EBOOK

Le Grand Livre de l'ingénierie des données

Une collection de blogs techniques, comprenant des échantillons de code et de notebooks

databricks

Table des matières

databricks

SECTION

01

Introduction à l'ingénierie des données sur Databricks

databricks

Les organisations sont conscientes de la valeur des données en tant qu'atout stratégique pour diverses initiatives liées à leurs activités, telles que la croissance du chiffre d'affaires, l'amélioration de l'expérience client, l'efficacité opérationnelle ou l'amélioration d'un produit ou d'un service. Cependant, l'accès et la gestion des données pour ces initiatives sont devenus de plus en plus complexes. La complexité est surtout due à l'explosion des volumes et des types de données, les organisations accumulant environ 80 % des données sous forme non structurée et semi-structurée. Alors que la collecte de données continue d'augmenter, 73 % des données ne sont pas utilisées pour l'analytique ou la prise de décision. Afin d'essayer de réduire ce pourcentage et de rendre plus de données utilisables, les équipes d'ingénierie des données sont chargées de construire des pipelines de données afin de fournir des données de manière efficace et fiable. Mais le processus de construction de ces pipelines de données complexes s'accompagne d'un certain nombre de difficultés :

- Pour introduire des données dans un data lake, les ingénieurs doivent passer un temps considérable à coder manuellement des tâches répétitives d'ingestion de données.
- Comme les plateformes de données changent continuellement, les ingénieurs des données passent du temps à construire et à maintenir, puis à reconstruire, une infrastructure évolutive complexe.
- Avec l'importance croissante des données en temps réel, des pipelines de données à faible latence sont nécessaires, mais ils sont encore plus difficiles à construire et à maintenir.
- Enfin, une fois tous les pipelines écrits, les ingénieurs de données doivent constamment se concentrer sur les performances, en ajustant les pipelines et les architectures pour respecter les SLA.

Comment Databricks peut vous aider ?

Avec la plateforme Lakehouse de Databricks, les experts internes ont accès à une solution d'ingénierie des données de bout en bout pour l'ingestion, la transformation, le traitement, la planification et la livraison des données. La plateforme Lakehouse automatise et simplifie la construction et la maintenance des pipelines ainsi que l'exécution des charges de travail ETL directement sur un data lake. Les ingénieurs des données peuvent ainsi se concentrer sur la qualité et la fiabilité pour obtenir des insights précieux.

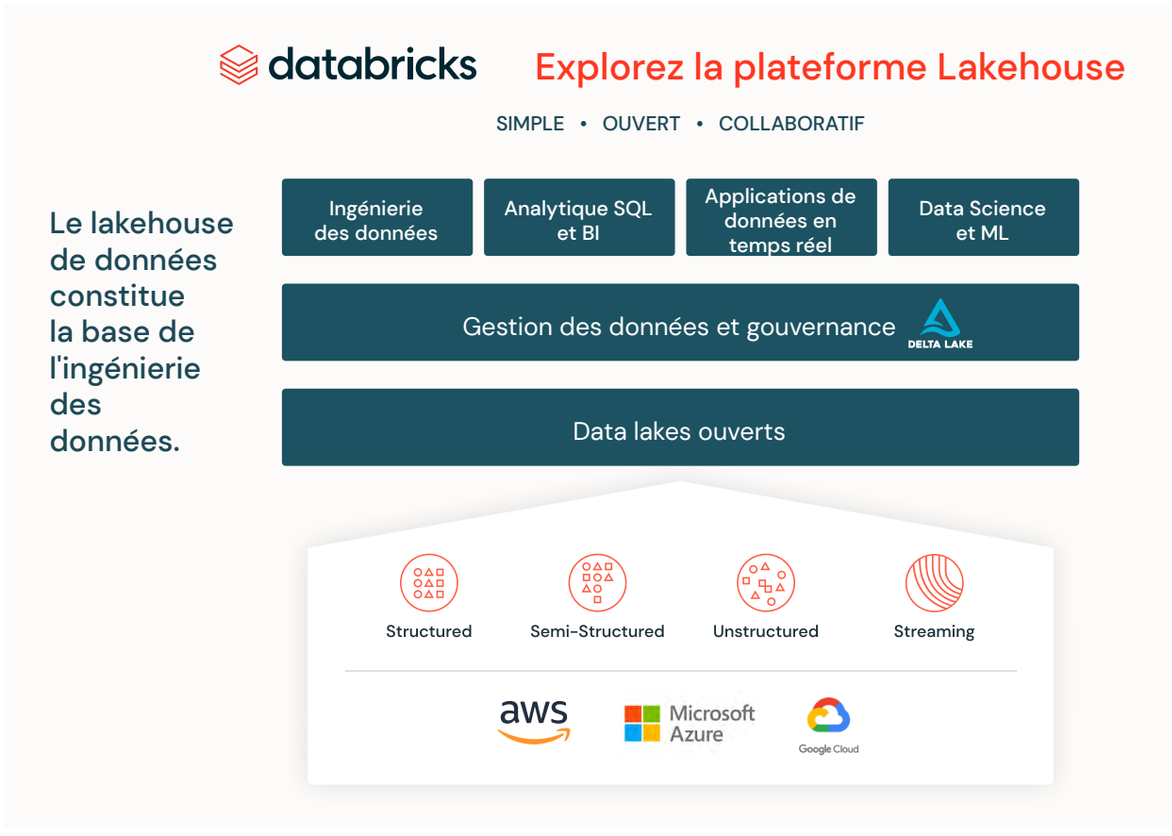

Figure 1
La plateforme Lakehouse de Databricks rassemble vos données, l'analytique et l'IA sur une plateforme commune pour tous vos cas d'usage liés aux données.

Facteurs clés de différenciation pour une ingénierie des données réussie avec Databricks.

Grâce à une architecture lakehouse simplifiée, les ingénieurs des données peuvent adopter une approche de qualité professionnelle et prête à l'emploi pour construire des pipelines de données.
Pour réussir, une équipe d'ingénierie des données orientée solutions doit adopter ces huit facteurs de différenciation :

Ingestion de données continue ou planifiée

Avec la possibilité d'ingérer des pétaoctets de données avec des schémas auto-évolutifs, les ingénieurs en donnée peuvent fournir des données rapides, fiables, évolutives et automatiques pour l'analytique, la data science ou le machine learning. Cela inclut :

- Traitement incrémentiel et efficace des données à mesure qu'elles arrivent de fichiers ou de sources de streaming de type Kafka, SGBD et NoSQL.
- Déduction automatique du schéma et détection des changements de colonnes pour les formats de données structurés et non structurés.
- Suivi automatique et efficace des données au fur et à mesure de leur arrivée, sans intervention manuelle
- Prévention des pertes de données grâce à la sauvegarde des colonnes de données

Pipelines ETL déclaratifs

Les ingénieurs des données peuvent réduire le temps et les efforts de développement afin de se concentrer sur la mise en œuvre de la logique métier et des contrôles de qualité des données au sein du pipeline de données à l'aide de SQL ou de Python. Cela peut être réalisé en :

- Utilisant le développement déclaratif par intention pour définir ce qui est à résoudre et simplifier la manière d'y arriver
- Créant automatiquement un lignage de haute qualité et en gérant les dépendances de tables à travers le pipeline de données.
- Vérifiant automatiquement les dépendances manquantes ou les erreurs de syntaxe, et en gérant la récupération du pipeline de données.

Validation et surveillance de la qualité des données

Améliorer la fiabilité des données dans l'ensemble du data lakehouse afin que les équipes chargées des données puissent se fier aux informations pour les initiatives en aval en :

- Définissant les contrôles d'intégrité et de qualité au sein du pipeline avec des attentes définies en matière de données
- Traitant les erreurs de qualité des données avec des politiques prédéfinies (échec, abandon, alerte, quarantaine).
- Exploitant les métriques de qualité des données qui sont capturées, suivies et rapportées pour l'ensemble du pipeline de données.

Tolérance aux pannes et récupération automatique

Gérer les erreurs transitoires et se rétablir des erreurs les plus courantes survenant pendant le fonctionnement d'un pipeline, avec une récupération automatique, rapide et évolutive qui comprend :

- Des mécanismes de tolérance aux pannes permettant de récupérer systématiquement l'état des données.
- La possibilité de suivre automatiquement la progression depuis la source grâce au checkpointing
- La possibilité de récupérer et de restaurer automatiquement l'état du pipeline de données

Observabilité du pipeline de données

Surveillez l'état global du pipeline de données à partir d'un tableau de bord graphique de flux de données et suivez visuellement la santé du pipeline de bout en bout pour les performances, la qualité et la latence. Les capacités d'observabilité du pipeline de données comprennent :

- Un diagramme de lignage de haute qualité et de haute fidélité qui fournit une visibilité sur la façon dont les données circulent pour l'analyse d'impact.
- Consignation précise des performances et de l'état du pipeline de données au niveau d'une ligne
- Suivi continu des tâches du pipeline de données pour assurer un fonctionnement continu

Traitement des données en batch et en flux

Permettre aux ingénieurs des données d'ajuster la latence avec des contrôles de coûts sans avoir besoin de traitement de flux complexe ou d'implémenter une logique de récupération.

- Exécuter des charges de travail de pipeline de données sur des clusters de calcul élastiques automatiquement provisionnés basés sur Apache Spark™ pour la mise à l'échelle et la performance.
- Utiliser des clusters d'optimisation des performances qui parallélisent les tâches et minimisent les mouvements de données.

Déploiements et opérations automatiques

Assurer une livraison fiable et prévisible des données pour les cas d'usage de l'analytique et du Machine Learning en permettant des déploiements et des retours en arrière du pipeline de données faciles et automatiques pour minimiser les temps d'arrêt. Parmi les avantages, on trouve :

- Déploiement complet, paramétré et automatisé pour la délivrance continue des données
- Orchestration, test et surveillance de bout en bout du déploiement du pipeline de données sur les principaux fournisseurs de cloud

Pipeline et flux de travail programmés

Orchestration simple, claire et fiable des tâches de traitement des données pour les pipelines de données et de Machine Learning, avec la possibilité d'exécuter plusieurs tâches non interactives sous forme de graphe orienté acyclique (Directed Acyclic Graph - DAG) sur un cluster de calcul Databricks.

- Orchestrer facilement des tâches dans un DAG en utilisant l'interface utilisateur et l'API de Databricks.
- Créer et gérer des tâches multiples dans des projets via l'interface utilisateur ou l'API, et des fonctionnalités telles que des alertes par e-mail pour le suivi.
- Orchestrer toute tâche disposant d'une API en dehors de Databricks et dans tous les clouds.

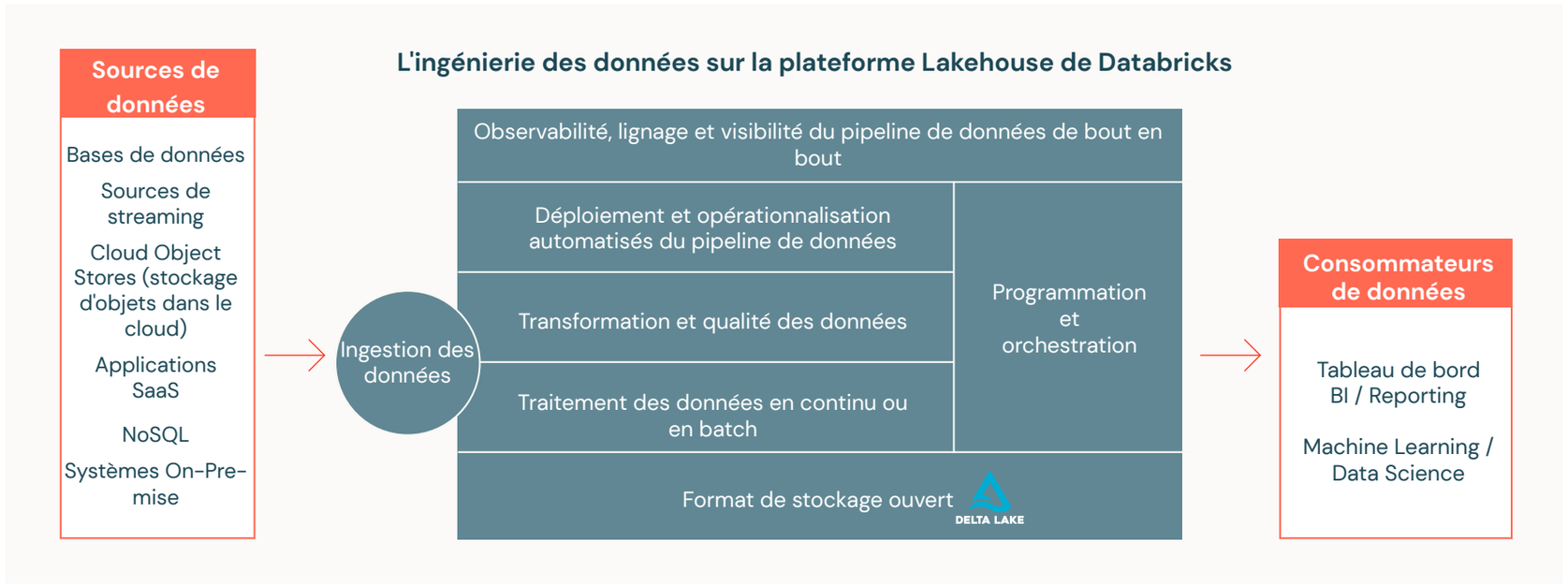

Figure 2
Architecture de référence de l'ingénierie des données sur Databricks

Conclusion

Alors que les organisations s'efforcent de devenir « data-driven », l'ingénierie des données est un point central de leur réussite. Pour fournir des données fiables et dignes de confiance, les ingénieurs des données ne devraient pas avoir à passer autant de temps à développer et à maintenir manuellement un cycle de vie ETL de bout en bout. Les équipes d'ingénierie des données ont besoin d'un moyen efficace et évolutif pour simplifier le développement ETL, améliorer la fiabilité des données et gérer les opérations.

Comme décrit précédemment, les huit facteurs de différenciation simplifient la gestion du cycle de vie de l'ETL en automatisant et en maintenant toutes les dépendances des données, en tirant parti des contrôles de qualité intégrés au monitoring et en fournissant une visibilité approfondie des opérations du pipeline avec la récupération automatique. Les équipes d'ingénierie des données peuvent désormais se concentrer sur la construction facile et rapide de pipelines de données fiables de bout en bout, prêts pour la production, en utilisant uniquement SQL ou Python pour le traitement en batch et le streaming, pour fournir des données de grande valeur pour l'analytique, la Data Science ou le Machine Learning.

Cas d'usage

Dans la section suivante, nous décrivons les bonnes pratiques pour les cas d'usage de bout en bout de l'ingénierie des données, tirés d'exemples concrets. De l'ingestion et du traitement des données à l'analytique et au Machine Learning, vous apprendrez à transformer des données brutes en données exploitables. Nous vous fournirons les ensembles de données et les échantillons de code, afin que vous puissiez mettre la main à la pâte et explorer tous les aspects du cycle de vie des données sur la plateforme Lakehouse de Databricks.

databricks

SECTION

02

Cas d'usage réels sur la plateforme Lakehouse de Databricks

Analytique en temps réel des points de vente avec le lakehouse de données

Construction d'un lakehouse de cybersécurité pour les événements CrowdStrike Falcon

Libérer la puissance des données de santé avec un lakehouse de données moderne

Délai et fiabilité de la transmission des rapports réglementaires

Solutions AML à l'échelle grâce à la plateforme Lakehouse de Databricks

Construire un modèle d'IA en temps réel pour détecter les comportements toxiques dans les contextes de gaming (jeux)

Transformation de Northwestern Mutual (plateforme d'insights) par l'adoption d'une architecture Lakehouse ouverte et évolutive

Comment l'équipe data de Databricks a construit un lakehouse au sein de trois clouds et plus de 50 régions

SECTION 2.1

Analytique en temps réel des points de vente avec le lakehouse de données

de **BRYAN SMITH** et **ROB SAKER**

9 septembre 2021

Les perturbations de la chaîne d'approvisionnement dues à la réduction de l'offre de produits et à la diminution de la capacité des entrepôts, associées à l'évolution rapide des attentes des consommateurs en matière d'expériences omnicanal sans faille, poussent les détaillants à repenser la manière dont ils utilisent les données pour gérer leurs activités. Avant la pandémie, 71 % des détaillants ont désigné le manque de visibilité en temps réel des stocks comme le principal obstacle à la réalisation de leurs objectifs omnicanal. La pandémie n'a fait qu'accroître la demande d'expériences intégrées en ligne et en magasin, exerçant encore plus de pression sur les détaillants afin qu'ils affichent la disponibilité précise des produits et gèrent les changements de commande à la volée. Un meilleur accès aux informations en temps réel est la clé pour répondre aux demandes des consommateurs dans la nouvelle normalité.

Dans ce blog, nous aborderons le besoin de données en temps réel dans le commerce de détail, et la manière de surmonter les défis liés à la transmission en continu et en temps réel des données des points de vente à l'échelle avec un lakehouse de données.

Le système de point de vente

Le système de point de vente (Point-of-Sale - POS) est depuis longtemps la pièce centrale de l'infrastructure en magasin, enregistrant l'échange de biens et de services entre le détaillant et ses clients. Pour mener à bien cet échange, le point de vente suit généralement les stocks de produits et facilite le réapprovisionnement lorsque le nombre d'unités tombe en dessous des niveaux critiques. On ne saurait trop insister sur l'importance du point de vente pour les opérations en magasin. En tant que système d'enregistrement des ventes et des activités d'inventaire, l'accès à ses données est d'un intérêt capital pour les business analysts.

Historiquement, la connectivité limitée entre les magasins individuels et les bureaux de l'entreprise impliquait que le système POS (et pas seulement ses interfaces terminales) se trouve physiquement dans le magasin. Pendant les heures creuses, ces systèmes peuvent transmettre des données récapitulatives qui, une fois consolidées dans un entrepôt de données, fournissent une vision des performances des opérations de vente au détail datant d'un jour. Ces données sont ensuite mises à jour par le cycle de la nuit suivante.

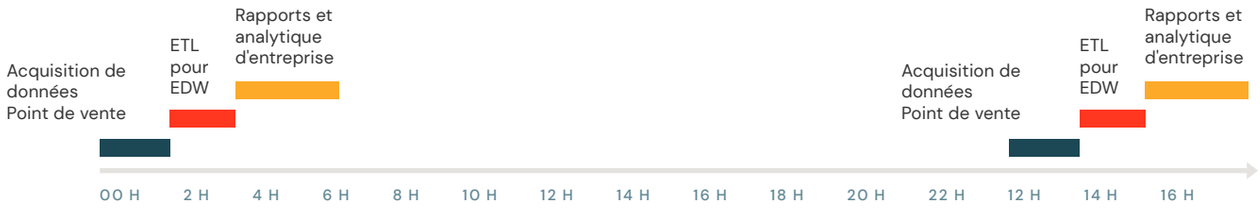

Figure 1
Disponibilité des stocks avec des modèles ETL traditionnels, orientés vers le traitement en batch

Les améliorations de la connectivité moderne ont permis à un plus grand nombre de détaillants de passer à un système de point de vente centralisé, basé dans le cloud, alors que beaucoup développent des intégrations presque en temps réel entre les systèmes en magasin et le back-office de l'entreprise. La disponibilité des informations presque en temps réel signifie que les détaillants peuvent actualiser en permanence leurs estimations de la disponibilité des articles. L'entreprise ne gère plus ses opérations en fonction de sa connaissance de l'état des stocks de la veille. Elle agit en fonction de sa connaissance de l'état des stocks en temps réel.

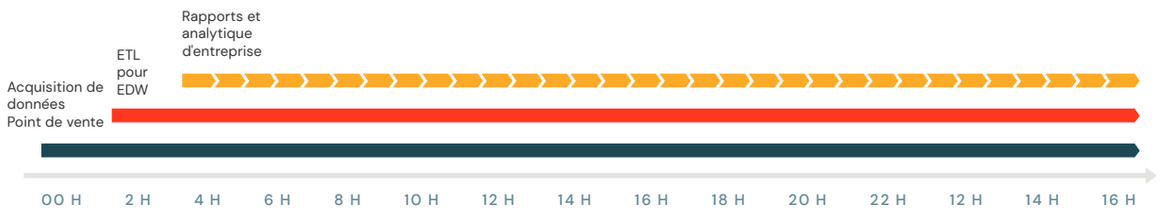

Figure 2
Disponibilité des stocks avec des modèles ETL traditionnels, orientés vers le traitement en batch

Perspectives presque en temps réel

Aussi importantes que soient les informations presque en temps réel sur l'activité des magasins, le passage de processus nocturnes à un flux continu d'informations pose des défis particuliers, non seulement pour l'ingénieur des données, qui doit concevoir un type de flux de traitement des données différent, mais aussi pour le consommateur d'informations. Dans ce billet, nous partageons quelques enseignements tirés de l'expérience de clients qui se sont récemment lancés dans cette aventure, et examinons comment les modèles et les compétences clés disponibles grâce au modèle lakehouse peuvent contribuer à la réussite d'un parcours.

LEÇON 1

Examinez attentivement le périmètre

Souvent, les systèmes POS ne se limitent pas à la gestion des ventes et des stocks. Ils peuvent également fournir une gamme exceptionnelle de fonctionnalités comprenant le traitement des paiements, la gestion des crédits en magasin, la facturation et la passation des commandes, la gestion des programmes de fidélisation, la gestion des plannings, le suivi des horaires et même la paie, ce qui en fait un véritable couteau suisse des fonctionnalités en magasin.

Par conséquent, les données hébergées dans le point de vente sont généralement réparties dans une structure de base de données aussi vaste que complexe. Si vous avez de la chance, la solution POS met à disposition une couche d'accès aux données, qui rend ces données accessibles via des structures plus faciles à interpréter. Mais si ce n'est pas le cas, l'ingénieur des données doit trier ce qui peut constituer un ensemble de tables opaque afin de déterminer ce qui est valable et ce qui ne l'est pas.

Quelle que soit la manière dont les données sont exposées, les recommandations habituelles restent valables : identifiez une justification commerciale convaincante pour votre solution et utilisez-la pour limiter la portée des actifs informationnels que vous consommez initialement. Cette justification émane souvent d'un sponsor « business » solide, qui est chargé de relever un défi commercial spécifique et qui considère que la disponibilité d'informations plus appropriées est essentielle à sa réussite.

Pour illustrer cela, prenons l'exemple d'un défi majeur pour de nombreuses organisations de vente au détail aujourd'hui : la mise en place de solutions omnicanal. Ces solutions, qui permettent d'effectuer des achats en ligne, des retraits en magasin et des transactions entre magasins, dépendent d'informations raisonnablement précises sur les stocks des magasins. Si nous devions limiter notre champ d'action initial à ce seul besoin, nos exigences en matière d'information pour notre système de monitoring et d'analytique seraient considérablement réduites. Une fois qu'une solution d'inventaire en temps réel est fournie et que sa valeur est reconnue par l'entreprise, nous pouvons étendre notre champ d'action pour prendre en compte d'autres besoins, tels que le suivi des promotions et la détection des fraudes, en élargissant l'étendue des actifs d'information exploités à chaque itération.

LEÇON 2

Aligner la transmission avec les modèles de génération de données et les sensibilités temporelles.

Les différents processus génèrent des données de manière différente dans le POS. Les transactions de vente sont susceptibles de laisser une trace de nouveaux enregistrements ajoutés aux tables concernées. Les retours peuvent suivre plusieurs voies, générant des mises à jour d'enregistrements de ventes antérieures, l'insertion de nouveaux enregistrements de ventes inversés et / ou l'insertion de nouvelles informations dans des structures spécifiques aux retours. La documentation du fournisseur, les connaissances « terrain » et même un travail d'investigation indépendant peuvent être nécessaires pour découvrir exactement comment et où les informations spécifiques à un événement se retrouvent dans le POS.

La compréhension de ces modèles peut aider à élaborer une stratégie de transmission des données pour des types d'informations spécifiques. Les modèles à fréquence plus élevée, avec une granularité plus fine et orientés vers l'insertion peuvent être parfaitement adaptés à la diffusion en continu (streaming). Les événements moins fréquents et de plus grande envergure peuvent être mieux adaptés aux styles de transmission de données en vrac. Mais si ces modes de transmission des données représentent les deux extrémités d'un même spectre, il est probable que la plupart des événements capturés par le POS se situent quelque part entre les deux.

L'ingéniosité de l'approche de l'architecture des données par le lakehouse de données se traduit par le fait que de multiples modes de transmission des données peuvent être employés en parallèle. Pour les données qui s'alignent naturellement sur la transmission continue, on peut recourir à la diffusion en streaming. Pour les données qui se prêtent mieux à la transmission de masse, on peut utiliser des processus en batch. Et pour les données qui se situent au centre, vous pouvez vous concentrer sur la rapidité d'exécution des données nécessaires à la prise de décision, et les laisser dicter la voie à adopter. Tous ces modes peuvent être abordés avec une approche cohérente de la mise en œuvre de l'ETL, un défi qui a contrecarré de nombreuses implémentations antérieures de ce qui était fréquemment appelé architectures lambda.

LEÇON 3

Déposer les données par étapes

Les données arrivent des systèmes de points de vente en magasin avec des fréquences, des formats et des prévisions variables pour une disponibilité en temps voulu. En tirant parti du modèle de conception Bronze, Silver & Gold populaire dans les lakehouses, vous pouvez séparer le nettoyage initial, le reformatage et la persistance des données de transformations plus complexes requises pour des livrables spécifiques alignés sur l'activité.

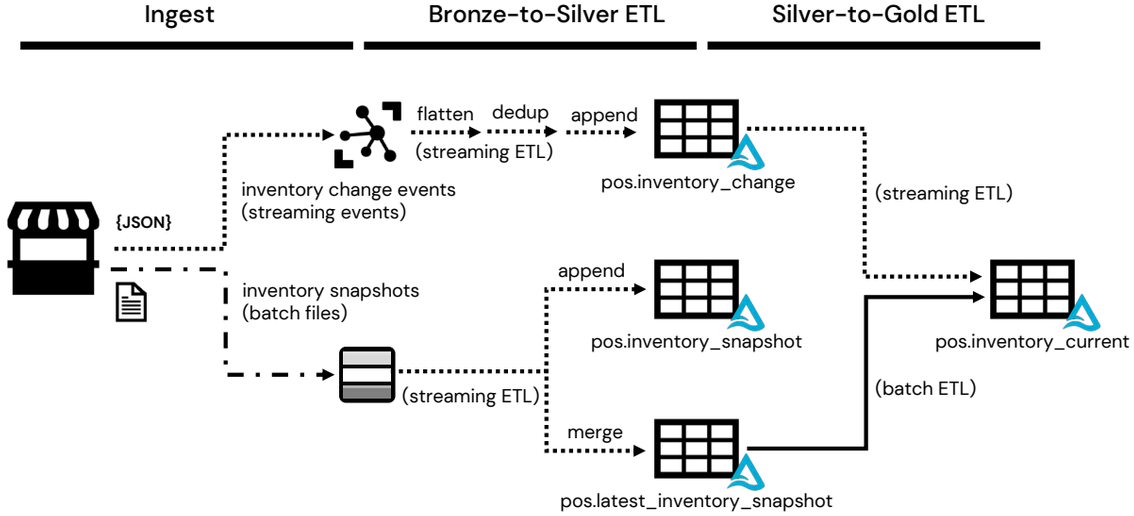

Figure 3
Une architecture de data lakehouse pour le calcul de l'inventaire actuel tirant parti du modèle de persistance des données Bronze, Silver et Gold.

LEÇON 4

Gérer les attentes

Le passage à l'analyse presque en temps réel nécessite un changement organisationnel. Gartner décrit ce phénomène à travers son Modèle de maturité pour l'analytique des données en diffusion continue dans lequel l'analyse des données en continu s'inscrit dans des opérations quotidiennes. Cela ne se fait pas du jour au lendemain.

Au lieu de cela, les ingénieurs en données ont besoin de temps pour reconnaître les défis inhérents à la diffusion en continu des livraisons, en partant des magasins physiques jusqu'à un back-office centralisé et basé dans le cloud. Les améliorations de la connectivité et de la fiabilité des systèmes, associées à des flux de travail ETL de plus en plus robustes, permettent d'obtenir des données plus actualisés, plus fiables et plus cohérentes. Cela implique souvent de renforcer les partenariats avec les ingénieurs systèmes et les développeurs d'applications, afin de soutenir un niveau d'intégration qui n'existait pas à l'époque des flux de travail ETL en batch uniquement.

Les business analysts devront se familiariser avec le bruit inhérent aux données mises à jour en permanence. Ils devront réapprendre à effectuer des tâches de diagnostic et de validation sur un ensemble de données, par exemple lorsqu'une requête exécutée quelques secondes auparavant renvoie un résultat légèrement différent. Ils doivent acquérir une conscience plus profonde des problèmes que posent les données et qui sont souvent cachés lorsqu'ils sont présentés sous forme d'agrégats quotidiens. Tout cela nécessitera des ajustements à la fois dans leur analyse et dans leur réponse aux signaux détectés dans leurs résultats.

Tout cela se passe durant les premiers stades de la maturation. Dans les étapes ultérieures, la capacité de l'organisation à détecter des signaux significatifs dans le flux peut conduire à des capacités de détection et de réponse plus automatisées. C'est ici que les plus hauts niveaux de valeur des flux de données sont débloqués. Mais la surveillance et la gouvernance doivent être mises en place et éprouvées avant que l'entreprise ne confie ses opérations à ces technologies.

Mise en œuvre du streaming POS

Pour illustrer comment l'architecture lakehouse peut être appliquée aux données des points de vente, nous avons développé un flux de travail de démonstration dans lequel nous calculons un inventaire presque en temps réel. Dans ce cas, nous envisageons deux systèmes de point de vente distincts qui transmettent des informations sur les stocks associées aux ventes, aux réapprovisionnements et aux données sur les pertes, ainsi qu'aux transactions d'achat en ligne et de retrait en magasin (initiées dans un système et exécutées dans l'autre) dans le cadre d'un flux de variation des stocks. Les comptages périodiques (instantanés) des unités de produits en rayon sont saisis par le POS et transmis en masse. Ces données sont simulées pour une période d'un mois et lues à une vitesse 10 fois supérieure à la vitesse nominale pour une meilleure visibilité des variations de stock.

Les processus ETL (comme illustré dans la figure 3) représentent un mélange de techniques de streaming et de batch. Une approche en deux étapes avec des données transformées capturées de façon minimale, dans des tables Delta représentant notre couche Silver, sépare notre approche ETL initiale (plus alignée sur la technique) de l'approche (plus alignée métier) requise pour les calculs d'inventaire actuels. La deuxième étape a été mise en œuvre à l'aide de capacités de diffusion structurée traditionnelles, ce que nous pourrions revoir avec la nouvelle fonctionnalité Delta Live Tables au fur et à mesure de sa disponibilité généralisée.

La démonstration fait appel à Azure IOT Hubs et Azure Storage pour l'ingestion des données, mais elle pourrait fonctionner de la même manière sur les clouds AWS et GCP avec des substitutions technologiques appropriées.

Découvrez ces notebooks Databricks gratuits

- **POS 01: Configuration de l'environnement**
- **POS 02: Génération de données**
- **POS 03: Ingestion ETL**
- **POS 04: Inventaire actuel**

SECTION 2.2

Construction d'un lakehouse de cybersécurité pour les événements CrowdStrike Falcon

de AEMRO AMARE, ARUN PAMULAPATI, YONG SHENG HUANG et JASON POHL

20 mai 2021

Les équipes de sécurité ont besoin des données relatives aux points de terminaison pour la détection et la lutte contre les menaces, les enquêtes sur les incidents et la mise en conformité réglementaire. Les volumes de données peuvent atteindre des téraoctets par jour ou des pétaoctets par an. La plupart des organisations ont du mal à collecter, stocker et analyser les journaux des points de terminaison en raison des coûts et de la complexité associés à de tels volumes de données. Mais il peut aussi en être autrement.

Dans cette série d'articles de blog en deux parties, nous verrons comment vous pouvez exploiter des pétaoctets de données sur les points de terminaison avec Databricks pour améliorer votre posture de sécurité grâce à une analytique avancée, et ce, de manière rentable. La première partie (ce blog) couvrira l'architecture de la collecte de données et l'intégration avec un SIEM (Splunk). À la fin de ce blog, avec les notebooks fournis, vous serez en mesure d'utiliser les données pour l'analyse. La deuxième partie traitera des cas d'usage spécifiques, de la manière de créer des modèles de ML, des enrichissements automatisés et de l'analytique. À la fin de la deuxième partie, vous serez en mesure de mettre en œuvre les notebooks pour détecter et étudier les menaces à l'aide des données des points de terminaison.

Nous allons utiliser les journaux Falcon de CrowdStrike comme exemple. Pour accéder aux journaux Falcon, on peut utiliser le Falcon Data Replicator (FDR) pour pousser les données brutes des événements de la plateforme CrowdStrike vers un stockage dans le cloud tel qu'Amazon S3. Ces données peuvent être ingérées, transformées, analysées et stockées à l'aide de la plateforme Lakehouse de Databricks, avec le reste de la télémétrie de sécurité. Les clients peuvent ingérer des données CrowdStrike Falcon, appliquer des détections en temps réel basées sur Python, effectuer des recherches dans les données historiques avec Databricks SQL, et effectuer des requêtes à partir d'outils SIEM comme Splunk avec Databricks Add-on pour Splunk.

Le défi de l'opérationnalisation des données de CrowdStrike

Bien que les données de CrowdStrike Falcon offrent des informations complètes sur l'enregistrement des événements, l'ingestion, le traitement et l'exploitation de volumes importants et complexes de données de cybersécurité, presque en temps réel et de manière rentable, constituent une tâche ardue. Ce sont là quelques-uns des défis les plus connus :

- **Ingestion de données en temps réel à l'échelle :** Il est difficile de garder la trace des fichiers de données brutes traités et non traités, qui sont écrits par FDR sur le stockage dans le cloud presque en temps réel.
- **Transformations complexes :** Le format des données est semi-structuré. Chaque ligne de fichier journal contient des centaines de charges utiles de différents types, et la structure des données d'événements peut changer au fil du temps.
- **Gouvernance des données :** Ce type de données peut s'avérer sensible et l'accès doit être limité aux utilisateurs qui en ont besoin.

- **Analyse simplifiée de la sécurité de bout en bout :** Des outils évolutifs sont nécessaires pour effectuer l'ingénierie des données, la modélisation mathématique et l'analytique de ces ensembles de données volumineux et en évolution rapide.
- **Collaboration :**Une collaboration efficace permet de tirer parti de l'expertise des ingénieurs des données, des analystes en cybersécurité et des ingénieurs en ML. Ainsi, le fait de disposer d'une plateforme collaborative améliore l'efficacité des charges de travail d'analyse et de réponse en matière de cybersécurité.

En conséquence, les ingénieurs en sécurité des entreprises se retrouvent dans une situation difficile, luttant pour gérer les coûts et l'efficacité opérationnelle. Ils doivent soit accepter d'être enfermés dans des systèmes propriétaires très coûteux, soit déployer des efforts considérables pour créer leurs propres outils de sécurité des points de terminaison, tout en luttant pour l'évolutivité et les performances.

Lakehouse de cybersécurité Databricks

Databricks offre aux équipes de sécurité et aux data scientists un nouvel espoir pour accomplir leur travail de manière efficace, ainsi qu'un ensemble d'outils pour lutter contre les défis croissants du big data et des menaces sophistiquées.

Lakehouse, une architecture ouverte combinant les meilleurs éléments des data lakes et des data warehouses, simplifie la construction d'un pipeline d'ingénierie de données à sauts multiples qui ajoute progressivement de la structure aux données. L'avantage d'une architecture multi-sauts est que les ingénieurs des données peuvent construire un pipeline qui commence par les données brutes comme « source unique de vérité » à partir de laquelle tout découle. Les données brutes semi-structurées de CrowdStrike peuvent être stockées pendant des années, et les transformations et agrégations ultérieures peuvent être effectuées en flux continu de bout en bout pour affiner les données, et introduire une structure spécifique au contexte afin d'analyser et de détecter les risques de sécurité à travers différents scénarios.

- **Ingestion des données :** Auto Loader (AWS | Azure | GCP) aide à lire immédiatement les données dès qu'un nouveau fichier est écrit par CrowdStrike FDR dans le stockage des données brutes. Il exploite les services de notification du cloud pour traiter de manière incrémentielle les nouveaux fichiers à mesure qu'ils arrivent dans le cloud. Auto Loader configure aussi automatiquement et écoute le service de notification des nouveaux fichiers et peut s'adapter à des millions de fichiers par seconde.
- **Traitement unifié des flux et des batches :** Delta Lake est une approche ouverte pour apporter la gestion et la gouvernance des données aux data lakes. Elle exploite la puissance de calcul distribuée d'Apache Spark™ pour d'énormes volumes de données et de métadonnées. Le moteur Delta de Databricks est un moteur hautement optimisé qui peut traiter des millions d'enregistrements par seconde.
- **Gouvernance des données :** Avec le contrôle d'accès aux tables Databricks (AWS | Azure | GCP), les administrateurs peuvent accorder différents niveaux d'accès aux tables Delta selon la fonction métier d'un utilisateur.

- **Outils d'analyse de la sécurité :** Databricks SQL aide à créer un tableau de bord interactif avec alerte automatique lorsque des modèles inhabituels sont détectés. De même, il peut facilement s'intégrer à des outils de BI largement adoptés tels que Tableau, Microsoft Power BI et Looker.
- **Collaboration sur les notebooks de Databricks :** Les notebooks collaboratifs de Databricks permettent aux équipes de sécurité de collaborer en temps réel. Plusieurs utilisateurs peuvent exécuter des requêtes dans plusieurs langues, partager des visualisations et apporter des commentaires au sein d'un même espace de travail afin de faire avancer les enquêtes sans interruption.

Architecture Lakehouse pour les données CrowdStrike Falcon.

Nous recommandons l'architecture Lakehouse suivante pour les charges de travail de cybersécurité, telles que les données CrowdStrike Falcon. Auto Loader et Delta Lake simplifient le processus de lecture des données brutes à partir du stockage dans le cloud et l'écriture dans une table Delta à faible coût, et avec un travail DevOps minimal.

Dans cette architecture, les données semi-structurées de CrowdStrike sont chargées sur le stockage dans le cloud du client au niveau de la zone d'atterrissage. Ensuite, Auto Loader utilise des services de notification dans le cloud pour déclencher automatiquement le traitement et l'ingestion de nouveaux fichiers dans les tables Bronze du client, qui serviront de source unique de vérité pour tous les travaux en aval. L'Auto Loader suit les fichiers traités et non traités à l'aide de points de contrôle afin d'éviter le traitement de données en double.

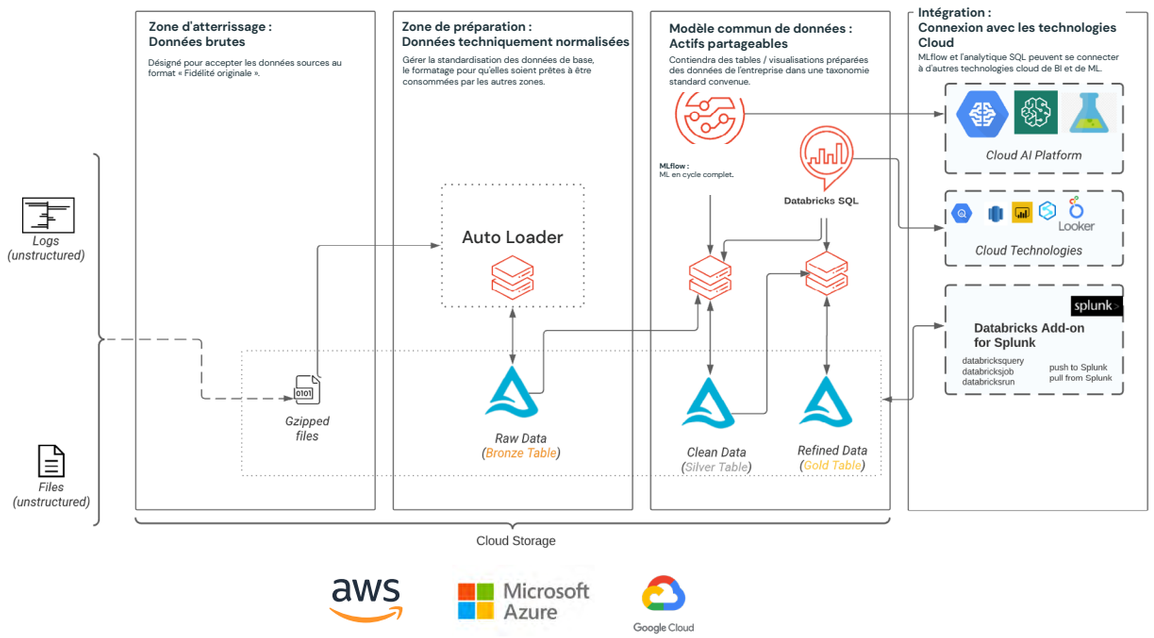

Figure 1
Architecture Lakehouse pour les données CrowdStrike Falcon.

Au fil de la transition du stade Bronze au stade Silver, des schémas seront ajoutés pour structurer les données. Comme la lecture se fait à partir d'une source unique de vérité, nous sommes en mesure de traiter l'ensemble des différents types d'événements et d'appliquer le schéma correct lorsqu'ils sont écrits dans leurs tables respectives. La possibilité de faire respecter les schémas au niveau de la couche Silver fournit une base solide pour la construction de charges de travail analytiques et de ML.

L'étape Gold, qui permet d'agréger les données pour accélérer les requêtes et les performances dans les tableaux de bord et les outils de BI, est facultative, en fonction du cas d'usage et des volumes de données. Des alertes peuvent être définies pour se déclencher lorsque des tendances inattendues sont observées.

Le Databricks Add-on for Splunk est une autre fonctionnalité optionnelle,qui permet aux équipes de sécurité de profiter du modèle économique de Databricks et de la puissance de l'IA sans avoir à se priver du confort de Splunk. Les clients peuvent exécuter des requêtes ad hoc contre les bases Databricks à partir d'un tableau de bord Splunk ou d'une barre de recherche avec l'add-on. Les utilisateurs peuvent également lancer des notebooks ou des tâches dans Databricks via un tableau de bord Splunk ou en réponse à une recherche Splunk. L'intégration de Databricks est bidirectionnelle, permettant ainsi aux clients de résumer les données bruitées ou d'exécuter des détections dans Databricks qui s'affichent dans Splunk Enterprise Security. Les clients peuvent même exécuter des recherches Splunk à partir d'un notebook Databricks afin d'éviter d'avoir à dupliquer les données.

L'intégration de Splunk et de Databricks permet aux clients de réduire les coûts, d'étendre les sources de données qu'ils analysent et de fournir les résultats d'un moteur d'analytique plus robuste, le tout sans modifier les outils utilisés par leur personnel au quotidien.

Guide du code

Étant donné qu'Auto Loader extrait la partie la plus complexe de l'ingestion de données basée sur des fichiers, le pipeline d'ingestion brut vers Bronze peut être créé en quelques lignes de code. Vous trouverez ci-dessous un exemple de code Scala pour un pipeline d'ingestion Delta. Les enregistrements d'événements CrowdStrike Falcon possèdent un nom de champ commun : "event_simpleName".

```
val crowdstrikeStream = spark.readStream
  .format(“cloudFiles”)
  .option(“cloudFiles.format”, “text”)    // text file doesn’t need schema
  .option(“cloudFiles.region”, “us-west-2”)
  .option(“cloudFiles.useNotifications”, “true”)
  .load(rawDataSource)
  .withColumn(“load_timestamp”, current_timestamp())
  .withColumn(“load_date”, to_date($”load_timestamp”))
  .withColumn(“eventType”, from_json($”value”, “struct”, Map.empty[String, String]))
.selectExpr(“eventType.event_simpleName”,”load_date”,”load_timestamp”, “value” )
  .writeStream
  .format(“delta”)
  .option(“checkpointLocation”, checkPointLocation)
  .table(“demo_bronze.crowdstrike”)
```

Dans la couche « brut à Bronze », seul le nom de l'événement est extrait des données brutes. En ajoutant un horodatage de chargement et des colonnes de date, les utilisateurs stockent les données brutes dans la table Bronze. La table Bronze est partitionnée par nom d'événement et par date de chargement, ce qui contribue à rendre les tâches « Bronze vers Silver » plus performantes, surtout lorsqu'il y a un intérêt pour un nombre limité de plages de dates d'événements. Ensuite, une tâche de streaming « Bronze-vers-Silver » lit les événements à partir

d'une table Bronze, applique un schéma et écrit dans des centaines de tables d'événements en fonction du nom de l'événement. Voici un exemple de code Scala :

```
spark
  .readStream
  .option("ignoreChanges", "true")
  .option("maxBytesPerTrigger", "2g")
  .option("maxFilesPerTrigger", "64")
  .format("delta")
  .load(bronzeTableLocation)
  .filter($"event_simpleName" === "event_name")
  .withColumn("event", from_json($"value", schema_of_json(sampleJson)) )
  .select($"event.*", $"load_timestamp", $"load_date")
  .withColumn("silver_timestamp", current_timestamp())
  .writeStream
  .format("delta")
  .outputMode("append")
  .option("mergeSchema", "true")
  .option("checkpointLocation", checkPoint)
  .option("path", tableLocation)
  .start()
```

Chaque schéma d'événement peut être stocké dans un registre de schémas ou dans une table Delta, dans le cas où un schéma doit être partagé entre plusieurs services pilotés par les données. Notez que le code ci-dessus utilise un exemple de chaîne JSON lue à partir de la table Bronze, et que le schéma est déduit du JSON en utilisant schema_of_json(). Plus tard, la chaîne JSON est convertie en une structure à l'aide de à partir de (from)_json(). Ensuite, la structure est aplatie, ce qui incite à l'ajout d'une colonne d'horodatage. Ces étapes fournissent un DataFrame avec toutes les colonnes requises pour être ajouté à une table d'événements. Enfin, nous écrivons ces données structurées dans un tableau d'événements en mode Append.

Il est également possible de répartir les événements sur plusieurs tables sur un seul flux avec foreachBatch en définissant une fonction qui traitera les microbatches. En utilisant foreachBatch(), il est possible de réutiliser les sources de données par batch existant pour filtrer et écrire dans plusieurs tables. Cependant, foreachBatch() ne fournit que des garanties d'écriture au moins une fois. Ainsi, une implémentation manuelle est nécessaire pour appliquer la sémantique exactement une fois.

À ce stade, les données structurées peuvent être interrogées avec l'un des langages pris en charge dans les notebooks et les tâches de Databricks : Python, R, Scala et SQL. Les données de la couche Silver sont pratiques à utiliser pour le ML et l'analyse des cyberattaques.

Le prochain pipeline de streaming serait alors « Silver vers Gold ». Lors de cette étape, il est possible d'agréger les données pour le tableau de bord et le processus d'alerte. Dans la deuxième partie de cette série d'articles de blogs, nous fournirons plus d'informations sur la façon dont nous créons des tableaux de bord à l'aide de Databricks SQL.

Et après ?

Suivez les autres articles de blog qui ajoutent encore plus de valeur sur ce cas d'usage en appliquant le ML et en utilisant Databricks SQL.

Vous pouvez utiliser ces notebooks dans votre propre déploiement Databricks. Chaque section des notebooks comporte des commentaires. Nous vous invitons à nous envoyer un e-mail à cybersecurity@databricks.com. Nous attendons avec impatience vos questions et suggestions pour rendre ce notebook plus facile à comprendre et à déployer.

Découvrez ces notebooks Databricks gratuits.

SECTION 2.3

Libérer le pouvoir des données de santé avec un lakehouse de données moderne

par MICHAEL ORTEGA, MICHAEL SANKY et AMIR KERMANY

19 mai 2021

Comment relever les défis des data warehouses et des data lakes dans les secteurs de la santé et des sciences de la vie

Un seul patient génère environ 80 mégaoctets de données médicales chaque année. Multipliez cela par des milliers de patients au cours de leur vie et vous obtenez des pétaoctets de données contenant des informations précieuses. Libérer ces insights peut aider à rationaliser les opérations cliniques, à accélérer la R&D sur les médicaments et à améliorer les résultats pour la santé des patients. Mais d'abord, les données doivent être préparées pour l'analytique en aval et l'IA. Malheureusement, la plupart des organisations de soins de santé et des sciences de la vie passent un temps démesuré à simplement rassembler, nettoyer et structurer leurs données.

Un seul patient génère environ 80 mégaoctets de données médicales chaque année

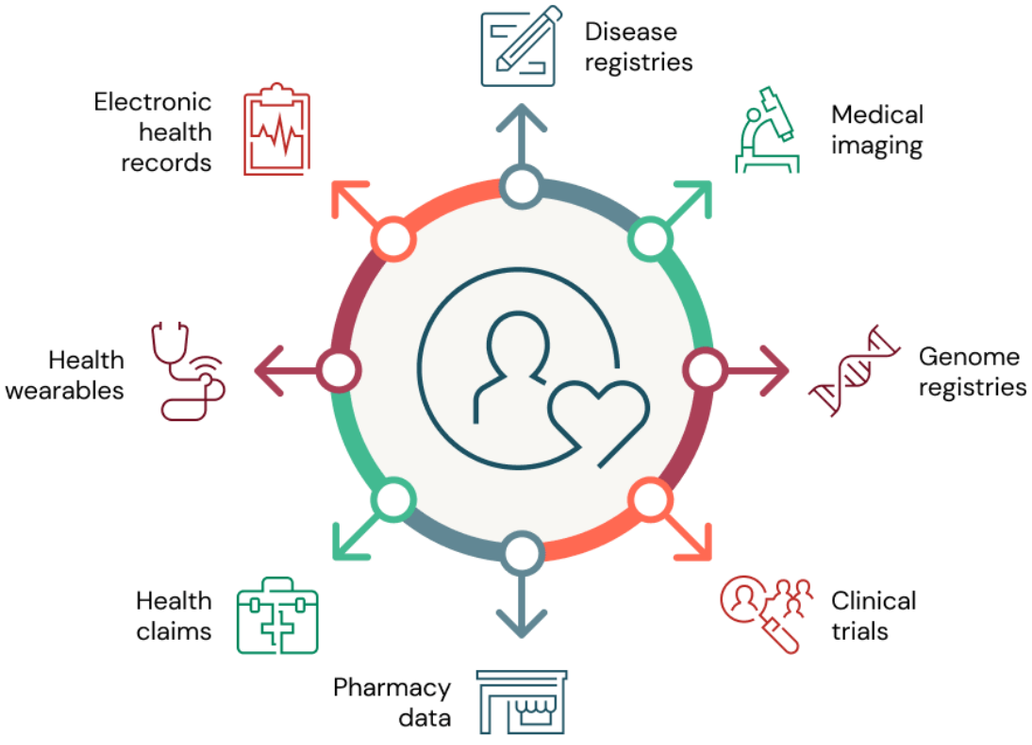

Figure 1
Les données de santé connaissent une croissance exponentielle, un seul patient générant plus de 80 mégaoctets de données par an

Les défis de l'analytique de données dans les soins de santé et les sciences de la vie

Il existe de nombreuses raisons pour lesquelles la préparation des données, l'analytique et l'IA représentent des défis pour les organisations du secteur de la santé. Mais nombre d'entre elles sont liées aux investissements dans des architectures de données héritées, construites sur des data warehouses. Voici les quatre défis les plus courants que nous rencontrons dans l'industrie :

DÉFI NUMÉRO 1 : VOLUME

Mise à l'échelle pour des données de santé en croissance rapide

La génomique est peut-être le meilleur exemple de la croissance explosive du volume de données dans le domaine de la santé. Le séquençage du premier génome a coûté plus d'un milliard de dollars. Compte tenu des coûts prohibitifs, les premiers efforts (et davantage encore) se sont concentrés sur le génotypage, un processus de recherche des variantes spécifiques dans une très petite fraction du génome d'une personne, généralement autour de 0,1 %. Cette technique a évolué vers le séquençage de l'exome entier, qui couvre les parties du génome codant pour les protéines, soit encore moins de 2 % de l'ensemble du génome. Les entreprises proposent désormais des tests directs aux consommateurs pour le séquençage du génome entier (WGS) qui coûtent moins de 300 dollars pour un séquençage 30x. Au niveau de la population, la biobanque britannique (UK Biobank) met à disposition plus de 200 000 génomes entiers pour la recherche cette année. Il n'y a pas que la génomique. L'imagerie, les appareils de santé connectés et les dossiers médicaux électroniques connaissent également une croissance fulgurante.

« L'échelle » est le facteur clé de succès pour des initiatives telles que l'analytique de santé de la population et la découverte de médicaments. Malheureusement, de nombreuses architectures traditionnelles sont construites on-premise et conçues pour une capacité maximale. Cette approche se traduit par une puissance de calcul inutilisée (et finalement de l'argent gaspillé) pendant les périodes de faible utilisation, et elle n'évolue pas rapidement lorsque des mises à niveau sont nécessaires.

DÉFI NUMÉRO 2 : VARIÉTÉ

Analyse de diverses données de santé

Les organisations de soins de santé et des sciences de la vie traitent une quantité énorme de données très variées, chacune ayant ses propres nuances. Il est largement admis que plus de 80 % des données médicales ne sont pas structurées, mais la plupart des organisations concentrent toujours leur attention sur les data warehouses conçus pour les données structurées et l'analytique traditionnelle basée sur SQL. Les données non structurées comprennent les données d'image, qui sont essentielles pour diagnostiquer et mesurer la progression de la maladie dans des domaines comme l'oncologie, l'immunologie et la neurologie (les domaines de coûts qui connaissent la croissance la plus rapide), et le texte narratif dans les notes cliniques, essentielles pour comprendre l'historique sanitaire et social complet du patient. Ignorer ces types de données, ou les mettre de côté, n'est pas une option.

Pour compliquer davantage les choses, l'écosystème de la santé devient de plus en plus interconnecté, obligeant les parties prenantes à gérer de nouveaux types de données. Par exemple, les prestataires ont besoin de données sur les réclamations pour gérer et statuer sur les accords de partage des risques Et les payeurs ont besoin de données cliniques pour soutenir des processus tels que les autorisations préalables, et pour piloter les mesures de qualité. Ces organisations manquent souvent d'architectures et de plateformes de données pour prendre en charge ces nouveaux types de données.

Certaines organisations ont investi dans des data lakes pour prendre en charge les données non structurées et l'analytique avancée, mais cela crée un nouvel ensemble de problèmes. Dans cet environnement, les équipes de données doivent désormais gérer deux systèmes, les data warehouses et les data lakes, où les données sont copiées entre des outils en silos, ce qui entraîne des problèmes de qualité et de gestion des données.

DÉFI NUMÉRO 3 : LA VITESSE

Ingérez des données transmises en continu pour des insights sur les patients en temps réel

Dans de nombreux contextes, les soins de santé sont une question de vie ou de mort. Les conditions peuvent s'avérer très dynamiques et le traitement des données en batch, même quotidien, reste souvent insuffisant. L'accès à des informations actualisées à la seconde près est essentiel à la réussite des soins interventionnels. Pour sauver des vies, les données transmises en continu sont utilisées par les hôpitaux et les systèmes de santé nationaux pour tout, de la prédiction de la septicémie à la mise en œuvre de prévisions de la demande en temps réel de lits de soins intensifs.

De plus, la vitesse des données est un élément majeur de la révolution digitale de la santé. Les individus ont accès à plus d'informations que jamais et peuvent adapter leurs soins en temps réel. Par exemple, les appareils portables, comme les glucomètres en continu fournis par Livongo, diffusent des données en temps réel dans des applications mobiles qui proposent des recommandations comportementales personnalisées.

Malgré certains de ces premiers succès, la plupart des organisations n'ont pas conçu leur architecture de données pour s'adapter à la vitesse de diffusion des données. Les problèmes de fiabilité et les défis liés à l'intégration de données en temps réel avec des données historiques freinent l'innovation.

DÉFI NUMÉRO 4 : VÉRACITÉ

Bâtir la confiance dans les données de santé et l'IA

Enfin, et surtout, les normes cliniques et réglementaires exigent le plus haut niveau de précision des données dans le domaine des soins de santé. Les organisations de soins de santé ont des exigences élevées de conformité en matière de santé publique, qui doivent être respectées. La démocratisation des données au sein des organisations nécessite une gouvernance.

De plus, les organisations ont besoin d'une bonne gouvernance des modèles lorsqu'elles intègrent l'intelligence artificielle (IA) et l'apprentissage automatique / machine learning (ML) dans un environnement clinique. Malheureusement, la plupart des organisations ont des plateformes distinctes pour les workflows de data science qui sont déconnectés de leur data warehouse. Cela crée de sérieux défis lorsque l'on essaie de renforcer la confiance et la reproductibilité dans les applications alimentées par l'IA.

Libérer les données de santé avec un lakehouse

L'architecture Lakehouse aide les organisations de soins de santé et des sciences de la vie à surmonter ces défis, avec une architecture de données moderne qui combine le faible coût, l'évolutivité et la flexibilité d'un data lake dans le cloud avec les performances et la gouvernance d'un data warehouse. Grâce à un lakehouse, les entreprises peuvent stocker tous les types de données et alimenter tous les types d'analytique et de ML dans un environnement ouvert.

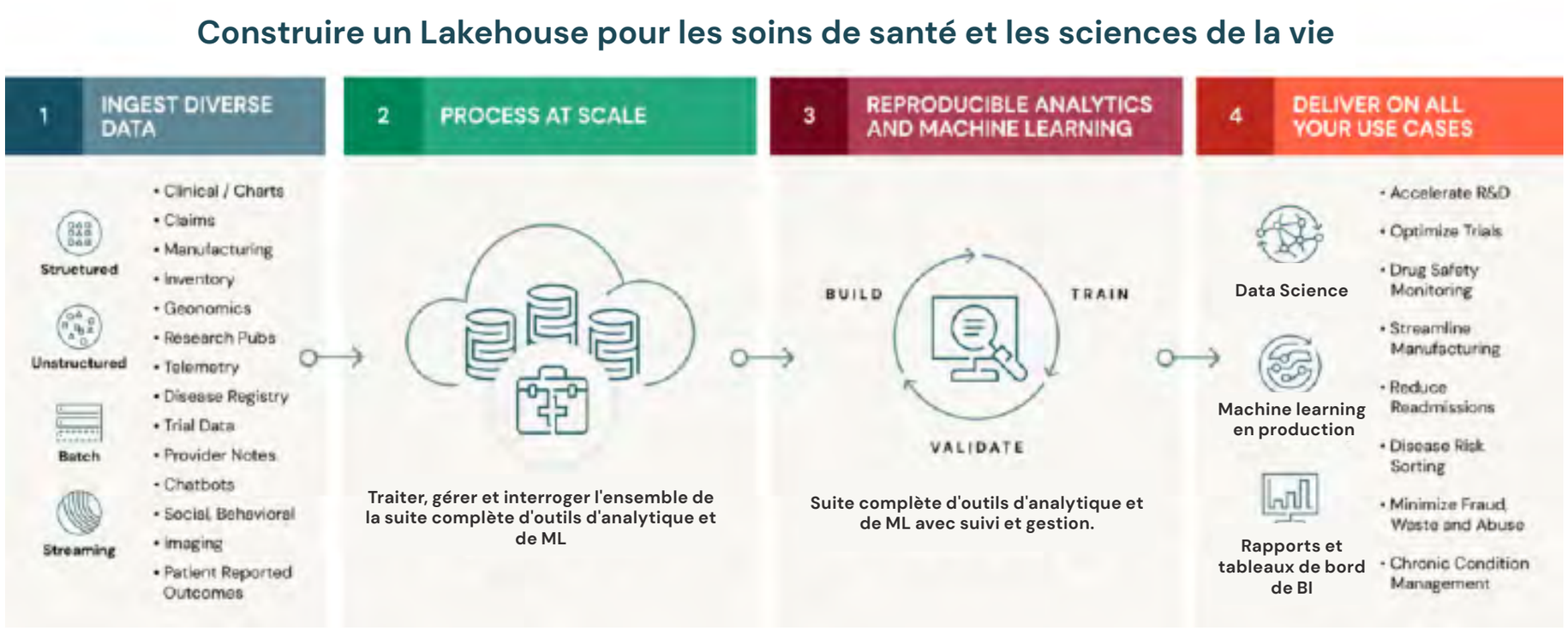

Figure 2

Répondez à tous vos cas d'utilisation d'analytique de données dans le domaine de la santé et des sciences de la vie avec une architecture Lakehouse moderne

Plus précisément, le lakehouse offre les avantages suivants aux organisations de soins de santé et des sciences de la vie :

- **Organisez toutes vos données de santé à grande échelle.** Au cœur de la plateforme Databricks Lakehouse se trouve Delta Lake, une couche de gestion de données open source qui offre fiabilité et performances à votre data lake. Contrairement à un data warehouse traditionnel, Delta Lake prend en charge tous les types de données structurées et non structurées. Et pour faciliter l'ingestion de données de santé, Databricks a construit des connecteurs pour des types de données spécifiques à un domaine comme les dossiers médicaux électroniques et la génomique. Ces connecteurs sont fournis avec des modèles de données standard de l'industrie dans un ensemble d'accélérateurs de solution à démarrage rapide. De plus, Delta Lake fournit des optimisations intégrées pour la mise en cache des données et l'indexation afin d'accélérer considérablement les vitesses de traitement des données. Grâce à ces fonctionnalités, les équipes peuvent déposer toutes leurs données brutes au même endroit, puis les conserver pour créer une vue holistique de la santé des patients.
- **Alimentez toutes l'analytique de vos patients et votre IA.** Avec toutes vos données centralisées dans un lakehouse, vos équipes peuvent créer une puissante analytique patients et des modèles prédictifs directement sur les données. Pour tirer parti de ces capacités, Databricks fournit des espaces de travail collaboratifs avec une suite complète d'outils d'analytique et d'IA ainsi que la prise en charge d'un large éventail de langages de programmation tels que SQL, R, Python et Scala. Cela permet à un groupe diversifié d'utilisateurs, tels que les data scientists, ingénieurs et informaticiens cliniques, de travailler ensemble pour analyser, modéliser et visualiser toutes vos données de santé.

- **Fournissez des insights patient en temps réel.** Le lakehouse fournit une architecture unifiée pour la diffusion en continu et les données en batch. Nul besoin de prendre en charge deux architectures différentes, ni de lutter contre des problèmes de fiabilité. De plus, en exécutant l'architecture Lakehouse sur Databricks, les organisations ont accès à une plateforme native du cloud qui s'adapte automatiquement en fonction de la charge de travail. Cela facilite l'ingestion de données en continu et leur fusion avec des pétaoctets de données historiques pour des informations presque en temps réel à l'échelle de la population.
- **Assurez la qualité et la conformité des données.** Pour garantir la véracité des données, le lakehouse inclut des fonctionnalités manquantes dans les data lakes traditionnels, telles que l'application des schémas, l'audit, la gestion des versions et les contrôles d'accès à fine granularité. Un avantage important de Lakehouse est la possibilité d'effectuer à la fois de l'analytique et du ML sur cette même source de données fiable. De plus, Databricks offre des fonctions de suivi et de gestion des modèles de ML pour permettre aux équipes de reproduire facilement les résultats dans tous les environnements et les aider à respecter les normes de conformité. Toutes ces fonctionnalités sont fournies dans un environnement analytique conforme à la loi HIPAA.

Ce lakehouse constitue la meilleure architecture pour la gestion des données de santé et des sciences de la vie. En associant cette architecture aux capacités de Databricks, les organisations peuvent prendre en charge un large éventail de cas d'usage à fort impact, de la découverte de médicaments aux programmes de gestion des maladies chroniques.

Construire un lakehouse pour les soins de santé et les sciences de la vie

Comme mentionné ci-dessus, nous sommes heureux de mettre à disposition une série d'accélérateurs de solution pour aider les organisations de soins de santé et des sciences de la vie, à commencer à construire un lakehouse adapté à leurs besoins spécifiques. Nos accélérateurs de solutions incluent des exemples de données, du code prédéfini et des instructions étape par étape dans un notebook Databricks.

Nouvel accélérateur de solutions : Lakehouse pour des preuves du monde réel. Les données du monde réel fournissent aux sociétés pharmaceutiques de nouvelles informations sur la santé des patients et l'efficacité des médicaments en dehors d'un essai. Cet accélérateur vous aide à créer un lakehouse pour des preuves du monde réel sur Databricks. Nous vous montrerons comment ingérer des échantillons de données EHR pour une population de patients, structurer les données à l'aide du modèle commun de données OMOP, puis exécuter des analyses à grande échelle pour relever des défis tels que l'étude des modèles de prescription de médicaments.

Découvrez ces notebooks Databricks gratuits.

En savoir plus sur toutes nos solutions de soins de santé et sciences de la vie.

databricks

SECTION 2.4

Délai et fiabilité de la transmission des rapports réglementaires

par **ANTOINE AMEND** et **FAHMID KABIR**

17 septembre 2021

La gestion des risques et de la conformité réglementaire est une entreprise de plus en plus complexe et coûteuse. Les changements réglementaires ont augmenté de 500 % depuis la crise financière mondiale de 2008 et ont augmenté les coûts réglementaires dans le même temps. Compte tenu des amendes associées à la non-conformité et aux violations des SLA (les banques ont atteint un niveau record d'amendes de 10 milliards de dollars en 2019 pour la lutte contre le blanchiment d'argent), le traitement des rapports doit se poursuivre même si les données sont incomplètes. D'autre part, un historique de mauvaise qualité des données est également sanctionné en raison de "contrôles insuffisants". Par conséquent, de nombreuses institutions de services financiers doivent souvent se battre entre une qualité de données médiocre et des accords de niveau de service stricts, et trouver un équilibre entre la fiabilité et la rapidité d'exécution.

Dans cet accélérateur de solution de reporting réglementaire, nous expliquonss comment Delta Live Tables peut garantir l'acquisition et le traitement des données réglementaires en temps réel afin de respecter les SLA réglementaires. Avec Delta Sharing et Delta Live Tables combinés, les analystes ont confiance en la qualité des données réglementaires transmises en temps réel. Dans cet article de blog, nous démontrons les avantages de l'architecture Lakehouse pour combiner les modèles de données de l'industrie des services financiers avec la flexibilité du cloud computing, pour permettre des normes de gouvernance élevées avec une faible surcharge de développement. Nous allons maintenant détailler ce qu'est un modèle de données FIRE et comment les Delta Live Tables peuvent être intégrées pour créer des pipelines de données robustes.

Modèle de données FIRE

La norme de données de réglementation financière (FIRE) définit une spécification commune pour la transmission de données précises entre les systèmes de réglementation de la finance. Les données réglementaires font référence aux données qui sous-tendent les soumissions, les exigences et les calculs réglementaires et sont utilisées à des fins de politique, de suivi et de supervision. La norme de données FIRE est soutenue par la Commission européenne, l'Open Data Institute et l'incubateur de données ouvertes FIRE data standard pour l'Europe via le programme de financement Horizon 2020. Dans le cadre de cette solution, nous avons fourni un module PySpark qui peut interpréter les modèles de données FIRE dans les pipelines d'exploitation Apache Spark™.

Delta Live tables

Databricks a récemment annoncé un nouveau produit pour l'orchestration des pipelines de données, Delta Live Tables (DLT), qui facilite la création et la gestion de pipelines de données fiables à l'échelle de l'entreprise. Avec la possibilité d'évaluer des attentes multiples, de rejeter ou de surveiller les enregistrements invalides en temps réel, les avantages de l'intégration du modèle de données FIRE sur Delta Live Tables sont évidents. Comme l'illustre l'architecture suivante, Delta Live Tables va **ingérer** les données réglementaires granulaires transférées vers le stockage dans le cloud, **schématiser** le contenu et **valider** les enregistrements pour en assurer la cohérence conformément à la spécification des données FIRE. Continuez la lecture de ce document pour tout savoir sur l'utilisation de Delta Sharing afin d'échanger des informations granulaires entre les systèmes de réglementation de manière sûre, évolutive et transparente.

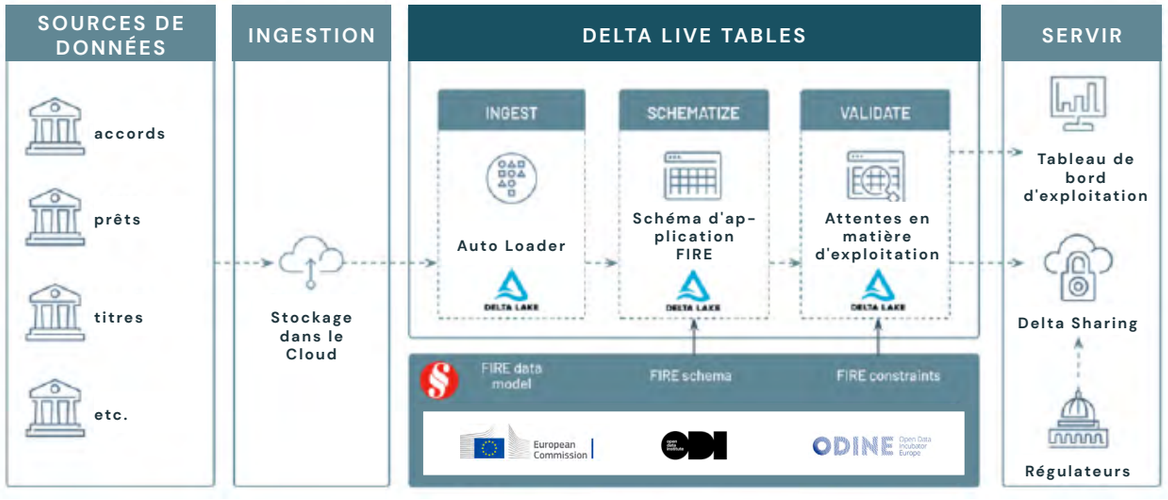

Figure 1

Schéma d'application

Même si certains formats de données peuvent « paraître » structurés (par exemple, les fichiers JSON), l'application d'un schéma n'est pas seulement une bonne pratique d'ingénierie. En entreprise, et en particulier dans l'espace de conformité réglementaire, l'application des schémas garantit que tout champ manquant est attendu, que les champs inattendus sont supprimés et que les types de données sont entièrement évalués (par exemple, une date doit être traitée comme un objet date et non comme une chaîne). Il teste également vos systèmes pour détecter d'éventuelles dérives de données. À l'aide du module FIRE PySpark, nous récupérons par programme le schéma Spark requis pour traiter une entité FIRE donnée (entité collatérale dans cet exemple) que nous appliquons sur un flux d'enregistrements bruts.

```
from fire.spark import FireModel
fire_model = FireModel().load("collateral")
fire_schema = fire_model.schema
```

Dans l'exemple ci-dessous, nous appliquons le schéma aux fichiers CSV entrants. En signalant ce processus à l'aide de l'annotation @dlt, nous définissons notre point d'entrée vers notre Delta Live Table, en lisant des fichiers CSV bruts depuis un répertoire monté et en écrivant des enregistrements schématisés dans une couche Bronze.

```
@dlt.create_table()
def collateral_bronze():
  return (
    spark
      .readStream
      .option("maxFilesPerTrigger", "1")
      .option("badRecordsPath", "/path/to/invalid/collateral")
      .format("csv")
      .schema(fire_schema)
      .load("/path/to/raw/collateral")
```

Attentes en matière d'évaluation

Appliquer un schéma est une chose, faire respecter ses contraintes en est une autre. Etant donné la définition de schéma d'une entité FIRE (voir exemple de définition de schéma collatéral), nous pouvons détecter si un champ est obligatoire ou non. Étant donné un objet d'énumération, nous nous assurons que ses valeurs sont cohérentes (par exemple, le code de la devise). En plus des contraintes techniques du schéma, le modèle FIRE fait également rapport des attentes de l'entreprise, telles que les minimum, maximum, monétaires et maxItems. Toutes ces contraintes techniques et business seront récupérées par programme à partir du modèle de données FIRE et interprétées comme une série d'expressions Spark SQL.

```
from fire.spark import FireModel
fire_model = FireModel().load("collateral")
fire_constraints = fire_model.constraints
```

Avec seulement quelques lignes de code, nous nous sommes assurés que notre table Silver soit à la fois correcte sur le plan syntaxique (schéma valide) et sémantique (attentes valides). Avec Delta Live Tables, les utilisateurs ont la possibilité d'évaluer plusieurs attentes à la fois, ce qui leur permet de supprimer les enregistrements invalides, de surveiller simplement la qualite des données ou d'abandonner un pipeline entier.

```
@dlt.create_table()
@dlt.expect_all_or_drop(fire_constraints)
def collateral_silver():
  return dlt.read_stream("collateral_bronze")
```

Avec seulement quelques lignes de code, nous nous sommes assurés que notre table Silver était à la fois correcte sur le plan syntaxique (schéma valide) et sémantique (attentes valides). Comme indiqué ci-dessous, les responsables de la conformité ont une visibilité totale sur le nombre d'enregistrements traités en temps réel. Dans cet exemple particulier, nous nous sommes assurés que notre entité de garantie soit complète à précisément 92,2 % (la quarantaine gère les 7,8 % restants).

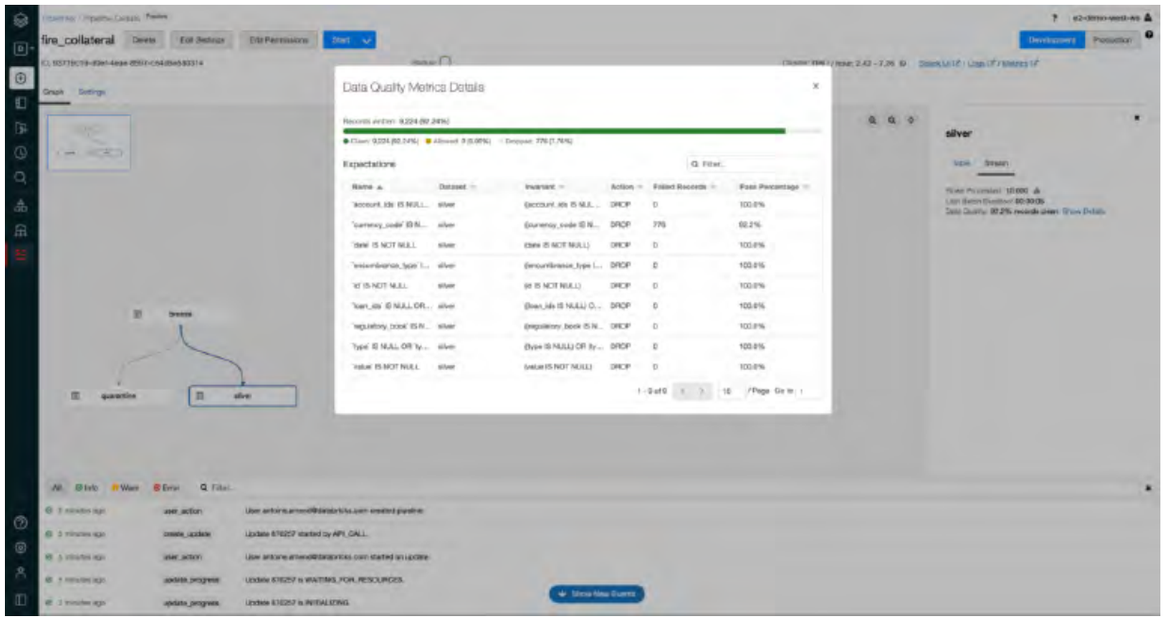

Figure 2

Le grand livre de l'ingénierie des données

En plus des données réelles stockées sous forme de fichiers Delta, Delta Live Tables stocke également les métriques d'exploitation au format « delta » sous le système / les événements. Nous suivons un modèle standard de l'architecture Lakehouse en « souscrivant » à de nouvelles métriques opérationnelles à l'aide d'Auto Loader, en traitant les événements du système au fur et à mesure que de nouvelles métriques se déploient, en batch ou en temps réel. Grâce au journal des transactions de Delta Lake qui garde une trace de toute mise à jour des données, les organisations peuvent accéder à de nouvelles métriques sans avoir à créer et maintenir leur propre processus de point de contrôle.

```
input_stream = spark \
    .readStream \
    .format(“delta”) \
    .load(“/path/to/pipeline/system/events”)

output_stream = extract_metrics(input_stream)

output_stream \
    .writeStream \
    .format(“delta”) \
    .option(“checkpointLocation”, “/path/to/checkpoint”) \
    .table(metrics_table)
```

Avec toutes les métriques disponibles de manière centralisée dans un magasin d'opérations, les analystes peuvent utiliser Databricks SQL pour créer des fonctionnalités de tableau de bord simples ou des mécanismes d'alerte plus complexes pour détecter les problèmes de qualité des données en temps réel.

	entity	expectation_name	expectation_value
1	adjustment	[id] is mandatory	`id` IS NOT NULL
2	adjustment	[date] is mandatory	`date` IS NOT NULL
3	adjustment	[col] is mandatory	`col` IS NOT NULL
4	adjustment	[contribution_amount] is mandatory	`contribution_amount` IS NOT NULL
5	adjustment	[currency_code] is mandatory	`currency_code` IS NOT NULL
6	adjustment	[currency_code] not allowed value	(`currency_code` IS NULL) OR (`currency_code` IN ('AED', 'AFN', 'ALL', 'AMD', 'ANG', 'AOA', 'ARS', 'AUD', 'AWG', 'AZN', 'BAM', 'BBD', 'BDT', 'BGN', 'BHD', 'BIF', 'BMD', 'BND', 'BOB', 'BOV', 'BRL', 'BSD', 'BTN', 'BWP', 'BYN', 'BZD', 'CAD', 'CDF', 'CHE', 'CHF', 'CHW', 'CLF', 'CLP', 'CNY', 'COP', 'COU', 'CRC', 'CUC', 'CUP', 'CVE', 'CZK', 'DJF', 'DKK', 'DOP', 'DZD', 'EGP', 'ERN', 'ETB', 'EUR', 'FJD', 'FKP', 'GBP', 'GEL', 'GHS', 'GIP', 'GMD', 'GNF', 'GTQ', 'GYD', 'HKD', 'HNL', 'HRK', 'HTG', 'HUF', 'I...

Figure 3

L'aspect immuable du format Delta Lake, associé à la transparence de la qualité des données offerte par Delta Live Tables, permet aux institutions financières de « voyager dans le temps » vers des versions spécifiques de leurs données, qui correspondent à la fois au volume et à la qualité requis pour la conformité réglementaire. Dans notre exemple spécifique, la relecture de nos 7,8 % d'enregistrements invalides stockés en quarantaine entraînera une version Delta différente attachée à notre table Silver, une version qui peut être partagée entre les organismes de réglementation.

```
DESCRIBE HISTORY fire.collateral_silver
```

Transmission des données réglementaires

Avec une confiance totale dans la qualité et le volume des données, les institutions financières peuvent échanger en toute sécurité des informations entre les systèmes réglementaires à l'aide de Delta Sharing, un protocole ouvert pour l'échange de données d'entreprise. En ne contraignant pas les utilisateurs finaux à la même plateforme et en ne s'appuyant pas sur des pipelines ETL complexes pour consommer les données (accès aux fichiers de données via un serveur SFTP, par exemple), la nature open source de Delta Lake permet aux utilisateurs d'accéder aux données schématisées de manière native depuis Python, Spark ou directement via des tableaux de bord MI / BI (tels que Tableau ou Power BI).

Bien que nous puissions partager notre table Silver telle quelle, nous pouvons aussi utiliser des règles métier qui ne partagent les données réglementaires que lorsqu'un seuil de qualité des données prédéfini est atteint. Dans cet exemple, nous clonons notre table Silver dans une version différente et à un emplacement spécifique disctinct de nos réseaux internes et accessible par les utilisateurs finaux (zone démilitarisée, ou DMZ).

```
from delta.tables import *

deltaTable = DeltaTable.forName(spark, “fire.collateral_silver”)
deltaTable.cloneAtVersion(
  approved_version,
  dmz_path,
  isShallow=False,
  replace=True
)

spark.sql(
  “CREATE TABLE fire.colleral_gold USING DELTA LOCATION ‘{}’”
    .format(dmz_path)
)
```

Bien que la solution open source Delta Sharing repose sur un serveur de partage pour gérer les autorisations, Databricks s'appuie sur Unity Catalog afin de centraliser et appliquer les politiques de contrôle d'accès, fournir aux utilisateurs une capacité complète de journaux d'audit et simplifier la gestion des accès via son interface SQL. Dans l'exemple ci-dessous, nous créons un PARTAGE qui inclut nos tables réglementaires et un DESTINATAIRE avec lequel partager nos données.

```
-- DEFINE OUR SHARING STRATEGY
CREATE SHARE regulatory_reports;

ALTER SHARE regulatory_reports ADD TABLE fire.collateral_gold;
ALTER SHARE regulatory_reports ADD TABLE fire.loan_gold;
ALTER SHARE regulatory_reports ADD TABLE fire.security_gold;
ALTER SHARE regulatory_reports ADD TABLE fire.derivative_gold;

-- CREATE RECIPIENTS AND GRANT SELECT ACCESS
CREATE RECIPIENT regulatory_body;

GRANT SELECT ON SHARE regulatory_reports TO RECIPIENT regulatory_body;
```

Tout régulateur ou utilisateur disposant d'autorisations accordées peut accéder à nos données sous-jacentes en utilisant un token d'accès personnel échangé dans le cadre de ce processus. Pour plus d'informations sur Delta Sharing, veuillez visiter notre page produit et contacter votre représentant Databricks.

Testez votre conformité

A travers cette série de notebooks et de tâches Delta Live Tables, nous avons démontré les avantages de l'architecture Lakehouse dans l'ingestion, le traitement, la validation et la transmission des données réglementaires. Plus précisément, nous avons abordé la nécessité pour les organisations d'assurer la cohérence, l'intégrité et l'actualité des pipelines réglementaires, ce qui pourrait être facilement réalisé en utilisant un modèle de données commun (FIRE) couplé à un moteur d'orchestration flexible (Delta Live Tables). Grâce aux capacités de partage Delta, nous avons enfin démontré comment les IFA pouvaient apporter une transparence et une confiance totales aux données réglementaires échangées entre les différents systèmes réglementaires, tout en répondant aux exigences de reporting, en réduisant les coûts d'exploitation et en s'adaptant aux nouvelles normes.

Faites connaissance avec le pipeline de données FIRE à l'aide des notebooks ci-joints et explorez notre hub d'accélérateurs de solutions pour vous rester informé de nos dernières solutions pour les services financiers.

Découvrez ces notebooks Databricks gratuits.

databricks

SECTION 2.5

Solutions AML à l'échelle grâce à la plateforme Lakehouse de Databricks

de SRI GHATTAMANENI, RICARDO PORTILLA et ANINDITA MAHAPATRA

16 JUILLET 2021

Résoudre les principaux défis liés à la création d'une solution aux délits financiers

La conformité à la lutte contre le blanchiment d'argent (AML) a sans aucun doute été l'un des principaux points à l'ordre du jour des régulateurs assurant la surveillance des institutions financières à travers le monde. Au fur et à mesure que l'AML a évolué et est devenue plus élaborée au fil des décennies, les exigences réglementaires conçues pour lutter contre les schémas modernes de blanchiment d'argent et de financement du terrorisme ont également évolué. La loi sur le secret bancaire de 1970 a fourni des orientations et un cadre aux institutions financières pour mettre en place des contrôles appropriés pour surveiller les transactions financières et signaler les activités fiscales suspectes aux autorités compétentes. Cette loi a fourni le cadre de la lutte des instituts financiers contre le blanchiment d'argent et le financement du terrorisme.

Pourquoi la lutte contre le blanchiment d'argent est-elle si complexe ?

Les opérations actuelles de lutte contre le blanchiment d'argent ne ressemblent guère à celles de la dernière décennie. Le passage à la banque numérique avec des institutions financières (IF) traitant quotidiennement des milliards de transactions, a entraîné une croissance continue du blanchiment d'argent, même avec des systèmes de surveillance des transactions plus stricts et des connaissances solides. Dans ce blog, nous partageons nos expériences de travail avec nos clients du secteur financier pour construire des solutions AML à l'échelle de l'entreprise sur la plateforme Lakehouse, assurant à la fois une surveillance solide et fournissant des solutions innovantes et évolutives, pour s'adapter à la réalité des menaces modernes de blanchiment d'argent en ligne.

Construire une solution AML avec Lakehouse

La charge opérationnelle du traitement de milliards de transactions par jour vient de la nécessité de stocker les données provenant de plusieurs sources et de solutions AML de nouvelle génération énergivores. Ces solutions fournissent l'analytique de risque et des rapports puissants tout en prenant en charge l'utilisation de modèles de ML avancés pour réduire les faux positifs et améliorer l'efficacité des enquêtes en aval. Les institutions financières ont déjà pris des mesures pour résoudre les problèmes d'infrastructure et de mise à l'échelle en passant du local au cloud pour une sécurité, une agilité et des économies d'échelle accrues nécessaires au stockage d'énormes quantités de données.

Mais se pose ensuite la question de savoir comment donner un sens aux quantités massives de données structurées et non structurées, collectées et sauvegardées sur un stockage d'objets bon marché. Si les fournisseurs de services cloud offrent un moyen peu coûteux de stocker les données, l'exploitation de ces données, pour les activités en aval de gestion du risque de blanchiment d'argent et de mise en conformité, commence par le stockage des données dans des formats de haute qualité et performants pour une utilisation

en aval. C'est ce que fait précisément la plateforme Databricks Lakehouse. En combinant les avantages des faibles coûts de stockage des data lakes avec les capacités de transaction robustes des data warehouses, les institutions financières peuvent véritablement construire la plate-forme AML moderne.

En plus des défis de stockage de données décrits ci-dessus, les analystes AML sont confrontés à des défis spécifiques à un domaine :

- Améliorez le délai de rentabilisation en analysant les données non structurées telles que les images, les données textuelles et les liens réseau
- Réduisez la charge DevOps pour prendre en charge les capacités de ML critiques telles que la résolution d'entités, la vision par ordinateur et l'analytique graphique sur les métadonnées d'entité
- Brisez les silos en introduisant une ingénierie analytique et une couche de tableau de bord sur les transactions AML et les tables enrichies

Heureusement, Databricks aide à résoudre ces problèmes en tirant parti de Delta Lake pour stocker et combiner des données non structurées et structurées afin d'établir des relations d'entité. De plus, le moteur Delta de Databricks fournit un accès efficace en utilisant le nouveau Photon Compute pour accélérer les requêtes de BI sur les tables. En plus de ces capacités, ML est un citoyen de première classe dans Lakehouse, ce qui signifie que les analystes et les data scientists ne perdent pas de temps à sous-échantillonner ou à déplacer des données pour partager des tableaux de bord et garder une longueur d'avance sur les mauvais acteurs.

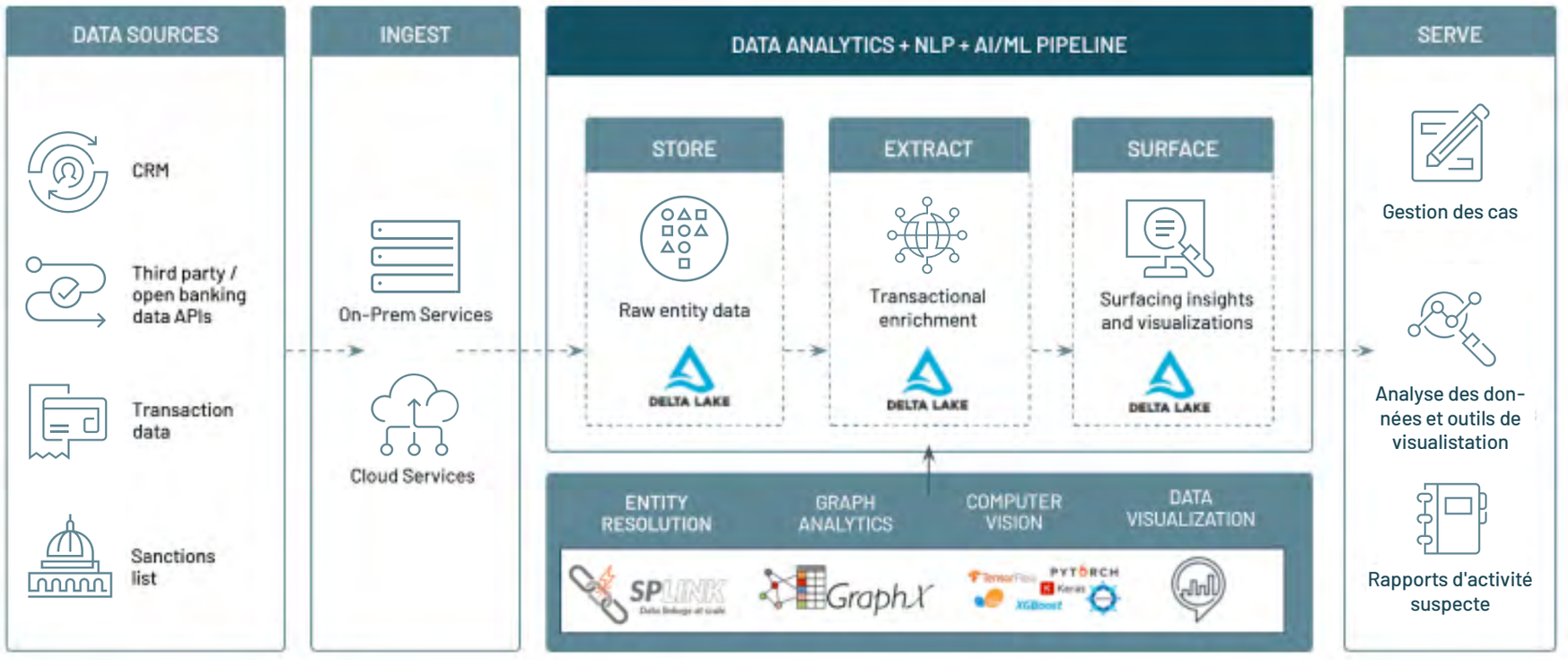

Figure 1

Détection des modèles AML avec des capacités graphiques

L'une des principales sources de données que les analystes AML utilisent dans le cadre d'un dossier sont les *données de transaction*. Même si ces données sont tabulaires et facilement accessibles avec SQL, il devient difficile de suivre des chaînes de transactions de trois couches ou plus avec des requêtes SQL. Pour cette raison, il est important de disposer d'une suite flexible de langages et d'API pour exprimer des concepts simples tels qu'un réseau connecté d'individus suspects effectuant ensemble des transactions illégales. Heureusement, cela est simple à réaliser à l'aide de GraphFrames, une API de graphes préinstallée dans Databricks Runtime pour Machine Learning.

Dans cette section, nous montrerons comment l'analytique graphique peut être utilisée afin de détecter les schémas AML tels que l'identité synthétique et la superposition / structuration. Nous allons utiliser un ensemble de données composé de transactions, ainsi que d'entités dérivées de transactions, pour détecter la présence de ces modèles avec Apache Spark™, GraphFrames et Delta Lake. Les modèles persistants sont enregistrés dans Delta Lake afin que Databricks SQL puisse être appliqué sur les versions agrégées de niveau Gold de ces résultats, offrant ainsi la puissance de l'analytique graphique aux utilisateurs finaux.

Scénario 1 : identités synthétiques

Comme mentionné ci-dessus, l'existence d'identités synthétiques peut être une cause d'alarme. Grâce à l'analytique graphique, toutes les entités de nos transactions peuvent être analysées en masse pour détecter un niveau de risque. Dans notre analyse, cela se fait en trois phases :

- Sur la base des données de transaction, extraire les entités
- Créer des liens entre les entités en fonction de l'adresse, du numéro de téléphone ou de l'e-mail
- Utiliser les composants connectés de GraphFrames pour déterminer si des entités multiples (identifiées par un ID et d'autres attributs ci-dessus) sont connectées via un ou plusieurs liens.

En fonction du nombre de connexions (c'est-à-dire d'attributs communs) existant entre les entités, nous pouvons attribuer un score de risque inférieur ou supérieur et créer une alerte basée sur les groupes à score élevé. Vous trouverez ci-dessous une représentation de base de cette idée.

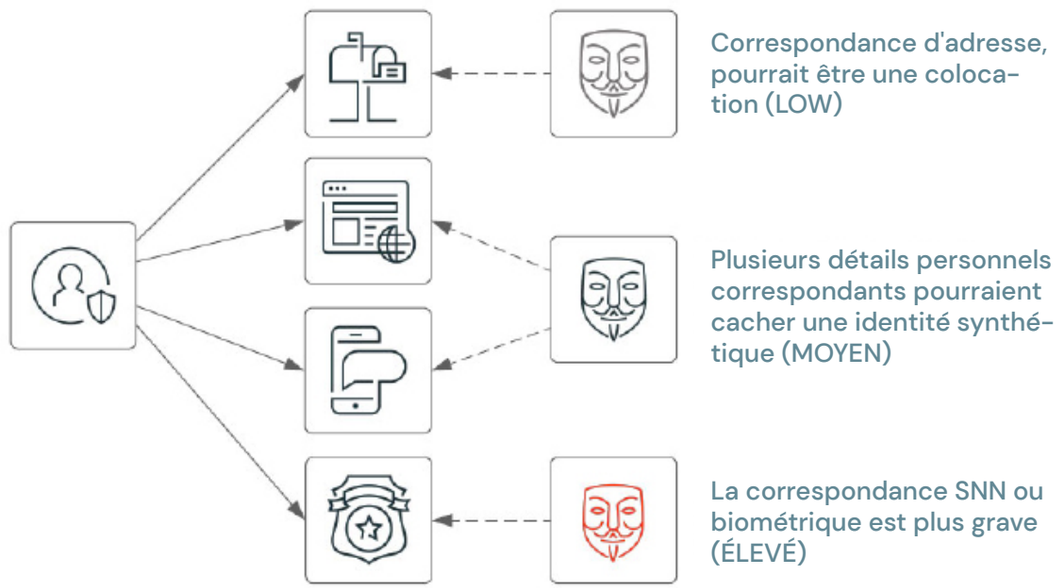

Figure 2

Tout d'abord, nous créons un graphique d'identité en utilisant une adresse, un e-mail et un numéro de téléphone pour relier les individus, s'ils correspondent à l'un de ces attributs.

```
e_identity_sql = '''
select entity_id as src, address as dst from aml.aml_entities_synth  where address is not
null
UNION
select entity_id as src, email as dst from aml.aml_entities_synth  where email_addr is not
null
UNION
select entity_id as src, phone as dst from aml.aml_entities_synth  where phone_number is not
null
'''

from graphframes import *
from pyspark.sql.functions import *
aml_identity_g = GraphFrame(identity_vertices, identity_edges)
result = aml_identity_g.connectedComponents()

result \
 .select("id", "component", 'type') \
 .createOrReplaceTempView("components")
```

Ensuite, nous exécuterons des requêtes pour identifier quand deux entités ont une identification personnelle et des scores qui se chevauchent. Sur la base des résultats de ces composants de graphe d'interrogation, nous nous attendons à une cohorte composée d'un seul attribut correspondant (comme l'adresse par exemple), ce qui n'est pas très inquiétant. Cependant, au fur et à mesure de la correspondance d'autres attributs, nous devons nous attendre à être alertés. Comme indiqué ci-dessous, nous pouvons signaler les cas où les trois attributs correspondent, permettant aux analystes SQL d'obtenir des résultats quotidiens à partir d'analytique graphique exécutée sur toutes les entités.

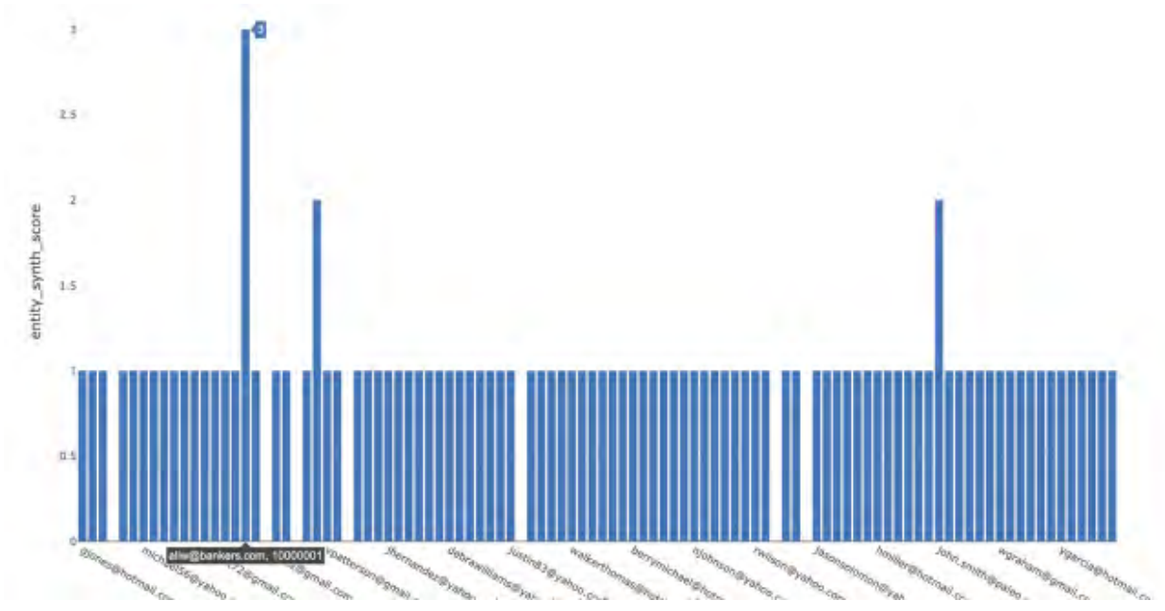

Figure 3

Scénario 2 : structurer

Un autre modèle courant est appelé *structuration*. Il se produit lorsque plusieurs entités s'entendent et envoient des paiements « sous le radar » plus petits à un ensemble de banques, qui acheminent ensuite des montants agrégés plus importants vers une institution finale (comme illustré ci-dessous à l'extrême droite). Dans ce scénario, toutes les parties sont restées sous le seuil de 10 000 dollars, ce qui alerterait généralement les autorités. Non seulement cela est facile à réaliser avec l'analytique graphique, mais la *technique de recherche de motifs* peut être automatisée pour s'étendre à d'autres permutations de réseaux et localiser d'autres transactions suspectes de la même manière.

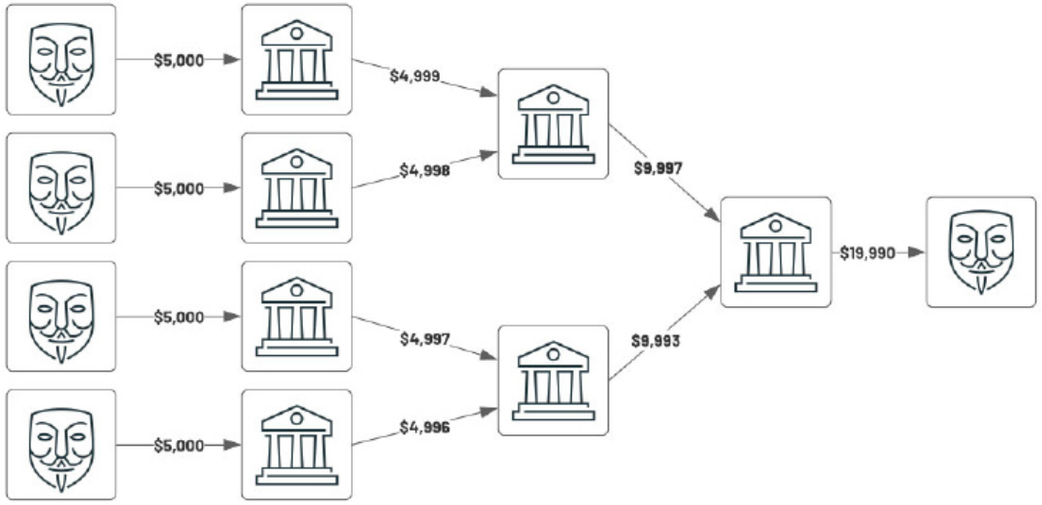

Figure 4

Nous allons maintenant écrire le code de base de recherche de motifs pour détecter le scénario ci-dessus à l'aide des capacités de graphe. Notez que la sortie ici est un JSON semi-structuré. Tous les types de données, y compris les types non structurés, sont facilement accessibles dans le lakehouse, nous enregistrerons ces résultats particuliers pour les rapports SQL.

```
motif = “(a)-[e1]->(b); (b)-[e2]->(c); (c)-[e3]->(d); (e)-[e4]->(f); (f)-[e5]-
>(c); (c)-[e6]->(g)”
struct_scn_1 = aml_entity_g.find(motif)

joined_graphs = struct_scn_1.alias(“a”) \
 .join(struct_scn_1.alias(“b”), col(“a.g.id”) == col(“b.g.id”)) \
 .filter(col(“a.e6.txn_amount”) + col(“b.e6.txn_amount”) > 10000)
```

À l'aide de la recherche de motifs, nous avons extrait des modèles intéressants où l'argent circule à travers quatre entités différentes et est maintenu sous un seuil de 10 000 dollars. Nous joignons nos métadonnées de graphique à des ensembles de données structurées pour générer des insights permettant à un analyste AML d'approfondir ses recherches.

	top_entity_id	first_entity	second_entity	third_entity	fourth_entity
1	1	Brenda Thomas	Teresa Gibson	Mary Strong	Robert Wilkinson
2	3	Lindsey Barber	Joshua Harris	Mary Strong	Robert Wilkinson
3	5	Bruce White	Kathleen Elliott	Victor Arias	Robert Wilkinson
4	7	Jeffrey Lara	Amy Campbell	Victor Arias	Robert Wilkinson

Figure 5

Scénario 3 : propagation du score de risque

Les entités à haut risque identifiées auront une influence (un effet de réseau) sur leur entourage. Ainsi, le score de risque de toutes les entités avec lesquelles ils interagissent doit être ajusté pour refléter la zone d'influence. En utilisant une approche itérative, nous pouvons suivre le flux des transactions à une profondeur donnée et ajuster les scores de risque des autres personnes concernées dans le réseau. Comme mentionné précédemment, l'exécution d'analytique de graphes évite plusieurs jointures SQL répétées et une logique métier complexe, qui peut avoir un impact sur les performances en raison de contraintes de mémoire. L'analytique graphique et l'API Pregel ont été conçues dans ce but précis. Développé initialement par Google, Pregel permet aux utilisateurs de « propager » récursivement des messages de n'importe quel sommet à ses voisins correspondants, en mettant à jour l'état du sommet (ici, son score de risque) à chaque étape. Nous pouvons représenter notre approche dynamique du risque en utilisant l'API Pregel comme suit.

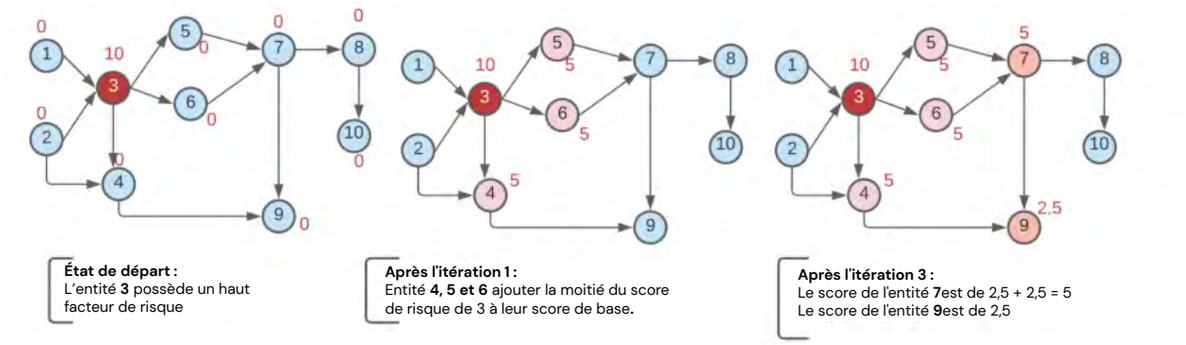

Figure 6

Le diagramme ci-dessous à gauche montre l'état de départ du réseau et deux itérations ultérieures. Supposons que nous commencions avec un mauvais acteur (Node 3) avec un score de risque de 10. Nous voulons pénaliser toutes les personnes qui effectuent des transactions avec ce nœud (à savoir les nœuds 4, 5 et 6) et reçoivent des fonds en transmettant, par exemple, la moitié du score de risque du mauvais acteur, qui est ensuite ajouté à son score de base. Dans la prochaine itération, tous les nœuds en aval des nœuds 4, 5 et 6 verront leurs scores ajustés.

Numéro de nœud	Itération numéro 0	Itération numéro 1	Itération numéro 2
1	0	0	0
2	0	0	0
3	10	10	10
4	0	5	5
5	0	5	5
6	0	5	5
7	0	0	5
8	0	0	0
9	0	0	2,5
10	0	0	0

En utilisant l'API Pregel de GraphFrame, nous pouvons effectuer ce calcul et conserver les scores modifiés pour que d'autres applications en aval puissent les consommer.

```
from graphframes.lib import Pregel

ranks = aml_entity_g.pregel \
    .setMaxIter(3) \
    .withVertexColumn(
       “risk_score”,
       col(“risk”),
       coalesce(Pregel.msg()+ col(“risk”),
       col(“risk_score”))
    ) \
    .sendMsgToDst(Pregel.src(“risk_score”)/2 )  \
    .aggMsgs(sum(Pregel.msg())) \
    .run()
```

Correspondance d'adresse

La correspondance d'adresses entre le texte et les images réelles de Street View est un modèle que nous souhaitons aborder brièvement. Souvent, un analyste AML est nécessaire pour valider la légitimité des adresses liées aux entités enregistrées. Cette adresse est-elle liée à un immeuble commercial, un quartier résidentiel ou une simple boîte aux lettres ? Cependant, l'analyse des images est souvent un processus manuel fastidieux, chronophage à obtenir, nettoyer et valider. Une architecture de données Lakehouse nous permet d'automatiser la plupart de cette tâche à l'aide des environnements d'exécution Python et ML avec PyTorch et des modèles open source pré-entraînés. Vous trouverez ci-dessous un exemple d'adresse valide à l'œil nu. Pour automatiser la validation, nous utiliserons un modèle VGG pré-entraîné pour lequel il existe des centaines d'objets valides que nous pouvons utiliser pour détecter une résidence.

Figure 7

En utilisant le code ci-dessous, qui peut être automatisé pour s'exécuter quotidiennement, nous aurons désormais une étiquette attachée à toutes nos images (nous avons chargé toutes les références et étiquettes d'image dans une table SQL pour une requête plus simple). Remarquez dans le code ci-dessous à quel point il est simple d'interroger un ensemble d'images pour les objets à l'intérieur. La possibilité d'interroger de telles données non structurées avec Delta Lake représente un énorme gain de temps pour les analystes et accélère le processus de validation pour le faire passer à quelques minutes au lieu de plusieurs jours ou semaines.

```
from PIL import Image
from matplotlib import cm

img = Image.fromarray(img)
...

vgg = models.vgg16(pretrained=True)
prediction = vgg(img)
prediction = prediction.data.numpy().argmax()
img_and_labels[i] = labels[prediction]
```

Alors que nous commençons à résumer, nous remarquons l'apparition de certaines catégories intéressantes. Comme on peut le voir dans la répartition ci-dessous, il existe quelques étiquettes évidentes telles que *patio*, *maison mobile* et *scooter* que nous nous attendrions à voir comme éléments détectés dans une adresse résidentielle. D'autre part, le modèle CV a étiqueté une parabole solaire à partir d'objets environnants dans une image. (*Remarque : étant donné que nous sommes limités à un modèle open source non formé sur un ensemble d'images personnalisé, l'étiquette de la parabole solaire n'est pas précise.*) Après une analyse plus approfondie de l'image, nous affinons et voyons immédiatement que i) il n'y a pas de vraie antenne solaire ici et plus important encore ii) cette adresse n'est pas une vraie résidence (représentée dans notre comparaison côte à côte sur la figure 7) . Le format Delta Lake nous permet de stocker une référence à nos données non structurées avec une étiquette pour une interrogation simple dans notre répartition de classification ci-dessous.

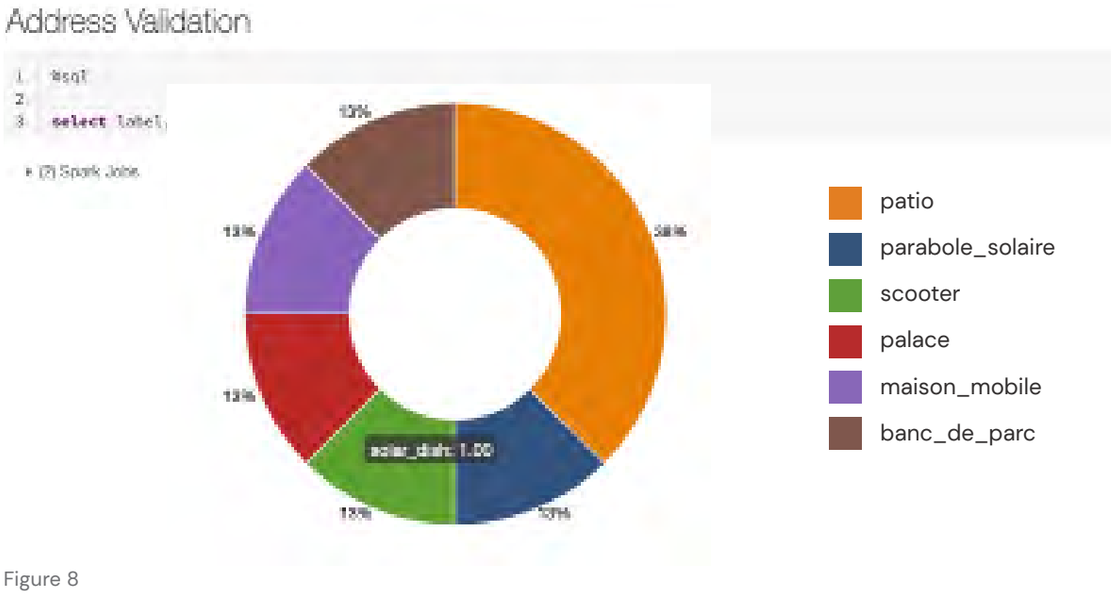

Figure 8

Image Name	Rendered Image	Main Object	Risk Level
img_0.jpg		Patio	Low
img_1.jpg		Solar Dish	High

Figure 9

Résolution d'entité

La dernière catégorie de défis AML sur laquelle nous allons nous concentrer est la résolution d'entités. De nombreuses bibliothèques open source s'attaquent à ce problème. Par conséquent, pour certaines correspondances approximatives d'entités de base, nous avons choisi de mettre en évidence Splink, qui réalise le couplage à l'échelle et offre des configurations pour spécifier les colonnes correspondantes et les règles de blocage.

Dans le contexte des entités dérivées de nos transactions, il s'agit d'un exercice simple pour insérer nos transactions Delta Lake dans le contexte de Splink.

```
settings = {
  “link_type”: “dedupe_only”,
  “blocking_rules”: [
      “l.txn_amount = r.txn_amount”,
  ],
  “comparison_columns”: [
      {
          “col_name”: “rptd_originator_address”,
      },
      {
          “col_name”: “rptd_originator_name”,
      }
  ]
}

from splink import Splink
linker = Splink(settings, df2, spark)
df2_e = linker.get_scored_comparisons()
```

Splink fonctionne en attribuant une probabilité de correspondance qui peut être utilisée pour identifier les transactions dans lesquelles les attributs d'entité sont très similaires, ce qui déclenche une alerte potentielle concernant une adresse, un nom d'entité ou un montant de transaction signalé. Étant donné que la résolution d'entités peut s'avérer hautement manuelle pour faire correspondre les informations de compte, le fait d'avoir des bibliothèques open source qui automatisent cette tâche et enregistrent les informations dans le Delta Lake peut augmenter la productivité des enquêteurs pour la résolution des cas. Bien qu'il existe plusieurs options disponibles pour la correspondance d'entités, nous vous recommandons d'utiliser le hachage sensible à la localité (LSH) pour identifier le bon algorithme pour la tâche. Vous pourrez en savoir plus sur le LSH et ses avantages dans cet article de blog.

Comme indiqué ci-dessus, nous avons rapidement trouvé des incohérences pour l'adresse bancaire de NY Mellon, avec « Canada Square, Canary Wharf, Londres, Royaume-Uni » similaire à « Canada Square, Canary Wharf, Londres, Royaume-Uni. » Nous pouvons stocker nos enregistrements dédupliqués dans une table Delta qui peut être utilisée pour l'enquête AML.

unique_id_l	unique_id_r	rptd_originator_address_l	rptd_originator_address_r
223254	223256	Canada Square, Canary Wharf, London, United Kingdom	Canada Square, Canary Wharf, London, UK

Figure 10

Tableau de bord AML Lakehouse

Databricks SQL sur Lakehouse comble l'écart par rapport aux data warehouses traditionnels en termes de gestion simplifiée des données, de performances avec le nouveau moteur de requête Photon et de concurrence utilisateur. Ceci est important car de nombreuses organisations n'ont pas le budget pour se procurer un logiciel AML propriétaire hors de prix pour prendre en charge la myriade de cas d'usage, tels que la lutte contre le financement du terrorisme (CFT), qui aident à lutter contre la criminalité financière. Sur le marché, il existe des solutions dédiées pouvant effectuer l'analytique graphique ci-dessus, des solutions dédiées au traitement de la BI dans un warehouse et des solutions spécialisées en ML. La conception du lakehouse AML unifie les trois. Les équipes de la plateforme de données AML peuvent tirer parti de Delta Lake à un coût de stockage inférieur (stockage dans le cloud), tout en intégrant facilement des technologies open source pour produire des rapports organisés basés sur la technologie graphique, la vision par ordinateur et l'ingénierie d'analytique SQL. Dans la figure 11, nous montrons la matérialisation du reporting pour l'AML.

Les notebooks joints ont produit un objet de transactions, un objet d'entités, ainsi que des résumés tels que des perspectives de structuration, des niveaux d'identité synthétiques et des classifications d'adresses à l'aide de modèles pré-entraînés. Dans la visualisation Databricks SQL ci-dessous, nous avons utilisé notre moteur SQL Photon pour exécuter des résumés sur ces derniers et une visualisation intégrée pour produire un tableau de bord de rapport en quelques minutes. Il existe des listes de contrôle d'accès complètes sur les deux tables, ainsi que le tableau de bord lui-même, pour permettre aux utilisateurs de partager avec les dirigeants et les équipes de données. Un planificateur pour exécuter ce rapport périodiquement est également intégré. Le tableau de bord est le point culminant de l'IA, de la BI et de l'ingénierie analytique intégrée à la solution AML.

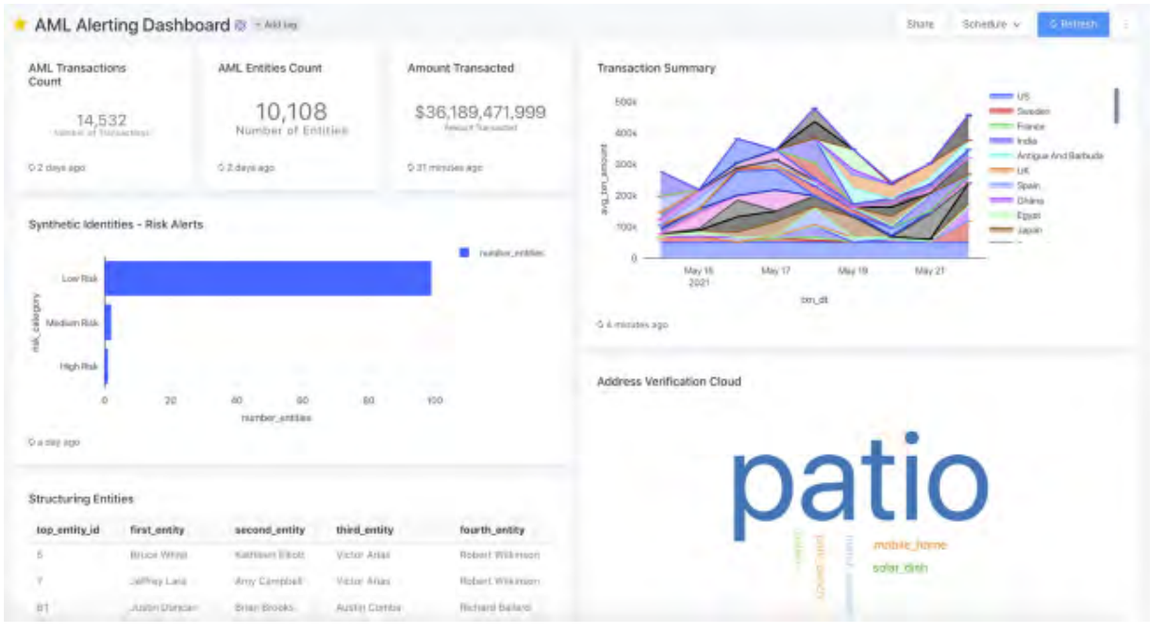

Figure 11

La transformation de l'open banking

L'essor de l'open banking permet aux institutions financières (IF) d'offrir une meilleure expérience client grâce au partage de données entre les consommateurs, les IF et les fournisseurs de services tiers via des API. Un exemple en est la Directive sur les services de paiement (PSD2) qui a transformé les services financiers dans la zone de l'UE dans le cadre de la réglementation Open Banking Europe. En conséquence, les IF ont accès à davantage de données provenant d'une multitude de banques et prestataires de services, y compris les données de compte client et de transaction. Cette tendance s'est étendue au monde de la fraude et des délits financiers avec les dernières directives du FinCEN en vertu de la section 314(b) de l'USA Patriot Act. Les IF couvertes peuvent désormais partager des informations avec d'autres IF et au sein de succursales nationales et étrangères concernant des individus, des entités, des organisations, etc., soupçonnés d'être impliqués dans un éventuel blanchiment d'argent.

Bien que la disposition sur le partage d'informations contribue à la transparence et protège les systèmes financiers des États-Unis contre le blanchiment d'argent et le financement du terrorisme, l'échange d'informations doit être effectué à l'aide de protocoles avec des protections de données appropriées. Pour résoudre le problème de la sécurisation du partage d'informations, Databricks a récemment annoncé le lancement de Delta Sharing, un protocole ouvert et sécurisé pour le partage de données. À l'aide d'API open source familières, telles que Pandas et Spark, les producteurs et les consommateurs de données peuvent désormais partager des données à l'aide de protocoles sécurisés et ouverts, et maintenir un audit complet de toutes les transactions de données préservant la conformité avec les réglementations FinCEN.

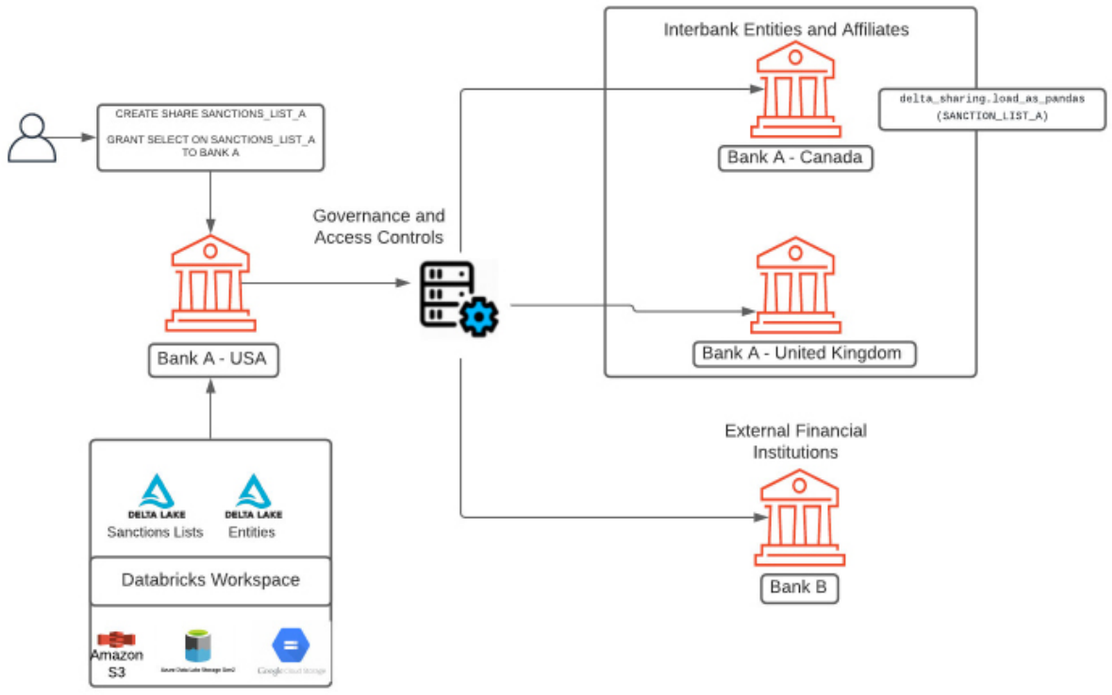

Figure 12

Conclusion

L'architecture Lakehouse est la plateforme la plus évolutive et la plus polyvalente pour permettre aux analystes d'effectuer l'analytique AML. Lakehouse prend en charge des cas d'usage allant de la correspondance approximative à l'analytique d'images, en passant par la BI avec des tableaux de bord intégrés. Toutes ces fonctionnalités permettront aux organisations de réduire le coût total de possession par rapport aux solutions AML propriétaires. L'équipe des services financiers de Databricks travaille sur une variété de problématiques business dans le domaine des services financiers et permet aux professionnels de l'ingénierie des données et de la data science de commencer le parcours Databricks via les accélérateurs de solution tels qu'AML.

Découvrez ces notebooks Databricks gratuits

- **Introduction à la théorie des graphes pour l'AML**
- **Introduction à la vision par ordinateur pour l'AML**
- **Introduction à la résolution d'entités pour l'AML**

databricks

SECTION 2.6

Construire un modèle d'IA en temps réel pour détecter les comportements toxiques dans les contextes de gaming (jeux)

de **DAN MORRIS** et **DUNCAN DAVIS**

16 juin 2021

Dans les jeux vidéo en ligne massivement multijoueurs (MMO), les jeux d'arène de combat en ligne multijoueur (MOBA) et d'autres formes de jeu en ligne, les joueurs interagissent en permanence en temps réel pour se coordonner ou s'affronter alors qu'ils se dirigent vers un objectif commun : gagner. Cette interactivité fait partie intégrante de la dynamique du jeu. Mais en même temps, c'est une ouverture de choix pour les comportements toxiques, un problème omniprésent dans la sphère du jeu vidéo en ligne.

Les comportements toxiques se manifestent sous de nombreuses formes, comme les différents degrés de « griefing » (anti-jeu), de cyberintimidation et de harcèlement sexuel illustrés dans la matrice ci-dessus à droite de Behaviour Interactive, qui répertorie les types d'interactions observées au sein du jeu multijoueur « Dead by Night »

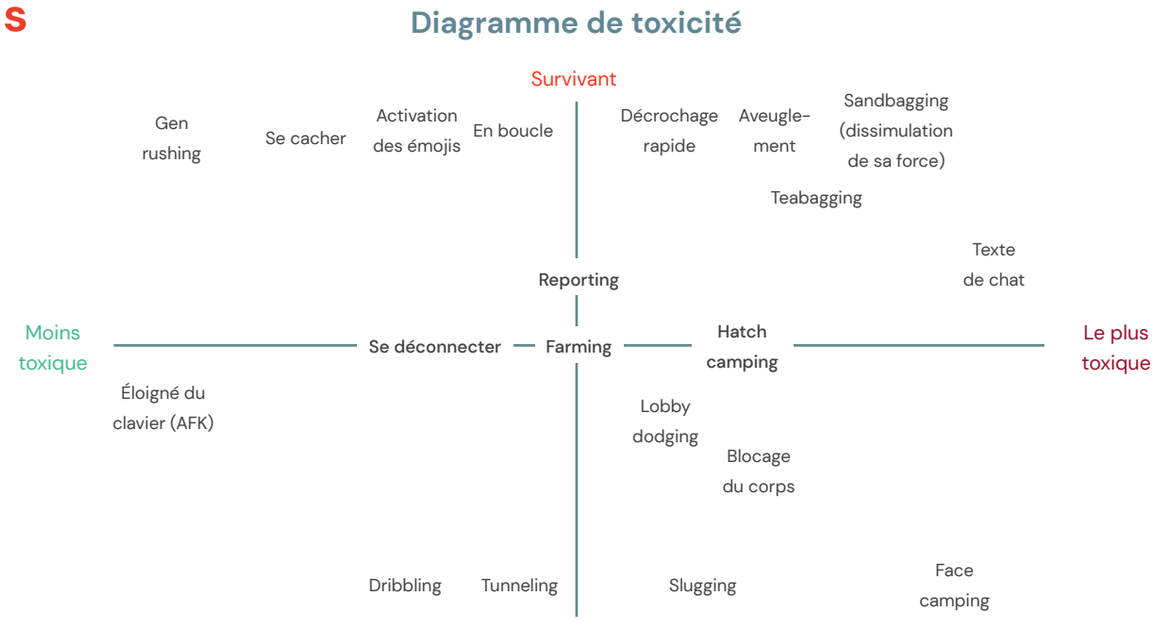

Figure 1
Matrice des interactions toxiques des joueurs

Outre les conséquences personnelles qu'un comportement toxique peut avoir sur les joueurs et la communauté - un problème que l'on ne saurait trop souligner -, il porte également atteinte aux résultats financiers de nombreux studios de jeux. Par exemple, une étude de la Michigan State University a révélé que 80 % des joueurs ont récemment subi un comportement toxique, et parmi eux, 20 % ont déclaré avoir quitté le jeu en raison de ces interactions. De même, une étude de l'Université de Tilburg a montré qu'une rencontre perturbatrice ou toxique lors de la première session du jeu conduit les joueurs à être plus de trois fois davantage susceptibles d'abandonner le jeu pour de bon. Étant donné que la fidélisation des joueurs est une priorité absolue pour de nombreux studios, notamment à l'heure où la distribution des jeux passe des supports physiques aux services à longue durée de vie, il est clair que la toxicité doit être jugulée.

databricks

En plus de ce problème lié au taux de désabonnement, certaines entreprises sont confrontées à des défis liés à la toxicité au début du développement, avant même le lancement. Par exemple, Amazon's Crucible a été lancé dans les tests sans chat textuel ou vocal, en partie parce qu'il n'y avait pas de système en place pour surveiller ou gérer les joueurs et les interactions toxiques. Cela montre que le périmètre de l'espace de jeu a largement dépassé la capacité de la plupart des équipes à gérer un tel comportement, par le biais de rapports ou en intervenant dans des interactions perturbatrices. Dans ces conditions, il est essentiel que les studios intègrent l'analytique dans les jeux dès le début du cycle de développement, puis qu'ils conçoivent la gestion continue des interactions toxiques.

La toxicité dans le jeu est résolument un problème à multiples facettes qui s'est intégré dans la culture du jeu vidéo. Elle ne peut pas être abordée de manière universelle et d'une seule façon. Cela dit, la lutte contre la toxicité dans le chat du jeu peut avoir un impact énorme compte tenu de la fréquence des comportements toxiques et de la possibilité d'automatiser leur détection à l'aide du traitement du langage naturel (NLP).

Présentation de l'accélérateur de détection de toxicité dans les solutions de jeu de Databricks

En utilisant les données de commentaires toxiques de Jigsaw et les données de matchs de jeu de Dota 2, cet accélérateur de solutions parcourt les étapes nécessaires pour détecter les commentaires toxiques en temps réel à l'aide du NLP et de votre lakehouse existant. Pour le NLP, cet accélérateur de solutions utilise Spark NLP de John Snow Labs, une solution open source de qualité professionnelle construite nativement sur Apache Spark.™

Les étapes que vous suivrez dans cet accélérateur de solution sont :

- Charger les données Jigsaw et Dota 2 dans des tables à l'aide de Delta Lake
- Classer les commentaires toxiques en utilisant la classification multi-label (Spark NLP)
- Suivre les expériences et enregistrer les modèles à l'aide de MLflow
- Appliquer l'inférence sur les données en batch et en continu
- Examiner l'impact de la toxicité sur les données de match

Détecter la toxicité dans le chat du jeu en production

Avec cet accélérateur de solution, vous pouvez désormais intégrer plus facilement la détection de toxicité dans vos propres jeux. Par exemple, l'architecture de référence ci-dessous montre comment récupérer des données de chat et de jeu à partir de diverses sources, telles que des flux, des fichiers, des bases de données vocales ou opérationnelles. L'architecture de référence montre aussi comment tirer parti de Databricks pour ingérer, stocker et organiser des données dans des tables de fonctionnalités pour les pipelines de machine learning (ML), le ML intégré dans le jeu, des tables de BI pour l'analyse et même l'interaction directe avec les outils utilisés pour la modération de la communauté.

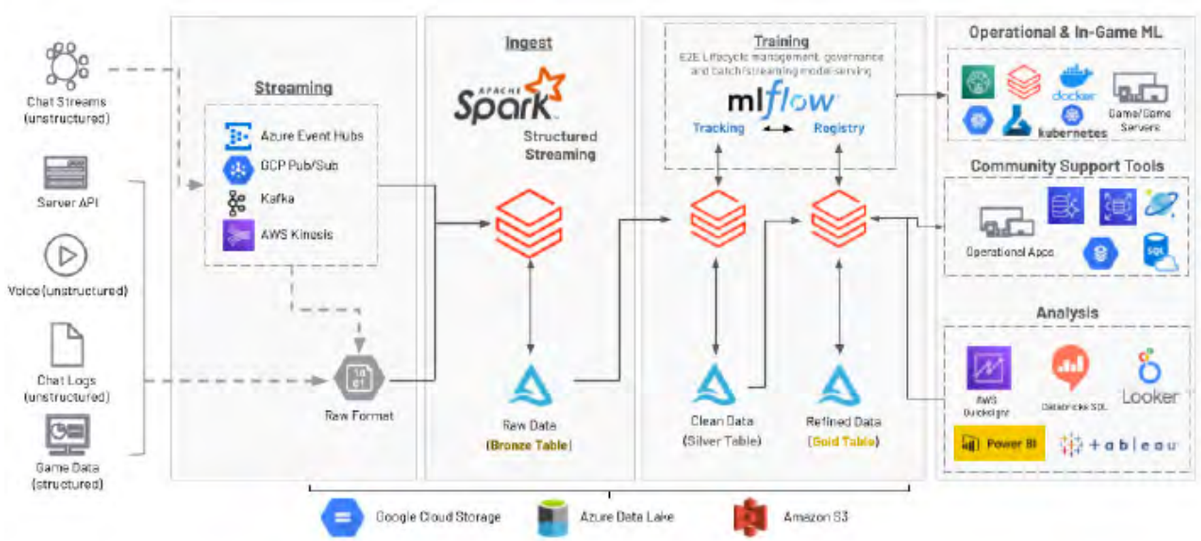

Figure 2
Architecture de référence pour la détection de la toxicité

databricks

Disposer d'une architecture évolutive en temps réel pour détecter la toxicité dans la communauté offre la possibilité de simplifier les flux de travail pour les responsables des relations avec la communauté et la possibilité de filtrer des millions d'interactions dans des charges de travail gérables. De même, la possibilité d'alerter en temps réel sur des événements très toxiques, voire d'automatiser une réponse telle que la mise en sourdine des joueurs ou l'alerte rapide d'un CRM sur l'incident, peut avoir un impact direct sur la fidélisation des joueurs. De même, disposer d'une plateforme capable de traiter de grands ensembles de données, provenant de sources disparates, peut être utilisé pour surveiller la perception de la marque via des rapports et des tableaux de bord.

Démarrer

L'objectif de cet accélérateur de solution est d'aider à prendre en charge la gestion continue des interactions toxiques dans les jeux en ligne en permettant la détection en temps réel des commentaires toxiques dans le chat du jeu. Commencez dès aujourd'hui en important cet accélérateur de solution directement dans votre espace de travail Databricks.

Une fois importés, vous aurez des notebooks avec deux pipelines prêts à passer en production.

- Pipeline ML utilisant la classification multi-label avec formation sur des ensembles de données anglaises du monde réel provenant de Google Jigsaw. Le modèle classera et étiquetera les formes de toxicité dans le texte.
- Pipeline d'inférence de transmission en continu et en temps réel exploitant le modèle de toxicité. La source du pipeline peut être facilement modifiée pour ingérer des données de chat à partir de toutes les sources de données courantes.

Avec ces deux pipelines, vous pouvez commencer à comprendre et à analyser la toxicité avec un effort minimal. Cet accélérateur de solutions fournit également une base pour construire, personnaliser et améliorer le modèle avec des données pertinentes pour les mécanismes de jeu et les communautés.

Découvrez ces notebooks Databricks gratuits.

SECTION 2.7

Transformation de Northwestern Mutual (plateforme d'insights) par l'adoption d'une architecture Lakehouse ouverte et évolutive

de **MADHU KOTIAN**
15 juillet 2021

La transformation digitale a été au centre de la plupart des initiatives récentes des entreprises en matière de big data, en particulier dans les entreprises ayant un lourd héritage. L'un des composants sous-jacents de la transformation numérique est constitué par les données et leur magasin de données associé. Depuis plus de 160 ans, Northwestern Mutual aide les familles et les entreprises à atteindre la sécurité financière. Avec plus de 31 milliards de dollars de chiffre d'affaires, plus de 4,6 millions de clients et plus de 9 300 professionnels de la finance, peu d'entreprises disposent d'un tel volume de données provenant de diverses sources.

L'ingestion de données est un défi à notre époque où les organisations traitent des millions de points de données de différents formats, provenant de multiples périodes et directions, avec un volume sans précédent. Nous voulons que les données soient prêtes à être analysées pour leur donner un sens. Aujourd'hui, je suis ravi de partager notre nouvelle approche pour transformer et moderniser notre processus d'ingestion de données, notre processus de planification et notre parcours avec les magasins de données. Une chose que nous avons apprise est qu'une approche efficace comporte de multiples facettes. C'est pourquoi, en plus des dispositions techniques, je passerai en revue le plan d'intégration de notre équipe.

Les problématiques rencontrées

Avant de nous lancer dans notre transformation, nous avons travaillé avec nos partenaires commerciaux pour vraiment comprendre nos contraintes techniques et nous aider à formuler l'énoncé du problème pour notre « business case ».

Le point sensible de l'entreprise que nous avons identifié était un manque de données intégrées, les données clients et commerciales provenant de différentes équipes et sources de données internes et externes. Nous étions conscients de la valeur des données en temps réel, mais nous disposions d'un accès limité aux données de production / en temps réel pouvant nous permettre de prendre des décisions commerciales au moment opportun. Nous avons également appris que les magasins de données construits par l'équipe commerciale entraînaient des silos de données qui, à leur tour, provoquaient des problèmes de latence, une augmentation des coûts de gestion des données et des contraintes de sécurité injustifiées.

De plus, il existait des défis techniques par rapport à l'état actuel que nous connaissons. Dans un contexte de demande accrue et de données supplémentaires nécessaires, nous avons rencontré des contraintes liées à l'évolutivité de l'infrastructure, la latence des données, le coût de gestion des silos, les limitations de taille et de volume des données et les problèmes de sécurité. Avec ces défis croissants, nous savions que nous avions beaucoup à faire et que nous devions trouver les bons partenaires pour nous accompagner dans notre parcours de transformation.

Analyse des solutions

Nous devions nous axer sur les données pour être compétitifs, mieux servir nos clients et optimiser les processus internes. Nous avons exploré diverses options et effectué plusieurs POC pour sélectionner une recommandation finale. Voici les éléments essentiels de notre stratégie d'avenir :

- Une solution complète pour nos besoins d'ingestion, de gestion et d'analytique des données
- Une plateforme de données moderne pouvant aider efficacement nos développeurs et business analysts à effectuer leurs analyses à l'aide de SQL
- Un moteur de données pouvant prendre en charge les transactions ACID en plus de S3 et activer la sécurité basée sur les rôles
- Un système capable de sécuriser efficacement nos données personnelles / informations personnelles de santé
- Une plateforme capable d'évoluer automatiquement en fonction du traitement des données et de la demande analytique

Notre infrastructure héritée était basée sur MSBI Stack. Nous avons utilisé SSIS pour l'ingestion, SQL Server pour notre magasin de données, Azure Analysis Service pour le modèle tabulaire et Power BI pour le tableau de bord et la reporting. Bien que la plateforme ait initialement répondu aux besoins de l'entreprise, nous avons rencontré des problèmes d'évolutivité avec un volume de données et une demande de traitement de données accrus, et nous avons limité nos attentes en matière d'analytique des données. Avec des besoins de données supplémentaires, nos problèmes de latence dus aux retards de chargement et à un magasin de données pour des besoins business spécifiques, ont entraîné des silos et une prolifération des données.

La sécurité est devenue un défi en raison de la dispersion des données dans plusieurs magasins. Nous avions environ 300 tâches ETL qui nécessitaient plus de sept heures sur nos tâches quotidiennes. Le délai de mise sur le marché pour tout changement ou nouveau développement était d'environ quatre à six semaines (selon la complexité).

Figure 1
Architecture héritée

Après avoir évalué plusieurs solutions sur le marché, nous avons décidé d'aller de l'avant avec Databricks pour nous aider à fournir une solution intégrée de gestion des données sur une architecture Lakehouse ouverte.

Le développement en cours de Databricks, au-dessus d'Apache Spark™, nous a permis d'utiliser Python pour créer un cadre personnalisé pour l'ingestion de données et la gestion des métadonnées. Il nous a fourni la flexibilité pour effectuer des analyses ad hoc et d'autres découvertes de données à l'aide du notebook. Delta Lake de Databricks (la couche de stockage construite au-dessus de notre data lake) nous a fourni la flexibilité nécessaire pour mettre en œuvre diverses fonctions de gestion de base de données (transactions ACID, gouvernance des métadonnées, voyage dans le temps, etc.), y compris la mise en œuvre des contrôles de sécurité requis. Databricks nous a épargné le casse-tête de la gestion et de la mise à l'échelle du cluster et a répondu efficacement à la demande croissante de nos ingénieurs et de nos utilisateurs.

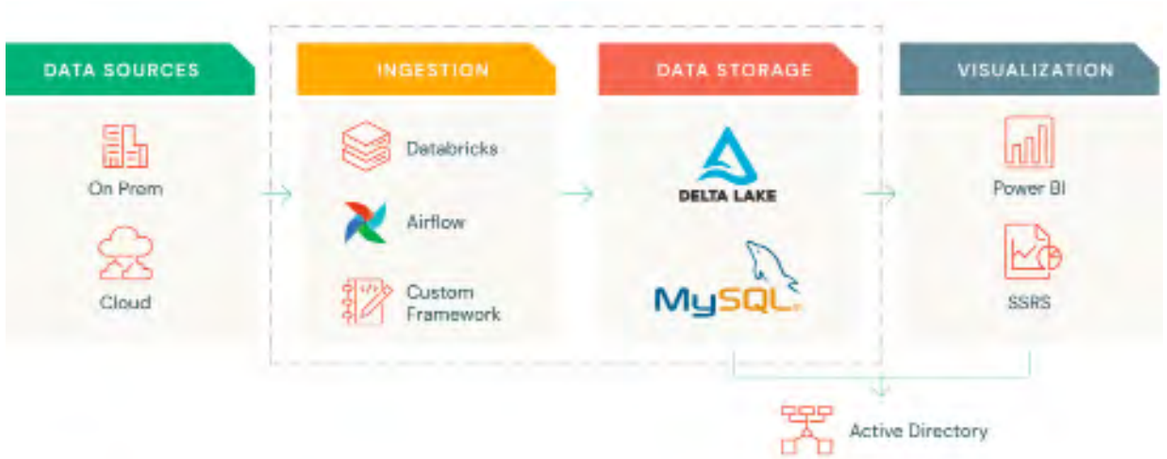

Figure 2
L'architecture avec Databricks

Approche de migration et ressources d'intégration

Nous avons commencé avec un petit groupe d'ingénieurs et les avons affectés à une équipe virtuelle de notre équipe Scrum existante. Leur objectif était d'exécuter différents POC, de s'appuyer sur la solution recommandée, de développer de bonnes pratiques et de revenir à leur équipe respective pour faciliter l'intégration. Tirer parti des membres de l'équipe existante nous a davantage favorisés car ils avaient des connaissances sur les systèmes en place, comprenaient le flux d'ingestion / les règles commerciales actuels et maîtrisaient bien au moins une compétence de la programmation (ingénierie des données et connaissances en ingénierie logicielle). Cette équipe s'est dans un premier temps formée à Python, a intégré les aspects complexes de Spark et Delta et a collaboré étroitement avec l'équipe Databricks pour valider la solution / l'approche. Pendant que l'équipe travaillait sur l'élaboration de l'état futur, le reste de nos développeurs travaillait sur la livraison aux professionnels de l'entreprise.

Étant donné que la plupart des développeurs étaient des ingénieurs MSBI Stack, notre plan d'action consistait à fournir une plateforme de données sans friction pour nos développeurs, nos utilisateurs professionnels et nos conseillers sur le terrain.

- Nous avons construit un cadre d'ingestion couvrant tous nos besoins de chargement et de transformation de données. Il comportait des contrôles de sécurité intégrés, qui conservaient toutes les métadonnées et les secrets de nos systèmes sources. Le processus d'ingestion a accepté un fichier JSON comprenant la source, la cible et la transformation requise. Il a permis une transformation à la fois simple et complexe.
- Pour la planification, nous avons fini par utiliser Airflow, mais étant donné la complexité des DAG, nous avons construit notre propre cadre personnalisé au-dessus d'Airflow, qui acceptait un fichier YAML contenant des informations sur le travail et ses interdépendances associées.
- Pour gérer les modifications au niveau du schéma à l'aide de Delta, nous avons construit notre propre cadre personnalisé qui a automatisé différentes opérations de type de base de données (DDL) sans exiger des développeurs qu'ils aient un accès par effraction au magasin de données. Cela nous a également aidé à mettre en œuvre différents contrôles d'audit sur le magasin de données.

En parallèle, l'équipe a également collaboré avec notre équipe sécurité pour s'assurer que nous comprenions et respections tous les critères de sécurité des données (chiffrement en transit, au repos et au niveau de la colonne pour protéger les données personnelles (PII).

Une fois ces cadres mis en place, l'équipe de la cohorte a déployé un flux de bout en bout (de la source à la cible avec toutes les transformations) et a généré un nouvel ensemble de rapports / tableaux de bord sur Power BI pointant vers Delta Lake. L'objectif était de tester les performances de notre processus de bout en bout, de valider les données et d'obtenir les éventuels retours de nos utilisateurs sur le terrain. Nous avons progressivement amélioré le produit en fonction des retours et des résultats de notre test de performance / validation.

Simultanément, nous avons créé des guides de formation et des procédures pour accompagner nos développeurs. Peu de temps après, nous avons décidé de déplacer les membres de l'équipe de cohorte dans leurs équipes respectives tout en conservant quelques-unes de ces personnes pour continuer à prendre en charge l'infrastructure de la plateforme (DevOps). Chaque équipe Scrum était responsable de la gestion et de la livraison de son ensemble respectif de capacités / fonctionnalités business. Une fois que les membres de l'équipe sont retournés dans leurs équipes respectives, ils se sont attelés à ajuster la vélocité de l'équipe pour inclure le backlog de l'effort de migration. Les chefs d'équipe ont reçu des conseils spécifiques et des objectifs appropriés pour réussir la migration pour différents incréments de sprint / programme. Les membres de l'équipe qui faisaient partie du groupe de cohorte étaient désormais les experts résidents et ils ont aidé leur équipe à intégrer la nouvelle plateforme. Ils se sont montrés disponibles pour toute question ou assistance ponctuelle.

Au fur et à mesure que nous construisions notre nouvelle plateforme, nous avons conservé l'ancienne pour validation et vérification.

Le début du succès

La transformation globale nous a pris environ un an et demi, ce qui constitue un exploit étant donné que nous avons dû construire tous les cadres, gérer les priorités de l'entreprise et les attentes en matière de sécurité, tout en rééquipant notre équipe et migrant la plateforme. Le temps de chargement global a considérablement diminué, passant de sept heures à seulement deux heures. Notre délai de mise sur le marché était d'environ une à deux semaines, en baisse significative de quatre à six semaines. Il s'agit d'une amélioration majeure qui, je le sais, s'étendra à notre entreprise à plusieurs titres.

Notre parcours n'est pas terminé. Alors que nous continuons à améliorer notre plateforme, notre prochaine mission sera d'étendre le modèle au lakehouse. Nous travaillons à la migration de notre plateforme vers E2 et au déploiement de Databricks SQL. Nous élaborons notre stratégie visant à fournir une plateforme en self-service à nos utilisateurs professionnels pour effectuer leurs analyses ad hoc et leur permettre d'apporter leurs propres données, avec la possibilité d'effectuer des analyses avec nos données intégrées. Ce que nous avons appris, c'est que nous avons grandement bénéficié de l'utilisation d'une plateforme ouverte, unifiée et évolutive. Au fur et à mesure que nos besoins et nos capacités augmentent, nous savons, avec Databricks, que nous disposons d'un partenaire solide.

En savoir plus sur le parcours de Northwestern Mutual vers le lakehouse

CONCERNANT MADHU KOTIAN

Madhu Kotian est vice-président de l'ingénierie (données sur les produits d'investissement, CRM, applications et rapports) chez Northwestern Mutual. Il possède plus de 25 ans d'expérience dans le domaine des technologies de l'information avec une expérience et une expertise dans l'ingénierie des données, la gestion des personnes, la gestion de programmes, l'architecture, la conception, le développement et la maintenance en utilisant des approches agiles. Il est également un expert des méthodologies de data warehouse et de la mise en œuvre de l'intégration et de l'analytique de données.

SECTION 2.8

Comment l'équipe chargée des données de Databricks a construit un lakehouse sur trois clouds et plus de 50 régions

de **JASON POHL** et **SURAJ ACHARYA**
14 juillet 2021

L'infrastructure de journalisation interne de Databricks a évolué au fil des ans et nous avons tiré quelques leçons en cours de route, sur la façon de maintenir un pipeline de journaux hautement disponible sur plusieurs clouds et zones géographiques. Ce blog vous donnera un aperçu de la façon dont nous collectons et administrons des métriques en temps réel à l'aide de notre plateforme Lakehouse, et comment nous exploitons plusieurs clouds pour nous aider à nous remettre des pannes du cloud public.

Lors de sa création, Databricks ne prenait en charge qu'un seul cloud public. Désormais, le service s'est développé pour prendre en charge les trois principaux clouds publics (AWS, Azure, GCP) dans plus de 50 régions du monde. Chaque jour, Databricks fait tourner des millions de machines virtuelles pour le compte de ses clients. Notre équipe de plateforme de données de moins de 10 ingénieurs est responsable de la création et de la maintenance de l'infrastructure de télémétrie de journalisation, qui traite un demi-pétaoctet de données chaque jour. L'orchestration, la surveillance et l'utilisation sont capturées via des journaux de service qui sont traités par notre infrastructure pour fournir des métriques précises et opportunes. En fin de compte, ces données sont stockées dans notre propre Delta Lake de la taille d'un pétaoctet. Notre équipe de plateforme de données utilise Databricks pour effectuer le traitement inter-cloud afin que nous puissions fédérer les données le cas échéant, atténuer la récupération après une panne de cloud régionale et minimiser les perturbations de notre infrastructure en direct.

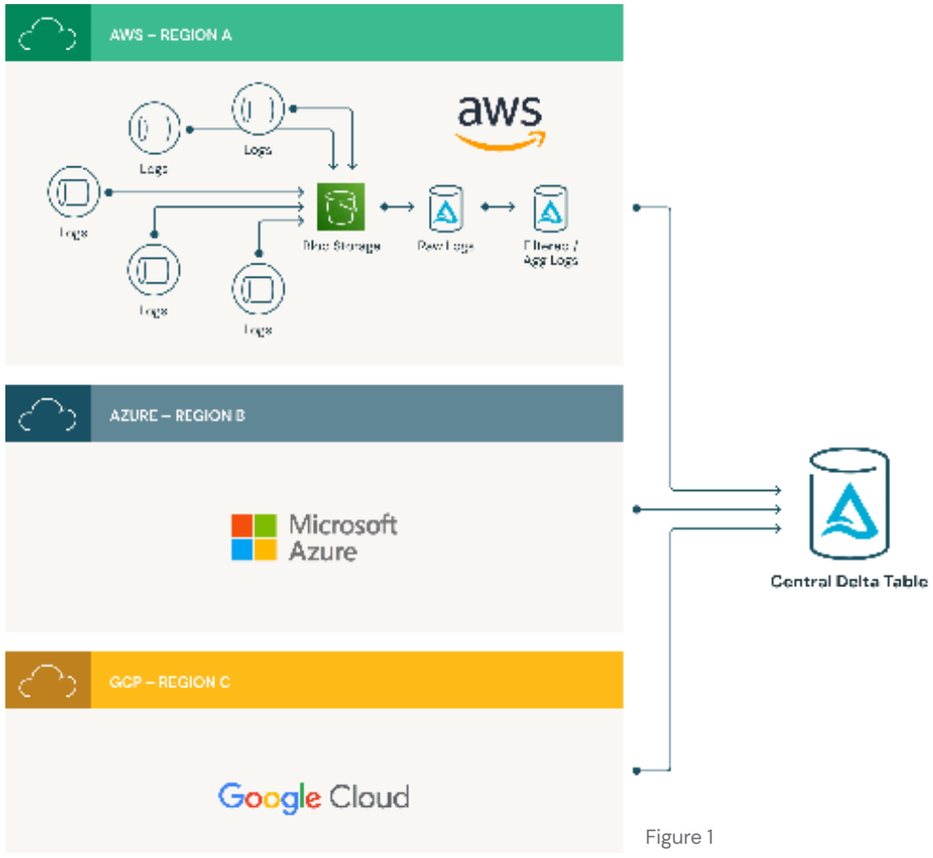

Figure 1

Architecture du pipeline

Chaque région cloud contient sa propre infrastructure et ses propres pipelines de données pour capturer, collecter et conserver les données de journal dans un Delta Lake régional. Les données de télémétrie du produit sont capturées dans l'ensemble du produit et dans nos pipelines par le même processus répliqué dans chaque région cloud. Un daemon capture les données de télémétrie et écrit ensuite les logs sur un compartiment de stockage dans le cloud régional (S3, WASBS, GCS). À partir de là, un pipeline planifié ingére les fichiers de log à l'aide d'Auto Loader (AWS | Azure | GCP) et écrit les données dans une table Delta régionale. Un pipeline différent lit les données de la table Delta régionale, les filtre et les écrit dans une table Delta centralisée dans une seule région cloud.

databricks

Avant Delta Lake

Avant Delta Lake, nous écrivions les données source dans leur propre table dans le lac centralisé, puis nous créions une vue représentant l'union entre toutes ces tables. Cette vue devait être calculée au moment de l'exécution et s'est avérée plus inefficace à mesure que nous ajoutions plus de régions :

```
CREATE OR REPLACE VIEW all_logs AS
SELECT * FROM (
  SELECT * FROM region_1.log_table
  UNION ALL
  SELECT * FROM region_2.log_table
  UNION ALL
  SELECT * FROM region_3.log_table
  ...
);
```

Qu'est-ce que Delta Lake ?

Aujourd'hui, nous n'avons qu'une seule table Delta qui accepte les instructions d'écriture simultanées de plus de 50 régions différentes. Tout en traitant simultanément les requêtes sur les données. Cela rend l'interrogation de la table centrale aussi simple que :

```
SELECT * FROM central.all_logs;
```

La transactionnalité est gérée par Delta Lake. Nous avons abandonné les tables régionales individuelles dans notre Delta Lake central et retiré la vue UNION ALL.

Le code suivant est une représentation simplifiée de la syntaxe exécutée pour charger les données approuvées pour la sortie des Delta Lakes régionaux vers le Delta Lake central :

```
spark.readStream.format("delta")
  .load(regional_source_path)
  .where("egress_approved = true")
  .writeStream
  .format("delta")
  .outputMode("append")
  .option("checkpointLocation", checkpoint_path)
  .start(central_target_path)
```

Récupération après catastrophe

L'un des avantages de l'exploitation d'un service inter-cloud est que nous sommes bien positionnés vis-à-vis de certains scénarios de reprise après catastrophe. Bien que rare, il n'est pas exclu que le service de calcul d'une région du cloud particulière subisse une panne. Lorsque cela se produit, le stockage dans le cloud est accessible, mais la possibilité de faire tourner de nouvelles machines virtuelles est entravée. Étant donné que nous avons conçu notre code de pipeline de données pour accepter la configuration des chemins source et de destination, cela nous permet de déployer et d'exécuter rapidement des pipelines de données dans une région différente de celle où les données sont stockées. Le cloud dans lequel le cluster est créé n'a aucune incidence sur le cloud dans lequel les données sont lues ou écrites.

Il existe quelques ensembles de données que nous protégeons contre les défaillances du service de stockage en répliquant en continu les données sur des fournisseurs de cloud. Cela peut facilement être fait en tirant parti de la fonctionnalité de clonage profond de Delta, comme décrit dans ce blog. Chaque fois que la commande clone est exécutée sur une table, elle met à jour le clone avec uniquement les modifications incrémentielles depuis la dernière fois qu'elle a été exécutée. C'est un moyen efficace de répliquer les données entre les régions et même les clouds.

Minimiser les perturbations des pipelines de données en direct

Nos pipelines de données sont la pierre angulaire de notre service géré, et font partie d'une entreprise mondiale qui ne dort pas. Nous ne pouvons pas nous permettre d'interrompre les pipelines pendant une période prolongée pour la maintenance, les mises à niveau ou le réapprovisionnement en données. Récemment, nous avons dû bifurquer nos pipelines pour filtrer un sous-ensemble des données normalement écrites dans notre table principale, afin qu'elles soient écrites dans un autre cloud public. Nous avons pu le faire sans perturber les activités, comme d'habitude.

En suivant ces étapes, nous avons pu déployer des modifications de notre architecture dans notre système en direct sans causer de perturbation.

> Tout d'abord, nous avons effectué un clone profond de la table principale vers un nouvel emplacement sur l'autre cloud. Cela copie à la fois les données et le journal des transactions de manière à assurer la cohérence.
>
> Deuxièmement, nous avons publié la nouvelle configuration dans nos pipelines afin que la majorité des données continue d'être écrite dans la table principale centrale, et que le sous-ensemble de données soit écrit dans la nouvelle table clonée dans les différents clouds. Cette modification peut être effectuée facilement en déployant simplement une nouvelle configuration et les tables reçoivent des mises à jour uniquement pour les nouvelles modifications qu'elles devraient recevoir.
>
> Ensuite, nous avons exécuté à nouveau la même commande de clonage profond. Delta Lake capture et copie uniquement les modifications incrémentielles de la table principale d'origine vers la nouvelle table clonée. Cela remplit essentiellement la nouvelle table avec toutes les modifications apportées aux données entre les étapes 1 et 2.
>
> Enfin, le sous-ensemble de données peut être supprimé de la table principale et la majorité des données peuvent être supprimées de la table clonée.
>
> Désormais, les deux tables représentent les données qu'elles sont censées contenir, avec un historique complet des transactions, et cela est fait en direct sans perturber la fraîcheur des pipelines.

Résumé

Databricks fait abstraction des détails des services cloud individuels, que ce soit pour faire tourner l'infrastructure avec notre gestionnaire de cluster, ingérer des données avec Auto Loader ou effectuer des écritures transactionnelles sur le stockage cloud avec Delta Lake. Cela nous offre l'avantage de pouvoir utiliser une seule base de code pour relier le calcul et le stockage à travers les clouds publics, à la fois pour la fédération des données et la récupération après catastrophe. Cette fonctionnalité inter-cloud nous donne la flexibilité de déplacer le calcul et le stockage là où ils nous servent le mieux, ainsi qu'à nos clients.

SECTION

03

Témoignages de clients

Atlassian

ABN AMRO

J.B. Hunt

SECTION 3.1

Atlassian

Atlassian est l'un des principaux fournisseurs de logiciels de collaboration, de développement et de suivi des incidents pour les équipes. Avec plus de 150 000 clients mondiaux (dont 85 du Fortune 100), Atlassian fait progresser la puissance de la collaboration avec des produits tels que Jira, Confluence, Bitbucket, Trello et plus encore.

CAS D'USAGE

Atlassian utilise la plateforme Databricks Lakehouse pour démocratiser les données dans toute l'entreprise et réduire les coûts d'exploitation. Atlassian propose actuellement un certain nombre de cas d'usage axés sur la mise en avant de l'expérience client.

Assistance et service à la clientèle
La majorité de ses clients étant basés sur des serveurs (utilisant des produits tels que Jira et Confluence), Atlassian a entrepris de déplacer ces clients vers le cloud pour tirer parti d'insights plus approfondis enrichissant l'expérience du support client.

Personnalisation du marketing
Les mêmes insights pourraient également être utilisés pour envoyer des e-mails marketing personnalisés afin de susciter l'engagement avec de nouvelles fonctionnalités et de nouveaux produits.

Détection anti-abus et fraude
Ils peuvent prédire les abus de licence et les comportements frauduleux grâce à la détection d'anomalies et à l'analytique prédictive.

Chez Atlassian, nous devons nous assurer que les équipes collaborent au mieux entre elles pour atteindre des objectifs en constante évolution. Une architecture simplifiée de type Lakehouse nous permettrait d'ingérer de gros volumes de données utilisateur et d'exécuter l'analytique nécessaire pour mieux prévoir les besoins des clients et améliorer leur expérience. Une plateforme analytique unique dans le cloud, facile à utiliser, nous permet d'améliorer rapidement et de créer de nouveaux outils de collaboration basés sur des insights exploitables.

Rohan Dhupelia
Data Platform Senior Manager, Atlassian

SOLUTION ET AVANTAGES

Atlassian utilise la plateforme Lakehouse de Databricks pour permettre la démocratisation des données à grande échelle, tant en interne qu'en externe. L'entreprise est passée d'un paradigme d'entreposage de données à une standardisation sur Databricks, ce qui lui a permis de devenir davantage axée sur les données. Plus de 3 000 utilisateurs internes dans des domaines allant des RH et du marketing à la finance et à la R&D - soit plus de la moitié de l'organisation - accèdent chaque mois aux insights de la plateforme via des technologies ouvertes telles que Databricks SQL. Atlassian utilise également la plateforme pour offrir à ses clients des expériences d'assistance et de service plus personnalisées.

- Delta Lake est la base d'un lakehouse unique pour les pétaoctets de données accessibles par plus de 3 000 utilisateurs dans les domaines des RH, du marketing, des finances, des ventes, de l'assistance et de la R&D
- Les charges de travail BI optimisées par Databricks SQL permettent la création de tableaux de bord pour un plus grand nombre d'utilisateurs
- MLflow rationalise les MLOps pour une livraison plus rapide
- L'unification de la plateforme de données facilite la gouvernance et les clusters autogérés permettent l'autonomie

Avec une architecture à l'échelle du cloud, une productivité améliorée grâce à la collaboration entre les équipes et la possibilité d'accéder à toutes les données clients pour l'analytique et le ML, l'impact sur Atlassian devrait être considérable. L'entreprise a déjà :

- Réduit le coût de ses opérations IT (en particulier les coûts de calcul) de 60 % en déplaçant plus de 50 000 tâches Spark d'EMR vers Databricks avec un minimum d'effort et un changement de code réduit
- Diminué le délai de livraison de 30 % avec des cycles de développement plus courts
- Réduit les dépendances de l'équipe de données de 70 % avec plus de self-service activé au sein de toute l'organisation

En savoir plus

databricks

SECTION 3.2

ABN AMRO

En tant que banque bien établie, ABN AMRO voulait moderniser ses activités, mais elle était paralysée par une infrastructure et des data warehouses hérités qui compliquaient l'accès aux données provenant de diverses sources et créaient des processus et des flux de données inefficaces. Aujourd'hui, Azure Databricks permet à ABN AMRO de démocratiser les données et l'IA pour une équipe de plus de 500 ingénieurs, scientifiques et analystes habilités travaillant en collaboration, pour améliorer les opérations commerciales et introduire de nouvelles capacités de mise sur le marché dans toute l'entreprise.

CAS D'USAGE

ABN AMRO utilise la plateforme Databricks Lakehouse pour fournir une transformation des services financiers à l'échelle mondiale, offrant une automatisation et des informations sur toutes les opérations.

Finance personnalisée
ABN AMRO exploite les données en temps réel et les informations clients pour fournir des produits et services adaptés aux besoins des clients. Par exemple, l'entreprise utilise le ML pour alimenter des messages ciblés dans ses campagnes marketing automatisées afin de favoriser l'engagement et la conversion.

Gestion des risques
En utilisant une prise de décision basée sur les données, elle se concentre sur l'atténuation des risques pour l'entreprise et ses clients. Par exemple, elle génère des rapports et des tableaux de bord que les décideurs internes et les dirigeants utilisent pour mieux comprendre les risques et éviter qu'ils n'affectent les activités d'ABN AMRO.

Détection de fraude
Dans le but de prévenir les activités malveillantes, elle utilise l'analytique prédictive pour identifier la fraude avant qu'elle n'affecte les clients. Parmi les activités qu'elle tente de combattre figurent le blanchiment d'argent et les fausses demandes de carte de crédit.

Databricks a changé la façon dont nous faisons des affaires. Elle nous a mis dans une meilleure position pour réussir notre transformation en matière de données et d'IA en tant qu'entreprise, en permettant aux professionnels des données de disposer de capacités avancées de manière contrôlée et évolutive.

Stefan Groot
Chef de l'ingénierie analytique,
ABN AMRO

SOLUTION ET AVANTAGES

Aujourd'hui, Azure Databricks permet à ABN AMRO de démocratiser les données et l'IA pour une équipe de plus de 500 ingénieurs, scientifiques et analystes habilités qui travaillent en collaboration, pour améliorer les opérations commerciales et introduire de nouvelles capacités de mise sur le marché dans toute l'entreprise.

- Delta Lake permet de profiter de pipelines de données rapides et fiables pour alimenter des données précises et complètes pour l'analytique en aval
- L'intégration avec Power BI permet une analytique SQL facile et fournit des insights à plus de 500 utilisateurs professionnels via des rapports et des tableaux de bord
- MLflow accélère le déploiement de nouveaux modèles qui améliorent l'expérience client, avec de nouveaux cas d'usage livrés en moins de deux mois

10x plus rapide
temps de mise sur le marché - cas d'usage déployés en deux mois

+ de 100
cas d'usage à livrer au cours de l'année prochaine

+ de 500
business et utilisateurs IT **responsabilisés**

En savoir plus

databricks

SECTION 3.2

J.B. HUNT

Ce que Databricks nous a vraiment apporté, c'est une base pour le marché du fret digital le plus innovant en nous permettant de tirer parti de l'IA pour offrir la meilleure expérience de transport possible.

Joe Spinelle
Directeur, Ingénierie et technologie,
J.B. Hunt

En savoir plus

S'étant fixé comme mission de construire le réseau de transport digital le plus efficace d'Amérique du Nord, JB Hunt souhaitait rationaliser la logistique du fret et offrir la meilleure expérience de transport. Mais son architecture héritée, son manque de capacités en matière d'IA et son incapacité à gérer en toute sécurité le Big Data ont créé des obstacles importants. Cependant, après avoir mis en œuvre la plateforme Databricks Lakehouse et Immuta, JB Hunt est désormais en mesure de fournir des solutions opérationnelles allant de l'amélioration de l'efficacité de la chaîne d'approvisionnement à l'augmentation de la productivité des conducteurs, ce qui se traduit par d'importantes économies d'infrastructure IT et des gains de chiffre d'affaires.

CAS D'USAGE

JB Hunt utilise Databricks pour fournir une analytique de pointe sur les transporteurs de fret via sa plateforme Carrier 360, réduisant les coûts tout en augmentant la productivité et la sécurité des conducteurs. Les cas d'usage comprennent la logistique du fret, la vue à 360° du client, la personnalisation et bien plus encore.

SOLUTION ET AVANTAGES

JB Hunt utilise la plateforme Lakehouse de Databricks pour créer le marché de fret le plus sûr et le plus efficace d'Amérique du Nord, en rationalisant la logistique, en optimisant les expériences des transporteurs et en réduisant les coûts.

- Delta Lake fédère et démocratise les données pour des optimisations d'itinéraires en temps réel et des recommandations de conducteurs via la plateforme Carrier 360
- Les notebooks améliorent la productivité de l'équipe de données pour fournir plus de cas d'usage, plus rapidement
- MLflow accélère le déploiement de nouveaux modèles qui améliorent l'expérience du conducteur

2,7 M $
d'économies sur l'infrastructure IT, augmentant la rentabilité.

5 %
de hausse des revenus grâce à l'amélioration de la logistique

99,8 % plus rapide
en termes de recommandations pour une meilleure expérience du transporteur

databricks

À propos de Databricks

Databricks est une entreprise axée sur les données et l'intelligence artificielle. Plus de 5 000 entreprises internationales, parmi lesquelles Comcast, Condé Nast, H&M et plus de 40 % des entreprises du Fortune 500, s'appuient sur la plateforme Databricks Lakehouse pour unifier leurs données, leurs capacités d'analytique et d'intelligence artificielle. Databricks possède différents bureaux à travers le monde. Son siège social est basé à San Francisco. Fondé par les créateurs d'Apache Spark™, Delta Lake and MLflow, Databricks a pour mission d'aider les équipes en charge des données à répondre aux problèmes les plus complexes. Pour en savoir plus, suivez Databricks sur Twitter, LinkedIn et Facebook.

COMMENCEZ VOTRE ESSAI GRATUIT

Contactez-nous pour une démonstration personnalisée
databricks.com/contact.

databricks