RÉPUBLIQUE DU BÉNIN
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR
ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITÉ D'ABOMEY-CALAVI

INSTITUT DE FORMATION ET DE RECHERCHE EN INFORMATIQUE

BP 526 Cotonou Tel : +229 21 14 19 88
https://www.ifri-uac.bj Courriel : contact@ifri.uac.bj

MÉMOIRE

pour l'obtention du

Diplôme de Licence en Informatique

Option : Sécurité Informatique

Présenté par :
Isségnon Jeanne **IDOLEKE**

Gestion centralisée des logs avec EFK(Elasticsearch, Fluentd, Kibana)

Sous la supervision :
Éric **ATTOU**
Administrateur réseau et système

Membres du jury :

Carlos ASSOGBA	Maître Assistant-CAMES	UCAO-UUC	Président
Victor OYETOLA	Ingénieur	IFRI-UAC	Examinateur
Eric ATTOU	Ingénieur	IFRI-UAC	Rapporteur

Année Académique : **2022-2023**

Sommaire

Dédicace ii

Remerciements iii

Résumé iv

Abstract v

Table des figures vi

Liste des tableaux vii

Liste des acronymes viii

Glossaire 1

Introduction 2

1 Revue de littérature 4

2 Présentation de EFK stack 12

3 La mise en place de EFK stack 18

Conclusion 38

Webographie 39

Table des matières 41

Dédicace

Je dédie ce travail,
À *ma mère*, **TOSSOU Philomène**
À *mon père*, **IDOLEKE Etienne**
À *mon conjoint*, **DAKIN Judicaël**
À tous ceux qui, de près ou de loin, ont contribué à faire de moi ce que je suis.

Remerciements

Je suis profondément reconnaissante envers l'Éternel Dieu, pour sa protection, l'inspiration et la force qu'il me donne tous les jours et qui nous a permis d'arriver là.
La réalisation de ce mémoire a été une expérience cruciale pour moi et cela n'aurait pas été possible sans le soutien et les conseils venant de nombreuses personnes. Pour cela, j'adresse mes sincères remerciements :

- À ma mère TOSSOU Philomène, qui s'est toujours battue pour mon bien-être et continue de le faire jusqu'aujourd'hui, merci maman ;
- À mon père IDOLEKE Etienne qui m'a toujours soutenu de loin et de près ;
- Au professeur Eugène C. EZIN, Directeur de notre Institut, pour ses conseils et orientations durant notre formation ;
- À mon encadreur, M. Éric ATTOU pour sa disponibilité et son sens de partage de connaissance ;
- Au Docteur M. C. Gaston G. EDAH, Directeur Adjoint de notre Institut pour ses précieux conseils et sa disponibilité à nous accompagner durant la formation ;
- Au corps professoral de l'Institut de Formation et de Recherche en Informatique de l'Université d'Abomey-Calavi (IFRI-UAC), qui œuvre pour fournir les ressources nécessaires à la réussite de notre étude universitaire ;
- À la coordination du programme de bourses de la Fondation MasterCard à l'Université d'Abomey-Calavi (MCFSP-UAC) ;
- À monsieur Boris KOUMONDJI notre maître de stage et directeur de FEDAPAY SAS pour nous avoir accordé un cadre adéquat pour le stage de fin de formation et qui nous ont également accompagnés dans la réalisation de notre travail ;
- À monsieur Hored SOSSOU pour son temps, son assistance et ses conseils ;
- Pour finir, nous remercions les membres du jury qui ont accepté d'évaluer notre projet. Nous leur présentons toute notre gratitude et nos profonds respects.

Que tous ceux qui nous ont aidé d'une manière ou d'une autre, spécialement l'entreprise Fedapay SAS, trouvent ici l'expression de nos sentiments, les meilleurs.

Résumé

Les entreprises dépendent de plus en plus des systèmes informatiques pour leurs opérations quotidiennes. Les réseaux informatiques, les serveurs, les applications en ligne jouent un rôle critique dans la prestation de services aux clients, la gestion des données et les processus opérationnels. Ces atouts pour l'entreprise méritent une attention particulière : d'où l'importance de la gestion des logs. Dans ce travail de fin d'études, nous nous intéressons à la mise en place d'une infrastructure de gestion centralisée des logs avec EFK stack au sein de l'entreprise Fedapay SAS. Au prime abord, nous avons fait une étude comparative de certains logiciels de gestion des logs existants. Ensuite, nous avons porté notre choix sur EFK stack tout en ressortant ses caractéristiques qui répondaient aux attentes et aux réalités de l'entreprise et puis nous avons procédé au déploiement et aux tests de fonctionnalités.

Mots clés : Réseaux informatiques, serveurs, applications, logs, EFK stack.

Abstract

Businesses are increasingly dependent on IT systems for their daily operations. Computer networks, servers, and online applications play a critical role in customer service delivery, data management and operational processes. These assets for the company deserve particular attention, hence the importance of log management. In this end-of-study work, we are interested in the implementation of a centralized log management infrastructure with the EFK stack within the company Fedapay SAS. At first glance, we did a comparative study of some existing log management software. Then, we chose the EFK stack while highlighting its characteristics, which met the expectations and realities of the company, and then we proceeded with the deployment and functionality testing.

Key words: Computer networks, servers, applications, logs, EFK stack.

Table des figures

2.1 Architecture de Fluentd [14] . . . 13
2.2 Fluentd plugin architecture [16] . . . 14
2.3 Tableau de bord de kibana [19] . . . 15
2.4 Architecture du système informatique de FEDAPAY SAS. . . . 17

3.1 Architecture virtuelle de la solution . . . 19
3.2 État de docker . . . 20
3.3 Vérification de docker-compose . . . 20
3.4 Configuration de base EFK stack . . . 21
3.5 Dockerfile de Fluentd . . . 22
3.6 Fichier de configuration de base d'Elasticsearch. . . . 22
3.7 Fichier de configuration de base de Kibana . . . 22
3.8 Vérification des conteneurs . . . 22
3.9 Vérification de Kibana . . . 23
3.10 Configuration finale de EFK Stack . . . 24
3.11 Configuration de la sécurité sur Elasticsearch . . . 24
3.12 Configuration finale d'Elasticsearch . . . 25
3.13 Configuration final de Kibana . . . 26
3.14 Fichier de configuration de Fluentd pour l'envoi des données à Elasticsearch . . . 27
3.15 Page de connexion de Kibana . . . 28
3.16 Etat de Td-agent sur centos . . . 28
3.17 Etat de Td-agent sur ubuntu . . . 29
3.18 Configuration de Td-agent sur centos . . . 30
3.19 Configuration de Td-agent sur les machines ubuntu . . . 31
3.20 Création d'index . . . 31
3.21 Filtrage des logs . . . 32
3.22 Exemple de logs sous forme de table . . . 32
3.23 Exemple de logs sous format JSON . . . 33
3.24 Recherche des authentifications échouées du 15/08/2023 . . . 33
3.25 Visualisation des logs Apache . . . 34
3.26 Logs liés à notre Application web . . . 34
3.27 Logs d'erreur de connexion à la base de donnée . . . 35
3.28 PHP erreur . . . 35
3.29 Visionnage des logs avec le tag efkclient2 . . . 36
3.30 Visionnage des logs avec le tag efkclient3 . . . 36
3.31 Dashboard crée sur Kibana . . . 37

Liste des tableaux

1.1 Points forts et limites des solutions de gestion de logs open sources étudiées 9
1.2 Différent entre Fluentd et Logstash [10] 10

Liste des acronymes

API :

Application Programming Interface 8

CNCF :

Cloud Native Computing Foundation 10, 14

CSV :

Comma Separated Values 10

ELK :

Elasticsearch Logstash Kibana 8

GELF :

Graylog Extended Lenght Format 8

HDFS :

Hadoop Distributed File System 8

HIPAA :

Health Insurance Portability and Accountability Act 4

HTTP :

Hypertext Transfer Protocol 25

HTTPS :

Hypertext Transfer Protocol Secure 27

JSON :

JavaScript Object Notation 10, 13, 15, 31

NoSQL :

Not only Structured Query Language 12

PCI DSS :

Payment Card Industry Data Security Standard 4

REST :

Representational State Transfer 8, 12

RGPD :

Règlement Général sur la Protection des Données 4

SGBD :

Système de Gestion de Base de Données 2

SI :

Système Informatique 10

SIEM :

Security Information and Event Management 7, 9, 41

SNMP :

Simple Network Management Protocol 5, 41

SSH :

Secur Shell 16

SSL :

Secure Sockets Layer 23

TCP :

Transmission Control Protocol 8

TLS :

Transport Layer Security 23

UDP :

User Datagram Protocol 5, 8

VPN :

Virtual Private Network 16

YAML :

Yet Another Markup Language 20, 38

Glossaire

Elastalert : est un framework simple pour alerter sur les anomalies, les pics ou d'autres modèles d'intérêt à partir de données dans Elasticsearch. 16

HBase : HBase est un système de gestion de bases de données distribué, NoSQL et orienté colonnes, développé au-dessus du système de fichier HDFS. 8

Open Source : S'applique aux logiciels dont la licence respecte des critères précisément établis par l'Open Source Initiative, c'est-à-dire les possibilités de libre redistribution, d'accès au code source et de création de travaux dérivés. 7–9, 42

Pipeline : est un ensemble de processus et d'outils utilisés pour collecter des données brutes à partir de plusieurs sources, les analyser et présenter les résultats dans un format compréhensible 8, 12

Redis : Remote Dictionary Server (Serveur de dictionnaire à distance), est un système de stockage de données clé-valeur en mémoire, open source et rapide. 10

Rsyslog : est un programme open source qui permet de transférer les messages de journalisation sur un réseau IP pour les systèmes UNIX et de type Unix. 30, 31

X-Pack : est une extension Elastic Stack qui fournit des fonctionnalités de sécurité, d'alerte, de surveillance, de création de rapports, d'apprentissage automatique et de nombreuses autres fonctionnalités. 23, 24

Introduction Générale

Les entreprises déploient de plus en plus de solutions informatiques qu'il faut savoir administrer, pour cela les logs s'avèrent être une source précieuse et incontournable pour la détection et la prévention d'éventuels risques. La gestion et l'exploitation de ce type d'informations s'avèrent délicates en pratique en raison de la grande quantité et de l'hétérogénéité des logs générés par les différents systèmes, routeurs, pare-feu, SGBD, serveur web, etc. Il devient alors indispensable de centraliser les logs dans un emplacement central pour pouvoir les gérer et les exploiter efficacement.

1. Problématique

Lorsqu'une entreprise gère un environnement informatique complexe comprenant de nombreux systèmes et applications, il est essentiel d'avoir une visibilité sur les événements qui se produisent dans ces systèmes et de détecter rapidement toute anomalie ou activité suspecte. Dans le cas de FEDAPAY SAS, une entreprise fournissant des services financiers, la gestion des logs revêt d'une importance particulière. Fedapay doit surveiller en temps réel les transactions financières, les activités des utilisateurs, les erreurs systèmes, les problèmes de sécurité et bien d'autres événements pour assurer la disponibilité, la performance et la sécurité de ses services. Ceci nous amène à nous poser une question importante qui mérite une attention particulière : comment gérer efficacement les logs provenant de multiples sources, tout en garantissant une bonne performance et une haute disponibilité du système de gestion des logs ? Notre problématique est donc d'identifier un outil de gestion des logs abordable et qui répondra aux attentes de l'entreprise.

2. Contexte

FEDAPAY SAS est une entreprise FinTech qui fournit des solutions de paiement en ligne pour les entreprises en Afrique, en utilisant une plateforme de paiement appelée Fedapay qui permet aux entreprises de recevoir des paiements à partir de diverses sources, telles que les mobiles money, les cartes bancaires, l'USSD et les portefeuilles électroniques. Elle dispose de plusieurs serveurs qui traitent un grand nombre de transactions chaque jour. Ces transactions produisent une quantité importante de logs qui doivent être collectés et stockés pour une analyse ultérieure. Actuellement, FEDAPAY SAS utilise Elastic stack appelé encore ELK stack pour Elasticsearch, Logstash et Kibana ainsi que les beats pour la gestion des logs qui présentent certaines limites. Logstash par exemple est plus gourmand en ressources mémoire. Fluentd traite plus efficacement et rapidement les logs et intègre nativement la collecte et le traitement de logs structurés ou non structurés. En termes de routage d'évènements,

Fluentd est meilleur parce que Fluentd utilise des tags pour router les événements alors que Logstash utilise des conditions. Le défi majeur de Fedapay en matière de gestion des logs est de collecter et d'acheminer efficacement ces logs vers un système centralisé en utilisant moins de ressources et doit être en mesure de détecter rapidement toute anomalie ou activité suspecte. Les logs peuvent provenir de plusieurs sources telles que les serveurs web, les bases de données, les applications mobiles, etc. La collecte, l'agrégation et l'acheminement de ces logs vers un système centralisé pour une analyse ultérieure est une tâche complexe qui nécessite un outil de collecte de logs efficace et flexible. C'est ce qui justifie le choix du thème : La Gestion centralisée des logs avec EFK stack.

3. Objectifs

L'objectif global de ce projet est de choisir parmi la panoplie de solution de gestion des logs, celui qui sera convenable pour l'entreprise objet de notre projet et qui permet d'avoir une visibilité sur les événements qui se produisent dans ces systèmes et de détecter rapidement toute anomalie ou activité suspecte Les objectifs spécifiques de ce projet consistent à :

- s'approprier du concept de la gestion des logs ;
- faire le choix d'une solution de gestion des logs qui répond à nos exigences ;
- mettre en place l'architecture ;
- installer et configurer la solution ;
- connecter la solution avec les systèmes à surveiller ;
- effectuer des tests après la mise en place de notre environnement ;

4. Organisation du mémoire

Le présent mémoire est organisé en trois chapitres. Le premier est une revue littéraire consacrée à l'état de l'art. Le deuxième chapitre aborde la présentation de EFK stack. Enfin, le chapitre 3 abordera la mise en place proprement dite de EFK stack et les tests de fonctionnalité du stack.

Chapitre 1

Revue de littérature

Introduction

Dans le monde des entreprises d'aujourd'hui, la gestion efficace des données et des informations est essentielle pour assurer le bon fonctionnement des opérations, la sécurité des systèmes et la prise de décisions éclairées. Parmi les différents types de données à prendre en compte, les logs jouent un rôle crucial dans la gestion des systèmes informatiques et la sécurité des données. Dans ce chapitre, il s'agira de faire une revue de littérature autour du concept de gestion des logs et de faire le choix d'un logiciel.

1.1 La gestion des logs

1.1.1 Définition

La gestion des journaux, appelée encore log management, est définie comme un processus ou flux de travail complet de gestion des données de journal capturées en grands volumes à partir de diverses sources et emplacements de stockage. L'idée derrière cette pratique est de simplifier la maintenance des applications ou des sites Web en utilisant les informations techniques générées par ceux-ci sous la forme de messages de log. [1]

1.1.2 Objectifs

L'objectif de la gestion des logs est de s'assurer que les logs se trouvent dans un endroit sûr et accessible rapidement. Elle permet d'avoir une visibilité complète et précise sur les activités et les événements qui se produisent au sein d'un système informatique. Voici quatre objectifs spécifiques de la gestion des logs :

- Exigences de conformité et d'audit : de nombreux secteurs et cadres réglementaires exigent des organisations qu'elles conservent des enregistrements de journaux complets pour démontrer leur conformité aux normes de sécurité et de confidentialité telles que la norme (PCI DSS), la loi HIPAA ou le règlement général sur la protection des données (RGPD).

- La centralisation des journaux : l'un des principaux objectifs de la gestion des journaux est de collecter les journaux à partir de diverses sources au sein de l'environnement informatique d'une organisation, y compris les serveurs, les applications, les périphériques réseau et les systèmes de sécurité. Ces journaux sont ensuite stockés de manière centralisée dans un emplacement sécurisé et accessible.
- Dépannage et analyse des causes profondes : les journaux contiennent des informations précieuses sur le comportement, les erreurs et les événements du système. Une gestion efficace des journaux permet de résoudre les problèmes et d'effectuer une analyse des causes profondes pour identifier les causes sous-jacentes des problèmes.
- Surveillance et optimisation des performances : les journaux fournissent des informations sur les performances du système, l'utilisation des ressources et le comportement des applications. Ainsi, elle permet d'optimiser l'allocation des ressources et d'améliorer l'efficacité des systèmes informatiques.

1.1.3 Les protocoles de gestion des logs

Pour rendre possible la communication entre les agents de gestion des logs et les équipements, il est obligatoire qu'ils utilisent le même protocole. En effet, la gestion des logs est rendue possible grâce à plusieurs protocoles tels que : Syslog, SNMP, et Windows Event Log.

1.1.3.1 Syslog

Syslog est un protocole définissant un service de journaux d'événements d'un système informatique. C'est aussi le nom du format qui permet ces échanges. En tant que protocole, Syslog se compose d'une partie client et d'une partie serveur. La partie client émet les informations sur le réseau, via le port UDP. Les serveurs collectent l'information et se chargent de créer les journaux. L'intérêt de Syslog est donc de centraliser les journaux d'événements, permettant de repérer plus rapidement et efficacement les défaillances d'ordinateurs présents sur un réseau. [3]

1.1.3.2 SNMP

SNMP est un protocole de communication simple qui permet aux administrateurs de gérer les équipements du réseau, superviser et de diagnostiquer les problèmes de réseau, matériels à distance et d'agir sur les paramètres d'une façon indépendante. Avec le protocole SNMP, le système de gestion de réseau est basé sur trois éléments principaux [4] :

- un superviseur : est la console qui permet à l'administrateur réseau d'exécuter des requêtes de gestion SNMP. Il est client SNMP qui obtient des informations d'un serveur.
- des agents et des nœuds (ou nodes) : sont des entités qui se trouvent au niveau de chaque interface, connectant au réseau l'équipement géré (nœud) et permettant de récupérer des informations sur différents objets. Les équipements gérés tels que des routeurs ou des commutateurs, mais aussi des serveurs et des matériels spécifiques des centres de données remplissent le rôle de serveur SNMP.

1.1.3.3 Windows Event Log

Windows Event Log protocole est spécifique aux systèmes d'exploitation Windows. Il permet la collecte et la gestion des logs générés par les événements système, de sécurité et d'application [5]. Les logs sont stockés dans la base de données des journaux d'événements de Windows et peuvent être consultés à l'aide de l'outil Event.

1.1.4 Les avantages de gestion des logs

Les principaux avantages de la gestion des journaux [6] :

1.1.4.1 Efficacité opérationnelle améliorée

À mesure que les entreprises évoluent, la taille et la complexité de leurs systèmes, applications et infrastructures de support augmentent également et posent divers problèmes de surveillance. Grâce à la gestion centralisée et automatisée des journaux, les entreprises peuvent surmonter les défis liés à l'inspection manuelle de fichiers journaux disparates de manière plus rapide et plus économique. De plus, l'agrégation centralisée des journaux garantit la normalisation des données, simplifie l'analyse des données et accélère l'ensemble du processus de gestion et de surveillance des fichiers journaux.

1.1.4.2 Utilisation efficace des ressources

La gestion des journaux aide à identifier les problèmes de performances avant qu'ils ne deviennent des goulots d'étranglement.

1.1.4.3 Dépannage proactif

La surveillance des journaux offre une meilleure vision des applications et des processus sur l'ensemble des systèmes et aide à détecter les problèmes pour un dépannage efficace. Les capacités d'analyse et de visualisation des données garantissent une recherche personnalisée facile, une analyse et une visualisation rapides d'une grande quantité de données de journal pour suivre et corréler les événements afin de déterminer la source du problème en temps réel.

1.1.4.4 Meilleure expérience de l'utilisateur final

La gestion des journaux garantit également que les applications sont exemptes de problèmes tels que les temps de réponse longs ou l'absence d'erreurs de réponse. Une gestion efficace des journaux aide les développeurs à surveiller les demandes à plusieurs niveaux pour détecter les failles ayant un impact sur l'expérience globale de l'utilisateur final.

1.1.4.5 Une plus grande sécurité

Les journaux sont l'une des principales sources de données pour détecter les failles de sécurité et prévoir les menaces. Avec un système efficace de gestion des journaux en place, les équipes de sécurité informatique peuvent surveiller les journaux d'accès au réseau, les journaux système ou les journaux d'authentification pour détecter les accès suspects aux systèmes et aux données de l'organisation.

1.1.5 Similitudes entre SIEM et la gestion des journaux

Le SIEM et la gestion des journaux partagent de nombreuses similitudes [7] :

- Ils utilisent tous les deux les journaux d'une manière ou d'une autre.
- Ils offrent tous deux une visibilité sur ce qui se passe dans un environnement d'application.
- Ils peuvent à la fois aider à trouver et à résoudre les problèmes en temps réel.
- Ils peuvent tous deux être utilisés pour faciliter la recherche ou l'audit de problèmes survenus dans le passé.

1.1.6 Différence entre SIEM et la gestion des logs

Dans les aspects les plus importants, les outils SIEM et de gestion des journaux sont des types de solutions fondamentalement différents, pour plusieurs raisons.[7]

- La première est que SIEM se concentre sur la recherche et la résolution des problèmes de sécurité. La gestion des journaux est également une partie importante des workflows de sécurité, car les données des journaux sont souvent essentielles pour résoudre les problèmes de sécurité. Cependant, la gestion des journaux va au-delà de la simple sécurité. Il est tout aussi important pour la gestion des performances des applications, le maintien de la disponibilité, la planification des capacités, etc.
- Le SIEM implique également un plus large éventail de sources de données. La gestion des journaux se concentre uniquement sur les journaux ; d'autres types de données qui peuvent être disponibles à partir d'un environnement, comme les métriques d'application, sont analysés et gérés d'autres manières.
- Un différenciateur final est les délais associés au SIEM et à la gestion des journaux. Bien que les données et les analyses d'un outil SIEM puissent parfois être utiles pour rechercher des événements passés, les plateformes SIEM se concentrent principalement sur l'analyse des modèles de sécurité en temps réel. Pendant ce temps, les outils de gestion des journaux offrent des informations en temps réel grâce au suivi et à l'analyse rapide des journaux, ainsi qu'à la collecte, l'analyse et la gestion des données des journaux après leur production.

1.1.7 Quelques solutions de gestion des logs

De nos jours, il existe plusieurs variétés d'outils de gestion des logs. Certains de ces outils sont Open Source et d'autres sont propriétaires. Faisons un tour d'horizon sur la particularité de quelques-uns d'entre eux pour la réussite de notre projet.

1.1.7.1 Les solutions payantes

Il existe une grande variété d'outils propriétaires. Parmi ces outils, nous pouvons énumérer : Splunk, SolarWinds Log & Event Manager, Logz.io, etc.

a. Avantages généraux des solutions propriétaires

   – Les solutions sont plus globales ;

- Les périmètres techniques et fonctionnels sont très étendus ;
- Présence d'un support réactif ;

b. Inconvénients

- les coûts d'acquisition et de support sont parfois très élevés ;
- Les périmètres techniques et fonctionnels sont très étendus ;
- Incompatibilité entre les fournisseurs (plusieurs produits différents) ;
- Développements additionnels restreints et très coûteux.

1.1.7.2 Les solutions Open Source

Ils sont utilisés par la plupart des PME et aussi par des organisations de grande taille. Voici la liste des plus fréquemment utilisés : SigNoz, Graylog, Fluentd, Syslog, Logwatch, Apache Flume, GoAccess, LOGalyze, Elastic Stack, Logit.io, Logary, etc. Mettons le zoom sur queslques uns d'entre eux.

1. Graylog
   Graylog est une plateforme de gestion de logs qui permet de collecter, d'analyser et de visualiser les logs provenant de diverses sources. Il offre des fonctionnalités de recherche, de corrélation d'événements, de surveillance en temps réel et de création de rapports.
   Il est découpé en 2 parties principales [**?**] :

   - Graylog-server : C'est une application java qui accepte les messages sur différents protocoles : UDP , TCP... Une API REST est également intégrée à l'outil et est notamment utilisée par la partie web-interface
   - Graylog-web : C'est l'interface web qui nous permet de gérer des utilisateurs, des streams et des Dashboard

   Il prend en charge plusieurs entrées et sorties de données, il peut collecter des données à partir de diverses sources telles que Syslog, GELF , des fichiers journaux et le journal des événements Windows, et il envoie des données à une base de données telles que Elasticsearch, Mongo-DB, etc.

2. Apache Flume
   Il a été développé à l'origine par Cloudera et a été accepté en tant que projet Apache Open Source en 2011. Il est écrit en Java et repose sur une architecture événementielle basée sur des flux de données. Son architecture est constituée de nombreux agents. Il faudra donc définir les agents distribués qui se chargent de récupérer les données en provenance d'une multitude de sources, consolider les logs pour les écrire dans un référentiel, centralisé (comme un cluster HDFS ou une base HBase).

3. Elastic stack
   Elastic stack appelé encore ELK stack est l'acronyme de trois projets Open Source : Elasticsearch, Logstash et Kibana. Elasticsearch est un moteur de recherche et d'analyse. Logstash est un côté serveur, destiné au traitement des données. Sa mission ? Ingérer simultanément des données provenant de multitude de sources, puis les transformer et les envoyer vers un système de stockage comme Elasticsearch. Kibana permet aux utilisateurs de visualiser les données avec des tableaux et des graphiques dans Elasticsearch. Un quatrième produit, Beat qui est une forme légère de Logstash, a ensuite été ajouté à la pile. [9]

4. Fluentd
   Fluentd est un collecteur de logs open source qui permet l'ingestion et l'envoi des logs vers différents systèmes de stockage et d'analyse. Il est utilisé pour collecter des logs à partir de différents types de sources et les acheminer vers des destinations telles que Elasticsearch, MongoDB, etc.

1.1.8 Étude Comparative de quelques solutions Open Source

Solutions	Points forts	Limites
Graylog	- Centralisation des journaux - Interface web - Capacité d'indexation, de recherche et d'analyse de données - Alerte par mail et gratuit - Communauté active	- Les capacités de visualisation sont limitées - Ne peut pas lire les fichiers Syslog, on doit donc soumettre les messages directement à Graylog - Le tableau de bord n'est pas convivial en termes de gestion
Apache Flume	- Centralisation des journaux - Une très bonne documentation - Ingestion de données de sources multiples dans le système Hadoop - Livraison fiable des messages - Un débit élevé avec une latence plus faible - La tolérance à la panne.	- Topologie complexe - Problèmes d'évolutivité et de fiabilité - Duplication : Apache Flume ne garantit pas que les messages qui parviennent sont uniques à 100 %
Elastic Stack	- Centralisation des journaux - Solution robuste - Variété de plugins - Indexation et stockage de données dans Elasticsearch - Incroyables visualisations avec Kibana - Implique des solutions de SIEM	- Pas de tableaux de bord de journalisation dans Kibana par défaut - L'authentification et les alertes sont des fonctionnalités premium qui doivent être achetées - Activation de premium pour plus de fonctionnalité - Logstash consomme assez de mémoire - Configuration complexe
Fluentd	- Open source et gratuit - Haute performance - Il fournit un système de mise en mémoire tampon - Plus de variété de plugins - Économe en thème de mémoire	- Configuration complexe - Fluentd est un collecteur de logs donc nécessite une base de donnée comme Elasticsearch pour le stockage et un outil de visualisation de données comme Kibana

TABLE 1.1 : Points forts et limites des solutions de gestion de logs open sources étudiées

1.1.8.1 Étude comparative entre Fluentd et Logstash

Nous mettons l'accent sur la comparaison entre Fluentd (notre choix) et Logstash, tous compatibles avec Elasticsearch et Kibana. Un tableau pour donner un bref aperçu des principales différences.

	Logstash	Fluentd
Entreprise d'approbation	Il fait partie de la pile ELK populaire gérée par Elastic lui-même.	Construit et géré par Treasure Data et fait partie de la CNCF.
Langue	Écrit en JRuby, il nécessite donc un runtime Java sur la machine hôte.	Écrit en CRuby donc pas besoin de dépendances d'exécution Java.
Plateforme	Fonctionne sur Windows et Linux à la fois. Naturellement multiplateforme.	Il n'était pas disponible sur Windows jusqu'en 2015. Maintenant, il fonctionne également sur Windows.
Écosystème et plugins	Il a environ 200 plugins	Il a environ 700 plugins.
Transport	Il ne prend pas en charge un système de mise en mémoire tampon intégré. Uniquement limité à une file d'attente en mémoire pouvant contenir jusqu'à 20 événements (fixe) et s'appuie principalement sur des files d'attente externes comme Redis pour la cohérence.	Il fournit un système de mise en mémoire tampon intégré qui peut être configuré en fonction des besoins. Il peut s'agir d'un système en mémoire ou sur disque.
Routage d'événement	Les événements sont acheminés en fonction des conditions if-else. Cela peut être un peu fastidieux cependant.	Les événements sont acheminés en fonction des balises.
Prise en charge de Docker	Un agent supplémentaire (filebeat) est requis sur le conteneur pour pousser les journaux vers Logstash	Docker fournit un pilote de journalisation Fluentd intégré. Le pilote de journalisation envoie les journaux de conteneur au collecteur Fluentd sous forme de données de journal structurées.
Analyse du journal	Logstash utilise des plugins	Fluentd utilise des analyseurs intégrés standard (JSON,CSV, regex, etc.)

TABLE 1.2 : Différent entre Fluentd et Logstash [10]

De l'étude comparative des solutions existantes, nous constatons que la première des limites majeures qu'elles ont est la consommation des ressources du SI. Un autre problème qui se pose avec ces solutions est celui du "single point of failure", donc un risque d'indisponibilité du serveur de gestion des logs. Pour résoudre ce dernier, il faut une redondance du ou des serveurs, ce qui demande des coûts supplémentaires de la part de l'entreprise. Notre objectif ici est de limiter la consommation de ressources, les dépenses informatiques de l'entreprise et de faciliter le travail tout en permettant une gestion complète et permanente. C'est la raison pour laquelle nous avons opté pour Fluentd en collaboration avec Elasticsearch et Kibana pour une gestion des logs complète.

1.2 La solution choisie : EFK stack

Dans le monde du log management, il existe toute une panoplie de logiciels qui ont chacun leurs avantages et inconvénients. Cela nous a amené à faire une étude comparative sur les logiciels les plus réputés. En effet, nous avons été séduits par deux solutions qui sont : Elastic stack et Fluentd. Fluentd présente de nombreux avantages au même titre que Logstash, mais son problème de mémoire, de mise en mémoire tampon nous a conduit à opter pour Fluentd, d'où EFK stack.

Conclusion

Il existe une panoplie de logiciels de gestions des logs, tant bien payants que libres. À travers ce chapitre, nous avons parcouru brièvement les solutions de gestion des logs existants et justifié notre choix.

Chapitre 2

Présentation de EFK stack

Introduction

Pour une meilleure mise en place de la solution choisie, il s'avère indispensable de maîtriser le fonctionnement de cette dernière et l'environnement informatique dont il s'agit. Pour ce faire, nous allons faire une présentation du fonctionnement de la pile EFK et de certains de ses composants.

2.1 Présentation de EFK stack

EFK stack est une suite d'outils combinant Elasticsearch, Fluentd et Kibana pour gérer les logs.

2.2 Elasticsearch

Elasticsearch est un moteur de recherche et d'analyse distribué, open source, RESTfull, basé sur Apache Lucene. Depuis sa publication en 2010, Elasticsearch est rapidement devenu le moteur de recherche le plus populaire. Il est couramment utilisé pour l'analyse de journaux, la recherche en texte intégral, les analyses à des fins commerciales [11]. C'est aussi une base de données NoSQL en charge de l'indexation et du stockage des données. Il est le composant principal de EFK stack. Il propose également des requêtes avancées pour effectuer une analyse détaillée et stocke toutes les données de manière centralisée. Il est également utilisé sur de nombreux projets hors EFK stack, car il permet d'exécuter une recherche rapide des documents.

2.3 FluentD

Fluentd est un collecteur de données open source pour une couche de journalisation unifiée. Fluentd vous permet d'unifier la collecte et la consommation de données pour une meilleure utilisation et compréhension des données [12]. Fluentd agit comme un Pipeline pour que les données de journaux circulent des générateurs de journaux (tels qu'un hôte ou une application) vers des destinations de

journaux (telles qu'Elasticsearch). Les sources lui envoient des messages de journal et Fluentd transmet ces messages à une destination en fonction d'un ensemble de règles configurables. Dans le cadre de ce processus, il convertit automatiquement chaque message de son format d'origine en JSON standard [13]. On parle d'unification des journaux.

2.3.1 Architecture de Fluentd

Ce schema illustre l'architecture de Fluentd

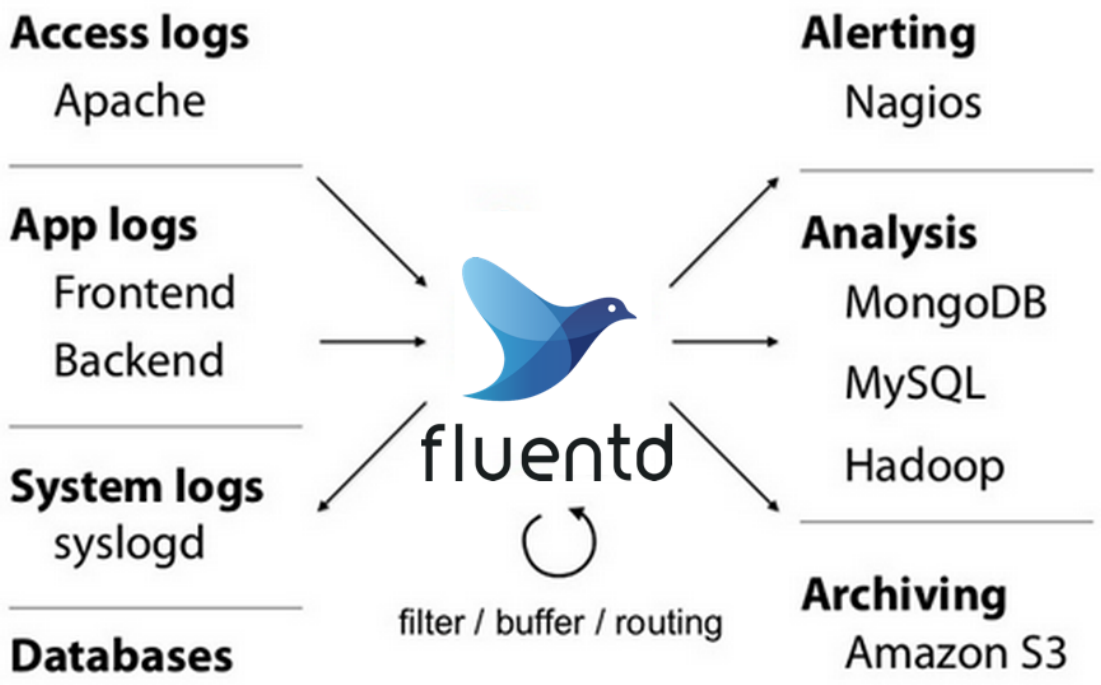


FIGURE 2.1 : Architecture de Fluentd [14]

Il nous faut retenir que Fluentd peut collecter des journaux à partir de plusieurs sources (Apache, les applications, les bases de données, etc) et structurer les données au format JSON. Cela permet un traitement unifié des données de journal, y compris la collecte, le filtrage, la mise en mémoire tampon et la sortie de journaux sur plusieurs sources et destinations.

2.3.2 Plugin

Fluentd a une architecture enfichable composée de plus de 650 plugins, permettant aux fonctionnalités de Fluentd de fonctionner avec différentes sources de données d'entrée et de sortie [15].

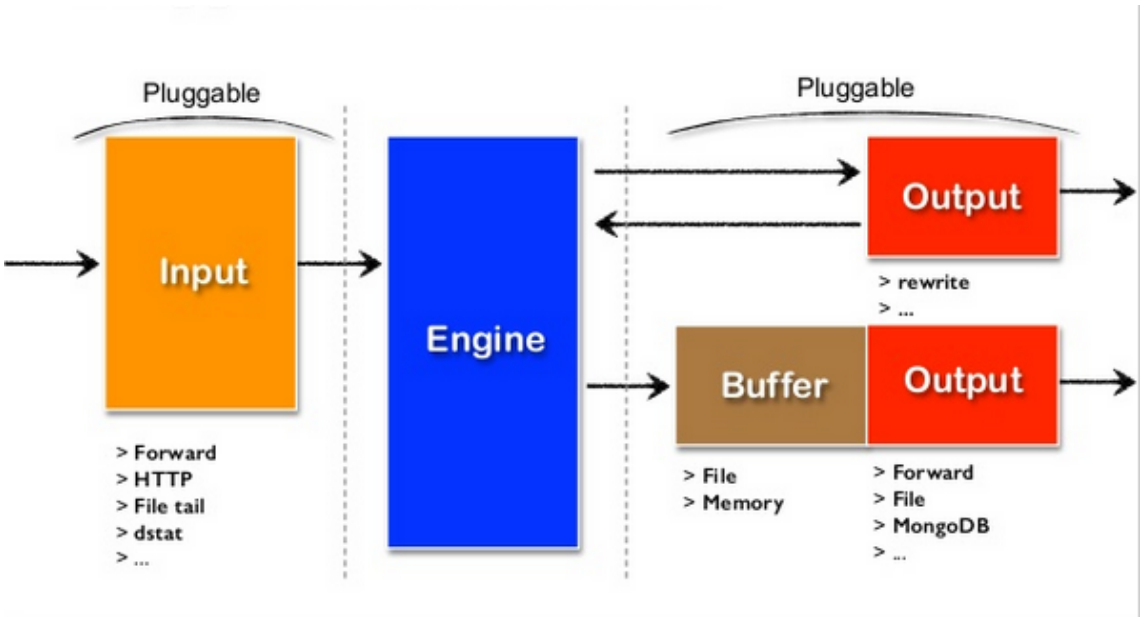


FIGURE 2.2 : Fluentd plugin architecture [16]

Ces plugins sont classés comme :

- Input : utilisée pour la collecte de données à partir de plusieurs sources.
- Analyseur : permet aux utilisateurs d'analyser le format de données personnalisé de la source, que Fluentd peut ensuite comprendre.
- Filter : permet à Fluentd de modifier les journaux entrants. Certains cas d'utilisation incluent enrichir des champs, supprimer/masquer des enregistrements, etc.
- Output : transfère les journaux vers le système de stockage souhaité. Le plugin de sortie peut fonctionner en mode buffered ou non-buffered.
- Buffer : utilisé par les plug-ins de sortie pour stocker temporairement les flux entrants avant de les envoyer dans le système de stockage. Ce plugin rend Fluentd fiable dans le transfert des journaux vers le point de terminaison souhaité.
- Formatter : Utilisé par le plugin Output pour modifier les formats de sortie en fonction des besoins de l'utilisateur.

2.3.3 TD-Agent

TD-Agent, également connu sous le nom de Treasure Agent, est un package de distribution stable de Fluentd, qui est maintenu par Treasure Data et la CNCF [17]. Il est souvent utilisé comme agent de collecte de journaux dans les grandes entreprises. Il joue un rôle important dans l'architecture de EFK stack en assurant la collecte fiable et l'acheminement des journaux vers Fluentd, qui est ensuite responsable de l'envoi des données vers Elasticsearch pour le stockage et la visualisation à l'aide de Kibana.

2.4 Kibana

Kibana est une interface de visualisation de données qui fonctionne au-dessus d'Elasticsearch et qui permet de créer des tableaux de bord et des visualisations à partir des données stockées dans Elasticsearch. Un tableau de bord kibana se présente comme suit :

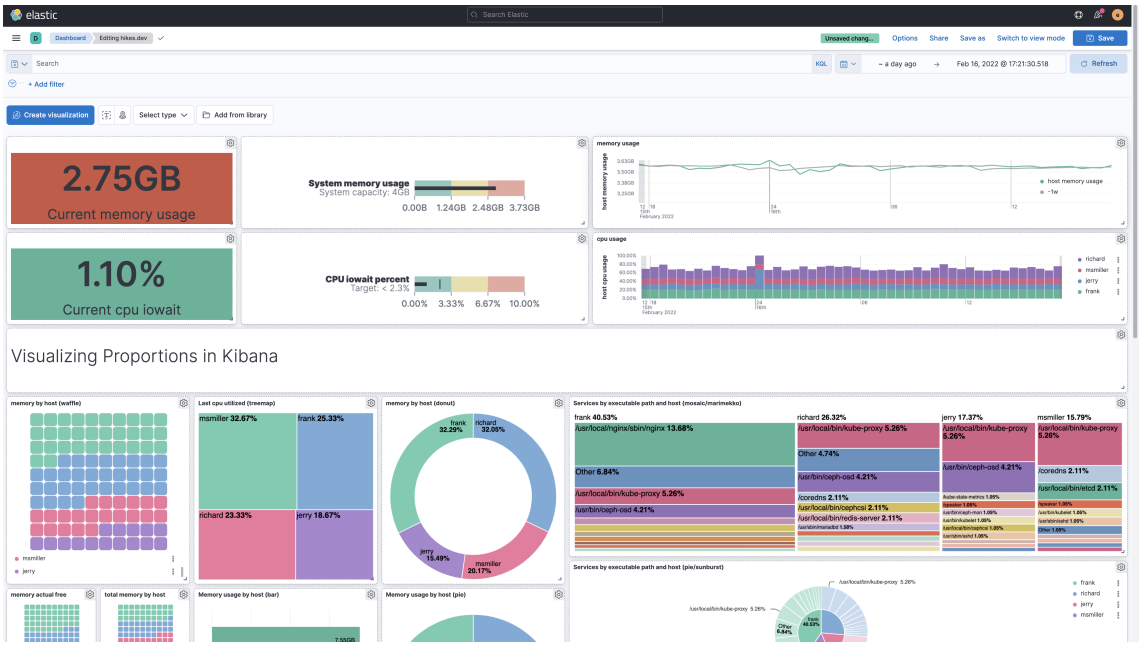


FIGURE 2.3 : Tableau de bord de kibana [19]

2.5 Fonctionnement de EFK stack

Fluentd collecte les journaux de différentes sources, les unifie dans un format JSON commun, puis les envoie à Elasticsearch pour le stockage et l'indexation. Elasticsearch crée des index pour faciliter la recherche et l'analyse des journaux, Kibana récupérera les journaux d'Elasticsearch et permet de visualiser et d'analyser les journaux de manière interactive, offrant ainsi une meilleure compréhension et une meilleure utilisation des données de journal[20].

2.6 Gestion des alertes

La gestion des alertes dans EFK stack peut être réalisée à partir de Kibana, Fluentd, ou en utilisant divers outils d'intégrations.

- Kibana : Kibana propose une option d'alerte appelée "Alerting" qui permet de créer des règles d'alerte basées sur des critères définis et recevoir des notifications par mail ou via des webhook lorsque ces conditions sont remplies. Cependant, cela nécessite une licence payante pour le niveau d'abonnement adéquat.

- FluentD : Une autre option consiste à utiliser Fluentd avec un plugin d'alerte tiers. Par exemple, le plugin Fluentd "fluent-plugin-alert" permet de surveiller les logs collectés par Fluentd et de déclencher des alertes en fonction de règles prédéfinies. On peut configurer des règles pour filtrer les logs et spécifier les actions à entreprendre lorsque des conditions particulières sont remplies, comme envoyer des notifications par e-mail, appeler des API externes, ou exécuter des scripts personnalisés. Ce plugin offre une flexibilité pour personnaliser les alertes selon les besoins spécifiques.
- Intégration : EFK stack peut-être intégrer avec d'autres outils d'alerte et de surveillance tiers tels, Elastalert. Ces outils peuvent être configurés pour collecter des métriques Elasticsearch, analyser les données et déclencher des alertes en fonction de seuils ou de conditions prédéfinis.

2.7 Les méthodes d'installation de EFK stack

EFK peut-être installée à l'aide d'une variété de méthodes et sur un large éventail de systèmes d'exploitation et d'environnements différents. On peut installer EFK localement, sur le cloud, à l'aide de Docker. Elle peut être également installée à l'aide de gestionnaire de paquets ou manuellement depuis les binaires officiels.

2.8 Description du système informatique

Le réseau de FEDAPAY SAS dispose d'une connexion Internet fournie par l'opérateur SBIN, d'un routeur, d'un pare-feu permettant de filtrer les paquets depuis internet. Tous les ordinateurs, serveur et imprimantes ont accès au réseau par le Wi-Fi.
Les serveurs de FEDAPAY SAS disposent du système d'exploitation Ubuntu et ils sont tous logés dans le cloud. Des règles de connexions (pare-feu) sont mis en place afin que toutes connexions depuis une machine extérieure, via SSH ou autres soit possible à travers un VPN.

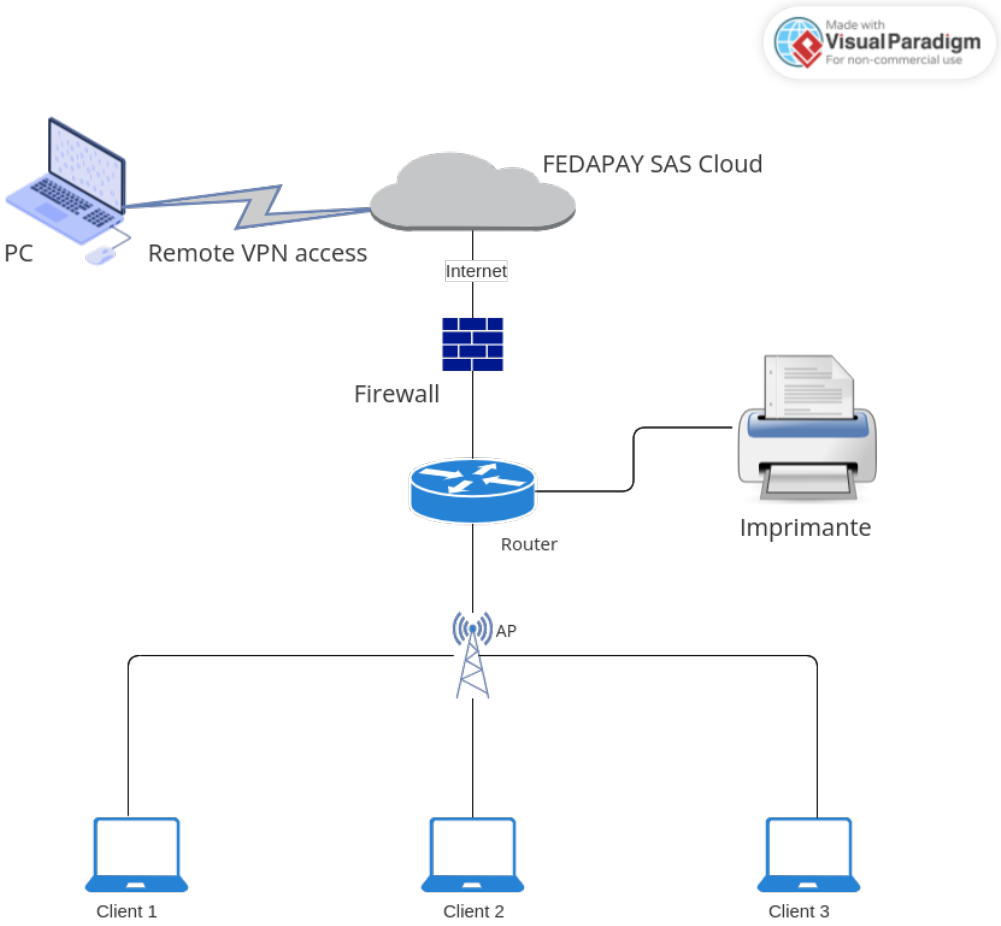


FIGURE 2.4 : Architecture du système informatique de FEDAPAY SAS.

Conclusion

Dans ce chapitre, nous avons présenté EFK stack, ces principaux composants, ses différentes méthodes d'installation, son fonctionnement ainsi que le système d'information de l'entreprise, objet de notre projet.

Chapitre 3

La mise en place de EFK stack

Introduction

Dans ce chapitre, il est question de l'implémentation de la solution de gestion des logs EFK stack et la visualisation des résultats obtenus par des tests de vérification. Nous procédons d'abord à l'installation des machines virtuelles dans VMWare, ensuite à l'installation des composants de EFK stack dans des conteneurs dockers, la configuration de la machine cliente puis nous terminons par quelques tests de vérification du fonctionnement de notre outil.

3.1 La virtualisation

La virtualisation consiste, en informatique, à exécuter sur une machine hôte, dans un environnement isolé, des systèmes d'exploitation ou des applications [21]. En effet, la virtualisation nous permettra, dans ce chapitre, d'avoir une architecture et une réaction semblable à l'environnement en entreprise. Nous avons donc opté pour VMWare comme logiciel de virtualisation.

3.1.1 Présentation de l'architecture virtuelle

Compte tenu des ressources de notre machine physique, nous utiliserons les machines virtuelles ci-après :

- Ubuntu-serveur : c'est la machine sur laquelle sera installé EFK stack. Elle possède les caractéristiques ci-après : Mémoire RAM : 3 GB ; Processeurs : 2 ; Hard Disk (SCSI) : 20 GB ; Carte réseaux : 1 (Bridge) ; Système d'exploitation : Ubuntu 22.10 ;
- CentOS : cette machine représente l'une des machines à surveiller sur laquelle tourne une application web en local. Elle est caractérisée par : Mémoire RAM : 1 GB ; Processeurs : 1 ; Hard Disk (SCSI) : 16 GB ; Carte réseaux : 1 (Bridge) ; Système d'exploitation : CentOS 7 ;
- deux machines Ubuntu desktop : Elles sont caractérisées par : Mémoire RAM : 1 GB ; Processeurs : 1 ; Hard Disk (SCSI) : 20 GB ; Carte réseaux : 1 (Bridge) ; Système d'exploitation : Ubuntu

16.04

Dans notre cas, EFK stack sera installée dans des conteneurs sur un système Ubuntu, c'est-à-dire que nous allons déployer Elasticsearch, Kibana, et Fluentd dans des conteneurs à nœud unique.

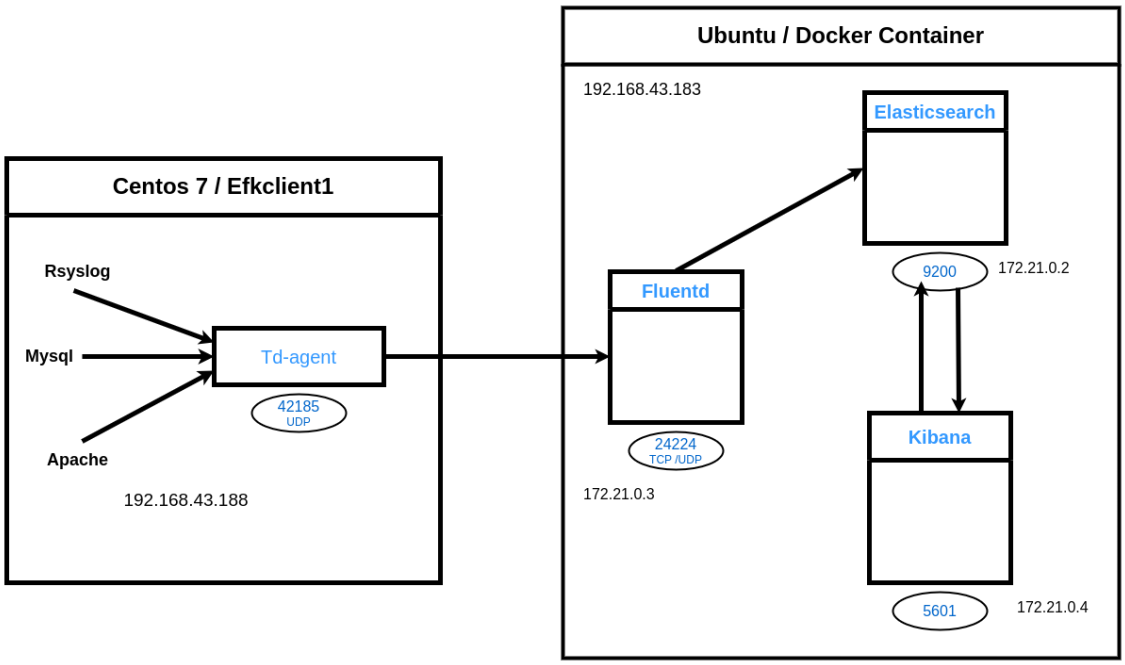


FIGURE 3.1 : Architecture virtuelle de la solution

3.2 Installation de EFK stack

Le processus d'installation se présente de la façon suivante :
Les conditions préalables à remplir pour un bon fonctionnement de notre outil.

3.2.1 Docker

Docker est une plateforme permettant de lancer certaines applications dans des conteneurs logiciels. [22] [23]

- Installation

```
$ sudo apt update
$ sudo apt install apt-transport-https ca-certificates
$ curl gnupg-agent software-properties-common -y
$ echo "deb [arch=$(dpkg --print-architecture) signed-
by=/usr/share/keyrings/docker-archive-keyring.gpg
https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable" | sudo tee
/etc/apt/sources.list.d/docker.list &gt; /dev/null
$ sudo mkdir -p /etc/apt/keyrings
$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg |
sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg
$ sudo apt update
$ sudo apt install docker-ce -y
$ docker --version
```

- Verifier l'état de docker

```
$ sudo systemctl status docker
```

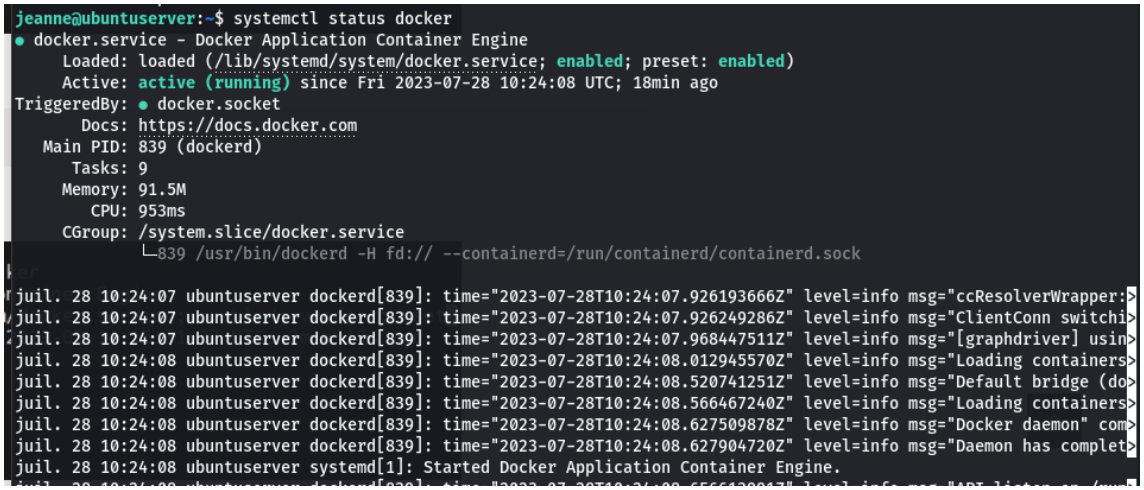


FIGURE 3.2 : État de docker

3.2.2 Docker compose

Docker compose est une fonctionnalité permettant d'orchestrer plusieurs conteneurs qui doivent travailler ensemble. Pour ce faire, on crée un fichier YAML à l'intérieur duquel on spécifie les configurations nécessaires à chaque service. Grâce à docker compose, tous les conteneurs dont on a besoin pourront être exécutés à l'aide d'une seule commande. [23] [24]

- Installation

```
$ sudo apt update
$ sudo curl -L https://github.com/docker/compose/releases/download/v2.16.0/docker-compose
`uname -s`-`uname -m` -o /usr/local/bin/docker-compose
$ sudo chmod +x /usr/local/bin/docker-compose
$ docker version
```

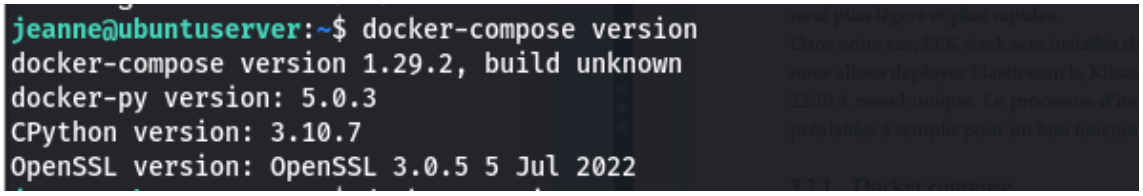


FIGURE 3.3 : Vérification de docker-compose

3.2.3 Création des conteneurs avec docker-compose [27]

Dans notre répertoire utilisateur (/home/jeanne), créons un nouveau dossier et un nouveau fichier de configuration pour EFK stack.

```
$ mkdir /efk/docker-efk
$ cd /efk/docker-efk
$ touch /efk/docker-compose.yml
```

L'étape suivante consiste à remplir le fichier de configuration que nous venons de créer. Commençons à éditer le fichier de configuration au format YAML.

```
$ nano docker-compose.yml
```

La configuration ci-dessous définit tous les conteneurs de EFK stack.

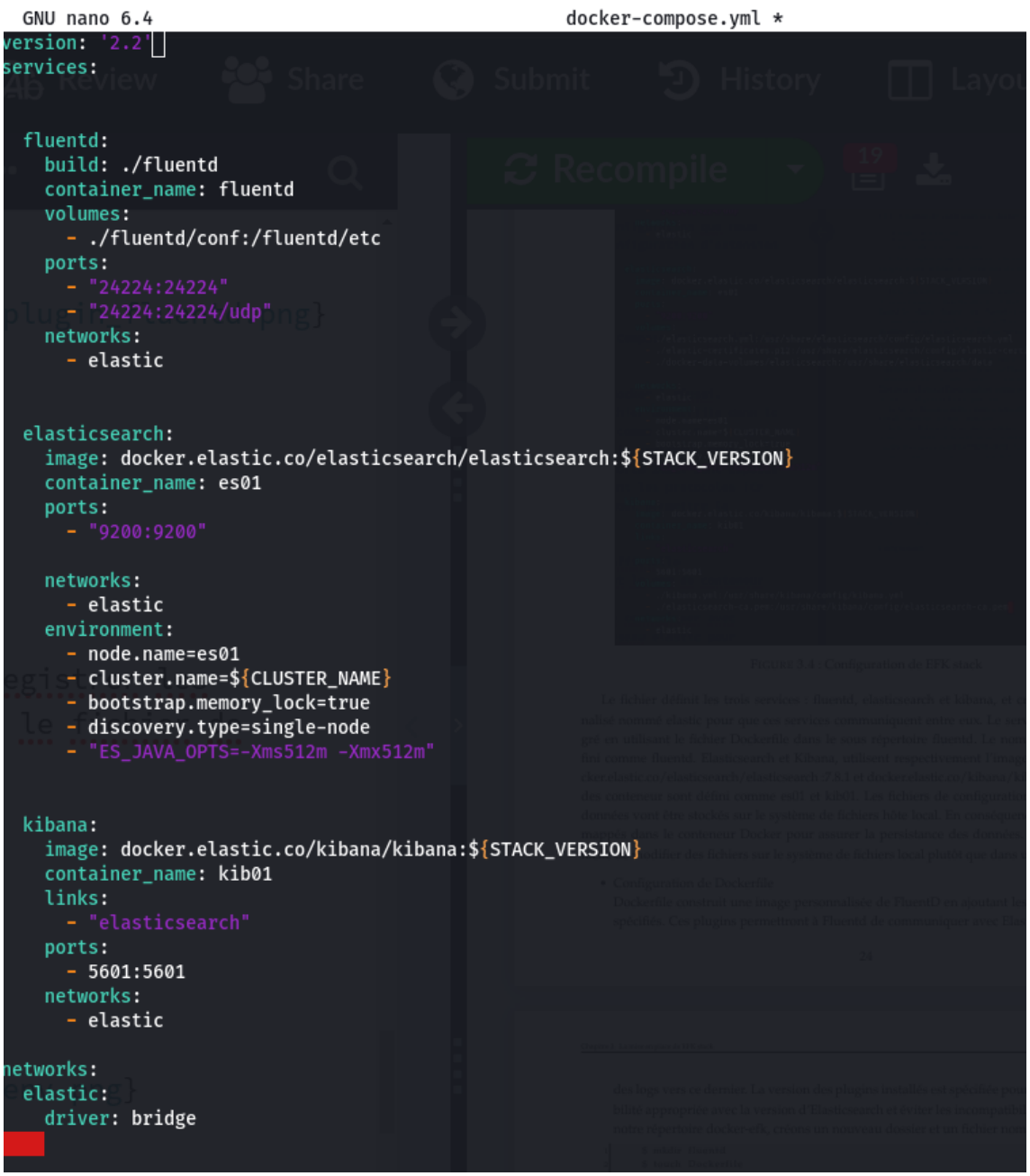


FIGURE 3.4 : Configuration de base EFK stack

Le fichier définit les trois services : Fluentd, Elasticsearch et kibana, et crée un réseau personnalisé nommé **elastic** pour que ces services communiquent entre eux. Le service Fluent, il est intégré en utilisant le fichier Dockerfile dans le sous-répertoire **Fluentd**.

- Configuration de Dockerfile
  Dockerfile construit une image personnalisée de FluentD en ajoutant les plugins Elasticsearch spécifiés. Ces plugins permettront à Fluentd de communiquer avec Elasticsearch et d'envoyer des logs vers ce dernier. Dans notre répertoire docker-efk, créons un nouveau dossier et un fichier nommé Dockerfile.

```
$ mkdir fluentd
```

```
$ touch Dockerfile
$ nano Dockerfile
```

```
GNU nano 6.4                                    fluentd/Dockerfile
# fluentd/Dockerfile

FROM fluent/fluentd:v0.12-debian
RUN fluent-gem install elasticsearch -v 7.5
RUN fluent-gem install elasticsearch-transport -v 7.5.0
RUN fluent-gem install fluent-plugin-elasticsearch -v 1.13.0 --no-rdoc --no-ri
```

FIGURE 3.5 : Dockerfile de Fluentd

- Configuration de Elasticsearch et Kibana
  Tout d'abord, créez deux fichiers elasticsearch.yml et kibana.yml dans le même répertoire que notre docke-compose.yml

```
$ touch elasticsearch.yml
$ touch kibana.yml
```

Editer nos fichier :

```
$ nano elasticsearch.yml kibana.yml
```

```
GNU nano 6.4                                    elasticsearch.yml *
network.host: 0.0.0.0
xpack.security.enabled: false
```

FIGURE 3.6 : Fichier de configuration de base d'Elasticsearch.

```
GNU nano 6.4                                    kibana.yml *
server.name: kibana
server.host: "0"
elasticsearch.hosts: [ "http://es01:9200" ]
```

FIGURE 3.7 : Fichier de configuration de base de Kibana

- Lancement de nos conteneurs
  Les fichiers elasticsearch.yml et kibana.yml sont liés à leurs conteneurs respectifs (on parle des volumes), on remplace les fichiers par défaut dans ces conteneurs. Après cela, on peut créer nos conteneurs.

```
$ docker-compse up build -d
```

Cette commande commencera à extraire les images du registre Docker et créera les conteneurs. L'option -d pour démarrer les conteneurs en arrière plan.

```
jeanne@ubuntuserver:~/efk/docker-efk$ docker-compose ps
 Name              Command               State                                    Ports
-------------------------------------------------------------------------------------------------------------------------
es01      /bin/tini -- /usr/local/bi ...   Up      0.0.0.0:9200->9200/tcp,:::9200->9200/tcp, 9300/tcp
fluentd   tini -- /bin/entrypoint.sh ...   Up      0.0.0.0:24224->24224/tcp,:::24224->24224/tcp,
                                                   0.0.0.0:24224->24224/udp,:::24224->24224/udp, 5140/tcp
kib01     /bin/tini -- /usr/local/bi ...   Up      0.0.0.0:5601->5601/tcp,:::5601->5601/tcp
```

FIGURE 3.8 : Vérification des conteneurs

On peut voir nos trois conteneurs avec le satut "up". A ce stade nous avons construit un cluster à noeud unique. On peut accéder à kibana à l'adresse url http ://192.168.43.188 :5601 et à elasticsearch http ://192.168.43.9200

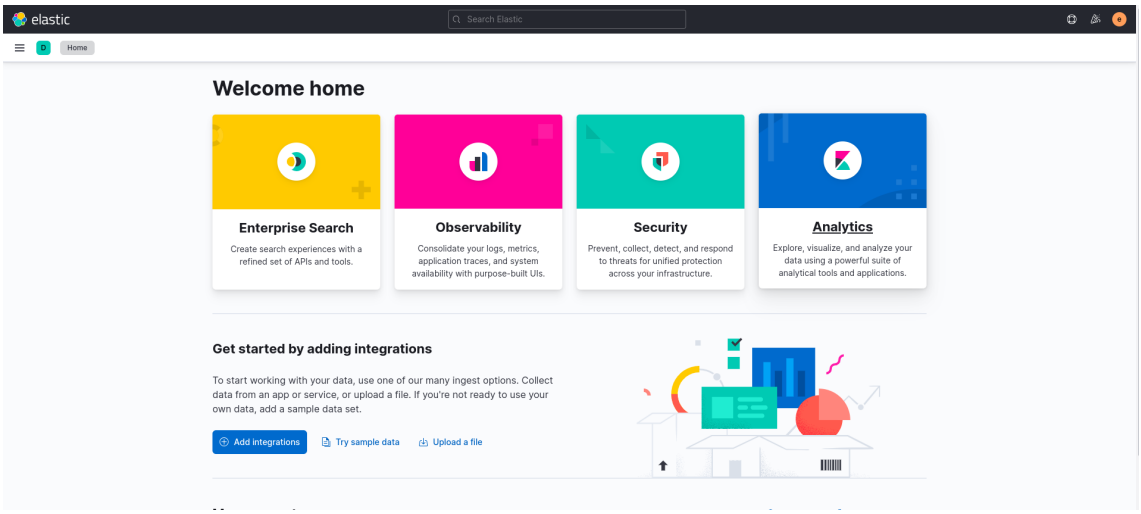


FIGURE 3.9 : Vérification de Kibana

Bien que cela fonctionne, il existe un défi de sécurité.

3.2.4 Configuration de la sécurité

- Création et activation d'un certificat SSL pour Elasticsearch

```
$ docker exec es01 bash
$ bin/elasticsearch-certutil ca
$ bin/elasticsearch-certutil cert --ca elastic-stack-ca.p12
$ docker cp "$(docker-compose ps -q elasticsearch)":/usr/share/
elasticsearch/elastic-certificates.p12 .
```

- Installation du certificat SSL sur Elasticsearch et activation de TLS dans la configuration
Maintenant que nous avons le certificat SSL disponible, nous pouvons activer la sécurité X-Pack sur notre nœud Elasticsearch et également activer TLS. Nous devons lier le montage de notre certificat de la machine hôte au conteneur. Pour cela, modifions notre fichier de configuration principale ainsi que le fichier elasticsearch.yml.

```
 GNU nano 6.4                                  docker-compose.yml *
    build: ./fluentd
    container_name: fluentd
    volumes:
      - ./fluentd/conf:/fluentd/etc
    ports:
      - "24224:24224"
      - "24224:24224/udp"
    networks:
      - elastic


  elasticsearch:
    image: docker.elastic.co/elasticsearch/elasticsearch:${STACK_VERSION}
    container_name: es01
    ports:
      - "9200:9200"
    volumes:
      - ./elasticsearch.yml:/usr/share/elasticsearch/config/elasticsearch.yml
      - ./elastic-certificates.p12:/usr/share/elasticsearch/config/elastic-certificates.p12
      - ./docker-data-volumes/elasticsearch:/usr/share/elasticsearch/data

    networks:
      - elastic
    environment:
      - node.name=es01
      - cluster.name=${CLUSTER_NAME}
      - bootstrap.memory_lock=true
      - discovery.type=single-node
      - "ES_JAVA_OPTS=-Xms512m -Xmx512m"


  kibana:
    image: docker.elastic.co/kibana/kibana:${STACK_VERSION}
    container_name: kib01
    links:
      - "elasticsearch"
    ports:
      - 5601:5601
    volumes:
      - ./kibana.yml:/usr/share/kibana/config/kibana.yml
    networks:
      - elastic

networks:
  elastic:
    driver: bridge
```

FIGURE 3.10 : Configuration finale de EFK Stack

```
 GNU nano 6.4                                  elasticsearch.yml *
network.host: 0.0.0.0
xpack.security.enabled: true
xpack.security.transport.ssl.enabled: true
xpack.security.transport.ssl.keystore.type: PKCS12
xpack.security.transport.ssl.verification_mode: certificate
xpack.security.transport.ssl.keystore.path: elastic-certificates.p12
xpack.security.transport.ssl.truststore.path: elastic-certificates.p12
xpack.security.transport.ssl.truststore.type: PKCS12
```

FIGURE 3.11 : Configuration de la sécurité sur Elasticsearch

Nous avons activé le X-Pack et le reste des lignes indique les paramètres SSL et l'emplacement de notre certificat et de notre clé privée, qui sont identiques.

- Génération des mots de passes
  Redémarrer nos services

```
$ docker-compose restart
```

  Générer les mots de passe

```
$ docker exec es01 bash
$ bin/elasticsearch-setup-passwords auto
```

  Les mots de passes sont générés pour tout les utilisateurs pas defaut.

- Chiffrer les communications du client HTTP pour Elasticsearch

```
$ docker exec es01 bash
$ bin/elasticsearch-certutil http
```

  Cette commande génère un fichier zip contenant des certificats et des clés à utiliser avec Elasticsearch et Kibana
  Installation

  – Décompresser le fichier généré elasticsearch-ssl-http.zip. Ce fichier compressé contient un répertoire pour Elasticsearch et Kibana.
  – Copier le certificat approprié http.p12 dans le répertoire de configuration d'elasticsearch

```
$ cp http.p12 /usr/share/elasticsearch/config
```

  – Modifier le fichier elasticsearch.yml pour activer la sécurité HTTPS
  – Spécifier l'emplacement du http.p12 certificat de sécurité.

```
 GNU nano 6.4                                  elasticsearch.yml
network.host: 0.0.0.0
xpack.security.enabled: true
xpack.security.transport.ssl.enabled: true
xpack.security.transport.ssl.keystore.type: PKCS12
xpack.security.transport.ssl.verification_mode: certificate
xpack.security.transport.ssl.keystore.path: elastic-certificates.p12
xpack.security.transport.ssl.truststore.path: elastic-certificates.p12
xpack.security.transport.ssl.truststore.type: PKCS12
xpack.security.http.ssl.enabled: true
xpack.security.http.ssl.keystore.path: http.p12
```

FIGURE 3.12 : Configuration finale d'Elasticsearch

- Chiffrer les communications du client HTTP pour Kibana

  – Chiffrer le trafic entre Kibana et Elasticsearch
  Lorsqu'on avait exécuté l'outil elasticsearch-certutil avec l'option HTTP, il a créé un répertoire /kibana contenant un fichier elasticsearch-ca.pem. Ce fichier permet de configurer Kibana afin qu'il fasse confiance à l'autorité de certification Elasticsearch pour la couche HTTP.
    * Copier le fichier elasticsearch-ca.pem dans le répertoire de configuration de Kibana

```
$ docker cp "$(docker-compose ps -q elasticsearch)":/usr/share/
elasticsearch/kibana/elasticsearch-ca.pem
"$(docker-compose ps -q kibana)":/usr/share/kibana/config
```

– Crypter le trafic entre le navigateur et Kibana

```
$ docker-compose ki01 bash
$ bin/elasticsearch-certutil ca --pem
```

Un fichier elastic-stack-ca.zip est généré.

* Décompresser le fichier. Ce fichier compressé contient un répertoire nommé ca contenant le certicat et la crée privé.
* Copier le certificat et dans le répertoire de configuration de kibana :

```
$  cp elasticsearch/kibana/elasticsearch-ca.pem
"$(docker-compose ps -q kibana)":/usr/share/kibana/config
```

* Modifier le fichier kabana.yml pour configurer Kibana afin d'accéder au certificat du serveur et à la clé privée.

```
 GNU nano 6.4                                          kibana.yml
server.name: kibana
server.host: "0"
elasticsearch.hosts: [ "https://es01:9200" ]
elasticsearch.username: kibana
elasticsearch.password: uBEraHN9LQ1YtpMEkdhR
elasticsearch.ssl.certificateAuthorities: /usr/share/kibana/config/elasticsearch-ca.pem

server.ssl.certificate: /usr/share/kibana/config/ca.crt
server.ssl.key: /usr/share/kibana/config/ca.key
server.ssl.enabled: true
server.publicBaseUrl: "https://172.21.0.4:5601"
```

FIGURE 3.13 : Configuration final de Kibana

- Configuration de FluentD
  Dans notre dossier fluentd, créons un nouveau dossier nommée conf et un fichier nommé fluent.conf dans ce dossier.

```
$ mkdir conf
$ touch fluent.conf
$ nano fluent.conf
```

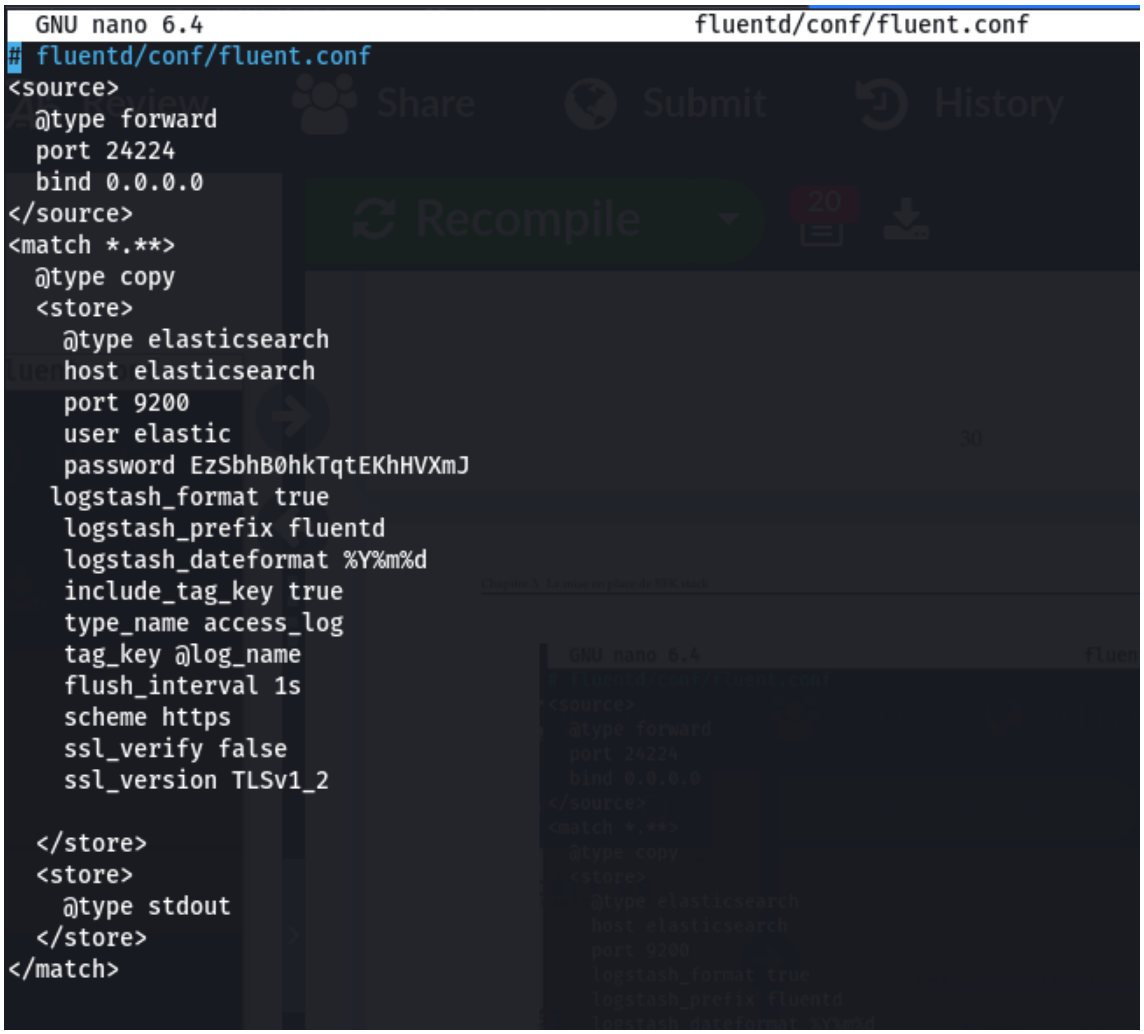


FIGURE 3.14 : Fichier de configuration de Fluentd pour l'envoi des données à Elasticsearch

Ce fichier configure Fluentd qui reçoit des journaux sur le port 24224 de n'importe quelle interface. Les journaux reçus sont ensuite envoyés à Elasticsearch via HTTPS. Les logs seront indexés sous l'indice Fluentd et seront également affichés dans la sortie standard du conteneur Fluentd.

- Redémarrer tous nos conteneurs

```
1 $ docker-compose restart
```

On peut maintenant accéder à kibana et elasticsearch via https.

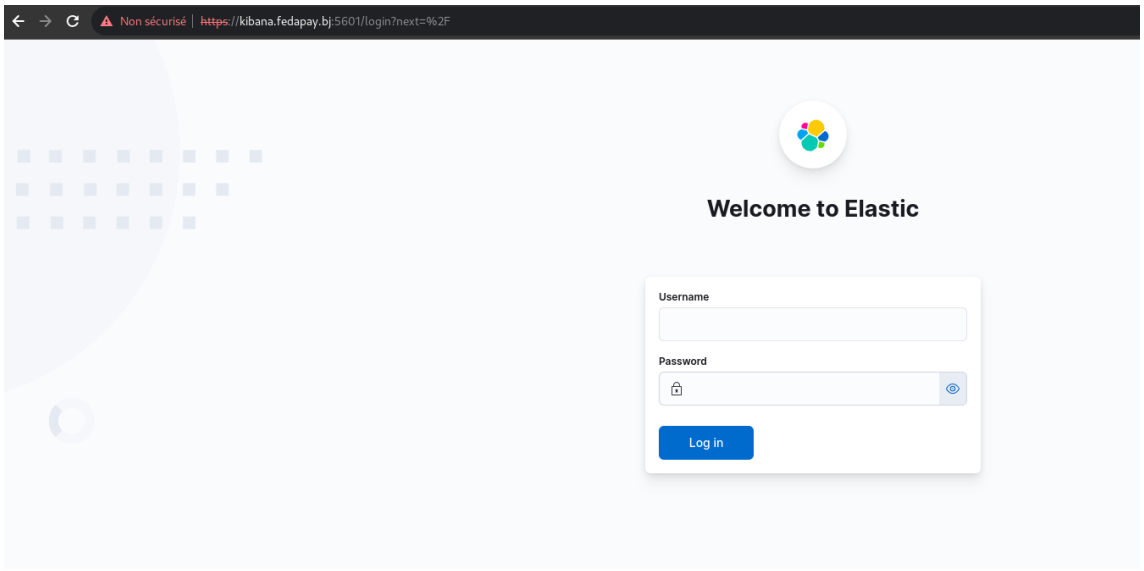


FIGURE 3.15 : Page de connexion de Kibana

3.3 Test de fonctionnalité

Rappelons que pour demarrer nos trois conteneur en même temps on utilise la commande **docker-compose start**

3.3.1 Configuration des machines à surveiller

Nous avons installer Td-agent sur les machines à surveiller. Td-agent se chagera de collecter les logs (rsyslog, apache, mysql..) et les envoyés à Fluentd.

- Installation et configuration de Td-agent sur centos [?]
  Installation

```
1     $ yum update -y
2     $ yum install curl
3     $ curl -L https://toolbelt.treasuredata.com/sh/install-redhat-td-agent4.sh | sh
```

- Vérification de l'état de Td-agent

```
1     $ systemctc status td-agent
```

```
[root@efkclient1 log]# systemctl status td-agent.service
● td-agent.service - td-agent: Fluentd based data collector for Treasure Data
   Loaded: loaded (/usr/lib/systemd/system/td-agent.service; enabled; vendor preset: disabled)
   Active: active (running) since dim. 2023-08-13 11:34:00 WAT; 12min ago
     Docs: https://docs.treasuredata.com/display/public/PD/About+Treasure+Data%27s+Server-Side+Agent
  Process: 1089 ExecStart=/opt/td-agent/bin/fluentd --log $TD_AGENT_LOG_FILE --daemon /var/run/td-agent/td-agent.pid $
TD_AGENT_OPTIONS (code=exited, status=0/SUCCESS)
 Main PID: 1782 (fluentd)
    Tasks: 12
   CGroup: /system.slice/td-agent.service
           ├─1782 /opt/td-agent/bin/ruby /opt/td-agent/bin/fluentd --log /var/log/td-agent/td-agent.log --daemon /v...
           └─1785 /opt/td-agent/bin/ruby -Eascii-8bit:ascii-8bit /opt/td-agent/bin/fluentd --log /var/log/td-agent/...

août 13 11:33:17 efkclient1 systemd[1]: Starting td-agent: Fluentd based data collector for Treasure Data...
août 13 11:34:00 efkclient1 systemd[1]: Started td-agent: Fluentd based data collector for Treasure Data.
```

FIGURE 3.16 : Etat de Td-agent sur centos

- Installation et configuration de Td-agent sur les machines ubuntu [21]
  Installation

```
$ yum update
$ yum install curl
$ curl -L https://toolbelt.treasuredata.com/sh/install-ubuntu-xenial-td-agent4.sh | sh
```

- Vérification de l'état de Td-agent sur l'une des machines

```
$ systemctc status td-agent
```

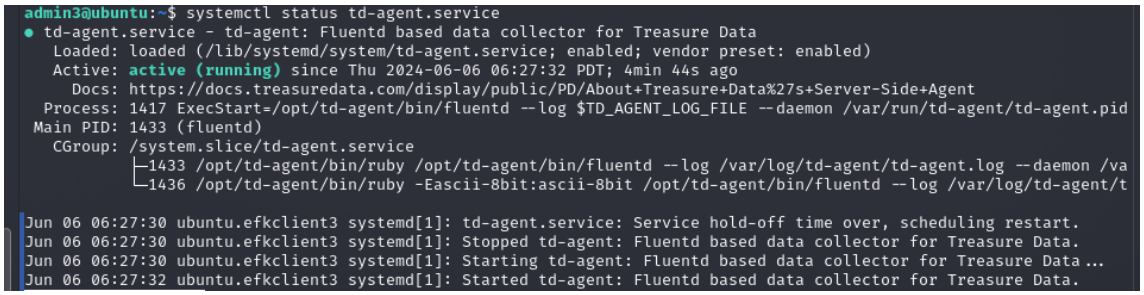


FIGURE 3.17 : Etat de Td-agent sur ubuntu

- Configuration de Td-agent sur centos La configuration se fait dans le fichier de configuration de Td-agent (généralement situé à /etc/td-agent/td-agent.conf). Il existe des configurations par défaut dans ce fichier. Nous avons modifier en fonction de nos objectifs.

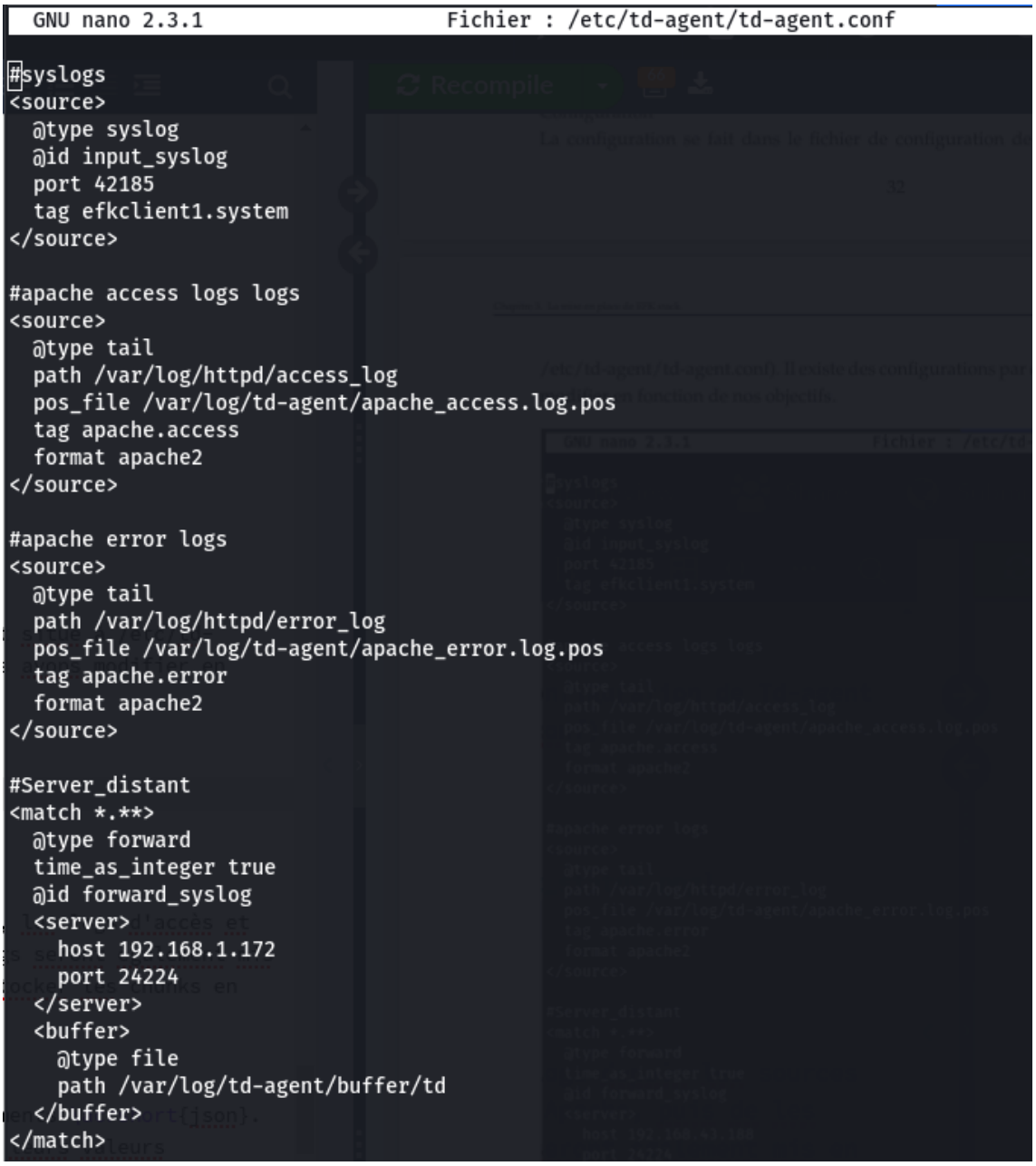


FIGURE 3.18 : Configuration de Td-agent sur centos

Cette configuration permet de collecter les logs depuis les sources Rsyslog, les logs d'accès et les logs d'erreur d'Apache donc les logs de notre application en local, puis de les envoyer vers nôtre serveur Fluentd avec un tampon en fichier en cas de problème de transfert.

- Configuration de Td-agent les machines ubuntu

```
  GNU nano 2.5.3                File: /etc/td-agent/td-agent.conf

#Syslogs
<source>
  @type syslog
  @id input_syslog
  port 42185
  tag efkclient2.system
</source>


#Server_distant
<match *.**>
  @type forward
  time_as_integer true
  @id forward_syslog
  <server>
    host 192.168.43.208
    port 24224
  </server>
  <buffer>
    @type file
    path /var/log/td-agent/buffer/td
  </buffer>
</match>
```

FIGURE 3.19 : Configuration de Td-agent sur les machines ubuntu

Cette configuration permet de collecter les logs systemes Rsyslog puis de les envoyer vers nôtre serveur Fluentd avec un tampon en fichier en cas de problème de transfert. Seul le tag chang pour chaque configuration des systèmes ubuntu

3.3.2 Création d'index sur Kibana

Un **index** est une structure de stockage qui contient un ensemble de documents JSON. Chaque document représente un enregistrement et peut contenir divers champs avec leurs valeurs associées.
Après cette étape accédon à Kibana
À l'ouverture de kibana une onglet s'affiche pour créer un index.



FIGURE 3.20 : Création d'index

L'index par defaut est *. L'index **fluentd*** que nous avons crée regroupe tout index de préfix fluentd. Dans 'time field' nous indiquons @timestamp qui correspond au champs d'horodatage.

3.3.3 Machine centos

3.3.3.1 Envoi de simple logs

Sur nôtre machine cliente(centos) nous avons envoyé de simple logs avec cette commande :

```
1    $ logger -t TEST1 hello I'm from efkclient1
```

L'Exploration des données de manière interactive se fait à partir de la page 'Discover'.
Pour vérifier l'arrivée des logs dans kibana, nous avons ajouté un filtre pour afficher tous les logs avec l'identité TEST1.

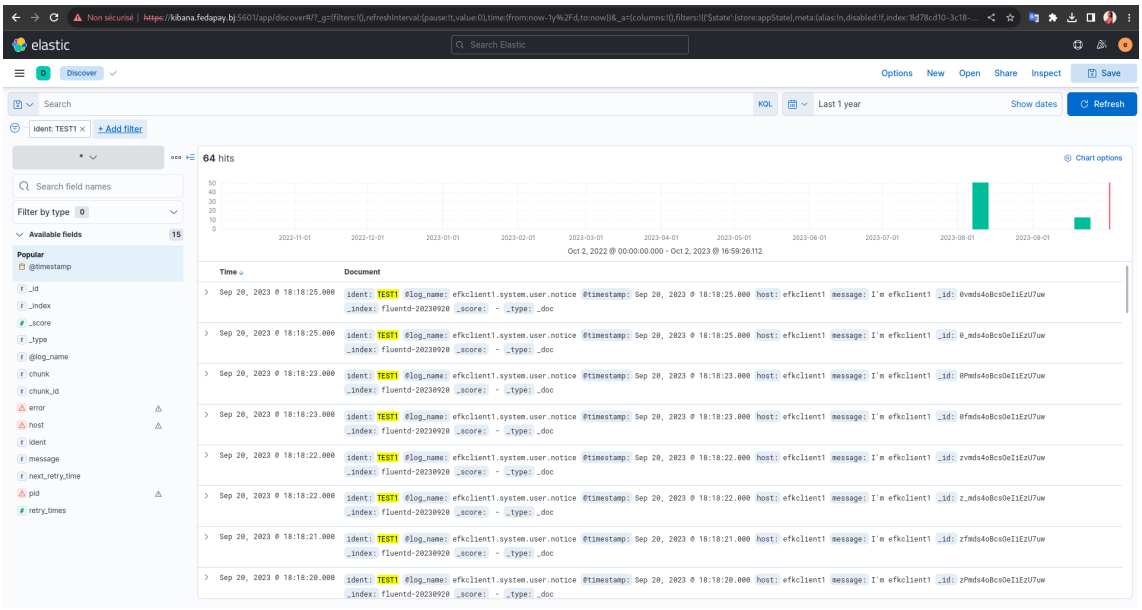


FIGURE 3.21 : Filtrage des logs

On peut consulter l'intégralité des logs sous forme de table ou sous format JSON. Le format table, particulièrement, permet de mieux visualiser tous les champs du document.

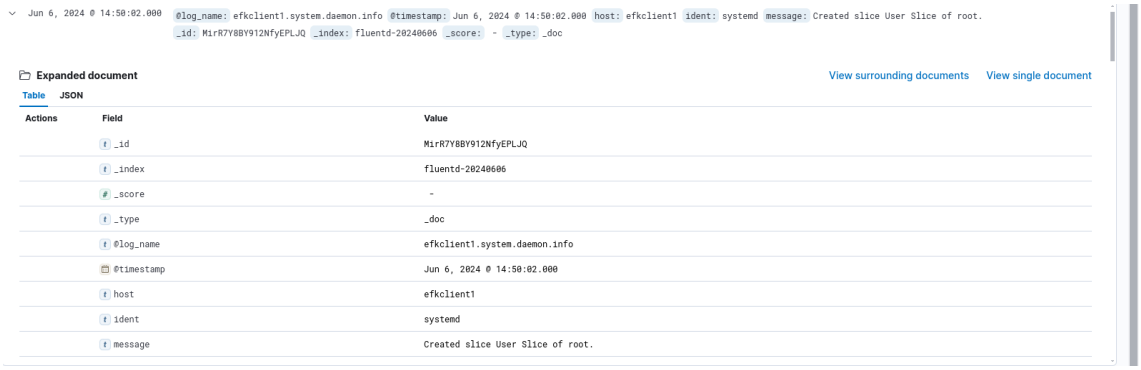


FIGURE 3.22 : Exemple de logs sous forme de table

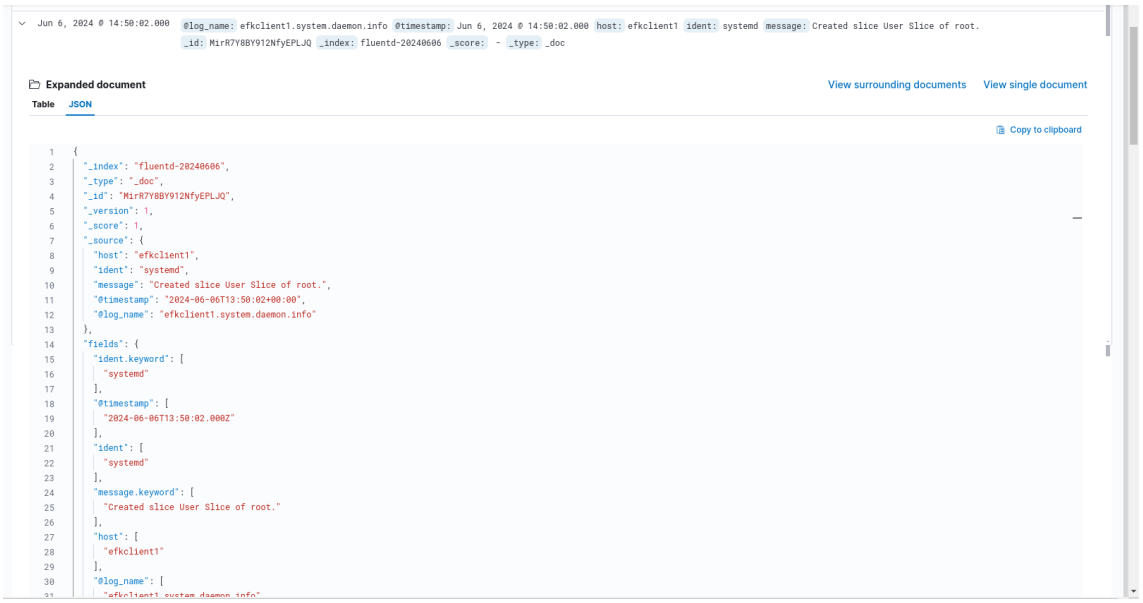


FIGURE 3.23 : Exemple de logs sous format JSON

3.3.3.2 Les logs SSH

Recherchons les évènements d'authentifications SSH échouer du 15/08/2023 à travers le filtre message : **"sshd" and message : "Failed password " and @timestamp : "2023-08-15" and @log-name : "efkclient1"** qui indique à Kibana de chercher tous les évènements dont le contenu du champ message contient le mot "sshd" et en plus les mots « Failed password » et dont le timestamp correspond à "2023-08-15 sur" le système CentOS avec le tag "efkclient1".

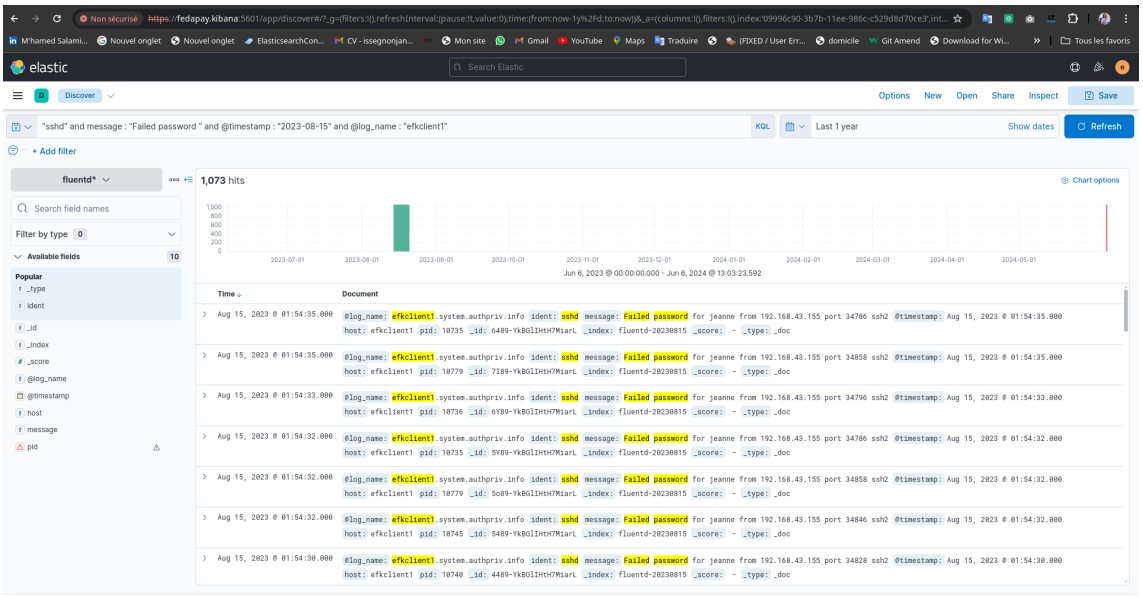


FIGURE 3.24 : Recherche des authentifications échouées du 15/08/2023

On peut envoyer une alerte aux équipes de sécurité après un nombre de tentatives de connexion échoué pendant une durée donné.

3.3.3.3 Les logs d'Apache

Notre application est accessible en local via http ://192.168.43.104/online-shopping-system/store.php
Les évènements générés pas notre application web et notre base de donnée MySQL, de phpMyAdmin sont directement envoyés au serveur web apache depuis le système. Les logs d'accès et les logs d'erreur d'Apache sont marqués respectivement avec apache.access et apache.error

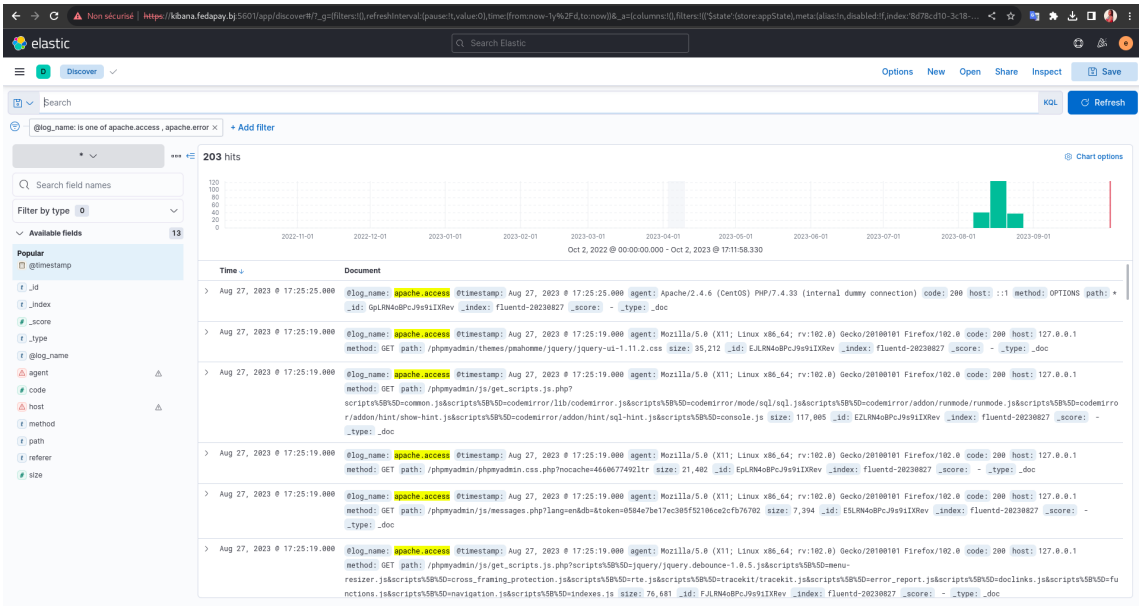


FIGURE 3.25 : Visualisation des logs Apache

Nous pouvons voir l'activité de chaque utilisé sur notre application (les pages consultés, l'adresse de l'utilisateur par exemple) ainsi que les erreurs liées à la base de donnée…,
Exemple de logs d'un utilisateur qui a consulté notre page avec un iPhone. Nous avons les informations de son téléphone, l'heure, les pages consultés, le navigateur utilisé ainsi que son adresse IP.

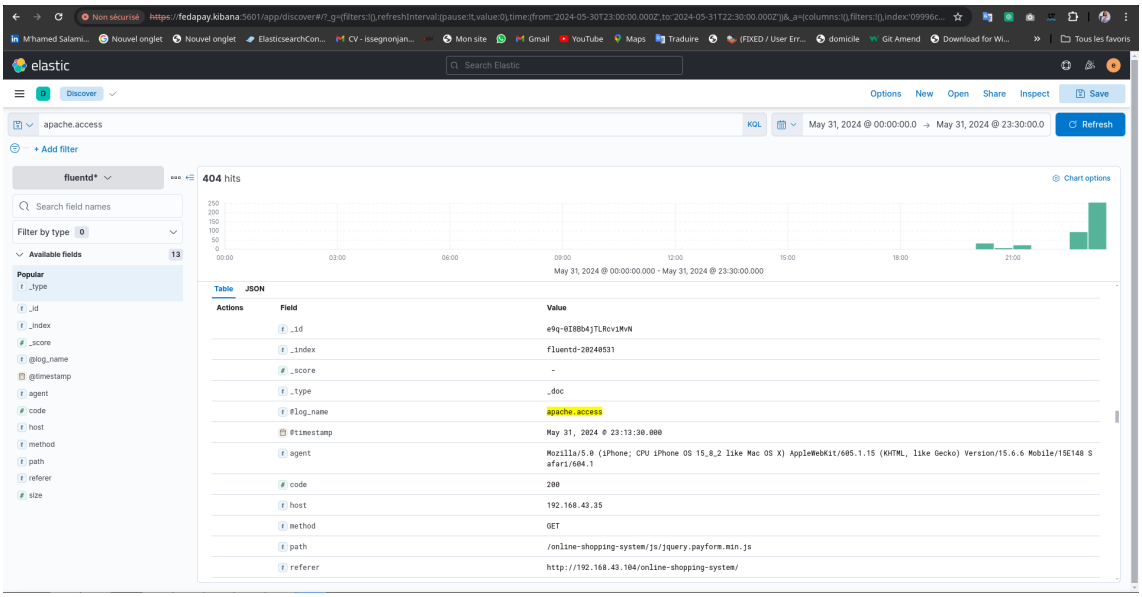


FIGURE 3.26 : Logs liés à notre Application web

Ce document montre que notre application n'arrive pas à se connecter à la base de donnée avec

les informations fournir. L'équipe sécurité peut corriger cette erreur avant que ça ne parvienne aux utilisateurs finaux.

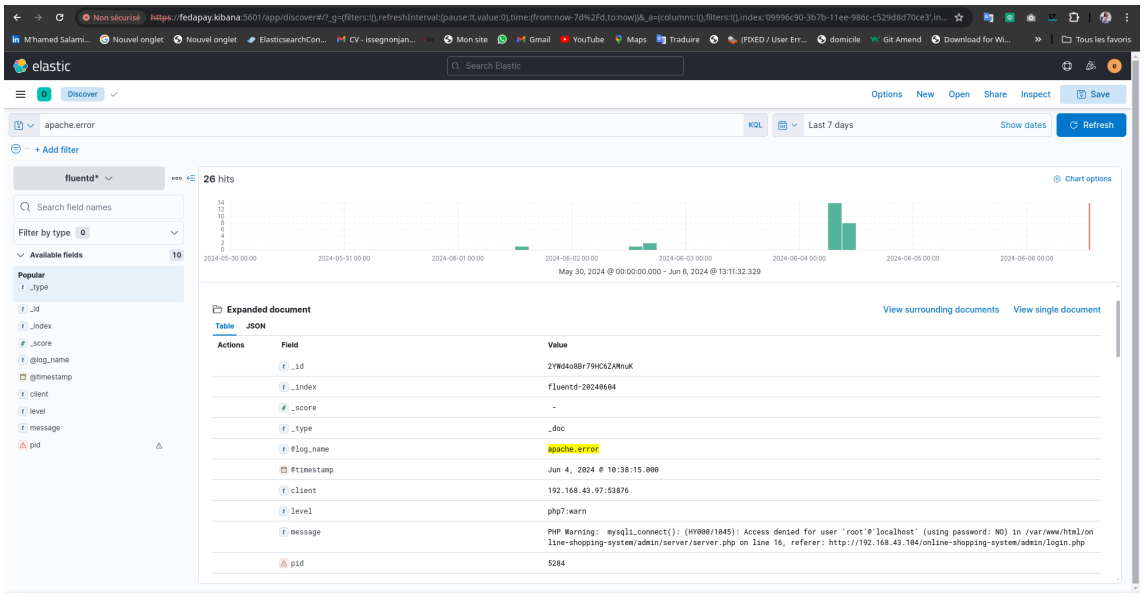


FIGURE 3.27 : Logs d'erreur de connexion à la base de donnée

Même les erreurs de code de notre application

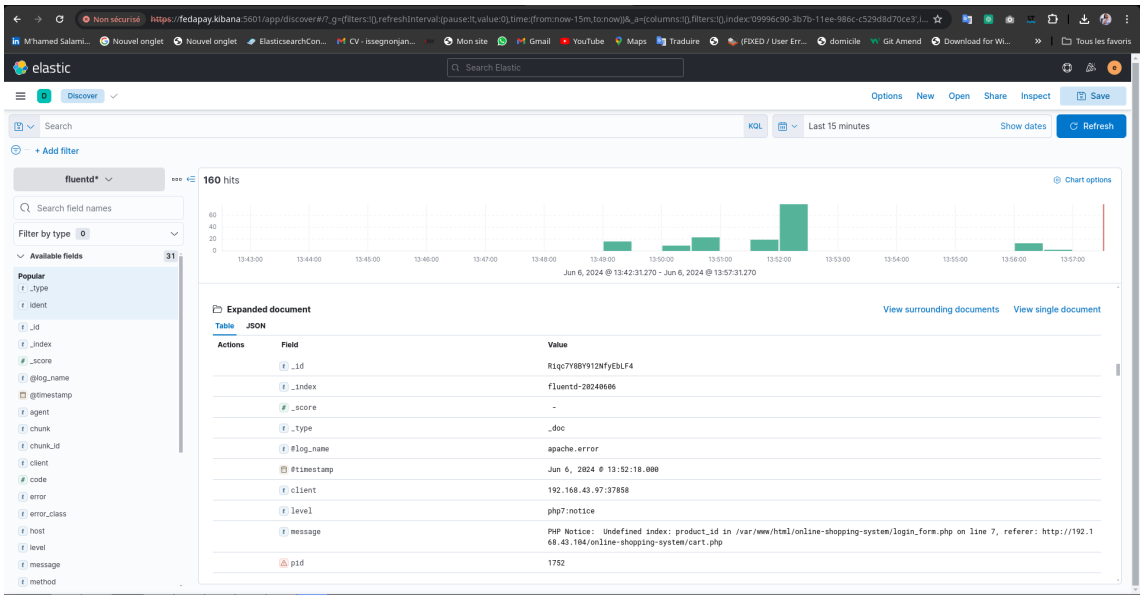


FIGURE 3.28 : PHP erreur

3.3.4 Les systèmes Ubuntu

On peut visionner les logs des systèmes Ubuntu avec le tag efkclient1 et efkclient2, rechercher les tentatives de connexion, les logs générés par les services du système...

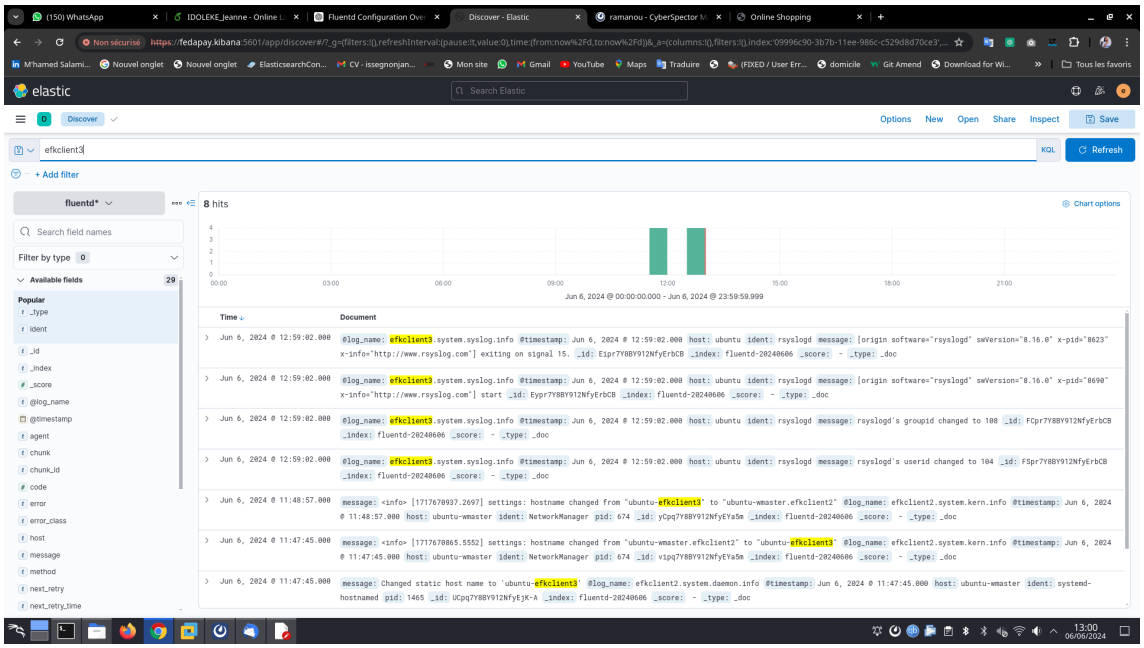


FIGURE 3.29 : Visionnage des logs avec le tag efkclient2

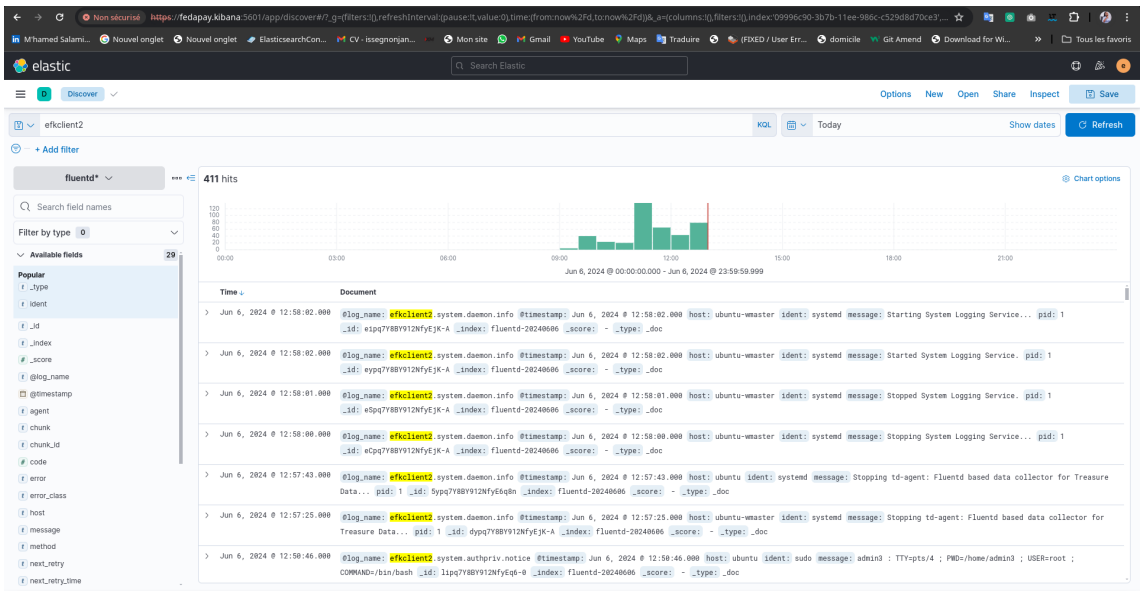


FIGURE 3.30 : Visionnage des logs avec le tag efkclient3

3.3.5 Kibana Dashboard

Pour ce faire, on se rend dans l'onglet "Visualize Library" puis "Create visualisation". Une fois dans cet onglet, on voit les différents types de visualisation qu'on peut créer. Nous avons créé plusieurs visualisations comme le nombre de logs, les messages, etc pour avoir un dashboard bien structuré.

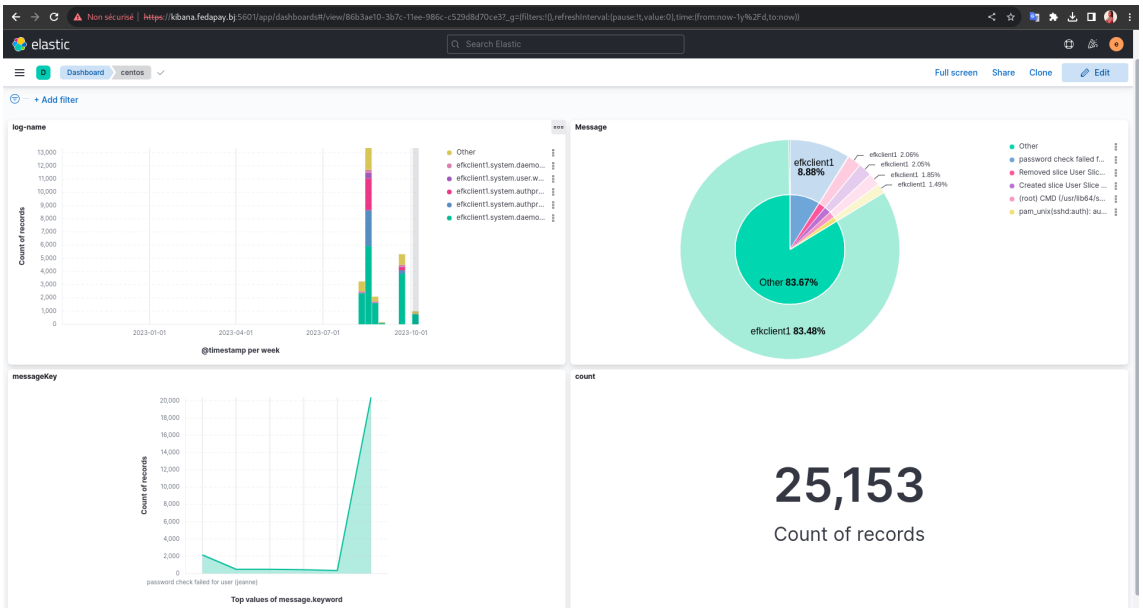


FIGURE 3.31 : Dashboard crée sur Kibana

Conclusion

Dans ce chapitre, nous avons expliqué les étapes de configuration de EFK stack en utilisant les conteneurs docker et de ses composants et de la configuration des machines de test. Ensuite, nous avons effectué des tests de fonctionnalité en créant des visualisations et des dashboards mettant ainsi en évidence ses fonctionnalités, ces dashboards nous ont permis de superviser notre système et d'avoir une vue globale et plus détaillée des informations de sécurité de nos systèmes.

Conclusion Générale

À travers ce document, nous avons fait une étude approfondie des protocoles et logiciels de gestion des logs disponibles dans le domaine informatique, en optant finalement pour le EFK stack pour répondre aux besoins spécifiques de Fedapay SAS. Notre approche a consisté à mettre en place cette solution à travers des conteneurs Docker, une décision stratégique permettant une gestion flexible et évolutive des logs au sein de l'environnement hautement dynamique. Nous avons été confrontés à plusieurs défis techniques tout au long du projet. D'une part, la création des différents conteneurs en utilisant des fichiers YAML a nécessité une compréhension approfondie de l'architecture et des interactions entre les composants Elasticsearch, Fluentd et Kibana. D'autre part, la configuration de Td-agent pour assurer l'acheminement efficace des logs vers notre serveur de gestion a exigé une expertise fine dans la manipulation et l'optimisation des flux de données. Nos difficultés ont été surmontées grâce à nos recherches persévérantes et aux apports de notre encadreur. Ce projet nous a permis d'approfondir les connaissances théoriques et pratiques que nous avons acquises tout au long de notre formation.
Par ailleurs, en termes de perspectives, nous envisageons plusieurs développements futurs pour améliorer encore notre infrastructure de gestion des logs :

- Étudier davantage EFK stack afin d'exploiter d'autres fonctionnalités de cet outil de log management très performant ;
- Collecter les logs d'un environnement entièrement conteneurisé sans utilisant un argent sur les conteneurs ;
- Configurer des alertes avec Fluentd ou Kibana pour notifier les administrateurs système via email ou Slack ;
- Intégrer des solutions de détection d'incidents comme Wazuh ;
- Mettre en place la haute disponibilité pour assurer la résilience de notre infrastructure.

Webographie

[1] https://www.tek-tools.com/apm/log-management-tips-and-tools. Consulté le 05 Juin 2023.

[2] https://fr.wikipedia.org/wiki/Gestion_des_logs. Consulté le 05 Janvier 2023.

[3] https://cisco.goffinet.org/ccna/gestion-infrastructure/gestion-des-logs-syslog/. Consulté le 05 Juin 2023.

[4] https://cisco.goffinet.org/ccna/gestion-infrastructure/supervision-snmp/. Consulté le 05 Juin 2023.

[5] https://www.solarwinds.com/resources/it-glossary/windows-event-log. Consulté le 05 Juin 2023.

[6] https://www.solarwinds.com/resources/it-glossary/log-management. Consulté le 05 Juin 2023.

[7] https://www.mezmo.com/learn-log-management/siem-vs-log-management. Consulté le 8 Juin 2023.

[8] https://datascientest.com/apache-flume-tout-savoir. Consulté le 8 Juin 2023.

[9] https://www.elastic.co/fr/what-is/elk-stack. Consulté le 12 Juin 2023.

[10] https://signoz.io/blog/fluentd-vs-logstash/. Consulté le 12 Juin 2023.

[11] https://aws.amazon.com/fr/elasticsearch-service/what-is-elasticsearch/. Consulté le 20 Juin 2023.

[12] https://docs.fluentd.org/. Consulté 20 Juin 2023.

[13] https://www.mezmo.com/blog/fluentd-vs-logdna. Consulté le 20 Juin 2023.

[14] https://docs.fluentd.org/quickstart. Consulté le 03 Juillet 2023.

[15] https://medium.com/ibm-cloud/fluentd-working-demystified-6ae5b46704f2. Consulté le 03 Juillet 2023.

[16] https://docs.fluentd.org/quickstart. Consulté le 03 Juillet 2023.

[17] https://docs.treasuredata.com/display/public/PD/Installing+td-agent+on+Ubuntu+and+Debian. Consulté le 10 Juillet 2023.

[18] https://www.elastic.co/fr/what-is/kibana. Consulté le 10 Juillet 2023.

[19] https://www.elastic.co/fr/kibana/. Consulté le 10 Juillet 2023.

[20] https://blogs.halodoc.io/logging-with-efk/. Consulté le 12 Juillet 2023.

[21] https://fr.wikipedia.org/wiki/Virtualisation. Consulté le 20 Juillet 2023.

[22] https://fr.wikipedia.org/wiki/Docker_(logiciel. Consulté le 25 Juillet 2023.

[23] https://cloudinfrastructureservices.co.uk/how-to-install-and-use-docker-compose-on-ubuntu-22-04/. Consulté le 25 Juillet 2023.

[24] https://www.data-transitionnumerique.com/docker-compose/. Consulté le 5 Aoû 2023.

[25] https://docs.treasuredata.com/articles/#!pd/installing-td-agent-on-ubuntu-and-debian. Consulté le 5 juin 2024.

[26] https://docs.fluentd.org/installation/install-by-rpm. Consulté le 5 Juin 2023.

[27] https://blog.yarsalabs.com/efk-setup-for-logging-and-visualization/. Consulté le 5 Juin 2023.

Table des matières

Dédicace ii

Remerciements iii

Résumé iv

. iv

Abstract v

. v

Table des figures vi

Liste des tableaux vii

Liste des acronymes viii

Glossaire 1

Introduction 2

1 Revue de littérature 4

Introduction 4

1.1 La gestion des logs 4

1.1.1 Définition 4

1.1.2 Objectifs 4

1.1.3 Les protocoles de gestion des logs 5

1.1.3.1 Syslog 5

1.1.3.2 SNMP 5

1.1.3.3 Windows Event Log 6

1.1.4 Les avantages de gestion des logs 6

1.1.4.1 Efficacité opérationnelle améliorée 6

1.1.4.2 Utilisation efficace des ressources 6

1.1.4.3 Dépannage proactif 6

1.1.4.4 Meilleure expérience de l'utilisateur final 6

1.1.4.5 Une plus grande sécurité 6

1.1.5 Similitudes entre SIEM et la gestion des journaux 7

1.1.6 Différence entre SIEM et la gestion des logs 7

1.1.7 Quelques solutions de gestion des logs 7

1.1.7.1 Les solutions payantes 7

1.1.7.2 Les solutions Open Source . . . 8
1.1.8 Étude Comparative de quelques solutions Open Source . . . 9
1.1.8.1 Étude comparative entre Fluentd et Logstash . . . 9
1.2 La solution choisie : EFK stack . . . 11
Conclusion . . . 11

2 Présentation de EFK stack **12**
Introduction . . . 12
2.1 Présentation de EFK stack . . . 12
2.2 Elasticsearch . . . 12
2.3 FluentD . . . 12
2.3.1 Architecture de Fluentd . . . 13
2.3.2 Plugin . . . 13
2.3.3 TD-Agent . . . 14
2.4 Kibana . . . 15
2.5 Fonctionnement de EFK stack . . . 15
2.6 Gestion des alertes . . . 15
2.7 Les méthodes d'installation de EFK stack . . . 16
2.8 Description du système informatique . . . 16
Conclusion . . . 17

3 La mise en place de EFK stack **18**
Introduction . . . 18
3.1 La virtualisation . . . 18
3.1.1 Présentation de l'architecture virtuelle . . . 18
3.2 Installation de EFK stack . . . 19
3.2.1 Docker . . . 19
3.2.2 Docker compose . . . 20
3.2.3 Création des conteneurs avec docker-compose [27] . . . 20
3.2.4 Configuration de la sécurité . . . 23
3.3 Test de fonctionnalité . . . 28
3.3.1 Configuration des machines à surveiller . . . 28
3.3.2 Création d'index sur Kibana . . . 31
3.3.3 Machine centos . . . 31
3.3.3.1 Envoi de simple logs . . . 31
3.3.3.2 Les logs SSH . . . 33
3.3.3.3 Les logs d'Apache . . . 34
3.3.4 Les systèmes Ubuntu . . . 35
3.3.5 Kibana Dashboard . . . 36
Conclusion . . . 37

Conclusion **38**

Webographie **39**

Table des matières **41**