Tuto2 hdp

Version 1.0

2019-01-31

DATASOFT CONSULTING

Historique des modifications du document

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Auteur** |
| 2019-01-31 | 1.0 | Tuto2 hdp | Abdelaziz HANI |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table des matières

1. Introduction 4

2. HDFS 4

2.1 Importer un fichier depuis ambari: 4

2.2 Importer un fichier depuis terminal: 5

3. HBase 6

4. PIG 8

4.1 Utiliser pig shell: 8

4.2 Utiliser pig script: 9

5. KAFKA 9

6. ZEPPELIN 11

6.1 Zeppelin avec Spark: 11

6.2 Zeppelin avec hdfs: 12

7. SQOOP 13

# Introduction

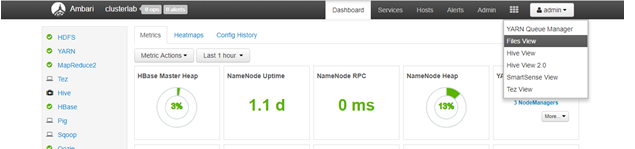
Dans ce tutorial nous allons voir les différents services big data avec des tests sur chaque service

Démarrons maintenant

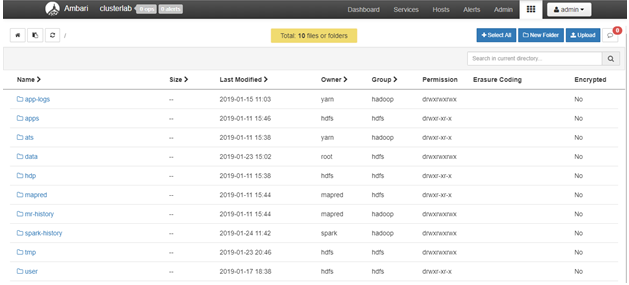
# HDFS

## Importer un fichier depuis ambari:

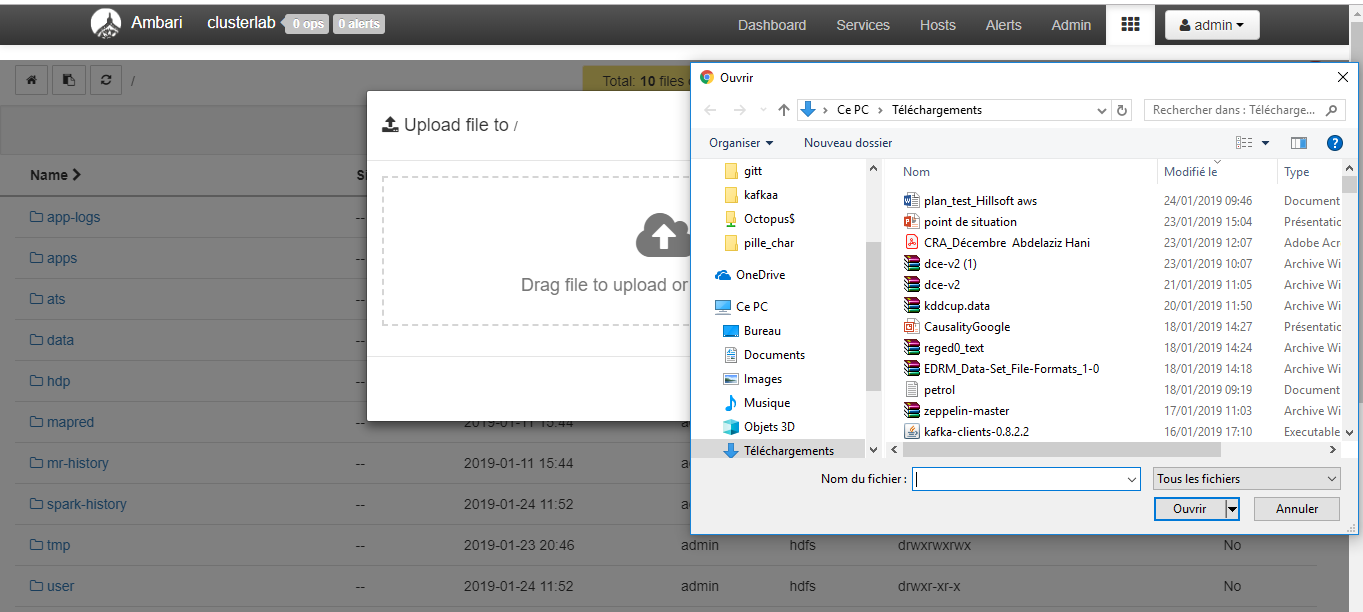
Dans le menu d’ambari cliquer sur Files View.



Vous aurez une page comme ceci



Ensuite cliquez sur upload (en haut a droite) afin de charger un ou plusieurs fichiers depuis le local vers hdfs.

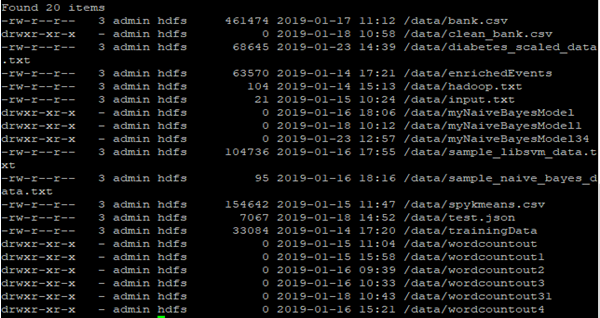


Et enfin Choisissez le fichier à charger dans HDFS.

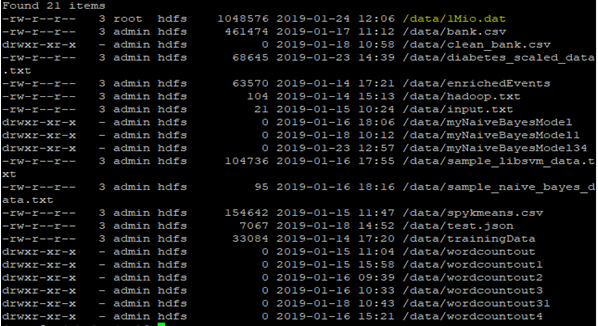
## Importer un fichier depuis terminal:

Choisissez n’importe quelle machine de cluster et tapez les commandes suivantes :

Tapez dans un premier temps la commande **hadoop fs -ls /data** pour vérifier que le fichier n’est pas encore importé

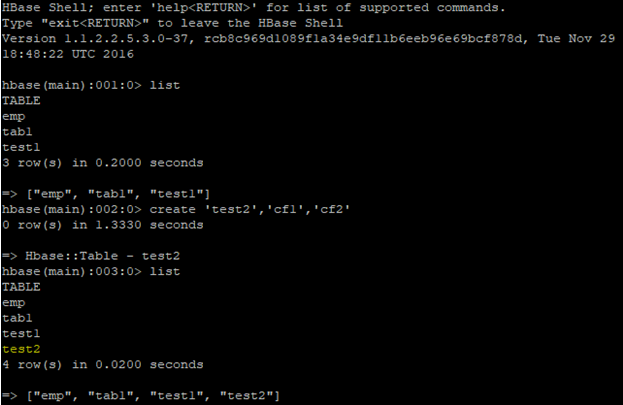


Ensuite on tape hadoop fs -copyFromLocal 1Mio.dat /data



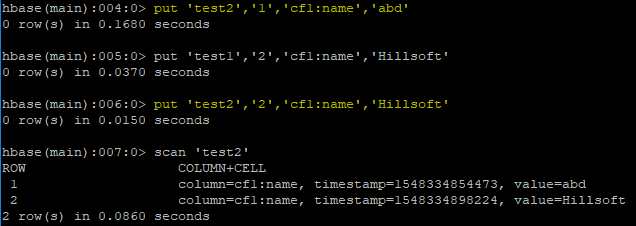
# HBase

Lancer la commande hbase shell



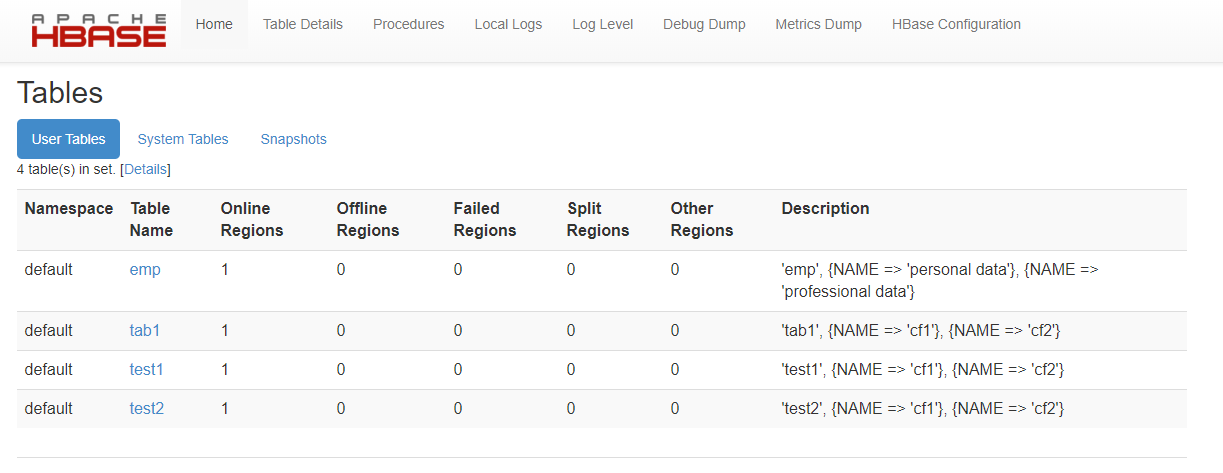
Donc on a crée la table ‘test2’ avec cf1 et cf2 sont les familles de colonne

Ensuite on a inséré deux lignes.



On verifie dans l’interface web de hbase master les tables créer

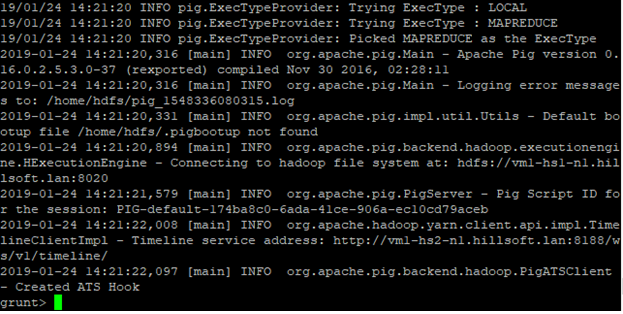
URL : <http://10.132.130.54:16010/master-status#userTables>



# PIG

## Utiliser pig shell:

Lancer la commande **pig**



On lit un fichier Dans un premier temps stocker sur HDFS

**grunt> bankText = load '/data/bank.csv' using PigStorage(';');**

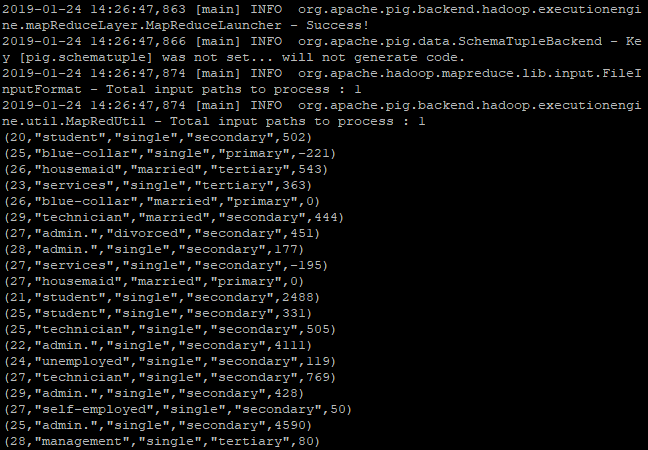
**grunt> bank = foreach bankText generate $0 as age, $1 as job, $2 as marital, $3 as education, $5 as balance;**

on fait un filtre suivant l’Age

**grunt> bank\_data = filter bank by age < 30;**

Enfin on affiche bank\_data

**grunt> dump bank\_data;**



## Utiliser pig script:

Écrire dans un fichier pig le script suivant

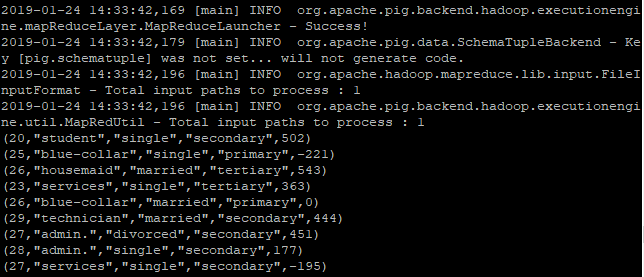
**bankText = load '/data/bank.csv' using PigStorage(';');**

**bank = foreach bankText generate $0 as age, $1 as job, $2 as marital, $3 as education, $5 as balance;**

**bank\_data = filter bank by age < 30;**

**dump bank\_data;**

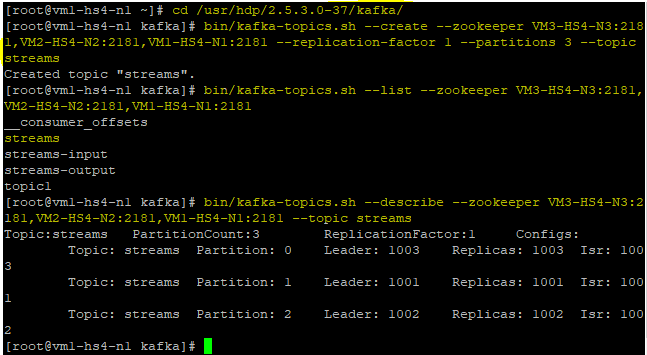
Exécuter le script en mode MapReduce avec la commande **pig bank.pig**



# KAFKA

On va considérer l’une de ces machines **vm1-hs4-n1, vm2-hs4-n2, vm3-hs4-n3** comme Producer et l’autre comme consumer

Mais avant nous devons créer un topic kafka qu’on le nomme **streams**



Donc on tape dans les machines Producer et consumer les commandes suivantes :

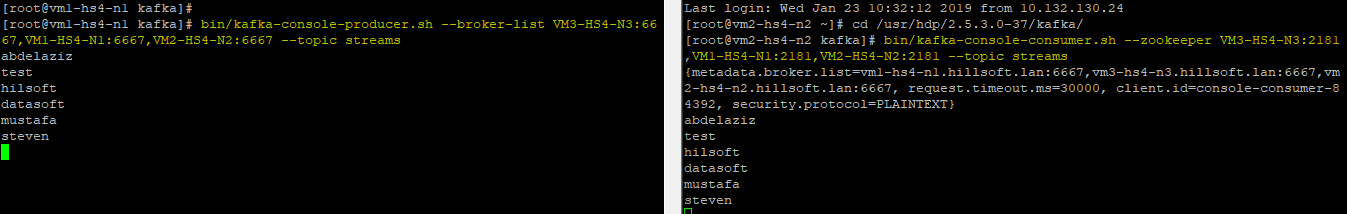
Dans la machine Producer :

**cd /usr/hdp/2.5.3.0-37/kafka**

**bin/kafka-console-producer.sh --broker-list VM3-HS4-N3:6667,VM1-HS4-N1:6667,VM2-HS4-N2:6667 --topic streams**

Dans la machine Consumer :

**bin/kafka-console-consumer.sh --zookeeper VM3-HS4-N3:2181,VM1-HS4-N1:2181,VM2-HS4-N2:2181 --topic streams**

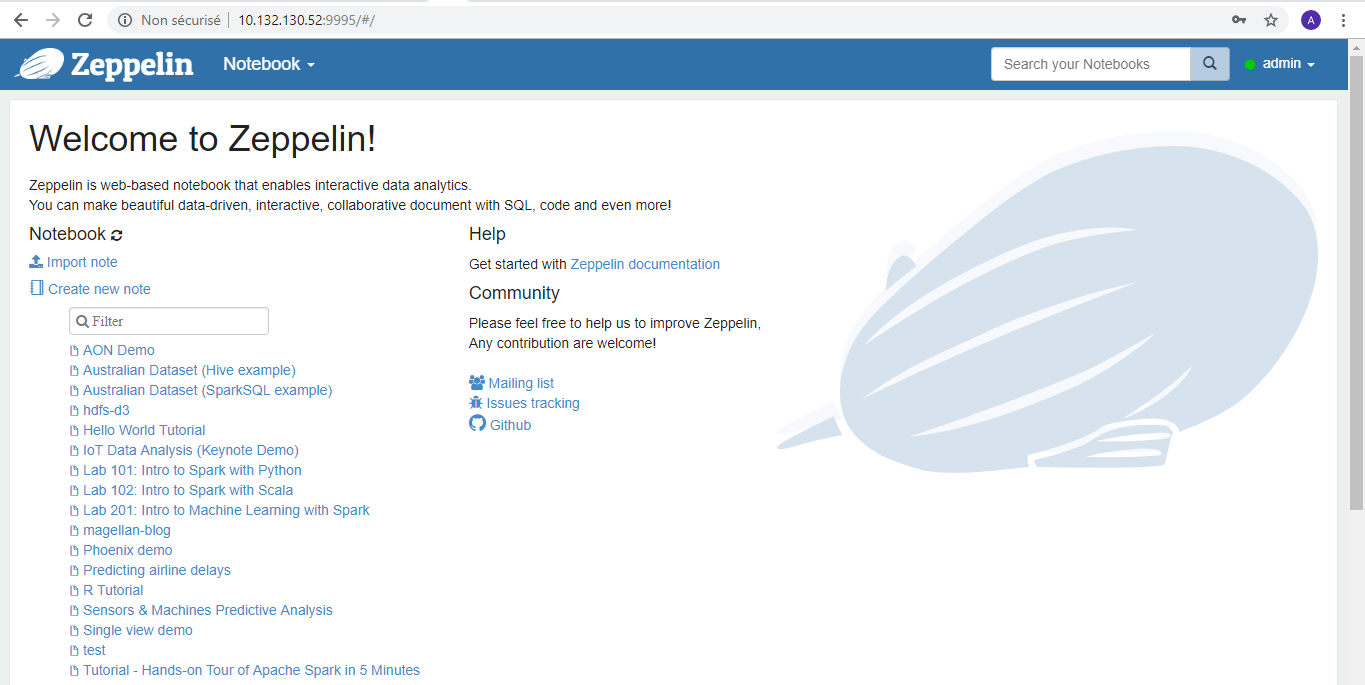


# ZEPPELIN

## Zeppelin avec Spark:

Tapez l’URL suivante pour acceder à zeppelin <http://10.132.130.52:9995>

Vous allez trouver une interface comme ceci où vous pouvez créer vos notebooks

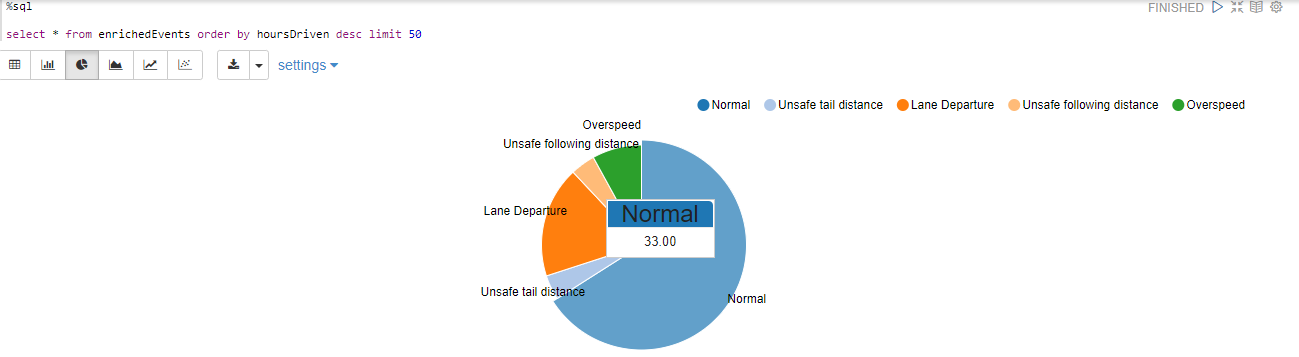


On commence par lire un fichier simple txt comme suit ,ensuite on applique un split suivant ‘,’

Et la fin on enregistre le résultat dans une table temporaire de spark

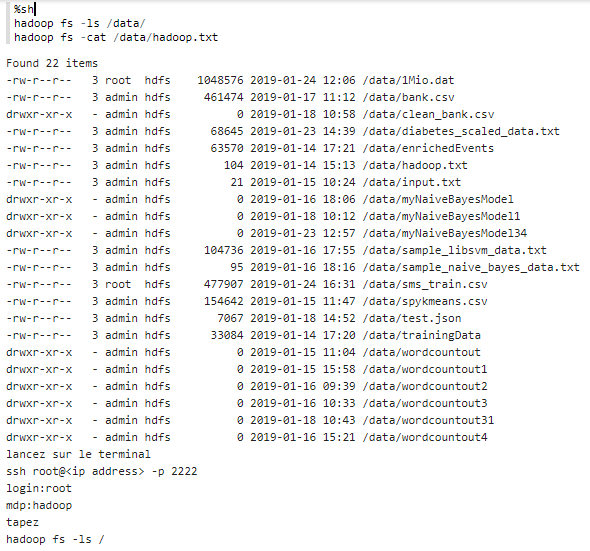


Maintenant on va utiliser spark SQL pour requêter la table **enrichedEvents**



## Zeppelin avec hdfs:

Dans zeppelin on peut utiliser des commandes linux ou hdfs, il suffit juste de tapez au début de %sh



# SQOOP

Lancer la commande mysql sur votre machine

La 1ere des choses c’est de créer une database

create database classe;

ensuite une table student2

create table student2(id integer,name char(20));

On insère des lignes avec la commande

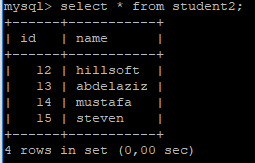
insert into student2 values(12,'hillsoft');

insert into student2 values(13,'abdelaziz');

insert into student2 values(14,'mustafa');

insert into student2 values(15,'steven');

on affiche pour voir la table

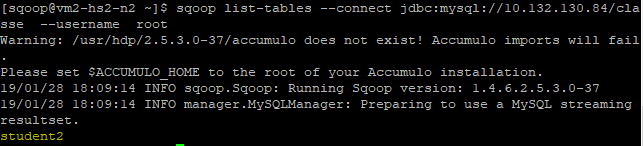


On passe maintenant à sqoop donc sur votre machine

$sqoop list-tables --connect jdbc:mysql://10.132.130.84/classe --username root

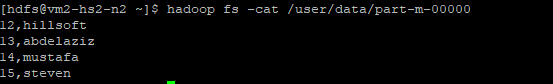
Où 10.132.130.84 est la machine serveur et classe est la base de données

Et list-tables : liste tous les tables dans classe

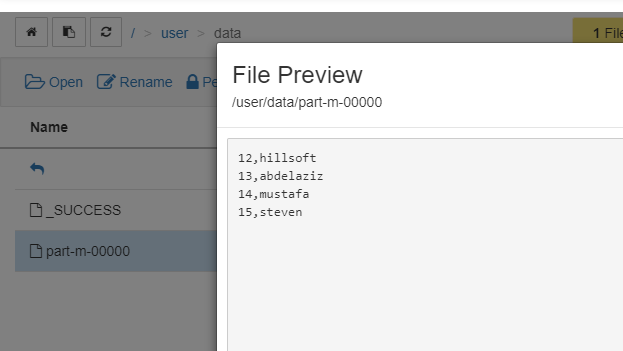


On lance maintenant la commande de l’import

sqoop import --connect jdbc:mysql://10.132.130.84/classe --username root --table student2 --m 1 --target-dir /user/data



Ou sur files view d’ambari



On passe maintenant à l’export depuis hdfs vers mysql

sqoop export --connect jdbc:mysql://10.132.130.84/classe --username root --table student3 --export-dir /user/data/part-m-00000

