

# Travail de Bachelor

Analyse et implémentation d'un système de logging multi-niveau pour  
une plateforme Smart Grid

Étudiant :	Jael Dubey
Travail proposé par :	Jonathan Bischof DEPsys SA Route du Verney 20B 1070 Puidoux
Enseignant responsable :	Nastaran Fatemi
Année académique :	2019-2020

# 1. Cahier des charges

## 1.1. Résumé du problème

DEPsys est une entreprise Suisse leader technologique du marché énergétique. Fondée en 2012 et basée à Puidoux, elle fournit des solutions évolutives basées sur sa plateforme GridEye permettant aux réseaux de distribution d'énergie traditionnels de faire face aux nouvelles contraintes de la production décentralisée des sources d'énergie renouvelable, tels que les systèmes photovoltaïques et les technologies de mobilité électrique.

La plateforme GridEye offre une solution technologique innovatrice pour les gestionnaires de réseau de distribution (GRD, p. ex. Romande Énergie). Positionné de manière unique avec sa simplicité de déploiement, c'est la seule solution réellement Plug & Play qui évite tous les problèmes d'installation. La surveillance du réseau électrique en temps réel et les statistiques fournissent des informations détaillées sur les conditions du réseau. Les algorithmes de contrôle et de gestion garantissent la qualité et la stabilité du réseau.

Ce Travail de Bachelor consiste en une évaluation de différentes technologies de logging et de l'analyse des performances, pour ensuite mettre en place la meilleure solution pour l'infrastructure GridEye.

## 1.2. Étapes de réalisation du projet

- Étude des systèmes de gestion de logs actuels.

Choix des systèmes à évaluer.

- Évaluation théorique des systèmes.
- Prise en main et configuration des outils pour une comparaison théorique & pratique.

Choix d'un système.

- Implémentation d'un cas d'utilisation concret fourni par DEPsys.

Avec un programme de simulation minimaliste de la plateforme GridEye.

- Implémentation d'une démonstration de faisabilité (Proof of Concept, PoC).
- Implémentation de bibliothèques (SDK) pour l'interfaçage avec le Back-End de la solution GridEye (puis Front-End, puis outils externes).

## 1.3. Informations diverses et technologies

Les logs sont constitués de plusieurs types (System, User Action, Notification, ...) avec différents niveaux de priorités (debug, info, ...). Les logs utilisateurs doivent pouvoir être consultés depuis le Front-End, alors que les logs systèmes peuvent être accessibles depuis une dashboard interne. Les notifications, dépendant du niveau de priorité, doivent pouvoir être transmises en temps réel aux gestionnaires de réseau de distribution électrique par e-mail, notification push, etc. ou sous forme de rapport journalier/hebdomadaire. La mise en place de bibliothèques (SDK) est nécessaire pour communiquer avec les différents composants de l'infrastructure. Le but étant de pouvoir s'interfacer avec l'infrastructure actuelle afin de pouvoir remplacer la solution de logging actuelle.

Technologies :

- Base de données : SQL, JSON, ... (dépendant de la solution choisie).
- Back-End : Java
- Front-End : Javascript
- Outils externes : Python 3

## Table des matières

<b>1. Cahier des charges</b>	<b>2</b>
1.1. Résumé du problème . . . . .	2
1.2. Étapes de réalisation du projet . . . . .	2
1.3. Informations diverses et technologies . . . . .	2
<b>2. Introduction</b>	<b>5</b>
<b>3. Évaluation</b>	<b>5</b>
3.1. Critères d'évaluation . . . . .	5
3.2. Choix des différents systèmes à évaluer . . . . .	6
3.3. Comparaison théorique . . . . .	8
3.3.1. Elastic Stack . . . . .	8
3.3.2. Graylog . . . . .	9
3.3.3. SolarWinds Loggly . . . . .	10
3.3.4. Splunk . . . . .	11
3.4. Tests réels et prise en main des logiciels . . . . .	12
3.4.1. Test de débit d'ingestion de log . . . . .	13
3.4.2. Test d'ingestion d'une grande quantité de logs . . . . .	14
3.4.3. Test de consommation CPU . . . . .	14
3.4.4. Popularité des systèmes . . . . .	16
3.5. Conclusion de l'évaluation . . . . .	17
3.5.1. Analyse théorique des systèmes . . . . .	17
3.5.2. Analyse pratique des systèmes . . . . .	17
<b>A. Bibliographie</b>	<b>21</b>
<b>B. Détails des tests</b>	<b>21</b>
B.1. Tests de débit d'ingestion . . . . .	21
<b>C. Journal de travail</b>	<b>22</b>
C.1. Jeudi 13 février . . . . .	22
C.2. Mercredi 19 février . . . . .	22
C.3. Jeudi 20 février . . . . .	22
C.4. Mercredi 26 février . . . . .	22
C.5. Jeudi 27 février . . . . .	22
C.6. Mercredi 04 mars . . . . .	22
C.7. Jeudi 05 mars . . . . .	22
C.8. Lundi 09 mars . . . . .	23
C.9. Mercredi 11 mars . . . . .	23
C.10. Jeudi 12 mars . . . . .	23
C.11. Mercredi 18 mars . . . . .	23
C.12. Jeudi 19 mars . . . . .	23
C.13. Samedi 21 mars . . . . .	23
C.14. Dimanche 22 mars . . . . .	23
C.15. Lundi 23 mars . . . . .	24
C.16. Mardi 24 mars . . . . .	24
C.17. Mercredi 25 mars . . . . .	24
C.18. Jeudi 26 mars . . . . .	24
C.19. Mercredi 01 avril . . . . .	24
C.20. Jeudi 02 avril . . . . .	24
C.21. Mercredi 08 avril . . . . .	24

C.22.Jeudi 09 avril . . . . .	24
C.23.Lundi 13 avril . . . . .	25
C.24.Mercredi 15 avril . . . . .	25
C.25.Jeudi 16 avril . . . . .	25
C.26.Lundi 20 avril . . . . .	25
C.27.Mercredi 22 avril . . . . .	25
C.28.Jeudi 23 avril . . . . .	25
C.29.Vendredi 24 avril . . . . .	25
C.30.Mercredi 29 avril . . . . .	26

## 2. Introduction

## 3. Évaluation

### 3.1. Critères d'évaluation

Pour réaliser une évaluation de différents système de gestion de log, il faut obligatoirement choisir des critères d'évaluation. Ces critères se basent sur deux sources. La première étant les demandes formulées par DEPSys, et la deuxième provient des différentes fonctionnalités nécessaires à un système de gestion de logs. Voici les critères retenus :

- La collecte des logs

Approche minimaliste ou maximaliste, est-ce que le système installe un agent sur le dispositif émetteur de log qui lui envoie uniquement les informations les plus importantes (approche minimaliste, méthode PUSH), ou est-ce que le système reçoit tous les logs et les enregistre tous (approche maximaliste, méthode PULL) ?
- L'agrégation centralisée des logs

L'agrégation des logs est un défi, car après avoir collecté les logs, il faut tous les regrouper dans un même endroit, alors qu'ils peuvent avoir des formats différents. De plus, ils peuvent être générés très rapidement (s'exprime en EPS, Event Per Second), il faut donc être capable de traiter et regrouper ces logs de manière efficace.
- Le stockage à long terme et la durée de rétention des logs

Après avoir agrégé ces informations, il faut maintenant faire des choix quant à leur stockage. L'idéal serait de garder tous les logs indéfiniment, mais chaque information stockée à un coût. Il faut donc avoir une stratégie de rétention qui permette de supprimer ou garder tel type de log.
- La rotation des fichiers de logs

La rotation consiste à rendre automatique la stratégie de rétention et/ou de stockage des logs.
- L'analyse des logs (en temps réel et en vrac après une période de stockage)

L'analyse des logs est, en quelque sorte, le but de tout le système de gestion de logs. En effet, il ne sert à rien de stocker de l'information sur un système si l'on en fait rien. L'analyse est donc là pour synthétiser les informations contenues dans les logs.
- Les rapports

Il doit être possible pour un système de gestion des logs d'effectuer des recherches sur les informations stockées et de rédiger des rapports.
- Visionnage et gestion des alertes

Un des buts d'un système de gestion de logs est de pouvoir réagir très vite à un problème, voire même de l'anticiper. Ceci passe par une émission d'alerte et d'une possibilité de visionnage des données, si possible en temps réel.
- Popularité

Voici un critère qui change en permanence, mais qui a son importance lors d'un choix d'outil informatique. En effet, un logiciel sur le déclin sera de plus en plus dur à supporter, alors qu'un outil trop jeune n'a souvent que trop peu d'utilisateurs qui pourraient partager leurs connaissances sur les forums. L'idéal étant donc un logiciel populaire et qui est en pleine croissance.

En plus de ces critères, les coûts d'utilisation seront évalués.

### 3.2. Choix des différents systèmes à évaluer

Comme pour le choix des critères d'évaluation, les différents systèmes qui vont être analysés ont été définis soit par DEPsys, soit par des recherches dans les nombreux classement de « Log Management Tools » disponible sur internet.

Quatre classements différents ont été sélectionnés afin d'avoir plusieurs avis différents, tout en restant dans une quantité raisonnable d'analyses. Les classements qui allaient être utilisés devaient être neutre. On entend par là que le site réalisant le classement ne doit pas proposer, par exemple, une solution cloud utilisant un certain outil de gestion de log, auquel cas son classement serait forcément biaisé. Il fallait également que le classement soit récent, étant donné la vitesse d'évolution générale des outils informatiques. Une période d'un an maximum a été définie.

Les classements suivants ont été choisis :

- Top 8 BEST Log Management Software

Classement datant de mars 2020, et publié par *SoftwareTestingHelp*.

- Best Log Manager & Monitoring Software & Tools

Classement datant de janvier 2020, et publié par *iTT Systems*.

- 6 Best Log Management Tools

Classement datant de août 2019, et publié par *AddictiveTips*.

- 13 Best Log Management & Analysis Tools

Classement datant de août 2019, et publié par *Comparitech*.

Les quatre sites internet retenus sont de simples portails informatiques, publiant des articles sur divers sujets informatiques. Lorsque l'on effectue une recherche Google afin de trouver des classements de système de gestion de logs, on tombe régulièrement sur des articles de *DNSStuff* et *Sematext*. Ceux-ci n'ont pas été utilisés car le premier appartient à SolarWinds, et le second propose des solutions cloud utilisant des outils de gestion de logs. Ils ont donc été jugés non-neutres.

Afin de réaliser un classement regroupant le contenu de ces 4 tops, une note a été attribuée à chaque position de chaque classement. Deux critères ont été intégrés dans le calcul de la note : la position (mieux on est positionné dans son propre classement, plus on aura de points), ainsi que le nombre total d'outils cités. Il est en effet plus difficile d'être bien classé dans un classement contenant 5 outils que dans un classement en contenant 10. Pour finir, afin de faciliter la lecture, la note attribuée se situe entre 0 (le plus mauvais), et 100 (la meilleure note possible). La formule suivante a donc été appliquée :

$$\frac{i * 100}{n}$$

$i$  étant la position dans le classement, et  $n$  le nombre total d'outil dans le classement.

Les tableaux 1 et 2 montrent les points obtenus par chaque système.

Et le tableau 3 montre le classement global, calculé d'après l'addition des points obtenus dans les différents classements.

Avec ces résultats, on constate qu'il y a un certain nombre de systèmes appartenant à SolarWinds. Après une étude légèrement plus approfondie sur ces différents systèmes, il a été décidé de garder les suivants pour l'évaluation :

1. Elastic Stack

Choisi par DEPsys, probablement la plus populaire.

SoftwareTestingHelp	
Système	Points
SolarWinds Log Analyzer	100
Sematext Logs	89
Splunk	78
ManageEngine EventLog Analyzer	67
LogDNA	56
Fluentd	44
Logalyze	33
Graylog	22
Netwrix Auditor	11

(a) Classement de SoftwareTestingHelp

Comparitech	
Système	Points
ManageEngine EventLog Analyzer	100
SolarWinds Papertrail	92
Loggly	83
PRTG Network Monitor	75
Splunk	67
Fluentd	58
Logstash	50
Kibana	43
Graylog	33
XpoLog	25
ManageEngine SyslogForwarder	17
TekWire Managelogs	8

(b) Classement de Comparitech

TABLE 1 – Classements SoftwareTestingHelp et Comparitech

AddictiveTips	
Système	Points
SolarWinds Log & Event Manager	100
PRTG Network Monitor	83
Lepide	67
McAfee Enterprise Log Manager	50
Veriato	33
Splunk	17

(a) Classement de AddictiveTips

iTT Systems	
Système	Points
Solarwinds Log & Event Manager	100
PRTG Network Monitor	83
Lepide	67
McAfee Enterprise Log Manager	50
Veriato	33
Splunk	17

(b) Classement de iTT Systems

TABLE 2 – Classements AddictiveTips et iTT Systems

	Système	Points
1	Splunk	229
2	SolarWinds Papertrail	192
3	ManageEngine EventLog Analyzer	184
4	SolarWinds Loggly	166
5	PRTG Network Monitor	158
6	Fluentd	102
7	SolarWinds Log Analyzer	100
8	SolarWinds Log & Event Manager	100
9	ELK Stack (Kibana + Logstash)	92
10	Sematext Logs	89
11	Graylog	88
12	Lepide	67
13	LogDNA	56
14	Nagios Log Server	50
15	McAfee Enterprise Log Manager	50
16	Logalyze	33
17	Veriato	33
18	XpoLog	25
19	ManageEngine Syslog Forwarder	18
20	Netwrix Auditor	11
21	TekWire Managelogs	8

TABLE 3 – Classement global des systèmes de gestion de log

## 2. Splunk

1<sup>er</sup> du classement.

## 3. SolarWinds Loggly

De tous les outils appartenant à SolarWinds, je voulais n'en choisir qu'un. Loggly me paraissait le plus approprié au cas d'utilisation de ce travail de Bachelor.

## 4. Graylog

Suggéré par DEPSys, et semble avoir une bonne documentation.

### 3.3. Comparaison théorique

#### 3.3.1. Elastic Stack

La « Elastic Stack », ou « Suite Elastic » en français, anciennement appelée « ELK Stack » est composée de plusieurs outils :

- Elasticsearch

Un moteur de recherche RESTful.

- Kibana

Un outil de visualisation.

- Logstash

Un pipeline d'ingestion de log.

- Beat



Une famille d'agent dédié au transfert de données.

La table 4 montre un résumé des fonctionnalités disponible ainsi que des coûts lié à la Suite Elastic.

Elastic Stack	
Collecte	La collecte des logs se fait en approche minimaliste. La suite Elastic contient l'outil Beat, qui est donc une famille d'agent. On installe un agent beat (p. ex. FileBeat, MetricBeat, etc.) sur le système générant les logs, et cet agent envoie les données vers le serveur.
Agrégation centralisée	Se fait via Logstash. Peut supporter beaucoup d'événements par seconde (> 10'000 EPS). Compatible avec énormément de type de logs. Permet d'analyser et transformer les logs en temps réel. Logstash dispose d'une API permettant de créer nos propres plug-in, si les sources de données ne sont pas compatible nativement.
Stockage et rétention	Le stockage se fait avec Elasticsearch. Il n'y a pas de rétention des données de bases avec Elasticsearch. Il est cependant possible de le faire avec Elastic-Curator, qui est un outil permettant de gérer un cluster Elasticsearch.
Rotation	La rotation se fait avec Elastic-Curator
Analyse	Elasticsearch et ses requêtes poussées permettent de faire des recherches avancées.
Rapport	Les rapports peuvent être générés depuis Kibana.
Visionnage et alertes	La visualisation des données en temps réels peut se faire avec Kibana. La gestion des alertes se fait également via Kibana. Il est possible de paramétrer des alertes classiques, qui se déclenchent suivant des règles précises. Et il est également possible de paramétrer des alertes suivant un algorithme d'apprentissage automatique, qui détectera des événements inhabituels.
Popularité	La suite Elastic est très certainement la plus populaire actuellement. Elle bénéficie d'une grande communauté active. Au niveau de la tendance, on peut voir une grande croissance entre les années 2016 et 2019. Ces derniers mois, cela semble se stabiliser.
Coûts	La suite Elastic propose différents abonnements. Il y a une offre gratuite, mais celle-ci ne contient pas de gestion d'alerte et de création de rapport. Les prix pour les offres payantes ne sont pas publics. Il faut contacter Elastic et les prix varient en fonction de la taille du système à implémenter.

TABLE 4 – Résumé théorique de Elastic Stack

### 3.3.2. Graylog

Graylog un outil de gestion de logs. Il dispose de deux versions : Open Source et Enterprise. La table 5 montre les fonctionnalités du systèmes Graylog.

Graylog	
Collecte	La collecte des logs se fait en approche minimaliste. Graylog possède un outil appelé « Sidecar » qui permet de gérer plusieurs type d'agent, y compris l'outil Beat de la suite Elastic.
Agrégation centralisée	Graylog permet de gérer « d'énormes » jeux de données et de les traiter selon des règles définies par l'utilisateur. En plus des règles classiques, comme la géographie, le type, etc., Graylog permet de faire des listes noires de logs.
Stockage et rétention	Le stockage se fait avec MongoDB et Elasticsearch. Graylog offre une solution (Graylog Archive) de rétention des données, disponible avec la version Enterprise, qui peut être paramétrée.
Rotation	La rotation se fait avec Graylog Archive.
Analyse	Graylog utilisant Elasticsearch, il dispose de ses requêtes poussées permettant de faire des recherches avancées. Graylog possède également son langage de requête, basé sur Apache Lucene.
Rapport	La création de rapport est une fonctionnalité de Graylog Enterprise. Il est possible de les configurer depuis l'interface de Graylog.
Visionnage et alertes	La visualisation des données en temps réels se fait avec l'interface de Graylog. La gestion des alertes est incluse dans le Graylog de base. Elle permet de définir des alertes selon des règles. Graylog possède également un « Store » proposant, entre autre, des fonctionnalités liées aux alertes, développées par la communauté.
Popularité	Graylog n'est pas arrivé très haut de manière générale dans les tops, mais il est en revanche souvent cité. La courbe de tendance de Graylog est en croissance régulière depuis 2008.
Coûts	Graylog propose deux versions : « Open Source » et « Enterprise ». La première est gratuite mais propose quelques fonctionnalités en moins, comme les rapports programmés ou le support technique. Les coûts de la version « Enterprise » ne sont pas disponibles, il faut contacter Graylog. À noter que la version « Enterprise » est gratuite jusqu'à une utilisation de 5 GB par jour.

TABLE 5 – Résumé théorique de Graylog

### 3.3.3. SolarWinds Loggly

SolarWinds Loggly un outil de gestion de logs SaaS (Software-as-a-Service, dans le cloud). Il dispose de plusieurs versions, dont une gratuite. La table 6 montre les fonctionnalités et des informations sur les coûts de SolarWinds Loggly.

SolarWinds Loggly	
Collecte	L'envoi de données à Loggly est relativement simple. La seule contrainte est qu'il s'agisse de texte. Il n'y a pas besoin d'avoir d'agent (envoi de log via un endpoint), mais il est également possible d'en utiliser.
Agrégation centralisée	
Stockage et rétention	La rétention des données est plutôt courte (7 jours dans la version gratuite), et ensuite, les logs peuvent être sauvegardés dans une instance S3 de AWS.
Rotation	La rétention se fait automatiquement après un certain nombre de jour.
Analyse	Loggly possède son propre langage de requête, qui est basé sur Apache Lucene. Il est également possible d'analyser les logs en temps réel via l'interface de Loggly.
Rapport	Il est possible de réaliser des rapport dans plusieurs format depuis l'interface graphique.
Visionnage et alertes	Il est possible de configurer des alertes selon des règles classiques, et également sur des événements inhabituels. Le visionnage des données en direct se fait via l'interface graphique de Loggly.
Popularité	Loggly était en croissance entre 2008 et 2016, mais depuis cette année-ci, sa courbe de tendance est en décroissance.
Coûts	Loggly propose une version gratuite et trois versions payantes. La version gratuite est limitée à un volume de 200 MB par jour, un seul utilisateur pouvant se connecter sur une instance, et ne contient pas certaines fonctionnalités comme la gestion des alertes. Les versions payantes varient entre 79 et 279 USD par mois, et proposent chacune quelques fonctionnalités en plus, et un volume de moins en moins limité.

TABLE 6 – Résumé théorique de SolarWinds Loggly

### 3.3.4. Splunk

Splunk est un outil de gestion de logs. Il est séparé en trois « parties », chacune étant responsable de plusieurs choses :

- Universal Forwarder

Effectue la collecte et envoie les données à l'indexer.

- Indexer

Effectue le stockage des logs.

- Search Head

Permet de lire dans l'index.

Chacune de ces fonctions peut être transformé en cluster si de grand volume de données sont à traiter. Splunk peut être disponible en version « sur site » ou « cloud ». La table 7 montre les fonctionnalités et les coûts relatif à Splunk.

Splunk	
Collecte	La collecte des logs se fait en approche minimaliste. Ceci via les « Universal Forwarder » qui sont des agents à installer sur le système générant les logs. Il envoie ensuite les données vers l'indexer.
Agrégation centralisée	
Stockage et rétention	Le stockage des logs se fait avec l'indexer. Concernant la rétention des données, Splunk la gère avec des « buckets ». Concrètement, un bucket possède une durée qui détermine le temps qu'une donnée va passer dedans. P. ex., on peut avoir un bucket de 30 jours où les logs seront accessibles et analysables librement. Puis, passé ces 30 jours, ils iront dans un autre bucket où ils seront compressés pour le stockage à long terme.
Rotation	Se fait via les buckets.
Analyse	L'analyse est possible via l'interface de Splunk. Celle-ci permet de trier et filtrer les logs selon de nombreux critères.
Rapport	La génération de rapport est possible depuis l'interface de Splunk.
Visionnage et alertes	Le visionnage et la gestion des alertes est également possible depuis l'interface de Splunk. Pour les alertes, elles sont définissables selon des règles classiques (p. ex. logs provenant d'une adresse IP particulière, etc.).
Popularité	D'après les courbes de Google Trends, le système Splunk est en constante croissance depuis 2010. Dans le classement des différents tops consultés, Splunk arrive en bonne position. Splunk bénéficie d'une documentation relativement grande et d'un forum.
Coûts	Les coûts sont en « infrastructure-based pricing » et ne sont donc pas fixes. Mais Splunk Enterprise commence à 150\$ par mois. Splunk ne propose pas de version gratuite (mise à part l'essai de 60 jours).

TABLE 7 – Résumé théorique de Splunk

### 3.4. Tests réels et prise en main des logiciels

Plusieurs tests ont été effectués afin de pouvoir effectuer une comparaison de ces systèmes. Comme la plupart de ces tests obligent une prise en main concrète des logiciels, ceux-ci ont également permis d'avoir une idée de la facilité d'utilisation, de la flexibilité ainsi que de l'utilisabilité (« user-friendliness ») des systèmes.

Voici les versions des logiciels utilisées pour effectuer ces tests :

- Elastic Stack
  - Elasticsearch 7.6.2
  - Filebeat 7.6.2
  - Logstash 7.6.2
  - Kibana 7.6.2
- Graylog
  - Graylog 3.2.4-1 (Virtual Appliance)
  - Filebeat 7.6.2
- Splunk
  - Splunk 8.0.2.1
  - Splunk Universal Forwarder 8.0.2.1

- Loggly

Loggly Lite (Cloud)

### 3.4.1. Test de débit d'ingestion de log

Le premier test de cette comparaison consiste en un test de débit. Cela permet d'avoir une première idée de la performance des applications.

#### Conditions de tests

- Chargement d'un fichier de logs.

Le fichier contient 98'305 logs identiques.

Log : 2020-02-27 09 :06 :24.596 INFO o.s.s.c.ThreadPoolTaskScheduler -> Shutting down ExecutorService 'taskScheduler'

- On effectue un test d'ingestion simple (on stocke le log brut).
- On effectue un test d'ingestion avec filtre (on parse les différentes parties du log).

Date et heure : 2020-02-27 09 :06 :24.596

Niveau du log : INFO

Classe Java : o.s.s.c.ThreadPoolTaskScheduler

Message : Shutting down ExecutorService 'taskScheduler'

- Le résultat est un débit exprimé en EPS (Event Per Second), qui correspond aux nombres de logs que le système aura pu ingérer en une seconde.

La table 8 montre **les résultats** des tests d'ingestion.

	Elastic Stack	Graylog	Splunk	Loggly
Sans filtrage	4'898	14'044	-	46'295 (*)
Avec filtrage	9'181	7'868	2'628	-

TABLE 8 – Résultats des tests de débit d'ingestion

(\*) Loggly n'acceptant que des fichiers de taille inférieure à 5 MB, le fichier a été réduit à 3'999 logs.

Splunk favorise l'extraction d'informations (parsing) après avoir stocké les logs dans le système. Il est donc compliqué d'ajouter un filtrage avant le stockage et ça n'a donc pas été réalisé. Par contre, Splunk effectue automatiquement une extraction de la date des logs. Son débit a donc été classé dans la catégorie « avec filtrage ». Loggly ayant été écarté assez tôt de l'évaluation, la partie « avec filtrage » n'a pas été testée.

## Conclusion

Sans filtrage, on constate que Graylog possède un débit supérieur à Elastic Stack. Loggly n'acceptant pas des fichiers de grande taille, il est difficilement comparable. Avec filtrage, étonnement, la Suite Elastic gagne en débit. Elle possède donc un meilleur débit que Graylog et Splunk. Après ce test, outre les résultats comptables, il se dessine surtout une tendance vers Graylog et Elastic Stack, car ces deux systèmes sont plus « ouvert » que les autres. Il n'y a pas eu de difficultés particulières lors de l'utilisation, pas de contraintes comme la taille du fichier ou encore le moment du filtrage au sein du pipeline.

### 3.4.2. Test d'ingestion d'une grande quantité de logs

Ce test a pour but de mettre en évidence les limites de certains systèmes lors de l'ingestion de gros fichier de logs. Il permet également d'avoir une brève vue sur la constance du débit mesuré lors du test précédent.

#### Conditions de tests

- Chargement d'un fichier de logs.

Le fichier contient 999'999 logs identiques.

Log : 2020-02-27 09 :06 :24.596 INFO o.s.s.c.ThreadPoolTaskScheduler -> Shutting down ExecutorService 'taskScheduler'

- On n'effectue aucun traitement sur le log.
- Le résultat est soit une limite supérieure, soit « > 1'000'000 »

La table 9 montre **les résultats** du test d'ingestion d'une grande quantité de logs.

Elastic Stack	Graylog	Splunk	Loggly
> 1'000'000	> 1'000'000	> 1'000'000	~4'000 (5 MB)

TABLE 9 – Résultats des tests d'ingestion de grande quantité de logs

#### Conclusion

Ce test permet simplement de montrer que les systèmes travaillant avec un agent, soit selon la méthode PUSH, n'ont pas de problème de taille de fichier, car l'agent lit et envoie les logs au fur et à mesure. Loggly est le seul ici utilisant la méthode PULL. Au niveau des débits, ceux d'Elastic Stack et Graylog ont été divisé par 2, alors que celui de Splunk est resté stable. Pour Loggly, il n'y a pas eu de différence, étant donné qu'il est limité en taille de fichier.

### 3.4.3. Test de consommation CPU

Ce test-ci a pour but de pouvoir comparer les systèmes de gestion de logs Elastic Stack et Graylog en terme de performance.

#### Conditions de test

- Chargement du même fichier de log que pour le test de débit.
- L'utilisation du CPU sera réduite au minimum hors système à tester.
- Ordinateur de test : ASUS UX360UAK
- Processeur : Intel Core i7-7500U CPU @ 2.70GHz x 4
- Lors de la réception, l'agent Filebeat se situera sur une autre machine.

#### Résultats

La figure 1 montre la consommation CPU et l'historique du trafic réseau lors de la réception des logs. La figure 2 montre les mêmes informations, mais pour le système Graylog. Enfin, la figure 3 montre ces informations-ci avec l'agent Filebeat.

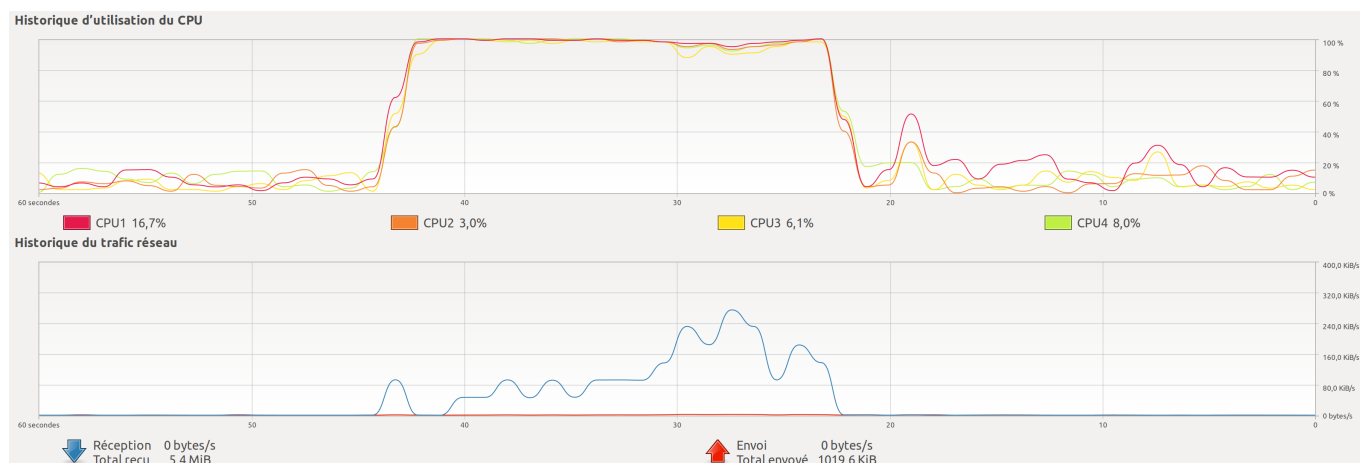


FIGURE 1 – Consommation CPU Elastic Stack

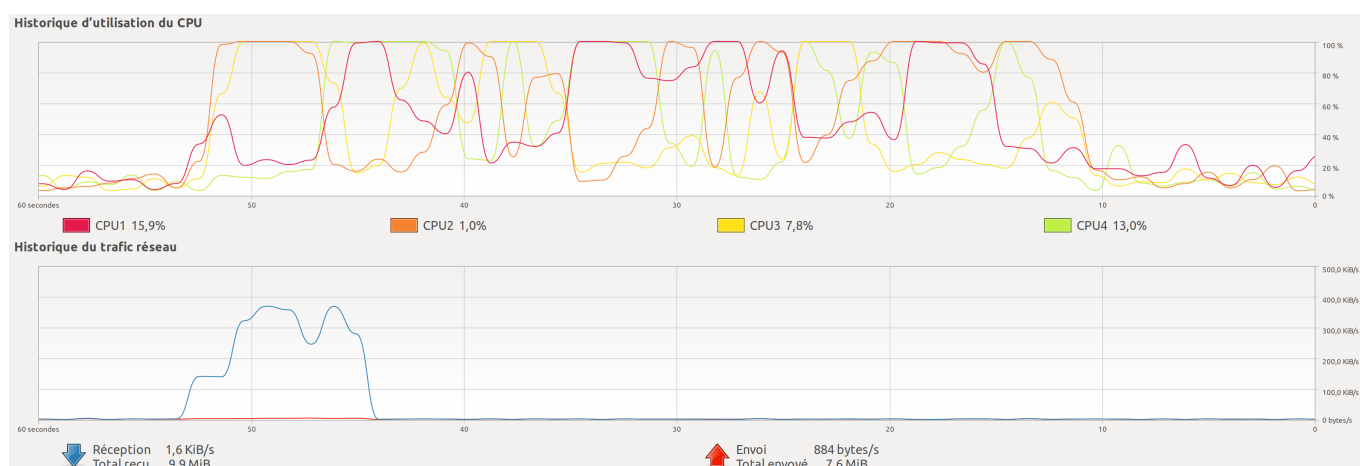


FIGURE 2 – Consommation CPU Graylog

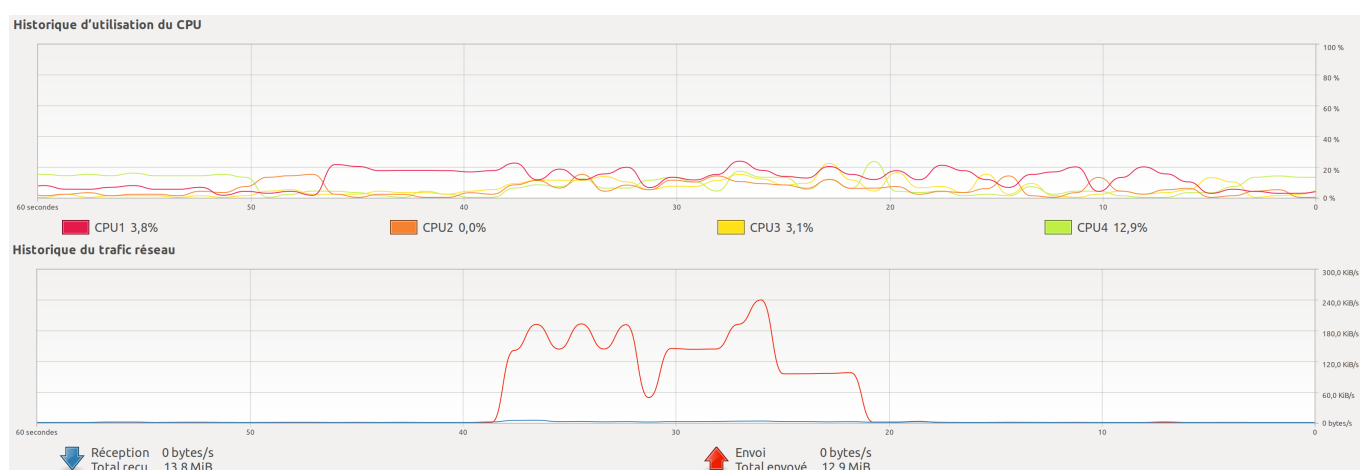


FIGURE 3 – Consommation CPU Filebeat

## Conclusion

Premièrement, on constate que l'agent Filebeat est un agent très léger puisque son action ne se fait quasiment pas ressentir sur le taux d'utilisation du CPU. Pour ce qui est du taux d'utilisation CPU par la Suite Elastic, on remarque que lors de la réception et le stockage des logs, tous les CPUs tournaient à 100% de leur capacité,

et la charge dure environ 25 secondes. Pour Graylog, les CPUs ne sont pas autant sollicités, ou du moins pas tous en même temps. Ils varient entre 20% et 100% et la distribution est plutôt égale entre les CPUs. On peut donc estimer un taux d'utilisation moyen à 60-70%. La charge dure cette fois-ci environ 45 sec. À noter que Graylog ingère les logs en env. 7 secondes, puis travaille encore pendant env. 35 secondes, alors que la Suite Elastic fait tout en même temps. On peut constater ceci grâce à la courbe de l'historique du trafic réseau.

### 3.4.4. Popularité des systèmes

La comparaison de popularité des différents systèmes permet de se faire une idée sur les tendances actuelles et passées d'utilisation de ces systèmes. Elle a également pour but de pouvoir éventuellement anticiper les futures tendances. Ces données sont importantes lors du choix d'un logiciel, afin de pouvoir compter sur un support de la communauté lors des problèmes qui arriveront pendant le développement.

#### Conditions de tests

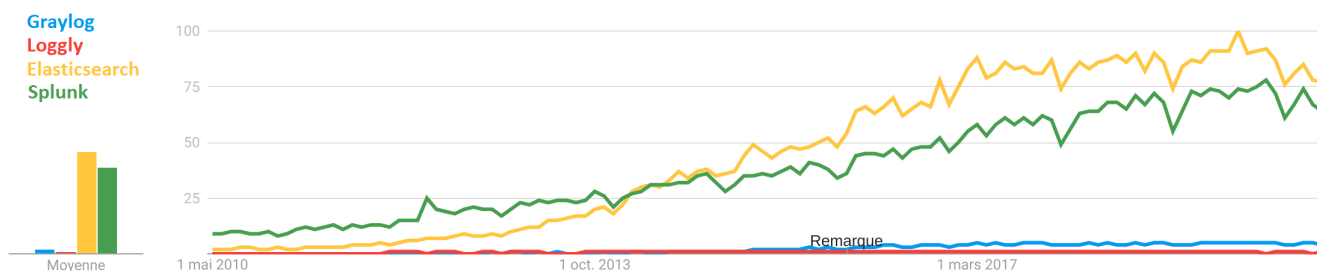
- Les courbes de tendances seront tirées de Google Trends.
- Les courbes porteront sur les 10 dernières années.

Du 23 avril 2010 au 23 avril 2020

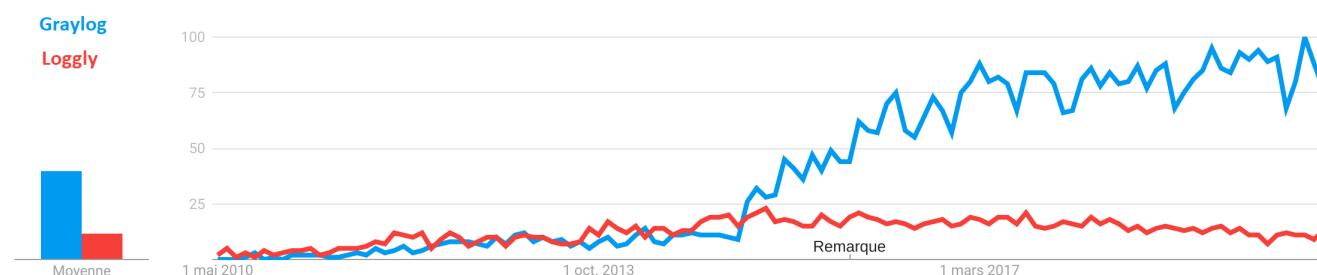
- Les courbes porteront sur les recherches dans tous les pays du monde.
- Il n'y aura pas d'autre restriction de recherche (catégorie, outil Google)

#### Résultats

Courbes de tendance des 4 systèmes :



Courbes de tendance de Graylog et Splunk :



La Suite Elastic contenant 4 logiciels séparés, le choix ici a été d'inclure uniquement « Elasticsearch », car c'est la plus connue des quatre applications, et que les résultats sont meilleurs qu'avec le terme « Elastic ».



Suite » (il faut également savoir que la suite a changé de nom en 2016, passant de « ELK Stack » à « Elastic Stack », ce qui ne favorise pas la recherche dans Google Trends).

## Conclusion

On distingue deux groupes dans cette analyse : Elasticsearch et Splunk d'un côté, Graylog et Loggly de l'autre. Le premier est très populaire et en croissance, alors que le deuxième ne que peu tendance, et Loggly n'est pas en croissance. Au sein même du groupe Graylog-Loggly, on peut encore constater une grande différence, Graylog étant beaucoup plus recherché sur Google que Loggly. On peut donc affirmer que la Suite Elastic est le système le plus populaire, devant Splunk. Loin derrière vient Graylog et encore plus loin, on trouve Loggly.

## 3.5. Conclusion de l'évaluation

### 3.5.1. Analyse théorique des systèmes

Au niveau fonctionnel, les quatre systèmes de gestion de logs ne se différencient pas énormément. Ils suivent à peu près tous le même schéma et permettent tous de faire les mêmes actions (reporting, alerting, analyse, etc.). Loggly est légèrement différent dans le sens où son approche de collecte de log par défaut est maximaliste (approche PULL). Il met en effet un endpoint à disposition, sur lequel on peut envoyer des données. Il s'agit là d'une différence mineure car bien que les autres systèmes privilégient et facilitent l'approche minimaliste, il est également possible de configurer un endpoint semblable pour les systèmes Elastic Stack, Graylog et Splunk. Pour ce qui est du non-fonctionnel, il y a quelques systèmes qui se démarquent. Premièrement, Loggly de SolarWinds est uniquement disponible dans une version Cloud. La Suite Elastic et Splunk sont eux disponibles tant dans une version Cloud que dans une version sur site. Graylog lui est uniquement disponible dans une version locale. Il y a également l'architecture des systèmes. La Suite Elastic est formée de 3 applications indépendantes, et en utilise une 4<sup>ème</sup> pour l'agent Beat. Graylog est composé d'un système, et utilise des agents ne lui appartenant pas, comme Beat. Splunk suit la même architecture que Graylog, mais possède son propre agent, le Splunk Forwarder. Et Loggly est une infrastructure cloud sans agent. Globalement, l'analyse théorique des systèmes de gestion de logs ne permet pas de faire un choix concret sur un système ou un autre pour la réalisation d'une solution destinée au système GridEye.

### 3.5.2. Analyse pratique des systèmes

# ANNEXES

## Liste des tableaux

1.	Classements SoftwareTestingHelp et Comparitech . . . . .	7
2.	Classements AddictiveTips et iTT Systems . . . . .	7
3.	Classement global des systèmes de gestion de log . . . . .	8
4.	Résumé théorique de Elastic Stack . . . . .	9
5.	Résumé théorique de Graylog . . . . .	10
6.	Résumé théorique de SolarWinds Loggly . . . . .	11
7.	Résumé théorique de Splunk . . . . .	12
8.	Résultats des tests de débit d'ingestion . . . . .	13
9.	Résultats des tests d'ingestion de grande quantité de logs . . . . .	14

## Table des figures

1.	Consommation CPU Elastic Stack . . . . .	15
2.	Consommation CPU Graylog . . . . .	15
3.	Consommation CPU Filebeat . . . . .	15

## A. Bibliographie

## B. Détails des tests

### B.1. Tests de débit d'ingestion

Graylog (3.2.4 Virtual appliance) avec Filebeat :

Sans extractor (module de Graylog permettant le parsing des logs) :

Début du chargement : 10 :09 :17

Fin du chargement : 10 :09 :24

Temps de chargement : 7 secondes

Débit moyen : ~14'043,47 EPS

Avec extractor :

Début du chargement : 07 :46 :47.153

Fin du chargement : 07 :46 :59.648

Temps de chargement : 12.495 secondes

Débit moyen : ~7'867,55 EPS

Elastic Stack sans Logstash :

Début du chargement : 10 :34 :34.200

Fin du chargement : 10 :35 :03.622

Temps de chargement : 29.422 secondes

Débit moyen : ~3'341,21 EPS

Elastic Stack complet (sans filtrage du log) :

Début du chargement : 10 :44 :11.851

Fin du chargement : 10 :44 :31.920

Temps de chargement : 20.069 secondes

Débit moyen : ~4'898,35 EPS

Elastic Stack complet (avec filtrage du log) :

Début du chargement : 10 :26 :37.308

Fin du chargement : 10 :26 :48.015

Temps de chargement : 10.707 secondes

Débit moyen : ~9'181,38 EPS

Splunk :

Avec filtrage :

Donné (metrics.log de Splunk) : 2'628 EPS

Solarwinds Loggly avec endpoint bulk : Avec Loggly, il est impossible de charger un fichier de plus de 5 MB. Il a donc été réduit à 39'999 lignes.

Début du chargement : 08 :42 :11.360

Fin du chargement : 18 :42 :12.224

Temps de chargement : 0.864 seconde

Débit moyen : ~46'295,14 EPS

## C. Journal de travail

### C.1. Jeudi 13 février

Première réunion avec Nastaran, Jonathan et Pascal. Jonathan et Pascal ont expliqué leur vision du TB à travers une présentation, puis nous avons planifié le travail de Bachelor. Notamment les dates de fin d'évaluation (avec la présentation à DEPsys), et de fin de développement du use-case.

### C.2. Mercredi 19 février

Début du Travail de Bachelor. J'ai commencé par suivre un tuto afin de maîtriser les bases du langage  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ , ce qui me sera utile pour tout ce qui est rédactionnel. Ensuite, j'ai commencé à revoir la présentation de Pascal afin de bien comprendre (notamment les technologies que je ne connais pas).

### C.3. Jeudi 20 février

J'ai regardé plusieurs vidéos qui présentent les différentes technologies que je dois évaluer.

### C.4. Mercredi 26 février

J'ai décidé de commencer à évaluer plus en profondeur Elasticsearch en premier, car Prometheus a comme contrainte de ne pas gérer les logs textuels, mais uniquement des métriques numériques. Cependant, d'après plusieurs lectures, je pense qu'il pourrait être intéressant de mixer les deux solutions. J'ai donc installé les outils de la suite ELK, et suivi des tutos plus concret en ce qui concerne Elasticsearch (insertion de donnée, recherches, etc.).

### C.5. Jeudi 27 février

Deuxième réunion avec Nastaran. Elle me propose de recentrer mes recherches sur la partie « Log Analysis », donc rechercher directement l'intégration de l'analyse de logs avec ELK par exemple. Après la réunion, j'ai donc continué mes recherches dans ce sens et ai suivi la vidéo de elastic qui concerne l'analyse de logs.

### C.6. Mercredi 04 mars

J'ai commencé à écrire ce journal afin de mieux me rappeler de ce que j'ai fait, ainsi que d'être plus structuré. Je commence également à utiliser Zotero, qui permet d'enregistrer tous les liens que je trouve intéressant, ainsi que de créer une bibliographie. J'ai également décidé de m'intéresser à Graylog en plus de ELK.

### C.7. Jeudi 05 mars

J'ai exploré plus en profondeur les articles de type « Elastic Stack versus Graylog », et je vais donc inclure la stack « Graylog server, MongoDB et Elasticsearch » dans le comparatif. Cette suite-là me semble très appropriée au traitement et à l'analyse de logs. J'ai été à la réunion avec Nastaran à 11h30. Suite à cette réunion, nous avons décidé qu'il fallait que je fasse une synthèse des mes recherches et que je la présente en quelques slides le jeudi 12 mars.

## C.8. Lundi 09 mars

J'ai commencé à faire la synthèse de mes recherches. Je vais donc la faire en 3 étapes :

1. Choix des critères d'évaluations
  - a) Selon des recherches au sujet des caractéristiques d'un « Log Management Tool »
2. Choix des outils à évaluer
  - a) Pour cette étape, je vais consulter plusieurs classements de système de gestion de logs et choisir ceux qui sont le plus souvent cités. Je vais probablement en prendre 5 ou 6.
3. Synthèse et rédaction des slides

Ce lundi, j'ai défini les critères d'évaluation, selon les demandes de DEPSys ainsi que les critères lu lors de mes recherches.

## C.9. Mercredi 11 mars

J'ai fait un tableau pour le choix des outils à tester. J'ai donc effectué un classement selon 4 tops de système de gestion de logs. J'ai également commencé l'évaluation à proprement parler, en particulier sur Elastic Stack et Graylog. J'ai également eu un problème de stockage de la base de donnée Zotero et j'ai perdu toute ma bibliographie.

## C.10. Jeudi 12 mars

J'ai continué l'évaluation avec Loggly, j'ai créé la présentation de synthèse pour la réunion avec Nastaran, puis je l'ai présentée.

## C.11. Mercredi 18 mars

J'ai remis en place Zotero, cette fois avec un synchronisation en ligne de ma bibliographie. J'ai analysé les différents systèmes de gestion de logs que j'hésitais à inclure dans l'évaluation. J'ai donc écarté ManageEngine EventLog Analyzer pour sa popularité vraiment faible et son manque de documentation, et PRTG Network Monitor, qui est très axé sur l'analyse d'un réseau, comme son nom l'indique. Je vais donc évaluer Splunk.

## C.12. Jeudi 19 mars

J'ai fait l'évaluation de Splunk. J'ai également eu la réunion hebdomadaire avec Nastaran.

## C.13. Samedi 21 mars

J'ai reformaté mon rapport avec le template L<sup>A</sup>T<sub>E</sub>X écrit par Mateo Tutic. J'ai également développé la partie Choix des différents systèmes à évaluer.

## C.14. Dimanche 22 mars

J'ai continué la partie Choix des différents systèmes à évaluer. J'ai également télécharger la suite Elastic et testé avec les logs systèmes de Ubuntu. Cela fonctionne normalement.

### C.15. Lundi 23 mars

J'ai mis en place les systèmes de gestion de logs Elastic Stack, Graylog, Splunk et Loggly. J'ai testé (en insérant des logs et regardant le débit) les 3 premiers. Encore quelques problèmes pour Loggly (pour l'instant, il sauvegarde 1 log avec n lignes dans le message plutôt que n logs avec 1 ligne). J'ai également terminé les tableaux récapitulatif de l'évaluation de chaque système.

### C.16. Mardi 24 mars

J'ai terminé les tests d'ingestions de logs pour les 4 systèmes. J'ai commencé à faire mes slides pour la présentation du 25 mars.

### C.17. Mercredi 25 mars

J'ai terminé les slides de la présentation. J'ai fais la présentation du travail de Bachelor à l'entreprise DEPsys. S'en est suivi une discussion avec Pascal, Jonathan, Nastaran et moi au sujet de la suite de l'évaluation de mon TB, puis un ajustement du cas d'utilisation à implémenter fourni par Pascal.

### C.18. Jeudi 26 mars

J'ai commencé à refaire des tests d'ingestion de log. Cette fois-ci avec tout le pipeline. Je commence avec Elastic Suite, en y intégrant logstash afin qu'il filtre les données.

### C.19. Mercredi 01 avril

J'ai continué les tests d'ingestion avec Elastic Suite et rencontré beaucoup de problème. J'arrive à faire fonctionner un pipeline Logstash-Elasticsearch-Kibana, et un pipeline Filebeat-Elasticsearch-Kibana, mais pas un contenant les 4 logiciels de la suite.

### C.20. Jeudi 02 avril

J'ai continué les tests en tentant plusieurs tutoriaux trouvé sur internet. Mais je rencontre toujours des problèmes. Ils sont probablement liés à la communication entre Filebeat et Logstash. J'ai également suivi les tuto officiels de la Suite Elastic, mais ça n'a pas fonctionné non plus. Ceci est peut-être dû à mes fichiers de configurations des logiciels Filebeat et Logstash. J'ai ensuite eu une réunion avec Nastaran.

### C.21. Mercredi 08 avril

J'ai continué les tests d'ingestion. En suivant les guides du site d'elastic, j'ai remarqué qu'il y avait toute une section expliquant l'utilisation de la Suite avec Docker. Je me suis dit qu'il y avait plus de chance que cela fonctionne étant donné l'uniformité que propose Docker. Malheureusement, j'ai toujours les mêmes problèmes. Même en prenant un git public sensé fonctionner. Je me dit alors que le problème vient peut-être de mes fichiers de logs de tests (ils contiennent le même log multiplié n fois).

### C.22. Jeudi 09 avril

Je suis repassé sur une version non dockerisée de la suite Elastic. J'ai téléchargé un fichier de log d'un serveur Apache afin de tester la Suite avec un fichier de log réel, et fait d'autres modifications, notamment sur les fichiers de configuration (j'ai créé la partie « filtrage » de Logstash avec un site internet permettant de créer ces filtres de manière incrémentale). Et ça a fonctionné. J'ai ensuite eu la réunion avec Nastaran.



### C.23. Lundi 13 avril

J'ai effectué les tests de performances avec filtrage de la Suite Elastic. Après ceci, je me suis lancé dans les tests avec filtrage de Graylog. Cette fois-ci, ce n'est pas avec un logiciel intermédiaire comme Logstash, mais avec une fonctionnalité intégrée à Graylog : les Graylog Extractors.

### C.24. Mercredi 15 avril

J'ai commencé à chercher une façon d'effectuer le filtrage avec le système de gestion de logs Splunk. Malheureusement, j'ai l'impression que cela va être plus compliqué car Splunk favorise l'extraction des informations après l'indexage. Je vais encore chercher une journée, et si ce n'est pas concluant, je passerai outre.

### C.25. Jeudi 16 avril

J'ai tenté d'effectuer l'extraction d'informations dans les logs durant la phase de "parsing" du pipeline de Splunk. J'ai vu qu'il devait être possible de le faire en modifiant des fichiers de configuration dans le répertoire de Splunk, mais cela n'a pas fonctionné.

### C.26. Lundi 20 avril

J'ai rédigé les résultats des tests d'ingestion. J'ai ensuite commencé à étudié les manières de faire des tests de consommation CPU. Sachant qu'avec Amazon Web Service (AWS), comme je l'avais vu quelques semaines plus tôt dans un cours de Cloud Computing, il est possible de monitorer différentes métriques d'une instance, entre autres l'utilisation du CPU, je me suis lancé dans une installation de la Suite Elastic sur une instance t2.micro d'AWS. Malheureusement, ces instances sont trop petites et ne supportent pas simplement Elasticsearch. Ne voulant pas payer pour des instances plus grosses, je me suis rabattu sur la solution locale. Je vais donc simplement stopper le maximum de processus et monitorer l'utilisation de mon CPU avec l'outil natif d'Ubuntu. Je dois aussi installer Filebeat et Graylog sur un autre ordinateur afin de pouvoir effectuer ces tests (dans la réalité, le serveur et le client ne seront pas sur la même machine).

### C.27. Mercredi 22 avril

J'ai effectué les tests de consommation CPU de la Suite Elastic, de Graylog, ainsi que de Filebeat. J'ai ensuite commencé à rédiger la synthèse de ces tests.

### C.28. Jeudi 23 avril

J'ai continué la rédaction, j'ai ajouté le test d'ingestion d'un grand nombre de logs, la comparaison de popularité.

### C.29. Vendredi 24 avril

J'ai rédigé le cahier des charges, puis ai eu une réunion avec Nastaran. Nous avons discuté du cahier des charges, des améliorations à y apporter. Ensuite, nous avons un peu regarder l'état du rapport actuel. Il faut entre autre pense a y ajouter les références (tableaux, image, bibliographie). Nous avons ensuite convenu de fixer une réunion avec Jonathan et Pascal afin de leur présenter les résultats de l'évaluation. J'ai donc fixé cette réunion à jeudi 30 avril.

### C.30. Mercredi 29 avril

J'ai modifié le rapport pour y ajouter les références sur les tableaux et sur les figures. J'ai également amélioré les annexes avec la table des tableaux et la table des figures. J'ai également commencé la conclusion finale de l'évaluation, en rédigeant le paragraphe concernant l'analyse théorique.