Une image contenant texte, carte de visite

Description générée automatiquement

Table des matières

[I. Etat de l’art 5](#_Toc91230909)

[a) Data Warehouse 5](#_Toc91230910)

[b) Le Data Lake 6](#_Toc91230911)

[a) Docker 7](#_Toc91230912)

[b) Hadoop 8](#_Toc91230913)

[II. Problématique 9](#_Toc91230914)

[III. Proposition de solution 10](#_Toc91230915)

[IV. Aspect pratique 12](#_Toc91230916)

[c) Installation 12](#_Toc91230917)

[d) Fichiers 12](#_Toc91230918)

[e) Utilisation - Ecriture dans HDFS 12](#_Toc91230919)

[f) Commandes importantes : 14](#_Toc91230920)

[1. Dans la commande Bash du namenode : 14](#_Toc91230921)

[2. Dans l'invite Windows : 15](#_Toc91230922)

[V. Conclusion 16](#_Toc91230923)

[VI. Références bibliographiques 18](#_Toc91230924)

[Figure 1 — Data Lake métaphore 6](file:///D:\TRSF%20DATA%20NE%20PAS%20SUPPRIMER\Downloads\Rapport_BIG_DATA.docx#_Toc91230902)

[Figure 2 – Data Lake 6](file:///D:\TRSF%20DATA%20NE%20PAS%20SUPPRIMER\Downloads\Rapport_BIG_DATA.docx#_Toc91230903)

[Figure 3 schéma des interfaces docker 7](file:///D:\TRSF%20DATA%20NE%20PAS%20SUPPRIMER\Downloads\Rapport_BIG_DATA.docx#_Toc91230904)

[Figure 4 docker logo 7](file:///D:\TRSF%20DATA%20NE%20PAS%20SUPPRIMER\Downloads\Rapport_BIG_DATA.docx#_Toc91230905)

[Figure 5 logo docker 7](file:///D:\TRSF%20DATA%20NE%20PAS%20SUPPRIMER\Downloads\Rapport_BIG_DATA.docx#_Toc91230906)

[Figure 6 lecture d’un fichier dans hdfs 8](file:///D:\TRSF%20DATA%20NE%20PAS%20SUPPRIMER\Downloads\Rapport_BIG_DATA.docx#_Toc91230907)

Introduction

Dans le cadre du cours de Big Data, il nous est demandé d’effectuer un travail de recherche afin d’en apprendre plus sur le stockage dans le Big Data et d’évaluer notre capacité à effectuer des recherches sur ce sujet.

Le stockage étant un sujet vaste, ce rapport fera l’objet de documentation sur l’ETL (Data Warehouse) et l’ELT (Data lake) dans un premier temps. Par la suite, le sujet se recentrera sur le data Lake et plus précisément sur la différence entre un mode distribué et un mode non-distribué.

# État de l’art

## Data Warehouse

Bien que la première émergence d’un data Warehouse d’entreprise, ou plus communément en français nommé entrepôt de données, date de la fin des années 1980 d’après l’article de Margot.P(2021). Les entreprises de maintenant l’utilisent encore énormément et justement essayons de le définir pour comprendre pourquoi il est encore si populaire de nos jours.

Selon Alain Fernandez dans son livre de 2013 : « Les nouveaux tableaux de bord des managers », la définition de l’entrepôt de données est Le Data Warehouse, ou entrepôt de données, est une base de données dédiée au stockage de l’ensemble des données utilisées dans le cadre de la prise de décision et de l’analyse décisionnelle. Le Data Warehouse est exclusivement réservé à cet usage. Il est alimenté en données depuis les bases de production grâce notamment aux outils d’ETL Extract Transform Load.

La société Oracle possédant un service pour louer des entrepôts de données autonomes précises que d’un point de la vue technique un Data Warehouse est défini comme un ensemble de données orientées sujet, intégrées, historisé et non volatiles*.*

Une ressource traduite en ligne de la « corporate finance institute » nous explique qu’il y a trois différents types de Data WareHouse :

Le Data Warehouse d’entreprise (EDW) qui fournis des services d’aide à la décision dans toute l’entreprise

Le data Stores opérationnels (ODS) qui a la particularité que ses données se mettent à jour en temps réel, ce qui convient parfaitement à des tâches quotidiennes. Généralement, ils sont associés à des EDW.

Le Data Mart qui est une sous-catégories d’un Data WareHouse. Elle porte sur des sujets précis.

Nous avons donc maintenant une définition bien complète de ce que représente un data Warehouse

## Le Data Lake

Le deuxième grand type de stockage en big data est le data Lake. Le Data Lake ou DL est une technologie qui possède de nombreux avantages dans l’ère de la collecte de données.

Le DL utilise le concept nommé ELT qui signifie Extraction, Insertion, Transformation. Ce nom est simplement la suite d’étape utilisée durant la mise en œuvre.

Une image contenant texte, carte de visite

Description générée automatiquementEffectivement, le lac de données (Data Lake) accorde la possibilité de stocker d’innombrables données dans n’importe quelle forme. Nous n’avons donc pas besoin de nous soucier du format de nos données lors de la récolte et ensuite du stockage de celle-ci. Ce n’est que lors de la récupération des données dans le data Lake que la transformation des données sera faite afin de les adapter à l’utilisation que l’on souhaite en faire.

Le machine Learning étant une méthodologie d’apprentissage pour ordinateur consistant à l’ordinateur d’apprendre sans que l’humain n’intervienne dans ses lignes de code. Cet apprentissage se fait par un biais d’étude mathématiques et statistiques en analysant les données. Plus le nombre de données est important, plus l’apprentissage est efficace dans toutes les situations. Il est indéniable que le stockage brut de données constitue un avantage considérable. De plus le stockage dans un Data Lake est volatile ce qui signifie que les données peuvent être alimentées en temps réel, cela pourrait permettre à un algorithme d’apprentissage de se mettre à jour constamment sur des données collecter continuellement.

Figure 1 - Data Lake métaphore

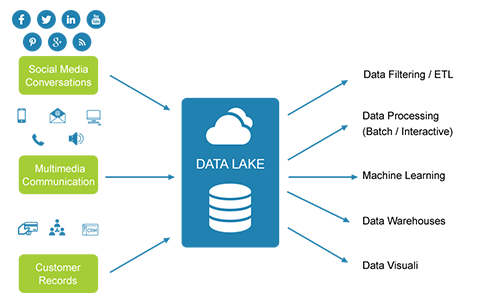
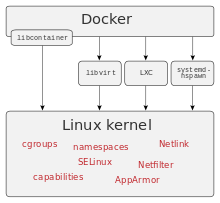
 Cependant, le lac de données possède aussi ses défauts. Une technologie aussi avantageuse pour le Machine Learning et le stockage de masse ne peut pas être que positif. La data Lake est un cout important et constitue une perte d’intégrité des données.  
  
En effet, un data Lake peut couter plusieurs centaines de milliers d’euros pour une très importante utilisation. Il est donc important pour choisir ce type de technologie, d’en avoir le besoin. De plus, pour une utilisation autre que le machine Learning, la perte d’intégrité des données peut poser de gros problèmes, la cohérence des données étant en causes. Le stockage de données brut et la perte d’intégrité des données n’étant pas un problème pour le machine Learning, le data Lake est principalement utilisé par des data scientifique et des data analyste.

Figure 2 — Data Lake

## Docker

Développer une application nécessite énormément plus que d’écrire de simples lignes de code, nous l’avons déjà bien remarqué durant nos années d’étude à Henallux Namur. Plein de Frameworks différents avec des API pas écrits dans le même langage de programmation et souvent tout cela nécessitent des connaissances pour les utiliser et/ou faire fonctionner. Cela créée énormément de complexité pour avoir accès à toutes ses ressources.

C’est cette complexité qui a fait Solomon Hykes propose, en 2010, à des investisseurs le concept de docker et de la technologie des conteneurs en 2010. Le concept séduit et soulève une première montée de fond de 800 000 $. L’année d’après, la société Docker inc. s’implémente de la Silicon Valley et en novembre 2019 docker inc. sera vendu à Mirantis pour la modique somme de 35 millions de dollars.

Docker permet donc le déploiement de conteneurs qui s’exécutent individuellement via une API de haut niveau construite sur les capacités du noyau Linux. Les conteneurs Docker contrairement aux machines virtuelles traditionnelles ne nécessitent pas de système d’exploitation séparé. Au lieu de cela il s’appuie sur les fonctionnalités du noyau et utilise l’isolation des ressources matérielles de l’hôte.

L’utilisation de Docker avec des conteneurs tel que ceux de la communauté peut simplifier le déploiement de systèmes distribués en permettant à plusieurs applications tâches d’arrière-plan et autres processus de s’exécuter automatiquement sur une seule machine physique ou sur une série de machines distinctes. Cela permet aux nœuds d’être déployés en tant que ressources à la demande offrant ainsi une plate-forme de déploiement de style PAAS et l’évolutivité de systèmes tels qu’apache Cassandra MongoDB ou HDFS et simplifie leur mise en œuvre.

Figure 3 schéma des interfaces docker

Nous avons utilisé docker pour réaliser la proposition de solution qui est disponible un peu plus loin dans ce rapport c’est pour cela que nous avons développé légèrement docker.

Figure 4 docker logo



Figure 5 logo docker

## Hadoop

Comme les SGBD traditionnels ne peuvent pas gérer des téraoctets de données, ce n’est qu’en utilisant des technologies comme Hadoop que de gros volumes de données non structurées peuvent être traités. Le Hadoop peut être interrogé en interrogeant dans son propre langage ou par SQL. De plus, les Frameworks Hadoop Open source sont nommés NoSQL pour Not Only SQL.

La technologie Hadoop est un Framework open source privilégié en entreprise pour faciliter l’exécution d’applications distribuées et stocker apache est le précurseur de cette technologie innovante qui sera la source ou plutôt l’inspiration de nombreuses solutions en big data.

Le Framework Hadoop de base se compose des modules suivants :

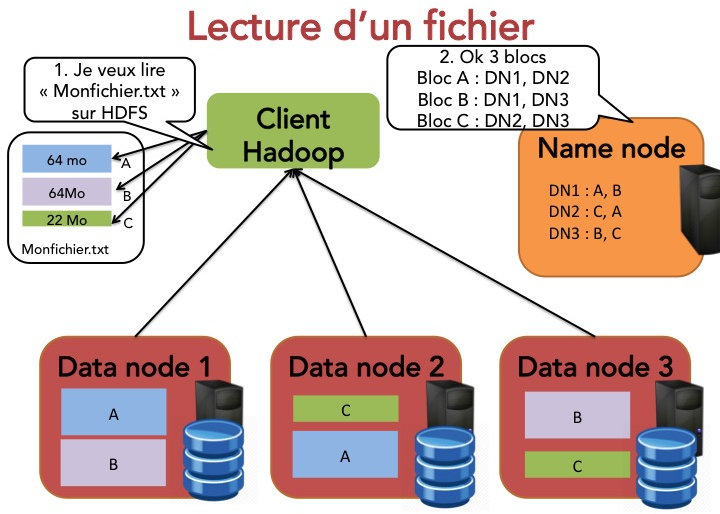
* Hadoop Common: Composants communs pour la gestion des systèmes de fichiers distribués. C’est la source de nombreux autres modules.
* Hadoop Distributed File System (HDFS), système de fichiers : Les systèmes de fichiers distribués sont conçus pour gérer des volumes élevés. HDFS effectue une réplication de données volumineuses.

Figure 6 lecture d’un fichier dans hdfs

* Hadoop YARN: Gestionnaire de cluster.
* Hadoop MapReduce : Modèle de programmation.

Le terme Hadoop fait référence non seulement aux modules de base ci-dessus, mais aussi à son écosystème et à l’ensemble des logiciels qui viennent s’y connecter comme Apache Hive, Apache Phoenix, Apache Spark, Apache ZooKeeper, etc. Ils ont tous des utilités allant du traitement en temps réel jusqu’à la gestion de cluster via une interface web.

Utiliser Hadoop veut généralement dire que votre entreprise a besoin de traiter en temps réel des données non structurées nécessitant une architecture évolutive en parfaite scalabilité avec le temps.

# Problématique

Dans le cadre du cours de big data, il nous est demandé de trouver une problématique sur le thème imposer. Après mure réflexion, nous nous sommes penché la différence de cluster possible dans HDFS. Très vite, quelques recherches nous ont menés sur deux modes de distribution possible. Le premier étant le mode distribué et le second, le mode non distribué. À la suite de cette découverte, plusieurs questions se sont posées ; quel est le plus performant ? Dans quel cadre utilisons-nous un mode non distribué ? Comment mettre en places ces types de clusters ?

Nous sommes arrivés à la conclusion qu’il serait important d’établir une comparaison des deux modes de distribution. Ainsi que d’établir les situations dans lesquelles nous utilisons l’un ou l’autre. Diverses recherches nous ont mené donc à centrer, à formuler et penser notre problématique comme suit :

Le mode distribué est-il la solution à tous les problèmes ?

Maintenant que la problématique est posée, nous essayerons par la suite d’y répondre le plus pertinemment possible. Ainsi que d’expliquer au mieux l’installation et l’utilisation de ces différents modes de distribution.

# Proposition de solution

Après avoir posé la problématique de notre rédaction, ainsi que la définition des différentes technologies que nous utilisons. Il est prioritaire de trouver une solution qui répondra avec certitude à notre problème posé plus tôt.  
  
Comme cité auparavant, il existe deux types de distribution dans Hadoop DFS. Le mode distribué possède de nombreux avantages, selon dikshantmalidev

« Hadoop est un modèle hautement évolutif. Une grande quantité de données est divisée en plusieurs machines peu coûteuses dans un cluster qui est traité en parallèle. Le nombre de ces machines ou nœuds peut être augmenté ou diminué selon les besoins de l’entreprise. »

De plus, une autre grande force d’Hadoop DFS est la tolérance aux pannes, comme l’explique dikshantmalidev dans son article :

« Hadoop utilise du matériel de base (systèmes peu coûteux) qui peut tomber en panne à tout moment. Dans Hadoop, les données sont répliquées sur divers DataNodes dans un cluster Hadoop, ce qui garantit la disponibilité des données en cas de panne de l’un de vos systèmes. Vous pouvez lire toutes les données d’une seule machine si cette machine rencontre un problème technique. Les données peuvent également être lues à partir d’autres nœuds d’un cluster Hadoop, car les données sont copiées ou répliquées par défaut. »

On peut complémentaires ces avantages avec le cout que cela implique comme ledit dikshantmalidev, un système distribué comporte un cout moindre. Comme ajoute Brien Posey dans son article pour Lemagit.fr ; le logiciel utilisé pour HDFS est open source.

HDFS est aussi reconnu pour sa fiabilité quant aux traitements des données en temps réel. Effectivement, Brien Posey nous explique que les capacités de traitement en parallèle de Hadoop participent à la rapidité de l’analyse, mais qu’il est possible qu’à un moment la puissance du cluster ne soit plus suffisante. Heureusement cela peut être réglé en agrandissant le cluster, donc en ajoutant des nœuds.

L’inconvénient principal d’HDFS est qu’il n’y a pas de protection contre la corruption de données comme l’explique Mike UZAN dans son article nommé « Hadoop OUI, HDFS NON ! »

D’après « ijasca.thesais.org », le tableau comparatif qui suit, que nous avons pris le temps de traduire. Nous montre les principales différences des deux types de distributions.

|  |  |
| --- | --- |
| Non-distribué | Distribué |
| Hadoop est installé sur une seule machine ou possède un datanode | Hadoop est installé sur plusieurs datanodes sur un réseau LAN |
| Tous les nœuds sont exécutés sur une seule et même machine | Tous les nœuds sont exécutés sur différentes machines |
| Le nombre de réplica est de 1 | Le nombre de réplicas est supérieur à 1 et devrait être installé sur plus d’une machine |
| L’utilisation prédominante de cette distribution est le testing et l’étude de la technologie | L’utilisation est pour le BigData pour de hautes volumétries de données |
| Utilisé pour MapReduce et des opérations basiques HDFS | Utilisé pour des besoins complexes comme notamment les analyses BigData |

Comme démontré plus haut dans le tableau comparatif, on peut voir que le non-distribué n’est utilisé que dans des cas d’apprentissage ou de découverte de la technologie HDFS. Cela peut s’expliquer par le faite qu’utiliser un mode de distribution non-distribué sur une technologie nommé « Hadoop Distributed File System » (Hadoop système de fichiers distribué) est un non-sens. Effectivement les avantages d’HDFS vu plus tôt ne s’appliquent pas. Il est donc préférable d’utiliser d’autres technologies pour un stockage simple et non morcelé. L’utilisation d’un Data Warehouse défini dans l’état de l’art peut surement s’avérer une bonne solution.

Nous avons quand même mis en place un système non-distribué malgré le peu de valeurs qu’il ajoute. Nous avons aussi implémenté un système distribué que vous pourrez retrouver en plus de l’autre distribution dans l’aspect pratique ci-dessous.

# Aspect pratique

Maintenant le cadre de notre problématique nous a demandé de mettre en place une solution Hadoop HDFS. Nous nous sommes basés sur un projet existant disponible en open source sur GitHub. Sur base de : [github/big-data-europe](https://github.com/big-data-europe/docker-hadoop).  
  
Nous avons mis à disposition différents types de clusters.

* Le premier cluster est un mode distribué possédant 5 datanodes.
* Le second est lui aussi un mode distribué, mais cette fois, il possède seulement 3 datanodes.
* Le dernier cluster que nous avons mis en place est un mode non distribué, cela veut dire qu’il n’y a qu’un seul datanodes dans ce cluster.

Afin de faciliter l’utilisation, nous avons créé 3 fichiers différents, les voici…

Cluster à disposition :

* Distribuer-3-nodes.yml
* Distribuer-5-nodes.yml
* Non-distribuer.yml

## Installation

Maintenant afin de pouvoir utiliser le projet, procédons à l’installation.

1. Téléchargez [Docker](https://www.docker.com/get-started) (https://www.docker.com/get-started)
2. Clonez-le [git](https://github.com/Kekermit/Projet-BigData-HUGO_D-QUENTIN_F-Emeric_L) (https://github.com/Kekermit/Projet-BigData-HUGO\_D-QUENTIN\_F-Emeric\_L)
3. Cd Chemin/accès/fichier
4. Choisissez le cluster que vous voulez démarrer
5. Lancez une invite de commande
6. Démarrer le cluster choisi avec cette commande

**docker-compose — -file <nomFichier>.yml up — d**

1. Accéder à la page web -> <http://localhost:9870/>
2. Vous pouvez éteindre le cluster avec

**docker-compose — -file <nomFichier>. yml down**

## Fichiers

L’installation est terminée ! Avant de passer à l’utilisation de notre cluster, il faut préparer quelques fichiers à écrire dans HDFS.   
Nous avons mis à disposition différents fichiers d’exemple à écrire dans HDFS.

Lien : [Fichiers](https://mega.nz/folder/slolVIbB#7RUxiXkNBGd1J-jYSD0i5g) (https://mega.nz/folder/slolVIbB#7RUxiXkNBGd1J-jYSD0i5g)

## Utilisation — Écriture dans HDFS

Procédons à l’utilisation d’HDFS, la première étape est d’ouvrir votre invite de commandes Windows.

Pour ouvrir le terminal Bash du namenode, exécutez cette commande dans l’invite.

**docker exec — it namenode bash**

Créez un répertoire pour stocker les fichiers d’entrée.

**mkdir input**

Si vous souhaitez utiliser mapReduce :

Créer un autre répertoire pour stocker les fichiers jar

**mkdir jars**

Dans un second terminal

Ajoutez le fichier dans le namenode, d’abord trouvez le container ID de votre namenode en utilisant :

**docker container ls**

Cela devrait ressembler à quelque chose comme cela cb0c13085cd3

**docker cp <Chemin>\<Chemin>\<nomFichier>.<Format> <NAMENODE-CONTAINER-ID> : /input/**

Si vous souhaitez utiliser map reduce :

Ajoutez le fichier. jar nommer wordcount dans le namenode

Dans le dossier « submit » du git, un dossier contient déjà un fichier. jar nommé ‘WordCount.jar’

**docker cp submit/WordCount.jar <NAMENODE-CONTAINER-ID>: /jars/**

Dans le premier terminal

Pour distribuer il faut faire ces deux commandes

**hadoop fs — mkdir — p input**

**hdfs dfs — put./input/\* input**

Pour map Reduce:

Démarrer l’exécutable

**hadoop jar jars/WordCount.jar org. apache. hadoop. examples. WordCount input output**

Pour voir la sortie

**hdfs dfs — ls output/**

**hdfs dfs — cat output/part-r-00000**

Vous pouvez vérifier l’écriture des donnes via l’adresse suivante :

<http://localhost:9870/>

Vous pourrez alors voir le stockage des datanodes.

## Commandes importantes :

Afin que vous puissiez utiliser HDFS par vous-mêmes, il est important de connaitre quelques commandes importantes.

### Dans la commande Bash du namenode :

Voir la liste des fichiers/dossiers dans le répertoire HDFS

**hdfs dfs — ls/**

Crée un dossier

**hdfs dfs — mkdir/<nomDossier>**

Permet de transférer un fichier local a hadoop

**hdfs dfs — put <chemin local> <chemin hadoop>**

Permet de lire le fichier dans hadoop

**hdfs dfs — cat <nomFichier>**

Vérifier si le fichier existe dans HDFS

**hadoop fs — ls**

**hadoop fs — ls input/**

Supprimer les fichiers

**hadoop fs — rm./input/\* input**

Vérifier la division des fichiers

**hdfs fsck input/\***

Augmenter le nombre de blocs diviser lors de l’écriture

**hdfs dfs — Ddfs.blocksize=1048576 — put./input/\* input**

Vérifier les nœuds en vie

**hdfs dfsadmin — report**

### Dans l’invite Windows :

Pour voir les images lancer + port

**Docker ps**

Pour voir les networks en cours

**Docker network ls**

# Conclusion

En conclusion, nous avons vu dans cette rédaction qu’il existait différentes façons de stocker nos données. La première façon abordée était le Data Warehouse, un système de stockage non volatile, soumis à un stockage formaté, utile pour un sujet précis. La seconde façon était le Data Lake, lui n’est pas soumis à un formatage des données pour le stockage, il est volatile.

Par la suite nous avons appris qu’il était possible de gérer la distribution de son Data Lake sous deux formes en utilisant Hadoop DFS. La première distribution se base sur un cluster dit distribué, le cluster distribué découpera et répliquera les données sur chaque nœud aléatoirement. Ce mode de fonctionnement est plutôt avantageux pour une utilisation en temps réel, notamment pour le Machine Learning, car comme citée plus tôt, le Data Lake n’a pas besoin de formatage de données.

Le mode non-distribué repose aussi sur l’utilisation d’Hadoop DFS. Cependant, ici, il n’y aura pas de réplication des données ni de découpe. Effectivement, le mode non distribué ne possède qu’un nœud comparé à son cousin le distribué. Le fait de ne posséder qu’un nœud enlève donc les avantages d’Hadoop DFS, c’est pourquoi cette méthode distribution est pratiquée dans un cadre d’apprentissage et de découverte de la technologie. Dans un cadre professionnel, cette technique n’existe donc pas. Dans un cas de volonté de stockage sans réplication et sans les avantages d’un mode distribué. Il est préférable de choisir une autre technologie telle que le Data Warehouse par exemple.

Au-delà d’apprentissage théorique, nous avons aussi vu l’installation, ainsi que l’utilisation d’HDFS et de docker. Lors de cet apprentissage pratique, différents clusters ont été mis à disposition, le premier est le mode non-distribué malgré son intérêt moindre. Les deux derniers sont des modes distribués, un possède 3 nœuds et l’autre 5 nœuds. Avoir choisir 3 et 5 nœuds facilite la compréhension que le cela peut donner à plus grande échelle.

Nous pouvons désormais répondre à notre problématique :

« Le mode distribué est-il la solution à tous les problèmes ? »

Nous avons vu que le mode distribué est la réponse au problème de stockage de masse avec un afflux continu, en gardant la sécurité des données, mais en perdant l’intégrité de celle-ci. Nous pouvons donc conclure que dans un cadre analytique précis, le mode distribué n’est pas adapté. C’est pourquoi nous pensons que le mode distribué est la solution aux problèmes de stockage dans un cadre de machine Learning. Et que dans le cas d’un besoin analytique performer par un humain, il serait préférable d’utiliser un Data Warehouse.

Nous tenons à remercier quelques personnes qui nous ont apporté leurs soutiens durant cette rédaction. La première personne est notre professeur M. Bouraada qui nous a épaulés et guider tout le long du projet avec ses conseils et ses remarques pertinentes. La seconde personne est notre collègue Hugo Devos qui pour des raisons médicales n’a pas pu poursuivre le projet dans toute sa durée. Malgré ça, il nous a été d’une aide précieuse que nous tenons à soulever.

Nous tenons aussi à vous remercier d’avoir lu ce rapport sur le stockage et plus particulièrement le data lake et HDFS.

# Références bibliographiques

big-data-europe/docker-hadoop. (2021). [Logiciel d’ordinateur]. https://github.com/big-data-europe/docker-hadoop

Brien Posey. (s. d.). Cluster Hadoop avantages et limites. lemagit. Consulté le 20 novembre 2021, à l’adresse https://www.lemagit.fr/article/Clusters-Hadoop-avantages-et-limites-pour-le-Big-Data

Corporate Finance Institute. (2021, 28 juillet). Data Warehouse. Consulté le 25 novembre 2021, à l’adresse https://corporatefinanceinstitute.com/resources/knowledge/data-analysis/data-warehouse/#:%7E : text=The%20three%20main%20types%20of, ODS)% 2C%20and%20data%20mart.

Data-Warehouse. (s. d.). Oracle. Consulté le 15 décembre 2021, à l’adresse https://www.oracle.com/fr/database/data-warehouse-definition.html

Fernandez, A. (2020, 9 mars). Qu’est-que le Data Warehouse, Entrepôt de Données ? Management et Performance, piloter.org. Consulté le 20 novembre 2021, à l’adresse https://www.piloter.org/business-intelligence/datawarehouse.htm

HADOOP : A Comparative Study between Single-Node and Multi-Node Cluster. (2021, novembre). https://thesai.org/Downloads/Volume12No2/Paper\_7-HADOOP\_%20A\_Comparative\_Study.pdf

L, B. (2017, 3 octobre). HDFS : fonctionnement, avantages et inconvénients du système de fichiers de Apache Hadoop. LeBigData.fr. Consulté le 20 novembre 2021, à l’adresse https://www.lebigdata.fr/hdfs-fonctionnement-avantages#Pourquoi\_a-t-on\_besoin\_de\_HDFS

Lima, A. (s. d.). Hadoop — Avantages et inconvénients. Acervo Lima. https://fr.acervolima.com/hadoop-avantages-et-inconvenients/

Mike UZAN. (2019, 6 mars). Hadoop OUI, Hdfs NON ! Programmez ! Consulté le 20 novembre 2021, à l’adresse https://www.programmez.com/avis-experts/hadoop-oui-hdfs-non-28658

P, M. (2021, 6 mai). Data Warehouse : qu’est-ce que c’est et comment les utiliser ? Formation Data Science | DataScientest.com. Consulté le 25 novembre 2021, à l’adresse https://datascientest.com/data-warehouse

Apache Hadoop 3.3.1 — Apache Hadoop YARN. (s. d.). https://hadoop.apache.org. Consulté le 22 décembre 2021, à l’adresse https://hadoop.apache.org/docs/stable/hadoop-yarn/hadoop-yarn-site/YARN.html

Docker. (s. d.). Docker logo [Logo]. https://fr.wikipedia.org/wiki/Docker\_ (logiciel). https://commons.wikimedia.org/w/index.php?title=File:Docker\_ (container\_engine)\_logo.svg&lang=fr&uselang=fr

Hadoop, stockage big data – Next Decision. (s. d.). https://www.next-decision.fr. Consulté le 22 décembre 2021, à l’adresse https://www.next-decision.fr/editeurs-big-data/hadoop#:%7E:text=La%20technologie%20Hadoop%20est%20un,des%20n%C5%93uds%20d’un%20cluster

Lecture d’un fichier dans hdfs. (s. d.). [Schéma]. https://openclassrooms.com. https://openclassrooms.com/fr/courses/4467481-creez-votre-data-lake/4509426-decouvrez-le-systeme-de-fichiers-distribue-hdfs

Schéma des interfaces de docker. (s. d.). [Schéma]. https://fr.wikipedia.org/wiki/Docker\_ (logiciel). https://commons.wikimedia.org/wiki/File : Docker-linux-interfaces.svg? uselang=fr

What is a Container? (s. d.). Docker. Consulté le 22 décembre 2021, à l’adresse https://www.docker.com/resources/what-container

Why ? (s. d.). Docker. Consulté le 22 décembre 2021, à l’adresse https://www.docker.com/why-docker

Wikipédia contributors. (s. d.). Hadoop. Wikipédia. Consulté le 22 décembre 2021, à l’adresse https://fr.wikipedia.org/wiki/Hadoop

Wikipédia contributors. (2021 a, juin 21). MapReduce. Wikipédia. Consulté le 22 décembre 2021, à l’adresse https://fr.wikipedia.org/wiki/MapReduce#Pr%C3%A9sentationWikipedia contributors. (2021 b, octobre 7).

Solomon Hykes. Wikipédia. https://fr.wikipedia.org/wiki/Solomon\_HykesWikipedia contributors. (2021 c, novembre 22). Docker (logiciel). Wikipédia. Consulté le 22 décembre 2021, à l’adresse https://fr.wikipedia.org/wiki/Docker\_ (logiciel)

DataLake. (s. d.). [Illustration]. Data lake illustration. https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.journaldunet.fr%2Fweb-tech%2Fdictionnaire-du-webmastering%2F1165409-data-lake-ou-lac-de-donnees-la-solution-reine-du-big-data%2F&psig=AOvVaw3i8ZzZWV8DHm9mcW61yY-p&ust=1639571463878000&source=images&cd=vfe&ved=0CAsQjRxqFwoTCMCW6eil4\_QCFQAAAAAdAAAAABAD

Data lake schéma. (s. d.). [Illustration schématique]. Data lake schéma. https://d2m6ke2px6quvq.cloudfront.net/uploads/2020/05/19/753a96c4-b8e7-45ef-b963-0b3ca9d65647.jpg