DATA ENGINEER

INPT
المعهد الوطني للبريد والمواصلات
Institut National des Postes et Télécommunications

APACHE Spark

TP2 : Apache Spark

Rapport

Réalisé par :

Abderrahim ALAKOUCHE
Abdellah AGHLALOU

Encadré par :

Mme. Dounia ZAIDOUNI

PLAN

Introduction

Apache Spark est un framework de calcul distribué in-memory principalement (mais pas que) qui permet de faire de l'ETL (Extract Transform and Load), de l'ELT (Extract Load and Transform), de l'analytique avec une librairie riche et complète, du Machine Learning et aussi du traitement de graphes sur des gros volumes de données, avec différents formats en batch ou en pseudo-temps réel.[1]

Les objectifs de ce TP sont les suivants :

- Installation et configuration de Hadoop dans un noeud unique en local.
- Manipulation des RDDs en utilisant le terminal « **pyspark** ».
- Connexion de Spark à une distribution de Hadoop.
- Exécution du traitement « **Word Count** » en utilisant le terminal scala et python.
- Exécution du traitement « **Word Count** » en utilisant un script python.
- Exécution d'une application Spark Batch en Java.
- Initiation au « **Spark SQL and Dataframes** »

[1]:https://meritis.fr/larchitecture-framework-spark/

I Installation et configuration de Hadoop dans un nœud unique en local

- On répète les mêmes étapes d'installation et de configuration de Hadoop détaillées dans la section (I) du TP1

Etape1 : Création d'un utilisateur hduser

```
alakouche@alakouche-VB:~$ sudo adduser hduser
[sudo] password for alakouche:
Adding user `hduser' ...
Adding new group `hduser' (1001) ...
Adding new user `hduser' (1001) with group `hduser' ...
Creating home directory `/home/hduser' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for hduser
Enter the new value, or press ENTER for the default
        Full Name []: Alakouche/Aghlalou
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] Y
alakouche@alakouche-VB:~$ sudo adduser hduser sudo
Adding user `hduser' to group `sudo' ...
Adding user hduser to group sudo
Done.
alakouche@alakouche-VB:~$
```

Création d'un compte normal (non root) **hduser**

Cette commande va nous permettre d'éviter les erreurs du types « hduser is not in the sudoers file»

On redémarre la machine virtuelle afin de basculer vers le compte **hduser**

Etape2 : Mise en place de la clé ssh

Installation de paquet nécessaire pour **ssh.**

```
hduser@alakouche-VM:~$ sudo apt-get install openssh-server
[sudo] password for hduser:
Reading package lists... Done
Building dependency tree
Reading state information... Done
openssh-server is already the newest version (1:7.6p1-4ubuntu0.3).
0 upgraded, 0 newly installed, 0 to remove and 101 not upgraded.
```

```
hduser@alakouche-VM:~$ ssh-keygen -t rsa -P ""
Generating public/private rsa key pair.
Enter file in which to save the key (/home/hduser/.ssh/id_rsa):
/home/hduser/.ssh/id_rsa already exists.
Overwrite (y/n)? Y
Your identification has been saved in /home/hduser/.ssh/id_rsa.
Your public key has been saved in /home/hduser/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:PVZkWJ5WrQjxCb5uaPAo3gJ2WspDSxcO8pknawYOLjw hduser@alakouche-VM
The key's randomart image is:
+---[RSA 2048]----+
|          o++ .. |
|         .o* +  .|
|          ..B. . |
|. . .    . +. .  |
| o = .. S =      |
|o X *  + + .     |
|=* %. . + o      |
|.EX..o . .       |
|.o.....          |
+----[SHA256]-----+
hduser@alakouche-VM:~$ cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
hduser@alakouche-VM:~$ chmod 0600 ~/.ssh/authorized_keys
hduser@alakouche-VM:~$
```

Mettre en place la clé **ssh** pour son propre compte.

Hadoop nécessite un accès SSH pour gérer les différents nœuds. Bien que nous soyons dans une configuration simple nœud, nous avons besoin de configurer l'accès vers localhost pour l'utilisateur hduser que nous venons de créer précédemment.

Autoriser l'accès au SSH de la machine avec cette nouvelle clé fraîchement créée

On copie la clé public sur le serveur localhost

```
hduser@alakouche-VM:~$ ssh-copy-id -i /home/hduser/.ssh/id_rsa.pub  hduser@localhost
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/hduser/.ssh/id_rsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are alre

/usr/bin/ssh-copy-id: WARNING: All keys were skipped because they already exist on the remote system.
                (if you think this is a mistake, you may want to use -f option)

hduser@alakouche-VM:~$
```

```
hduser@alakouche-VM:~$ ssh localhost
Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 5.4.0-53-generic x86_64)

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


 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch

108 packages can be updated.
1 update is a security update.

New release '20.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Your Hardware Enablement Stack (HWE) is supported until April 2023.
Last login: Sat Nov 21 17:20:21 2020 from 127.0.0.1
hduser@alakouche-VM:~$ exit
logout
Connection to localhost closed.
hduser@alakouche-VM:~$
```

On teste la connexion à localhost

Etape 3 : Installation de JAVA 8

```
hduser@alakouche-VM:~/Documents$ sudo -i
[sudo] password for hduser:
root@alakouche-VM:~# mkdir /opt/java
```

Création de répertoire /opt/java

```
root@alakouche-VM:~# cd /home/hduser/Documents/
root@alakouche-VM:/home/hduser/Documents# tar -zxvf jdk-8u71-linux-x64.tar.gz
jdk1.8.0_71/
jdk1.8.0_71/db/
jdk1.8.0_71/db/lib/
jdk1.8.0_71/db/lib/derbyLocale_pl.jar
```

On va ensuite décompresser l'archive jdk8u71linuxx64.tar.gz

```
root@alakouche-VM:/home/hduser/Documents# mv jdk1.8.0_71/ /opt/java/
```

On déplace le jdk vers /opt/java/

On informe le système où java et ses exécutables sont installés.

```
root@alakouche-VM:/opt/java/jdk1.8.0_71# update-alternatives --install  /usr/bin/java java /opt/java/jdk1.8.0_71/bin/java 100
root@alakouche-VM:/opt/java/jdk1.8.0_71# update-alternatives --config java
There is only one alternative in link group java (providing /usr/bin/java): /opt/java/jdk1.8.0_71/bin/java
Nothing to configure.
root@alakouche-VM:/opt/java/jdk1.8.0_71#
root@alakouche-VM:/opt/java/jdk1.8.0_71# update-alternatives --install  /usr/bin/javac javac /opt/java/jdk1.8.0_71/bin/javac 100
root@alakouche-VM:/opt/java/jdk1.8.0_71# update-alternatives --config javac
There is only one alternative in link group javac (providing /usr/bin/javac): /opt/java/jdk1.8.0_71/bin/javac
Nothing to configure.
root@alakouche-VM:/opt/java/jdk1.8.0_71#
```

```
root@alakouche-VM:~# apt install vim
Reading package lists... Done
Building dependency tree
Reading state information... Done
```

Installation de l'éditeur de texte « **Vim** ».

```
root@alakouche-VM:~# vim /etc/profile
root@alakouche-VM:~#
```

Mettre en place de manière permanente les variables d'environnement JAVA pour tous les utilisateurs

```
  done
  unset i
fi
export JAVA_HOME=/opt/java/jdk1.8.0_71/
export JRE_HOME=/opt/java/jdk1.8.0._71/jre
export PATH=$PATH:/opt/java/jdk1.8.0_71/bin:/opt/java/jdk1.8.0_71/jre/bin
-- INSERT --
```

```
root@alakouche-VM:~# source /etc/profile
root@alakouche-VM:~# su - hduser
hduser@alakouche-VM:~$ source /etc/profile
hduser@alakouche-VM:~$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin:/opt/java/jdk1.8.0_71/bin:/op
t/java/jdk1.8.0_71/jre/bin:/opt/java/jdk1.8.0_71/bin:/opt/java/jdk1.8.0_71/jre/bin:/opt/apache-maven-3.5.0/bin:/usr/local/hadoop
/bin:/usr/local/hadoop/sbin:/usr/local/spark/bin:/opt/java/jdk1.8.0_71/bin:/opt/java/jdk1.8.0_71/jre/bin:/opt/java/jdk1.8.0_71/b
in:/opt/java/jdk1.8.0_71/jre/bin:/opt/apache-maven-3.5.0/bin
hduser@alakouche-VM:~$
```

On recharge le fichier **/etc/profile**

On teste la mise en place des variables d'environnement dans le terminal hadoop

Assurent que la version Java est correctement installée

```
hduser@alakouche-VM:~$ java -version
java version "1.8.0_71"
Java(TM) SE Runtime Environment (build 1.8.0_71-b15)
Java HotSpot(TM) 64-Bit Server VM (build 25.71-b15, mixed mode)
hduser@alakouche-VM:~$
```

Etape 4 : Installation d'Apache Hadoop 3.2.1

```
hduser@alakouche-VM:~/Documents$ tar -zxvf hadoop-3.2.1.tar.gz
```

On décompresse l'archive : hadoop3.1.2.tar.gz

```
hduser@alakouche-VM:~$ cd Documents/
hduser@alakouche-VM:~/Documents$ ls
 apache-maven-3.5.0              hadoop-3.2.1              jdk1.8.0_71
 apache-maven-3.5.0-bin.tar.gz  'hadoop-3.2.1(1).tar.gz'   jdk-8u71-linux-x64.tar.gz
hduser@alakouche-VM:~/Documents$ mv hadoop-3.2.1 hadoop
hduser@alakouche-VM:~/Documents$ sudo mv hadoop /usr/local/hadoop/
[sudo] password for hduser:
```

```
hduser@alakouche-VM:~/Documents$ sudo chown -R hduser /usr/locla/hadoop
```

On affecte les droits à notre utilisateur **hduse.**

```
hduser@alakouche-VM:~$ sudo mkdir -p /usr/local/hadoop_store/hdfs/namenode
hduser@alakouche-VM:~$ sudo mkdir -p /usr/local/hadoop_store/hdfs/datanode
hduser@alakouche-VM:~$ sudo chown -R hduser /usr/local/hadoop_store
hduser@alakouche-VM:~$
```

Création de datanode et namenode.
On affecte ensuite les droits à notre utilisateur **hduser**.

Etape 5 : Configuration d'Apache Hadoop 3.2.1

- Il faut maintenant définir la configuration de Hadoop et pour cela plusieurs fichiers de configurations doivent être modifiés

```
hduser@alakouche-VM:~$ vim .bashrc
hduser@alakouche-VM:~$ source .bashrc
hduser@alakouche-VM:~$
```

```
fi
#HADOOP VARIABLES START
export JAVA_HOME=/opt/java/jdk1.8.0_71/
export HADOOP_INSTALL=/usr/local/hadoop
export PATH=$PATH:$HADOOP_INSTALL/bin
export PATH=$PATH:$HADOOP_INSTALL/sbin
export HADOOP_MAPRED_HOME=$HADOOP_INSTALL
export HADOOP_COMMON_HOME=$HADOOP_INSTALL
export HADOOP_HDFS_HOME=$HADOOP_INSTALL
export YARN_HOME=$HADOOP_INSTALL
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_INSTALL/lib/native
#export HADOOP_OPTS=" Djava.library.path=$HADOOP_INSTALL/lib"
#HADOOP VARIABLES END
:x
```

```
hduser@alakouche-VM:~$ cd /usr/local/hadoop/etc/hadoop
hduser@alakouche-VM:/usr/local/hadoop/etc/hadoop$ vim hadoop-env.sh
hduser@alakouche-VM:/usr/local/hadoop/etc/hadoop$
```

```
#  JAVA_HOME=/usr/java/testing hdfs dfs -ls
#
export JAVA_HOME=/opt/java/jdk1.8.0_71/ # Therefore, the vast ma
# are configured for substitution and not append.  If append
# is preferable, modify this file accordingly.

###
```

```
hduser@alakouche-VM:~$ sudo mkdir -p /app/hadoop/tmp
hduser@alakouche-VM:~$ sudo chown hduser /app/hadoop/tmp
hduser@alakouche-VM:~$
```

Création de répertoire des fichiers temporaires de Hadoop

- Modification des autres fichiers de configurations : on ajoute les lignes suivantes entre les balises de configurations.

```
<configuration>
 <property>
  <name>hadoop.tmp.dir</name>
  <value>/app/hadoop/tmp</value>
 </property>
 <property>
  <name>fs.default.name</name>
  <value>hdfs://localhost:54310</value>
 </property>
</configuration>
"core-site.xml" 28L, 958C
```

```
<configuration>
 <property>
  <name>dfs.replication</name>
  <value>1</value>
 </property>
 <property>
  <name>dfs.namenode.name.dir</name>
  <value>file:/usr/local/hadoop_store/hdfs/namenode</value>
 </property>
 <property>
  <name>dfs.datanode.data.dir</name>
  <value>file:/usr/local/hadoop_store/hdfs/datanode</value>
 </property>
</configuration>
```

```
hduser@alakouche-VM:~$ cd /usr/local/hadoop/etc/hadoop
hduser@alakouche-VM:/usr/local/hadoop/etc/hadoop$ vim core-site.xml
hduser@alakouche-VM:/usr/local/hadoop/etc/hadoop$ vim hdfs-site.xml
hduser@alakouche-VM:/usr/local/hadoop/etc/hadoop$ vim mapred-site.xml
hduser@alakouche-VM:/usr/local/hadoop/etc/hadoop$ vim yarn-site.xml
hduser@alakouche-VM:/usr/local/hadoop/etc/hadoop$
```

```
<!-- Put site-specific property overrides

<configuration>
 <property>
  <name>mapred.job.tracker</name>
  <value>localhost:54311</value>
 </property>
</configuration>
```

```
<configuration>
<!-- Site specific YARN configuration propertie
<property>
<name>yarn.nodemanager.aux-services</name>
<value>mapreduce_shuffle</value>
</property>
</configuration>
```

- Avant de démarrer le serveur Hadoop, il faut formater le système de fichiers HDFS.

```
hduser@alakouche-VM:/usr/local/hadoop_store$ cd /usr/local/hadoop/etc/hadoop
hduser@alakouche-VM:/usr/local/hadoop/etc/hadoop$ hdfs namenode -format
2020-11-22 15:21:00,007 INFO namenode.NameNode: STARTUP_MSG:
/************************************************************
STARTUP_MSG: Starting NameNode
STARTUP_MSG:   host = alakouche-VM/127.0.1.1
STARTUP_MSG:   args = [-format]
STARTUP_MSG:   version = 3.2.1
```

```
2020-11-22 15:21:12,461 INFO namenode.FSNamesystem: Retry cache will use 0.03 of total heap and retry cache entry expiry time is 600000 millis
2020-11-22 15:21:12,515 INFO util.GSet: Computing capacity for map NameNodeRetryCache
2020-11-22 15:21:12,529 INFO util.GSet: VM type       = 64-bit
2020-11-22 15:21:12,537 INFO util.GSet: 0.029999999329447746% max memory 953.2 MB = 292.8 KB
2020-11-22 15:21:12,538 INFO util.GSet: capacity      = 2^15 = 32768 entries
2020-11-22 15:21:12,990 INFO namenode.FSImage: Allocated new BlockPoolId: BP-513619828-127.0.1.1-1606038472869
2020-11-22 15:21:13,222 INFO common.Storage: Storage directory /usr/local/hadoop_store/hdfs/namenode has been successfully formatted.
2020-11-22 15:21:13,710 INFO namenode.FSImageFormatProtobuf: Saving image file /usr/local/hadoop_store/hdfs/namenode/current/fsimage.ckpt_0000
000000000000000 using no compression
2020-11-22 15:21:14,753 INFO namenode.FSImageFormatProtobuf: Image file /usr/local/hadoop_store/hdfs/namenode/current/fsimage.ckpt_00000000000
00000000 of size 401 bytes saved in 1 seconds .
2020-11-22 15:21:14,904 INFO namenode.NNStorageRetentionManager: Going to retain 1 images with txid >= 0
2020-11-22 15:21:14,966 INFO namenode.FSImage: FSImageSaver clean checkpoint: txid=0 when meet shutdown.
2020-11-22 15:21:14,967 INFO namenode.NameNode: SHUTDOWN_MSG:
/************************************************************
SHUTDOWN_MSG: Shutting down NameNode at alakouche-VM/127.0.1.1
************************************************************/
hduser@alakouche-VM:/usr/local/hadoop/etc/hadoop$
```

- Démarrage de Hadoop

```
hduser@alakouche-VM:/usr/local/hadoop/etc/hadoop$ start-dfs.sh
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [alakouche-VM]
2020-11-22 15:47:20,984 WARN util.NativeCodeLoader: Unable to load native-hadoo
 applicable
hduser@alakouche-VM:/usr/local/hadoop/etc/hadoop$ start-yarn.sh
Starting resourcemanager
Starting nodemanagers
hduser@alakouche-VM:/usr/local/hadoop/etc/hadoop$ jps
23378 Jps
22370 SecondaryNameNode
23108 NodeManager
21557 DataNode
22886 ResourceManager
21371 NameNode
hduser@alakouche-VM:/usr/local/hadoop/etc/hadoop$
```

```
hduser@alakouche-VM:/usr/local/hadoop/etc/hadoop$ hdfs dfsadmin -report
2020-11-21 17:03:22,979 WARN util.NativeCodeLoader: Unable to load native-hadoo
 applicable
-------------------------------------------------
Live datanodes (1):

Name: 127.0.0.1:9866 (localhost)
Hostname: alakouche-VM
Decommission Status : Normal
Configured Capacity: 21001486336 (19.56 GB)
DFS Used: 24576 (24 KB)
Non DFS Used: 7693438976 (7.17 GB)
DFS Remaining: 12217610240 (11.38 GB)
DFS Used%: 0.00%
DFS Remaining%: 58.17%
```

On vérifie le bon fonctionnement de notre nœud

II Installation et configuration de Spark en local

- On va utiliser Spark en local sur la VM Ubuntu créée précédemment. la version quand va utilisé est : **spark-2.4.3**

```
hduser@alakouche-VM:~$ cd /home/hduser/Documents/
hduser@alakouche-VM:~/Documents$ tar xvzf spark-2.4.3-bin-hadoop2.7.tgz spark-2.4.3-bin-hadoop2.7/
spark-2.4.3-bin-hadoop2.7/
spark-2.4.3-bin-hadoop2.7/python/
spark-2.4.3-bin-hadoop2.7/python/setup.cfg
```

On décompresse l'archive : **spark-2.4.3-bin-hadoop2.7.tgz**

```
hduser@alakouche-VM:~/Documents$ ls
'hadoop-3.2.1(1).tar.gz'   jdk-8u71-linux-x64.tar.gz   poeme.text   spark-2.4.3-bin-hadoop2.7   spark-2.4.3-bin-hadoop2.7.tgz
hduser@alakouche-VM:~/Documents$ mv spark-2.4.3-bin-hadoop2.7 spark
hduser@alakouche-VM:~/Documents$ sudo mv spark /usr/local/
[sudo] password for hduser:
hduser@alakouche-VM:~/Documents$
```

On déplace **spark** vers **/usr/local/**

On renomme le répertoire pour plus de simplicité

- Il faut maintenant configurer le PATH dans le fichier **.bashrc**

```
hduser@alakouche-VM:~$ vim .bashrc
hduser@alakouche-VM:~$ source .bashrc
hduser@alakouche-VM:~$
```

```
#HADOOP VARIABLES START
export JAVA_HOME=/opt/java/jdk1.8.0_71/
export HADOOP_INSTALL=/usr/local/hadoop
export PATH=$PATH:$HADOOP_INSTALL/bin
export PATH=$PATH:$HADOOP_INSTALL/sbin
export HADOOP_MAPRED_HOME=$HADOOP_INSTALL
export HADOOP_COMMON_HOME=$HADOOP_INSTALL
export HADOOP_HDFS_HOME=$HADOOP_INSTALL
export YARN_HOME=$HADOOP_INSTALL
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_INSTALL/lib/native
export HADOOP_OPTS="-Djava.library.path=$HADOOP_INSTALL/lib"
#HADOOP VARIABLES END
#Spark
export SPARK_HOME=/usr/local/spark
export PATH=$PATH:$SPARK_HOME/bin
```

- Installation de Python :

```
hduser@alakouche-VM:~$ sudo apt-get install python
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libpython-stdlib libpython2.7 libpython2.7-minimal libpython2.7-stdlib python-minimal python2.7 python2.7-minimal
Suggested packages:
  python-doc python-tk python2.7-doc binfmt-support
The following NEW packages will be installed:
  libpython-stdlib python python-minimal python2.7 python2.7-minimal
```

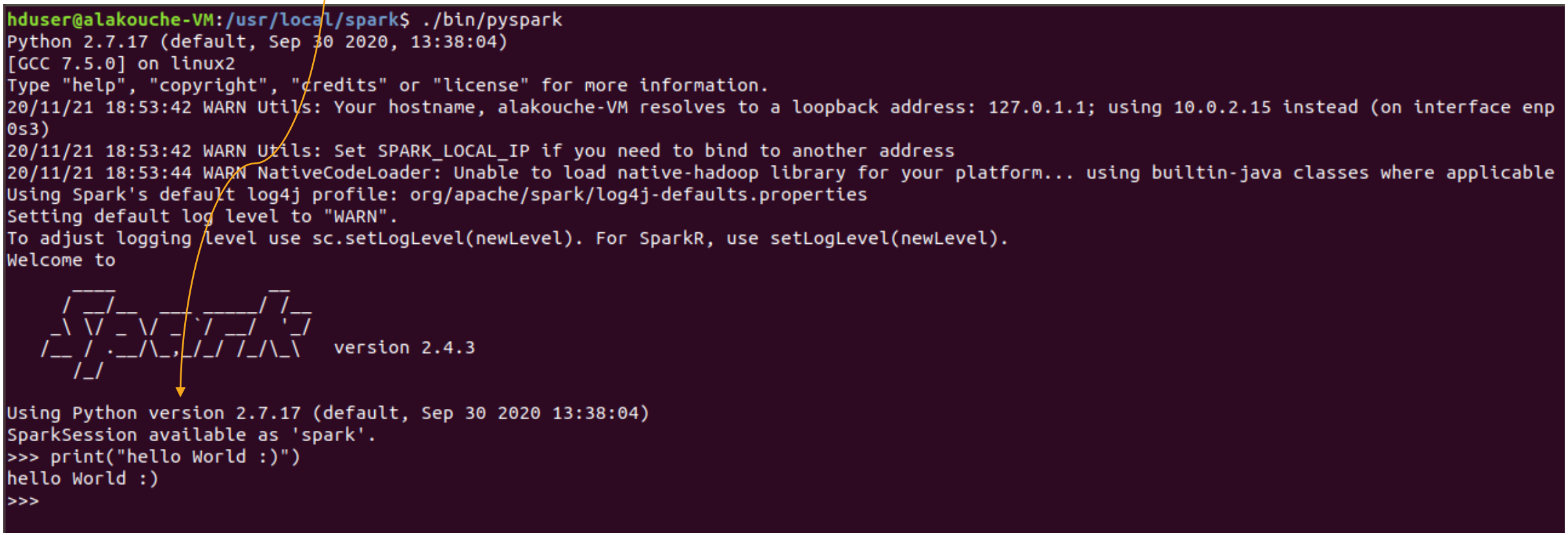
- On accède au **terminal Scala**

```
hduser@alakouche-VM:~$ cd /usr/local/spark/
hduser@alakouche-VM:/usr/local/spark$ ./bin/spark-shell
20/11/21 18:50:19 WARN Utils: Your hostname, alakouche-VM resolves to a loopback address: 127.0.1.1; using 10.0.2.15 instead (on interface enp
0s3)
20/11/21 18:50:19 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
20/11/21 18:50:21 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
Spark context Web UI available at http://10.0.2.15:4040
Spark context available as 'sc' (master = local[*], app id = local-1605984656591).
Spark session available as 'spark'.
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 2.4.3
      /_/

Using Scala version 2.11.12 (Java HotSpot(TM) 64-Bit Server VM, Java 1.8.0_71)
Type in expressions to have them evaluated.
Type :help for more information.

scala>
```

- On peut aussi accéder au **terminal Python**

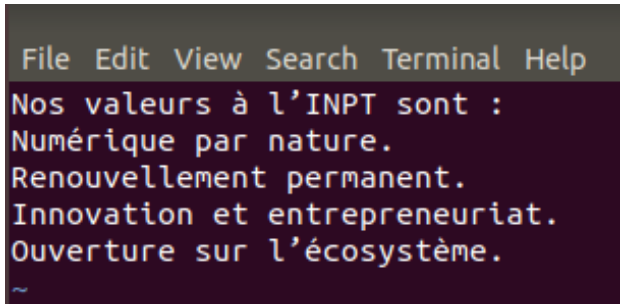


Dans cette partie, nous allons appliquer les différents exemples vus dans le cours.

1-Enregistrement et chargement des Textfile :

- La création de fichier **ValeursINPT.txt**

```
hduser@alakouche-VM:/usr/local/spark$ touch ValeursINPT.txt
hduser@alakouche-VM:/usr/local/spark$ vim ValeursINPT.txt
hduser@alakouche-VM:/usr/local/spark$
```



```
hduser@alakouche-VM:/usr/local/spark$ ./bin/pyspark
```

```
Using Python version 2.7.17 (default, Sep 30 2020 13:38:04)
SparkSession available as 'spark'.
>>> mydata = sc.textFile("file:/usr/local/spark/ValeursINPT.txt")
>>> for line in mydata.collect():
...     print line
...
Nos valeurs à l'INPT sont :
Numérique par nature.
Renouvellement permanent.
Innovation et entrepreneuriat.
Ouverture sur l'écosystème.
>>> mydata_filt = mydata.filter(lambda s: s.startswith('N'))
>>> mydata_filt.saveAsTextFile("file:/usr/local/spark/values_starts_withN")
>>> mydata.count()
5
>>> mydata.count()
5
>>> mydata_filt.count()
2
>>>
```

On accède au terminal **pyspark** pour taper les commandes suivantes

Vérification : le répertoire **values_starts_withN** était bien crée, ce répertoire contient deux fichiers : **part-00000** et **_SUCCESS.**

```
hduser@alakouche-VM:/usr/local/spark$ ls
bin   data      fileseq1  kubernetes  licenses  python  README.md  sbin             values_starts_withN
conf  examples  jars      LICENSE     NOTICE    R       RELEASE    ValeursINPT.txt  yarn
hduser@alakouche-VM:/usr/local/spark$ cd values_starts_withN/
hduser@alakouche-VM:/usr/local/spark/values_starts_withN$ ls
part-00000  _SUCCESS
hduser@alakouche-VM:/usr/local/spark/values_starts_withN$
```

2-Enregistrement et chargement d es SequenceFiles :

```
hduser@alakouche-VM:/usr/local/spark$ ./bin/pyspark
Python 2.7.17 (default, Sep 30 2020, 13:38:04)
[GCC 7.5.0] on linux2
Type "help", "copyright", "credits" or "license" for more information.
20/11/21 19:38:26 WARN Utils: Your hostname, alakouche-VM resolves to a loopback address: 127.0.1.1; usi
0s3)
20/11/21 19:38:26 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
20/11/21 19:38:28 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 2.4.3
      /_/

Using Python version 2.7.17 (default, Sep 30 2020 13:38:04)
SparkSession available as 'spark'.
>>> rdd = sc.parallelize(range(1, 4)).map(lambda x: (x, "a" * x))
>>> rdd.saveAsSequenceFile("file:/usr/local/spark/fileseq1")
>>> sorted(sc.sequenceFile("file:/usr/local/spark/fileseq1").collect())
[(1, u'a'), (2, u'aa'), (3, u'aaa')]
>>> exit()
hduser@alakouche-VM:/usr/local/spark$ ls
bin   data      fileseq1  kubernetes  licenses  python  README.md  sbin            values_starts_withN
conf  examples  jars      LICENSE     NOTICE    R       RELEASE    ValeursINPT.txt  yarn
hduser@alakouche-VM:/usr/local/spark$ cd fileseq1/
hduser@alakouche-VM:/usr/local/spark/fileseq1$ ls
part-00000  _SUCCESS
hduser@alakouche-VM:/usr/local/spark/fileseq1$
```

Vérification : le répertoire **fileseq1** était bien crée, ce répertoire contient deux fichiers : **part-00000** et **_SUCCESS.**

3-Utilisation d’une fonction nommée :

```
Python 2.7.17 (default, Sep 30 2020, 13:38:04)
[GCC 7.5.0] on linux2
Type "help", "copyright", "credits" or "license" for more information.
20/11/21 19:43:39 WARN Utils: Your hostname, alakouche-VM resolves to a loopback address: 127.0.1.1; using 10.0.2.15 instead (on interface enp
0s3)
20/11/21 19:43:39 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
20/11/21 19:43:40 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 2.4.3
      /_/

Using Python version 2.7.17 (default, Sep 30 2020 13:38:04)
SparkSession available as 'spark'.
>>>
>>> def toUpper(s):
...     return s.upper()
...
>>> mydata = sc.textFile("file:/usr/local/spark/ValeursINPT.txt")
>>> mydataupper = mydata.map(toUpper)
>>> for line in mydataupper.collect():
...     print line
...
NOS VALEURS À L’INPT SONT :
NUMÉRIQUE PAR NATURE.
RENOUVELLEMENT PERMANENT.
INNOVATION ET ENTREPRENEURIAT.
OUVERTURE SUR L’ÉCOSYSTÈME.
>>>
```

4- Utilisation d'une fonction anonyme :

```
>>> mydata = sc.textFile("file:/usr/local/spark/poeme.txt")
>>> mydata_upper = mydata.map(lambda line: line.upper()).take(5)
>>> for line in mydata_upper:
...     print line
...
CELUI QUI CROYAIT AU CIEL
CELUI QUI NY CROYAIT PAS
TOUS DEUX ADORAIENT LA BELLE
PRISONNIERE DES SOLDATS
LEQUEL MONTAIT A LECHELLE
>>>
```

5- Utilisation de « parallelize » :

```
>>> data = [10, 20, 30, 40, 50, 100, 250]
>>> distData = sc.parallelize(data)
>>> total = distData.reduce(lambda a,b: a + b)
>>> print total
500
>>>
```

6- Utilisation de « wholeTextFiles » :

- La création et le remplissage des deux fichier json **file1** et **file2** dans le répertoire **/usr/local/spark**

```
hduser@alakouche-VM:~$ cd /usr/local/spark
hduser@alakouche-VM:/usr/local/spark$ ls
bin    examples  kubernetes  NOTICE     R           sbin                 yarn
conf   fileseq1  LICENSE     poeme.txt  README.md   ValeursINPT.txt
data   jars      licenses    python     RELEASE     values_starts_withN
hduser@alakouche-VM:/usr/local/spark$ clear
hduser@alakouche-VM:/usr/local/spark$ mkdir json_files
hduser@alakouche-VM:/usr/local/spark$ cd json_files/
hduser@alakouche-VM:/usr/local/spark/json_files$ touch file1.json file2.json
hduser@alakouche-VM:/usr/local/spark/json_files$ vim file1.json
hduser@alakouche-VM:/usr/local/spark/json_files$ vim file2.json
hduser@alakouche-VM:/usr/local/spark/json_files$
```

File Edit View Search Terminal He

```
{
"firstName": "Fred",
"lastName": "Flintstone",
"userid": "123"
}
```

file1.json

File Edit View Search Terminal Help

```
{
"firstName": "Barney",
"lastName": "Rubble",
"userid": "234"
}
```

file2.json

```
>>>
>>> import json
>>> myrdd1 = sc.wholeTextFiles("file:/usr/local/spark/json_files")
>>> myrdd2 = myrdd1.map(lambda (fname,s): json.loads(s))
>>> for record in myrdd2.take(2):
...     print record.get("firstName",None)
...
Fred
Barney
>>>
```

7- Utilisation de « flatMap » et « distinct »:

```
>>>
>>> mydata = sc.textFile("file:/usr/local/spark/poeme.txt")
>>> mynewdata = mydata.flatMap(lambda line: line.split(' ')).distinct()
>>> for line in mynewdata.collect():
...     print line
...
```

```
croyait

prisonniere
trompa
coeur
couleur
nouvelle
soldats
rouge
tombe
derobât
citadelle
```

```
au
il
mirabelle
amour
par
pas
quand
belle
seul
ne
ny
deux
lautre
```

```
fois
est
vient
court
fideles
a
sy
songe
querelles
lequel
delicat
murisse
reseda
```

```
deux
lautre
lechelle
combat
bretagne
flute
fois
est
vient
court
fideles
a
sy
```

....

8- Utilisation de « subtract » et « zip »:

```
>>>
>>> mydata = ["Chicago", "Boston", "Paris", "San Francisco", "Tokyo"]
>>> rdd1 = sc.parallelize(mydata)
>>> data = ["San Francisco", "Boston", "Amsterdam", "Mumbai", "McMurdo Station"]
>>> rdd2 = sc.parallelize(data)
>>> newrdd = rdd1.subtract(rdd2)
>>> for line in newrdd.collect():
...     print line
...
Paris
Tokyo
Chicago
>>> ziprdd = rdd1.zip(rdd2)
>>> for line in ziprdd.collect():
...     print line
...
('Chicago', 'San Francisco')
('Boston', 'Boston')
('Paris', 'Amsterdam')
('San Francisco', 'Mumbai')
('Tokyo', 'McMurdo Station')
>>>
```

9- Utilisation de intersection et union :

```
>>>
>>> interdd = rdd1.intersection(rdd2)
>>> for line in interdd.collect():
...     print line
...
Boston
San Francisco
>>> unionrdd = rdd1.union(rdd2)
>>> for line in unionrdd.collect():
...     print line
...
Chicago
Boston
Paris
San Francisco
Tokyo
San Francisco
Boston
Amsterdam
Mumbai
McMurdo Station
>>>
```

IV Connexion de Spark à une distribution de Hadoop

Spark peut utiliser les bibliothèques clientes Hadoop pour HDFS et YARN.
Pour cela on va modifier **SPARK_DIST_CLASSPATH** afin d'inclure les fichiers jar des packages de projet Hadoop free.

```
hduser@alakouche-VM:/usr/local/spark$ cd conf/
hduser@alakouche-VM:/usr/local/spark/conf$ cp spark-env.sh.template spark-env.sh
hduser@alakouche-VM:/usr/local/spark/conf$ vim spark-env.sh
hduser@alakouche-VM:/usr/local/spark/conf$
```

Modification de fichier **spark-env.sh**

```
File  Edit  View  Search  Terminal  Help
#!/usr/bin/env bash
### in conf/spark-env.sh ###
# If 'hadoop' binary is on your PATH
export SPARK_DIST_CLASSPATH=/usr/local/hadoop
# With explicit path to 'hadoop' binary
export SPARK_DIST_CLASSPATH=/usr/local/hadoop/bin
# Passing a Hadoop configuration directory
export SPARK_DIST_CLASSPATH=/usr/local/hadoop/etc/hadoop
#
# Licensed to the Apache Software Foundation (ASF) under
# contributor license agreements.  See the NOTICE file di
# this work for additional information regarding copyrigh
# The ASF licenses this file to You under the Apache Lice
# (the "License"); you may not use this file except in co
# 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 writi
```

V Exécution du « Word Count » en utilisant le terminal scala et python

Dans cette section, nous allons déposer le **poeme.txt** dans le HDFS

```
hduser@alakouche-VM:/usr/local/hadoop$ bin/hdfs dfs -put /usr/local/spark/poeme.txt /
2020-11-21 20:58:51,015 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where
 applicable
2020-11-21 20:58:55,089 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
```

```
hduser@alakouche-VM:/usr/local/hadoop$ bin/hdfs dfs -ls /
2020-11-21 20:59:46,685 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where
 applicable
Found 1 items
-rw-r--r--   1 hduser supergroup       1670 2020-11-21 20:58 /poeme.txt
hduser@alakouche-VM:/usr/local/hadoop$
```

Show 25 entries Search:

	Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
☐	-rw-r--r--	hduser	supergroup	1.63 KB	Nov 21 21:58	1	128 MB	poeme.txt

Showing 1 to 1 of 1 entries Previous 1 Next

Hadoop, 2019.

On exécute le traitement du « **Word Count** » en utilisant le terminal **spark-shell**:

```
hduser@alakouche-VM:/usr/local/hadoop$ cd /usr/local/spark
hduser@alakouche-VM:/usr/local/spark$ ./bin/spark-shell
2020-11-21 21:01:42,145 WARN util.Utils: Your hostname, alakouche-VM resolves to a loopback address: 127.0.1.1; using 10.0.2
terface enp0s3)
2020-11-21 21:01:42,153 WARN util.Utils: Set SPARK_LOCAL_IP if you need to bind to another address
2020-11-21 21:01:44,043 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-
 applicable
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
Spark context Web UI available at http://10.0.2.15:4040
Spark context available as 'sc' (master = local[*], app id = local-1605992537979).
Spark session available as 'spark'.
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 2.4.3
      /_/

Using Scala version 2.11.12 (Java HotSpot(TM) 64-Bit Server VM, Java 1.8.0_71)
Type in expressions to have them evaluated.
Type :help for more information.

scala> val lines = sc.textFile("/poeme.txt")
lines: org.apache.spark.rdd.RDD[String] = /poeme.txt MapPartitionsRDD[1] at textFile at <console>:24

scala> val words = lines.flatMap(_.split("\\s+"))
words: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[2] at flatMap at <console>:25

scala> val wc = words.map(w => (w, 1)).reduceByKey(_ +_)
wc: org.apache.spark.rdd.RDD[(String, Int)] = ShuffledRDD[4] at reduceByKey at <console>:25

scala> wc.saveAsTextFile("file1.count")
```

```
hduser@alakouche-VM:/usr/local/spark$ cd /usr/local/hadoop
hduser@alakouche-VM:/usr/local/hadoop$ hadoop fs -get file1.count
2020-11-21 21:05:03,205 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where
 applicable
2020-11-21 21:05:08,939 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
hduser@alakouche-VM:/usr/local/hadoop$ bin/hdfs dfs -ls .
2020-11-21 21:05:54,483 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where
 applicable
Found 1 items
drwxr-xr-x   - hduser supergroup          0 2020-11-21 21:03 file1.count
hduser@alakouche-VM:/usr/local/hadoop$ bin/hdfs dfs -cat file1.count/part-00000
2020-11-21 21:07:10,833 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where
 applicable
2020-11-21 21:07:14,389 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
```

```
(jura,1)
(ils,1)
(violoncelle,1)
(comment,1)
(rose,1)
(soldats,1)
(que,2)
(celui,20)
(levres,1)
(bretagne,1)
(ses,1)
(quelle,1)
(de,5)
(les,4)
(cruelle,1)
(tombe,1)
(ruisselle,1)
(combat,1)
(prefere,1)
```

```
(qui,25)
(lechelle,1)
(est,1)
(lhirondelle,1)
(fait,1)
(tira,1)
(court,1)
(pour,1)
(croyait,20)
(bas,1)
(plus,2)
(passent,1)
(commun,1)
(font,1)
(qua,1)
(le,6)
(muscat,1)
(tous,3)
(gele,1)
```

```
(le,6)
(muscat,1)
(tous,3)
(gele,1)
(lalouette,1)
(fois,1)
(citadelle,1)
(aima,1)
(eclat,1)
(sang,1)
(terre,1)
(nouvelle,1)
(framboise,1)
(etaient,1)
(guettait,1)
(brula,1)
(couleur,1)
(rechantera,1)
(vient,1)
```

- Exécutent le traitement du « **Word Count** » en utilisant le terminal **Pyspark**

```
hduser@alakouche-VM:/usr/local/hadoop$ cd /usr/local/spark
hduser@alakouche-VM:/usr/local/spark$ ./bin/pyspark
Python 2.7.17 (default, Sep 30 2020, 13:38:04)
[GCC 7.5.0] on linux2
Type "help", "copyright", "credits" or "license" for more information.
2020-11-21 21:09:51,999 WARN util.Utils: Your hostname, alakouche-VM resolves to a loopback address: 127.0.1.1; using 10.0.2.15 inste
terface enp0s3)
2020-11-21 21:09:52,004 WARN util.Utils: Set SPARK_LOCAL_IP if you need to bind to another address
2020-11-21 21:09:53,711 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java clas
 applicable
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 2.4.3
      /_/

Using Python version 2.7.17 (default, Sep 30 2020 13:38:04)
SparkSession available as 'spark'.
>>> words = sc.textFile("/poeme.txt").flatMap(lambda line: line.split(" "))
>>> wordCounts = words.map(lambda word: (word, 1)).reduceByKey(lambda a,b:a +b)
>>> wordCounts.saveAsTextFile("file2.count")
>>> exit()
```

```
hduser@alakouche-VM:/usr/local/spark$ cd /usr/local/hadoop
hduser@alakouche-VM:/usr/local/hadoop$ hadoop fs -get file2.count
2020-11-21 21:12:59,137 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where
 applicable
2020-11-21 21:13:02,985 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
hduser@alakouche-VM:/usr/local/hadoop$ bin/hdfs dfs -ls .
2020-11-21 21:13:42,775 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where
 applicable
Found 2 items
drwxr-xr-x   - hduser supergroup          0 2020-11-21 21:03 file1.count
drwxr-xr-x   - hduser supergroup          0 2020-11-21 21:11 file2.count
hduser@alakouche-VM:/usr/local/hadoop$ bin/hdfs dfs -cat file2.count/part-00000
2020-11-21 21:14:28,020 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where
 applicable
2020-11-21 21:14:32,842 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
```

```
(u'croyait', 20)
(u'', 2)
(u'prisonniere', 1)
(u'trompa', 1)
(u'coeur', 2)
(u'couleur', 1)
(u'nouvelle', 1)
(u'soldats', 1)
(u'rouge', 1)
(u'tombe', 1)
(u'derob\xe2t', 1)
(u'citadelle', 1)
(u'tira', 1)
```

```
(u'passent', 1)
(u'bas', 1)
(u'framboise', 1)
(u'ruisselle', 1)
(u'sang', 1)
(u'leur', 2)
(u'gele', 1)
(u'chancelle', 1)
(u'violoncelle', 1)
(u'fait', 1)
(u'en', 2)
(u'rats', 1)
(u'bles', 1)
```

```
(u'adoraient', 1)
(u'et', 11)
(u'eclat', 1)
(u'fou', 2)
(u'font', 1)
(u'ailes', 1)
(u'levres', 1)
(u'sur', 1)
(u'sanglots', 1)
(u'muscat', 1)
(u'lun', 3)
(u'mele', 1)
(u'terre', 1)
```

...

V Exécution du « Word Count » en utilisant un script python

- La création de fichier **word_count.py**

```
hduser@alakouche-VM:/usr/local/hadoop$ cd /usr/local/spark
hduser@alakouche-VM:/usr/local/spark$ touch word_count.py
hduser@alakouche-VM:/usr/local/spark$ vim Word_count.py
```

```
File  Edit  View  Search  Terminal  Help
import sys
from pyspark import SparkContext, SparkConf
if __name__ == "__main__":
    # create Spark context with necessary configuration
    sc = SparkContext("local","PySpark Word Count Example")
    # read data from text file and split each line into words
    words = sc.textFile("/poeme.txt").flatMap(lambda line:line.split(" "))
    # count the occurrence of each word
    wordCounts = words.map(lambda word: (word,1)).reduceByKey(lambda a,b:a +b)
    # save the counts to output
    wordCounts.saveAsTextFile("file0.count")
```

```
hduser@alakouche-VM:/usr/local/spark$ ./bin/spark-submit word_count.py
2020-11-21 22:02:50,288 WARN util.Utils: Your hostname, alakouche-VM resolves to a loopback address: 127.0.1.1; using 10.0.2.15 instead (on in
terface enp0s3)
2020-11-21 22:02:50,293 WARN util.Utils: Set SPARK_LOCAL_IP if you need to bind to another address
2020-11-21 22:02:51,808 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where
 applicable
2020-11-21 22:02:53,529 INFO spark.SparkContext: Running Spark version 2.4.3
2020-11-21 22:02:53,603 INFO spark.SparkContext: Submitted application: PySpark Word Count Example
2020-11-21 22:02:53,767 INFO spark.SecurityManager: Changing view acls to: hduser
```

```
hduser@alakouche-VM:/usr/local/spark$ hadoop fs -get file0.count
2020-11-21 22:04:13,529 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where
 applicable
2020-11-21 22:04:17,227 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
hduser@alakouche-VM:/usr/local/spark$ cd /usr/local/hadoop
hduser@alakouche-VM:/usr/local/hadoop$ bin/hdfs dfs -ls .
2020-11-21 22:05:06,228 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where
 applicable
Found 3 items
drwxr-xr-x   - hduser supergroup          0 2020-11-21 22:03 file0.count
drwxr-xr-x   - hduser supergroup          0 2020-11-21 21:03 file1.count
drwxr-xr-x   - hduser supergroup          0 2020-11-21 21:11 file2.count
```

```
hduser@alakouche-VM:/usr/local/hadoop$ bin/hdfs dfs -cat file0.count/part-00000
2020-11-21 22:10:44,148 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where
 applicable
2020-11-21 22:10:47,693 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
```

```
(u'croyait', 20)
(u'', 2)
(u'prisonniere', 1)
(u'trompa', 1)
(u'coeur', 2)
(u'couleur', 1)
(u'nouvelle', 1)
(u'soldats', 1)
(u'rouge', 1)
(u'tombe', 1)
(u'derob\xe2t', 1)
(u'citadelle', 1)
```

```
(u'sang', 1)
(u'leur', 2)
(u'gele', 1)
(u'chancelle', 1)
(u'violoncelle', 1)
(u'fait', 1)
(u'en', 2)
(u'rats', 1)
(u'bles', 1)
(u'adoraient', 1)
(u'et', 11)
(u'eclat', 1)
```

```
(u'sappelle', 1)
(u'qua', 1)
(u'les', 4)
(u'que', 2)
(u'qui', 25)
(u'lalouette', 1)
(u'jura', 1)
(u'bras', 1)
(u'grabat', 1)
(u'celle', 1)
(u'coule', 2)
(u'cette', 1)
```

Affichage de résultats

...

VI Exécution d'une application Spark Batch en Java

- Dans cette section, nous allons créer une application Spark Batch en Java (un simple **WordCount**), le charger sur le nœud en local et le lancer.

```
hduser@alakouche-VM:/usr/local/hadoop$ cd /home/hduser/Documents/
hduser@alakouche-VM:~/Documents$ ls
 apache-maven-3.5.0-bin.tar.gz  'hadoop-3.2.1(1).tar.gz'   jdk-8u71-linux-x64.tar.gz   spark-2.4.3-bin-hadoop2.7.tgz
hduser@alakouche-VM:~/Documents$ tar -zxvf apache-maven-3.5.0-bin.tar.gz
apache-maven-3.5.0/README.txt
apache-maven-3.5.0/LICENSE
apache-maven-3.5.0/NOTICE
apache-maven-3.5.0/lib/
apache-maven-3.5.0/lib/cdi-api.license
```

Après le téléchargement d'apache Maven on va On décompresse l'archive: Apache-maven-3.5.0-bin

```
hduser@alakouche-VM:~/Documents$ sudo -i
root@alakouche-VM:~# mv /home/hduser/Documents/apache-maven-3.5.0 /opt/
root@alakouche-VM:~# vim /etc/profile
root@alakouche-VM:~# logout
hduser@alakouche-VM:~/Documents$ source /etc/profile
```

mettent en place de manière permanente la variable d'environnement PATH pour tous les utilisateurs

```
    fi
  done
  unset i
fi
export JAVA_HOME=/opt/java/jdk1.8.0_71/
export JRE_HOME=/opt/java/jdk1.8.0._71/jre
export PATH=$PATH:/opt/java/jdk1.8.0_71/bin:/opt/java/jdk1.8.0_71/jre/bin
export PATH=$PATH:/opt/java/jdk1.8.0_71/bin:/opt/java/jdk1.8.0_71/jre/bin:/opt/apache-maven-3.5.0/bin
~
~
:x
```

```
hduser@alakouche-VM:~$ source /etc/profile
hduser@alakouche-VM:~$ mvn -v
Apache Maven 3.5.0 (ff8f5e7444045639af65f6095c62210b5713f426; 2017-04-03T20:39:06+01:00)
Maven home: /opt/apache-maven-3.5.0
Java version: 1.8.0_71, vendor: Oracle Corporation
Java home: /opt/java/jdk1.8.0_71/jre
Default locale: en_US, platform encoding: UTF-8
OS name: "linux", version: "5.3.0-28-generic", arch: "amd64", family: "unix"
hduser@alakouche-VM:~$
```

Assurent que la l'apache Maven a était correctement installée

```
hduser@alakouche-VM:~$ mvn archetype:generate -DarchetypeArtifactId=maven-archetype-quickstart -DarchetypeVersion=1.1
[INFO] Scanning for projects...
[INFO]
[INFO] ------------------------------------------------------------------------
[INFO] Building Maven Stub Project (No POM) 1
[INFO] ------------------------------------------------------------------------
[INFO]
[INFO] >>> maven-archetype-plugin:3.2.0:generate (default-cli) > generate-sources @ standalone-pom >>>
[INFO]
```

Création d'un projet Maven en utilisent les paramètres suivantes

```
[INFO] Parameter: basedir, Value: /home/hduser
[INFO] Parameter: package, Value: DataEngineer.myapp
[INFO] Parameter: groupId, Value: DataEngineer.myapp
[INFO] Parameter: artifactId, Value: myapp
[INFO] Parameter: packageName, Value: DataEngineer.myapp
[INFO] Parameter: version, Value: 1.0-SNAPSHOT
[INFO] project created from Old (1.x) Archetype in dir: /home/hduser/myapp
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 14:26 min
[INFO] Finished at: 2020-11-22T14:59:55Z
[INFO] Final Memory: 16M/60M
[INFO] ------------------------------------------------------------------------
hduser@alakouche-VM:~$ clear
```

- On compile notre application java

```
hduser@alakouche-VM:~$ cd myapp/
hduser@alakouche-VM:~/myapp$ mvn package
[INFO] Scanning for projects...
[INFO]
[INFO] ------------------------------------------------------------------------
[INFO] Building myapp DataEngineer.myapp
[INFO] ------------------------------------------------------------------------
Downloading: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-resources-plugin/2.6/maven-resources-plugin-2.6.pom
Downloaded: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-resources-plugin/2.6/maven-resources-plugin-2.6.pom (8.1 kB at
Downloading: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-archiver/2.1/plexus-archiver-2.1.jar
Downloading: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-io/2.0.2/plexus-io-2.0.2.jar
Downloading: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-interpolation/1.15/plexus-interpolation-1.15.jar
Downloading: https://repo.maven.apache.org/maven2/commons-lang/commons-lang/2.1/commons-lang-2.1.jar
Downloaded: https://repo.maven.apache.org/maven2/org/apache/maven/maven-archiver/2.5/maven-archiver-2.5.jar (22 kB at 12 kB/s)
Downloading: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-utils/3.0/plexus-utils-3.0.jar
Downloaded: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-io/2.0.2/plexus-io-2.0.2.jar (58 kB at 17 kB/s)
Downloaded: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-interpolation/1.15/plexus-interpolation-1.15.jar (60 kB at 12 kB/s
)
Downloaded: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-utils/3.0/plexus-utils-3.0.jar (226 kB at 33 kB/s)
Downloaded: https://repo.maven.apache.org/maven2/commons-lang/commons-lang/2.1/commons-lang-2.1.jar (208 kB at 26 kB/s)
Downloaded: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-archiver/2.1/plexus-archiver-2.1.jar (184 kB at 17 kB/s)
[INFO] Building jar: /home/hduser/myapp/target/myapp-DataEngineer.myapp.jar
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 04:14 min
[INFO] Finished at: 2020-11-21T23:28:15Z
[INFO] Final Memory: 19M/60M
[INFO] ------------------------------------------------------------------------
```

Installation de la commande **tree**

```
hduser@alakouche-VM:~/myapp$ sudo apt install tree
[sudo] password for hduser:
Reading package lists... Done
```

Visualisation de toute l'arborescence de notre projet

```
hduser@alakouche-VM:~$ tree myapp/
myapp/
├── pom.xml
├── src
│   ├── main
│   │   └── java
│   │       └── DataEngineer
│   │           └── myapp
│   │               └── App.java
│   └── test
│       └── java
│           └── DataEngineer
│               └── myapp
│                   └── AppTest.java
└── target
    ├── classes
    │   └── DataEngineer
    │       └── myapp
    │           └── App.class
    ├── maven-archiver
    │   └── pom.properties
    ├── maven-status
    │   └── maven-compiler-plugin
    │       ├── compile
    │       │   └── default-compile
    │       │       ├── createdFiles.lst
    │       │       └── inputFiles.lst
    │       └── testCompile
    │           └── default-testCompile
    │               ├── createdFiles.lst
    │               └── inputFiles.lst
    ├── myapp-1.0-SNAPSHOT.jar
    ├── surefire-reports
    │   ├── DataEngineer.myapp.AppTest.txt
    │   └── TEST-DataEngineer.myapp.AppTest.xml
    └── test-classes
        └── DataEngineer
            └── myapp
                └── AppTest.class

24 directories, 13 files
```


Remarque : Le projet Maven créé correspond à un projet java « **Hello World** » sur lequel nous allons nous baser pour créer notre application java « **WordCount** ».

- Reconfiguration du projet Maven :

```
hduser@alakouche-VM:~$ cd myapp/
hduser@alakouche-VM:~/myapp$ vim pom.xml
hduser@alakouche-VM:~/myapp$
```

Modification de fichier xml **pom.xml**

```
  <properties>
          <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
          <maven.compiler.source>1.8</maven.compiler.source>
          <maven.compiler.target>1.8</maven.compiler.target>
  </properties>

  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>3.8.1</version>
      <scope>test</scope>
    </dependency>
    <dependency>
        <groupId>org.apache.spark</groupId>
        <artifactId>spark-core_2.11</artifactId>
        <version>2.1.0</version>
        </dependency>
        <dependency>
        <groupId>org.slf4j</groupId>
        <artifactId>slf4j-log4j12</artifactId>
        <version>1.7.22</version>
     </dependency>
  </dependencies>
"pom.xml" 37L, 1124C
```

```
hduser@alakouche-VM:~/myapp$ cd /home/hduser/myapp/src/main/java/DataEngineer/myapp
hduser@alakouche-VM:~/myapp/src/main/java/DataEngineer/myapp$ ls
App.java
hduser@alakouche-VM:~/myapp/src/main/java/DataEngineer/myapp$ mv App.java WordCountTask.java
hduser@alakouche-VM:~/myapp/src/main/java/DataEngineer/myapp$ ls
WordCountTask.java
hduser@alakouche-VM:~/myapp/src/main/java/DataEngineer/myapp$ gedit WordCountTask.java
hduser@alakouche-VM:~/myapp/src/main/java/DataEngineer/myapp$
```

Renomment **App.java** en **WordCountTask.java**

Open ▾ *WordCountTask.java
~/myapp/src/main/java/DataEngineer/myapp

```
package DataEngineer.myapp;

import org.apache.spark.SparkConf;
import org.apache.spark.api.java.JavaPairRDD;
import org.apache.spark.api.java.JavaRDD;
import org.apache.spark.api.java.JavaSparkContext;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import scala.Tuple2;
import java.util.Arrays;
import static jersey.repackaged.com.google.common.base.Preconditions.checkArgument;

public class WordCountTask {
private static final Logger LOGGER = LoggerFactory.getLogger(WordCountTask.class);
public static void main(String[] args) {
checkArgument(args.length > 1, "Please provide the path of input file and output dir as parameters.");
new WordCountTask().run(args[0], args[1]);
}
public void run(String inputFilePath, String outputDir) {
SparkConf conf = new SparkConf().setAppName(WordCountTask.class.getName());
JavaSparkContext sc = new JavaSparkContext(conf);
JavaRDD<String> textFile = sc.textFile(inputFilePath);
JavaPairRDD<String, Integer> counts = textFile
        .flatMap(s -> Arrays.asList(s.split(" ")).iterator())
        .mapToPair(word -> new Tuple2<>(word, 1))
        .reduceByKey((a, b) -> a + b);
counts.saveAsTextFile(outputDir);
}
}
```

Le contenu de **WordCountTask.java**

- Compilation de notre app **WordCountTask**

```
hduser@alakouche-VM:~/myapp/src/main/java/DataEngineer/myapp$ cd ~/myapp
hduser@alakouche-VM:~/myapp$ mvn package
[INFO] Scanning for projects...
[INFO]
[INFO] ------------------------------------------------------------------------
[INFO] Building myapp 1.0-SNAPSHOT
[INFO] ------------------------------------------------------------------------
[INFO]
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ myapp ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory /home/hduser/myapp/src/main/resources
[INFO]
[INFO] --- maven-compiler-plugin:3.1:compile (default-compile) @ myapp ---
[INFO] Changes detected - recompiling the module!
```

```
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0

[INFO]
[INFO] --- maven-jar-plugin:2.4:jar (default-jar) @ myapp ---
[INFO] Building jar: /home/hduser/myapp/target/myapp-1.0-SNAPSHOT.jar
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 28.699 s
[INFO] Finished at: 2020-11-22T15:15:44Z
[INFO] Final Memory: 32M/76M
[INFO] ------------------------------------------------------------------------
hduser@alakouche-VM:~/myapp$
```

- Nettoyage et Formatage du nœud Hadoop

```
hduser@alakouche-VM:~/myapp$ cd /usr/local/hadoop_store/
hduser@alakouche-VM:/usr/local/hadoop_store$ rm -rf *
hduser@alakouche-VM:/usr/local/hadoop_store$ mkdir -p /usr/local/hadoop_store/hdfs/namenode
hduser@alakouche-VM:/usr/local/hadoop_store$ mkdir -p /usr/local/hadoop_store/hdfs/datanode
hduser@alakouche-VM:/usr/local/hadoop_store$ chown -R hduser /usr/local/hadoop_store/hdfs/namenode
hduser@alakouche-VM:/usr/local/hadoop_store$ chown -R hduser /usr/local/hadoop_store/hdfs/datanode
hduser@alakouche-VM:/usr/local/hadoop_store$
hduser@alakouche-VM:/usr/local/hadoop_store$ cd /usr/local/hadoop/etc/hadoop
hduser@alakouche-VM:/usr/local/hadoop/etc/hadoop$ hdfs namenode -format
2020-11-22 15:21:00,007 INFO namenode.NameNode: STARTUP_MSG:
/************************************************************
STARTUP_MSG: Starting NameNode
STARTUP_MSG:   host = alakouche-VM/127.0.1.1
STARTUP_MSG:   args = [-format]
STARTUP_MSG:   version = 3.2.1
```

```
2020-11-22 15:21:12,515 INFO util.GSet: Computing capacity for map NameNodeRetryCache
2020-11-22 15:21:12,529 INFO util.GSet: VM type       = 64-bit
2020-11-22 15:21:12,537 INFO util.GSet: 0.029999999329447746% max memory 953.2 MB = 292.8 KB
2020-11-22 15:21:12,538 INFO util.GSet: capacity      = 2^15 = 32768 entries
2020-11-22 15:21:12,990 INFO namenode.FSImage: Allocated new BlockPoolId: BP-513619828-127.0.1.1-1606058472869
2020-11-22 15:21:13,222 INFO common.Storage: Storage directory /usr/local/hadoop_store/hdfs/namenode has been successfully formatted.
2020-11-22 15:21:13,710 INFO namenode.FSImageFormatProtobuf: Saving image file /usr/local/hadoop_store/hdfs/namenode/current/fsimage.ckpt_0000
000000000000000 using no compression
2020-11-22 15:21:14,753 INFO namenode.FSImageFormatProtobuf: Image file /usr/local/hadoop_store/hdfs/namenode/current/fsimage.ckpt_00000000000
00000000 of size 401 bytes saved in 1 seconds .
2020-11-22 15:21:14,904 INFO namenode.NNStorageRetentionManager: Going to retain 1 images with txid >= 0
2020-11-22 15:21:14,966 INFO namenode.FSImage: FSImageSaver clean checkpoint: txid=0 when meet shutdown.
2020-11-22 15:21:14,967 INFO namenode.NameNode: SHUTDOWN_MSG:
/************************************************************
SHUTDOWN_MSG: Shutting down NameNode at alakouche-VM/127.0.1.1
************************************************************/
hduser@alakouche-VM:/usr/local/hadoop/etc/hadoop$
```

- Démarrage de Hadoop

```
hduser@alakouche-VM:/usr/local/hadoop/etc/hadoop$ start-dfs.sh
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [alakouche-VM]
2020-11-22 15:47:20,984 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes
 applicable
hduser@alakouche-VM:/usr/local/hadoop/etc/hadoop$ start-yarn.sh
Starting resourcemanager
Starting nodemanagers
hduser@alakouche-VM:/usr/local/hadoop/etc/hadoop$ jps
23378 Jps
22370 SecondaryNameNode
23108 NodeManager
21557 DataNode
22886 ResourceManager
21371 NameNode
hduser@alakouche-VM:/usr/local/hadoop/etc/hadoop$
```

```
hduser@alakouche-VM:/usr/local/hadoop/etc/hadoop$ hdfs dfsadmin -report
2020-11-22 15:50:09,413 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes
 applicable
Configured Capacity: 21001486336 (19.56 GB)
Present Capacity: 10665234432 (9.93 GB)
```

```
        Block groups with corrupt internal blocks: 0
        Missing block groups: 0
        Low redundancy blocks with highest priority to recover: 0
        Pending deletion blocks: 0

-------------------------------------------------
Live datanodes (1):

Name: 127.0.0.1:9866 (localhost)
Hostname: alakouche-VM
Decommission Status : Normal
Configured Capacity: 21001486336 (19.56 GB)
DFS Used: 24576 (24 KB)
```

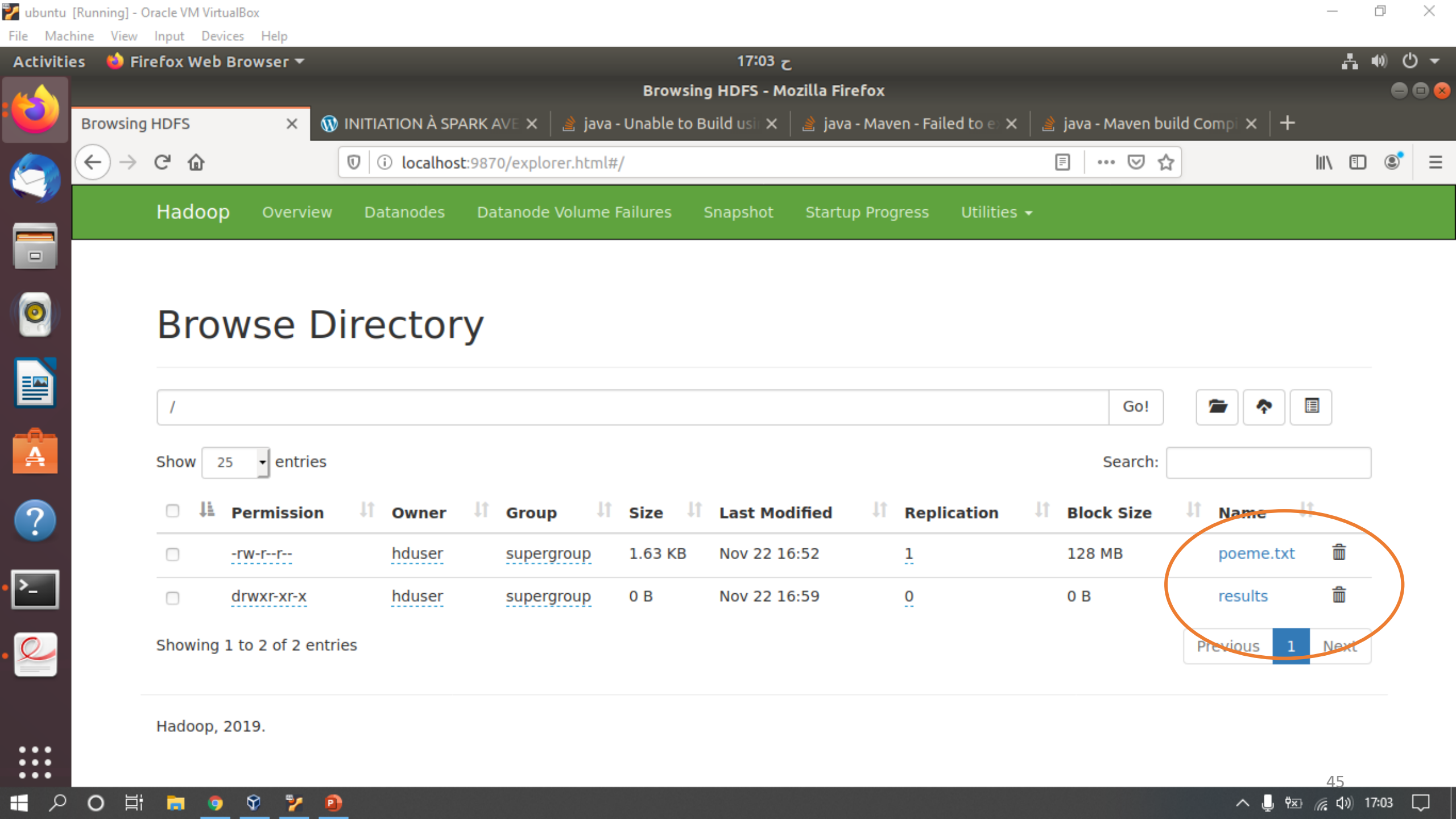
On vérifie le bon fonctionnement de notre nœud

- Dépôt du poeme.txt dans HDFS

```
hduser@alakouche-VM:/usr/local/hadoop/etc/hadoop$ cd ..
hduser@alakouche-VM:/usr/local/hadoop/etc$ cd ..
hduser@alakouche-VM:/usr/local/hadoop$ bin/hdfs dfs -put /usr/local/spark/poeme.txt /
2020-11-22 15:52:46,994 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where
 applicable
2020-11-22 15:52:57,164 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
hduser@alakouche-VM:/usr/local/hadoop$ bin/hdfs dfs -ls /
2020-11-22 15:53:36,727 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where
 applicable
Found 1 items
-rw-r--r--   1 hduser supergroup       1670 2020-11-22 15:52 /poeme.txt
hduser@alakouche-VM:/usr/local/hadoop$
```

- On lance la commande : spark-submit

```
hduser@alakouche-VM:/usr/local/hadoop$ spark-submit --class DataEngineer.myapp.WordCountTask /home/hduser/myapp/target/myapp-1.0-SNAPSHOT.jar
/poeme.txt /results
2020-11-22 15:58:12,603 WARN util.Utils: Your hostname, alakouche-VM resolves to a loopback address: 127.0.1.1; using 10.0.2.15 instead (on in
terface enp0s3)
2020-11-22 15:58:12,614 WARN util.Utils: Set SPARK_LOCAL_IP if you need to bind to another address
2020-11-22 15:58:19,279 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where
 applicable
2020-11-22 15:58:22,283 INFO spark.SparkContext: Running Spark version 2.4.3
2020-11-22 15:58:22,451 INFO spark.SparkContext: Submitted application: DataEngineer.myapp.WordCountTask
2020-11-22 15:58:22,882 INFO spark.SecurityManager: Changing view acls to: hduser
2020-11-22 15:58:22,888 INFO spark.SecurityManager: Changing modify acls to: hduser
2020-11-22 15:58:22,889 INFO spark.SecurityManager: Changing view acls groups to:
2020-11-22 15:58:22,889 INFO spark.SecurityManager: Changing modify acls groups to:
2020-11-22 15:58:22,891 INFO spark.SecurityManager: SecurityManager: authentication disabled; ui acls disabled; users  with view permissions:
Set(hduser); groups with view permissions: Set(); users  with modify permissions: Set(hduser); groups with modify permissions: Set()
2020-11-22 15:58:26,653 INFO util.Utils: Successfully started service 'sparkDriver' on port 41839.
```


ubuntu [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities
Firefox Web Browser
17:03
Browsing HDFS - Mozilla Firefox
Browsing HDFS
INITIATION À SPARK AVE
java - Unable to Build usi
java - Maven - Failed to e
java - Maven build Compi
localhost:9870/explorer.html#/
Hadoop
Overview
Datanodes
Datanode Volume Failures
Snapshot
Startup Progress
Utilities
Browse Directory
/
Go!
Show 25 entries
Search:
Permission
Owner
Group
Size
Last Modified
Replication
Block Size
Name
-rw-r--r--
hduser
supergroup
1.63 KB
Nov 22 16:52
1
128 MB
poeme.txt
drwxr-xr-x
hduser
supergroup
0 B
Nov 22 16:59
0
0 B
results
Showing 1 to 2 of 2 entries
Previous
1
Next
Hadoop, 2019.
17:03

- Affichage de résultats

```
hduser@alakouche-VM:/usr/local/hadoop$ bin/hdfs dfs -cat /results/part-00000
2020-11-22 16:01:22,593 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where
 applicable
2020-11-22 16:01:30,878 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
```

```
(jura,1)
(ils,1)
(violoncelle,1)
(comment,1)
(rose,1)
(soldats,1)
(que,2)
(celui,20)
(levres,1)
(bretagne,1)
(ses,1)
(quelle,1)
(de,5)
(les,4)
(cruelle,1)
(tombe,1)
(ruisselle,1)
(combat,1)
(prefere,1)
(fideles,1)
(au,11)
(glas,1)
(un,4)
(quil,1)
(coule,2)
```

```
(montait,1)
(quand,2)
(ciel,10)
(coeur,2)
(double,1)
(saison,1)
(qui,25)
(lechelle,1)
(est,1)
(lhirondelle,1)
(fait,1)
(tira,1)
(court,1)
(pour,1)
(croyait,20)
(bas,1)
(plus,2)
(passent,1)
(commun,1)
(font,1)
(qua,1)
(le,6)
(muscat,1)
(tous,3)
(gele,1)
```

```
gele,1)
lalouette,1)
fois,1)
citadelle,1)
aima,1)
eclat,1)
sang,1)
terre,1)
nouvelle,1)
framboise,1)
(prison,1)
(disaient,1)
(sur,1)
(adoraient,1)
(en,2)
(la,8)
(ne,1)
(par,1)
(deux,6)
(fut,1)
(sous,1)
(a,6)
(rats,1)
(cette,1)
(pas,11)
```

```
deux,6)
fut,1)
sous,1)
a,6)
rats,1)
cette,1)
pas,11)
,1)
repetant,1)
derobât,1)
quaucun,1)
querelles,1)
celle,1)
lequel,5)
sy,1)
trepas,1)
grele,1)
bles,1)
lautre,4)
se,1)
haut,1)
reseda,1)
ny,10)
belle,1)
laube,1)
```

- Dans cette Etude bibliographique on va détailler les points suivantes

✓ **c'est quoi Spark SQL ?**

✓ **Comment spark SQL fonctionne-t-il?**

✓ **C’est quoi un Dataframes ?**

✓ **Démonstration en Python**

Spark SQL

Spark introduit un module de programmation pour le traitement structuré des données appelé Spark SQL. Il fournit une abstraction de programmation appelée DataFrame et peut agir comme moteur de requête SQL distribué.

Caractéristiques de Spark SQL:

- **Intégré** : Mélangez parfaitement les requêtes SQL avec les programmes Spark. Spark SQL vous permet d'interroger des données structurées en tant qu'ensemble de données distribués (RDD) dans Spark, avec des API intégrées dans Python, Scala et Java. Cette intégration étroite facilite d'exécuter des requêtes SQL aux côtés d'algorithmes analytiques complexes.

- **Accès unifié aux données** :Chargez et interrogez les données provenant de diverses sources. Les Schémas-RDD fournissent une interface unique pour travailler efficacement avec des données structurées, y compris les tables Apache Hive, les fichiers parquet et les fichiers JSON.

- **Compatibilité Hive** : Exécutez des requêtes hive non modifiées sur les entrepôts existants. Spark SQL réutilise le frontend hive et metastore, vous donnant une compatibilité complète avec les données, les requêtes et les UDF de Hive existants. Il suffit de l'installer aux côtés de Hive.

- **Connectivité standard :** Connectez-vous via JDBC ou ODBC. Spark SQL inclut un mode serveur avec la connectivité JDBC et ODBC standard de l'industrie.

- **Évolutivité** : Utilisez le même moteur pour les requêtes interactives et longues. Spark SQL tire parti du modèle RDD pour soutenir la tolérance aux défauts à mi-requête, ce qui lui permet également de passer à de gros travaux. Ne vous inquiétez pas de l'utilisation d'un moteur différent pour les données historiques.

Spark SQL Architecture :

Cette architecture contient trois couches, à savoir, API , Schema RDD et Sources de données.

API : Spark est compatible avec différentes langues et Spark SQL. Il est également, pris en charge par ces langues- API (python, scala, java, HiveQL).

Schema RDD : Spark Core est conçu avec une structure de données spéciale appelée RDD. En général, Spark SQL fonctionne sur des schémas, des tableaux et des enregistrements. Par conséquent, nous pouvons utiliser le RdD Schema comme table temporaire. Nous pouvons appeler ce Schema RDD comme cadre de données.

Sources de données : Habituellement, la source de données pour spark-core est un fichier texte, un fichier Avro, etc. Toutefois, les sources de données de Spark SQL sont différentes. Il s'appelle le fichier Parquet, le document JSON, les tableaux HIVE et la base de données Cassandra.

Dataframes :

Un DataFrame est une collection distribuée de données, qui est organisée en colonnes nommées. Conceptuellement, il est équivalent à des tables relationnelles avec de bonnes techniques d'optimisation.

Un DataFrame peut être construit à partir d'un éventail de sources différentes telles que les tables hive, les fichiers de données structurées, les bases de données externes ou les RDD existants. Cette API a été conçue pour les applications modernes de Big Data et de science des données en s'inspirant de DataFrame dans la programmation R et pandas en Python.

Démonstration en Python :

Le point d'entrée dans toutes les fonctionnalités relationnelles de Spark est la classe SQLContext, ou l'un de ses décédants. Pour créer un SQLContext de base, tout ce dont vous avez besoin est un SparkContext.

- **Exemple avec JSON Dataset:**

```
>>> from pyspark.sql import SQLContext
>>> sqlContext = SQLContext(sc)
#cree schema rdd from json file
>>> people = sqlContext.jsonFile("people.json")
#register scheme rdd as a table
>>> people.registerTempTable("people")
>>> teenagers = sqlContext.sql("SELECT name FROM people WHERE age >= 13 AND
age <= 19")
```

- **Création d'une Dataframe :**

1)From csv :

```
>>> df = spark.read.csv("/src/resources/file.csv")
```

2)From text :

```
>>> df = spark.read.text("/src/resources/file.text")
```

3)From RDD :

```
>>> from  pyspark.sql  import SparkSession
>>> spark = SparkSession.builder.appName('pyspark – example
join').getOrCreate()
>>> sc = spark.sparkContext
```

```
>>> dataset1 = [{'id' : '1','name' : 'SMITH','firstname' : 'LEA'},
  {'id' : '2','name' : 'JOHN','firstname' : 'PAUL'},
  {'id' : '4','name' : 'CRUZ','firstname' : 'TOM'}]
>>> dataset2 = [{'id' : '1','Profession' :'Lawyer','Salary':60000},
  {'id' : '3','Profession' : 'Teacher','Salary' : 30000},
   {'id' : '4','Profession' : 'Actor','Salary' : 20000000}]

>>> rdd1 = sc.parallelize(dataset1)
>>> df1 = spark.createDataFrame(rdd1)
>>> print('df1')
>>> df1.show()

df1
+---------+---+------+
|firstname| id|  name|
+---------+---+------+
|      LEA|  1| SMITH|
|     PAUL|  2|  JOHN|
|      TOM|  4|CRUISE|
+---------+---+------+
```

```
>>> rdd2 = sc.parallelize(dataset2)
>>> df2 = spark.createDataFrame(rdd2)
>>> print('df2’)
>>> df2.show()
```

```
df2
+----------+--------+---+
|Profession|  Salary| id|
+----------+--------+---+
|    Lawyer|   60000|  1|
|   Teacher|   30000|  3|
|     Actor|20000000|  4|
+----------+--------+---+
```

```
>>> df = df1.join(df2, on=['id'], how='inner’)
>>> df.show()
```

```
+---+---------+------+----------+--------+
| id|firstname|  name|Profession|  Salary|
+---+---------+------+----------+--------+
|  1|      LEA| SMITH|    Lawyer|   60000|
|  4|      TOM|CRUISE|     Actor|20000000|
+---+---------+------+----------+--------+
```

```
>>> df  = df1.join(df2, df1.id > df2.id, how='inner’)
>>> df.show()
```

```
+---------+---+------+----------+------+---+
|firstname| id|  name|Profession|Salary| id|
+---------+---+------+----------+------+---+
|     PAUL|  2|  JOHN|    Lawyer| 60000|  1|
|      TOM|  4|CRUISE|    Lawyer| 60000|  1|
|      TOM|  4|CRUISE|   Teacher| 30000|  3|
+---------+---+------+----------+------+---+
```

Fin

Merci