**Membres du Jury:**

* President :
* Rapporteurs :
* Docteur
* Monsieur
* Examinateur :

Présenté le

par Monsieur **RAVELOMAHARAVO Rotsy Maminiaina Thierry Michael**

**ETUDE DE MISE EN PLACE DE CI/CD ET ORCHESTRATION D’UNE APPLICATION EN ARCHITECTURE MICROSERVICES**

Option: Administration des systèmes et réseaux

Intitulé

**MEMOIRE DE FIN D’ETUDES**

**POUR L’OBTENTION DU DIPLOME DE MASTER PROFESSIONNEL EN INFORMATIQUE**

**ECOLE NATIONALE D’INFORMATIQUE**

**UNIVERSITE DE FIANARANTSOA**

Annee universtaire 2024-2025

### CURRICULUM VITAE

ETAT CIVIL

RAVELOMAHARAVO Rotsy Maminiaina Thierry Michael

23 ans, Celibataire

240 Bis | 3611 Tanambao Fianarantsoa 301

Nationnalite: Malagasy

Contact: 034 88 359 57

1. mail: [thierrymichael2001@gmail.com](mailto:thierrymichael2001@gmail.com)

FORMATIONS ET DIPLOMES

2022-2023: L3 Informatique General, ENI Fianarantsoa

2021-2022: L2 Informatique General, ENI Fianarantsoa

2019-2021: L1 Informatique General, ENI Fianarantsoa

COMPETENCE EN INFORMATIQUE

Logiciels : Virtualbox, EoN, GNS3, Jenkins, Asterisk, Git, Vagrant, Docker, Kubernetes

Services : SSH, Apache2

Langages de programmation: PHP, Python, nodeJS, ReactJS, ReactNative

Systeme de Gestion de Base de donnees: MySQL, PostGreSQL

Syteme d’exploitation: Linux distribution Debian

EXPERIENCE PROFESSIONNELLE

Octobre 2022-Decembre 2022: Stagiare en tant qu’administrateur Systemes et Reseaux a l’ESD Nanisana, Antananarivo

COMPETENCES LINGUISTIQUES

|  |  |  |  |
| --- | --- | --- | --- |
| **Langues** | **Parlé** | **Ecrit** | **Compréhension** |
| Français | Très bien | Très bien | Très bien |
| Anglais | Assez bien | Très bien | Très bien |

LOISIRS

Lecture, Volley-ball

### REMERCIEMENTS

Nous tenons tout d’abord à remercier Dieu Tout Puissant sans qui rien n’aurait été possible.

Nous tenons aussi à remercier:

* Monsieur HAJALALAINA Aimé Richard, Docteur et Président de l’Université de Fianarantsoa;
* Monsieur THOMAS Mahatody, Docteur HDR et Directeur de l’Ecole Nationale d’Informatique, pour son dévouement dans la direction de l’Ecole;
* Monsieur le Directeur de l’Ecole Supérieure de Droits publiques et sciences politiques, de nous avoir acceulli dans son établissement pour effectuer notre stage;
* Monsieur Anny Sebastien, Responsable des Systemes d’Information et Administrateur Réseau de l’ESD, pour son encadrement tout au long de notre stage à l’ESD
* Le personnel de l’ESD pour leur bonne collaboration dans le déroulement de notre stage et leur acceuil très chaleureux au sein de leur établissement;
* Chacuns et chacunes des enseignants de l’ENI sans qui nous n’aurions pas eu les compétences et les atouts nécéssaires à la réalisation de notre stage;
* Le personnel administratif de l’ENI pour leur travail pour assurer le bon déroulement de notre année universitaire
* Nos amis, nos parents, nos familles pour leur aide précieuse et leur soutien moral, financier et l’hébérgement.

A vous tous, veuillez agréer à nos remerciements les plus sincères.

### SOMMAIRE

[PARTIE I : PRESENTATION GENERALE 12](#_Toc555429746)

[Chapitre 1. Présentation de l’Ecole Nationale d’Informatique 13](#_Toc63470259)

[Chapitre 2. Présentation OpenData-Madagascar 20](#_Toc1314147381)

[Chapitre 3. Description du projet 33](#_Toc1012135227)

[PARTIE II : ANALYSE DE L’EXISTANT 35](#_Toc1416063651)

[Chapitre 4. Analyse de l’existant proprement dit 36](#_Toc1254325855)

[Chapitre 5. Proposition de solution 41](#_Toc1375101511)

[PARTIE III : CHOIX DE L’OUTILS 42](#_Toc289696513)

[PARTIE IV : MISE EN OEUVRE 44](#_Toc1492469257)

[Chapitre 6. Mise en oeuvre proprement dit 45](#_Toc1111461535)

[Chapitre 7. Validations avec contre-analyse et perspectives 46](#_Toc179125881)

[Chapitre 8. Conclusion 47](#_Toc1295183240)

[CONCLUSION GENERALE 48](#_Toc946466339)

### LISTE DES ABBREVIATIONS

### LISTE DES TABLEAUX

### LISTE DES FIGURES

### **INTRODUCTION GÉNÉRALE**

L’évolution rapide des technologies de développement logiciel a profondément transformé la manière dont les entreprises conçoivent, déploient et maintiennent leurs applications. Aujourd’hui, les organisations cherchent avant tout la rapidité, la fiabilité et la flexibilité dans leurs processus de livraison logicielle. C’est dans ce contexte qu’émergent les approches **DevOps**, reposant sur la collaboration entre les équipes de développement et d’exploitation, et favorisant l’automatisation du cycle de vie des applications.

L’intégration continue (**Continuous Integration – CI**) et le déploiement continu (**Continuous Deployment – CD**) constituent les piliers de cette approche. Ils permettent d’automatiser les différentes étapes de compilation, de test, et de mise en production des applications, réduisant ainsi les erreurs humaines et améliorant la qualité des livrables. Combinées à l’**orchestration**, ces pratiques facilitent la gestion d’applications complexes, notamment celles basées sur une **architecture microservices**, où chaque composant fonctionne de manière indépendante tout en collaborant avec les autres.

Le présent mémoire, intitulé **« Mise en place d’un CI/CD et orchestration pour une application en architecture microservices »**, s’inscrit dans cette dynamique. Il a pour objectif principal de **concevoir et implémenter une chaîne d’intégration et de déploiement automatisée** adaptée à une application microservices, en assurant un déploiement fiable, rapide et reproductible. Ce projet vise également à **optimiser la gestion et la scalabilité** des différents services applicatifs grâce à des outils modernes d’orchestration.

Le travail s’appuie sur le contexte de l’entreprise **Tetika**, qui propose une large gamme de produits logiciels destinés à divers domaines de gestion. Dans un environnement où les mises à jour sont fréquentes et les déploiements encore majoritairement manuels, l’adoption d’une approche DevOps s’impose comme une nécessité afin d’améliorer la productivité, la qualité et la disponibilité des services.

Dans le cadre de ce mémoire, nous présenterons tout d’abord l’entreprise d’accueil et le contexte du projet. Nous définirons ensuite la problématique et les objectifs poursuivis, avant de décrire les moyens nécessaires à la réalisation du projet. Enfin, nous détaillerons les étapes de mise en œuvre de la solution proposée et les résultats obtenus, tout en envisageant les perspectives d’évolution.

# PARTIE I : PRESENTATION GENERALE

# Chapitre 1. Présentation de l’Ecole Nationale d’Informatique

* 1. **Information d’ordre générale**

L’Ecole Nationale d’Informatique, en abrégé ENI, est un établissement d’enseignement supérieur rattaché académiquement et administrativement à l’Université de Fianarantsoa. Le siège de l’Ecole se trouve à Tanambao-Antaninarenina à Fianarantsoa. L’adresse pour la prise de contact avec l’Ecole est la suivante : Ecole Nationale d’Informatique (ENI) Tanambao, Fianarantsoa. Le numéro de sa boîte postale est 1487 avec le code postal 301. Téléphone : 034 05 733 36 ou 032 15 204 28. Son adresse électronique est la suivante : **eni@eni.mg**. Il dispose également d'un site web : **www.eni.mg**

* 1. **Missions et historiques**

L’ENI se positionne sur l’échiquier socio-éducatif malgache comme étant le plus puissant secteur de diffusion et de vulgarisation des connaissances et des technologies informatiques.

Cette Ecole Supérieure peut être considérée aujourd’hui comme la vitrine et la pépinière des élites informaticiennes du pays.

De façon formelle, l’ENI était créée par le décret N° 83- 185 du 24 Mai 1983, comme étant le seul établissement Universitaire Professionnalisé au niveau national, destiné à former des techniciens et des Ingénieurs de haut niveau, aptes à répondre aux besoins et exigences d’Informatisation des entreprises, des sociétés et des organes implantés à Madagascar.

L’ENI a pour conséquent pour mission de former des spécialistes informaticiens compétents et opérationnels de différents niveaux notamment :

* en fournissant à des étudiants des connaissances de base en informatique ;
* en leur transmettant le savoir-faire requis, à travers la professionnalisation des formations dispensées et en essayant une meilleure adéquation des formations par rapport aux besoins évolutifs des sociétés et des entreprises ;
* en initiant les étudiants aux activités de recherche dans les différents domaines des Technologies de l’Information et de la Communication (TIC) ;

La filière de formation d’Analystes Programmeurs a été mise en place à l’Ecole en 1983, et a été gelée par la suite en 1996, tandis que la filière de formation d’ingénieurs a été ouverte à l’Ecole en 1986.

Une formation de troisième cycle a été ouverte à l’Ecole a été ouverte à l’Ecole depuis l’année 2003 – 2004 grâce à la coopération académique et scientifique entre l’Université de Fianarantsoa pour le compte de l’ENI et l’Université Paul Sabatier de Toulouse (UPST).

Cette filière avait pour objectif de former certains étudiants à la recherche dans les différents domaines de l’Informatique, et notamment pour préparer la relève des Enseignants-Chercheurs qui étaient en poste.

Pendant l’année 2007-2008, la formation en vue de l’obtention du diplôme de Licence Professionnelle en Informatique a été mise en place à l’ENI avec les deux parcours de formation :

* Génie Logiciel et base de Données.
* Administration des Système et réseaux.

La mise en place à l’Ecole de ces deux options de formation devait répondre au besoin de basculement vers le système Licence – Master – Doctorat (LMD).

En vue de surmonter les difficultés de limitation de l’effectif des étudiants accueillis à l’Ecole, notamment à cause du manque d’infrastructures, un système de « Formation Hybride » a été mise en place à partir de l’année 2010. Il s’agit en effet d’un système de formation semi présentielle et à distance avec l’utilisation de la visioconférence pour la formation à distance. Le système de formation hybride a été ainsi créé à Fianarantsoa ainsi qu’Université de Toliara. Cette formation est à l’origine du parcours Informatique Générale.

En 2023, une nouvelle mention Intelligence Artificielle (IA) a été ouvert au sein de l’Ecole pour répondre les besoins des entreprises. La formation est destinée aux étudiants titulaires du diplôme de licence (Bac +3) en Mathématiques ou en Statistiques ou en Informatique, etc. La mention IA comporte deux parcours :

* Gouvernance et Ingénierie de Données (GID),
* Objets connectés et Cybersécurité (OCC).

Le principe de l’enseignement pour le parcours GID offre aux l’étudiants des compétences scientifiques et techniques spécialisées en Science de données. Pour le parcours OCC, les étudiants octroient la double spécialité premièrement en internet des objets et deuxièmement en cybersécurité. La formation de master est axée sur l’ensemble d’applications de l’Intelligence Artificielle.

* 1. **Organigramme institutionnel**

L’organigramme de l’Ecole est inspiré des dispositions du décret N° 83-185 du 24 Mai 1983. L’ENI est administrée par un Conseil d’Ecole, et dirigée par un directeur nommé par un décret adopté en Conseil des Ministres. Le Collège des enseignants regroupant tous les enseignants-chercheurs permanents de l’Ecole est chargé de résoudre les problèmes liés à l’organisation pédagogique des enseignements. Le Conseil Scientifique propose les orientations pédagogiques et scientifiques de l’établissement, en tenant compte notamment de l’évolution du marché de travail et de l’adéquation des formations dispensées par rapport aux besoins des entreprises. La figure 1 représente l’organigramme actuel de l’ENI.

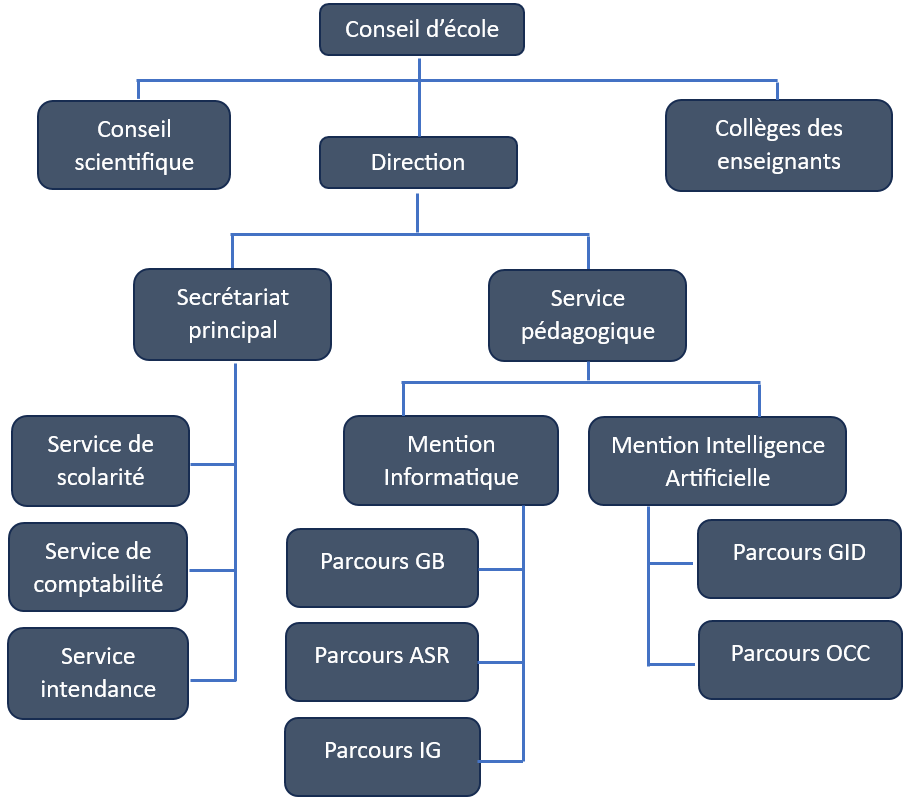


Figure 1. Organigramme actuel de l’Ecole

* 1. **Domaine de spécialisation**

Les activités de formation et de recherche organisées à l’ENI portent sur les domaines suivants :

* Génie logiciel et Base de Données ;
* Administration des Systèmes et Réseaux ;
* Informatique Générale ;
* Modélisation informatique et mathématique des Systèmes complexes ;
* Intelligence artificielle.

Le tableau 1 décrit l’organisation du système de formation pédagogique de l’Ecole.

Tableau 1. Organisation du système de formation pédagogique de l’Ecole

|  |  |
| --- | --- |
| Formation Théorique | Formation Pratique |
| * Enseignement théorique * Travaux dirigés * Travaux pratiques * Conférences | * Etude de cas * Travaux de réalisation * Projets/ Projets tutorés * Voyages d’Etudes * Stages en entreprise |

* 1. **Architecture des formations pédagogiques**

Le recrutement des étudiants à l’ENI se fait uniquement par voie de concours d’envergure nationale en première année. Les offres de formation organisées à l’Ecole ont été validées par la Commission Nationale d’Habilitation (CNH). Au sein de l’ENI, il existe deux mentions et cinq parcours. Le tableau 2 récapitule les mentions et les parcours au sein de l’Ecole :

Tableau 2. Mention et parcours au sein de l’ENI

|  |  |
| --- | --- |
| **Mention** | **Parcours** |
| **Informatique** | Génie logiciel et Base de Données (GB) |
| Administration des Systèmes et Réseaux (ASR) |
| Informatique Générale (IG) |
| **Intelligence Artificielle** | Gouvernance et Ingénierie de Données (GID) |
| Objets Connectés et Cyber sécurités (OCC) |

La figure 2 représente l’architecture des études correspondant au système LMD.

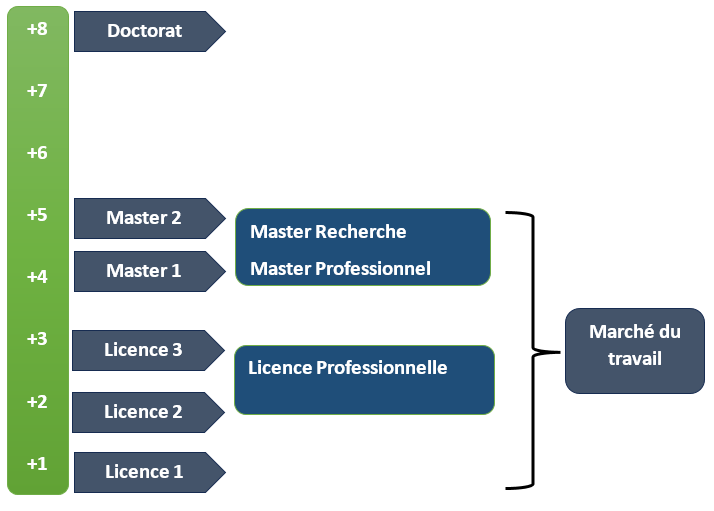


Figure 2. Architecture des études correspondant au système LMD

La licence peut avoir une vocation générale ou professionnelle. Le master peut avoir une vocation professionnelle ou de recherche. L’accès en première année de MASTER se fait automatiquement pour les étudiants de l’Ecole qui ont obtenu le diplôme de Licence Professionnelle. Le tableau 3 illustre la liste des formations existantes à l’ENI.

Tableau 3. Liste des formations existantes à l’ENI

|  |  |  |
| --- | --- | --- |
|  | FORMATION | |
|  | LICENCE PROFESSIONNELLE | MASTER |
| Condition admission | Par voie de concours | Par voie de concours pour la mention IA |
| Condition d’accès | Bac de série C, D ou Technique | Être titulaire de licence professionnelle |
| Durée de Formation | 3 ans | 2 ans |
| Diplôme délivré | Diplôme de Licence Professionnelle | Diplôme de Master Professionnel  Diplôme de Master Recherche |

Le Master Recherche permet à son titulaire de poursuivre directement des études en doctorat et de s’inscrire directement dans une Ecole Doctorale.

Les étudiants diplômés de l’Ecole sont plutôt bien accueillis dans les instituts universitaires étrangères (Canada, Suisse, France, …)

* 1. **Relation de l’ENI avec les organismes externes**

Les stages effectués chaque année par les étudiants mettent l’Ecole en rapport permanent avec plus de 400 entreprises et organismes publics, semi-publics et privés, nationaux et internationaux. L’Ecole dispose ainsi d’un réseau d’entreprises, de sociétés et d’organismes publics et privés qui sont des partenaires par l’accueil en stage de ses étudiants, et éventuellement pour le recrutement après l’obtention des diplômes par ces derniers. Les compétences que l’Ecole cherche à développer chez ses étudiants sont l’adaptabilité, le sens de la responsabilité, du travail en équipe, le goût de l’expérimentation et l’innovation.

En effet, la vocation de l’ENI est de former des licenciés et des ingénieurs de niveau MASTER avec des qualités scientifiques, techniques et humaines reconnues, capables d’évoluer professionnellement dans des secteurs d’activité variés intégrant l’informatique. Les stages en milieu professionnel permettent de favoriser une meilleure adéquation entre les formations à l’Ecole et les besoins évolutifs du marché de l’emploi.

Parmi les sociétés, les entreprises et les organismes partenaires de l’Ecole, on peut citer : ACCENTURE Mauritius, AKATA Goavana, Air Madagascar, Ambre Associates, Airtel, Agence Universitaire de la Francophonie (AUF), AXIAN, B2B, Banque Centrale, , BIANCO, BlueLine, CNaPS, Bureau National de Gestion des Risques et des Catastrophes (BNGRC), CEDII-Fianarantsoa, Data Consulting, Central Test, Centre National Antiacridien, CNRE, COLAS, Direction Générale des Douanes, DLC, E-Tech Consulting, , FID, FIHARY Soft, FTM, GNOSYS, GENIUS AT WORK, Hello Tana, IBONIA, INGENOSIA, INSTAT, IOGA, JIRAMA, JOUVE, MADADEV, MAEP, MANAO, MEF, MEN, MESupRES, MFB, , MININTER, Min des Postes/Télécommunications et du Développement Numérique, NEOV MAD, Ny Havana, Madagascar National Parks, OMNITEC, ORANGE, OTME, PRACCESS, QMM Fort-Dauphin, SG Madagasikara SMMC, SMMEC, SNEDADRS Antsirabe, Sénat, Société d’Exploitation du Port de Toamasina (SEPT), SOFTWELL, Strategy Consulting, TELMA, VIVETEC, Société LAZAN’I BETSILEO, WWF, UGD, ARATO, MANAO, MNDPT, NG ACADEMY.NG, Relia …

* 1. **Débouchés professionnels et diplômés**

Les formations proposées par l’Ecole permettent aux diplômés d’être immédiatement opérationnels sur le marché du travail avec la connaissance d’un métier complet lié à l’informatique aux TIC.

L’Ecole apporte à ses étudiants un savoir-faire et un savoir-être qui les accompagnent tout au long de leur vie professionnelle. Elle a une vocation professionnalisante. Les diplômés en LICENCE et en MASTER issus de l’ENI peuvent faire carrière dans différents secteurs.

L’Ecole bénéficie aujourd’hui de 40 années d’expériences pédagogiques et de reconnaissance auprès des sociétés, des entreprises et des organismes. C’est une Ecole Supérieure de référence en matière informatique.

D’une manière générale, les diplômés de l’ENI n’éprouvent pas de difficultés particulières à être recrutés au terme de leurs études. Cependant, l’ENI recommande à ses diplômés de promouvoir l’entrepreneuriat en TIC et de créer des cybercafés, des SSII ou des bureaux d’études. Le tableau 4 représente les débouchés éventuels des jeunes diplômés.

Tableau 4. Débouchés éventuels des jeunes diplômés

|  |  |
| --- | --- |
| **LICENCE** | **MASTER** |
| * Analyste - Programmeur * Administrateur de site web/de portail web * Assistant Informatique et internet * Chef de projet web ou multimédia * Développeur Informatique ou multimédia * Intégrateur web ou web designer * Hot liner/Hébergeur Internet * Agent de référencement * Technicien/Supérieur de help desk sur Informatique * Responsable de sécurité web * Administrateur de réseau | * Administrateur de réseau et système * Architecture de système d’information * Développeur d’applications * Ingénieur réseau * Webmaster / Web Designer * Concepteur et réalisateur d’application * Directeur du système d’informations * Chef de projet informatique * Responsable de sécurité informatique * Consultant fonctionnel ou freelance |

* 1. **Ressources humaines**

Les ressources humaines sont citées ci-dessous selon leurs responsabilités :

* Directeur de l’Ecole : Monsieur MAHATODY Thomas, Docteur HDR
* Responsable de la Mention « Informatique » : Monsieur RABETAFIKA Louis Haja, Maître de Conférences
* Responsable de la Mention « Intelligence Artificielle » : Monsieur DIMBISOA William Germain, Maître de Conférences
* Responsable du Parcours « Génie Logiciel et Base de Données » : Monsieur RALAIVAO Jean Christian, Assistant d’Enseignement Supérieur et de Recherche
* Responsable du Parcours « Administration Systèmes et Réseaux » : Monsieur SIAKA, Assistant d’Enseignement Supérieur et de Recherche
* Responsable du Parcours « Informatique Générale » : Monsieur GILANTE Gesazafy, Assistant d’Enseignement Supérieur et de Recherche
* Responsable du Parcours « Gouvernance et Ingénierie de Données » : Madame RATIANANTITRA Volatiana Marielle, Maître de Conférences
* Responsable du Parcours « Objets Connectés et Cybersécurité » : Monsieur RAZAFIMAHATRATRA Hajarisena, Maître de Conférences

L’ENI compte quinze (15) enseignants permanents dont un (01) Professeur Titulaire, un (01) Professeur, un (01) Docteur HDR, huit (08) Maîtres de Conférences, quatre (04) Assistants d’Enseignement Supérieur et de Recherche, dix (10) enseignants vacataires, quarante un (41) personnel administratif.



# Chapitre 2. Présentation OpenData-Madagascar

|  |  |  |
| --- | --- | --- |
| **Rédigé par** | Serge Rakotoarivony  (Fondateur et Gérant d’Opendata-Madagascar) | [opendata.m@gmail.com](mailto:opendata.m@gmail.com) |
| **Nous contacter** | OpenData-Madagascar 0120 B 282 Ivory  Antsirabe 110 – Madagascar Tel : +261 34 08 728 10  Skype : excellenz.serge  Site web : [http://www.open-data-madagascar.branded.me](http://www.open-data-madagascar.branded.me/) | |
| **Situation Juridique** | **NUMERO D'IMMATRICULATION FISCALE : 2000705304**  **STATISTIQUE : 72301 12 2011 0 02890**  **REGISTRE DE COMMERCE ET DES SOCIETES** (Tribunal d’Antsirabe - Madagascar) **: A 00085** | |

CONFIDENTIALITE : Le présent document est protégé par les lois et les traités internationaux du copyright. Le nom OpenData-Madagascar® et son logo sont des marques déposées de la société OpenData-Madagascar dont le siège social est situé 1116 C 79 Mahazoarivo Antsirabe 110. Il est interdit de reproduire intégralement ou partiellement sur quelque support que ce soit la présente documentation sans l’autorisation expresse et préalable de notre accord.

OpenData-Madagascar est une plateforme de traitement externalisée de données.



Présentation et démarche

Nos solutions permettent aux entreprises d'externaliser les tâches à faible et moyenne valeur ajoutée, pour se concentrer sur leur cœur du métier.

OpenData-Madagascar intervient auprès de nombreux Sociétés, Entreprises, Bibliothèques,... sur des missions ponctuelles ou récurrentes.

OpenData-Madagascar s’engage à mettre à votre disposition des équipes dédiées à faible cout qui effectuent des taches chronophages liées à la gestion de vos flux.

* Nos services vous feront économiser du temps et de l’argent
* Ce service pouvant être stratégique pour votre entreprise
* Un processus de saisie de données fiable, vous pouvez faire confiance
* Services disponible 7j/7 (durant les jours fériés, une équipe sera à votre disposition)
* Réductions de vos coûts de productions car nos services à un taux incroyablement bas (Pas de minimum de facturation)
* Devis sous 2h, gratuit et sans engagement



Nos Services

## TRAITEMENTS DE DONNEES

### Conversion rétrospective, Rétroconversion des catalogues des bibliothèques en format Unimarc norme iso2709, encodage de caractère aux choix : Iso5426 ou UTF8

* **Récupérations ou Dérivation des notices Bibliographiques et notices d’Autorités,**

### Création, Qualification et Vérification des fichiers

* **Indexation, Extraction, Gestion et Traitement de flux de données : saisie en masse des Chèques, des avis de décès, des enquêtes, des coupons, des annuaires, des notices, des formulaires...**

### Saisir ou retranscrire sous format informatique des informations écrites (tapuscrits et manuscrites),

* **Saisies des pièces comptables,**

### Structuration des données, Enrichissements de données, Archivages, Conversion, Zoning, intégration des données et rétro-balisage des textes en XML, XML-EAD, SGML, html, csv...,

* **Découpe, indexation et nettoyage d'État-civil...**

### OCRisation, Dématérialisation des flux documentaires, Archivages électroniques,

* **Numérisation en masse des factures, des documents, des photos...,**

### Back-office web,

* **Traitement d'images,**

### Animation et modération des sites

* **Mise en ligne des articles (copywriting), des informations, des annonces, des avis etc.**
* **Sauvegarde des données informatiques**

## SOLUTIONS POUR DES ÉQUIPES HYBRIDES

### OpenData-Madagascar est conçu pour aider les entreprises à déléguer (sans embaucher) des équipes en télétravail

* **OpenData-Madagascar est la solution pour identifier un (des) talent à Madagascar**
* **Des solutions adaptées à vos besoins**

## INFOGERANCE

### Contrôles matériels et logiciels détaillés, diagnostic de votre architecture, préconisation initiale

* **Gestion du réseau des serveurs et des postes de travail, gestion des droits des utilisateurs, gestion de la messagerie**

### Prise de contrôle à distance, assistance en illimité

* **Paramétrage, configuration et suivi des firewall, anti-virus, anti-spam, surveiller les sauvegardes**

### Une équipe humaine de qualité et compétente :



Une prestation de qualité

OpenData-Madagascar dispose d’une expérience avérée en traitement numérique de l’information. Tous nos collaborateurs ont des compétences dans les domaines de traitement de données informatique. Un personnel composé d’experts, techniciens cherchant l’excellence dans le domaine du traitement de données.

### Les principales caractéristiques de notre offre sont :

* 1. **Le respect des spécifications :**

De par son expérience, OpenData-Madagascar est en mesure de prendre en compte l’ensemble des contraintes de votre projet et nos processus nous permet de suivre tous types d'opérations quelle que soit leur nature et les volumes concernés.

### Un engagement sur la qualité des données livrées :

Nos procédés de production reposent sur des chaînes de production éprouvées. Chaînes de production paramétrées et contrôlées en tenant compte des données fournies par le client afin de produire les données attendues.

Les fichiers livrés seront conformes aux informations contenues dans les originaux (pas d’information perdue, sous réserve qu’elle soit présente et visible).

### Un engagement sur le respect des délais :

OpenData-Madagascar est particulièrement concernée par la satisfaction de ses clients et par la tenue de tous ses engagements (notamment, le respect des délais). Les ressources matérielles et logicielles sont en permanence adaptées afin d’offrir une capacité de traitement adéquate à toutes les prestations (logistique, numérisation, livraison des données).

### Un engagement sur la qualité de service :

Dans le cadre d’un projet dont l’objectif est important, une qualité de service client sera mise en place pour tenir informé continuellement du processus, du déroulement et des aspects techniques de la prestation. Des tableaux de bord, de suivi, de planning, d’objectifs ainsi que des outils de gestion de projet (MS Projet et outils de gestion de production adaptée à l’activité de la société OpenData-Madagascar) seront utilisés par l’ensemble de notre équipe. Notre chef de projet est disponible à tout moment pour apporter des précisions sur la prestation en cours. OpenData-Madagascar met un point d’honneur a apporté à ses clients écoute, soutien et conseils tout au long de la prestation. Pour que le travail se déroule dans les meilleures conditions, chaque étape est validée point par point et des relations collaboratives sont rapidement instaurées afin de démarrer dans les délais.



Charte éthique

OpenData-Madagascar a la volonté d’appuyer son développement sur un ensemble de valeurs et de principes qui affirment clairement son engagement éthique et sa responsabilité dans tous ses domaines d’activité.

La société a pour principe la loyauté à l’égard de ses fournisseurs et de ses clients afin d’instaurer et de maintenir des relations de confiance durables.

Chaque membre du personnel met en œuvre ses compétences professionnelles au meilleur niveau possible et contribue activement au respect des spécificités du client.

### La responsabilité d’OpenData-Madagascar s’exerce notamment vis-à-vis de :

***Ses clients,*** auxquels il s’engage à fournir des produits et services de qualité, dans le strict respect des normes de sécurité et d’archivage.

***Ses collaborateurs.*** OpenData-Madagascar est attentif au développement de leurs compétences ainsi qu’à leurs conditions de travail. Il est attendu d’eux qu’ils respectent les principes d’actions et de comportement : établir des rapports clairs et honnêtes avec les clients et fournisseurs, respecter la confidentialité des informations professionnelles, être attentifs à la qualité des relations humaines au sein des équipes de travail.

### Plan assurance qualité

Tout au long des étapes de réalisation de la prestation, OpenData-Madagascar met en œuvre tous les moyens nécessaires à la réalisation de la prestation dans les meilleures conditions de sécurité et de qualité.

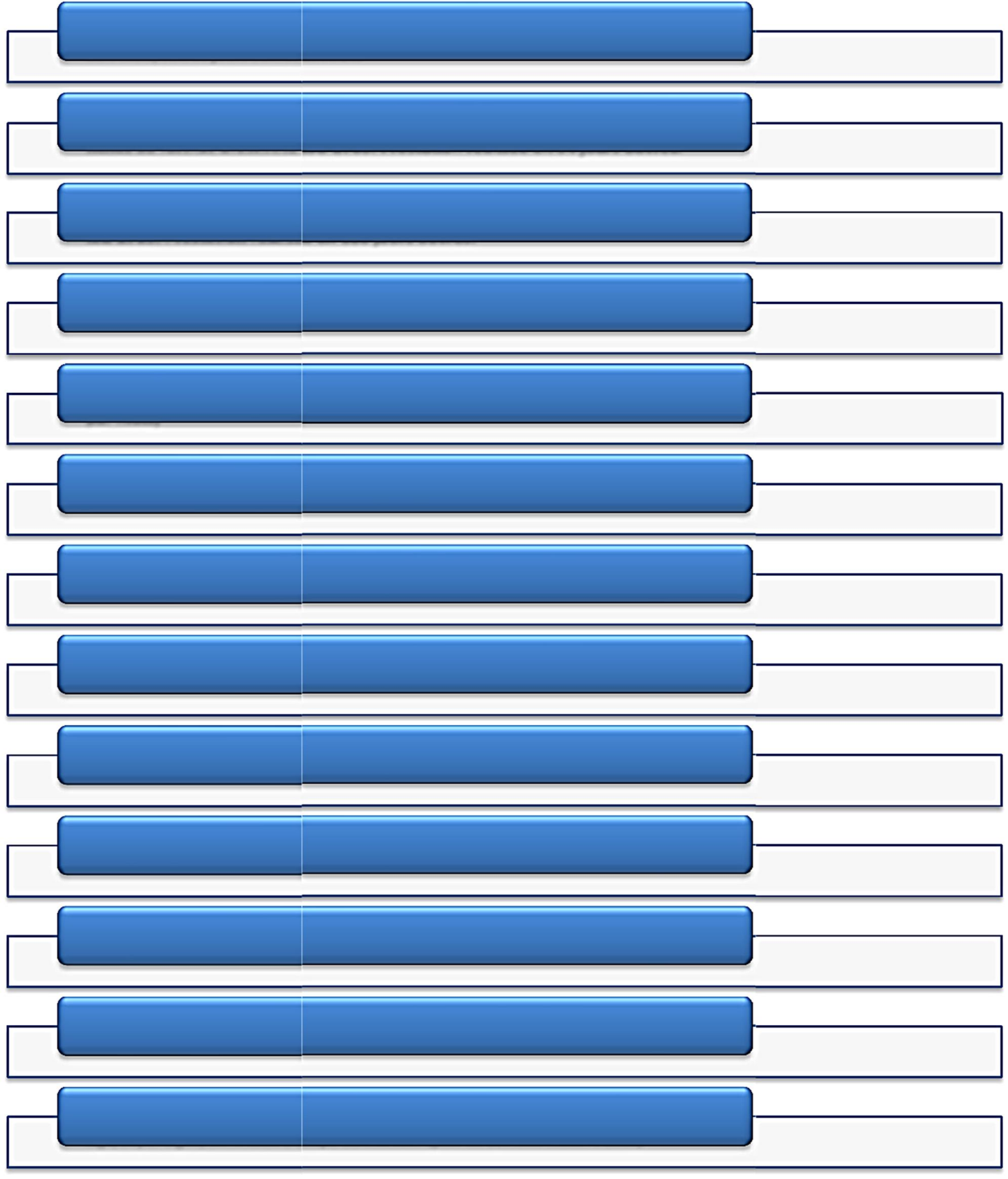


Références

OpenData-Madagascar a réalisé de nombreux projets. Nous joignons avec cette note technique l’extrait des travaux déjà effectuées.

Veuillez trouver ci-dessous les listes des projets réalisés par OpenData-Madagascar (N.B. : Liste non exhaustive)

**Rétroconversion et informatisation des catalogues de la Bibliothèque médicale de l’Hôpital Ste Anne au format UNIMARC de 19 000 fiches d’inventaire pendant une durée de 4 mois, soit 80 jours ouvrés environ.**



**Rétroconversion et informatisation des catalogues de la Bibliothèque francophone multimédia d la ville de Limoges au format UNIMARC de 20 000 fiches au format UNIMARC ISO 2709. Prestation réalisée en 94 jours ouvrés.**

**Rétroconversion et informatisation des catalogues de la Médiathèque de Bar le Duc au format UNIMARC d 40 000 notices à rétroconvertir au format UNIMARC ISO 2709. Prestation réalisée en 120 jours ouvrés.**

**Indexation de thesaurus d’Archives Departementales de La Seine-Maritime**

**Depuis le mois de juin 2015, nous assurons quotidiennement la saisie ainsi que la mise en ligne des avis de décès (~40000 Avis de décès traiter et mis en ligne par mois)**

**Depuis le mois de juin 2015, nous indexons quotidiennement des condoléances**

Saisie des pièces comptables

**Nous avons déjà travaillé avec la société BNF, Prestalia, Fondadev, etc. pour des traitements des données informatique**

Création, Qualification et Vérification des données pour le compte de Fondadev, IFFRES, Global Exam, TS-Data... (plus de 70.000 contacts)

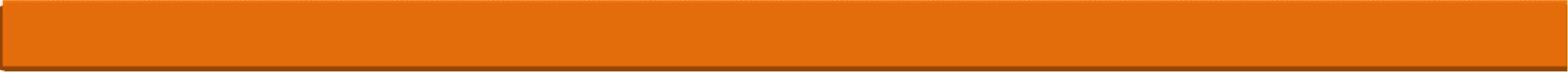
Indexation et Extraction manuelle de données pour le compte de Prestalia, BNF, Allmakes,...

Conversion rétrospective de 5237 fiches en format Unimarc norme iso-2709 des catalogues des Arvhives Départmentale de Touraine

Saisie et mettre en ligne des biens à louer : Villa, Maison, Appartement... (+ de 25000 annonces traitées)

Saisie et mettre en ligne des fiches match de Foot pour tous les compétitions dans tous les continents, ce depuis 1928 jusqu'à ce jour (LDC, Ligue 1, Première Ligue, Bundesligua, Laliga, Liga, Super Ligue, Eliminatoire coupe de monde... (plus de 125.000 match traités)

### Méthodologie de conduite du projet



### Environnement de travail

Hormis la saisie et les relectures humaines, toutes les interventions sont automatisées au maximum. Nous utilisons plusieurs utilitaires développés en interne qui facilitent le traitement :

Des utilitaires développés sous VB (masque de saisie, utilitaire de comparaison, assemblage des fichiers)

### Processus de qualité

Nous travaillons avec un logiciel développé en interne, à partir duquel nous créons un masque de saisie sur mesure avec des informations et une structure répondant aux consignes de saisie du client. Ce logiciel nous permet de limiter le nombre d’erreur (surtout pour le codage et la saisie) en étant plus pratique et plus rapide. Notre traitement est sécurisé par l’intermédiaire de diverses procédures de contrôles et gestions des anomalies.

### Mode de communication

Nous allons installer un moyen de communication efficace et fiable pour permettre d’établir une relation de travail et de confiance :

* Mail : les échanges de rapports qui permettent d’assurer tous les historiques
* Une espace Workspace (de type Windows live, SharePoint…) qui permet de partager entre OpenData et les clients des fichiers de récolement, des suivis de projet, des rapports, tout type de document de suivi… cette espace permet également de mettre à jour les fichiers par le client avec des remarques éventuelles.
* Pour les transferts des fichiers (FTP, Hubic, WeTransfer, Google Drive, etc.)
* Vidéo conférence (sur SKYPE par exemple)

### Les moyens humains

OpenData-Madagascar assure la continuité du service pendant toute la durée du marché, et met à votre service l’équipe dont la décomposition est comme suit :

### Chef de projet

* Garant de l'organisation et de la mise en œuvre des opérations, le chef de Projet assure la coordination de toutes les phases de production, de suivi et d'évolution du projet. Il est votre interlocuteur unique.
* Son rôle consiste à suivre de manière opérationnelle le déroulement du projet, vérifier le respect du planning préparer les points à aborder en réunion, et les validations.
* Il est expérimenté, spécialisé depuis au moins 5 ans. Il sera un interlocuteur technique, et administratif et pourra répondre à toutes les questions, pour toute la durée du projet.
* Sa mission consiste à piloter le projet :

 Faire respecter des orientations définies dans le cahier des charges et la proposition

 Avoir un rôle de conseil à côtés du client,

 Faire respecter le planning,

 Prendre les décisions et définir les plans d'action,

 Coordonner et évaluer les interventions associées au projet,

 Superviser les phases d'organisation,

 Communiquer avec les équipes du client,

 Mettre en place les outils de suivi (tableaux d'actions, récolement, comptes rendu, etc.)

 Assurer la validation sur proposition du chef de projet client de chaque phase et des livrables associés.

 Etre le garant de la bonne fin du projet,

 Etre l’Interlocuteur privilégié du client.

### Equipe de saisie

OpenData-Madagascar met à votre disposition en plus du chef de projet :

* Contrôleurs qualité
* Superviseurs
* Correcteurs
* Opérateurs de saisie

Nos opérateurs sont tous francophones et ont au moins 6 années d’expérience. Ils ont des connaissances en latin, en vieux français et en langues étrangères.



Méthode de traitement

### Nous vous proposons deux types de méthodes :

* + 1. **L’outsourcing-Saisie**

Les documents ou les fiches seront traités à partir de leurs copies numériques.

 Les fiches ou les documents à traiter seront scannées sous forme d'images ou en PDF, ou autres

 Les informations à traiter sont dans un fichier sous forme de listing (Word, Excel) ou dans une base de données (Access, XML, csv, etc.)

### L’outsourcing-IPAC (Image public access catalog)

 Les fiches, les documents ou les informations à traiter sont accessible en ligne ou dans le Cloud.



Processus de traitement

### A noté que nos processus nous permet de suivre tous types d'opérations quelle que soit leur nature et les volumes concernés :

1. **Avant production**

 Analyser le projet et réaliser la mise en place pour la production (création outils de traitement informatique ou masque de saisie,…)

 Étudier le Cahier des Clauses Techniques Particulières (CCTP) et élabore les consignes internes,

 Former les équipes,

 Résoudre les complexités, problèmes parcouru durant les tests et proposé aux clients des solutions, améliorations, etc.

 Établir un planning de livraison hebdomadaire pour le client et assurer son respect,

 Placer un système pour mettre à jour les connaissances des équipes en fonction des retours clients,

 Objectif : Fournir un travail de qualité répondant aux normes,

 Feu vert du client pour démarrer la production.

### Pendant la production

Pour éviter toutes fautes pendant les traitements, OpenData-Madagascar est en mesure de vous proposer deux méthodes de traitements. Ces deux méthodes de saisie ont fait ses preuves pour limiter les erreurs et les oublis de données, l’objectif c’est d’avoir un traitement soigné pour chaque Lot traité.

* 1. La méthode **« Double saisie plus comparaison »** est une technique qui consiste à faire saisir une fiche source par deux personnes indépendamment l’une de l’autre et sur deux postes différents.
  2. Méthode **« Système binôme »** : La technique consiste à 2 personnes travaillant sur un (01) dossier (ou 1 Batch) en même temps sur un PC, mais avec 2 écrans, l'un pour le traitement, l'autre pour la visualisation et la vérification (ou inversement), l'écran où ces deux personnes travaillent est partagé via un logiciel comme TeamViewer. Un Chef de projet suit à partir de l’écran partager les travaux effectués par ces 2 personnes, toutes les doutes et les soucis seront traités et résolu au moment dès qu’elle se présente (en temps réel). L'Avantage c'est que le risque d’une erreur est très minime.

### Après la production

 Assistance après les livraisons.

 Le contrôle de l’exhaustivité sera faite en confrontant le nombre des fiches ou documents reçu et celui traités.

 Tous les lots seront assemblés en un seul fichier.

 La livraison finale sera effectuée sur un support (DVD, USB ou Disque Dur).



Sauvegarde de sécurité

Notre serveur en Raid 5 nous permet d’accéder aux sauvegardes sans délai via notre matériel de secours. Nous mettons aussi en place une sauvegarde sur bande s’il est nécessaire.

OpenData-Madagascar s’engage à sauvegarder les résultats de saisie de données sur ses serveurs pendant les douze mois suivant la livraison. Après ce délai les fichiers seront détruits définitivement (après validation client).



Nos engagements

* Nous sommes à votre disposition 7j/7 et tout au long de la prestation
* Nous pouvons créer pour vous une équipe de nuit
* Mettre en place des contrôles à toutes les étapes de la prestation
* Respect de la confidentialité
* Service de relecture inclut
* OpenData-Madagascar s’engage à ne pas reproduire les documents en dehors de la prestation et à ne pas les diffuser de quelque manière que ce soit
* Aucun droit d’auteur ne sera revendiqué sur les données traitées.



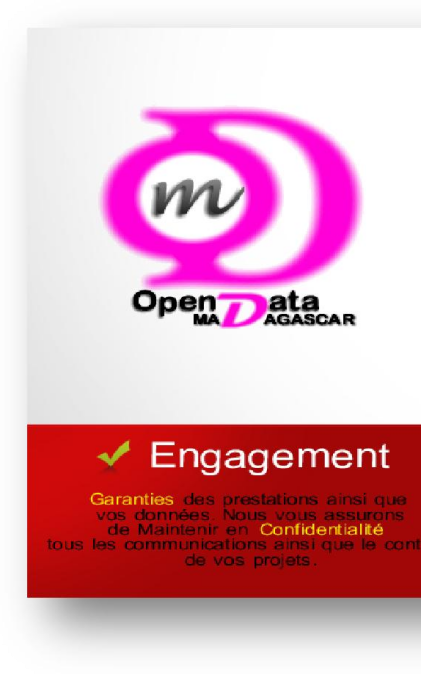
Secret professionnel

OpenData-Madagascar traitera les données du Client dans les meilleures conditions de sécurité et de confidentialité. Aucune information en provenance du Client ne pourra être communiquée par OpenData- Madagascar sans accord écrit préalable du Client.



Droit de citation

Sauf opposition expresse du client, OpenData-Madagascar a le droit de le citer dans ses références commerciales.



https:/[/www.facebook.c](http://www.facebook.com/opendatamadagascar)o[m/opendatamadagascar](http://www.facebook.com/opendatamadagascar)

-



- Follow Us

<http://www.linkedin.com/in/opendatamad/> <http://www.viadeo.com/fr/profile/open.data.madagascar>

https://twitter.com/opendatamada

# Chapitre 3. Description du projet

# **3.1. Formulation**

Le projet consiste à étudier la mise en place d’un pipeline CI/CD et d’un système d’orchestration adaptés à une application reposant sur une architecture microservices. L’objectif global est d’automatiser les processus d’intégration, de déploiement et de validation, afin d’assurer la fiabilité, la rapidité et la cohérence des mises en production.

### **3.2. Objectif et besoins de l’utilisateur**

L’entreprise **Tetika** propose une large gamme de produits logiciels destinés à la gestion de plusieurs domaines d’activité. Chacune de ces applications fonctionne de manière indépendante, bien qu’elles soient toutes centralisées dans une licence principale regroupant l’ensemble des solutions offertes. Ces logiciels connaissent une évolution constante, ce qui implique des mises à jour et des déploiements fréquents. Ces opérations, actuellement réalisées de manière manuelle, concernent aussi bien la mise en production que les tests post-déploiement.

Cependant, cette méthode manuelle présente plusieurs limites. Elle accroît les risques d’erreurs humaines et rallonge considérablement les délais de déploiement. De plus, l’absence d’automatisation des tests favorise l’apparition de régressions lors des mises à jour successives. En cas de dysfonctionnement après une mise en production, le retour vers une version antérieure fonctionnelle s’effectue également manuellement, ce qui engendre une perte de temps et une baisse de productivité. L’entreprise fait aussi face à des pics de charge récurrents, provoquant des surcharges au niveau des serveurs et une dégradation ponctuelle des performances.

Face à ces difficultés, **Tetika** ressent la nécessité d’optimiser et de moderniser son processus de déploiement afin de garantir une meilleure fiabilité et une disponibilité continue de ses applications. L’objectif principal de ce projet est donc de concevoir et de mettre en place une chaîne d’intégration et de déploiement continus (CI/CD) adaptée à l’environnement de production de l’entreprise. Cette approche vise à automatiser les différentes étapes du cycle de vie des applications, allant du développement au déploiement, tout en assurant la cohérence et la stabilité du système.

L’automatisation des processus permettra de réduire les interventions manuelles, d’accélérer la livraison des nouvelles fonctionnalités et d’améliorer la qualité globale des produits déployés. De plus, l’intégration d’un outil d’orchestration offrira une meilleure gestion des microservices, garantissant leur disponibilité et leur mise à l’échelle dynamique en fonction des besoins de l’entreprise. Enfin, la mise en œuvre de ce projet contribuera à instaurer un environnement de travail plus structuré et reproductible, facilitant ainsi la collaboration entre les équipes techniques et l’intégration de nouveaux développeurs au sein de l’organisation.

### **3.3. Moyens nécessaires à la réalisation du projet**

La mise en œuvre de ce projet repose sur trois facteurs essentiels : **humain**, **matériel** et **logiciel**. Ces éléments constituent l’ensemble des ressources indispensables pour assurer la conception et la mise en place de l’infrastructure DevOps de production.

#### **a. Facteur humain**

Le facteur humain joue un rôle déterminant dans la réussite du projet.  
 Un **responsable technique** ou **superviseur DevOps** est chargé de la configuration du pipeline d’intégration et de déploiement continus, ainsi que de la gestion du cluster Kubernetes. Il veille au bon déroulement des déploiements en production et à la supervision de l’infrastructure.

Un **stagiaire ou ingénieur DevOps junior** participe à l’automatisation des tâches, à la conteneurisation des applications et au suivi des déploiements sous la supervision du responsable.  
 Enfin, les **développeurs** interviennent pour assurer la maintenance du code source, la mise à jour des microservices et la gestion des dépendances.

#### **b. Facteur matériel**

Le projet nécessite un environnement matériel capable de supporter la virtualisation et l’orchestration en production.  
 Une **machine hôte** équipée d’au moins **8 Go de RAM** et d’un **processeur quad-core** est requise pour exécuter les machines virtuelles à l’aide de **Vagrant** et **VirtualBox**.

Deux machines virtuelles principales sont prévues :

* une jouant le rôle de **nœud maître** pour la gestion du cluster Kubernetes ;
* une ou plusieurs agissant comme **nœuds travailleurs (workers)** pour l’exécution des conteneurs applicatifs.

Une **connexion Internet stable** est indispensable pour le téléchargement des images Docker, la synchronisation avec GitHub, et la mise à jour des dépendances nécessaires à l’exécution des microservices en production.

#### **c. Facteur logiciel**

Le bon fonctionnement de l’infrastructure repose sur un ensemble cohérent d’outils logiciels open source, regroupés selon leurs fonctions principales :

* **Outils de développement :**
  + *Python Flask* : framework léger pour la création du microservice en Python.
  + *Node.js* : environnement d’exécution JavaScript pour les microservices Node.
  + *MySQL* : système de gestion de base de données relationnelle pour le stockage persistant des données.
  + *Visual Studio Code (VSCode)* : environnement de développement intégré utilisé pour la rédaction et la gestion du code.
* **Outils d’intégration et de déploiement continus :**
  + *Git & GitHub* : pour la gestion du code source et le déclenchement des workflows d’automatisation.
  + *Docker* : pour la conteneurisation et l’isolation des applications.
  + *Kubernetes* : pour l’orchestration et la gestion dynamique des conteneurs sur les nœuds du cluster.
  + *GitHub Actions* : pour l’automatisation du pipeline CI/CD et le déploiement automatique en production.
  + *Vagrant* : pour la création et la configuration des machines virtuelles utilisées comme serveurs du cluster.

Ces outils, tous **open source**, permettent de construire une infrastructure **fiable, scalable et économique**, répondant pleinement aux besoins d’automatisation et de déploiement continu du projet.

### **3.4. Résultat attendu**

La mise en place de ce projet vise à produire une infrastructure DevOps complète et opérationnelle, permettant d’automatiser le cycle de vie des applications en architecture microservices au sein d’un environnement unique de production.

L’objectif principal est de parvenir à une **intégration et un déploiement continus (CI/CD)** pleinement fonctionnels, afin d’assurer un flux de livraison rapide, fiable et reproductible. Grâce à cette automatisation, chaque modification apportée au code source doit pouvoir être intégrée, testée, conteneurisée, puis déployée automatiquement dans le cluster de production sans intervention manuelle.

Le résultat attendu se traduit par plusieurs réalisations concrètes :

* **Une infrastructure conteneurisée** à l’aide de **Docker**, garantissant l’isolation des microservices et la portabilité du système sur n’importe quel hôte compatible.
* **Un cluster Kubernetes opérationnel**, orchestrant l’exécution des conteneurs sur plusieurs machines virtuelles (nœud maître et nœuds travailleurs). Ce cluster assure la disponibilité, la scalabilité et la tolérance aux pannes de l’application en production.
* **Un pipeline CI/CD automatisé via GitHub Actions**, permettant d’exécuter automatiquement les étapes de build, test, création d’images Docker, et déploiement continu vers le cluster Kubernetes à chaque mise à jour du code source sur GitHub.
* **Une supervision simplifiée du déploiement**, où le responsable technique peut suivre en temps réel les différentes étapes du processus et intervenir uniquement en cas d’anomalie détectée.
* **Une réduction du temps de mise en production**, puisque les tâches manuelles (test, build, déploiement, redémarrage des services) sont remplacées par des workflows automatisés.
* **Une meilleure fiabilité du système**, grâce à la traçabilité des versions et à la standardisation des opérations de déploiement.

En somme, le projet doit permettre à Tetika de disposer d’un **système DevOps complet**, intégrant les bonnes pratiques d’automatisation, de conteneurisation et d’orchestration, tout en assurant la continuité de service et la stabilité des applications en production.  
  
Parfait, restons fidèles à la fiche et poursuivons avec **3.5 Chronogramme de travail**. Voici une rédaction structurée adaptée au contexte du mémoire :

### **3.5 Chronogramme de travail**

Le chronogramme de travail permet de planifier les différentes étapes du projet et de visualiser l’avancement attendu sur la durée du stage/mémoire. Il est construit sur une base hebdomadaire et mensuelle selon les tâches définies dans la méthodologie.

|  |  |  |
| --- | --- | --- |
| **Phase** | **Tâches principales** | **Durée estimée** |
| **Phase 1 : Analyse et étude préliminaire** | - Compréhension du contexte et des besoins- Étude des outils existants et documentation | Semaine 1 à 2 |
| **Phase 2 : Planification et conception** | - Définition du pipeline et des étapes clés- Préparation des modèles et schémas | Semaine 3 à 4 |
| **Phase 3 : Réalisation / Développement** | - Mise en place des outils et scripts- Développement des composants nécessaires- Tests unitaires et intégration | Semaine 5 à 8 |
| **Phase 4 : Suivi et ajustements** | - Contrôle de la qualité des livrables- Ajustements en fonction des problèmes rencontrés | Semaine 9 à 10 |
| **Phase 5 : Rédaction et restitution** | - Compilation des résultats- Rédaction du rapport de mémoire- Présentation finale | Semaine 11 à 12 |

**Remarques :**

* Les durées sont indicatives et peuvent être ajustées selon les contraintes rencontrées.
* Les points de validation avec le responsable sont prévus à la fin de chaque phase pour assurer le suivi et la qualité du projet.

# 

# PARTIE II : ANALYSE DE L’EXISTANT

# Chapitre 4. Critique de l’existant

### **Chapitre 3 : Analyse préalable**

#### **3.1 Analyse de l’existant**

**Organisation actuelle :** L’entreprise Tetika exploite un écosystème logiciel composé de plusieurs produits couvrant différents domaines fonctionnels : gestion des utilisateurs, produits, commandes, inventaire, etc. Ces logiciels sont :

* Développés indépendamment par différentes équipes.
* Hébergés sur des machines virtuelles locales, sans orchestration centralisée.
* Déployés manuellement via des scripts shell et des transferts FTP/SCP.

Les déploiements manuels entraînent des problèmes fréquents :

* **Erreurs humaines** : oublis, mauvaises configurations, versionning incorrect.
* **Retards dans les mises en production** : chaque microservice nécessite une intervention humaine pour son déploiement.
* **Manque de traçabilité** : difficulté à savoir quelle version exacte est en production et quelles modifications ont été appliquées.
* **Tests post-déploiement lents** : l’absence de tests automatisés retarde la détection des régressions et des bugs critiques.

**Topologie réseau actuelle :**

* Les machines hôtes communiquent via un réseau interne classique (LAN), avec un nœud central pour certaines bases de données.
* Chaque microservice expose ses endpoints sur un port dédié.
* Les flux réseau ne sont pas surveillés en temps réel, ce qui complique la détection des surcharges ou anomalies de performance.

#### **3.2 Critique de l’existant**

L’existant présente des limitations majeures pour une entreprise souhaitant évoluer vers une infrastructure DevOps moderne :

1. **Absence de pipeline CI/CD :**
   * Les tests, builds et déploiements sont réalisés manuellement, augmentant le risque d’erreurs et allongeant les délais.
   * Le retour à une version stable est complexe et peu fiable.
2. **Pas d’orchestration centralisée :**
   * Les microservices ne peuvent pas être automatiquement scalés selon la charge.
   * La haute disponibilité n’est pas assurée.
3. **Maintenance difficile :**
   * La mise à jour ou la suppression d’un service nécessite une intervention humaine et une planification spécifique.
   * Les dépendances entre microservices sont peu documentées, rendant les déploiements sensibles aux erreurs.
4. **Surveillance et supervision limitées :**
   * Aucune métrique centralisée pour suivre la santé des services.
   * Difficulté à anticiper les incidents liés à des pics de trafic.

#### **3.3 Conception avant-projet**

**Proposition de solutions :**

1. **Pipeline CI/CD automatisé via GitHub Actions :**
   * Déclenchement automatique du build à chaque push sur le repository GitHub.
   * Exécution des tests unitaires et d’intégration.
   * Construction des images Docker pour chaque microservice.
   * Déploiement automatique sur le cluster Kubernetes.
2. **Conteneurisation avec Docker :**
   * Isolation des microservices pour éviter les conflits de dépendances.
   * Portabilité sur n’importe quelle machine hôte compatible Docker.
   * Gestion des versions via des tags d’image Docker.
3. **Orchestration avec Kubernetes :**
   * Déploiement et gestion automatique des conteneurs.
   * Scalabilité horizontale : ajout ou suppression de pods selon la charge.
   * Résilience : redémarrage automatique des conteneurs en cas de panne.
   * Load balancing intégré pour distribuer les requêtes entre les pods.
4. **Provisionnement avec Vagrant et VirtualBox :**
   * Création rapide d’un environnement de test/production sur des machines virtuelles locales.
   * Réplication exacte de l’infrastructure cible pour éviter les erreurs liées à la configuration.

**Comparaison des solutions proposées :**

|  |  |  |  |
| --- | --- | --- | --- |
| **Solution** | **Avantages** | **Inconvénients** | **Justification pour Tetika** |
| GitHub Actions (CI/CD) | Automatisation complète, intégration directe avec GitHub | Dépendance à la plateforme GitHub | Facilite l’intégration continue et réduit les erreurs humaines |
| Docker | Isolation, portabilité, versionning des microservices | Nécessite des compétences techniques pour la gestion des images | Garantit la reproductibilité et la cohérence entre environnements |
| Kubernetes | Scalabilité, haute disponibilité, orchestration | Complexité de configuration | Assure la disponibilité et la tolérance aux pannes des services |
| Vagrant | Provisionnement rapide et identique des VM | Consommation en ressources | Permet de simuler la production sur une machine locale avant déploiement réel |

### **Chapitre 4 : Choix des outils**

#### **4.1 Choix des outils de modélisation et de développement**

**Python Flask :**

* Framework léger, idéal pour créer rapidement des microservices.
* Permet de définir facilement des routes, gérer les requêtes HTTP et interagir avec des bases de données.
* Compatible avec Docker pour la conteneurisation.

**Node.js :**

* Exécution JavaScript côté serveur, adapté aux microservices nécessitant des opérations asynchrones et des I/O intensives.
* Large écosystème de modules via npm.
* Facilement intégrable dans un pipeline CI/CD et conteneurisable.

**MySQL :**

* Base de données relationnelle robuste et mature.
* Supporte le stockage persistant des données et les transactions ACID.
* Facilement déployable dans des conteneurs Docker pour chaque microservice nécessitant une DB dédiée.

**Visual Studio Code (VSCode) :**

* IDE polyvalent, léger et extensible.
* Plugins pour Docker, Git, Python, Node.js et Kubernetes.
* Permet une gestion complète du code et du workflow DevOps depuis un seul environnement.

#### **4.2 Choix des outils d’intégration et de déploiement**

**Git & GitHub :**

* Gestion centralisée du code source.
* Déclenchement automatique des workflows CI/CD à chaque commit.

**Docker :**

* Conteneurisation des microservices avec gestion des versions via tags.
* Isolation totale pour éviter les conflits entre dépendances.

**Kubernetes :**

* Orchestration des conteneurs : déploiement, scalabilité et redondance.
* Services et Ingress pour exposer les microservices au réseau.
* ConfigMaps et Secrets pour la configuration et la sécurité.

**GitHub Actions :**

* Pipeline CI/CD automatisé : build, test, image Docker et déploiement.
* Intégration directe avec les repositories GitHub.
* Possibilité de créer des workflows conditionnels et parallèles.

**Vagrant :**

* Création et configuration de machines virtuelles locales pour simuler l’environnement de production.
* Permet de tester la configuration du cluster Kubernetes avant déploiement réel.

# Chapitre 5. Proposition de solution

# PARTIE III : CHOIX DES OUTILS

# Chapitre 6. Comparatifs des outils disponibles

#### 6.1. Outils d’automatisations

Les technologies d’automatisations et les plateformes de deploiement continues existent en grand nombre grace a l’avance des technologies.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Jenkins** | **GitLab CI/CD** | **Azure** |
| Integration avec VCS | Github via plugins | Integre avec GitLab | Integre avec Azure Repos |
| Automatisation / Configuration | Groovy scripts  Pipeline as code | Fichiers YAML | YAML  Interface graphique |
| Open Source | Oui | Oui | Non |
| Limite de stockage | Dependant du systeme | Dependant du plan | Limite par compte Azure |
| Cout d’utilisation | Gratuit | Plan gratuit disponnible, tarification pour fonctionnalites avancees | Inclus dans l’abonnement Azure, tarification supplementaire pour certains services |

Table 7 : Outils de CI/CD

Nous allons opter pour Jenkins vis-a-vis de sa communaute tres etendue (open source), le stockage flexible dependant du systeme dans lequel on l’installe. Le suivi de la progression des taches sont visible grace a l’interface graphique du pipeline. Le cout d’utilisation etant gratuite, permettant ainsi d’economiser sur le long terme.

#### 6.2. Outils d’Infrastructures as Code

L’infrastructure as Code permet de mettre en place une infrastructure pour un projet en tant que code, c’est a dire que l’installation des dependances, les ressources et les services se font via un programme. Cela permet de gagner du temps en deplacement et verification des ressources par l’administrateur. Il suffira de lancer l’IaC pour un projet et l’outil met en place automatiquement les configurations necessaires.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Outils** | **Langage supportee** | **Prix** | **Communaute** | **Difficulte (1-5, 5 etant le plus eleve)** |
| Terraform | HCL | Gratuit | Tres active | 3 |
| Ansible | YAML | Gratuit | Tres active | 2 |
| Chef | Ruby | Gratuit (Version communautaire)  Payant (Version entreprise) | Active | 4 |

Table 8 : Outils d'IaC

Bien que Terraform soit un outil d’IaC tres populaire, la difficulte d’utilisation et le support du langage pour le programmer ne sont pas a la portee d’un aspirant. Aussi, nous nous proposons d’utiliser Ansible en raison de la programmation en YAML facile a comprendre et a mettre en place.

#### 6.3. Outils de deploiements

Le deploiement de l’application constitue l’etape finale du processus de developpement d’application. Il est donc necessaire de choisir l’outil adequat pour permettre le lancement en production de l’application d’une maniere optimale.

|  |  |  |  |
| --- | --- | --- | --- |
| **Caracteristiques** | **Cloud** | **Conteneurisation** | **Serveur Linux VM** |
| Ressources materielles | Utilise les ressources materielle du data center.  Evolutif en fonction des besoins | Partages les ressources materielles du serveur hote avec les autres conteneurs.  Utilise une partie des ressources systemes | Alloue les ressources materielles specifiques a la machine virtuelle pour les applications. |
| Base de donnees | Service de base de donnees disponnible en tant que service gere | Peut etre execute dans un conteneur ou sur la machine hote avec des configurations supplementaires | Peut etre execute directement sur le serveur |
| Stockages des donnees | Presistant et distribues pour garantir la disponnibilite des donnees | Persistant ou ephemeres selon la configuration du conteneur | Persistantes sur le serveur |
| Bandes passantes | Evolutive en fonction des besoins | Depend des capacites de l’hote et de l’infrastructure reseau | Depend des capacites du serveur et de l’infrastructure reseau |
| Prix | Base sur la consommation des ressources et des services utilises | Gratuit (version communauteaire)  Payant (version d’entreprise) | Image telechargeable gratuitement  Outils de deploiement open source |
| Exemples | AWS, Azure, Google Cloud | Docker, Kubernets | Ubuntu Server, CentOS, Debian |

Table 9 : Outils de deploiement

Le Cloud computing prend un grand place sur l’hebergement d’application de nos jours. Il presente neanmoins un inconvenient majeur pour le PME en raison du politique de Pay As You Use.

La conteneurisation quant a elle presente des solutions simple, legere et adequates pour une application evolutive, mais certaines applications ne sont pas conteneurisable. Pour palier a cela, nous allons utiliser la virtualisation d’un serveur linux pour heberger et deploier les applications.

# PARTIE IV : MISE EN OEUVRE

# Chapitre 7. Mise en oeuvre proprement dit

Tout au long de la mise en oeuvre, nous allons nous centrer sur la CI/CD d’une application en NodeJS avec Jenkins installe sur une machine linux de la distribtion debian : Kali Linux.

Ce systeme d’exploitation possede deja de nombreux outils necessiares a la mise en place du projet tel que Git ou SSH. Nous n’aurons donc plus besoins d’installer ces outils.

De plus, Kali Linux est dote d’une interface utilisateur, il est donc plus facile a utiliser pour un aspirant.

#### 7.1. Installation de Jenkins

Jenkins est un outil en open source qui vise à automatiser les tâches associées au développement logiciel. Son utilisation principale réside dans la gestion de l'intégration continue et du déploiement continu (CI/CD).

1. Installation de Jenkins

* Importer la cle GPG de Jenkins dans le systeme et ajouter l’URL de referentiel de Jenkins dans le fichier sources:

$ wget -q -O - https://pkg.jenkins.io/debian/jenkins.io.key | sudo apt-key add -

$ sudo sh -c 'echo deb https://pkg.jenkins.io/debian-stable binary/ > /etc/apt/sources.list.d/jenkins.list'

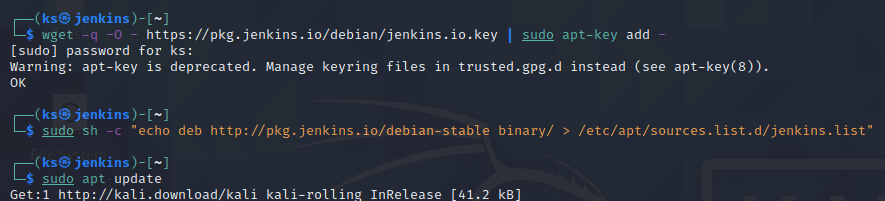


Figure 2 : Import cle GPG et mis a jour des paquets

* Installer Jenkins:

$ sudo apt update

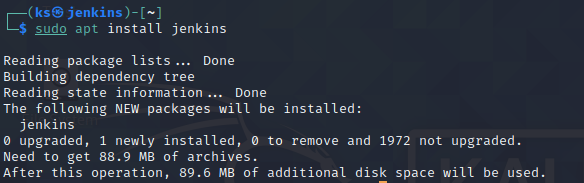


Figure 3 : Installer Jenkins avec apt

* Activer le service:

$ sudo systemctl enable jenkins

$ sudo systemctl start jenkins

Ces commandes activeront le service jenkins a chaque demarrage du systeme. Pour acceder a l’interface web de jenkins, il uffit d’ouvrir un navigateur et taper dans la barre d’adresse: **<http://localhost:8080>** ou <http://Adresse_IP:8080>, avec Adresse\_IP l’adresse IP de la machine sur laquelle jenkins est actif.

L’interface demandera le mot de passe de l’administrateur par defaut. On le trouvera en lancant la commande :

$ sudo cat /var/lib/jenkins/secrets/initialAdminPassword

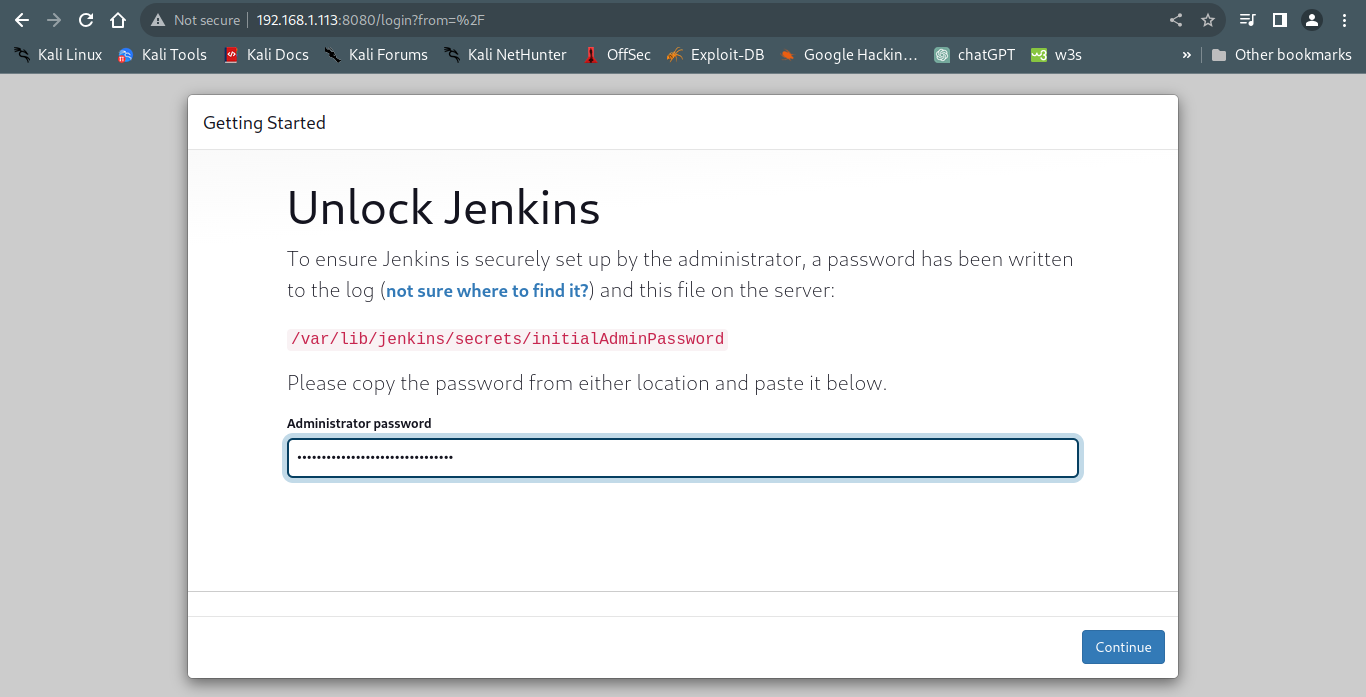


Figure 4 : Mot de passe initial de Jenkins

Une interface pour creer un utilisateur s’affiche. Il faut remplir les champs et bien se souvenir des informations, car ils seront necessaires aux prochaines authentifications.

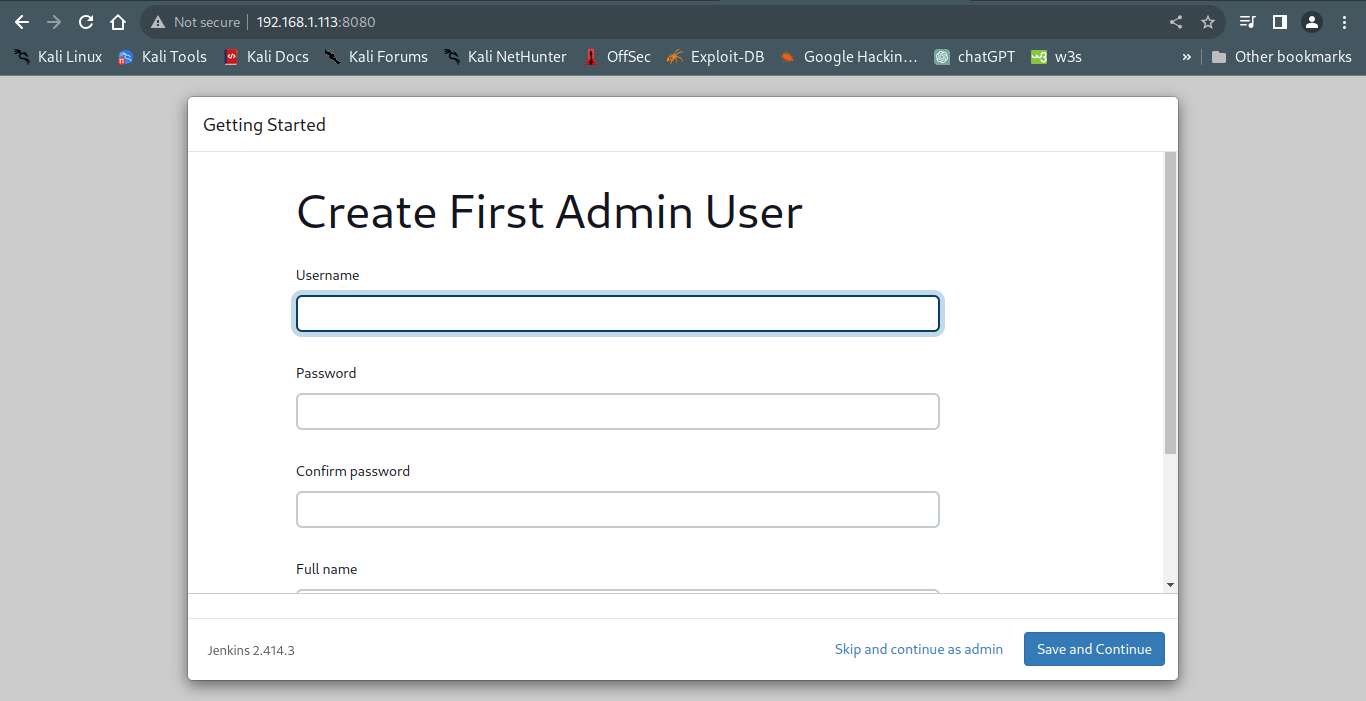


Figure 5 : Creation d'un compte administrateur

Jenkins passera ensuite a l’installation des paquets de bases

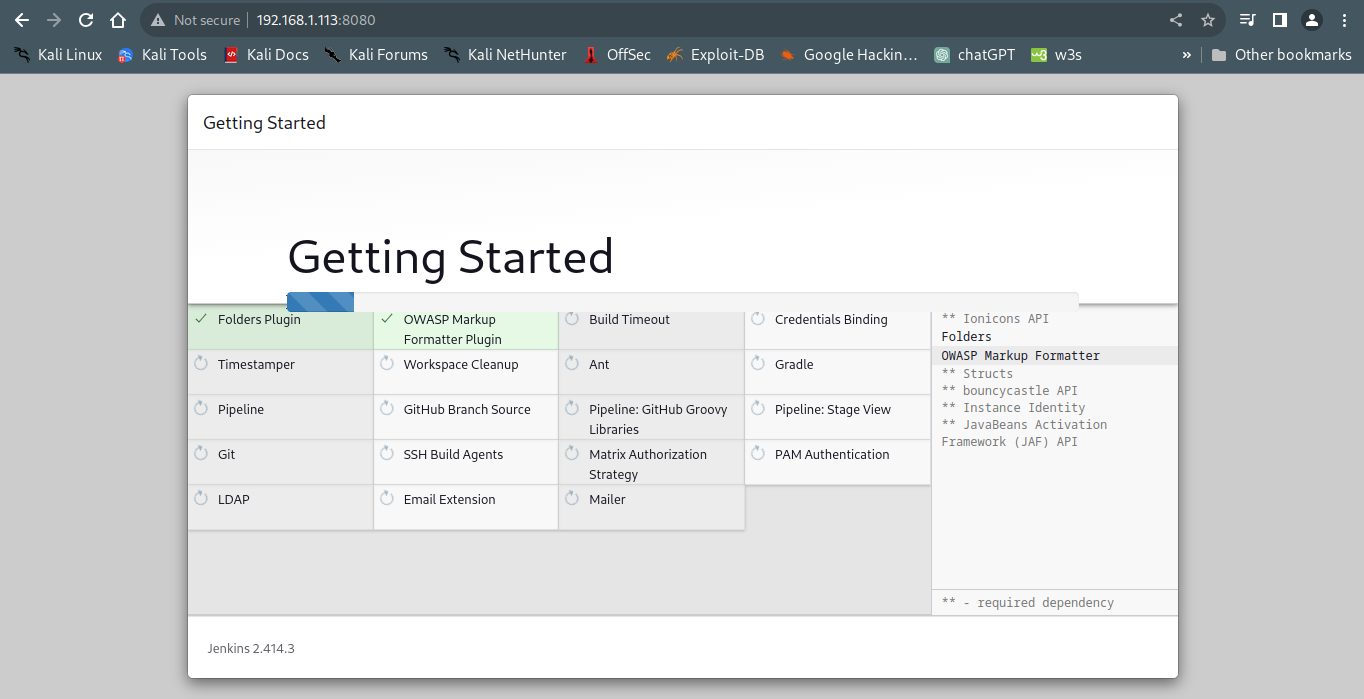


Figure 6 : Installation des paquets suggeres

Ces paquets sont les composants de bases qui permettront a Jenkins de proceder au CI/CD de base.

Une fois l’installation des paquets termines, on aura acces au tableau de bord de Jenkins fraichement installee.

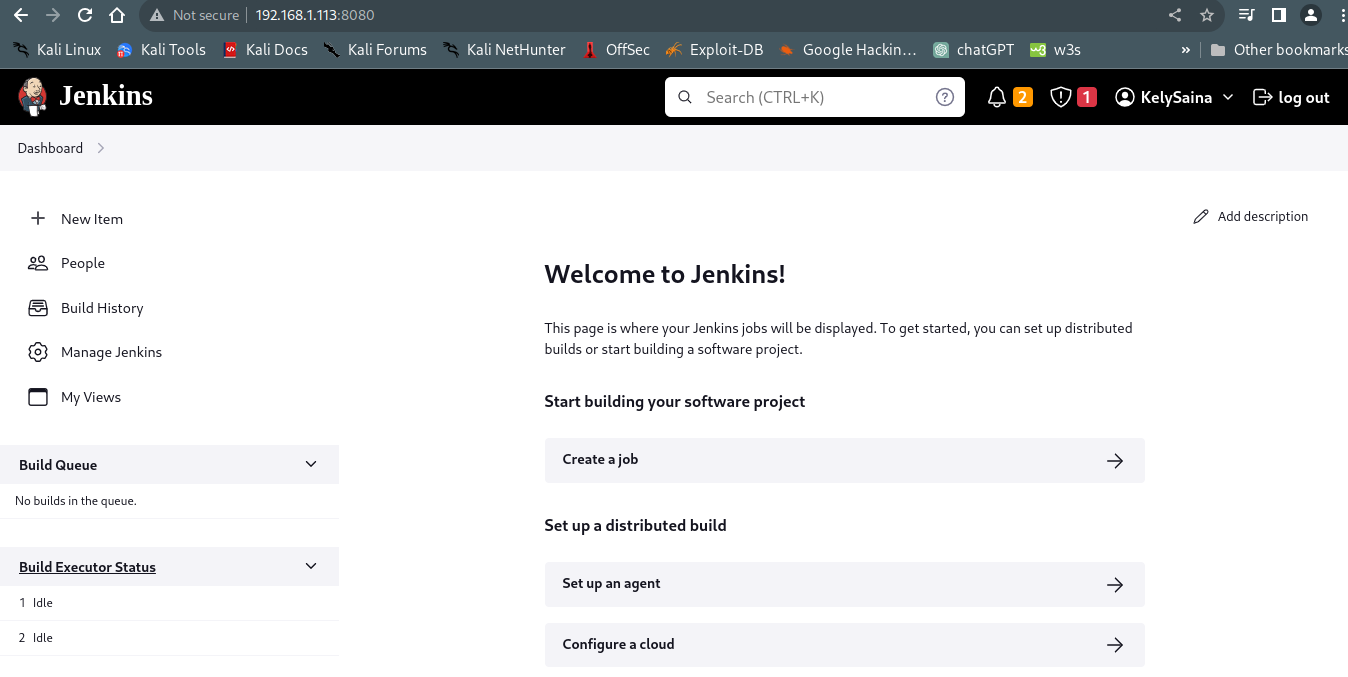


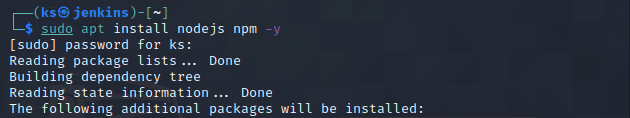
Figure 7 : Dashboard de Jenkins

1. Installation des plugins

Pour une application NodeJS, l’installation du runtime et des gestionnaires de paquets sont necessaires. Soit on les installe sur le systeme, soit en tant qu’outil ou plugin dans Jenkins.

Voici les commandes pour l’installation dans le systeme:

$ sudo apt install nodejs npm -y



Pour l’installation en tant que plugin dans Jenkins:

* Depuis le tableau de bord, naviguer: Dashboard > Manage Jenkins > Plugins > Available plugins.
* Rechercher NodeJS et cocher sur le plugin NodeJS puis cliquer sur installer.
* Une fois l’installation terminee, naviguuer: Dashboard > Manage Jenkins > Tools, scroller vers la rubrique NodeJS Installations et cliquer sur Add NodeJS
* Entrer un nom pour l’outil NodeJS et choisir la version NodeJS a utiliser. Le nom de l’outil est primordial pour l’utiliser dans le pipeline par la suite.

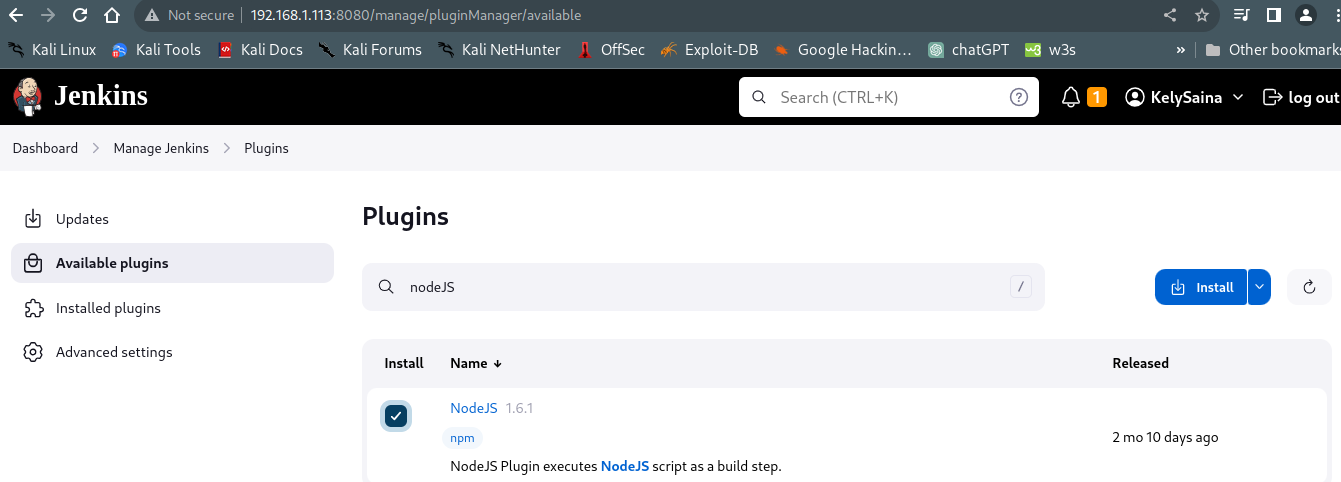


Figure 8 : Installation plugin NodeJS

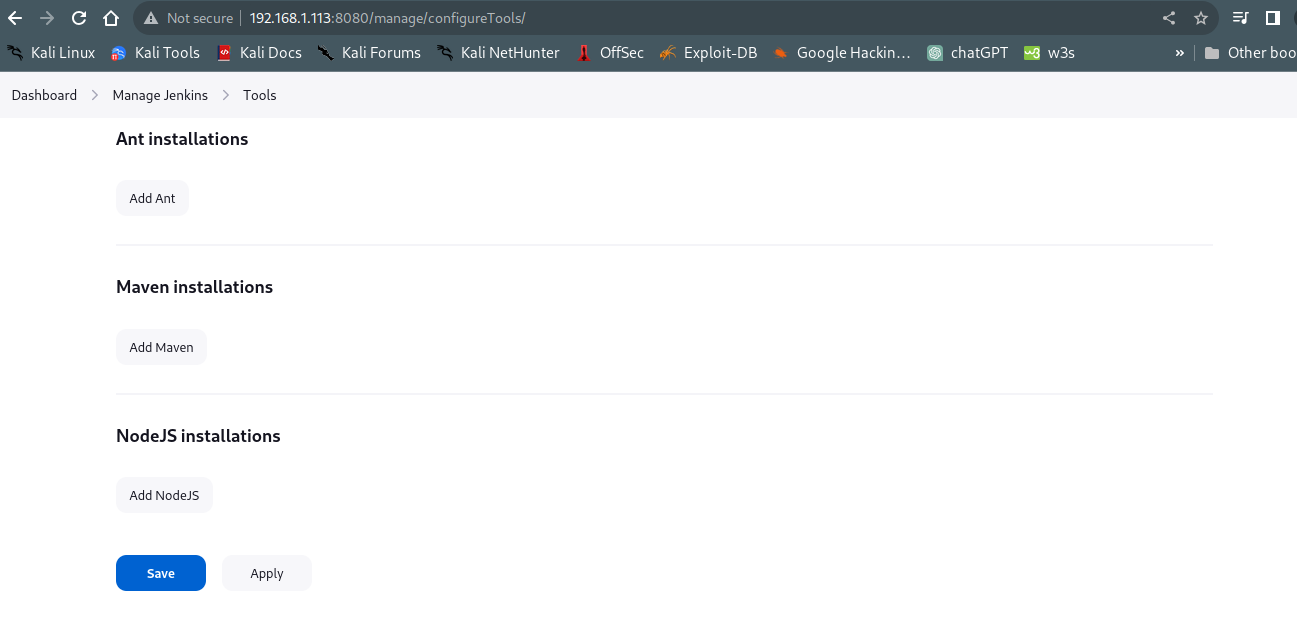


Figure 9 : Installation de l'outil NodeJS 1

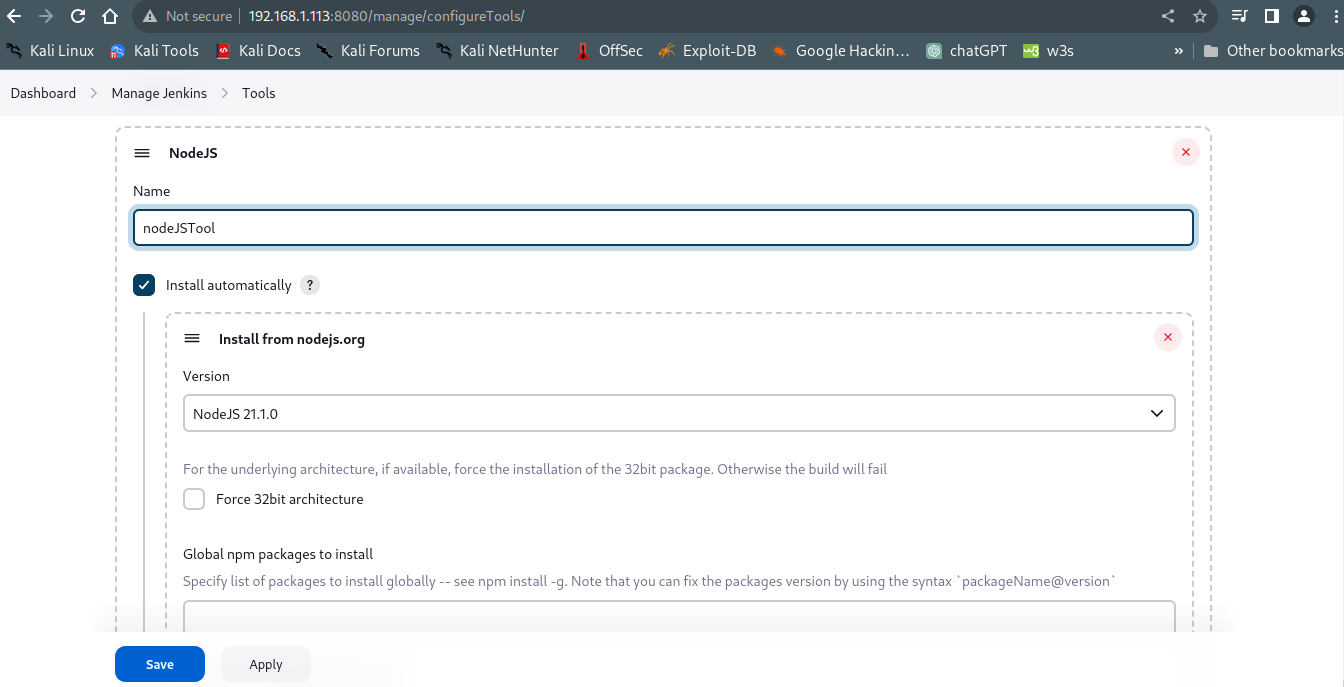


Figure 10 : Installation de l'outil NodeJS 2

1. Configuration du systeme de notification par mail

Jenkins offre aussi un systeme de notification pour alerter l’administrateur en cas de succes ou d’echec durant les etapes de tests ou de deploiement dans le pipeline.

L’envoi d’une notification par mail est un plugin de base dans Jenkins, il suffira de le configurer.

Pour l’envoi d’un mail, un compte d’envoi devra etre ajoute. Nous utiliserons un compte google en tant qu’expediteur. Il faut s’assurer que ce compte dispose d’un mot de passe d’application a cause des mesures de securite impose par google.

Pour créer un mot de passe d'application Google, accédez à votre compte Google en vous rendant sur <https://myaccount.google.com/.> Dans le menu à gauche, cliquez sur "Sécurité". Faites défiler vers le bas jusqu'à la section "Mots de passe d'application". Cliquez sur "Gérer les mots de passe d'application". Sélectionnez l'application pour laquelle vous souhaitez générer le mot de passe d'application. Si l'application spécifique n'est pas répertoriée, choisissez "Autre (personnalisé)". Choisissez l'appareil sur lequel vous utilisez l'application. Cliquez sur le menu déroulant et sélectionnez l'appareil et l'application. Cela générera un mot de passe unique pour cette application spécifique. Copiez le mot de passe généré et utilisez-le dans l'application pour vous connecter à votre compte Google. Une fois généré, assurez-vous de conserver le mot de passe en sécurité. Vous ne pourrez pas le voir à nouveau, mais vous pourrez en générer un nouveau si nécessaire.

L’etape suivante consiste a configurer le compte expediteur dans Jenkins:

* Naviguer : Dashoard > Manage Jenkins > System
* Scroller vers la rubrique Extended E-mail Notification
* Remplir les champs SMTP Server par **smtp.google.com** et SMTP Port par **465**.
* Cliquer en suite sur Advanced et cocher la case Use SSL.
* Cliquer sur Add pour ajouter l’expediteur et remplir les informations. Choisir Username with password et remplir Username par l’adresse mail de l’expediteur et Password par le mot de passe d’application et cliquer sur Add.
* Le couple mail/mot de passe devrait apparaitre dans l’option Credentials. Choisissez-le et cliquer sur Save.

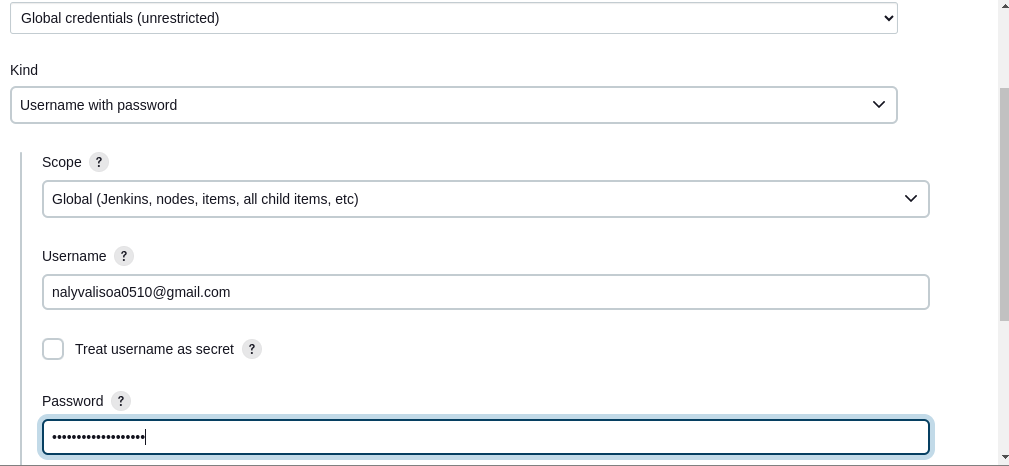


Figure 11 : Ajout de l'expediteur

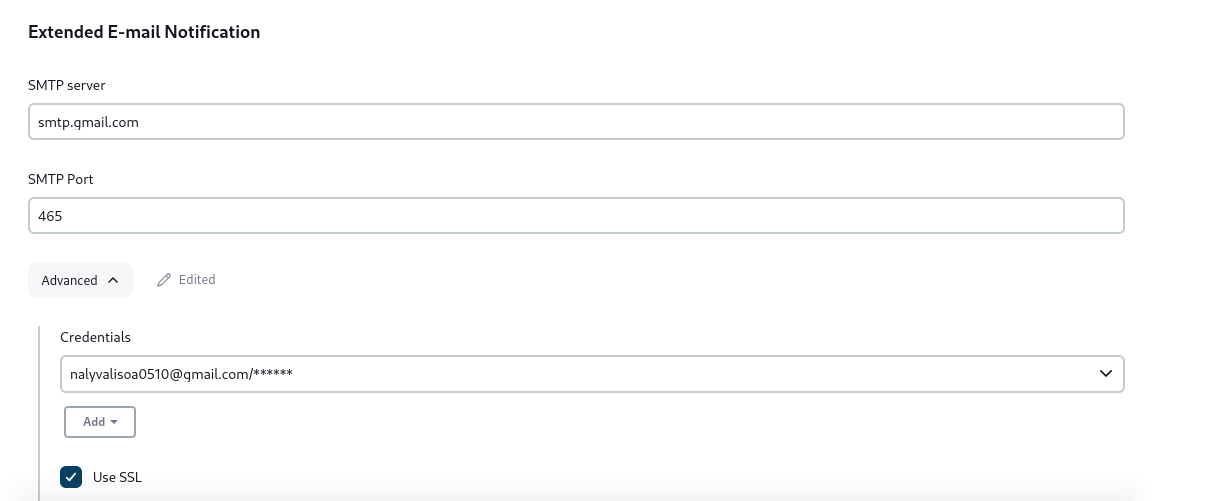


Figure 12 : Configuration de l'envoi de mail

#### 7.2. Installation d’Ansible

Ansible est un outil d'automatisation informatique open-source qui permet de configurer, orchestrer et administrer des systèmes informatiques sans nécessiter l'installation d'agents sur les machines cibles.

Utiliser la commande suivante pour l’installer:

$ sudo apt install ansible

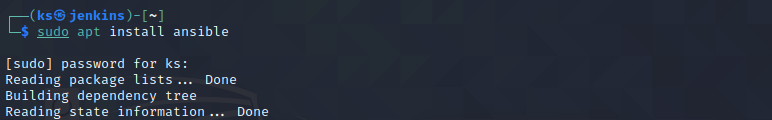


Figure 13 : Installation d'Ansible

#### 7.3. Installation de divers outils

1. Installer Jest

Jest est un framework de test JavaScript largement employé, notamment pour évaluer les API développées en Node.js. Pour l’installer globalement sur le systeme:

$ sudo npm install -g jest

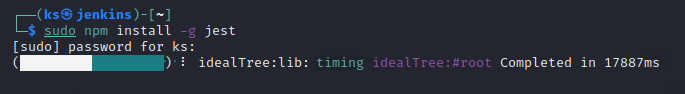


Figure 14 : Installation de Jest

1. Installer autossh

AutoSSH représente une amélioration du protocole SSH (Secure Shell) en permettant la création et la gestion automatique de tunnels SSH persistants. Ces derniers sont rétablis automatiquement en cas de déconnexion, assurant ainsi une continuité de la connexion sécurisée. Pour l’installer sur le systeme:

$ sudo apt install autossh

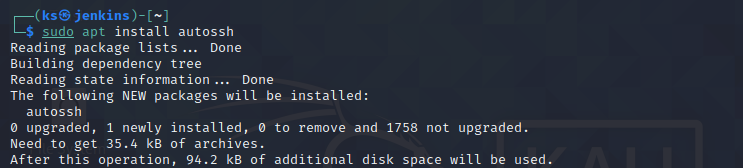


Figure 15 : Installation autossh

1. Installer sshpass

`sshpass` est une commande en ligne permettant d'automatiser le processus d'authentification SSH en transmettant le mot de passe via une instruction. Il s'avère particulièrement utile dans les scripts et les flux de travail automatisés qui requièrent une connexion SSH. En combinaison avec Ansible, `sshpass` peut être employé pour intégrer des actions automatisées d'authentification SSH dans les playbooks et les tâches de déploiement. Cela facilite la gestion et l'exécution de configurations système à distance. Installer sshpass avec la commande:

$ sudo apt install sshpass

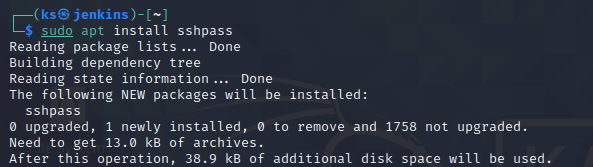


Figure 16 : Installation de sshpass

1. Creation de tunnels

Un tunnel de redirection de port est indispensable pour activer l’automatisation des taches dans Jenkins lorsqu’il est relies a un systeme de gestion de version. Nous allons utiliser **serveo.net** en tant que proxy et rediriger les requetes pour le port 8080 de jenkins vers un sous-domaine jenkins\_ks.serveo.net. La commande est:

$ autossh -M 0 -R jenkins\_ks:80:localhost8080 serveo.net

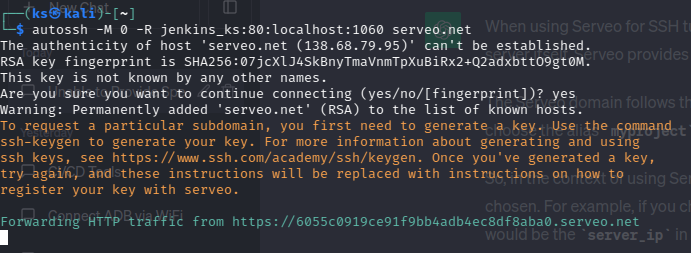


Figure 17 : Redirection via serveo.net premiere utilisation

A la premiere utilisation, le sous-domaine sera une suite aleatoire de chiffre et de lettre. Pour obtenir un sous-domaine personnalise, on doit d’abord generer une cle ssh pour l’authentification sans mot de passe:

$ ssh-keygen

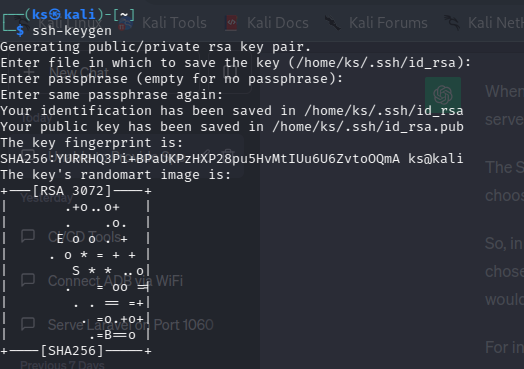


Figure 18 : Generation cle

Et re-tenter la redirection de port apres avoir genere une cle

$ autossh -M 0 -R jenkins\_ks:80:localhost8080 serveo.net

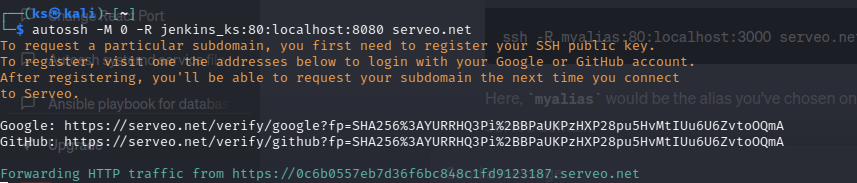


Figure 19 : Redirection via serveo.net apres generation de la cle

La commande de redirection ne demande plus d’accepter l’hote serveo.net, mais nous n’avons pas encore obtenu de sous-domaine personnalise. Pour ce faire, il faut se rendre sur un des lien propose pour s’authentifier par Google ou Github. Une fois l’authentification termine, nous devrions etre en mesure d’avoir un sous-domaine personnalise.

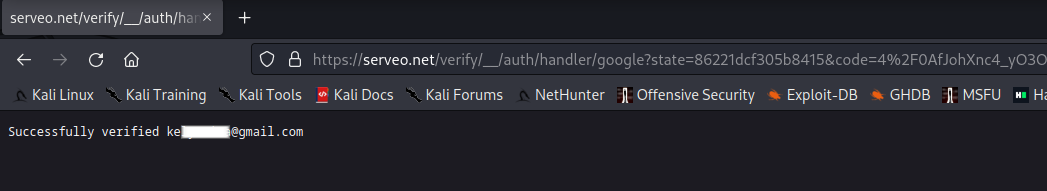


Figure 20 : Verification aupres de serveo.net

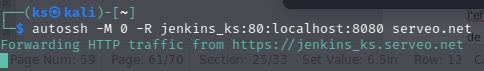


Figure 21 : Obtention sous-domaine via serveo.net

#### 7.4. Notion de DevOps

DevOps est une méthodologie axée sur la promotion de la collaboration et de la communication entre les équipes de développement (Dev) et d'exploitation (Ops), dans le but d'optimiser l'efficacité et la fiabilité du processus de création, de déploiement et de maintenance de logiciels.

Supposons que nous ayons developpes une API de CRUD en NodeJS. Cette application est encore en local dans la machine du developpeur, et au fur et a mesure qu’il ajoute les fonctionnalites du CRUD, l’on voudrait etre capable de pouvoir les utiliser. Pour cela, le cycle de vie de l’application se presente globalement dans une serie d’etapes jusqu’a terme du projet:

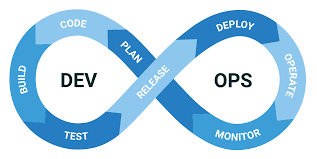


Figure 22 : Cycle DEVOPS

Les taches dans la partie DEV (Code, Build, Test) sont fait localement sur la machine. Pour les taches dans OPS (Deploy, Operate, Monitor), le tout est de s’assurer que l’application fonctionne aussi bien en developpement qu’en production.

#### 7.5. Mise en CI/CD d’une API NodeJS

1. Creation du depot

Pour commencer, il faut relier le projet a un outil de controle de version comme GitHub. Il faut alors creer un depot et affecter un webhook de jenkins pour declencher le Build automatiquement.

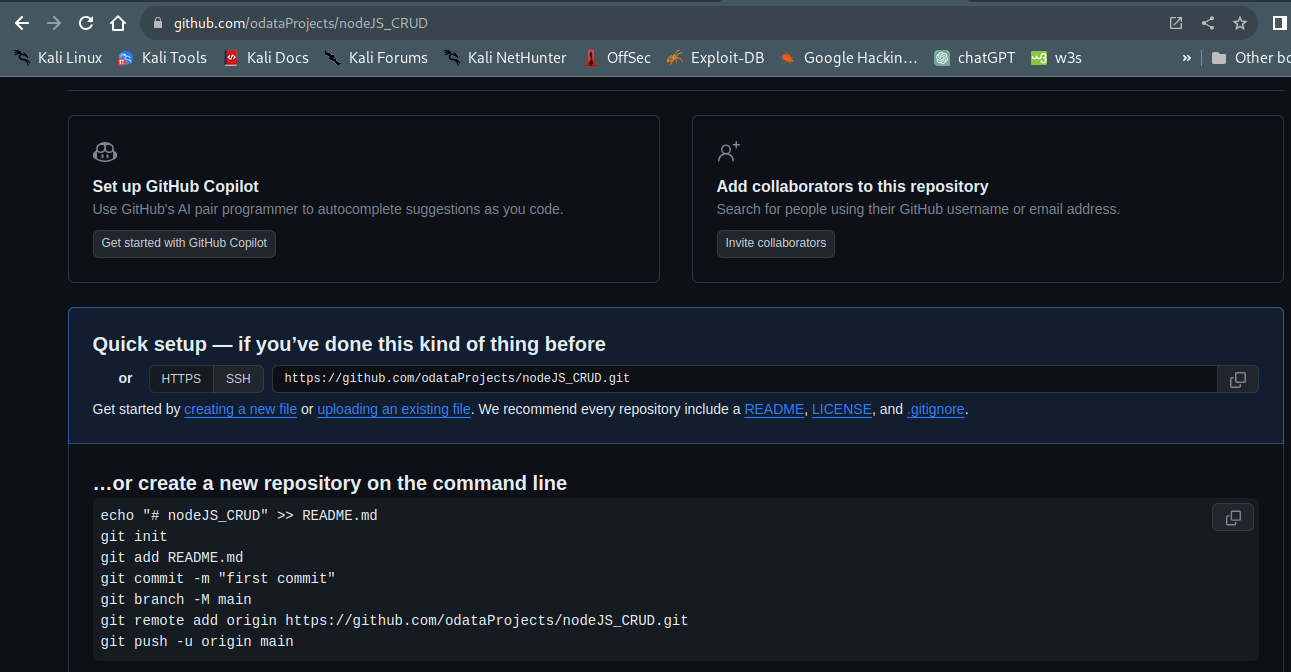


Figure 23 : Creation depot sur github

Pour la mise en place du webhook, naviguer vers l’onglet Settings et sur la partie droite, cliquer Webhook. Cliquer sur le bouton Add webhook et entrez votre mot de passe du compte GitHub .

Dans le champ Payload URL, copier l’URL suivant : **<https://jenkins_ks.serveo.net/github-webhook/>** puis valider en appuyant sur le bouton Add webhook en bas.

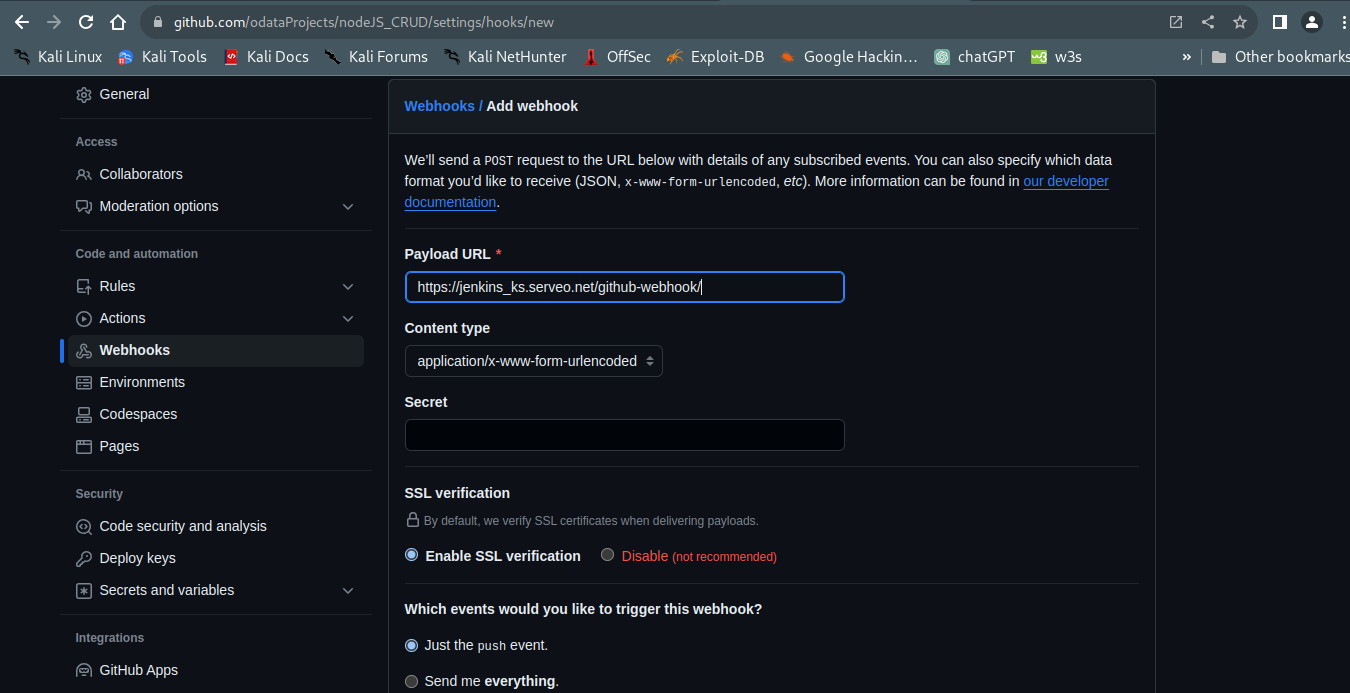


Figure 24 : Creation du webhook

1. Creation des fichiers de configuration

Dans le projet NodeJs, nous allons creer 4 fichiers :

* inventory.ini, dans lequel nous allons specifier les configuration du serveur dans lequel on hebergera l’application
* Jenkinsfile, le script d’automatisation du CI/CD de l’application
* nodeJS\_CRUD.js, un fichier Jinja2 portant le nom du depot pour creer un service qui lancera l’application
* playbook.yaml, un fichier qu’Ansible lancera pour la creation d’infrastrcture sur le serveur

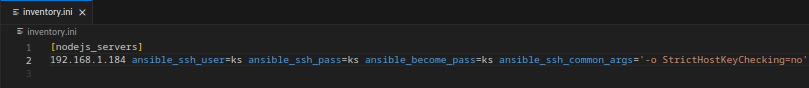


Figure 25 : inventory.ini

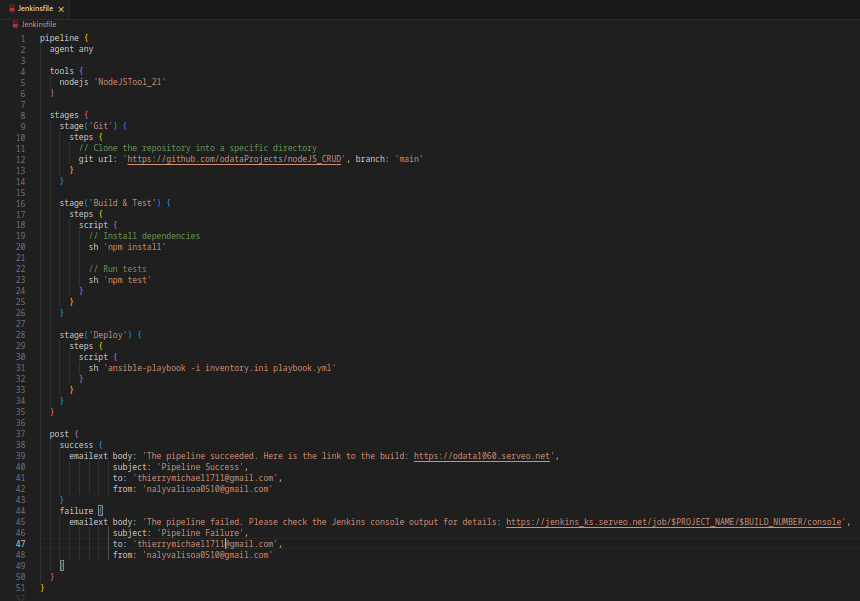


Figure 26 : Jenkinsfile

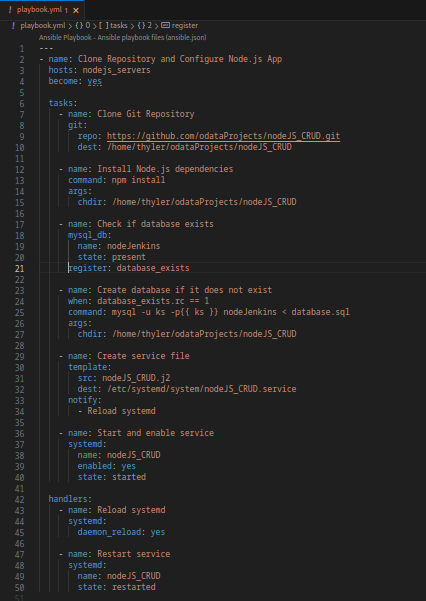


Figure 27 : playbook.yml

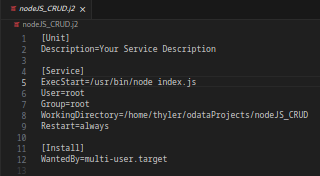


Figure 28 : nodeJS\_CRUD.j2

Une fois ces fichiers crees, on peut envoyer le projet vers GitHub.

* Sans depot local:

$ git init

$ git add .

$ git commit -m ”COMMIT\_MESSAGE”

$ git branch -M main

$ git remote add origin GITHUB\_REPOSITORY\_URL

$ git push -u origin main

* Avec depot local:

$ git add .

$ git commit -m “COMMIT\_MESSAGE”

$ git push -u origin main

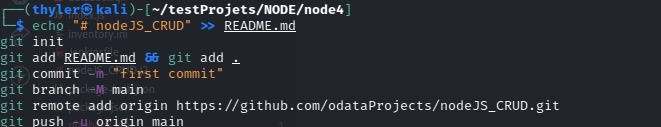


Figure 29 : Envoi du projet sur github

1. Creation du projet sur Jenkins

La creation d’un projet sur Jenkins pour l’application NodeJS consiste en 3 etapes:

* Creer un nouveau projet

Pou creer un nouveau projet, naviguer sur la page d’acceuil et cliquer sur le bouton New Item, en haut a gauche. Ensuite, saisir le nom du projet dans le champs indique, il serait preferable de mettre le nom du projet sur GitHub. Ensuite, selectionner Pipeline comme type de projet et appuyer sur Add.

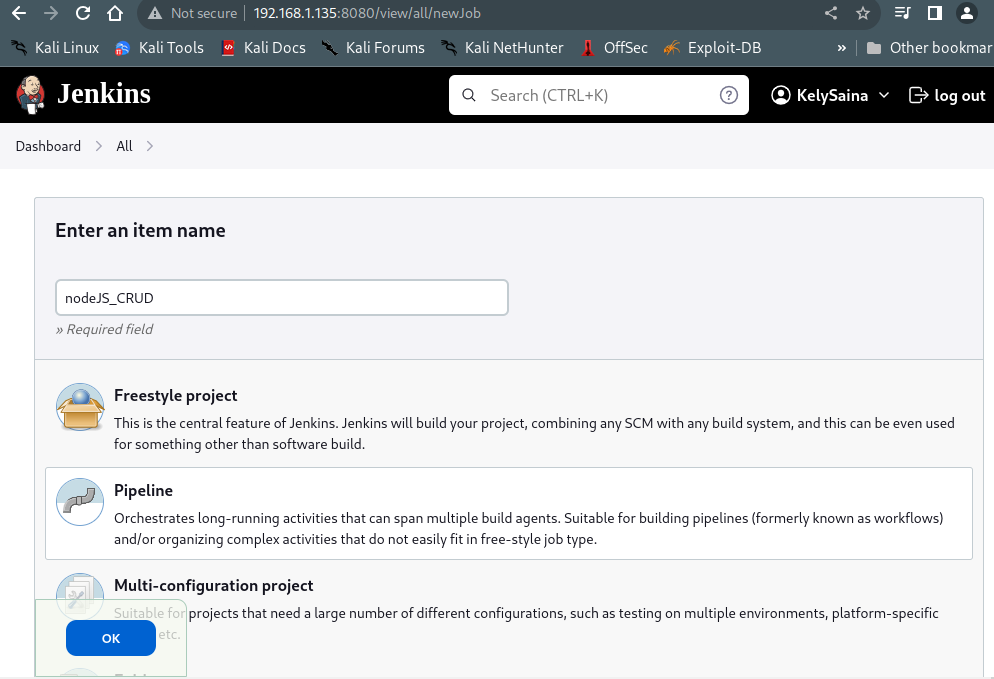


Figure 30 : Creation de projet

* Configurer les instructions

Lorsque le projet est cree, Jenkins redirige vers la page de configuration.

Premierement, cocher la case GitHub project, et copier sur le champ l’URL du projet. Ensuite, Dans la section Build triggers, cocher la case GitHub hook trigger for GITScm polling. Dans la section Pipeline, selectionner l’option Pipeline Script from SCM. Finalement, choisir le Git comme SCM et copier l’URL du projet GItHub, dans la partie branches to build, changer \*/master en \*/main. Appuyer sur sauver pour sauver les configurations.

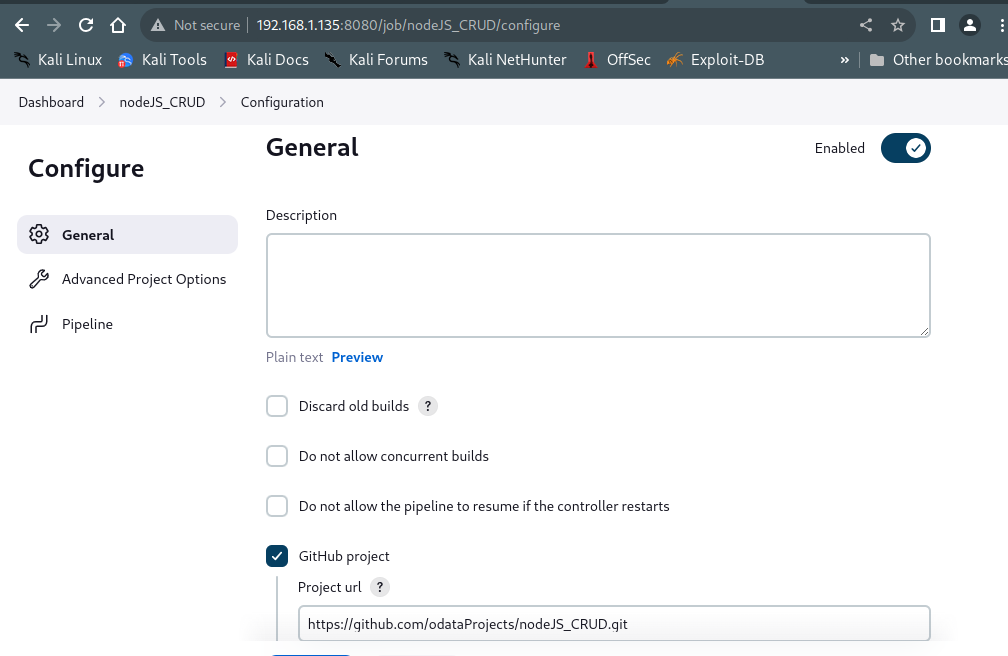


Figure 31 : Configuration 1

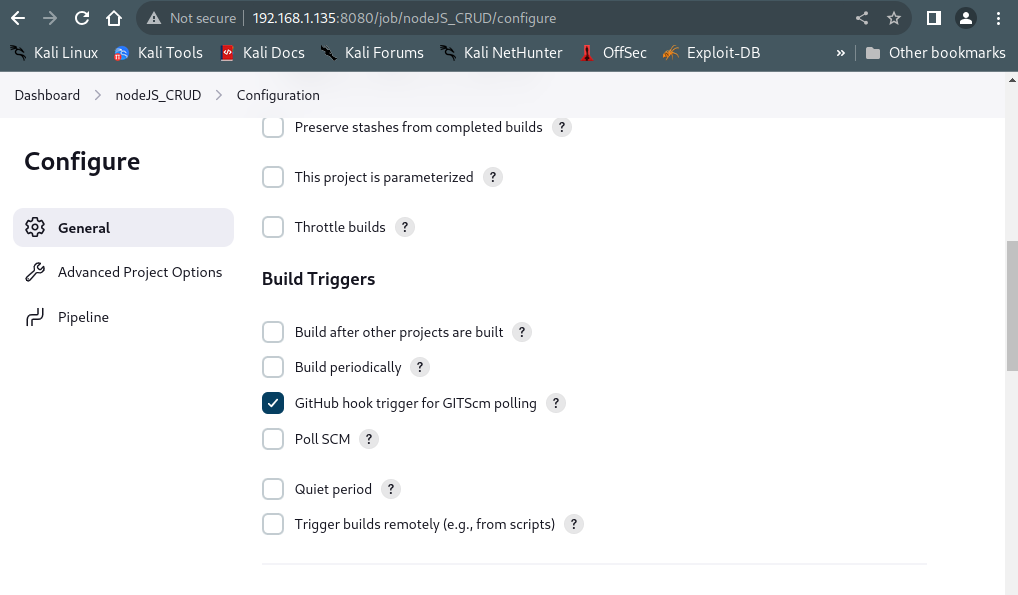


Figure 32 : Configuration 2

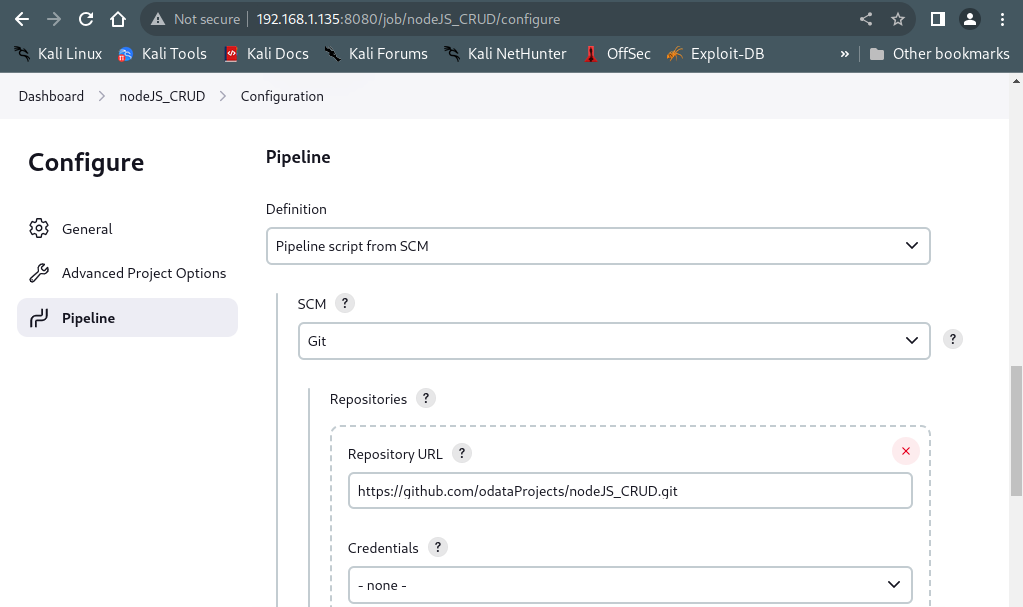


Figure 33 : Configuration 3

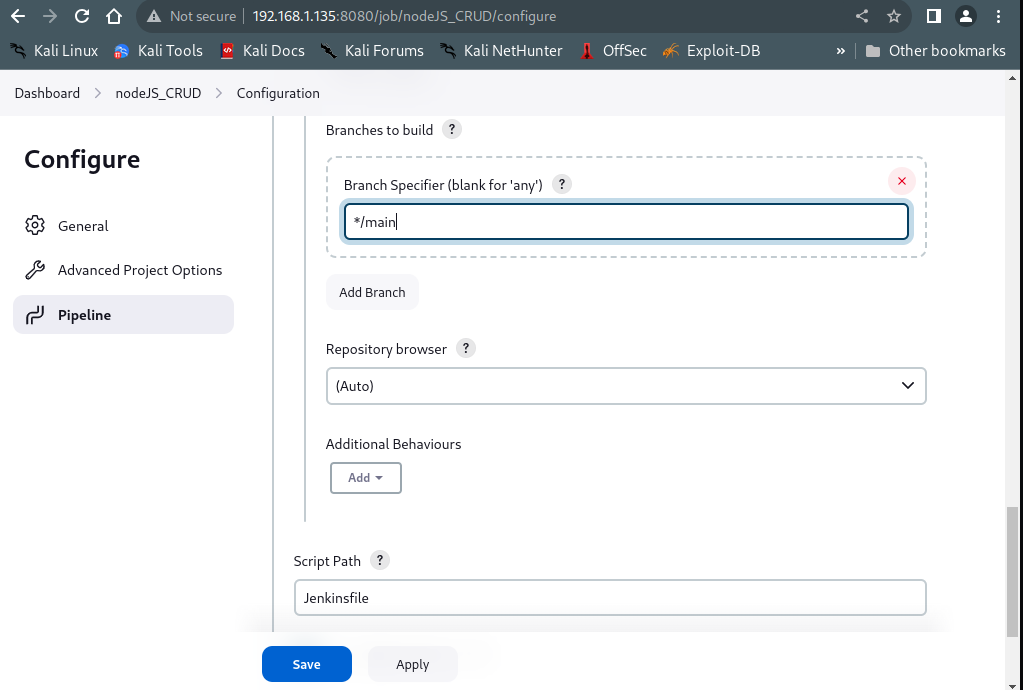


Figure 34 : Configuration 4

* Lancer le 1er Build

# Chapitre 8. Validations avec contre-analyse et perspectives

# Chapitre 9. Conclusion

# CONCLUSION GENERALE

### REFERENCES BIBLIOGRAPHIQUES

### REFERENCES WEBOGRAPHIQUES

### GLOSSAIRE

### ANNEXES

### TABLE DES MATIERES