

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
http://www.ifri-uac.net Courriel : contact@ifri.uac.bj



MÉMOIRE

pour l'obtention du

Diplôme de Licence en Informatique

Option : Sécurité Informatique

Présenté par :
Séna Nérys Marie-Ellen **TCHIBOZO**

Mise en place d'un système de gestion des logs pour les serveurs de l'Université d'Abomey-Calavi

Sous la supervision :
Ing Victor **OYETOLA**

Membres du jury :

COMLAN Maurice	Docteur	IFRI	Président
GNONLONFOUN Miranda	Ingénieure	IFRI	Examinateur
OYETOLA Victor	Ingénieur	IFRI	Rapporteur

Année Académique : **2023-2024**

Sommaire

Dédicace ii

Remerciements iii

Résumé iv

Abstract v

List of Figures vi

List of Tables vii

Liste des acronymes viii

Introduction 2

1 Revue de littérature 4

2 Choix de la solution et présentation du matériel 8

3 Implémentation et collecte des logs avec opensearch 14

Conclusion 34

Bibliographie 35

Webographie 36

Table des matières 37

Dédicace

Je dédie ce travail à :

À mon père, dont la mémoire et les valeurs continuent d'inspirer chaque pas de ma vie.
À ma mère, pour sa force et son soutien indéfectible.
À mes frères et sœurs, pour leur amour, leur complicité et leurs encouragements constants.
Et à mes amis et proches, pour leur amitié et leur soutien tout au long de ce parcours.

Remerciements

Je rends grâce à Dieu pour tous ses bienfaits dans ma vie et pour la réalisation de ce mémoire. J'adresse mes sincères remerciements

- au professeur EZIN C. Eugène, Directeur de l'Institut de Formation et de Recherche en Informatique à l'Université d'Abomey-Calavi
- à mon maître de mémoire M. Victor OYETOLA
- aux membres de la coordination du programme de bourse de la fondation mastercard à l'université de Abomey-Calavi
- à mes amis et collègues, pour leur amitié, leur écoute, et les moments de partage qui ont rendu ce parcours plus agréable
- à tous ceux qui, de près ou de loin, ont cru en moi et m'ont encouragé à donner le meilleur de moi-même

Résumé

Dans un environnement où la gestion des données et la sécurité informatique sont devenues des enjeux cruciaux, les institutions académiques, telles que l'Université d'Abomey-Calavi (UAC), doivent adopter des solutions technologiques avancées pour garantir la performance et la sécurité de leurs infrastructures. Ce mémoire propose l'implémentation d'un système de logging avancé, intégrant des outils modernes pour optimiser la surveillance des serveurs de l'UAC, améliorer la sécurité des données, et soutenir les objectifs académiques et administratifs de l'université. Ce travail examine le déploiement d'un système de logging avancé aux serveurs de l'UAC, utilisant Fluentd, OpenSearch, et OpenSearch Dashboards pour améliorer la surveillance, la sécurité et la performance informatique en optimisant l'agrégation, le stockage, et l'analyse des logs. Grâce à ces outils, l'UAC peut identifier les anomalies plus efficacement et prendre des décisions éclairées pour renforcer la gestion des risques informatiques.

Mots clés : Logging, Fluentd, OpenSearch, OpenSearch Dashboard, risques informatiques

Abstract

In an environment where data management and IT security have become crucial concerns, academic institutions like the University of Abomey-Calavi (UAC) must adopt advanced technological solutions to ensure the performance and security of their infrastructures. This thesis proposes the implementation of an advanced logging system, integrating modern tools to optimize server monitoring at UAC, enhance data security, and support the university's academic and administrative goals. This work examines the deployment of an advanced logging system for UAC servers, utilizing Fluentd, OpenSearch, and OpenSearch Dashboards to improve monitoring, security, and IT performance by optimizing the aggregation, storage, and analysis of logs. With these tools, UAC can more effectively identify anomalies and make informed decisions to strengthen IT risk management.

Key words: Logging, Fluentd, OpenSearch, OpenSearch Dashboard, IT risks

List of Figures

3.1 Configuration du mot de passe administrateur pour OpenSearch . . . 15
3.2 Configuration du mot de passe administrateur pour OpenSearch . . . 15
3.3 Vérification du bon fonctionnement d'OpenSearch . . . 16
3.4 Vérification de l'installation des plugins OpenSearch . . . 17
3.5 Configuration d'un utilisateur dans OpenSearch . . . 18
3.6 Configuration d'un utilisateur dans OpenSearch . . . 19
3.7 État du service OpenSearch . . . 19
3.8 Installation de Fluentd . . . 20
3.9 État de l'installation de Fluentd . . . 20
3.10 Test du fonctionnement de fluentd . . . 21
3.11 Activation d'OpenSearch Dashboards . . . 21
3.12 Interface d'OpenSearch Dashboards . . . 22
3.13 Site web utilisé . . . 23
3.14 Affichage du Dashboards . . . 25
3.15 Affichage du Dashboards . . . 26
3.16 Visualisation du Dashboards . . . 27
3.17 Visualisation du Dashboards . . . 28
3.18 Résultat du filtre . . . 29
3.19 Configuration d'alerte . . . 30
3.20 Configuration d'alerte . . . 30
3.21 Configuration d'alerte . . . 31
3.22 Génération d'alertes . . . 31
3.23 Notification . . . 32

List of Tables

2.1 Comparaison des systèmes de gestion de logs : OpenSearch, Fluentd, ELK, Splunk et Graylog . . 10

Liste des acronymes

UAC :

Université d'Abomey-Calavi 2, 9

Liste des acronymes

Introduction Générale

Contexte

Dans un environnement académique en rapide évolution, l'Université d'Abomey-Calavi fait face à une augmentation exponentielle du volume de données générées par ses activités quotidiennes et de recherche. Compte tenu de cette croissance et des risques potentiels associés à la gestion et à la sécurité des systèmes informatiques, l'université reconnaît la nécessité d'améliorer ses capacités de surveillance et d'analyse. L'efficacité opérationnelle et la sécurité des données deviennent des priorités stratégiques pour garantir la continuité académique et protéger les ressources de l'institution.

Problématique

Malgré la présence de systèmes informatiques en place, l'UAC a identifié plusieurs lacunes en termes de surveillance en temps réel et de réactivité face aux incidents de sécurité. La fragmentation des données et l'absence d'un système de logging cohérent limitent la capacité de l'université à détecter et à répondre efficacement aux anomalies de performance et aux menaces de sécurité. De plus, avec les réglementations de plus en plus strictes en matière de protection des données et les exigences de conformité, l'UAC est pressée de mettre en œuvre une solution robuste qui peut non seulement surveiller mais aussi analyser de manière proactive les activités au sein de ses systèmes informatiques

Objectifs

Le projet vise donc à déployer un système de logging avancé, utilisant des outils modernes tels que Fluentd, OpenSearch, et OpenSearch Dashboard, pour centraliser la collecte de logs et améliorer les capacités d'analyse des données. L'objectif est de créer une infrastructure capable de répondre aux besoins en surveillance, en performance et en sécurité de l'université tout en facilitant la conformité aux normes réglementaires en vigueur. Ce mémoire examine comment la mise en place de ce système peut transformer la gestion des risques informatiques à l'UAC, en évaluant l'efficacité, les défis et les bénéfices de cette initiative.

Organisation du mémoire

Le travail est structuré en trois chapitres. Le chapitre 1 présente une revue de littérature concernant la problématique. Nous abordons ensuite dans le chapitre 2 la question des choix techniques ainsi que les étapes de conception de notre projet puis dans le chapitre 3 nous présenterons notre solution et les résultats obtenus

Chapitre 1

Revue de littérature

Introduction

Pour mener à bien ce travail, il est nécessaire d'effectuer une collecte d'informations afin de prendre connaissance des notions et termes techniques nécessaires à la mise en place de notre solution. Dans ce chapitre, nous allons définir les concepts clés, explorer les différentes technologies et outils de gestion de logs disponibles, et sélectionner la solution la plus appropriée pour la mise en place d'un système de gestion de logs pour les serveurs de l'Université d'Abomey-Calavi.

1.1 Introduction à la gestion de logs

1.1.1 Définition du logging

Le logging, également appelé journalisation en français, est le processus par lequel des événements, des actions ou des transactions sont enregistrés dans un système informatique ou une application. Ces enregistrements, appelés logs, fournissent un historique détaillé des activités qui se sont produites, ce qui permet aux administrateurs, aux développeurs ou aux analystes de comprendre et d'analyser le comportement d'un système.

1.1.2 Définition d'un système de logging

Un système de logging est un ensemble de processus qui permet de collecter, analyser et visualiser les logs générés par différents services et applications. Il facilite la détection d'anomalies, la surveillance en temps réel et la gestion des incidents. Parmi les solutions populaires, la stack ELK (Elasticsearch, Logstash, Kibana) est largement utilisée pour ingérer et analyser les logs. Jeanne IDOLEKE, dans son mémoire "Gestion centralisée des logs avec ELK (Elasticsearch, Logstach et Kibana)", défend cette approche pour sa capacité à centraliser et visualiser les logs efficacement. Dans notre étude, nous optons pour une solution alternative avec OpenSearch, Fluentd et OpenSearch Dashboards, offrant des avantages similaires, notamment en termes de scalabilité et de flexibilité.

1.1.3 Fonctionnement d'un Système de Logging

Le fonctionnement d'un système de logging peut être divisé en plusieurs étapes clés, qui couvrent la collecte des logs, leur traitement, leur stockage, et leur analyse. Voici un aperçu de chacune de ces étapes pour mieux comprendre comment ces systèmes fonctionnent :

Collecte des Logs : La première étape dans un système de logging est la collecte des données de logs. Ces données peuvent provenir de diverses sources telles que des applications logicielles, des systèmes d'exploitation, des serveurs web, des bases de données, des équipements réseau, et d'autres composants informatiques. Les logs sont généralement générés automatiquement par les logiciels et les systèmes en réponse à des événements spécifiques. Ils peuvent inclure des informations telles que des horodatages, des types d'événement, des messages d'erreur, des statuts de transaction, et d'autres détails pertinents.

Transport des Logs : Une fois collectés, les logs doivent souvent être transportés vers un système central de gestion de logs. Ce transport peut être réalisé via des protocoles réseau sécurisés tels que TLS (Transport Layer Security) pour assurer que les données de logs restent intactes et sécurisées pendant la transmission. Le transport peut aussi intégrer un processus de filtrage ou d'agrégation pour réduire le volume de données envoyées à un serveur central.

Stockage des Logs : À leur arrivée dans le système de gestion de logs central, les données doivent être stockées de manière sécurisée et efficiente. Le stockage peut être réalisé sur des disques durs, des bases de données, ou des solutions de stockage en cloud. Le système de stockage doit être capable de gérer de grands volumes de données tout en permettant un accès rapide et efficace pour les requêtes et les analyses. Les pratiques de gestion des données, telles que la rotation des logs et l'archivage, sont également importantes pour gérer l'espace de stockage et assurer la conformité avec les politiques de rétention des données.

Analyse des Logs : L'analyse des logs est cruciale pour extraire des informations utiles à partir des grandes quantités de données collectées. Cette étape peut impliquer l'utilisation de logiciels d'analyse de logs, de systèmes de gestion d'événements et d'informations de sécurité (SIEM), ou d'autres outils d'analytique avancée. L'analyse peut inclure la détection d'anomalies, la corrélation d'événements pour identifier des menaces potentielles, la génération d'alertes en temps réel, et la production de rapports pour les audits et les revues de conformité.

Réponse et Correction : Enfin, le système de logging peut être intégré à des procédures de réponse aux incidents pour permettre une action rapide en cas de détection d'un problème. Cela peut inclure des notifications automatiques, des scripts de correction, ou des intégrations avec d'autres systèmes de gestion informatique pour résoudre les problèmes identifiés par les analyses de logs.

1.1.4 Gestion Actuelle des Logs à l'UAC

À l'Université d'Abomey-Calavi (UAC), la gestion des logs reste embryonnaire et n'exploite pas pleinement les potentialités qu'un tel système pourrait offrir. Actuellement, les données de journaux générées par les différents serveurs et infrastructures sont rarement collectées de manière centralisée. Ces logs, bien qu'essentiels, sont souvent dispersés, stockés localement sur les machines sources, et accessibles uniquement sur demande, ce qui rend leur exploitation difficile.

L'absence d'un système structuré pour la gestion des logs complique la surveillance proactive des systèmes et limite la capacité à détecter rapidement les anomalies ou les incidents de sécurité. Les équipes techniques, bien qu'engagées, manquent parfois des outils nécessaires pour diagnostiquer efficacement les pannes ou retracer les événements critiques. Cela engendre des pertes de temps importantes et des risques accrus pour la sécurité des données et des services numériques.

1.1.5 Pourquoi la Gestion des Logs est Cruciale pour l'UAC

Sans système de gestion des logs, l'UAC ne peut détecter rapidement les intrusions ou activités suspectes. Cela expose ses systèmes à des risques majeurs de cyberattaques. Aussi,lors de pannes, les équipes techniques perdent un temps précieux à chercher les causes sans outils ni logs centralisés, prolongeant les interruptions de service.De même, l'absence de logs fiables empêche de retracer les incidents. Cela compromet la crédibilité de l'UAC et l'expose à des risques juridiques. Les logs offrent des pistes pour optimiser les systèmes et prévenir les pannes. Leur absence prive l'UAC d'améliorations potentielles et engendre des coûts inutiles.

1.2 Utilité et Objectifs d'un Système de Logging

1.2.1 Importance de la Surveillance et de la Gestion des Logs

Pour l'UAC, la geston des logs est essentielle dans la protection des ressources de l'université. En surveillant en continu les activités des systèmes et en enregistrant les événements, elle permet de détecter rapidement les menaces potentielles, telles que les intrusions ou les comportements anormaux. Cette traçabilité offre aux équipes informatiques les moyens d'identifier et d'analyser efficacement les incidents pour intervenir avant qu'ils n'entraînent des perturbations majeures. De plus, le logging contribue à une surveillance proactive des ressources dispersées de l'UAC, garantissant leur disponibilité et leur sécurité.

1.3 Types et Approches de Systèmes de Logging

1.3.1 Logging Local vs Logging Centralisé

Logging Local : Le logging local stocke les données directement sur chaque serveur ou appareil. À l'UAC, cette méthode permet un accès rapide en cas de problème sur un serveur spécifique. Cependant, elle devient inefficace lorsque les logs sont dispersés sur plusieurs serveurs, rendant difficile une vue d'ensemble de l'infrastructure.
Logging Centralisé : Avec le logging centralisé, toutes les données sont collectées et stockées dans une plateforme unique. Pour l'UAC, cette approche améliorerait la visibilité globale, faciliterait la détection des incidents et simplifierait l'analyse forensique. Elle nécessite toutefois des investissements pour sécuriser le transfert et le stockage des données, mais les bénéfices en termes de réactivité et de gestion proactive des risques en font une solution idéale pour une institution aux ressources limitées.

1.4 Solutions Actuelles et Limitations

1.4.1 Revue des Outils de Logging

L'UAC peut s'appuyer sur des outils modernes tels qu'Elasticsearch, Logstash, Graylog, reconnus pour leur capacité à gérer de grandes quantités de données en temps réel. Ces solutions permettent de collecter, filtrer et analyser les logs de manière centralisée, offrant des tableaux de bord intuitifs pour surveiller l'état des systèmes et détecter les incidents.

1.4.2 Défis et Limitations des Solutions Actuelles

Malgré leur puissance, ces outils nécessitent des compétences techniques avancées pour l'installation et la maintenance. Les ressources nécessaires, en termes de CPU, mémoire, et stockage, peuvent représenter un défi pour l'UAC, où les infrastructures sont souvent limitées. La sécurisation des données lors du transfert et du stockage reste également une contrainte majeure, particulièrement dans le contexte académique où la sensibilité des informations est critique.

1.5 Notre Solution

1.5.1 Description de la Solution Proposée

Pour répondre aux besoins spécifiques de l'UAC, notre solution repose sur l'intégration d'OpenSearch, Fluentd et OpenSearch Dashboards :

Fluentd collecte les logs des différents serveurs de l'UAC, en les filtrant et en les normalisant pour réduire la complexité. OpenSearch par contre stocke et analyse les données de logs, offrant des capacités avancées pour détecter les anomalies et produire des rapports détaillés. OpenSearch Dashboards permet de visualiser les données via des tableaux de bord, facilitant la surveillance en temps réel et l'analyse approfondie.

Conclusion

Ce chapitre a abordé l'importance cruciale des systèmes de gestion de logs dans la surveillance et la sécurité informatique. Après avoir exploré les défis posés par les solutions actuelles comme Elasticsearch, Logstash, Graylog, notamment en termes de complexité et de gestion des ressources, nous avons proposé une solution intégrée utilisant OpenSearch, Fluentd, et OpenSearch Dashboard. Cette solution vise à simplifier la gestion des logs tout en améliorant la sécurité et l'efficacité, répondant ainsi mieux aux besoins des organisations modernes.

Chapitre 2

Choix de la solution et présentation du matériel

Introduction

Ce chapitre se concentre sur la sélection rigoureuse et la présentation détaillée du système de logging adapté aux besoins des serveurs de l'Université d'Abomey-Calavi (UAC). L'objectif est de détailler les critères de choix qui ont guidé cette sélection et de fournir une description exhaustive de la solution retenue, en soulignant comment elle répond spécifiquement aux exigences de l'UAC.

2.1 Critères de Choix de la Solution

2.1.1 Exigences Fonctionnelles

Pour l'Université d'Abomey-Calavi , qui gère une volumétrie importante de données due à ses nombreuses activités académiques et administratives, le système de logging doit être particulièrement robuste et sophistiqué. Il est crucial que ce système soit capable de capturer de manière exhaustive et précise toutes les activités informatiques sur les serveurs de l'université, garantissant qu'aucun événement significatif ne soit omis. Le système doit également être équipé de fonctionnalités avancées de filtrage et d'agrégation pour gérer et réduire efficacement le volume des données collectées, tout en conservant les informations pertinentes pour l'analyse. L'analyse en temps réel est essentielle pour identifier rapidement toute anomalie ou activité suspecte, permettant ainsi une intervention proactive pour sécuriser les systèmes. En outre, il est nécessaire que le système propose des options flexibles pour la génération de rapports et la configuration d'alertes, répondant aux besoins spécifiques des différents départements de l'UAC. Enfin, il est impératif que l'accès au système soit rigoureusement contrôlé et que les permissions soient gérées avec précision pour assurer la sécurité et la conformité des données traitées.

2.1.2 Exigences Techniques

La mise en place d'un système de logging pour les serveurs de l'UAC doit prendre en compte plusieurs contraintes techniques inhérentes à l'environnement informatique de l'université. Tout d'abord, la diversité des systèmes et des applications utilisés nécessite une solution de logging compatible avec une grande variété de plateformes, y compris les systèmes d'exploitation Windows, Linux et les applications spécifiques à l'enseignement et à l'administration. De plus, l'architecture distribuée des serveurs de l'UAC implique que le système de logging doit être

capable de collecter des données à partir de différents emplacements géographiques et de les agréger de manière centralisée pour une analyse efficace. Cette architecture distribuée soulève également des défis en termes de latence et de bande passante pour la transmission des logs vers le système central. Par conséquent, la solution de logging doit être optimisée pour minimiser l'impact sur les performances du réseau et des serveurs. En outre, l'évolutivité est une considération importante étant donné la croissance potentielle du nombre de serveurs et de volumes de données à gérer à l' UAC. La solution choisie doit être capable de s'adapter facilement à ces évolutions sans compromettre sa fiabilité et ses performances. Enfin, la sécurité est une priorité absolue, et le système de logging doit être conforme aux normes de sécurité les plus strictes pour protéger les données sensibles de l'université contre les menaces internes et externes.

2.2 Options d'Implémentation

2.2.1 Étude des Solutions Existantes

Syslog est l'un des protocoles de logging les plus anciens et les plus largement adoptés. Il permet aux appareils de réseau, serveurs, et autres systèmes d'envoyer leurs messages de log à un serveur de logging centralisé. Syslog est hautement configurable et supporte une gestion efficace des logs à travers différents systèmes et applications.

Logstash, faisant partie de la suite Elastic (anciennement ELK Stack), Logstash est un pipeline de traitement de données côté serveur qui ingère les données de diverses sources, les transforme, et les envoie à un stockage comme Elasticsearch. Logstash est particulièrement apprécié pour sa flexibilité et sa capacité à filtrer et enrichir les logs avant de les stocker.

Fluentd est un collecteur de données open source qui unifie la collecte et la consommation de logs pour une meilleure compréhension de l'environnement informatique. Il est conçu pour être léger et facile à intégrer, supportant les architectures de microservices et les environnements de conteneurisation comme Kubernetes.

Graylog est une solution de gestion de logs centralisée qui offre un stockage puissant des logs, une interface utilisateur intuitive pour la visualisation des données, et des capacités d'alerte avancées. Il est souvent choisi pour sa facilité d'utilisation et ses fonctionnalités complètes de recherche et d'analyse.

Splunk est une plateforme complète qui non seulement gère les logs mais offre également des capacités d'analyse de données opérationnelles. Splunk est réputé pour sa capacité à analyser de grands volumes de données en temps réel, fournissant des insights précieux pour la prise de décision stratégique.

ELK Stack/Elastic Stack Composé d'Elasticsearch, Logstash, et Kibana, l'ELK Stack est l'une des solutions les plus populaires pour la surveillance et l'analyse des logs en temps réel. Elasticsearch sert de moteur de recherche et de base de données, Logstash comme pipeline de traitement de logs, et Kibana offre une interface de visualisation des données.

OpenSearch, un fork d'Elasticsearch et de Kibana, est un moteur de recherche et d'analyse distribué open source conçu pour un large éventail d'applications, y compris la gestion de logs. OpenSearch Dashboard remplace Kibana pour offrir des visualisations et une exploration des données.

2.2.2 Comparaison des solutions

Critère	OpenSearch/ Fluentd/Dash-boards	ELK	Splunk	Graylog
Coût	Totalement open-source, sans frais de licence.	Open-source partiel, fonction-nalités avancées payantes.	Très coûteux, basé sur le volume.	Gratuit en ver-sion de base, mais fonction-nalités avancées payantes.
Flexibilité	Fluentd est très flexible avec des plugins pour di-vers scénarios.	Logstash est puis-sant mais lourd à configurer.	Très flexible, mais fonctionnalités li-mitées sans paie-ment.	Moins flexible que Fluentd.
Scalabilité	Scalabilité élevée grâce à l'architec-ture OpenSearch et Fluentd.	Très scalable, mais Logstash peut de-venir un goulot d'étranglement.	Scalabilité élevée mais coûteuse.	Scalabilité cor-recte, mais moins performante que ELK ou Open-Search.
Simplicité d'installation	Facile à installer et bien documenté.	Installation bien documentée mais Logstash peut être complexe.	Facile, mais nécessite une infrastructure lourde.	Relativement simple, mais fonc-tionnalités avan-cées complexes à configurer.
Communauté et support	Communauté ac-tive avec le sou-tien d'AWS pour OpenSearch.	Large commu-nauté, mais certaines parties nécessitent un support payant.	Support commer-cial excellent mais coûteux.	Communauté ac-tive, mais plus pe-tite qu'ELK.
Adaptation académique	Idéal grâce à l'ab-sence de coûts et à la simplicité de gestion.	Bon mais peut devenir coûteux pour des fonction-nalités avancées.	Peu adapté à cause des coûts élevés.	Convient pour un projet simple, mais limité en personnalisation.

TABLE 2.1 : Comparaison des systèmes de gestion de logs : OpenSearch, Fluentd, ELK, Splunk et Graylog

Explication du Tableau
Type : Catégorise le logiciel en tant que protocole, pipeline, collecteur ou plateforme complète
Flexibilité : Indique à quel point la solution est adaptable aux besoins spécifiques d'une organisation.
Scalabilité : Aptitude du système à s'adapter à l'augmentation de la charge sans dégradation significative des performances.
Coût : Indique si le logiciel est disponible gratuitement, ou s'il offre des options payantes pour des fonctionnalités avancées.
Communauté et Support : Taille et activité de la communauté autour du logiciel, ce qui peut influencer le support et les ressources disponibles.

2.3 Choix de la solution

2.3.1 Justification du choix

Lorsqu'il s'agit de choisir une solution de logging pour les serveurs de l'Université d'Abomey-Calavi (UAC), plusieurs facteurs cruciaux ont été pris en compte pour justifier la sélection d'OpenSearch, appuyée par Fluentd et OpenSearch Dashboard. Ce choix stratégique est fondé sur les besoins spécifiques de l'UAC en matière de traitement et d'analyse de données, ainsi que sur les avantages techniques et opérationnels offerts par OpenSearch.

OpenSearch, Fluentd et OpenSearch Dashboards sont privilégiés pour leur nature entièrement open-source, ce qui les rend exempts de coûts de licence, contrairement à ELK, où certaines fonctionnalités essentielles (comme la sécurité avancée ou les alertes) sont payantes, et à Splunk, dont les coûts élevés sont souvent prohibitifs pour des projets à budget limité. Fluentd, plus flexible et léger que Logstash ou Graylog, permet de collecter, transformer et enrichir les logs avec une simplicité remarquable grâce à ses nombreux plugins. OpenSearch offre une scalabilité comparable à Elasticsearch tout en étant libre de restrictions de licence, et OpenSearch Dashboards propose une interface intuitive pour l'analyse des données. Ensemble, ces outils forment une solution performante, économique et adaptée aux besoins d'un projet académique ou à budget restreint, sans sacrifier la qualité ni les fonctionnalités.

2.4 Architecture et Composants du Système

2.4.1 Architecture Globale

L'architecture de ce système de logging est conçue pour optimiser la collecte, le stockage, l'analyse et la visualisation des logs à travers l'infrastructure informatique de l'UAC. Elle s'articule autour de plusieurs couches clés :

Couche de Collecte de Données : Fluentd est utilisé comme agent de collecte de logs. Il est déployé sur chaque serveur et dispositif réseau au sein de l'UAC pour intercepter et collecter les logs générés par les systèmes et applications. Fluentd est choisi pour sa légèreté, sa flexibilité dans la gestion des différents formats de logs et sa capacité à transmettre les données de manière efficace.

Couche de Traitement et d'Agrégation : Les logs collectés sont envoyés à OpenSearch, où ils sont indexés et stockés. OpenSearch agit comme la couche de traitement central, offrant des capacités puissantes d'indexation et de recherche nécessaires pour gérer de grands volumes de données. Cette couche traite les logs pour les rendre consultables et les agrège pour une analyse plus approfondie.

Couche de Visualisation : OpenSearch Dashboard est utilisé pour la visualisation des logs. Cette plateforme permet aux administrateurs et aux techniciens de l'UAC de créer des tableaux de bord dynamiques qui affichent des analyses en temps réel et des résumés des données de logs. Cela aide à surveiller les performances du système, à détecter rapidement les problèmes, et à prendre des décisions basées sur les données observées.

La communication entre Fluentd, OpenSearch et OpenSearch Dashboard est sécurisée à travers des protocoles de cryptage pour garantir que les données sont protégées pendant la transmission. De plus, des stratégies de redondance et de sauvegarde sont mises en place pour assurer la haute disponibilité et la résilience du système de logging en cas de défaillance.

Des politiques de sécurité rigoureuses sont appliquées à tous les niveaux de l'architecture pour protéger contre les accès non autorisés et les attaques externes. Cela inclut l'utilisation de pare-feu, l'authentification forte, le contrôle d'accès basé sur les rôles et les audits réguliers du système.

2.5 Intégration et Déploiement

- **Évaluation des Besoins et des Ressources**
  Examiner les exigences en termes de capacité de stockage, de puissance de traitement et de bande passante réseau.
  S'assurer que l'infrastructure actuelle peut supporter l'ajout de nouvelles applications de logging.
- Installation des Dépendances Nécessaires Installer Java pour OpenSearch
  S'assurer que toutes les autres dépendances logicielles, comme Ruby pour Fluentd , sont installées et configurées correctement.
- **Déploiement de Fluentd** Télécharger et installer Fluentd sur chaque serveur ou dispositif qui doit collecter les logs.
  Configurer Fluentd pour collecter les logs de différentes sources (serveurs, applications).
- **Configuration des Plugins** Configurer les plugins nécessaires pour permettre à Fluentd de transmettre les données collectées à OpenSearch.
- **Déploiement d'OpenSearch** Installer OpenSearch sur les serveurs dédiés en suivant les instructions officielles pour garantir une installation sécurisée.
- **Configuration du Cluster OpenSearch** Configurer les nœuds OpenSearch pour former un cluster, si nécessaire, pour améliorer la redondance et la disponibilité des données. Paramétrer les réglages de performance, tels que la gestion de la mémoire et le partitionnement des données.
- **Déploiement d'OpenSearch Dashboard** Installer OpenSearch Dashboard sur un serveur accessible aux administrateurs système.
- Configuration de la Connexion Configurer OpenSearch Dashboard pour se connecter au cluster OpenSearch, permettant ainsi la visualisation et l'analyse des logs.
- **Intégration et Tests** Intégrer la solution de logging avec les systèmes de surveillance et d'alerte existants à l'UAC.
  S'assurer que Fluentd et OpenSearch fonctionnent de manière synchrone pour collecter, stocker et indexer les logs.
- **Tests de Fonctionnalité** Effectuer des tests complets pour vérifier que les logs sont correctement collectés, transmis à OpenSearch, stockés, indexés et disponibles via OpenSearch Dashboard.
  Tester les performances du système sous différentes charges pour s'assurer qu'il répond aux exigences de rapidité et d'efficacité.
- **Mise en Production et Surveillance** Une fois les tests validés, mettre le système en production.
  Former les utilisateurs finaux, comme les administrateurs et le personnel de sécurité, sur l'utilisation d'OpenSearch Dashboard.

- **Surveillance Continue** Surveiller le système de logging pour s'assurer qu'il continue de fonctionner comme prévu et pour détecter tout problème potentiel.
  Maintenance et Mises à Jour : Planifier des maintenances régulières et des mises à jour du système pour assurer la sécurité et l'efficacité continues. En suivant ces étapes détaillées, l'UAC peut s'assurer que le système de logging est correctement intégré dans son environnement informatique existant, offrant une solution robuste et efficace pour la gestion des logs.

Conclusion

Dans ce chapitre, nous avons passé en revue les différentes solutions de logging disponibles sur le marché, en analysant leurs avantages et inconvénients. Nous avons finalement justifié le choix d'OpenSearch comme la solution la plus adaptée pour améliorer la gestion des logs des serveurs de l'Université d'Abomey-Calavi, en raison de sa robustesse, de ses capacités d'indexation et d'analyse avancées, ainsi que de sa flexibilité dans un environnement universitaire en constante évolution.

Chapitre 3

Implémentation et collecte des logs avec opensearch

Introduction

Ce chapitre est consacré à l'implémentation et aux diverses expériences effectuées pour démontrer l'efficacité et le bon fonctionnement de notre solution de gestion des logs. Nous aborderons ici les étapes clés de l'installation, de la configuration et des tests effectués sur OpenSearch.

3.1 Environnement de travail

Pour réaliser ce projet, nous avons utilisé un serveur EC2 sur Amazon Web Service fonctionnant sous le système d'exploitation Ubuntu. Ce choix a été motivé par la robustesse d'Ubuntu, sa large adoption dans le milieu des serveurs et sa compatibilité avec les outils nécessaires pour la mise en œuvre de notre solution.

3.2 Installation et configuration de Opensearch

Après avoir téléchargé le package 'opensearch-2.16.0-linux-x64.deb' à partir du site officiel d'OpenSearch, nous avons procédé à l'installation en décompressant l'archive. Cette étape initiale est cruciale pour garantir que tous les fichiers nécessaires soient disponibles sur notre machine.

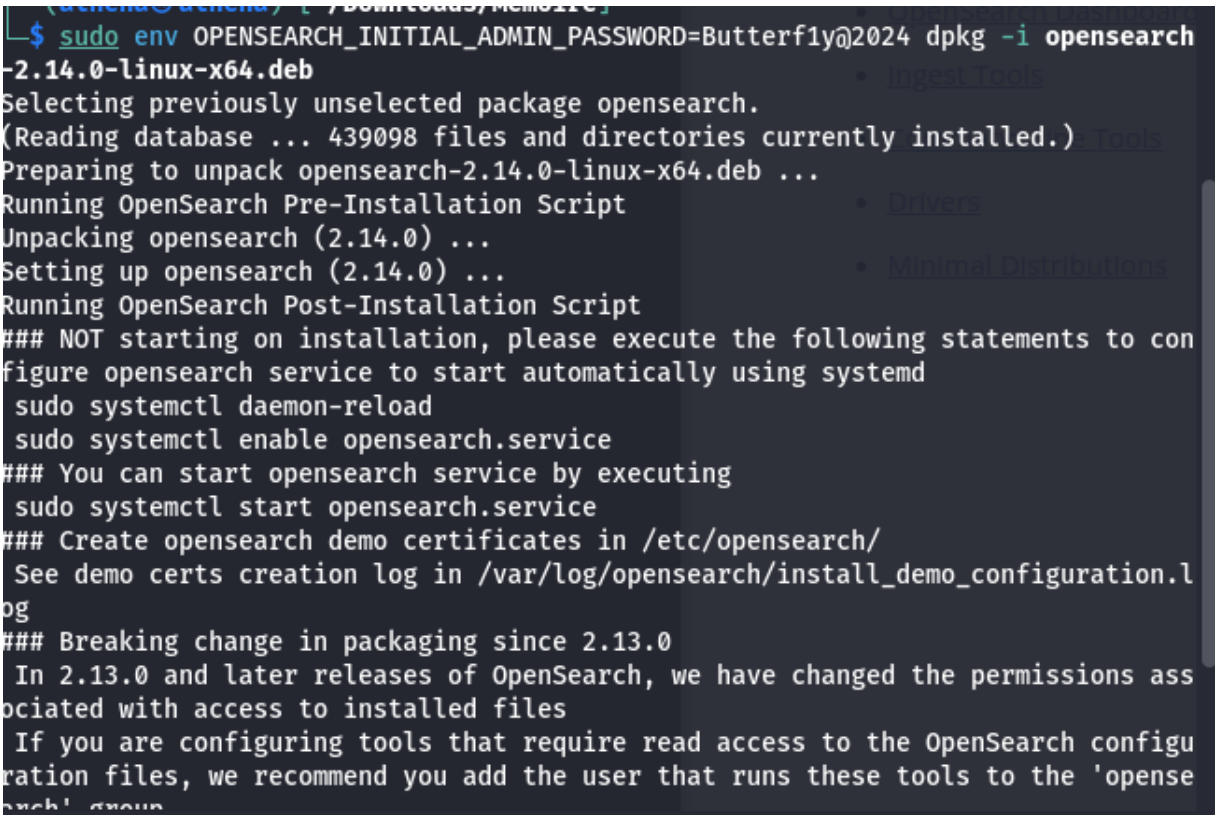


FIGURE 3.1 : Configuration du mot de passe administrateur pour OpenSearch

Démarrons OpenSearch et vérifions qu'il a été lancé correctement. Lors de cette étape, il est essentiel de surveiller les journaux pour s'assurer qu'aucune erreur ne se produit et que le service est opérationnel.

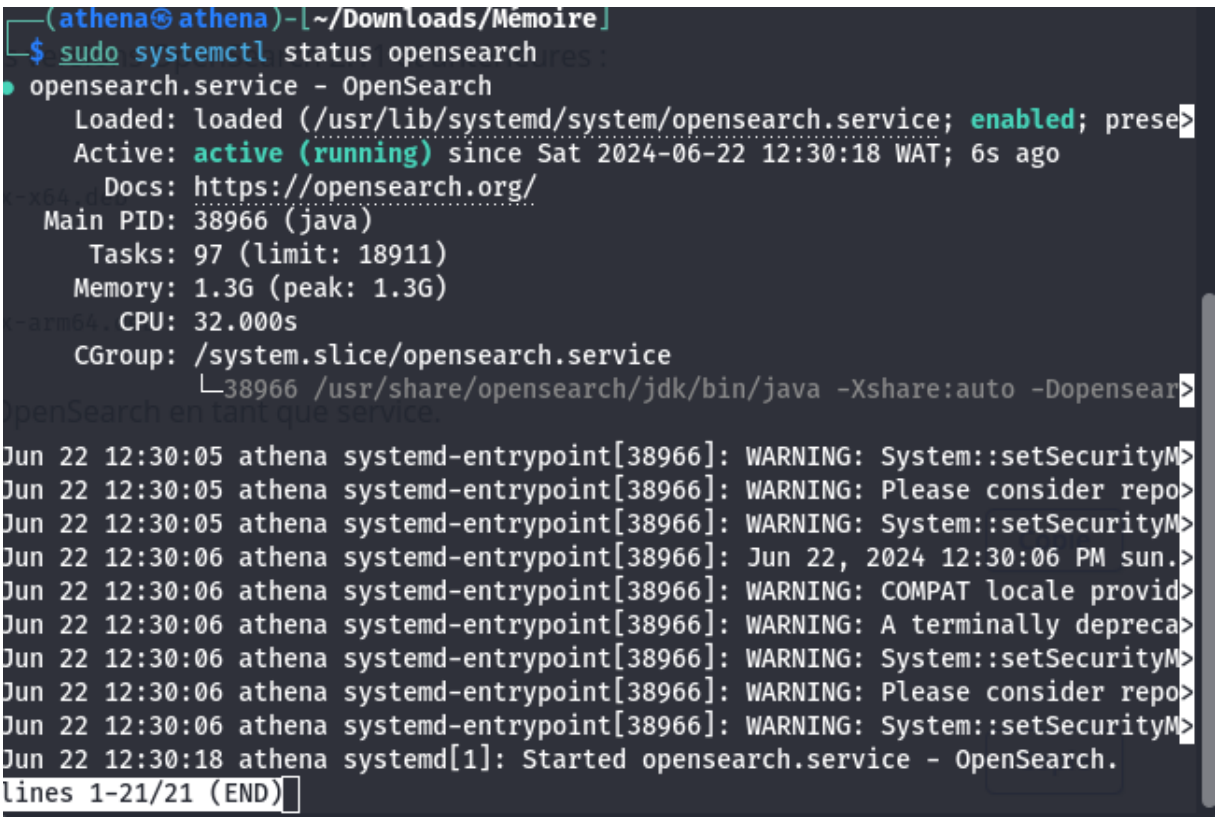


FIGURE 3.2 : Configuration du mot de passe administrateur pour OpenSearch

On constate donc que le service OpenSearch fonctionne bien. Après vérification, il est clair que le démarrage a été réussi, permettant ainsi à toutes les fonctionnalités du système de devenir accessibles.
Nous avons ensuite vérifié le bon fonctionnement d'OpenSearch en lançant une requête vers le serveur sur le port 9200, qui est le port par défaut pour accéder à l'API REST d'OpenSearch. Cette étape est cruciale pour valider que le serveur répond correctement aux requêtes entrantes.

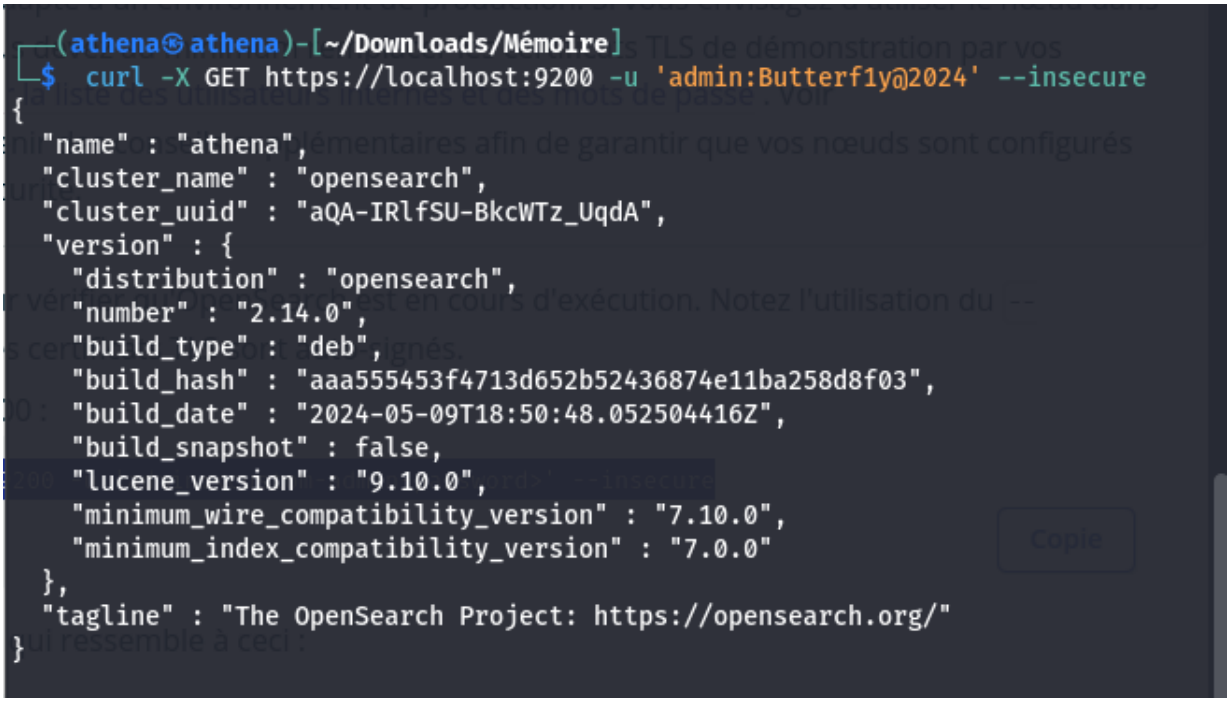


FIGURE 3.3 : Vérification du bon fonctionnement d'OpenSearch

Interrogeons maintenant le point de terminaison du plugin afin de vérifier que les plugins nécessaires sont correctement installés. Cela permet de garantir que nous pouvons tirer parti des fonctionnalités supplémentaires offertes par OpenSearch.

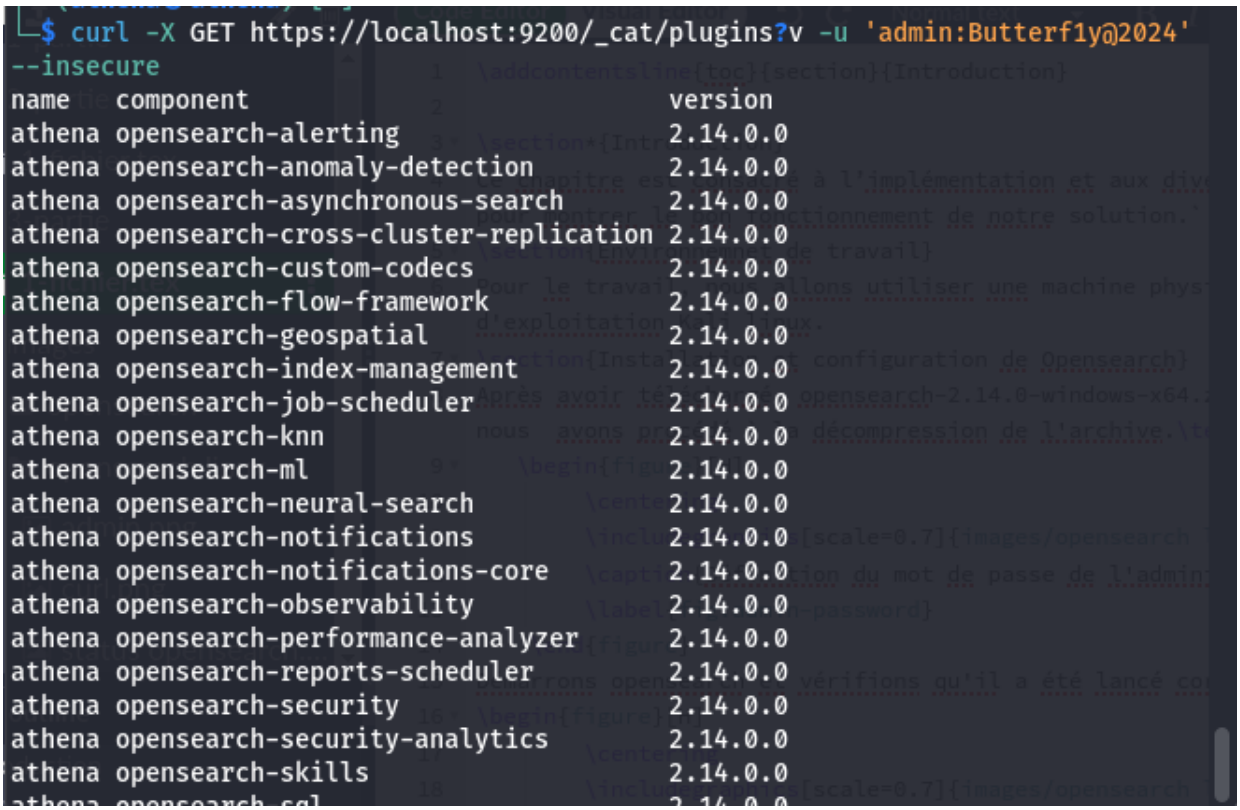


FIGURE 3.4 : Vérification de l'installation des plugins OpenSearch

Nous avons alors la liste des plugins installés, ce qui nous permet de confirmer que l'environnement est configuré selon nos besoins.

Nous allons lier OpenSearch à un réseau sur l'hôte. Cette liaison est essentielle pour que le serveur puisse communiquer correctement avec les autres services et utilisateurs sur le réseau. Pour ce faire, nous devons configurer le fichier `opensearch.yml` en y ajoutant les lignes suivantes :

- `network.host: 0.0.0.0`
- `discovery.type: single-node`
- `plugins.security.disabled: false`

Une bonne allocation de la mémoire est cruciale pour optimiser les performances d'OpenSearch. Pour cela, il est important de spécifier les tailles de tas JVM initiales et maximales dans le fichier `jvm.options`. En général, ces valeurs doivent être définies à la moitié de la mémoire système disponible. Nous allons définir les tailles de tas initiales et maximales à 4 Go. Cela permettra à OpenSearch de fonctionner de manière efficace, en utilisant judicieusement les ressources système.

Configuration du certificat TLS

Nous allons configurer le certificat TLS, qui permettra aux hôtes de s'identifier de manière sécurisée sur le réseau.

Dans le cadre de la sécurisation des communications entre OpenSearch, OpenSearch Dashboards et les utilisateurs, des certificats SSL/TLS sont indispensables. Pour cette installation, nous avons opté pour des certificats Let's Encrypt, qui offrent une solution de sécurisation gratuite et automatisée.

Types de certificats

Les certificats Let's Encrypt ont été choisis pour leur fiabilité et leur facilité d'utilisation. Ils disposent d'une durée de validité de trois mois, nécessitant un renouvellement régulier tous les trois mois. Ce processus de renouvellement peut être automatisé, simplifiant ainsi la gestion des certificats et garantissant une continuité dans la sécurité des communications.

Noms des certificats utilisés

Pour l'installation d'OpenSearch, les certificats suivants ont été employés :

- `SSLCertificateFile : fullchain.pem`
- `SSLCertificateKeyFile : privkey.pem`

Processus d'installation

Le processus d'installation a impliqué la génération des certificats via Let's Encrypt, permettant ainsi d'obtenir les fichiers nécessaires pour sécuriser les connexions entre les différents services.

Configuration dans OpenSearch

Les certificats ont été intégrés dans la configuration d'OpenSearch, garantissant que toutes les communications soient cryptées et sécurisées. Les fichiers de certificats ont été spécifiés dans le fichier de configuration d'OpenSearch, ce qui permet une mise en œuvre fluide des protocoles HTTPS.

Impact sur la sécurité L'utilisation de certificats Let's Encrypt a considérablement renforcé la sécurité de l'installation OpenSearch. Cela protège les données en transit contre les interceptions et assure aux utilisateurs que leurs communications sont sécurisées et fiables.

Défis rencontrés

Un défi potentiel réside dans le renouvellement des certificats tous les trois mois. Cependant, des stratégies peuvent être mises en place pour gérer cette tâche efficacement, minimisant ainsi les risques d'expiration des certificats et assurant une continuité dans la sécurité des communications.

Maintenant, configurons un utilisateur pour accéder à OpenSearch. Cela implique la création d'identifiants qui permettront de contrôler l'accès et d'assurer la sécurité des opérations au sein du système.

```
┌──(athena㉿athena)-[/usr/…/opensearch/plugins/opensearch-security/tools]
└─$ ./hash.sh
************************************************************************
** This tool will be deprecated in the next major release of OpenSearch **
** https://github.com/opensearch-project/security/issues/1755          **
************************************************************************
WARNING: nor OPENSEARCH_JAVA_HOME nor JAVA_HOME is set, will use /usr/bin/java
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
[Password:]
$2y$12$zikzXXEP7yizLYvDGPbqGeeZibpG/.ZdOhIWOlQJ51QSS2VZQ5RHe
```

FIGURE 3.5 : Configuration d'un utilisateur dans OpenSearch

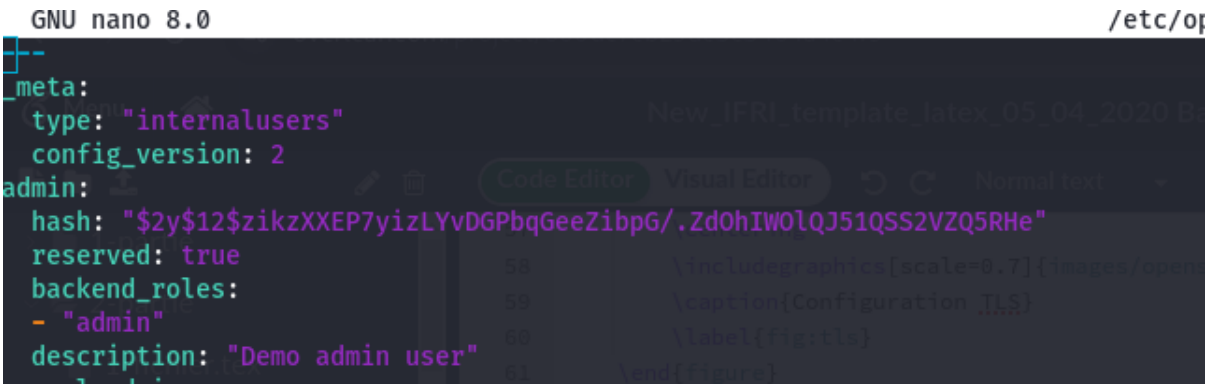


FIGURE 3.6 : Configuration d'un utilisateur dans OpenSearch

Nous allons maintenant redémarrer OpenSearch pour nous assurer qu'il fonctionne correctement avec toutes les configurations que nous avons appliquées. Ce redémarrage est une étape critique pour valider que toutes les modifications ont été prises en compte et que le service est opérationnel.

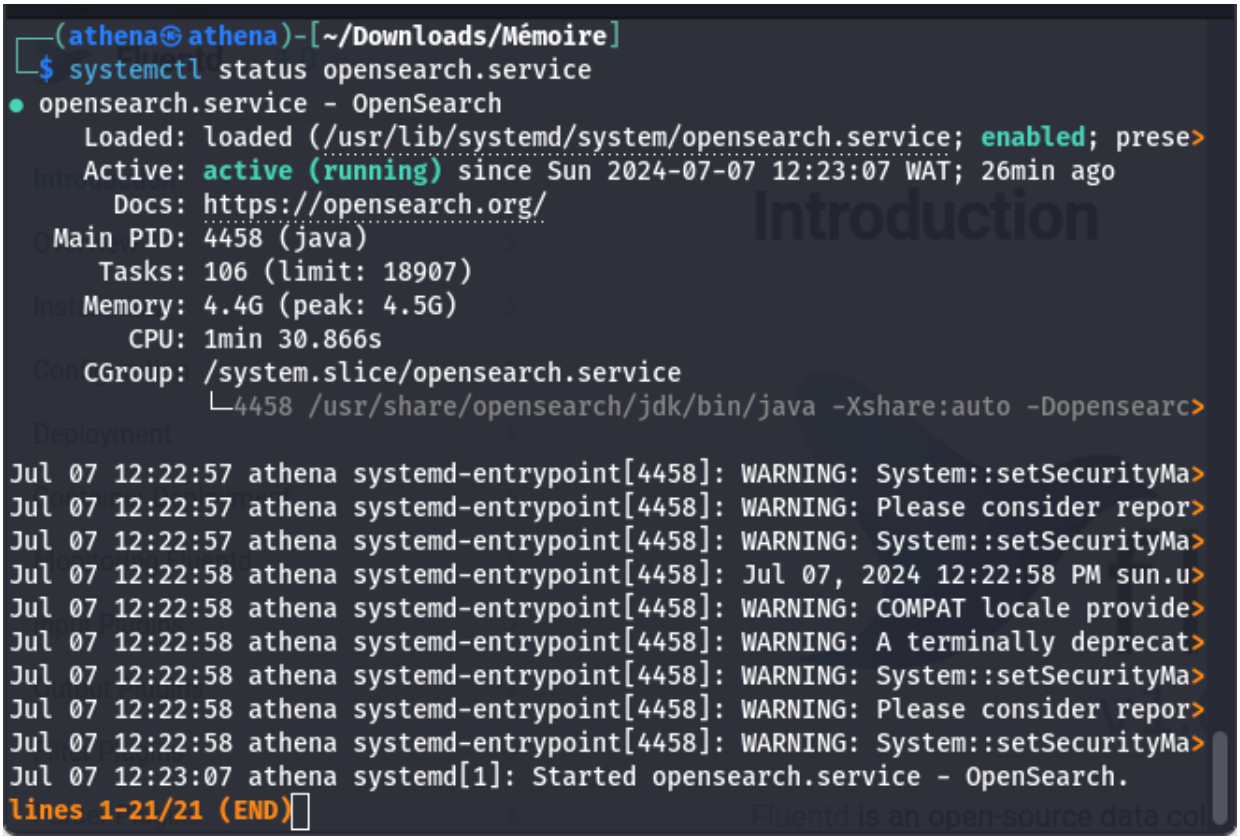


FIGURE 3.7 : État du service OpenSearch

Après le téléchargement du package, nous avons vérifié son fonctionnement en lançant des requêtes sur le port 9200 et en validant l'installation des plugins. Nous avons configuré l'interface réseau, ajusté les tailles de tas JVM, et mis en place un certificat TLS pour sécuriser les communications. Enfin, nous avons créé des utilisateurs et redémarré le service pour appliquer les modifications.

3.3 Configuration de fluentd

Nous allons procéder à l'installation de Fluentd, un outil essentiel pour la collecte et le traitement des logs. Son intégration dans notre système permettra une gestion efficace des données générées par les différents services.

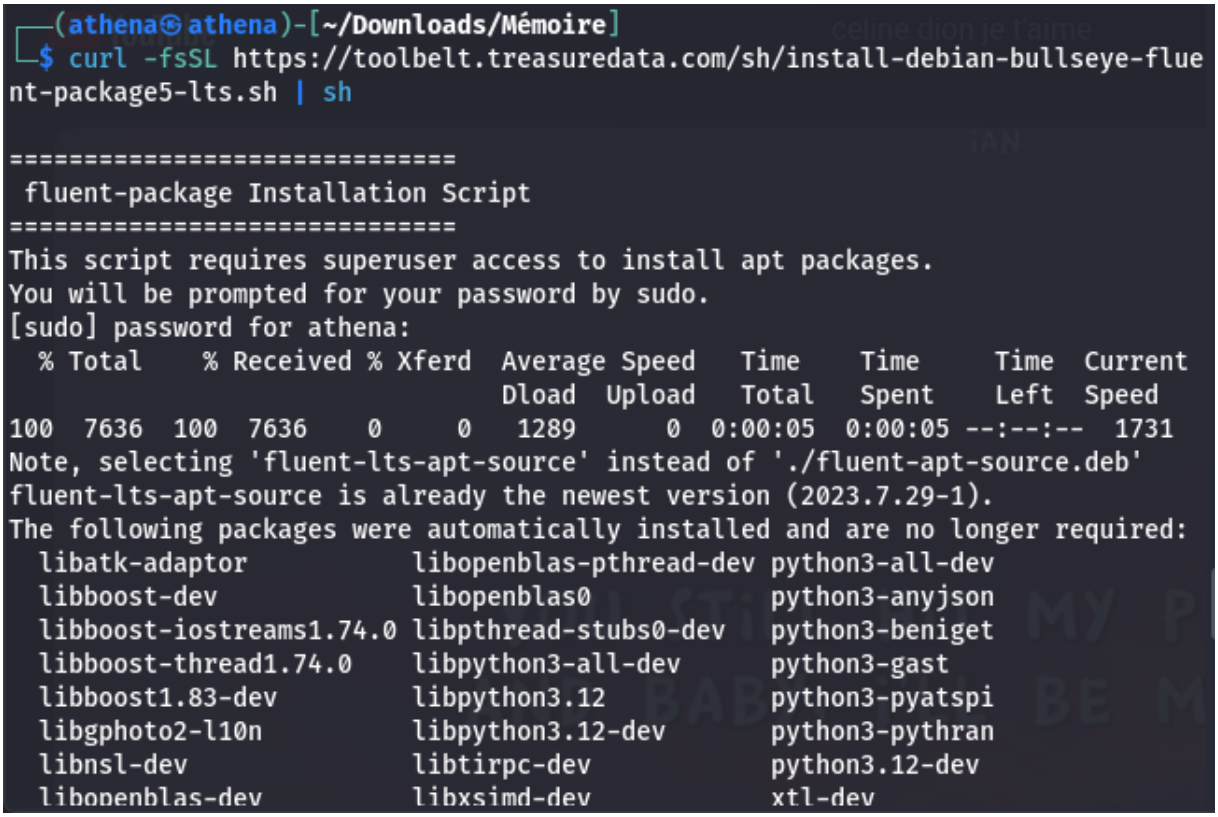


FIGURE 3.8 : Installation de Fluentd

Après l'installation, il est crucial de s'assurer que Fluentd fonctionne correctement. Nous allons vérifier son état pour confirmer que le service est opérationnel et prêt à être utilisé.

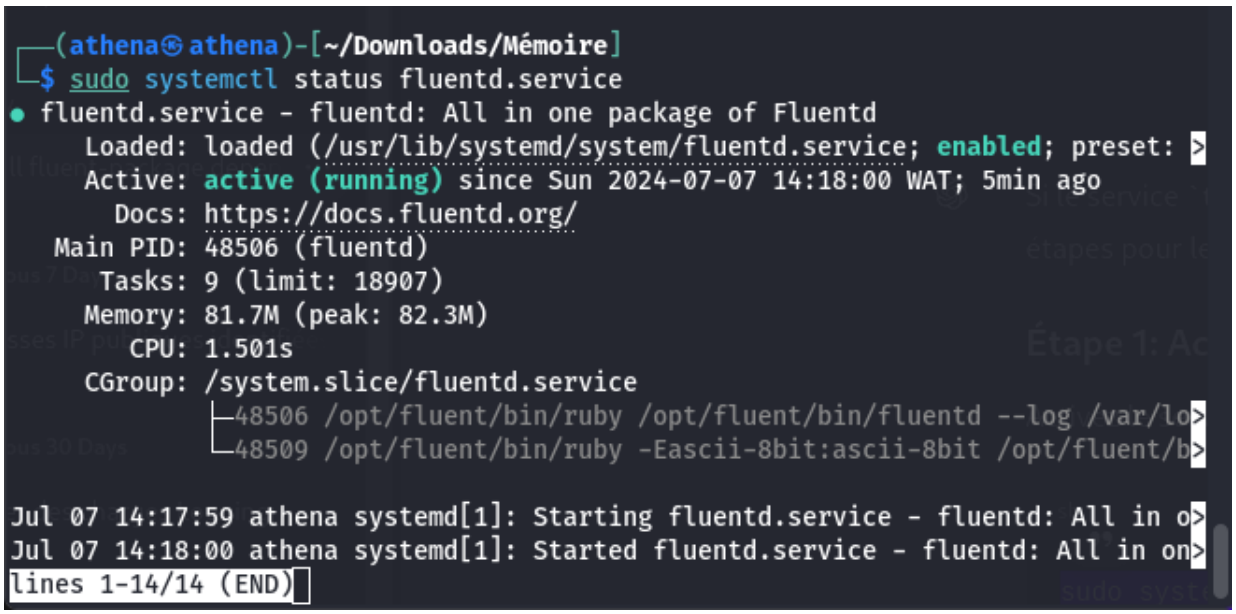


FIGURE 3.9 : État de l'installation de Fluentd

Pour tester le bon fonctionnement de Fluentd, nous allons publier un exemple d'enregistrement de journal en utilisant la commande curl. Cela nous permettra de vérifier que les données sont correctement traitées et acheminées par Fluentd.



FIGURE 3.10 : Test du fonctionnement de fluentd

3.4 Installation d'OpenSearch Dashboards

Pour commencer, nous allons télécharger et installer le paquet d'OpenSearch Dashboards. Cette étape est essentielle pour nous permettre d'accéder à une interface graphique conviviale qui facilitera la visualisation et l'analyse des données d'OpenSearch.
Une fois l'installation terminée, nous procéderons au redémarrage du service et à l'activation d'OpenSearch Dashboards. Cela permettra de s'assurer que toutes les configurations sont prises en compte et que le service est opérationnel.

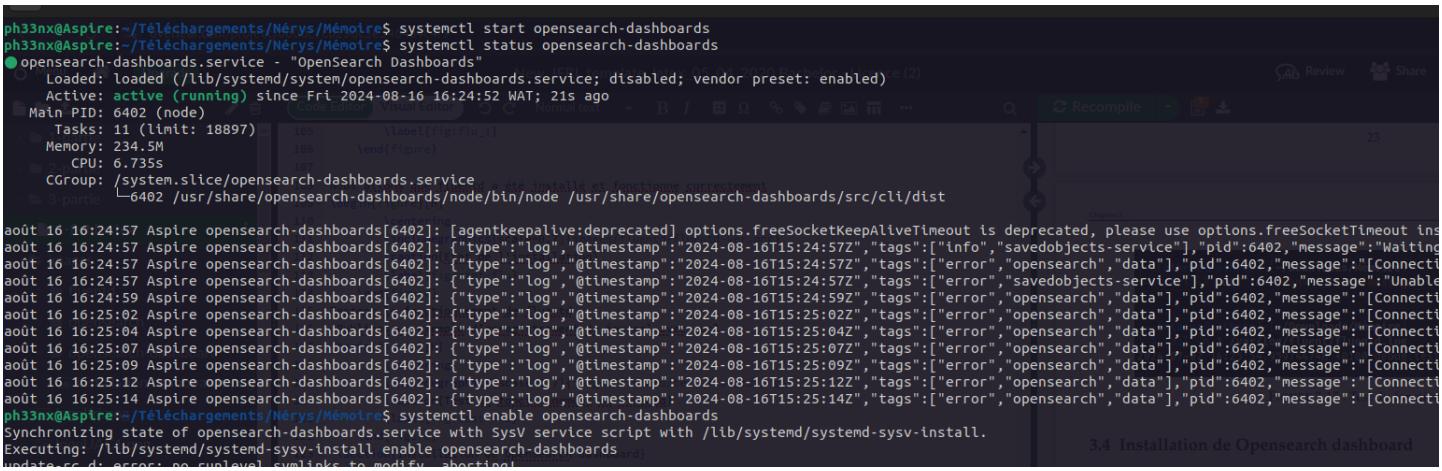


FIGURE 3.11 : Activation d'OpenSearch Dashboards

Après avoir activé le service, nous pouvons constater que tout fonctionne correctement. L'interface d'OpenSearch Dashboards est maintenant accessible, offrant ainsi une plateforme intuitive pour interagir avec nos données.

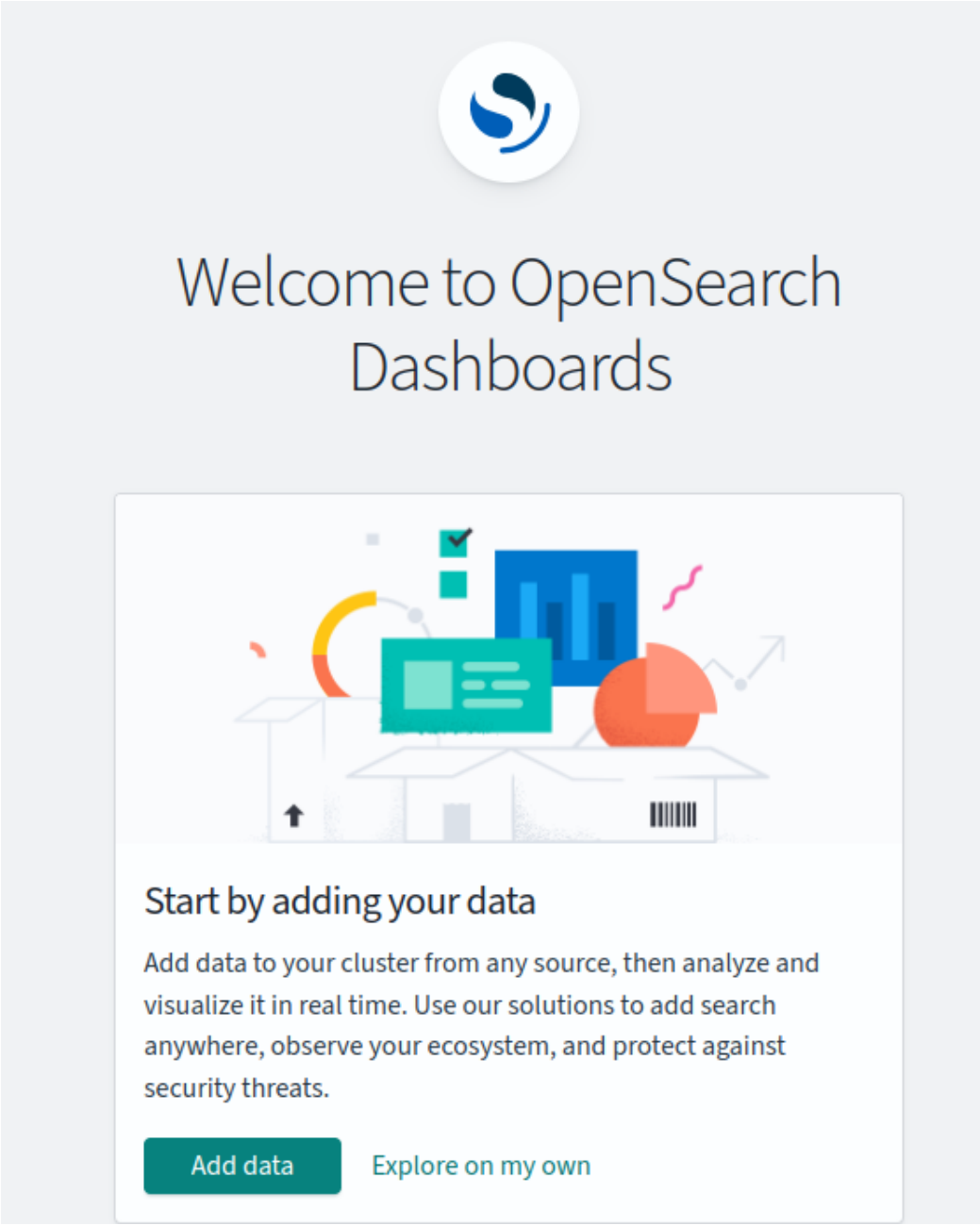


FIGURE 3.12 : Interface d'OpenSearch Dashboards

3.5 Mise en œuvre de la solution

Nous avons configuré OpenSearch, Fluentd, et OpenSearch Dashboards pour centraliser et gérer les logs Apache du site hébergé sur le serveur uac-serveur-log, accessible via l'adresse sécurisée logscenter.uac.bj. Afin de garantir la confidentialité et l'intégrité des communications entre le serveur et ses utilisateurs, nous avons mis en place des certificats SSL/TLS pour chiffrer toutes les transmissions de données. Cette configuration robuste assure non seulement la sécurité des échanges, mais également un accès fiable et protégé aux services de gestion des logs.

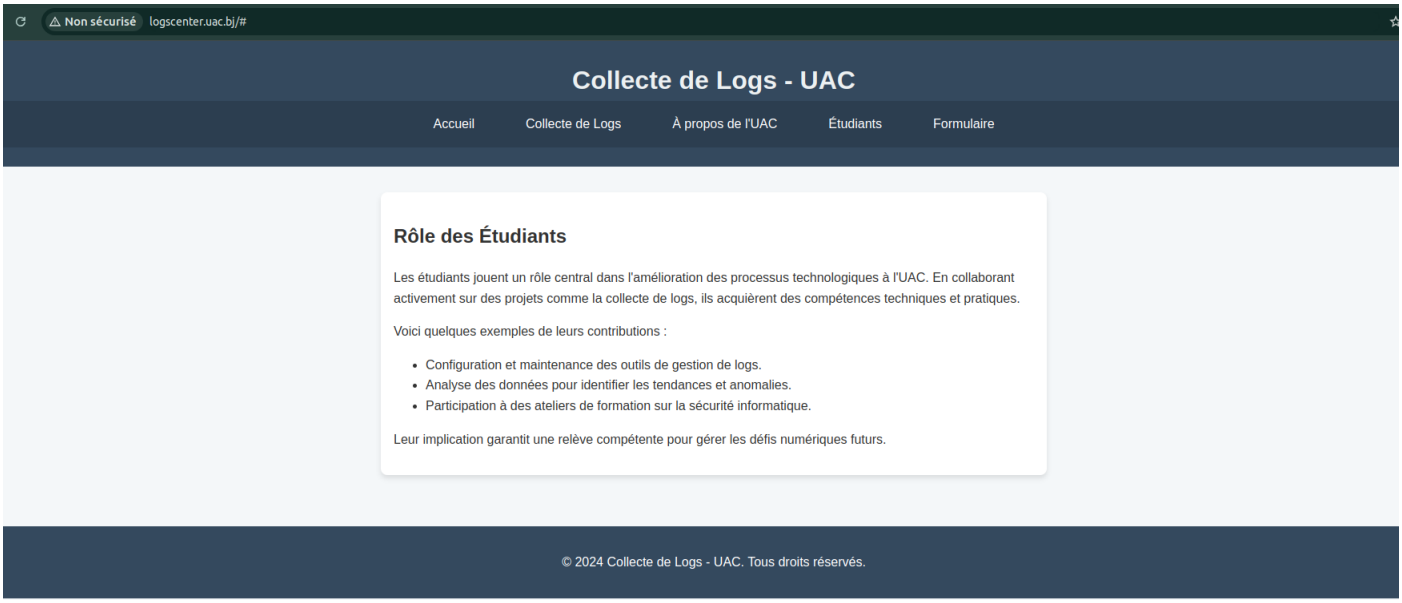


FIGURE 3.13 : Site web utilisé

L'interface d'OpenSearch Dashboards a été configurée pour être accessible à l'adresse https ://logscenter.uac.bj :5601. Nous collecterons les logs générés par le site logcenter.uac.bj pour les analyser et les visualiser via OpenSearch-dashboards. Nous allons utiliser le script suivant pour collecter les logs avec fluentd et les envoyer à opensearch-dashboards.

```
<source>
  @type tail
  path "/var/log/apache2/access.log"
  pos_file "/var/log/td-agent/access.log.pos"
  tag "apache.access"
  <parse>
    @type "regexp"
    expression /^(?<remote_addr>[^ ]+) - (?<remote_user>[^ ]+) \[\[(?<time>[^\]]+)\]\]
    "(?<request>[^"]+)" (?<status>\d+) (?<size>\d+|-) \
    "(?<referer>[^"]+|)" "(?<user_agent>[^"]+)"/
    time_format "%d/%b/%Y:%H:%M:%S %z"
    time_key "time"
  </parse>
</source>

<filter apache.access>
  @type grep
  <exclude>
    key request
    pattern /-/
  </exclude>
</filter>

<match apache.access>
```

```
  @type opensearch
  host "logscenter.uac.bj"
  port 9200
  scheme https
  ssl_verify false
  user "admin"
  password "Butterf1y@2024"
  index_name "apache-logs"
</match>
```

Ce script Fluentd collecte les logs d'accès Apache à partir du fichier /var/log/apache2/access.log et les envoie à l'index apache-logs dans OpenSearch. Il utilise une expression régulière pour extraire des informations clés comme l'adresse IP, l'utilisateur et le statut de la requête, tout en excluant les lignes avec des requêtes non valides. Les données sont ensuite transférées à OpenSearch via une connexion sécurisée en utilisant les informations d'identification fournies.

Lorsque nous interagissons avec le site, les informations sont enregistrées et deviennent accessibles sur le Dashboards. Pour intégrer l'index apache-logs dans l'onglet Discover, nous devons d'abord vérifier que l'index a été créé et qu'il contient des données. Ensuite, nous accédons à la section **Dashboards Management > Index Patterns** pour créer un nouveau modèle d'index basé sur apache-logs. Une fois le modèle d'index créé, l'index sera visible dans l'onglet **Discover**, ce qui nous permettra de consulter et d'analyser facilement les journaux Apache collectés.

Après avoir interagit avec le site, nous avons consulté les informations collectées sur le dashboard d'OpenSearch. Pour cela, nous sommes allés dans l'onglet Discover, puis nous avons choisi l'index apache-logs, que nous avions créé précédemment.

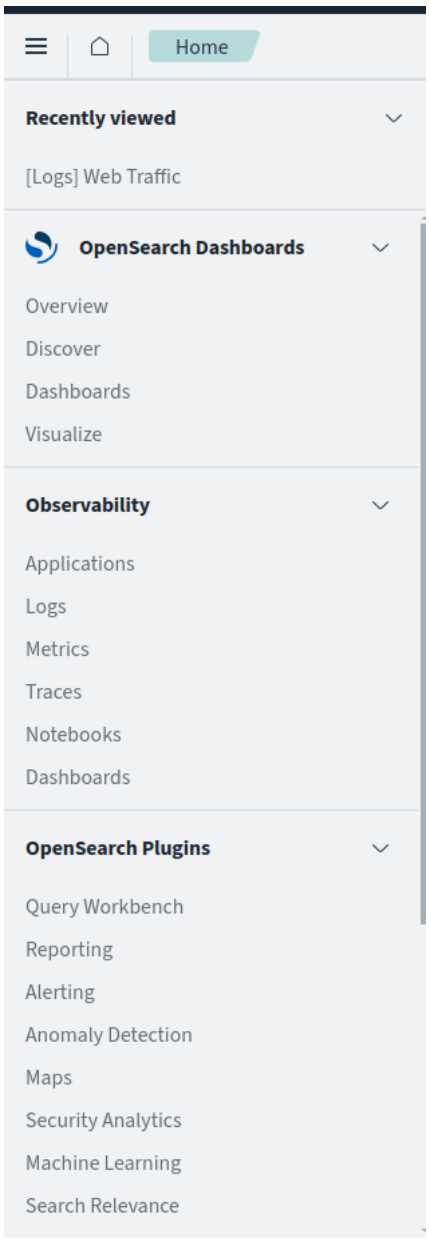


FIGURE 3.14 : Affichage du Dashboards

Nous pouvons voir dans l'onglet Discover les informations suivantes

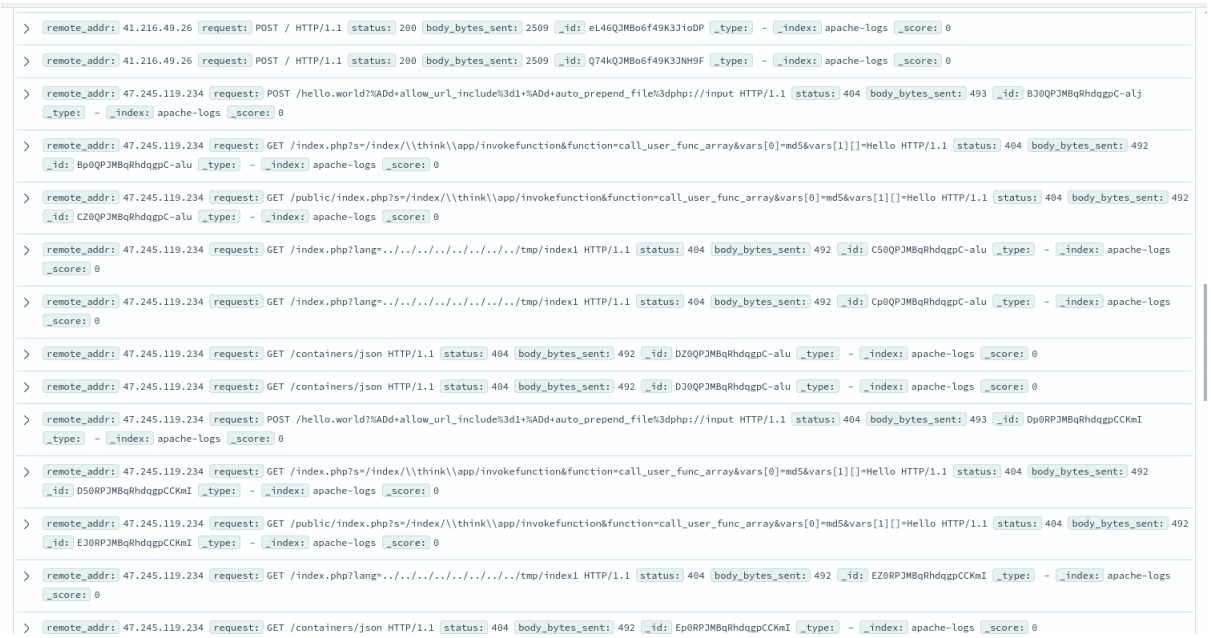


FIGURE 3.15 : Affichage du Dashboards

Explication des champs du log

- **remote_addr :** Adresse IP du client qui a effectué la requête. *Exemple :* `47.245.119.234`.
- **request :** Détail de la requête HTTP effectuée par le client.
  - **Méthode HTTP :** `GET`.
  - **Chemin demandé :** `/public/index.php?s=/index/think app/invokefunction&function=call_user_func_array&vars[0]=md5&vars[1][]=Hello`.
- **status :** Code de statut HTTP renvoyé par le serveur. *Exemple :* `404` (ressource non trouvée).
- **body_bytes_sent :** Taille (en octets) de la réponse envoyée par le serveur. *Exemple :* `492` octets.
- **_id :** Identifiant unique généré pour ce log dans OpenSearch. *Exemple :* `CZ0QPJMBqRhdqgpC-alu`.
- **_type :** Champ désuet dans OpenSearch, ici laissé vide (-).
- **_index :** Nom de l'index où ce log est stocké dans OpenSearch. *Exemple :* `apache-logs`.
- **_score :** Score de pertinence du document dans le cadre d'une recherche. *Exemple :* `0` (pas lié à une recherche spécifique).

Dans ce cas, l'adresse IP `47.245.119.234` a tenté d'accéder au chemin `/public/index.php...`, en utilisant la méthode HTTP `GET`, ce qui semble être une tentative d'exécution d'une fonction via des paramètres (potentiellement malveillante). Le serveur a répondu avec un code `404`, indiquant que la ressource demandée n'existe pas, et a renvoyé une réponse de `492` octets. Ce log a été enregistré dans l'index `apache-logs` avec un identifiant unique, permettant de le retrouver facilement pour analyser. Cela est une tentative d'accès non autorisé ou une exploration automatisée des failles du serveur.

Nous allons maintenant procéder à la visualisation des données à l'aide de l'onglet Visualize. À titre d'exemple, nous allons compter le nombre de requêtes ayant échoués c'est-à-dire dont le statut est 404. Pour ce faire, nous commençons par accéder à l'onglet Visualize . Ensuite, nous cliquons sur : Create new visualization, et choisissons le type de graphique souhaité. Nous sélectionnons ensuite l'index que nous avons créé, dans notre cas apache-logs.

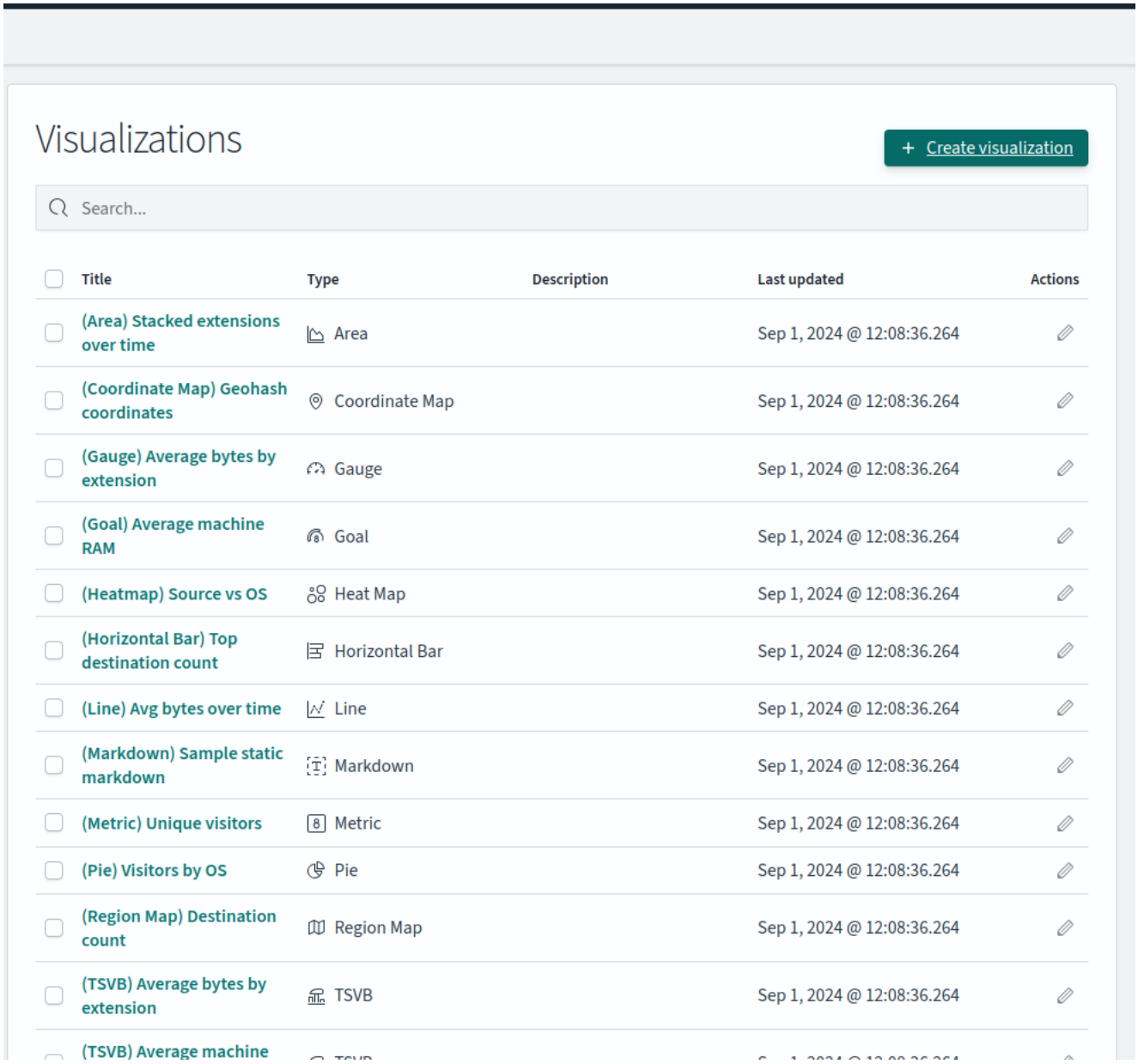


FIGURE 3.16 : Visualisation du Dashboards

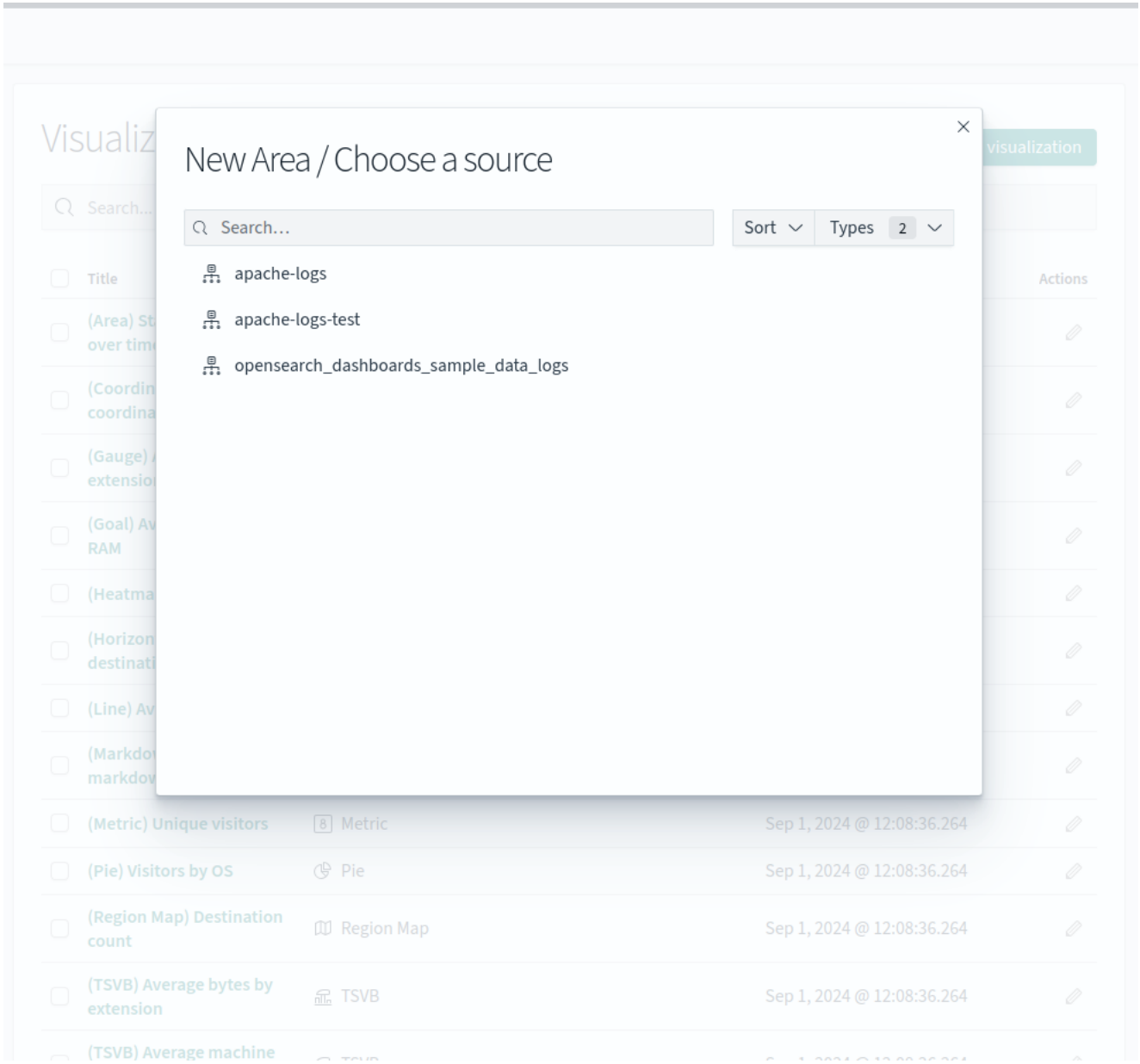


FIGURE 3.17 : Visualisation du Dashboards

Nous allons filtrer les requêtes dont le champ `status` est égal à : 404. Pour cela, nous allons ajouter un filtre et définir nos conditions et avoir le résultat suivant

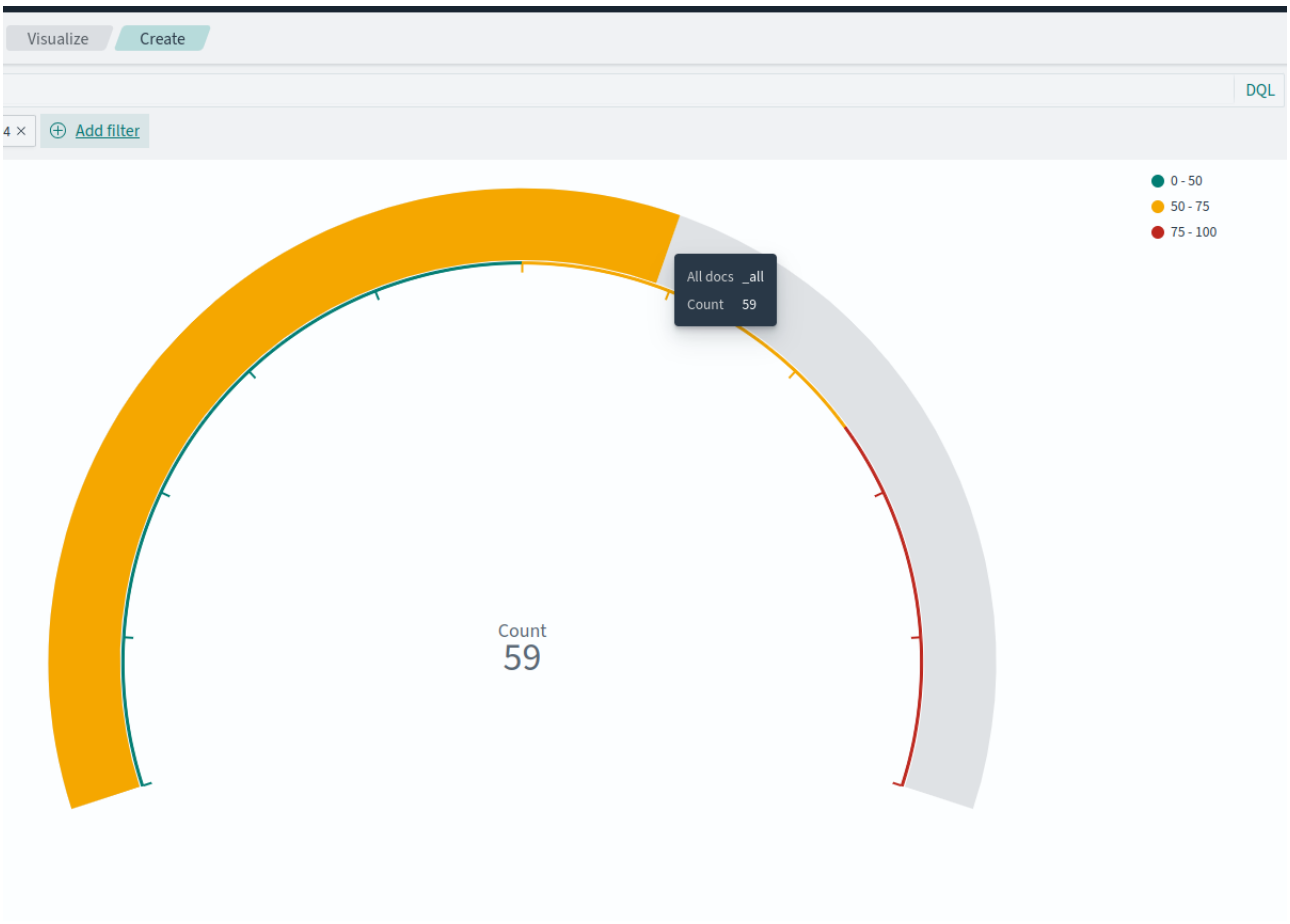


FIGURE 3.18 : Résultat du filtre

Nous constatons donc qu'il y a 59 requêtes qui ont échouées sur le site. Afin d'être informé de l'échec d'une requête sur le site, nous allons configurer des alertes. Pour cela , nous irons dans l'onglet Alerting pour définir nos conditions. Nous allons également créer un déclencheur et lui dire l'action à exécuter. Nous avons alors créé des alertes pour les requêtes dont le statut est 404 et avons demandé l'envoi d'une notification si l'alerte se déclenche. Cette notification nous sera envoyée par Slack que nous avions lié à opensearch-dashboards.

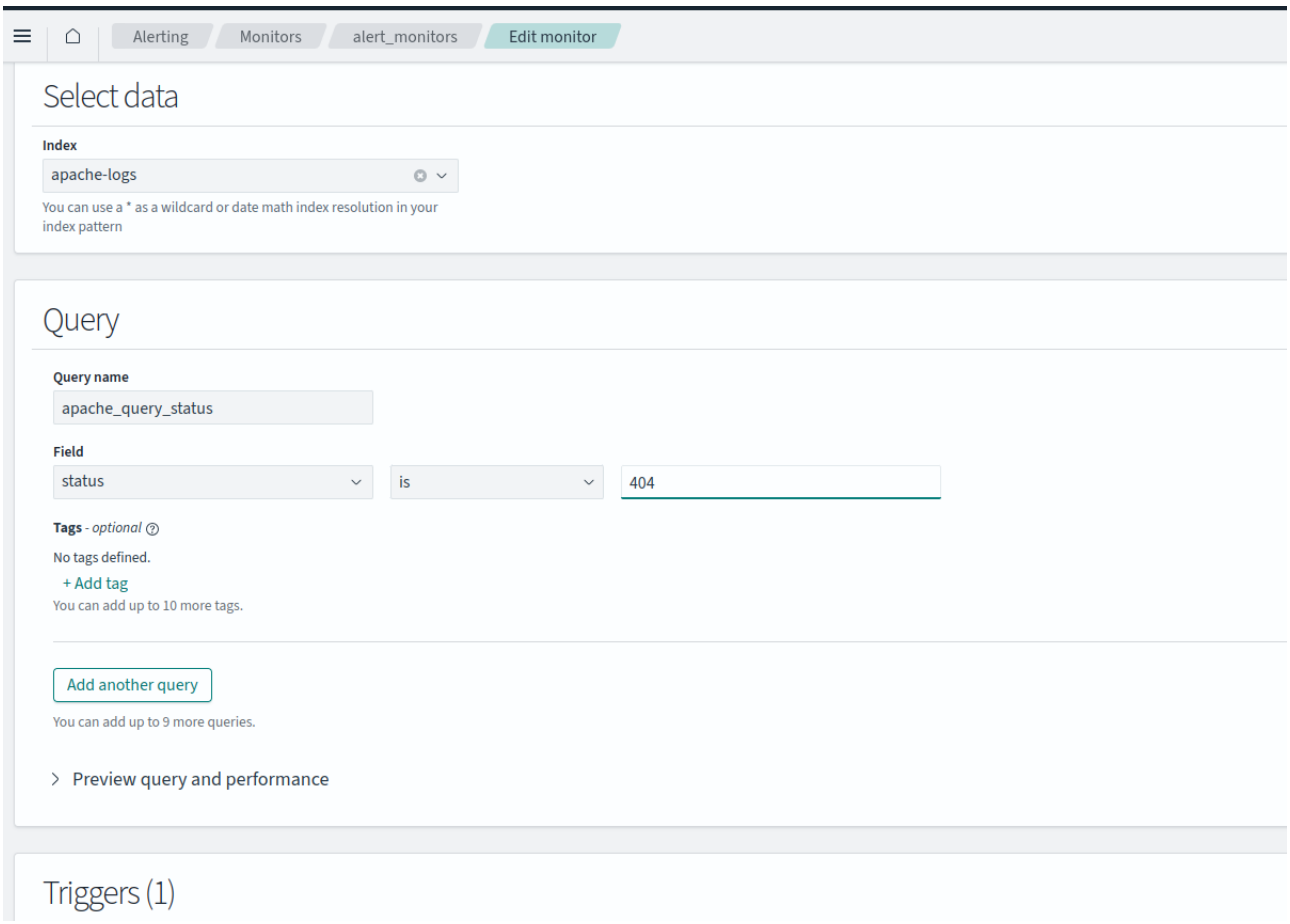


FIGURE 3.19 : Configuration d'alerte

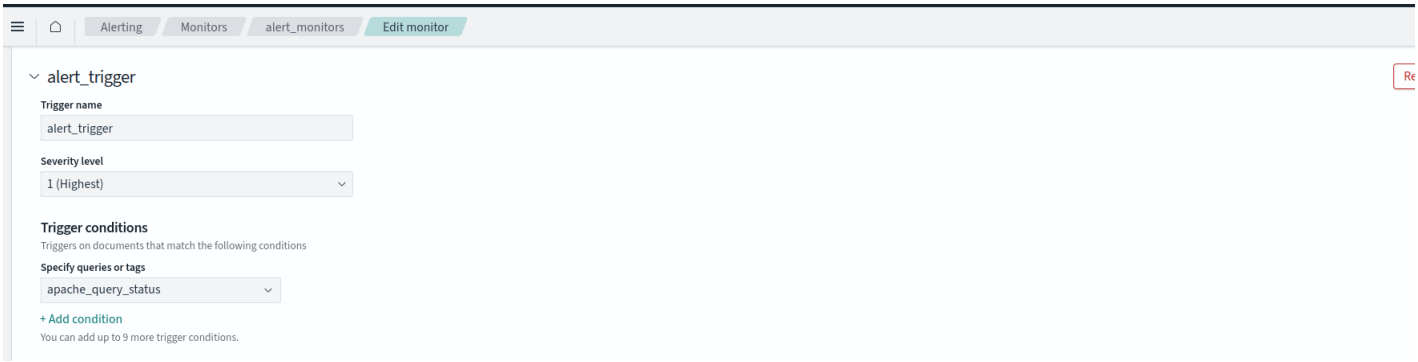


FIGURE 3.20 : Configuration d'alerte

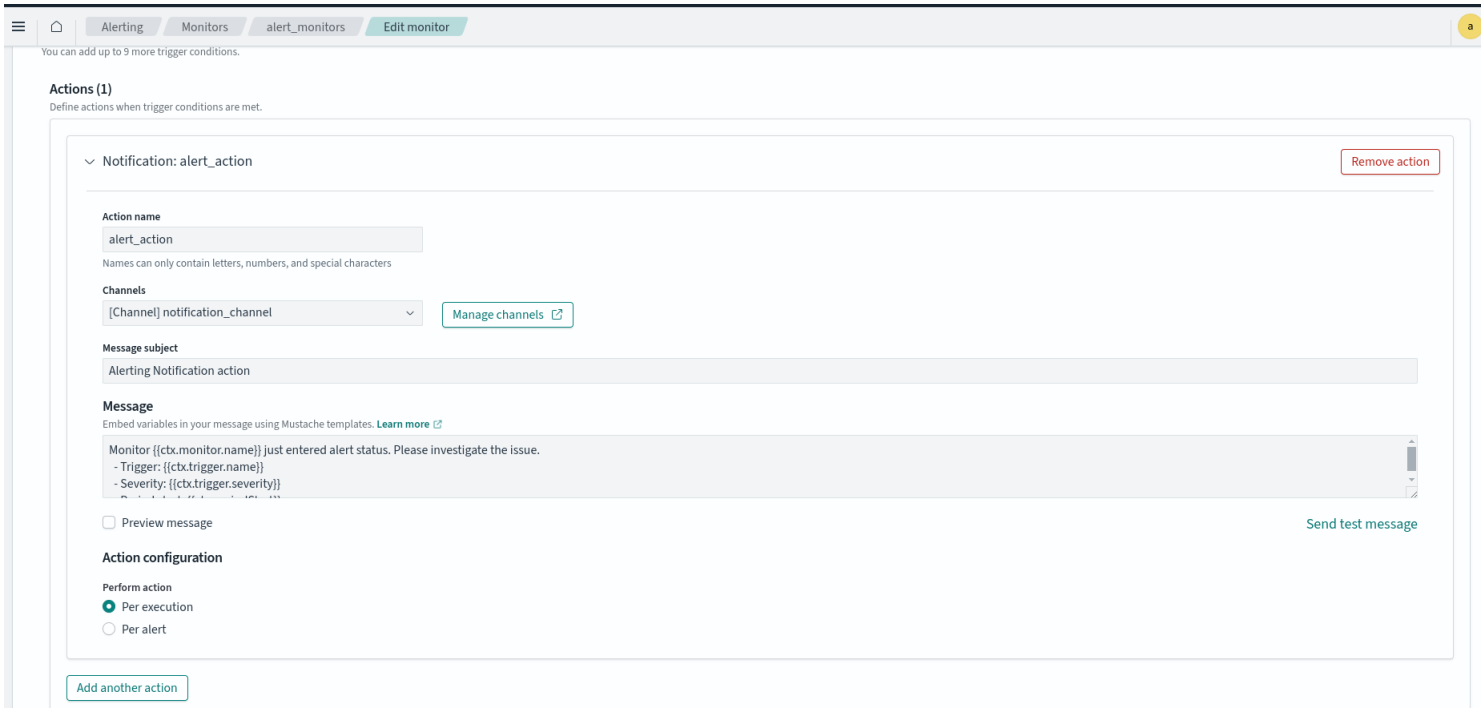


FIGURE 3.21 : Configuration d'alerte

Des alertes ont été configurées pour détecter les requêtes avec un statut HTTP 404, indiquant des ressources non trouvées sur le serveur. Chaque fois qu'une telle requête est détectée, une notification est automatiquement envoyée via Slack grâce à un webhook intégré, permettant une réactivité immédiate pour analyser et résoudre ces incidents. Ce mécanisme de surveillance a permis d'identifier un total de 19 alertes, ce qui témoigne de l'efficacité du système d'alerte mis en place pour assurer un suivi précis des erreurs et optimiser la gestion des logs.

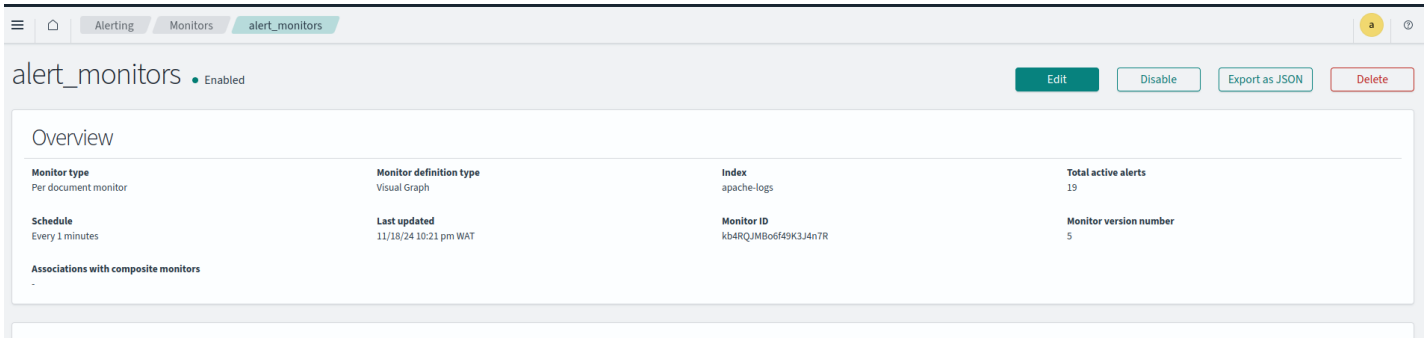


FIGURE 3.22 : Génération d'alertes

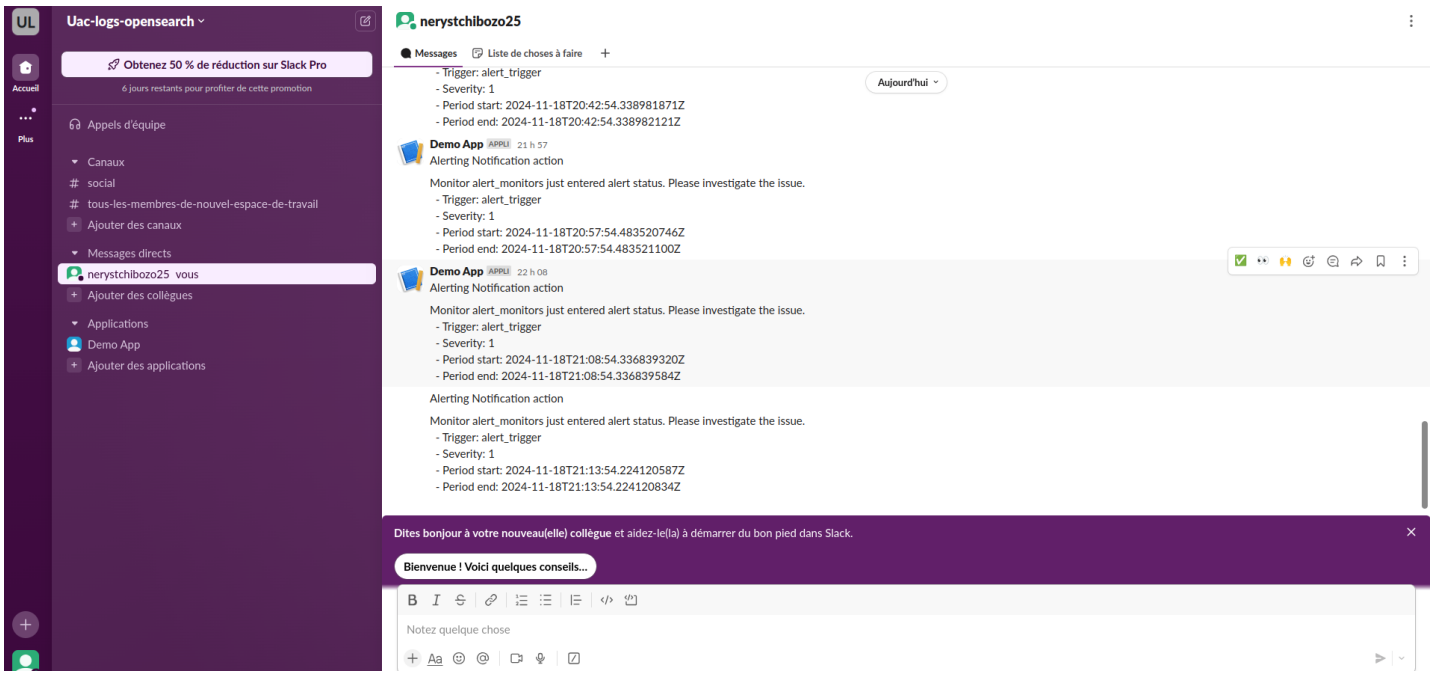


FIGURE 3.23 : Notification

3.6 Interprétation des notifications d'alertes

Les notifications suivantes montrent les alertes critiques déclenchées par le système de surveillance. Voici leur contenu brut suivi de leur interprétation.

3.6.1 Interprétation des notifications

Les notifications ci-dessus indiquent que le système de surveillance a détecté des événements critiques nécessitant une investigation immédiate. Voici une analyse détaillée :

- **Notification 1** :
  - Déclencheur : `alert_trigger`.
  - Gravité : `Severity: 1`, indiquant un problème critique.
  - Intervalle : entre `06:18:54.441569518Z` et `06:18:54.441569899Z`.
  - Interprétation : une condition anormale a été détectée dans les métriques surveillées.
- **Notification 2** :
  - Déclencheur : `alert_trigger`.
  - Gravité : `Severity: 1`, toujours critique.
  - Intervalle : entre `06:22:54.336811322Z` et `06:22:54.336811584Z`.
  - Interprétation : la même anomalie ou une nouvelle condition a été détectée.

Le système de surveillance a détecté deux événements critiques, signalés par des notifications, nécessitant une attention immédiate. Ces alertes, déclenchées par le mécanisme `alert_trigger`, ont une gravité maximale (`Severity: 1`), ce qui indique des problèmes sérieux dans les métriques surveillées. La première notification

a été enregistrée entre `06:18:54.441569518Z` et `06:18:54.441569899Z`, révélant une condition anormale. Quelques minutes plus tard, une seconde alerte, survenue entre `06:22:54.336811322Z` et `06:22:54.336811584Z`, indique soit une persistance du problème, soit une nouvelle anomalie similaire. Cela montre un problème récurrent ou potentiellement aggravé qui doit être rapidement investigué pour éviter des impacts sur le système.

3.7 Présentation des résultats

La mise en place du système de gestion centralisée des journaux a produit des résultats significatifs, renforçant à la fois la supervision et la sécurité des serveurs de l'UAC. Pour le site hébergé sur le serveur Apache, les logs collectés ont permis d'identifier et d'analyser les erreurs, souvent causées par des tentatives d'accès à des liens inexistants. Ces informations ont été utilisées pour créer un mécanisme d'alerte, avec des notifications en temps réel via Slack, permettant une intervention rapide pour corriger les anomalies.

En parallèle, la collecte des logs SSH a révélé des tentatives de connexion parfois suspectes. Grâce à OpenSearch Dashboards, des tableaux de bord interactifs ont permis de visualiser les adresses IP utilisées, la fréquence des tentatives, facilitant ainsi l'identification des menaces. L'implémentation technique a nécessité la configuration de Fluentd pour capturer les journaux des fichiers système, l'établissement d'une connexion sécurisée par SSL/TLS avec OpenSearch, et la création de tableaux de bord personnalisés pour surveiller les activités critiques.

Le système répond aux objectifs initiaux, à savoir améliorer la visibilité sur les activités des serveurs, tout en offrant des outils efficaces pour une surveillance proactive.

Conclusion

Dans ce chapitre, nous avons présenté la mise en place d'un système de centralisation des journaux et de supervision en temps réel. Grâce à la collecte, au stockage, à l'analyse des logs, ainsi qu'à un mécanisme d'alertes et de notifications, ce système renforce la gestion proactive des serveurs, offrant à la fois réactivité et visibilité accrue pour assurer leur bon fonctionnement. en les acheminant vers OpenSearch. OpenSearch, en tant que moteur de recherche et d'analyse, a permis de stocker ces logs de manière centralisée et d'assurer leur accessibilité rapide. Enfin, OpenSearch Dashboards a été utilisé pour visualiser les données sous forme de tableaux de bord dynamiques. L'implémentation a consisté à configurer Fluentd pour capturer les journaux à partir de fichiers système, à établir une connexion sécurisée entre Fluentd et OpenSearch à l'aide de certificats SSL/TLS, et à créer des visualisations adaptées dans OpenSearch Dashboards pour analyser les logs collectés. Par exemple, un tableau de bord a permis d'identifier les tentatives de connexion anormales sur SSH, localisant les adresses IP suspectes et leurs fréquences. Ce système offre une visibilité accrue sur les activités des serveurs de l'UAC, renforçant ainsi leur sécurité et facilitant la surveillance proactive des ressources.

Conclusion Générale

Ce mémoire a proposé une solution performante pour la gestion centralisée des logs des serveurs de l'Université d'Abomey-Calavi, en s'appuyant sur les outils OpenSearch, Fluentd et OpenSearch Dashboards. Cette solution répond aux besoins croissants de suivi, d'analyse et de sécurisation des infrastructures informatiques en permettant une collecte automatisée et systématique des journaux provenant de divers services, leur formatage structuré selon des normes cohérentes, et leur présentation à travers des tableaux de bord interactifs. et dynamiques. La centralisation des logs simplifie considérablement la gestion des infrastructures en renforçant la capacité des administrateurs à détecter rapidement les anomalies, les activités suspectes ou les failles potentielles. En plus d'améliorer la visibilité sur les activités des systèmes, cette solution fournit des outils essentiels pour le diagnostic précis des incidents, permettant d'identifier les causes profondes des problèmes et de prendre des mesures correctives efficaces et les impacts sur les services critiques. Une des fonctionnalités clés de cette solution est l'intégration d'un système d'alertes automatisé et configurable, capable de notifier les équipes techniques en temps réel lors d'événements critiques tels que des dépassements de seuils, des tentatives d'accès non autorisées. ou des dysfonctionnements du système. Ce mécanisme d'alerte proactive garantit une réactivité accrue, améliorant ainsi la résilience opérationnelle et permettant une gestion plus proactive des risques. Par ailleurs, cette solution jette les bases d'une gestion des logs évolutive et adaptable aux défis technologiques futurs. Elle ouvre la voie à l'intégration de modules avancés comme l'analyse prédictive basée sur des algorithmes d'apprentissage automatique, qui permettrontient de prévoir les pannes et de planifier des interventions préventives, ou encore des outils d'intelligence artificielle pour une gestion autonome. et intelligente des incidents. Des améliorations futures pourraient également inclure des visualisations plus riches et personnalisées adaptées aux besoins spécifiques des différents utilisateurs, consolidant ainsi la sécurité, l'efficacité et la flexibilité des infrastructures informatiques de l'Université. En définitive, cette solution représente une avancée majeure pour la gestion des systèmes, tout en posant les bases pour des innovations qui renforceront durablement la capacité de l'Université à répondre aux exigences de son environnement numérique en constante évolution.

Bibliographie

[1] Cheswick, William R., et Steven M. Bellovin. *Firewalls and Internet Security : Repelling the Wily Hacker*. Addison-Wesley, 1994.

[2] Bejtlich, Richard. *The Practice of Network Security Monitoring : Understanding Incident Detection and Response*. No Starch Press, 2013.

[3] Rusu *Vărzaru, A. (2020). Advances in log data analysis for anomaly detection. International Journal of Information Security*.

Webographie

[4] Logging, *Logging Documentation*, https://www.vaadata.com/blog/fr/logging-monitoring-definitions-et-bonnes-pratiques/, Consulté en 2024.

[5] Logging, *Meilleures pratiques de journalisation sécurisée* , https://owasp.org/owasp-istg/01_introduction/, Consulté en 2024.

[6] Let's Encrypt., *How to secure your application with Let's Encrypt certificates*, https://letsencrypt.org/docs/, Consulté en 2024.

[7] Linux Foundation *Open source tools for log management.*, https://linuxfoundation.org/tools/log-management, Consulté en 2024.

[8] ELK, *ELK Documentation*, https://www.elastic.co/fr/elastic-stack, Consulté en 2024.

[9] Kibana, *Kibana Documentation*, https://www.elastic.co/fr/kibana, Consulté en 2024.

[10] Risques-informatiques, *Risques-informatiques Documentation*, https://syxperiane.com/types-attaques-informatiques/, Consulté en 2024.

[11] Gestion des logs, *Gestion des logsDocumentation*, https://www.tek-tools.com/apm/log-management-tips-and-tools, Consulté en 2024.

[12] Fluentd vs Logstash, *Fluentd vs Logstash*, https://signoz.io/blog/fluentd-vs-logstash/, Consulté en 2024.

[13] Journalisation des événements++, *Journalisation des événements systèm Documentation*, https://www.techtarget.com, Consulté en 2024.

[14] OpenSearch, *OpenSearch Documentation*, https://opensearch.org/, Consulté en 2024.

[15] Fluentd, *Fluentd Documentation*, https://www.fluentd.org/, Consulté en 2024.

[16] Opensearch-Fluentd, *Opensearch-Fluentd Documentation*, https://opensearch.org/blog/getting-started-with-fluentd-and-opensearch/, Consulté en 2024.

[17] Comparaison opensearch et ELK, *Comparaison opensearch et ELK*, https://www.chaossearch.io/blog/opensearch-vs-elasticsearch-comparison, Consulté en 2024.

Table des matières

Dédicace ii

Remerciements iii

Résumé iv

iv

Abstract v

v

List of Figures vi

List of Tables vii

Liste des acronymes viii

Introduction 2

1 Revue de littérature 4

Introduction . . . 4

1.1 Introduction à la gestion de logs . . . 4

1.1.1 Définition du logging . . . 4

1.1.2 Définition d'un système de logging . . . 4

1.1.3 Fonctionnement d'un Système de Logging . . . 5

1.1.4 Gestion Actuelle des Logs à l'UAC . . . 5

1.1.5 Pourquoi la Gestion des Logs est Cruciale pour l'UAC . . . 6

1.2 Utilité et Objectifs d'un Système de Logging . . . 6

1.2.1 Importance de la Surveillance et de la Gestion des Logs . . . 6

1.3 Types et Approches de Systèmes de Logging . . . 6

1.3.1 Logging Local vs Logging Centralisé . . . 6

1.4 Solutions Actuelles et Limitations . . . 6

1.4.1 Revue des Outils de Logging . . . 6

1.4.2 Défis et Limitations des Solutions Actuelles . . . 7

1.5 Notre Solution . . . 7

1.5.1 Description de la Solution Proposée . . . 7

Conclusion . . . 7

2 Choix de la solution et présentation du matériel 8

Introduction . . . 8

2.1 Critères de Choix de la Solution . . . 8
2.1.1 Exigences Fonctionnelles . . . 8
2.1.2 Exigences Techniques . . . 8
2.2 Options d'Implémentation . . . 9
2.2.1 Étude des Solutions Existantes . . . 9
2.2.2 Comparaison des solutions . . . 10
2.3 Choix de la solution . . . 11
2.3.1 Justification du choix . . . 11
2.4 Architecture et Composants du Système . . . 11
2.4.1 Architecture Globale . . . 11
2.5 Intégration et Déploiement . . . 12
Conclusion . . . 13

3 Implémentation et collecte des logs avec opensearch 14
Introduction . . . 14
3.1 Environnement de travail . . . 14
3.2 Installation et configuration de Opensearch . . . 14
3.3 Configuration de fluentd . . . 20
3.4 Installation d'OpenSearch Dashboards . . . 21
3.5 Mise en œuvre de la solution . . . 22
3.6 Interprétation des notifications d'alertes . . . 32
3.6.1 Interprétation des notifications . . . 32
3.7 Présentation des résultats . . . 33
Conclusion . . . 33

Conclusion 34

Bibliographie 35

Webographie 36

Table des matières 37