efrei PARIS	**RAPPORT DE STAGE**	**M1 / M2**

ELEVE

Nom : THOIREY Prénom : Romane Filière : Big Data et Machine Learning

SUJET

Stage Technique M2 en tant que Data Engineer à la Société Générale

ENTREPRISE

Nom : Société Générale

Adresse : 29 Bd Haussmann, 75009 Paris

Adresse du lieu du stage si différent : 5&7 Avenue du Val de Fontenay, 94120 Fontenay-sous-Bois

DATE DU STAGE Date de remise du rapport aux membres du Jury : 16/09/2021

du : 01/03/2021 au : 27/08/2021 durée effective en semaines : 26 semaines

SOUTENANCE Date : 23/09/2021 Heure : **18h30**

Composition du Jury :

- Président (responsable EFREI) : Manar BADDOUR
- Responsable du stage (Entreprise) : Fréderic KRANTZ
- Invité(e) :

PUBLICATION DU RAPPORT DE STAGE

Le Responsable du stage autorise le stagiaire à publier le rapport de stage sur l'Intranet de l'Ecole.
Signature

Mots clés

Big Data, Spark, Scala, HDFS, Streaming, S3, API, UI

Table des matières

Remerciements

Dans un premier temps, je souhaite remercier la Société Générale pour m'avoir permis d'effectuer mon stage de fin d'étude de M2.

Je remercie tout particulièrement mon maître de stage Fréderic KRANTZ, pour m'avoir fait confiance et pour m'avoir confié une telle mission. Il a su prendre le temps et faire preuve d'écoute durant les périodes où tout n'était pas clair pour moi techniquement.

Un grand merci pour Othman SEFRAOUI, pour son temps accorder à m'expliquer la partie fonctionnelle du projet. Nous avons pu compter l'un sur l'autre pour avancer dans le projet et nous aider mutuellement.

Merci à Sandeep RAMANATH, pour sa grande disponibilité malgré la distance et son aide au sein du projet.

Enfin, je remercie toute l'équipe BDE et d'autres collègues au sein de la Société Générale que j'ai rencontré durant mon stage et qui m'ont apporté aussi bien techniquement qu'humainement.

Introduction

Actuellement en 3ème année du cycle ingénieur dans la majeure Big Data et Machine Learning, j'ai réalisé dans le cadre de ma formation d'ingénieur à l'EFREI (École d'ingénieurs généraliste en informatique et technologies du numérique) un stage technique à la **Société Générale**.

La majeure Big Data et Machine Learning forme les étudiants à répondre aux problématiques spécifiques au secteur de la donnée et à développer des solutions logicielles adaptées. Ce stage a été réalisé au sein de la Société Générale et plus précisément au sein de l'entité **DDS (Digital & Data Service)** pendant une durée de 26 semaines. Ce stage de fin d'étude a pour but de présenter aux élèves ingénieurs un environnement technique professionnel ainsi que les différentes méthodes appliquées par l'entreprise. Selon moi, il est important de finir par un stage technique afin de pouvoir être confronté dès le début aux différentes problématiques technologiques pour pouvoir par la suite apporter son expérience dans ce domaine bien que l'on ne se destine pas à faire uniquement un métier technique durant sa carrière d'ingénieur.

Ce stage constitue pour moi, une occasion d'acquérir une expérience professionnelle supplémentaire et de me permettre également d'élargir mon réseau professionnel.
J'ai choisi cette entreprise car j'ai pensé que c'était une bonne occasion pour développer mes compétences dans le domaine de la donnée, ayant choisi la filière Big Data. En outre, une entreprise de cette taille m'a paru être un bon choix en raison de la quantité des données qu'elle est amenée à traiter.

Ainsi, au-delà de l'enrichissement de mes connaissances, ce stage m'a permis d'en apprendre plus sur les relations professionnelles en entreprise et le travail en équipe et de m'adapter à cette nouvelle situation à laquelle nous faisons face avec la COVID-19. En effet aux vues de la situation actuelle, j'ai dû apprendre à m'adapter pour travailler à distance et contacter mon équipe via les outils mis à disposition.

Ce rapport montrera dans un premier temps le contexte de mon stage, avec une présentation de **l'entreprise**, **de son fonctionnement et son environnement métier** afin de comprendre le milieu dans lequel j'ai évolué au cours de ces 6 mois. Puis dans un second temps, j'exposerai **l'aspect technique du stage avec une présentation des outils utilisés ainsi qu'une étude critique de mes missions principales**. Enfin je me concentrerai sur **l'intégration et les notions importantes** que j'ai pu tirer de ces travaux.

1. L'entreprise

1) Historique de la Société Générale

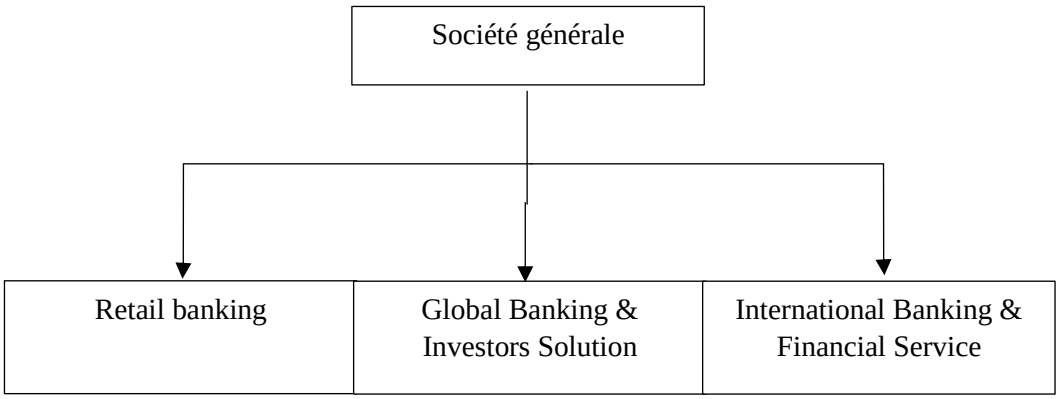
Fondée par un groupe d'entrepreneurs en 1864 "pour favoriser le développement du commerce et de l'industrie en France", la Société Générale assume pleinement sa mission d'accompagnement des entreprises.
Elle a été nationalisée en 1945 puis reprivatisée en 1987. Elle constitua la première banque d'envergure importante au niveau français et européen.

Aujourd'hui encore, la Société Générale cultive cet esprit d'entreprendre pour construire l'avenir. Société Générale est l'un des tous premiers groupes européens de services financiers. S'appuyant sur un modèle diversifié et intégré, il s'allie sur une solidité financière, dynamique d'innovation et stratégie de croissance durable afin d'être le partenaire de confiance de ses clients, engagé dans les transformations positives du monde.
Le modèle adopté est celui d'une banque universelle avec cinq métiers complémentaires, celui de **Banque de détail en France**, celui de **Banque de détail et services financiers internationaux**, celui de **Banque de financement et d'investissement**, **Banque privée**, **Gestion d'actifs** et enfin **Métiers titres**. Au total, le Groupe comptabilise environ 150 000 collaborateurs présents dans 76 pays ainsi que 31 millions de clients particuliers, professionnels, entreprises et institutions financières.

2) Société Générale aujourd'hui

Aujourd'hui, la Société Générale est solidement installée avec 31 millions de clients particuliers, entreprises et investisseurs institutionnels, un capital de 29,06 milliards d'euros en 2019. Le groupe est séparé en 3 pôles métiers complémentaires, à savoir :

- La banque de détail en France (Société Générale Banque & Assurance, Crédit du Nord, Boursorama Banque)
- La banque de détail et les services financiers internationaux
- La banque de financement et d'investissement (La Banque de Grande Clientèle & Solutions Investisseurs (GBIS) accompagne les grandes entreprises, les PME & ETI, les institutions financières, le secteur public, ainsi que les Family Offices et clients privés).



3) Les concurrents

Les principaux concurrents de la Société Générale sont :

- BNP Paribas,
- Crédit Agricole,
- LCL

Tout comme la Société Générale, ces banques bénéficient d'une présence importante en France.
D'autres concurrents à la banque sont les néo-banques, qui sont des établissements de paiement ayant reçu une licence bancaire. Parmi celles-ci on compte par exemple :

- N26,
- Revolut,
- Nickel

Leur particularité est leur accès 100% en ligne. De plus, elles proposent des produits gratuits comme la carte bancaire, frais de transaction. Cependant, elles ne permettent pas toujours le dépôt d'espèces ou de chèques. Banques et néo-banques ne répondent donc pas aux mêmes besoins.

4) Organisation du pôle Data

La Société Générale est organisée en entités autrement appelées **Business Units (BU)** et **Service Units (SU)**. Les BU sont responsables de la définition de leur stratégie et de leur exécution, elles représentent des périmètres de métiers dans lesquels les décisions opérationnelles sont concentrés afin de servir de façon plus agile le client.
Durant mon stage j'ai travaillé au sein d'une des Services Units. Les SU sont au service des ambitions des Business Units et des clients du Groupe, elles sont alignées avec leur fonctionnement, les Services Units assurent des fonctions de support et de contrôle. Voici la liste des Service Units :

- CPLE (Compliance)
- DFIN (Direction Financière du développement)
- GBSU (Global Business Service Unit)
- DRHG/COMM (Direction des Ressources Humaines du Groupe & Communication du Groupe)
- IGAD (Inspection Générale et Audit)
- ITIM (Innovation, Technologies & Informatique)
- RESG (Direction des Ressources et de la Transformation Numérique du Groupe)
- RISQ (Direction des Risques du Groupe)
- SEGL (Secrétariat Général du Groupe)

Ces Directions Centrales sont organisées en filières transversales à travers les piliers et les principales filiales du Groupe.
Durant mon stage, j'ai eu l'occasion de travailler parmi les équipes RESG dans le Département **Digital & Data Service** (RESG/DDS). Les équipes RESG accompagne la transformation numérique de la Société Générale et contribue à développer l'efficacité opérationnelle du Groupe. La première mission de RESG est d'assurer la production de services mutualisés et optimisés autour de 5 métiers :

- Les Systèmes d'information
- Les Achats
- L'Immobilier
- Le Conseil et Transformation
- Les Centres de services partagés.

Dans cette entité DDS, il existe le pôle **RESG/DDS/DAT/BDX** (Big Data Expérience) qui garantit les fonctions de Centre de Solutions transverses. Ses missions sont focalisées sur une offre de service tout autour de la donnée : Big Data, référentiels, Data Quality, distributions de données, et regroupent les activités d'innovation de DAT.

Au sein de l'entité DAT, j'ai fait mon stage dans l'équipe **RESG/DDS/DAT/BDX/BDE** (Big Data Expertise). Elle a pour principal objectif l'animation de la communauté Tech Lead sur le Big Data, notamment les réunions hebdomadaires du comité Tech Lead qui sont des réunions d'échange sur le thème du Big Data. L'équipe BDE participe également à des ateliers, des présentations ou bien des interventions d'externe pour former les techs leads et fédérer la communauté (Intervenant de chez Flink, Kafka, Databricks…).

L'entité BDE fait également office de support niveau 3 sur les applications concernant le Big Data. Le niveau 3 étant le dernier support en termes d'aide lorsque des problèmes sont rencontrés sur les applications de Big Data. Ils travaillent également sur les architectures du Big Data avec les architectes qui ne sont pas spécialisés dans la Data. Ils participent à la validation de profils qui souhaitent rejoindre la communauté Big Data lors de recrutement.

2. Contexte du stage et objectifs

1) Contexte de la mission

Les clusters informatiques de calcul en mémoire ont pris de l'ampleur ces dernières années, en raison de leur capacité à analyser de grandes quantités de données en parallèle. Ces plateformes sont des environnements complexes et difficiles à gérer. De plus, il y a un manque d'outils pour mieux comprendre et optimiser ces plateformes qui forment la base des technologies de Big Data.

Cela conduit directement à une sous-utilisation des ressources disponibles dans un tel environnement. L'un des aspects clés qui peut résoudre ce problème est l'optimisation du parallélisme des tâches d'application dans de tels environnements. Afin de relever les défis du Big Data, de nombreux frameworks de programmation parallèle, comme Map Reduce, Apache Spark, ou Flink ont été développés.

Dans le cadre de mon stage, j'ai travaillé sur le projet **AOD** (Application Optimisation Dashboard). À terme, l'objectif du projet **AOD** est de pouvoir offrir à la communauté des Data engineers de la Société Générale tout une panoplie de services pour optimiser, et profiler leurs applications de Data Processing.

Le but est de leur offrir des outils pour l'optimisation de l'utilisation des clusters et un Dashboard. Ils pourront visualiser les performances de leurs applications en termes de ressources et de temps, le coût en termes de performance pour chaque partie de leurs applications, et tout cela, afin de pouvoir cibler les parties qui consomment le plus et ainsi les optimiser.

Nous avons concentré nos travaux sur l'optimisation et le monitoring de jobs **Spark**, qui est un framework open source de calcul distribué. Il existe déjà une interface utilisateur de Spark qui permet aux développeurs de débugger leurs applications. Sur cette interface, il est possible de retrouver plusieurs informations à propos des applications, notamment celles qui échouent, ou des informations telles que la durée et les ressources qu'elle a utilisées.
Cependant l'interface de Spark n'est pas correctement exploitée par les développeurs, en particulier lorsqu'il s'agit d'une grosse application avec plusieurs jobs. L'interface devient très difficile à comprendre et s'il y a une information critique, elle est perdue derrière un grand nombre d'information.

Nous avons donc développé une application web qui a pour but de proposer de nouvelles visualisations pour faciliter la tâche aux développeurs. Des visualisations qui permettent de répondre aux questions que se posent un développeur comme :

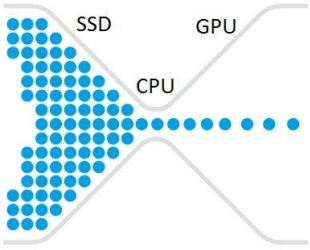
- Quel est le stage/job qui prend le plus de temps dans le code ?
- Quel est le Bottleneck de l'application ?

C'est dans le but de répondre à ces questions que nous avons implémenter l'outil de recommandation et de monitoring **AOD**.

2) Les termes techniques

Définition technique

Les **Bottlenecks**, autrement appelés « Goulot d'étranglement » sont des points de système où des instructions dans le code limitent les performances globales et pouvant avoir un effet sur les temps de traitement et de réponse. Un Bottleneck peut se produire au niveau du réseau d'utilisateurs, de la topologie de stockage ou des serveurs lorsque trop de contraintes pèsent sur les ressources serveur internes, comme la puissance traitement du processeur, la mémoire ou les entrées/sorties (E/S).



Spark est un outil complexe qui émet un événement chaque fois que l'application exécute une action au niveau de la tâche, du Stage, du Job et de l'application. Je vais dans un premier temps définir certains termes qu'utilise Spark pour chaque application.

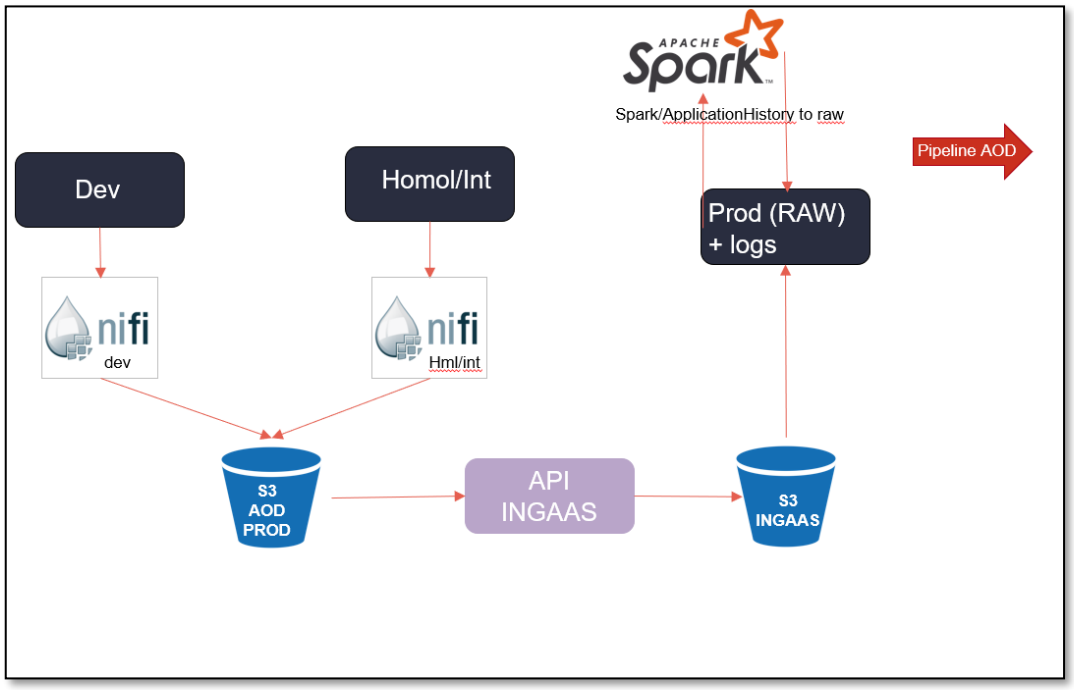
- **Une Application** : un programme utilisateur construit sur Spark. Elle consiste en un driver programme et plusieurs exécuteurs.
- **Driver program** : c'est le processus qui exécute la fonction `main()` de l'application et qui créer son contexte.
- **Les exécuteurs** : un processus lancé par une application sur un nœud « worker » qui exécute les tâches et gardent les données en mémoire ou sur le disque de stockage. Chaque application possède ses exécuteurs.
- **Les tâches** : une unité de calcul qui sera envoyé à un des exécuteurs. C'est une opération unique (`.map` ou `.filter`) appliqué à une seule partition de donnée.
- **Un job** : un job est une séquence de stages déclenchée par une action Spark comme : `.count()`, `.read()`, `.write()`
- **Les stages** : les stages sont des séquences de tâches qui peuvent être exécuté en parallèles.
- **Les logs** : les logs Spark sont stockés dans un répertoire dédié et consistent en un fichier qui récapitule l'exécution de l'application. Elle stocke les données par événements et contient également les tâches et stages.

Architecture de l'application

Un des objectifs de notre application est d'accéder à tous les environnements du cluster Big Data. Il en existe 4 :

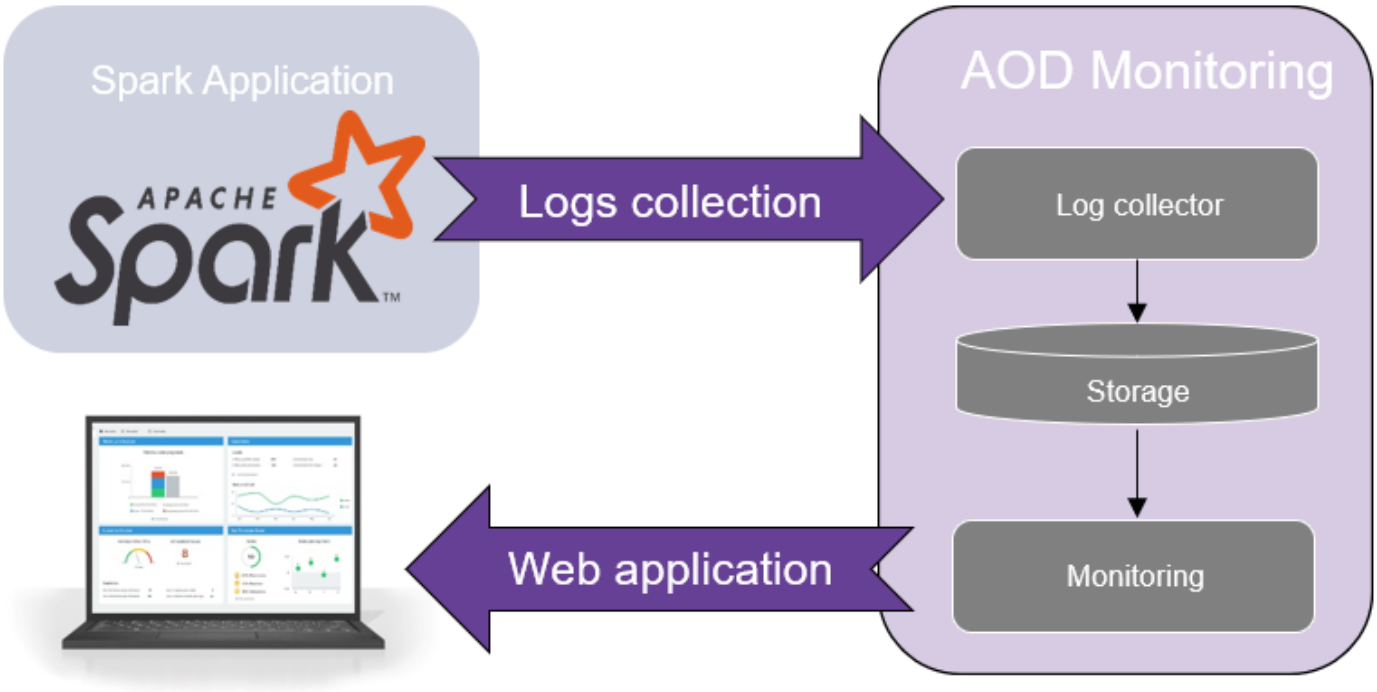
- L'environnement de développement, l'objectif est de construire une solution retenue, mis à jour très fréquemment.
- L'environnement d'homologation, qui a pour but de valider la conformité de la solution construite.
- L'environnement d'intégration est une phase d'acceptation avant la mise en production
- L'environnement de production héberge une version stable du projet exempt de bug et de mauvais comportements.

Cependant en termes de sécurité, ces environnements ne communiquent pas d'informations entre eux. Nous avons donc dû penser à une solution qui nous permettait de récupérer les logs de chaque environnement au même endroit pour que notre projet soit le plus complet possible au niveau de sa recommandation. L'idée est de récupérer les logs d'une application en développement pour pouvoir prédire son temps d'exécution en homologation ou production.



Une solution qui permet de récupérer les données sans prendre en compte les environnements respectifs de chaque log existe. En effet, nous avons fait appel à **Nifi**, un logiciel de flux de données qui permet de gérer et d'automatiser des flux de données entre plusieurs systèmes informatiques dans un environnement distribué.
Ces données sont ensuite envoyées dans des **Buckets S3** (Simple Storage Service), puis nous utilisons une offre de service que propose la Société Générale afin d'envoyer nos données directement sur le **cluster HDFS** (Hadoop File System) un système de stockage et de traitement de fichiers adapté au Big Data puisqu'il permet la gestion de données volumineuses. Depuis ce cluster de production, nous réalisons les traitements afin d'avoir ensuite notre Dashboard d'applications.

Une fois les données traitées, voici l'architecture de notre application :

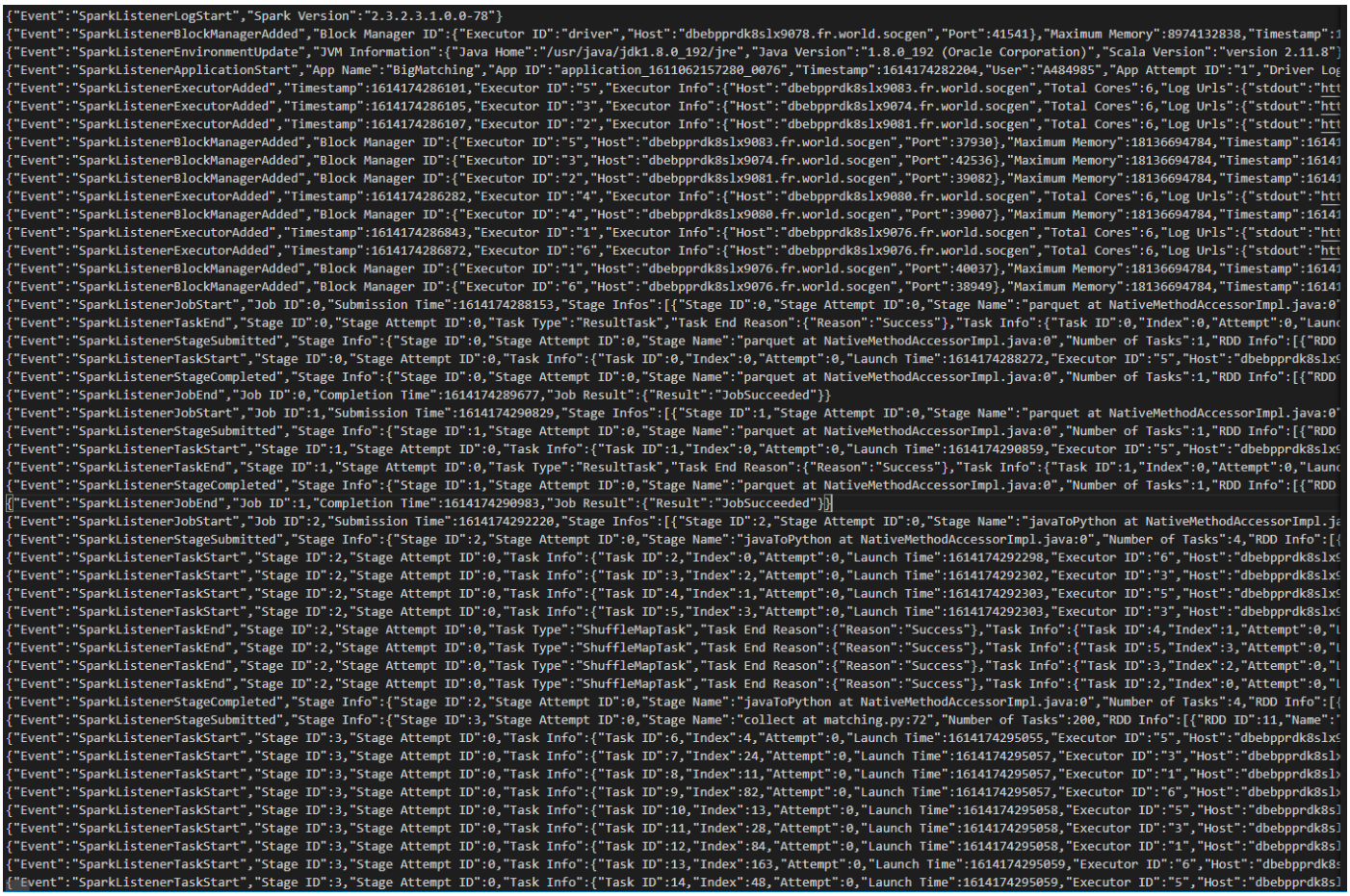


Les données envoyées en production sont ensuite collectées et traitées puis enfin stockées pour être affichées dans notre application web. Pour récupérer les données depuis le cluster de production HDFS, nous utilisons l'outil **Presto**, un moteur de requête SQL pour optimiser les interactions temps réel. Nous connectons le cluster Presto à notre base de données **Hive** pour ensuite requêter sur cette base de données grâce à Presto sur notre API Backend **Springboot**. L'application web récupère ses données pour les afficher sur le Dashboard.

3) Les principales missions

Parsing des logs

A mon arrivée, une première tâche qui m'a été donnée était de reprendre un projet déjà développé sur le parsing (analyse) des logs Spark et de l'adapter à notre besoin. Il s'agissait de construire un nouveau pipeline pour notre projet qui sélectionnait les colonnes appropriées d'un fichier de logs pour ensuite les stocker sur une base de données **PostgreSQL** en attendant le développement des clusters Presto. Cette partie a été développé dans un projet Scala Spark.



{"Event":"SparkListenerLogStart","Spark Version":"2.3.2.3.1.0.0-78"}
{"Event":"SparkListenerBlockManagerAdded","Block Manager ID":{"Executor ID":"driver","Host":"dbebpprdk8slx9078.fr.world.socgen","Port":41541},"Maximum Memory":8974132838,"Timestamp":1
{"Event":"SparkListenerEnvironmentUpdate","JVM Information":{"Java Home":"/usr/java/jdk1.8.0_192/jre","Java Version":"1.8.0_192 (Oracle Corporation)","Scala Version":"version 2.11.8"}
{"Event":"SparkListenerApplicationStart","App Name":"BigMatching","App ID":"application_1611062157280_0076","Timestamp":1614174282204,"User":"A484985","App Attempt ID":"1","Driver Log
{"Event":"SparkListenerExecutorAdded","Timestamp":1614174286101,"Executor ID":"5","Executor Info":{"Host":"dbebpprdk8slx9083.fr.world.socgen","Total Cores":6,"Log Urls":{"stdout":"htt
{"Event":"SparkListenerExecutorAdded","Timestamp":1614174286105,"Executor ID":"3","Executor Info":{"Host":"dbebpprdk8slx9074.fr.world.socgen","Total Cores":6,"Log Urls":{"stdout":"htt
{"Event":"SparkListenerExecutorAdded","Timestamp":1614174286107,"Executor ID":"2","Executor Info":{"Host":"dbebpprdk8slx9081.fr.world.socgen","Total Cores":6,"Log Urls":{"stdout":"htt
{"Event":"SparkListenerBlockManagerAdded","Block Manager ID":{"Executor ID":"5","Host":"dbebpprdk8slx9083.fr.world.socgen","Port":37930},"Maximum Memory":18136694784,"Timestamp":1614
{"Event":"SparkListenerBlockManagerAdded","Block Manager ID":{"Executor ID":"3","Host":"dbebpprdk8slx9074.fr.world.socgen","Port":42536},"Maximum Memory":18136694784,"Timestamp":1614
{"Event":"SparkListenerBlockManagerAdded","Block Manager ID":{"Executor ID":"2","Host":"dbebpprdk8slx9081.fr.world.socgen","Port":39082},"Maximum Memory":18136694784,"Timestamp":1614
{"Event":"SparkListenerExecutorAdded","Timestamp":1614174286282,"Executor ID":"4","Executor Info":{"Host":"dbebpprdk8slx9080.fr.world.socgen","Total Cores":6,"Log Urls":{"stdout":"htt
{"Event":"SparkListenerBlockManagerAdded","Block Manager ID":{"Executor ID":"4","Host":"dbebpprdk8slx9080.fr.world.socgen","Port":39007},"Maximum Memory":18136694784,"Timestamp":1614
{"Event":"SparkListenerExecutorAdded","Timestamp":1614174286843,"Executor ID":"1","Executor Info":{"Host":"dbebpprdk8slx9076.fr.world.socgen","Total Cores":6,"Log Urls":{"stdout":"htt
{"Event":"SparkListenerExecutorAdded","Timestamp":1614174286872,"Executor ID":"6","Executor Info":{"Host":"dbebpprdk8slx9076.fr.world.socgen","Total Cores":6,"Log Urls":{"stdout":"htt
{"Event":"SparkListenerBlockManagerAdded","Block Manager ID":{"Executor ID":"1","Host":"dbebpprdk8slx9076.fr.world.socgen","Port":40037},"Maximum Memory":18136694784,"Timestamp":1614
{"Event":"SparkListenerBlockManagerAdded","Block Manager ID":{"Executor ID":"6","Host":"dbebpprdk8slx9076.fr.world.socgen","Port":38949},"Maximum Memory":18136694784,"Timestamp":1614
{"Event":"SparkListenerJobStart","Job ID":0,"Submission Time":1614174288153,"Stage Infos":[{"Stage ID":0,"Stage Attempt ID":0,"Stage Name":"parquet at NativeMethodAccessorImpl.java:0"
{"Event":"SparkListenerTaskEnd","Stage ID":0,"Stage Attempt ID":0,"Task Type":"ResultTask","Task End Reason":{"Reason":"Success"},"Task Info":{"Task ID":0,"Index":0,"Attempt":0,"Launc
{"Event":"SparkListenerStageSubmitted","Stage Info":{"Stage ID":0,"Stage Attempt ID":0,"Stage Name":"parquet at NativeMethodAccessorImpl.java:0","Number of Tasks":1,"RDD Info":[{"RDD
{"Event":"SparkListenerTaskStart","Stage ID":0,"Stage Attempt ID":0,"Task Info":{"Task ID":0,"Index":0,"Attempt":0,"Launch Time":1614174288272,"Executor ID":"5","Host":"dbebpprdk8slx
{"Event":"SparkListenerStageCompleted","Stage Info":{"Stage ID":0,"Stage Attempt ID":0,"Stage Name":"parquet at NativeMethodAccessorImpl.java:0","Number of Tasks":1,"RDD Info":[{"RDD
{"Event":"SparkListenerJobEnd","Job ID":0,"Completion Time":1614174289677,"Job Result":{"Result":"JobSucceeded"}}
{"Event":"SparkListenerJobStart","Job ID":1,"Submission Time":1614174290829,"Stage Infos":[{"Stage ID":1,"Stage Attempt ID":0,"Stage Name":"parquet at NativeMethodAccessorImpl.java:0"
{"Event":"SparkListenerStageSubmitted","Stage Info":{"Stage ID":1,"Stage Attempt ID":0,"Stage Name":"parquet at NativeMethodAccessorImpl.java:0","Number of Tasks":1,"RDD Info":[{"RDD
{"Event":"SparkListenerTaskStart","Stage ID":1,"Stage Attempt ID":0,"Task Info":{"Task ID":1,"Index":0,"Attempt":0,"Launch Time":1614174290859,"Executor ID":"5","Host":"dbebpprdk8slx
{"Event":"SparkListenerTaskEnd","Stage ID":1,"Stage Attempt ID":0,"Task Type":"ResultTask","Task End Reason":{"Reason":"Success"},"Task Info":{"Task ID":1,"Index":0,"Attempt":0,"Launc
{"Event":"SparkListenerStageCompleted","Stage Info":{"Stage ID":1,"Stage Attempt ID":0,"Stage Name":"parquet at NativeMethodAccessorImpl.java:0","Number of Tasks":1,"RDD Info":[{"RDD
{"Event":"SparkListenerJobEnd","Job ID":1,"Completion Time":1614174290983,"Job Result":{"Result":"JobSucceeded"}}
{"Event":"SparkListenerJobStart","Job ID":2,"Submission Time":1614174292220,"Stage Infos":[{"Stage ID":2,"Stage Attempt ID":0,"Stage Name":"javaToPython at NativeMethodAccessorImpl.ja
{"Event":"SparkListenerStageSubmitted","Stage Info":{"Stage ID":2,"Stage Attempt ID":0,"Stage Name":"javaToPython at NativeMethodAccessorImpl.java:0","Number of Tasks":4,"RDD Info":[{
{"Event":"SparkListenerTaskStart","Stage ID":2,"Stage Attempt ID":0,"Task Info":{"Task ID":2,"Index":0,"Attempt":0,"Launch Time":1614174292298,"Executor ID":"6","Host":"dbebpprdk8slx
{"Event":"SparkListenerTaskStart","Stage ID":2,"Stage Attempt ID":0,"Task Info":{"Task ID":3,"Index":2,"Attempt":0,"Launch Time":1614174292302,"Executor ID":"3","Host":"dbebpprdk8slx
{"Event":"SparkListenerTaskStart","Stage ID":2,"Stage Attempt ID":0,"Task Info":{"Task ID":4,"Index":1,"Attempt":0,"Launch Time":1614174292303,"Executor ID":"5","Host":"dbebpprdk8slx
{"Event":"SparkListenerTaskStart","Stage ID":2,"Stage Attempt ID":0,"Task Info":{"Task ID":5,"Index":3,"Attempt":0,"Launch Time":1614174292303,"Executor ID":"3","Host":"dbebpprdk8slx
{"Event":"SparkListenerTaskEnd","Stage ID":2,"Stage Attempt ID":0,"Task Type":"ShuffleMapTask","Task End Reason":{"Reason":"Success"},"Task Info":{"Task ID":4,"Index":1,"Attempt":0,"
{"Event":"SparkListenerTaskEnd","Stage ID":2,"Stage Attempt ID":0,"Task Type":"ShuffleMapTask","Task End Reason":{"Reason":"Success"},"Task Info":{"Task ID":5,"Index":3,"Attempt":0,"
{"Event":"SparkListenerTaskEnd","Stage ID":2,"Stage Attempt ID":0,"Task Type":"ShuffleMapTask","Task End Reason":{"Reason":"Success"},"Task Info":{"Task ID":3,"Index":2,"Attempt":0,"
{"Event":"SparkListenerTaskEnd","Stage ID":2,"Stage Attempt ID":0,"Task Type":"ShuffleMapTask","Task End Reason":{"Reason":"Success"},"Task Info":{"Task ID":2,"Index":0,"Attempt":0,"
{"Event":"SparkListenerStageCompleted","Stage Info":{"Stage ID":2,"Stage Attempt ID":0,"Stage Name":"javaToPython at NativeMethodAccessorImpl.java:0","Number of Tasks":4,"RDD Info":[{
{"Event":"SparkListenerStageSubmitted","Stage Info":{"Stage ID":3,"Stage Attempt ID":0,"Stage Name":"collect at matching.py:72","Number of Tasks":200,"RDD Info":[{"RDD ID":11,"Name":"
{"Event":"SparkListenerTaskStart","Stage ID":3,"Stage Attempt ID":0,"Task Info":{"Task ID":6,"Index":4,"Attempt":0,"Launch Time":1614174295055,"Executor ID":"5","Host":"dbebpprdk8slx
{"Event":"SparkListenerTaskStart","Stage ID":3,"Stage Attempt ID":0,"Task Info":{"Task ID":7,"Index":24,"Attempt":0,"Launch Time":1614174295057,"Executor ID":"3","Host":"dbebpprdk8sl
{"Event":"SparkListenerTaskStart","Stage ID":3,"Stage Attempt ID":0,"Task Info":{"Task ID":8,"Index":11,"Attempt":0,"Launch Time":1614174295057,"Executor ID":"1","Host":"dbebpprdk8sl
{"Event":"SparkListenerTaskStart","Stage ID":3,"Stage Attempt ID":0,"Task Info":{"Task ID":9,"Index":82,"Attempt":0,"Launch Time":1614174295057,"Executor ID":"6","Host":"dbebpprdk8sl
{"Event":"SparkListenerTaskStart","Stage ID":3,"Stage Attempt ID":0,"Task Info":{"Task ID":10,"Index":13,"Attempt":0,"Launch Time":1614174295058,"Executor ID":"5","Host":"dbebpprdk8s
{"Event":"SparkListenerTaskStart","Stage ID":3,"Stage Attempt ID":0,"Task Info":{"Task ID":11,"Index":28,"Attempt":0,"Launch Time":1614174295058,"Executor ID":"3","Host":"dbebpprdk8s
{"Event":"SparkListenerTaskStart","Stage ID":3,"Stage Attempt ID":0,"Task Info":{"Task ID":12,"Index":84,"Attempt":0,"Launch Time":1614174295058,"Executor ID":"1","Host":"dbebpprdk8s
{"Event":"SparkListenerTaskStart","Stage ID":3,"Stage Attempt ID":0,"Task Info":{"Task ID":13,"Index":163,"Attempt":0,"Launch Time":1614174295059,"Executor ID":"6","Host":"dbebpprdk8s
{"Event":"SparkListenerTaskStart","Stage ID":3,"Stage Attempt ID":0,"Task Info":{"Task ID":14,"Index":48,"Attempt":0,"Launch Time":1614174295059,"Executor ID":"5","Host":"dbebpprdk8s

Voici un exemple de fichier de logs que nous recevons directement sur le répertoire où sont stockés les logs sur HDFS. Grâce à un job Spark, j'écrivais ensuite ses données dans un premier temps sur la base de données PostgreSQL puis à terme sur Hive puisque notre solution finale stocke les données traitées sur Hive.

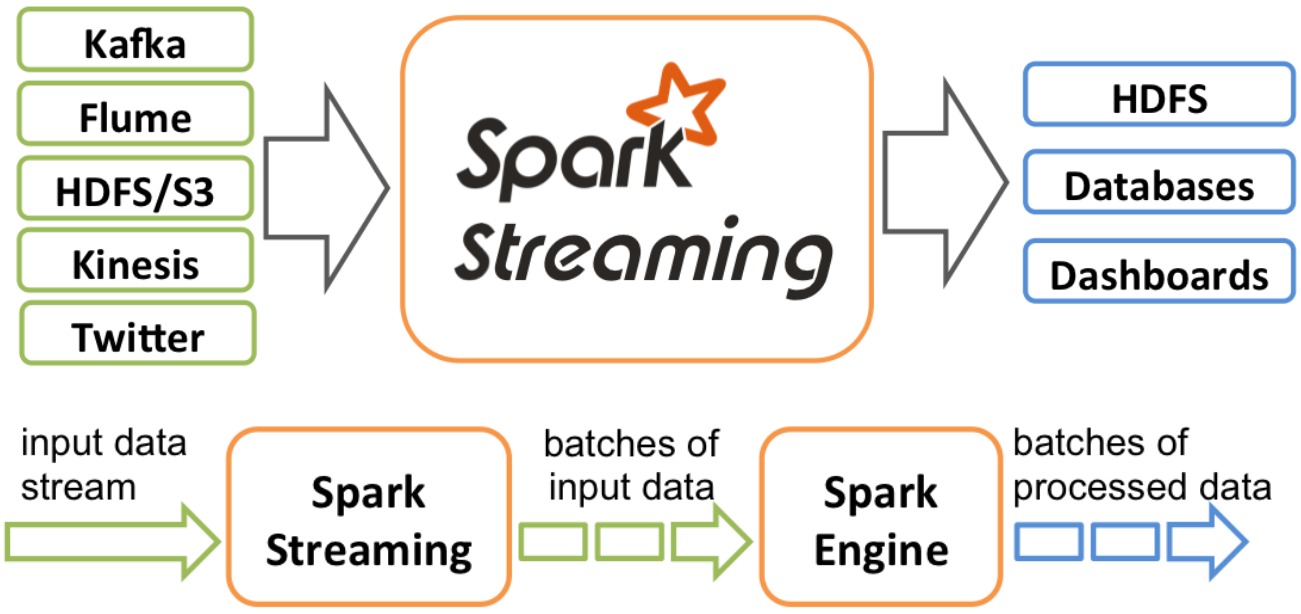
De ces jobs Spark, j'extrait plusieurs informations, tout d'abord les informations de l'application à son lancement (notamment le nombre d'exécuteur, le nom de l'application, sa date d'exécution…). Ensuite je récupérais ses stages et tâches pour les stockés dans la base de données.
Dans un second temps il fallait également vérifier le contenu des fichiers. Dans le cas où un job échoue par exemple, ou ne contient pas de stage, il n'est pas possible de récupérer les données du job. J'ai mis en place des tests pour ne sélectionner que les jobs n'ayant pas échoué et contenant des stages.

log_id text	app_name text	app_id text	User text	timestamp bigint	date timestamp without time zone	spark_executor_memory text	timestamp_end bigint	environment text	app_duration bigint	executors bigint	cores bigint
a3783064...	DataQuality	applicatio...	bc4-5c...	1625749923696	2021-07-08 15:12:03	4	1625749946523	HOMOLOGATION	22827	1	3
c3601a9c...	CLK	applicatio...	ba1-92...	1626955875654	2021-07-22 14:11:15	4	1626958580212	PRODUCTION	2704558	18	4
3b6c51c5...	DataQuality	applicatio...	bc4-5c...	1622099919241	2021-05-27 09:18:39	4	1622099990929	DEVELOPMENT	71688	3	3
315b0dc8...	Placideo_MB	applicatio...	8a7-87...	1627458375628	2021-07-28 09:46:15	1	1627458409909	DEVELOPMENT	34281	5	2
574c839b...	DataCleaner	applicatio...	a3b-a2...	1627426966183	2021-07-28 01:02:46	1	1627426998158	DEVELOPMENT	31975	2	4
e38419e8...	MUSED	applicatio...	d32-00...	1627457436849	2021-07-28 09:30:36	4	1627457530798	PRODUCTION	93949	15	5
b2f92e63...	CLK	applicatio...	4b0-0e...	1627479944562	2021-07-28 15:45:44	4	1627480057309	DEVELOPMENT	112747	6	4
ea80810b...	CLK	applicatio...	4b0-0e...	1625747167108	2021-07-08 14:26:07	4	1625747324842	HOMOLOGATION	157734	10	4
3045ca13...	DataQuality	applicatio...	a01-9c...	1627509953115	2021-07-29 00:05:53	4	1627510083451	PRODUCTION	130336	5	3
2b2dfdfc-...	MUSED	applicatio...	ba6-b5...	1627545605513	2021-07-29 10:00:05	4	1627545663389	DEVELOPMENT	57876	3	5
df82b03f-...	MUSED	applicatio...	ba6-b5...	1627543095844	2021-07-29 09:18:15	4	1627543174272	HOMOLOGATION	78428	1	5
9c0d23d7...	Monitoring Hdf...	applicatio...	0a2-5a...	1627466428702	2021-07-28 12:00:28	1	1627466447241	DEVELOPMENT	18539	4	2
92711505...	Monitoring Hdf...	applicatio...	0a2-5a...	1627542014963	2021-07-29 09:00:14	1	1627542050027	HOMOLOGATION	35064	3	2
44881cdf...	Monitoring Hdf...	applicatio...	e9b-01...	1627509615471	2021-07-29 00:00:15	1	1627509643510	PRODUCTION	28039	4	2
50897858...	DSR	applicatio...	643-91...	1628681061693	2021-08-11 13:24:21	8	1628682121139	PRODUCTION	1059446	241	5
5c1116ab...	DataQuality	applicatio...	a01-9c...	1628682530330	2021-08-11 13:48:50	4	1628683705333	PRODUCTION	1175003	5	3
17c2a99c...	Oozie monitori...	applicatio...	ba6-b5...	1628691856227	2021-08-11 16:24:16	4	1628691967449	DEVELOPMENT	111222	6	5
248ecea1...	Placideo	applicatio...	8a7-87...	1628693169310	2021-08-11 16:46:09	1	1628693342138	DEVELOPMENT	172828	5	2
89f94f7d-...	QU.R.E.	applicatio...	4f0-d9e...	1628690861736	2021-08-11 16:07:41	25	1628691418459	PRODUCTION	556723	54	5
436143ae...	DHR-DEV-com...	applicatio...	976-02...	1629446816222	2021-08-20 10:06:56	8	1629446929425	DEVELOPMENT	113203	13	2

Voici un exemple de la base de données PostgreSQL qui représente les données de l'information de l'application.
En parallèle de ces opérations pour chaque fonction développée, je réalisais des tests unitaires pour vérifier le bon fonctionnement des méthodes développées.

Spark Streaming

Le traitement de notre application et des logs Spark a été fait en temps réel. J'ai développé un job de **Spark Streaming** qui lit les fichiers depuis un répertoire sur HDFS pour ensuite les traiter et les stocker. Grâce au module Spark Streaming, il est possible de traiter des flux de données qui arrivent en continu, et donc de traiter ces données au fur et à mesure de leur arrivée.



En fonction de l'environnement dans lequel sont les logs, le job de Streaming va lire différents répertoires. Pour la production, nous lirons directement sur le répertoire où les fichiers de logs sont écris. Pour les autres environnements notamment le développement et l'homologation, c'est à ce moment qu'entre en jeu l'offre de service pour aller écrire dans un répertoire HDFS.

L'Interface Homme Machine

L'objectif final du projet AOD étant de pouvoir monitorer ses applications, une interface homme machine a été développée dans le but de pouvoir consulter ces métriques de nos applications.
Une des tâches qui m'a été confiée était également de développer une première interface assez basique pour avoir un point de départ de l'application.

Pour réaliser cette interface, nous avons utilisé plusieurs technologies.

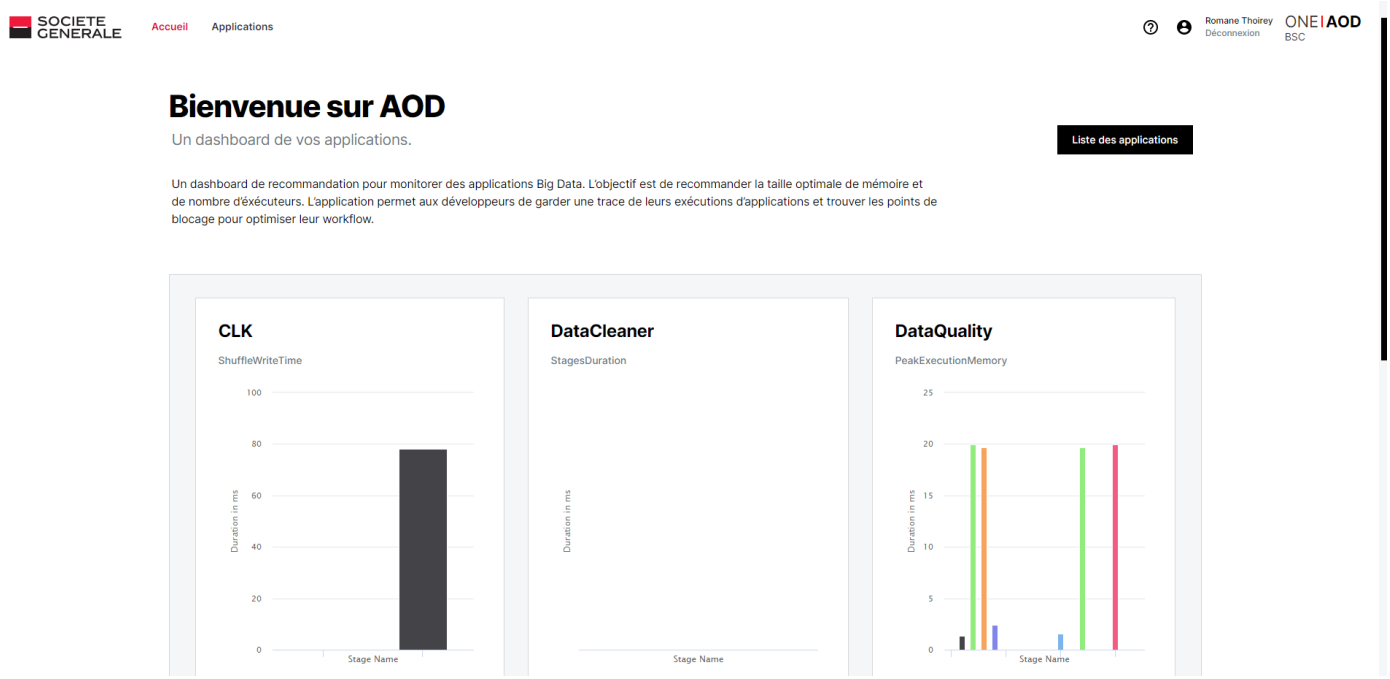
L'interface a été réalisée en Javascript avec la librairie **React**. Cette librairie facilite la création d'application web mono page, grâce à la création de code déclaratif.

Pour communiquer avec les données stockées sur Hive, nous passons par **PostgreSQL** afin d'interagir avec notre interface. PostgreSQL est un système de gestion de base de données relationnelle et objet (SGBDRO).

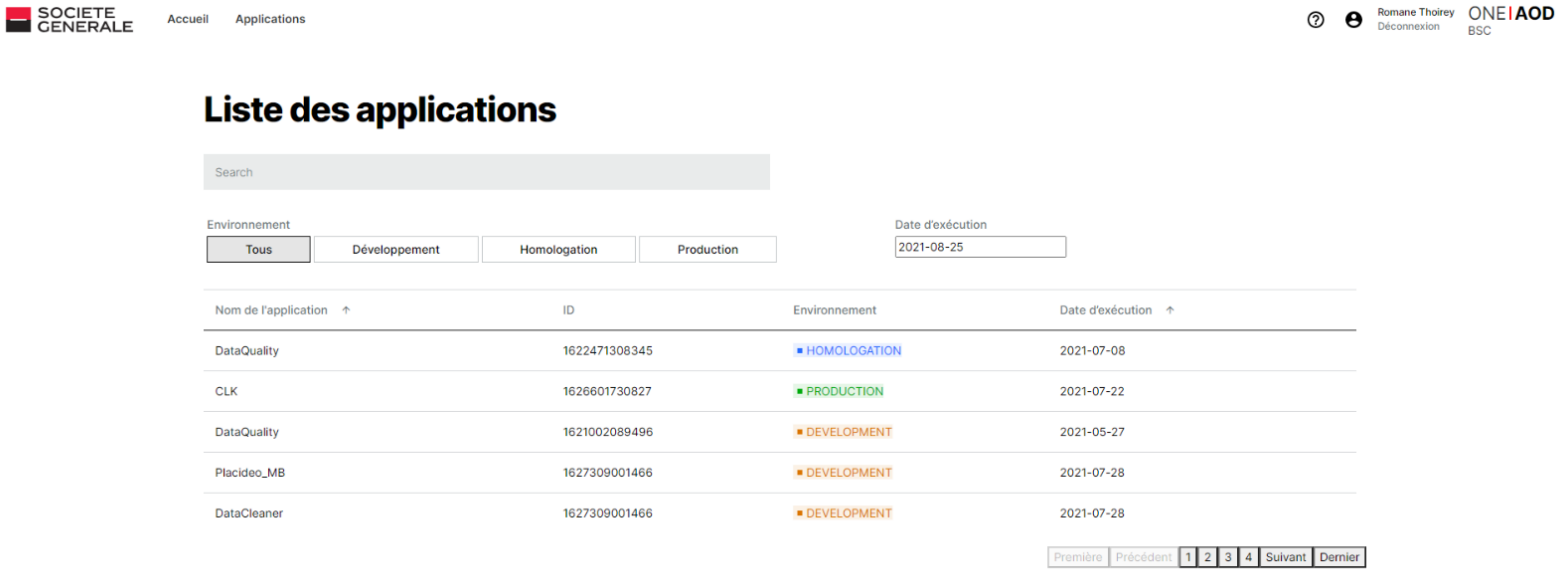
Afin de récupérer les données sur PostgreSQL, nous avons créé une API pour mettre en lien la base de données et l'interface. Elle a été développée dans le langage Java avec la librairie **SpringBoot**.
Cette API se base sur l'architecture logicielle MVC (Model, Vue, Controller) pour créer des applications web.

J'ai développé les premières pages de l'interface pour avoir un début et quelques APIs pour pouvoir interagir avec la base de données. Plus tard dans le projet nous avons fait appel à un UX designer pour avoir une application web la plus facile et agréable à utiliser. Un développeur Frontend a pris la main sur l'interface web, j'ai pu me concentrer sur la partie Big Data du projet.

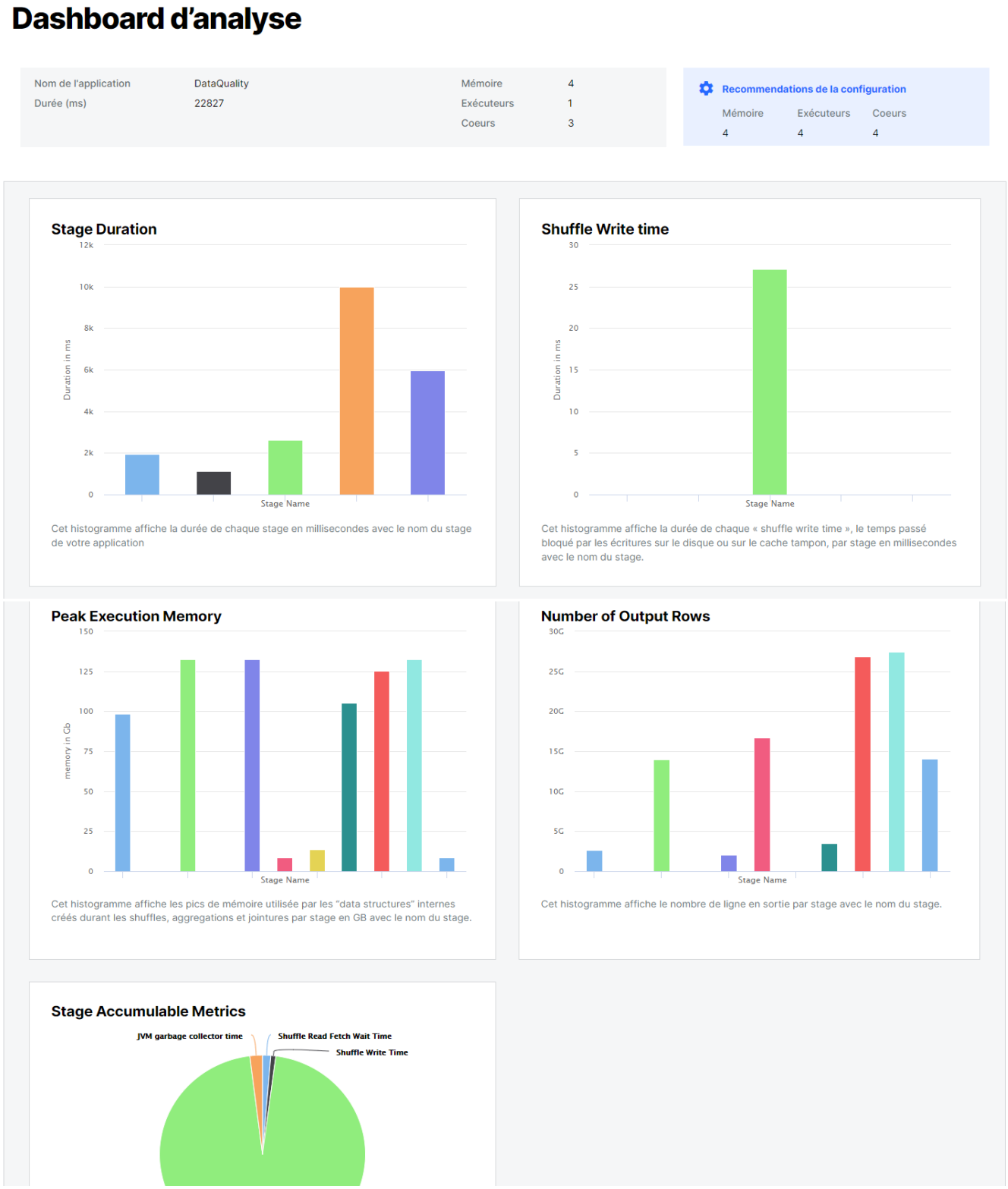
Nous avons réfléchi avec les UX designer a une interface facile d'utilisation. Voici les écrans qui ont été développé :



L'écran d'accueil est composé d'une brève description de notre application et d'un récapitulatif des dernières applications exécutées avec un aperçu des graphes affichés dans le Dashboard.



Cette page liste les applications traitées. Nous avons récupéré et affiché les informations concernant le nom de l'application, l'environnement dans lequel elle a été lancé, et sa date d'exécution. Cette page est faite pour permettre à l'utilisateur de rechercher les applications qu'il aurait lancé.

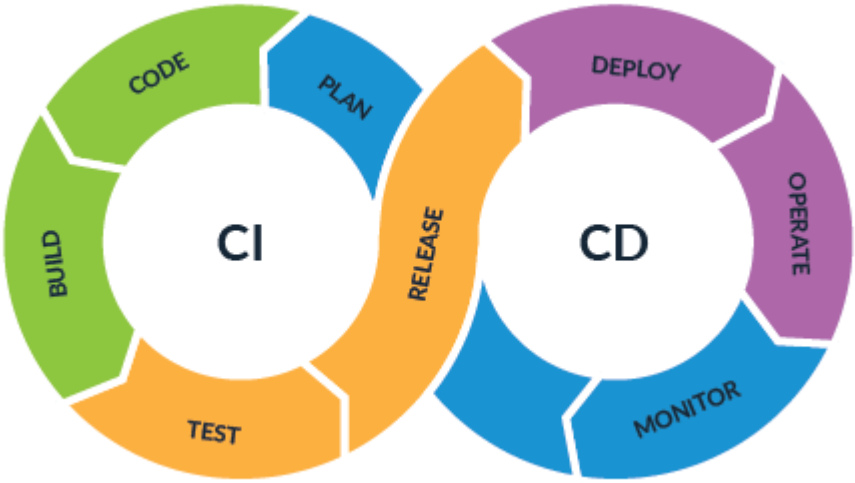


La page de Dashboard contient les informations récupérées de l'application. Grâce aux endpoints de l'API, nous affichons les données de l'application sous forme de graphiques dont par exemple la durée de chaque stage. Cela permet aux utilisateurs d'avoir une interface accessible pour avoir un récapitulatif de leur projet et leur code.
Nous avons préparé la recommandation de configuration avec notamment le nombre de cœurs, de mémoire et d'exécuteurs, cependant cette fonctionnalité n'est pas encore développée.

Déploiement DevOps, CI/CD et automatisation du projet

L'objectif du projet est d'automatiser la récupération des logs Spark ainsi que leur traitement et leur affichage. Le déploiement concerne aussi bien la partie Big Data développée en Scala Spark et la partie Web avec le Frontend et Backend. Nombreuses sont les technologies utilisées pour le déploiement des jobs Spark ou bien pour celui du déploiement de l'IHM.

Tout d'abord, notre projet utilise l'approche CI/CD. Elle permet d'augmenter la fréquence de distribution des applications grâce à l'introduction de l'automatisation au niveau des étapes de développement des applications. Les principaux concepts liés à l'approche CI/CD sont l'intégration continue, la distribution continue et le déploiement continu.
L'acronyme « CI/CD » a plusieurs significations. Le « CI » de CI/CD désigne toujours l'« intégration continue », à savoir un processus d'automatisation pour les développeurs. L'intégration continue consiste, pour les développeurs, à apporter régulièrement des modifications au code de leur application, à les tester, puis à les fusionner dans un référentiel partagé. Cette solution permet d'éviter de travailler en même temps sur un trop grand nombre d'éléments d'une application, qui pourraient entrer en conflit les uns avec les autres.
Le « CD » de CI/CD désigne la « distribution continue » et/ou le « déploiement continu », qui sont des concepts très proches, parfois utilisés de façon interchangeable. Les deux concepts concernent l'automatisation d'étapes plus avancées du pipeline, mais ils sont parfois dissociés pour illustrer le haut degré d'automatisation.



Pour automatiser les traitements réalisés avec Spark, j'ai écrit un job **Oozie** qui lance le job de Spark Streaming. Apache Oozie est un logiciel de la Fondation Apache servant à l'ordonnancement de flux dédié au logiciel Hadoop. Il est ensuite lancé durant le pipeline de déploiement de la partie Big Data.

Le job Oozie est lancé automatiquement via un pipeline **Jenkins**. Jenkins est un serveur d'automatisation gratuit et open source. Il aide à automatiser les parties du développement logiciel liées à la construction, aux tests et au déploiement, facilitant l'intégration continue et la livraison continue.

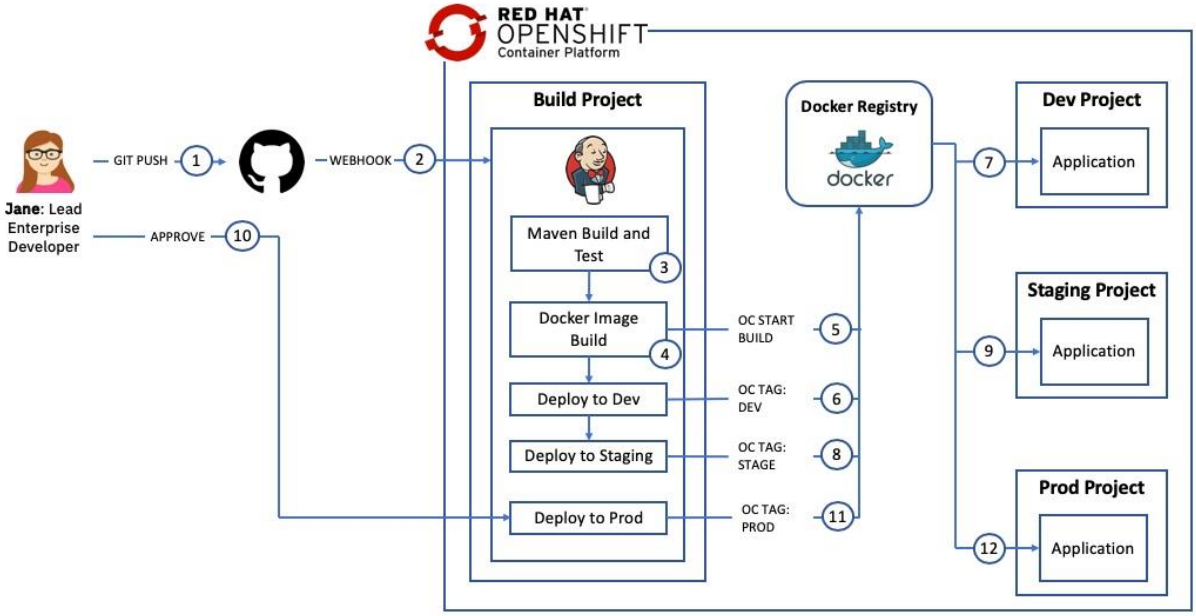
Il va construire notre projet Spark de traitement des logs, appliquées les tests unitaires et enfin le déployer sur Ansible **AWX**. Ansible est un outil DevOps qui automatise la mise en service, la gestion de configuration et entre autres le déploiement d'applications.
J'ai dû mettre à jour le projet Big Data afin qu'il soit prêt pour le déploiement sur le cluster.

Pour l'interface utilisateur et l'API Backend, nous avons utilisé les technologies déjà présentes pour beaucoup de projets à la Société Générale : **Jenkins** et **Openshift**. Une équipe externe à notre projet nous a préparé les technologies pour le déploiement. Jenkins fonctionne par paire avec Openshift pour déployer des projets. Openshift est un service de plate-forme de la société Red Hat qui permet de déployer des projets dans des containers. Pour ce faire, Openshift utilise les technologies Docker et Kubernetes. Grâce à une image du projet générée par Jenkins, Openshift l'interprète via ses outils pour la déployer.

Jenkins vient récupérer le code déposer sur des plateformes utilisées par le logiciel Git parmi GitHub et GitLab. Git est un logiciel de gestion de versions décentralisé. C'est donc grâce à cet outil que nous avions du code à jour tout en pouvant travailler à plusieurs dessus. Nous avions un répertoire pour chaque technologie utilisée, la partie Scala/Spark, l'interface utilisateur développé en React, le Backend avec SpringBoot codé en Java.

Pour expliquer la mise en pratique de ce déploiement, lorsque que je souhaitais commencer une nouvelle tâche, je commençais tout d'abord avec Git. Je créais une nouvelle branche qui s'appelle donc une branche de `feature` depuis la branche la plus fonctionnelle, à savoir la branche `develop` qui est celle qui est à jour avec les nouvelles fonctionnalités développées. La branche `master` est la plus fonctionnelle de toutes les branches. Elle correspond à ce qui est déployé en production. Elle doit donc être la plus propre possible et ne doit pas être mise en échec. Lorsque plusieurs fonctionnalités ont été fusionnées avec la branche `develop`, elles sont ensuite fusionnées avec la branche `master` et une nouvelle version du projet est alors mis à jour.

A chaque dépôt sur le répertoire Git et sur la branche `develop` ou `master`, il peut y avoir un déclencheur qui lance un nouveau build sur Jenkins ou bien il peut être lancé à la main. Jenkins va alors lancer plusieurs étapes du déploiement sur ce qu'on appelle le `pipeline`. Il lance les builds et les tests de qui sont développés sur nos applications. Quand ils échouent, c'est aux développeurs de les régler eux-mêmes et de redéposer le code sur le répertoire Git. Quand ils réussissent, le build Jenkins continue de s'exécuter. Le Quality Gate exécute des contrôles SonarQube qui inspecte la qualité du code. Et enfin à la fin du build Jenkins, il y a le build et le déploiement sur Openshift ou bien sur AWX. Ensuite Openshift s'occupe du routage de l'application, c'est grâce à ce routage que les liens de notre interface sont accessibles.



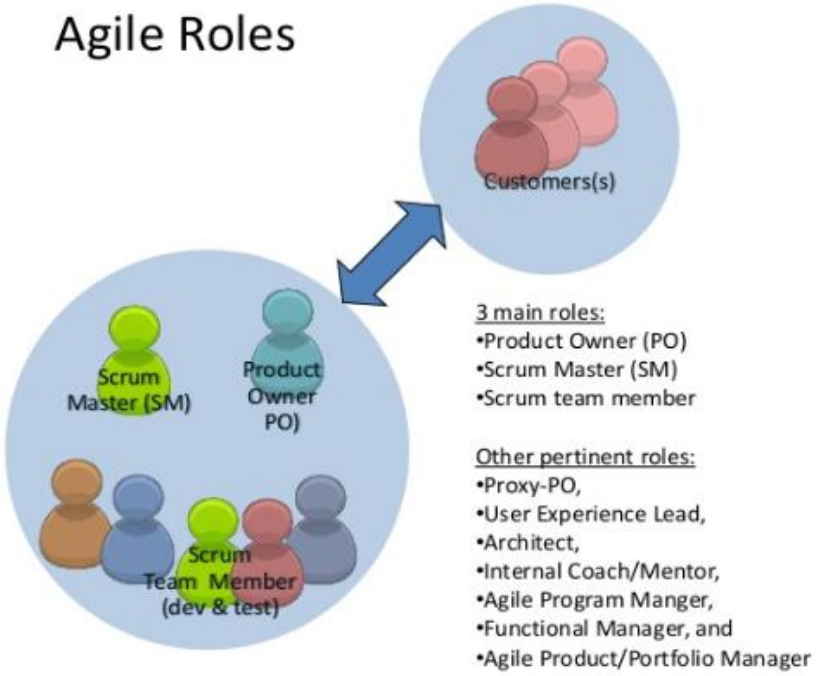
Voici un exemple de Pipeline d'un projet qui utilise les technologies Github, Jenkins et Openshift.

3. Environnement de travail

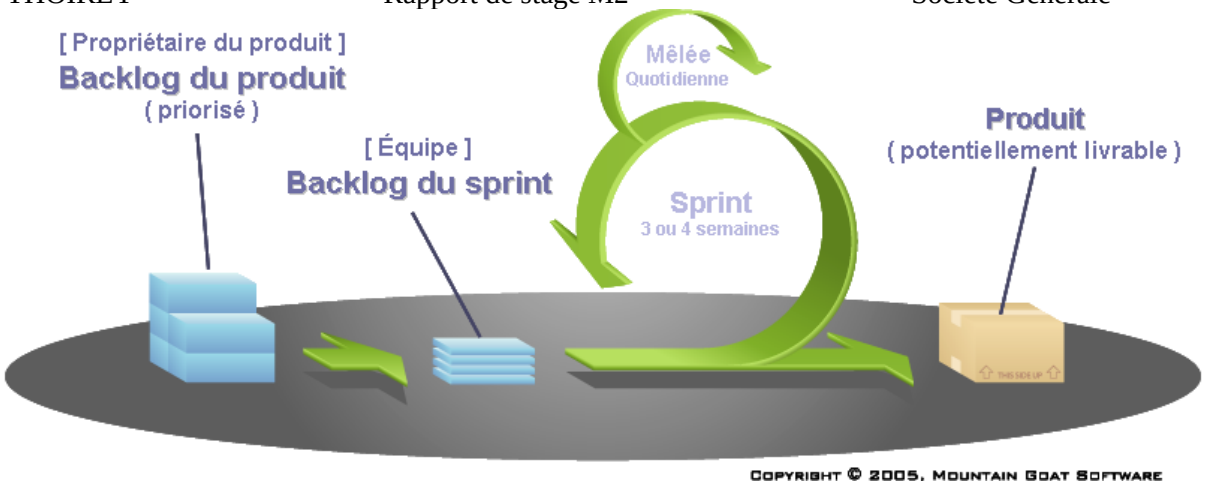
1) La méthode Agile et l'équipe

Au sein de l'équipe AOD, notre méthode de travail en équipe s'appuie sur l'approche Agile et notamment le cadre méthodologique de Scrum. Nous sommes donc une équipe composée de 4 personnes : un product owner, et 3 développeurs.

- Le Product Owner (PO), Frederic KRANTZ. Il est à l'origine de la demande du projet. Nous discutions avec lui lorsqu'il s'agissait de l'interface web ou bien de l'architecture du projet.
- Les Développeurs Big Data, Othman SEFRAOUI et moi-même Romane THOIREY, spécialisé dans les technologies tel que Spark Scala. Nous avons travaillé ensemble sur la partie Big Data du projet et nous répartissions les tâches concernant l'avancement du déploiement et de l'architecture.
- Le Développeur Frontend, Sandeep RAMANATH, est un développeur web, spécialisé dans les technologies comme Javascript situé à Bangalore. Grâce à son expérience et sa disponibilité, il a développé l'interface homme machine basée sur les maquettes des UX designer.



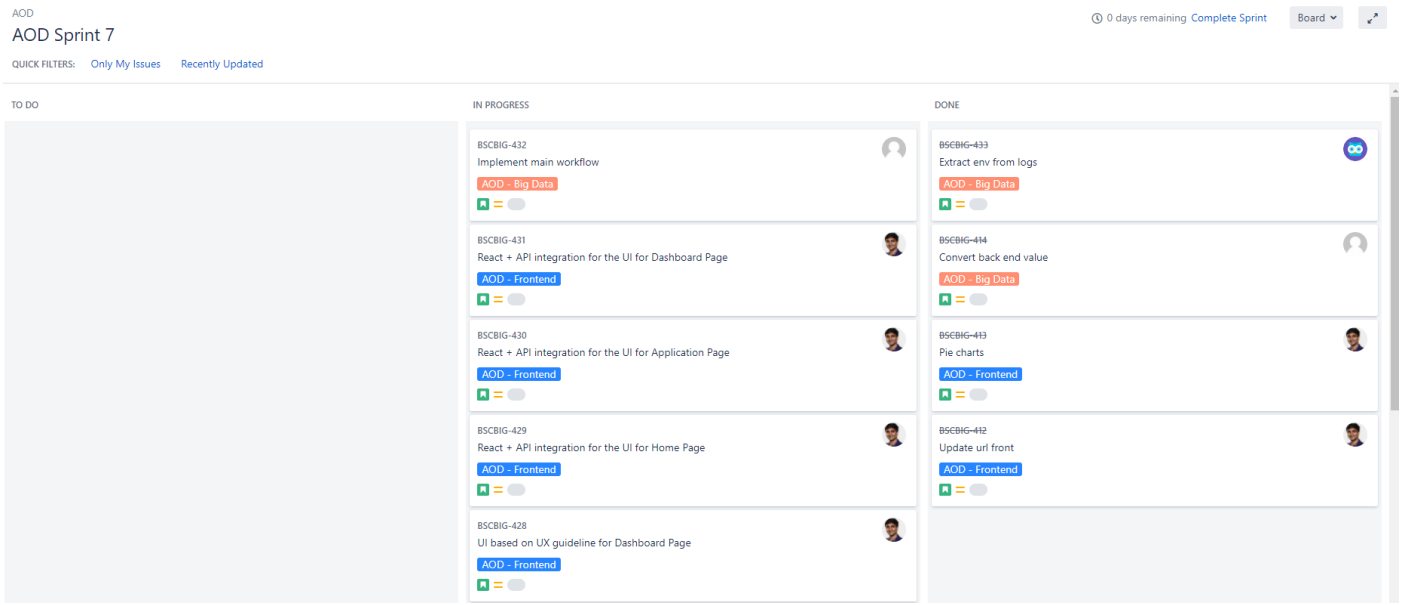
Comme je le présentais précédemment, notre équipe travaillant selon l'approche Scrum, nous avions des sprints toutes les 2 semaines avec des rendus se rapprochant au mieux du produit final. A la fin des sprints, nous faisions la revue du sprint qui venait d'être fini et parfois, nous faisions une démonstration à notre PO. Une fois le sprint fini, nous définissions les nouvelles tâches pour le sprint et leur ordre d'importance. Nous avions également des mêlées quotidiennes durant lesquelles nous présentions nos tâches faites la veille, celle que nous allons faire dans la journée et les problèmes rencontrées quand il y en avait.



2) Les outils de travail

JIRA

Afin d'appliquer au mieux l'approche Agile, nous travaillions avec un outil qui permettait le suivi des sprints en cours. Nous utilisions JIRA qui est un système de gestion de projet. Il nous permet de voir le sprint en cours et les tâches à faire durant ce sprint. Nous nous occupions d'assigner, de créer, de prioriser les tâches. Nous traitions ensuite des tâches en fonction de nos disponibilités et des tâches déjà effectuées.



IDE de développement

Pour le développement de l'API et de la partie Spark, j'utilisais l'IDE **Intelliji IDEA** pour développer en Java et Scala. Quant au développement de l'interface, l'outil le plus adapté est **Visual Studio Code**.

Github

Une fois le code fonctionnel, nous utilisions les plates-formes de Git notamment **Github** pour y déposer notre code. Dans un projet déposé sur Git, il y a 2 versions du projet. La branche master représente l'état du code en production. Les bonnes pratiques veulent qu'aucun développeur ne fasse l'erreur de travailler dans master qui est considéré comme un reflet du code en production. La branche supplémentaire develop permet de gérer les développements pour les versions à venir, c'est ce qu'on appelle souvent la branche d'intégration.

Jenkins et Openshift

Tout au long de notre stage, nous travaillions sur la version de développement, le projet n'étant déployé en production qu'une fois toutes les fonctionnalités attendues soient terminées. Nous avons donc assisté au déploiement en développement de notre application, ce qui veut dire qu'elle était disponible et accessible, et non plus accessible seulement en local sur notre machine. Pour se faire, nous avons utilisé les outils Jenkins et Openshift comme mentionné plus tôt. Jenkins nous permet également de vérifier que le code déployé était bien fonctionnel, puisqu'il exécute des commandes que nous ne faisions pas en local.

4. Difficultés rencontrées

1) La barrière de la langue

Notre développeur web était situé à Bangalore en Inde, il ne parlait donc pas français mais nous pouvions tous communiquer en anglais. Beaucoup de réunion était alors en anglais. Même si ce n'était pas nos langues maternelles, nous pouvions nous appuyer chacun sur nos bases pour discuter de termes techniques et échanger chacun sur nos problèmes. Il arrivait cependant que cette barrière soit un frein à certaines conversations puisqu'il peut arriver que nous ayons du mal à comprendre et à se faire comprendre.

2) Le travail à distance

C'est durant ce stage que j'ai pu expérimenter le travail à distance autrement dit télétravail. Dans le contexte actuel, il s'est vite imposé comme la façon de travailler. J'ai commencé mon stage en télétravail sans avoir pu rencontrer mes collègues en personne. Ils étaient joignables via l'outil Skype Entreprise, mais le contact était différent. En effet, il est plus facile et spontané de demander de l'aide ou de discuter avec un collègue qui est présent sur site plutôt que sur Skype.
Cependant après quelques semaines, nous avons pu reprendre le travail sur site sur un rythme de 2 jours sur site et 3 jours en télétravail. En privilégiant des jours similaires pour mes collègues d'équipe.

Ayant expérimentée ces deux façons de travailler, je peux affirmer que je préfère travailler sur site avec mes collègues. Cela me permet de poser des questions plus facilement et d'aller voir directement mon collègue. Notre développeur web qui est à l'étranger est le seul de mes collègues que je n'ai pas pu rencontrer, le décalage horaire n'était pas assez important pour que nous ne puissions pas échanger avec lui, nous avons donc communiquer principalement sur Skype avec lui. Quant aux autres de mes collègues, les jours sur site nous permettaient d'organiser des réunions et des points techniques où nous pouvions poser nos questions et éclaircir certains points d'ombres du projet.

Au sein de l'équipe BDE, nous avons des réunions organisées tous les jours afin d'échanger sur nos tâches respectives. Elles permettent d'avoir un contact quotidien avec des collègues de notre équipe avec lesquels je ne te travaillais pas. Nous avons également tous les vendredis un point technique où nous nous réunissons pour discuter de différents sujets. Si l'un d'entre nous souhaite nous présenter une technologie ou bien présenter un problème qu'il rencontre, il est beaucoup plus facile de communiquer ensemble quand tout le monde est sur site et se voir pour se parler.

3) Les connaissances techniques et fonctionnelles

Ce stage est mon deuxième stage technique en tant que Data Engineer, j'ai donc mis en pratique mes connaissances acquises durant l'année et au cours de mon précédent stage au sein du projet AOD. De plus, le projet utilisant une interface web, je devais également utiliser mes connaissances précédemment acquises sur ces technologies et en découvrir de nouvelles (notamment avec AWX ou bien Presto).

La difficulté était surtout au niveau de la mise en place du pipeline de Streaming. C'était une technologie que je ne connaissais et l'utilisation que j'en avais était très spécifique au projet. Le Streaming est habituellement utilisé pour lire le contenu des fichiers et non leur nom. La difficulté résidait donc dans l'extraction du nom du fichier entrant dans le pipeline de Streaming. Cette tâche m'a pris plusieurs semaines à réaliser.

Je n'avais pas eu l'occasion durant mon dernier stage de voir la mise en pratique du déploiement de la partie Big Data et surtout du déploiement du projet sur AWX et Jenkins. Il s'agissait de reconstruire tout le projet pour qu'il soit adapté au déploiement. Notamment d'utiliser Oozie et les playbooks Ansible afin que tout soit mis sous paramètre.

Certains de mes collègues ont pris le temps de m'expliquer durant certaines réunions ou lorsque je posais la question, les points sur lesquels je bloquais. Le point technique était également une bonne opportunité pour avoir plusieurs retours d'un coup et plusieurs points de vue sur un sujet sur lequel je bloquais.
Grâce à eux, j'ai pu approfondir mes connaissances sur le déploiement d'une application en développement et sur le fonctionnement pratique en Big Data dont le Streaming.

L'interface web n'est pas le principal pôle de notre entité étant tourné au cœur de la Data et du Data Engineering, Sandeep a été d'une extrême aide et efficacité sur le rendu qu'il nous a fourni sur l'IHM. Une équipe externe était présente en tant que support pour le déploiement de l'application web et de la partie Big Data. Des revues de notre code ont été prévues afin de nous guider sur les bonnes pratiques.

La partie fonctionnelle du projet était aussi difficile à assimiler. En effet, ce sujet de stage était la mise en production d'un projet qui avait déjà vu le jour l'année dernière avec Othman SEFRAOUI.
J'ai dû assimiler beaucoup d'informations concernant les métriques de Spark et comment elles pouvaient être traiter pour les rendre accessible aux développeurs. Spark est un framework assez complexe et comprendre les tenants et aboutissants peut prendre beaucoup de temps.
Heureusement nous avions des réunions régulières durant lesquels nous pouvions discuter des informations dans les logs.

4) La complexité

N'ayant pas de scrum master sur le projet, avec Othman SEFRAOUI, nous nous partagions les tâches concernant les demandes de création d'environnements et autres réunions avec les architectes. Certains de ces points prennent beaucoup de temps en raison de la complexité de notre projet.
En effet nous avons dû réfléchir à une solution d'architecture sécurisée pour accéder aux données de tous les environnements. Cette solution nous a pris environ 3 à 4 mois pour être finalisée. Ensuite nous devions faire valider cette solution par un architecte et une équipe sécurité afin de récupérer un document d'empreinte de notre projet.
Cette validation était nécessaire pour pouvoir ensuite accéder aux outils mentionnés dans l'architecture.

Beaucoup de traitements pour déployer notre application sont délégués à d'autres équipes. Nous sommes donc dépendant de ces équipes. Seulement pendant cette période estival les équipes sont en sous nombres et les processus prennent plus de temps à être réalisé.
La création d'environnement pour notre application ou bien la création d'outils comme le cluster Presto ont été complexes à réaliser. Plusieurs étapes de créations doivent passer par la validation d'une personne extérieur au projet et peut allonger les temps de processus.

5. Points positifs apportés

1) Les softs skills

Durant ce stage technique, j'ai aussi pu développer mes compétences sociales notamment. Ce projet était donc en équipe de 3/4 personnes et j'ai pu améliorer mes compétences de travail en équipe et plus particulièrement travailler en équipe avec la méthode Agile.
A l'école, lors des projets en équipe, il m'arrive souvent de me mettre avec des camarades que je connais déjà et avec qui je m'entends déjà. En entreprise, nous ne choisissons pas notre équipe et nous ne la connaissons pas non plus. Je me suis donc adaptée et j'ai su travailler en équipe pour le projet. Il faut aussi s'adapter à la façon dont chacun a de développer et lors des revues de code que nous demandions j'ai développé mon esprit critique sur ce sujet.

J'ai également développé mes capacités de leadership puisque sur certains points, j'ai pris les devants pour faire avancer le projet et mettre en place des réunions avec notre expert technique afin d'en apprendre plus et avoir des retours sur la partie technique. En effet, concernant les parties administratives du projet, nous travaillions à deux dessus. J'ai eu plus de responsabilité concernant le projet puisqu'il est question de tout construire nous-même (l'environnement, les outils, le déploiement…)

J'ai appris à travailler depuis chez moi et à avoir un rapport avec mes collègues plus distant puisque les contacts sont restreints à Skype Entreprise lorsque nous télétravaillons. J'ai donc dû développer mon autonomie puisque les échanges étaient plus limités, il est d'ailleurs parfois plus rapide d'effectuer des recherches de son côté lorsque j'ai un problème plutôt que d'attendre la réponse d'un collègue sur Skype.

2) Les hards skills

Le projet AOD était orienté Big Data mais restait assez multi-technologies. En effet, j'ai eu l'occasion de manipuler de nombreuses technologies : Scala, Spark, Hive, ReactJs, SpringBoot… J'ai pu assister à toutes les étapes du déploiement d'un projet. De sa création, à sa mise en production. J'ai donc pu voir comment les projets sont déployés en entreprise et utiliser les outils qui servent au déploiement comme Jenkins et Openshift ou bien AWX.

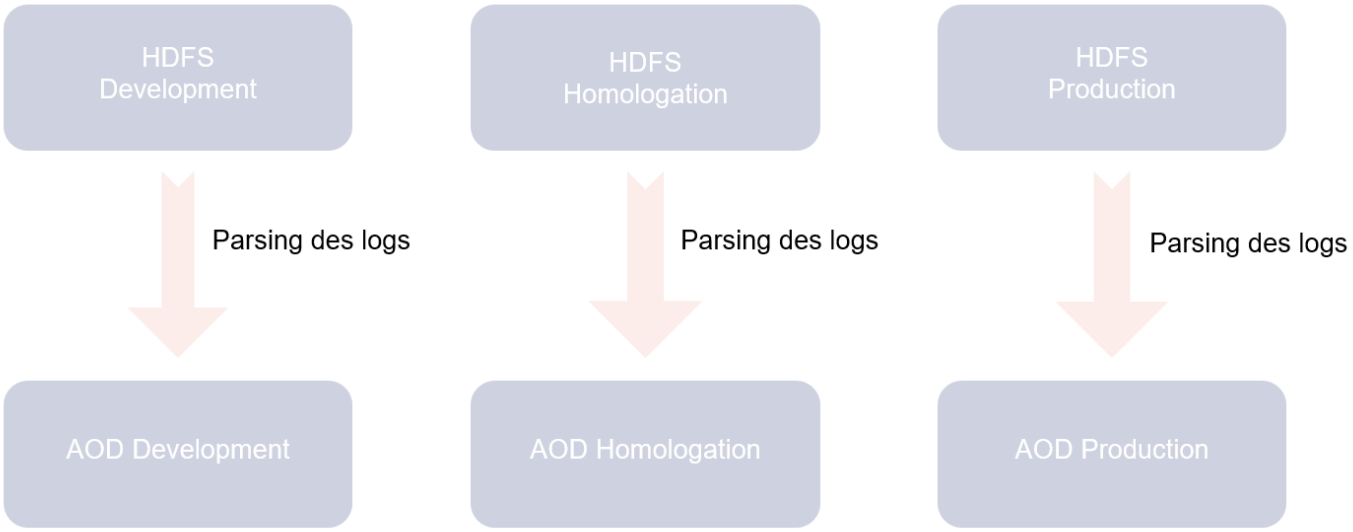
ReactJs étant une technologie que j'avais déjà utilisé dans une précédente expérience professionnelle, j'ai réappris à l'utiliser pour le projet. Pour l'API, SpringBoot étant une librairie Java et fonctionnant sur le modèle MVC, la prise en main de cette technologie fut assez facile.
Pour les autres technologies, surtout celle Big Data notamment Hive et HDFS, j'avais déjà eu l'occasion de travailler dessus lors de mon précédent stage. C'était plus facile pour moi de les prendre en main comme ne je ne les découvrais pas.

3) Le projet AOD aujourd'hui

Il faut comprendre que au vu de la crise sanitaire, le projet a pris du retard sur ces rendus initiaux. En août, le projet commençait déjà à être déployé en environnement de développement. Lors du dernier sprint auquel j'ai assisté, le mot d'ordre était la mise en production du projet. Nous étions en train de préparer l'environnement d'homologation.

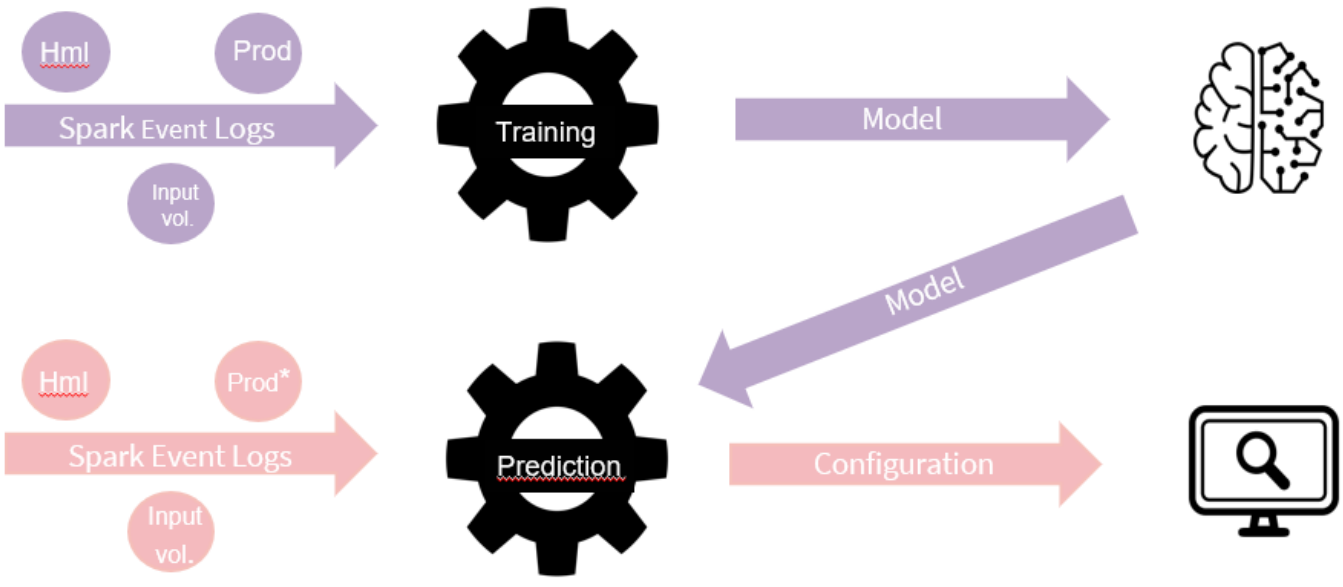
Certains outils n'étaient pas encore au point, comme l'offre de service qui proposait un bucket S3 ou bien Nifi.
Une première version d'AOD consisterait à récupérer les logs Spark par environnement (les logs de production sur le projet AOD mis en production, les logs de développement sur AOD de développement…) au lieu de récupérer tous les logs de chaque environnement et de les traiter en production.

Voici un schéma explicatif de l'application AOD pour sa future mise en production. Elle ne correspond pas à l'architecture finale du projet. Cependant sans les outils pour récupérer les logs de plusieurs environnements, nous nous contentons de déployer cette solution dans un premier temps.



Une autre partie du projet qui n'a pas encore été développé est la recommandation de la configuration Spark. En effet l'idée de cette recommandation est basée sur des algorithmes de Machine Learning qui prédirait le temps de chaque stage en fonction de sa configuration et des données en entrée. Nous voulons prédire pour une application, sa configuration optimale en fonction de ses anciens lancements.
Cette fonctionnalité n'a pas été développé car pour entraîner les modèles de Machine Learning, il faut beaucoup de logs Spark. Cependant nous n'avons pas encore eu l'occasion d'automatiser la collecte des logs.

J'avais commencé à développer des algorithmes de Machine Learning pour prédire la durée de chaque stage, cependant le nombre de données était trop peu nombreuses pour que les recherches faites à ce sujet soit pertinente.
Nous nous sommes concentrés sur le monitoring d'une application Spark dans un premier temps. Une fois tous les outils déployés et les logs récupérées, les algorithmes de Machine Learning vont pouvoir être entraîné.



6. Conclusion

Dans le cadre de ma formation d'ingénieur à l'EFREI, j'ai réalisé mon stage technique au sein de la Société Générale. Ce stage a pour but de faire découvrir aux élèves ingénieurs un environnement technique professionnel ainsi que les différentes méthodes appliquées par l'entreprise, ce qui fut le cas lors du mien. Au vu de ces nombreuses semaines passées au sein de l'équipe BDE, je tire un bilan plus que positif de mon expérience. En effet, la richesse de la mission qui m'a été confiée m'a permis d'élargir mon éventail de compétences et de parcourir un ensemble de notions qui me paraissent aujourd'hui primordiales. De ce fait, j'ai pu participer à l'entièreté du cycle de vie d'un produit à travers la participation en équipe à la totalité des phases d'un projet de développement.

La principale difficulté que j'ai rencontrée au début de ma mission a été la confiance que je pouvais avoir en mes capacités en tant que Data Engineer, et plus particulièrement au sein d'une grande entreprise telle que la Société Générale, et ce, pour deux raisons.

La première étant de prendre en main les métriques de Spark et de comprendre tous les tenants et aboutissants du projet. J'ai dû assimiler très vite les connaissances fonctionnelles et les mettre en pratique via AOD et son Dashboard. La documentation et l'aide que m'ont fourni mes collègues m'ont permis de m'adapter assez et de pouvoir avancer comme je voulais sur le projet.

La deuxième, la mise en place du projet. Dans une grosse société comme la Société Générale, il arrive souvent que pour que les projets avancent, il faut l'aide de plusieurs équipes externes au projet, et ces procédures peuvent prendre du temps. Comprendre les composants que nous devions utiliser a pris du temps car nous devions passer par des réunions avec toute l'équipe et d'autres membres des équipes externes pour nous expliquer certains processus.

J'ai réussi à surmonter ces difficultés, et j'en suis particulièrement fière. Si je ne devais choisir qu'une seule des réalisations dont je suis le plus fière, le déploiement de la CI/CD du projet. Ces étapes dans un projet était encore floue pour moi, notamment le déploiement de la partie Big Data que je n'avais pas eu l'occasion de voir dans mes précédentes expériences. Selon moi, la meilleure façon d'apprendre une notion est de pratiquer continuellement. Tout au long de ce rapport, j'ai pu mettre en évidence mes réalisations dans le cadre de ce stage technique en tant que Data Engineer. Je suis convaincue que mon entente avec l'équipe et ma rapidité, à comprendre le besoin de l'équipe ont permis de mener à bien mes missions dans les plus brefs délais. Ce stage m'a permis de me rendre compte de mon potentiel et d'accroître mon autonomie ainsi que ma productivité, mais également de tester mes limites.

Au début de mon stage, j'appréhendais la situation de télétravail dans laquelle nous étions car elle ne facilite pas le contact avec ses collègues. Rencontrer son équipe via Skype sans pouvoir mettre un visage sur les noms affichés fut un réel obstacle dans l'intégration selon moi.

Les réunions quotidiennes durant lesquelles nous faisions le point sur nos avancées me permettaient de garder le cap sur les tâches que je devais accomplir et également sur le reste de mon équipe. Grâce aux outils de gestion d'équipe comme JIRA que notre Product Owner gérait, je savais quelles étaient les tâches les plus importantes. Les points hebdomadaires avec mon maître de stage me permettaient également d'avoir un point de vue extérieur sur l'avancement du projet, notamment sur son déploiement.

A la fin de ma période de stage, nous avons donc dû réfléchir à la façon d'effectuer un transfert de connaissances afin que le projet puisse continuer une fois partie. Pour l'instant, Othman SEFRAOUI va prendre la suite du projet, il sera sûrement aidé d'un autre collègue pour pouvoir avancer au plus vite sur la mise en production du projet.

Ce stage m'a ainsi appris des éléments essentiels au niveau des méthodes de travail, j'ai pu m'améliorer en matière de rédaction de comptes rendus ou de dossiers de conception. Grâce à ces expériences, j'arrive à mieux sélectionner l'information que je veux exprimer ce qui se traduit par une meilleure communication avec mes interlocuteurs. Je me suis aussi rendu compte de l'importance du travail d'équipe. Les missions regroupant souvent plusieurs services différents, la communication est primordiale si l'on veut avancer dans le projet. Un bon travail d'équipe se traduit alors par le fait d'avoir des objectifs clairs, une bonne communication et une organisation précise.

Selon moi, il est important de réaliser ce stage technique afin de pouvoir se confronter dès le début aux différentes problématiques technologiques et surtout dans le monde d'une entreprise comme la Société Générale. Nous permettant par la suite d'apporter son expérience dans ce domaine bien que l'on ne se destine pas à faire uniquement un métier technique durant sa carrière d'ingénieur.

J'arrive à la fin de mes études et je vais pouvoir évoluer dans le monde professionnel. J'ai décidé que les prochaines années seraient l'occasion pour moi d'apprendre le plus de choses possibles sur les technologies du Big Data et surtout sur la technologie du cloud.

Glossaire

API: Application Programming Interface

AOD: Application Optimization Dashboard

BDE: Big Data Expertise

BDX: Big Data Experience

BU: Business Unit

CD: Continuous Delivery

CI: Continuous Integration

CPLE: Compliance

DDS : Digital & Data Service

DFIN: Direction Financière du développement

DRHG/COMM: Direction des Ressources Humaines du Groupe & Communication du Groupe

DSI : Direction des systèmes d'informations

GBIS : Global Banking & Investors Solutions

GBSU : Global Business Service Unit

HDFS : Hadoop File System

IDE : Integrated development environment

IGAD : Inspection Générale et Audit

IHM: Interface Homme-Machine

IITM: Innovation, Technologie & Informatique

LCL: Le Crédit Lyonnais

MVC: Model Vue Controller

Parsing: Analyse

SU: Service Unit

S3: Simple Storage Service

RESG : Ressources et de la Transformation Numérique Groupe

RISQ : Direction des Risques du Groupe

SEGL : Secrétariat Général du Groupe

UX : User Experience

Bibliographie

Pour des raisons de confidentialité, certaines informations sont extraites des sites internes à la Société Générale et n'apparaissent pas dans la Bibliographie.

- https://fr.wikipedia.org/wiki/Soci%C3%A9t%C3%A9_g%C3%A9n%C3%A9rale
- https://fr.wikipedia.org/wiki/Apache_Oozie
- https://fr.wikipedia.org/wiki/Scrum_(d%C3%A9veloppement)
- https://www.openshift.com/blog/jenkins-pipelines
- https://fr.wikipedia.org/wiki/Apache_Spark
- https://blog.sodifrance.fr/un-modele-de-versionnement-efficace-avec-git/
- https://www.redhat.com/fr/topics/devops/what-is-ci-cd#:~:text=L'approche%20CI%2FCD%20permet,continue%20et%20le%20d%C3%A9ploiement%20continu.