

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université de Abderrahmane Mira - Bejaia
Faculté des Sciences Exactes
Département d'informatique



Mémoire du Projet de Fin d'Études

En vue de l'obtention du diplôme de master Professionnel en informatique
Option : Administration et Sécurité des Réseaux

Thème

Mise en place d'un cluster Hadoop automatisé sur Docker

Réalisé par :

MERRI Nassim BOUHAOUI Walid

Composition du jury :

Président : Mr. LARBI Ali - Grade : MCB Univ de Béjaïa

Examineur : Mme. TIAB Amal - Grade : MCB Univ de Béjaïa

Encadreur : Mr. AMROUN Kamal - Grade : MCA Univ de Béjaïa

Co-Encadreur : Mr. EL SAKAAN Nadim - Grade : Doctorant.LMD Univ de Béjaïa

Promo 2019/2020

REMERCIEMENTS

A l'issue de ce mémoire nous remercions d'abord Allah de nous avoir donné l'aide et nous donné la patience et le courage durant nos études.

Nous voudrions présenter nos sincères remerciements à notre co-encadreur Mr El-SAKAAN Naddim pour les nombreuses et systématiques corrections, ses orientations, ses conseils, ses remarques, ses suggestions et sa disponibilité.

Nous remercions vivement le docteur AMROUN Kamal d'avoir accepté d'être notre encadreur.

Aussi nous exprimons notre profonde gratitude et sincère reconnaissance aux membres du jury, d'avoir accepté d'examiner ce travail.

Enfin, nous tenons à remercier toutes les personnes qui nous ont conseillés et relus pendant l'élaboration de ce travail : famille, amis , les enseignants de l'université de Béjaia ainsi qu'aux camarades de promotion.

DÉDICACES

A nos chers parents, source de vie, d'amour et d'affection.

A nos chers frères et soeur d'avoir étaient toujours là pour nous et de nous avoir encourager.

A tous nos proches, tous les ami(e)s, tous ceux et celles qui ont cru en nous, On vous dit aujourd'hui merci.

Walid et Nassim

TABLE DES MATIÈRES

Table des matières	i
Liste des abréviations	iv
Liste des figures	vi
Liste des tableaux	viii
Introduction générale	1
1 Les systèmes d'informations au cœur des métiers	3
1 Introduction :	3
2 Le Parc informatique et ses composants :	3
2.1 Parc matériel :	4
2.1.1 Les matériels clients :	4
2.1.2 Les matériels réseau :	4
2.1.3 Les baies de stockages :	4
2.1.4 Les serveurs :	4
2.1.5 Les matériels d'impression :	5
2.2 Parc Logiciel :	5
3 Les applications de gestion orientées métiers :	6
3.1 Enterprise Resources Planning (ERP) :	6
3.1.1 Définition :	7
3.1.2 Type des ERP :	7
3.1.3 Comparaison entre ERP Propriétaire et ERP Open source :	8
3.1.4 Avantage des systèmes à ERP :	8
3.1.5 Inconvénients des systèmes à ERP :	9

3.2	Customer Relationship Management (CRM) :	9
3.2.1	Définition :	9
3.2.2	Rôle de CRM :	10
3.2.3	Type de CRM :	10
3.3	Gestion de la Maintenance Assistée par Ordinateur (GMAO) :	11
3.4	Transport Management System (TMS) :	12
3.4.1	Définition :	12
3.4.2	Fonctionnalités de TMS :	13
4	Les technologies Big data :	14
4.1	Fedora :	14
4.1.1	Présentation de Fedora :	14
4.2	Docker :	15
4.2.1	Présentation :	15
4.2.2	Architecture de Docker :	15
4.3	Kubernetes :	17
4.4	Hadoop :	17
4.4.1	Présentation de Hadoop :	17
4.4.2	Composants de Hadoop :	18
4.4.3	Architecture de Hadoop :	20
4.4.4	Caractéristiques de Hadoop :	21
5	Conclusion :	21
2	Introduction du cas d'étude	22
1	Introduction :	22
2	Présentation de la société d'accueil :	22
2.1	Présentation de l'entreprise	22
2.2	Historique de l'entreprise :	22
2.3	Pourquoi le lait UHT :	23
2.4	Contrat de franchise Candia/Tchin-Lait :	23
2.5	La laterie Tchin-Lait :	25
2.6	La gestion de l'unité :	26
2.7	Réseau de distribution :	26
2.8	Les missions de l'entreprise :	28
2.9	Structure de la direction du système d'information :	28
2.10	Architecture réseau de l'entreprise :	29
2.11	Les différents serveurs du réseau de l'entreprise :	29
2.12	Les équipements utilisés à Tchin-Lait :	30
2.13	Les applications métiers :	31

2.14	Les logiciels utilisés :	31
2.15	Les systemes d'exploitation utilisés :	31
3	Etude de cas : Tchiv-Lait	32
3.1	Problématique :	32
3.2	Objectif de l'étude :	32
4	Conclusion :	33
3	Mise en place de la solution	34
1	Introduction :	34
2	Présentation de l'environnement de travail :	34
2.1	VMware Workstation 15 :	34
3	Installation et configuration de Fedora 29 sur VMware Workstation 15 :	35
4	Installation de Docker sur fedora :	41
5	Installation de Hadoop :	45
6	Conclusion :	48
4	Déploiement et exploitation de la solution	49
1	Introduction :	49
2	Importation des données :	49
3	Insertion de données dans le HDFS :	50
4	Programme MapReduce :	51
4.1	Code Source :	51
4.2	Compilation du programme :	54
4.3	Exécution du programme :	55
4.4	Obtention des résultats :	56
5	Conclusion :	57
	Conclusion générale	58
	Références Webliographiques	59
	Références Bibliographiques	60
	Résumé	61

LISTE DES ABRÉVIATIONS

ABC	Activity Based Coasting
ADSL	Asymmetric Digital Subscriber Line
API	Interface de Programmation d'Application
BDD	Base De Donnée
BI	Business Intelligence
BT	Bon de Travaux
CE	Community Edition
CPU	Central processing unit
CRM	Customer Relationship Management
DI	Demandes d'Intervention
EDI	Echange de Données Informatisées
EE	Enterprise Edition
ERP	Enterprise Resources Planning
GFS	Google File System
GMAO	Gestion de la Maintenance Assistée par Ordinateur
GNU	GNU's Not Unix
GRC	Gestion de la Relation Client
HDFS	Hadoop Distributed File System
ISO	International Organisation for Standardization
JAR	Java ARchive
KPI	Key Performance Indicator
LAN	Local Area Network
MPT	Maintenance Productive Totale
My SQL	Microsoft Structured Query Language
NAS	Network Attached Storage
NAT	Network Address Translation

OFBiz Open For Business
OLPC One Laptop Per Child
OM Ordre de Maintenance
OS Operating System
OT Ordre de Travaux
PC Personal Computer
PDU Power Distribution Unit
PGI Progiciels de Gestion Intégrée
PME Petites et Moyennes Entreprises
PMI Project Management Institute
PoE Power over Ethernet
RAM Random Access Memory
RHEL Red Hat Enterprise Linux
SAN Storage Area Network
SAP Systems, Applications and Products
SI Système Information
SLC Single Level Cell
SSA Static Single Assignment
TCP/IP Transmission Control Protocol/Internet Protocol
TMS Transport Management System
UHT Ultra Haute Température
VM Virtual Machine

TABLE DES FIGURES

1.1	Les données relatives aux fonctions de l'ERP [4]	6
1.2	Les données Relatives aux fonctions de CRM [26]	10
1.3	Architecture de Docker [8]	15
1.4	Conteneur Docker [8]	16
1.5	Architecture de HDFS [11]	19
1.6	Architecture de Mapreduce [25]	20
1.7	Architecture de Hadoop [12]	21
2.1	Organigramme générale de Tchiv-Lait [14]	26
2.2	Réseau de la distribution [14]	26
2.3	Architecture réseau de Tchiv-Lait [14]	29
2.4	Les équipements utilisés [14]	30
3.1	VMware Workstation 15	35
3.2	Création d'une nouvelle machine virtuelle	35
3.3	Choix de type de configuration	35
3.4	Choix d'installation de système d'exploitation	36
3.5	Sélection de système d'exploitation	36
3.6	Nom de la machine virtuelle	37
3.7	Configuration de la taille de disque dur	37
3.8	Configuration de la memoire vive (RAM)	38
3.9	Configuration des processeurs (CPU) de la machine virtuelle	38
3.10	Choix de support d'installation du système d'exploitation de la machine virtuelle	39
3.11	Configuration de la mise en réseau de la machine virtuelle	39
3.12	Démarrage de la machine virtuelle avec l'image ISO	40
3.13	Vérification de support d'installation	40

3.14	La fenetre de connexion après le démarrage de la machine virtuelle	41
3.15	Installations des mise a jour sur fedora	42
3.16	Redémarrage de système	42
3.17	Activation et configuration de dépôt Docker CE sur le serveur Fedora 29	42
3.18	Mettre en place de dépôt Docker stable	43
3.19	Installation de paquet Docker CE	43
3.20	Démarrage et activation de service Docker	43
3.21	Vérification de l'installation du docker	44
3.22	Lancement de l'image Docker hello-world	44
3.23	La version de java installé sur fedora	45
3.24	Installation de hadoop sur docker	45
3.25	Vérification de la liste des images de docker présentes sur votre système	46
3.26	Lancement de l'image du docker Hadoop à l'intérieur d'un conteneur de docker	46
3.27	Vérification des demons hadoop	47
3.28	Obtention des details sur le conteneur docker	47
3.29	Obtention de l'adresse IP sur laquelle le conteneur fonctionne	47
4.1	Fichier Excel des données	50
4.2	Création d'un répertoire d'entrée	50
4.3	Stockage de fichier de données dans HDFS	50
4.4	Vérification du fichier enregistré	51
4.5	Compilation du programme Mapreduce	54
4.6	Création d'un fichier JAR	55
4.7	Exécution du programme Mapreduce	55
4.8	Le contenu du répertoire de sortie	56
4.9	Les résultats de l'exécution	57

LISTE DES TABLEAUX

1.1	Tableau comparatif entre les logiciels libres et propriétaire [2] [3]	5
1.2	Tableau comparatif entre ERP Propriétaire et ERP Open source [24] [19]	8
2.1	Les centres de distribution [14]	27

INTRODUCTION GÉNÉRALE

Le monde d'aujourd'hui connaît des progrès technologiques considérables dans tous les domaines de l'informatique. Il s'agit d'une science qui étudie la représentation, le traitement et la communication de l'information dans les systèmes naturels et artificiels.

Le développement informatique concerne tout ce qui est lié à l'étude, la conception, le développement, la maintenance, la création d'applications personnalisées, l'amélioration de logiciels existants ou d'autres supports informatiques.

Dans ce monde hautement numérique et extrêmement compétitif, les entreprises cherchent des solutions informatiques de qualité. Afin de satisfaire les besoins d'une entreprise, le développement informatique permet à ces structures de bénéficier des programmes spécialisés. Ces programmes pourront répondre à leurs besoins en termes de promotion du travail sous la responsabilité des services informatiques, d'amélioration de la gestion du cycle de vie et d'optimisation du travail en équipe.

Tchin-lait était à l'origine Tchin Tchin une société spécialisée dans les boissons gazeuses. Cela lui confère une longue expérience dans le conditionnement d'articles liquides. L'arrivée de grandes multinationales sur le marché des boissons gazeuses l'a obligé à se convertir au lait Ultra Haute Température (UHT), d'où la naissance de Tchin-lait. Cette entreprise est un droit du constructeur Candia, elle a l'autorisation du franchisé de construire et de vendre ses produits sous le nom de Candia. De son côté, Candia s'engage à aider le franchisé pour la conception et la vente de ses produits en Algérie.

Les Big Data constituent une combinaison de données structurées, semi-structurées et non structurées collectées par les organisations, qui peuvent être exploitées pour obtenir des informations et utilisées dans des projets d'apprentissage automatique, de modélisation prédictive et

d'autres applications analytiques avancées. Les entreprises utilisent les Big Data accumulées dans leurs systèmes pour améliorer leurs opérations, fournir un meilleur service à la clientèle, créer des campagnes de marketing personnalisées basées sur les préférences spécifiques des clients et, en fin de compte, augmenter leur rentabilité.

Afin de mener à bien notre travail, nous avons organisé ce rapport en quatre chapitres :

- Le premier chapitre intitulé « Les systèmes d'informations au cœur des métiers » comporte trois parties : dans la première on décrira le parc informatique et ses composants, dans la deuxième les applications de gestion orientées métiers et dans la troisième les technologies Big data.
- Le deuxième chapitre nommé « Introduction du cas d'étude » porte en premier lieu, présentation de la société d'accueil (Tchin-Lait Candia Bejaia) et en second lieu, l'étude de cas Tchin-Lait.
- Le troisième chapitre titré « Mise en place de la solution » est consacré aux installations et configurations, en premier lieu, installation et configuration de fedora server sur vmware workstation, en second lieu, installation de docker sur fedora et enfin installation de hadoop sur docker.
- Enfin, le dernier chapitre « Déploiement et exploitation de la solution » contient les réalisations appliquées, ainsi que les tests utilisés pour vérifier si les objectifs ont effectivement été atteints.

A la fin, nous avons présenté une conclusion générale décrivant les points clés développés dans ce mémoire ainsi que les différentes réalisations théoriques et pratiques.

CHAPITRE 1

LES SYSTÈMES D'INFORMATIONS AU CŒUR DES MÉTIERS

1 Introduction :

Comme toute nouveauté le Système d'information (SI) des entreprises a tout d'abord été utilisé comme un élément stratégique permettant d'améliorer la productivité des entreprises. Au fil du temps, dans la majorité des secteurs d'activités, le SI est devenu un élément critique indispensable au fonctionnement des entreprises en question (système de paiement, réservation des trains, gestion des stocks. . .).

Dans ce chapitre, nous allons introduire les systèmes d'informations numériques et présenter les principales applications métiers qui sont exploitées par les entreprises.

2 Le Parc informatique et ses composants :

Toute entreprise travaillant quotidiennement avec des produits informatiques doit gérer son parc informatique. Une médiocre gestion du parc serait capable d'avoir d'importantes conséquences sur le bon fonctionnement de l'entreprise.

Par définition, le parc informatique d'une entreprise est un assemblage, parfois hétérogène de matériels et de logiciels accumulés tout au long des années. On y trouve des matériels différents (pc portable, pc fixe, imprimantes, téléphones, éléments d'interconnexions, etc.), des logiciels et systèmes d'exploitations variés (Windows, Linux, Mac OS, etc.). [1]

Par conséquent, on distingue deux catégories de dispositifs composant un parc informatique, à savoir les matériels et les programmes.

2.1 Parc matériel :

Les matériels dont il est question ici sont bien évidemment les solutions qui peuvent être considérés comme étant des machines, dénommées hardware, à savoir les ordinateurs et tous ses composants internes mais également les matériels accessoires... Pour mieux s'y retrouver, séparons les nombreux genres d'éléments du parc en 5 catégories : les matériels clients, les matériels réseaux, les baies de stockages, les serveurs et les matériels d'impression.

2.1.1 Les matériels clients :

Il s'agit des équipements finaux utilisés par les collaborateurs. Ils regroupent : les ordinateurs (bureautique ou ordinateurs portables), les moniteurs, les périphériques (du clavier à la souris, en passant par la tablette graphique), les téléphones, les tablettes, etc. [1]

2.1.2 Les matériels réseau :

Il s'agit des équipements qui permettent la communication ou l'exploitation des données sur un réseau informatique. Nous y retrouvons : [1]

- 1) **Les matériels d'interconnexions** : Switch, routeur, borne Wifi, etc.
- 2) **Les Power Distribution Unit (PDU)** : sont les dispositifs permettant de délivrer de l'électricité dans les salles serveurs.

2.1.3 Les baies de stockages :

il s'agit des équipements qui permettent la sauvegarde des données informatiques. Nous y retrouvons deux éléments :

- 1) **Network Attached Storage (NAS)** : désigne un périphérique de stockage (généralement un ou plusieurs disques durs) relié à un réseau par un protocole de communication tel que TCP/IP par exemple. [21]
- 2) **Storage Area Network (SAN)** : est un réseau spécialisé permettant de partager de l'espace de stockage à une bibliothèque de sauvegarde et à des serveurs. [21]

2.1.4 Les serveurs :

Un serveur réseau est un ordinateur spécifique partageant ses ressources avec d'autres ordinateurs appelés clients. [1]

On peut distinguer quelques types de serveurs : Serveur Central, Serveur d'Application, Serveur de fichiers, Serveur d'impression et Serveur de messagerie électronique.

2.1.5 Les matériels d'impression :

Les équipements d'impression regroupent les imprimantes, mais aussi les cartouches et toners. [1]

2.2 Parc Logiciel :

Pour ce qui est des programmes ou software, ce sont toutes les applications informatiques qui peuvent être considérées comme étant des logiciels et qui sont nécessaires au fonctionnement des machines (ordinateurs) ainsi que du réseau. . . Pour mieux s'y retrouver, séparons les divers types d'éléments du parc en 2 catégories : les logiciels libres et les logiciels propriétaires. [1]

Critères	Logiciel propriétaire	Logiciel libre
Licence	Payant/Gratuit	Gratuit
Mise à jour nouvelles version	Payante	Payante dans la plupart des cas
Accès au code source	Non	Oui, modification possible
Conception	Individuelle : c'est l'entreprise qui développe	Collaborative : c'est toute une communauté qui développe
Sécurité logiciel	Tout système connaît des problèmes de sécurité Le système propriétaire n'a pas intérêt que ça se sache	Tout le monde a intérêt à ce que cela se sache pour que ce soit résolu
Exemples	- Windows, Macintosh - Microsoft Office - Oracle - Outlook - Internet Explorer	- GNU Linux (Ubuntu..) - OpenOffice - MySQL - Mozilla Thunderbird - Firefox

TABLE 1.1 – Tableau comparatif entre les logiciels libres et propriétaire [2] [3]

3 Les applications de gestion orientées métiers :

Au sein d'un secteur en constante progression, synthétisé par une mondialisation et une compétitivité augmentée, la prise de décisions endosse une fonction toujours plus gros. Les entreprises de classe internationale doivent affronter différents intégration ; les consommateurs et les acheteurs sont toujours plus exigeants et la compétition plus agressive et globale.

Pour s'en sortir, les entreprises doivent vouloir maximiser la totalité de leurs processus, de l'approvisionnement jusqu'à la livraison physique des produits. Donc, l'amélioration classique d'éléments individuels de l'entreprise (taille et cadence des commandes, lotissement, entreposage, distribution, . . .) cède sa position à l'amélioration intégrée de la chaîne logistique rendue envisageable avec l'ampleur des techniques de l'information et de la communication comme les Progiciels de Gestion Intégrée (PGI) des Ressources ou, en anglais, Enterprise Resource Planning (ERP).

3.1 Enterprise Resources Planning (ERP) :

Depuis le commencement des années 90, les Progiciels de Gestion Intégrée (PGI) ou plus habituellement nommés ERP en anglais Enterprise Resources Planning est devenue de plus en plus populaire et continue de se propager à une vitesse vertigineuse dans les entreprises. du monde entier. Qu'est-ce qu'un ERP ? [4]

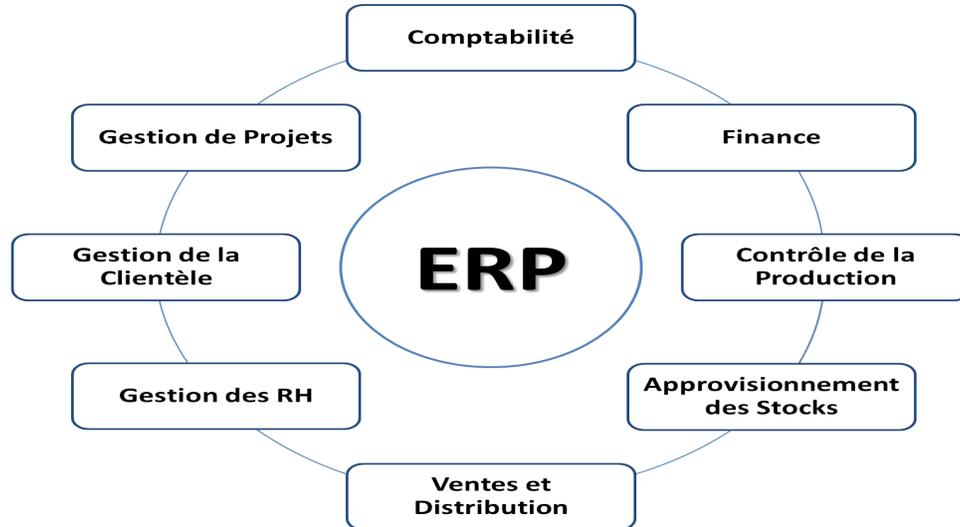


FIGURE 1.1 – Les données relatives aux fonctions de l'ERP [4]

3.1.1 Définition :

Les ERP sont avant tout ce que l'on appelle des progiciels de gestion intégrés. Ils sont destinés aux sociétés ayant besoin de logiciels performants adaptés à leurs activités, afin d'assurer la meilleure gestion possible, dans chacun de ces domaines (la gestion des stocks, la comptabilité, les achats, les ventes, les ressources humaines, entre autres).

En maîtrisant l'usage du PGI, l'entreprise remplit toutes les conditions pour baisser le plus possible les délais de mise en marché des produits et des services, et d'amoinrir ses dépenses. Les PGI cherchent à mettre en réseau, par une base de données unique, la totalité des données se rapportant à des fonctions de l'entreprise (comptabilité, finance, contrôle de la fabrication, approvisionnement des stocks, ventes et distribution, gestion des ressources humaines, gestion de la clientèle, gestion de programmes, etc.) Donc, les PGI améliorent la diffusion de l'information en interne pour une excellente vivacité de l'entreprise. [18]

3.1.2 Type des ERP :

1) Les ERP propriétaires :

Un ERP propriétaire est un progiciel créé par une société spécialisée dans la conception et la mise en place de logiciels et de systèmes informatiques. Comme tout ERP, c'est un progiciel de gestion intégrée. Il permet aux entreprises de proposer des modules indépendants à chaque service tout en travaillant sur une seule et même base de données. Chaque information est créée ou actualisée en temps réel et sa traçabilité est garantie. Ces ERP ont un coût élevé et sont destinés aux grandes entreprises et multinationales, quelque ERP spécifiques sont également conçus pour des PME/PMI. [19]

Exemples d'ERP propriétaire sur le marché : SAP (leader mondial avec + de 40% du marché), Oracle/Peoplesoft, Microsoft, SSA Global, Intenia/Lawson, Infor Global Solutions, CeGid...

2) Les ERP open source :

C'est un logiciel libre qui n'exige pas l'achat d'une licence, ce qui offre un moyen de faire de sérieuses économies, il est 20% à 50% moins cher qu'un ERP propriétaire. Le manque de licence sur les ERP open source offre une forme d'autonomie aux sociétés qui ne prennent aucun engagement. [24]

Parmi les ERP Open Source les plus connus on trouve : Odoo, Aria, Compiere /Adempiere, ERP5, Fistera, Open for Business (OFBiz), OpenBravo, PGI Suite, Value Enterprise, Tiny ERP...

Odoo : Connue sous l'appellation Open-ERP (aussi TinyERP) auparavant, est un progiciel open source apparu au mois de février 2005 au royaume de Belgique par Fabien Pinckaers renommé Odoo en 2014. Il est créé sur une suite de module de gestion de donnée d'entreprise

inclu sous une base de données unique. Cette BDD centrale est liée à une couche opérationnelle très innovante qui introduit en relation des données d'origines diverses et garantit un déroulement utile des processus transversaux de création de valeur ajoutée de l'entreprise. Des milliers de modules fonctionnels sont installés sur cette fondation technique solide et efficace. Ils donnent les applications métier dont chacun a besoin au sein de l'entreprise. Cette approche constituée de modules simplifie l'intégration de nouvelles fonctionnalités ayant l'aspect de modules additionnels. [23]

3.1.3 Comparaison entre ERP Propriétaire et ERP Open source :

Critères	ERP Propriétaire	ERP Open source
- Licence	- Achat des licences	- Pas de licences de base
- Prise en charge	- Prise en charge complète	- Manque d'accompagnement au changement
- Assistance, Support	- Oui	- Non
- Accès	- Les éditeurs d'ERP propriétaires donnent un certain accès très limité à leur ERP via des webservices.	- L'ERP open source, c'est l'ensemble du code qui est ouvert et pas seulement ses webservices.

TABLE 1.2 – Tableau comparatif entre ERP Propriétaire et ERP Open source [24] [19]

3.1.4 Avantage des systèmes à ERP :

Les ERP ont beaucoup d'avantages dans le monde des entreprises, parmi ces avantages nous citons :

- Définition homogène et unique des données (même structure). [18]
- Informations cohérentes (mêmes données). [18]
- Information en temps réel. [18]
- Accès aux informations en toute sécurité et avec des habilitations bien définis. [18]
- Garantir la fiabilité et l'intégrité des différentes données et informations stockés. [18]
- Adaptation aux multinationales (Outils multilingues et multidevises). [18]
- Maintenance simplifiée : [18]
 - Maintenance corrective assurée par l'éditeur.
 - Maintenance évolutive par le service informatique (règles/fonctions).

Pour assurer une utilisation optimale des ERP, un paramétrage et une adaptation des besoins spécifiques de l'entreprise et nécessaire. [18]

3.1.5 Inconvénients des systèmes à ERP :

Bien que la progression des progiciels de gestion sur le marchés, notamment avec le développement des techniques de l'information, l'installation d'une telle solution reste de nouveau un projet et une phase très critique, dont le succès et l'atteinte des objectifs reste sans assurance directement. [18]

Parmi les inconvénients de la mise en place d'un ERP : [18]

- Mise en œuvre complexe :
 - Analyse et étude préalable de longue durée.
 - Intégration de l'ERP au SI existant.
 - Adaptation organisationnelle et opérationnelle.
- Dépendance au fournisseur (problèmes techniques).
- Disparition de postes (reclassement ou licenciement).
- Le prix haut d'une tel solution niveau matériels, licence d'utilisation, effort d'intégration, formation utilisateurs et entretien.
- Investissements lourds. . .

3.2 Customer Relationship Management (CRM) :

3.2.1 Définition :

La Gestion de la Relation Client (GRC), connue aussi sous son nom anglais de Customer Relationship Management (CRM), est un terme utilisé dans l'industrie de l'information pour désigner des méthodologies, des logiciels et généralement des fonctionnalités Internet qui aide une entreprise à gérer les relations avec les clients de manière organisée. [17]

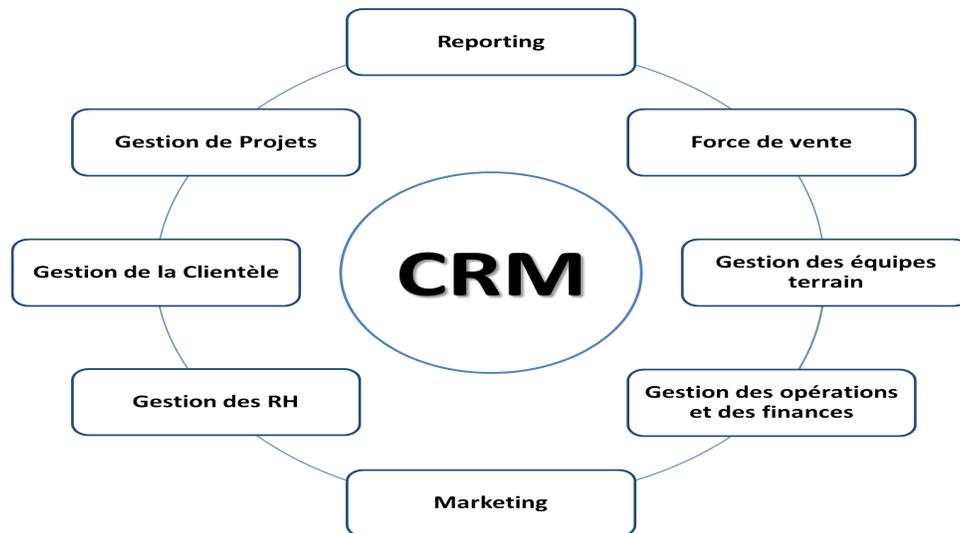


FIGURE 1.2 – Les données Relatives aux fonctions de CRM [26]

3.2.2 Rôle de CRM :

La mise en œuvre de CRM génère les avantages commerciaux et techniques suivants :

- Fournir la capacité de connaître et de mettre en œuvre les meilleures pratiques mondiales et donne les meilleurs moyens d'étalonner la compétitivité de l'organisation axée sur le client. [26]
- Fournir un support réactif pour entreprendre toutes les différentes versions de programmes et de méthodologies d'amélioration de processus, y compris l'innovation de processus, l'amélioration de processus et les canaux d'interaction. [26]
- Permettre à une entreprise d'augmenter considérablement son niveau d'exploitation ou même d'entrer dans différents entreprises, sans aucune interruption. [26]
- Intègre les données de l'organisation dans une seule base de données complète. [26]
- Réduire considérablement les coûts de maintenance des systèmes. [26]

3.2.3 Type de CRM :

Les sociétés d'analystes, y compris Meta Group, classifient le CRM en plusieurs types :

- 1) **CRM opérationnel** : C'est le domaine concerné par l'automatisation des processus métier impliquant des points de contact client front-office. L'automatisation des ventes, l'automatisation du marketing et les relations clients. Le CRM opérationnel a toujours été un outil principal dans le secteur des dépenses des entreprises. [15]
- 2) **CRM analytique** : Il s'agit de la saisie, du stockage, de l'organisation, de l'analyse, de l'interprétation et l'utilisation des données créées du côté opérationnel de l'entreprise. L'in-

tégration de solutions CRM analytiques avec un CRM opérationnel sont une considération importante. [15]

- 3) **CRM collaboratif** : l'utilisation de services collaboratifs et infrastructure permettant d'interagir entre une entreprise et ses multiples canaux possibles. Cela permet une interaction entre les clients, l'entreprise et ses employés. [15]

3.3 Gestion de la Maintenance Assistée par Ordinateur (GMAO) :

La Gestion de la Maintenance Assistée par Ordinateur (GMAO) est une phase que les sociétés franchissent pour évoluer vers plus d'efficacité, de rentabilité et de productivité

Une GMAO doit prendre en charge, d'un côté, la plupart des données et des moyens associées à une organisation d'entretien, ensuite d'en tirer les indicateurs stratégiques requis à une prise des bonnes dispositions. D'un autre côté, elle doit répondre aux besoins quotidiens des gens du suivi, cela signifie permettre d'effectuer des initiatives sur des accessoires avec le plus d'efficacité, pour cela, une GMAO constituera la mémoire et l'expérience du service de maintenance.

Habituellement, cette mémoire et cette expérience constituent une base commune pour le travail entre spécialistes et experts techniques, ce qui pose de gros soucis en leur absence. De toute évidence, dans de nombreux cas, un historique papier des interventions est cependant déposé dans une mémoire journalier, mais les informations sont longues à trouver et la plupart du temps, ces rapports manquent de détails. La GMAO, avec l'avantage des nouvelles technologies, a réussi à gérer une quantité considérable d'informations et simplifier leur exploration, à estimer les coûts et les ratios et d'offrir aux gestionnaires un planning d'entretien. "Une GMAO est une base de données (historique) qui est alimentée par le personnel de maintenance via un formulaire. Les problèmes sont stockés pour certains matériaux (date, heure passée, intervenant, équipement renouvelé....etc). [5]

Les fonctions les plus courantes de ces progiciels sont :

- 1) **la gestion des équipements** : inventaire des équipements, localisation, gestion d'information dédiée par type d'équipement (production, bâtiments, véhicules, réseaux, ordinateurs).
- 2) **la gestion de la maintenance** : corrective avec Ordre de Travaux (OT), ou Bon de Travaux (BT), ou Ordre de Maintenance (OM), préventive (systématique, conditionnelle, prévisionnelle), ce plugin comprend la plupart du temps des fonctionnalités ouvertes à des utilisateurs au-delà du service d'entretien, en tant que une gestion des Demandes d'intervention (DI), consentant à toute personne autorisée de l'entreprise de déclarer un défaut devant être prise en considération par la maintenance.
- 3) **la gestion de la mise en sécurité des installations pour les travaux de maintenance** : consignation, centralisation, accord de sécurité, déconsignation pour autoriser le blocage optimal d'une mise en place durant des opérations de maintenance.

- 4) **la gestion des stocks** : magasins, quantités minimum ou maximum de réapprovisionnement, analyse ABC, listes de sélection, positionnement et exploration, articles de rechange, catalogue fournisseurs.
- 5) **la gestion des achats** : de pièces détachées ou de services (sous-traitance, forfait ou régie), cycle devis, demande d'achat, commande, réception et retour fournisseur, facturation.
- 6) **la gestion du personnel et le planning** : activités, métiers, planning de charge, prévisionnel, pointage des heures.
- 7) **la gestion des coûts et budget** : de main-d'œuvre, de stocks, d'achat, de location de matériel, préparation des budgets, suivi périodique, rapports d'écart.
- 8) **Indicateurs clés de performance** : cockpit de pilotage ou tableau de bord pour le manager (requêtes de base de données concernant des statistiques, des alertes).

Parmi les résultats attendus de l'utilisation d'une GMAO on trouve :

- meilleure gestion et réduction des coûts (main-d'œuvre, pièces détachées, traitement administratif, etc.). [5]
- amélioration de la fiabilité et de la disponibilité des équipements. [5]
- optimisation des achats (aide aux appels d'offres, gestion des contrats de prestataires externes, etc.). [5]
- amélioration de la planification des interventions. [5]
- amélioration de la gestion des stocks (meilleur contrôle des sorties, aide aux inventaires, optimisation du taux de rotation, etc.). [5]
- traçabilité des équipements, parfois pour répondre à des contraintes réglementaires. [5]
- participation à une démarche de Maintenance Productive Totale (MPT). [5]
- aide à la décision grâce à la fourniture d'indicateurs plus objectifs, notamment les décisions de renouvellement de matériel. [5]

3.4 Transport Management System (TMS) :

3.4.1 Définition :

Le TMS ou Transport Management System est un progiciel qui regroupe tout ou partie des fonctions de gestion du transit de marchandises.

L'extrême variété des besoins fonctionnels des différents acteurs a dirigé à une proposition très diverse, allant du design d'un réseau logistique au suivi en temps réel des camions, en passant par l'optimisation de voyages, l'amélioration du taux de remplissage des véhicules, le contrôle des factures, l'EDI automatisé des commandes et des documents.

Cette offre peut répondre selon les cas à la demande des chargeurs, des transporteurs, des organisateurs de transport, des cabinets de conseil. A l'intérieur même de ces organisations, elle répondra à des attentes différentes en fonction du rôle de chacun. [16]

3.4.2 Fonctionnalités de TMS :

- 1) **Visibilité** : La visibilité sur les données et l'activité de la chaîne d'approvisionnement est le principal facteur de valeur derrière TMS et le retour sur l'investissement. Des quantités massives de données circulent dans la chaîne d'approvisionnement. Une partie est capturée dans des systèmes internes ; certains sont hébergés dans des systèmes externes de transporteurs et de fournisseurs ; certains ne sont jamais capturés du tout. Et si la connectivité et l'accès à ces données constituent un sujet totalement différent, la visibilité sur l'activité est améliorée pour les parties prenantes internes et externes. Les expéditeurs peuvent consulter les expéditions pour préparer et planifier la main-d'œuvre pour des volumes de quais lourds ou légers. En même temps, le client final reçoit des liens de suivi vers les sites Web des opérateurs pour les mises à jour de localisation des envois, les messages d'échange de Données Informatisées (EDI) des opérateurs, ou les mises à jour peuvent provenir d'une Interface de Programmation d'Application (API) directe. [22]
- 2) **Rapports** : Les rapports de Business Intelligence (BI) doivent apporter une visibilité significative et exploitable aux données et aux tendances. Le reporting BI entraîne des économies de coûts et une amélioration continue des mesures de Key Performance Indicator (KPI), Les rapports de synthèse, d'exception, de performance, d'activité et d'avance fournissent tous des informations sur les performances de la chaîne logistique et apportent une visibilité aux données. [22]
- 3) **Opérations** : Avec TMS, les expéditeurs conservent le contrôle tactique stratégique et quotidien de leurs opérations. Les employés de l'entreprise gèrent le processus logistique via le TMS, en commençant par la planification et la sélection du transporteur, en passant par le cycle de vie complet de la commande, et par l'audit de facturation et le paiement. Dans cet environnement de «True TMS», les expéditeurs appuient sur les boutons et gèrent tous les événements d'un envoi .[22]
- 4) **Optimisation** : Une autre fonction courante parmi la plupart des systèmes de gestion des transports est la capacité d'analyser un lot d'envois en fonction de divers paramètres sophistiqués afin de déterminer le plan d'itinéraire le plus rentable. TMS offre la possibilité de déterminer le mode le moins coûteux, d'analyser et de planifier des itinéraires et d'analyser les possibilités de consolidation en combinant les envois en un seul chargement. Le moteur d'optimisation d'un TMS permet également un routage dynamique, statique et en boucle fermée. [22]
- 5) **Automatisation** : Avec TMS, personne ne saisit les commandes. Cela permet non seulement

de gagner beaucoup de temps, car cela extrait les données de leur système dans le TMS, mais cela minimise également le risque d'erreur de frappe, ce qui augmente la précision. [22]

L'automatisation élimine également les processus manuels, permettant d'effectuer les opérations suivantes en un seul clic :

- **Classement** : saisissez les détails de l'envoi et recevez les tarifs du fournisseur de base. [22]
- **Réservation** : consultez les tarifs des opérateurs et réservez des charges via TMS. [22]
- **Appel d'offres** : appels d'offres par courrier électronique / EDI / API, documents d'appel d'offres personnalisés, réponses sans souci. [22]
- **Suivi** : les opérateurs peuvent mettre à jour les charges, suivi du tableau de bord disponible. [22]
- **Audit** : Identifiez et rapprochez les écarts de facture, approuvez et traitez automatiquement les factures correspondantes. [22]
- **Facturation** : une fois les informations saisies, l'utilisateur peut générer des factures pour tout chargement en un seul clic, extraire les informations AP approuvées vers votre ERP pour le traitement du paiement. [22]
- **Reporting** : rapport de synthèse, rapports d'activité, rapports de performance et rapports avancés. [22]

4 Les technologies Big data :

L'explosion des données produites ces dernières années dans divers domaines a exigé de nouvelles techniques de traitement, de nouvelles grandes architectures de traitement des données et des algorithmes intelligents permettant d'exploiter efficacement d'énormes ensembles de données pour obtenir des informations utiles et améliorer la découverte des connaissances. L'explosion des données entraîne de nombreux défis pour faire face à la complexité de la surcharge d'informations. De nombreux outils et techniques ont été développés au fil des années pour faire face à ces défis.

4.1 Fedora :

4.1.1 Présentation de Fedora :

Fedora est un système d'exploitation open source populaire basé sur Linux et sponsorisé par Red Hat. [6]

Fedora est conçu comme un système d'exploitation sécurisé et d'usage général. Le système d'exploitation est développé selon un cycle de publication de six mois, sous les auspices du projet Fedora. [6]

Selon le projet Fedora, Fedora est "toujours libre pour quiconque d'utiliser, de modifier et de distribuer". Fedora serait la deuxième distribution Linux la plus utilisée, après Ubuntu. Il existe plus d'une centaine de distributions basées sur Fedora, dont Red Hat Enterprise Linux (RHEL) et le système d'exploitation du projet One Laptop Per Child (OLPC). [6]

4.2 Docker :

4.2.1 Présentation :

Docker est un projet open-source qui automatise le déploiement d'applications dans des conteneurs logiciels et facilite la portabilité et l'intégration d'applications dans différents environnements. Ces conteneurs d'applications sont similaires à des machines virtuelles légères dans la mesure où ils peuvent fonctionner indépendamment les uns des autres et de l'hôte qui les exécute. L'idée derrière Docker est de permettre aux développeurs de développer facilement des applications, de les expédier dans des conteneurs qui peuvent ensuite être déployés n'importe où. [7]

4.2.2 Architecture de Docker :

L'architecture de Docker est basée sur une architecture de type client-serveur. Il y a un client Docker et un Docker daemon qui prend le rôle de serveur. Voici les principales actions qu'il est possible de réaliser avec l'outil de ligne de commande de Docker : [7]

- Récupérer une image depuis un registre et la transférer vers un Docker daemon ou transférer une image vers un registre à partir d'un Docker daemon.
- Créer une image Docker.
- Démarrer un conteneur sur un serveur Docker.

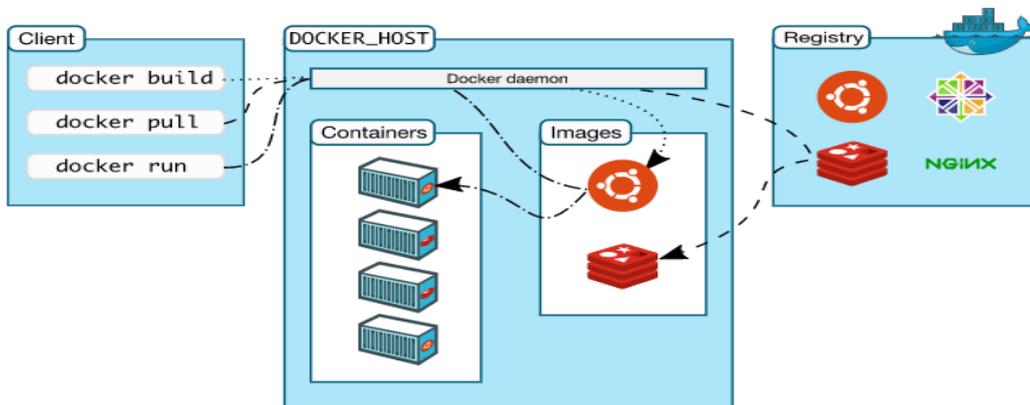


FIGURE 1.3 – Architecture de Docker [8]

1) **Docker Daemon :**

Le service en arrière-plan exécuté sur l'hôte qui gère la construction, l'exécution et la distribution des conteneurs Docker. Le daemon est le processus qui s'exécute dans le système d'exploitation et avec lequel les clients parlent. [9]

2) **Docker Client :**

L'outil en ligne de commande qui permet à l'utilisateur d'interagir avec le démon. Plus généralement, il peut y avoir d'autres formes de clients, comme Kitematic qui fournit une interface graphique aux utilisateurs. [9]

3) **Docker Conteneur :**

À l'origine, le concept de conteneurisation utilisé par Docker pour ses solutions logicielles vient des conteneurs physiques utilisés depuis des dizaines d'années dans le domaine du transport de marchandises. Les différentes actions suivantes peuvent être effectuées sur un conteneur à partir de l'outil de ligne de commande de Docker ou d'une API. Cette application client-serveur est composée de trois éléments principaux : [9]

- Exécuter un conteneur
- Arrêter un conteneur
- Déplacer un conteneur
- Supprimer un conteneur

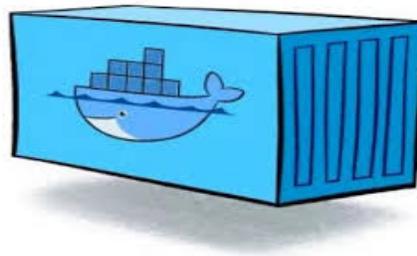


FIGURE 1.4 – Conteneur Docker [8]

4) **Docker image :**

Une Docker image est un modèle permettant de créer un conteneur. En fait, une Docker image correspond à une manière simple de réutiliser et déplacer des données. Ces données représentées par une Docker image correspondent au contenu d'un conteneur. Il est possible de créer des images ou de télécharger et d'utiliser des images créées et mises à disposition par d'autres personnes. La description d'une Docker image est faite dans un simple fichier texte appelé Dockerfile. [9]

Un Dockerfile est un fichier de configuration contenant plusieurs instructions permettant de construire une Docker image. Les instructions concernent plusieurs tâches comme la création d'un répertoire, la copie de fichiers, etc. [9]

5) **Docker registry :**

Un registre d'images de Docker. Vous pouvez considérer le registre comme un répertoire de toutes les images Docker disponibles. Si nécessaire, on peut héberger ses propres registres de Docker et les utiliser pour extraire des images. [9]

4.3 **Kubernetes :**

Kubernetes est un orchestrateur développé par Google. il est largement le plus utilisé sur le marché. Il laisse à l'utilisateur une grande liberté sur la gestion de son réseau de conteneurs tout en proposant de nombreuses fonctions pré-codées. Il présente aussi de nombreux autres avantages, comme sa compatibilité avec tous les types de conteneur qui existent ainsi que sa robustesse. [20]

Kubernetes, est donc un outil très pratique pour l'orchestration mais sa mise en place est compliquée. Bien qu'il existe de nombreux outils annexes développés dans le but de simplifier cette mise en place, tel que Minikube, pour pouvoir bénéficier de tous les avantages de Kubernetes, notamment sa robustesse, l'infrastructure à mettre en place est complexe et grossit très rapidement avec le nombre de conteneurs à gérer. [20]

4.4 **Hadoop :**

4.4.1 **Présentation de Hadoop :**

Hadoop est un framework open source écrit en java qui permet le traitement distribué de grands ensembles de données sur des grappes d'ordinateurs en utilisant des modèles de programmation simples. L'application Hadoop fonctionne dans un environnement qui fournit un stockage et un calcul distribué sur des grappes d'ordinateurs. Hadoop est conçu pour passer d'un seul serveur à des milliers de machines, chacune offrant des possibilités de calcul et de stockage local. [10]

4.4.2 Composants de Hadoop :

Hadoop se compose de deux sous-projets :

- 1) **Hadoop Distributed File System (HDFS)** : s'occupe de la partie stockage des applications Hadoop. Les applications MapReduce consomment les données du HDFS. HDFS crée de multiples répliques de blocs de données et les distribue sur des nœuds de calcul dans un cluster. Cette distribution permet des calculs fiables et extrêmement rapides. [25]
 - a) **Définition et caractéristiques** : Le Hadoop Distributed File System (HDFS) est un système de fichiers distribués, extensible et portable pour Hadoop inspiré par le Google File System (GFS). Il a été conçu pour stocker de très gros volumes de données dans le cluster. Il se diffère du reste des systèmes de gestion de fichier traditionnel par les principales caractéristiques suivantes : [25]
 - Portabilité : le système de fichiers HDFS est indépendant du noyau du système d'exploitation. [25]
 - Distributivité : HDFS est un système distribué où chaque nœud d'un cluster correspond à un sous-ensemble du volume global de données du cluster. [25]
 - HDFS utilise des tailles de blocs largement supérieures à ceux des systèmes classiques. [25]
 - HDFS fournit un système de réplication des blocs dont le nombre de répliquions est configurable. [25]
 - b) **Architecture de HDFS** : Il contient une architecture maître/esclave. Cette architecture consiste en un seul NameNode qui joue le rôle de maître, et plusieurs DataNodes qui jouent le rôle d'esclave. [25]
 - **NameNode (le maître)** : Il s'agit d'un nœud maître unique qui existe dans le cluster et qui s'occupe de gérer l'état du HDFS et l'espace de noms du système de fichiers en effectuant une opération telle que l'ouverture, le renommage et la fermeture de fichiers. Il simplifie l'architecture du système, mais comme il s'agit d'un nœud unique, il peut devenir un point de défaillance. [25]
 - **SecondaryNameNode** : est aussi un nœud maître. Le SecondaryNameNode est chargé de la maintenance des nœuds du cluster, ou il garde sous contrôle l'espace disque utilisé, limite la charge processeur du NameNode et il permet la continuité de fonctionnement du cluster Hadoop en cas de panne. [25]
 - **DataNode (l'esclave)** : Le cluster HDFS contient plusieurs DataNodes (nœuds esclaves), chaque DataNode contenant plusieurs blocs. Ces blocs sont utilisés pour stocker des données. Il est de la responsabilité du DataNode de lire et d'écrire les requêtes des clients du système de fichiers. Il effectue la création, la suppression et la réplication des blocs sur instruction du NameNode.[25]

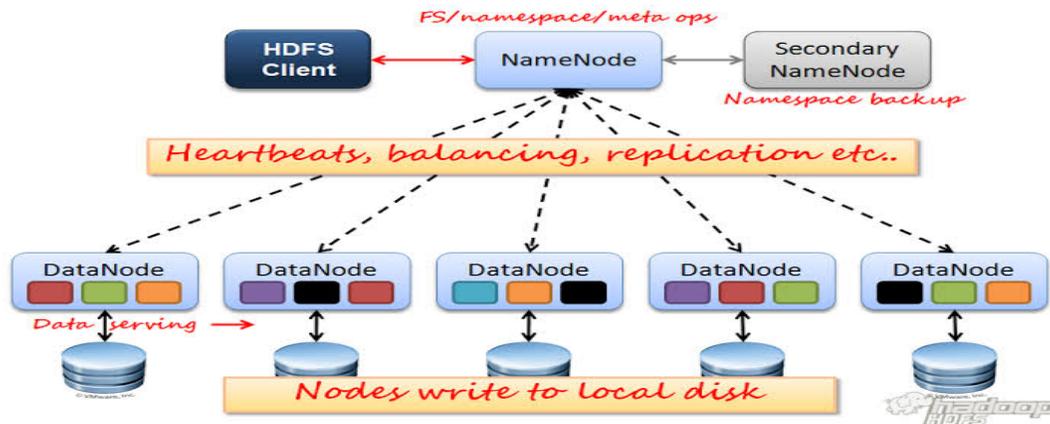


FIGURE 1.5 – Architecture de HDFS [11]

2) **MapReduce** : est un cadre logiciel et un modèle de programmation utilisé pour le traitement d'énormes quantités de données. Le programme MapReduce fonctionne en deux phases, à savoir Map et Reduce. Les tâches Map traitent du fractionnement et du mappage des données tandis que les tâches Reduce mélangent et réduisent les données. [25]

a) **Fonctionnement de MapReduce** :

Au cœur de MapReduce se trouvent deux fonctions : Map et Reduce. Elles sont séquençées l'une après l'autre. [25]

- **Map** : prend une paire clef/valeur du lecteur d'entrée, effectue un calcul sur celle-ci, puis produit le résultat sous forme de paire clef/valeur également. Les résultats des tâches de Map sont d'abord transmis à la mémoire tampon de la mémoire principale, puis lorsqu'elle est presque pleine, ils sont transférés sur le disque. La fonction Map s'écrit de la manière suivante : $\text{Map}(\text{clé1}, \text{valeur1}) \rightarrow \text{List}(\text{clé2}, \text{valeur2})$. [25]
- **Reduce** : est définie par l'utilisateur et est invoquée une fois pour chaque clé distincte et est appliquée sur l'ensemble des valeurs associées à cette clé ; c'est-à-dire que les paires ayant la même clé seront traitées comme un seul groupe. La fonction Reduce s'écrit de la manière suivante : $\text{Reduce}(\text{clé2}, \text{List}(\text{valeur2})) \rightarrow \text{List}(\text{valeur2})$. [25]

b) **Architecture de MapReduce (purement maître-esclave)** :

Le MapReduce possède une architecture maître-esclave :

- **Le maître JobTracker** : [25]
 - Ordonne les différentes tâches des jobs soumis ;
 - Assigne les tâches aux TaskTrackers ;
 - Gère l'ensemble des ressources du système ;
 - Reçoit les jobs des clients.

- L'esclave **TaskTracker** : [25]
 - Exécution des tâches dans une autre JVM (Child) ;
 - Heartbeat avec le JobTracker ;
 - Exécute les tâches données par le Jobtracker ;
 - A une capacité en termes de nombres de tâches qu'il peut exécuter.

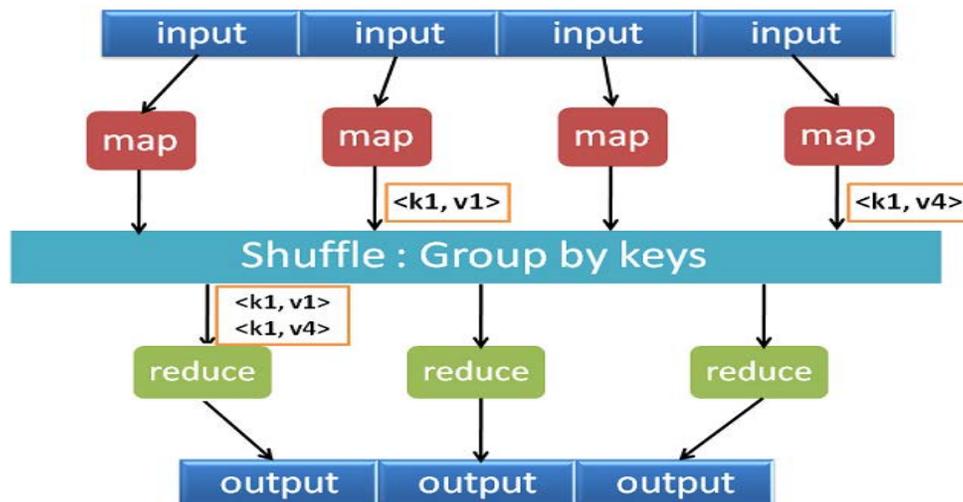


FIGURE 1.6 – Architecture de Mapreduce [25]

4.4.3 Architecture de Hadoop :

Hadoop dispose d'une architecture maître-esclave pour le stockage des données et le traitement distribué des données à l'aide des méthodes MapReduce et HDFS. [12]

- **NameNode** : représente tous les fichiers et répertoires utilisés dans l'espace de noms. [12]
- **DataNode** : vous aide à gérer l'état d'un nœud HDFS et vous permet d'interagir avec les blocs. [12]
- **MasterNode** : vous permet d'effectuer un traitement parallèle des données à l'aide de Hadoop MapReduce. [12]
- **SlaveNode** : sont les machines supplémentaires du cluster Hadoop qui vous permettent de stocker des données pour effectuer des calculs complexes. De plus, tous les nœuds esclaves sont livrés avec un Task Tracker et un DataNode. Cela vous permet de synchroniser les processus avec le NameNode et le Job Tracker respectivement. [12]

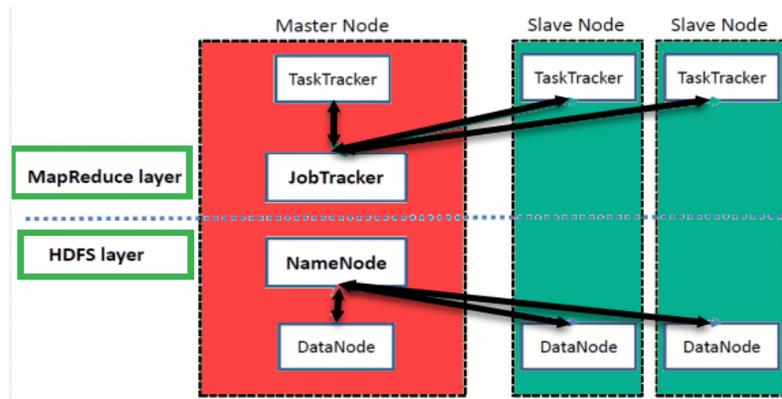


FIGURE 1.7 – Architecture de Hadoop [12]

4.4.4 Caractéristiques de Hadoop :

- Convient pour l'analyse de données volumineuses : comme les Big Data ont tendance à être distribuées et non structurées par nature, les clusters Hadoop sont les mieux adaptés à l'analyse des Big Data. Comme c'est la logique de traitement (et non les données réelles) qui circule vers les nœuds de calcul, la consommation de bande passante du réseau est moindre. Ce concept est appelé "concept de localité des données", ce qui permet d'accroître l'efficacité des applications basées sur Hadoop. [12]
- Évolutivité : les clusters Hadoop peuvent être facilement mis à l'échelle en ajoutant des nœuds de cluster supplémentaires et permettent ainsi la croissance des Big Data. De plus, la mise à l'échelle ne nécessite pas de modifications de la logique d'application. [12]
- Tolérance aux fautes : l'écosystème Hadoop a une disposition permettant de reproduire les données d'entrée sur d'autres nœuds du cluster. Ainsi, en cas de défaillance d'un nœud de grappe, le traitement des données peut toujours se poursuivre en utilisant les données stockées sur un autre nœud de grappe. [12]

5 Conclusion :

Dans la première partie de ce chapitre, nous avons effectué une présentation de parc informatique et ses composants tels que parc matériel et parc logiciel, tandis que dans la seconde partie nous avons présenté quelques applications de gestion orientées métiers ERP, CRM, GMAO et TMS, nous avons aussi présenté dans la dernière partie quelques technologies big data Fedora, Docker, Kubernetes et Hadoop. Le chapitre suivant sera consacré à la présentation de l'organisme d'accueil et l'étude du cas.

CHAPITRE 2

INTRODUCTION DU CAS D'ÉTUDE

1 Introduction :

Dans ce chapitre, nous allons présenter l'organisme d'accueil, en donnant un aperçu sur l'architecture de l'entreprise Tchaine-Lait Candia et son système, ainsi que les différentes machines et logiciels utilisées par l'entreprise pour comprendre le fonctionnement de cette dernière. Puis Dans la deuxième, partie nous allons exposer la problématique qui nous a motivé à choisir le sujet en question.

2 Présentation de la société d'accueil :

2.1 Présentation de l'entreprise

Implantée sur l'ancien site de la limonaderie Tchaine-Tchaine, à l'entrée de la ville de Béjaïa, Tchaine-Lait produit et commercialise le lait longue conservation UHT (Ultra Haute Température) sous le label CANDIA. Tchaine-Lait est une société privée de droit algérien, constituée juridiquement en SARL. Elle est dotée d'un capital social de 1.000.000.000 DZD, détenu majoritairement par M.Fawzi BERKATI, gérant de la société. [14]

2.2 Historique de l'entreprise :

Tchaine-Tchaine était, à l'origine, une entreprise familiale, spécialisée dans les boissons gazeuses depuis 1952. Elle a, de ce fait, capitalisé une longue expérience dans le conditionnement des produits sous forme liquide. L'arrivée des grandes firmes multinationales sur le marché des boissons gazeuses et la multiplication du nombre de limonadiers locaux l'a contraint à réviser sa stratégie ; d'où l'idée d'une reconversion vers le lait UHT, qui a donné naissance à Tchaine-Lait. [14]

2.3 Pourquoi le lait UHT :

Le lait constitue l'un des produits de base de notre alimentation. Il apparaît comme un produit indispensable à la santé, source de vie et de croissance, possédant des vertus nutritionnelles spécifiques et très bénéfiques, en particulier sa teneur en calcium. [14]

Le lait représente l'un des plus importants marchés de l'univers alimentaire. L'Algérie est l'un des plus grands importateurs mondiaux de lait ; elle représente un marché de plus de 3 milliards de litres/an, soit 100 litres/habitant/an. [14]

Le choix du procédé UHT (lait traité à Ultra Haute Température, permettant une conservation longue durée hors chaîne de froid) résulte du fait que le lait existant en Algérie est un lait frais pasteurisé, il requiert la continuité et la non rupture de la chaîne de froid, depuis son conditionnement jusqu'à sa consommation finale, en passant par son stockage et son transport. Or, la température peut atteindre les 40°C en été dans les régions Nord de l'Algérie et plus dans les régions du sud. La mise en place et le respect de la chaîne de froid nécessitent une organisation tenant compte de la courte durée de conservation du lait et une flotte dotée de camions réfrigérés, ce qui n'est pas habituellement le cas : [14]

- Le lait pasteurisé est parfois vendu au consommateur à une température deux à six fois supérieure à la température exigée par la législation (+6°C).
- la chaîne de froid est quasi inexistante chez le détaillant et souvent défaillante chez le distributeur et le producteur.

2.4 Contrat de franchise Candia/Tchin-Lait :

N'étant pas laitier de tradition, Tchin-Lait a opté pour un partenariat avec Candia, leader européen du lait. [14]

Candia, c'est 40 ans d'expérience dans le traitement et le conditionnement du lait. [14]

Ce contrat de franchise n'est rien de plus qu'un partenariat entre l'entreprise Tchin-Lait et Candia, où chacune des parties trouve son intérêt : Candia peut, grâce aux contrats de franchise, étendre le marché et la notoriété de ses produits à l'échelle internationale ; Tchin-Lait, quant à elle, peut bénéficier du savoir-faire Candia pour produire des produits de bonne qualité qui, de plus, sont déjà bien connus du marché. [14]

Voici, à titre d'exemple, certains engagements des deux parties dans ce contrat de franchise. [14]

- Engagements du Franchiseur (Candia) : L'engagement le plus important de Candia est, sans doute, le fait d'autoriser le franchisé à produire et vendre ses produits sous la marque Candia ; de ce fait, il pourra bénéficier de la notoriété internationale de cette marque.

De plus, Candia s'engage à fournir l'assistance nécessaire au franchisé, pour la fabrication et la commercialisation de ses produits en Algérie.

Cette assistance se matérialise, tant sur le plan technique que sur le plan commercial :

- Sur le plan technique :
 - Assistance au franchisé en ce qui concerne les procédés et techniques de fabrication ;
 - Faire bénéficier le franchisé de l'expérience de Candia pour le choix des équipements et des approvisionnements ;
 - Assistance au franchisé pour la mise en œuvre du contrôle de la qualité des matières premières, de l'emballage et des produits finis ;
 - Mettre à la disposition de Tchou-Lait toute innovation technologique développée par Candia (nouveaux produits, nouveaux emballages, nouvelles techniques de fabrication et de contrôle qualité, etc.).
- Sur le plan commercial/Marketing :
 - Assistance pour l'élaboration de plans marketing et campagnes de communication ;
 - Le franchisé bénéficie de l'expérience internationale en commercialisation, marketing et merchandising de Candia.

Pour assurer cette assistance, Candia s'engage à envoyer en Algérie un ou plusieurs membres qualifiés de son personnel à des périodes choisies de l'année.

Candia s'engage, aussi, à accueillir et former, dans ses usines en Europe, le personnel du franchisé.

- Engagements du Franchise(Tchin-Lait) : En plus des redevances annuelles (royalties représentant un certain pourcentage du chiffre d'affaires) et du droit d'entrée que le franchisé doit payer après la signature du contrat, le franchisé doit prendre d'autres engagements qui ont pour but de préserver l'image de marque de Candia ou de se prémunir de toute concurrence déloyale aux produits Candia.

Nous pouvons citer quelques-uns de ces engagements :

- Contrôle de la qualité : La préservation de la notoriété et de l'image de marque Candia est un point primordial pour Candia. Pour cette raison, le franchisé se doit de respecter certaines contraintes concernant le contrôle de la qualité de ses produits. En conséquence, il a été convenu que :
 - Toutes les matières premières, les ingrédients, les emballages et les accessoires utilisés par le franchisé, pour la fabrication des produits Candia, doivent être conformes aux normes de qualité prescrites par Candia.
 - Le franchisé doit s'assurer de l'accord préalable de Candia sur le choix des fournisseurs d'équipements et de matières premières, Candia se réservant le droit de refuser l'intervention de tout fournisseur s'il estime que celui-ci est de nature à compromettre le niveau de qualité ou de productivité exigée pour les produits Candia.

2.5 La laterie Tchín-Lait :

Tchin-Lait est une laiterie moderne, construite sur une superficie totale de 6.000m², comprenant :

- Un atelier de production : reconstitution du lait, traitement thermique et conditionnement. [14]
- Un laboratoire : pour analyses micro biologiques et physico-chimiques du lait. [14]
- Les utilités : Chaudières, station de traitement des eaux, compresseurs, groupes électrogènes, onduleurs, station de froid. [14]
- Administration Générale (Direction générale et administration, Direction marketing et vente, Direction qualité, Direction achats et approvisionnements, Direction finances et comptabilité). [14]
- Dépôt de stockage des produits finis, pouvant contenir près de 3 millions de litres. Ce dépôt sert aussi de plateforme d'expédition, pour la livraison des distributeurs, à travers tout le territoire national. [14]

La gamme de produits Tchín-Lait est constituée actuellement de : [14]

- Lait longue conservation : Conditionné en emballage Tetra Pak ou Combibloc 1litre.
 - Lait stérilisé UHT, partiellement écrémé, à dominante Bleue. Existe aussi en conditionnement 50cl.
 - Lait stérilisé UHT, Entier, à dominante Rouge.
 - Lait stérilisé UHT Silhouette, écrémé (sans matière grasse), à dominante verte, enrichi en vitamine D.
 - Lait stérilisé UHT Viva, partiellement écrémé, enrichi en vitamines B1, B2, B3, B5, B6, B8, B9, B12, E, D.
- Laits chocolatés :
 - Lait stérilisé UHT au chocolat, dénommé « Candy Choco », en emballage 1l et 20cl.
- Laits jus :
 - Lait additionné de jus de fruits (Orange-Ananas, Orange-fraise-banane, Orange -Mangue et Pêche-Abricot), dénommé « Twist », en emballage 20cl, avec paille.
- Poudre Instantanée : lait entier en poudre, enrichi en vitamine A et D. Contenance : étui de 500g.
- Boissons aux fruits : Conditionné en emballage Tetra Pak 20cl avec paille et en emballage Combibloc 1L.
 - Boisson à l'Orange.
 - Cocktail de fruits.

— Citronnade (Boisson au Citron) : disponible au format 1 litre seulement.

Les capacités de conditionnement actuelles sont comme suit : [14]

- Format 1l : 740.000 litres/jour.
- Format 20cl : 96.000 litres/jour (480.000 emballages 20cl).

Tchin-Lait emploie 539 personnes (dont 29 femmes). 9.46% d'entre eux sont des cadres, 39.33% des agents de maîtrise et le reste sont des agents d'exécution. [14]

2.6 La gestion de l'unité :

La gestion de l'unité est subdivisée en plusieurs directions : [14]

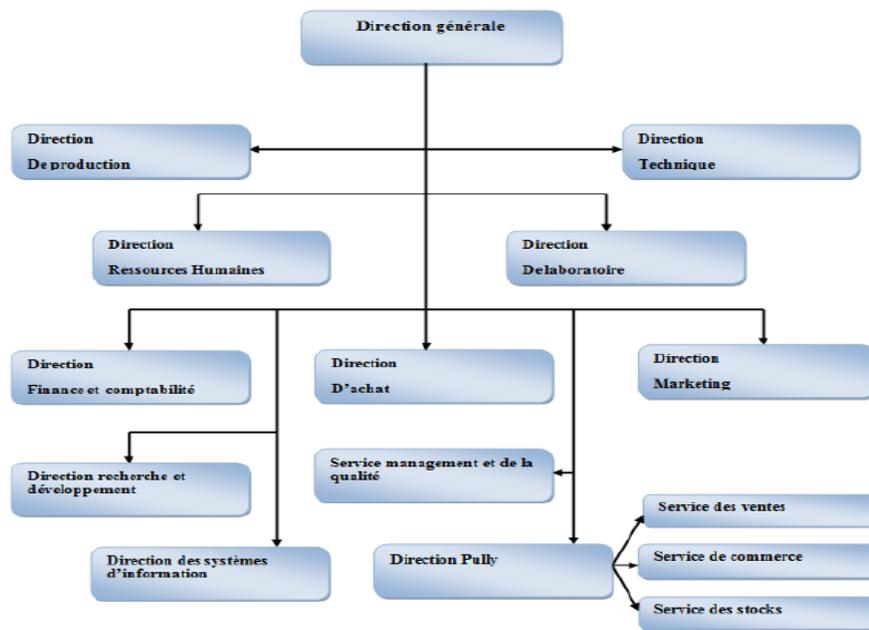


FIGURE 2.1 – Organigramme générale de Tchin-Lait [14]

2.7 Réseau de distribution :

Le réseau de distribution de Tchin-Lait est illustrée dans la figure suivante :



FIGURE 2.2 – Réseau de la distribution [14]

Tchin-Lait dispose de 51 clients distributeurs. Ils sont répartis comme suit :

Centre	Est
Alger (4)	Batna (1)
Médéa (1)	Tébessa (1)
Boumerdes (1)	Jijel (1)
Tipaza (1)	Sétif (1)
Béjaia (2)	Annaba (1)
Blida (1)	Guelma (1)
Tizi-Ouzou (1)	Constantine (2)
Bouira (1)	M'sila (1)
	Bordj Bou Arreridj (1)
	Khenchela (1)
	Mila (1)
	Oum El Bouaki (1)
	Skikda (1)
	El Taref (1)
West	Sud
Oran	Djelfa (1)
Tlemcen (1)	El Oued (1)
Aïn Timouchent (1)	Ghardaia (1)
Mascara (1)	Laghouat (1)
Mostaganem (1)	Ouargla (2)
Chlef (1)	Biskra (1)
Tiaret (1)	Béchar (1)
Sidi-Bel-Abbès (1)	Adrar (2)
Ain Defla (1)	Tindouf (1)
	Tamanrasset (1)
	Naâma - El Beyadh (1)
	Illizi (1)

TABLE 2.1 – Les centres de distribution [14]

2.8 Les missions de l'entreprise :

La mission d'entreprise est la déclaration de la raison d'être de l'entreprise et de la façon dont elle entend atteindre ses buts. Tchiv-Lait a pour mission principale de : [14]

- Mobiliser les ressources internes en motivant les employés qui peuvent s'identifier à des valeurs fortes.
- Aligner les décisions et actions prises au quotidien par l'ensemble du personnel.
- Communiquer une image forte et claire aux clients et aux actionnaires de l'entreprise.
- Forcer les managers à se poser des questions fondamentales sur les valeurs et les comportements qu'ils doivent chercher à promouvoir.

2.9 Structure de la direction du système d'information :

Le système d'information d'une entreprise est l'ensemble des actions coordonnées de recherche, de traitement, de distribution et protection des, il met les technologies informatiques et les réseaux au service du personnel et de la clientèle de l'entreprise. [14]

Le département d'informatique est composé de : [14]

- Un chef de département.
- Un administrateur réseau et système.
- Un administrateur des bases de données.
- Un ingénieur support.
- Un ingénieur réseau et système.

2.10 Architecture réseau de l'entreprise :

L'architecture réseau de Tchir-Lait est démontrée dans la figure suivante : [14]

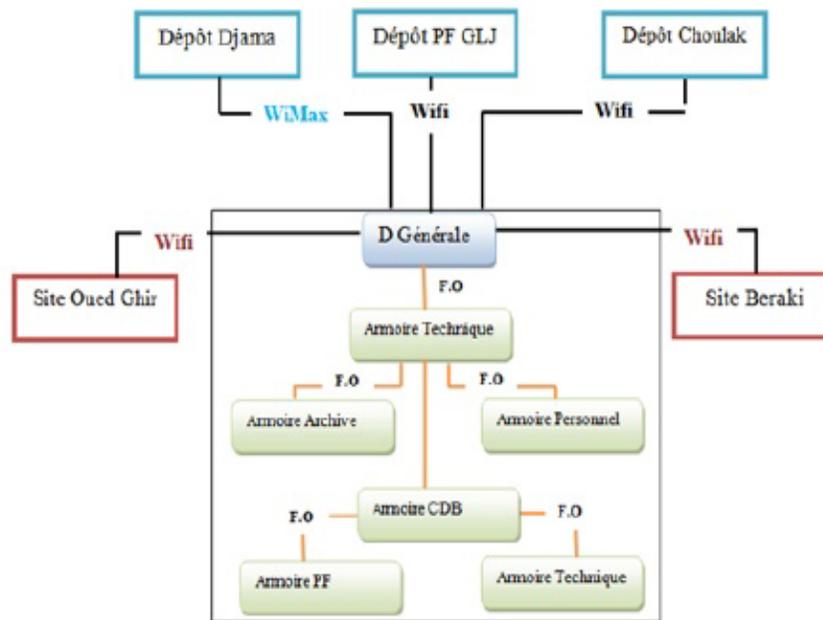


FIGURE 2.3 – Architecture réseau de Tchir-Lait [14]

2.11 Les différents serveurs du réseau de l'entreprise :

Le réseau de Tchir Lait se base sur le mode de communication (architecture) client-serveur dont plusieurs serveurs sont disponibles pour fournir des services aux différents clients de l'entreprise.[14]

Chaque serveur s'occupe des tâches spécifiques comme suit : [14]

- DC-server (Domain Control Server) : serveur contrôleur de domaine , il exécute les services de Active Directory ;
- DB-server (Data Base Server) : est un serveur de base de données, sur lequel un système de gestion de base de données (SGBD) ici SQL Server 2008 est installé ;
- TERMINAL-server : est un serveur pour les applications de sauvegardes ;
- WMS-server : est un serveur de gestion d'entrepot et de traçabilité ;
- EXCHANGE-server : est un serveur de messagerie Microsoft Exchange. Pour cette entreprise, les comptes de messagerie sont configurés par Microsoft Office Outlook ;
- PLMS-server : est un serveur pour l'application des statistiques de production ;
- KSC-server : est un serveur des applications antivirus. Pour l'entreprise Tchir Lait kaspersky est installé sur tous ses ordinateurs ;

- HHT-server : est un serveur des applications des ventes mobiles ;
- TSE-server : est un serveur de bureau à distance ;
- Serveur-DATA : est un serveur pour le partage de fichiers entre les différents services et il contient les fichiers partagés ;

2.12 Les équipements utilisés à Tchín-Lait :

Désignation	Modèle
170 Ordinateurs	Portable & Bureau
37 Imprimantes	Epson, Canon, Canon IR25
Deux modems	ADSL
Un pare-feu	Sophos

FIGURE 2.4 – Les équipements utilisés [14]

- Pour relier les différents équipements qui sont utilisés dans le réseau de l'entreprise, Tchín-Lait opte pour la fibre optique.
- Tchín-Lait est aussi dotée d'une technologie Power over Ethernet (PoE) est une technologie de réseaux locaux (LAN) Ethernet filaires qui fait passer le courant électrique nécessaire au fonctionnement de chaque appareil par les cables de données, au lieu des cordons d'alimentation.
- Tous les PC sont dotés d'un anti-virus KASPERSKY 10 end point.
- Candia dispose de 3 connexions :
 - Une connexion Wimax SLC : Un réseau Wimax composé de deux connexions internet (Algérie Télécom, Icosnet), elles sont reliées directement à un pare-feu Sophos configuré afin de garantir la haute disponibilité des dispositifs de sécurité et un controle total du flux entrant et flux sortant.
 - Deux connexions ADSL (une de 4 giga et l'autre de 8 giga)

2.13 Les applications métiers :

- Entreprise Resource Planning (ERP) : est un outil informatise qui permet le pilotage de l'entreprise. Sa particularité est d'embarquer, en un meme logiciel et une seule base de données, les fonctionnalités nécessaires à la gestion de l'ensemble de l'activité d'une entreprise : gestion comptable, gestion commerciale, gestion des stocks. [14]
- Assabil : est une solution de gestion de la force de vente mobile, elle s'adresse aux différents industriels et entreprises de distribution pour renforcer leur positionnement sur le marché et accroitre leur vente. [14]
- Logitrace : Logitrace est la dénomination du logiciel de traçabilité des produits après leur identification sur ligne de production. Logitrace, relié à l'ERP, permet de gérer : [14]
 - les différents mouvements de stock.
 - la préparation des commandes.
 - la traçabilité dans la prise d'échantillonnage.
 - la relation avec le laboratoire qualité.

2.14 Les logiciels utilisés :

- Microsoft office Word 2007, 2010, 2013, 2016. [14]
- Microsoft office Excel 2007, 2010, 2013, 2016. [14]

2.15 Les systemes d'exploitation utilisés :

- Windows server 2008 R2. [14]

3 Etude de cas : Tchén-Lait

3.1 Problématique :

La croissance continue des données numériques dans les organisations a conduit à l'émergence des Big Data.

Cependant, les technologies traditionnelles ne sont pas suffisantes pour traiter de grands volumes de données ou pour diffuser des données en temps réel.

C'est pourquoi Hadoop, un framework open source pour le calcul distribué, a été développé. Récemment, les entreprises ont commencé à adopter et à intégrer le framework Hadoop pour améliorer leurs capacités de traitement des données.

Tout cela nous a conduit à choisir notre thème et à essayer d'intégrer la société Tchén-lait Candia en tant que stagiaire afin d'accomplir les tâches/missions suivantes :

- Comprendre et analyser l'architecture actuelle de l'entrepôt de données.
- Etudier les solutions d'intégration de la technologie Hadoop dans l'environnement de l'entrepôt de données déjà en cours.
- Mettre en œuvre le processus d'extraction, de transformation et de chargement, suivi de la mise en œuvre physique de Hadoop et de la localisation du cluster Hadoop.

Cela nous amène à poser les questions suivantes :

- Comment devons-nous procéder pour gérer le stockage des données et le nombre de palettes transportées ?
- Comment procéder pour remplacer le système de stockage classique par le Big data ?
- Quelles sont les solutions à proposer pour convaincre l'entreprise de prendre le relais et de se lancer dans le Big data ?

3.2 Objectif de l'étude :

L'étude de notre thème consistera donc à faire des propositions concrètes en rapport avec la problématique évoquée ci-dessus. L'objectif de notre étude est donc la création d'un programme mapreduce pour accéder aux données stockées dans le système de fichiers Hadoop (HDFS) qui pourra répondre aux besoins. d'où la nécessité des solutions qui sont définies en conséquence :

Premièrement, le stockage des fichiers de données dans le HDFS qui consiste à :

- Eviter le risque de perdre des données, chaque donnée est stockée à trois emplacements.
- Eviter également la congestion du réseau en déplaçant les opérations plutôt que les données.
- Des dizaines de millions de fichiers peuvent être pris en charge par une seule instance.

Deuxièmement, la mise en œuvre d'un modèle de programmation mapreduce qui permet :

- Le traitement et la génération de grands ensembles de données sur des clusters d'ordinateurs.

4 Conclusion :

Au cours de ce chapitre, nous avons présenté l'ensemble des informations recueillies à Tchinalait (Candia), en premier lieu la présentation des différents produits et services. Ensuite, nous avons cité le problème et les solutions que nous avons proposées pour le résoudre.

CHAPITRE 3

MISE EN PLACE DE LA SOLUTION

1 Introduction :

Dans le chapitre précédent nous avons présenter l'organisme d'accueil. Nous allons dans ce chapitre présenter les différents outils et environnements de travail que nous avons utilisé afin de réaliser les objectifs de notre projet.

2 Présentation de l'environnement de travail :

La virtualisation désigne la technologie qui permet de créer, sur un ordinateur physique, des ordinateurs virtuels appelés machines virtuelles, sur lesquels nous pourrions installer n'importe quel système d'exploitation. Il s'agit de la manière la plus utile de diminuer les dépenses informatiques tout en excitant la performance et la flexibilité des entreprises de toute taille.

2.1 VMware Workstation 15 :

VMware est un hyperviseur de niveau 2 qui permet la création d'une ou plusieurs machines virtuelles simulant plusieurs systèmes d'exploitation au sein d'un même système d'exploitation. Celles-ci, pouvant être reliées au réseau local avec des adresses IP différentes, tout en étant sur la même machine physique qui existe réellement. Il est possible de faire fonctionner plusieurs machines virtuelles en même temps, Workstation maximise les ressources de votre ordinateur de façons à permettre l'exécution des applications plus gourmandes dans un environnement virtuel. Il permet aussi de construire des machines virtuelles complexes pour l'exécution des applications Big Data sous la forme d'un cluster Apache Hadoop. [13]



FIGURE 3.1 – VMware Workstation 15

3 Installation et configuration de Fedora 29 sur VMware Workstation 15 :

Ci-dessous se trouve le premier écran lorsque VMware Workstation s'ouvre pour créer une nouvelle machine virtuelle. On clique sur "Create New Virtual Machine".

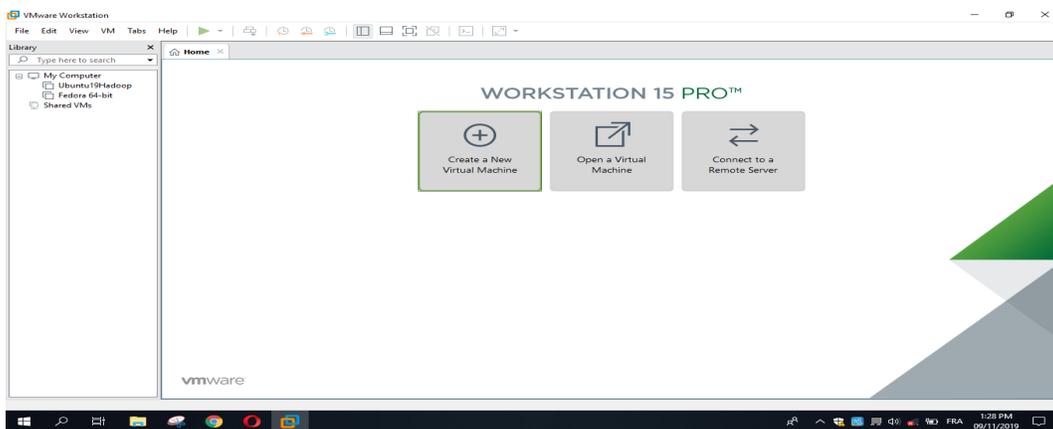


FIGURE 3.2 – Création d'une nouvelle machine virtuelle

La nouvelle fenêtre apparaît pour la création d'une machine virtuelle ; ici, on sélectionne l'option "Typical (recommended)".

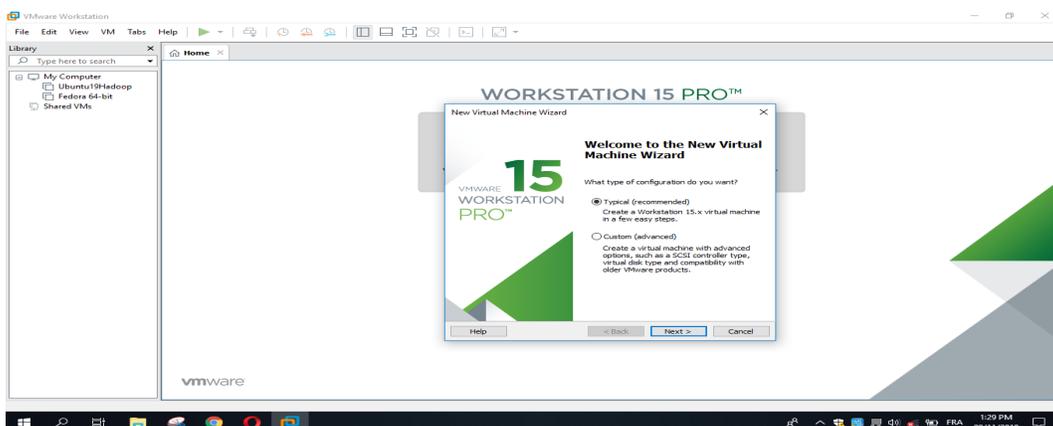


FIGURE 3.3 – Choix de type de configuration

Cette fenêtre apparaîtra pour choisir comment installer le système d'exploitation invité. on sélectionne "i will install the operating system later" et on clique sur le bouton "Next" pour poursuivre la procédure.

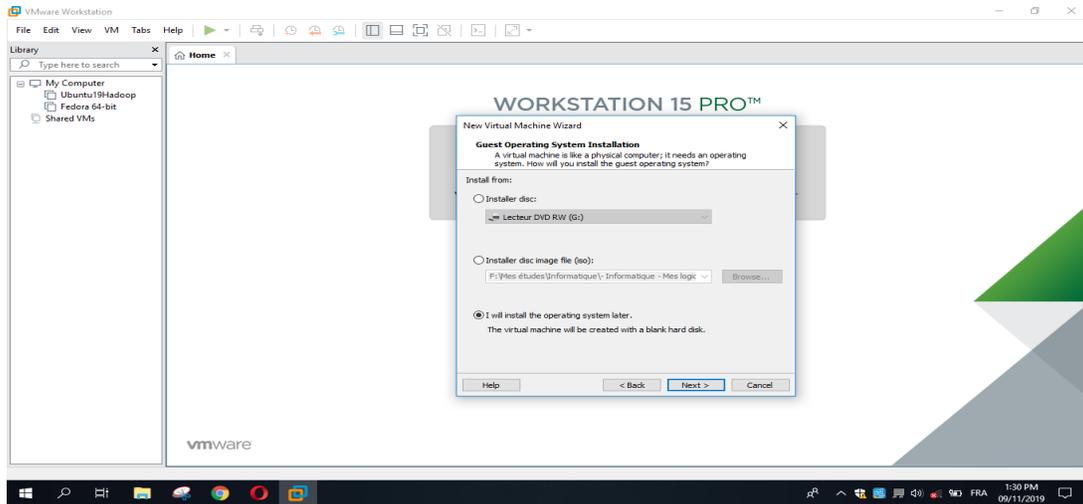


FIGURE 3.4 – Choix d'installation de système d'exploitation

Cette fenêtre apparaîtra pour choisir le système d'exploitation invité. on sélectionne "Linux" pour le system d'exploitation, "Fedora 64-bit" pour la version et on clique sur le bouton "Next" pour poursuivre la procédure.

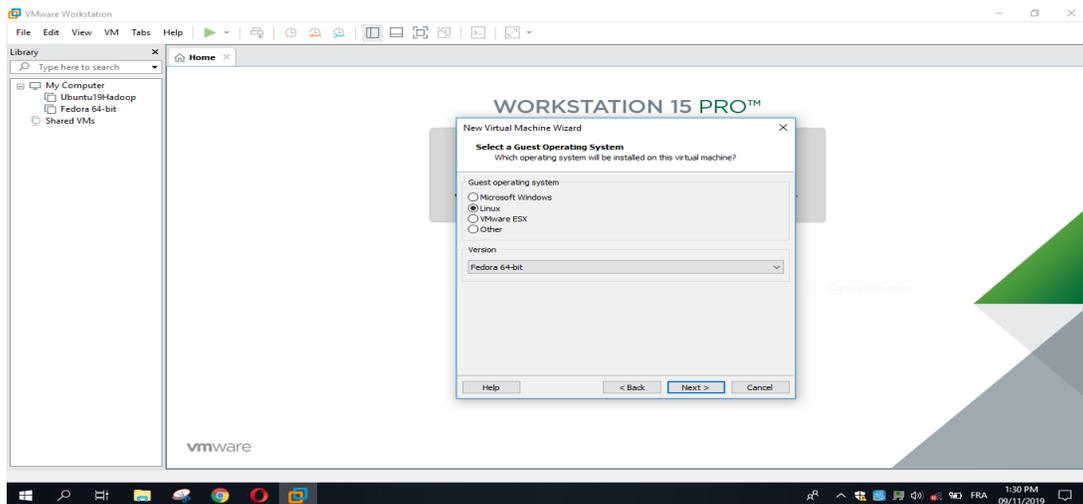


FIGURE 3.5 – Sélection de système d'exploitation

Dans cette fenêtre, nous pouvons configurer le nom de la machine virtuelle et l'emplacement de l'installation, car nous configurons un nom pour cette machine virtuelle qui est "Fedora 64 bits" dans l'image ci-dessous. Après avoir configuré le nom de la machine virtuelle, nous cliquons sur le bouton "Suivant" pour continuer le processus.

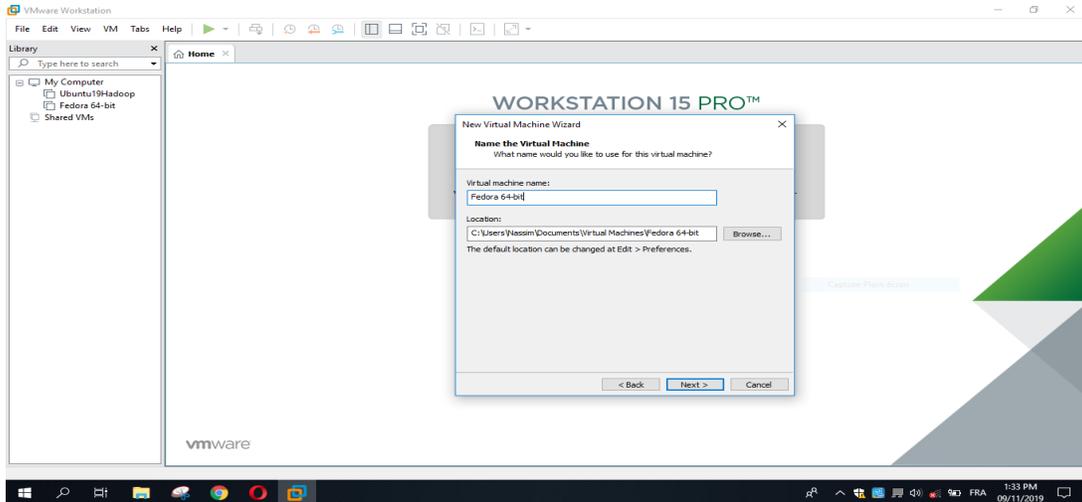


FIGURE 3.6 – Nom de la machine virtuelle

Dans cette fenêtre, nous pouvons configurer une taille de disque virtuel. Nous avons défini 100 Go pour le disque virtuel et nous avons choisi "Stocker le disque virtuel dans un seul fichier" pour utiliser un seul fichier pour le disque virtuel.

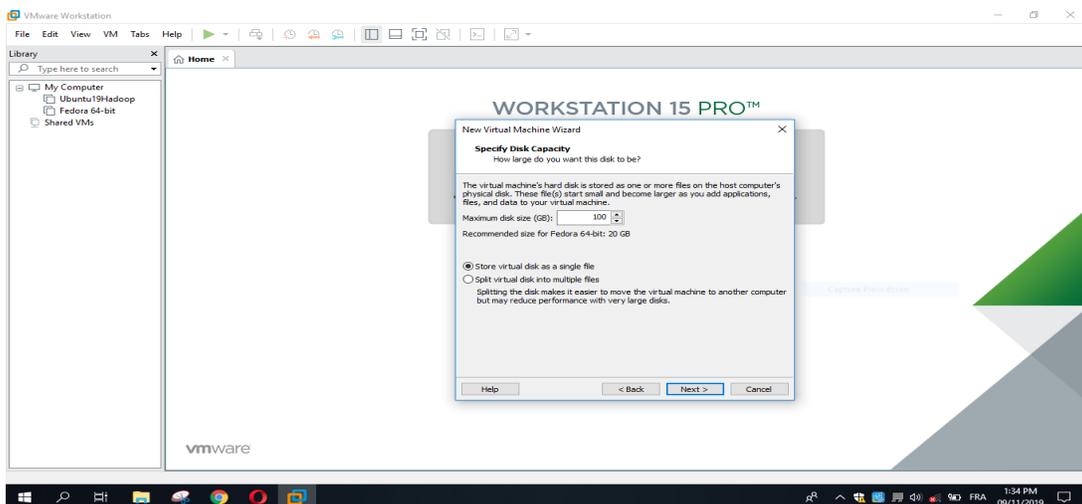


FIGURE 3.7 – Configuration de la taille de disque dur

Il est maintenant temps de configurer la mémoire vive (RAM) de la machine virtuelle.

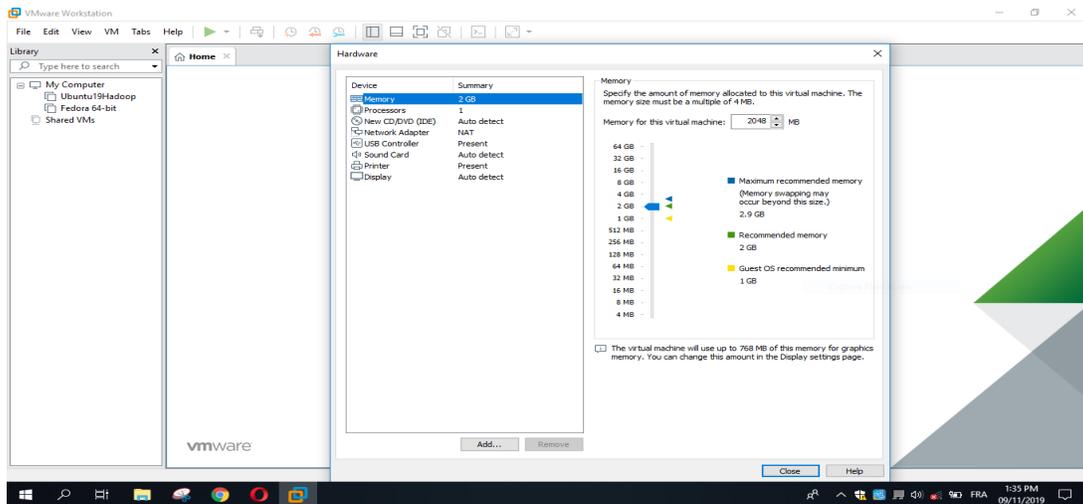


FIGURE 3.8 – Configuration de la memoire vive (RAM)

Dans la nouvelle fenêtre, nous pouvons configurer les processeurs (CPU) pour une machine virtuelle. Ici, nous pouvons sélectionner "Nombre de processeurs" et "Nombre de cœurs par processeur" et cliquer sur le bouton "Suivant" pour l'étape suivante.

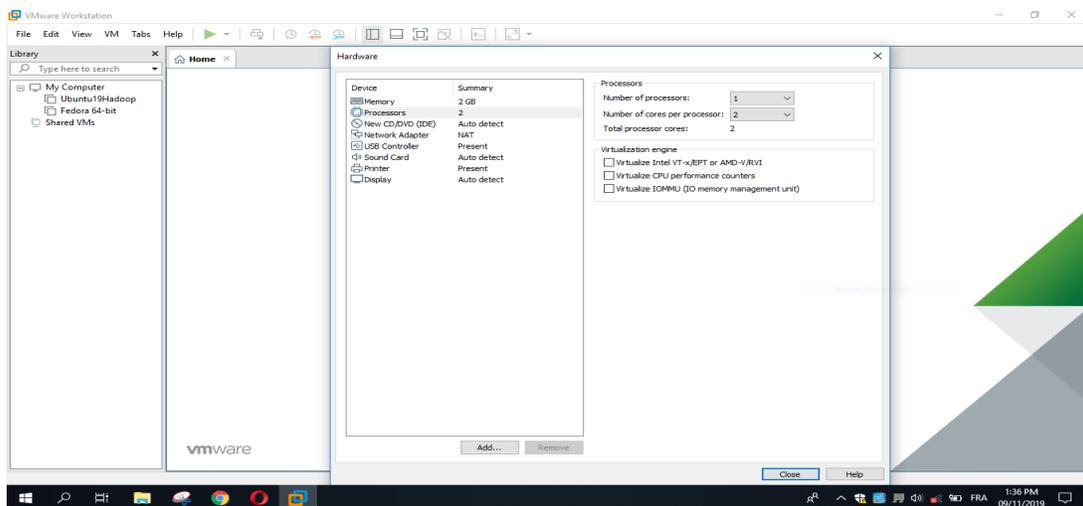


FIGURE 3.9 – Configuration des processeurs (CPU) de la machine virtuelle

La nouvelle fenêtre apparaît pour choisir le support d'installation du système d'exploitation de la nouvelle machine virtuelle. Nous sélectionnons ici "Use ISO image file" (qui est utilisé lorsque nous avons une image ISO pour le système d'exploitation et que nous allons utiliser ce fichier ISO pour l'installation dans la nouvelle machine virtuelle) et nous choisissons l'image ISO de Fedora.

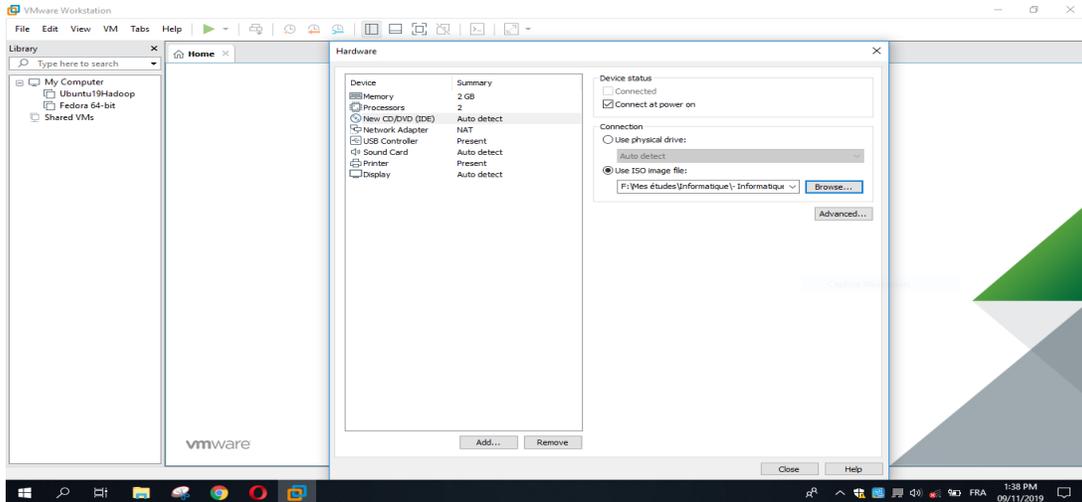


FIGURE 3.10 – Choix de support d'installation du système d'exploitation de la machine virtuelle

Dans cette fenêtre de configuration des postes de travail VMware, nous configurons le réseau pour la nouvelle machine virtuelle. Il existe plusieurs options pour configurer la mise en réseau comme ci-dessous. Nous sélectionnons "Custom" : Réseau virtuel spécifique" et nous choisissons "VMnet8(NAT)".

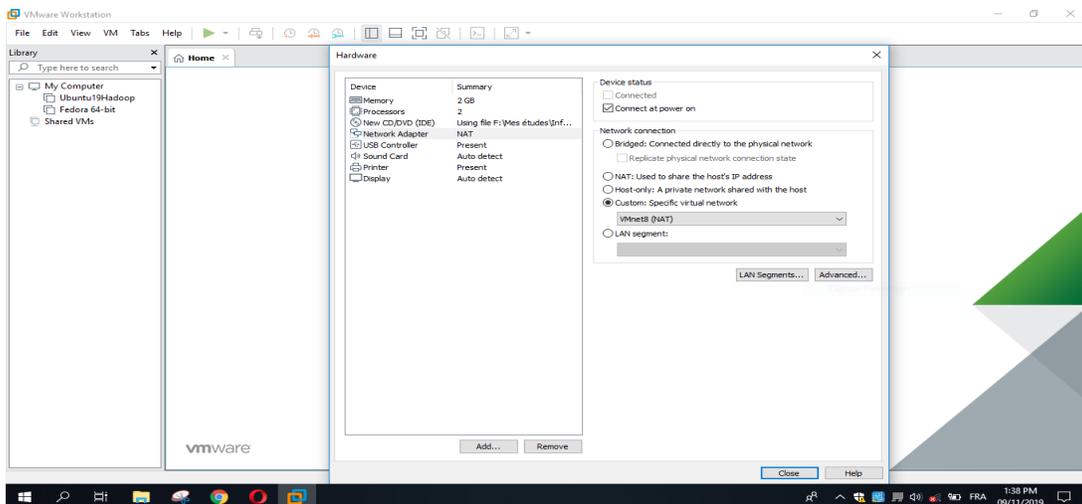


FIGURE 3.11 – Configuration de la mise en réseau de la machine virtuelle

Maintenant, l'écran ci-dessous représente le début d'une nouvelle machine virtuelle avec l'image ISO de Fedora. Dans cette fenêtre, nous sélectionnons l'option "Tester ce média installer Fedora 29".

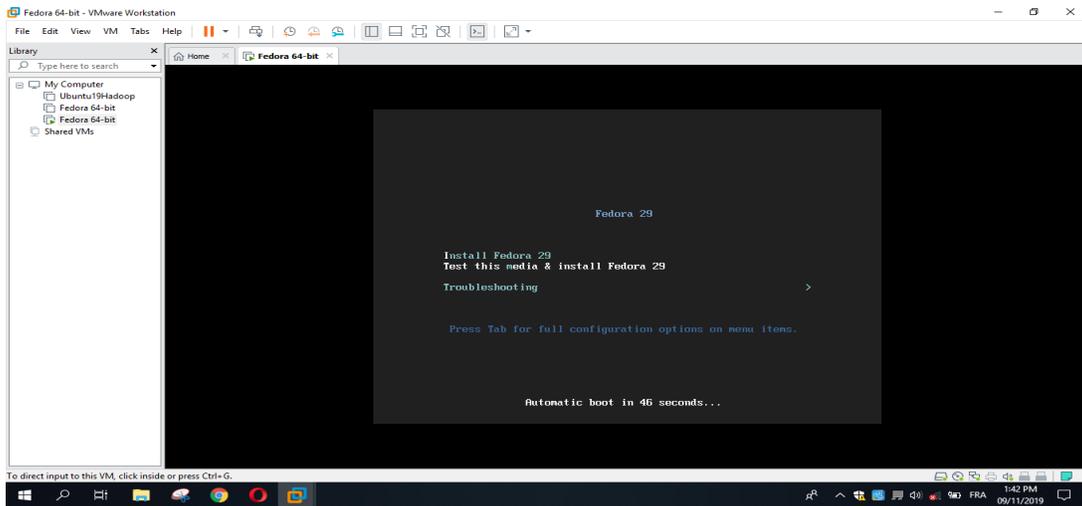


FIGURE 3.12 – Démarrage de la machine virtuelle avec l'image ISO

Affichage en dessous de l'écran lorsque le processus d'installation consiste à vérifier le support d'installation et à être prêt pour l'installation.

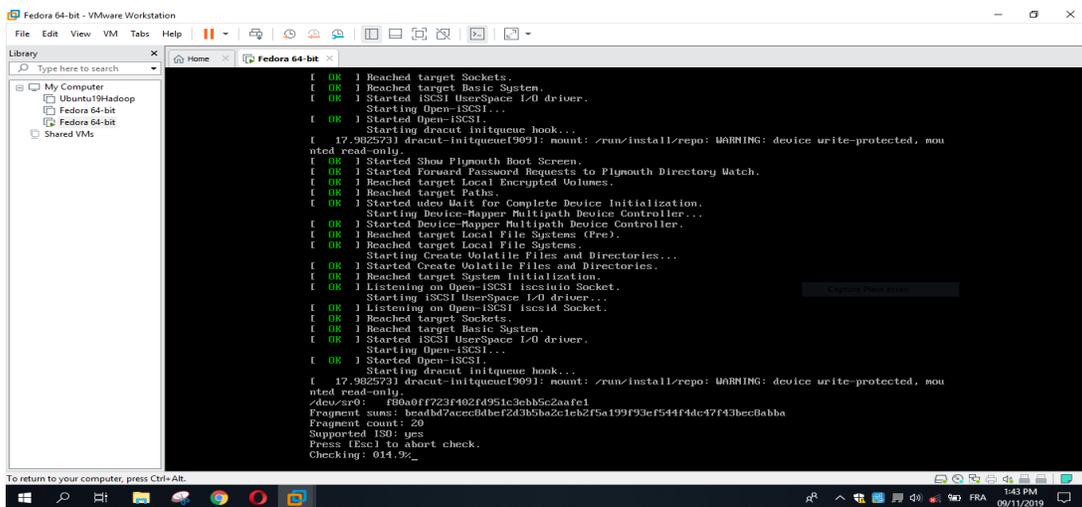


FIGURE 3.13 – Vérification de support d'installation

Ici, nous obtenons l'écran de connexion après le démarrage de la machine virtuelle ; nous nous connecterons au système en utilisant le nom d'utilisateur et le mot de passe configurés au moment de l'installation.

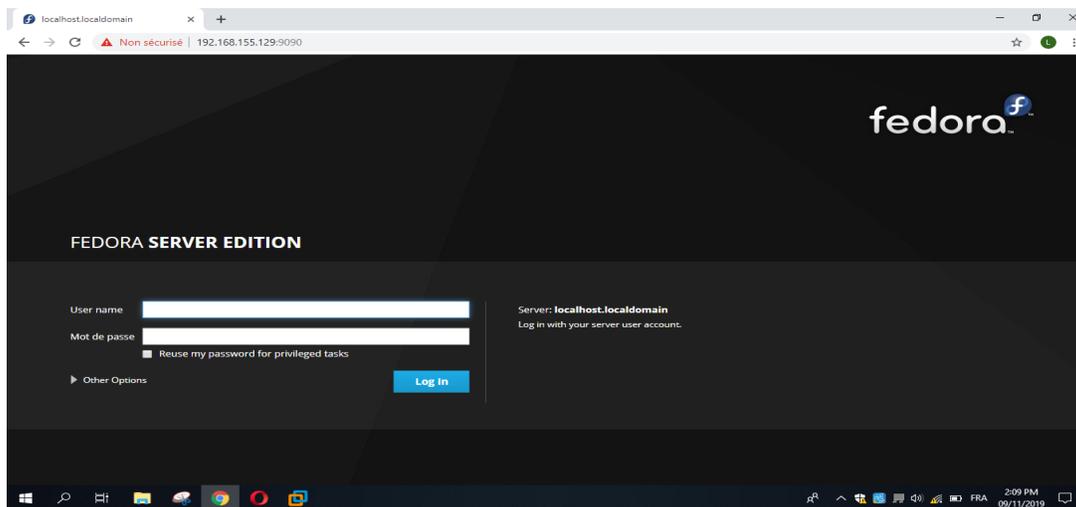


FIGURE 3.14 – La fenetre de connexion après le démarrage de la machine virtuelle

4 Installation de Docker sur fedora :

Docker est un outil qui est utilisé pour effectuer une virtualisation au niveau du système d'exploitation, appelée conteneurisation. En utilisant Docker, nous déployons des conteneurs ou des progiciels, Docker déploie des conteneurs en utilisant des images de conteneurs.

Parfois, les conteneurs Docker sont connus sous le nom de machines virtuelles légères. Docker existe en deux éditions différentes :

Docker CE (Community Edition) Il est généralement utilisé par une petite équipe de développeurs ou d'administrateurs pour les tests fonctionnels d'applications basées sur des conteneurs. Il s'agit d'une édition opensource qui peut être installée sur presque toutes les distributions Linux.

Docker EE (Enterprise Edition) Comme son nom l'indique, il s'agit d'une édition entreprise, utilisée pour exécuter des applications critiques basées sur des conteneurs. La version de Docker que nous allons installer est Docker Community Edition (CE). Afin de réussir l'installation nous avons suivi les étapes ci-dessous :

Étape 1 : Nous commençons toujours nos installations par la mise à jour et la mise à niveau des paquets OS. Sur Fedora, cela peut être fait facilement en exécutant la commande :

```

FEDORA SERVER EDITION Merri Nassim
root@localhost:~# su -
Mot de passe :
[root@localhost ~]# dnf update -y
Fedora Modular 29 - x86_64 - Updates 3.2 kB/s | 45 kB 00:13
Fedora Modular 29 - x86_64 - Updates ] --- B/s | 0 B  ---:-- ETA
    
```

FIGURE 3.15 – Installations des mise a jour sur fedora

Il est recommandé de redémarrer le système après une mise à niveau

```

FEDORA SERVER EDITION Merri Nassim
root@localhost:~# reboot
    
```

FIGURE 3.16 – Redémarrage de système

Étape 2 : Activation et configuration de dépôt Docker CE, Les commandes dnf suivantes sont exécutées pour activer et configurer le dépôt Docker CE sur le serveur Fedora 29

```

FEDORA SERVER EDITION Merri Nassim
root@localhost:~# dnf install dnf-plugins-core -y
Fedora Modular 29 - x86_64 - Updates 1.2 kB/s | 45 kB 00:38
Fedora Modular 29 - x86_64 - Updates ] 12 kB/s | 73 kB 02:56 ETA
    
```

FIGURE 3.17 – Activation et configuration de dépôt Docker CE sur le serveur Fedora 29

Cette commande permet de mettre en place le dépôt Docker stable.



```
FEDORA SERVER EDITION Merri Nassim
root@localhost:~# dnf config-manager --add-repo https://download.docker.com/linux/fedora/docker-ce.repo
Ajout du dépôt depuis : https://download.docker.com/linux/fedora/docker-ce.repo
```

FIGURE 3.18 – Mettre en place de dépôt Docker stable

Étape 3 : Maintenant, nous installons le paquet Docker CE en utilisant la commande dnf.



```
FEDORA SERVER EDITION Merri Nassim
root@localhost:~# dnf install docker-ce -y
Docker CE Stable - x86_64 245 B/s | 3.5 kB 00:14
Fedora Modular 29 - x86_64 33% [=====] 10 kB/s | 16 kB 00:03 ETA
```

FIGURE 3.19 – Installation de paquet Docker CE

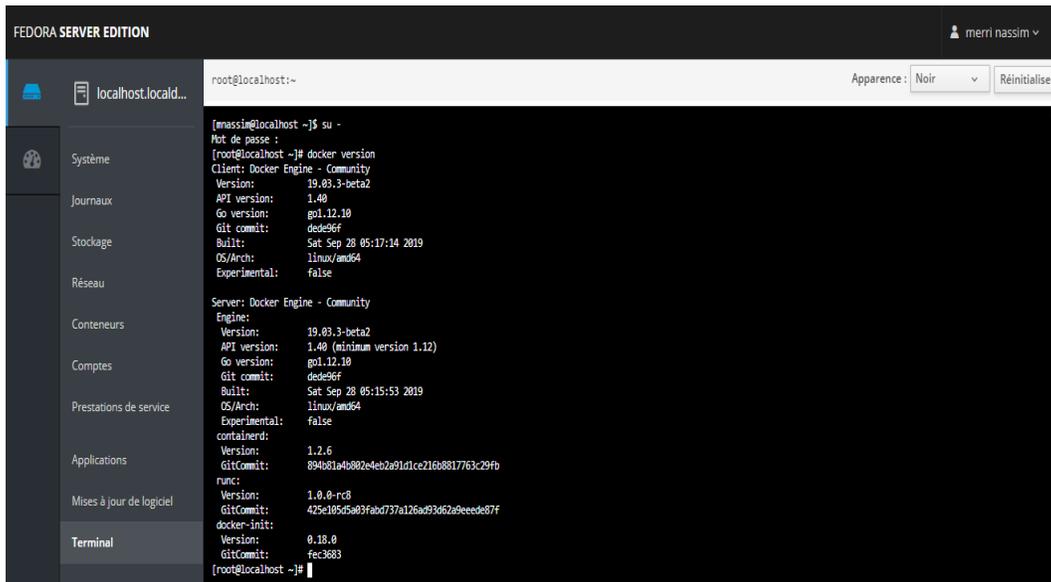
Une fois que le paquet Docker est installé avec succès, nous exécutons les commandes suivantes pour démarrer et activer le service Docker.



```
FEDORA SERVER EDITION Merri Nassim
root@localhost:~# systemctl start docker
root@localhost:~# systemctl enable docker
root@localhost:~#
```

FIGURE 3.20 – Démarrage et activation de service Docker

Étape 4 : Vérification de l'installation du docker On lance la commande "docker version" pour vérifier la version du docker.



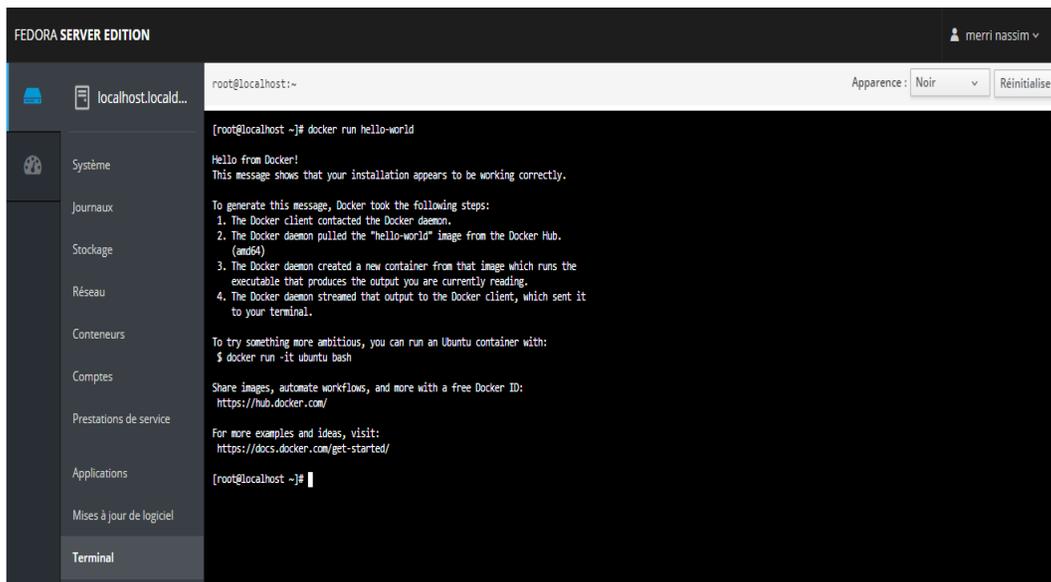
```
FEDORA SERVER EDITION
localhost.local...
root@localhost:~
Apparence: Noir Réinitialiser

[merri@localhost ~]$ su -
Mot de passe :
[root@localhost ~]# docker version
Client: Docker Engine - Community
Version: 19.03.3-beta2
API version: 1.40
Go version: go1.12.10
Git commit: dede96f
Built: Sat Sep 28 05:17:14 2019
OS/Arch: linux/amd64
Experimental: false

Server: Docker Engine - Community
Engine:
Version: 19.03.3-beta2
API version: 1.40 (minimum version 1.12)
Go version: go1.12.10
Git commit: dede96f
Built: Sat Sep 28 05:15:53 2019
OS/Arch: linux/amd64
Experimental: false
containerd:
Version: 1.2.6
GitCommit: 894814b08024eb2691d1ce21608817763c9fb
runc:
Version: 1.0.0-rc8
GitCommit: 425c18545a03fabd737a126a93462a9eede87f
docker-init:
Version: 0.18.0
GitCommit: fec3683
[root@localhost ~]#
```

FIGURE 3.21 – Vérification de l'installation du docker

Maintenant, nous pouvons vérifier que Docker a été correctement installé et fonctionne en lançant l'image Docker hello-world.



```
FEDORA SERVER EDITION
localhost.local...
root@localhost:~
Apparence: Noir Réinitialiser

[root@localhost ~]# docker run hello-world

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
   (amd64)
3. The Docker daemon created a new container from that image which runs the
   executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it
   to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/

[root@localhost ~]#
```

FIGURE 3.22 – Lancement de l'image Docker hello-world

5 Installation de Hadoop :

Le composant java est un outil indispensable pour l'exécution des travaux Hadoop, dans notre cas nous avons installé la version de java qui est compatible avec la version de Docker 19.3.3-beta2.



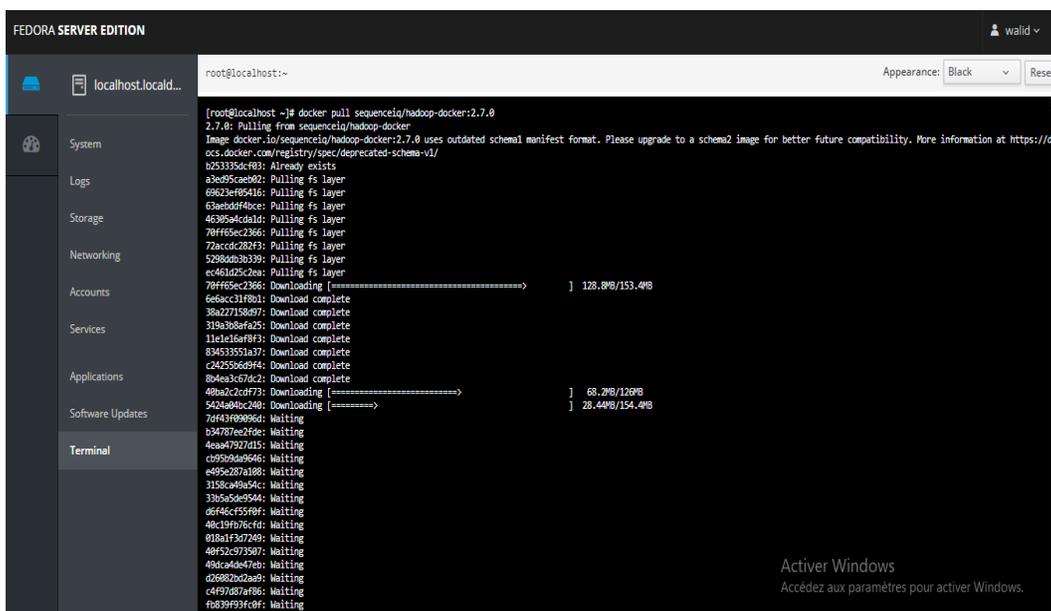
```

root@localhost:~# java -version
java version "1.7.0_71"
Java(TM) SE Runtime Environment (build 1.7.0_71-b14)
Java HotSpot(TM) 64-Bit Server VM (build 24.71-b01, mixed mode)
bash-4.1#

```

FIGURE 3.23 – La version de java installé sur fedora

Nous pouvons maintenant installer Hadoop dans un conteneur de docker. Pour ce faire, nous lançons une commande pull pour obtenir l'image du docker sur Hadoop. Plus précisément, ce que nous allons voir est une image du docker, un fichier multicouche que nous utiliserons pour déployer les conteneurs.



```

root@localhost:~# docker pull sequenceia/hadoop-docker:2.7.0
2.7.0: Pulling from sequenceia/hadoop-docker
Image docker.io/sequenceia/hadoop-docker:2.7.0 uses outdated schema1 manifest format. Please upgrade to a schema2 image for better future compatibility. More information at https://docs.docker.com/registry/spec/deprecated-schema-v1/
b23335dcf03: Already exists
a3e05fcae02: Pulling fs layer
69623ef05416: Pulling fs layer
63a0d040ce: Pulling fs layer
46306a4c016: Pulling fs layer
70ff05ec2366: Pulling fs layer
72acdc282f3: Pulling fs layer
5280db30339: Pulling fs layer
ec461d252ea: Pulling fs layer
70ff05ec2366: Downloading [=====] 128.80B/153.40B
6688cc31f01: Download complete
38a22715807: Download complete
319a3080fa25: Download complete
11e1e160f03: Download complete
83453351a37: Download complete
c2825506094: Download complete
80a3a307022: Download complete
40ba2c2c0f73: Downloading [=====] 68.20B/129B
542a040c240: Downloading [=====] 28.440B/154.40B
7d43f09096d: Waiting
b34787ee2fe: Waiting
4eaa7927d15: Waiting
c0909a0a966: Waiting
e495e267a108: Waiting
3158ca49a54c: Waiting
3305a5de9544: Waiting
d6f46cf55f0f: Waiting
0b1397076cfd: Waiting
018a1f307099: Waiting
40f52c073507: Waiting
494ca0de47eb: Waiting
d260820d2aa9: Waiting
c4f97087af06: Waiting
fd839f93fc0f: Waiting

```

FIGURE 3.24 – Installation de hadoop sur docker

Ensuite, nous lançons la commande ci-dessous pour vérifier la liste des images de docker présentes sur notre système.

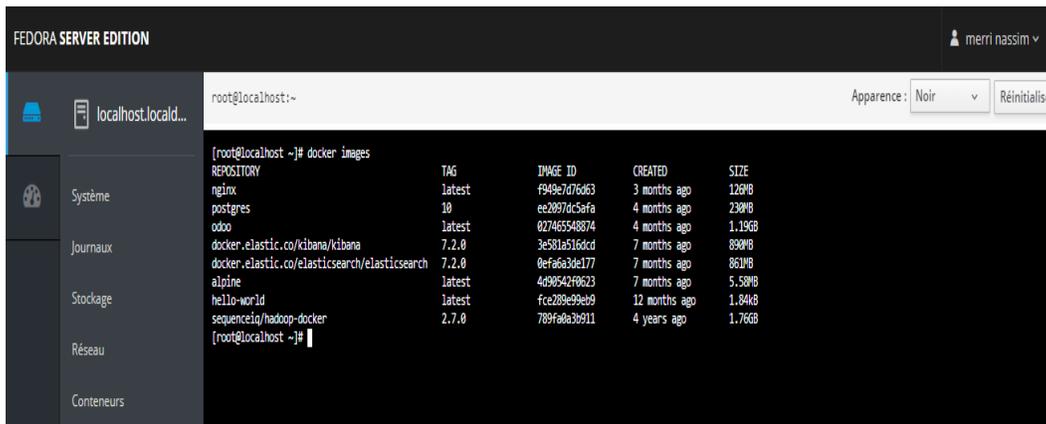


FIGURE 3.25 – Vérification de la liste des images de docker présentes sur votre système

Une fois cela fait, nous lançons l'image du docker Hadoop à l'intérieur d'un conteneur de docker.

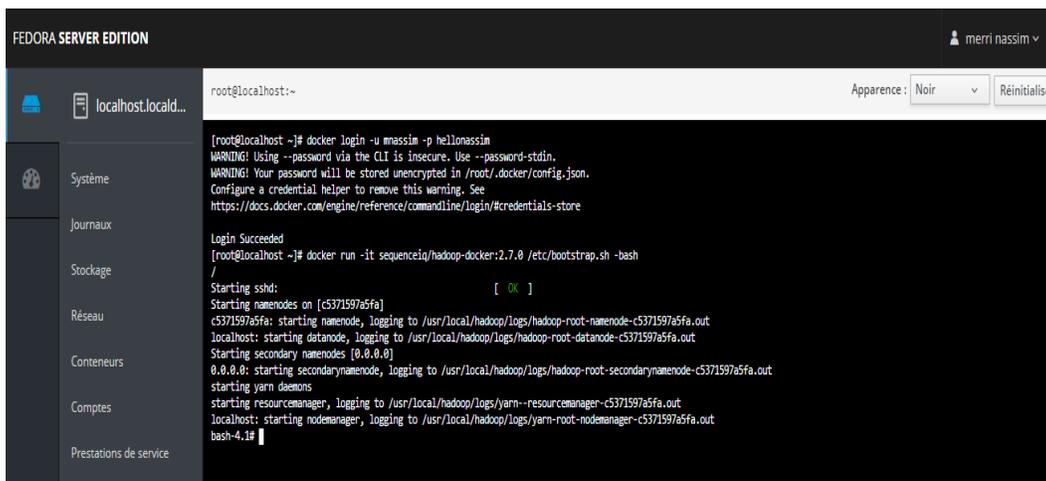


FIGURE 3.26 – Lancement de l'image du docker Hadoop à l'intérieur d'un conteneur de docker

Dans le résultat ci-dessus, on peut voir que le conteneur lance un par un tous les démons de Hadoop. Pour s'assurer que tous les démons sont lancés, nous lançons la commande `jps`.

```

FEDORA SERVER EDITION
localhost.locald... root@localhost:~
Apparence : Noir Réinitialiser

bash-4.1# jps
564 ResourceManager
354 SecondaryNameNode
972 Jps
219 DataNode
139 NameNode
659 NodeManager
bash-4.1#
    
```

FIGURE 3.27 – Vérification des demons hadoop

Si nous obtenons le résultat ci-dessus après avoir exécuté la commande `jps`, nous pouvons être sûrs que tous les démons hadoop fonctionnent correctement. Ensuite, nous exécutons la commande `docker ps` ci-dessous pour obtenir les détails sur le conteneur docker.

```

FEDORA SERVER EDITION
localhost.locald... root@localhost:~
Apparence : Noir Réinitialiser

[root@localhost ~]# docker ps
CONTAINER ID        IMAGE               COMMAND             NAMES             CREATED          STATUS          PORTS
9170e0ea947        sequenceiq/hadoop-docker:2.7.0   "/etc/docker/strap.sh -"   4 minutes ago    Up 3 minutes    2122/tcp, 8030-8033/tcp, 8040/tcp, 8042/tcp, 8080/tcp, 19080/tcp
, 49707/tcp, 50010/tcp, 50020/tcp, 50070/tcp, 50075/tcp, 50090/tcp   distracted_boyd
[root@localhost ~]#
    
```

FIGURE 3.28 – Obtention des details sur le conteneur docker

Nous exécutons la commande suivante pour obtenir l'adresse IP sur laquelle tourne le conteneur.

```

FEDORA SERVER EDITION
localhost.locald... root@localhost:~
Apparence : Noir Réinitialiser

bash-4.1# ifconfig
eth0: Link encap:Ethernet  Haddr: 02:42:4C:11:00:03
      inet addr:172.17.0.3  Bcast:172.17.255.255  Mask:255.255.0.0
      UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
      RX packets:42  errors:0  dropped:0  overruns:0  frame:0
      TX packets:30  errors:0  dropped:0  overruns:0  carrier:0
      collisions:0  txqueuelen:0
      RX bytes:2842 (2.7 KiB)  TX bytes:1906 (1.8 KiB)

lo: Link encap:Local Loopback
      inet addr:127.0.0.1  Mask:255.0.0.0
      UP LOOPBACK RUNNING  MTU:65536  Metric:1
      RX packets:665  errors:0  dropped:0  overruns:0  frame:0
      TX packets:665  errors:0  dropped:0  overruns:0  carrier:0
      collisions:0  txqueuelen:1000
      RX bytes:97910 (95.6 KiB)  TX bytes:97910 (95.6 KiB)

bash-4.1#
    
```

FIGURE 3.29 – Obtention de l'adresse IP sur laquelle le conteneur fonctionne

6 Conclusion :

Au cours de ce chapitre, nous avons implémenté la mise en place de la solution. En premier lieu, nous avons présenté les étapes nécessaires à l'Installation et configuration de Fedora server sur VMware Workstation, ensuite nous avons mis en évidence les étapes d'installation de Docker sur fedora, enfin nous sommes passé à l'installation et la configuration de Hadoop sur Docker.

CHAPITRE 4

DÉPLOIEMENT ET EXPLOITATION DE LA SOLUTION

1 Introduction :

Ce chapitre est consacré au déploiement et au fonctionnement de la solution. Nous présenterons d'abord les données utilisées. Ensuite, nous décrirons les étapes d'insertion du fichier de données dans le hdfs et le programme mapreduce pour la réalisation de notre projet. Enfin, nous présenterons les étapes de la compilation et de l'exécution de ce programme afin d'obtenir les résultats finaux.

2 Importation des données :

- Nous avons sélectionné les données pour le travail dans le projet et sont de type Excel.
- Dans notre cas, nous choisirons la destination A.

Voyages	Destinations / nombre palettes				
	A	B	C	D	..
V1	47	34	58	47	
V2	63	60	47	48	
V3	51	68	41	59	
V4	55	59	69	68	
V5	50	67	74	57	
V6	48	54	75	49	
V7	100	51	76	58	
V8	80	25	84	62	
V9	69	100	95	48	
V10	66	147	68	100	
V11	80	59	57	14	
V12	70	83	48	150	
V13	63	72	156	174	
V14	48	96	258	80	
V15	59	174	479	70	
V16	44	93	96	70	
V17	50	60	85	86	
V18	60	48	85	96	
..					

FIGURE 4.1 – Fichier Excel des données

3 Insertion de données dans le HDFS :

Les étapes mentionnées ci-dessous sont suivies pour insérer le fichier requis dans le système de fichiers Hadoop (HDFS).

- **Étape 1** : la création d'un répertoire d'entrée :

```

merri@nassim: ~
Nassim:~$ hadoop fs -mkdir /data1
Nassim:~$
    
```

FIGURE 4.2 – Création d'un répertoire d'entrée

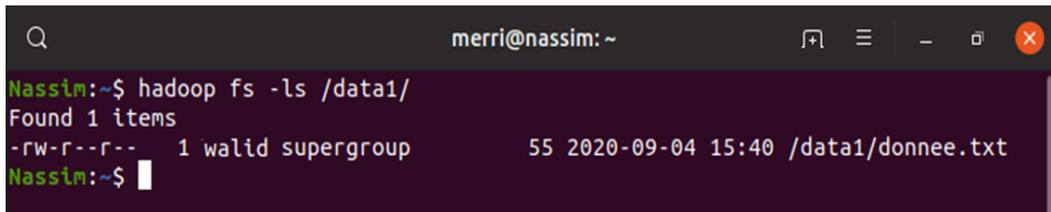
- **Étape 2** : Nous stockons le fichier de données des systèmes locaux dans le HDFS en utilisant la commande suivante dans le terminal :

```

merri@nassim: ~
Nassim:~$ hadoop fs -copyFromLocal '/home/walid/Bureau/dos1/donnee.txt' /data1
Nassim:~$
    
```

FIGURE 4.3 – Stockage de fichier de données dans HDFS

- **Etape 3** : Vérification du fichier à l'aide de la commande suivante :



```

merri@nassim: ~
Nassim:~$ hadoop fs -ls /data1/
Found 1 items
-rw-r--r-- 1 walid supergroup      55 2020-09-04 15:40 /data1/donnee.txt
Nassim:~$

```

FIGURE 4.4 – Vérification du fichier enregistré

4 Programme MapReduce :

Nous allons commencer, à partir de l'API Hadoop de MapReduce, par écrire le code correspondant à :

- **Opération Map** : Nous utilisons ici les types LongWritable, IntWritable et Text de Hadoop. En java, nous écrivons la classe EOMapper qui implémente l'opération Map.
- **Opération Reduce** : Nous utilisons ici les types IntWritable et Text de Hadoop. nous écrivons la classe EOReducer qui implémente l'opération Reduce.
- Nous pouvons ensuite écrire le code correspondant au EODriver ainsi que la méthode main.

4.1 Code Source :

Nous allons examiner le code exécutable ci-dessous :

```

import java.io.IOException;
import java.util.Iterator;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapred.MapReduceBase;
import org.apache.hadoop.mapred.Mapper;
import org.apache.hadoop.mapred.OutputCollector;
import org.apache.hadoop.mapred.Reporter;
import org.apache.hadoop.mapred.Reducer;
import org.apache.hadoop.conf.Configured;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.mapred.FileInputFormat;
import org.apache.hadoop.mapred.FileOutputFormat;

```

```

import org.apache.hadoop.mapred.JobClient;
import org.apache.hadoop.mapred.JobConf;
import org.apache.hadoop.util.Tool;
import org.apache.hadoop.util.ToolRunner;

public class palette {

public static class EOMapper extends MapReduceBase implements Mapper
<LongWritable, Text, Text, IntWritable> {

@Override
//Map function

public void map(LongWritable key, Text value, OutputCollector
<Text, IntWritable> output, Reporter rep) throws IOException {

    String data[] = value.toString().split(" ");

        for (String num : data)
        {

            int numner = Integer.parseInt(num):
            output.Collect(new Text("Nombre de palettes = "),
            new IntWritable(number));

        }
    }
}

public static class EOReducer extends MapReduceBase implements
Reducer <Text, IntWritable, Text, IntWritable> {

@Override
//Reduce Function

public void reduce(Text key, Iterator <IntWritable> value, OutputCollector
<Text, IntWritable> output, Reporter rep) throws IOException {

```

```

int sum = 0, count = 0;

    while (value.hasNext())
        {

            IntWritable i = value.next();
            sum += i.get();
            count++;
            output.collect(new Text("Nombre de Voyages = "), new IntWritable(count));
            output.collect(key, new IntWritable(sum));

        }

}

}

```

```

public static class EODriver extends Configured implements Tool {

    @Override
    //Driver Function

    public int run(String[] args) throws Exception {

        if (args.length < 2)
            {

                System.out.println("Please enter valid arguments");
                return -1;

            }

        JobConf conf = new JobConf(EODriver.class);
        FileInputFormat.setInputPaths(conf, new Path(args[0]));
        FileOutputFormat.setOutputPath(conf, new Path(args[1]));
        conf.setMapperClass(EOMapper.class);
        conf.setReducerClass(EOReducer.class);
        conf.setMapOutputKeyClass(Text.class);
        conf.setMapOutputValueClass(IntWritable.class);

```

```
    conf.setOutputKeyClass(Text.class);
    conf.setOutputValueClass(IntWritable.class);
    JobClient.runJob(conf);
    return 0;
}
}

//Main Method

public static void main(String args[]) throws Exception {

    int exitcode = ToolRunner.run(new EODriver(), args);
    System.out.println(exitcode);

}
}
```

4.2 Compilation du programme :

1) Compilation :



```
merri@nassim: ~/Bureau/dos1
Nassim:~$ cd Bureau
Nassim:~/Bureau$ cd dos1
Nassim:~/Bureau/dos1$ javac -classpath /home/walid/HadoopAdmin/hadoop-2.8.1/share/hadoop/common/hadoop-common-2.8.1.jar:/home/walid/HadoopAdmin/hadoop-2.8.1/share/hadoop/mapreduce/hadoop-mapreduce-client-core-2.8.1.jar:/home/walid/HadoopAdmin/hadoop-2.8.1/share/hadoop/commons-cli-1.2.jar: palette.java
Nassim:~/Bureau/dos1$
```

FIGURE 4.5 – Compilation du programme Mapreduce

2) Création d'un fichier jar :



```

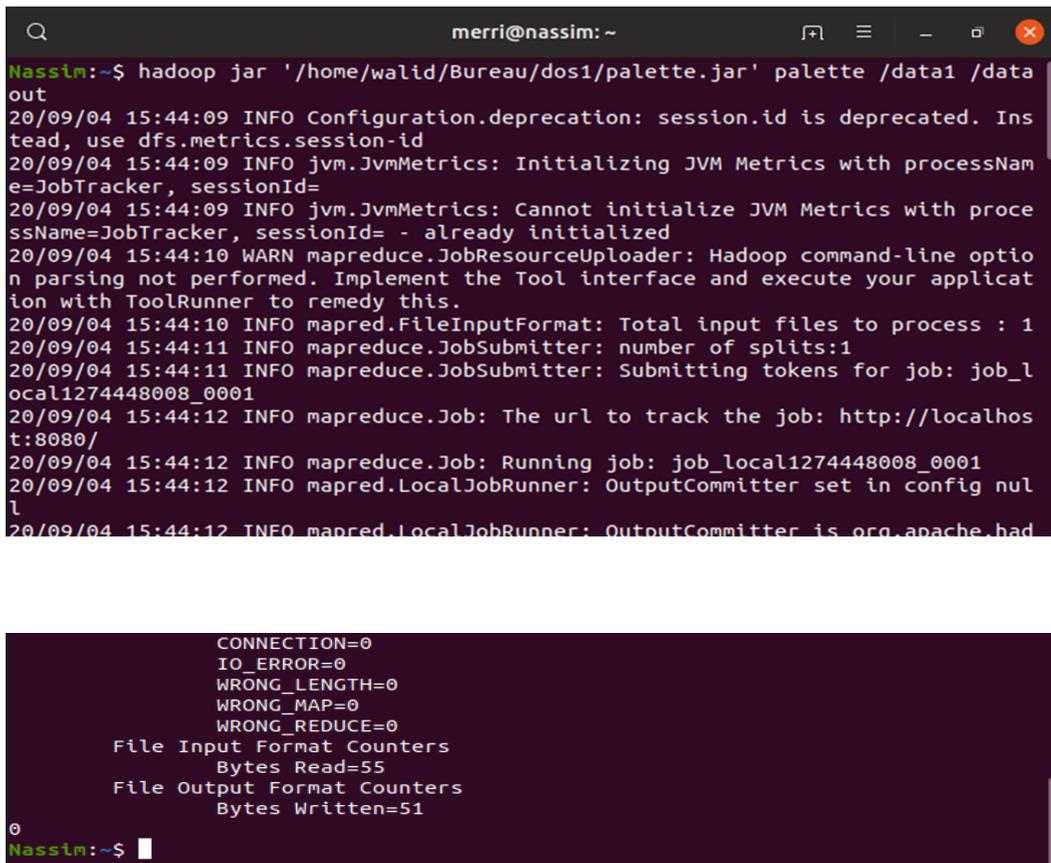
merri@nassim: ~/Bureau/dos1
Nassim:~/Bureau/dos1$ jar -cvf palette.jar *.class
manifeste ajouté
ajout : palette$EODriver.class(entrée = 1632) (sortie = 846)(compression : 48 %
)
ajout : palette$EOMapper.class(entrée = 1903) (sortie = 778)(compression : 59 %
)
ajout : palette$EOReducer.class(entrée = 1702) (sortie = 719)(compression : 57
%)
ajout : palette.class(entrée = 710) (sortie = 451)(compression : 36 %)
Nassim:~/Bureau/dos1$

```

FIGURE 4.6 – Création d'un fichier JAR

4.3 Exécution du programme :

le programme est exécuté comme suit :



```

merri@nassim: ~
Nassim:~$ hadoop jar '/home/walid/Bureau/dos1/palette.jar' palette /data1 /data
out
20/09/04 15:44:09 INFO Configuration.deprecation: session.id is deprecated. Ins
tead, use dfs.metrics.session-id
20/09/04 15:44:09 INFO jvm.JvmMetrics: Initializing JVM Metrics with processNam
e=JobTracker, sessionId=
20/09/04 15:44:09 INFO jvm.JvmMetrics: Cannot initialize JVM Metrics with proce
ssName=JobTracker, sessionId= - already initialized
20/09/04 15:44:10 WARN mapreduce.JobResourceUploader: Hadoop command-line optio
n parsing not performed. Implement the Tool interface and execute your applicat
ion with ToolRunner to remedy this.
20/09/04 15:44:10 INFO mapred.FileInputFormat: Total input files to process : 1
20/09/04 15:44:11 INFO mapreduce.JobSubmitter: number of splits:1
20/09/04 15:44:11 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_l
ocal1274448008_0001
20/09/04 15:44:12 INFO mapreduce.Job: The url to track the job: http://localhos
t:8080/
20/09/04 15:44:12 INFO mapreduce.Job: Running job: job_local1274448008_0001
20/09/04 15:44:12 INFO mapred.LocalJobRunner: OutputCommitter set in config nul
l
20/09/04 15:44:12 INFO mapred.LocalJobRunner: OutputCommitter is org.apache.had

```

```

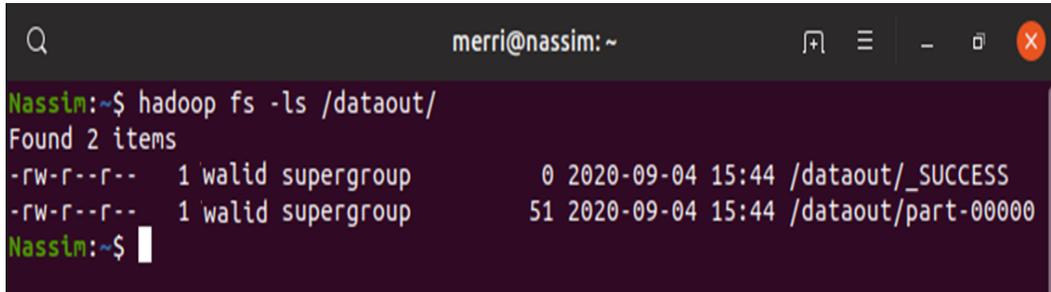
CONNECTION=0
IO_ERROR=0
WRONG_LENGTH=0
WRONG_MAP=0
WRONG_REDUCE=0
File Input Format Counters
  Bytes Read=55
File Output Format Counters
  Bytes Written=51
Nassim:~$

```

FIGURE 4.7 – Exécution du programme Mapreduce

4.4 Obtention des résultats :

- 1) Le contenu du répertoire de sortie :

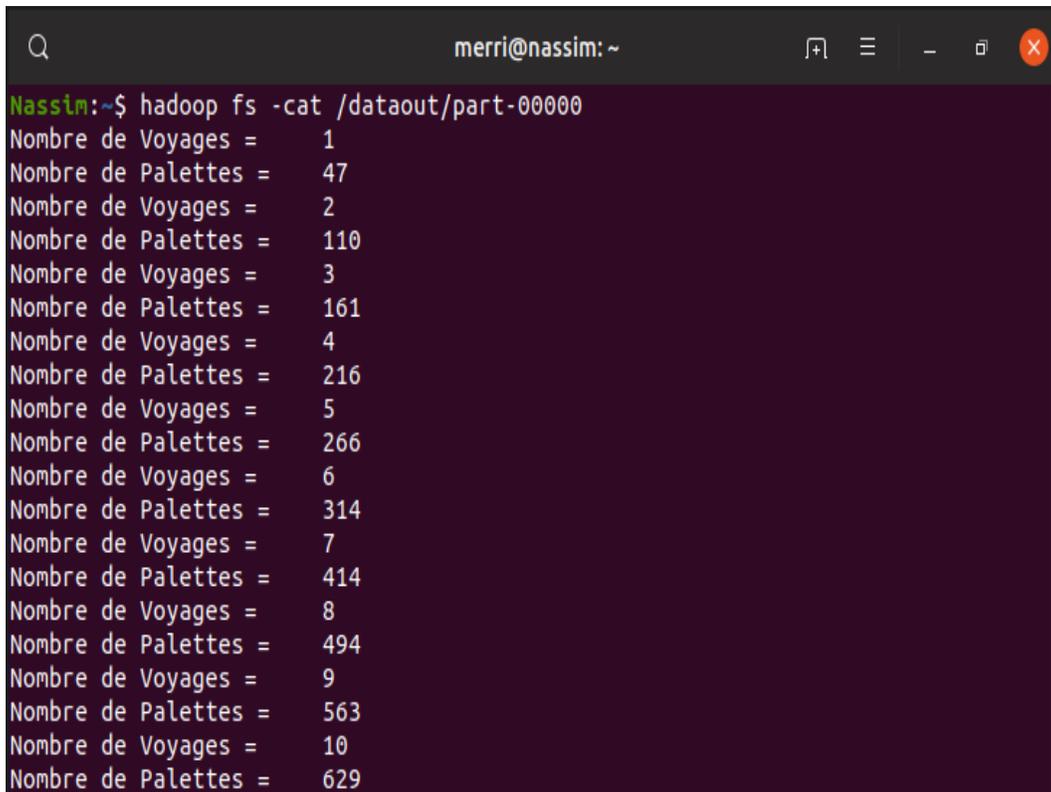


```
merri@nassim: ~  
Nassim:~$ hadoop fs -ls /dataout/  
Found 2 items  
-rw-r--r--  1 walid supergroup      0 2020-09-04 15:44 /dataout/_SUCCESS  
-rw-r--r--  1 walid supergroup    51 2020-09-04 15:44 /dataout/part-00000  
Nassim:~$
```

FIGURE 4.8 – Le contenu du répertoire de sortie

- 2) Enfin, examinons les résultats :

- Les résultats sont le cumul du nombre de voyages et le nombre de palettes par Client (la destination A).



```
merri@nassim: ~  
Nassim:~$ hadoop fs -cat /dataout/part-00000  
Nombre de Voyages = 1  
Nombre de Palettes = 47  
Nombre de Voyages = 2  
Nombre de Palettes = 110  
Nombre de Voyages = 3  
Nombre de Palettes = 161  
Nombre de Voyages = 4  
Nombre de Palettes = 216  
Nombre de Voyages = 5  
Nombre de Palettes = 266  
Nombre de Voyages = 6  
Nombre de Palettes = 314  
Nombre de Voyages = 7  
Nombre de Palettes = 414  
Nombre de Voyages = 8  
Nombre de Palettes = 494  
Nombre de Voyages = 9  
Nombre de Palettes = 563  
Nombre de Voyages = 10  
Nombre de Palettes = 629
```

```
Nombre de Voyages = 11
Nombre de Palettes = 709
Nombre de Voyages = 12
Nombre de Palettes = 779
Nombre de Voyages = 13
Nombre de Palettes = 842
Nombre de Voyages = 14
Nombre de Palettes = 890
Nombre de Voyages = 15
Nombre de Palettes = 949
Nombre de Voyages = 16
Nombre de Palettes = 993
Nombre de Voyages = 17
Nombre de Palettes = 1043
Nombre de Voyages = 18
Nombre de Palettes = 1103
Nassim:~$
```

FIGURE 4.9 – Les résultats de l'exécution

5 Conclusion :

Dans ce chapitre, nous avons mis en œuvre le déploiement et l'exploitation de la solution. Tout d'abord, nous avons présenté le fichier de données utilisé et les étapes nécessaires à l'insertion des données dans le hdfs. Ensuite, nous avons décrit le programme mapreduce que nous avons créé. Enfin, nous avons présenté les étapes nécessaires à la compilation et à l'exécution de ce programme pour obtenir les résultats finaux.

CONCLUSION GÉNÉRALE

Ce travail nous a permis d'acquérir une expérience personnelle et professionnelle remarquable. Nous avons fait progresser nos connaissances et nos compétences au niveau de configuration dans un environnement virtuel, qui est VMware Workstation. En outre, nous avons amélioré nos connaissances dans les technologies big data grâce à la mise en place d'un cluster hadoop automatisé sur docker.

En effet, nous avons présenté un travail divisé en deux parties, dont la première est axée sur l'aspect théorique et qui comprend les deux premiers chapitres. Le premier portait sur les systèmes d'information au cœur des métiers, afin de mettre en évidence les systèmes d'information numériques et les principales applications métiers exploitées par les entreprises. Puis le second chapitre, intitulé "Introduction à l'étude de cas", s'est concentré sur l'étude consacrée à l'établissement d'accueil, son système d'information, et son système informatique.

L'aspect pratique a fait l'objet de la deuxième partie, elle-même composée de deux chapitres, dont le premier traite de la mise en œuvre de la solution. Cela nous a permis de présenter les différents outils et environnements de travail que nous avons utilisés pour atteindre les objectifs de notre projet. Le second chapitre, consacré au déploiement et à l'exploitation de la solution, nous a permis d'apporter des améliorations en tant que solution. Tout d'abord, la présentation du fichier de données utilisé et les étapes nécessaires à l'insertion des données dans le hdfs. Ensuite, la création du programme mapreduce. Enfin, la présentation des étapes nécessaires à la compilation et à l'exécution de ce programme pour l'obtention des résultats.

RÉFÉRENCES WEBLIOGRAPHIQUES

- [1] <https://www.reseaucerta.org/sites/default/files/coteLaboOcsGlpiv3.pdf>, Consulté le 20/09/2019.
- [2] <https://aful.org/ressources/presentation/logiciels>, Consulté le 02/10/2019.
- [3] <http://opensource4ever.canalblog.com/archives/2013/12/19/28699485.html>, Consulté le 02/10/2019.
- [4] <https://www.ingenosya.com/page/entreprise-ressource-planning>, Consulté le 26/04/2020.
- [5] <https://www.supinfo.com/articles/single/4269-gestion-maintenance-assiste-ordinateur-gmao>, Consulté le 25/09/2019.
- [6] <https://searchdatacenter.techtarget.com/definition/Fedora?amp=1>, Consulté le 26/03/2020.
- [7] <https://riptutorial.com/Download/docker.pdf>, Consulté le 28/08/2020.
- [8] <https://docs.docker.com/get-started/overview/>, Consulté le 20/08/2020.
- [9] <https://docker-curriculum.com/>, Consulté le 28/08/2020.
- [10] https://www.tutorialspoint.com/hadoop/hadoop_tutorial.pdf, Consulté le 23/02/2020..
- [11] <https://yoyoclouds.wordpress.com/2011/12/15/hdfsarchitecture/>, Consulté le 28/02/2020.
- [12] <https://www.guru99.com/learn-hadoop-in-10-minutes.html>, Consulté le 23/02/2020.
- [13] <https://docs.vmware.com/en/VMware-Workstation-Pro/15.0/workstation-pro-15-user-guide.pdf>, Consulté le 20/05/2020.

RÉFÉRENCES BIBLIOGRAPHIQUES

- [14] *"Le document interne de Tchir-Lait Candia Bejaia"*.
- [15] Payne ADRIAN. *Handbook of CRM : Achieving Excellence in Customer Management*. 1er edition, 2005.
- [16] Borri ALAIN, Keller YVAN et Vienney ALEXANDRE. *Livre Blanc : Comment appréhender l'acquisition et le déploiement d'un logiciel TMS*. bp2r-5, rue des Mathurins, 75009 Paris, France, aout 2014.
- [17] Buttle FRANCIS. *Customer Relationship Management : Concepts and Technologies*. 2eme edition, 2009.
- [18] Baglin GÉRARD, Lamouril SAMIR et André THOMAS. *Maîtriser les Progiciels ERP*. Economica, Collection Gestion, 2015.
- [19] Deixonne JEAN-LUC. *Piloter un projet ERP*. Dunod, Paris, 2006.
- [20] Hightower KELSEY, Burns BRENDAN et Joe BEDA. *KUBERNETES Maîtrisez l'orchestrateur des infrastructures du futur*. O'Reilly Media Inc, 2017.
- [21] KOMAR, MANCEL et Le CAM. *Nouvelles Technologies Réseau : Les solutions de stockage*. 11/01/2006.
- [22] Trinity LOGISTICS. *Guide to : Transportation Management Systems (TMS)*. avril 2017.
- [23] Gajjar MANTAVYA. *Odoo Online Book*. Utgåva 12.0, aug. 13, 2018.
- [24] Valyi RAPHAËL. *Livre blanc,ERP open-source*. smile, 2008.
- [25] Eric SAMMER. *Hadoop Operations*. O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472, 2012.
- [26] Kale VIVEK. *Implementing SAP® CRM : The Guide for Business and Technology Managers*. 1er edition, 2014.

Nous vivons à une époque où les données sont générées par tout ce qui nous entoure. Le taux de génération de données est si alarmant qu'il a créé un besoin pressant de mécanismes de stockage et d'extraction de données faciles et rentables. En outre, de grandes quantités de données doivent être analysées pour en tirer des bénéfices et établir des relations d'attributs, ce qui peut conduire à une meilleure prise de décision et à des stratégies commerciales efficaces. Les grandes données nous permettent d'en apprendre davantage sur les nouvelles technologies, les compétences et les processus de notre architecture d'information et sur les personnes qui les exploitent, les conçoivent et les utilisent. Hadoop Distributed File System, également connu sous le nom de HDFS, et Hadoop MapReduce sont suffisamment bons pour mettre en évidence les implications des Big Data sur le marché. Dans ce rapport, nous présentons la mise en œuvre d'un cluster hadoop pour stocker et analyser de grandes quantités de données.

Mots clés : Fedora Server, Big data, Hadoop, HDFS, MapReduce, Docker, Kubernetes.

We live in an age where data is generated by everything around us. The rate of data generation is so alarming that it has created a pressing need for easy and cost-effective data storage and retrieval mechanisms. In addition, large amounts of data need to be analyzed for benefits and attribute relationships, which can lead to better decision making and effective business strategies. Large amounts of data allow us to learn more about the new technologies, skills, and processes in our information architecture and the people who operate, design, and use them. Hadoop Distributed File System, also known as HDFS, and Hadoop MapReduce are good enough to highlight the implications of Big Data in the marketplace. In this report, we present the implementation of a hadoop cluster to store and analyze large amounts of data.

Keywords : Fedora Server, Big data, Hadoop, HDFS, MapReduce, Docker, Kubernetes.