

UNIVERSITÉ PARIS 8

CADRE LOGICIEL POUR LE BIG DATA - AUTOMNE 2021

---

## Compte rendu de projet

---

*Auteur:*  
ROQUI DAVID

*Encadrant:*  
Mme.Jaziri RAKIA

Janvier, 2022



# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Problématique</b>	<b>2</b>
<b>3</b>	<b>Les données</b>	<b>2</b>
<b>4</b>	<b>L'architecture</b>	<b>3</b>
4.1	L'installation du cluster . . . . .	4
4.1.1	Création et lancement des instances . . . . .	4
4.1.2	Configuration du cluster Hadoop . . . . .	8
4.1.3	Installation de Hive . . . . .	11
4.1.4	Installation de Sqoop . . . . .	12
4.1.5	Création de la base de données hive . . . . .	14
<b>5</b>	<b>Les approches</b>	<b>17</b>
5.1	Postulat . . . . .	17
5.2	Le produit . . . . .	17
5.3	Le lieu . . . . .	20
5.4	Sensibilité à la pub . . . . .	21
<b>6</b>	<b>Conslusion</b>	<b>21</b>

# 1 Introduction

Depuis les années 2010 la data est omniprésente chez les entreprises. Cela s'explique de part le fait qu'elle procure une quantité d'information importante à l'entreprise ce que la rend donc extrêmement rentable. En effet les données entreprises, bien stockée, utilisée et récoltée permettent de mieux cibler les clients, prédire leurs comportements et de ce fait adapter au mieux le produit vendu afin qu'ils répondent au mieux aux besoins actuels et futurs du client. C'est pourquoi il est maintenant vital pour une entreprise d'avoir une bonne politique data.

## 2 Problématique

Le profilage client existe depuis bien avant l'arrivée des solutions big data. Cependant, avant l'arrivée de ces solutions le profilage était parfois assez rudimentaire et peu précis de par le fait que les données n'étaient pas aussi bien traitées qu'aujourd'hui, le savoir faire des data scientist et analyst n'existait pas ou alors se limitait aux statistiques, etc... Mon choix c'est porté sur un dataset contenant des données clients avec divers informations sur leurs vies privées et habitudes d'achat. A partir du dataset que j'ai sélectionné on se pose la question suivante : "Est-il possible de faire un profil des clients afin de mieux répondre à leurs besoins actuels et futurs ?". L'intérêt de répondre à une telle problématique est qu'une entreprise souhaitant mieux cibler sa clientèle actuelle ou future pourra désormais avoir une vision d'ensemble du comportement d'achat des clients et donc comment répondre au mieux aux besoins de ces derniers.

## 3 Les données

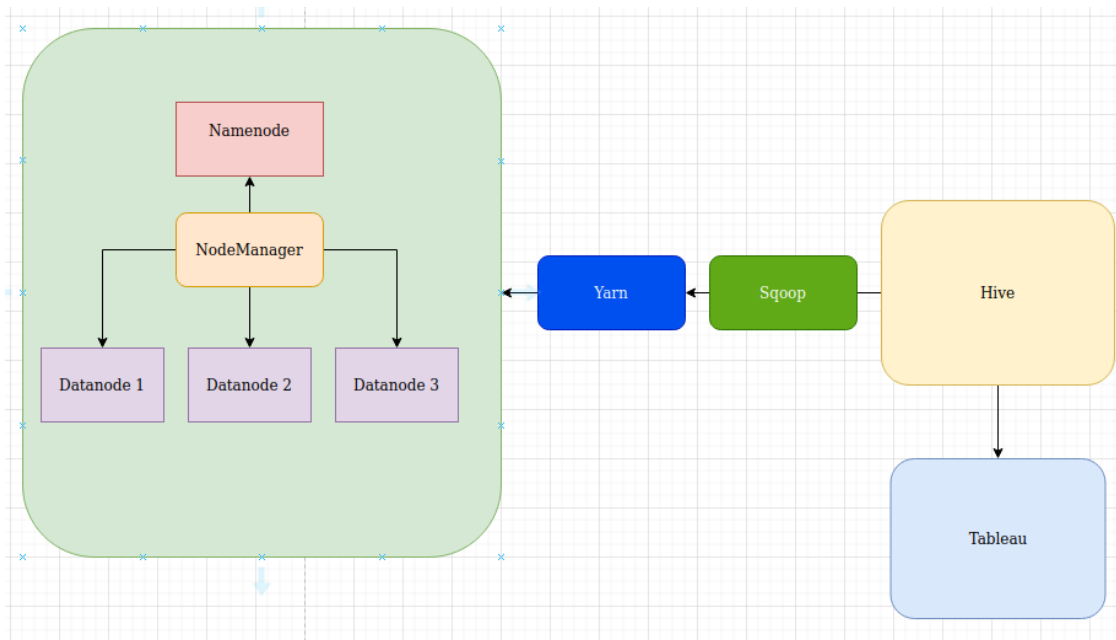
Le dataset utilisé (<https://www.kaggle.com/imakash3011/customer-personality-analysis>) contient les données d'achat de plusieurs clients et va nous permettre de prendre en entrée l'historique d'achat de plusieurs types de clients( Exemple: Diplômé de master avec deux enfants) et de voir comment ses différents types se comportent durant leurs achats afin de produire en sortie des représentations graphiques permettant de visualiser facilement quel type de client il est rentable de démarcher pour tel ou tel type de produits. Voici un tableau résumant le contenu de chaque colonne du dataset.

ID	L'identifiant du client
Year <sub>Birth</sub>	La date d'anniversaire du client.
Education	Le plus haut diplôme obtenu du client.
Marital <sub>status</sub>	Le statut matrimonial du client.
Income	Le revenu annuel du client.
Kidhome	Le nombre d'enfants du client.
Teenhome	Le nombre d'adolescents du client.
Dt <sub>Customer</sub>	La date d'embauche du client.
Recency	Le nombre de jours depuis le dernier achat du client.
MntWines	Nombre d'argent dépensé en vin (Sur deux ans).
MntFruits	Nombre d'argent dépensé en fruits (Sur deux ans).
MntMeatProducts	Nombre d'argent dépensé en viandes (Sur deux ans).
MntFishProducts	Nombre d'argent dépensé en poissons (Sur deux ans).
MntSweetProducts	Nombre d'argent dépensé en produits sucrés(On suppose ici que ce sont des bonbons ou autres confiseries sur deux ans).
MntGoldProds	Nombre d'argent dépensé en produits de luxe (Sur deux ans).

NumDealsPurchases	Nombre d'achats effectués.
NumWebPurchases	Nombre d'achats fait via internet.
NumCatalogPurchases	Nombre d'achats fait via le catalogue d'une marque.
NumStorePurchases	Nombre d'achat fait en magasin.
NumWebVisitsMonth	Nombre de visits du site web par mois.
AcceptedCmp1	Achat du produit après la première campagne publicitaire.
AcceptedCmp2	Achat du produit après la deuxième campagne publicitaire.
AcceptedCmp3	Achat du produit après la troisième campagne publicitaire.
AcceptedCmp4	Achat du produit après la quatrième campagne publicitaire.
AcceptedCmp5 text	Achat du produit après la cinquième campagne publicitaire.

## 4 L'architecture

L'architecture utilisée pour ce projet est la suivante:



- Un cloud aws comoosé de 4 machine EC2 sous Linux 16.04.
- Une base (Par base on entend la brique principale sur laquelle vont reposer les autres) Hadoop contenant:
  - Un namdenode.
  - Trois datanode.
  - Une brique hive qui va permettre de stocker les données.
  - Une brique mysql qui va nous permettre de stocker les données de façon (La base sql dans ce projet est utilisé pour pouvoir justifier l'utilisation de Sqoop, mais en soit un import csv des données directement sur Hive aurait suffi).
  - Une brique Sqoop qui va nous permettre d'importer récupérer les donées de la base de donnée mysql vers hive.

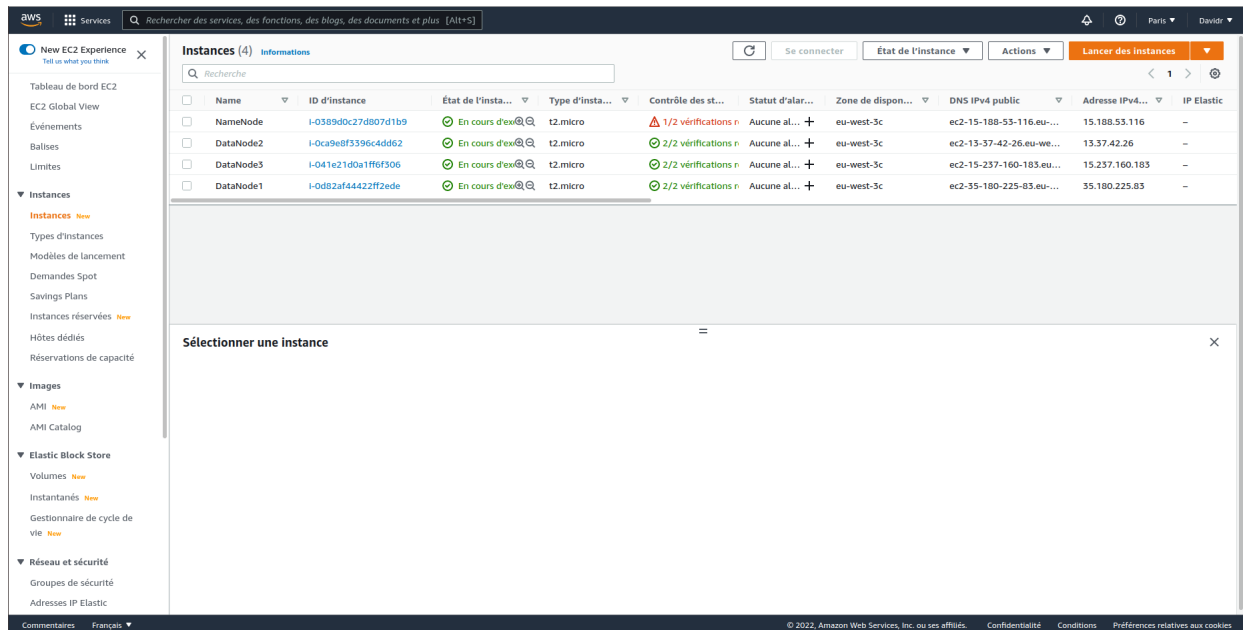
- Enfin Tableau est le dernier élément (qui n'est pas une brique) va nous permettre d'avoir une représentation graphique des résultats des différentes requêtes hive.

## 4.1 L'installation du cluster

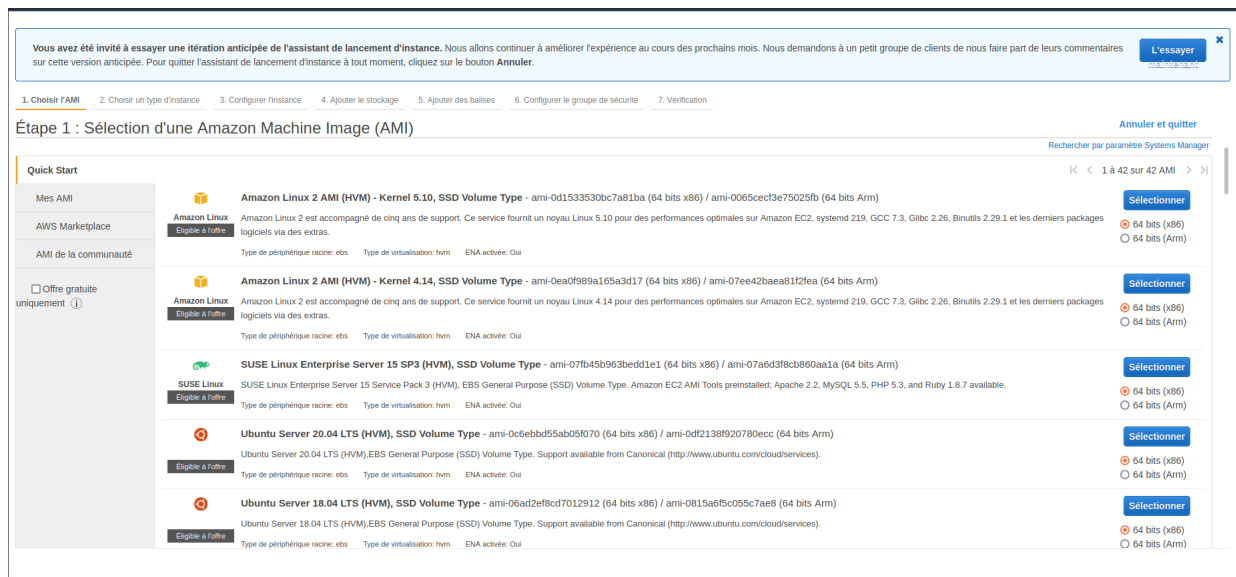
Le choix d'un cluster EC2 aws a été fait pour ce projet afin d'apprendre le fonctionnement d'un environnement cloud. La marche suivie va être détaillée dans les parties ci-dessous.

### 4.1.1 Création et lancement des instances

Dans un premier temps il s'agit de se rendre sur le panneau de configuration d'AWS EC2 et de cliquer sur "Lancer des instances"



Une fois cela fait on choisit la distribution de l'instance. Pour ce projet le choix d'Ubuntu 18.04 a été fait.



On sélectionne le type de machine qu'on souhaite utilisé. Dans ce projet les machines T2.micro on été sélectionnées de par le fait qu'elles font partis de l'offre gratuite d'AWS.

aws Services Rechercher des services, des fonctions, des blogs, des documents et plus [Alt+S]

1. Choisir l'AMI 2. Choisir un type d'instance 3. Configurer l'instance 4. Ajouter le stockage 5. Ajouter des balises 6. Configurer le groupe de sécurité 7. Vérification

### Étape 2 : Choisir un type d'instance

Amazon EC2 fournit un vaste éventail de types d'instances optimisés pour différents cas d'utilisation. Les instances sont des serveurs virtuels qui peuvent exécuter des applications. Les types d'instances se composent de différentes combinaisons de processeur, de mémoire, de stockage et de capacité réseau, et vous offrent une flexibilité dans le choix de l'association de ressources adaptées à vos applications. [En savoir plus](#) à propos des types d'instances et de la manière dont ils peuvent répondre à vos besoins informatiques.

Filtrer par: Toutes les familles d'instances Génération actuelle Afficher / Masquer les colonnes

Actuellement sélectionné : t2.micro (- ECU, 1 vCPU, 2,5 GHz, -, 1 Gio mémoire, EBS uniquement)

	Famille	Type	vCPU	Mémoire (Gio)	Stockage d'instance (Go)	Disponible en version optimisée pour EBS	Performances réseau	Prise en charge IPv6
<input type="checkbox"/>	t2	t2.nano	1	0.5	EBS uniquement	-	Faibles à modérées	Oui
<input checked="" type="checkbox"/>	t2	t2.micro <small>Eligible à l'offre gratuite</small>	1	1	EBS uniquement	-	Faibles à modérées	Oui
<input type="checkbox"/>	t2	t2.small	1	2	EBS uniquement	-	Faibles à modérées	Oui
<input type="checkbox"/>	t2	t2.medium	2	4	EBS uniquement	-	Faibles à modérées	Oui
<input type="checkbox"/>	t2	t2.large	2	8	EBS uniquement	-	Faibles à modérées	Oui
<input type="checkbox"/>	t2	t2.xlarge	4	16	EBS uniquement	-	Modérées	Oui
<input type="checkbox"/>	t2	t2.2xlarge	8	32	EBS uniquement	-	Modérées	Oui
<input type="checkbox"/>	t3	t3.nano	2	0.5	EBS uniquement	Oui	Jusqu'à 5 gigabits	Oui
<input type="checkbox"/>	t3	t3.micro	2	1	EBS uniquement	Oui	Jusqu'à 5 gigabits	Oui
<input type="checkbox"/>	t3	t3.small	2	2	EBS uniquement	Oui	Jusqu'à 5 gigabits	Oui
<input type="checkbox"/>	t3	t3.medium	2	4	EBS uniquement	Oui	Jusqu'à 5 gigabits	Oui
<input type="checkbox"/>	t3	t3.large	2	8	EBS uniquement	Oui	Jusqu'à 5 gigabits	Oui
<input type="checkbox"/>	t3	t3.xlarge	4	16	EBS uniquement	Oui	Jusqu'à 5 gigabits	Oui

Annuler Précédent Vérifier et lancer Suivant : Configurer les détails de l'instance

On configure le nombre d'instance que l'on souhaite avoir

1. Choisir l'AMI 2. Choisir un type d'instance 3. Configurer l'instance 4. Ajouter le stockage 5. Ajouter des balises 6. Configurer le groupe de sécurité 7. Vérification

### Étape 3 : Configurer les détails de l'instance

Configurez l'instance en fonction de vos besoins. Vous pouvez lancer plusieurs instances à partir de la même AMI, demander des instances Spot pour bénéficier d'un tarif inférieur, attribuer un rôle de gestion d'accès à l'instance et bien d'autres choses encore.

Nombre d'instances 1 Lancer dans le groupe Auto Scaling

Option d'achat ☐ Demander des instances Spot

Réseau vpc-0c13223a533b4fd40 (par défaut) Créer un nouveau VPC

Sous-réseau Aucune préférence (sous-réseau (subnet) par défaut) Créer un nouveau sous-réseau (subnet)

Attribuer automatiquement l'adresse IP publique Utiliser le paramètre de sous-réseau (subnet) (active)

Type de nom d'hôte Utiliser le paramètre de sous-réseau (subnet) (Nom)

DNS Hostname ☒ Enable IP name IPv4 (A record) DNS requests

☒ Activer les demandes DNS IPv4 (enregistrement A) basées sur les ressources

☐ Activer les demandes DNS IPv6 (enregistrement AAAA) basées sur les ressources

Groupe de placement ☐ Ajoutez une instance au groupe de placement.

Réserve de capacité Ouvrir

Répertoire de jonction de domaines Aucun annuaire répertoire Créer un nouveau

Rôle IAM Aucun(e) Créer un nouveau rôle IAM

Comportement d'arrêt Arrêter

Arrêt - Activer le comportement de veille prolongée ☐ Activer la mise en veille prolongée comme comportement d'arrêt supplémentaire

Annuler Précédent Vérifier et lancer Suivant : Ajouter le stockage

On attribue un groupe de sécurité à notre instance avec des règles personnalisée comme le montre l'image ci-dessous

**Étape 6 : Configurer le groupe de sécurité**

Un groupe de sécurité est un ensemble de règles de pare-feu qui contrôlent le trafic de votre instance. Sur cette page, vous pouvez ajouter des règles pour permettre qu'un trafic spécifique atteigne votre instance. Par exemple, si vous voulez configurer un serveur Web et permettre au trafic Internet d'atteindre votre instance, ajoutez des règles qui autorisent un accès restreint aux ports HTTP et HTTPS. Vous pouvez créer un nouveau groupe de sécurité ou en sélectionner un parmi les groupes existants ci-dessous. [En savoir plus](#) à propos des groupes de sécurité Amazon EC2.

Attribuer un groupe de sécurité : ☐ Créez un nouveau groupe de sécurité ☒ Sélectionnez un groupe de sécurité existant

ID de groupe de sécurité	Nom	Description	Actions
<input checked="" type="checkbox"/> sg-0fad7b81da39c3e84	default	default VPC security group	<a href="#">Copier vers le nouveau</a>
<input type="checkbox"/> sg-03c2327de6ca3b9d9	Framework_SG	projet S1.M1.framework big data	<a href="#">Copier vers le nouveau</a>
<input type="checkbox"/> sg-08e03c039b3935ac	launch-wizard-1	launch-wizard-1 created 2022-01-19T19:20:50.373+01:00	<a href="#">Copier vers le nouveau</a>
<input type="checkbox"/> sg-03f9fe6ec69ae0b48	SG	b	<a href="#">Copier vers le nouveau</a>

Règles entrantes pour sg-0fad7b81da39c3e84 (Groupes de sécurité sélectionnés : sg-0fad7b81da39c3e84)

Type	Protocole	Plage de ports	Source	Description
HTTP	TCP	80	83.202.142.124/32	
Tout le trafic	Tous	Tous	sg-0fad7b81da39c3e84 (default)	
SSH	TCP	22	83.202.142.124/32	
HTTPS	TCP	443	83.202.142.124/32	

Enfin on applique un pair de clé RSA que l'on génère via le menu "paire de clés" sur le site AWS et qui nous permet de nous connecter à nos instances et garantissant à AWS que nous sommes bien autorisé.

**Étape 7 : Examiner le lancement de l'instance**

Veuillez vérifier les détails de votre lancement d'instance. Vous pouvez revenir en arrière pour modifier les changements pour chaque section. Cliquez sur **Lancer** pour affecter une paire de clés à votre instance et terminer la procédure de lancement.

**Détails de l'AMI**

Ubuntu Server 18.04 LTS (HVM), SSD Volume Type - ami-06ad2ef...

**Type d'instance**

t2.micro

**Groupes de sécurité**

sg-0fad7b81da39c3e84

**Toutes les règles de trafic entrant sélectionnées des groupes de sécurité**

Type	Protocole	Plage de ports	Source	Description
HTTP	TCP	80	83.202.142.124/32	
Tout le trafic	Tous	Tous	sg-0fad7b81da39c3e84 (default)	
SSH	TCP	22	83.202.142.124/32	
HTTPS	TCP	443	83.202.142.124/32	

**Détails de l'instance**

Modifier l'AMI

Modifier le type d'instance

Modifier les groupes de sécurité

Modifier les détails de l'instance

Annuler Précédent Lancer

**Sélectionnez une paire de clés existante ou créez une nouvelle**

**paire de clés**

Une paire de clés se compose d'une **clé publique** détenue par AWS et d'une **clé privée, incluse dans un fichier**, que vous conservez. Ensemble, elles vous permettent de vous connecter à votre instance en toute sécurité. Avec les AMI Windows, une clé privée est requise afin d'obtenir le mot de passe utilisé pour se connecter à votre instance. Avec les AMI Linux, la clé privée vous permet d'accéder en toute sécurité à votre instance via SSH. Amazon EC2 prend en charge les types de paire de clés ED25519 et RSA.

Remarque : La paire de clés sélectionnée sera ajoutée à l'ensemble de clés autorisé pour cette instance. En savoir plus sur la suppression de paires de clés existantes d'une AMI publique.

Choisir une paire de clés existante

Sélectionner une paire de clés

AWS\_Framework | RSA

☒ Je confirme avoir accès au fichier de clé privée approprié et j'ai conscience que, sans ce dernier, je ne pourrai pas me connecter à mon instance.

Annuler Lancer des instances

Nous avons maintenant une instance créer et lancée. Nous allons maintenant installer les prérequis de base sur cette instance pour ensuite en faire une image et créer nos namenode. Pour cela on commence par ce connecter à l'instance via ssh -i et en spécifiant le chemin de la paire de clé aws.

```
hadoop@david-UX410UAR:~$ ssh -i /home/hadoop/documents/M1/PROJET/AWS/Private_keys/AWS_Framework.pem ubuntu@ec2-15-188-53-116.eu-west-3.compute.amazonaws.com
Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 5.4.0-1063-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

System Information as of Sat Jan 29 15:33:08 UTC 2022

System load:  0.03               Processes:    100
Usage of /:   13.7% of 29.02GB   Users logged in:  0
Memory usage: 35%               IP address for eth0: 172.31.34.249
Swap usage:   0%

0 updates can be applied immediately.

Last login: Sat Jan 22 11:07:29 2022 from 172.31.34.249
```

On met à jours la machine.

```
ubuntu@ip-172-31-34-249:~$ sudo apt-get update && sudo apt-get dist-upgrade
```

On installe Java.

```
ubuntu@ip-172-31-34-249:~$ sudo apt-get install openjdk-8-jdk
```

On récupère Hadoop 2.8.1.

```
ubuntu@ip-172-31-34-249:~$ wget http://apache.mirrors.tds.net/hadoop/common/hadoop-2.8.1/hadoop-2.8.1.tar.gz -P ~/Downloads
```

On l'extrait.

```
ubuntu@ip-172-31-34-249:~$ sudo tar zxvf ~/Downloads/hadoop-* -C /usr/local
```

On déplace hadoop dans son repertoire.

```
ubuntu@ip-172-31-34-249:~$ sudo mv /usr/local/hadoop-* /usr/local/hadoop
```

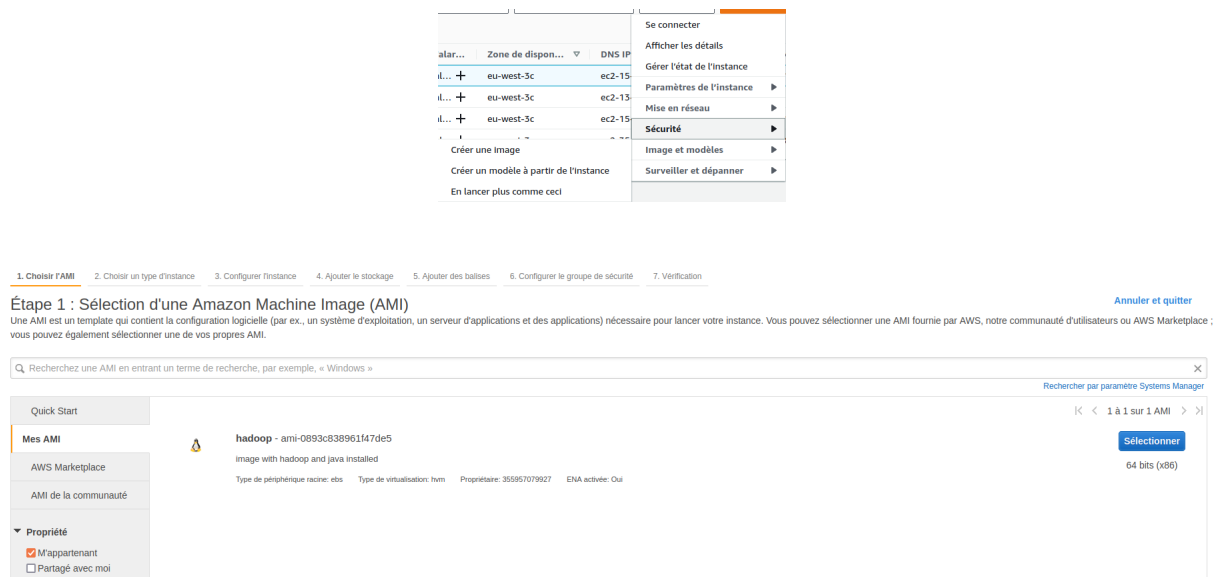
On met à jour le bashrc afin de renseigner le chemin du répertoire d'Hadoop.

```
#JAVA
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
export PATH=$PATH:$JAVA_HOME/bin

#Hadoop
export HADOOP_HOME=/usr/local/hadoop
export PATH=$PATH:$HADOOP_HOME/bin

#Hadoop conf directory
export HADOOP_CONF_DIR=/usr/local/hadoop/etc/hadoop
```

Maintenant que nous avons fait tout le nécessaire de base pour une machine EC2 d'un cluster Hadoop nous créons une image et lançons 3 autres machines (Les namenode) EC2 à partir de cette image. Nous faisons cela afin d'éviter d'avoir à faire la manipulation 3 fois de suites. La marche à suivre pour les configurations d'instances est la même que les précédentes à la différence que nous allons sélectionner l'AMI que nous venons de créer.



The screenshot displays the AWS Management Console interface for creating a new EC2 instance. The 'Choose an AMI' step is active, showing a list of available AMIs. The 'hadoop' AMI (ami-0893c838961147de5) is selected. The console provides details about the AMI, including its name, description, and properties. The 'hadoop' AMI is described as 'Image with hadoop and java installed'. The console also shows the 'Quick Start' section with 'Mes AMI' selected, and the 'Propriété' section with 'M'appartenant' checked.



1. Choisir l'AMI 2. Choisir un type d'instance 3. Configurer l'instance 4. Ajouter le stockage 5. Ajouter des balises 6. Configurer le groupe de sécurité 7. Vérification

## Étape 2 : Choisir un type d'instance

Amazon EC2 fournit un vaste éventail de types d'instances optimisés pour différents cas d'utilisation. Les instances sont des serveurs virtuels qui peuvent exécuter des applications. Les types d'instances se composent de différentes combinaisons de processeur, de mémoire, de stockage et de capacité réseau, et vous offrent une flexibilité dans le choix de l'association de ressources adaptées à vos applications. [En savoir plus](#) à propos des types d'instances et de la manière dont ils peuvent répondre à vos besoins informatiques.

Filtrer par: Toutes les familles d'instances Génération actuelle Afficher / Masquer les colonnes

Actuellement sélectionné : t2.micro (- ECU, 1 vCPU, 2.5 GHz, -, 1 Gio mémoire, EBS uniquement)								
	Famille	Type	vCPU	Mémoire (Gio)	Stockage d'instance (Go)	Disponible en version optimisée pour EBS	Performances réseau	Prise en charge IPv6
<input type="checkbox"/>	t2	t2.nano	1	0.5	EBS uniquement	-	Faibles à modérées	Oui
<input checked="" type="checkbox"/>	t2	t2.micro	1	1	EBS uniquement	-	Faibles à modérées	Oui
<input type="checkbox"/>	t2	t2.small	1	2	EBS uniquement	-	Faibles à modérées	Oui
<input type="checkbox"/>	t2	t2.medium	2	4	EBS uniquement	-	Faibles à modérées	Oui
<input type="checkbox"/>	t2	t2.large	2	8	EBS uniquement	-	Faibles à modérées	Oui
<input type="checkbox"/>	t2	t2.xlarge	4	16	EBS uniquement	-	Modérées	Oui
<input type="checkbox"/>	t2	t2.2xlarge	8	32	EBS uniquement	-	Modérées	Oui
<input type="checkbox"/>	t3	t3.nano	2	0.5	EBS uniquement	Oui	Jusqu'à 5 gigabits	Oui

aws Services Rechercher des services, des fonctions, des blogs, des documents et plus [Alt+S]

1. Choisir l'AMI 2. Choisir un type d'instance 3. Configurer l'instance 4. Ajouter le stockage 5. Ajouter des balises 6. Configurer le groupe de sécurité 7. Vérification

## Étape 3 : Configurer les détails de l'instance

Configurez l'instance en fonction de vos besoins. Vous pouvez lancer plusieurs instances à partir de la même AMI, demander des instances Spot pour bénéficier d'un tarif inférieur, attribuer un rôle de gestion d'accès à l'instance et bien d'autres choses encore.

Nombre d'instances  Lancer dans le groupe Auto Scaling

Vous pouvez envisager de lancer ces instances dans un groupe Auto Scaling pour garantir la disponibilité de l'application et pour un dimensionnement aisé à l'avenir. [Découvrez comment Auto Scaling peut aider votre application à rester saine et rentable.](#)

Nous avons maintenant 4 instances EC2 AWS de créer. Il s'agit maintenant de configurer les cluster afin de créer un cluster Hadoop/Hive/Sqoop.

### 4.1.2 Configuration du cluster Hadoop

<input type="checkbox"/>	NameNode	i-0389d0c27d807d1b9	En cours d'exé	t2.micro	2/2 vérifications	Aucune al...	eu-west-3c	ec2-15-188-53-116.eu...	15.188.53.116	-
<input type="checkbox"/>	DataNode2	i-0ca9e0f3396c4d62	En cours d'exé	t2.micro	2/2 vérifications	Aucune al...	eu-west-3c	ec2-13-37-42-26.eu-we...	13.37.42.26	-
<input type="checkbox"/>	DataNode3	i-041e21d0a1ff6f306	En cours d'exé	t2.micro	2/2 vérifications	Aucune al...	eu-west-3c	ec2-15-237-160-183.eu...	15.237.160.183	-
<input type="checkbox"/>	DataNode1	i-0d82af44422ff2ede	En cours d'exé	t2.micro	2/2 vérifications	Aucune al...	eu-west-3c	ec2-35-180-225-83.eu...	35.180.225.83	-

Afin que les machine puissent communiquer entre elles il faut utiliser le protocole SSH, afin de réaliser cela on génère une clé privé et une clé publique qui va être transmise autres autres machine en tant que "authorized key"

```
ubuntu@ip-172-31-34-249:~$ ssh-keygen -f ~/.ssh/id_rsa -t rsa -P ""
```

On ajout la clé publique aux "clés autorisées" pour chaque noeuds(Le Namenode compris)

```
cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
```

Lors du lancement d'Hadoop etc il faut que tous les noeuds se soient déjà connecté entre eux en SSH afin d'autorisé la nouvelle emprente. C'est dans cet objectif qu'on se connecte en ssh depuis le namenode en ssh à tous le datanode afin de pouvoir autoriser l'empreinte ssh

```
ubuntu@ip-172-31-34-249:~$ ssh nnode
Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 5.4.0-1063-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Sat Jan 29 15:50:58 UTC 2022

System load:  0.0               Processes:    100
Usage of /:   13.8% of 29.02GB   Users logged in: 1
Memory usage: 37%              IP address for eth0: 172.31.34.249
Swap usage:   0%

0 updates can be applied immediately.

New release '20.04.3 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Sat Jan 29 15:33:09 2022 from 83.202.142.124
```

```
ubuntu@ip-172-31-34-249:~$ ssh dnode1
Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 5.4.0-1063-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Sat Jan 29 15:51:03 UTC 2022

System load:  0.0               Processes:    123
Usage of /:   10.5% of 29.02GB   Users logged in: 0
Memory usage: 36%              IP address for eth0: 172.31.43.80
Swap usage:   0%

 * Ubuntu Pro delivers the most comprehensive open source security and
compliance features.

https://ubuntu.com/aws/pro

0 updates can be applied immediately.

Last login: Sat Jan 22 11:07:38 2022 from 172.31.34.249
```

Maintenant que les différents noeuds se connaissent, on peut commencer la configuration communes à tous les noeuds. On commence par spécifier le(s) type(s) de filesysteme

```
ubuntu@ip-172-31-43-80:~$ cd $HADOOP_CONF_DIR
ubuntu@ip-172-31-43-80:~/usr/local/hadoop/etc/hadoop$ nano core-site.xml
```

```
<configuration>
  <property>
    <name>fs.defaultFS</name>
    <value> hdfs://ec2-15-188-53-116.eu-west-3.compute.amazonaws.com:9000</value>
  </property>

  <property>
    <name>fs.AbstractFileSystem.ec2-15-188-53-116.eu-west-3.compute.amazonaws.com.impl</name>
    <value>org.apache.hadoop.fs.local.LocalFs</value>
  </property>
</configuration>
```

On configure ensuite Yarn(A noté qu'on met l'adresse DNS public du namenode dans "value").

```
<configuration>

<!-- Site specific YARN configuration properties -->

<property>
  <name>yarn.nodemanager.aux-services</name>
  <value>mapreduce_shuffle</value>
</property>

<property>
  <name>yarn.resourcemanager.hostname</name>
  <value>ec2-15-188-53-116.eu-west-3.compute.amazonaws.com</value>
</property>

</configuration>
```

On copie la configuration de base de mapreduce (le template) dans le fichier de configuration de mapreduce.

```
ubuntu@ip-172-31-43-86:/usr/local/hadoop/etc/hadoop$ sudo cp mapred-site.xml.template mapred-site.xml
```

On configure mapreduce avec l'adresse DNS public du namenode afin de setup le jobtracker.

```
<configuration>
<property>
  <name>mapreduce.jobtracker.address</name>
  <value>ec2-15-188-53-116.eu-west-3.compute.amazonaws.com:54311</value>
</property>
<property>
  <name>mapreduce.framework.name</name>
  <value>yarn</value>
</property>

</configuration>
```

La configuration communes est maintenant terminé, il s'agit maintenant de configurer le Namenode. On configure ensuite le hdfs-site.xml afin de renseigner le nombre de réplication (Le nombre de namenode)

```
<configuration>
<property>
  <name>dfs.replication</name>
  <value>3</value>
</property>
<property>
  <name>dfs.namenode.name.dir</name>
  <value>file:///usr/local/hadoop/data/hdfs/namenode</value>
</property>
</configuration>
```

On créer un répertoire pour stocker les données

```
ubuntu@ip-172-31-43-80:/usr/local/hadoop/etc/hadoop$ sudo mkdir -p $HADOOP_HOME/data/hdfs/namenode
```

On créer un fichier slaves dans lequel on renseigne tout les "esclaves" soit les datanodes. On fait de même pour le maitre avec un fichier master (Non représenter ici afin d'éviter des redondances).

```
localhost
ec2-35-180-225-83.eu-west-3.compute.amazonaws.com
ec2-13-37-42-26.eu-west-3.compute.amazonaws.com
ec2-15-237-160-183.eu-west-3.compute.amazonaws.com
```

On arrive maintenant à la configuration des datanodes. On commence par la modification du fichier `hdfs-site.xml`.

```
<configuration>
<property>
  <name>dfs.replication</name>
  <value>3</value>
</property>
<property>
  <name>dfs.namenode.name.dir</name>
  <value>file:///usr/local/hadoop/data/hdfs/namenode</value>
</property>
</configuration>
```

On créer un répertoire pour les données.

```
ubuntu@ip-172-31-43-80:/usr/local/hadoop/etc/hadoop$ sudo mkdir -p $HADOOP_HOME/data/hdfs/datanode
```

On accorde les droits afin d'éviter des problèmes d'accès lors du lancement du cluster.

```
ubuntu@ip-172-31-43-80:/usr/local/hadoop/etc/hadoop$ sudo chown -R ubuntu $HADOOP_HOME
```

Nous avons maintenant finit la configuration globale, namenode et datanode. De ce fait nous pouvons donc maintenant lancer le cluster en executant le formatage, puis en lançant dfs, yarn et le jobhistory

```
ubuntu@ip-172-31-43-80:/usr/local/hadoop/etc/hadoop$ hdfs namenode -format^C
ubuntu@ip-172-31-43-80:/usr/local/hadoop/etc/hadoop$ $HADOOP_HOME/sbin/start-dfs.sh^C
ubuntu@ip-172-31-43-80:/usr/local/hadoop/etc/hadoop$ $HADOOP_HOME/sbin/start-yarn.sh^C
ubuntu@ip-172-31-43-80:/usr/local/hadoop/etc/hadoop$ $HADOOP_HOME/sbin/mr-jobhistory-daemon.sh start historyserver^C
```

Quand on regarde ce que retourne la commande `jps` du côté namenode on voit bien que tous les composants sont actifs et de même du côté d'un datanode.

```
ubuntu@ip-172-31-34-249:~$ $HADOOP_HOME/sbin/start-dfs.sh
Starting namenodes on [ec2-15-188-53-116.eu-west-3.compute.amazonaws.com]
ec2-15-188-53-116.eu-west-3.compute.amazonaws.com: starting namenode, logging to /usr/local/hadoop/logs/hadoop-ubuntu-namenode-ip-172-31-34-249.out
ec2-35-180-225-83.eu-west-3.compute.amazonaws.com: starting datanode, logging to /usr/local/hadoop/logs/hadoop-ubuntu-datanode-ip-172-31-43-80.out
ec2-15-237-160-183.eu-west-3.compute.amazonaws.com: starting datanode, logging to /usr/local/hadoop/logs/hadoop-ubuntu-datanode-ip-172-31-45-128.out
ec2-13-37-42-26.eu-west-3.compute.amazonaws.com: starting datanode, logging to /usr/local/hadoop/logs/hadoop-ubuntu-datanode-ip-172-31-42-160.out
Starting secondary namenodes [0.0.0.0]
0.0.0.0: starting secondarynamenode, logging to /usr/local/hadoop/logs/hadoop-ubuntu-secondarynamenode-ip-172-31-34-249.out
ubuntu@ip-172-31-34-249:~$ $HADOOP_HOME/sbin/start-yarn.sh
starting yarn daemons
starting resourcemanager, logging to /usr/local/hadoop/logs/yarn-ubuntu-resourcemanager-ip-172-31-34-249.out
ec2-13-37-42-26.eu-west-3.compute.amazonaws.com: starting nodemanager, logging to /usr/local/hadoop/logs/yarn-ubuntu-nodemanager-ip-172-31-42-160.out
ec2-15-237-160-183.eu-west-3.compute.amazonaws.com: starting nodemanager, logging to /usr/local/hadoop/logs/yarn-ubuntu-nodemanager-ip-172-31-45-128.out
ec2-35-180-225-83.eu-west-3.compute.amazonaws.com: starting nodemanager, logging to /usr/local/hadoop/logs/yarn-ubuntu-nodemanager-ip-172-31-43-80.out
ubuntu@ip-172-31-34-249:~$ $HADOOP_HOME/sbin/mr-jobhistory-daemon.sh start historyserver
starting historyserver, logging to /usr/local/hadoop/logs/mapred-ubuntu-historyserver-ip-172-31-34-249.out
ubuntu@ip-172-31-34-249:~$ jps
2435 ResourceManager
2275 SecondaryNameNode
2007 NameNode
2727 JobHistoryServer
2764 Jps
```

```
ubuntu@ip-172-31-43-80:~$ jps
2500 NodeManager
2703 Jps
```

#### 4.1.3 Installation de Hive

J'ai fais le choix de prendre hive 2.3.9 (Petite erreur lors de la capture d'écran) qui est une version compatible avec Hadoop 2.8.1. A noté que Hive n'a besoin d'être installé que sur le namenode. On commence par récupérer hive via le dépôt puis on le décompresse.

```
ubuntu@ip-172-31-34-249:/opt$ wget https://dlcdn.apache.org/hive/hive-3.1.2/apache-hive-3.1.2-bin.tar.gz ^C
ubuntu@ip-172-31-34-249:/opt$ sudo tar -xzf apache-hive-2.3.9-bin.tar.gz ^C
ubuntu@ip-172-31-34-249:/opt$ sudo mv apache-hive-2.3.9-bin hive-2.3.9^C
```

On met à jour le chemin de hive dans bashrc.

```
#Hive
export HIVE_HOME=/opt/hive-2.3.9
export PATH=$HIVE_HOME/bin:$PATH
```

On initialise le type de schema de la base donnée hive.

```
ubuntu@ip-172-31-34-249:/opt$ schematool -initSchema -dbType derby
```

#### 4.1.4 Installation de Sqoop

Afin de récupérer les données d'une base de donnée externe et de les importer sur hdfs ou hive le choix d'utiliser Sqoop a été fait. Afin d'installer Sqoop on commence par télécharger l'archive sur le dépôt. On décompresse l'archive et on la déplace dans le répertoire sqoop

```
ubuntu@ip-172-31-34-249:/usr/local$ wget https://archive.apache.org/dist/sqoop/1.4.6/sqoop-1.4.6.bin__hadoop-2.0.4-alpha.tar.gz ^C
ubuntu@ip-172-31-34-249:/usr/local$ tar -xzf ^Cooop-1.4.6.bin__hadoop-2.0.4-alpha.tar.gz
ubuntu@ip-172-31-34-249:/usr/local$ mv /sqoop-1.4.6.bin__hadoop-2.0.4-alpha /usr/local/sqoop/^C
```

On copie le template de base de sqoop dans le fichier de configuration d'environnement de sqoop.

```
#Sqoop
export SQOOP_HOME=/usr/local/sqoop/sqoop-1.4.6.bin__hadoop-2.0.4-alpha
export PATH=$PATH:$SQOOP_HOME/bin
```

On modifie le fichier d'environnement de Sqoop en renseignant les chemins utiles pour le bon fonctionnement de Sqoop en fonction des briques présentes au sein de notre architecture. Pour ce projet j'ai fais le choix d'importer les données d'une base mysql vers hdfs puis de hdfs vers hive il n'y a donc besoin de renseigner uniquement le chemin hadoop.

```
ubuntu@ip-172-31-34-249:/usr/local/sqoop/sqoop-1.4.6.bin__hadoop-2.0.4-alpha/conf$ cp sqoop-env-template.sh sqoop-env.sh
```

```

# Licensed to the Apache Software Foundation (ASF) under one or more
# contributor license agreements. See the NOTICE file distributed with
# this work for additional information regarding copyright ownership.
# The ASF licenses this file to You under the Apache License, Version 2.0
# (the "License"); you may not use this file except in compliance with
# the License. You may obtain a copy of the License at
#
# http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.
#
# included in all the hadoop scripts with source command
# should not be executable directly
# also should not be passed any arguments, since we need original $*
#
# Set Hadoop-specific environment variables here.

#Set path to where bin/hadoop is available
export HADOOP_COMMON_HOME=/usr/local/hadoop

#Set path to where hadoop-*-core.jar is available
export HADOOP_MAPRED_HOME=/usr/local/hadoop

#set the path to where bin/hbase is available
#export HBASE_HOME=

#Set the path to where bin/hive is available
#export HIVE_HOME=

#Set the path for where zookeeper config dir is
#export ZOO_CFG_DIR=

```

On créer un répertoire afin de stocker les "sqoop work"

```

ubuntu@ip-172-31-34-249:/usr/local/sqoop/sqoop-1.4.6.bin__hadoop-2.0.4-alpha$ cd $SQOOP_HOME
ubuntu@ip-172-31-34-249:/usr/local/sqoop/sqoop-1.4.6.bin__hadoop-2.0.4-alpha$ mkdir sqoop_work/

```

Enfin on vérifie que Sqoop est bien installé (Ici la version 1.4.6).

```

ubuntu@ip-172-31-34-249:/usr/local/sqoop/sqoop-1.4.6.bin__hadoop-2.0.4-alpha$ sqoop-version
Warning: /usr/local/sqoop/sqoop-1.4.6.bin__hadoop-2.0.4-alpha/./hbase does not exist! HBase imports will fail.
Please set $HBASE_HOME to the root of your HBase installation.
Warning: /usr/local/sqoop/sqoop-1.4.6.bin__hadoop-2.0.4-alpha/./hcatalog does not exist! HCatalog jobs will fail.
Please set $HCAT_HOME to the root of your HCatalog installation.
Warning: /usr/local/sqoop/sqoop-1.4.6.bin__hadoop-2.0.4-alpha/./accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
Warning: /usr/local/sqoop/sqoop-1.4.6.bin__hadoop-2.0.4-alpha/./zookeeper does not exist! Accumulo imports will fail.
Please set $ZOOKEEPER_HOME to the root of your Zookeeper installation.
22/01/29 16:49:28 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6
Sqoop 1.4.6
git commit id c0c5a81723759fa575844a0a1eae8f510fa32c25
Compiled by root on Mon Apr 27 14:38:36 CST 2015

```

Afin que Sqoop puisse importer les données il faut garantir une connexion entre lui et la base de données. Pour cela il s'agit d'installer JDBC driver ainsi que Ojdbc et des les ajouter aux fichiers lib (libraries) afin que Sqoop puisse utiliser leurs fonctionnalités de connexion à la base de données mysql.

```
#JDBC DRIVER
cd /usr/local/sqoop
wget http://cdn.mysql.com/Downloads/Connector-J/mysql-connector-java-5.0.8.tar.gz
tar -xzf mysql-connector-java-5.0.8.tar.gz >> /dev/null
mv mysql-connector-java-5.0.8/mysql-connector-java-5.0.8-bin.jar /usr/local/sqoop/lib/
rm -rf mysql-connector-java-5.0.8
rm mysql-connector-java-5.0.8.tar.gz

#OJDBC
wget http://download.oracle.com/otn/utilities_drivers/jdbc/11204/ojdbc6.jar?AuthParam=1472102440_acf2d63e2d3673651122947bf8cba738
mv ojdbc6.jar?AuthParam=1472102440_acf2d63e2d3673651122947bf8cba738 ojdbc6.jar
mv ojdbc6.jar /usr/local/sqoop/lib/
```

Le cluster AWS contient maintenant une brique Hadoop, Hive et Sqoop. Malheureusement comme évoqué en cours AWS ayant changé sa politique il n'est pas possible d'exécuter le projet sur ce cluster, car les machines T2.micro refusent ce genre de traitement. J'ai donc décidé de basculé sur une installation locale en suivant les même démarches que décrivent précédemment à la différence que je suis en mono-noeud.

#### 4.1.5 Création de la base de données hive

Afin de stocker les données une base de données hive a été créée. Elle contient deux tables. Une table temporaire ayant pour but de stocker toutes les données du sqoop import et une autre définitive étant quant à elle partitionné par le niveau universitaire (colonne Education) afin d'avoir des groupes prédéfinis en plus d'améliorer le temps d'exécution des requête.

Dans un premier temps les données sont chargées sur un base de données mysql en local

nom	type	intercasement	AUTO_INCREMENT	Actions
1 id_prima	int(11)	Non	Aucune	Modifier Supprimer Primaire Unique Index Spatial Texte entier Valeurs distinctes
2 ID	int(11)	Oui	NULL	Modifier Supprimer Primaire Unique Index Spatial Texte entier Valeurs distinctes
3 Year_Birth	int(50)	Oui	NULL	Modifier Supprimer Primaire Unique Index Spatial Texte entier Valeurs distinctes
4 Education	varchar(255)	Oui	NULL	Modifier Supprimer Primaire Unique Index Spatial Texte entier Valeurs distinctes
5 Marital_Status	varchar(255)	Oui	NULL	Modifier Supprimer Primaire Unique Index Spatial Texte entier Valeurs distinctes
6 Income	int(50)	Oui	NULL	Modifier Supprimer Primaire Unique Index Spatial Texte entier Valeurs distinctes
7 Kidhome	int(50)	Oui	NULL	Modifier Supprimer Primaire Unique Index Spatial Texte entier Valeurs distinctes
8 Teenhome	int(50)	Oui	NULL	Modifier Supprimer Primaire Unique Index Spatial Texte entier Valeurs distinctes
9 Dt_Customer	varchar(250)	Oui	NULL	Modifier Supprimer Primaire Unique Index Spatial Texte entier Valeurs distinctes
10 Recency	int(50)	Oui	NULL	Modifier Supprimer Primaire Unique Index Spatial Texte entier Valeurs distinctes
11 MntWines	int(50)	Oui	NULL	Modifier Supprimer Primaire Unique Index Spatial Texte entier Valeurs distinctes
12 MntFruits	int(50)	Oui	NULL	Modifier Supprimer Primaire Unique Index Spatial Texte entier Valeurs distinctes
13 MntMeatProducts	int(50)	Oui	NULL	Modifier Supprimer Primaire Unique Index Spatial Texte entier Valeurs distinctes
14 MntFishProducts	int(50)	Oui	NULL	Modifier Supprimer Primaire Unique Index Spatial Texte entier Valeurs distinctes
15 MntSweetProducts	int(50)	Oui	NULL	Modifier Supprimer Primaire Unique Index Spatial Texte entier Valeurs distinctes
16 MntGoldProds	int(50)	Oui	NULL	Modifier Supprimer Primaire Unique Index Spatial Texte entier Valeurs distinctes
17 NumDealsPurchases	int(50)	Oui	NULL	Modifier Supprimer Primaire Unique Index Spatial Texte entier Valeurs distinctes
18 NumWebPurchases	int(50)	Oui	NULL	Modifier Supprimer Primaire Unique Index Spatial Texte entier Valeurs distinctes
19 NumCatalogPurchases	int(50)	Oui	NULL	Modifier Supprimer Primaire Unique Index Spatial Texte entier Valeurs distinctes
20 NumStorePurchases	int(50)	Oui	NULL	Modifier Supprimer Primaire Unique Index Spatial Texte entier Valeurs distinctes
21 NumWebVisitsMonth	int(50)	Oui	NULL	Modifier Supprimer Primaire Unique Index Spatial Texte entier Valeurs distinctes
22 AcceptedCmp3	int(50)	Oui	NULL	Modifier Supprimer Primaire Unique Index Spatial Texte entier Valeurs distinctes
23 AcceptedCmp4	int(50)	Oui	NULL	Modifier Supprimer Primaire Unique Index Spatial Texte entier Valeurs distinctes
24 AcceptedCmp5	int(50)	Oui	NULL	Modifier Supprimer Primaire Unique Index Spatial Texte entier Valeurs distinctes
25 AcceptedCmp1	int(50)	Oui	NULL	Modifier Supprimer Primaire Unique Index Spatial Texte entier Valeurs distinctes
26 AcceptedCmp2	int(50)	Oui	NULL	Modifier Supprimer Primaire Unique Index Spatial Texte entier Valeurs distinctes
27 Complain	int(50)	Oui	NULL	Modifier Supprimer Primaire Unique Index Spatial Texte entier Valeurs distinctes
28 Z_CostContact	int(50)	Oui	NULL	Modifier Supprimer Primaire Unique Index Spatial Texte entier Valeurs distinctes
29 Z_Revenue	int(50)	Oui	NULL	Modifier Supprimer Primaire Unique Index Spatial Texte entier Valeurs distinctes
30 Response	int(50)	Oui	NULL	Modifier Supprimer Primaire Unique Index Spatial Texte entier Valeurs distinctes

phpMyAdmin

Les données de cette base sont ensuite extraites via Sqoop et importer sur le hdfs via un script bash

```
#On supprime le repertoire de sotckage s'il existe
hdfs dfs -rm -r /user/hadoop/data_sqoop

#On importe les données de mysql vers hdfs via sqoop (Changer par les bons identifiants)
sqoop import --connect jdbc:mysql://localhost/Framework_big_data --table data_sqoop --username root --password Rowe0169007157*
```

Une fois que mapreduced a terminé son "job" 4 les données sont chargées dans le hdfs et ce en 4 partie. Un script hql est ensuite appliqué pour:

- Créer la table temporaire.
- Charger les données dans la table temporaire.
- Créer la table finale partitionnée.
- Insérer les données de la table temporaire vers la table partitionnée.



```

CREATE DATABASE IF NOT EXISTS projet_framework_big_data;

use projet_framework_big_data;

DROP TABLE IF EXISTS dataset;
DROP TABLE IF EXISTS dataset_partitioned;
set hive.exec.dynamic.partition.mode=nonstrict;
create table dataset(id_prima int,
                    ID int,
                    Year_Birth int,
                    Education string,
                    Marital_Status string,
                    Income int,
                    Kidhome int,
                    Teenhome int,
                    Dt_Customer string,
                    Recency int,
                    MntWines int,
                    MntFruits int,
                    MntMeatProducts int,
                    MntFishProducts int,
                    MntSweetProducts int,
                    MntGoldProds int,
                    NumDealsPurchases int,
                    mWebPurchases int,
                    NumCatalogPurchases int,
                    NumStorePurchases int,
                    NumWebVisitsMonth int,
                    AcceptedCmp3 int,
                    AcceptedCmp4 int,
                    AcceptedCmp5 int,
                    AcceptedCmp1 int,
                    AcceptedCmp2 int,
                    Complain int,
                    Z_CostContact int,
                    Z_Revenue int,
                    Response int)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ',';

```

```

create table dataset_partitioned(id_prima int,
                                ID int,
                                Year_Birth int,
                                Marital_Status string,
                                Income int,
                                Kidhome int,
                                Teenhome int,
                                Dt_Customer string,
                                Recency int,
                                MntWines int,
                                MntFruits int,
                                MntMeatProducts int,
                                MntFishProducts int,
                                MntSweetProducts int,
                                MntGoldProds int,
                                NumDealsPurchases int,
                                mWebPurchases int,
                                NumCatalogPurchases int,
                                NumStorePurchases int,
                                NumWebVisitsMonth int,
                                AcceptedCmp3 int,
                                AcceptedCmp4 int,
                                AcceptedCmp5 int,
                                AcceptedCmp1 int,
                                AcceptedCmp2 int,
                                Complain int,
                                Z_CostContact int,
                                Z_Revenue int,
                                Response int)
PARTITIONED BY (Education string)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ',';
|

```

```

load data infile '/user/hadoop/data_sqoop/part-m-00000' into table dataset;
load data infile '/user/hadoop/data_sqoop/part-m-00001' into table dataset;
load data infile '/user/hadoop/data_sqoop/part-m-00002' into table dataset;
load data infile '/user/hadoop/data_sqoop/part-m-00003' into table dataset;

INSERT OVERWRITE TABLE dataset_partitioned PARTITION(Education) SELECT id_prima,
ID,
Year_Birth,
Marital_Status,
Income,
Kidhome,
Teenhome,
Dt_Customer,
Recency,
MntWines,
MntFruits,
MntMeatProducts,
MntFishProducts,
MntSweetProducts,
MntGoldProds,
NumDealsPurchases,
MWebPurchases,
NumCatalogPurchases,
NumStorePurchases,
NumWebVisitsMonth,
AcceptedCmp3,
AcceptedCmp4,
AcceptedCmp5,
AcceptedCmp1,
AcceptedCmp2,
Complain,
Z_CostContact,
Z_Revenue,
Response,
Education
FROM dataset;

```

## 5 Les approches

Nous avons maintenant notre architecture Hadoop/Hive/Sqoop d'installer avec les données du dataset chargée sur hive et partitionné selon le niveau universitaire. Il s'agit maintenant de réfléchir à différentes analyses afin de savoir quel type de profil ciblé pour quel type de produit et retourner les différents résultats de façon graphique via la solution "Tableau".

### 5.1 Postulat

Un client peut-être caractériser par un certains nombres de critères. Cependant, il y a dans ce dataset deux critères qui prédominent. Ces derniers sont:

- Le niveau universitaire
- Le nombre d'enfants/adolescent présent au sein du foyer du client

Ces deux critères sont les deux critères à partir des quels le résultats vont être groupé. A noté qu'il y a pleins d'approche possibles, j'ai ici fait le choix de prendre le niveau universitaire et le nombre d'enfant, car ce sont ces deux critères qui impactent le plus le niveau de revenu et les dépenses mensuelles. Cependant, on aurait aussi très bien pu imaginer une approche ou le status matrimonial à un impact sur le comportement d'achat (Exemple: A la saint-Valentin, il ne faut cibler que les personne marié ou en couple et délaissier les personnes célibaire d'un point de vue marketing, etc...).

### 5.2 Le produit

La première étape pour optimiser la vente d'un produit est de savoir quel produit vendre et à qui. C'est pour cela on applique un script hql qui va selectionner les colonnes nous intéressant et faire un ratio achats web/achats magasin (Il n'y aura qu'une seule capture d'écran pour montrer le schéma classique du script et des requêtes afin d'éviter des redondances)

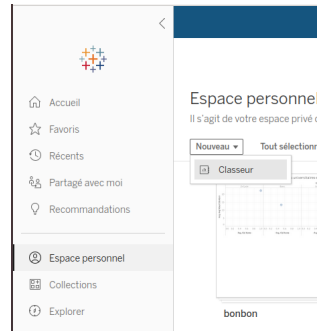
```

use projet_framework_btg_data;

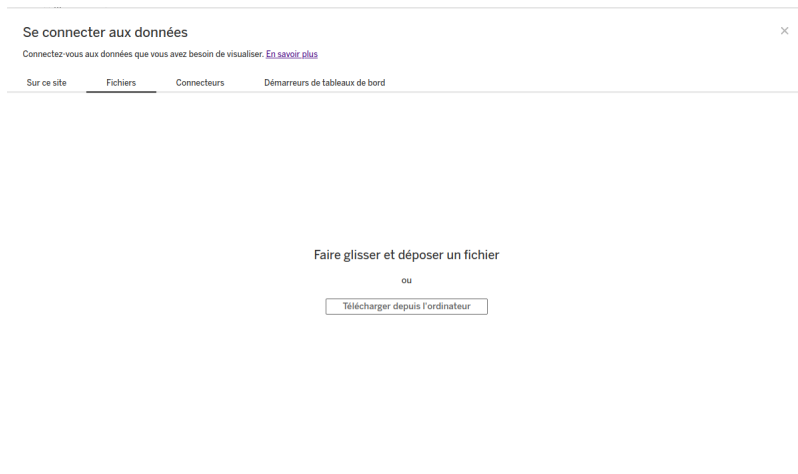
INSERT OVERWRITE LOCAL DIRECTORY '/home/hadoop/Documents/M1/PROJET/rapport/script/hql_script/result' ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' select Education,Kidhome,
round(percentile(MWebPurchases,0.5)/percentile(NumStorePurchases,0.5),2) from dataset_partitioned group by Education, Kidhome,Teenhome;

```

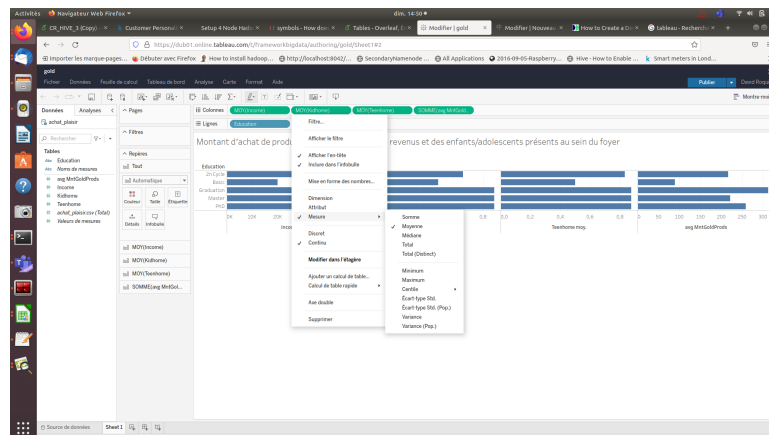
Cela nous donne un fichier txt, que je modifie en fichier csv avec les bons noms de colonne. Une fois cela fait je me rends sur mon espace personnel Tableau et je crée un nouveau classeur.

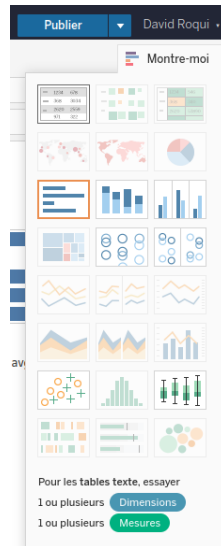


J'importe mes données.

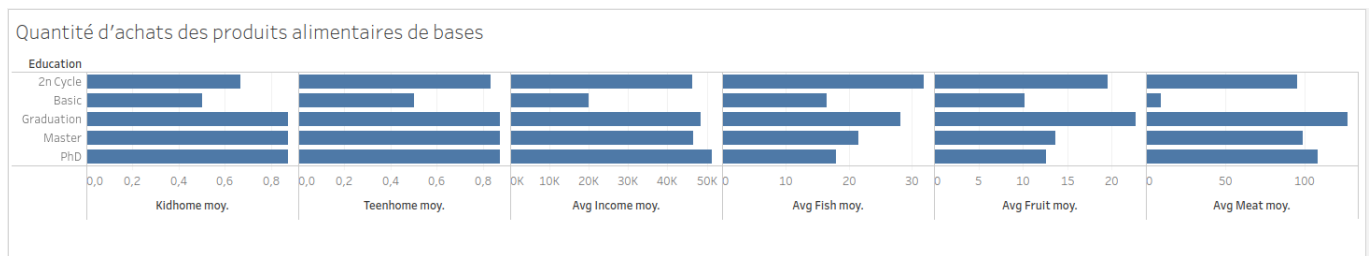


Et enfin je sélectionne mon mode d'affichage, mes valeurs, mes calculs etc...

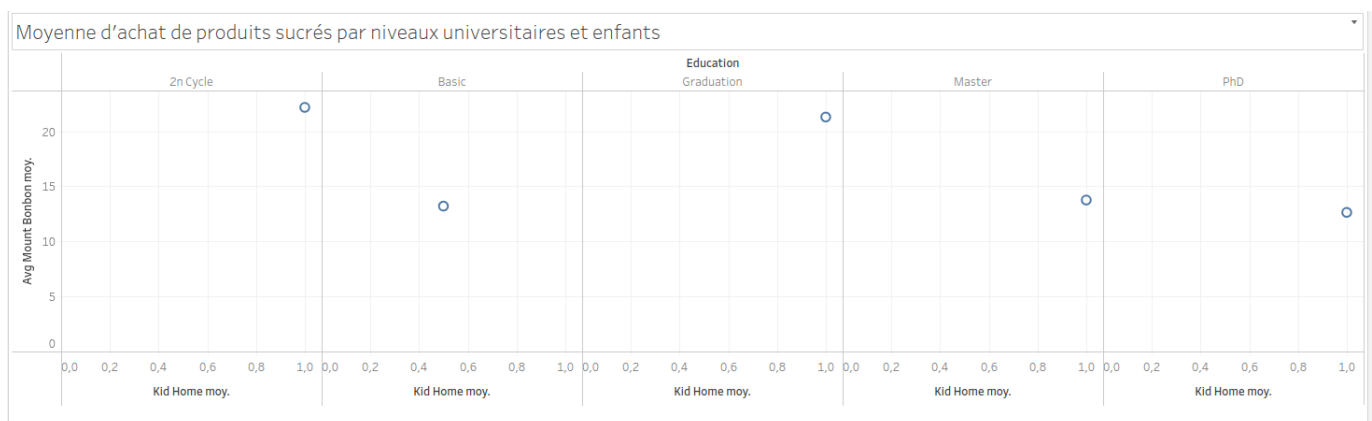




Il ressort de ce graphique que le client moyen ayant un niveau universitaire "Graduation" consomme le plus de produits. Notamment au niveau des produits viandes et poissons. Sachant que ces produits valent chers il peut-être intéressante de se concentrer sur la population "graduation"



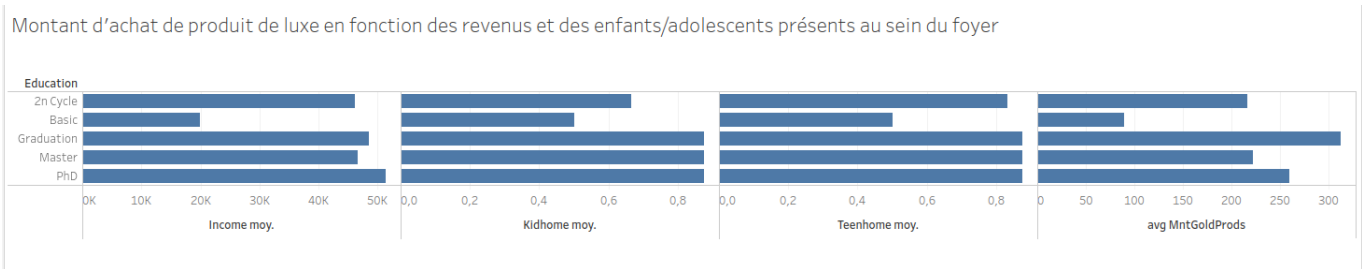
On s'intéresse maintenant à la quantité d'achat de produit sucrés (bonbon, friandises, etc...) afin de savoir si les enfants ont un si fort impact sur l'achat de ses produits et si oui quelle tranche de la population ciblée.



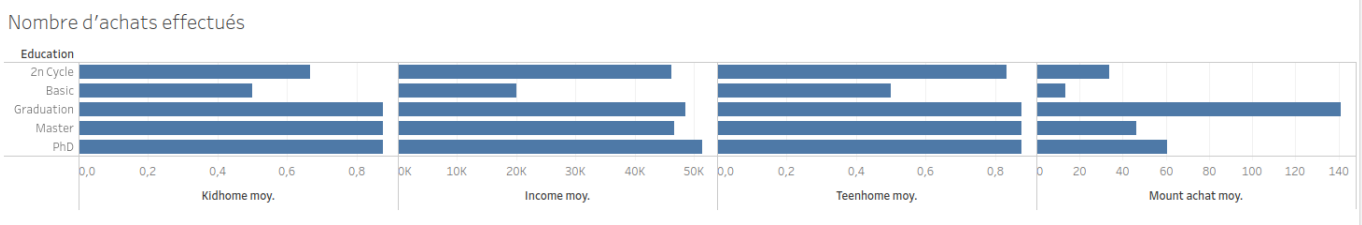
Il ressort de ce graphique qu'en effet comme on pouvait s'en douter avoir un enfant impacte sur la quantité de produit sucré acheté. Cependant, encore une fois malgré le fait que les détenteurs de PhD ont une meilleure revenu ils consomment moins que les personnes ayant "graduation" ou "2n cycle".

On souhaite maintenant savoir quelle tranche de la population est la plus sensible aux achats "plaisirs" ici les achats plaisir sont représentés par les achats contenant de l'or donc plus que des achats plaisir ce sont

des achats de luxe. Il ressort de ce graphique que malgré le fait que les populations phd et master consomme certes moins de produits alimentaires, mais dépense plus que les autres populations en ce qui concerne les achats de luxe/plaisir. Cette tranche de population est donc a visé pour ce type d'achat.



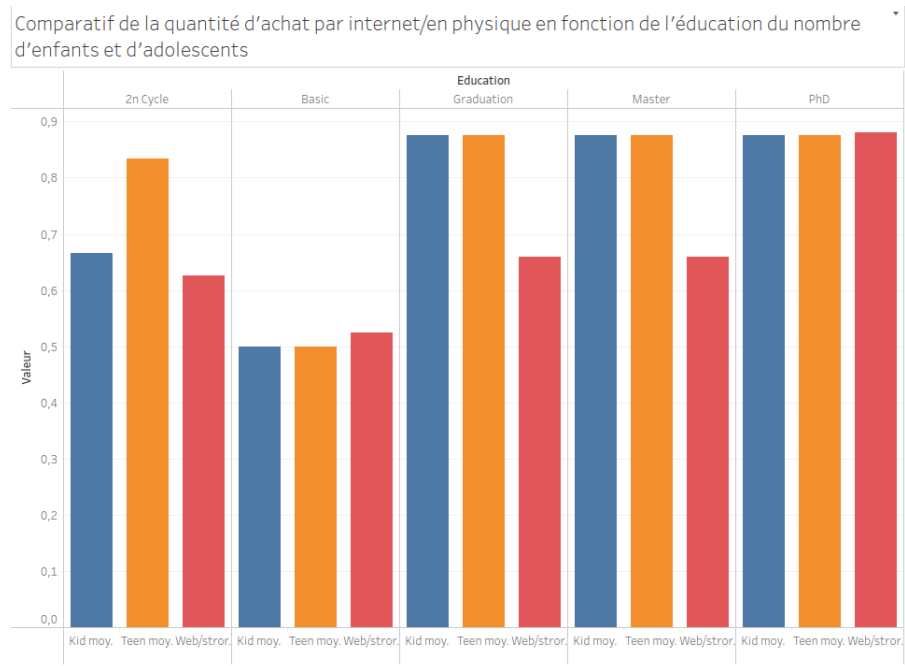
Enfin il est important de savoir combien de fois une population achète afin de savoir si cette dite population est plus enclin aux gros achats, mais rares ou bien à plusieurs petits d'achats. Déterminé sa mentalité en somme.



Le constat reste le même que pour les analyses précédentes. La population graduation effectue beaucoup d'achat au terme d'une année comparée à la moyenne des autres populations.

### 5.3 Le lieu

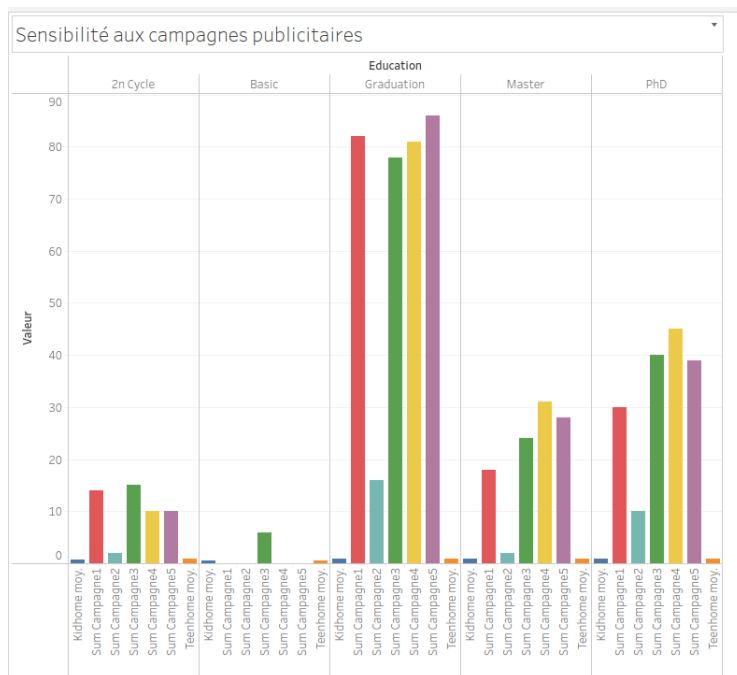
Le rendu graphique du ratio achat web/ achat magasin est le suivant:



On remarque que plus le niveau d'étude augmente plus la quantité d'achat par internet augmente. Cela peut s'expliquer de par le fait que plus le niveau universitaire est élevé, plus la probabilité d'occuper un poste à responsabilité est grande. Sachant qu'un post à responsabilité demande beaucoup d'investissement on peut imaginer que le client préfère utiliser son temps libre à autre chose que se déplacer en magasin et donc de ce fait utilise les achats web.

## 5.4 Sensibilité à la pub

Nous savons maintenant quel type de population à quel type de comportement et quel lieu ciblé pour vendre le produit. Il reste maintenant à promouvoir ce produit et pour cela il s'agit de savoir si la catégorie de population que l'on vise est sensible à la publicité/démarches.



Il ressort encore une fois que la catégorie "graduation" est très sensible à la pub dans le sens où elle répond quasiment à toutes les campagnes publicitaire à l'exception de la deuxième qui est bien plus basse que les autres. Cela est peut-être dû à un intervalle trop rapproché entre les campagnes ou bien à une mauvaise campagne publicitaire. Quand on regarde les autres catégories de population on y voit une très faible sensibilité à l'exception des master et phd qui on l'air de mieux réagir à la répétition, car ce sont souvent les campagnes 3,4,5 qui font que la population achète le produit.

## 6 Conclusion

Nous avons dans ce projet sélectionné un dataset permettant l'analyse du comportement d'achat de clients afin de créer divers profils et de mieux répondre à leurs attentes. Afin de réaliser cela une solution cloud AWS avec un cluster a été utilisé dans un premier temps, puis abandonné, car la politique des clouds fait qu'il est maintenant difficile de réaliser un tel projet gratuitement. J'ai malheureusement perdu trop de temps à vouloir insister sur le cloud afin de ne pas perdre le travail fournis. J'ai finalement opté pour une implémentation locale avec Hadoop/Hive/Sqoop permettant de récupérer les données depuis Sql et les transférer vers HDFS puis Hive afin de pouvoir appliquer diverses requête permettant de récupérer les données importantes. Ces données on ensuite été transformée en fichier CSV puis l'outil de visualisation Tableau a été utilisé. A l'issus de l'utilisation de tableau des analyses présentes dans ce compte rendu il ressort très clairement que la population graduation est la plus sûre pour la majeure partie des produits de

part le fait qu'elle dépense plus que les autres et est sensible à la publicité. Cependant il y a des exceptions comme les produits de luxe/plaisir qui eux conviennent plutôt aux populations de type master/phd.

Les résultats rendu ici sont cependant à vérifier notamment de par le fait que la population phd et master sont bien moins fournis que la population graduation ce qui est logique vu qu'il y a moins de phd que de master et moins de master que de graduation. Ce manque d'équité dans les données peut donc amener à des résultats biaisés. Une des améliorations à l'avenir serait donc de récolter plus de données sur ces dites population et aussi d'avoir des avis métier sur les variables et requête envisager pour l'analyse de comportement afin d'être sûr que notre interprétation des résultats est la bonne.