des latences de bout en bout de l'ordre de la milliseconde, de l'ingestion au traitement, en passant par l'émission des résultats pour cibler les magasins de données et d'autres systèmes. Par exemple, Netflix utilise Kinesis Streams pour surveiller les communications entre toutes ses applications ce qui lui permet de détecter et résoudre rapidement les problèmes, garantissant une haute disponibilité et durée de fonctionnement de ses services à ses clients. Bien que le cas d'utilisation le plus courant soit la surveillance des performances des applications, il y a un nombre croissant d'applications en temps réel dans le domaine des technologies publicitaires, des jeux, et de l'IoT qui relèvent de cette catégorie.

Différence entre le traitement par lots et le traitement de flux

Vous avez besoin d'un ensemble d'outils différents pour collecter, préparer et traiter en temps réel des données diffusées en continu par rapport aux outils que vous avez traditionnellement utilisés pour l'analyse par lots. Avec les outils d'analyse traditionnels, vous collectez les données, vous les chargez périodiquement dans une base de données et vous les analysez des heures, des jours ou des semaines plus tard. L'analyse de données en temps réel requiert une approche différente. Au lieu d'exécuter des requêtes de base de données sur des données stockées, les applications de traitement de flux traitent les données de manière continue en temps réel, même avant qu'elles ne soient stockées. Les données diffusées en continu peuvent arriver à un rythme effréné et les volumes de données peuvent varier à la hausse et à la baisse à tout moment. Les plateformes de traitement de données en diffusion doivent être en mesure de gérer la vitesse et la variabilité des données entrantes et les traiter à mesure qu'elles arrivent, souvent des millions à des centaines de millions d'événements par heure.

Défis de traitement des flux

Le traitement des données en temps réel à mesure qu'elles arrivent peut vous permettre de prendre des décisions beaucoup plus rapidement qu'avec les technologies d'analyse de données traditionnelles. Toutefois, la création et la gestion de vos propres pipelines de données en continu personnalisés sont complexes et gourmandes en ressources. Vous devez construire un système qui puisse, à moindre coût, collecter, préparer et transmettre des données entrantes simultanément à partir des milliers de sources de données. Vous devez ajuster les ressources de stockage et de calcul afin que les données soient regroupées et transmises de manière efficace pour atteindre un débit maximal et une faible latence. Vous devez déployer et gérer une flotte de serveurs pour mettre à l'échelle le système afin de pouvoir gérer les différentes vitesses de données que vous allez y lancer. Une fois que vous avez construit cette plateforme, vous devez surveiller le système et récupérer les défaillances du serveur ou du réseau en rattrapant le traitement des données à partir du point approprié du flux, sans créer de doublons. Tout cela prend du temps et de l'argent et, en fin de compte, la plupart des entreprises n'y parviennent jamais et doivent se contenter du statu quo et exploiter leur entreprise avec des informations qui datent de quelques heures ou de quelques jours.

De lot à temps réel : Un exemple

Pour mieux comprendre comment les entreprises passent d'un traitement de lot à un traitement de flux avec AWS, nous allons passer en revue un exemple. Dans cet exemple, nous allons examiner un scénario et discuter en détail de la façon dont les services AWS [Amazon Kinesis Streams](#),¹ [Amazon Kinesis Firehose](#)² et [Amazon Kinesis Analytics](#)³ sont utilisés pour résoudre le problème.

Le traitement par lots est une pratique courante pour le traitement des données. Les organisations exécutent souvent des tâches régulières pour analyser leurs données à une fréquence applicable pour leurs cas d'utilisation. Par exemple, une organisation peut exécuter un processus à la fin du mois pour déterminer combien facturer à chacun de ses clients. Ou, elle peut exécuter une tâche horaire pour analyser les journaux de ses applications informatiques pour déterminer quelles erreurs se sont produites au cours de la dernière heure. Bien que ces processus mensuels ou par heures soient utiles, que se passerait-il si les mêmes données pouvaient être analysées au fur et à mesure de leur création ?

Y a-t-il des informations supplémentaires qui pourraient être glanées, ou une valeur additionnelle qui pourrait être créée?

Considérez le scénario de facturation mensuelle à nouveau. En analysant les données d'utilisation d'un client à mesure qu'elles sont générées, une organisation peut activer des fonctionnalités utiles, telles que la notification aux utilisateurs du fait qu'ils approchent d'une limite de facturation prédéfinie. Si les journaux des applications informatiques peuvent être analysés en temps réel, un administrateur système peut être immédiatement averti pour enquêter et prendre des mesures correctives.

Maintenant, nous allons combiner ces deux cas en un seul scénario et examiner comment nous pouvons créer une solution.

Exemple de scénario : Facturation de la route à péage et notification

Dans cet exemple simplifié, une société fictive, ABC Tolls, exploite des autoroutes à péage dans tout le pays. Les clients qui s'inscrivent auprès d'ABC Tolls reçoivent un émetteur-récepteur pour leur automobile. Lorsque le client passe à travers la zone de péage, un capteur reçoit des informations de l'émetteur-récepteur et enregistre les détails de la transaction à une base de données relationnelle. ABC Tolls dispose d'une architecture de traitement par lots traditionnelle. Chaque jour, un processus d'extraction-transformation-chargement (ETL) est exécuté pour traiter les transactions quotidiennes et les transformations afin qu'elles puissent être chargées dans leur entrepôt de données. Le lendemain, les analystes d'ABC Tolls analysent les données à l'aide d'un outil de rapport. En outre, une fois par mois (à la fin du cycle de facturation) un autre processus agrège toutes les transactions pour chacun des clients ABC Tolls pour calculer leur paiement mensuel.

ABC Tolls aimerait apporter quelques modifications à son système. La première exigence vient de son équipe d'analystes d'affaires. Ils ont demandé la possibilité d'exécuter des rapports à partir de leur entrepôt de données avec des données qui ne datent pas de plus de 30 minutes.

ABC Tolls développe également une nouvelle application mobile pour ses clients. Pendant le développement de l'application, ils ont décidé de créer de nouvelles fonctionnalités. Une fonctionnalité permet aux clients de définir un

seuil de dépenses pour leur compte. Si la facture de péage cumulative d'un client dépasse ce seuil, ABC Tolls souhaite envoyer un message d'application au client pour le prévenir que le seuil a été violé dans les 10 minutes suivant la violation.

Enfin, l'équipe d'exploitation d'ABC Tolls a des exigences supplémentaires qu'elle aimerait introduire dans le système. Tout en surveillant leurs stations de péage, ils veulent être immédiatement avertis lorsque le trafic de véhicules d'une station de péage tombe en dessous d'un seuil prédéfini pour chaque période de 30 minutes dans une journée. Par exemple, ils savent, d'après des données historiques, que l'une de leurs stations de péage voit environ 360 véhicules mercredi entre 14h00 et 14h30. Dans cette fenêtre de 30 minutes, l'équipe d'exploitation souhaite être avertie si la station de péage voit moins de 100 véhicules. Leurs opérateurs peuvent alors examiner pour déterminer si le trafic est normal ou si un autre facteur a contribué à la valeur inattendue (par exemple, un capteur défectueux ou un accident de voiture sur l'autoroute).

L'équipe d'ingénierie ABC Tolls a déterminé que son architecture actuelle nécessite certaines modifications pour répondre à ces exigences. Ils décident de construire un système d'analyse et d'ingestion de données en continu pour répondre aux exigences. Examinons chaque exigence et jetons un coup d'œil sur les améliorations de l'architecture qui vont prendre en charge chacune d'entre elles.

Condition 1 : Données plus récentes dans l'entrepôt de données

Actuellement, les données de l'entrepôt de données ABC Tolls peuvent durer jusqu'à 24 heures en raison de leur traitement par lots quotidien. Leur solution actuelle d'entrepôt de données est Amazon Redshift. Lors de l'examen des fonctionnalités des services Amazon Kinesis, ils ont reconnu que Kinesis Firehose peut recevoir un flux d'enregistrements de données et les insérer dans Amazon Redshift. Ils ont créé un flux de diffusion Kinesis Firehose et l'ont configuré pour copier les données sur leur table Amazon Redshift toutes les 15 minutes. Leur solution actuelle stocke les enregistrements dans un système de fichiers dans le cadre de leur traitement par lots. Dans le cadre de cette nouvelle solution, ils ont utilisé l'Amazon Kinesis Agent sur leurs serveurs pour transmettre leurs données de journal vers Kinesis Firehose. Étant donné que Kinesis Firehose utilise Amazon S3 pour stocker les données de streaming brutes avant qu'elles ne soient copiées dans Amazon Redshift, ABC Tolls n'a pas eu besoin de créer une autre solution pour archiver ses données brutes.

La figure 1 représente cette solution.

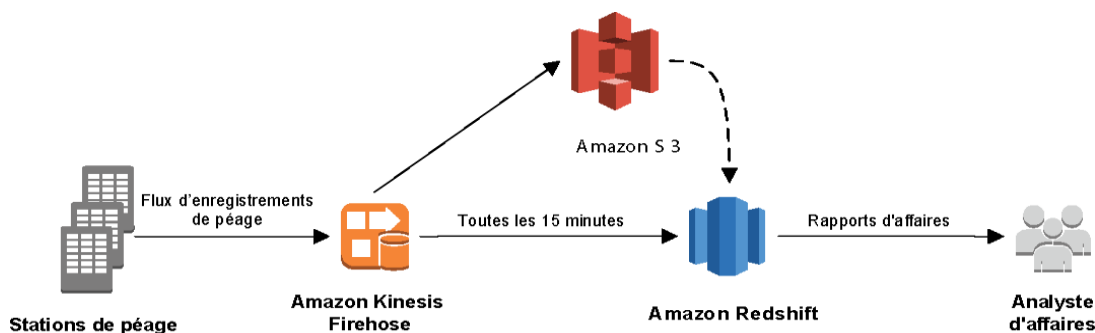


Figure 1 : Nouvelle solution utilisant Amazon Kinesis Firehose

Pour cette partie de l'architecture, ABC Tolls a choisi Kinesis Firehose. Passons en revue les fonctionnalités de Kinesis Firehose en détail.

Amazon Kinesis Firehose

Amazon Kinesis Firehose est la solution la plus simple pour charger les données de streaming dans AWS. Il peut capturer, transformer et charger les données de diffusion en continu dans Amazon Kinesis Analytics, Amazon S3, Amazon Redshift et Amazon Elasticsearch Service, ce qui permet une analyse quasiment en temps réel à l'aide des outils de veille stratégique et les tableaux de bord que vous utilisez déjà. C'est un service entièrement géré qui s'adapte automatiquement à votre débit de données et ne nécessite aucune administration continue. Il peut également regrouper, compresser et chiffrer les données avant de les charger, en limitant le volume de stockage utilisé à destination et en optimisant la sécurité.

Kinesis Firehose est un service entièrement géré. Vous n'avez pas besoin d'écrire des applications ou de gérer des ressources. Vous pouvez configurer vos producteurs de données pour envoyer des données vers Kinesis Firehose, qui délivre automatiquement les données à la destination que vous avez spécifiée. Vous pouvez également configurer Kinesis Firehose pour transformer vos données avant de les diffuser.

Envoi de données à un flux de diffusion Amazon Kinesis Firehose

Pour envoyer des données à votre flux de diffusion, il existe plusieurs options. AWS propose des kits SDK pour les langages de programmation populaires,

chacun fournissant des API pour Kinesis Firehose. AWS a également créé un utilitaire pour aider à envoyer des données à votre flux de diffusion.

Utilisation de l'API

L'API Kinesis Firehose propose deux opérations pour l'envoi des données à votre flux de diffusion. `PutRecord` envoie un enregistrement de données au sein d'un seul appel. `PutRecordBatch` peut envoyer plusieurs enregistrements de données au sein d'un seul appel.

Dans chaque méthode, vous devez spécifier le nom du flux de diffusion et l'enregistrement de données, ou un ensemble d'enregistrements de données, lors de l'utilisation de la méthode. Chaque enregistrement de données se compose d'une forme indistincte de données pouvant atteindre jusqu'à 1000 Ko et n'importe quel type de données.

Pour obtenir des informations détaillées et un exemple de code pour les opérations d'API Kinesis Firehose, reportez-vous à la section [écriture dans un flux de diffusion Firehose à l'aide du kit de développement logiciel \(SDK\) AWS](#)⁴.

Utilisation de l'agent Amazon Kinesis

L'agent Amazon Kinesis est une application logicielle Java autonome qui permet de collecter et d'envoyer facilement des données vers Kinesis Streams et Kinesis Firehose. L'agent surveille en permanence un ensemble de fichiers et envoie de nouvelles données à votre flux. L'agent gère la rotation de fichier, les points de contrôle et réessaie en cas de défaillance. Il transmet toutes vos données de manière fiable, rapide et simple. Il émet également des mesures Amazon CloudWatch pour vous aider à mieux surveiller et dépanner le processus de diffusion en continu.

Cet agent peut être installé dans des environnements basés sur des serveurs Linux tels que des serveurs Web, serveurs de journaux ou encore serveurs de base de données. Après avoir installé l'agent, configurez-le en spécifiant les fichiers à surveiller et le flux de destination pour les données. Une fois que l'agent est configuré, il collecte de façon durable les données depuis les fichiers et les envoie en toute fiabilité dans le flux de diffusion.

L'agent peut surveiller plusieurs répertoires de fichiers et écrire sur plusieurs flux. Il peut également être configuré pour pré-traiter les enregistrements de données avant leur envoi vers votre flux ou votre flux de diffusion.

Si vous envisagez une migration d'un système de fichiers par lots traditionnel vers la diffusion de données, il est possible que vos applications enregistrent déjà des événements dans des fichiers sur les systèmes de fichiers de vos serveurs d'applications. Ou, si votre application utilise une bibliothèque de journalisation populaire (par exemple, Log4j), il s'agit généralement d'une tâche simple pour la configurer à écrire dans les fichiers locaux. Quelle que soit la façon dont les données sont écrites dans un fichier journal, vous devriez envisager d'utiliser l'agent dans ce scénario. Il fournit une solution simple qui ne nécessite aucune modification, ou très peu, de votre système existant. Dans de nombreux cas, il peut être utilisé simultanément avec votre solution existante de traitement par lots. Dans ce scénario, il fournit un flux de données vers Kinesis Streams, à l'aide des fichiers journaux en tant que source de données pour le flux.

Dans notre exemple de scénario, ABC Tolls a choisi d'utiliser l'agent pour envoyer des données en continu à son flux de diffusion. Ils étaient déjà en train de créer des fichiers journaux, donc la transmission des entrées de journal à Kinesis Firehose était une simple installation et configuration de l'agent. Aucun code supplémentaire n'était nécessaire pour commencer à diffuser leurs données.

Transformation de données

Dans certains scénarios, vous souhaitez peut-être transformer ou améliorer vos données de diffusion en continu avant qu'elles ne soient livrées à leur destination. Par exemple, les producteurs de données peuvent envoyer des textes non structurés dans chaque enregistrement de données, et vous devez les convertir au format JSON avant de les transférer vers Amazon Elasticsearch Service.

Pour activer les transformations de données en continu, Kinesis Firehose utilise une fonction [AWS Lambda](#) que vous créez pour transformer vos données.⁵

Flux de transformation de données

Lorsque vous activez la transformation de données Kinesis Firehose, Kinesis Firehose met en mémoire tampon les données entrantes à hauteur de 3 MB ou la taille de mise en mémoire tampon que vous avez spécifiée pour le flux de diffusion, la taille la plus petite étant retenue. Kinesis Firehose appelle alors la fonction Lambda spécifiée avec chaque lot mis en mémoire tampon de manière asynchrone. Les données transformées sont envoyées à partir de Lambda pour

Kinesis Firehose pour la mise en mémoire tampon. Les données transformées sont diffusées à la destination quand la taille de mise en mémoire tampon ou l'intervalle de mise en mémoire tampon spécifiée est atteinte, selon la situation qui survient en premier. La figure 2 illustre ce processus pour un flux de diffusion qui fournit des données dans Amazon S3.

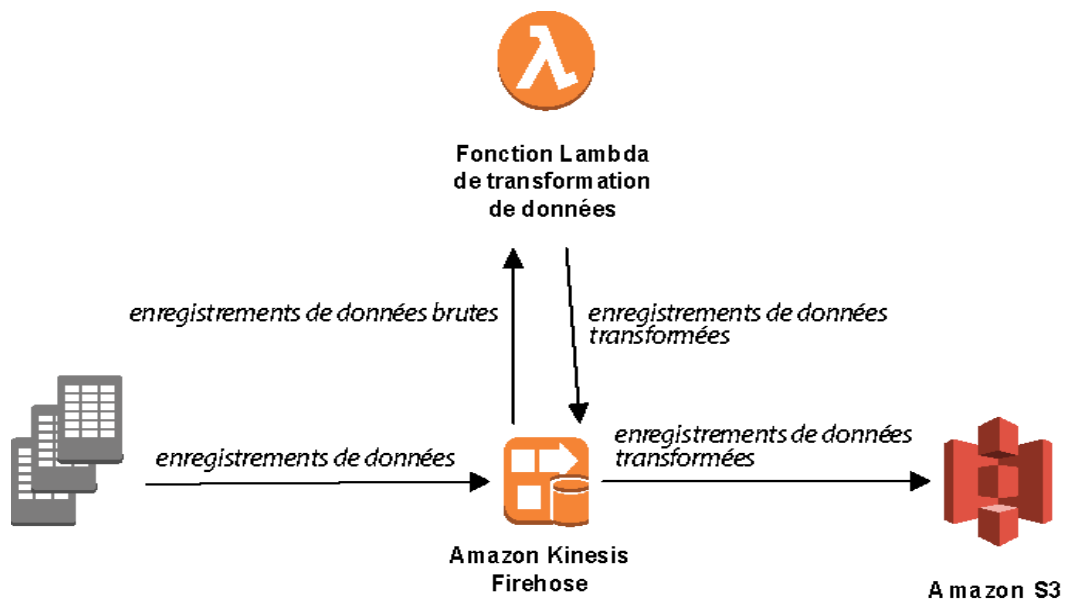


Figure 2 : Mise en mémoire tampon les données à l'aide de Kinesis Firehose et les fonctions Lambda

Livraison de données

Une fois les seuils de mise en mémoire tampon de votre flux de diffusion atteints, vos données sont livrées à la destination que vous avez configurée. Il existe quelques différences dans la façon dont Kinesis Firehose fournit des données pour chaque destination, nous allons les passer en revue dans les sections suivantes.

Service Amazon de simple stockage

[Amazon S3](#) est un stockage d'objet doté d'une interface de services web simple pour stocker et récupérer n'importe quelle quantité de données de n'importe où sur le Web.⁶ Il est conçu pour offrir une durabilité de 99,99999999 % et évoluer à travers des milliards d'objets dans le monde entier. Vous pouvez utiliser Amazon S3 en tant que stockage principal pour les applications natives

du cloud; en tant que référentiel en masse ou « lac de données » pour l'analyse; en tant que cible pour la sauvegarde, la récupération et la reprise après sinistre.

Format de livraison des données

Pour la livraison des données vers Amazon S3, Kinesis Firehose concatène plusieurs enregistrements entrants basés sur la configuration de mise en mémoire tampon de votre flux de diffusion, puis les envoie à Amazon S3 en tant qu'objet S3. Vous pouvez ajouter un séparateur d'enregistrement à la fin de chaque enregistrement avant de l'envoyer à Kinesis Firehose afin de pouvoir diviser un objet S3 transmis en enregistrements individuels.

Fréquence de livraison des données

La fréquence de livraison des données vers Amazon S3 est déterminée par la taille du tampon S3 et la valeur d'intervalle de mise en mémoire tampon configurée pour votre flux de diffusion. Kinesis Firehose met en mémoire tampon les données entrantes avant de les transférer vers Amazon S3. Vous pouvez configurer les valeurs de la taille de tampon Amazon S3 (1 MB à 128 MB) ou de l'intervalle de mise en mémoire tampon (60 à 900 secondes). La première condition remplie déclenche la transmission de données vers Amazon S3. Notez que dans les cas où la diffusion des données vers la destination prend du retard par rapport à l'écriture des données dans le flux de diffusion, Kinesis Firehose augmente dynamiquement la taille du tampon pour rattraper le retard et veiller à ce que toutes les données soient livrées à la destination.

Flux de données

La figure 3 montre le flux de données pour les destinations Amazon S3.

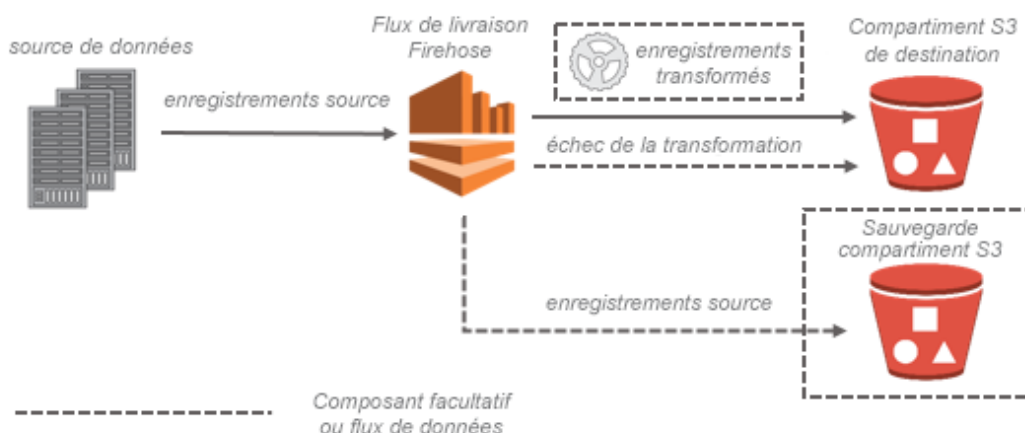


Figure 3 : Flux de données à partir de Kinesis Firehose pour les compartiments S3

Amazon Redshift

[Amazon Redshift](#) est un service d'entrepôt de données rapide, et entièrement géré qui simplifie et rentabilise l'analyse de toutes vos données à l'aide du langage SQL standard et de vos outils de veille stratégique (Business Intelligence -BI) existants.⁷ Il vous permet d'exécuter des requêtes analytiques complexes sur des pétaoctets de données structurées utilisant l'optimisation de requête sophistiquée, d'un stockage en colonnes sur des disques locaux haute performance et d'une exécution de requête massivement parallèle. La plupart des résultats reviennent en quelques secondes.

Dans notre exemple, ABC Tolls utilisait déjà Amazon Redshift comme solution d'entrepôt de données. Lorsqu'ils ont mis en œuvre leur solution de diffusion de données en continu, ils ont configuré leurs flux de diffusion pour transmettre leurs données diffusées en continu à leur cluster Amazon Redshift.

Format de livraison des données

Pour la livraison des données vers Amazon Redshift, Kinesis Firehose délivre tout d'abord les données entrantes à votre compartiment S3 au format indiqué précédemment. Amazon Kinesis Firehose émet ensuite une commande COPY Redshift pour charger les données à partir de votre compartiment S3 vers votre cluster Amazon Redshift. Vous devez vous assurer qu'une fois que Kinesis Firehose concatène plusieurs enregistrements entrants dans un objet S3, cet objet S3 peut être copié dans votre cluster Amazon Redshift. Pour plus

d'informations, consultez la section [Amazon Redshift Paramètres du format des données de commande COPY](#).

Fréquence de livraison des données

La fréquence des opérations COPY de données depuis Amazon S3 vers Amazon Redshift est déterminée par la rapidité avec laquelle votre cluster Amazon Redshift peut traiter la commande COPY. S'il y a encore des données à copier, Kinesis Firehose délivre une nouvelle commande COPY dès que la commande COPY précédente a bien été traitée par Amazon Redshift.

Flux de données

La figure 4 montre les flux de données pour les destinations Amazon Redshift.

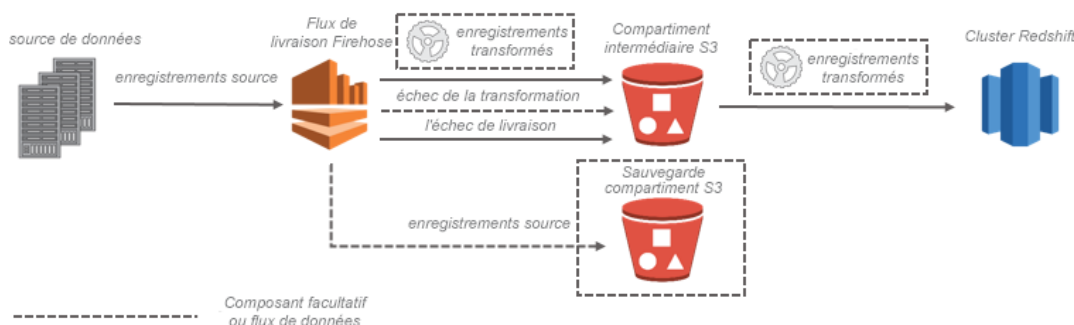


Figure 4 : Flux de données de Kinesis Firehose vers Amazon Redshift

Amazon Elasticsearch Service

[Amazon Elasticsearch Service](#) est un service entièrement géré qui fournit des API faciles à utiliser et des capacités en temps réel parallèlement à la disponibilité, l'évolutivité et la sécurité nécessaire aux charges de production.⁸ Amazon ES facilite le déploiement, le fonctionnement et la mise à l'échelle d'Elasticsearch pour l'analyse des journaux, la recherche en texte intégral, la surveillance des applications, etc.

Format de livraison des données

Pour la livraison des données à Amazon ES, Kinesis Firehose met en mémoire tampon les enregistrements entrants basés sur la configuration de mise en tampon de votre flux de diffusion, puis génère une requête groupée Elasticsearch dans votre cluster Elasticsearch. Vous devez vous assurer que

vos enregistrements sont codés en UTF-8 et aplatis en un seul objet JSON à une seule ligne avant de l'envoyer à Kinesis Firehose.

Fréquence de livraison des données

La fréquence de livraison des données vers Amazon ES est déterminée par la taille de la mémoire tampon Elasticsearch et les valeurs d'intervalle de mémoire tampon configurées pour votre flux de diffusion. Kinesis Firehose met en mémoire tampon les données entrantes avant de les transférer vers Amazon ES. Vous pouvez configurer les valeurs de la taille de la mémoire tampon Elasticsearch (1 MB à 100 MB) ou de l'intervalle de mise en mémoire tampon (60 à 900 secondes). La première condition remplie déclenche la transmission de données vers Amazon ES. Notez que dans les cas où la diffusion des données vers la destination prend du retard par rapport à l'écriture des données dans le flux de diffusion, Kinesis Firehose augmente dynamiquement la taille du tampon pour rattraper le retard et veiller à ce que toutes les données soient livrées à la destination.

Flux de données

La figure 5 illustre le flux de données pour les destinations Amazon ES.

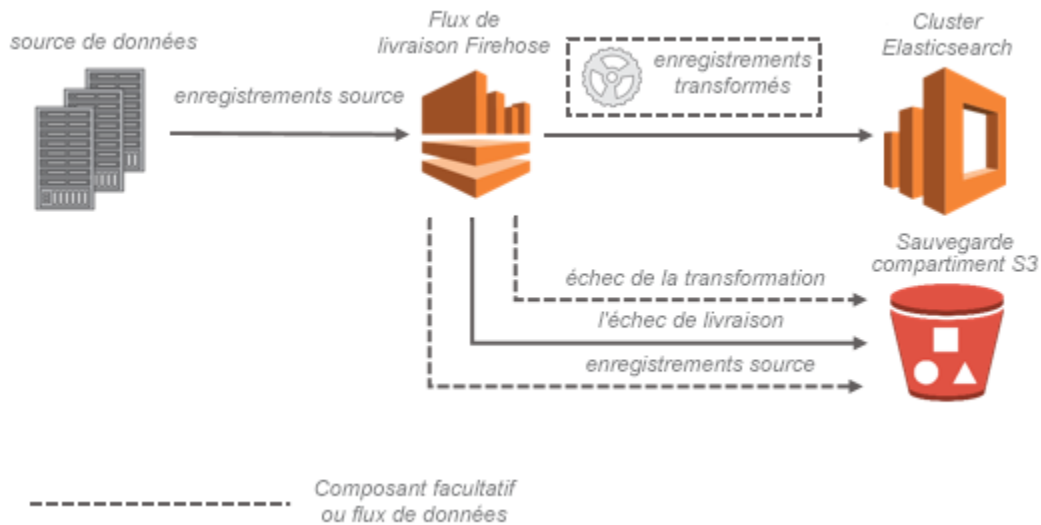


Figure 5 : Livraison de données de Kinesis Firehose au cluster Amazon ES

Résumé

Kinesis Firehose est la solution la plus simple pour conserver vos données diffusées en continu vers une destination prise en charge. Il s'agit d'une solution entièrement gérée, nécessitant peu ou pas de développement pour l'utiliser. Pour ABC Tolls, l'utilisation de Kinesis Firehose était un choix naturel. Ils utilisaient déjà Amazon Redshift comme solution d'entrepôt de données. Et parce que leurs sources de données écrivaient continuellement dans les journaux de transactions, ils pouvaient utiliser l'agent Amazon Kinesis pour diffuser ces données sans écrire de code supplémentaire.

Maintenant que ABC Tolls a créé un flux d'enregistrements de péage et reçoit ces enregistrements via Kinesis Firehose, ils peuvent l'utiliser comme base pour leurs autres exigences en matière de diffusion de données en continu.

Condition 2 : Alertes de seuil de facturation

Pour soutenir la fonctionnalité d'envoyer une notification lorsqu'un seuil de dépenses est dépassé, l'équipe de développement ABC Tolls a créé une application mobile et une table [Amazon DynamoDB](#)⁹. L'application permet aux clients de définir leur seuil, et la table stocke cette valeur pour chaque client. La table est également utilisée pour stocker le montant de dépense cumulée de chaque client, chaque mois. Pour fournir des notifications en temps opportun, ABC Tolls a besoin pour mettre à jour la valeur cumulative dans ce tableau en temps opportun et comparer cette valeur avec le seuil pour déterminer si une notification doit être envoyée au client. Étant donné que leurs transactions de péages sont déjà diffusées en continu via Kinesis Firehose, ils ont décidé d'utiliser ces données de streaming comme source pour leur agrégation et leur alerte. Il s'agit d'une solution idéale au problème car Kinesis Analytics leur a permis d'utiliser SQL pour regrouper les données diffusées en continu. Dans cette solution, Kinesis Analytics totalise la valeur de transaction pour chaque client sur une période de 10 minutes (fenêtre). A la fin de la fenêtre, elle envoie les totaux à un flux Kinesis. Ce flux est la source d'événement pour une fonction AWS Lambda. La fonction Lambda interroge la table DynamoDB pour récupérer les seuils et le total actuel dépensé par chaque client représenté dans la sortie de Kinesis Analytics. Pour chaque client, la fonction Lambda met à jour le montant total actuel dans DynamoDB et compare le total avec le seuil. Si le seuil est dépassé, il utilise le kit de développement logiciel (SDK) AWS pour indiquer à Amazon Simple Notification Service (SNS) d'envoyer une notification aux clients.

La figure 6 montre l'architecture pour cette solution.

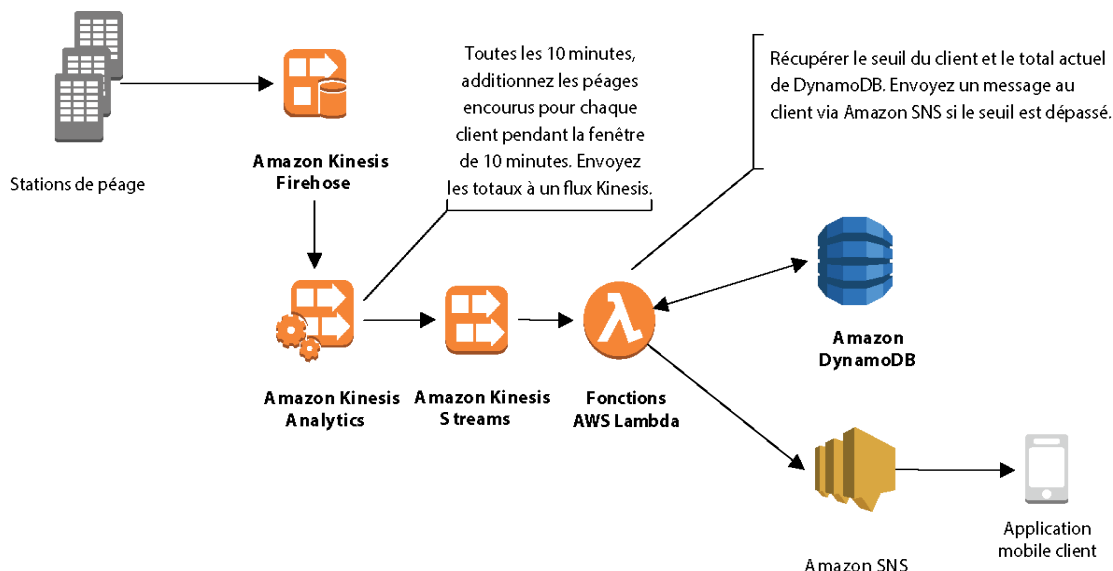


Figure 6 : Architecture pour les alertes de seuil de facturation et les notifications

Avec cette solution, ABC Tolls fournit à ses clients une notification lorsqu'ils approchent des limites de dépenses.

Pour recueillir des informations en temps réel à partir de leurs données de en continu, ABC Tolls a choisi d'utiliser Kinesis Analytics pour analyser ses données en continu. Avec Kinesis Analytics, ABC Tolls utilisait SQL, un langage qu'il connaissait déjà, pour inspecter ses données au fur et à mesure de leur diffusion. Examinons Kinesis Analytics plus en détail.

Amazon Kinesis Analytics

Avec Kinesis Analytics, vous pouvez traiter et analyser des données en diffusion à l'aide de SQL. Le service vous permet de créer et d'exécuter rapidement un code SQL puissant sur des sources de diffusion en continu pour effectuer des analyses de séries chronologiques, générer des tableaux de bord en temps réel et créer des métriques en temps réel.

Pour démarrer avec Kinesis Analytics, vous créez une application Kinesis Analytics, qui lit et traite en continu des données en diffusion continue. Le service prend en charge l'intégration des données provenant des sources de diffusion Kinesis

Streams et Kinesis Firehose. Vous créez ensuite votre code SQL à l'aide de l'éditeur interactif et vous le testez avec des données de diffusion. Vous pouvez également configurer des destinations dans lesquelles vous voulez que Kinesis Analytics conserve les résultats. Kinesis Analytics prend en charge Kinesis Firehose (Amazon S3, Amazon Redshift et Amazon Elasticsearch Service), et Kinesis Streams en tant que destinations.

Concepts clés

Une *application* est la ressource principale de Kinesis Analytics que vous pouvez créer dans votre compte. Les applications Kinesis Analytics lisent et traitent continuellement des données de diffusion en temps réel. Vous écrivez le code d'application à l'aide de SQL pour traiter les données de diffusion entrantes et produire une sortie. Kinesis Analytics écrit ensuite les données sortantes sur une destination configurée. La figure 7 illustre une architecture d'application typique.

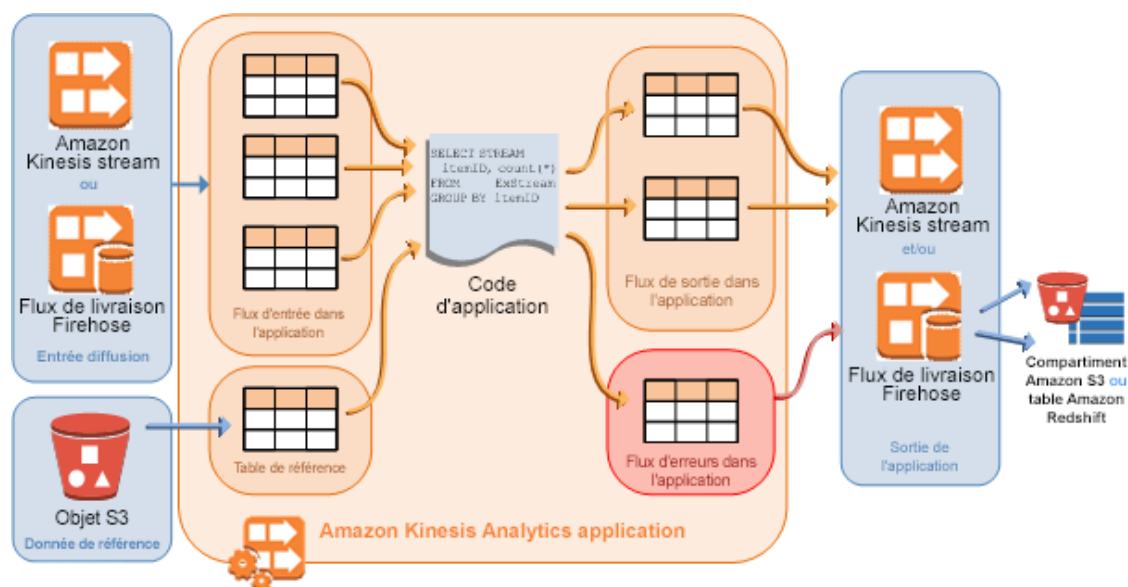


Figure 7 : Architecture pour une application Kinesis Analytics

Chaque application possède un nom, une description, un ID de version et un statut. Lorsque vous créez une application, vous pouvez configurer l'entrée, créer le code d'application, et configurer la sortie.

Entrée

L'entrée d'application est la source de diffusion pour votre application. Vous pouvez sélectionner un flux Kinesis ou un flux de diffusion comme source de

diffusion. Vous pouvez configurer une source de données de référence pour enrichir votre flux de données d'entrée au sein de l'application. Celle-ci crée une table de référence intégrée à l'application. Vous devez stocker vos données de référence en tant qu'objet dans un compartiment S3. Lorsque l'application démarre, Kinesis Analytics lit l'objet S3 et crée une table intégrée à l'application.

ABC Tolls ont utilisé leur flux de diffusion comme entrée pour leur application Kinesis Analytics.

Code d'application

Votre code d'application se compose d'un ensemble d'instructions SQL qui traitent les entrées et produisent des sorties. Vous pouvez écrire des instructions SQL par rapport aux flux dans l'application, aux tables de référence, et vous pouvez écrire des requêtes JOIN pour combiner des données provenant de ces deux sources.

Dans sa forme la plus simple, le code d'application peut être une instruction SQL unique qui effectue une sélection à partir d'une entrée de diffusion et insère les résultats dans une sortie de diffusion. Il peut également s'agir d'un ensemble d'instructions SQL où la sortie d'une instruction alimente les données entrantes de la déclaration SQL suivante. En outre, vous pouvez rédiger un code d'application pour scinder un flux de données entrantes en plusieurs flux, puis appliquer des requêtes supplémentaires pour traiter ces flux.

Sortie

Dans votre code d'application, les résultats d'une requête sont transmis à des flux intégrés à l'application. Dans votre code d'application, vous pouvez créer un ou plusieurs flux intégrés à l'application pour stocker des résultats intermédiaires. Vous pouvez ensuite configurer, le cas échéant, une sortie d'application pour conserver des données des flux intégrés à l'application qui contiennent votre sortie d'application (également appelés flux de sortie intégrés à l'application) dans des destinations externes. Les destinations externes peuvent être un flux de diffusion ou d'un flux Kinesis.

ABC Tolls a utilisé Kinesis Streams en tant que destination pour leurs valeurs regroupées.

Résumé

Kinesis Analytics vous permet d'utiliser SQL pour recueillir des informations sur vos données car il diffuse via le système. ABC Tolls a écrit son SQL pour effectuer des agrégations de 10 minutes pour totaliser les péages supportés par leurs clients. Les valeurs de sortie de ces agrégations de 10 minutes pourraient être comparées aux seuils de leurs clients.

Comme mentionné précédemment, Kinesis Analytics génère ses résultats dans Kinesis Streams ou Kinesis Firehose. Dans cet exemple, ABC Tolls a choisi d'envoyer la sortie de Kinesis Analytics vers un flux Kinesis en raison de l'intégration de Kinesis Streams avec AWS Lambda. Apprenons-en plus sur les flux Kinesis.

Amazon Kinesis Streams

Amazon Kinesis Streams vous permet de concevoir des applications personnalisées en temps réel, à l'aide des structures de traitement de flux courants et du chargement automatique des données de streaming dans un magasin de données. Vous pouvez configurer des centaines de milliers de producteurs de données de manière à ce qu'ils ajoutent des données en continu à un flux Kinesis, par exemple, les données issues des parcours de navigation sur un site Web, des journaux d'application, des capteurs IoT et des flux de réseaux sociaux. En moins d'une seconde, ces données sont accessibles à votre application, qui peut les lire et les traiter depuis le flux.

Lors de l'implémentation d'une solution avec Kinesis Streams, vous allez créer des applications de traitement de données personnalisées appelées *applications Kinesis Streams*. Une application Kinesis Streams lit les données d'un flux Kinesis comme des enregistrements de données.

Même si vous pouvez utiliser Kinesis Streams pour résoudre divers problèmes de transmission de données en continu, une utilisation courante est l'agrégation en temps réel ou l'analyse des données suivie du chargement de l'ensemble des données dans un entrepôt de données ou un cluster map-reduce.

Les données sont placées dans Kinesis Streams, ce qui garantit la durabilité et l'élasticité. Le délai entre le moment où un enregistrement est placé dans le flux et le moment où il peut être récupéré (délai de mise en attente) est généralement inférieur à 1 seconde - en d'autres termes, une application Kinesis

Streams peut commencer à consommer les données du flux presque immédiatement après l'ajout des données. Etant donné que Kinesis Streams est un service géré, il vous libère du fardeau opérationnel lié à la création et à l'exécution d'un pipeline d'entrée de données.

Envoi de données à Amazon Kinesis Streams

Il existe plusieurs mécanismes pour envoyer des données à votre flux. AWS propose des kits SDK pour les langages de programmation populaires, chacun d'entre eux fournit des API pour Kinesis Streams. AWS a également créé plusieurs utilitaires pour vous aider à envoyer des données à votre flux. Examinons chacune des approches que vous pouvez utiliser et les raisons de la choisir.

Agent Amazon Kinesis

L'agent Amazon Kinesis a été présenté précédemment comme un outil qui peut être utilisé pour envoyer des données à Kinesis Firehose. Le même outil peut être utilisé pour envoyer des données vers Kinesis Streams. Pour plus d'informations sur l'installation et la configuration de l'agent Kinesis, consultez [Ecriture sur Amazon Kinesis Firehose à l'aide d'Amazon Kinesis Agent](#).¹⁰

Producteur de bibliothèque Amazon Kinesis (KPL)

Le KPL simplifie le développement d'applications de producteurs, en permettant aux développeurs d'atteindre un débit élevé d'écriture pour un ou plusieurs flux Kinesis. Le KPL est une bibliothèque facile à utiliser et hautement configurable que vous installez sur les hôtes qui génèrent les données que vous souhaitez diffuser en continu vers Kinesis Streams. IL joue le rôle d'intermédiaire entre votre producteur de code d'application et les actions d'API Kinesis Streams. Le KPL effectue les tâches principales suivantes :

- Il écrit à un ou plusieurs flux Kinesis avec un mécanisme de tentatives automatique et configurable
- Il permet de collecter les enregistrements et utilise `PutRecords` pour écrire plusieurs enregistrements sur plusieurs fragments par demande
- Il regroupe les enregistrements utilisateur pour augmenter la capacité de données et d'améliorer le débit

- Il s'intègre de façon transparente à la bibliothèque client KCL (Amazon Kinesis Client Library) afin de regrouper les enregistrements par lots sur le consommateur
- Il envoie des statistiques Amazon CloudWatch en votre nom pour fournir davantage de visibilité sur les performances du producteur

Le KPL peut être utilisé dans des cas d'utilisation synchrones ou asynchrones. Nous vous suggérons d'utiliser les meilleures performances de l'interface asynchrone sauf s'il existe une raison précise d'utiliser le comportement synchrone. Pour plus d'informations sur ces deux cas d'utilisation et des exemples de code, veuillez consulter [Ecriture dans votre flux Streams à l'aide du KPL](#).¹¹

Le KPL peut vous aider à construire des producteurs à haute performance. Prenons l'exemple d'une situation où vos instances Amazon Elastic Compute Cloud (EC2) servent de proxy pour la collecte d'événements de 100 bytes à partir de centaines ou de milliers de périphériques de faible puissance et d'enregistrements d'écriture dans un flux Kinesis. Ces instances EC2 doivent chacune d'écrire des milliers d'événements par seconde dans votre flux Kinesis. Pour atteindre le débit nécessaire, les producteurs doivent mettre en œuvre une logique complexe telle que le traitement par lot ou le multi chargement, en plus de réessayer la logique et la désagrégation d'enregistrements du côté du consommateur. Le KPL effectue toutes ces tâches pour vous.

Etant donné que le KPL met en tampon vos enregistrements avant leur envoi à un flux Kinesis, le KPL peut entraîner un délai de traitement supplémentaire, en fonction de la durée pendant laquelle vous avez configuré le fichier KPL pour tamponner les enregistrements avant de les envoyer à Kinesis. Un temps tampon plus long permet d'obtenir des rendements d'emballage plus élevés et de meilleures performances. Les applications qui ne peuvent pas tolérer ce délai supplémentaire peuvent avoir besoin d'utiliser le kit de développement logiciel (SDK) AWS directement.

Si votre application ne consigne pas les enregistrements dans un fichier local et crée un grand nombre de petits enregistrements par seconde, envisagez d'utiliser la clé KPL.

Pour plus d'informations sur l'utilisation du KPL pour produire des données, consultez [Développement des producteurs Amazon Kinesis Streams à l'aide de la bibliothèque Amazon Kinesis Producer](#).¹²

API Amazon Kinesis

Une fois qu'un flux est créé, vous pouvez ajouter vos enregistrements de données. Un enregistrement est une structure de données qui contient les données qui doivent être traitées sous une forme indistincte de données. Après avoir stocké les données dans l'enregistrement, Kinesis Streams n'inspecte, n'interprète ou ne modifie en aucun cas les données.

Il existe deux opérations différentes dans l'API Kinesis Streams qui ajoutent des données à un flux : `PutRecords` et `PutRecord`. L'opération `PutRecords` envoie plusieurs enregistrements à votre flux par requête HTTP et l'opération particulière `PutRecord` envoie les enregistrements à votre flux un par un (une requête HTTP distincte est requise pour chaque enregistrement). Vous préférerez utiliser `PutRecords` pour la plupart des applications, car il permettra d'augmenter le débit par producteur de données.

Etant donné que les API sont exposées dans tous les kits de développement logiciel (SDK) AWS, l'utilisation de l'API pour écrire les enregistrements fournit la solution la plus flexible pour envoyer des données à un flux Kinesis. Si vous ne pouvez pas utiliser l'agent Kinesis ou KPL (par exemple, vous souhaitez écrire les messages directement depuis une application mobile, ou vous souhaitez minimiser la latence de bout en bout des messages autant que possible), alors utilisez l'API pour écrire les enregistrements dans votre flux Kinesis.

Pour plus de détails sur ces API, consultez [Utilisation de l'API](#) dans la documentation Kinesis Streams.¹³ Les détails pour chaque opération d'API sont disponibles dans la section [Références de l'API d'Amazon Kinesis Streams](#).¹⁴

Traitement des données dans Amazon Kinesis Streams

Un consommateur est une application qui lit et traite des données provenant de Kinesis Streams. Vous pouvez créer des utilisateurs Kinesis Streams de différentes manières. Dans cette section, nous allons aborder quatre des approches les plus courantes : à l'aide de Kinesis Analytics, à l'aide de la bibliothèque KCL, à l'aide d'Amazon Lambda, et à l'aide de Kinesis Streams directement via l'API.

Utilisation d'Amazon Kinesis Analytics

Auparavant, nous avons décrit comment Kinesis Analytics peut être utilisé pour analyser des données de diffusion à l'aide de code SQL standard. Kinesis

Analytics peut lire les données à partir de votre flux Kinesis et les traiter à l'aide du SQL que vous fournissez. Pour en savoir plus sur le traitement de vos données de diffusion en continu à l'aide de Kinesis Analytics, consultez la [Configuration de l'entrée d'application](#) dans le Guide du développeur de Kinesis Analytics.

Utilisation de la bibliothèque client Amazon Kinesis (KCL)

Vous pouvez développer une application consommateur pour Kinesis Streams à l'aide du KCL. Même si vous pouvez utiliser l'API Kinesis Streams pour obtenir des données d'un flux Amazon Kinesis, nous vous recommandons d'utiliser les modèles de conception et du code pour les applications consommateur fournies par la KCL.

Le KCL vous aide à utiliser et traiter des données provenant d'un flux Kinesis. Ce type d'application est également appelée consommateur. Le KCL s'occupe de la plupart des tâches complexes associés aux calculs distribués, telles que l'équilibrage de charge entre plusieurs instances, la réaction aux instances défaillantes, le point de contrôle des enregistrements traités et la réaction au repartitionnement. Elle vous permet ainsi de vous concentrer sur l'écriture de la logique de traitement de l'enregistrement.

Le KCL est une bibliothèque Java; une prise en charge des langages autres que Java est fournie à l'aide d'une interface multilingue. Lors de l'exécution, une application KCL instancie un programme avec des informations de configuration, puis utilise un processeur d'enregistrements pour traiter les données reçues d'un flux Kinesis. Vous pouvez exécuter une application KCL sur un nombre illimité d'instances. Plusieurs instances de la même application coordonnent les défaillances et équilibrent la charge de manière dynamique. Vous pouvez également avoir plusieurs applications KCL travaillant sur le même flux, sous réserve des limites de débit. Le KCL joue le rôle d'intermédiaire entre votre logique de traitement d'enregistrements et Kinesis Streams.

Pour plus d'informations sur la manière de créer votre propre application KCL, reportez-vous à [Développement d'applications consommateur Amazon Kinesis Streams à l'aide de la bibliothèque client Amazon Kinesis](#).¹⁵

Utilisation d'AWS Lambda

[AWS Lambda](#) est un service de calcul qui vous permet d'exécuter du code sans provisionnement ou gestion de serveurs.¹⁶ AWS Lambda exécute le code

uniquement lorsque cela est nécessaire et s'adapte automatiquement. Avec AWS Lambda, vous pouvez exécuter le code sans aucune administration. AWS Lambda exécute votre code sur une infrastructure de calcul à haute disponibilité et effectue toute l'administration des ressources de calcul, y compris la maintenance des serveurs et du système d'exploitation, le provisionnement de capacités et la mise à l'échelle automatique, ainsi que la surveillance et la journalisation du code. Il vous suffit de fournir le code dans l'un des langages que AWS Lambda prend en charge.

Vous pouvez vous abonner aux fonctions Lambda pour lire automatiquement les lots d'enregistrements de votre flux Kinesis et les traiter si des enregistrements sont détectés dans le flux. AWS Lambda interroge ensuite régulièrement le flux (une fois par seconde), à la recherche de nouveaux enregistrements. Lorsqu'il détecte de nouveaux enregistrements, il appelle votre fonction Lambda en transmettant les nouveaux enregistrements en tant que paramètre. Si aucun nouvel enregistrement n'est détecté, votre fonction Lambda n'est pas appelée.

Pour plus d'informations sur l'utilisation d'AWS Lambda pour la consommation des données provenant de Kinesis Streams, consultez [Utilisation d'AWS Lambda avec Amazon Kinesis](#).¹⁷

Utilisation de l'API

Dans la plupart des cas d'utilisation, vous devez utiliser KCL ou AWS Lambda pour récupérer et traiter les données d'un flux. Toutefois, si vous préférez écrire votre propre application consommateur à partir de zéro, il existe plusieurs méthodes qui le permettent. L'API Kinesis Streams fournit les méthodes `GetShardIterator` et `GetRecords` pour extraire des données d'un flux. Il s'agit d'un modèle tiré, où votre code tire des données directement à partir des fragments du flux. Pour plus d'informations sur l'écriture de votre propre application consommateur à l'aide de l'API, consultez [Développement d'applications consommateur Amazon Kinesis Streams à l'aide du Amazon Kinesis Streams API](#). Les détails sur l'API sont disponibles dans la section [Référence de l'API Amazon Kinesis Streams](#).¹⁸

Choisir le meilleur modèle de consommateur pour votre application

Comment pouvez-vous savoir quel modèle de consommateur est le plus adapté à votre cas d'utilisation ? Chaque approche a son propre ensemble de compromis et vous devez décider ce qui vous importe. Voici quelques conseils généraux pour vous aider à choisir le bon modèle.

Dans la plupart des cas, nous vous conseillons de commencer avec AWS Lambda. Sa facilité d'utilisation et le modèle de déploiement simple vont vous permettre de créer rapidement un consommateur de données. Le compromis à l'aide d'AWS Lambda est que chaque appel de votre fonction Lambda doit être considéré comme statique. En d'autres termes, vous ne pouvez pas facilement utiliser les résultats d'appels de votre fonction précédente (par ex., des lots d'enregistrements antérieurs de votre flux). De même, tenez compte du fait que le délai d'exécution maximal pour une seule fonction Lambda est de 5 minutes. Si un seul lot d'enregistrements prend plus de 5 minutes à traiter, AWS Lambda peut ne pas être le meilleur consommateur pour votre cas d'utilisation.

Si vous décidez que vous ne pouvez pas utiliser AWS Lambda, envisagez de concevoir votre propre application de traitement avec le KCL. Étant donné que vous déployez des applications KCL dans des instances EC2 au sein de votre compte AWS, vous disposez d'une grande flexibilité et contrôlez la persistance des données locales et les conditions d'état de vos données.

Votre troisième option consiste à créer directement votre propre application à l'aide des API. Cela vous offre davantage de maîtrise et de flexibilité mais vous devez également générer votre propre logique pour gérer les fonctions communes de l'application consommateur telles que le contrôle, la mise à l'échelle et le basculement.

Résumé

Kinesis Streams vous permet de recevoir des données diffusées en continu. Vous pouvez dimensionner un flux Kinesis afin de prendre en charge quelques enregistrements par seconde ou des millions d'enregistrements par seconde. Pour ABC Tolls, le débit de données dans leurs diffusions en continu n'a pas été volumineux. Cependant, ils ont bénéficié d'une intégration directe avec AWS Lambda, ce qui leur a permis de passer facilement des appels d'API vers Amazon SNS pour les notifications d'utilisateur.

Condition 3 : Autres seuils d'alertes

L'exigence finale est semblable à la précédente, mais elle présente un problème supplémentaire. Pour récapituler cette exigence finale, les opérateurs de péage d'ABC Tolls souhaitent être immédiatement informés lorsque le trafic de véhicules d'une station de péage est inférieur à un seuil prédéfini pour chaque période de 30 minutes dans une journée. Par exemple, ils savent, d'après des données

historiques, que l'une de leurs stations de péage voit environ 360 véhicules mercredi entre 14h00 et 14h30. Dans cette fenêtre de 30 minutes, si une station de péage voit moins de 100 véhicules, ils souhaitent recevoir une notification.

ABC Tolls souhaite comparer les totaux actuels des véhicules pour chaque station avec un taux moyen connu pour cette station. Pour ce faire, ils ont créé un fichier contenant les valeurs de trafic de seuil pour chaque créneau de 30 minutes, pour chaque station. Comme décrit précédemment, Kinesis Analytics prend en charge l'utilisation de données de référence. Il créera un flux intégré à l'application (comme une table) basé sur les données d'un fichier stocké dans un compartiment S3. Avec cette mise en place, les développeurs d'ABC Tolls ont été en mesure d'écrire le code SQL dans leur application Kinesis Analytics pour compter le nombre de véhicules observés à chaque poste sur un créneau de 30 minutes et de comparer ces valeurs avec les seuils dans le fichier. Si le seuil a été dépassé, Kinesis Analytics génère un enregistrement dans un flux Kinesis. Lorsque des enregistrements arrivent dans le flux, une fonction Lambda est exécutée. Elle utilise Amazon SNS pour envoyer une notification à des opérateurs de péage d'ABC Tolls AWS. La figure 8 illustre l'architecture de ce scénario.

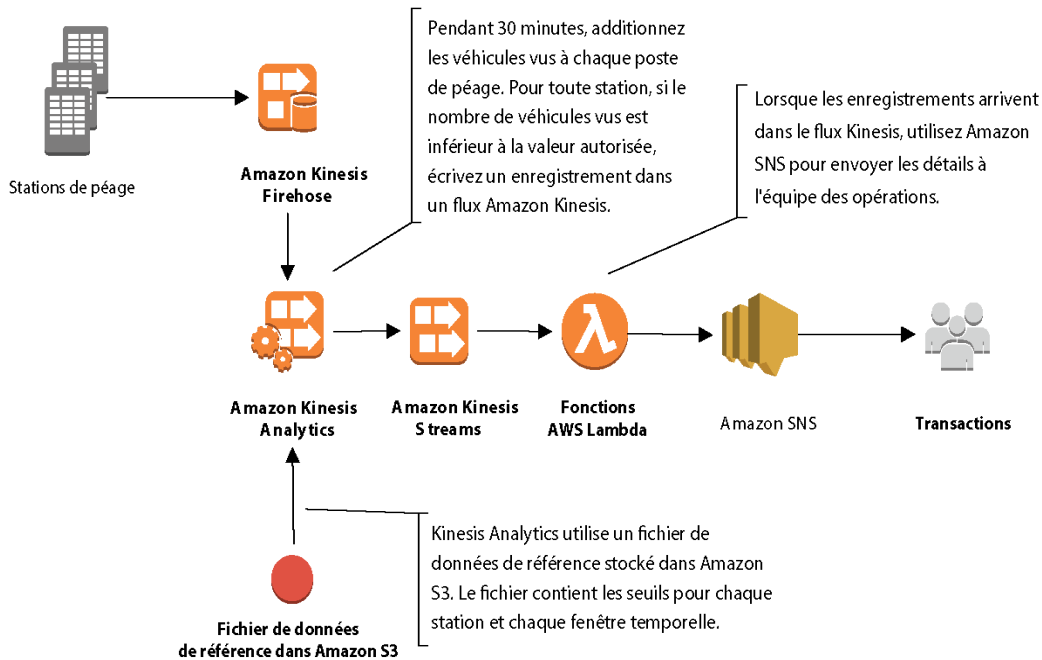


Figure 8 : Architecture pour les alertes et les notifications par tranches de 30 minutes

Architecture complète

Avec une solution à chaque besoin, nous avons maintenant notre solution globale de diffusion en continu, comme le montre la figure 9.

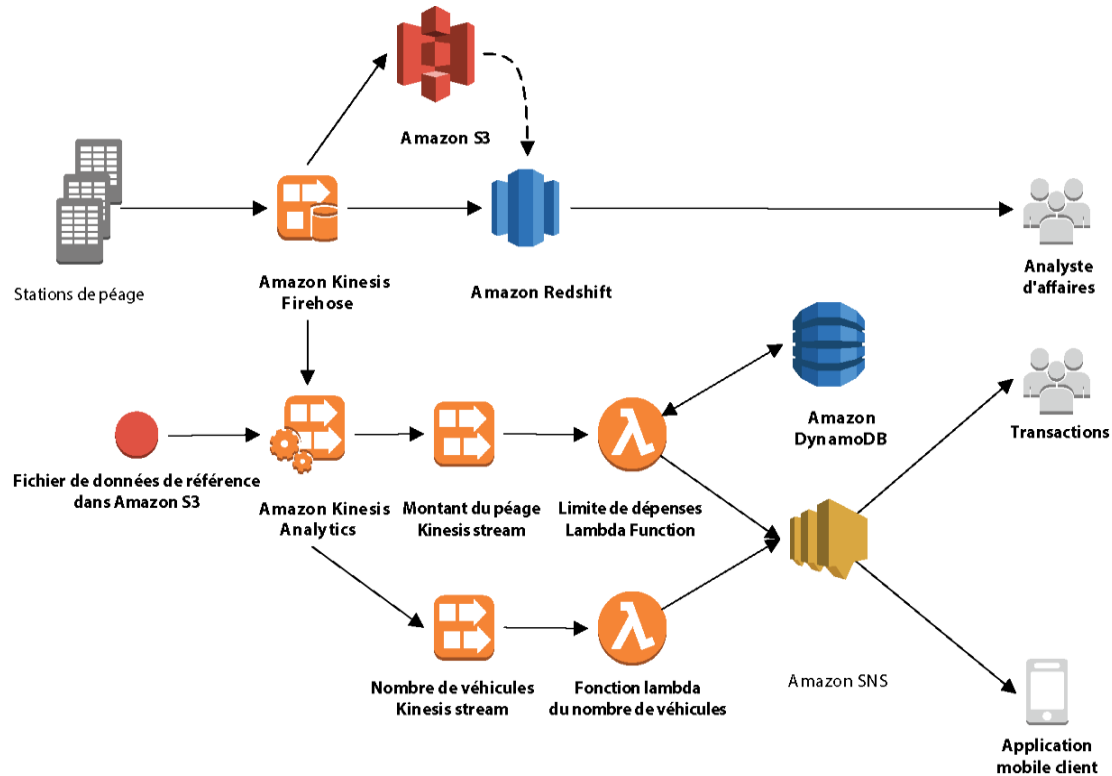


Figure 9 : L'architecture de la solution globale de diffusion en continu

Cette conception fournit à ABC Tolls une architecture réactive flexible. Grâce à la diffusion en continu de transactions de leurs clients en temps réel, ils sont en mesure de réaliser leurs exigences avec très peu d'efforts de développement et une infrastructure minimale à gérer.

Conclusion

Dans ce document, nous avons passé en revue la façon dont la société fictive, ABC Tolls, a utilisé les services Amazon Kinesis pour déplacer un flux de production par lots traditionnel vers un flux de production en continu. Cette migration leur a permis d'ajouter de nouvelles fonctions et fonctionnalités qui n'étaient pas possibles avec leur solution par lots héritée.

En analysant les données au fur et à mesure de leur création, vous obtiendrez un aperçu de ce que fait votre entreprise en ce moment. Les services Amazon Kinesis vous permettent de vous concentrer sur votre application pour prendre des décisions commerciales urgentes, plutôt que de déployer et de gérer l'infrastructure.

Participants

Les personnes et organisations suivantes ont participé à l'élaboration de ce document :

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

Remarques

¹ <https://aws.amazon.com/kinesis/streams/>

² <https://aws.amazon.com/kinesis/firehose/>

³ <https://aws.amazon.com/kinesis/analytics/>

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

⁵ <https://aws.amazon.com/lambda/>

⁶ <https://aws.amazon.com/s3/>

⁷ <https://aws.amazon.com/redshift/>

⁸ <https://aws.amazon.com/elasticsearch-service/>

⁹ <https://aws.amazon.com/dynamodb/>

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

- ¹¹ <http://docs.aws.amazon.com/streams/latest/dev/kinesis-kpl-writing.html>
- ¹² <http://docs.aws.amazon.com/streams/latest/dev/developing-producers-with-kpl.html>
- ¹³ <http://docs.aws.amazon.com/streams/latest/dev/developing-producers-with-sdk.html>
- ¹⁴ <http://docs.aws.amazon.com/kinesis/latest/APIReference/Welcome.html>
- ¹⁵ <http://docs.aws.amazon.com/streams/latest/dev/developing-consumers-with-kcl.html>
- ¹⁶ <https://aws.amazon.com/lambda/>
- ¹⁷ <http://docs.aws.amazon.com/lambda/latest/dg/with-kinesis.html>
- ¹⁸ <http://docs.aws.amazon.com/kinesis/latest/APIReference/Welcome.html>