Implémentation d'une infrastructure automatisée pour la gestion de données médicales

Par

Yugurta SARI

MÉMOIRE PRÉSENTÉ À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
COMME EXIGENCE PARTIELLE À L'OBTENTION DE DIPLÔME DE
MAITRISE AVEC MÉMOIRE EN GÉNIE DES RÉSEAUX
DE TÉLÉCOMMUNICATION
M.Sc. A.

MONTRÉAL, LE 09 JUILLET 2023

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

PRÉSENTATION DU JURRY

CE MÉMOIRE A ÉTÉ ÉVALUÉ

PAR UN JURRY COMPOSÉ DE **:**

M. Chakib Tadj, directeur de mémoire
Département de génie électrique à l'École de technologie supérieure

M. Tony Wong, président du jury
Département de génie des systèmes à l'École de technologie supérieure

M. Chamseddine Talhi, membre du jury
Département de génie des logiciel et des technologies de l'information à l'École de technologie supérieure

IL A FAIT L'OBJET D'UNE SOUTENANCE DEVANT JURY ET PUBLIC

LE 19 JUIN 2023

À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

REMERCIEMENTS

Les opportunités que j'ai eues, ces deux dernières années, ont été de belles opportunités d'apprentissage continu et de développement personnel. Par conséquent, je me considère comme une personne très chanceuse d'avoir eu l'opportunité de faire partie de L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE. Je suis également reconnaissant d'avoir eu l'occasion de rencontrer tant de personnes formidables qui m'ont guidée tout au long de cette période magistrale.

Je tiens à exprimer ma plus profonde gratitude à M. Tadj Chakib, mon directeur de recherche, pour avoir pris part à des prises de décision utiles et fourni des conseils et des orientations approfondies et fourni toutes les facilités pour réaliser ce projet de recherche.

Je veux dédier ce master à mon père qui m'a appris que rien n'est impossible si tu crois suffisamment en toi, m'a appris à ne pas être complaisant et à continuer d'apprendre à être meilleur. Aussi à ma mère bien-aimée qui prie toujours pour moi et qui croit en moi et me dit toujours que même les plus grandes tâches peuvent être accomplies si elles sont faites étape par étape. A mes frères bien-aimés, ma fiancée bien-aimée, les personnes qui me soutiennent dans ma vie et à mes chers amis.

Implémentation d'une infrastructure automatisée pour la gestion de données médicales

Yugurta SARI

RÉSUMÉ

Les épidémies sont imprévisibles, souvent meurtrières. Quel que soit l'endroit de la planète où elles surgissent, nous sommes tous potentiellement concernés. Avec la mondialisation des transports aériens, un microbe peut faire le tour du monde en moins de 24h.

Mon mémoire s'intéresse à mettre en place une infrastructure automatisée. En effet, elle porte sur une forte implication des techniques modernes de l'automatisation et de l'intelligence artificielle pour la gestion des données médicales telles les données d'une épidémie, en assurant un moyen efficace pour collecter un grand nombre d'informations en peu de temps.

Nous avons réussi à développer un Playbook Ansible responsable de l'implémentation des bases de données PostgreSQL, ainsi un Playbook dédié pour l'agrégation de données enregistrées dans plusieurs serveurs. De plus, nous avons traité l'aspect sécurité de notre infrastructure avec l'application de plusieurs techniques qui assurent l'étanchéité des serveurs de base de données contre les différentes attaques possibles.

Comme preuve de concept, nous avons décidé d'exploiter la capacité de cette infrastructure dans la gestion de données, en particulier la possibilité de collecter et de consolider des données à partir de plusieurs serveurs. Nous pourrons ainsi utiliser les données collectées pour entraîner des modèles prédictifs, en particulier des données médicales réelles comprenant des échantillons de cris de nourrissons appartenant à deux classes : des échantillons de cas sains et des échantillons de cas atteints de pathologies.

Les modèles développés sont basés sur la technique du Deep Feed Forward Neural Network. Grâce aux données collectées à l'aide de notre infrastructure, nous pourrons entraîner les modèles pour prédire les classes de nos échantillons tout en visant à obtenir les meilleurs résultats possibles.

Mots-clés : automatisation d'infrastructure, Ansible, PostgreSQL

Implementation of an automated infrastructure for medical data management

Yugurta SARI

ABSTRACT

Epidemics are unpredictable, often deadly. Wherever they appear on the planet, we are all potentially concerned. With the globalization of air transport, a microbe can travel around the world in less than 24 hours.

My thesis focuses on building an automated infrastructure. Indeed, it focuses on a strong involvement of modern automation and artificial intelligence techniques for the management of medical data such as epidemic data, ensuring an efficient way to collect a large amount of information in a short time.

We managed to develop an Ansible Playbook responsible for the implementation of PostgreSQL databases, as well as a dedicated Playbook for the aggregation of data stored on several servers. In addition, we have addressed the security aspect of our infrastructure with the application of several techniques that ensure that database servers are sealed against various possible attacks.

As proof of concept, we decided to exploit the capacity of this infrastructure in data management, in particular the ability to collect and consolidate data from several servers. To then use the data collected to train predictive models. The data used to form the models are real medical data consisting of infant cry samples, of which there are two classes of samples: samples of healthy cases, and samples of cases with pathologies.

The models developed are based on the technique (Deep Feed Forward Neural Network), with the data we will collect using our infrastructure, we will be able to train the models to predict the classes of our samples, while aiming for the best possible results.

Keywords : infrastructure automation, Ansible, PostgreSQL

TABLE DES MATIERES

LISTE DES TABLEAUX

Page

LISTE DES FIGURES

Page

LISTE DES ABRÉVIATION, SIGLES ET ACRONYMES

Psql	Postgresql
SSH	Secure Shell
DSL	Digital Subscriber Line
IP	Internet Protocol Address
IA	Intelligence Artificielle
SI	Systèmes Information
DEV	Développement
OPS	Opérations
CPU	Central Processing Unit
SGBD	Système gestion de base de données
TPL	Task Parallel Library
BSD	Berkeley Source Distribution
SQL	Structured Query Language
PL	Procedural Language
FDW	Foreign Data Wrapper
WAL	Write-Ahead Logging
IoT	Internet Of Things
DES	De Santé Électroniques
RFID	Radio Frequency Identification
OMS	Organisation Mondiale De La Santé
SIH	Systèmes d'information hospitalier (SIH)

SIS	Systèmes d'information de santé
DME	Dossiers médical électronique
RIS	Radiology Information System
PACS	Picture Archiving and Communication System

CHAPITRE 1

INTRODUCTION

1.1 Contexte et motivation

L'être humain à travers son histoire a été touché par un nombre important d'épidémies qui ont causé d'énormes impacts négatifs sur la santé, l'économie et même la sécurité nationale. Ces dernières se généralisent à la suite de la propagation de maladies entre les humains, à l'instar des plus connus, telle que la Grippe espagnole, la grippe de Hong Kong, le SRAS, la H7N9, l'Ébola, le Zika. Néanmoins il existe certaines caractéristiques clés d'une épidémie, y compris une large extension géographique, le mouvement de la maladie, la nouveauté, la gravité, les taux d'attaque élevés et l'explosivité, l'immunité minimale de la population, l'infectiosité et la contagiosité, qui nous aident à comprendre mieux leurs concepts (Qiu, Rutherford, Mao & Chu, 2017).

Une maladie épidémique est la propagation rapide d'une maladie infectieuse à un grand nombre de populations en peu de temps. Ces maladies ont coûté la vie à un grand nombre de personnes et continuent de le faire. Bien que certains remèdes et vaccins existent, les chercheurs ont encore du mal à trouver le remède pour beaucoup de type de maladies. Dans ce cas, la meilleure chose à faire est de prévenir les épidémies en prenant des précautions et en comprenant les facteurs responsables de leur propagation.

De nombreux facteurs jouent un rôle dans la propagation épidémique et ces facteurs peuvent varier en fonction du mode de transmission (Chaudhari, Argade, Jadhav, Saikar, & Ligade, 2020).

Vue la sévérité et la complexité des conséquences qui peuvent être causées par ces épidémies, des parties prenantes à travers le monde ont lutté de diverses manières pour freiner la propagation des épidémies. La science et la technologie ont contribué à la mise en œuvre de nombreuses politiques gouvernementales visant à réduire l'impact des épidémies et à diagnostiquer et fournir des traitements contre ces maladies.

Des outils technologiques récents, en particulier des outils qui se basent sur l'intelligence artificielle (IA), ont également été explorés pour suivre la propagation, identifier les patients à haut risque de mortalité et diagnostiquer les patients atteints de la maladie (Abdulkareem & Petersen, 2021).

L'IA peut aider avec de nombreuses façons, par exemple, l'IA pourrait être utilisée pour suivre la propagation du virus, identifier les patients présentant un risque de mortalité élevé, diagnostiquer et dépister une population ou réduire le temps de diagnostic (Abdulkareem & Petersen, 2021). L'IA a un énorme potentiel dans la lutte contre les épidémies, mais les applications pratiques réussies de ces outils ont jusqu'à présent été limitées en raison de différents défis. Les scientifiques doivent se contenter d'une collecte de données en temps réel, des données en proie à des pénuries dues aux tests, à la documentation et aux rapports pratiques qui varient dans le temps et entre les pays, ce qui pose un majeur problème en termes de traitement de données et aux processus de prise de décision.

Devant de telles crises sanitaires mondiales, l'industrie médicale est à la recherche de nouveaux moyens pour surveiller et contrôler les épidémies. Alors que les organisations de soins de santé s'appuient davantage sur des technologies nouvelles pour soutenir les patients et les médecins, l'automatisation des infrastructures informatiques de soins peuvent être une stratégie précieuse pour réduire le gaspillage et améliorer les performances.

L'automatisation et l'intelligence artificielle commencent à aller de pair, en particulier lorsque les organisations traitent de grandes quantités de données et essaient de rendre ces données exploitables. Alors que l'automatisation peut conduire à l'apprentissage automatique, l'IA et l'automatisation ne sont pas la même chose. L'automatisation suit des règles préprogrammées configurées manuellement par le personnel informatique. Lorsque des corrections ou des améliorations doivent être apportées à la configuration automatique, toute modification doit être effectuée par un membre du personnel. Les changements ne se produisent pas automatiquement (O'dowd, 2018).

Au lieu de cela, l'IA est censée de simuler la façon dont les humains pensent et apprennent. Alors que les fonctionnalités de l'IA peuvent être configurées manuellement si des changements importants doivent être apportés, l'IA est dynamique et peut souvent s'améliorer

en réintroduisant les résultats précédents dans des algorithmes d'auto-apprentissage pour corriger les événements passés. Les deux processus sont pilotés par les données, dont l'automatisation s'appuie sur la collecte des données et l'IA va encore plus loin en comprenant les données (O'dowd, 2018).

L'industrie de la santé devrait bénéficier de manière significative de l'automatisation du point de vue des soins, ainsi que d'un point de vue commercial. L'automatisation a fait progresser la science et a amélioré la qualité des soins de santé. Il est bien compris que l'automatisation est un élément essentiel pour fournir de meilleurs soins de santé (D Arezzo, 2019).

Les progrès technologiques continus dans le secteur de la santé encouragent les grands hôpitaux et les organisations. Les organisations de soins de santé doivent étendre leurs infrastructures au-delà des murs de leurs établissements. L'ajout des outils plus avancés pour relever ces défis peut être compliqué, mais l'automatisation des processuspeut aider à alléger la charge du personnel informatique, on éliminant les erreurs humaines quand il s'agit des tâches gourmandes en données et elle aide également les organisations à maintenir des solutions de sauvegarde et de récupération fiables et correctes.

1.2 Problématique

Trois principaux défis doivent être pris en compte lors du déploiement d'une infrastructure automatisée utilisée pour la gestion de données medicales :

- P1 :Création des bases de données ou conteneurs de données en peu de temps.
- P2 : Manque d'accès à des ensembles de données suffisamment volumineux pour l'entraînement et la validation externe des modèles d'IA (accessibilité des données).
- P3 : la compatibilité et la cohérence des données récupérées à partir de diffirentes sources.

1.3 Objectifs

Dans ce mémoire, notre objectif est de concevoir un système qui automatise le déploiement optimal d'infrastructures dédiées à la gestion de données médicales. En prenant en compte les besoins croissants en termes de données et les différents défis existants, nous avons défini les principaux objectifs que nous souhaitons atteindre au cours de ce travail :

1. Améliorer l'efficacité opérationnelle en automatisant les processus manuels et en réduisant la dépendance aux tâches manuelles, ce qui permettra de gagner du temps et des ressources.
2. Renforcer la sécurité des données en intégrant des mesures de sécurité robustes dans l'infrastructure automatisée, telles que le chiffrement des données, l'authentification des utilisateurs, la gestion des accès, afin de protéger les informations sensibles.
3. Améliorer l'accessibilité et le partage des données pour permettre un accès facile et sécurisé aux données médicales pertinentes, à la fois au sein de l'organisation médicale et entre les différents acteurs du secteur de la santé.
4. Assurer la cohérence et la qualité de données afin d'entraîner des modèles predictifs et d'obtenir de meilleurs précisions.

La figure suivante décrit les étapes suivies durant la réalisation de ce travail.

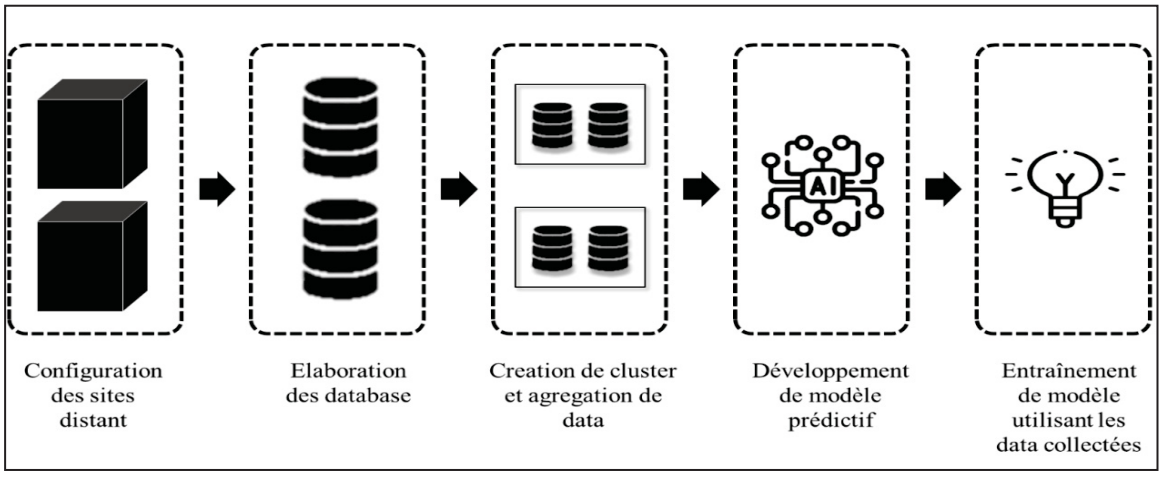

Figure 1.1 Les étapes de mis en place de l'infrastructure

1.4 Plan

Le présent travail est divisée en six chapitres organisés comme suit :

- Le premier chapitre est une introduction générale. Nous présentons d'abord le contexte général et les motivations de cette recherche. Ensuite, les problèmes et les objectifs à atteindre**.**
- Le deuxième chapitre aborde le contexte technique. Il est divisé en deux parties. La première partie présente le concept d'automatisation avec la structure de Ansible et la comparaison entre Ansible et d'autres outils d'automatisation. La deuxième partie présente le système de gestion de base de données ainsi qu'une explication de PostgreSQL**.**
- Le troisième chapitre est une revue de la littérature dont nous allons Parler plus précisément sur les épidémies ainsi sur l'utilisation de l'intelligence artificielle dans le domaine de la santé en traitant le concept de (deep feed forward network).
- Le quatrième chapitre est consacré à la mise en œuvre de l'application conformément à l'objectif de notre mémoire qui sera présenté en deux parties. La première partie est consacrée à la préparation de l'environnement de travail par la création de machines hôtes, le developpement de scripts d'automatisation et la configuration des bases de données Postgresql. La seconde partie montre l'aspect sécurité de l'environnement avec des explications concernant la gestion des accès et la haute disponibilité des bases de données.
- Dans lechapitre cinq nous allons traiter les données que nous allons collecter au chapitre quatre, ainsi que l'entrainnement des modèles prédictifs en utilisant ces données et en fin une discussion sur les résultats obtenus.
- Le dernier chapitre présentera une conclusion génerale de mémoire et les perspectives de ce travail.

CHAPITRE 2

TECHNIQUES ET BACKGROUND

Ce chapitre présente les techniques et le contexte de ce mémoire, notamment les concepts d'automatisation et de système de gestion de base de données.

2.1. DevOps et automatisation

Le développement des systèmes d'information (SI) évolue en permanence, passant d'une approche basée sur la documentation à des méthodes de travail plus agiles et collaboratives. Initialement, les méthodes de travail agiles étaient principalement axées sur le développement (Dev) du SI et peu d'attention a été accordée aux opérations (Ops). Plus récemment, les approches agiles ont introduit le concept intégré de DevOps.

Il a été rapporté que, DevOps est un ensemble de pratiques qui tentent de combler le fossé entre les développeurs et les opérations au cœur des choses et en même temps couvre tous les aspects qui contribuent à un logiciel rapide, optimisé et de haute qualité (Ghantous & Gill, 2017).

Selon Krohn (2022), une équipe DevOps comprend des développeurs et des opérations informatiques travaillant en collaboration tout au long du cycle de vie du produit, afin d'augmenter la vitesse et la qualité du déploiement des logiciels. C'est une nouvelle façon de travailler, un changement culturel, qui a des implications importantes pour les équipes et les organisations pour lesquelles elles travaillent. Dans un modèle DevOps, les équipes de développement et d'exploitation ne sont plus cloisonnées. Parfois, ces deux équipes fusionnent en une seule équipe où les ingénieurs travaillent sur l'ensemble du cycle de vie de l'application, à partir du développement et tests au déploiement et opérations. Les équipes DevOps utilisent des outils pour automatiser et accélérer les processus, ce qui contribue à accroître la fiabilité, l'intégration continue, la livraison continue, l'automatisation et la collaboration (Atlassian, s. d.)

La figure 2.1 représente la boucle infinie pour indiquer comment les phases du cycle de vie DevOps sont liées les unes aux autres.

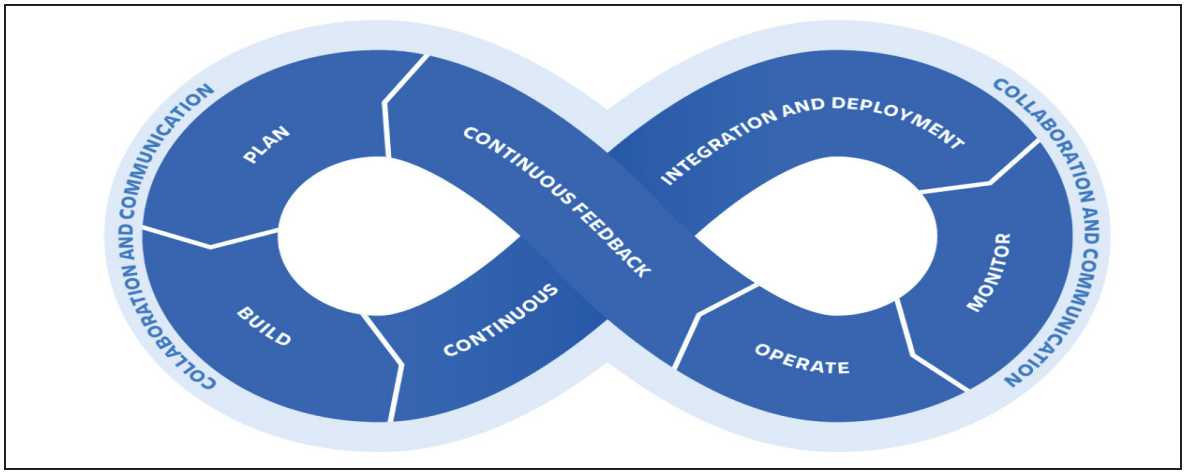

Figure 2.1 Cycle de vie DevOps
tirée de Atlassian, s. d

La séparation traditionnelle entre les développeurs et les opérations que l'on trouve aujourd'hui dans de nombreuses organisations est un obstacle majeur à la rapidité des versions fréquentes de logiciels. Cela est dû à des objectifs différents, des mentalités contraires et des processus incompatibles de ces deux groupes. Par exemple, les développeurs veulent pousser les changements en production le plus rapidement possible, alors que le personnel des opérations vise à maintenir les environnements de production stables. Pour cette raison, la collaboration et la communication entre les développeurs et le personnel d'exploitation reposent principalement sur des processus lents, manuels et sujets aux erreurs. Par conséquent, il faut beaucoup de temps pour mettre en production les modifications, les nouvelles fonctionnalités et les corrections de bogues. Cependant, les besoins des utilisateurs s'accroît en termes de rapidité. Ainsi, c'est un avantage concurrentiel de mettre en œuvre des processus automatisés pour permettre des versions d'applications rapides et fréquentes. Mais cela n'est possible qu'en comblant le fossé entre le développement et les opérations (Bass & al., 2015).

DevOps est un paradigme émergent qui devrait combler ce fossé entre ces deux groupes, permettant ainsi une collaboration efficace. Outre les défis organisationnels et culturels pour

éliminer la scission, le processus de déploiement doit être hautement automatisé pour permettre la livraison continue des logiciels. La communauté DevOps en croissance constante soutient cela en fournissant une grande variété d'approches individuelles telles que des outils et des artefacts pour mettre en œuvre une automatisation globale du déploiement. Les artefacts DevOps réutilisables tels que les scripts, les modules et les modèles sont accessibles au public pour être utilisés pour l'automatisation du déploiement (Wettinger & Breitenbucher, 2014).

Aujourd'hui et à l'avenir, les gens auront besoin de plus d'automatisation. Le secteur de la santé se rend également compte que l'automatisation des services leur apportera des avantages. L'automatisation dans le domaine de la santé présente de nombreux avantages (Bigelow, 2019), elle peut améliorer la prestation de services, d'où la fourniture de services sera de meilleure qualité et en temps opportun.

L'automatisation a un fort potentiel dans le domaine de santé, mais n'a pas été pleinement réalisée. L'automatisation peut aider les établissements de santé à mieux gérer les charges de travail, à améliorer l'efficacité et à numériser les tâches manuelles chronophages sur papier. Cependant, le potentiel de l'automatisation pour aider à protéger et à prendre soin des professionnels de la santé est souvent négligé. Il existe encore de nombreux services de santé qui le font encore manuellement ou sur papier, ce qui prend beaucoup de temps (Kim & al., 2016).

Voici une explication de ce qu'est l'automatisation.

L'automatisation informatique est le processus de création de logiciels et de systèmes pour remplacer les processus reproductibles et réduire les interventions manuelles. Il accélère la livraison de l'infrastructure et des applications informatiques en automatisant les processus manuels qui nécessitaient auparavant une intervention humaine (roth,2022).

Avec l'automatisation, les logiciels sont utilisés pour configurer et répéter des instructions, des processus ou des politiques qui permettent de gagner du temps et de libérer le personnel informatique pour un travail plus stratégique. Avec l'essor des réseaux virtualisés et des services cloud qui nécessitent un provisionnement rapide et complexe, l'automatisation est

une stratégie indispensable pour aider les équipes informatiques à fournir des services avec une vitesse, une cohérence et une sécurité améliorées (Karamitsos et al., 2020).

Dans ce travail l'automatisation va nous servir à créer des bases de données pour accueillir les informations ainsi que pour effectuer toutes les tâches reliées à l'ouverture des comptes utilisateurs et super utilisateurs d'une façon centralisée et automatique. Pour cela, nous allons utiliser un des outils d'automatisation les plus puissants, ce choix est dû à plusieurs raisons dont nous allons parler durant la suite de ce chapitre.

2.2. Comparaison entre Ansible et les autres outils d'automatisation

La plupart des gens comparent aujourd'hui Ansible à des outils tels que Puppet et Chef, qui sont les plus couramment utilisés. Puppet et Chef sont plus similaires l'un à l'autre qu'à Ansible, mais ils peuvent remplir les mêmes fonctions. Alors que Puppet et Chef utilisent chacun un serveur central pour stocker l'état souhaité des machines et toutes les métadonnées qui leur sont associées, Ansible n'a aucun serveur central (ce qui le rend sans agent). Ceci est important, car lors de l'utilisation d'outils tels que Puppet et Chef, chaque serveur se connecte périodiquement au serveur central pour voir s'il y a des mises à jour. Ansible repose entièrement sur l'utilisateur final qui applique lui-même les modifications.

La figure 2.2 illustres les outils de gestion de configuration les plus populaires.

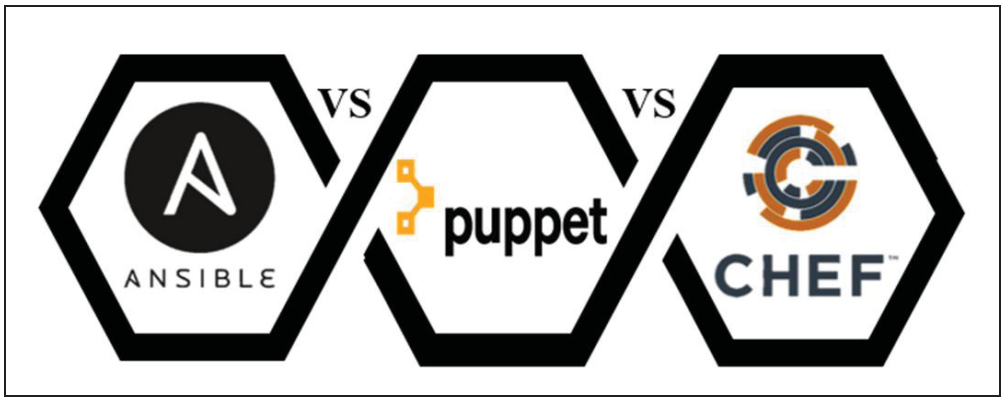

Figure 2.2 Outils de gestion de configuration
tirée de educba.com

L'intérêt de comparer Ansible, Chef et Puppet est évidente dans la nécessité d'identifier le meilleur outil de gestion de configuration pour un projet donné. Ces trois outils sont simples en termes de facilité d'utilisation. Ils offrent également de puissantes capacités d'automatisation d'environnements informatiques complexes à plusieurs niveaux. La discussion suivante vise à étudier les différences entre Ansible, Chef et Puppet en fonction de divers facteurs, cette discussion permettra de comparer les forces et les faiblesses des trois outils.

Les différences entre Ansible, Puppet et Chef sont présenté dans le tableau 2.1.

Tableau 2. 1 Comparaison des caractéristiques des outils d'automatisation

Ansible	Puppet	Chef
Un programmeur n'est pas nécessaire pour gérer cet outil.	Un programmeur connaissant Puppet DSL est nécessaire pour gérer Puppet.	Un programmeur Ruby est nécessaire pour gérer l'outil Chef.
L'exécution de la configuration est un processus simple.	L'exécution de la configuration n'est pas un processus facile par rapport à Ansible.	L'exécution de la configuration n'est pas un processus facile.
Nous pouvons appeler Ansible comme immature.	La marionnette est vieille et mature.	Chef est aussi mature qu'Ansible.
Ansible n'a pas beaucoup de fonctionnalités.	Les fonctionnalités de Puppet sont plus.	Chef a de nombreuses fonctionnalités comme Ansible.
Ansible signale les erreurs survenues lors de l'installation.	Les erreurs ne sont pas signalées dans Puppet lors de l'installation, ce qui rend le processus difficile.	Le processus d'installation est difficile dans Chef car les erreurs ne sont pas affichées.
Le système de communication est plus rapide.	Le système de communication est plus lent.	Le processus de communication est très lent.

Tableau 2. 1 Comparaison des caractéristiques des outils d'automatisation

Ansible	Puppet	Chef
 L'environnement mis à l'échelle est ralenti lors de la communication avec ssh.	 Lors du déploiement à grande échelle, en raison d'un code DSL volumineux, la mise à l'échelle devient difficile.	 Chef est également confronté à des problèmes lors de la mise à l'échelle de l'environnement en raison de son code volumineux.
 Si les nœuds primaires tombent en panne, le nœud secondaire reprend la tâche.	 Lorsque le maître échoue, un autre maître prend la place. Fondamentalement, il s'agit d'un système multi-maître.	 Il existe un serveur de secours si le serveur principal tombe en panne dans Chef.
 Ansible est orienté administrateur.	 Puppet est basé sur son administrateur système.	 Chef est principalement orienté développeur.
 Ansible a un style de codage procédural.	 Puppet a un style déclaratif.	 Chef suit le codage de style procédural.
 Ansible n'a pas d'architecture maître et donc pas de serveurs supplémentaires.	 En raison de son architecture maître, un serveur supplémentaire doit être exécuté.	 Chef nécessite également un serveur supplémentaire pour exécuter le serveur maître.
 L'entretien n'est pas une tâche fastidieuse.	 La maintenance de tous les serveurs supplémentaires	 Un entretien élevé est nécessaire.

	doit être effectuée.	

Tableau 2. 1 Comparaison des caractéristiques des outils d'automatisation

Ansible	**Puppet**	**Chef**
 Ansible est plus populaire.	 La marionnette n'est pas très populaire.	 Chef n'est pas aussi populaire qu'Ansible.
 Les applications sont facilement déployées à l'aide d'Ansible.	 Le déploiement d'applications n'est pas si facile.	 Chef n'effectue pas de déploiement d'application.
 Plusieurs serveurs s'interrogent.	 Interroger entre les serveurs n'est pas facile.	 L'interrogation n'est pas effectuée dans Chef.

2.3. La structure de Ansible

Ansible est un outil d'automatisation disponible en source ouverte. Par le passé, on s'assurait que l'état d'un système correspondait à ce à quoi on s'attendait, puis on créait un document certifiant que c'est le cas. Aujourd'hui, nous écrivons ce document comme une spécification déclarant l'état attendu du système, puis nous nous appuyons sur des outils tels que Ansible pour mettre en œuvre les transformations requises. Bien sûr, nous avions l'habitude d'écrire de petits scripts shell pour créer un utilisateur ou installer et configurer un logiciel spécifique, mais cela ne suffit plus. L'infrastructure en tant que code exige le même respect que nous accordons à nos logiciels essentiels à la mission : examens de code, suites de tests et processus de développement de logiciels éprouvés. Avec Ansible, nous pouvons effectuer le provisionnement des serveurs, la gestion de la configuration et le déploiement d'applications sur les serveurs.

Ansible utilise les fichiers YAML (yet another markup language) comme principale source d'information au moment de son exécution. YAML est un langage de représentation de données couramment utilisé pour la configuration, Ansible est entièrement écrit en Python. Le coureur principal et tous les modules sont compatibles avec Python 2.6, ce qui signifie qu'ils fonctionneront avec n'importe quelle version de Python2 au-dessus de la version 2.6, donc il n'y aurait pas de dépendances supplémentaires sur les machines que nous devons gérer (Heap, 2016).

Non seulement il n'y a pas de dépendances linguistiques supplémentaires pour nos machines, mais aussi, il n'y a pas de dépendances supplémentaires du tout. Ansible fonctionne en exécutant des commandes via SSH. Il n'est pas nécessaire d'installer un logiciel serveur. C'est un énorme plus pour deux raisons :

1. Les machines exécutent les applications sans CPU ou des démons avides de mémoire en arrière-plan.
2. Nous pouvons exploiter tout ce que SSH fournit gratuitement, en utilisant les fonctionnalités avancées, telles que ControlPersist et Kerberos. En outre, il n'est pas nécessaire de configurer notre propre mécanisme d'authentification, nous pouvons laisser SSH prendre soin de cette étape.

2.3.1. Création de Playbook

Un playbook Ansible est un fichier YAML utilisé par Ansible, un outil d'automatisation et de gestion de configuration. Il permet d'automatiser des tâches telles que le déploiement, la configuration et la gestion de systèmes informatiques.

La structure d'un playbook Ansible repose sur une séquence de tâches, où chaque tâche correspond à une action spécifique à effectuer. Ces tâches sont exécutées séquentiellement sur les machines cibles, conformément à l'ordre défini dans le playbook. Chaque tâche utilise des modules Ansible pour réaliser des actions particulières, par exemple le module "copy" pour copier des fichiers, le module "apt" pour gérer les paquets sur les systèmes basés sur Debian, ou encore le module "service" pour gérer les services système.

Le playbook Ansible offre également la possibilité de définir des variables, des conditions, des boucles et d'autres structures de contrôle afin de personnaliser le comportement de l'automatisation. Il peut également inclure des rôles Ansible, qui sont des ensembles de tâches réutilisables organisées de manière modulaire.

En somme, les playbooks Ansible sont des fichiers YAML qui utilisent une terminologie spécifique pour indiquer à Ansible les actions à effectuer. Ils contiennent une liste de commandes, d'arguments et de configurations nécessaires. Les tâches peuvent être exécutées de manière synchrone ou asynchrone en fonction des besoins. Cette syntaxe concise et lisible par les humains et les machines facilite la création de playbooks Ansible, qui se présentent essentiellement comme des manuels YAML avec un format spécialisé.

Une fois le fichier playbook est prêt, la première chose que nous devons faire est de dire à Ansible où ce manuel devrait fonctionner en spécifiant un groupe hôte sur lequel il sera exécuté.

Pour exécuter ce playbook sur tous les hôtes disponibles, il suffit d'ajouter - hosts : all à notre fichier sur une nouvelle ligne.

Maintenant que Ansible sait où s'exécuter, nous pouvons lui dire ce que nous voulons qu'il exécute. Ceci est fait en ajoutant une section nommée tasks, dans cette section, nous allons dire à Ansible d'exécuter un ping pour s'assurer que nous pouvons se connecter sur nos machines.

Nous pouvons ensuite exécuter Ansible sur la machine en exécutant le Playbook (figure 2.3). Nous devrions voir des résultats qui ressemblent à ce qui suit :

```
GATHERING FACTS
*********************************************************************
ok: [default]

TASK: [ping ]
*********************************************************************
ok: [default]

PLAY RECAP
*********************************************************************
default                    : ok=2    changed=0    unreachable=0    failed=0
```

Figure 2.3 Résultat d'exécution de Playbook

Le bloc TASK : [ping] nous fait savoir que notre action a été exécutée avec succès. Cela montre qu'Ansible peut se connecter à notre hôte. Bien que l'exécution de ce guide soit plutôt modeste, il est facile de suivre ce qui se passe. Au fur et à mesure que la complexité augmente, nous pouvons imaginer que les choses deviendront de plus en plus compliquées, et donc plus difficiles à suivre. Heureusement, Ansible nous permet d'ajouter un nom à chaque tâche pour expliquer son but.

2.3.2. L'inventaire Ansible

Dans la gestion de configuration, l'outil que nous utilisons doit savoir sur quelles machines il doit fonctionner. C'est ce qu'on appelle un inventaire. Sans inventaire, nous aurions un ensemble de Playbooks qui définissent l'état de notre système souhaité, mais nous ne saurions pas contre quelles machines nous devons les exécuter.

Avec les autres outils d'automatisation tels que Puppet et Chef, cette information est stockée sur un serveur central. Comme il n'y a pas de serveur central dans Ansible, nous aurons besoin d'une autre façon d'obtenir toutes ces informations, c'est là que le fichier d'inventaire entre en jeu.

Par défaut, Ansible va considérer le fichier /etc/ansible/hosts comme son inventaire.
Cependant, nous devrions maintenir un fichier d'inventaire différent pour chaque projet que nous avons et le passer à la fois Ansible et Ansible-playbooken en utilisant l'option –i.

Ceci est un exemple de passage d'un fichier d'inventaire personnalisé (figure 2.4) à Ansible avant d'exécuter le module ping :

ansible all –i /path/to/inventory –m ping

```
host1.example.com
host2.example.com
host3.example.com
192.168.9.29
```

Figure 2.4 Aperçu sur un inventaire personnalisé

2.3.3. Les roles Ansible

Les rôles sont un concept de base dans Ansible. Ils remplissent une telle fonction de base, en fait, qu'ils ont même leur propre dépôt et outil de ligne de commande. Ansible Galaxy est un site Web où les gens peuvent télécharger des rôles qu'ils ont développés pour d'autres personnes à utiliser. Lors de l'installation d'un rôle, nous pouvons soit l'installer globalement sur notre machine ou localement à un projet. Pour télécharger un rôle, nous utilisons la commande ansible-galaxy, néanmoins nous devrions exécuter la commande ansible-galaxy dans le même répertoire où le fichier playbook.yml peut être trouvé (Heap, 2016).

2.4. Système de gestion de base de données

Le système de gestion de base de données (Système de Gestion de Base de Données-SGBD) est un logiciel qui permet aux organisations de centraliser facilement les données, de les gérer efficacement et de fournir un accès aux données pour les programmes d'application. Le SGBD agit comme une interface entre les programmes d'application et les fichiers de données physiques.

Lorsqu'un programme d'application appelle un fichier de données, tel que le salaire brut, le SGBD recherche ces données dans la base de données et les transmet au programme d'application (Susanto & Meiryani, 2019).

Si nous utilisons des fichiers de données traditionnels, un programmeur doit déterminer la taille et le format de chaque élément de données utilisé dans le programme, puis indiquer à l'ordinateur où se trouve le fichier. Le SGBD facilite la tâche du programmeur ou de l'utilisateur final pour comprendre où et comment les données sont réellement stockées, en les séparant logiquement et physiquement. L'affichage logique présente les données telles qu'elles sont vues par les utilisateurs finaux ou les spécialistes de l'entreprise, tandis que les affichages physiques montrent comment les données sont concrètement organisées et structurées sur des supports de stockage physiques.

Selon les recherches de (Jumardi, Osmond & Saputra, 2019), le SGBD est un logiciel fonctionnant par la création, la maintenance, le traitement et l'utilisation de bases de données à grande échelle. Ainsi, l'entrée et la sortie des données dans la base de données seront gérées par le SGBD. Le SGBD est la partie qui relie la base de données à l'application pour s'assurer que la base de données reste organisée de manière harmonieuse et facilement accessible.

L'utilisation de bases de données pour les ressources informatiques est le principal élément requis pour prendre en charge le stockage centralisé des données. Ainsi, la redondance de ces dernières peut être minimisée, de ce fait en gagne en termes de précision, de fiabilité, de facilité d'utilisation grâce à la simplicité de la structure des BDD, sans avoir oublié que l'utilisation de l'énergie et le coût de maintenance du système de base de données peuvent être optimisés (Mansur & Kasmawi, 2017).

2.5. PostgreSQL

PostgreSQL est un système de gestion de base de données relationnelle open source qui a commencé comme un projet de l'Université de Californie à Berkeley. Il était à l'origine sous licence BSD, mais s'appelle désormais la licence PostgreSQL (TPL). Il a une longue histoire, remontant presque aux débuts des bases de données relationnelles. Il dispose de fonctionnalités de classe entreprise telles que les fonctions de fenêtrage SQL, la possibilité de créer et agrégez les fonctions et utilisez-les également dans les constructions de fenêtre, les expressions de table commune et de table commune récursive, et la réplication en continu. Ces fonctionnalités sont rarement trouvées dans d'autres plates-formes de bases de données open source, mais on les trouve couramment dans les nouvelles versions des bases de données propriétaires telles qu'Oracle, SQL Server et IBM DB2. Ce qui la distingue des autres bases de données, y compris les bases de données propriétaires que nous venons de mentionner, est la facilité avec laquelle nous pouvons l'étendre sans changer la base sous-jacente - et dans de nombreux cas, sans aucune compilation de code (Obe & Hsu, 2017).

PostgreSQL est spécial, car ce n'est pas seulement une base de données, mais c'est aussi une plateforme d'application, ça nous permet d'écrire des procédures stockées et des fonctions dans plusieurs langages de programmation. Son architecture nous offre la flexibilité de

prendre en charge davantage de langages tels que SQL (intégré), PL/pgSQL (intégré), PL/Perl, PL/Python, PL/Java et PL/R. Cette prise en charge d'une grande variété de langages nous permet de résoudre les problèmes d'une façon plus détaillée, par exemple, en utilisant R fonctions statistiques et R idiomes de domaine succincts pour résoudre des problèmes de statistiques ; appeler un service Web via Python ; ou écrire des constructions map reduce puis utiliser ces fonctions dans une instruction SQL.

2.5.1. Outils d'administration PostgreSQL

Il existe trois outils populaires pour gérer PostgreSQL et ceux-ci sont pris en charge par les principaux développeurs de PostgreSQL ; ils ont tendance à rester synchronisés avec les versions de PostgreSQL. De plus, il existe également de nombreuses offres commerciales.

Psql

psql est une interface de ligne de commande qui permet d'écrire des requêtes et de gérer PostgreSQL. Il est livré avec quelques extras intéressants, tels que des commandes d'importation et d'exportation pour les fichiers délimités, et une fonction de rapport qui peut générer une sortie HTML. Psql existe depuis le début de PostgreSQL et est un favori des utilisateurs inconditionnels de PostgreSQL.

PgAdmin

Il s'agit de l'outil d'administration graphique gratuit et largement utilisé pour PostgreSQL, pgAdmin s'exécute sur le bureau et peut se connecter à plusieurs serveurs PostgreSQL indépendamment de la version ou du système d'exploitation, avec PgAdmin nous obtiendrons une excellente vue d'ensemble en explorant tous les objets de la base de données dans l'interface principale.

2.6. Réplication et données externes

PostgreSQL a un certain nombre d'options pour partager des données avec des serveurs externes ou des sources de données. La première option est la propre réplication intégrée de PostgreSQL, qui nous permet d'avoir une copie prête de notre serveur sur un autre serveur PostgreSQL. La deuxième possibilité est le Foreign Data Wrapper (FDW), qui nous permet d'interroger et de copier des données à partir de nombreux types de ressources de données externes en utilisant la norme SQL/Management.

Nous pouvons probablement énumérer d'innombrables raisons justifiant la nécessité de répliquer nos données, mais elles se résument toutes à deux : la disponibilité et l'évolutivité. En cas de défaillance au niveau de notre serveur principal, nous voulons qu'un autre assume immédiatement son rôle. Pour les petites bases de données, nous pouvons simplement nous assurer que nous disposons d'un autre serveur physique prêt et restaurez la base de données dessus, mais pour les bases de données volumineuses, la restauration elle-même peut prendre plusieurs heures. Pour éviter les temps d'arrêt, la réplication de données est le moyen le plus sûr et efficace pour résoudre de tels problèmes.

Afin de bénéficier de la réplication de données sur PostgreSQL, nous ferions mieux de définir quelques points communs.

Maître

Le serveur maître est le serveur de base de données qui est la source des données en cours de réplication et où toutes les mises à jour ont lieu.

Esclave

Un esclave est un serveur sur lequel les données sont copiées. Des termes plus esthétiques tels que l'abonné ou l'agent ont été évoqués, mais l'esclave est toujours le plus à propos. La réplication intégrée de PostgreSQL ne prend actuellement en charge que les esclaves en lecture seule.

Write-ahead Log (WAL)

WAL est le journal qui garde une trace de toutes les transactions. Il est souvent appelé journal des transactions dans d'autres bases de données. Pour configurer la réplication, PostgreSQL met simplement les journaux à la disposition des esclaves. Une fois que les esclaves ont les journaux, ils n'ont plus qu'à exécuter les transactions qu'ils contiennent.

Synchrone

Une transaction sur le maître ne sera pas considérée comme terminée tant que tous les esclaves n'auront pas été mis à jour, garantissant l'absence de perte de données.

Asynchrone

Une transaction sur le maître sera validée même si les esclaves n'ont pas été mis à jour. Ceci est utile dans le cas de serveurs distants où nous ne voulons pas que les transactions attendent à cause de la latence du réseau, mais l'inconvénient est que notre ensemble de données sur l'esclave peut être en retard et que l'esclave peut manquer certaines transactions en cas d'échec de transmission.

Streaming

Le modèle de réplication en continu a été introduit dans la version 9.0. Contrairement aux versions précédentes, il ne nécessite pas d'accès direct aux fichiers entre le maître et les esclaves. Au lieu de cela, il s'appuie sur le protocole de connexion PostgreSQL pour transmettre les WAL.

Réplication en cascade

Les esclaves peuvent recevoir des journaux d'esclaves proches au lieu de les recevoir directement du maître. Cela permet à un esclave de se comporter également comme un maître à des fins de réplication tout en n'autorisant que les requêtes en lecture seule.

2.7. Conclusion

Dans ce chapitre nous avons présenté l'outil d'automatisation Ansible, Nous avons mentionné les différences avec Pupete et Chef. Comme nous avons introduit le système de gestion de bases de données PostgreSQl.

Dans le chapitre suivant, nous présentons une revue de la littérature sur les pandémies, l'intelligence artificielle et l'implémentation des infrastructures dans le domaine de la santé.

CHAPITRE 3

REVUE DE LA LITTERATURE

3.1 Introduction

Ce chapitre présente une revue de la littérature sur la gestion de données médicales et l'utilisation de l'intilligence artificielle dans le domaine de la santé ainsi que l'implémentation des infrastructures.

3.2 Introduction à la gestion de données médicales

En 2012, une étude remarquait que les données médicales représentaient la part la plus importante, soit 30 %, de l'ensemble du stockage de données électroniques à l'échelle mondiale (Brown, 2015). Cette constatation souligne l'ampleur et l'importance croissante des données de santé dans le paysage numérique. En effet, le secteur de la santé connaît une croissance exponentielle au sein de l'univers numérique en expansion. Selon un rapport d'EMC Corporation et de la société de recherche IDC (International Data Corporation), les données numériques relatives aux soins de santé augmentent à un taux annuel de 48 %, dépassant ainsi le taux de croissance de 40 % de l'univers numérique global (IDC, 2014). Cette progression rapide met en évidence l'urgence et la nécessité d'une gestion efficace des données médicales pour optimiser les soins de santé et permettre des avancées significatives dans la recherche médicale.

3.2.1 Les défis spécifiques liés à la gestion des données médicales

La gestion des données médicales est confrontée à plusieurs défis importants qui nécessitent une attention particulière. L'un de ces défis majeurs réside dans le volume massif de données générées dans le domaine de la santé. Avec l'adoption croissante des dossiers médicaux électroniques, des images médicales haute résolution, des données génomiques et d'autres sources, le volume de données médicales a atteint des proportions colossales. La capacité de stocker, gérer et analyser efficacement ces quantités massives de données devient donc primordiale.

Un autre défi de taille est la variété des formats de données. Les données médicales proviennent de diverses sources telles que les systèmes d'information hospitaliers, les appareils médicaux connectés, les laboratoires et les essais cliniques. Ces données peuvent être structurées, comme les résultats de laboratoire et les données démographiques des patients, ou non structurées, comme les notes cliniques et les images radiologiques. La gestion de cette variété de formats de données nécessite des outils et des méthodes adaptés pour intégrer, normaliser et interpréter ces données hétérogènes (Deng et al., 2015).

En outre, les problèmes de qualité des données constituent un défi significatif dans la gestion des données médicales. Les erreurs de saisie, les valeurs manquantes, les duplications et les incohérences peuvent compromettre la qualité des données. Une mauvaise qualité des données peut entraîner des résultats incorrects, des décisions cliniques erronées et des conclusions erronées dans la recherche médicale. Il est donc essentiel d'établir des processus rigoureux de contrôle de la qualité et de nettoyage des données afin de garantir la fiabilité et l'intégrité des données utilisées (Kahn et al., 2012).

Ces défis complexes de gestion des données médicales requièrent des approches stratégiques et des solutions technologiques avancées. Les progrès continus dans le domaine de l'informatique, tels que l'intelligence artificielle, l'apprentissage automatique et l'analyse des données, ouvrent de nouvelles possibilités pour surmonter ces défis et tirer pleinement parti du potentiel des données médicales pour améliorer les soins de santé et promouvoir la recherche médicale (Chen et al., 2018).

3.2.2 Les systèmes d'information de santé (SIS)

Les systèmes d'information de santé (SIS) jouent un rôle essentiel dans la gestion des données médicales. Les SIS sont des systèmes informatiques utilisés pour collecter, stocker, gérer et transmettre les informations relatives à la santé des individus, aux soins de santé et aux activités cliniques. Ils constituent une infrastructure cruciale pour la gestion efficace des données médicales et la prise de décisions cliniques informées (HealthIT.gov, 2020).

Les SIS sont utilisés dans divers contextes de soins de santé, tels que les hôpitaux, les cliniques, les centres de santé et les établissements de recherche. Ils permettent de centraliser

les informations sur les patients, les antécédents médicaux, les résultats des tests, les ordonnances médicales et les rendez-vous. Les SIS facilitent également la communication et la coordination entre les différents acteurs de la santé, tels que les médecins, les infirmières, les pharmaciens et les professionnels de la santé (World Health Organization, 2018).

Ces systèmes jouent un rôle clé dans la gestion des données médicales en offrant plusieurs avantages. Tout d'abord, ils permettent la collecte et le stockage sécurisés des données médicales, éliminant ainsi le besoin de documents papier et facilitant l'accès aux informations pertinentes. De plus, les SIS peuvent intégrer des fonctionnalités d'interopérabilité, permettant le partage sécurisé des données entre les différents systèmes et fournisseurs de soins de santé (Blumenthal & Tavenner, 2010).

Les SIS contribuent également à l'analyse des données médicales, en permettant l'extraction d'informations utiles à partir des ensembles de données. Cela peut faciliter la recherche clinique, l'identification de tendances de santé, la surveillance des épidémies et la prise de décisions basées sur des données probantes (Hersh & Totten, 2018).

Cependant, la mise en place et la gestion des SIS ne sont pas sans défis. Certains des défis courants comprennent la nécessité d'assurer la confidentialité et la sécurité des données, la standardisation des formats de données et des terminologies, ainsi que la formation et l'adoption efficaces par les utilisateurs finaux.

Parmi les systèmes d'information de santé (SIS) qui sont largement utilisés dans le domaine de santé on trouve :

- Les systèmes d'information hospitalier (SIH) : Les SIH sont des SIS utilisés dans les hôpitaux pour gérer les données médicales des patients, les rendez-vous, les résultats de laboratoire, les prescriptions, les informations sur les traitements et d'autres informations cliniques. Un exemple populaire de SIH est le système EPIC, largement utilisé dans de nombreux établissements de santé à travers le monde.

- Les dossiers médical électronique (DME) : Les DME sont des SIS utilisés pour stocker et gérer les informations médicales complètes d'un patient. Ils regroupent les données

provenant de différentes sources, telles que les résultats de tests, les antécédents médicaux, les allergies, les traitements et les informations de facturation. Parmi les exemples de DME figurent Cerner, Allscripts et Meditech.

- Les systèmes de gestion de l'imagerie (RIS/PACS) : Les systèmes RIS (Radiology Information System) et PACS (Picture Archiving and Communication System) sont des SIS spécialisés utilisés dans les départements d'imagerie médicale, tels que la radiologie, pour gérer et stocker les images radiologiques et les rapports d'examens.

3.2.3 La sécurité et la confidentialité des données médicales

La sécurité et la confidentialité des données médicales revêtent une importance capitale en raison de la sensibilité et de la valeur de ces informations. La protection de la vie privée des patients et la sécurité des données de santé sont essentielles pour garantir la confiance et maintenir l'intégrité du système de santé. Voici quelques mesures de sécurité couramment utilisées pour protéger les données médicales :

- Anonymisation des données : L'anonymisation est une méthode qui consiste à supprimer ou à modifier les informations personnellement identifiables dans les données médicales, afin de les rendre anonymes. Cette technique permet de préserver la confidentialité des patients tout en permettant l'utilisation des données à des fins de recherche et d'analyse.

- Chiffrement des données : Le chiffrement est le processus de conversion des données en un format illisible, sauf pour les parties autorisées disposant de la clé de chiffrement appropriée. Le chiffrement des données médicales garantit qu'elles ne peuvent être lues que par des personnes autorisées, même en cas d'accès non autorisé.

- Authentification : L'authentification est un mécanisme qui vérifie l'identité des utilisateurs avant de leur accorder l'accès aux données médicales. Cela peut inclure l'utilisation de mots de passe forts, de codes d'accès, de systèmes de reconnaissance biométrique ou d'autres méthodes d'authentification à plusieurs facteurs.

- Contrôles d'accès : Les contrôles d'accès permettent de définir des autorisations d'accès spécifiques pour les utilisateurs en fonction de leur rôle et de leur responsabilité. Ces

contrôles garantissent que seules les personnes autorisées ont accès aux informations médicales sensibles, limitant ainsi les risques d'accès non autorisé.

- Gestion des droits d'accès : La gestion des droits d'accès consiste à attribuer des privilèges d'accès appropriés à chaque utilisateur en fonction de sa fonction et de ses besoins. Cela garantit que chaque utilisateur ne peut accéder qu'aux informations nécessaires à l'exécution de ses tâches spécifiques, réduisant ainsi les risques de divulgation ou d'utilisation abusive des données.

3.3 L'intelligence artificielle dans le domaine de la santé

La complexité et l'augmentation des données dans le domaine de la santé signifient que l'intelligence artificielle (IA) sera de plus en plus appliquée sur le terrain. Plusieurs types d'IA sont déjà utilisés par les payeurs et les prestataires de soins, ainsi que par les entreprises des sciences de la vie. Les principales catégories d'applications concernent les recommandations de diagnostic et de traitement, l'engagement et l'observance des patients, et les activités administratives. Bien qu'il existe de nombreux cas dans lesquels l'IA peut effectuer des tâches de santé aussi bien ou mieux que les humains, les facteurs de mise en œuvre empêcheront l'automatisation à grande échelle des emplois des professionnels de la santé pendant une période considérable. Les questions éthiques dans l'application de l'IA aux soins de santé sont également abordées (Davenport & Kalakot, 2019).

Le plus grand défi pour l'IA dans ces domaines de santé n'est pas de savoir si les technologies seront suffisamment capables d'être utiles, mais plutôt d'assurer leur adoption dans la pratique clinique quotidienne. Pour qu'une adoption généralisée ait lieu, les systèmes d'IA doivent être approuvés par les régulateurs, intégrés aux systèmes de dossiers de santé électroniques (DSE), standardisés à un degré suffisant pour que des produits similaires fonctionnent de la même manière, enseignés aux cliniciens, payés par un payeur public ou privé, et mis à jour au fil du temps sur le terrain. Ces défis finiront par être surmontés, mais ils prendront beaucoup plus de temps qu'il n'en faudra pour que les technologies elles-mêmes arrivent à maturité. En conséquence, nous attendons à voir une utilisation limitée de l'IA dans la pratique clinique d'ici 5 ans et une utilisation plus étendue d'ici 10 ans. Il semble également

de plus en plus clair que les systèmes d'IA ne remplaceront pas les cliniciens humains à grande échelle, mais augmenteront plutôt leurs efforts de soins pour les malades. Au fil du temps, les cliniciens humains peuvent évoluer vers des tâches et des conceptions de travail qui s'appuient sur des compétences humaines uniques telles que l'empathie, la persuasion et l'intégration globale. Peut-être que les seuls prestataires de soins de santé qui perdront leur emploi au fil du temps seront peut-être ceux qui refusent de travailler aux côtés de l'intelligence artificielle (Davenport & Kalakot, 2019).

L'intelligence artificielle vise à imiter les fonctions cognitives humaines. Elle apporte un changement de paradigme aux soins de santé, alimenté par la disponibilité croissante des données de santé et les progrès rapides des techniques d'analyse. Nous examinons l'état actuel des applications de l'IA dans les soins de santé et discutons de son avenir. L'IA peut être appliquée à différents types de données de santé (structurées et non structurées). Les techniques d'IA populaires incluent les méthodes d'apprentissage automatique pour les données structurées, telles que la machine à vecteurs de support classique et le réseau de neurones, et l'apprentissage en profondeur moderne, ainsi que le traitement du langage naturel pour les données non structurées. Les principaux domaines de maladies qui utilisent des outils d'IA comprennent la cancérologie, la neurologie et la cardiologie (Jiang, 2017).

Durant les dernières années, l'accent a été mis sur l'utilisation de l’IA dans divers domaines dans le but de résoudre des problèmes complexes. De même, l'adoption de l’IA dans le domaine des soins de santé se développe tout en changeant radicalement le visage de la prestation des soins de santé. L'IA est utilisée dans une myriade de contextes, notamment dans les hôpitaux, les laboratoires cliniques et les centres de recherche. Les approches de l'IA employant des machines pour détecter et comprendre les données comme des humains ont ouvert des opportunités précédemment indisponibles ou non reconnues pour les praticiens cliniques et les organisations de services de santé. Parmi les exemples, on peut citer l'utilisation d'approches d'IA pour analyser des données non structurées telles que des photos, des vidéos, des notes de médecin, afin de faciliter la prise de décision clinique, l'utilisation d'interfaces intelligentes pour améliorer l'engagement des patients et l'observance du traitement, et la modélisation prédictive pour gérer le flux des patients et la capacité des

hôpitaux en assurant une meilleure répartition des ressources. Pourtant, la compréhension de l'IA est incomplète et il existe même une confusion quant à sa nature. De même, les implications de l'utilisation de l'IA en général et pour les cliniciens en particulier ne sont pas tout à fait claires. Ce chapitre vise à couvrir ces sujets et à présenter le concept d'IA, les théories qui sous-tendent la programmation de l'IA et les diverses applications de l'IA dans le domaine médical (Sandeep Reddy, 2018).

Au fil des ans, la prestation des soins de santé est devenue complexe et difficile. Une grande partie de la complexité de la prestation des soins de santé est due aux données volumineuses qui sont générées dans le processus de soins de santé, qui doivent être interprétées de manière intelligente. Les systèmes d'IA, avec leur approche de résolution de problèmes, peuvent répondre à ce besoin. Leur architecture intelligente, qui intègre l'apprentissage et le raisonnement et la capacité d'agir de manière autonome sans nécessiter une attention humaine constante est séduisante. Ainsi, le domaine médical a fourni un terrain fertile aux chercheurs en IA pour tester leurs techniquesdans de nombreux cas, les applications d'IA ont résolu avec succès des problèmes avec des résultats comparables à ceux des cliniciens humains. Comme la prestation des soins de santé devient plus coûteuse, les parties prenantes chercheront de plus en plus des solutions qui peuvent remplacer les éléments coûteux dans les soins aux patients et les solutions d'IA seront recherchées dans ces situations. Cependant, la technologie froide ne peut pas remplacer totalement les éléments humains dans les soins aux patients, et un modèle qui intègre à la fois les innovations technologiques et les soins humains doit être étudié.

3.4 Cris de nourrissons

Dans le cadre de ce projet, et afin de traiter et gérer des données médicales critiques, nous avons opté pour travailler sur un jeu de données déjà existant qui contient des échantillons de cris de nourrissons sains et malades.

Chaque année, environ 130 millions bébés naissent dans le monde. Prendre soin des nouveau-nés est un grand défi, surtout pour les nouveaux parents. Suivre les suggestions des autres parents et des livres ne suffit pas pour résoudre les problèmes dans la pratique. La

raison principale est qu'il est difficile de comprendre le sens des pleurs de l'enfant. Les bébés communiquent avec le monde en pleurant, les parents, les soignants, les médecins et les infirmières d'expérience comprennent les cris en fonction de leur expérience.

Les jeunes parents sont frustrés et ont de la difficulté à calmer leurs bébés parce que tous les signaux de pleurs sonnent de la même façon. Une interprétation précise du cri des bébés peut aider les parents à mieux prendre soin de leur bébé.

La recherche sur les pleurs infantiles a commencé dès les années 1960, lorsque le groupe de recherche Wasz–Hockert a identifié les quatre types de cris (douleur, faim, naissance et plaisir) auditivement par des infirmières formées.

Dans les premières années, les recherches ont déterminé que différents types de cris peuvent être différenciés auditivement par des auditeurs adultes formés. Mais former la perception humaine aux pleurs infantiles est beaucoup plus difficile que former des modèles d'apprentissage automatique.

Dans l'étude de Mukhopadhyay, la plus grande précision de classification en formant un groupe de personnes à reconnaître certains sons de cri est de 33,09% tandis que l'algorithme d'apprentissage automatique basé sur des caractéristiques spectrales et prosodiques peut reconnaître le même ensemble de données et atteindre 80,56% de précision (Ji, Mudiyanselage, Yutong & Pan, 2021).

3.4.1 Système vocal humain

En tant que principal moyen de communication, la voix joue un rôle important dans la vie quotidienne. La voix transmet également des informations personnelles telles que le statut social, les traits personnels et l'état émotionnel de l'orateur. Mécaniquement, la production de la voix implique une interaction fluide-structure complexe dans la glotte et son contrôle par l'activation des muscles laryngés (Zhang, 2016).

la voix, fait référence aux sons produits par la vibration des cordes vocales ou aux sons vocaux. Cela contraste avec les sons non vocaux qui sont produits sans vibration des cordes

vocales, par exemple, les fricatives qui sont produites par le flux d'air à travers des constrictions dans le tractus vocal, les plosives produites par la libération soudaine d'une fermeture complète du tractus vocal, ou d'autres mécanismes de production de sons tels que comme chuchotant. Pour la production sonore vocale, la vibration des cordes vocales module le flux d'air à travers la glotte et produit un son (la source vocale), qui se propage à travers le tractus vocal et est sélectivement amplifié ou atténué à différentes fréquences. Cette modification sélective du spectre de la source vocale produit des contrastes perceptibles, qui sont utilisés pour transmettre différents sons et significations linguistiques. La source vocale contient également des informations importantes sur la hauteur, le volume, la prosodie et la qualité de la voix, qui transmettent un sens (Zhang, 2016).

3.4.2 Sources sonores de la production vocale

Le processus de phonation commence à partir de l'adduction des cordes vocales, qui se rapproche des cordes vocales pour réduire ou fermer la glotte. La contraction des poumons initie le flux d'air et établit une accumulation de pression sous la glotte. Lorsque la pression sous-glottale dépasse un certain seuil de pression, les cordes vocales sont excitées dans une vibration auto-entretenue. La vibration des cordes vocales module à son tour le flux d'air glottal en un flux de jet pulsé, qui se développe finalement en un flux turbulent dans le conduit vocal (Zhao & all, 2002).

En général, trois mécanismes majeurs de production sonore sont impliqués dans ce processus, dont une source sonore monopolistique due au volume d'air déplacé par la voix. vibration du pli, une source sonore dipolaire due à la force fluctuante appliquée par les cordes vocales au flux d'air, et une source sonore quadripolaire due à la turbulence développée immédiatement en aval de la sortie glottale (Zhang, 2016).

3.4.3 Classification des pathologies infantiles basée sur les pleurs de nourrissons

Les cris du nourrisson peuvent fournir des informations précieuses sur l'état de santé du nourrisson. Par exemple, des études ont montré que les nourrissons atteints de certaines maladies génétiques ont des modèles de cris différents de ceux des nourrissons en bonne santé (Nakamura et al., 2013). De plus, des études ont montré que les nourrissons qui sont

nés prématurément ont des cris différents de ceux des nourrissons nés à terme (Esposito et Venuti, 2008).

Ces différences de cris peuvent être liées à des différences dans les propriétés acoustiques des cris, telles que la fréquence et l'intensité, ainsi que dans la structure temporelle des cris. Par exemple, les cris des nourrissons atteints de certaines maladies génétiques peuvent avoir des fréquences plus élevées que ceux des nourrissons en bonne santé (Nakamura et al., 2013).

Le rôle principal dans la classification du cri du nourrisson est de savoir comment faire une distinction scientifique entre les différents états de santé néonatals, uniquement sur la base de leurs signaux de cri en plus de l'examen de santé des nourrissons et d'autres prédicteurs de la santé de l'enfant. Une analyse acoustique détaillée, qui mesure et compare les caractéristiques acoustiques des signaux de cri du nouveau-né, montre le potentiel diagnostique caché des signaux de cri pour les types de cri de base et les cris des nourrissons dans des conditions pathologiques telles que des lésions cérébrales, des maladies du système nerveux central et la trisomie 21 (Abou-Abbas et al., 2017).

La figure 3.1 illustre quelques causes de la mortalité des enfants âgés de moins de 5 ans.

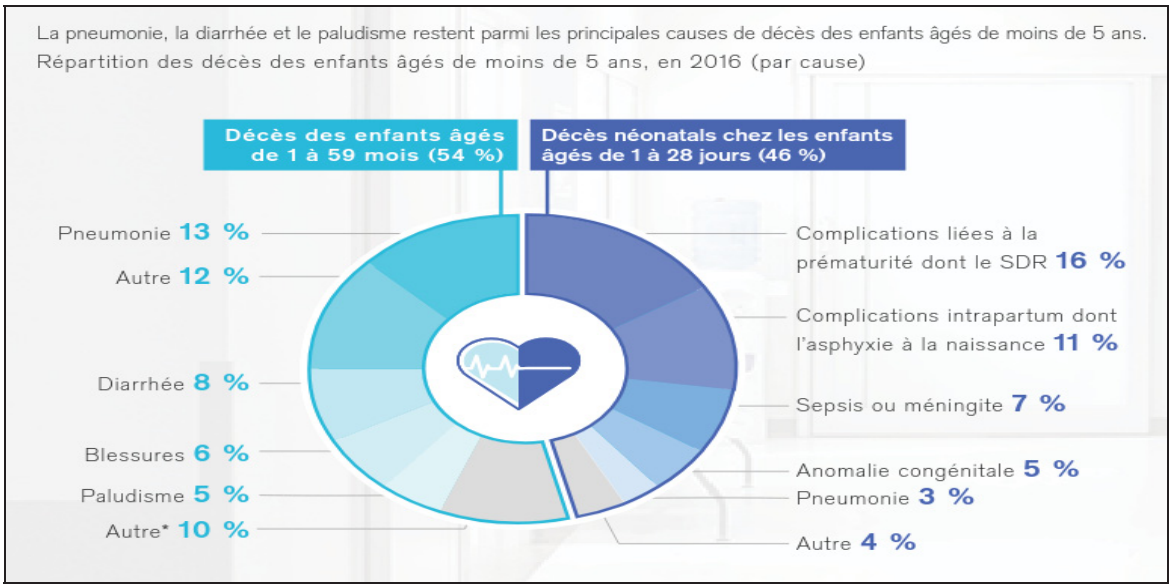

Figure 3.1 Les causes de la mortalité néonatale dans les pays en voie de développement tirée de draeger.com

La recherche automatique sur les pleurs infantiles comporte généralement cinq étapes (Figure 3.2) : acquisition des données, prétraitement, extraction des caractéristiques, sélection des caractéristiques et classification (Ji & all, 2021).

Figure 3.2 Les cinq étapes de la recherche sur le cri du nourisson

La découverte de nouvelles méthodes à n'importe laquelle des étapes peut aider à améliorer l'exactitude de la classification finale.

Dans ce contexte, plusieurs études ont été effectuées afin de faciliter le travail des professionnels de santé, pour diagnostiquer les différentes pathologies qui peuvent affecter les bébés.

Cependant, la réalisation d'une procédure de diagnostic néonatal systématique pour tous les nouveau-nés exige des coûts élevés, car elle implique la participation de nombreux professionnels de la santé et d'équipements spécialisés. Parmi les solutions adoptées, on trouve la reconnaissance automatique qui consiste à classer les pleurs en utilisant des techniques avancées de traitement des signaux, et une analyse du cri spectral fondée sur l'observation des spectrogrammes de signaux de cri et des outils logiciels.

Des paramètres acoustiques qui qualifient le tractus vocal, tels que le MFCC (Mel-Frequency Cepstral Coefficients) et la dysrégulation des fréquences de résonance (RFsdys), et le pli vocal, comme le glissement de fréquence fondamentale (F0glide) sont utilisés afin de permettre l'analyse d'un nombre considérable de signaux de cris pathologiques en peu de temps (Kheddache & Tadj, 2019).

Les techniques d'analyse de signal conventionnelles fonctionnent avec des trames à court terme de signaux avec une dynamique non stationnaire, comme la parole humaine. Pour extraire des caractéristiques à partir de ces signaux, il est courant d'utiliser les MFCC, qui

sont une technique d'analyse cepstrale utilisée pour extraire les caractéristiques fréquentielles des signaux vocaux. Les MFCC sont calculés à partir de la sortie log-énergie des filtres passe-bande triangulaires qui couvrent la bande de fréquences efficace des signaux de cri. La transformée de Fourier rapide (FFT) est d'abord effectuée pour obtenir la fréquence d'amplitude de chaque trame fenêtrée, puis les MFCC sont calculés en reconvertissant le spectre log Mel dans le domaine temporel à l'aide de la transformée en cosinus discrète (DCT).

Dans le contexte de l'analyse des signaux de cri, l'extraction de caractéristiques peut être effectuée en utilisant deux durées de trame différentes, 10 et 30 ms, avec le même pourcentage de chevauchement entre deux fenêtres consécutives. Avant toute analyse dans le domaine fréquentiel, le fenêtrage de Hamming est appliqué pour réduire les discontinuités aux bords de la région sélectionnée. Pour détecter la fréquence de coupure supérieure de la bande de fréquence efficace des signaux de pleurs du nourrisson, où la puissance cesse presque d'augmenter, le spectre de puissance cumulé est utilisé. Les résultats des expériences montrent que les signaux de cri des nourrissons nés à terme en bonne santé et malades ont respectivement près de 94 % et 98 % de leurs énergies en dessous de 4 kHz et 6,8 kHz. Les énergies des segments d'inspiration pour les nourrissons malades ont tendance à s'accumuler à un rythme plus lent que les énergies des segments d'expiration, en particulier dans les cas de troubles du syndrome de détresse respiratoire.

3.5 L’implémentation des infrastructures de soin de santé

Une infrastructure nationale d’information dans le domaine de la santé devrait répondre aux besoins de la gestion de la santé personnelle, de la prestation des soins de santé, de la santé publique et de la recherche. Il devrait également aborder les dimensions mondiales pertinentes (par exemple, les normes pour le partage des données et des connaissances au-delà des frontières nationales). Les secteurs public et privé devront collaborer pour construire une infrastructure nationale d’information sur la santé robuste, essentiellement un système de soins de santé sans papier. Des progrès sont nécessaires dans les domaines du financement, des incitatifs, des normes et de l’amélioration continue d’un cadre de protection de la vie

privée (c.-à-d. Confidentialité et sécurité) afin de faciliter l'identification personnelle à des fins de santé.

Ces infrastructures existent pour connecter les utilisateurs entre eux, à l'information et aux outils d'analyse et pour permettre la gestion et la génération de connaissances. La connectivité est obtenue grâce à une combinaison de technologies, de normes pour la transmission de données et de règles et de processus convenus, en reliant la multitude de participants du secteur de la santé qui interagissent régulièrement et fournis les moyens de gérer les volumes massifs de données, d'informations et de connaissances sur la santé qui augmentent d'heure en heure. Une fois pleinement mis en œuvre, ces infrastructures permettrait également l'automatisation des tâches de routine, la simplification des tâches complexes, la démocratisation des fonctions, la personnalisation des services, la gestion de la base de connaissances et une plus grande collaboration dans les domaines du secteur de la santé (D Detmer, 2003).

D'atres études ont mis l'accent sur l'importance d'introduire Internet des objets (IoT) dans l'implémentation des infrastructures de santé.

L'IoT est une combinaison d'objets intelligents distribués spatialement qui ont des capacités de détection et d'identification intégrée grâce à la technologie radio frequency identification (RFID). Plus précisément, l'intégration de capteurs, d'étiquettes RFID et de technologies de communication constitue le fondement de l'IoT. Il traite de la traçabilité, de la visibilité et de la contrôlabilité des objets intelligents. Il s'agit d'une vision d'avenir à travers laquelle les objets numériques et physiques peuvent être interconnectés et communiqués pour fournir certains services spécifiques à un domaine (Adil, M., & Khan, 2021). L'IoT transforme des objets du monde réel en objets intelligents capables de détecter la quantité physique de l'environnement et de la communiquer en conséquence. La technologie RFID est maintenant largement utilisée dans le suivi des personnes, des objets et des animaux. Les codes de produits électroniques sont codés dans des étiquettes RFID qui peuvent être utilisées pour suivre les objets intelligents dans l'IoT. Pour le stockage des données entrantes à partir de ces appareils intelligents, la technologie Cloud est l'une des options appropriées (Jabbar, Ullah, Khalid & Khan, 2017).

Les infrastructures de soin de santé sont les meilleurs moyens pour ralentir la propagation des pandémies telle que la Covid19 (sharma, borah & moses, 2021). L'OMS mesure les dépenses de santé par pays comme l'investissement total dans le développement des infrastructures de soins de santé, la recherche et l'amélioration de l'accès et de l'abordabilité (OMS, 2020), et elle peut affecter l'apprentissage organisationnel de trois manières. Premièrement, l'augmentation des infrastructures de soins de santé aide un pays à mettre l'accent sur les experts en soins de santé à l'interne et les institutions de santé mondiales à l'externe. Conformément à la littérature qui suggère que les organisations peuvent s'appuyer sur les parties prenantes internes pour développer des réponses aux crises par l'apprentissage (Bundy, 2017), l'augmentation de l'infrastructure de soins de santé peut aider un pays à accumuler des connaissances tacites sur le pays auprès d'experts internes et à les intégrer dans le système. Entre-temps, des connaissances explicites sur le contrôle des maladies fournies par des experts externes peuvent aider le système à répondre à ses besoins immédiats. De plus, l'augmentation des investissements dans les soins de santé permet aux individus au sein d'un système de santé d'utiliser l'intuition pour rechercher de nouvelles connaissances, les interpréter selon le contexte existant et les intégrer dans le système existant (Chadwick & Raver, 2015). De surcroît, l'augmentation des dépenses de santé permet à un pays d'institutionnaliser l'apprentissage (Nonaka, 1994), ce qui a une incidence positive sur les réponses organisationnelles en cas de crise.

3.6 Conclusion

Nous avons traité dans ce chapitre la gestion de données médicales ainsi l'application de l'intelligence artificielle dans le domaine de santé.

Nous allons présenter dans le prochain chapitre les différentes étapes suivies durant l'implémentation de notre solution.

CHAPITRE 4

IMPLEMENTATION ET EVALUATION

4.1. Introduction :

Dans ce chapitre, nous allons procéder à la préparation de notre environnement de travail et nous exécutons le Playbook sur les serveurs pour créer les bases de données, et en fin nous allons traiter l'aspect sécurité de notre environnement de travail.

La figure 4.1 illustre un aperçu sur l'infrastructure et les différents serveurs que nous avons créée.

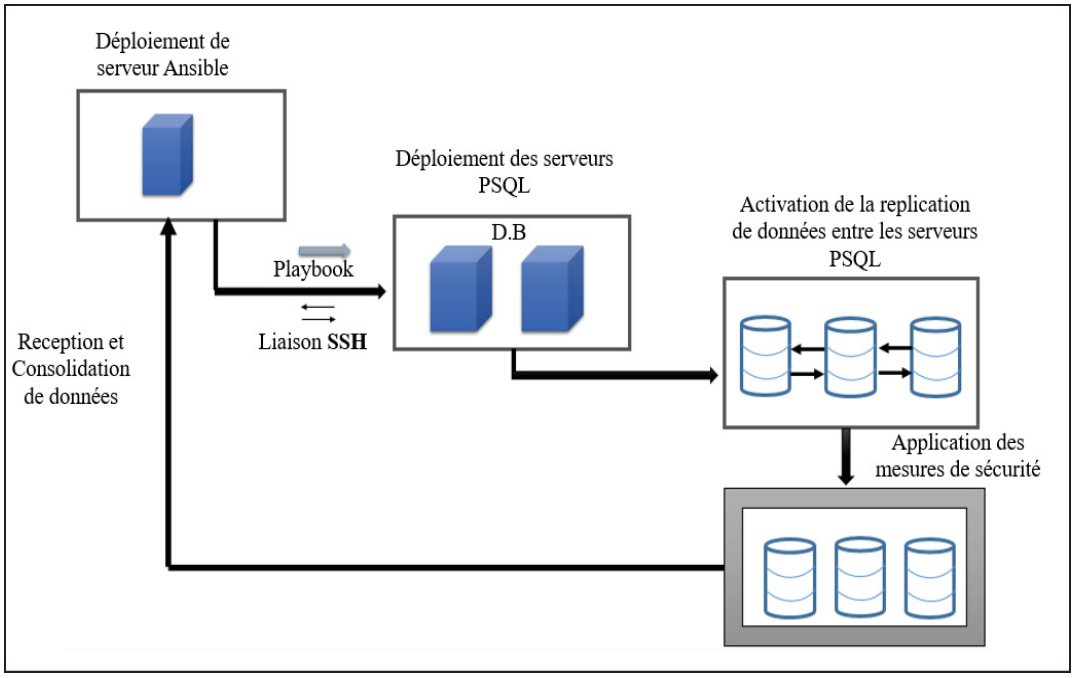

Figure 4.1 illustration de l'infrastructure

4.2. Déploiement des serveurs :

Notre environnement de travail se compose d'un serveur maitre avec lequel nous exécutons le Playbook Ansible et 3 autres serveurs pour les bases de données Postgresql. Ces serveurs seront déployés dans des conteneurs docker

Les étapes nécessaires pour déployer l'environnement de travail sont présentées à l'Annexe I.

- *Execution de playbook:*

Apres l'installation de ansible nous allons pouvoir exécuter les playbook qui contiennent l'ensemble des tâches que nous volons appliquer sur les serveurs distants. La figure 4.2 nous montre le résultat de l'exécution de playbook à partir de la machine master.

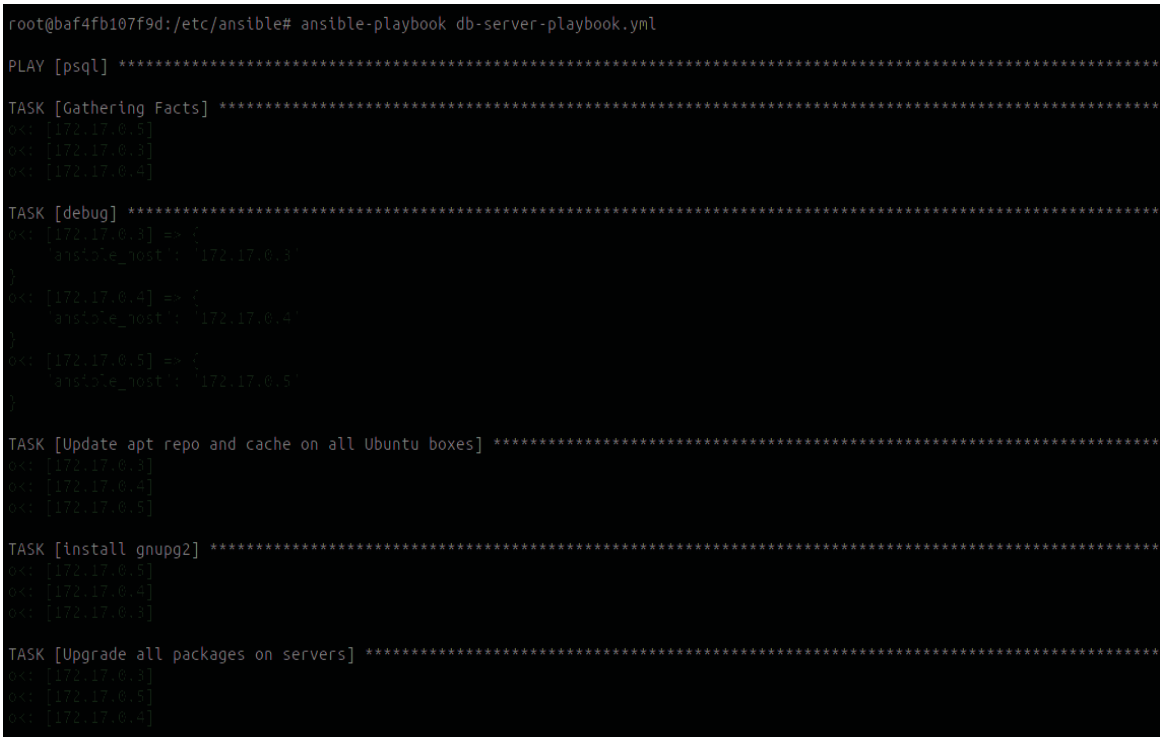

Figure 4.2 Le Playbook Ansible

Après la préparation de toutes les machines host sur docker, nous exécutons le Playbook Ansible à partir de la machine master dont nous pouvons suivre le déroulement de l'exécution de chaque tâche.

La figure 4.3 illustre le résultat de l'exécution de playbook

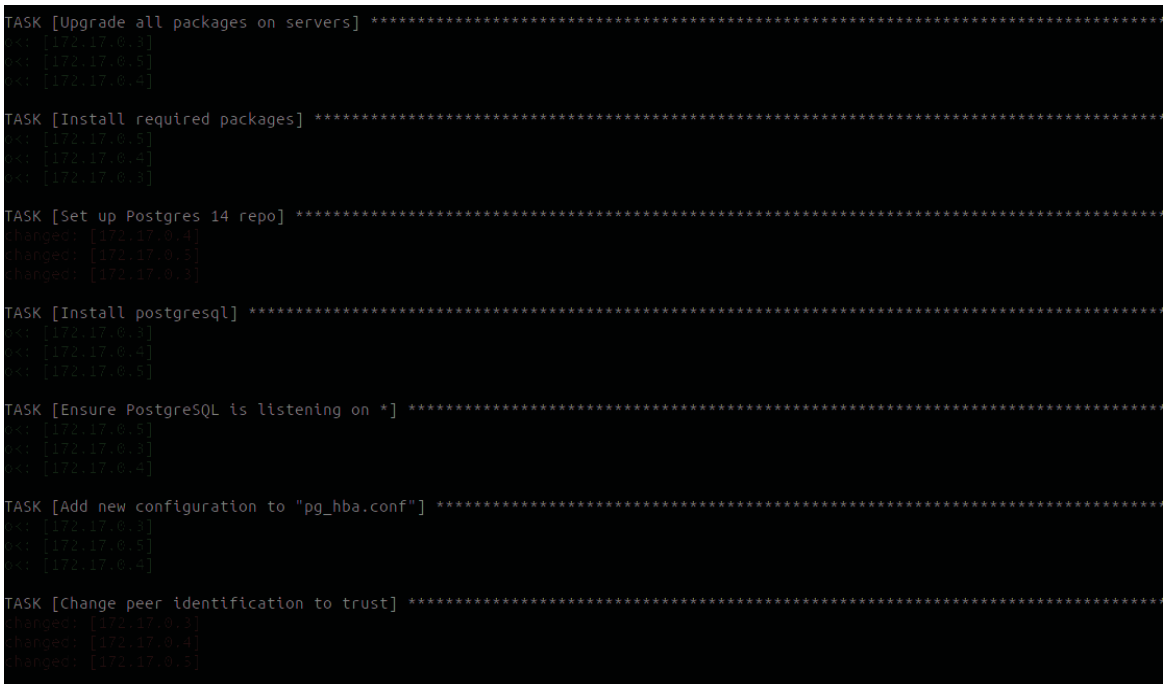

Figure 4.3 exécution de Playbook Ansible

Durant l'exécution, nous pouvons vérifier si la tâche a été bien exécuté sur chaque serveur comme c'est démontré sur la figure 4.4.

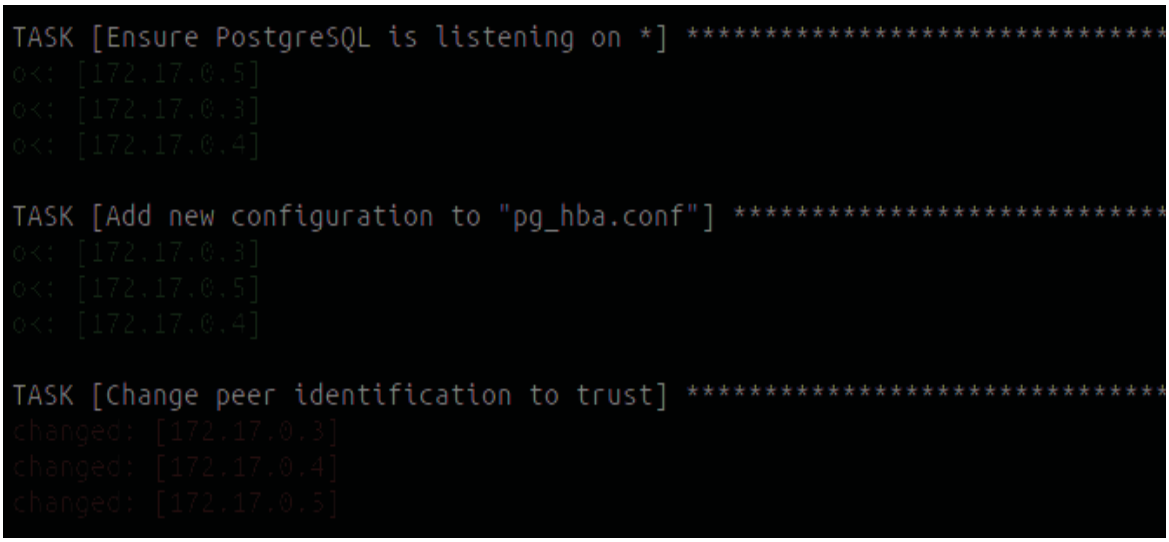

Figure 4.4 exécution de Playbook Ansible

À la fin de l'exécution de Playbook Ansible, les bases de donnes Psql seront disponibles et prêtes pour l'utilisation. Et à l'aide de la commande suivante, nous pouvons se connecter à l'une de nos bases de données avec le nom d'utilisateur et le mot de passe précédemment définis dans le Playbook.
La figure 4.5 présente la connexion au serveur de base de données PSQL

```
root@844a10b682a0:/etc# psql -h localhost -p 5432 -U root postgres
Password for user root:
psql (14.4 (Ubuntu 14.4-1.pgdg20.04+1))
SSL connection (protocol: TLSv1.3, cipher: TLS_AES_256_GCM_SHA384, bits: 256, compression: off)
Type "help" for help.

postgres=#
```

Figure 4.5 Connexion au serveur PostgreSQL

➢ *Installation de Pgadmin*

PostgreSQl propose plusieurs méthodes pour gérer les bases de données, parmi ces outils Pgadmin, dans la figure 4.6 nous pouvons voire les étapes à suivre pour installer Pgadmin

```
# Setup the repository
#

# Install the public key for the repository (if not done previously):
sudo curl https://www.pgadmin.org/static/packages_pgadmin_org.pub | sudo apt-key add

# Create the repository configuration file:
sudo sh -c 'echo "deb https://ftp.postgresql.org/pub/pgadmin/pgadmin4/apt/$(lsb_release -cs) pgadmin4 main" > /etc/apt

#
# Install pgAdmin
#

# Install for both desktop and web modes:
sudo apt install pgadmin4
```

Figure 4.6 installation de Pgadmin

Pgadmin est doté d'une interface graphique (Figure 4.7) qui nous permettre de réaliser tous sorte d'opérations sur les données telles que la création des tables l'insertion la suppression de données.

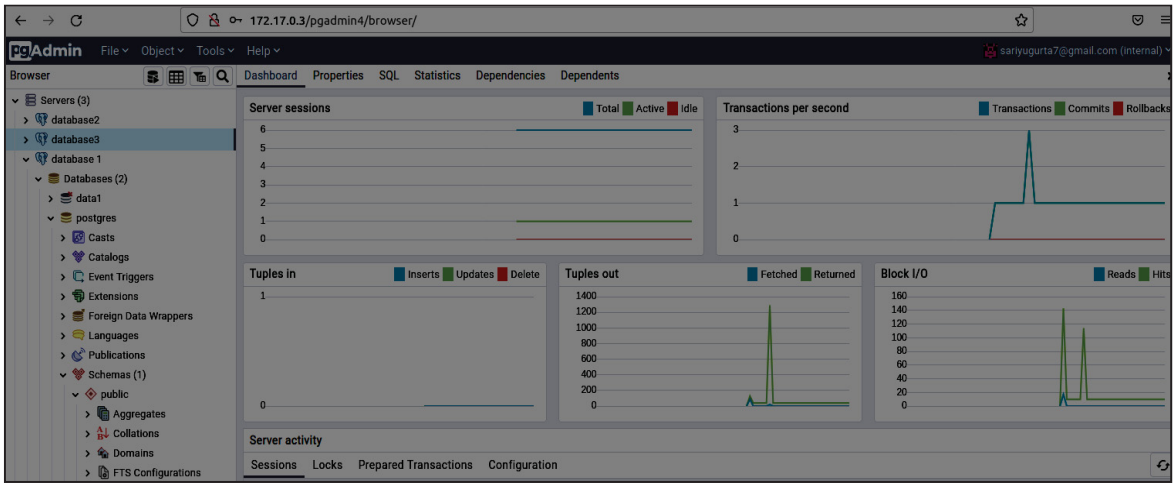

Figure 4.7 interface graphique de Pgadmin

➢ *Implémentation d'une base de données dédiés aux données d'apprentissage*

A partir de serveur de base de données nous allons procéder à la création de notre base de données qui vas stocker nos données d'apprentissage.

La figure 4.8 présente la création de base de données postgesSQL.

```
postgres=# \c postgres
SSL connection (protocol: TLSv1.3, cipher: TLS_AES_256_GCM_SHA384, bits: 256, compression: off)
You are now connected to database "postgres" as user "root".
postgres=# CREATE DATABASE datasetcsv ;
CREATE DATABASE
postgres=# \l
                                  List of databases
    Name    |  Owner   | Encoding | Collate |  Ctype  |   Access privileges
------------+----------+----------+---------+---------+-----------------------
 data1      | root     | UTF8     | C.UTF-8 | C.UTF-8 |
 datasetcsv | root     | UTF8     | C.UTF-8 | C.UTF-8 |
 postgres   | postgres | UTF8     | C.UTF-8 | C.UTF-8 |
 template0  | postgres | UTF8     | C.UTF-8 | C.UTF-8 | =c/postgres          +
            |          |          |         |         | postgres=CTc/postgres
 template1  | postgres | UTF8     | C.UTF-8 | C.UTF-8 | =c/postgres          +
            |          |          |         |         | postgres=CTc/postgres
(5 rows)

postgres=# \c datasetcsv
SSL connection (protocol: TLSv1.3, cipher: TLS_AES_256_GCM_SHA384, bits: 256, compression: off)
You are now connected to database "datasetcsv" as user "root".
datasetcsv=#
```

Figure 4.8 création de base de données

➢ *Activation de l'option réplication de données*

La réplication est une technologie de base pour tout serveur de base de données, car le temps d'arrêt ou la perte de données peut entraîner une réduction de l'accessibilité et de la productivité.

L'utilisation de la réplication de données d'un serveur principal (maître) vers un ou plusieurs serveurs de secours (esclaves) réduit le risque de perte de données, au même temps, on assure la haute disponibilité de nos serveurs. En conséquence, nous allons définir les serveurs maitres et les serveurs esclave afin de s'assurer de la continuité de service en cas de défaillance. D'autre part la fonction de réplication va nous aider à agréger les données enregistrées dans différents serveurs.

➢ *Étape 1:*

Comme première étape, nous allons accéder au fichier de configuration de Postgres Pg_hba.conf qui est responsable de l'authentification avec les clients. Nous ajoutons la plage d'adresses IP et les ports avec lesquelles nous pouvons communiquer comme présenté dans la figure 4.9.

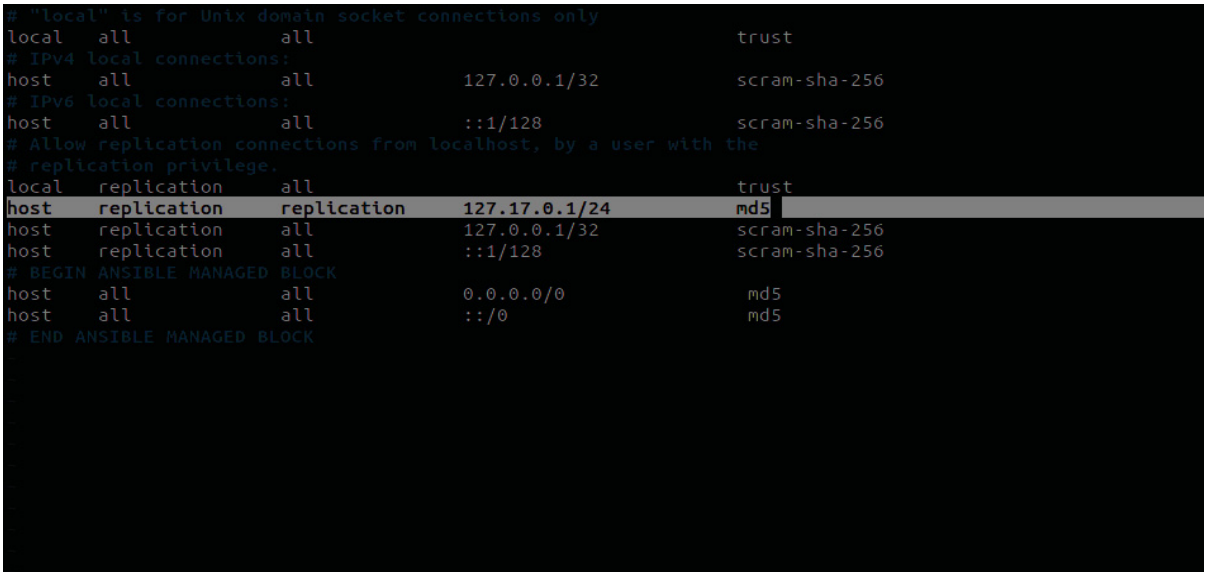

Figure 4.9 fichier Pg_hba.conf

➢ *Étape 2:*

Une fois les adresses IP d'autres serveurs étaient ajoutées, nous allons créer un rôle pour la réplication sur le serveur maitre. La figure 4.10 présente la création de rôle pour activer La réplication.

```
postgres=# CREATE USER replication LOGIN CONNECTION LIMIT 1 ENCRYPTED PASSWORD 'replication';
CREATE ROLE
postgres=#
```

Figure 4.10 création de rôle réplication

Nous pouvons ainsi visualiser toutes nos bases de données et les privilèges accordés pour chacune comme démontré dans la figure 4.11.

```
postgres=# DROP USER replication;
DROP ROLE
postgres=# \du
                             List of roles
 Role name |                         Attributes                         | Member of
-----------+------------------------------------------------------------+-----------
 postgres  | Superuser, Create role, Create DB, Replication, Bypass RLS | {}
 root      | Superuser, Create role, Create DB                          | {}

postgres=# CREATE USER replication WITH SUPERUSER PASSWORD 'replication';
CREATE ROLE
postgres=# \du
                              List of roles
  Role name  |                         Attributes                         | Member of
-------------+------------------------------------------------------------+-----------
 postgres    | Superuser, Create role, Create DB, Replication, Bypass RLS | {}
 replication | Superuser                                                  | {}
 root        | Superuser, Create role, Create DB                          | {}

postgres=#
```

Figure 4.11 visualisation des bases de données

- *Étape 3:*

Il faut apporter ensuite les modifications suivantes sur le fichier Postgresql.conf_dont nous définissons :

- Les adresses écoutées
- Wal_level : détermine la quantité d'informations écrite dans les journaux de transactions.
- Max_wal_sender : indique le nombre maximum de serveurs standby (autrement dit, le nombre maximum de processus walsender en cours d'exécution).
- Wal_keep_segment : iindiquer le nombre minimum de journaux de transactions passés à conserver dans le répertoire pg_wal, au cas où un serveur en attente a besoin de les récupérer pour la réplication en flux.
- Hot_stanby: indique si on peut se connecter et exécuter des requêtes lors de la restauration, activé par défaut. Ce paramètre peut seulement être configuré au lancement du serveur. Il a un effet uniquement lors de la restauration des archives ou en mode serveur en attente.

La figure 4.12 présente la configuration de fichier Postgresql.conf.

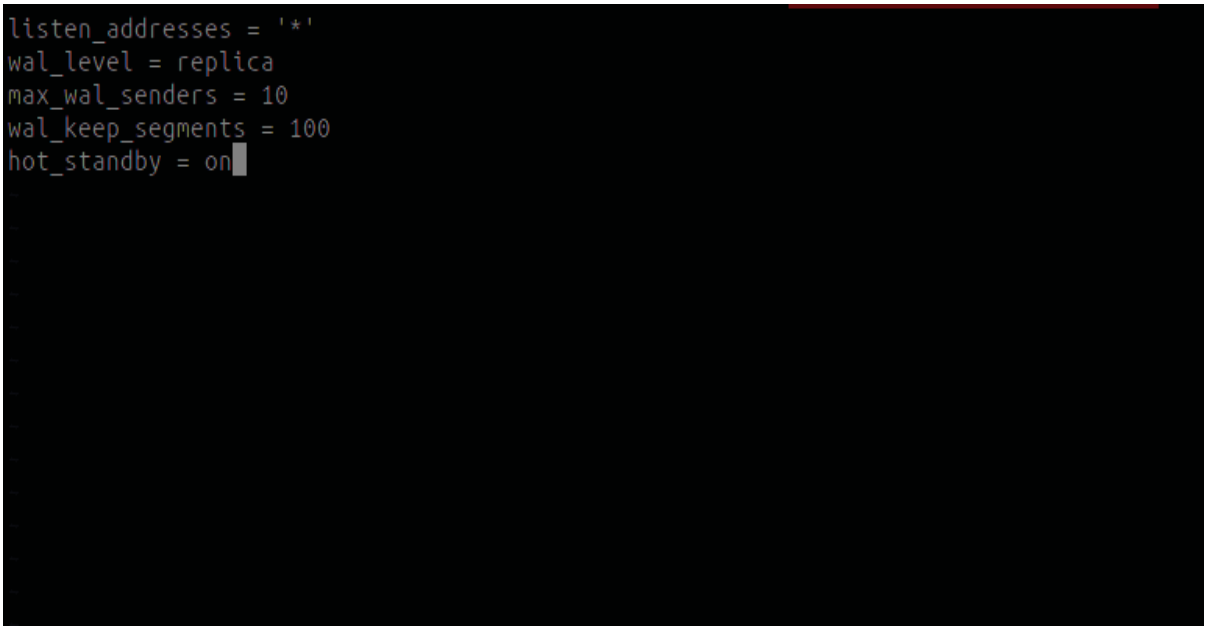

Figure 4.12 configuration de fichier Postgresql.conf

➢ *Étape 4:*

- La dernière étape sert à synchroniser les serveurs esclaves avec le serveur maitre (Figure 4.13).
- Pg_basebackup est utilisé également pour prendre une sauvegarde de base d'une instance PostgreSQl en cours d'exécution. Cette sauvegarde se fait via une connexion PostgreSQl standard et utilise le protocole de réplication. La connexion doit se faire avec un utilisateur doté de l'attribut REPLICATION ou SUPERUSER, et le fichier Pg_hba.conf que nous avons précédemment configurée, doit explicitement permettre la connexion de réplication. Le serveur doit aussi être configuré avec un max_wal_senders suffisamment élevé pour laisser au moins une connexion disponible pour la sauvegarde.

Figure 4.13 synchronisation de serveur esclave avec le serveur maitre

4.3. Agrégation de données

Dans cette partie, nous allons consolider les données présentes dans chaque serveur à partir de serveur Ansibleserver, à l'aide d'un Playbook (Figure 4.14) que nous avons développé pour effectuer cette tâche.

Pour cela, il faut que nous définissions les éléments suivants :

- les serveurs que nous voudrons solliciter.
- les base de donnes et les tables où se trouve les informations que nous désirons récupérer.
- la source et la destination des fichiers qui contient les informations sollicitées.

```
---
- hosts: psql
  become: yes
  become_method: sudo
  gather_facts: true
  ignore_errors: true
  ignore_unreachable: true

  tasks:
      - name: Select the query from Databases2

        shell: |
          psql -U root -d Database2 -c "SELECT * FROM Dataset" >> /root/Database2.csv

        args:
          executable: /bin/bash
        delegate_to: 172.17.0.4

      - name: collect les donnees depuis le serveur distant2
        run_once: yes
        fetch: src=/root/Database2.csv dest=/tmp/centraldatabase.csv flat=yes

        delegate_to: 172.17.0.4

      - name: Select the query from Databases3
        shell: |
            psql -U root -d Database3 -c "SELECT * FROM Dataset" >> /root/Database3.csv

        args:
          executable: /bin/bash
        delegate_to: 172.17.0.5
      - name: collect les donnees depuis le serveur distant3
        run_once: yes
        fetch: src=/root/Database3.csv dest=/tmp/centraldatabase1.csv flat=yes

        delegate_to: 172.17.0.5
```

Figure 4.14 Playbook de collecte de données

Ensuite, nous exécutons le playbook pour solliciter les serveurs distants (Figure 4.15).

```
root@baf4fb107f9d:/etc/ansible# ansible-playbook collection.yml

PLAY [psql] ********************************************************************************************************

TASK [Gathering Facts] *********************************************************************************************
ok: [172.17.0.5]
ok: [172.17.0.4]
ok: [172.17.0.3]

TASK [Select the query from Databases2] ****************************************************************************
changed: [172.17.0.4]
changed: [172.17.0.3 -> 172.17.0.4]
changed: [172.17.0.5 -> 172.17.0.4]

TASK [collect les donnees depuis le serveur distant2] **************************************************************
changed: [172.17.0.3 -> 172.17.0.4]

TASK [Select the query from Databases3] ****************************************************************************
changed: [172.17.0.3 -> 172.17.0.5]
changed: [172.17.0.5]
changed: [172.17.0.4 -> 172.17.0.5]

TASK [collect les donnees depuis le serveur distant3] **************************************************************
changed: [172.17.0.3 -> 172.17.0.5]

PLAY RECAP *********************************************************************************************************
172.17.0.3                 : ok=5    changed=4    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
172.17.0.4                 : ok=3    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
172.17.0.5                 : ok=3    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

Figure 4.15 exécution de Playbook

Après l'exécution de Playbook, nous pouvons voir que les données existantes sur les serveurs distants étaient copiées sur notre serveur maitre (Figure 4.16).

```
root@baf4fb107f9d:/tmp# ls
centraldatabase.csv  centraldatabase1.csv  tmp8ssowrgo  tmp_w56b45b
root@baf4fb107f9d:/tmp#
```

Figure 4.16 les bases de données colletées

4.3.1. L'aspect sécurité de l'environnement de travail

4.3.1.1. Système d'exploitation

4.3.1.2. Désactivation des fichiers de systèmes inutiles

Le système d'exploitation vient avec certains fichiers système qui ne sont pas utiles ou nécessaires. La désactivation de support de ces derniers va nous aider à minimiser les risques. Parmi ces fichiers, nous trouverons :

i. CRAMFS : c'est un fichier compressé qui vient par défaut sur le noyau linux. La suppression de ce fichier minimise l'accès au serveur par des vulnérabilités en exploitant ce dernier vu qu'il peut être lu et modifié sans décompression. Pour le désactiver, il faut créer un fichier dans le répertoire (/etc/modprobe.d/CIS.conf) et ajouter la ligne suivante :

```
install cramfs /bin/true
```

ii. FREEVXFS : ce fichier de système est primaire pour le type HP-UX OS. La suppression de ce type minimise la surface d'attaque s'il n'est plus nécessaire sur notre application. Pour le désactiver, il faut créer un fichier dans le répertoire (/etc/modprobe.d/CIS.conf) et ajouter la ligne suivante :

```
install freevxfs /bin/true
```

iii. JFFS2 : Un type de FS utilisé dans les équipements FLASH memory, pour le désactiver, il faut créer un fichier dans le répertoire (/etc/modprobe.d/CIS.conf) et ajouter la ligne suivante :

```
install jffs2 /bin/true
```

iv. HFS : Un type de fichier de système dédié pour les systèmes d'exploitation MAC-OS et qui sera inutile sur notre cas vu qu'il existe par défaut dans notre système linux. Pour le désactiver, nous devons ajouter la ligne suivante dans le répertoire (/etc/modprobe.d/CIS.conf).

```
install hfs /bin/true
```

v. AUTOFS : AutoFS permet le montage automatique de périphériques, y compris généralement des CD/DVD et des clés USB. Lorsque le montage automatique est activé, toute personne disposant d'un accès physique peut connecter une clé USB ou un disque et avoir son contenu disponible dans le système même si elle n'a pas les autorisations nécessaires. Pour le désactiver, il faut lancer la commande suivante :

```
systemctl --now disable autofs && apt purge -y autofs
```

4.3.2. Sécurité SSH

La sécurité d'accès est un élément vital et obligatoire afin de minimiser tous les trous d'accès et minimiser toute option nécessaire pour avoir un taux de risque qui est proche au zéro.

1. Accès Limité : Pour limiter l'accès au serveur, nous pouvons ajouter les deux options AllowUsers/AllowGroups afin d'autoriser uniquement la liste d'utilisateurs qui peuvent se connecter sur notre serveur en utilisant le protocole SSH sur le fichier /etc/ssh/sshd_config**.**
2. Configuration au niveau des événements SSH (LOG) : pour toujours garder les évènements et pour avoir une visibilité sur ce qui se passe sur la partie accès, la définition de niveau de log est importante. Nous pouvons la configurer sur la partie (/etc/sshd/sshd_config) avec l'option Loglevel info**.**
3. Maximum d'essai d'authentifications : Le paramètre MaxAuthTries spécifie le nombre maximal de tentatives d'authentification autorisées par connexion pour répondre aux

attaques de BRUTE FORCE. Pour configurer cette partie, il faut toujours se positionner sur le fichier (/etc/ssh/sshd_config) et définir l'option MaxAuthTries**.**

4. Désactiver l'accès ROOT : Pour interdire les connexions root via SSH, les administrateurs système doivent s'authentifier à l'aide de leur propre compte individuel afin de collecter tous les évènements qui ont eu lieu sur la partie système, pour cela et pour fournir une piste d'audit clair en cas d'incident de sécurité, l'accès ROOT est désactivé à distance via l'option PermitRootLogin no sur le fichier (/etc/ssh/sshd_config).
5. Strong Ciphers : les clés de cryptage sont utilisées pour fournir un niveau de confidentialité importante sur la transaction de donnée, pour cela nous devons utiliser les forts Ciphers sur le fichier (/etc/ssh/sshd_config**)** comme suivant :

```
Ciphers chacha20-poly1305@openssh.com,aes256-gcm@openssh.com,aes128-gcm@openssh.com,aes256-ctr,aes192-ctr,aes128-ctr
```

4.3.3. Réseau

4.3.3.1. Firewalls

Dans le but d'assurer la sécurité pour les bases de données, il faut dans un premier temps configurer la partie pare-feu le plus approprié afin de minimiser le risque et assurer un niveau de sécurité important.

D'abord nous devons autoriser la communication entre le serveur Postgres et toutes les autres parties à travers une commande IPTABLES :

```
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
```

Ensuite, et pour les communications qui traverse le pare-feu, il faut ajouter les deux règles qui autorisent les services SSH et PostgreSQL.

```
iptables -A INPUT -p tcp -m state --state NEW --dport 22 -j ACCEPT
iptables -A INPUT -p tcp -m state --state NEW --dport 5432 -j ACCEPT
```

Il faut bloquer toutes les autres connexions une fois avoir établi les connexion autorisées.

```
iptables -A OUTPUT -j ACCEPT
iptables -A INPUT -j DROP
iptables -A FORWARD -j DROP
```

Enfin, pour avoir un état de visibilité sur tous les évènements, nous avons configuré les fichier stderr, csvlog et syslog.

```
postgres=# alter system set log_destination = 'csvlog';
ALTER SYSTEM
postgres=# select pg_reload_conf();
```

➢ *Assurer que le répertoire de log existe :*

Le paramètre log_directory spécifie le répertoire de destination des fichiers journaux

```
postgres=# alter system set
log_directory='/var/log/postgres';
ALTER SYSTEM
postgres=# select pg_reload_conf()
```

➢ *Assurer que les permissions sont établies pour les fichiers de log*

Le paramètre log_file_mode détermine les autorisations de fichier pour les fichiers journaux lorsque logging_collector est activé. Les autorisations doivent être définies pour autoriser uniquement l'accès nécessaire au personnel autorisé. Dans la plupart des cas, le meilleur réglage est 0600.

```
postgres=# alter system set log_file_mode = '0600';
ALTER SYSTEM
postgres=# select pg_reload_conf();
```

➢ *Activer l'option « Log_connection » sur Postgresql*

L'activation du paramètre log_connections entraîne l'échec de chaque tentative de connexion au serveur. PostgreSQL ne conserve pas les enregistrements internes des tentatives de connexion à la base de données pour un audit ultérieur. Ce n'est qu'en activant la journalisation de ces tentatives que l'on peut déterminer si des tentatives étaient faites. La configuration suivante illustre comment activer cette option sur postgresql :

```
postgres=# alter system set log_connections = 'on';
ALTER SYSTEM
postgres=# select pg_reload_conf();
```

➢ *Activation de l'option « log_hostname » sur Postgresql*

L'activation du paramètre log_hostname donne comme résultat la journalisation du nom de l'hôte qui se connecte en plus de son adresse IP. Pour le configurer il faut lancer les deux commandes suivantes :

```
postgres=# alter system set log_hostname='on';
ALTER SYSTEM
postgres=# select pg_reload_conf();
pg_reload_conf
```

4.4 Conclusion

Dans ce chapitre nous avons préparé la plateforme pour gérer et stocker les données, en installant les serveurs PostgreSQL, ainsi nous avons évoqué l'aspect sécurité de notre plateforme.

Dans le chapitre suivant nous allons procéder au traitement de données collectées et comme deuxième étape, les données que nous avons pu collectées seront utilisées pour l'entrainement de notre modèle intelligent.

CHAPITRE 5

ENTRAINEMENT DU MODELE PREDICTIF

5.1. Introduction

L'accès et le traitement d'un grand volume de données de santé sont bénéfiques pour les professionnels de la santé, car cela leur permet de mieux comprendre les cas et les symptômes de leurs patients, et ainsi de fournir un meilleur retour d'information, des conseils et un suivi adapté.

Le chapitre précédent a été consacré à la mise en place d'une plateforme permettant la gestion de ces données de manière automatisée et rapide. Dans ce chapitre, les données collectées seront utilisées pour entraîner un réseau de neurones à détecter précocement différentes pathologies. Nous effectuons également les prétraitements de ces données avant d'alimenter le réseau de neurones.

5.2. Introduction sur les données utilisées

Les caractéristiques acoustiques des cris sont une exposition de l'état de santé d'un nourrisson et ces caractéristiques ont été reconnues comme des indicateurs de diverses pathologies. Néanmoins, les motifs de pleurs n'étaient pas les mêmes pour tous les nourrissons ; par exemple, les pleurs peuvent être dus à des couches mouillées, à la faim, à la peur, etc. Ces raisons ont été déterminées en fonction des conditions à l'origine du cri avec l'aide du personnel médical et des tuteurs du nourrisson. Ils étaient également basés sur les différents tests pratiqués après la naissance (Khalilzad, Kheddache & Tadj, 2022).

Dans notre étude nous allons utiliser une base de données provenant d'un projet sur le diagnostic des pleurs du nouveau-né. Cette base de données se composent de 633 échantillons, dont on trouve 317 échantillons de cas sains et 316 échantillons de cas avec des pathologies.

Le but de ce travail est d'exploiter notre infrastructure automatisée pour gérer des données médicales réelles. Comme une preuve de concept, nous allons utiliser ces données pour entrainer des modèles prédictifs en utilisant les moyens classiques, où chaque modèle va utiliser une base de données Psql contenant une portion de la base de données totale, puis nous allon nous servir de notre serveur Ansible afin de lancer des requêtes Playbook. Ces requêtes vont solliciter toutes les données enregistrées dans les différents serveurs Psql afin d'entrainer le modèle prédictif.

Dernièrement, nous allons évaluer et comparer les résultats obtenus à partir de chaque modèle.

5.3. Traitement de données

Dans un premier temps, nous allons charger les données sur Google collab afin de commencer leur traitement.

La figure 5.1 nous donne un aperçu sur les étapes suivies durant l'entraînement du modèle prédictif.

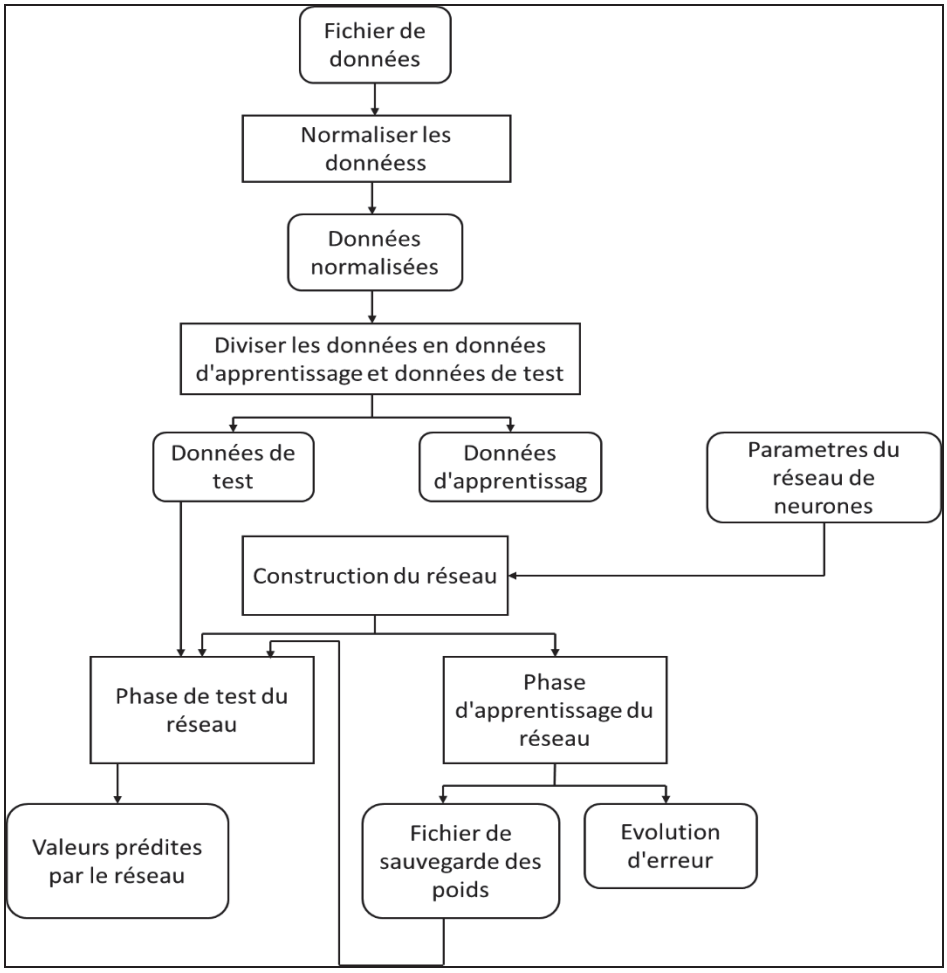

Figure 5.1 Les étapes d'entraînement d'un modèle prédictif

Ensuite, nous allons effectuer une modification sur les étiquettes de classe dont tous les enfants sains vont avoir l'étiquette 1 et les enfants qui ont des pathologies vont avoir l'étiquette 0 (Figure 5.2).

```
print(data[0:100,:])

[[ 1.      -1.6338   0.44991 ... -0.66329 -0.74595 -1.0904 ]
 [ 1.      -1.3714   0.99388 ...  0.2877  -0.92997 -1.5562 ]
 [ 1.       1.0511  -1.2667  ... -0.7194  -1.0529  -1.2046 ]
 ...
 [ 0.      -0.5565  -0.005121 ...  0.82342  1.0848   1.8242 ]
 [ 0.       0.21251 -0.1951  ...  1.5956   1.7331   1.3713 ]
 [ 0.      -0.23784 -0.50569 ... -1.246   -0.69005 -0.92011 ]]
```

Figure 5.2 visualisation de la base de données

Afin de garantir la fiabilité et pour éviter d'avoir toutes les données positives (échantillons représentant les enfants sains) dans la partie consacrée à l'entrainement de réseau de neurones, nous devons mélanger les ligne de cette base de données. Ce brassage des données est une tâche très utile pour créer des ensembles d'entrainement et de test plus représentatifs.

Une fois le brassage de données est fait, nous procédons à la division de ces données en deux parties, les données d'entrainement et les données de validation.

Dans ce modèle, nous allons utiliser 20 % de nos données pour la validation et le reste pour l'entrainement de modèle.

5.4. Normalisation de données d'entrée

L'objectif de la normalisation est de changer les valeurs des colonnes numériques dans l'ensemble de données à une échelle commune, sans déformer les différences dans les plages de valeurs (Figure 5.3).

Prenons une colonne aléatoire de notre base de données, dans notre cas, nous avons choisi la colonne 260.

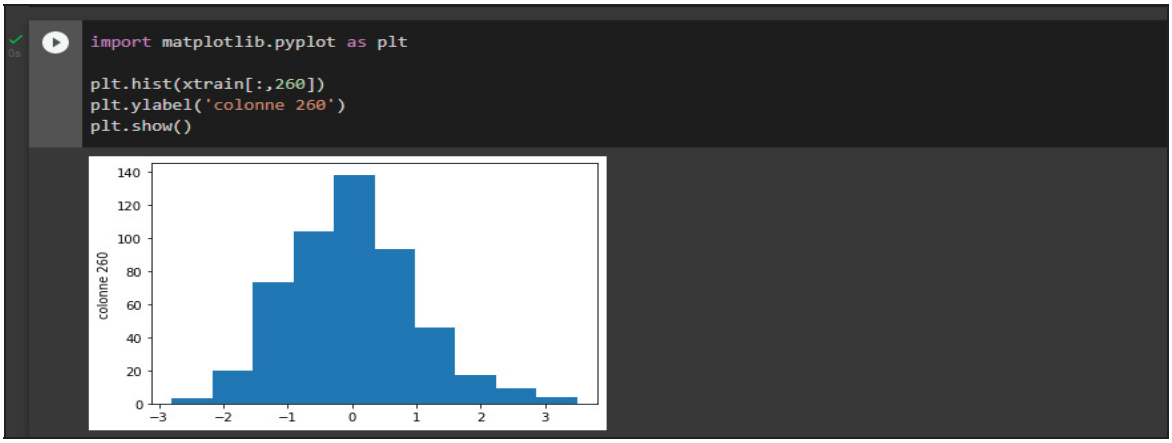

Figure 5.3 normalisation de données

Ce graphe démontre que les valeurs de la colonne 260 sont comprises entre -3 et 4 d'où nous recourons à la normalisation afin de stabiliser ces valeurs entre 1 et -1.

Pour cela, nous allons utiliser la technique de standardisation et nous obtenons les deux paramètres : la moyenne et l'écart-type à partir de nos données d'entrainement.

Nous allons ensuite vérifier la distribution de nos sorties (Figure 5.4).

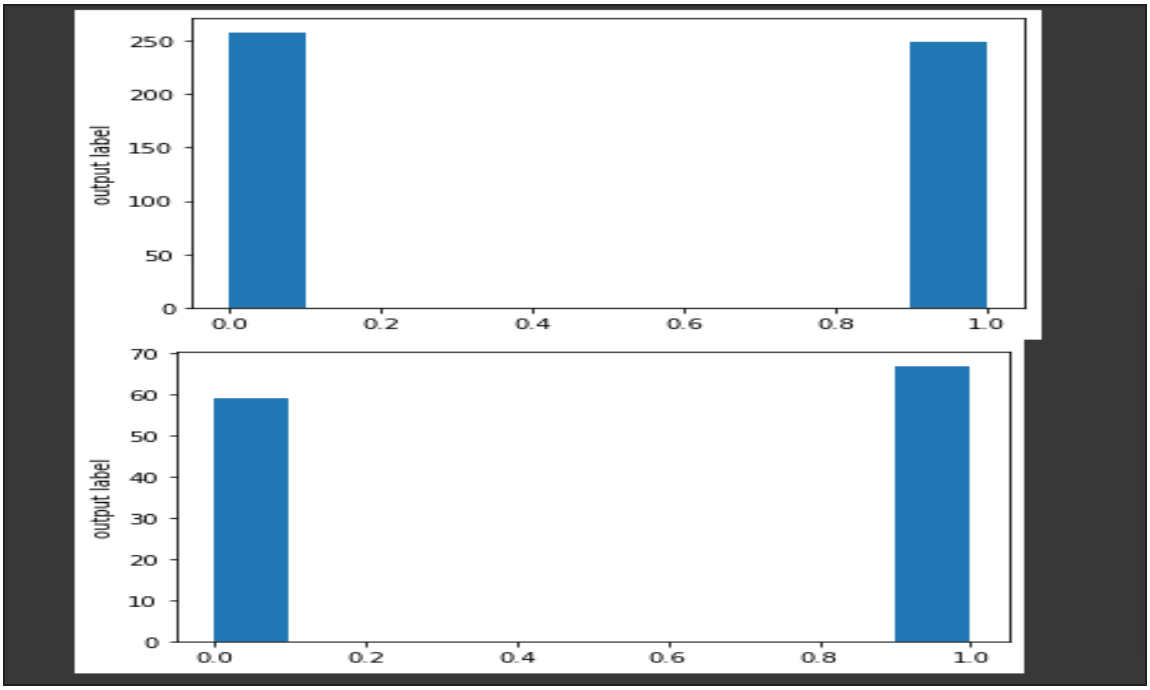

Figure 5.4 Visualisation de la distributions des cas sains et malades

5.5. Création d'un modèle de réseau de neurones

Dans cette partie, nous allons procéder à la création de modèle de réseau de neurones (Figure 5.5).

Le modèle est construit à l'aide de la classe Sequential de Keras, ce qui signifie que chaque couche est ajoutée au modèle dans l'ordre où elle est spécifiée.

La première couche est une couche dense (complètement connectée) avec 8 neurones. Elle prend en entrée un tenseur de dimensions (batch_size, input_dim), où input_dim est le nombre de fonctionnalités (features) dans les données d'entrée. L'activation de cette couche est définie comme étant ReLU (fonction d'activation rectifiée linéairement).

La deuxième couche est également une couche dense avec 4 neurones. Elle prend en entrée les sorties de la première couche et applique la fonction d'activation ReLU.

La dernière couche est une couche dense avec 1 neurone, qui prend en entrée les sorties de la deuxième couche et applique la fonction d'activation sigmoïde. Cette couche produit une sortie de dimension (batch_size, 1) qui représente la prédiction de la probabilité d'appartenance à la classe positive pour chaque exemple d'entrée.

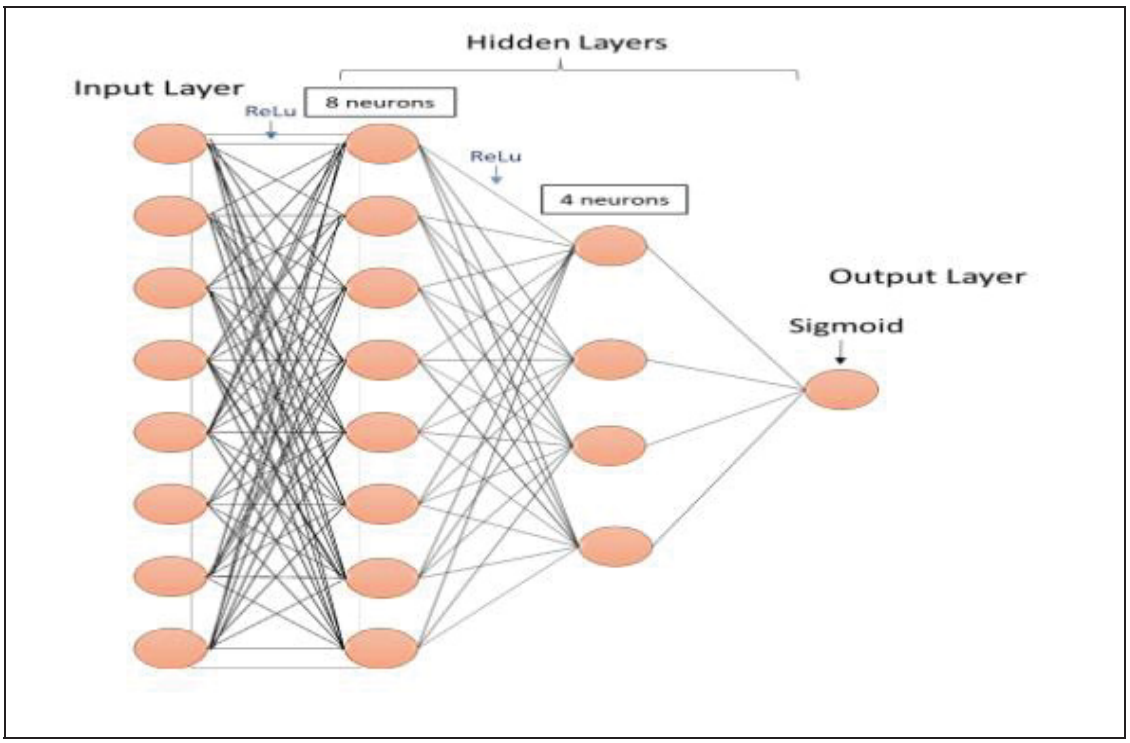

Figure 5.5 Architecture de réseau de neurones

5.6. La compilation de modèle :

Dans cette étape, nous allons définir les trois fonctions suivantes :

- La fonction de perte : La fonction "binary_crossentropy" est une fonction de coût souvent utilisée en apprentissage automatique pour les tâches de classification binaire. Elle est également connue sous le nom de "perte d'entropie croisée binaire" ou "logarithmic loss". Cette fonction calcule la distance entre les prédictions du modèle et les valeurs réelles pour

chaque exemple dans un ensemble de données d'apprentissage. Elle est souvent utilisée en conjonction avec une fonction d'activation sigmoid, qui produit une valeur de sortie entre 0 et 1, pour prédire la probabilité que chaque exemple soit positif.

La fonction "binary_crossentropy" est définie comme suit :

$$binary\,crossentropy = -\frac{1}{N}\sum_{i=1}^{N} yi.\log\big(p(yi)\big) + (1-yi)\cdot\log\big(1-p(yi)\big) \quad (5.1)$$

Où (N) est le nombre d'exemples dans l'ensemble de données, (y_i) est la valeur réelle (0 ou 1) pour l'exemple (i), et (py_i) est la prédiction du modèle pour l'exemple (i).

- Adam (Adaptive Moment Estimation) un algorithme d'optimisation couramment utilisé en apprentissage automatique pour entraîner des réseaux de neurones profonds. C'est une variante de la descente de gradient stochastique (SGD) qui utilise des taux d'apprentissage adaptatifs et de l'élan pour converger plus rapidement et atteindre de meilleures performances.

  Adam calcule des taux d'apprentissage individuels pour chaque paramètre en combinant des informations provenant des premiers et seconds moments des gradients. Le premier moment est la moyenne du gradient, tandis que le deuxième moment est la variance du gradient. Ces moments sont estimés en utilisant des moyennes mobiles pondérées exponentiellement du gradient et de son carré. Le taux d'apprentissage de chaque paramètre est ensuite mis à jour en fonction de ces estimations.

- La fonction métrique : métrique (accuracy) sert à évaluer la performance de notre modèle de classification, en mesurant le taux de prédictions correctes du modèle par rapport au nombre total de prédictions effectuées.

$$accuracy = \frac{(TP+TN)}{(TP+TN+FP+FN)} \quad (5.2)$$

5.7. Performance sur les données d'apprentissage et de validation

Après avoir développé notre modèle prédictif, nous avons procéder à l'étape d'entrainement avec trois bases de données Psql dont chacune contient 200 échantillons.

Tableau 5.1 répartition d'échantillons dans chaque base de données

Base de données	Echantillons de cas sain	Echantillons de cas avec pathologie
Bdd1	111	89
Bdd2	104	96
Bdd3	115	85

Nous pouvons voir les résultats obtenus dans la Figure 5.6 en utilisant la métrique d'exactitude (accuracy), qui est une mesure courante utilisée en apprentissage automatique et en statistiques pour évaluer les performances d'un modèle de classification. L'exactitude représente la proportion de prédictions correctes du modèle par rapport au nombre total de prédictions effectuées.

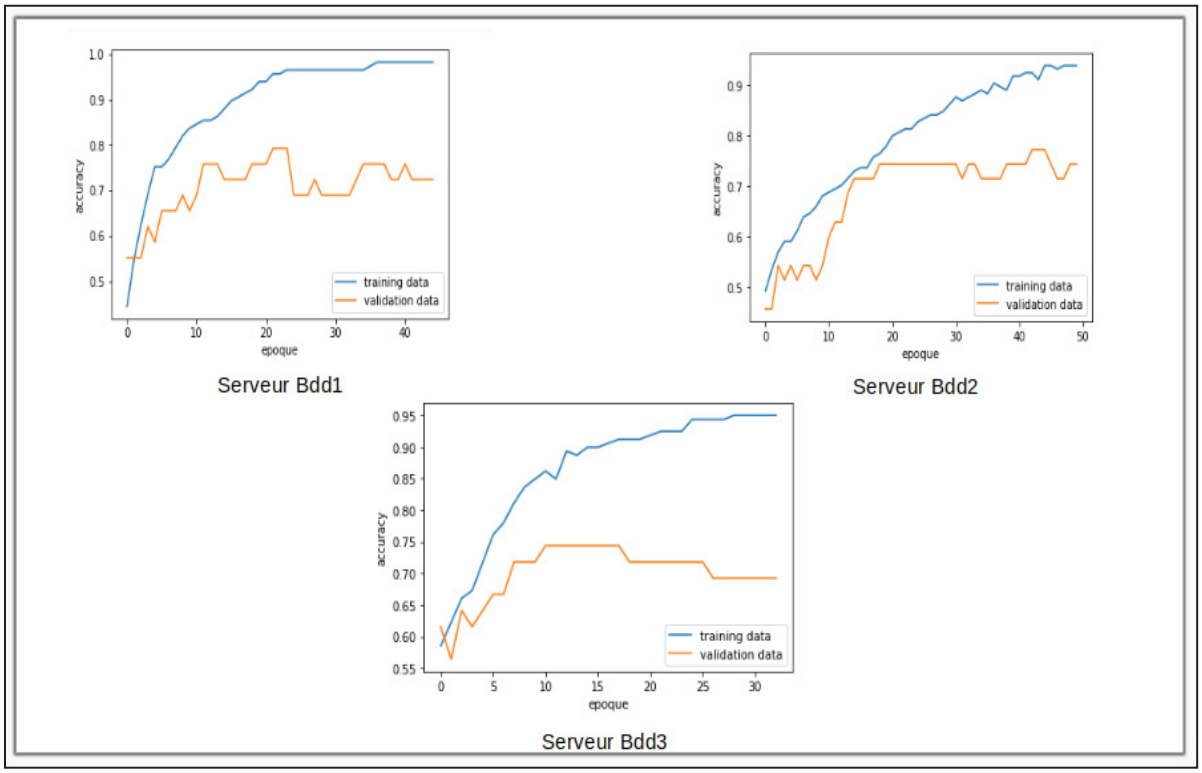

Figure 5.6 Visualisation de processus d'apprentissage et de validation

Les graphiques présentés dans la figure 5.6 montrent que les modèles ont de bonnes performances sur les données de test, mais ils ne sont pas satisfaisants sur les données de validation. Les résultats détaillés sont présentés dans le tableau 5.1. Ensuite, le tableau 5.2 présente les performances des modèles sur les données d'apprentissage et de validation pour trois serveurs différents (Bdd1, Bdd2 et Bdd3). On peut constater que les modèles ont des performances élevées sur les données d'apprentissage pour tous les serveurs, mais les performances sont moins bonnes sur les données de validation.

Tableau 5.2 Performance des modèles en terme d'exactidude (accuracy)

	Performance sur les données d'apprentissage	Performance sur les données de validation
Serveur Bdd1	96.58 %	68.97 %
Serveur Bdd2	86.81 %	74.29 %
Serveur Bdd3	89.31 %	74.36 %

Nous allons par la suite exécuter le Playbook ansible (Figure 4.20) pour solliciter tous les serveurs PostgreSQL et récupérer toutes les données enregistrées dans ces serveurs, puis nous allons entrainer à nouveau le modèle prédictif avec les données collectées et nous observons les résultats obtenus avec la mesure d'exactitude (Figure 5.7).

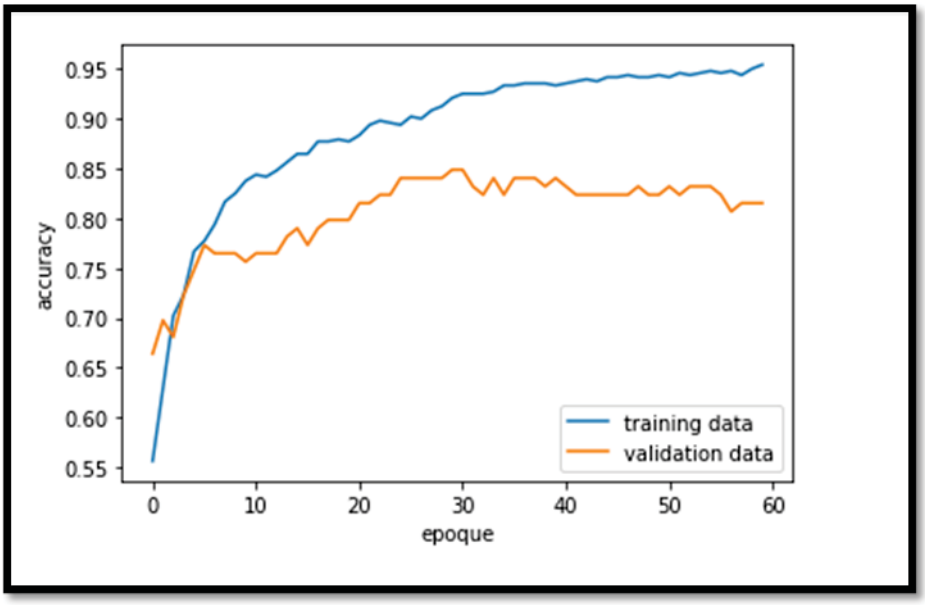

Figure 5.7 Performance de modèle avec toutes les données collectées

Selon la courbe d'apprentissage, le modèle a gagné plus en termes d'exactitude (accuracy) sur les données de validation par rapport aux modèles précédents, dont nous pouvons

expliquer ça à la présence de plus de donnes d'entrainement et de validation et une meilleure balance entre les échantillons des cas sain et cas avec pathologie.

Tableau 5.3 Performance du modèle avec toutes les données collectées

	Performance sur les données d'apprentissage	Performance sur les données de validation
Serveur Ansible	93.54 %	84.03 %

Pour une meilleure comparaison entre les résultats obtenus à partir des trois serveurs PostgreSQL et le serveur master (Ansible), nous avons pris d'autres mesures (métrique) autre que l'exactitude (accuracy) qui présente cependant de fortes limites en présence de données déséquilibrées (imbalanced data). Ce qui est le cas dans nos trois bases de données Psql avec la différence dans le nombre d'échantillons de cas sain et cas malade. Pour cela, nous faisons recours aux métriques de précision et de rappel (recall) qui se basent sur la matrice de confusion (Figure 5.8).

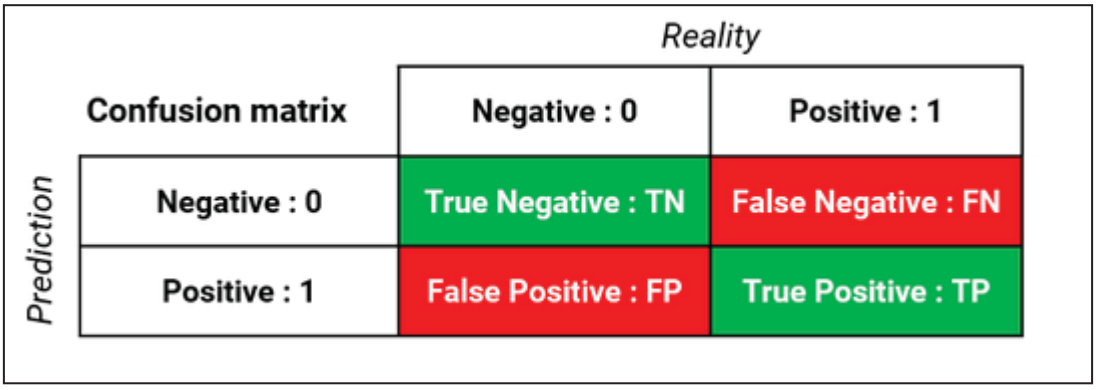

Figure 5.8 Matrice de confusion

La précision et le rappel (recall) sont deux métriques qui se concentrent sur la performance du modèle concernant les individus positifs :

La précision est également appelée (Positive Predictive Value). Elle correspond au taux de prédictions correctes parmi les prédictions positives :

$$Precision = \frac{tp}{tp+fp} \quad (5.3)$$

Elle mesure la capacité du modèle à ne pas faire d'erreur lors d'une prédiction positive.

Le recall est également appelé sensibilité (sensitivity), (true positive rate) ou encore taux de détection (hit rate). Il correspond au taux d'individus positifs détectés par le modèle :

$$Recall = \frac{tp}{tp+fn} \quad (5.4)$$

Il mesure la capacité du modèle à détecter l'ensemble des individus positifs.

Tableau 5.4 Les résultats de précision et recall des modèles prédictifs

	La précision	Recall
Bdd1	53.85 %	70.00 %
Bdd2	83.33%	71.43%
Bdd3	73.33%	68.75%
Bdd master	86.57%	85.29%

Le modèle entraîné sur la base de données master a obtenu la meilleure précision de tous les modèles avec un taux de 86,57%.

De plus, la base de données master a également permis d'obtenir le meilleur rappel, avec un taux de 85,29%. Le rappel mesure la proportion de cas positifs réellement identifiés parmi tous les cas positifs présents dans les données. Cela signifie que le modèle a correctement identifié 85,29% de tous les cas positifs de la base de données master.

Ces résultats montrent l'importance de la qualité des données d'entraînement pour obtenir de bonnes performances de modélisation. Dans ce cas, la base de données master a permis d'obtenir les meilleurs résultats pour les deux mesures de performance clés.

5.8. Discussion

Dans ce travail, nous avons mis en évidence l'importance de développer une infrastructure automatisée pour gérer les données médicales dans des situations d'urgence telles que les pandémies. Nous avons donc choisi l'outil d'automatisation Ansible pour gérer les serveurs de bases de données PostgreSQL à distance. Cette solution nous permet de déployer rapidement des mesures de défense pour protéger les données médicales sensibles et de mettre à jour les serveurs en temps réel pour garantir leur sécurité.

En outre, nous avons souligné l'importance de la sécurité de nos serveurs de bases de données étant donné la sensibilité et l'importance des données médicales qu'ils stockent. Nous avons donc pris toutes les mesures nécessaires pour garantir leur sécurité, notamment en utilisant des protocoles de cryptage et de sécurité avancés.

En fin, pour démontrer l’intérêt de cette infrastructure qui peut servir dans plusieurs applications, telles que le recensement et l’analyse de données, nous avons exploité la capacité de notre infrastructure dans la collecte de donnée afin d’entrainer des modèles prédictifs en utilisant des données médicales réelles.

Selon les résultats que nous avons obtenus après l’entrainement de trois modèles avec des bases de données contenant 200 échantillons dans chacune, et un modèle entrainé après la consolidation de données à l’aide de serveurs Ansible, nous pouvons remarquer que le modèle atteint de meilleurs résultats en termes d’exactitude (accuracy) 84.03%, de précision 86.57% et dans le test de rappel (recall) 85.29%.

5.9. Conclusion

Nous avons consacré ce chapitre à l'utilisation de nos données médicales pour entraîner des modèles prédictifs. Nous avons tout d'abord appliqué un prétraitement sur nos données, ensuite, nous avons créé les modèles et nous les avons entraîner avec des données enregistrées dans trois bases de données Psql. Enfin, nous nous sommes focalisés sur l'évaluation des résultats obtenus à partir de chaque modèle.

CHAPITRE 6

CONCLUSION ET RECOMMANDATION

6.1. Conclusion

Notre étude a porté sur les épidémies et leur mode de propagation. Nous avons cherché à mettre en évidence l'importance cruciale de collecter le maximum d'informations pour mieux combattre ce type de maladies.

La solution que nous avons proposée se représente dans une infrastructure automatisée qui assure la collecte rapide des données de différents serveurs en assurant un environnement fortement sécurisé.

Notre premier objectif a été atteint après l'automatisation des processus de création des bases de données et la gestion des accès pour les utilisateurs.

Nous avons ensuite pu atteindre les deux objectifs suivants en nous focalisant sur la collecte de données et l'implémentation des techniques agissants sur la haute disponibilité des serveurs de bases de données, tout en renforçant l'aspect sécurité de notre environnement.

Le dernier volet de notre projet a été consacré à l'implémentation et à l'évaluation des modèles prédictifs où nous avons obtenu des précisions de 93.57% avec les données d'entrainement et 84.03% avec les données de validation après l'utilisation de serveur Ansible qui nous a permis d'assurer le maximum nombre de données possible en consolidant les données de plusieurs serveurs.

Nous avons pour ambition de continuer à améliorer notre modèle actuel en augmentant sa précision et en réduisant son temps d'apprentissage, tout en maintenant un haut niveau de précision. Pour y parvenir, nous prévoyons d'agrandir et d'améliorer le jeu de données d'entraînement, d'utiliser des techniques d'apprentissage par transfert et d'apprentissage actif pour améliorer les performances du modèle. De plus, nous prévoyons d'exploiter les plateformes Cloud pour rendre notre infrastructure plus interopérable avec les technologies existantes.

En ce qui concerne l'exploitation des plateformes Cloud, nous pouvons utiliser des services tels que Amazon Web Services (AWS) ou Google Cloud Platform (GCP) pour héberger notre infrastructure de collecte de données et notre modèle de machine learning. Cela nous permettra de bénéficier de la puissance de calcul et de stockage des serveurs Cloud, ainsi que d'une plus grande flexibilité dans le déploiement et la gestion de notre infrastructure.

ANNEXE I

INSTALLATION DE ANSIBLE

L'installation d'Ansible est facile. Cependant, l'installation de la bonne version de Ansible n'est pas tout à fait aussi simple.

Ansible est disponible dans les dépôts de paquets pour les systèmes d'exploitation tels que Ubuntu (apt-get) et Fedora (yum), mais les versions de ces dépôts sont généralement obsolètes. Comme Ansible est en développement rapide, nous souhaitons que la dernière version soit disponible, soit en la construisant nous-même ou en l'installant via un gestionnaire de paquets.

La meilleure façon de s'assurer que nous obtenons toujours la dernière version disponible est de saisir le code source de https://github.com/ansible/ansible.

Une fois toutes les dépendances requises installées, nous pouvons télécharger Ansible de Github et le construire :

- git clone git://github.com/ansible/ansible.git --recursive
- cd ansible
- make
- sudo make install

Cela permettra de construire et d'installer la dernière version de développement de Ansible à partir de la source.

Préparation de l'environnement de travail

Nous allons suivre les étapes suivantes afin de déployer notre environnement de travail.

1. Suppression de tous les fichiers Docker en cours d'exécution sur le système à l'aide de La commande suivante :
   # Apt-get remove docker docker-engine docker.io
2. Mettre à jour le système avec la commande suivante :
   # Apt-get update
3. Installation de Docker à l'aide de la commande suivante :
   # Apt install docker.io
4. Installation de tous les packages de dépendance avec la commande suivante :
   # Snap install docker
5. Avant de tester Docker, nous vérifions la version installée (Figure I-1) avec la commande suivante :
   # docker –version

```
yogo@yogo-Aspire-E5-772G: $ docker --version
Docker version 20.10.12, build e91ed57
yogo@yogo-Aspire-E5-772G: $
```

Figure-A I-1 la version de Docker

L'étape suivante sert à créer les conteneurs et l'installation des images linux.
D'abord nous téléchargeons l'images Ubuntu sur docker-engine (Figure I-2).

```
yogo@yogo-Aspire-E5-772G: $ sudo docker pull ubuntu
[sudo] password for yogo:
Using default tag: latest
latest: Pulling from library/ubuntu
405f018f9d1d: Pull complete
Digest: sha256:b6b83d3c331794420340093eb706a6f152d9c1fa51b262d9bf34594887c2c7ac
Status: Downloaded newer image for ubuntu:latest
docker.io/library/ubuntu:latest
yogo@yogo-Aspire-E5-772G: $
```

Figure-A I-2 téléchargement de l'image Ubuntu

Ensuite, nous allons créer des conteneurs avec les images Ubuntu téléchargées et nous attribuons les noms pour chaque conteneur (Figure I-3).

```
yogo@yogo-Aspire-E5-772G: $ sudo docker run -d -t --name ansibleserver0 ubuntu
c45c8eb15f98eab619c047640b4cb18877a53d3bed7361450b3b4900d05736f9
yogo@yogo-Aspire-E5-772G: $
```

Figure-A I-3 création de conteneur

Dans cette étape, nous devons nous assurer que le serveur master pourra établir des connexions SSH avec les serveurs esclave. Pour cela, il faut activer les services SSH sur tous les serveurs.
La figure suivante présente comment activer les services SSH (Figure I-4).

```
root@9fa033f24f8d:/# apt-get install openssh-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
openssh-server is already the newest version (1:8.2p1-4ubuntu0.3).
openssh-server set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@9fa033f24f8d:/# service ssh restart
 * Restarting OpenBSD Secure Shell server sshd
root@9fa033f24f8d:/# service ssh status
 * sshd is running
root@9fa033f24f8d:/#
```

Figure-A I-4 Installation de serveur SSH

Il faut ensuite générer une clé SSH sur le serveur master et la faire copier sur les serveurs distants comme c'est présenté dans la figure I-5.

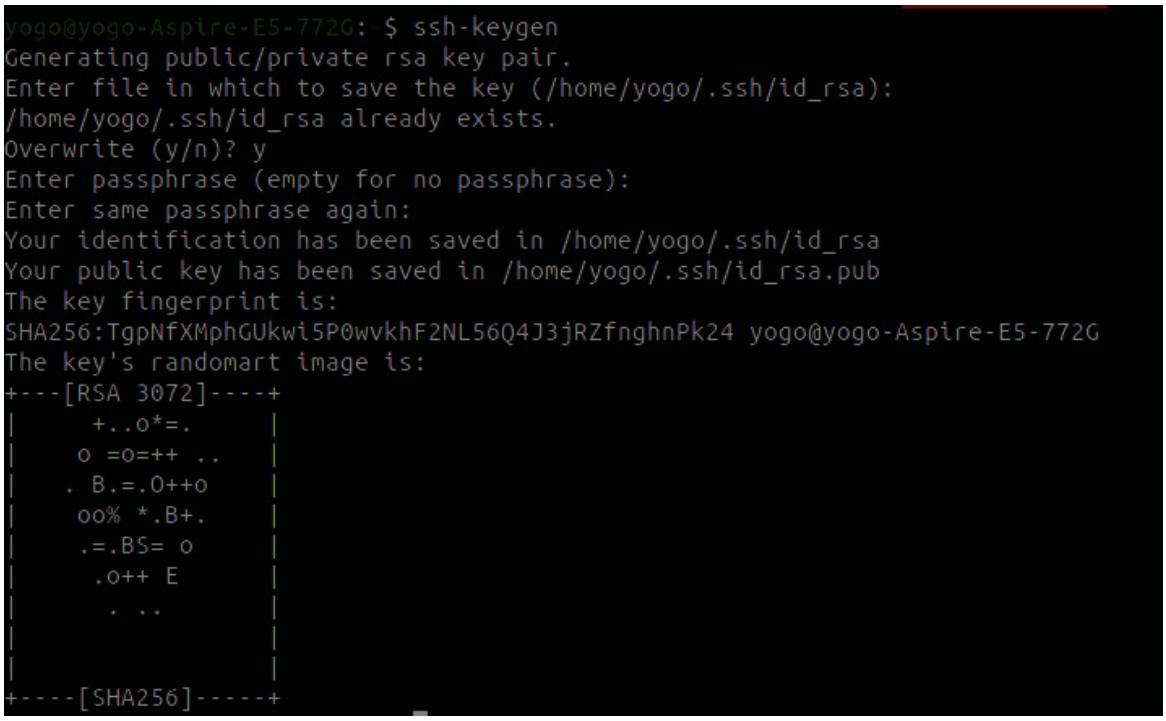

Figure-A I-5 création de clé publique

Une fois la clé créée, nous allons copier cette dernière aux serveurs distant à l'aide de leurs adresses IP, dont la figure suivante montre les étapes suivies pour copier cette clé (figure I-6). À la fin de ces étapes, le serveur master va pouvoir lancer le Playbook Ansible sur les serveurs esclaves.

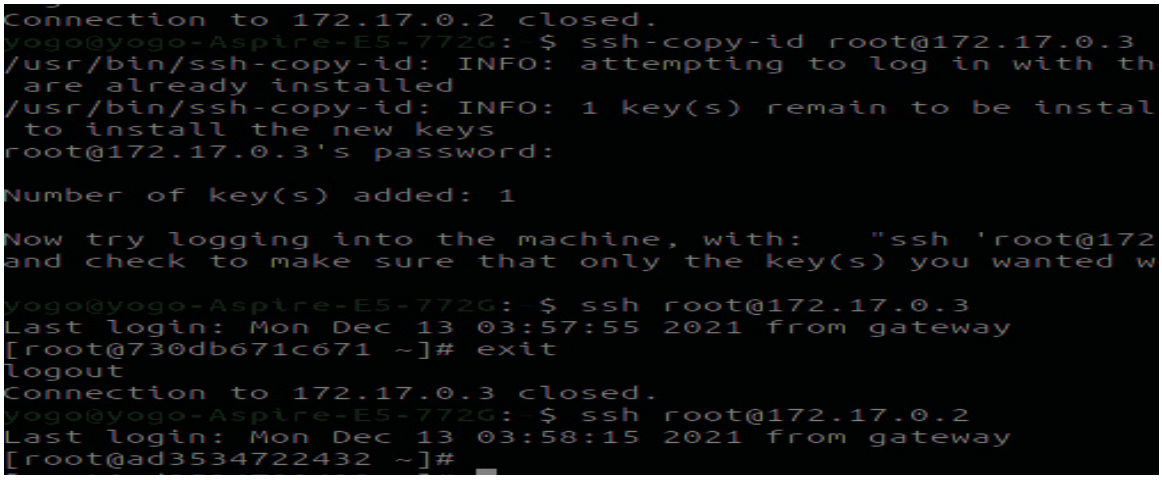

Figure-A I-6 transfert de la clé publique

Installation de Ansible et exécution de Playbook

Ansible est un outil agentless, c'est-à-dire qu'il n'installe pas d'agent sur les nœuds. Il travaille donc en mode push : il pousse les installations sur les nœuds. Pour cela nous allons installer Ansible seulement sur notre machine master comme c'est démontré dans la figure I-7.

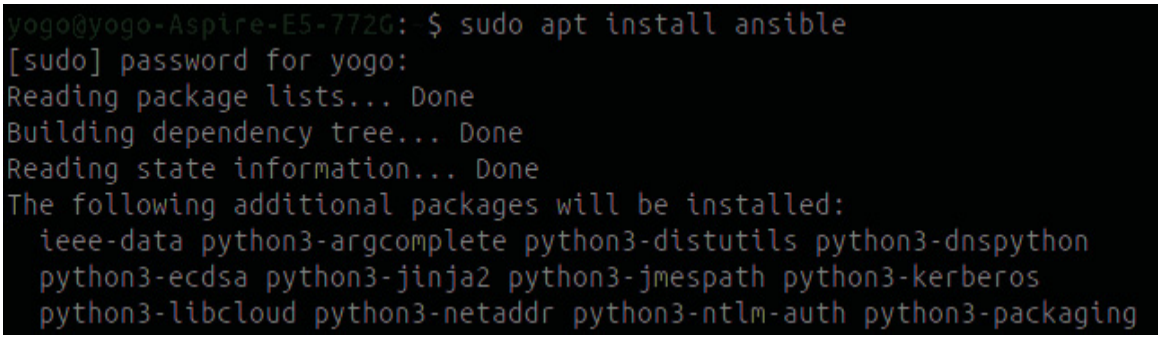

Figure-A I-7 installation de Ansible

LISTE DE RÉFÉRENCES

Abdulkareem, M., & Petersen, S. E. (2021). The Promise of AI in Detection, Diagnosis, and Epidemiology for Combating COVID-19 : Beyond the Hype. Frontiers in Artificial Intelligence, 4. https://www.frontiersin.org/articles/10.3389/frai.2021.652669

Abou-Abbas, L., Tadj, C., & Fersaie, H. A. (2017). A fully automated approach for baby cry signal segmentation and boundary detection of expiratory and inspiratory episodes. The Journal of the Acoustical Society of America, 142(3), 1318-1331. https://doi.org/10.1121/1.5001491

Adil, M., & Khan, M. K. (2021). Emerging IoT Applications in Sustainable Smart Cities for COVID-19 : Network Security and Data Preservation Challenges with Future Directions. Sustainable Cities and Society, 75, 103311. https://doi.org/10.1016/j.scs.2021.103311

Atlassian. (s. d.). What is DevOps? Atlassian. Consulté 21 janvier 2023, à l'adresse https://www.atlassian.com/devops

Atlassian. (s. d.). Des outils DevOps pour chaque phase du cycle de vie DevOps. Atlassian. Consulté 28 février 2023, à l'adresse https://www.atlassian.com/fr/devops/devops-tools

Bass, L., Weber, I., & Zhu, L. (2015). DevOps : A Software Architect's Perspective (1st éd.). Addison-Wesley Professional

Blumenthal, D., & Tavenner, M. (2010). The "meaningful use" regulation for electronic health records. New England Journal of Medicine, 363(6), 501-504.

Brown, M. (2015). How much data is created every minute? [Blog post]. Retrieved from https://www.domo.com/blog/data-never-sleeps-2-0

Chadwick, I. C., & Raver, J. L. (2015). Motivating organizations to learn : Goal orientation and its influence on organizational learning. Journal of Management, 41, 957-986. https://doi.org/10.1177/0149206312443558

Chen, H., Chiang, R. H., & Storey, V. C. (2012). Business intelligence and analytics: From big data to big impact. MIS quarterly, 36(4), 1165-1188.

Davenport, T., & Kalakota, R. (2019). The potential for artificial intelligence in healthcare. Future Healthcare Journal, 6(2), 94-98. https://doi.org/10.7861/futurehosp.6-2-94

Deng, Z., Liu, S., Huang, H., Lu, X., & Cai, Y. (2015). Understanding health information seeking from an actor-centric perspective. Journal of the Association for Information Science and Technology, 66(5), 946-959.

Detmer, D. E. (2003). Building the national health information infrastructure for personal health, health care services, public health, and research. BMC Medical Informatics and Decision Making, 3(1), Art. 1. https://doi.org/10.1186/1472-6947-3-1

Ghantous, G. B., & Gill, A. (2017). DevOps : Concepts, Practices, Tools, Benefits and Challenges. PACIS 2017 Proceedings. https://aisel.aisnet.org/pacis2017/96

HealthIT.gov. (2020). Health information systems. Retrieved from https://www.healthit.gov/topic/health-information-systems

Heap, M. (2016). Getting Started. In M. Heap (Éd.), Ansible : From Beginner to Pro (p. 1-17). Apress. https://doi.org/10.1007/978-1-4842-1659-0_1

Hersh, W., & Totten, A. M. (2018). Health information technology: Integration, utilization, and optimization. Academic Press.

IDC. (2014). The digital universe of opportunities: Rich data and the increasing value of the internet of things. Retrieved from https://www.emc.com/leadership/digital-universe/2014iview/executive-summary.htm

Jiang, F., Jiang, Y., Zhi, H., Dong, Y., Li, H., Ma, S., Wang, Y., Dong, Q., Shen, H., & Wang, Y. (2017). Artificial intelligence in healthcare : Past, present and future. Stroke and Vascular Neurology, 2(4), 230-243. https://doi.org/10.1136/svn-2017-000101

Ji, C., Mudiyanselage, T. B., Gao, Y., & Pan, Y. (2021). A review of infant cry analysis and classification. EURASIP Journal on Audio, Speech, and Music Processing, 2021(1), Art. 1. https://doi.org/10.1186/s13636-021-00197-5

Jumardi, J., Osmond, A. B., & Saputra, R. E. (2019). Aplikasi Back End Game Edukasi Berbasis Web Menggunakan Qr Code. eProceedings of Engineering, 6(1), Art. 1. https://doi.org/10.34818/eoe.v6i1.8600

Kahn Jr, M. G., Callahan, T. J., Barnard, J., & Bauck, A. E. (2012). A harmonized data quality assessment terminology and framework for the secondary use of electronic health record data. EGEMS (Generating Evidence & Methods to Improve Patient Outcomes), 1(1), 2.

Karamitsos, I., Albarhami, S., & Apostolopoulos, C. (2020). Applying DevOps Practices of Continuous Automation for Machine Learning. Information, 11(7), Art. 7. https://doi.org/10.3390/info11070363

Khalilzad, Z., Kheddache, Y., & Tadj, C. (2022). An Entropy-Based Architecture for Detection of Sepsis in Newborn Cry Diagnostic Systems. Entropy (Basel, Switzerland), 24(9), 1194. https://doi.org/10.3390/e24091194

Kheddache, Y., & Tadj, C. (2019). Identification of Diseases in Newborns Using Advanced Acoustic Features of Cry Signals. Biomedical Signal Processing and Control, 50, 35-44. https://doi.org/10.1016/j.bspc.2019.01.010

Kim, G., Debois, P., Willis, J. (Writer on information technology), Humble, J., & Allspaw, J. (2016). The DevOps handbook : How to create world-class agility, reliability, & security in technology organizations (First edition). IT Revolution Press, LLC.

Mansur, M., & Kasmawi, K. (2017). Pengembangan Sistem Database Terpadu Berbasis Web Untuk Penyediaan Layanan Informasi Website Desa. Jurnal Teknologi dan Sistem Informasi, 3, 73. https://doi.org/10.25077/TEKNOSI.v3i1.2017.73-82

McCloskey, B., Dar, O., Zumla, A., & Heymann, D. L. (2014). Emerging infectious diseases

Nonaka, I. (1994). A dynamic theory of organizational knowledge creation. Organization Science, 5, 14-37. https://doi.org/10.1287/orsc.5.1.14

Obe, R. O., & Hsu, L. S. (2017). PgRouting : A Practical Guide. Locate Press LLC.

O'Dowd, R. (2018). From telecollaboration to virtual exchange : State-of-the-art and the role of UNICollaboration in moving forward. Journal of Virtual Exchange, 1, 1-23. https://doi.org/10.14705/rpnet.2018.jve.1

One Health. (2017). Repéré à l'adresse https://www.who.int/news-room/questions-and-answers/item/one-health

Organisation mondiale de la santé. (2020). Rapport sur la santé dans le monde 2020 : Couverture sanitaire universelle pour tous. Genève : Organisation mondiale de la santé.

Reddy, S. (s. d.). Use of Artificial Intelligence in Healthcare Delivery. Consulté 22 janvier 2023, à l'adresse https://www.academia.edu/38071530/Use_of_Artificial_Intelligence_in_Healthcare_Delivery

Roth, G. (2022). L'avenir de l'automatisation (D. Buxton, Trad.). Variations. Revue internationale de théorie critique, 25, Art. 25. https://doi.org/10.4000/variations.2149

Sharma, A., Borah, S., & Moses, A. (2021). Responses to COVID-19 : The role of governance, healthcare infrastructure, and learning from past pandemics. Journal of Business Research, 122, 597-607. https://doi.org/10.1016/j.jbusres.2020.09.011

Susanto, A., & Meiryani, M. (2019). The impact of Environmental Accounting Information System Alignment on Firm Performance and Environmental Performance : A case of Small and Medium Enterprises s of Indonesia. International Journal of Energy Economics and Policy, 9(2), Art. 2.

Wettinger, J., Breitenbucher, U., & Leymann, F. (2014). Standards-Based DevOps Automation and Integration Using TOSCA. IEEE. https://doi.org/10.1109/ucc.2014.14

World Health Organization (WHO). (s. d.). Consulté 21 janvier 2023, à l'adresse https://www.who.int/

World Health Organization. (2018). Strengthening health information systems for better health and health care. Retrieved from https://www.who.int/health-topics/health-systems#tab=tab_1