MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR
DE LA RECHERCHE SCIENTIFIQUE ET DE L'INNOVATION

UNIVERSITÉ SULTAN MOULAY SLIMANE

ECOLE NATIONALE DES SCIENCES APPLIQUÉS DE KHOURIBGA

Mémoire de fin d'études

pour l'obtention du

Diplôme d'Ingénieur d'État

Filière : Informatique et Ingénierie des Données

Présenté par :
Ziyad **RAKIB**

Participation à la migration d'une arichitecture monolithique à une arichitecture microservices

Sous la supervision :
Abdelghani **Ghazdali**
Nidal **Lamghari**

Membres du jury :

Nom et prénoms du président	Entité	Président
Nom et prénoms de l'examinateur	Entité	Examinateur
Nom et prénoms du rapporteur	Entité	Rapporteur
Nom et prénoms du rapporteur	Entité	Rapporteur

Année Académique : **2023**

Sommaire

Dédicace

“

À ma mère, qui m'a comblée de son soutien et dévouée moi avec un amour inconditionnel. Tu es pour moi un exemple de courage et sacrifice continuel. Que cet humble travail porte témoigner de mon affection, de mon attachement éternel et qu'il appelle sur moi ta bénédiction continuelle.,

À mon père, aucune dédicace ne peut exprimer l'amour, l'estime, dévouement et le respect que j'ai toujours eu pour vous. Rien dans le monde vaut les efforts faits jour et nuit pour mon éducation et mon bien-être. Ce travail est le fruit de vos sacrifices que vous avez fait pour mon éducation et ma formation,

À mes chers frères, merci pour votre amour, soutien, et encouragements,

À tous mes chers amis, pour le soutien que vous m'avez apporté, je dis Merci encore une fois à tous ceux qui me sont chers, à vous tous

Merci.

”

- Ziyad

Remerciements

Tout d'abord, je remercie le grand Dieu puissant de nous donné la puissance pour continuer et dépasser toutes les difficultés.

J'adresse mes remerciements à, **Monsieur Abdelghani Ghazdali**, chef de filière Informatique et Ingénierie des Données qui a su assurer le bon déroulement des séances d'encadrement du début à la fin.

Je tiens également à adresser mes sincères remerciements à **Professeur Nidal Lamghari**, mon encadrante interne, pour tous les conseils qu'elle m'a prodigués ainsi que ses encouragements.

Au terme de ce travail, je tiens également à exprimer mes sincères remerciements et ma gratitude envers tous ceux qui, par leur enseignement, par leur soutien et leurs conseils, ont contribué au déroulement de ce projet.

J'adresse mes remerciements à **Monsieur Reda EL Mejad**, PDG et co-fondateur de Izicap et lauréat de l'ENSA, pour l'opportunité de passer mon stage au sein de cet entreprise.

Je tiens à remercier mon encadrant à Izicap, **Monsieur Bilal SLAYKI** qui m'a supervisée avec patience et n'a épargné aucun effort pour mettre à ma disposition les explications nécessaires et les directives précieuses. Son assistance et ses conseils permanents m'ont été un apport remarquable. Je veux remercier aussi Nasr, Youness et Anas; l'équipe avec laquelle j'ai travaillé et qui ont généreusement contribué à ce travail.

Je remercie également l'agente RH **Madame Sanaa ARROUCHE** pour ses bons conseils car elle est toujours là pour accéder à nos demandes et répondre au mieux à nos questions.

Un remerciement particulier aux membres du jurys qui m'ont honoré en acceptant de juger ce travail et de me faire profiter de leurs remarques et conseils.

Finalement, mes remerciements s'adressent aussi à l'ensemble du corps professoral et administratif de l'ENSA Khouribga pour l'effort qu'ils fournissent afin de nous garantir une bonne formation et à l'équipe administrative et technique pour tous les services offerts.

Résumé

Ce rapport fournit une analyse approfondie du projet complexe entrepris pour faire évoluer la source de données de Izicap. Le projet a nécessité la suppression de MariaDB et la mise en place de Delta Lake, une solution de stockage et de traitement de données hautes performances. De plus, l'architecture monolithique existante a été divisée en microservices utilisant Spring Boot comme backend avec un connecteur approprié pour Delta Lake, et les micro-frontends ReactJS comme frontend.

Le projet présentait de nombreux défis qui nécessitaient l'application de technologies et de stratégies avancées. Celles-ci comprenaient la migration des données, la garantie de la cohérence et de l'exactitude des données, la gestion de la complexité des systèmes distribués et l'intégration de diverses technologies et services. Le rapport décrit les différentes solutions développées pour relever ces défis, notamment l'utilisation d'algorithmes de traitement de données avancés, d'architectures informatiques distribuées et de la conteneurisation.

Malgré les défis rencontrés, le projet reste un travail en cours. Le rapport donne un aperçu des efforts en cours pour améliorer l'évolutivité, les performances et la fonctionnalité globale du projet.

Mots clés : Projet, Source de données, Delta Lake, Monolith, Microservice, Spring Boot, ReactJS, Micro-Frontend, Back-End, Front-End, Données, Évolutivité, Performance.

Abstract

This report provides an in-depth analysis of the complex project undertaken to scale the data source of Izicap. The project required the removal of MariaDB and implementation of Delta Lake, a high-performance data storage and processing solution. In addition, the existing monolithic architecture was divided into microservices utilizing Spring Boot as the backend with a suitable connector to Delta Lake, and ReactJS micro-frontends as the frontend.

The project presented numerous challenges that required the application of advanced technologies and strategies. These included data migration, ensuring data consistency and accuracy, managing the complexities of distributed systems, and integrating various technologies and services. The report outlines the various solutions developed to address these challenges, including the use of advanced data processing algorithms, distributed computing architectures, and containerization.

Despite the challenges encountered, the project remains a work in progress. The report provides insights into the ongoing efforts to improve the project's scalability, performance, and overall functionality.

Keywords: Project, Data Source, Delta Lake, Monolith, Microservice, Springboot, ReactJS, Micro-frontends, Back-End, Front-End, Data, Scalability, Performance.

ملخص

يقدم هذا التقرير تحليلاً متعمقًا للمشروع المعقد الذي تم إجراؤه لتوسيع نطاق مصدر بيانات الشركة. يتطلب المشروع إزالة MariaDB وتنفيذ Delta Lake ، وهو حل تخزين ومعالجة بيانات عالي الأداء. بالإضافة إلى ذلك ، تم تقسيم المونوليث الحالي إلى خدمات مصغرة باستخدام Spring Boot كواجهة خلفية مع موصل مناسب لـ Delta
Lake ، وواجهة ReactJS كواجهة أمامية. قدم المشروع العديد من التحديات التي تطلبت تطبيق التقنيات والاستراتيجيات المتقدمة. وشمل ذلك ترحيل البيانات ، وضمان اتساق البيانات ودقتها ، وإدارة تعقيدات الأنظمة الموزعة ، ودمج التقنيات والخدمات المختلفة. يحدد التقرير الحلول المختلفة التي تم تطويرها لمواجهة هذه التحديات ، بما في ذلك استخدام خوارزميات معالجة البيانات المتقدمة ، وبنى الحوسبة الموزعة ، والحاويات. على الرغم من التحديات التي واجهتها ، لا يزال المشروع قيد التنفيذ. يقدم التقرير رؤى حول الجهود الجارية لتحسين قابلية المشروع للتوسع والأداء والوظائف العامة

كلمات مفتاحية : المشروع، مصدر البيانات، Delta Lake ، المونوليث، الميكروسيرفس، Springboot ، ReactJS ، الميكروواجهات، الخلفية، الواجهة الأمامية، البيانات، القابلية للتوسع، الأداء. Delta Lake ، Springboot ، ReactJS ، الميكروواجهات.

Table des matières

Table des figures

Liste des tableaux

TABLE 1 : List of Abbreviations

API	Application programming interface
VSC	Visual Studio Code
DL	Delta Lake
REST	RepresEntational State Transfer
AWS	Amazon Web Services
GCP	Google Cloud Platform
AZR	Microsoft Azure
S3	Simple Storage Service
IAM	Identity and Access Management
MinIO	Minimal Object Storage
SQL	Structured Query Language
DB	Database
ACID	Atomicity, Consistency, Isolation, Durability
OLTP	Online Transaction Processing
OLAP	Online Analytical Processing
ReactJS	React JavaScript
SPA	Single Page Application
JS	JavaScript
CSS	Cascading Style Sheets
HTML	Hypertext Markup Language
UI	User Interface
API	Application Programming Interface
Spark	Apache Spark
Hadoop	Apache Hadoop
HDFS	Hadoop Distributed File System
YARN	Yet Another Resource Negotiator
MapReduce	MapReduce Programming Model
Metadata	Data about Data
ETL	Extract, Transform, Load
ELT	Extract, Load, Transform
CRM	Customer Relationship Management
RDBMS	Relational Database Management System
SPI	Server Provider Interface

Introduction Générale

Le monde des données continue de se développer à un rythme sans précédent. Cela a conduit au besoin de systèmes de stockage et de gestion de données efficaces capables de suivre cette croissance. Dans notre quête d'une meilleure solution, nous avons rencontré le problème de l'évolutivité avec notre système de gestion de base de données actuel, MariaDB. En conséquence, nous nous sommes lancés dans un projet visant à trouver une source de données évolutive qui répondrait à nos besoins. Après de nombreuses recherches et réflexions, nous avons décidé d'utiliser Delta Lake avec un stockage S3.

Cependant, nous avons rencontré plusieurs défis lors de la mise en œuvre de Delta Lake. L'un des principaux problèmes auxquels nous avons été confrontés était le manque de popularité et de documentation de Delta Standalone. Il s'est également avéré lent et ne pouvait pas être interrogé avec SQL. Par conséquent, nous avons opté pour Trino, un moteur de requête SQL distribué qui pourrait facilement interagir avec Delta Lake.

Un autre défi important était l'architecture monolithique de notre système, qui entravait sa flexibilité et son évolutivité. Par conséquent, nous avons décidé de passer aux microservices pour rendre notre système plus efficace et flexible.

Dans ce rapport, nous donnerons un aperçu de la société Izicap au chapitre 1, suivi d'une explication de Delta Lake et de ses défis de mise en œuvre au chapitre 2. Au chapitre 3, nous discuterons de Trino et de la manière dont il nous a aidés à surmonter les défis auxquels nous étions confrontés. avec le lac Delta.

Chapitre 1

Izicap

Introduction

1.1 About

Izicap est un pionnier dans l'utilisation des données de cartes de paiement et les transforme en une connaissance client et des informations commerciales puissantes pour les commerçants, leur permettant de gérer leurs propres programmes de fidélité et campagnes de marketing numérique. Ces services marketing, fournis par les acquéreurs utilisant les solutions SaaS d'Izicap, redonnent rapidement de la croissance aux entreprises marchandes en renforçant les dépenses et la fidélité de leurs clients (les titulaires de carte). La solution innovante de CRM et de fidélité liée aux cartes d'Izicap donne aux acquéreurs un avantage concurrentiel en monétisant leurs données de transactions de paiement, en générant de nouvelles sources de revenus et en améliorant leurs capacités de rétention. Après s'être solidement implanté en France grâce à des partenariats avec le Groupe BPCE et le Crédit Agricole, Izicap s'est associé à Nexi, le premier acquéreur et Fintech en Italie et a rejoint le programme StartPath de Mastercard dans le but d'étendre considérablement sa portée mondiale. Izicap s'associe aux principaux fournisseurs de solutions de paiement tels qu'Ingenico, Verifone, Poynt et PAX, et rend sa solution CRM et Fidélité liée à la carte disponible sur les terminaux de paiement les plus populaires et les plus innovantes.

1.2 Organigramme

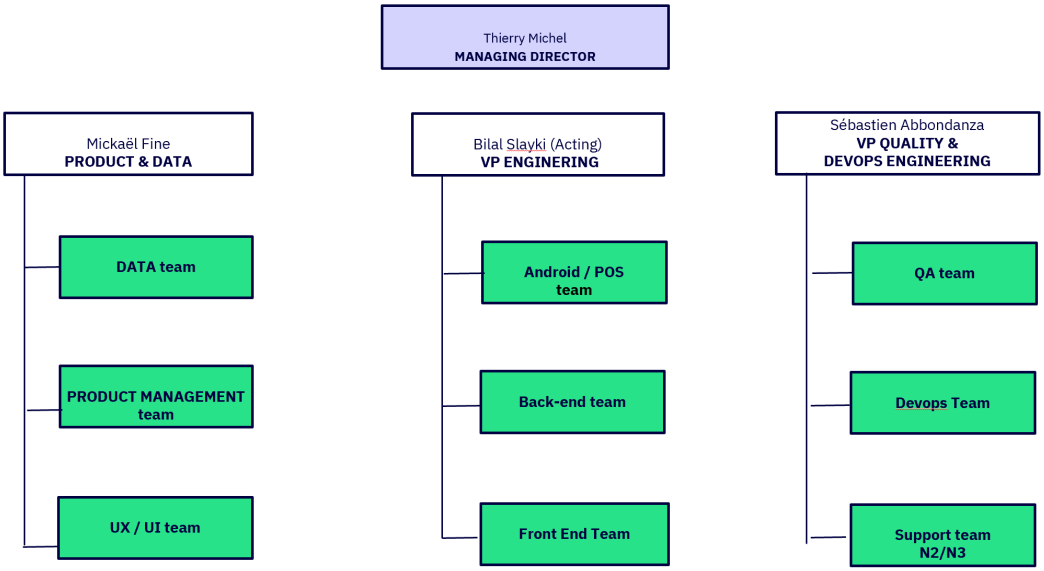

FIGURE 1.1 : Organigramme de Izicap

Conclusion

Chapitre 2

Delta Lake

Introduction

Dans ce chapitre, nous explorerons la schématisation détaillée du projet visant à remplacer notre architecture monolithique par une approche basée sur des microservices et des micro-frontends, tout en migrer notre système de gestion de base de données de MariaDB vers Delta Lake. Cette transition marque une étape cruciale dans notre parcours d'évolution technologique, offrant des avantages significatifs en termes de flexibilité, d'évolutivité et de maintenabilité.

2.1 Smart Data

La smart data fait référence au processus d'extraction d'informations précieuses à partir de grandes quantités de données afin de prendre des décisions commerciales éclairées.

La smart data nous permet de :

1. **Collecte des données** : Izicap aide les entreprises à collecter des données clients à partir de différents points de contact tels que les systèmes de point de vente, les programmes de fidélité, les interactions en ligne, etc. Assurez-vous d'intégrer leurs outils de collecte de données dans vos systèmes ou processus existants.

2. **Analyse des données** : La solution de smart data d'Izicap vous permet d'analyser les données collectées pour découvrir des tendances, des modèles et des comportements clients. Utilisez leurs outils d'analyse et leurs algorithmes pour obtenir des informations exploitables à partir des données.

3. **Segmentation des clients** : Segmentez votre base de clients en fonction de leurs préférences, de leur historique d'achat, de leurs caractéristiques démographiques ou d'autres critères pertinents. La solution de smart data d'Izicap peut vous aider à identifier différents segments de clients et à créer des stratégies marketing personnalisées pour chaque segment.

4. **Marketing personnalisé** : Exploitez les informations tirées de la solution de smart data d'Izicap pour créer des campagnes marketing ciblées. Envoyez des offres personnalisées, des promotions ou des recommandations à des segments spécifiques de clients, augmentant ainsi les chances de conversion et de satisfaction client.

5. **Fidélisation de la clientèle** : Utilisez la solution de smart data d'Izicap pour identifier les clients qui risquent de résilier leur abonnement ou de ne plus acheter chez vous. Développez des stratégies de fidélisation en leur offrant des incitations, des programmes de fidélité ou des communications personnalisées afin de les maintenir engagés et fidèles à votre marque.

6. **Suivi des performances** : Surveillez régulièrement les performances de vos campagnes marketing et de vos efforts d'engagement client. La solution d'Izicap peut vous fournir des métriques et des rapports pour évaluer l'efficacité de vos stratégies et apporter des ajustements basés sur les données lorsque nécessaire.

7. **Amélioration continue** : Les solutions de smart data sont les plus efficaces lorsqu'elles sont utilisées de manière itérative. Analysez régulièrement les données, adaptez vos stratégies et peaufinez votre approche en fonction de nouvelles informations et de l'évolution du comportement client.

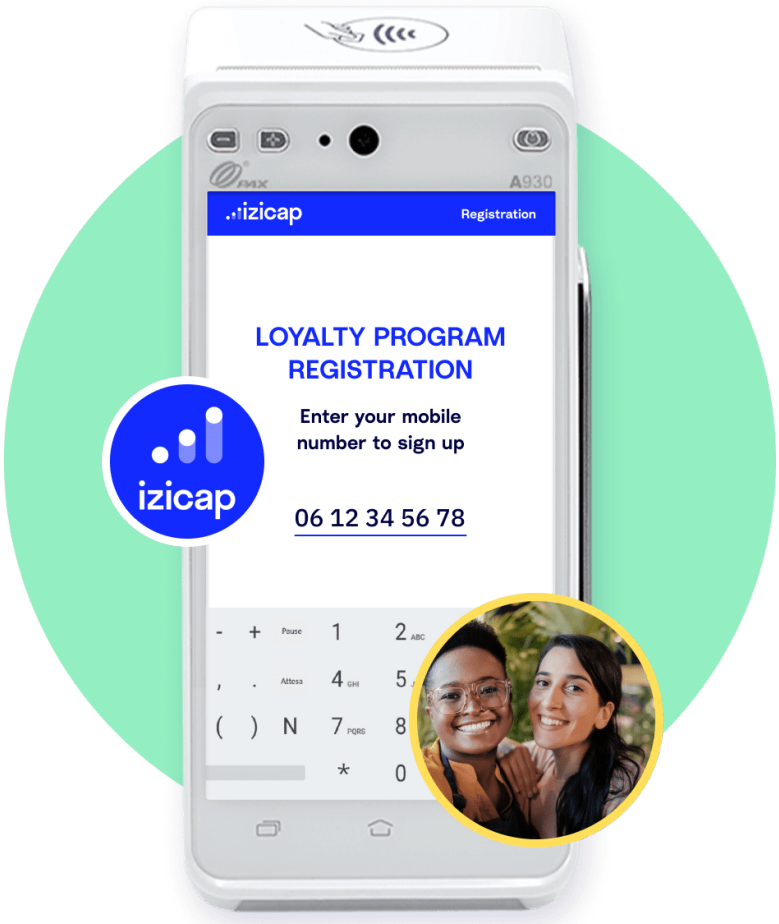

FIGURE 2.1 : Smart data Izicap

2.2 L'architecture du système

La figure ci-dessous visualise la structure et les composants clés de notre système, ainsi que les interactions entre eux, elle constitue ainsi le fondement de notre système et joue un rôle essentiel dans la fourniture de fonctionnalités et de services à nos utilisateurs.

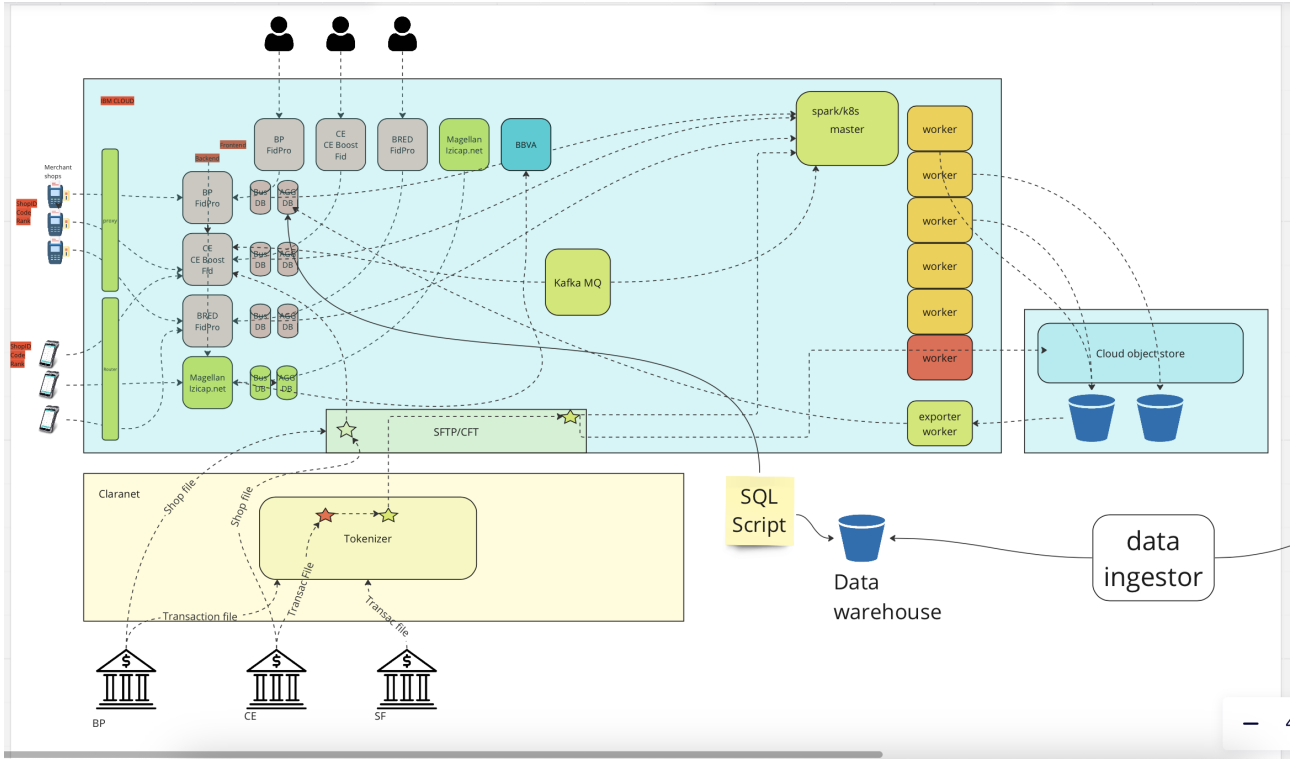

FIGURE 2.2 : Architecture du système actuelle

Ce schéma représente le fonctionnement de l'architecture monolithique actuelle qui présente plusieurs inconvénients qui peuvent entraver l'efficacité, la flexibilité et la maintenabilité du système. Voici une description détaillée des principaux inconvénients :

1. **Complexité et dépendances** : L'architecture monolithique implique que tous les modules et fonctionnalités du système sont regroupés en un seul bloc. Cela crée une forte interdépendance entre les différents composants, rendant la compréhension, la gestion et les mises à jour complexes. Les modifications apportées à une partie du système peuvent avoir des répercussions sur d'autres parties, ce qui rend les tests, les déploiements et les corrections d'erreurs plus difficiles.

2. **Scalabilité limitée** : L'architecture monolithique a souvent des difficultés à s'adapter à une augmentation de la charge ou à une demande croissante. Étant donné que tous les composants sont regroupés dans un seul monolithe, il est difficile de faire évoluer sélectivement une partie spécifique du système. Cela peut entraîner des goulots d'étranglement, des performances réduites et une mauvaise répartition des ressources lorsqu'il s'agit de traiter des volumes de données importants ou de gérer une augmentation du nombre d'utilisateurs.

3. **Déploiements complexes et risques élevés** : Avec une architecture monolithique, les déploiements nécessitent la mise à jour de l'ensemble du système, même pour des modifications mineures. Cela augmente les risques d'erreurs et de régressions, car une seule erreur peut entraîner l'indisponibilité du système tout entier. De plus, les déploiements doivent être soigneusement planifiés et coordonnés, ce qui peut entraîner des temps d'arrêt plus longs et des interruptions de service pour les utilisateurs.

4. **Difficulté de choix technologiques** : Dans une architecture monolithique, les technologies utilisées sont souvent liées et intégrées de manière étroite. Cela peut rendre difficile l'adoption de nouvelles technologies ou l'intégration de composants spécialisés. Les mises à niveau de versions ou les ajouts de nouvelles fonctionnalités peuvent être limités par les choix technologiques initiaux, ce qui peut entraver l'innovation et l'adaptation aux évolutions du marché.

5. **Cohésion et responsabilités** : L'architecture monolithique ne facilite pas une séparation claire des responsabilités et des fonctionnalités. Les différents modules du système sont souvent intimement liés et peuvent partager des fonctionnalités communes. Cela rend difficile l'isolation des problèmes, la maintenance spécifique des modules et la réutilisation de code spécifique à un domaine.

Conclusion

Delta Lake est un outil important pour le traitement du Big Data, fournissant une gestion fiable des données et garantissant l'intégrité des données à grande échelle. Ses transactions ACID, l'application des schémas et les fonctionnalités de gestion des versions des données en font un choix populaire pour les entreprises qui doivent traiter de grandes quantités de données avec une grande précision et fiabilité.

En utilisant Delta Lake, les ingénieurs de données et les scientifiques des données peuvent facilement gérer la qualité des données, suivre la lignée des données et collaborer sur des projets d'analyse de données. Grâce à son intégration transparente avec d'autres outils de Big Data. Delta Lake fournit une solution puissante pour le traitement du Big Data qui peut aider les entreprises à mieux comprendre leurs données plus rapidement et plus efficacement.

Chapitre 3

Technologies Utilisées

Introduction

Dans ce chapitre, nous examinerons en détail les technologies clés qui sont utilisées dans notre solution pour fournir des fonctionnalités avancées et répondre aux besoins spécifiques de notre projet. Les trois principales technologies que nous aborderons sont Delta Lake, Trino et les Microfrontends.

3.1 Delta Lake

Delta Lake est une technologie de gestion des données qui permet de stocker, gérer et analyser des volumes massifs de données de manière efficace et fiable. Il repose sur une architecture basée sur des fichiers parquet et offre des fonctionnalités avancées telles que la gestion des transactions ACID (Atomicité, Cohérence, Isolation, Durabilité) et la compatibilité avec des outils d'analyse populaires. Delta Lake garantit également l'intégrité des données, la cohérence des requêtes et la prise en charge de la réplication et de la récupération en cas de défaillance.

Le concept de "lakehouse" est rendu possible grâce à Delta Lake. Il s'agit d'une architecture de données qui combine les avantages des entrepôts de données et des lacs de données, en offrant une approche unique et cohérente pour la gestion des données. Les données sont stockées au format Parquet dans le lac de données, permettant ainsi un traitement continu et par lots.

- **Permet une architecture Lakehouse :** Delta Lake permet une architecture de données continue et simplifiée qui permet aux organisations de gérer et de traiter d'énormes volumes de données en continu et par lots sans les tracas de gestion et d'exploitation liés à la gestion séparée du streaming, des data warehouses et des lacs de données.
- **Permet une gestion intelligente des données pour les lacs de données :** Delta Lake offre une gestion efficace et évolutive des métadonnées, qui fournit des informations sur les volumes de données massifs dans les lacs de données. Grâce à ces informations, les tâches de gouvernance et de gestion des données se déroulent plus efficacement.

- **Application du schéma pour une meilleure qualité des données :** Étant donné que les lacs de données n'ont pas de schéma défini, il devient facile pour les données mauvaises/incompatibles d'entrer dans les systèmes de données. La qualité des données est améliorée grâce à la validation automatique du schéma, qui valide la compatibilité DataFrame et table avant les écritures.
- **Permet la transaction ACID :** La plupart des architectures de données organisationnelles impliquent de nombreux mouvements ETL et ELT entrant et sortant du stockage de données, ce qui l'ouvre à plus de complexité et d'échecs aux points d'entrée des nœuds. Delta Lake garantit la durabilité et la persistance des données pendant l'ETL et d'autres opérations de données. Delta Lake capture toutes les modifications apportées aux données pendant les opérations de données dans un journal des transactions, garantissant ainsi l'intégrité et la fiabilité des données pendant les opérations de données.

3.2 Avantages de Delta Lake

Avec Delta Lake, les données sont stockées dans un format optimisé, tel que Parquet, dans un lac de données. Ce format facilite le traitement efficace des requêtes, quel que soit le mode d'accès aux données, qu'il s'agisse d'un traitement streaming ou par batch.

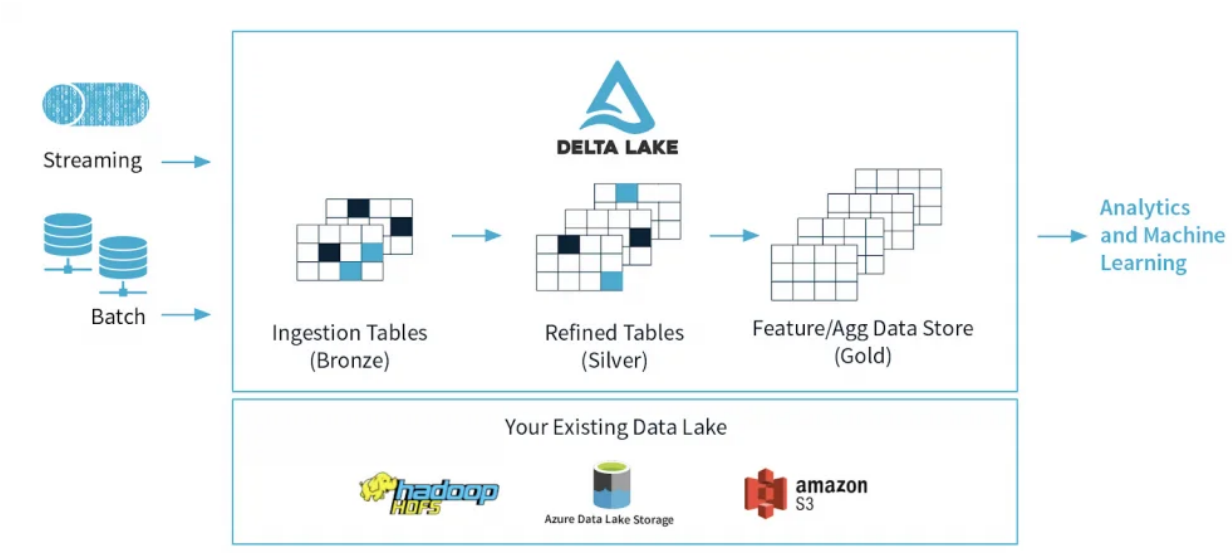

FIGURE 3.1 : Architecture multi-sauts de Delta Lake

3.3 Principaux avantages et caractéristiques de Delta Lake

- **Pistes d'audit et historique :**Dans Delta Lake, chaque écriture existe en tant que transaction et est enregistrée en série dans un journal des transactions. Par conséquent, toutes les modifications ou validations apportées au journal des transactions sont enregistrées, laissant une trace

complète à utiliser dans les audits historiques, la gestion des versions ou à des fins de voyage dans le temps. Cette fonctionnalité de Delta Lake permet de garantir l'intégrité et la fiabilité des données pour les opérations de données d'entreprise.

- **Voyager dans le temps et versionner les données :** Étant donné que chaque écriture crée une nouvelle version et stocke l'ancienne version dans le journal des transactions, les utilisateurs peuvent afficher/restaurer les anciennes versions de données en fournissant l'horodatage ou le numéro de version d'une table ou d'un répertoire existant à l'API de lecture Sparks. À l'aide du numéro de version fourni, Delta Lake construit ensuite un instantané complet de la version avec les informations fournies par le journal des transactions. Les retours en arrière et la gestion des versions jouent un rôle essentiel dans l'expérimentation de l'apprentissage automatique, où les scientifiques des données modifient de manière itérative les hyperparamètres pour former des modèles et peuvent revenir aux modifications si nécessaire.

- **Unifie le traitement par lots et par flux :** Chaque table d'un Delta Lake est un puits de lot et de flux. Avec le streaming structuré Sparks, les organisations peuvent diffuser et traiter efficacement les données de streaming. De plus, grâce à la gestion efficace des métadonnées, à la facilité d'évolutivité et à la qualité ACID de chaque transaction, l'analyse en temps quasi réel devient possible sans utiliser une architecture de données à deux niveaux plus compliquée.

- **Gestion efficace et évolutive des métadonnées :** Delta Lakes stocke les informations de métadonnées dans le journal des transactions et exploite la puissance de traitement distribuée de Spark pour traiter rapidement, lire et gérer efficacement de gros volumes de métadonnées de données, améliorant ainsi la gouvernance des données.

- **transactions ACID :** Delta Lakes garantit que les utilisateurs voient toujours une vue de données cohérente dans une table ou un répertoire. Il garantit cela en capturant chaque modification effectuée dans un journal de transactions et en l'isolant au niveau d'isolation le plus fort, le niveau sérialisable. Au niveau sérialisable, chaque opération existante a et suit une séquence en série qui, lorsqu'elle est exécutée une par une, fournit le même résultat que celui indiqué dans le tableau.

- **Opérations du langage de manipulation de données :** Delta Lakes prend en charge les opérations DML telles que les mises à jour, les suppressions et les fusions, qui jouent un rôle dans les opérations de données complexes telles que la capture de données de modification (CDC), les upserts en continu et la dimension à évolution lente (SCD). Des opérations comme CDC assurent la synchronisation des données dans tous les systèmes de données et minimisent le temps et les ressources consacrés aux opérations ELT. Par exemple, en utilisant le CDC, au lieu d'ETL toutes les données disponibles, seules les données récemment mises à jour depuis la dernière opération subissent une transformation.

- **Schema Enforcement :** Delta Lakes effectue une validation automatique du schéma en vérifiant un ensemble de règles pour déterminer la compatibilité d'une écriture d'un DataFrame vers une table. L'une de ces règles est l'existence de toutes les colonnes DataFrame dans la table cible. Une occurrence d'une colonne supplémentaire ou manquante dans le DataFrame génère

une erreur d'exception. Une autre règle est que le DataFrame et la table cible doivent contenir les mêmes types de colonnes, ce qui, sinon, déclenchera une exception. Delta Lake utilise également DDL (Data Definition Language) pour ajouter explicitement de nouvelles colonnes. Cette fonctionnalité de lac de données permet d'éviter l'ingestion de données incorrectes, garantissant ainsi une qualité élevée des données.

- **Compatibilité avec l'API de Spark :** Delta Lake est basé sur Apache Spark et est entièrement compatible avec l'API Spark, qui permet de créer des pipelines de données volumineuses efficaces et fiables.
- **Flexibilité et intégration :** Delta Lake est une couche de stockage open source et utilise le format Parquet pour stocker des fichiers de données, ce qui favorise le partage de données et facilite l'intégration avec d'autres technologies et stimule l'innovation.

3.4 Trino

Trino, anciennement connu sous le nom de Presto, est un moteur de requêtes SQL distribué et open-source. Il est conçu pour exécuter des requêtes interactives et analytiques à grande échelle sur des données hétérogènes et distribuées. Trino offre une grande polyvalence en permettant l'accès à différents types de sources de données, qu'il s'agisse de bases de données relationnelles, de systèmes de fichiers, de sources de données en temps réel ou de services de stockage cloud. Grâce à sa conception distribuée, Trino permet des performances élevées et une scalabilité horizontale, ce qui en fait un outil essentiel pour l'analyse des données dans notre solution.

Conclusion Générale