

DEVELOPPEMENT D'UN OUTIL D'ANALYSE DE LOGS

STAGE TN10

Université de
Technologie de
Compiègne

Département
Informatique

5 février - 19 juillet 2024

Suiveur UTC : M. Marc Shawky
Tuteur : M. Tristan Campserveux

Entreprise : SOLUTEC
Adresse : 63 av. Galline
69100 Villeurbanne



REMERCIEMENTS

Je souhaite commencer ces remerciements en exprimant ma sincère reconnaissance envers SOLUTEC pour m'avoir accueillie au sein du Lab'SOLUTEC pour mon stage de fin d'études. Ce fut une expérience incroyablement enrichissante, tant sur le plan professionnel que personnel et je suis très heureuse d'avoir eu l'opportunité de participer à un projet aussi passionnant.

Je tiens à remercier tout particulièrement mon tuteur de stage, responsable du Lab'SOLUTEC, M. Tristan Campserveux. Son accompagnement, ses conseils avisés et son approche bienveillante m'ont permis de progresser tout au long de ce stage.

Un immense merci à mon coéquipier et binôme de stage, Yanni Mansour. Travailler à ses côtés a été un véritable plaisir, nos échanges quotidiens, notre collaboration et notre partage de compétences ont largement contribué à la réussite de ce projet.

Je souhaite également exprimer ma reconnaissance à notre Product Owner, Baptiste Flysaintemarie, pour son rôle déterminant dans la gestion du projet.

Je remercie également l'UTC et en particulier mon suiveur de stage, Marc Shawky, pour son accompagnement tout au long de mon parcours.

Enfin, je souhaite adresser un grand merci à l'ensemble des stagiaires et intervenants que j'ai eu la chance de côtoyer durant ce stage. Leur esprit d'équipe, leur bienveillance et leurs partages ont rendu cette période non seulement formatrice, mais également très agréable. L'ambiance de travail et l'entraide que nous avons partagées resteront parmi mes meilleurs souvenirs.

SOMMAIRE

Introduction	3
Première partie : Présentation de SOLUTEC et de son Lab'	4
I. Solutec	4
II. Le Lab'SOLUTEC	7
III. IMPALA	10
Deuxième partie : Les missions du stage.....	11
I. Sujet	11
II. Planning.....	13
III. Contributions.....	17
IV. Outils et technologies	18
V. Prise de recul	22
Troisième partie : La création d'un outil d'analyse de logs.....	23
Organisation du projet	23
Première sous-partie : Collecte des données et centralisation.....	25
Deuxième sous-partie : Log parsing.....	28
Troisième sous-partie : Log Mining	34
Quatrième sous-partie : Site Web	45
Conclusion.....	49

RESUME TECHNIQUE

J'ai effectué mon stage de fin d'étude d'ingénieur sur une durée de six mois au sein de SOLUTEC, entreprise située à Villeurbanne. SOLUTEC est une Entreprise de Service du Numérique spécialisée en informatique principalement pour des clients grands comptes. J'ai pu intégrer le Lab'SOLUTEC, pôle d'innovation et de professionnalisation, en tant que développeuse.

L'objectif de ce stage est de réaliser un outil d'analyse de logs pour permettre le traitement de ceux-ci dans les Direction des Systèmes d'Information (DSI) de différentes entreprises. Un log est un fichier textuel numérique contenant des événements ou des messages générés par un système ou une application. Les logs sont souvent utilisés pour le suivi, le diagnostic ou l'analyse des activités et des erreurs. Pour cela, j'ai travaillé en équipe avec mon binôme co-développeur et notre Product Owner (PO) chargé du suivi du projet. Cette entraide et cet accompagnement ont permis le bon déroulé et la réussite du projet.

La stage a débuté par une phase de compréhension et délimitation du sujet, notamment avec la création d'un dossier de cadrage de projet. Par la suite une phase de formation a été réalisée pour permettre une bonne prise en main des outils et des informations. La réalisation du projet a ensuite été découpée en quatre parties. Tout d'abord, la première partie consiste en la collecte des données sur les différentes machines virtuelles disponibles au sein de l'environnement de stage. Les données collectées proviennent de différents services informatiques essentiels pour assurer l'activité d'une entreprise ainsi que sa sécurité. La deuxième partie repose sur le parsing de ces logs obtenus. Les logs semi-structurés sont ainsi découverts à l'aide d'un algorithme de machine learning, Drain 3, pour être enregistrés dans une base de données. La troisième partie concerne quant à elle le log mining, soit la recherche d'anomalies ou d'incidents sur ces données de logs à l'aide de différents algorithmes de machine learning non supervisés. Finalement, la dernière partie de ce projet s'intéresse à l'application informatique web, qui se décompose en différentes pages. Cette interface web s'adresse aux DSI et utilise les logs parsés ainsi que les informations d'anomalies obtenues pour créer une visualisation et permettre différentes actions sur ces données.

INTRODUCTION

Dans le contexte actuel de transformation numérique, les systèmes informatiques génèrent une quantité croissante de données, parmi lesquelles les logs occupent une place prépondérante. Les logs, ou journaux de bord informatiques, sont des enregistrements séquentiels et horodatés des événements et des activités des systèmes, des applications et des utilisateurs. Ils constituent une source essentielle d'informations pour la gestion, la maintenance et la sécurité des infrastructures informatiques. Les besoins croissants en matière de gestion et d'analyse des logs ont conduit au développement de solutions innovantes pour exploiter au mieux ces données.

Avec la prolifération des environnements informatiques distribués et des architectures microservices, la gestion des logs est devenue un défi de plus en plus complexe. La diversité des sources, la volumétrie croissante des données générées et la nécessité d'une réactivité en temps réel imposent des exigences nouvelles en matière d'outils d'analyse de logs. Dans ce contexte, la création d'un outil qui permet la gestion des logs devient non seulement pertinente mais également indispensable pour les entreprises cherchant à maintenir la disponibilité et la fiabilité de leurs systèmes.

Ce rapport de stage s'inscrit dans le cadre de la création d'un outil d'analyse de logs. Ce projet a été motivé par le besoin de fournir une solution complète et efficace pour la collecte, la centralisation, le parsing, la détection d'anomalies et la visualisation des logs dans une application web. Le développement de cet outil repose sur plusieurs phases techniques essentielles, chacune visant à répondre à des problématiques spécifiques rencontrées dans les différentes étapes du processus.

La première étape de ce projet consistera en la collecte et la centralisation des logs provenant de diverses sources. Dans un environnement où de très nombreux services informatiques génèrent un nombre impressionnant de logs par jour, le regroupement de ceux-ci relève d'un enjeu essentiel pour les DSI. Une fois les logs centralisés, il est indispensable de les parser, c'est-à-dire de les découper en éléments structurés selon leurs attributs spécifiques. Le parsing des logs permet de traduire les informations brutes en données exploitables, puis de les stocker dans une base de données structurée.

L'analyse des logs ne se limite pas à leur simple collecte et parsing. La détection d'anomalies est une étape cruciale qui permet d'identifier les comportements inhabituels ou suspects au sein des systèmes informatiques. Différents algorithmes d'apprentissage automatique non supervisés seront étudiés pour cette tâche. Ces algorithmes permettent d'anticiper et de réagir rapidement aux incidents, améliorant ainsi la sécurité et la fiabilité des infrastructures.

Enfin, pour rendre les informations contenues dans les logs accessibles et compréhensibles, une interface web servira de visualisation. La visualisation joue un rôle clé dans l'analyse des logs, offrant une vue d'ensemble des performances du système et des éventuelles anomalies détectées.

Ce rapport détaillera tout d'abord, l'entreprise d'accueil ainsi que l'équipe intégrée ; par la suite nous observerons les missions et méthodes établies pour la mise en pratique du projet, finalement nous étudierons les étapes de la création de l'outil d'analyse de logs, en mettant en lumière les défis techniques rencontrés et les solutions mises en place. De la collecte et centralisation des logs au parsing, en passant par la détection d'anomalies et la visualisation, chaque phase du projet contribuera à fournir une solution complète pour la gestion des logs.

PREMIERE PARTIE : PRESENTATION DE SOLUTEC ET DE SON LAB'

I. SOLUTEC

Solutec est une Entreprise de Service du Numérique (ESN) fondée en 1991, spécialisée en informatique et plus précisément en conseil dans le domaine de l'ingénierie. Son activité principale se focalise sur la transition numérique et la digitalisation des entreprises, ainsi que sur l'accompagnement de projets et le conseil en informatique. Elle est également impliquée dans la production et l'exploitation de systèmes d'information.

1) Son positionnement

Solutec est implantée dans plusieurs agences en France : Lyon, son siège social et Paris où se concentrent la majorité de ses collaborateurs. De plus, trois autres agences plus petites sont présentes en France, Nantes, Bordeaux et Toulouse.

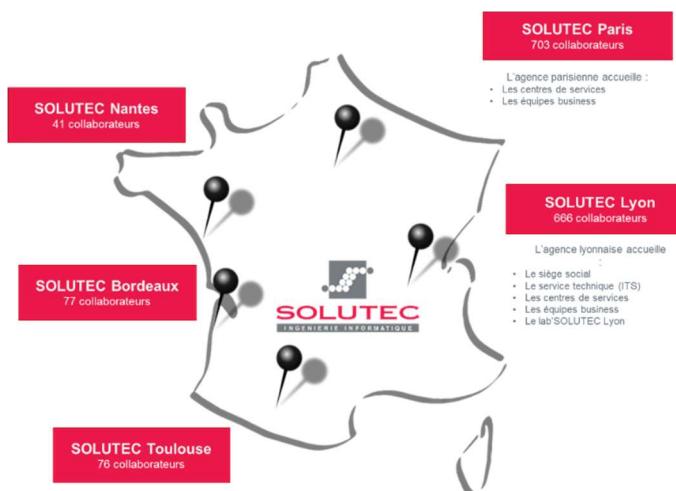


Figure 1 : L'implantation de Solutec en France

L'agence lyonnaise accueille sept services distincts, ayant chacun un rôle clé pour l'entreprise.

Tout d'abord, le siège social qui correspond au domicile juridique de Solutec.

Ensuite, le service technique appelé Infrastructure des Télécommunications et de Support (ITS), qui correspond au service informatique de l'entreprise. Il est chargé de mener à bien les études d'infrastructure, de garantir la cybersécurité et de développer des logiciels internes. En outre, ITS joue un rôle auprès de certains clients de Solutec en leur offrant un accompagnement tout au long de leurs projets et de les aider à les maintenir en bon état de fonctionnement par la suite.

Les équipes business ou commerciales sont également essentielles au fonctionnement de Solutec. Ces équipes s'occupent de l'interaction avec les clients de Solutec et recherchent de

nouvelles missions pour les consultants. Ils sont également à même de prospector des nouveaux clients, c'est-à-dire en proposant les services de l'entreprise à de nouvelles entreprises.

Par la suite, le service d'avant-vente a pour objectif de vendre les outils aux clients. Ce service propose notamment les versions packagées et préconçues par l'entreprise.

Ensuite, il existe le service de recrutement. Ce service est chargé de trouver de nouveaux profils pour l'entreprise. L'objectifs étant de trouver des consultants, des ingénieurs d'affaires, mais également tous les profils liés aux besoins de l'activité de Solutec comme des personnes chargées des ressources humaines ou de l'administration.

De plus, se trouve le service administratif qui gère administrativement l'entreprise que ce soit pour la paye, les facturations ou la comptabilité.

Finalement, on retrouve le Lab'SOLUTECH, intégré au service des recrutements qui accueille et forme les stagiaires. Ce service est détaillé par la suite car c'est là que j'ai pu effectuer mon stage.

Ces différents services permettent le bon fonctionnement de l'entreprise et notamment la gestion des différents employés.

2) Ses collaborateurs

Le nombre de collaborateurs est en nette augmentation durant ces dernières années, avec une progression de 8.8% en 2023. Cette croissance est importante mais maîtrisée. En effet, en janvier 2024, il y a 1550 collaborateurs en France, ce qui correspond à une entreprise, pour le pays, de taille moyenne.

Solutec étant une entreprise de conseil, la majeure partie de ses employés sont des consultants. En effet, les consultants correspondent à 85% des employés de l'entreprise. La plupart d'entre eux travaillent directement en mission chez le client. Ces employés intègrent les locaux et les équipes du client. Le reste des consultants, qui correspondent à une minorité, travaillent en mission dans un centre de services. Le travail du consultant est alors en interne, dans les locaux de Solutec pour des projets clients au forfait. Ces projets ne sont pas réalisés directement chez le client, ceux-ci n'ayant pas la possibilité ou l'envie d'intégrer les équipes chez eux. C'est par exemple le cas pour Enedis, dont un service est dédié pour eux dans les locaux de Solutec à Villeurbanne.

Le fonctionnement de Solutec correspond à celui d'une entreprise de conseil. En effet, les collaborateurs effectuent des missions qui durent en général entre six mois et trois ans et qui sont souvent suivies d'une période d'intercontrat durant laquelle les consultants se forment dans l'attente d'une nouvelle mission.

Le reste des collaborateurs n'effectuant pas un travail de consultant, représentant 15% des employés sont dédiés à la structure. Ces employés s'occupent du fonctionnement de l'entreprise. Notamment, la direction par Jean Bruyère le président de Solutec et Nicolas Invernizzi, le directeur général. Ainsi que les différents services présentés précédemment.

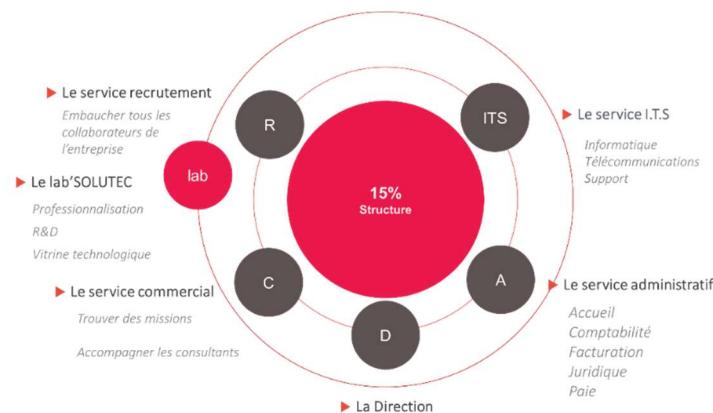


Figure 2 : L'organisation de Solutec

Par ailleurs, le chiffre d'affaires de Solutec est en croissance de 17%, atteignant 100 millions d'euros en 2022. Ce chiffre est assez important pour une entreprise de cette taille.

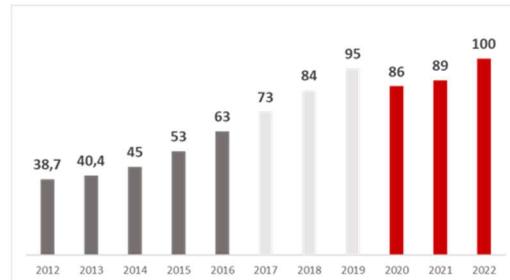


Figure 3 : Evolution du chiffre d'affaires

L'entreprise dispose donc de nombreux employés qui exercent chacun divers métiers selon les besoins des clients.

3) Ses métiers

Solutec exerce son activité dans quatre domaines principaux du conseil en informatique.

- **Le développement applicatif** : Ces métiers correspondent à la mise en place, à la conception ou encore la maintenance de projets d'applications informatiques et logicielles. Ces projets interviennent sur toutes les phases du projet que ce soit lors de l'analyse du besoin ou encore du développement mais également lors des tests, de la mise en production et aussi pour la maintenance des projets. Les projets sont divers que ce soit des développements web, des applications mobiles, des applications clients ou encore du big data.

- **L'infrastructure et l'exploitation** : Ces métiers interviennent davantage sur l'infrastructure des services informatiques. Ces projets correspondent à la gestion de l'architecture du système informatique d'une entreprise ou encore de ses outils. Elle intervient également dans la sécurité et le cloud. Ces projets s'occupent également de gérer les changements, de coordonner les mises en production, mais également de gérer les incidents et les crises. Enfin, Solutec s'engage dans une démarche d'amélioration continue en effectuant une veille technologique constante.

- **L'accompagnement et la gestion de projet** : Ces métiers encadrent les projets afin de permettre la réussite des différentes phases. Que ce soit lors de l'étude des opportunités d'un projet, en passant par l'analyse des besoins et la création du cahier des charges et des

spécifications et jusqu'à la conduite du changement. Ces métiers constituent un accompagnement pour le client ainsi que la coordination d'un projet de bout en bout.

- Le **business développement** : Ces métiers sont orientés sur la partie commerciale de l'entreprise. C'est-à-dire, d'être en lien avec les clients pour obtenir de nouveaux projets que ce soit par la prospection de nouveaux clients mais également en fidélisant les clients actuels. Il y a également un aspect de management, d'analyse des besoins et de réponses aux appels d'offres.

Ces différents métiers ont pour objectifs de répondre aux besoins des différents clients tout en se démarquant de la concurrence.

4) Ses clients et concurrents

Solutec a une clientèle très variée, aussi bien dans le secteur public que privé, dans le secondaire et tertiaire principalement. En effet, l'entreprise fournit ses services à plus de 140 clients grands comptes issus de divers secteurs d'activité, tels que les médias, les télécommunications, les services publics, la finance, le commerce de détail et l'industrie. Parmi ses clients les plus importants se trouvent EDF, Enedis, la SNCF, mais encore Carrefour, le Crédit Agricole France Télévisions, La Métropole Grand Lyon, M6, Orange, SFR, Thalès, Veolia, Vinci ou même Volvo.

Solutec s'appuie sur différentes valeurs pour offrir un service de qualité à ses clients. L'entreprise privilégie la proximité avec ses clients pour mieux comprendre leurs ambitions, et l'écoute active pour entretenir des relations humaines solides. La transparence et une vision à long terme sont également essentielles pour Solutec, qui souhaite établir des partenariats durables avec ses clients. Enfin, l'entreprise garantit une qualité de service à haute valeur ajoutée et un professionnalisme en constante évolution.

Les principaux concurrents de Solutec sont d'autres ESN et cabinets de conseil, tels que Capgemini, Sopra Steria, Astek ou bien IBM.

Solutec est donc l'entreprise dans laquelle j'ai pu réaliser mon stage. Au sein de celle-ci se trouve un service dédié au stagiaire qui m'a accueilli sur toute la durée du stage.

II. LE LAB'SOLUTEC

Le Lab'SOLUTEC est un pôle de Solutec ayant pour objectif l'accueil, l'encadrement et la formation futurs ingénieurs en stage de fin d'étude.

1) Son histoire et ses objectifs

Le Lab a été créé en janvier 2016 pour accueillir des stagiaires issus de différentes écoles d'ingénieurs en France. C'est un pôle de professionnalisation et d'innovation destiné au monde

de l'entreprise. L'objectif étant de recréer un espace d'entraide, formateur et professionnel où différents types de métiers se croisent.

C'est un lieu de création, en effet, les sujets sont innovants sur des thèmes variés. L'objectif étant de répondre à des besoins provenant de divers clients pour tester ou bien d'être le point de départ d'une idée. Le but du Lab étant d'être une vitrine technologique sur des sujets différents en informatique.

C'est aussi un lieu de professionnalisation, très formateur et autonome avec des responsabilités à prendre. C'est un lieu où on apprend des méthodes de travail rigoureuses et des outils adaptés pour pouvoir l'appliquer par la suite.

Finalement, c'est un lieu de partage. En effet, les équipes sont pour la plupart des binômes où des trinômes, tous encadrés par des product owner et un consultant ayant également le rôle de tuteur. Les équipes interagissent entre elles au sein d'un open-space et s'entraident. Le Lab permet la mutualisation des compétences et expériences. De plus, d'autres consultants, clients ou encore ingénieurs d'affaires interagissent avec les stagiaires ou organisent des présentations.

2) Son organisation

Le Lab est organisé à l'image d'une société. Il y a trois encadrants : Tristan Campserveux le Responsable du Lab, Antoine Richard et Léna Deroche des consultants-tuteurs de Solutec. Leurs missions sont multiples. Tout d'abord, de s'occuper des stagiaires du Lab mais également de répondre aux clients, de promouvoir le Lab et les projets qui s'y trouvent.

Le reste du Lab est composé de trente-six stagiaires répartis selon différents postes et projets.

Tout d'abord, un chef de projet interagit avec tous les projets du Lab. Sa fonction est de reproduire le rôle du client en assistant aux réunions d'avancements et suivant en parallèles les différents projets. Il a également pour mission de démarcher des clients pour leur proposer de nouveaux projets.

Ensuite, trois proxy product owner qui représentent le lien entre le client et les développeurs. Chacun est en charge de plusieurs projets. Ils organisent les daily quotidien et correspondent davantage avec les tuteurs et le chef de projet. Leur rôle est de s'occuper du management des projets, organiser les sprints et leurs revues et permettre le bon déroulement du projet.

Par la suite, deux DevSecOps et un DevOps s'occupent des outils informatiques du Lab, les DevSecOps étant axés davantage sur le côté sécurité.

Finalement, on retrouve vingt-neuf développeurs répartis dans douze projets différents et variés. Ces projets sont composés d'une à quatre personnes.

Voici, ci-après, l'organigramme du Lab qui permet de mieux comprendre les interactions entre les métiers qui sont exercés.

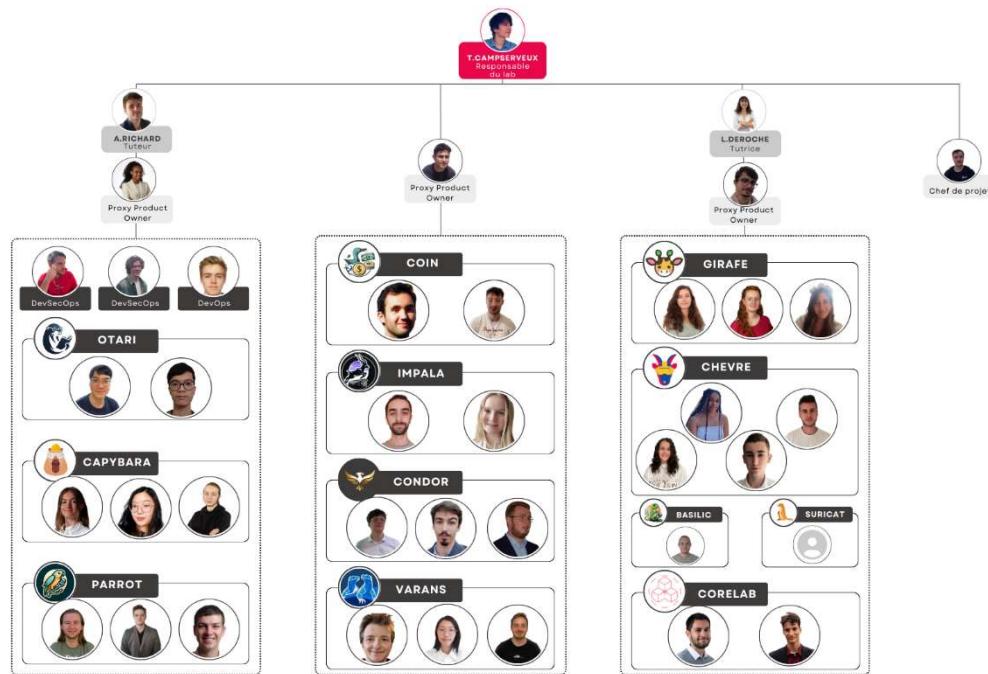


Figure 4 : Organigramme du Lab'

Par ailleurs, le Lab est un lieu de passage et d'échange de connaissance.

Au sein du Lab'SOLUTECH, de nombreux ateliers sont proposés. En effet, tout au long du stage des formations ont été organisées. Que ce soit par les tuteurs ou bien par d'autres projets nous avons pu assister à diverses présentations, comme les formations sur des outils pour Figma ou GitLab, mais aussi sur les tests, les méthodes agiles ou encore des formations de l'équipe de Green IT sur l'accessibilité, l'empreinte carbone, des fresques du numérique et également une formation sécurité provenant des DevSecOps. Nous avons aussi assisté à une formation RGPD de l'équipe Varans.

Ces diverses formations permettent un partage des connaissances au sein du Lab et un enrichissement sur de nombreux sujets liés au monde du numérique.

Nous avons également rencontré de nombreux ingénieurs d'affaires venus observer et comprendre nos réalisations que nous présentions. Nous avons aussi eu des présentations et retours d'expérience provenant de consultants de SOLUTEC. Finalement, des clients sont venus observer nos projets et nous questionner tout au long du stage.

L'ambiance au sein du Lab est très chaleureuse avec la mise en place d'une journée de Teambuilding au début du stage pour permettre par le biais de serious games de mieux comprendre les méthodes de travail utilisées à Solutec mais également faire connaissance avec tous les collègues de travail.

Impala est l'équipe que j'ai intégrée et dans laquelle je me suis formée et j'ai évolué durant ce stage.

III. IMPALA



L'équipe IMPALA (Implémentation d'Analyse de Logs Avancée), dans laquelle j'étais durant ce stage se compose d'un consultant-tuteur Tristan Campserveux, d'un product owner Baptiste Fly Sainte Marie et de deux développeurs Yanni Mansour et moi-même. Cette équipe, entité du Lab Solutec est aussi en interactions avec le chef de projet et les DevOps et DevSecOps.

Figure 5 : Logo d'IMPALA

Le projet répond à un besoin exprimé par la ville de Lyon. Cependant, avant de poursuivre leur collaboration avec Solutec, ils souhaitaient s'assurer du bon fonctionnement d'un tel projet. Ainsi, le projet a été lancé en février 2024 au Lab. Le but de ce projet est de répondre au besoin du Lab et notamment d'être un outil utilisable par les DevOps et DevSecOps qui voudraient pouvoir mieux gérer les différents services informatiques du Lab. Un objectif futur serait de l'élargir à la DSI de Solutec puis de l'envisager pour des structures et des clients externes.

Les objectifs du projet sont donc gérés de manière autonome par l'équipe et suivis par notre tuteur.

DEUXIEME PARTIE : LES MISSIONS DU STAGE

I. SUJET

L'objectif global de ce stage est la réalisation d'un outil d'analyse de logs. Cet outil est une interface destinée à un client pour qu'il puisse gérer les logs provenant de ses différents systèmes informatiques.

1) Définition

Un log ou journal correspond à un fichier informatique textuel enregistré de manière séquentielle. Les données conservées sont des évènements datés d'un service informatique. Ces évènements permettent de suivre l'activité interne du processus et ses différentes actions avec les autres services.

Les logs sont générés de manière automatique par les softwares et permettent de communiquer différents types d'informations, comme des connexions à un serveur, des erreurs, ou des modifications. L'analyse de ces fichiers se révèle être particulièrement utile pour les entreprises, qui sont devenues dépendantes de l'analyse de données, leur permettant de mieux comprendre le fonctionnement de leur infrastructure et prendre les décisions adéquates.

Le projet s'appuie donc sur ces données de logs particuliers à des systèmes informatiques comme GitLab, HAProxy, Sonarqube...

```
Dec 10 07:28:25 LabSZ sshd[24263]: Failed password for root from 112.95.230.3 port 40388 ssh2
Dec 10 07:28:25 LabSZ sshd[24263]: Received disconnect from 112.95.230.3: 11: Bye Bye [preauth]
Dec 10 07:28:25 LabSZ sshd[24265]: Invalid user utsims from 112.95.230.3
```

Figure 6 : Exemple de logs Openssh

Les logs sont des données difficiles à déchiffrer par un humain car trop nombreuses. En effet, pour certains systèmes plusieurs lignes de logs sont produites chaque seconde, ce qui rend la recherche d'incidents fastidieuse. De plus, les logs peuvent être difficiles à comprendre et prendre en main, en effet, les logs sont souvent mécanisés et suivent un schéma textuel encadré empêchant une compréhension simple.

Ainsi, des outils informatiques peuvent être utilisés afin de permettre une compréhension rapide des informations contenues dans ces logs, que ce soit par la présence de graphiques ou encore par la détection d'anomalies par le biais d'algorithmes de machines Learning. En étant supplié par l'IA la détection de problèmes et la compréhension des logs peut être facilitée. Notre objectif était donc de réaliser cet outil destiné aux entreprises.

1) Contexte

Ce projet se situe dans la continuation d'un POC (Proof Of Concept ou preuve de concept) intitulé LOGNESS, réalisé en 2023. Ce POC avait été commandé par le Grand Lyon, un client de Solutec qui souhaitait obtenir un outil d'analyse de logs.

En effet, le Lab'SOLUTEC est un environnement de création et d'innovation qui permet notamment de réaliser des projets interne. Ces projets sont réalisés à titre de vitrine qui permet aux clients d'aborder un projet dans un contexte d'entreprise pour par la suite le demander en implémentation à SOLUTEC. Ainsi, notre projet d'analyse de logs est avant tout destiné au Lab'SOLUTEC mais également, par cet intermédiaire aux futurs clients intéressés et notamment le grand Lyon. Par extension, le projet sera aussi proposé à la DSI de Solutec pour gérer ses différents services.

2) Histoire des analyses des logs

L'analyse de logs n'a pas toujours été un processus automatique. Aux premiers jours d'Unix, dans les années 1970, il n'existe pas d'outils pour agréger des fichiers de logs provenant de différentes sources, ou pour surveiller leur génération en temps réel. De plus, aucune interface graphique ne permettait de visualiser de manière claire et résumée les données issues des fichiers logs.

Au fur et à mesure de l'évolution des systèmes informatiques, la quantité de logs générés, à des emplacements différents a conduit à la création d'outils permettant la centralisation de ces fichiers. Cette centralisation a rendu leur accès plus facile, mais tout en ne supprimant pas la nécessité d'une analyse manuelle, qui peut être réalisée à la suite de la mise en place du traitement automatique.

Aujourd'hui, les infrastructures informatiques se diversifient de plus en plus, et leur taille augmente en parallèle, en incluant une grande variété d'applications et de systèmes, générant une quantité massive de logs. L'analyse de ces logs de façon manuelle serait une tâche bien trop pénible. En effet, un fichier de log ne possède pas un format universel et dépend très souvent du système l'ayant généré. De plus, la taille de ces fichiers, en quantité d'informations, est dans la majorité des cas très importante et n'est plus adaptée à un traitement et une lecture manuelle de ceux-ci.

C'est dans ce contexte qu'ont émergé un grand nombre de solutions d'analyse de logs automatisées, souvent couplées à des fonctionnalités assurées par une intelligence artificielle. Ces solutions, pouvant être open-source et gratuites ou commerciales et payantes, sont aujourd'hui largement développées et diversifiées, chacune avec ses avantages et inconvénients.

Ainsi, les DSI, expriment de plus en plus le besoin d'avoir à leur disposition un outil d'analyse de logs performant et adapté à leur infrastructure particulière. En effet, la DSI est le point central de génération de fichiers de logs au sein d'une entreprise, et leur exploitation leur est cruciale.

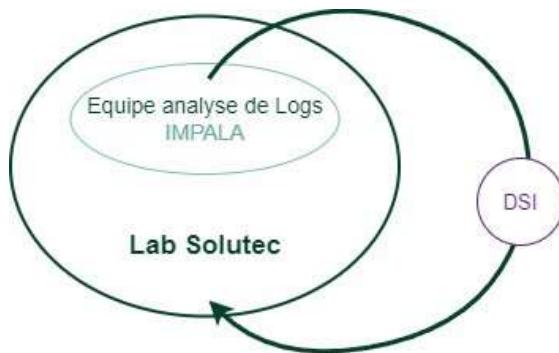


Figure 7 : Position d'IMPALA au sein de SOLUTEC

Le projet du développement d'un outil d'analyse de logs a été mis en place à la suite du POC LOGNESS, en ayant cependant la volonté de créer le nouvel outil en partant de zéro. Ce choix a été fait pour faire évoluer le concept de LOGNESS afin d'obtenir un outil fonctionnel et adapté à une utilisation concrète par des utilisateurs.

3) Besoins du projet

Concrètement, l'analyse de logs est souvent utilisée pour couvrir trois axes majeurs :

- L'identification et l'anticipation d'incidents : en analysant les logs, il est possible de détecter des anomalies sur le réseau pouvant indiquer un problème de sécurité afin de le résoudre le plus rapidement possible et d'empêcher sa récurrence.
- L'analyse de comportements utilisateurs : les logs permettent de dresser le profil des utilisateurs d'un site ou d'une application, en fonction du comportement adopté. Cette analyse sert par la suite à améliorer l'expérience utilisateur.
- L'analyse de performances : il est courant d'utiliser les logs pour comprendre comment les ressources sont réparties au sein d'une infrastructure globale, afin de pouvoir répartir aux mieux celles-ci et améliorer les performances du système.

Comme vu précédemment, l'analyse des logs reçus par la DSI est essentielle. Ces logs sont trop nombreux pour être traités individuellement et un outil doit être créé pour permettre à la DSI d'utiliser l'information qui se cache derrière ces données.

L'objectif principal du projet est de créer notre propre outil d'analyse de logs, qui puisse de manière automatisée traiter ces différentes lignes de données et renvoyer une information plus condensée et lisible par un employé de la DSI. Cet objectif principal s'inscrit donc dans la recherche de l'algorithme le plus performant en termes de rapidité et d'utilisation d'espace mémoire. Un objectif secondaire est donc d'être capable de comprendre le format de ces données ainsi que les besoins de la DSI quant à ces données de logs.

Voici un diagramme de bête à corne permettant de cadrer au mieux l'analyse des besoins.

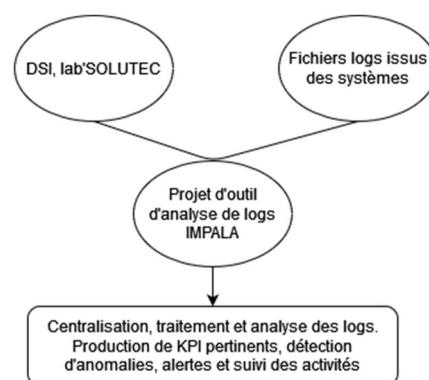


Figure 8 : Diagramme du projet

Afin d'organiser nos tâches durant ce stage et de permettre un bon développement de la solution nous avons prévu notre travail à l'aide d'un planning.

II. PLANNING

Le stage s'est déroulé pendant le semestre de printemps 2024. Il a eu lieu du 5 février au 19 juillet et a duré 24 semaines. Le stage s'est déroulé en compagnie de mon binôme Yanni Mansour sur presque toute la durée du stage. Nous avons choisi d'organiser notre projet en fonction du temps dont nous disposions pour le réaliser. Cependant, nous avons principalement

travaillé en méthode agile, ainsi l'objectif final et les étapes de notre projet n'étaient pas fixes dans le temps.

1) Cadrage et état de l'art du projet

Nous avons débuté notre travail par une phase de cadrage, au cours de laquelle nous avons élaboré un dossier préliminaire pour mieux définir et délimiter notre mission, nommé dossier de cadrage de projet. Cela nous a permis de revoir les besoins et les attentes. Nous avons également effectué un état de l'art des solutions existantes. Ces différentes étapes ont favorisé une gestion optimale, permettant de définir les objectifs, les périmètres, ainsi qu'une description fonctionnelle et conceptuelle de notre projet.

Lors du cadrage du projet, deux possibilités d'orientation quant à la conduite du projet s'offraient à nous. En effet, le projet étant à sa première élaboration et ayant pour vocation à être repris, la réalisation entière et complète de toutes les attentes définies et les besoins n'était pas prévu sur la durée du stage. Il était donc primordial d'organiser notre travail selon notre durée de stage. La première possibilité était donc la formation d'un squelette du projet, en réalisant chacune des étapes de manière distinctes sans rentrer dans les détails pour obtenir une première version fonctionnelle mais peu approfondie du projet. La deuxième solution consistait à enchaîner les étapes dans l'ordre des besoins du projet en approfondissant davantage ces étapes pour obtenir une version plus détaillée et utilisable mais en enlevant la possibilité d'obtenir un travail fini et utilisable jusqu'au bout.

Nous avons choisi la deuxième option, préférant nous pencher davantage sur des détails plus complexes et mieux approfondir nos tâches plutôt que de rester en surface sur les objectifs.

2) Formation et bases du projet

Après avoir mis en place le cadre du projet nous avons entamé le sprint 0, un sprint entièrement destiné à notre formation sur les outils et les technologies que nous utiliserons sur le projet.

Nous avons pu apprendre les langages informatiques que nous ne connaissions pas et que nous aurions besoin d'utiliser pour ce projet, notamment ce qui concerne l'interface web pour moi et plutôt ce qui concerne le machine learning pour Yanni.

Nous avons pu nous former en conditions réelles en réalisant un mini-projet durant cette phase de formation. Notre mini-projet sur les jeux de sociétés du lab centralisait toutes les connaissances que nous avions besoin d'apprendre pour mettre en pratique notre projet par la suite. Que ce soit l'interface web, la connexion avec une base de données ou encore une partie IA relié à un script de clusterisation, qui suggérait des jeux de sociétés adaptés.

Durant cette phase, nous avons également mis en place le projet en réalisant un poker planning et une première planification de sprint.

3) Début des sprints

Notre phase de réalisation et développement du projet s'est déroulé du 13 avril 2024 jusqu'à la fin du stage. Nous avons pu effectuer cinq sprints d'une durée approximative de 2 à 3 semaines.

Notre projet initial se découpe en plusieurs phases ainsi les moments importants de notre développement se trouvent à la réalisation de ces étapes.

Le développement de l'outil de parsing a été plus complexe que prévu dépassant le temps du premier sprint. Le développement web a commencé lors du deuxième sprint et s'est étalé sur le reste du stage. En parallèle le développement des algorithmes de détection d'anomalies a été effectué en plusieurs parties dédiées.

La partie qui m'a le plus intéressée est celle de recherche d'anomalie en utilisant divers algorithmes de machine learning. Cette partie m'a beaucoup plu par sa dimension de recherche et par les diverses méthodes employées. J'ai également beaucoup apprécié la collecte des données par sa nouveauté pour moi de par la navigation dans les machines virtuelles et la création de programme de collecte en continu. Cette partie du stage était nouvelle pour moi, n'ayant pas eu beaucoup l'occasion de le pratiquer durant mes études.

L'étape qui m'a semblé la plus fastidieuse fut le développement web et notamment la partie de frontend, soit la visualisation et l'apparence du site et notamment les nombreux tests que nous devions réaliser pour être sûr d'une bonne implémentation.

La partie du parsing des logs fut également difficile, au vu de la très faible documentation de l'algorithme drain 3 et de la présence de certains bugs. Cependant, sa réalisation fut très enrichissante et le travail accompli très satisfaisant.

Dans l'ensemble, les différentes étapes traversées furent très diversifiées, très formatrices et le travail accompli m'a beaucoup plu.

4) Dates importantes

Des dates importantes furent ajoutées au calendrier en cours de progression. Notamment, le 11 juin pour la venue d'un client de la SNCF intéressé par notre projet. Puis le 4 juillet pour la journée de présentation des différents clients de SOLUTEC. De plus, au cours du stage de nombreuses personnes comme des responsables d'affaires, des clients ou d'autres collaborateurs sont venus voir notre progression et assister à la présentation de nos réalisations.

Ces journées furent l'occasion pour nous de présenter plus précisément notre travail aux clients, ainsi qu'une date butoir d'un certain avancement à réaliser.

Ces expériences ont permis d'améliorer l'explication et la synthétisation du projet sur lequel nous travaillons. De plus, l'arrivée d'un avis extérieur sur le projet sur lequel nous passons

tout notre temps permet de remettre en question certains points et d'apporter de nouvelles perspectives. Ces expériences furent enrichissantes.

5) Répartition du travail

Le développement du projet a été organisé autour de missions ou tâches principales qui correspondent aux étapes nécessaires à la réalisation de la solution. Ces missions comprennent la collecte des logs, le parsing, l'apprentissage sur ces logs via des algorithmes de machine learning et la visualisation de ceux-ci dans une interface web.

L'approche que nous avons choisie, impliquait un développement progressif, ajoutant les fonctionnalités successivement. Cette méthode m'a permis d'explorer toutes les fonctionnalités de l'application finale tout en entrant dans le détail pour certaines.

Durant ce stage, j'ai collaboré avec Yanni, mon binôme de travail et nous avons organisé nos tâches entre nous deux. Au début de chaque sprint, nous nous répartissions les missions en fonction de nos préférences. Cette répartition n'était jamais fixe et avait tendance à évoluer au cours du sprint selon nos difficultés ou de l'avancement du sprint. Pendant les phases de sprint, nous consacrons la majeure partie de notre temps au développement des fonctionnalités définies lors de la planification du sprint avec notre Product Owner. À la fin de chaque sprint, nos tâches se concentraient sur la rédaction des tests unitaires et d'intégration, ainsi que sur la documentation. Nous pouvons voir le déroulement du projet via le diagramme de GANTT ci-dessous.

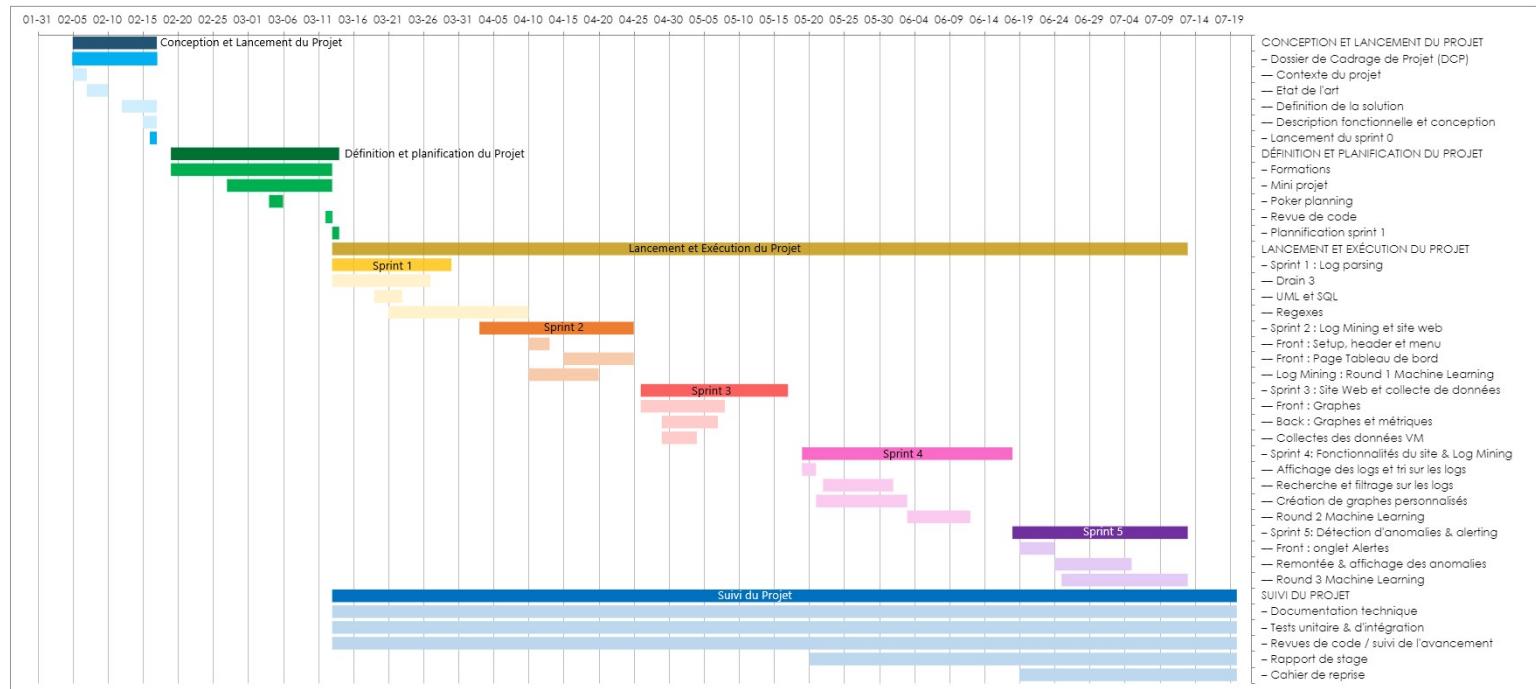


Figure 9 : Diagramme de GANTT du projet IMPALA du 5 février au 19 juillet 2024

Le travail en binôme avec Yanni a permis au projet de meilleurs résultats, combinant nos différentes connaissances et permettant une entraide continue sur tout le long du stage. Ce travail en équipe fut enrichissant pour le projet mais également pour moi qui a appris le travail collectif et l'entraide.

Durant le stage, nous avons alloué environ 40% de notre temps au développement de l'application web (back-end et front-end), 45% au développement des modèles de machine learning pour la détection d'anomalies et pour le parsing, et 15% à la rédaction des tests, de la documentation et à d'autres tâches annexes. Le diagramme de Gantt résume l'ensemble des tâches effectuées.

Ces tâches ont été effectuées en équipe pour permettre tout le déroulement du projet.

III. CONTRIBUTIONS

Le projet a été réalisé en binôme avec un autre développeur, en accompagnement sur la partie de gestion de projet par un product owner et supervisé par notre consultant tuteur.

1) Réalisations avant et après le projet

Nous avons débuté le projet à notre arrivé à partir de rien. Un POC avait été réalisé en amont mais nous ne nous sommes pas appuyé dessus car nous devions avoir des idées neuves et différentes de ce qui avait été réalisé. Nous avons donc fait une analyse de l'existant sur le marché en cherchant notamment des inspirations sur ce qui se faisait déjà, puis nous avons débuté le projet sur de nouvelles bases.

L'objectif de ce projet consiste en la création d'une application Web complète ainsi qu'en la mise en place de récupération des logs des différents services informatique, parsing de ceux-ci et visualisation sous différentes formes : graphes, métriques, anomalies.

Le projet à la fin de ce stage est en cours de développement. Les trois pages principales de l'application ont été réalisées. La collecte et le parsing des logs ont été effectués sur certains services informatiques. Ceux-ci ont été enregistrés dans notre base de données et sont visualisables par le biais de notre application.

La partie de parsing a été réalisé pour certains types de logs, notamment pour HAProxy, GitLab, Windows, OpenSSH et Sonarqube qui fournissent des clusters satisfaisants et des résultats conservés dans notre base de données.

La partie web contient certaines des pages principales de notre application avec notamment le tableau de bord, la visualisation des graphes et des métriques et la fonction de recherche avancé, ainsi qu'une partie dédiée aux alertes des anomalies détectées. La partie concernant mes équipes ainsi que la connexion et les volets d'aides sont encore à prévoir dans la future reprise de l'application.

Le partie d'apprentissage des données et de détection des anomalies a été réalisée et certaines anomalies sont détectées. Il pourra être intéressant d'approfondir avec des données labellisées. De plus, la prédiction des incidents ou le diagnostic des échecs reste encore à réaliser.

La partie de centralisation des logs et réception en temps réels n'est pas encore implémentée mais pourra l'être facilement à l'aide d'un outil comme fluentd ou ELK qui sont open source. L'interface de connexion doit également être implémentée, pour permettre aux utilisateurs une sécurisation des données.

Une documentation précise du code a été réalisée au fur et à mesure des développements et un cahier de reprise est disponible sur bookstack, pour permettre une meilleure reprise du projet.

Notre projet a pour objectif d'être repris par d'autres stagiaires pour continuer le travail mis en place et mettre en œuvre de nouvelles fonctionnalités.

2) Travail en équipe

Le travail réalisé a été mené en équipe. Notre product owner Baptiste a géré la bonne réalisation du projet, en s'occupant de notre product backlog à jour sur Jira (outil de ticketing), nos revues de sprint et tout ce qui correspondait à l'encadrement du stage. De plus, il a également créé la maquette de visualisation de l'interface de notre site web.

Le développement de l'application a été mené en binôme avec Yanni, et nous avons collaboré étroitement tout au long du projet. Nos différentes réalisations ont été effectuées en parallèle, nous permettant de progresser efficacement. Nous avons divisé le travail en sous-tâches spécifiques, chacun se concentrant sur des aspects particuliers de l'application, tout en veillant à ce que notre travail se complète harmonieusement. Cette répartition des tâches a permis de couvrir un large éventail de fonctionnalités et d'optimiser notre productivité. Nous avons également construit une forte entraide, partageant régulièrement nos avancées, et nous apportant un soutien mutuel pour résoudre les problèmes rencontrés. Grâce à cette collaboration, nous avons pu échanger des idées, améliorer nos compétences respectives, et garantir une qualité constante dans le développement de l'application.

Pour permettre au mieux la réalisation de notre projet, nous nous sommes appuyés sur différents outils qui nous ont accompagnés et aidés dans la mise en place du projet.

IV. OUTILS ET TECHNOLOGIES

Différentes technologies ont été choisies afin de mener à bien notre projet. Ces choix ont été fait lors du cadrage du projet en comparant les différentes possibilités selon différents critères nécessaires à notre réalisation.

1) Techniques utilisées

Les techniques choisies pour le développement web ont été considérés selon leur facilité de prise en main et d'implémentation notamment pour la partie de visualisation et d'interface

utilisateur. Le langage de développement est le Javascript. Ce langage était nouveau pour moi et j'ai pu m'y former dès le début du stage. Le framework utilisé pour le frontend est le Vue js 3 connu pour sa simplicité d'application et sa possibilité de créer de nombreux composants. L'HTML et le CSS sont également utilisable avec le javascript pour encadrer l'affichage. Concernant le backend, l'utilisation combiné de Node js et d'Express ont permis une bonne synchronisation des pages et une facilitation de correspondance avec la base de données.

Le choix de la base de données a également été réfléchi. En effet, les bases de données sont soit en SQL (Structured Query Language) pour des données structurées, qui peuvent être découpées en tables avec des colonnes et des champs qui reviennent pour chacun des logs. Les bases des données peuvent également être en NoSQL (Not Only SQL) pour des données non structurées. Les logs étant des données textuelles semi-structurées il était plus difficile de faire un choix. De plus au vu du nombre de logs, le NoSQL aurait pu être une bonne option. Cependant après réflexion et notamment la prise en compte de la taille du Lab correspondant à une petite structure nous avons défini que le SQL avec PostgreSQL serait une bonne alternative. Il serait possible dans les futures améliorations du projet de basculer vers un autre type de base de données.

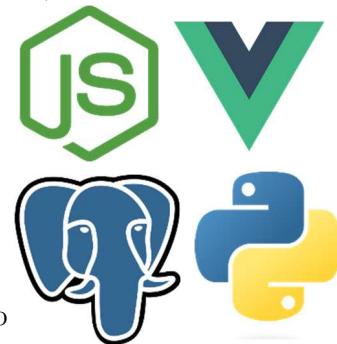


Figure 10 : Langages de programmation choisis

Finalement, concernant le parsing des données et la détection d'anomalies réalisés à partir d'algorithmes de Machine Learning le choix du Python a logiquement été adopté car c'est le leader actuel dans ce domaine, permettant grâce à de nombreuses librairies comme pandas, numpy, matplotlib, scikit-learn et pleins d'autres encore de lire les données, utiliser des algorithmes et obtenir des résultats.

2) Tests et documentation

Durant chacune des phases de développement que ce soit pour chacun des langages présentés ci-dessus nous réalisions différents tests et une documentation des fonctions et variables utilisées pour permettre l'assurance de nos résultats et la bonne reprise du projet. Ces tests étaient principalement des tests unitaires soit des tests sur nos composants ou bien des tests d'intégrations pour vérifier la bonne liaison entre nos éléments.

Pour le javascript nos tests ont été réalisés avec Cypress, qui permet une visualisation des tests sur l'interface, pour les tests d'intégration et Vitest pour les tests unitaires. Pour le Python nos tests ont été traités avec la librairie unittest.

Le coverage de ces tests peut être affiché et rassemblés via l'interface de visualisation de Sonarqube qui permet une bonne

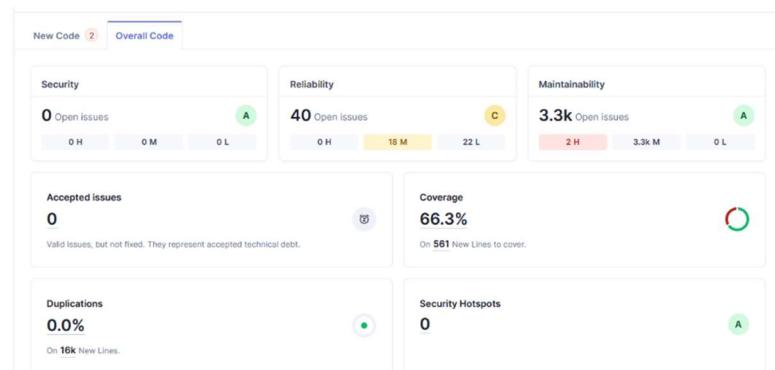


Figure 11 : Extrait du diagnostic Sonarqube du projet

analyse du code selon les différentes fonctionnalités testées que ce soit en python ou bien en javascript.

Concernant la documentation, pour la partie python elle a été réalisée avec Doxygen fonctionnant directement dans le code. Tandis que pour le javascript et l'interface web la documentation utilisée est celle de Vitepress. Finalement une documentation globale de notre projet est visualisable sur Bookstack.

3) Outils de travail

Pour notre projet nous avions besoin de différents outils pour le mener à bien. Pour notre développement nous avons utilisé l'IDE (Interface de développement) Visual Studio Code. Pour gérer et sauvegarder au mieux notre développement et permettre un travail d'équipe nous avons utilisé GitLab. Notre système d'exploitation au sein de l'entreprise était Windows et pour accéder aux VM ou lancer notre code nous utilisions WSL soit Linux pour Windows.



Figure 12 : Outils utilisés pour la programmation



Nos discussions étaient centralisées avec l'outil Teams et tous les documents créés étaient enregistrés via Sharepoint qui permet une modification en temps réel de plusieurs utilisateurs.

Figure 13 : Outils utilisés pour les comptes rendus et la communication

Finalement nos mots de passe étaient centralisés et

gérés par VaultWarden.

Un autre outil utilisé est Figma qui permet la création de maquette dédié au développement à partir de ces designs. Ces maquettes réalisées par notre PO, nous ont servi de support tout au long de l'avancement de notre projet.

4) Méthode Agile

Dans le but de former les stagiaires aux méthodes de management en entreprise, le lab'SOLUTECH fonctionne suivant la méthode Agile (SCRUM et Kanban). Le logiciel utilisé est Jira, de l'éditeur Atlassian.



La méthodologie Agile est une méthode de management de projet souvent utilisée dans le développement logiciel. L'objectif de cette méthode est de promouvoir la collaboration entre les clients et les équipes mais également au sein de l'équipe. Son principe est la mise en avant du besoin et la mise en perspective du développement sur le besoin, tout en permettant une grande flexibilité et de nombreuses modifications tout au long du projet selon



Figure 14 : Résumé des étapes de la méthode Agile

l'avancée effectuée et les nouvelles attentes. Cette méthode favorise également la communication entre les différentes parties prenantes.

a) SCRUM

Concernant la méthode SCRUM, le projet a été découpé en plusieurs cycles de 2 semaines nommés « sprint ». Lors d'un sprint sont développées les fonctionnalités déterminées comme prioritaires par le product owner. Ces « User Stories » sont définies lors de la création du Product Backlog qui consiste à lister l'ensemble des tâches à effectuer et les classer selon leur difficulté d'implémentation et leur valeur métier définie par le client (dans notre cas le chef de projet).

Pendant toute la durée du sprint, notre équipe s'est retrouvée tous les matins pour les daily. Ces réunions journalières sont organisées par notre PO et permettent d'expliquer ce qui a été fait la veille, ce qui est prévu pour la journée et les éventuelles difficultés rencontrées. À la fin de chaque sprint a lieu une rétrospective permettant ainsi de résumer l'ensemble des difficultés rencontrées et les points d'amélioration pour le prochain sprint. Tout au long du sprint, un Scrum Master est désigné au sein de l'équipe, son rôle est d'animer les daily et la rétrospective. Nous avons été Scrum Master à tour de rôle durant chacun des sprints avec Yanni.

b) KANBAN

La méthode Kanban vient du japonais qui signifie étiquettes. Cette méthode de management se traduit par la structuration du sprint sous forme de tâches qui correspondent à des tickets. Elle permet notamment de structurer ce qui sera accompli durant un sprint et d'attribuer les différentes tâches au sein de l'équipe. Cette méthode nous a permis d'organiser notre travail et de se répartir les différentes tâches selon nos envies.

Un certain nombre de points d'effort est défini chaque sprint (25 par développeur travaillant sur le sprint) et les tickets sont choisis selon les points d'effort, les valeurs métier correspondantes et les volontés des développeurs et du PO. Ainsi une phase de planification de sprint est réalisée avant chaque sprint, pour choisir les tickets qui seront effectués pour le sprint.

Ces étiquettes sont ensuite rangées par état dans un tableau et ordonnées en fonction de leur priorité. Les états sont : A faire (liste des tâches qui ne sont pas commencées mais planifiées), En cours (liste des tâches en cours de traitement), A tester (liste des tâches en cours de tests), Fait (liste des tâches terminées).

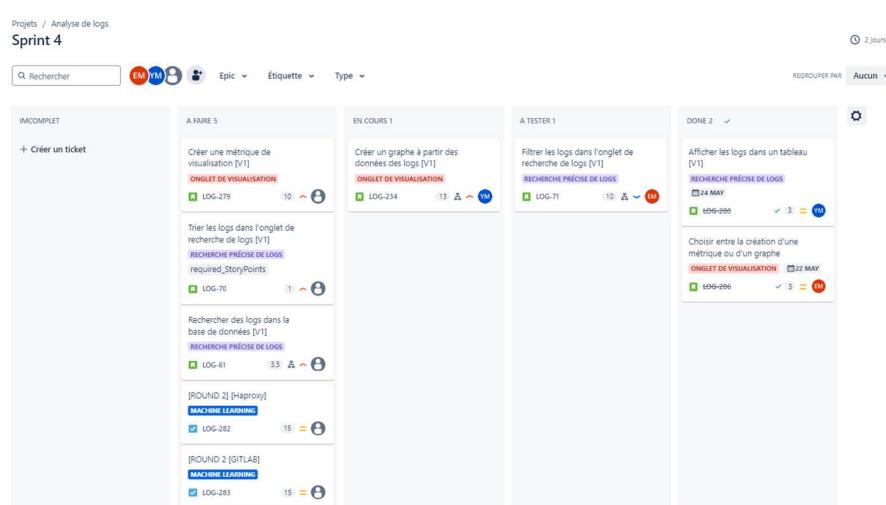


Figure 15 : Visualisation des tâches durant le sprint 4 dans l'outil Jira

V. PRISE DE RECUL

Le travail réalisé durant ce stage est le lancement du projet d'outil d'analyse de logs. Le projet a pour vocation d'être repris par d'autres stages au cours des prochains semestres de printemps.

L'objectif final étant de le proposer au DevSecOps et DevOps du Lab pour leur permettre de gérer au mieux les différents systèmes informatiques du Lab en visualisant leur log et en étant avertis lors d'anomalies et potentiellement, dans un prochain temps, de le proposer à la DSI de Solutec. Par intermédiaire, l'objectif serait de convaincre le grand Lyon de l'intérêt de ce projet pour l'implémenter par la suite, chez eux.

Finalement, nous avons cherché tout au long de ce projet à réduire notre impact environnemental et améliorer notre projet aux besoins du développement durable, notamment grâce à l'équipe de Green IT qui a encadré et formé tous les projets du Lab à ces préoccupations.

Parmi nos actions, nous avons décidé de limiter notre utilisation de Python sur le développement de notre projet au Machine Learning au vu de son impact écologique très supérieur à d'autres langages comme le Javascript que nous avons préféré choisir pour le reste du projet, notamment pour le développement Web.

De plus, nous avons mis en place d'autres pratiques de Green IT comme l'utilisation de Lighthouse sur nos sites qui nous propose un score et des améliorations à réaliser pour augmenter ce score. Nous nous sommes notamment penchés sur l'aspect de l'accessibilité, notre score étant de 88/100 pour la page principale de l'application. Ce score a pu être amélioré grâce aux différentes pratiques de 18%. Ce score est plutôt bon comparés aux sites web connus.

TROISIEME PARTIE : LA CREATION D'UN OUTIL D'ANALYSE DE LOGS

IMPALA est un outil d'analyse de logs qui réalise toutes les étapes clés de l'obtention des logs jusqu'à la visualisation de ceux-ci dans une application web. Le projet a débuté en mars et a été précédé d'une phase de définition et planification qui a permis d'organiser le projet.

ORGANISATION DU PROJET

La phase de planification du projet s'est étendue sur deux semaines et a permis d'explorer les différentes possibilités et configuration pour que le projet se déroule au mieux. De plus, il était nécessaire de bien comprendre les objectifs pour prendre en main le projet et pouvoir le réaliser selon les besoins.

1) Définition des besoins

Le besoin de créer un outil d'analyse de logs découle de plusieurs exigences. Tout d'abord, il y a une nécessité de tracer les actions des utilisateurs des systèmes informatiques pour comprendre leur comportement et leurs interactions. Cela permet de garantir le bon fonctionnement des applications et des services associés. Analyser et comprendre ces interactions est essentiel pour optimiser les performances et l'efficacité des systèmes.

Ensuite, il y a des impératifs de sécurité. L'analyse des logs permet de détecter les failles de sécurité, de surveiller les tentatives d'intrusion et de répondre rapidement aux incidents de sécurité. Pouvoir identifier et corriger ces vulnérabilités est fondamental pour protéger les données sensibles et protéger les utilisateurs.

En outre, il est indispensable de diagnostiquer les pannes, les erreurs et les dysfonctionnements. L'outil d'analyse de logs doit donc être capable de fournir des informations détaillées sur les incidents techniques, permettant ainsi une résolution rapide et efficace des problèmes.

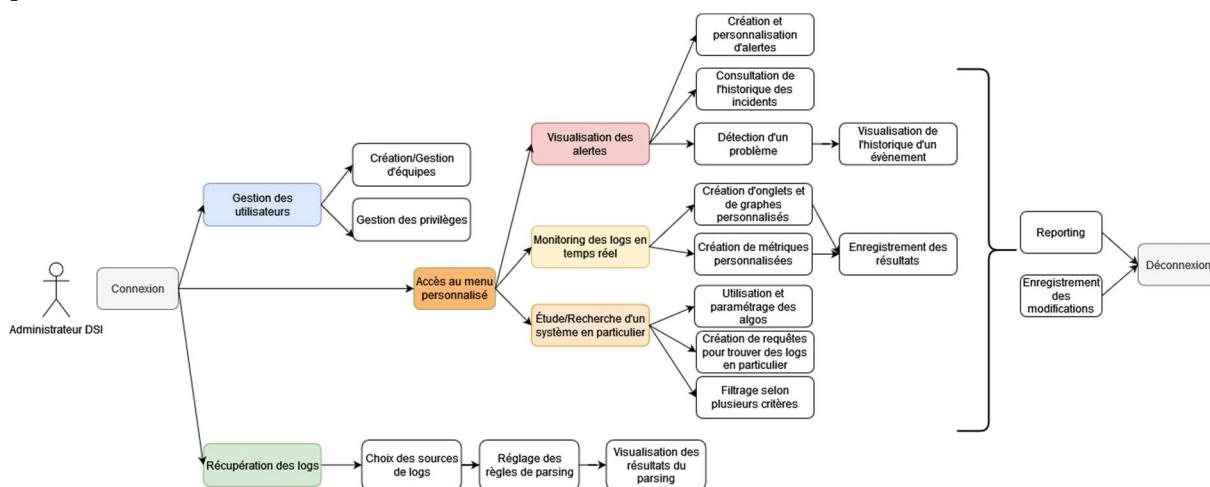


Figure 16 : Parcours utilisateur de l'application

Pour concrétiser cette vision, il a été nécessaire de définir toutes les fonctionnalités que devrait inclure la solution finale. En élaborant le parcours utilisateur de notre application, présent ci-dessus, nous avons pu identifier les étapes clés et les fonctionnalités requises. Cette approche nous a permis de structurer et de prioriser le développement des différentes composantes de l'application, garantissant ainsi une solution cohérente et répondant pleinement aux besoins identifiés.

2) Découpage en quatre phases de réalisation

Notre projet a été divisé en plusieurs sous-parties, chacune étant réalisée individuellement comme des briques ajoutées au projet. Ces briques doivent également s'emboîter harmonieusement. L'objectif est de décomposer le projet en modules pouvant être modifiés sans altérer l'ensemble, offrant ainsi une solution modulable selon les besoins des futurs clients. Ainsi, chaque partie doit être travaillée de manière indépendante.

Tout d'abord, les logs arrivent de différentes sources et doivent être reçus et centralisés pour ensuite être traités. Les logs sont donc parsés selon leur format pour être découpés en éléments dans un tableau. Cette table est stockée dans une base de données. Les données stockées peuvent ensuite être présentées via une interface web. Elles peuvent également être traitées par des algorithmes de détection et de prédiction des anomalies pour être visualisées par l'utilisateur.

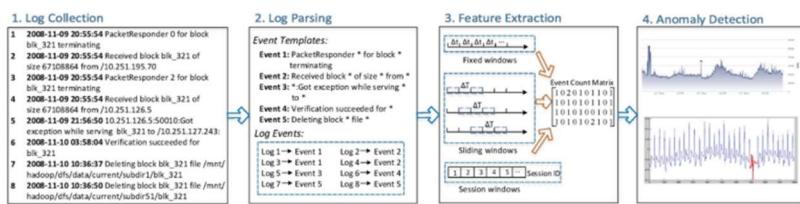


Figure 17 : Résumé des étapes à réaliser pour réaliser notre outil

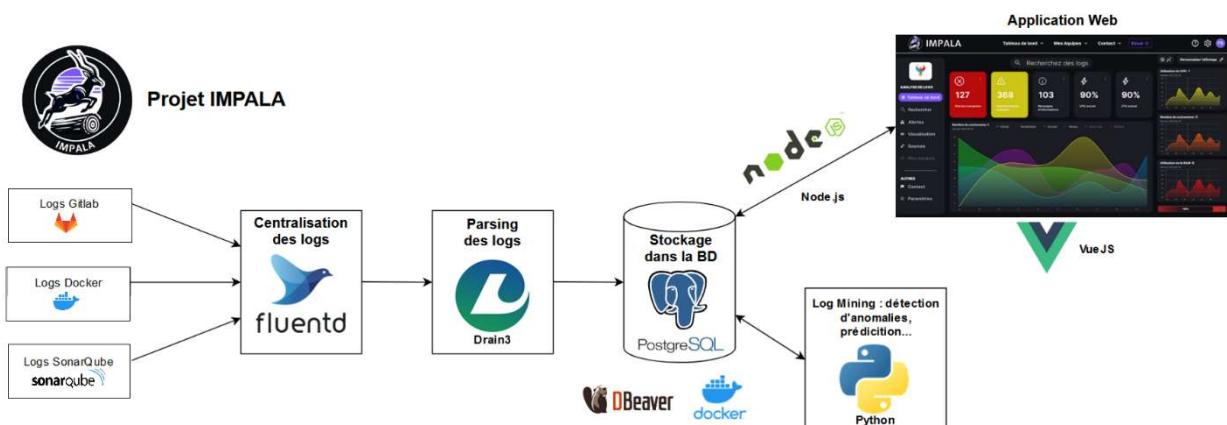


Figure 18 : Résumé des étapes de réalisation pour obtenir notre outil d'analyse de logs

Les quatre étapes principales sont récapitulées dans le schéma ci-dessus. La première partie correspond à la centralisation des données : les données sont collectées puis centralisées dans un espace unique. Ensuite, les logs peuvent éventuellement être compressés pour éviter la surcharge de l'espace de stockage intermédiaire. La deuxième étape concerne le parsing des logs, où ceux-ci sont découpés selon leurs attributs pour être stockés dans une base de données. Vient ensuite l'étape d'apprentissage sur ces données : les logs sont envoyés dans un modèle destiné à

déetecter les anomalies et les problèmes potentiels en amont. Finalement, la dernière étape consiste en la visualisation des données sous format de graphes et de métrique sur une interface web.

Ces étapes seront détaillées et explicitées dans la partie suivante. Avant cela, les méthodes employées pour réaliser ces étapes ont été recherchées et comparées.

3) Recherches effectuées

Nous avons débuté par réaliser un l'état de l'art des solutions existantes sur le marché. La première étape consistait à recenser les différents outils d'analyse de logs disponibles afin de comprendre leurs spécificités et d'identifier les pratiques courantes dans ce domaine.

Par la suite, une analyse plus poussée des outils et algorithmes à utiliser pour chacune des étapes de notre développement a été réalisée. [1][2][3][4][5]

Concernant la centralisation des logs, plusieurs outils ont été étudiés en détail. Nous avons comparé et analysé les différentes solutions disponibles sur le marché pour déterminer laquelle répondrait le mieux à nos besoins. Finalement, nous avons opté pour Fluentd, un outil open-source reconnu pour sa capacité à récupérer, centraliser et unifier les données des logs sous format JSON. Cet outil se distingue par sa simplicité d'utilisation, ce qui le rend facilement intégrable au sein des DSI des futurs clients.

Pour le parsing des logs, nous avons effectué une recherche approfondie et consulté de nombreux articles comparant divers algorithmes de parsing. Parmi les solutions performantes identifiées, on peut citer IPLoM, AEL, et Spell. Cependant, notre choix s'est porté sur Drain 3, un algorithme open-source réputé pour son efficacité dans la majorité des cas de parsing des logs. En outre, Drain 3 est compatible avec Python, ce qui facilite son intégration dans notre environnement de développement et converge avec nos connaissances.

Enfin, pour la détection des anomalies de manière non supervisée sur les données de logs, il existe différents algorithmes qui ont été proposés. Ceux-ci sont étudiés dans une partie suivante qui y est dédiée.

PREMIERE SOUS-PARTIE : COLLECTE DES DONNEES ET CENTRALISATION

La première étape essentielle de ce projet consiste à collecter et centraliser les logs provenant de diverses sources. Cette phase permet de rassembler toutes les données nécessaires en un seul endroit, facilitant ainsi leur traitement ultérieur et assurant une base solide pour les étapes suivantes.

1) Les types de logs

Le nombre de logs différents a considérablement augmenté ces dernières années, en grande partie au vu de l'explosion des multiservices et des micro services. Cette évolution a entraîné une

augmentation significative des fichiers de logs à analyser et regrouper, rendant la collecte de ces données cruciale.

Les logs, générés par divers systèmes informatiques, doivent être collectés de manière ciblée pour assurer un traitement efficace. Il est essentiel d'identifier les services spécifiques à surveiller et d'extraire les logs pertinents.

Au sein du Lab, plusieurs services sont utilisés, chacun fonctionnant sur des machines virtuelles. Voici la liste exhaustive de ces services :

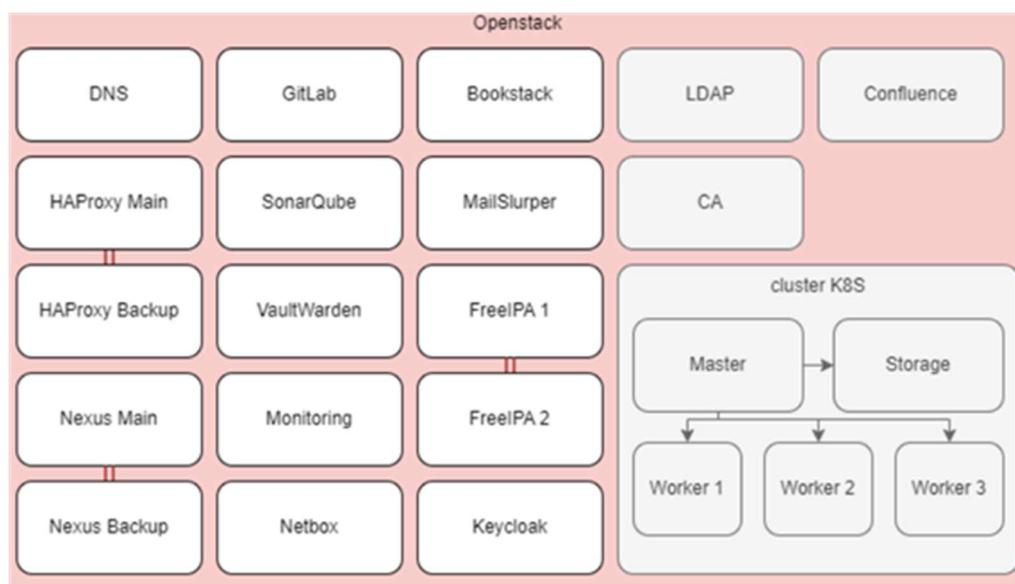


Figure 19 : Services informatiques du Lab présents sur des VM (en gris : dépréciés ou en cours de relance)

Ces machines virtuelles génèrent un grand nombre de logs. La première étape a donc consisté à cibler les VM où le besoin de surveillance était le plus important, ainsi qu'à identifier les types de logs les plus propices à être utilisables.

Après diverses discussions avec les DevSecOps, futurs utilisateurs de l'application, nous avons identifié les logs à cibler : GitLab, HAProxy, Sonarqube, Valtwarden, FreeIPA, Keycloack et Nexus. A l'avenir, il pourrait être intéressant d'explorer également Kubernetes. De plus, les logs OpenSSH et Windows, couramment traités dans les outils d'analyse de logs classiques, posent des défis supplémentaires. Ils sont plus difficiles à obtenir étant donné qu'ils sont gérés par les services informatiques de Solutec et non pas par le Lab.

Ces logs sont variés et fournissent différentes informations. Ces données sont particulièrement utiles en cas d'anomalies ou de problèmes, car elles proviennent des systèmes les plus critiques, retraçant les actions des utilisateurs ou les fonctionnements des services.

2) La collecte des logs

Chaque machine virtuelle produit différents types de logs. Il y a d'abord les logs directement liés aux services hébergés, mais aussi ceux relatifs au fonctionnement de la VM et aux authentifications. Nous nous sommes concentrés sur les logs générés par les services des VM, en particulier ceux décrits dans les documentations techniques spécifiques, suivant un schéma propre à chaque type.

Par exemple, sur la machine virtuelle HAProxy, les logs d'intérêt se trouvent dans le fichier nommé haproxy.log.

```
debian@haproxy-prod-node1:/var/log$ ls
alternatives.log      bttmp.1           debug       haproxy.log.1    messages     syslog.6.gz
alternatives.log.1    cloud-init.log   debug.1     haproxy.log.2.gz  messages.1   syslog.7.gz
alternatives.log.2.gz  cloud-init-output.log  debug.2.gz  haproxy.log.3.gz  messages.2.gz  user.log
alternatives.log.3.gz  cron.log        debug.3.gz  haproxy.log.4.gz  messages.3.gz  user.log.1
alternatives.log.4.gz  cron.log.1      dpkg.log    haproxy.log.5.gz  messages.4.gz  user.log.2.gz
apt                   cron.log.2.gz   dpkg.log.1  haproxy.log.6.gz  private     user.log.3.gz
auth.log               cron.log.3.gz   dpkg.log.2.gz journal    README     user.log.4.gz
auth.log.1              cron.log.4.gz   dpkg.log.3.gz kern.log   runit      wtmp
auth.log.2.gz            daemon.log     dpkg.log.4.gz kern.log.1  syslog
auth.log.3.gz            daemon.log.1   dpkg.log.5.gz kern.log.2.gz syslog.1
auth.log.4.gz            daemon.log.2.gz  dpkg.log.6.gz kern.log.3.gz syslog.2.gz
bootstrap.log          daemon.log.3.gz  faillog    kern.log.4.gz  syslog.3.gz
bttmp                 daemon.log.4.gz  haproxy.log lastlog   syslog.4.gz
```

Figure 20 : Liste des différents fichiers de logs présents la machine virtuelle HAProxy

En réalité, les logs sont générés en continu sur les VM, avec de nouveaux logs produits chaque minute. Pour conserver ces logs, nous avons mis en place des logrotates. Ces logrotates permettent d'enregistrer quotidiennement les logs dans un fichier, qui est ensuite conservé pendant une semaine avant d'être supprimé définitivement.

C'est pour cette raison qu'il est intéressant de sauvegarder et traiter ces logs avant qu'ils ne soient perdus. Il est nécessaire de les sauvegarder ailleurs que dans la VM qui a une taille de stockage limitée. Pour ce faire, nous avons d'abord configuré des tâches crontab qui enregistrent quotidiennement un fichier de logs compressé, renommé en fonction de la date du jour. Chaque crontab, spécifique à chaque type de log, exécute un script bash sur la machine virtuelle. Ce script bash localise les fichiers de logs générés par logrotate, les renomme, les compresse et les enregistre dans un emplacement défini.

Une fois ces étapes réalisées, les fichiers de logs peuvent être extraits à l'aide de la méthode scp (secure copy), qui permet de copier de manière sécurisée et chiffrée entre différents systèmes. Ainsi, les fichiers peuvent être transférés de la machine virtuelle vers nos machines personnelles pour une utilisation immédiate.

Selon les différents types de logs, des méthodes d'extraction spécifiques ont été nécessaires, car les VM ne fonctionnent pas toutes de la même manière.

3) La centralisation

Afin de traiter efficacement les informations contenues dans les logs, il est essentiel de les obtenir rapidement, bien avant la fin de la journée ou du mois. Ainsi il est important

d'automatiser le processus afin de récupérer les logs peu de temps après leur génération. La mise en place du traitement des logs en temps réel passe par une méthode de centralisation.

La centralisation des logs garantit la sécurité de ces logs. Centraliser les logs garantit non seulement leur sécurité, mais permet également une réponse plus rapide aux incidents et un gain de temps considérable lors des futures analyses.

Après une étude nous avons conclu que le système le plus performant pour les DSI des futurs clients de notre outil d'analyse de logs est Fluentd. Cet outil open-source collecte les logs directement sur les VMs et les centralise en un point unique. Grâce à sa modularité, Fluentd facilite l'envoi direct des logs vers le système de parsing.

La collecte des logs est une étape essentielle au bon fonctionnement de l'infrastructure informatique d'une entreprise. En obtenant les logs rapidement et en temps réel, la réactivité face aux incidents est améliorée. Une fois ces logs collectés, ils peuvent être analysés pour en extraire des informations.

Avant cette étape d'analyse, les logs doivent être parsés puis stockés dans une base de données.

DEUXIEME SOUS-PARTIE : LOG PARSING

Une fois les logs obtenus et centralisés il est important de les parser afin d'étiqueter les différentes informations qu'ils contiennent. Le parsing consiste à découper chaque ligne de logs selon les attributs présents. Cette opération permet de traduire le fichier de log afin de lire, indexer, et stocker les données. Le but du parsing est d'enregistrer ces données dans une base de données structurée en fonction des attributs détectés.

1) Méthodologie

Chaque type de logs possèdent une syntaxe particulière, nécessitant un traitement distinct pour chacun. En effet, les données contenues dans les logs varient selon le système d'origine. Cependant, certains points communs existent entre les logs, comme une date détaillée à la seconde près. Ils peuvent également contenir d'autres informations communes, telles qu'un PID ou une adresse IP, bien que certains attributs soient spécifiques au système. Par exemple, pour HAProxy, les attributs frontend et backend sont propres à son fonctionnement.

Nous avons donc opté pour une approche consistant à enregistrer les logs dans

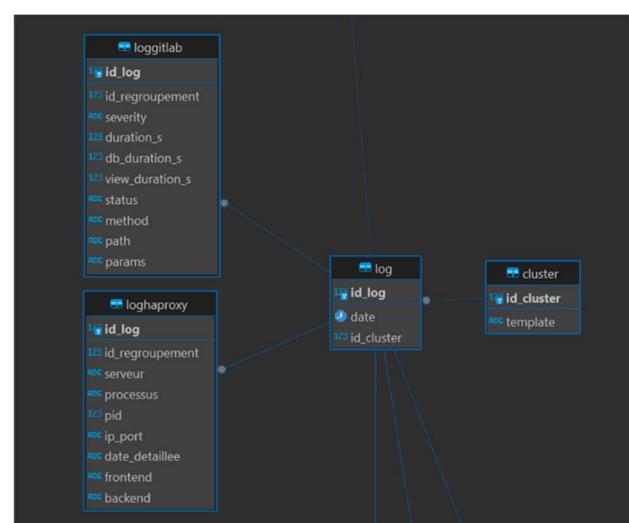


Figure 21 : Architecture du stockage des logs dans la base de données (visualisation : DBeaver)

une table générale en conservant les informations communes. En complément, des tables spécifiques sont créées pour chacun des types de logs parsés, permettant de capturer et de structurer les attributs uniques à chaque système.

2) Drain 3

Il a ensuite fallu décider de la méthode à adopter pour parser nos données. Plusieurs propositions étaient possibles. Nous pouvions retrouver les différents éléments en les détectant selon leur forme ou bien nous pouvions utiliser des algorithmes. D'après les différents articles de recherches étudiés sur le log parsing, Drain 3 semblait être l'algorithme de parsing le plus performant.

a) L'algorithme

Drain 3 est un algorithme de machine learning non supervisé spécialisé dans l'extraction de modèles ou de clusters à partir d'un flux continu de logs. L'objectif étant de transformer ces logs de format textuel, non structuré en un format utilisable pour une future analyse. Ce modèle a été conçu dans l'objectif de parser les logs.

Le principal objectif de Drain 3 est de transformer des logs non structurés en une structure organisée, facilitant ainsi l'analyse et le traitement ultérieurs. Cela se fait par l'extraction de templates qui représentent des schémas répétitifs dans les messages de logs.

L'algorithme se subdivise en plusieurs étapes en utilisant une approche hiérarchique pour organiser et traiter les logs.

Tout d'abord la tokenisation. C'est-à-dire que chaque ligne est découpée en sous-groupes, en jetons individuels. Ces jetons peuvent être des mots ou des segments particuliers et



Figure 22 : Exemple de parsing d'un log selon les différents attributs présents

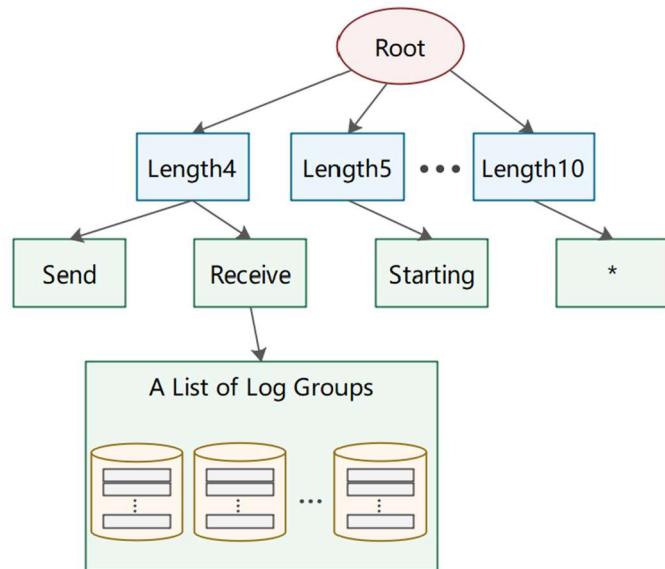


Figure 23 : Illustration de l'algorithme de Drain 3

significatifs, soit des nombres ou des groupes comme une adresse IP. Chaque ligne de log est ainsi désassemblée en plus petits composants.

Ensuite, l'algorithme utilise une structure arborescente pour organiser les différents jetons créés. La ligne est présentée sous forme d'arbre avec les jetons les plus proches qui sont regroupés en un seul noeud comme des groupes de mots par exemple.

L'arbre est surtout organisé selon sa profondeur. En effet, plus un noeud est profond, plus les jetons qui s'y trouvent sont spécifiques. Cet arbre est construit de manière à refléter les structures des logs, où chaque noeud représente un niveau de granularité différent. Les niveaux supérieurs de l'arbre capturent les aspects plus généraux des logs, tandis que les niveaux inférieurs capturent les détails plus spécifiques. Le choix de la formation de l'arbre se repose sur la structure de la phrase et ainsi les mots clés se retrouveront aux mêmes emplacements. Les mots clés étant à une même profondeur avec de mêmes parents peuvent être repérés et extraits. L'algorithme permet ainsi d'obtenir la structure du message.

En observant les différents schémas créés lors de la formation de l'arbre, Drain 3 forme des clusters à partir des chemins les plus communs empruntés dans l'arborescence. La fonction de similarité compare les nouveaux messages de log avec les templates existants pour déterminer s'il y a correspondance ou si un nouveau template doit être créé. Cette comparaison est basée sur les tokens extraits lors de la tokenization. Ainsi les structures similaires sont regroupées et les patterns sont identifiés. De cette manière il détecte également les attributs similaires qu'il regroupe.

Drain 3 s'appuie également sur une intervention humaine en utilisant des schémas de masquage. Ces schémas sont créés à partir de regex.

Cet algorithme est adaptable et évolutif, le modèle est capable de s'adapter aux nouveaux formats de logs et d'évoluer avec le temps. Les nouveaux patterns de logs sont automatiquement intégrés au modèle grâce à l'ajout de nouveaux templates ou à la mise à jour des templates existants.

Finalement, l'un des avantages d'utiliser Drain 3 est sa rapidité d'exécution. La structure hiérarchique de Drain 3 permet un traitement efficace des logs et la hiérarchisation des tokens accélère le processus de correspondance des templates. De plus, grâce à l'utilisation de caches, Drain 3 peut accélérer le traitement des logs en mémorisant les templates déjà rencontrés et en réduisant le temps de calcul nécessaire pour les nouveaux messages de logs similaires. Drain 3 utilise des mécanismes de cache pour stocker les résultats intermédiaires et les templates fréquents. Cela permet de réduire le temps de traitement pour les logs récurrents, améliorant ainsi la rapidité globale du système.

```

1 connected to 10.0.0.1
2 connected to 192.168.0.1
3 Hex number 0xDEADBEAF
4 user yanni.mansour logged in
5 user elise.maistre logged in

```



```

1 Cluster 1 : Logs trouvés=2 : connected to <*:>
2 cluster 2 : Logs trouvés=1 : Hex number <*:>
3 Cluster 3 : Logs trouvés=2 : user <*:> logged in

```

Figure 25 : Groupement des logs en cluster et capture de l'information différente par Drain 3

b) Les regex : intervention sur l'algorithme

Afin d'améliorer les résultats fournis par Drain 3, il est possible de lui fournir des patterns nommés prédéfinis, intitulés des regex. Un regex ou une expression régulière décrit à l'aide de symboles précis un motif que nous voulons capturer et localiser dans un texte. Ce motif peut-être de toute forme avec des chiffres ou des signes particuliers. Cette méthode peut notamment être utilisée pour retrouver des dates.

En donnant des regex à drain 3 nous pouvons étiqueter les données capturées et ainsi obtenir une base de données plus précise.

```
[MASKING]
masking = [
    {"regex_pattern": "(?<^(\d{1-9}|\[12\]\d|3[\01])\d{1-9}|1\d{2})(\d{2})\\s([\01]\\d{2}\\d{3})[0-5]\\d{5}\\d\\s(\d)+", "mask_with": "pid"},
    {"regex_pattern": "^(0[1-9]|\[12\]\\d|3[\01])(0[1-9]|1\d{2})(\d{2})\\s([\01]\\d{2}\\d{3})[0-5]\\d{5}\\d", "mask_with": "date"},
    {"regex_pattern": "(INFO|WARN)", "mask_with": "level"},
    {"regex_pattern": "dfs\\.[a-zA-Z$]+", "mask_with": "composant"}
]
```

Figure 26 : Exemple de regex qui capturent de nouveaux éléments pour aider Drain 3

Après la tokenisation des éléments et durant la mise en place de l'arbre, Drain 3 recherche les éléments et vérifie leur forme pour les étiqueter ou non. L'étiquetage simplifie la décision sur la profondeur pour l'algorithme. Les éléments étiquetés sont ensuite disponibles pour chaque ligne.

La mise en évidence des tokens semblables par la profondeur formée par Drain 3 a notamment permis la perception de ces éléments et leur étiquetage par des regex.

1 \d{1,3}\.\.\d{1,3}\.\.\d{1,3}\.\.\d{1,3}	↔	1 192.168.0.1
---	---	------------------

1 - Cluster 1 : Logs trouvés=2 : connected to <*:*> 2 - cluster 2 : Logs trouvés=1 : Hex number <*:*> 3 - Cluster 3 : Logs trouvés=2 : user <*:*> logged in		1 - Cluster 1 : Logs trouvés=2 : connected to <:IP_ADDRESS:> 2 - cluster 2 : Logs trouvés=1 : Hex number <:HEX_NUM:> 3 - Cluster 3 : Logs trouvés=2 : user <:USERNAME:> logged in
---	---	---

Figure 27 : Illustration de l'utilisation de regex pour étiqueter les éléments capturés

c) L'adaptation et les modifications

Pour utiliser l'algorithme il est nécessaire de créer des scripts qui permettent l'entraînement des données avec une création de clusters puis l'utilisation de ces clusters sur les données à enregistrer.

Ces scripts permettent notamment d'enregistrer des fichiers permettant de retenir les clusters créés et leur forme. Ces clusters correspondent à un enchainement de clés détectées par l'algorithme.

On peut voir ci-dessous l'exemple des six clusters formés pour HAProxy.

```
Parsing > résultats > clusters > haproxy_clusters.csv > data
1 1,"<:date:> <:serveur:> <:processus:>[<:pid:>]: <:ip_port:> [<:date_detaillee:>] <:frontend:> <:backend:> <:temps:> <:temp
2 2,<:date:> <:serveur:> <:processus:>[<:pid:>]: <:ip_port:> [<:date_detaillee:>] <:frontend:> <:backend_serveur:> <:temp
3 3,<:date:> <:serveur:> <:processus:>[<:pid:>]: <:ip_port:> [<:date_detaillee:>] <:frontend:> <:backend:> SSL handshake failure
4 6,<:date:> <:serveur:> <:processus:>[<:pid:>]: <:ip_port:> [<:date_detaillee:>] <:frontend:> <:backend:> <:temps:> <:temp
5 4,<:date:> <:serveur:> <:processus:>[<:pid:>]: [<:level:>] (<:pid:>) : <:message_particulier:>
6 5,<:date:> <:serveur:> <:processus:>[<:pid:>]: <:message_particulier:>
```

Figure 28 : Exemple des 6 clusters créés par Drain 3 pour le type de log HAProxy

Une fois les clusters définitifs formés sur les données d'entraînements. Les données de tests sont ensuite transformées ligne par ligne via notre script à l'aide de l'algorithme et des clusters entraînés. Les lignes de données parsées sont enregistrées dans la base de données en parallèles.

Lors de l'utilisation de l'algorithme Drain 3 pour le parsing des logs, un bug aléatoire a été identifié et corrigé. Le problème se situait au niveau de l'utilisation des expressions régulières (regex) pour nommer les tokens trouvés. Le bug se manifestait de manière non déterminée en raison de l'ordre aléatoire dans lequel les regex étaient traitées. Cet ordre aléatoire était dû à l'utilisation d'un ensemble (set) pour stocker les regex, ce qui ne garantissait pas l'ordre précis.

Pour corriger ce bug, j'ai remplacé l'ensemble par une liste ordonnée, traitée en ordre inverse. Cette modification a permis de stabiliser le traitement des regex, éliminant ainsi le caractère aléatoire du bug. Identifier ce problème a été particulièrement complexe en raison de la nature aléatoire du bug et de la complexité de l'algorithme Drain 3, qui comprend de nombreux scripts et étapes de traitement.

De plus, la difficulté à reproduire systématiquement le bug compliquait l'identification de sa source. Après avoir mis en place la correction, les résultats ont montré que les logs étaient correctement traités, et nous avons pu poursuivre le projet en organisant le stockage des données dans notre base de donnée.

3) Stockage dans la BD

Lors de la conception de notre projet, nous avons dû déterminer le type de base de données le mieux adapté pour enregistrer nos données. La majorité des éléments à enregistrer consistent en des logs soit des lignes d'informations semi-structurées. Ces logs sont prévus pour être lus par un humain ou par une machine de par sa structure et son organisation. Les informations présentes dans ces lignes conservent un schéma directeur de formes se succédant. Les séparateurs de champs correspondent à des espaces qui peuvent se perdre au sein d'un même attribut mais leur forme reste organisée.

Ce format particulier a soulevé la question du choix entre une base de données SQL et une base de données NoSQL. Étant donné la nature semi-structurée des données, les bases de données NoSQL pouvaient être envisagées. Cependant, nous avons décidé de commencer avec une base de données SQL et de garder l'option d'élargir dans le futur vers une base de données NoSQL si nécessaire.

Le choix d'utiliser une base de données SQL a été fait conjointement avec l'idée de parser les données via un algorithme spécialisé. En effet, une fois les logs parsés, les paires clé-valeur peuvent être extraites et enregistrées facilement dans une base de données structurée.

Pour choisir le type de base de données le plus adapté, nous avons estimé la quantité de logs générés au sein du lab'SOLUTEC. GitLab et SonarQube, par exemple, produisent environ 60 % du total des logs, avec une dizaine de logs par seconde. En comparaison, les autres services génèrent environ une dizaine de logs par minute.

Au total, le nombre de logs générés quotidiennement est d'environ 400 000 logs, occupant environ 700 Mo de stockage par jour. En extrapolant sur une année, cela représente environ 400 Go de logs, une taille raisonnable pour une base de données SQL, capable de supporter plusieurs téraoctets de données.

En analysant le format des logs reçus et le résultat du log parsing, qui vise à uniformiser les logs, il est apparu judicieux de se tourner vers une base de données SQL. Les données étant destinées à être formatées de manière cohérente, une base de données relationnelle est bien adaptée pour maintenir cette structure. Nous avons donc choisi de nous tourner vers une base de données PostgreSQL. Ce type de base de données présentant l'avantage d'être extensible et polyvalente.

Lors de la création de la base de données plusieurs réflexions ont été prises en compte et celle-ci a beaucoup évolué. Lors de sa mise en place deux points principaux de l'application ont été considérés : comment stocker les logs et comment stocker les informations liées à l'application : les utilisateurs et les sauvegardes.

Voici notre schéma de base de données représentée sous forme d'un UML pour mieux comprendre les interactions qui existent entre les informations et saisir la manière dont celles-ci sont stockées.

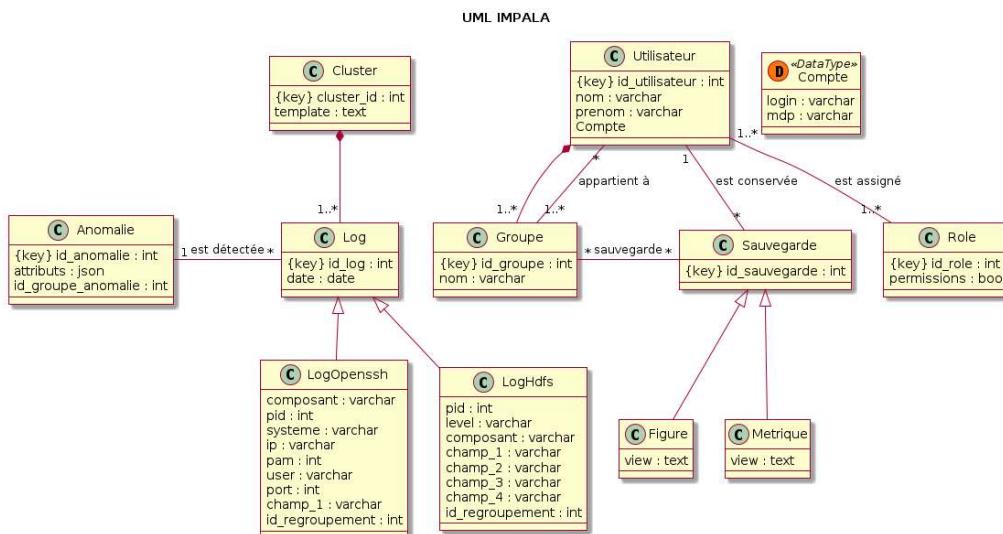


Figure 29 : UML du projet IMPALA, avec le stockage des logs, des anomalies et la gestion des éléments et utilisateurs créés sur le site

4) Les logs

Les logs présentent des schémas d'architecture variés selon leur type, chaque type ayant des champs spécifiques. Par conséquent, chaque type de log doit être enregistré dans une table distincte, comme on peut voir un extrait des différentes tables créées ci-dessus. Chaque type de log est traité comme une sous-classe de la classe principale Log, qui contient les informations

communes. Par exemple, la date est un paramètre clé commun à tous les logs et est donc renseignée de manière globale.

Les logs doivent être associés à un cluster, stocké dans une table dédiée aux différents formats de clusters générés par l'algorithme Drain 3 lors du parsing. Les anomalies, qui correspondent aux alertes détectées par les algorithmes d'analyse, sont liées aux logs et peuvent faire partie d'un groupe de logs. Ces alertes détectées seront vues plus en détail dans la partie suivante.

L'enregistrement des logs représente une part majeure du stockage de la base de données. Lors de l'ajout d'un nouveau type de log, une nouvelle table est créée pour contenir les champs spécifiques à ce type. Cependant d'autres informations doivent également être enregistrées.

5) Les sauvegardes et utilisateurs

Pour assurer le bon fonctionnement du site web, il est essentiel de sauvegarder les informations qui permettent son activité. Cela inclut l'enregistrement des utilisateurs, ainsi que leurs rôles et groupes, qui déterminent leur statut et leurs accès au sein de l'interface.

Les autres sauvegardes effectuées concernent les métriques et graphiques qui sont générés par les utilisateurs. Ces éléments sont créés via notre interface Web et peuvent être enregistrés ou supprimés directement par cet intermédiaire.

Une fois les logs parsés et leurs informations annexes enregistrées et stockées dans une base de données, les logs peuvent être analysés et traités pour apprendre de ces données.

TROISIÈME SOUS-PARTIE : LOG MINING

Après avoir récupéré, parsé et stocké les logs ceux-ci peuvent être affichés sur le site web et explorés. Cependant l'exploration peut être accompagnée par un modèle analysant en amont les logs reçus. Cette analyse peut être faite directement par des algorithmes de Machine learning et permettrait un gain de temps considérable dans l'analyse de ces logs. Pour améliorer les recherches lors de l'analyse des logs le Machine learning peut accompagner les humains dans leurs études pour retrouver les erreurs ou problématiques rencontrées.

L'objectif de l'analyse de logs repose sur trois grands points :

- La détection d'anomalies : L'objectif étant d'identifier les schémas et les comportements inhabituels, de vérifier les performances des applications et serveurs, de repérer les pannes et aussi de veiller à la sécurité en observant les comportements des utilisateurs.
- La prédiction des incidents : L'objectif étant de prédire les comportements et données après l'obtention de certains résultats dans les logs. Cette prédiction est faite sur les incidents ou les pannes qui peuvent survenir.

- Le diagnostic des échecs : L'objectif étant de diagnostiquer les raisons des problèmes survenus une fois l'arrivée d'une problématique en remontant jusqu'à la source de la difficulté.

Lors de ce stage, l'analyse s'est portée davantage sur la détection des anomalies car la détection semble être au cœur des problématiques lors de l'analyse des logs.

Nos données n'étant pas labellisées nous nous sommes concentrés sur des algorithmes de Machine learning non supervisés pour détecter les anomalies issues des logs.

Plusieurs méthodes ont pu être employées selon les logs et les résultats. La première consiste à considérer toutes les données numériques et former des clusters qui s'éloigneraient des autres valeurs pour détecter les anomalies. La deuxième méthode consiste à considérer les anomalies en observant les séries temporelles.

La première étape consistait tout d'abord à nettoyer les données pour permettre l'application des divers algorithmes.

1) Nettoyage des données

Le nettoyage des données a été effectué en plusieurs étapes. Tout d'abord les données ont été extraites de la base de données. Certaines données n'avaient pas été séparées correctement lors du parsing lié à l'absence de séparateurs et il a ainsi fallu les dissocier. Les données ont ensuite été converties selon leur format.

Après diverses analyses statistiques et visualisations sur les données, elles ont pu être traitées. Les données les plus importantes ont été conservées. Les données identiques ont été supprimées.

La détection des anomalies s'est différenciée selon les types de logs choisis pour cette étape. Le choix s'est porté sur les services qui semblaient les plus essentiels au fonctionnement du Lab avec des logs prédisposés à la détection.

2) Choix des types de logs à traiter

Nous avons étudié majoritairement deux types de logs lors de ces analyses : HAProxy et GitLab. Le choix a été fait au vu de l'utilité de cette détection.

En effet, HAProxy (High Availability Proxy) est un logiciel très utilisé au Lab comme équilibrEUR de charge des couches TCP et HTTP, en effectuant des redirections sur les requêtes. Cet outil prend en charge les communications entre les applications, les appareils et les navigateurs web lors du transfert de fichiers (textes, images, vidéos). HAProxy permet également d'obtenir la chaîne de confiance via des certificats pour les services du Lab. Les logs HAProxy renseignent ainsi sur toutes les communications qui ont lieu entre les différents services lors de l'utilisation et permet un aperçu global des défaillances qui peuvent survenir lors de l'activité du

Lab. Obtenir ce type de logs et détecter des anomalies sur son activité est essentiel au fonctionnement des services informatiques du Lab ou d'une DSi.

Au Lab, GitLab n'est pas pris en charge par HAProxy et fonctionne sur une machine virtuelle particulière. GitLab est l'un des logiciels les plus utilisés au Lab. Tous les projets y sont créés, testés et déployés pour permettre une collaboration tout le long de l'étape de développement. Ce logiciel produit de très nombreux logs avec des lignes contenant d'importantes informations sur les temps d'exécutions et les processus en cours. C'est le service qui produit le plus de logs avec plusieurs centaines de logs générés par minute sur certains fichiers. Ainsi le traitement automatisé de ceux-ci semble être prioritaire au vu de sa taille et de son importance au sein des projets.

Le choix de ces types de logs a été également fait au vu de leur syntaxe plutôt structurée et de leur ancienneté. En effet, Sonarqube est également très présent au Lab et peut-être utile pour détecter des anomalies mais les logs sont moins structurés, avec de nombreuses phrases en langage naturel. La détection d'anomalie sur des textes complexifie les recherches car les mêmes champs ne se retrouvent pas sur chaque ligne. De plus, Sonarqube est un logiciel plutôt récent, la documentation est donc moins clair et explicite que pour HAProxy ou encore GitLab qui sont bien établis. De plus la structure des logs évolue encore rapidement avec des nouvelles versions rapidement apparues. Les logs Sonarqube sont donc voués à évoluer rapidement, ainsi le parsing et la détection sont moins prioritaires.

Les algorithmes mis en place sur ces deux types de logs pourront être réemployés et réutilisés pour d'autres types de logs car les structures entre les logs restent similaires.

Ces deux types de logs ont été collectés sur les VM, puis parsés via Drain3 avec l'aide de regex et finalement stockés sur la base de données.

Pour détecter les anomalies, deux méthodes non supervisées ont principalement été employées, notamment au vu des types de logs sélectionnés. Tout d'abord, la première méthode fut de tenter de séparer les anomalies par le biais de clusters en utilisant des algorithmes de classifications.

3) Clusterisation et PCA

Les méthodes de clusterisation, ou encore de partitionnement des données correspondent aux différentes méthodes de classifications non supervisées utilisant des algorithmes d'apprentissage pour rassembler les données similaires par groupes appelés classes ou clusters. Ces classes présentent des propriétés proches entre elles et divergent avec les éléments des autres classes. La proximité des éléments est calculée par le biais d'une distance sur les attributs.

Afin d'obtenir des clusters des différents logs les plus proches, il a fallu tout d'abord sélectionner les attributs qui pourraient être utilisés. Les données numériques comme les tailles, les temps ou d'autres informations quantitatives ont pu être conservées. Pour les données qualitatives, certaines données ont pu être gardées en réalisant une vectorisation sur les variables catégorielles. Cette méthode permet de convertir les données catégorielles en données numériques

binaires, où chaque catégorie est représentée par une variable binaire distincte. Ainsi des informations comme les PID, les services ou encore les codes ont pu être conservés de manière binaire car leur nombre de valeurs distinctes n'étaient pas trop élevés.

Les données conservées doivent être normalisées par la suite pour ne pas donner trop de poids à un attribut ou une valeur lors du calcul de la distance. La normalisation permet de transformer la majorité des données en des valeurs entre -1 et 1 en fonction des autres valeurs sur la colonne.

Deux méthodes de clusterisation ont été employées sur ces données. K-means et CAH.

a) K-means

K-means est un algorithme de clustering non supervisé utilisé pour regrouper les données en k groupes distincts en fonction de leur similarité. Cet algorithme fonctionne en sélectionnant initialement k points centraux, appelés centroïdes, et en attribuant chaque point de données au centroïde le plus proche. Les centroïdes sont ensuite mis à jour en fonction des moyennes des points de données qui leur sont attribués, et le processus est répété jusqu'à ce que les centroïdes ne changent plus ou qu'un nombre maximal d'itérations soit atteint.

L'utilisation de l'Analyse en Composantes Principales (ACP) en combinaison avec k-means peut améliorer ses performances. En effet, k-means est sensible à la présence de variables corrélées et à la dimensionnalité des données. Lorsque les données ont une forte dimensionnalité, l'espace de recherche devient très grand, ce qui rend la tâche de clustering plus difficile et plus longue. L'ACP est une technique de réduction de la dimensionnalité qui permet de transformer un ensemble de variables corrélées en un ensemble de variables non corrélées, appelées composantes principales. Ces composantes principales sont ordonnées en fonction de leur variance, ce qui permet de conserver les informations les plus importantes dans les premières composantes. En réduisant la dimensionnalité des données, l'ACP permet de réduire le bruit et les effets négatifs de la dimensionnalité, améliorant ainsi les performances de k-means. Ainsi, lorsque l'ACP est utilisée en prétraitement avant l'application de k-means, les données sont projetées sur un espace de dimension inférieure, ce qui permet de réduire le temps de calcul et d'améliorer la qualité des clusters obtenus. En effet, l'ACP permet de réduire les effets de la corrélation entre les variables, ce qui peut conduire à une meilleure séparation des clusters.

Le choix du nombre de clusters a été déterminé à partir des méthodes du coude et de la silhouette qui déterminent le moment optimal selon les résultats. Pour GitLab nous obtenons des résultats intéressants pour détecter des clusters qui divergent beaucoup de la moyenne.

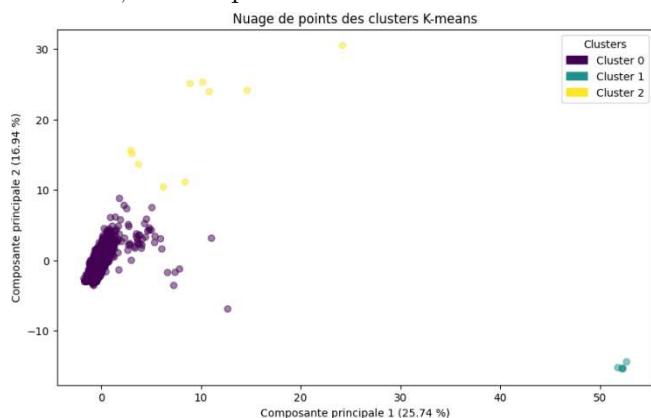


Figure 30 : K-means selon les deux composantes principales pour les logs GitLab

Comme nous l'observons dans la répartition des clusters, ceux-ci sont très déséquilibrés, avec la grande majorité des logs dans le cluster 0. On peut dans un premier temps considérer les logs présents dans les clusters les plus petits comme des "anomalies", dans le sens où ils sont différents de ceux présents dans le cluster principal. La présence d'"anomalies" se confirme en observant les logs du cluster 1, qui possèdent des différences visibles par rapport aux logs des autres clusters lorsque l'on compare leur contenu.

Pour HAProxy le clustering n'est pas idéal car les données sont assez proches selon leurs attributs. Cependant les classes se démarquent, ainsi il pourrait être intéressant d'explorer d'autres méthodes de détection d'anomalies.

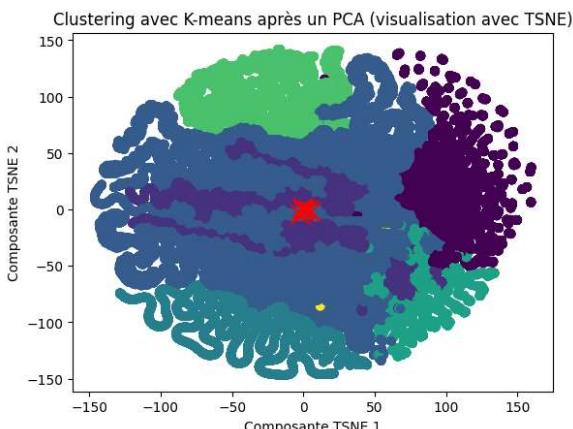


Figure 31 : K-means selon TSNE pour les logs HAProxy

b) CAH

La Classification Ascendante Hiérarchique (CAH) est une méthode de clustering utilisée pour regrouper des données similaires en clusters hiérarchiques. La CAH commence par traiter chaque point de données comme un cluster individuel, puis elle fusionne les clusters les plus similaires en un seul cluster à chaque étape, jusqu'à ce qu'il ne reste qu'un seul cluster contenant toutes les données.

La similarité entre les clusters est mesurée à l'aide d'une métrique de distance, telle que la distance euclidienne, et il existe différentes méthodes de liaison pour déterminer la distance entre les clusters, comme la liaison moyenne qui calcule la distance moyenne entre les points des deux clusters.

Cette méthode a davantage été employée pour les logs HAProxy. Cette clusterisation n'a pas été faite avec les attributs des données, trop peu représentatif, mais avec la quantité de logs obtenus par minutes sur une journée. Pour chaque minute on peut observer une courbe qui représente la fréquence des logs obtenus pendant la journée. A partir de cette courbe une distance peut être calculée sur chaque minute de la journée. Cette méthode permet de séparer les jours en différents

jours types.

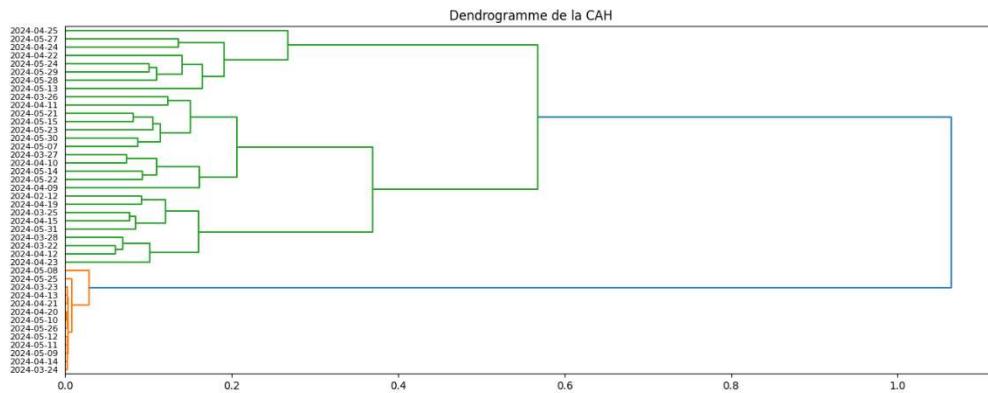


Figure 32 : Dendrogrammes montrant la proximité des jours selon la quantité de logs reçus par minute (HAProxy)

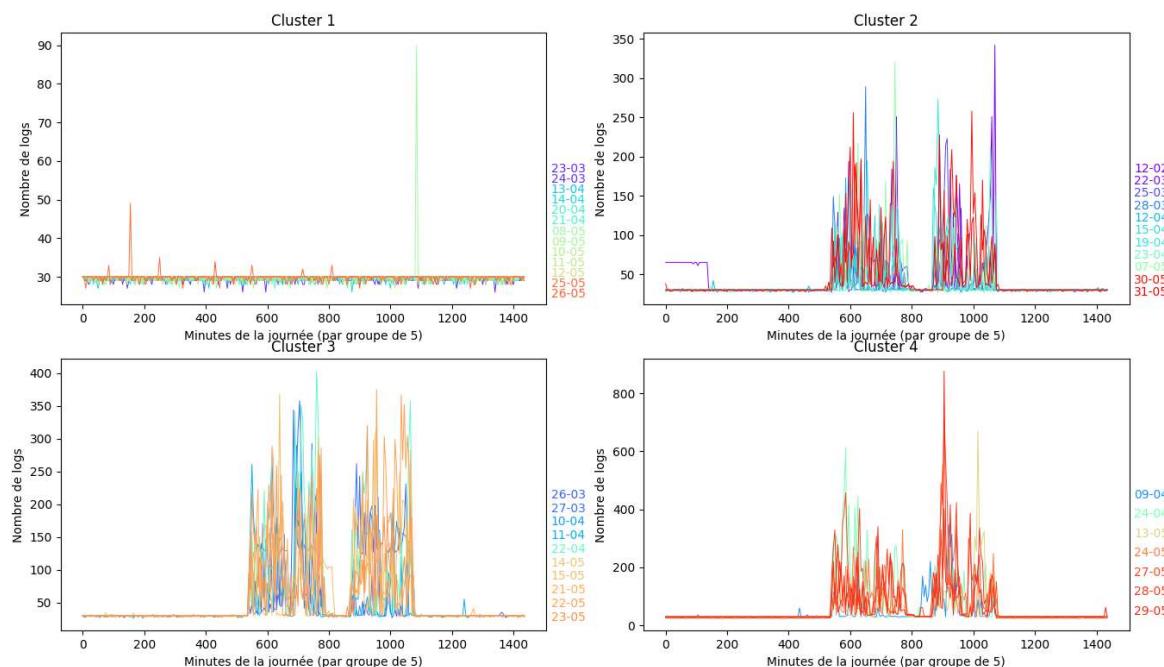
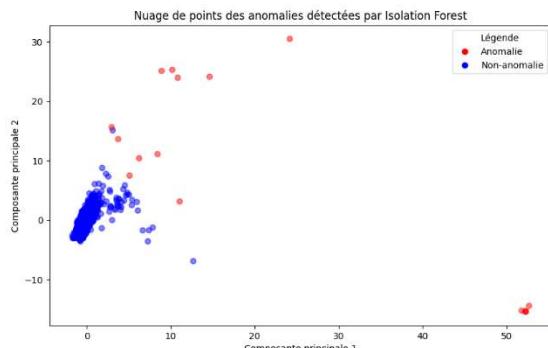


Figure 33 : Les différents clusters créés à partir de la distance représentée sur le dendrogramme précédent, on obtient 4 jours types (logs HAProxy)

On peut observer l'activité du Lab à partir des logs HAProxy. On observe notamment les week-end et jours fériés en haut à gauche. Certains jours ont davantage d'activités que d'autres et sont ainsi clusterisés. En réalisant cette clusterisation on peut utiliser ces jours types pour voir quels jours s'éloigneraient de la moyenne habituelle en utilisant notamment des algorithmes sur les séries temporelles que nous verrons ci-après.

Une autre méthode peut-être également employée sur les données pour améliorer la détection des anomalies, comme nous avons testé avec isolation forest.

Figure 34 : Anomalies détectées par Isolation Forest (GitLab)



c) Isolation Forest

Isolation Forest est basé sur l'idée d'isoler les anomalies plutôt que de modéliser les données normales. L'algorithme construit un ensemble d'arbres de décision binaires aléatoires, appelés arbres d'isolation, qui partitionnent l'espace de données en régions plus petites. Les anomalies sont plus susceptibles d'être isolées dans des régions plus petites et plus profondes de l'arbre.

Lors de la classification, Isolation Forest calcule un score d'anomalie pour chaque échantillon en fonction de la profondeur moyenne des feuilles dans lesquelles l'échantillon se trouve dans les arbres d'isolation. Les échantillons avec des scores d'anomalie élevés sont considérés comme des anomalies. L'algorithme est très efficace pour détecter les anomalies dans des jeux de données de grande taille avec un nombre élevé de dimensions, car il ne nécessite pas de calculer des distances ou des densités entre les échantillons.

Cet algorithme a davantage fonctionné sur les logs GitLab que les logs HAProxy qui ont obtenu des résultats peu concluants. Les logs GitLab ont réellement détecté des anomalies qui se distinguaient des autres points selon les différents attributs numériques qui ont été utilisés.

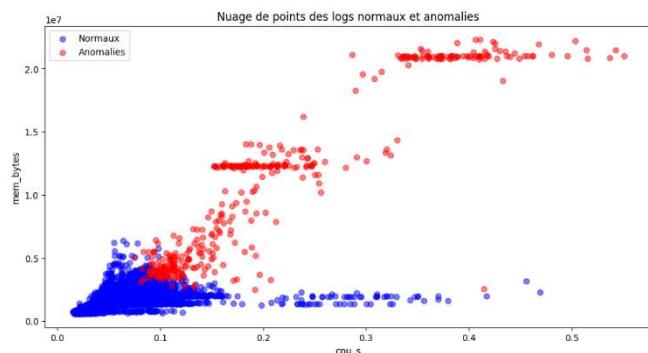


Figure 35 : Anomalies détectés avec Isolation Forest (GitLab)

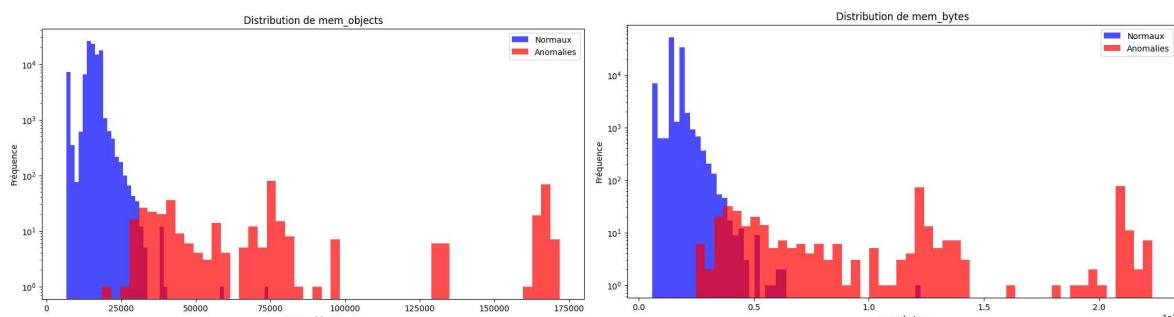


Figure 36 : Distributions sur certains aspects pour observer si les anomalies se distinguent sur certains aspects (ici la mémoire des logs GitLab)

On peut voir que les données détectées comme des anomalies ont bien des caractéristiques différentes, en observant sur plusieurs attributs différents les anomalies se distinguent en effet du reste. Le nombre d'anomalie détectées reste faible au vu du nombre de logs. Ces anomalies sont enregistrées dans la base de données avec d'autres paramètres comme la moyenne et la variance. Ces informations seront affichées sur l'onglet alerte de notre site web.

Comme on peut le voir ci-contre sur le temps d'exécution d'HAProxy, les anomalies ne s'éloignent pas des autres données et restent très similaire il est donc difficile de les différencier.

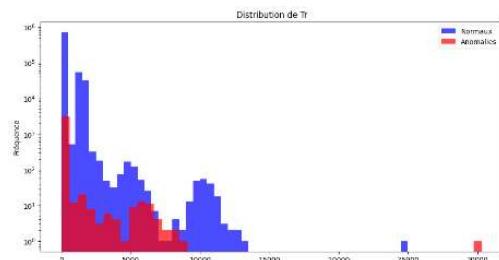


Figure 37 : Distribution du temps pour les logs HAProxy (cas de non distinction)

Les tentatives réalisées sur HAProxy n'ont pas abouti c'est pourquoi d'autres tentatives sur les séries temporelles ont été réalisées.

4) Séries temporelles

- a) Isolation Forest

L'utilisation d'Isolation Forest pour la détection d'anomalies dans les séries temporelles est une approche efficace pour identifier les points aberrants dans les données. Contrairement aux méthodes de clustering, Isolation Forest ne nécessite pas de définir un nombre prédéfini de clusters ou de définir une mesure de distance entre les points de données. Au lieu de cela, l'algorithme isole les points aberrants en construisant un ensemble d'arbres de décision binaires aléatoires.

Dans notre implémentation, nous avons diviser les données de la série temporelle en segments de taille égale. Pour chaque segment, nous avons appliqué Isolation Forest pour détecter les points aberrants. Nous avons utilisé le paramètre "contamination" pour spécifier le pourcentage de points aberrants prévus dans les données. Les points identifiés comme aberrants ont été marqués et visualisés dans le graphique de la série temporelle.

Les segments utilisés sont une journée de données divisée en minutes. A partir de la moyenne sur les minutes ou bien de la somme, pour les logs HAProxy.

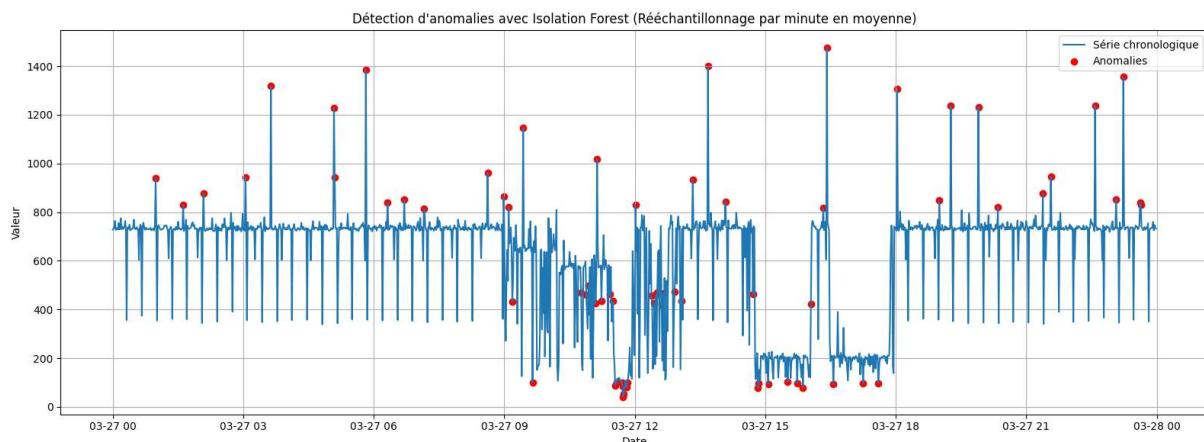


Figure 38 : Détection d'anomalies du 27/03 des logs HAProxy avec Isolation Forest sur des séries temporelles

On peut observer que l'algorithme détecte bien certaines anomalies qui semblent se distinguer des autres valeurs.

Une autre idée pour améliorer les performances a été d'utiliser la clusterisation réalisée avec CAH et de faire la détection en fonction d'un modèle pré-entraîné sur les données moyennes des clusters formés puis d'obtenir les anomalies en fonction de cette courbe de référence selon le cluster de la courbe observée.

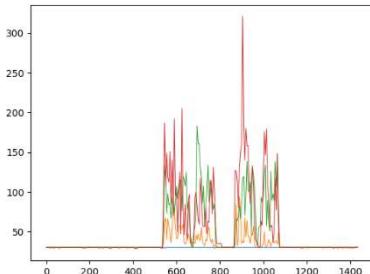


Figure 39 : Jours-types moyens sur une journée(HAProxy)

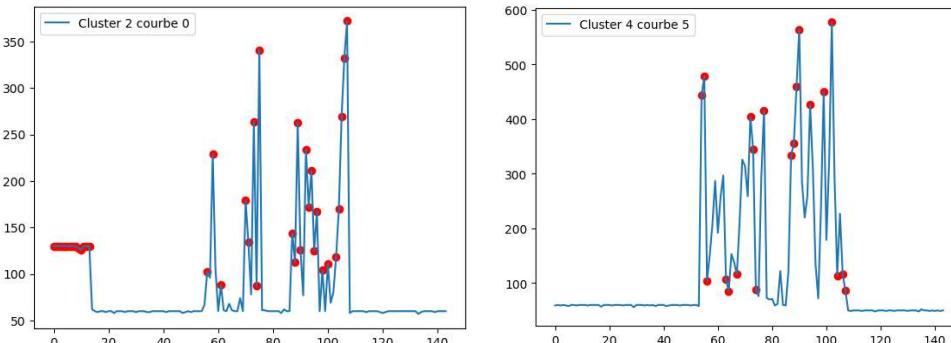


Figure 40 : Anomalies détectées en utilisant les moyennes des jours-types créés

L'algorithme a ainsi pu détecter des anomalies sur les données selon les différents clusters. La détection sur les clusters des week-end et jours fériés n'étaient pas les plus performants, c'est pourquoi l'idée a été d'utiliser un autre algorithme pour améliorer la détection de ces clusters.

b) Moyennes mobiles

Les moyennes mobiles sont un outil couramment utilisé pour la détection d'anomalies dans les séries temporelles. Elles permettent de lisser les données en supprimant les fluctuations aléatoires et en mettant en évidence les tendances sous-jacentes.

La moyenne mobile simple est calculée en faisant la moyenne des n dernières valeurs de la série temporelle. Elle donne donc le même poids à chaque valeur dans la fenêtre de temps considérée. La longueur de la fenêtre de temps est un paramètre important à choisir, car elle détermine la sensibilité de la moyenne mobile aux fluctuations de la série temporelle. Plus la fenêtre est longue, plus la moyenne mobile sera lissée, mais moins elle sera sensible aux changements récents.

Pour détecter les anomalies à l'aide des moyennes mobiles, on peut utiliser une approche basée sur des seuils. On calcule d'abord la moyenne mobile de la série temporelle, puis on définit un seuil supérieur et un seuil inférieur autour de cette moyenne mobile. Si une valeur de la série temporelle dépasse le seuil supérieur ou tombe en dessous du seuil inférieur, elle est considérée comme une anomalie. Les seuils peuvent être fixes ou adaptatifs, c'est-à-dire qu'ils peuvent être ajustés en fonction de la volatilité de la série temporelle.

La détection d'anomalie par les moyennes mobiles a également été réalisée sur des journées sous forme de courbe de somme ou moyenne du nombre de logs par minutes à propos des logs HAProxy.

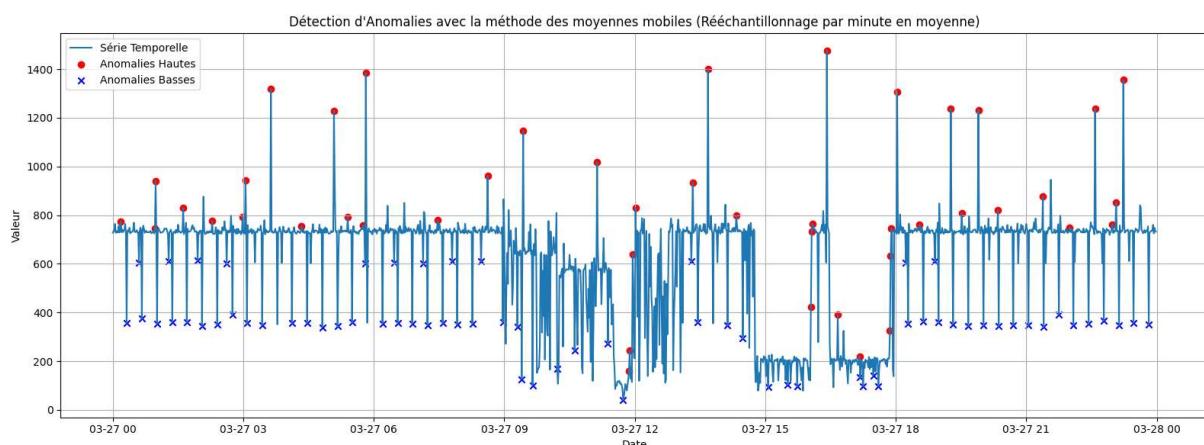


Figure 41 : Détection d'anomalies du 27/03 des logs HAProxy avec les moyennes mobiles sur des séries temporelles

Les résultats sont divisés en deux types d'anomalies et semblent plus précis que les résultats obtenus avec Isolation Forest sur ce type de données. De plus, les clusters des weekends et jours fériés semblent être mieux détectés, comme on peut le voir ci-contre. Les résultats obtenus détectent de très nombreuses anomalies et les résultats pourraient être recoupés avec d'autres algorithmes sur les séries temporelles comme ARIMA ou encore Prophet que j'ai eu l'occasion d'essayer.

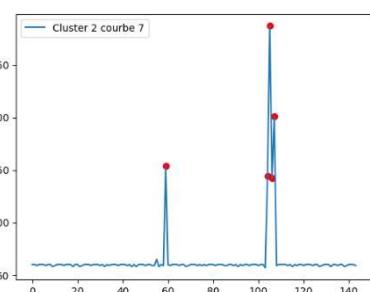


Figure 42 : Anomalies détectées sur le jour type « feriens et week-end »

c) ARIMA

ARIMA (AutoRegressive Integrated Moving Average) est une méthode statistique utilisée pour modéliser et prévoir des séries temporelles. Elle combine trois composantes principales : l'autoregression (AR), la moyenne mobile (MA) et l'intégration (I).

Dans le contexte de la détection d'anomalies, ARIMA peut être utilisé pour modéliser le comportement normal d'une série temporelle et identifier les points qui s'écartent significativement de ce comportement. Pour ce faire, on commence par ajuster un modèle ARIMA sur les données historiques de la série temporelle. Ensuite, on utilise le modèle pour prévoir les valeurs futures de la série et calculer les résidus (c'est-à-dire la différence entre les valeurs observées et les valeurs prédites). Les résidus sont alors analysés pour identifier les points qui s'écartent significativement de la distribution normale.

L'un des avantages d'ARIMA pour la détection d'anomalies est qu'il prend en compte les tendances, les saisons et les fluctuations aléatoires de la série temporelle. Cela signifie qu'il peut détecter des anomalies qui ne seraient pas visibles en utilisant des méthodes plus simples, telles que la moyenne mobile ou la décomposition en séries temporelles.

On peut voir ci-dessous la décomposition selon la saisonnalité et la tendance des logs HAProxy sur une heure.

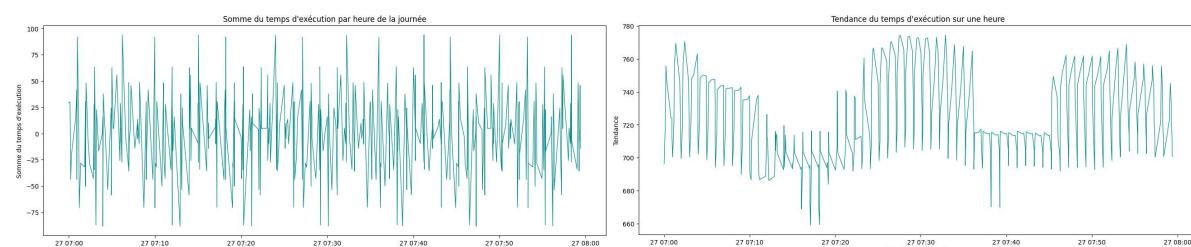


Figure 44 : Saisonnalité et tendance des logs HAProxy entre 7h et 8h

En plus de la détection d'anomalies, ARIMA peut également être utilisé pour prévoir les valeurs futures d'une série temporelle. Cela peut être utile pour détecter les anomalies avant qu'elles ne se produisent ou pour prévoir les tendances futures de la série. ARIMA pourra donc être utilisé pour prédire les incidents sur les logs.

ARIMA a été réalisé sur un attribut spécifique, les temps d'exécution d'HAProxy

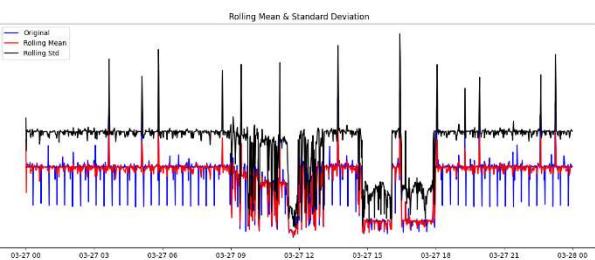


Figure 43 : Courbes sur une journée des moyennes et écart-types mobiles

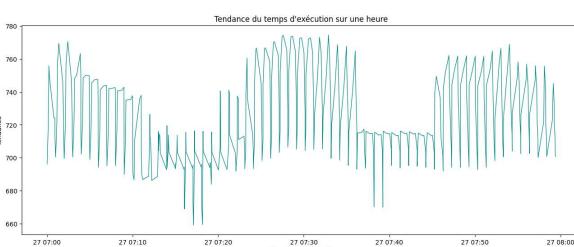


Figure 44 : Saisonnalité et tendance des logs HAProxy entre 7h et 8h

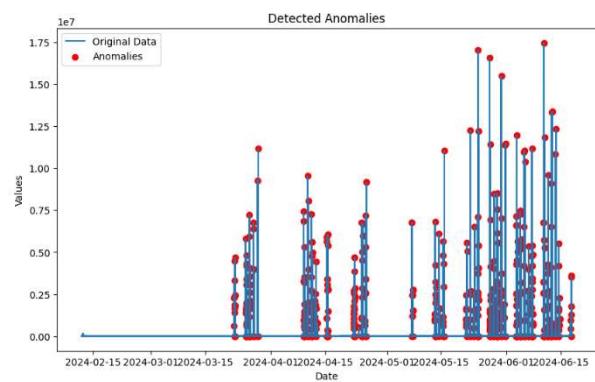


Figure 45 : Détection des anomalies par ARIMA sur une journée

qui semble être l'un des éléments les plus importants présents dans les informations extraites des logs.

Les résultats obtenus semblent être plutôt intéressants et moins nombreux que ceux obtenus par les algorithmes précédents.

d) Prophet

Prophet est une méthode alternative à ARIMA qui fonctionne de manière similaire mais possède certaines particularités qu'il est intéressant d'observer.

Prophet est une bibliothèque open source développée par Facebook pour la prévision de séries temporelles. Elle est basée sur un modèle additif qui prend en compte les tendances, les saisons et les effets des jours fériés. La détection d'anomalies via Prophet consiste à entraîner un modèle sur les données historiques d'une série temporelle, puis à utiliser ce modèle pour prévoir les valeurs futures de la série. Les écarts entre les valeurs prédites et les valeurs réelles sont ensuite analysés pour détecter les anomalies.

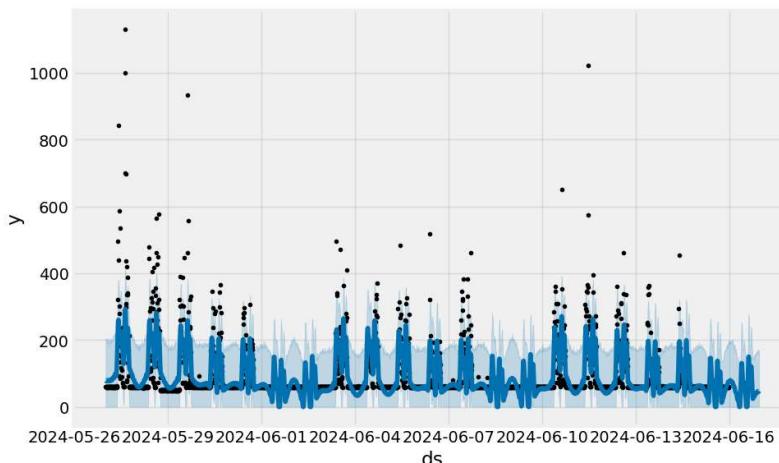
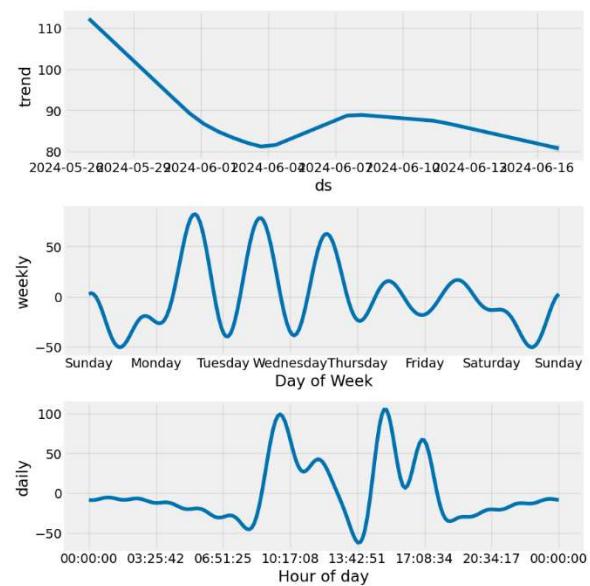


Figure 46 : Approximation et détection d'anomalies par Prophet de fin mai à mi-juin (HAProxy)

En observant les résultats obtenus sur trois semaines de données étudiées (fin mai à début juin), des anomalies sont détectées. A partir de tendances sur les jours de la semaine et des heures de la journée certains logs diffèrent des autres et ont pu être observés. On observe que les courbes produites sur lesquelles le modèle s'est entraîné approximent correctement les données et permettent une bonne estimation des anomalies reçues.

Figure 47 : Courbes utilisées par Prophet pour approximer les logs selon leur tendance, une semaine et une journée

Pour détecter les anomalies, Prophet utilise une approche basée sur les résidus. Les résidus sont calculés comme la différence entre les valeurs réelles et les valeurs prédites par le modèle. Les résidus sont ensuite analysés pour détecter les valeurs aberrantes qui s'écartent significativement de la distribution attendue de ceux-ci.



e) Autre tentative et améliorations futures

D'autres tentatives ont été réalisées au cours du stage mais n'ont pas pu être finalisées. Elles consistent en des améliorations possibles pour la suite du projet et permettraient potentiellement une meilleure approche de détection des anomalies sur des données de logs.

L'une des idées consiste, tout d'abord, en la création de deux clusters de données avec la classification hiérarchique ascendante. Ces clusters sépareraient les jours de repos des jours de travail car les activités en ces deux clusters diffèrent fortement. A partir de ces données l'idée serait de calculer une distance sur chacune des minutes de la journée en fonction des données des logs reçues cette journée par rapport à la moyenne sur la minute selon le cluster. On obtiendrait ainsi des courbes de journées sur chacune des minutes de la journée avec un chiffre calculé à partir d'une distance comme la distance euclidienne pour les données numériques et une distance particulière pour les données catégorielles.

Finalement, en obtenant une courbe de données sur chaque minute pour chaque jour il serait possible d'appliquer ARIMA et Prophet sur les données et détecter des anomalies en prenant en compte davantage d'informations.

Cette méthode pourrait fournir des résultats intéressants et exploitables et serait à observer dans le futur.

Les données collectées et parsées ainsi que certaines des anomalies détectées ont pu être affichées et rendues disponible sur le site Web qui a été conçu durant le stage.

QUATRIEME SOUS-PARTIE : SITE WEB

Un fois les données obtenues celles-ci peuvent être visualisées par l'intermédiaire d'un site ou d'une application. Ce site web a été réalisé à partir de maquettes conçues par notre product owner qui a donné les grands axes de configuration au niveau du design du site. Nous avons ensuite eu l'occasion de réaliser les trois onglets principaux de l'application. Lors du développement à la fois le frontend et le backend ont été réalisés. Le travail a été effectué page par page durant les sprints, nous avons contribué à chacune des pages par fonctionnalités.

1) Maquettes

Afin d'avoir une vision globale de l'apparence du site il est essentiel de réaliser une maquette en amont du développement. Cette maquette a été réalisée sous figma par notre product owner. Certaines fonctionnalités ont davantage été mises en avant mais le

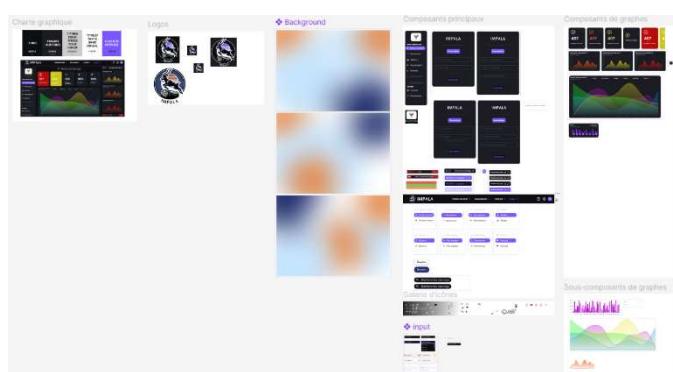


Figure 48 : Composants, boutons, logo en maquette sous Figma

style graphique et colorimétrique a été conservé. Le logo a également été réalisé lors de cette phase.

Différents composants, boutons et éléments visuels nous ont été proposés ainsi qu'une charte graphique. Ces éléments ont pu être repris et adaptés lors du développement du frontend.

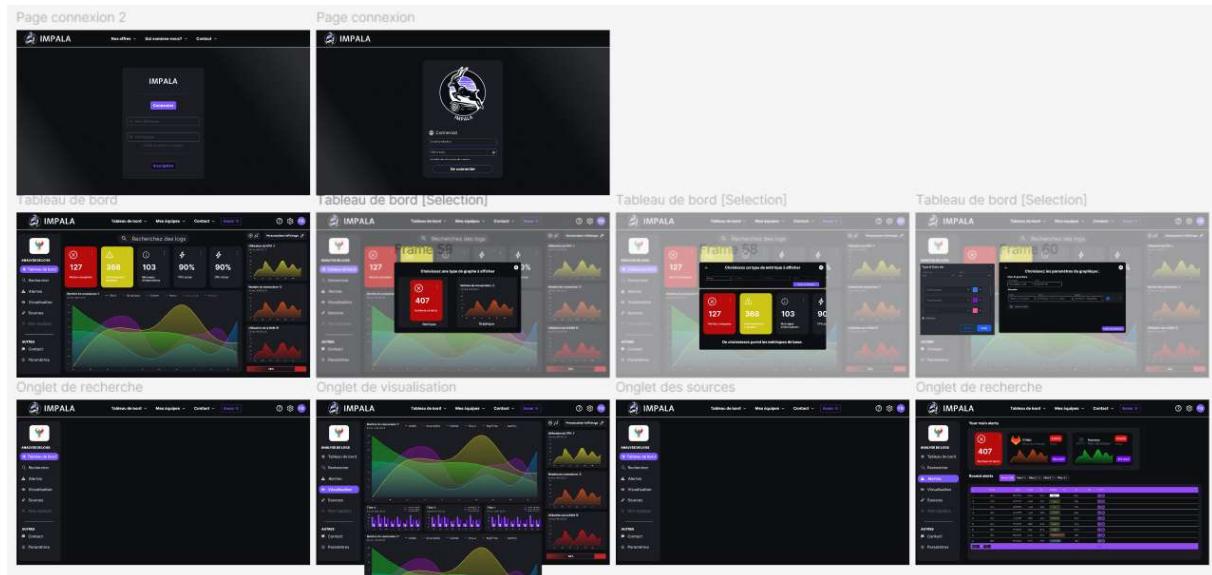


Figure 49 : Maquette Figma des pages du futur Site Web

On peut voir ci-dessus les différentes pages et sections qui ont pu être réalisées. L'application possède un header et la navigation est rendue possible par un menu latéral à gauche. Il est possible de naviguer grâce à ce menu entre les trois pages principales réalisées au cours du stage.

L'accent a été davantage mis sur la page principale de l'application, le tableau de bord qui fut la première étape du développement.

2) Tableau de bord

Le tableau de bord (ou dashboard) est l'écran principal de l'application. L'utilisateur peut visualiser les métriques définies qui correspondent à des indicateurs de performances et de logs issues de la base de données. Ces métriques

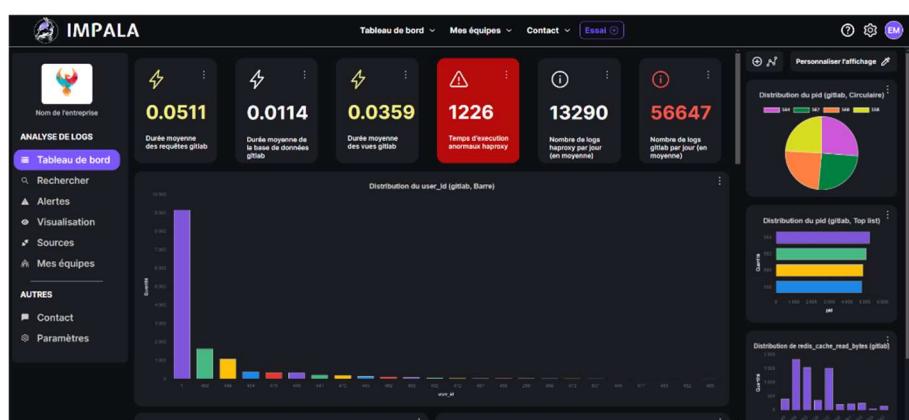


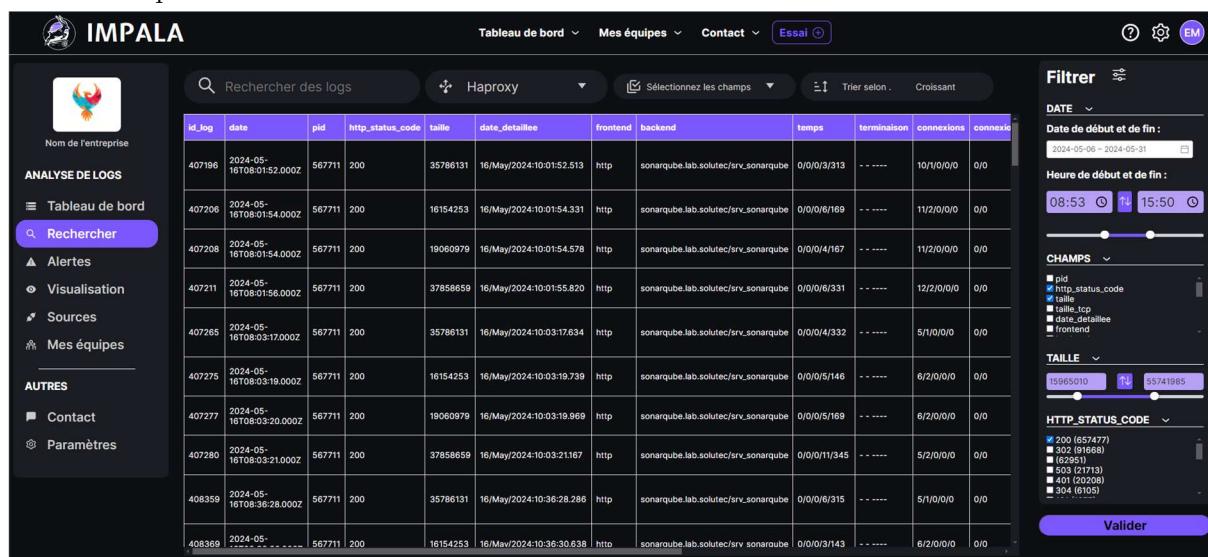
Figure 50 : Tableau de bord (page principale de l'application Web)

sont présentes sur la page sous forme de vignettes. De plus, les différents graphes créés par l'utilisateur sur les données des logs présents dans la base sont également visualisables. Les graphes peuvent être ajoutés et créés directement par l'utilisateur. Il peut ainsi personnaliser ces graphes en sélectionnant le type souhaité, les logs, les caractéristiques et les couleurs des éléments. Le titre et l'emplacement sur le panneau d'affichage sont également ajustables.

Les données utilisées sont celles qui ont été obtenues et parsées dans les étapes précédentes. Pour visualiser les logs plus précisément, l'onglet rechercher permet une recherche plus avancée sur ceux-ci.

3) Rechercher

Dans cet onglet, la recherche s'effectue sur les logs de la base de données visualisés dans un tableau. Ce tableau contient toutes les informations obtenues en colonnes sur les lignes de logs. Il est possible de rechercher sur le contenu, de sélectionner le type de logs souhaités, de définir les champs à conserver et également d'effectuer un tri croissant ou décroissant sur la caractéristique souhaitée.



The screenshot shows the IMPALA dashboard interface. On the left, there's a sidebar with navigation links like 'Tableau de bord', 'Rechercher', 'Alertes', 'Visualisation', 'Sources', 'Mes équipes', 'Contact', and 'Paramètres'. The main area features a search bar ('Rechercher des logs') and a dropdown menu ('Haproxy'). Below these are buttons for 'Trier selon...' and 'Filtrer'. A large table lists log entries with columns: id_log, date, pid, http_status_code, taille, date_detaillee, frontend, backend, temps, terminaison, connexions, and connexe. The table contains several rows of log data. To the right of the table is a 'Filtrer' panel with sections for 'DATE', 'CHAMPS', 'TAILLE', and 'HTTP_STATUS_CODE', each with specific filter settings. A 'Valider' button is at the bottom of the filter panel.

Figure 51 : Tableau affichant les logs présents dans la base de données (tri, filtre, sélection possible)

De plus, il est possible de filtrer les logs, que ce soit de par leur date et heure dans un sens ou dans l'autre, mais également à partir des champs. Deux types de champs ont été configurés. Pour les champs quantitatifs, l'affichage choisi est celui d'une barre de slide avec deux poignées en permettant une sélection inverse. Pour les champs qualitatifs l'affichage correspond à une liste de check boxes sélectionnables.

Les deux onglets précédents ont été réalisés en parallèle des travaux de recherches sur les anomalies, le dernier onglet d'alertes n'a été réalisé qu'en fin de stage et est encore incomplet. En effet, il a fallu attendre nos résultats de détection d'anomalies avant de pouvoir le réaliser.

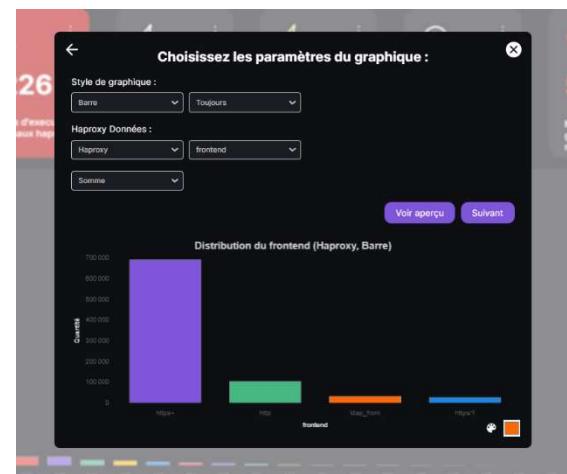


Figure 52 : Rubrique de création personnalisée de graphes (axes, titres, couleurs)

4) Alertes

L'onglet d'Alertes propose une visualisation des anomalies détectées par nos algorithmes.

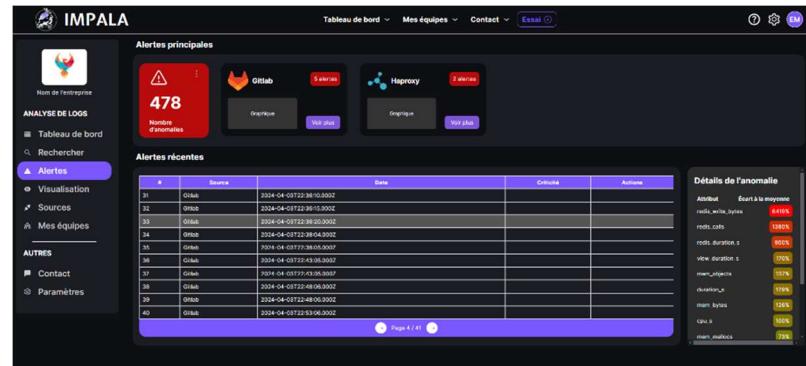


Figure 53 : Page dédiée aux anomalies détectées

Il est possible de visualiser des informations supplémentaires en sélectionnant une anomalie sur l'écart à la moyenne pour chacun des attributs significatifs ainsi que d'autres composantes statistiques.

Cet onglet pourra être poursuivi et amélioré en prenant en considération davantage de résultats.

Les autres onglets pourront être réalisés dans le futur, comme notamment les équipes qui permettront une centralisation des informations selon l'équipe de la personne. La connexion ainsi que les paramètres et les contacts qui permettent la gestion du site pourra également être implémenté dans le futur.

Cette visualisation a permis de concrétiser et structurer nos données et résultats. Celle-ci est essentiel au travail des DSI et DevOps. Elle permet une navigation plus claire et simplifiée des données et permettra de détecter les problèmes plus rapidement.

CONCLUSION

Ce stage de fin d'études, réalisé au sein de l'entreprise SOLUTEC, m'a offert une opportunité unique de mettre en pratique les connaissances acquises au cours de ma formation à l'UTC, tout en découvrant de nouveaux aspects du domaine de l'informatique. Le projet que j'ai mené, centré sur la création d'un outil d'analyse de logs, m'a permis d'approfondir mes compétences en machine learning, tout en explorant d'autres domaines tels que le développement web, le parsing de données, et la gestion de projets.

Le rapport a détaillé le processus de développement de cet outil en plusieurs phases, chacune jouant un rôle essentiel dans la réussite du projet. La phase initiale de collecte et de centralisation des logs, issue de divers services informatiques, a posé les bases d'un traitement efficace des données. Le parsing, réalisé grâce à l'algorithme de machine learning Drain 3, a permis de structurer ces données, rendant leur analyse plus pertinente. Ensuite, la phase de log mining, où j'ai mis en œuvre différents algorithmes non supervisés, a été cruciale pour identifier les anomalies et les incidents potentiels. Enfin, la création d'une interface web a permis de visualiser ces informations et d'offrir aux DSI un outil pratique et opérationnel.

Les perspectives de ce projet sont nombreuses. Tout d'abord, l'outil pourrait être étendu pour intégrer des fonctionnalités supplémentaires, telles que des alertes en temps réel ou des analyses prédictives plus avancées. De plus, l'optimisation des algorithmes de détection d'anomalies pourrait être explorée pour améliorer la précision et la performance de l'analyse. En termes de développement, l'intégration de compte utilisateur et la connexion devront être réalisés. Concernant l'algorithme de parsing, celui-ci est opérationnel mais pourra être enrichi avec l'ajout de nouveaux logs. A l'avenir, le projet a pour objectif d'être repris pour un nouveau sujet de stage qui prendra en compte un plus grand nombre de logs et de types différents.

Le bilan de ce stage est largement positif. Parmi les points forts, je retiens l'expérience précieuse acquise en gestion de projet grâce à la méthodologie Agile, qui a structuré chaque étape du développement et m'a permis d'adopter une approche itérative et collaborative. Le travail en équipe avec mon binôme et sous la supervision de notre product owner ainsi que de notre tuteur a été extrêmement enrichissant, tant sur le plan technique que relationnel. J'ai également apprécié la variété des tâches, allant du développement pur à la recherche algorithmique, ce qui a permis une diversification de mes compétences.

Ce stage a non seulement confirmé mon choix de filière en intelligence artificielle et sciences des données, mais a aussi élargi mes horizons sur d'autres domaines de l'informatique. Par ailleurs, l'expérience en gestion de projet, notamment à travers la méthode Agile, a enrichi mon parcours, ajoutant une dimension pratique et stratégique à ma formation.

En conclusion, ce stage a été une étape décisive dans mon parcours professionnel. Cette expérience m'a non seulement aidée à affiner mes compétences techniques, mais m'a également donné une meilleure compréhension des dynamiques de travail en entreprise, m'a permis de découvrir différents métiers d'ingénieur en informatique et m'a montré certains des défis liés à la gestion de projets.

TABLE DES MATIERES DETAILLEE

Remerciements	1
Sommaire	1
Résumé technique	2
Introduction	3
Première partie : Présentation de SOLUTEC et de son Lab'	4
I. Solutec	4
II. Le Lab'SOLUTEC	7
III. IMPALA	10
Deuxième partie : Les missions du stage.....	11
I. Sujet	11
II. Planning.....	13
III. Contributions	17
IV. Outils et technologies.....	18
V. Prise de recul	22
Troisième partie : La création d'un outil d'analyse de logs.....	23
Organisation du projet	23
Première sous-partie : Collecte des données et centralisation.....	25
Deuxième sous-partie : Log parsing.....	28
Troisième sous-partie : Log Mining	34
Quatrième sous-partie : Site Web	45
Conclusion.....	49
Table des matières	50
Bibliographie	51
Annexes.....	52

BIBLIOGRAPHIE

[1] He, S., He, P., Chen, Z., Yang, T., Su, Y., & Lyu, M. R. (2021). A survey on Automated Log Analysis for Reliability Engineering. *ACM Computing Surveys*, 54(6), 1–37.

<https://doi.org/10.1145/3460345>

[2] He, S., Zhu, J., He, P., & Lyu, M. R. (2016). Experience report: System log analysis for anomaly detection. 2016 IEEE 27th International Symposium on Software Reliability Engineering (ISSRE). <https://doi.org/10.1109/issre.2016.21>

[3] Fu, Y., Yan, M., Xu, Z., Xia, X., Zhang, X., & Yang, D. (2022). An empirical study of the impact of log parsers on the performance of log-based anomaly detection. *Empirical Software Engineering*, 28(1). <https://doi.org/10.1007/s10664-022-10214-6>

[4] Zhu, J., He, S., Liu, J., He, P., Xie, Q., Zheng, Z., & Lyu, M. R. (2019). Tools and benchmarks for automated log parsing. 2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP). <https://sci-hub.3800808.com/10.1109/icse-seip.2019.00021>

[5] Zhang, T., Qiu, H., Castellano, G., Rifai, M., Chen, C. S., & Pianese, F. (2023). System log parsing: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 1–20. <https://doi.org/10.1109/tkde.2022.3222417>

ANNEXES

I. GLOSSAIRE

ACP ou PCA : Analyse en Composantes Principales (CP : composante principale)

Back-end : Logique, bases de données, et communication serveur d'un site web.

DBeaver : Outil de gestion de base de données utilisé durant le projet

DCP : Dossier de Cadrage de Projet

DSI : Direction des Systèmes d'Information

ESN : Entreprise de services du numérique

Front-end : Partie visible et interactive d'un site web ou d'une application

GitLab : Plateforme de gestion de dépôts Git et de CI/CD.

Green IT : Equipe du Lab cherchant à modifier nos comportements en informatique pour réduire l'impact environnemental tout en considérant le développement durable

HAProxy : Logiciel libre offrant des solutions de répartition de charge et de proxy.

HTTP : Hypertext Transfer Protocol

IMPALA : Implémentation d'analyse de logs avancée

IT : Technologies de l'information

ITS : Infrastructure Télécommunications et Support

K-means : Algorithme de classification non supervisée utilisé en machine learning.

Machine learning : Technique permettant aux systèmes d'apprendre et de s'améliorer automatiquement.

Parsing : Analyse syntaxique d'un texte ou d'un fichier

PO : Product Owner

POC : Proof Of Concept ou preuve de concept

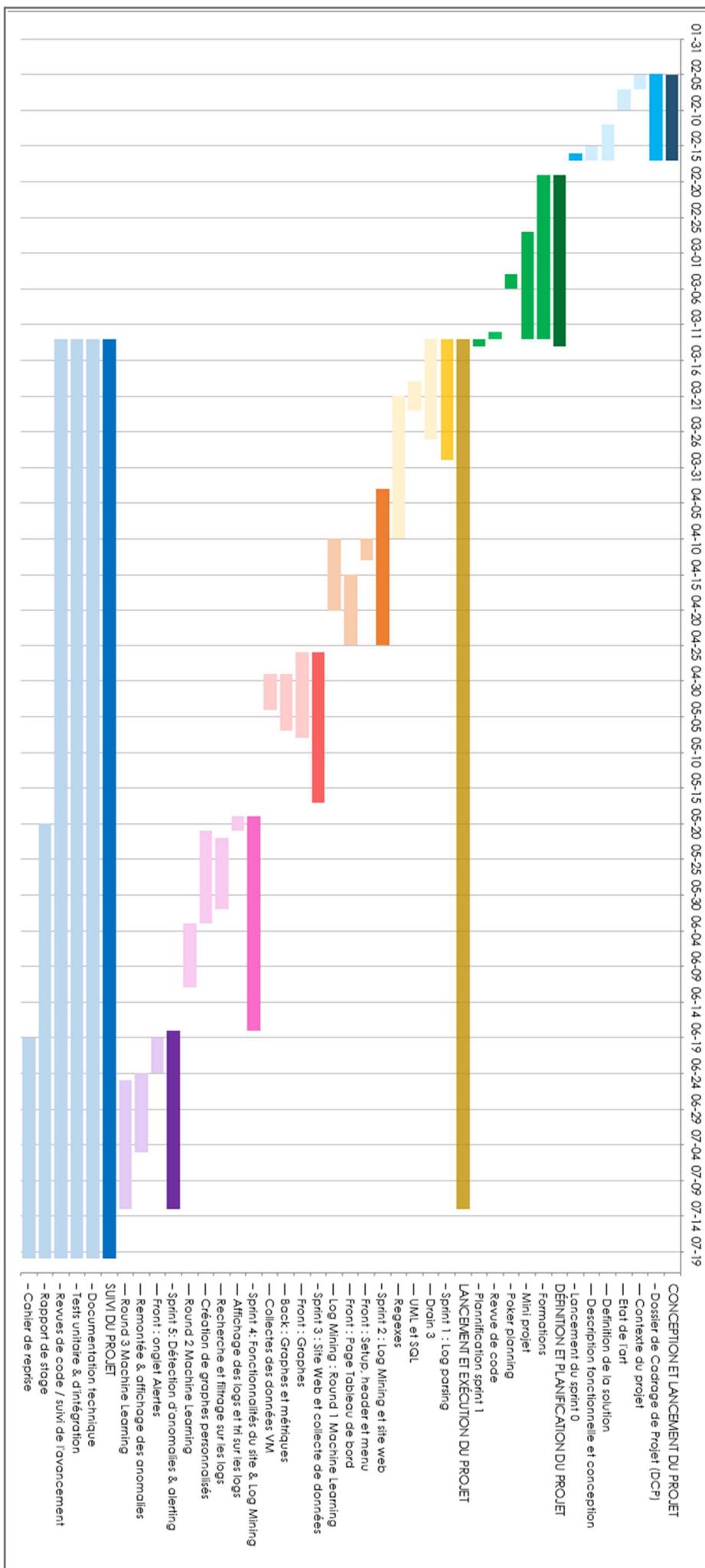
Sprint : Période de travail en méthodologie agile (dans notre cas d'une durée de 2 semaines)

SQL : Langage de requête pour gérer et manipuler les bases de données

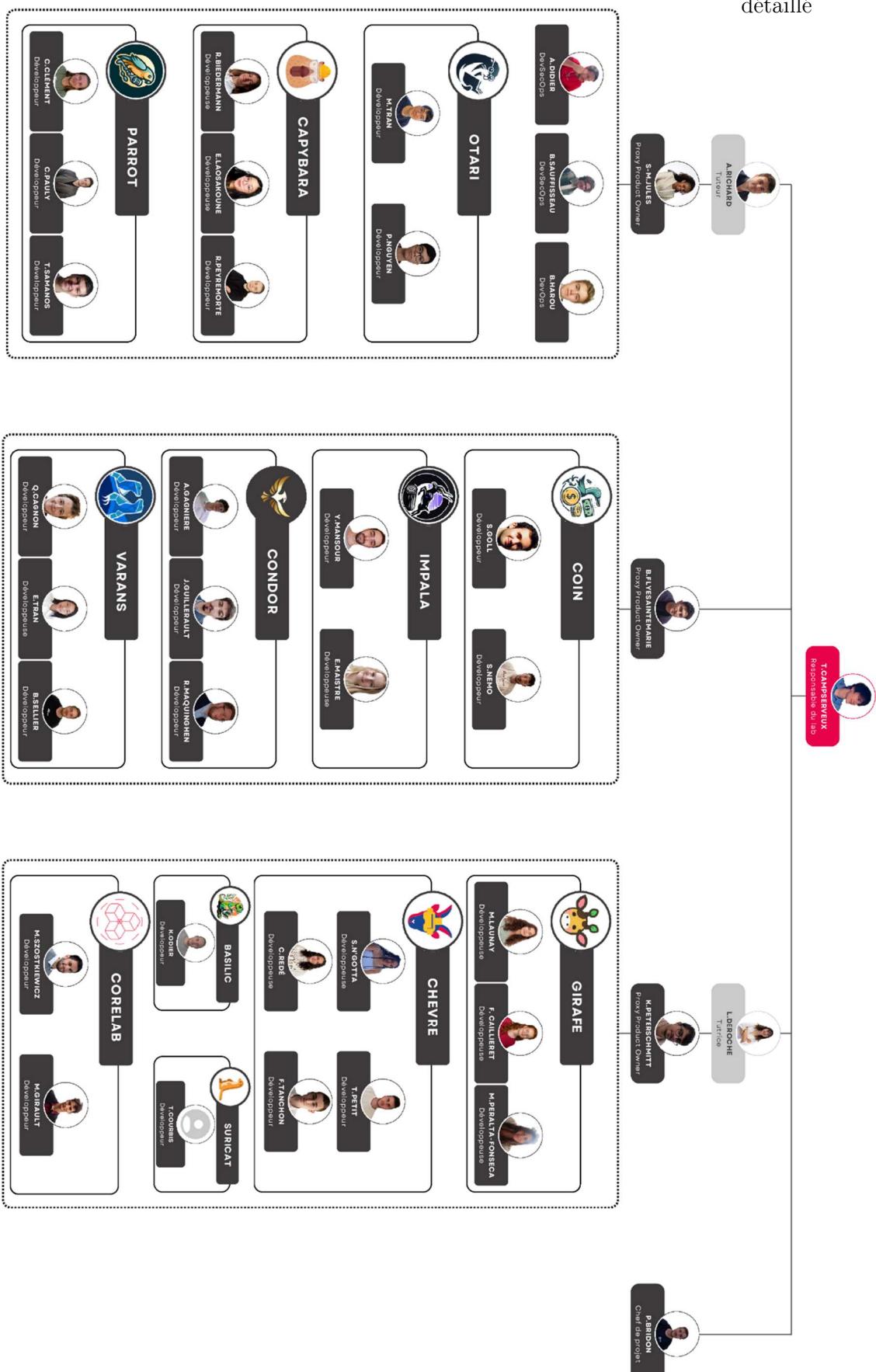
TCP : Transmission Control Protocol

VM : Machine virtuelle

II. IMAGES DETAILLEES



2) Organigramme détaillé



3) Méthode agile

