# Chap I: Le Big-Data

# Tables des matières

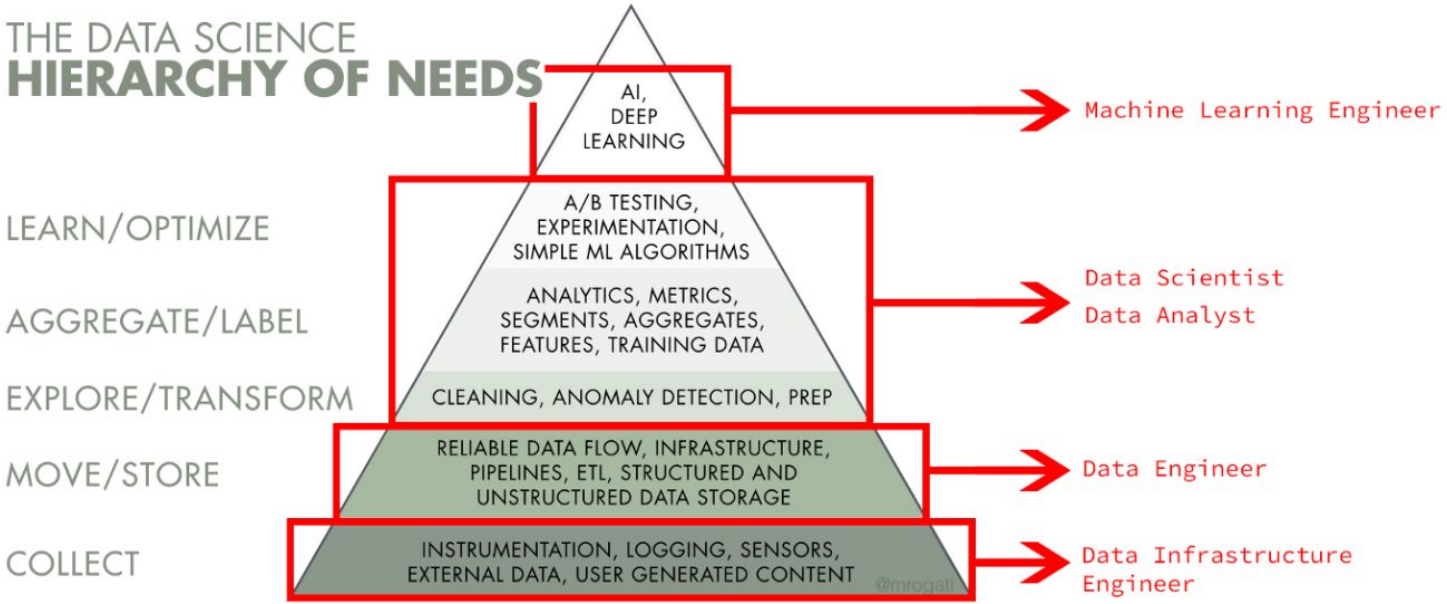
# Tour de table

# About Me

# About Me

Data-Engineer, +3 ans d’expériences avec les techno Big-Data et Cloud.

# Introduction



# Introduction

*“ There was 5 exabytes of information created between the dawn of civilization through 2003, but that much informations now created every 2 days, and the pace is increasing."*

Eric Schmidt, Former PDG Google, 2010

# Focus sur la quantité de données produites par les réseaux sociaux à chaque minutes de la journée

527,760 photos

120 nouveaux utilisateurs

4,146,600 vidéos regardées

456,000 tweets

46,740 nouvelles photos

# Définition

Le Big Data (ou mégadonnées) représente les collections de données caractérisées par un **volume**, une **vélocité** et une **variété** si grands que leur transformation en valeur utilisable requiert l'utilisation de **technologies et de méthodes analytiques spécifiques**.

# Caractéristiques du big-data

| Volume | Vélocité | Variété | Véracité | Variabilité | Valeur |
|---|---|---|---|---|---|
| Quantité importante de données issues de multiples sources. | La vitesse à laquelle la donnée est créée et consommée. | Différentes types de données (Structurées, non-structurées, semi-structurées) | A quelle mesure la donnée est fiable | Comportement de la donnée au cours du temps (anomalies, ). | La valeur associée à l'information extraite de la donnée |

# Pourquoi le Big-Data

- Augmentation exponentielle de la quantité de données (non structurées notamment: chat, blog, web, musique, photo, vidéo, etc)
- L'augmentation de la capacité de stockage à un coût relativement bas
- Nouvelles techniques d'analyse/d'exploitation de la donnée (Visualisation, Data-Science, etc)
- De nouvelles technologies plus adaptées (Hadoop)
- L'avènement du CLoud

# Les Outils de stockages

# Les différentes types de bases de données

- Les bases de données relationnelles (Sql)
- Les bases de données Nosql
- Le Stockage Objet

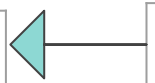
# Les bases de données relationnelles

## Définition

Une base de données relationnelle est un ensemble d'éléments de données dotés de relations prédéfinies entre eux. Ces éléments sont organisés en un jeu de tables composées de colonnes et de lignes.

students

| id_stud | name | dept |
|---|---|---|
| 1 | Abdou | 2 |
| 2 | Jeanne | 1 |
| 3 | Ibou | 3 |



| id_dept | name |
|---|---|
| 1 | Maths |
| 2 | Infos |
| 3 | Bio |

départements

# Quelques notions clés

**Attribut:** Un attribut est un identificateur (un nom) décrivant une information stockée dans une base.
**Schéma de relation:** Un schéma de relation précise le nom de la relation ainsi que la liste des attributs avec leurs domaines.
**Clé primaire:** La clé primaire d'une relation est un attribut unique qui identifie de manière unique un élément d'une table.
**Clé étrangère:** Une clé étrangère dans une relation est formée d'un ou plusieurs attributs qui constituent une clé primaire dans une autre relation.
**Transaction:** Une transaction de base de données désigne une ou plusieurs instructions SQL exécutées en tant que séquence d'opérations (requête).

| id_stud | name | dept |
|---|---|---|
| 1 | Abdou | 2 |
| 2 | Jeanne | 1 |
| 3 | Ibou | 3 |

# Propriété ACID

- **Atomicité:** La transaction est exécutée dans son intégralité ou invalidée dans son intégralité.
- **Cohérence:** requiert que les données écrites dans la base de données dans le cadre d'une transaction soient conformes à toutes les règles et restrictions définies.
- **Isolation:** L'indépendance des différentes transactions en cas de concurrence.
- **Durabilité:** requiert que toutes les modifications réalisées sur la base de données soient permanentes une fois la transaction exécutée avec succès

# Avantages et Inconvénients

- **Les ++**
  - Un modèle de données simple
  - Préservation de l’intégrité des données
  - Un système de requêtage assez mature et facile à prendre en main

- **Les - -**
  - Un schéma parfois très rigide moins pratique pour des données qui varient constamment
  - Scalabilité horizontale parfois difficile à mettre en place
  - Souvent moins performant que le modèle non-relationnel

# Cas d’usage

- Données structurées avec de fortes contraintes sur le modèle de données
- Pour des transactions complexes et/ou fréquentes sur les données
- Quand on a des relations entre les différentes entités présentes.

# Exemples de systèmes de bases de données relationnelles

# Le modèle non relationnel, le NoSql

Le NoSQL est un ensemble de technologies de base de données qui **repose sur un modèle différent de celui des BDD relationnelles**. Il se passe de l'exigence d’avoir un schéma prédéfini offrant ainsi plus de flexibilité avec la possibilité de stocker des données non-structurées, semi-structurées et structurées.

Il existe 4 grandes familles de base de données NoSql: Les bases de données orientées **clé-valeur,** les bases de données **document**, les bases de données orientées **colonne** et les bases de données type **graphe**.

# Le modèle clé-valeur

| Clé | Valeur |
| --- | --- |
| 1 | https://adresseweb.com |
| 2 | 356 |
| 3 | **mail**: monmail@gmail.com<br>**date**: 25/10/2020 13:42:12 |

**Exemples**:

-  DynamoDB
  amazon web services
- redis

# **Le modèle document** (basé sur le json et le xml)

Document (id: 5baf47)

```
{
 "nom": "Liquide vaisselle",
 "images": [
   "https://...",
   "https://..."
 ],
 "specs": {
   "parfum": "Orange"
 }
}
```

Document (id: ea53aa)

```
{
 "nom": "Shampooing",
 "images": [
   "https://...",
   "https://..."
 ],
 "specs": {
   "parfum": "Vanille"
 }
}
```

Document (id: d710bb)

```
{
 "nom": "Fromage blanc",
 "images": [
   "https://...",
   "https://..."
 ],
 "specs": {
   "mat_grasses": "0%"
 }
}
```

**Exemples**:

- 

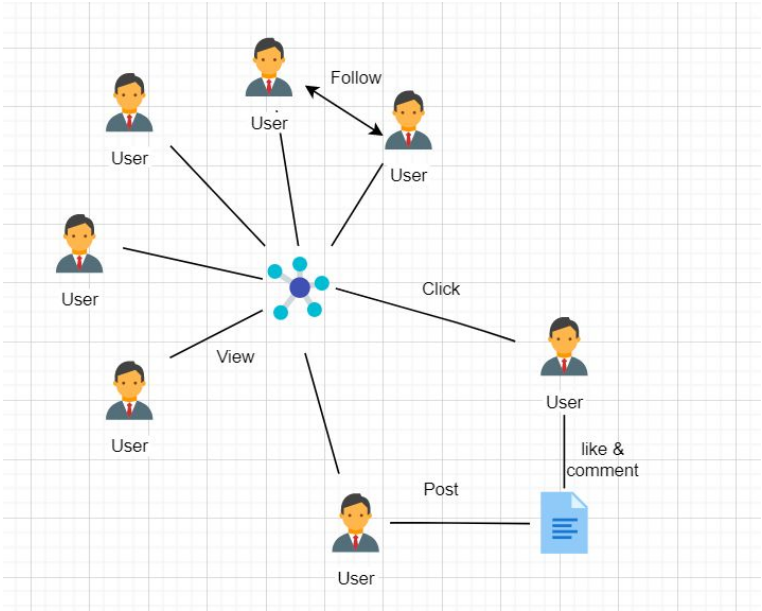
- 

# Le modèle orienté colonne


**Exemples**:

- APACHE HBASE
- cassandra

# Le modèle Graphe




**Exemple**:

- neo4j

# **Le Stockage Objet** (un modèle souvent basé sur le cloud)

Le **stockage Objet** est un format de stockage dans lequel les données sont stockées sous forme d'unités discrètes appelées objets. Chaque unité a un identifiant unique ou une clé qui permet de la retrouver où qu'elle soit stockée dans un système distribué.

# Les avantages

- **Évolutivité :** une architecture simple

- **Données à la demande :** avec le stockage d'objets, il est plus facile de payer seulement pour la capacité de stockage effectivement utilisée.

- **Supporte multiples Api :** vous pouvez accéder et gérer les données dans des systèmes de stockage d'objets par différentes manières

- **Meilleure intégrité des données :** grâce au codage à effacement, les systèmes de stockage d'objets peuvent protéger l'intégrité des données en reconstruisant des morceaux de vos données et en effectuant des contrôles d'intégrité pour prévenir la corruption.
- **Disponibilité des données:** Profitant du cloud, ces systèmes offrent une haute disponibilité de la donnée.
- **Scalabilité** (jusqu'à plusieurs Teras de données)

# cas d’usage

- En tant que **Datalake**
- Pour les gros volumes de données non-structurées (Photos, vidéos, fichiers etc)
- Machine-Learning, Data-Science
- Backup and archivage

# Exemples


Azure Data Lake Storage Gen2

# L’avènement du cloud

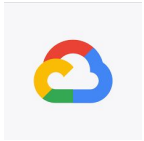
# Définition

Le terme « cloud » désigne les serveurs accessibles sur Internet, ainsi que les logiciels, bases de données et autres services qui fonctionnent sur ces serveurs.
Les serveurs situés dans le cloud sont hébergés au sein de datacenters répartis dans le monde entier.
L'utilisation du cloud computing (informatique cloud) permet aux utilisateurs et aux entreprises de s'affranchir de la nécessité de gérer des serveurs physiques eux-mêmes ou d'exécuter des applications logicielles sur leurs propres équipements.

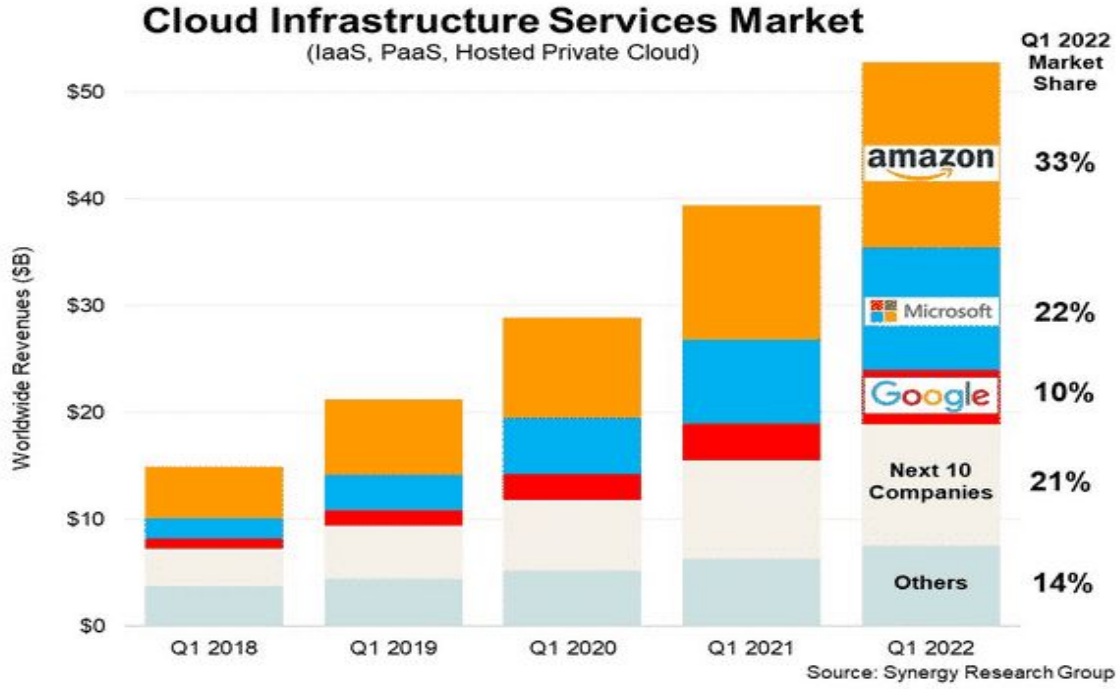
# Les Cloud-Providers

- Amazon Web Service (AWS) 

- Microsoft Azure 

- Google Cloud Platform (GCP) 
- IBM Cloud 
- Oracle 

# Le marché du Cloud (évolution et répartition)



# Quelques avantages du Cloud

- Un accès illimité et démocratisé à des serveurs puissants et à des services de haute qualité.
- Une technologie évolutive et flexible (Passage à l’échelle et Agilité)
- Une réduction des coûts
- Permettre aux entreprises de se focaliser sur leur coeurs de métiers
- Une meilleure sécurité des outils
- Réduction des pertes accidentelles de données

# Focus sur AWS

- Le premier Cloud Provider public créé en 2008
- Présentation de l’interface
- Quelques services

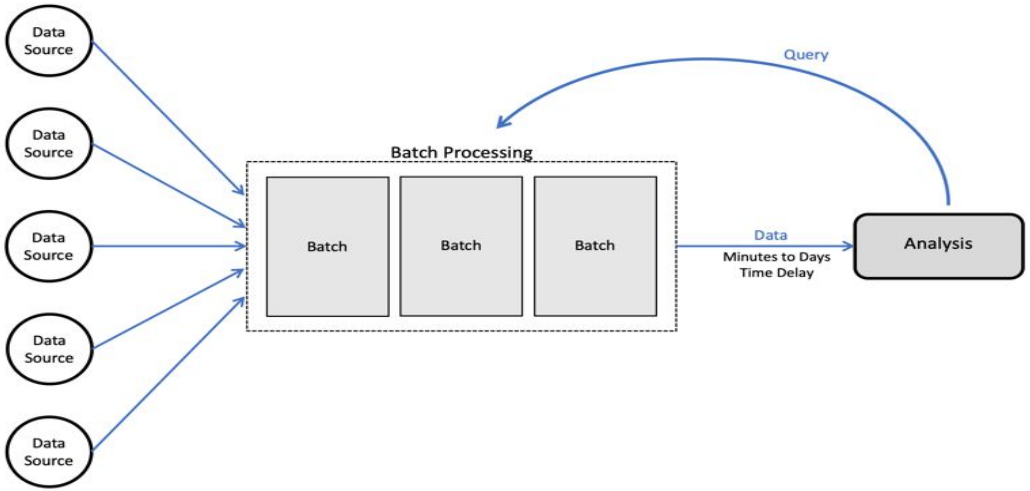
# Quelques services aws

- ***RDS:*** Pour la gestion de base de données relationnelles
- ***EC2*** pour utiliser des instances/serveurs
- ***S3*** pour le stockage
- ***Amazon-Sagemaker*** pour faire du Machine-Learning/Data-Science
- ***Lambda*** pour lancer des fonctions sans se soucier de l'infra

# Notions de Streaming & Batch processing

# Batch Processing (Traitement par lots)

Le *traitement en batch* consiste à exécuter des jobs de traitements répétitifs contenant des volumes importants. Le traitement se fait généralement sur des données cumulées sur une période bien précise.



# Exemples

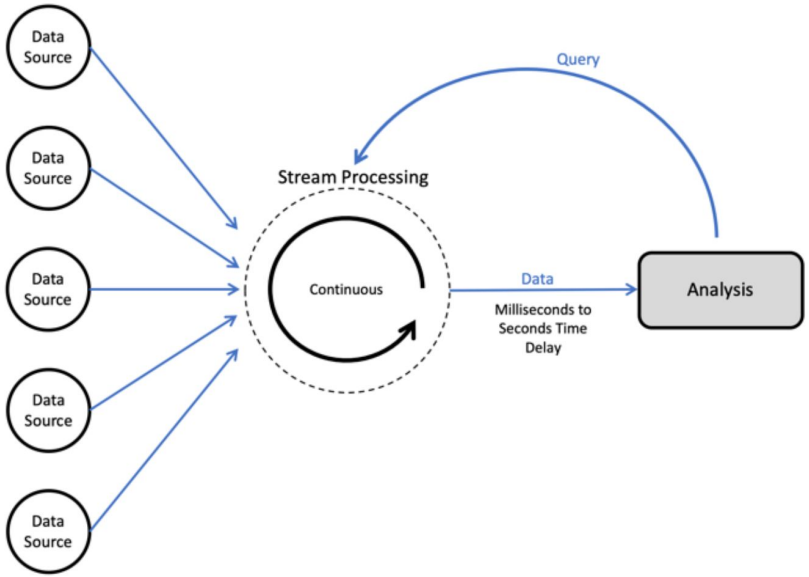
- Les cron jobs
- Reporting

# Quelques Outils pour faire du batch processing




# Streaming Processing (Traitement en continue)

A l'inverse du batch streaming, ici les données sont traitées au fil de l'eau, c'est-à-dire aussitôt que les données nous arrivent elles sont directement traitées.



# Exemples

- Parcours client sur un site de e-commerce (panier, validation paiement )
- Recommandations de films sur Netflix/Youtube

# Quelques outils pour faire du streaming



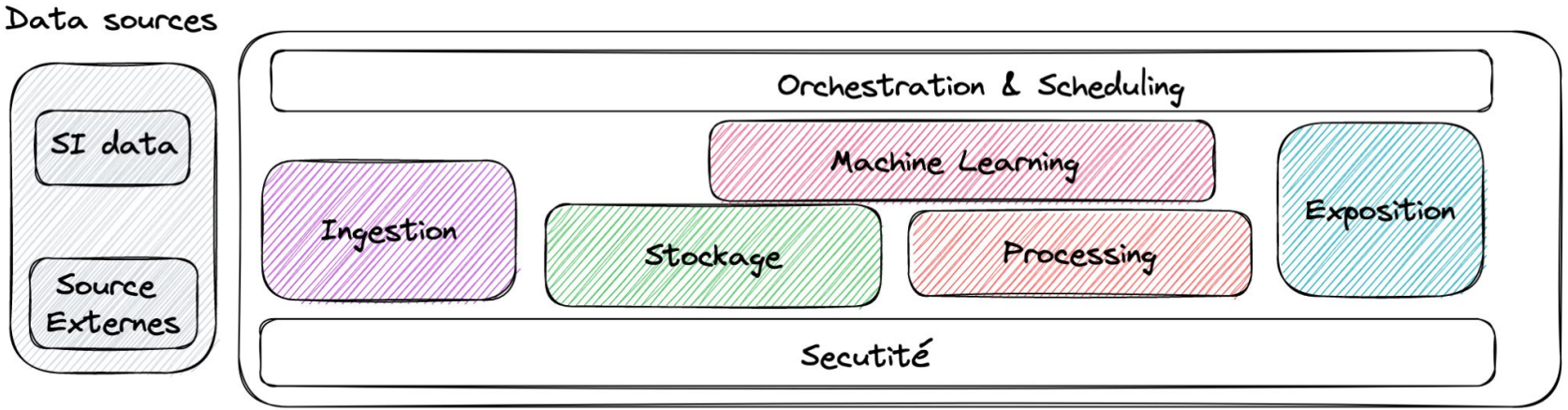

# Data Plateformes

# Définition - DataPlateformes

Une solution unifiée et complète pour ingérer, traiter, analyser et exposer  la donnée pour des besoins BI, Machine-Learning etc. La data plateforme assure également la sécurité, la conformité et la gouvernance de la donnée.

On y retrouve donc plusieurs acteurs: des Data-Engineers, des Data-Scientists, des ingénieurs devops et des data-owner, etc.

# Illustration



# Les différentes types de Data plateformes

- Le Datalake
- Le Datawarehouse
- *Le Lakehouse*

# Le datalake

Le datalake désigne un espace de stockage global des informations issues de sources internes à une organisation ou issues de sources externes, traitées au préalable ou pas.
Il s'agit de le faire avec suffisamment de flexibilité pour interagir avec les données, qu'elles soient brutes ou très raffinées.

Grâce à leur coût de stockage relativement bas, le datalake permet d'ingérer toutes les données, quels que soient leur formats, leurs types ou leurs sources.

Le stockage objet est l'outil idéal pour servir de datalake.

E L T : Extract, Load & Transform

# Les avantages et inconvénients du datalake

Les ++:

- Un modèle flexible
- A très faible coût
- Peut répondre à une multitude de cas d’usage

Les - -:

- Problème de qualité et de gouvernance de données
- Peut facilement devenir encombrant

# Le Data Warehouse

Un Data Warehouse est une base de données relationnelle hébergée sur un serveur dans un Data Center ou dans le Cloud. Il recueille des données de sources variées et hétérogènes dans le but principal de soutenir l'analyse et faciliter le processus de prise de décision.
Le Data Warehouse est donc utilisé principalement pour des besoins de BI. Il ne contient que des données pré-traitées et structurées. Les données y arrivent par un processus d’ETL (Extract, Transform and Load)

Il s'appuie sur les bases de données relationnelles.

# Les avantages et inconvénients du datalake

Les ++:

- Qualité et gouvernance des données
- Un excellent outil pour faire de la BI

Les - -:

- Un modèle de schéma rigide
- Usage limité
- coût parfois élevé (Traitement et stockage)

# Le lakehouse

L'architecture Lakehouse vise à résoudre les limitations des data lakes et des data warehouse en fournissant une plateforme centralisée et cohérente.

Elle exploite la scalabilité et la flexibilité des data lakes tout en incorporant des éléments des data warehouse tels que l'application de schémas, l'indexation et l'optimisation des requêtes.

# Hadoop

# Who Uses Hadoop?

# Définition

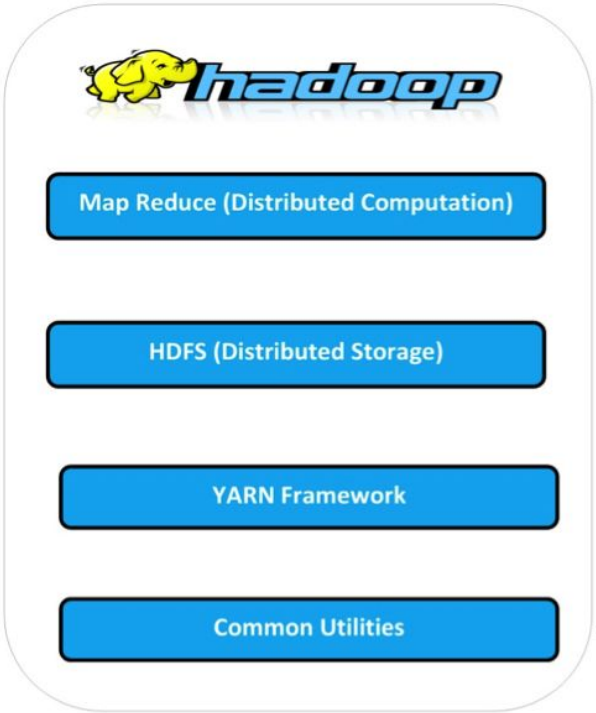
Hadoop est un framework open source permettant de stocker et de traiter de gros volumes de données de manière distribuée sur plusieurs machines.

Développé initialement chez Google, le projet est aujourd'hui en Open-Source et est maintenu par la communauté Apache.

Hadoop est constitué principalement de trois composantes:

- Yarn Manager pour la gestion des ressources du cluster
- HDFS qui est la couche de stockage de Hadoop
- Map-Reduce pour le traitement des données.

# Composants


hadoop
Map Reduce (Distributed Computation)
HDFS (Distributed Storage)
YARN Framework
Common Utilities

# Composants

**Yarn Manager** (Yet Another Resource Negotiator)**:**

Ce composant assure l’allocation et la gestion des ressources au sein du cluster (Remplacement de noeuds).

# Composants

**HDFS** (Hadoop Distributed File System)**:**

Ce composant assure le stockage des données et leurs réplications au sein des différentes machines sous formes de blocs. On a deux types de machines: le Namenode et les Workers Node.

Le Namenode (Master) qui contient les métadonnées et les workers nodes qui contiennent les données  et les réplications associées.

Les métadonnées constituent entre autre les endroits de stockage des données, les tailles des fichiers etc.

# Composants

**Map-Reduce** assure le traitement distribué des données, les tâches sont parallélisées sur les différents workers. Elle comporte deux phases:

La phase de Map qui est une étape de processing, filtre ou de tri sur l’ensemble des datapoints

La phase de Reduce qui se sert des résultats issus de la phase de Map pour ensuite faire des agrégations.

# Composants

**Map-Reduce** Exemple:

**Nombre d’occurence de chaque lettre dans un texte**

ce cours est un cours de big data

# Composants

**Map-Reduce** Exemple:

**Nombre d'occurence de chaque lettre dans un texte**

ce cours est un cours de big data

***Map***:

(c,1)  , (e,1) , (c,1), (o, 1), (u, 1), (r, 1), (s, 1), (e, 1), (s, 1), (t, 1), (u, 1), (n, 1), (c, 1), (o, 1), (u, 1), (r, 1), (s, 1), (d, 1), (e, 1), (b, 1), (i, 1), (g, 1), (d, 1), (a, 1), (t, 1), (a,1)

***Reduce*:**

(a, 2), (b, 1), (c, 3), (e, 2), (i, 1), (g, 1), (o, 2), (r, 2), (s, 2), (t,1)

# Avantages de Hadoop

- Réduction des coûts
- Distribution/parallélisation des tâches
- Résilience et tolérance aux pannes
- Capacité de stockage et de traitements de gros volumes de données de différents typologies

# **Les différents Format de stockage** (données structurées ou semi-structurées)

## - Csv

Comma separated values: “row-based”, en plus des lignes On trouve généralement une entête (header) représentant les noms des différentes colonnes.Ce format permet de représenter des données tabulaire sous forme de texte brut. On peut également utiliser d’autres séparateurs autre que la , comme ; ou \t etc.

- **Avantages:**
  - human readable
  - facile à manier/modifier/analyser
  - S'intègre facilement avec la plupart des systèmes
- **Inconvénients:**
  - Ne prend pas en charge le type des colonnes
  - Les données  peuvent facilement être corrompues (séparateurs, caractères spéciaux etc)
  - Ne permet pas la représentation hiérarchique
  - Peu adapté au big data, gourmand en taille de stockage

# **Les différents Format de stockage** (données structurées ou semi-structurées)

## - Json

JavaScript object notation: “key/value pair”, permet de représenter des données semi-structurées sous formes de paires clé/valeur, supporte la représentation hiérarchique des données.

Les documents JSON sont couramment utilisés dans la communication réseau, en particulier avec des services web basés sur REST.

- **Avantages:**
  - human readable
  - facile à manier/modifier/analyser
  - S'intègre facilement avec la plupart des systèmes
  - Représentation hiérarchique
  - Très utilisé par beaucoup de base de données
- **Inconvénients:**
  - Peut facilement être complexe à parser
  - Peu adapté au big data, gourmand en taille de stockage

# **Les différents Format de stockage** (données structurées ou semi-structurées)

## - Parquet

Pour le parquet on a un stockage en colonnes (columnar storage format). Ce format est très bien adapté au big data car une compression est automatiquement appliquée sur les données. Prend en compte les types des colonnes et plus généralement le schéma des données.

- **Avantages:**
  - Très bien adapté au big data: compression, schéma
  - Représentation hiérarchique
  - S'intègre à la plupart des systèmes big-data
- **Inconvénients:**
  - Not human readable
  - Ne prend pas en charge l’évolution de schéma

# **Les différents Format de stockage** (données structurées ou semi-structurées)

## - Avro

Row-based, le avro stocke efficacement les données au format binaire et le schéma au format JSON. Il bénéficie d'un support fiable pour l'évolution du schéma, ce qui facilite la gestion des changements dans les données au fil du temps.

- **Avantages:**
  - Très bien adapté au big data: compression, schéma
  - Représentation hiérarchique
  - Evolution de schéma supportée
  - Utilisation de plus en plus dans le domaine du streaming
- **Inconvénients:**
  - Not human readable
  - Moins adopté, pas encore intégré à certains systèmes/langage de prog

# **Les différents Format de stockage** (données structurées ou semi-structurées)

## - Orc

Orc (Optimized Row Columnar) est un format de stockage de données orienté colonnes créé et rendu open source par Hortonworks en collaboration avec Facebook.

Il contient des groupes de données par ligne appelés "stripes" et est fortement compressible, réduisant la taille des données d'origine jusqu'à 75%.

- **Avantages:**
  - Adapté au big data: compression, schéma
  - Efficace pour les applications OLAP
- **Inconvénients:**
  - Not human readable
  - Ne Prend pas en charge l’évolution de schéma
  - Moins adopté, pas encore intégré à certains systèmes/langage de prog

# extrait d’un article sur le blog de Linkedin

Les enseignements tirés de l'expérience concernant les formats de stockage de données sont les suivants :

- JSON est la norme pour la communication sur Internet.
- Parquet et Avro sont plus optimisés pour le Big Data, mais la lisibilité et la vitesse d'écriture sont assez faibles.
- Parquet est le meilleur choix en termes de performances lors du choix d'un format de stockage de données dans Hadoop, en tenant compte de facteurs tels que l'intégration avec des applications tierces, l'évolution du schéma et le support de types de données spécifiques.
- Les algorithmes de compression jouent un rôle significatif dans la réduction de la quantité de données et l'amélioration des performances.
- Apache Avro est un encodeur universel rapide pour les données structurées, garantissant de bonnes performances lors de l'accès à tous les attributs d'un enregistrement en même temps.
- Le format CSV est généralement le plus rapide à écrire, JSON est le plus facile à comprendre pour les humains, et Parquet est le plus rapide à lire pour un sous-ensemble de colonnes, tandis qu'Avro est le plus rapide à lire toutes les colonnes en une fois.
- Les formats colonnaires tels que Parquet conviennent à une ingestion rapide de données, à une récupération rapide de données aléatoires et à des analyses de données évolutives.
- Avro est généralement utilisé pour stocker des données brutes, et Parquet est utilisé pour des analyses ultérieures après prétraitement, lorsque toutes les champs ne sont pas requis.

# Quelques bonnes pratiques

- Choisir le bon format, privilégier le parquet ou le Avro, sauf quand simple ou on peut choisir le csv ou le json
- Compresser les données pour réduir la taille des fichiers donc le coût de stockage
- Partitionner les données en fonction de la manière dont ces dernières seront lues/accédées
- Mettre du versionning (si besoin)
- Mettre en place un lifecycle afin de déplacer automatiquement les données dans la bonne classe de stockage mais également de supprimer les données non utilisées

# Quiz 1

- Citez et expliquez  3 V’s parmi les caractéristiques du Big Data.

# Quiz 2

Citez les différents types de stockage et leurs spécificités respectives.

# Quiz 3

Différence(s) majeure(s) entre les bd relationnelles et bd non-relationnelles ?

# Quiz 4

Citez quelques avantages du Cloud.

# Quiz 5

Citez les trois principaux Cloud Providers

# Quiz 6

Citez les principaux types de traitement et quelques outils associés à chacun de ces types de traitement.

# Quiz 7

La différence entre un datalake et un data warehouse ?

# Quiz 8

Les différents formats de stockage de données (structurées/semi structurées)