Rapport de :

TP 2 : “Installation et utilisation d’Apache Spark ”

Réalisé par :

→ Riali Mouad

→ Addi Kamal

encadré par :

→ Pr. D.Zaidouni

2021/2020

Table de matières :

VII. Exécution d’une application Spark Batch en Java.

1. Installation d’Apache Maven.
2. Reconfiguration du projet Maven.
3. Nettoyage et Formatage du nœud hadoop.
4. Dépôt du poeme.txt dans HDFS.

Pré-requis techniques :

✗ Oracle AM VirtualBox-6.0 :

Oracle VM VirtualBox (anciennement VirtualBox) est un logiciel libre de virtualisation publié par Oracle.

Lien de Téléchargement :

https://download.virtualbox.org/virtualbox/6.0.12/virtualbox-6.0_6.0.12-133076~Ubuntu~bionic_amd64.dddeb

✗ ***Ubuntu 18.04.3*** **:**

Ubuntu est un système d’exploitation GNU/Linux basé sur la distribution Linux Debian. Il est développé, commercialisé et maintenu pour les ordinateurs individuels (desktop), les serveurs (Server) et les objets connectés (Core) par la société Canonical.

Lien de Téléchargement de la version Ubuntu 20.04 :

https://ubuntu.com/download/desktop/thank-you?version=20.04.1&architecture=amd64

✗ ***Apache Hadoop version=3.2.1 :***

est un framework libre et open source écrit en Java destiné à faciliter la création d'applications distribuées et échelon nables permettant aux applications de travailler avec des milliers de nœuds et des pétaoctets de données.données. Ainsi chaque nœud est constitué de machines standard regroupées en grappe.

Lien de Téléchargement :

https://downloads.apache.org/hadoop/common/hadoop-3.2.1/hadoop-3.2.1.tar.gz

pour Java 8 : https://github.com/sanyoushi/java-buildpack.git

✗ ***Apache Spark version=2.4.3 :***

Spark est un système de traitement rapide et parallèle. Il fournit des APIs de haut niveau en Java, Scala, Python et R, et un moteur optimisé qui supporte l’exécution des graphes. Il supporte également un ensemble d'outils de haut niveau tels que Spark SQL pour le support du traitement de données structurées, MLlib pour l'apprentissage des données, GraphX pour le traitement des graphes, et Spark Streaming pour le traitement des données en streaming.

Lien de Téléchargement :

https://archive.apache.org/dist/spark/spark-2.4.3/spark-2.4.3-bin-hadoop2.7.ttgz

✗ ***Scala :***

Scala est un langage de programmation multi-paradigme conçu à l’École polytechnique fédérale de Lausanne pour exprimer les modèles de programmation courants dans une forme concise et élégante.

✗ ***Python :***

est un langage de programmation interprété, multi-paradigme et multi plateformes. Il favorise la programmation impérative structurée, fonctionnelle et orientée objet. Il est doté d'un typage dynamique fort, d'une gestion automatique de la mémoire par ramasse-miettes et d'un système de gestion d'exceptions ; il est ainsi similaire à Perl, Ruby, Scheme, Smalltalk et Tcl.

✗ ***Apache Maven :***

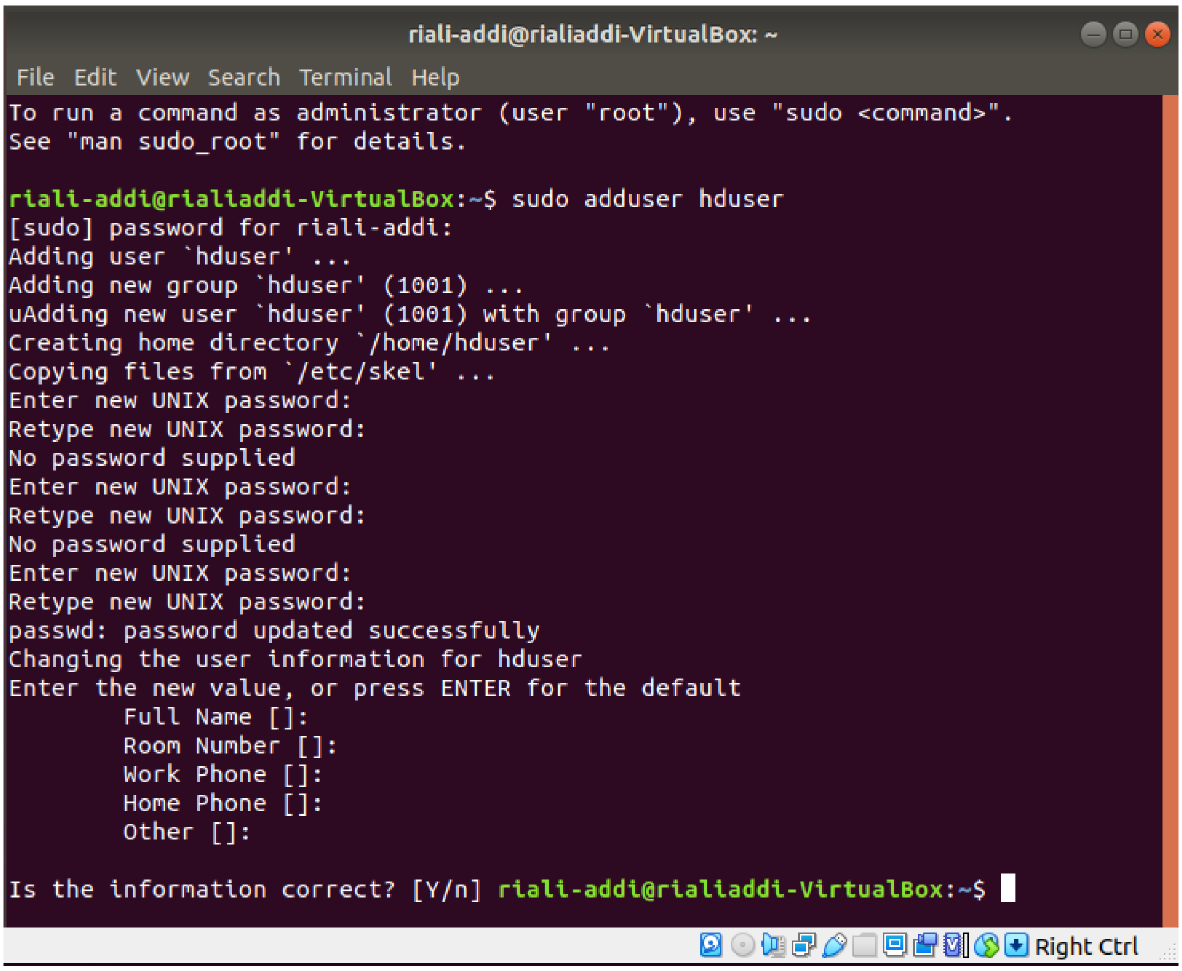
Apache Maven est un outil de gestion et d'automatisation

de production des projets logiciels Java en général et Java EE en particulier. Il est utilisé pour automatiser l'intégration continue lors d'un développement de logiciel. Maven est géré par l'organisation Apache Software Foundation.

Lien de Téléchargement :

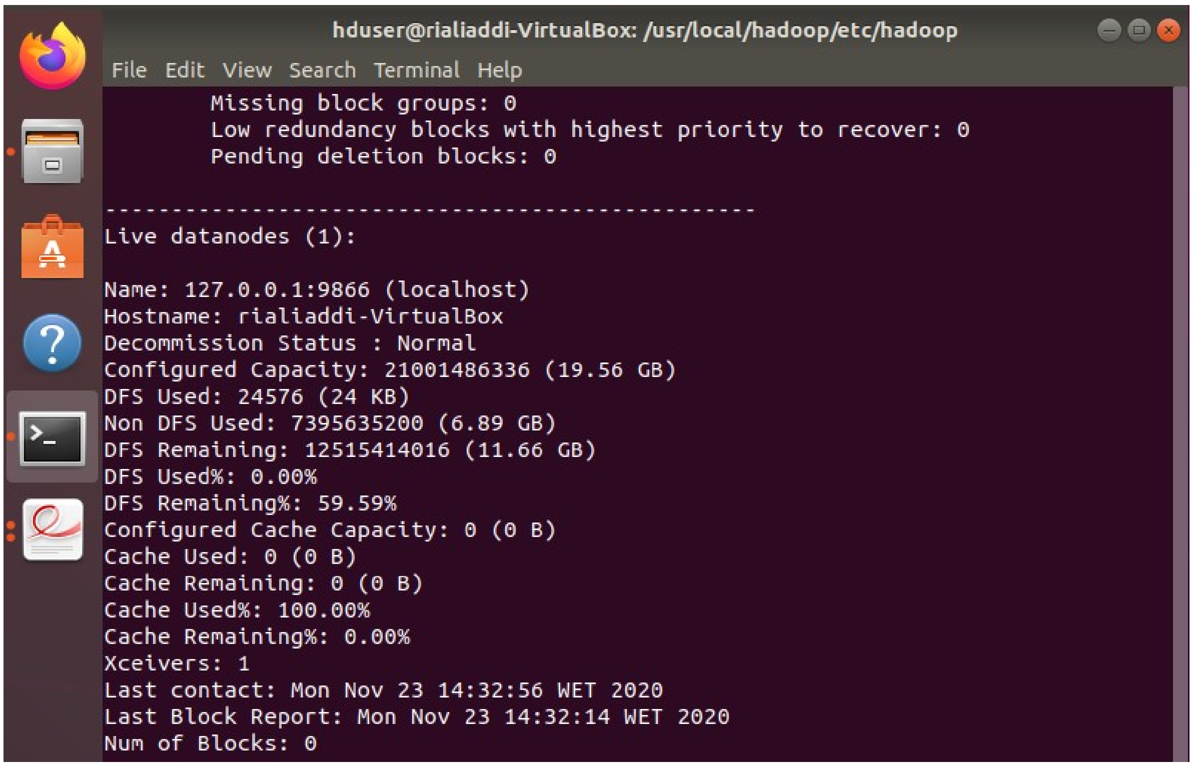
https://repo.maven.apache.org/maven2/org/apache/maven/apache-maven/3.5.0/apache-maven-3.5.0-bin.tar.gz

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


riali-addi@rialiaddi-VirtualBox: ~
File Edit View Search Terminal Help
Right Ctrl

```
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

riali-addi@rialiaddi-VirtualBox:~$ sudo adduser hduser
[sudo] password for riali-addi:
Adding user `hduser' ...
Adding new group `hduser' (1001) ...
uAdding 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:
No password supplied
Enter new UNIX password:
Retype new UNIX password:
No password supplied
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 []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:

Is the information correct? [Y/n] riali-addi@rialiaddi-VirtualBox:~$
```

- **On va suivre les mêmes étapes que le premier TP pour installer et configurer de Hadoop dans un nœud unique en serveur local pour arriver enfin au résultat suivant :**



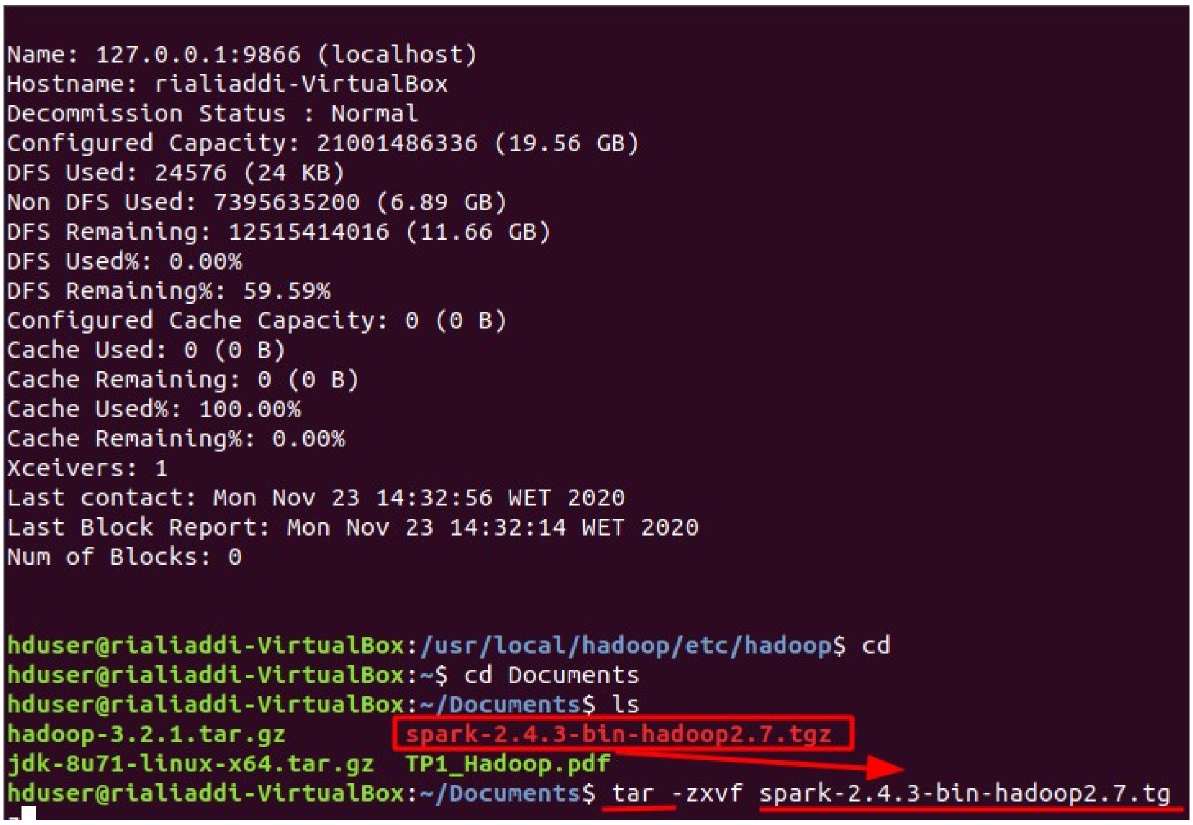
hduser@rialiaddi-VirtualBox: /usr/local/hadoop/etc/hadoop
File Edit View Search Terminal Help

```
        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: rialiaddi-VirtualBox
Decommission Status : Normal
Configured Capacity: 21001486336 (19.56 GB)
DFS Used: 24576 (24 KB)
Non DFS Used: 7395635200 (6.89 GB)
DFS Remaining: 12515414016 (11.66 GB)
DFS Used%: 0.00%
DFS Remaining%: 59.59%
Configured Cache Capacity: 0 (0 B)
Cache Used: 0 (0 B)
Cache Remaining: 0 (0 B)
Cache Used%: 100.00%
Cache Remaining%: 0.00%
Xceivers: 1
Last contact: Mon Nov 23 14:32:56 WET 2020
Last Block Report: Mon Nov 23 14:32:14 WET 2020
Num of Blocks: 0
```

II. Installation et configuration de spark en local :

Récupération les fichiers sources :

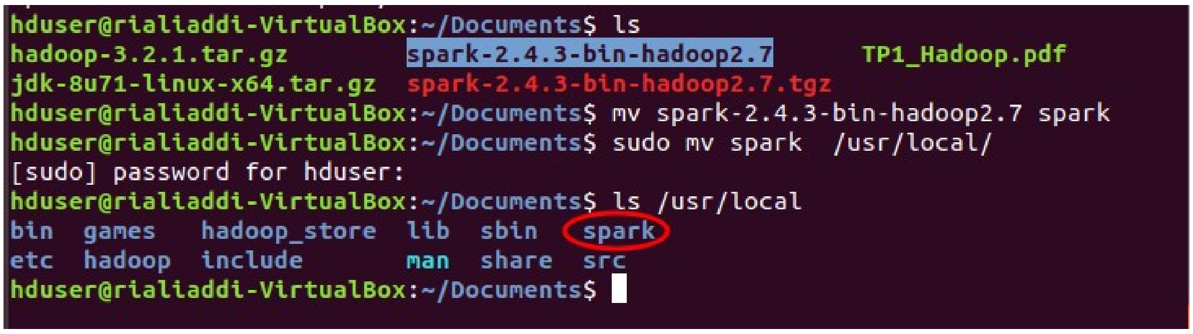
Après avoir téléchargé : **spark-2.4.3-bin-hadoop2.7.tgz**, On va l’extraire :



```
Name: 127.0.0.1:9866 (localhost)
Hostname: rialiaddi-VirtualBox
Decommission Status : Normal
Configured Capacity: 21001486336 (19.56 GB)
DFS Used: 24576 (24 KB)
Non DFS Used: 7395635200 (6.89 GB)
DFS Remaining: 12515414016 (11.66 GB)
DFS Used%: 0.00%
DFS Remaining%: 59.59%
Configured Cache Capacity: 0 (0 B)
Cache Used: 0 (0 B)
Cache Remaining: 0 (0 B)
Cache Used%: 100.00%
Cache Remaining%: 0.00%
Xceivers: 1
Last contact: Mon Nov 23 14:32:56 WET 2020
Last Block Report: Mon Nov 23 14:32:14 WET 2020
Num of Blocks: 0


hduser@rialiaddi-VirtualBox:/usr/local/hadoop/etc/hadoop$ cd
hduser@rialiaddi-VirtualBox:~$ cd Documents
hduser@rialiaddi-VirtualBox:~/Documents$ ls
hadoop-3.2.1.tar.gz          spark-2.4.3-bin-hadoop2.7.tgz
jdk-8u71-linux-x64.tar.gz    TP1_Hadoop.pdf
hduser@rialiaddi-VirtualBox:~/Documents$ tar -zxvf spark-2.4.3-bin-hadoop2.7.tg
```

Décompression le fichier récupéré dans le répertoire de votre choix :

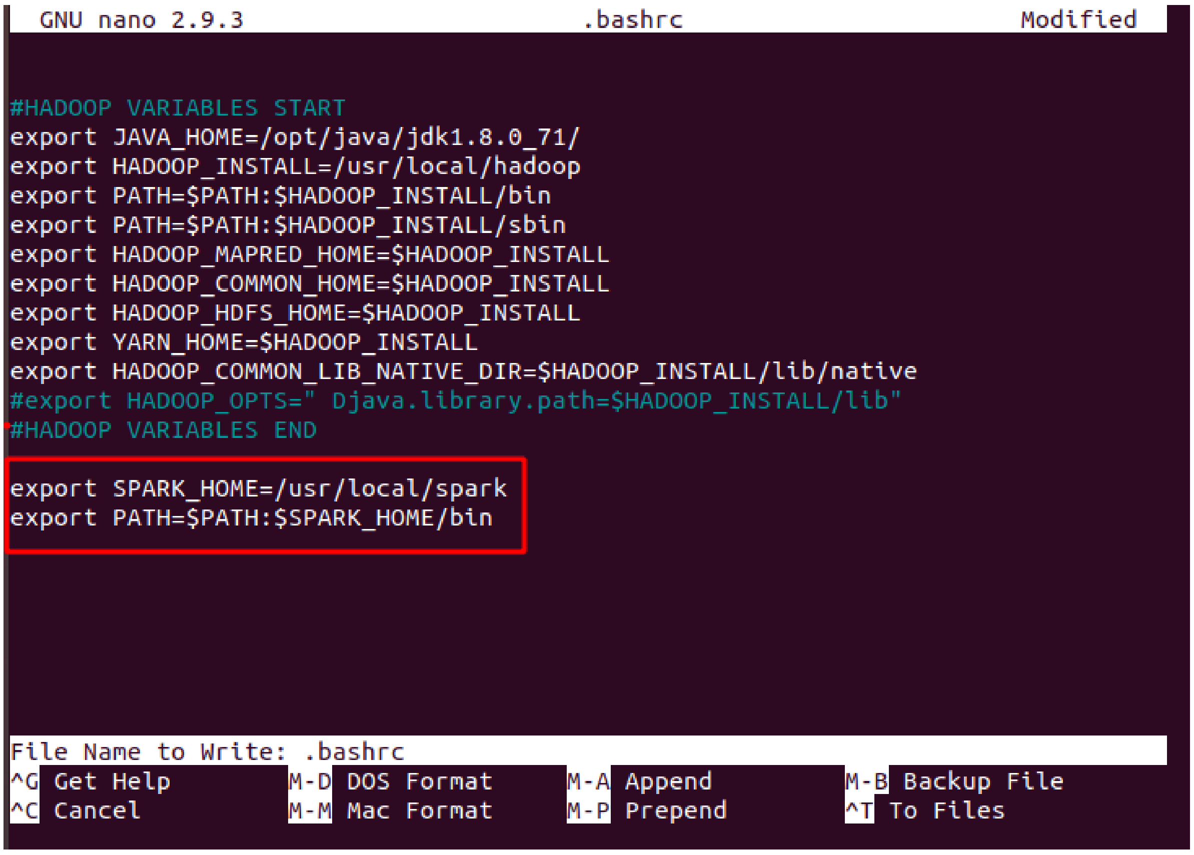
Puis, on va déplacer le fichier extrait vers le répertoire ***/usr*/local/Spark** :



```
hduser@rialiaddi-VirtualBox:~/Documents$ ls
hadoop-3.2.1.tar.gz        spark-2.4.3-bin-hadoop2.7      TP1_Hadoop.pdf
jdk-8u71-linux-x64.tar.gz  spark-2.4.3-bin-hadoop2.7.tgz
hduser@rialiaddi-VirtualBox:~/Documents$ mv spark-2.4.3-bin-hadoop2.7 spark
hduser@rialiaddi-VirtualBox:~/Documents$ sudo mv spark  /usr/local/
[sudo] password for hduser:
hduser@rialiaddi-VirtualBox:~/Documents$ ls /usr/local
bin  games   hadoop_store  lib  sbin  spark
etc  hadoop  include       man  share  src
hduser@rialiaddi-VirtualBox:~/Documents$
```

Configuration le PATH dans le .bashrc :

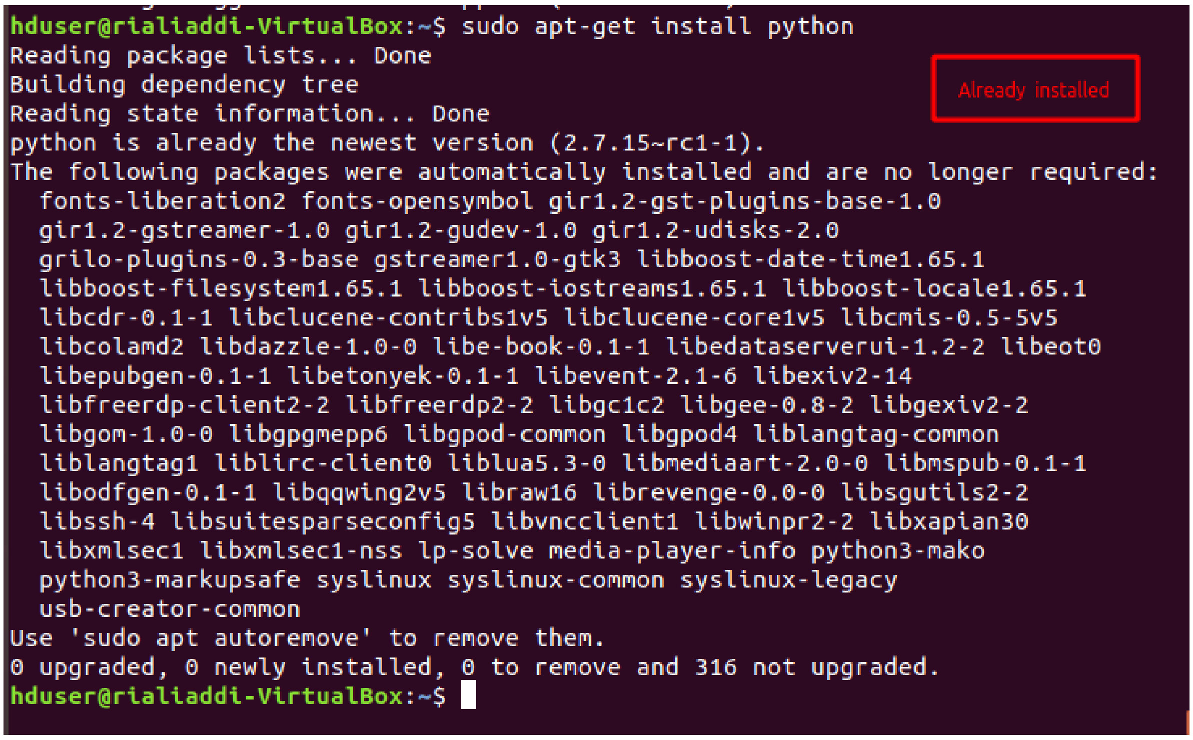
Ensuite, il faut configurer le fichier **.bashrc** en ajoutant les chemins liées au Spark comme: **SPARK_HOME….etc:**



```
  GNU nano 2.9.3                     .bashrc                          Modified


#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

export SPARK_HOME=/usr/local/spark
export PATH=$PATH:$SPARK_HOME/bin




File Name to Write: .bashrc
^G Get Help        M-D DOS Format      M-A Append          M-B Backup File
^C Cancel          M-M Mac Format      M-P Prepend         ^T To Files
```

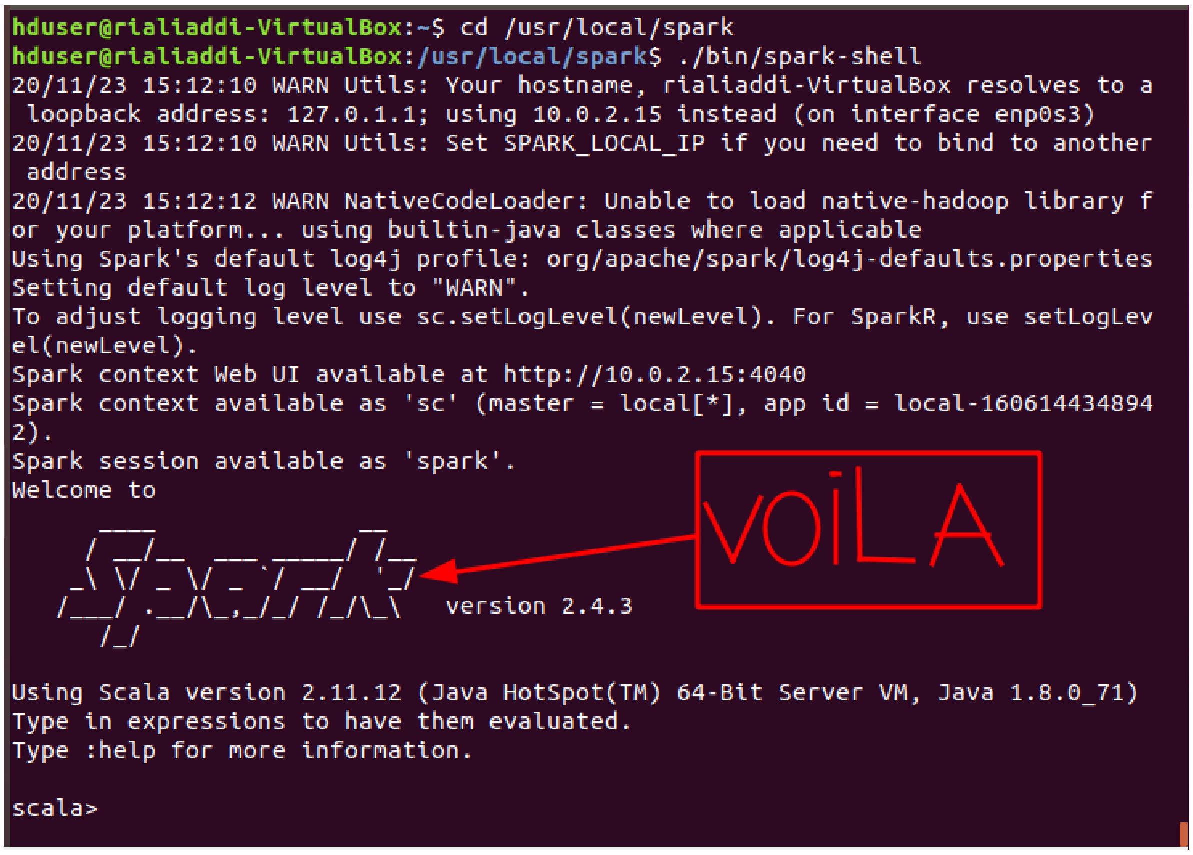
Installation de Python :

D’abord c’est le moment de l’installation de **Python 2.7 :**

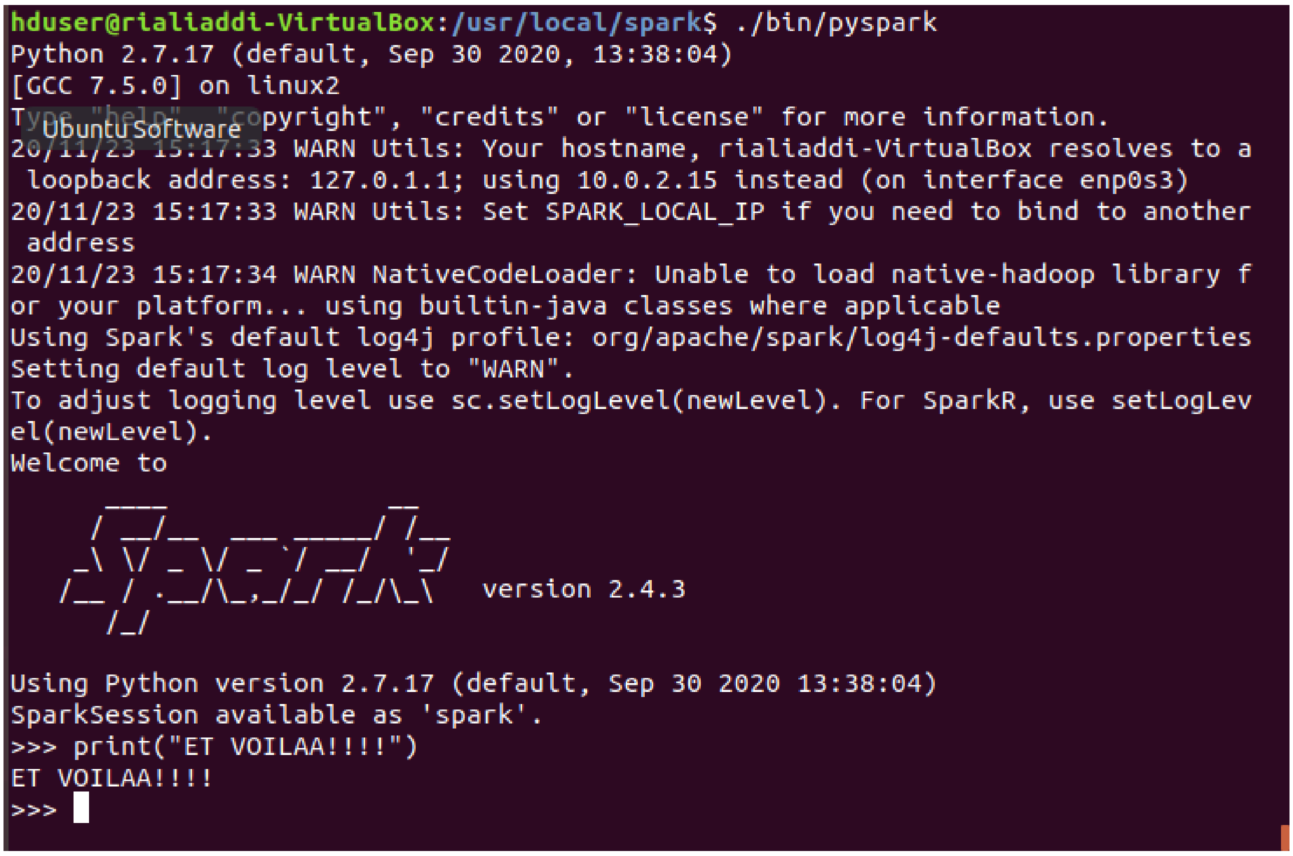


```
hduser@rialiaddi-VirtualBox:~$ sudo apt-get install python
Reading package lists... Done
Building dependency tree
Reading state information... Done
python is already the newest version (2.7.15~rc1-1).
The following packages were automatically installed and are no longer required:
  fonts-liberation2 fonts-opensymbol gir1.2-gst-plugins-base-1.0
  gir1.2-gstreamer-1.0 gir1.2-gudev-1.0 gir1.2-udisks-2.0
  grilo-plugins-0.3-base gstreamer1.0-gtk3 libboost-date-time1.65.1
  libboost-filesystem1.65.1 libboost-iostreams1.65.1 libboost-locale1.65.1
  libcdr-0.1-1 libclucene-contribs1v5 libclucene-core1v5 libcmis-0.5-5v5
  libcolamd2 libdazzle-1.0-0 libe-book-0.1-1 libedataserverui-1.2-2 libeot0
  libepubgen-0.1-1 libetonyek-0.1-1 libevent-2.1-6 libexiv2-14
  libfreerdp-client2-2 libfreerdp2-2 libgc1c2 libgee-0.8-2 libgexiv2-2
  libgom-1.0-0 libgpgmepp6 libgpod-common libgpod4 liblangtag-common
  liblangtag1 liblirc-client0 liblua5.3-0 libmediaart-2.0-0 libmspub-0.1-1
  libodfgen-0.1-1 libqqwing2v5 libraw16 librevenge-0.0-0 libsgutils2-2
  libssh-4 libsuitesparseconfig5 libvncclient1 libwinpr2-2 libxapian30
  libxmlsec1 libxmlsec1-nss lp-solve media-player-info python3-mako
  python3-markupsafe syslinux syslinux-common syslinux-legacy
  usb-creator-common
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 316 not upgraded.
hduser@rialiaddi-VirtualBox:~$
```

Et pour être sûr que tout va bien on peut exécuter ces commandes-là pour accéder au Spark-shell et pyspark, pour nous on a trouvé ces résultats suivants :



```
hduser@rialiaddi-VirtualBox:~$ cd /usr/local/spark
hduser@rialiaddi-VirtualBox:/usr/local/spark$ ./bin/spark-shell
20/11/23 15:12:10 WARN Utils: Your hostname, rialiaddi-VirtualBox resolves to a
 loopback address: 127.0.1.1; using 10.0.2.15 instead (on interface enp0s3)
20/11/23 15:12:10 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another
 address
20/11/23 15:12:12 WARN NativeCodeLoader: Unable to load native-hadoop library f
or 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 setLogLev
el(newLevel).
Spark context Web UI available at http://10.0.2.15:4040
Spark context available as 'sc' (master = local[*], app id = local-160614434894
2).
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>
```



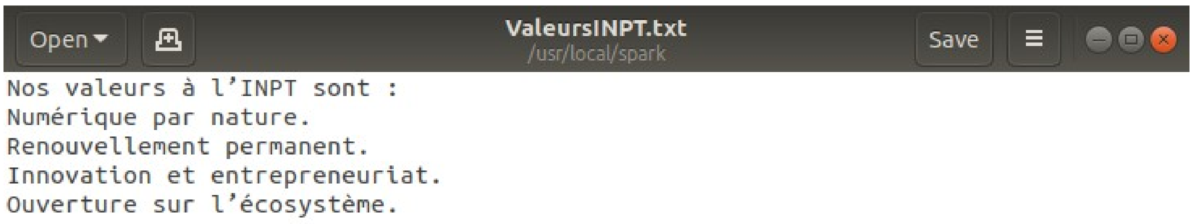
```
hduser@rialiaddi-VirtualBox:/usr/local/spark$ ./bin/pyspark
Python 2.7.17 (default, Sep 30 2020, 13:38:04)
[GCC 7.5.0] on linux2
Type "help", "copyright", "credits" or "license" for more information.
20/11/23 15:17:33 WARN Utils: Your hostname, rialiaddi-VirtualBox resolves to a
 loopback address: 127.0.1.1; using 10.0.2.15 instead (on interface enp0s3)
20/11/23 15:17:33 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another
 address
20/11/23 15:17:34 WARN NativeCodeLoader: Unable to load native-hadoop library f
or 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 setLogLev
el(newLevel).
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 2.4.3
      /_/

Using Python version 2.7.17 (default, Sep 30 2020 13:38:04)
SparkSession available as 'spark'.
>>> print("ET VOILAA!!!!")
ET VOILAA!!!!
>>>
```

III. Manipulation des RDDs en utilisant le terminal pyspark :

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

Enregistrement et chargement des Textfile. :

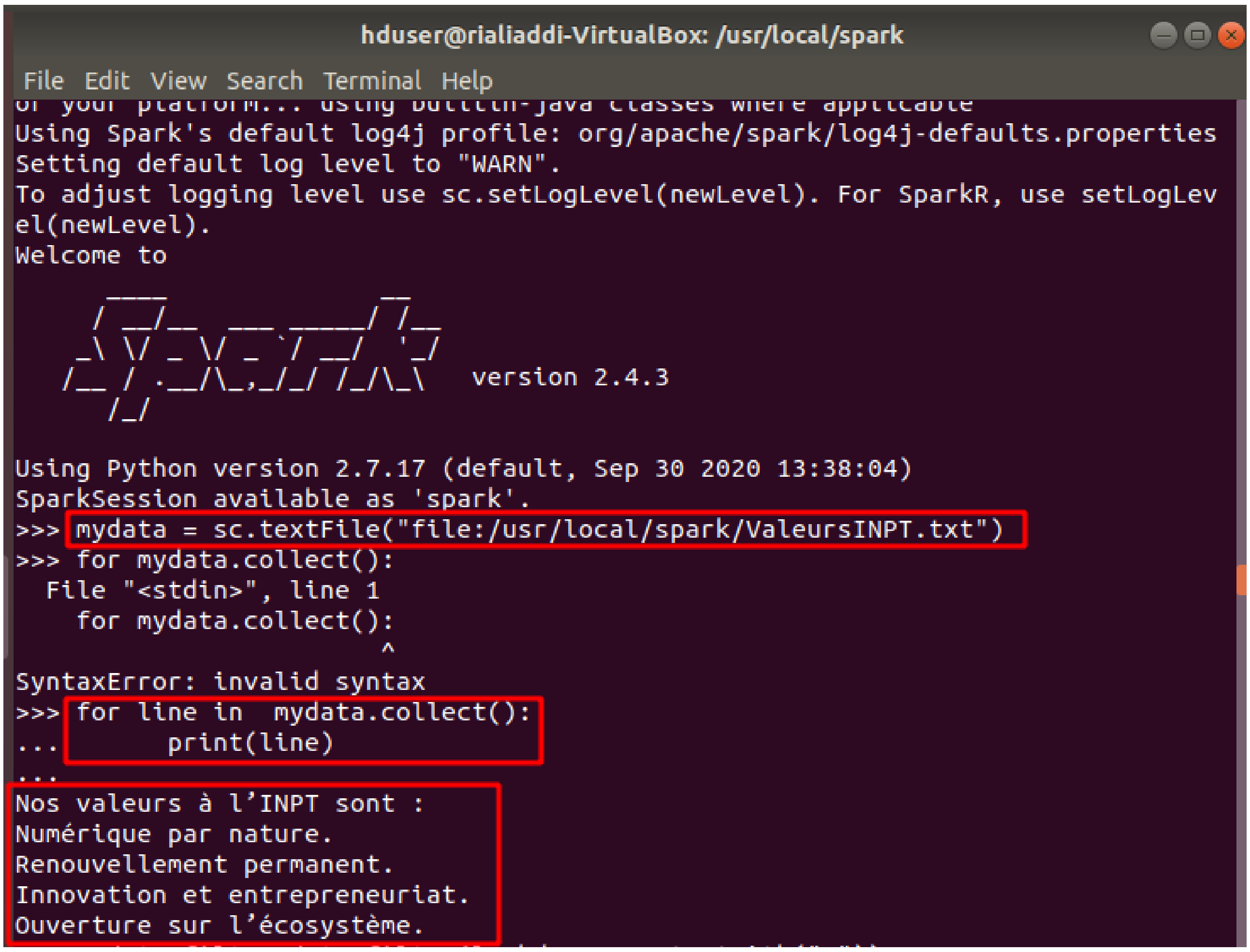
Créer un fichier : **ValeursINPT.txt** dans le répertoire : **/usr/local/spark** :



```
Nos valeurs à l'INPT sont :
Numérique par nature.
Renouvellement permanent.
Innovation et entrepreneuriat.
Ouverture sur l'écosystème.
```

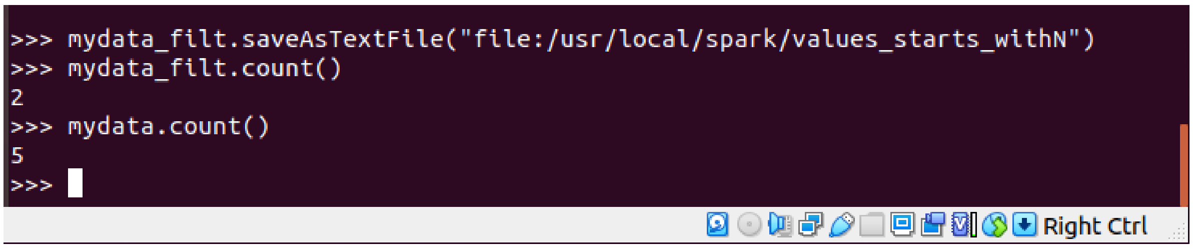
Accéder au terminal Python en tapant : $**./bin/pyspark** puis taper les commandes suivantes :

```
>>> mydata =
sc.textFile("file:/usr/local/spark/ValeursINPT.txt")
>>> for line in mydata.collect():
    ... print line

    >>> mydata.count()

    >>> mydata_filt = mydata.filter(lambda s:
s.startswith('N'))
>>>mydata_filt.saveAsTextFile("file:/usr/local/spark/va
lues_starts_withN")
```



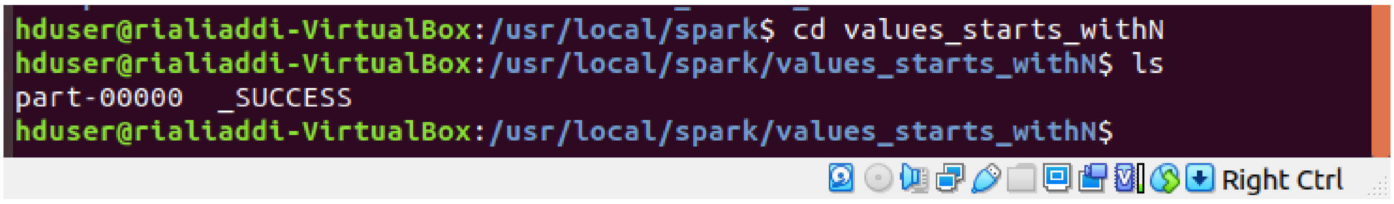
```
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLev
el(newLevel).
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 2.4.3
      /_/

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 mydata.collect():
  File "<stdin>", line 1
    for mydata.collect():
                        ^
SyntaxError: invalid syntax
>>> 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.
```

hduser@rialiaddi-VirtualBox: /usr/local/spark
File Edit View Search Terminal Help



```
>>> mydata_filt.saveAsTextFile("file:/usr/local/spark/values_starts_withN")
>>> mydata_filt.count()
2
>>> mydata.count()
5
>>> 
```

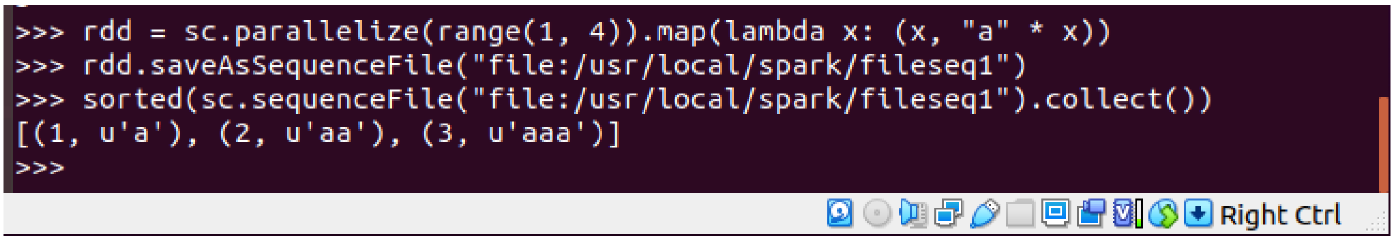
Vérifiez dans le répertoire /usr/local/spark qu'un répertoire nommé values_starts_withN est bien crée, ce répertoire doit contenir deux fichiers : part-00000 et _SUCCESS.



```
hduser@rialiaddi-VirtualBox:/usr/local/spark$ cd values_starts_withN
hduser@rialiaddi-VirtualBox:/usr/local/spark/values_starts_withN$ ls
part-00000  _SUCCESS
hduser@rialiaddi-VirtualBox:/usr/local/spark/values_starts_withN$
```

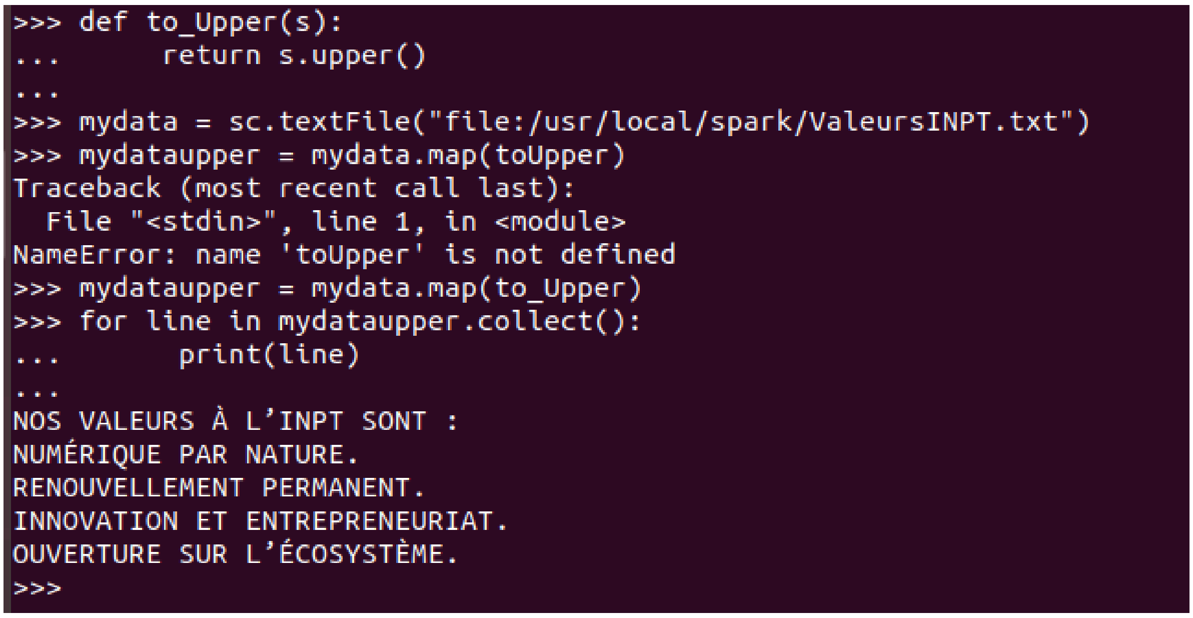
Enregistrement et chargement des SequenceFiles :

```
>>> 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())
```



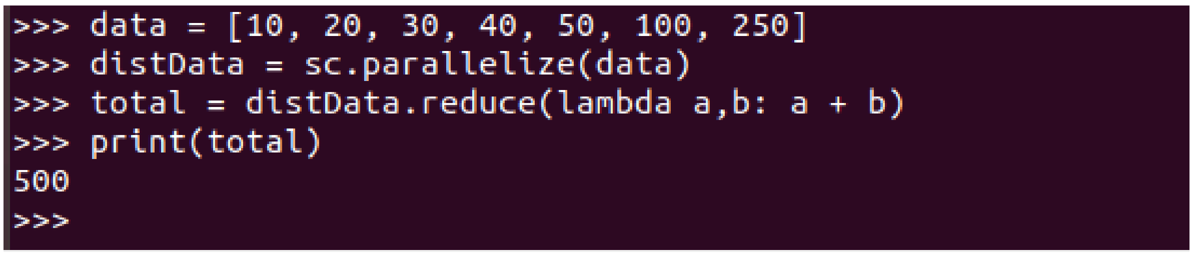
```
>>> rdd = sc.parallelize(range(1, 4)).map(lambda x: (x, "a" * x))
>>> rdd.saveAsSequenceFile("file:/usr/local/spark/fileseq1")
>>> sorted(sc.sequenceFile("file:/usr/local/spark/fileseq1").collect())
[(1, u'a'), (2, u'aa'), (3, u'aaa')]
>>>
```

Utilisation d'une fonction nommée :



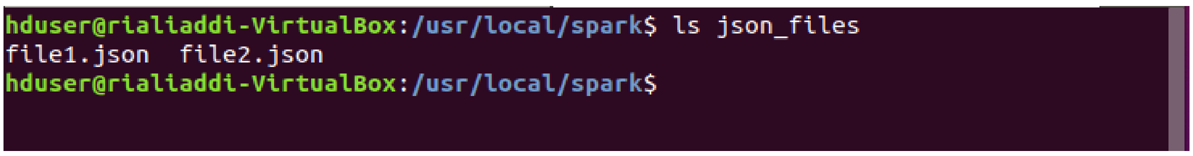
```
>>> def to_Upper(s):
...       return s.upper()
...
>>> mydata = sc.textFile("file:/usr/local/spark/ValeursINPT.txt")
>>> mydataupper = mydata.map(toUpper)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'toUpper' is not defined
>>> mydataupper = mydata.map(to_Upper)
>>> 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.
>>>
```

Utilisation de « parallelize » :

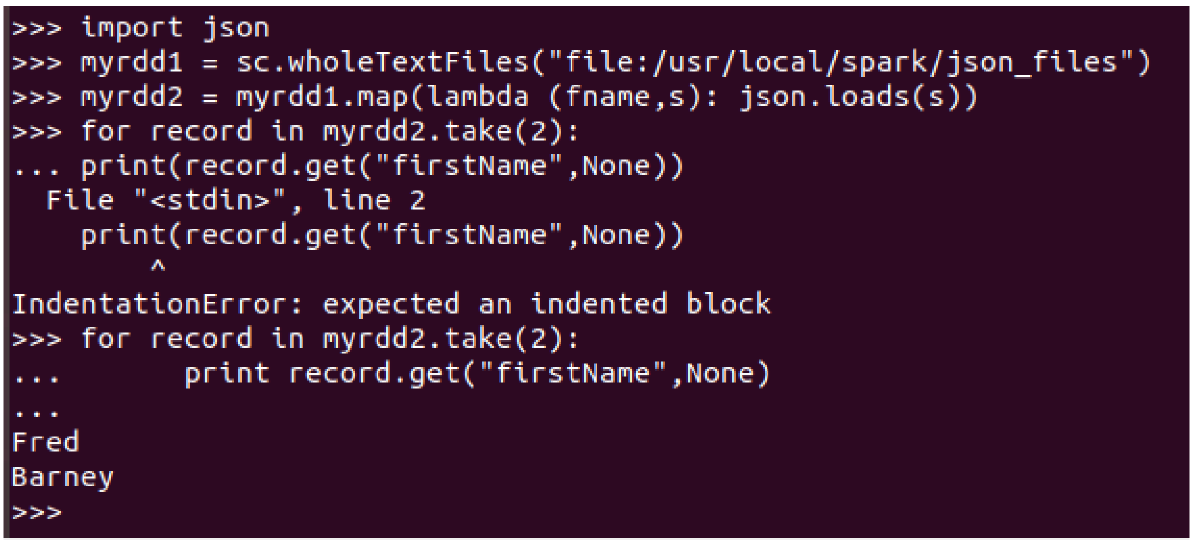


```
>>> data = [10, 20, 30, 40, 50, 100, 250]
>>> distData = sc.parallelize(data)
>>> total = distData.reduce(lambda a,b: a + b)
>>> print(total)
500
>>>
```

Utilisation de « wholeTextFiles » :

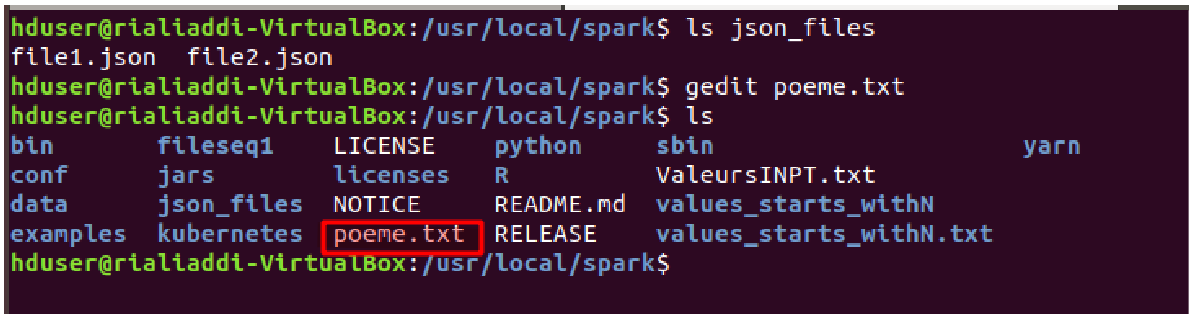


```
hduser@rialiaddi-VirtualBox:/usr/local/spark$ ls json_files
file1.json  file2.json
hduser@rialiaddi-VirtualBox:/usr/local/spark$
```

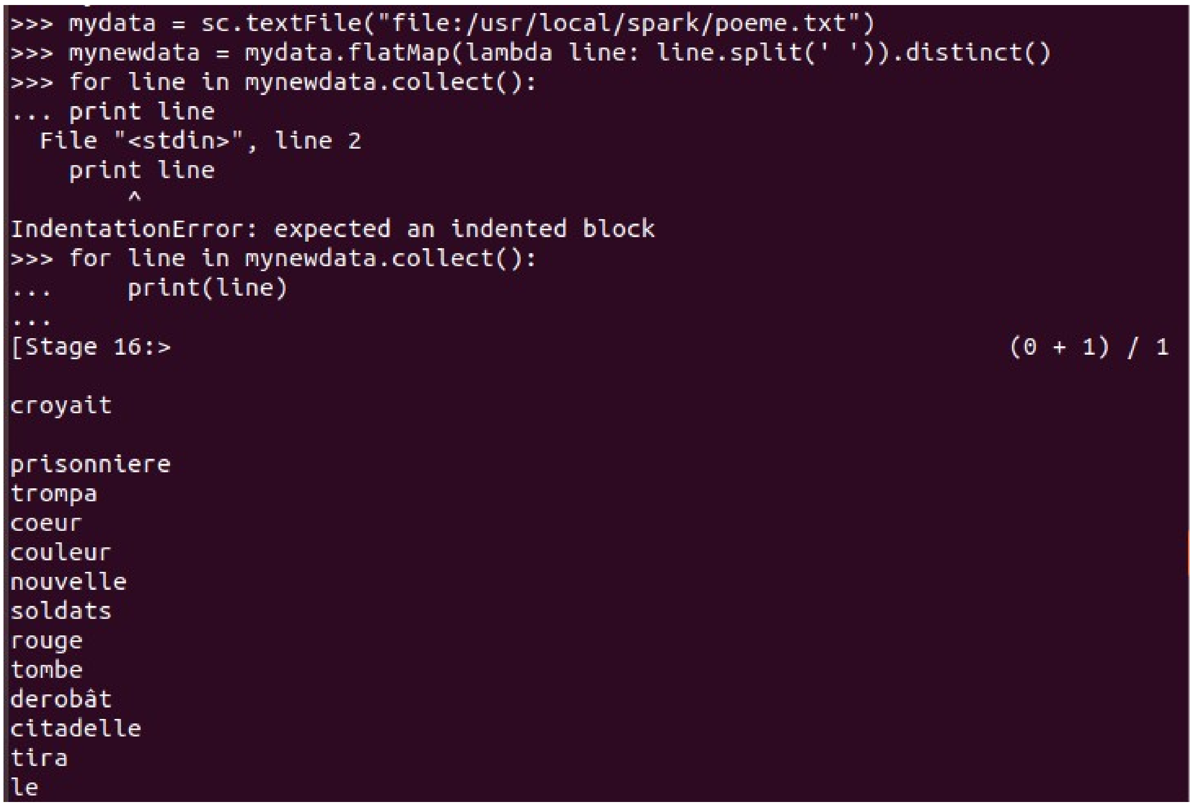


```
>>> import json
>>> myrdd1 = sc.wholeTextFiles("file:/usr/local/spark/json_files")
>>> myrdd2 = myrdd1.map(lambda (fname,s): json.loads(s))
>>> for record in myrdd2.take(2):
... print(record.get("firstName",None))
  File "<stdin>", line 2
    print(record.get("firstName",None))
        ^
IndentationError: expected an indented block
>>> for record in myrdd2.take(2):
...        print record.get("firstName",None)
...
Fred
Barney
>>>
```

Utilisation de « flatMap » et « distinct »:

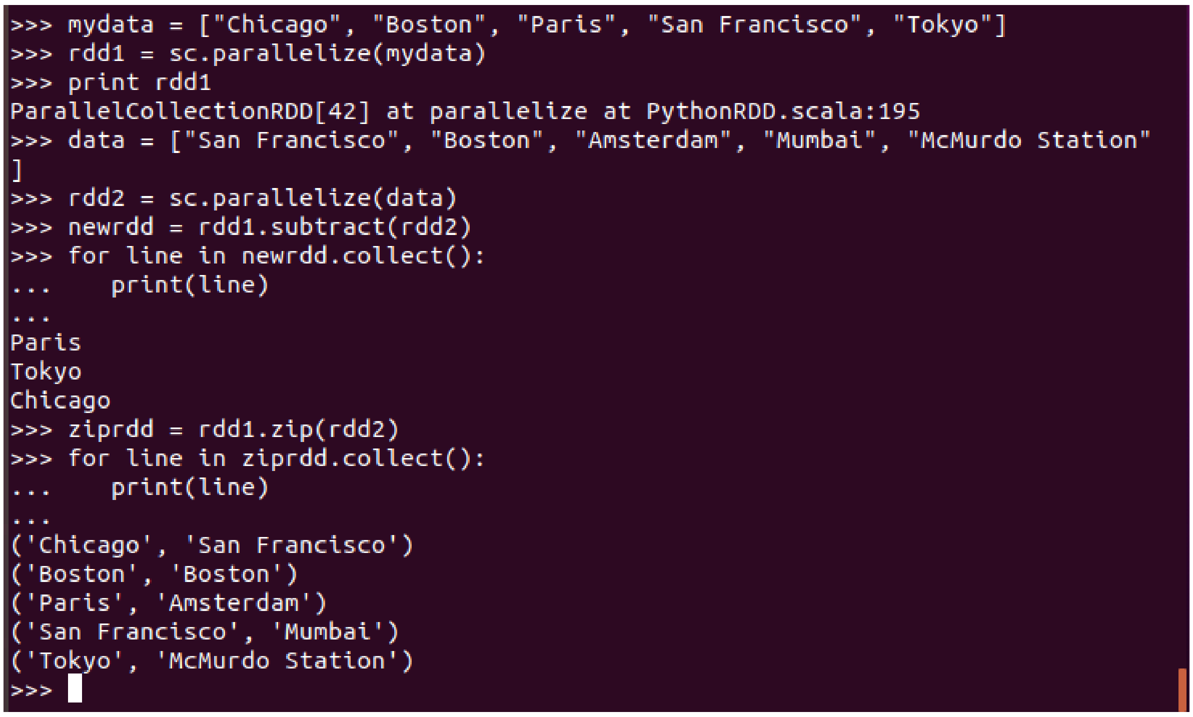


```
hduser@rialiaddi-VirtualBox:/usr/local/spark$ ls json_files
file1.json  file2.json
hduser@rialiaddi-VirtualBox:/usr/local/spark$ gedit poeme.txt
hduser@rialiaddi-VirtualBox:/usr/local/spark$ ls
bin        fileseq1    LICENSE    python     sbin                        yarn
conf       jars        licenses   R          ValeursINPT.txt
data       json_files  NOTICE     README.md  values_starts_withN
examples   kubernetes  poeme.txt  RELEASE    values_starts_withN.txt
hduser@rialiaddi-VirtualBox:/usr/local/spark$
```



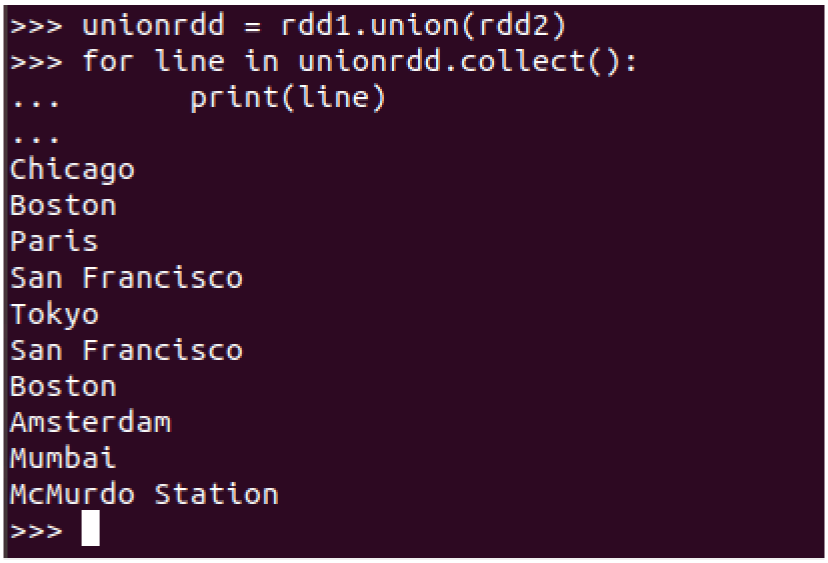
```
>>> mydata = sc.textFile("file:/usr/local/spark/poeme.txt")
>>> mynewdata = mydata.flatMap(lambda line: line.split(' ')).distinct()
>>> for line in mynewdata.collect():
... print line
  File "<stdin>", line 2
    print line
        ^
IndentationError: expected an indented block
>>> for line in mynewdata.collect():
...     print(line)
...
[Stage 16:>                                                          (0 + 1) / 1

croyait

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

Utilisation de « subtract » et « zip »:



```
>>> mydata = ["Chicago", "Boston", "Paris", "San Francisco", "Tokyo"]
>>> rdd1 = sc.parallelize(mydata)
>>> print rdd1
ParallelCollectionRDD[42] at parallelize at PythonRDD.scala:195
>>> 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')
>>>
```

Utilisation d'intersection et union :



```
>>> 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.

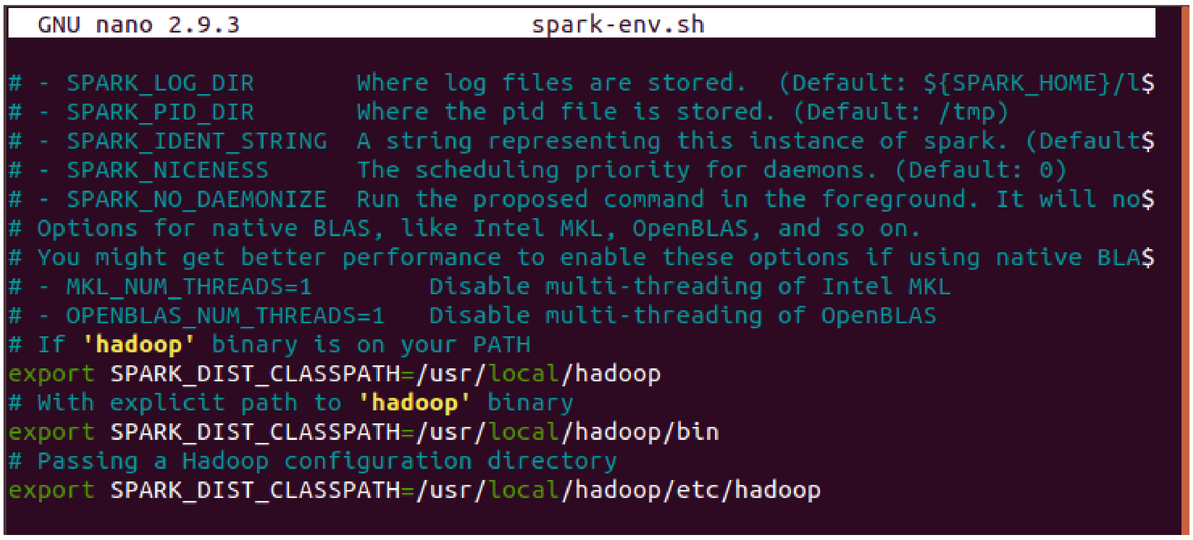
À partir de la version Spark 1.4, les packages de projet «Hadoop free » sont conçus pour vous permettre de

connecter plus facilement un seul fichier binaire Spark à n'importe quelle version de Hadoop.

Pour utiliser ces packages de Hadoop, vous devez modifier SPARK_DIST_CLASSPATH afin d'inclure les

Fichiers jar relatifs à ces packages. Pour ce faire, il est préférable d'ajouter une entrée dans conf / spark-env.sh :

```
:/usr/local/spark$ cd conf/

:/usr/local/spark/conf$

cp spark-env.sh.template spark-env.sh

:/usr/local/spark/conf$ sudo nano spark-env.sh
```



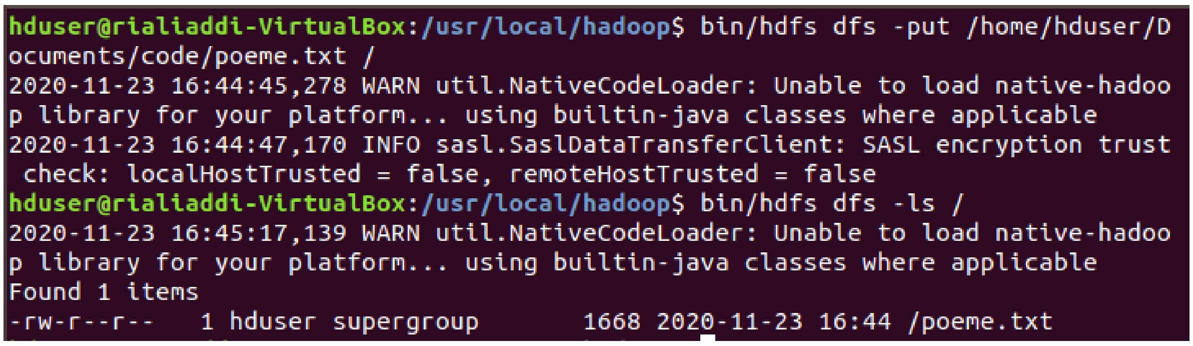
```
 GNU nano 2.9.3                                 spark-env.sh

# - SPARK_LOG_DIR       Where log files are stored.  (Default: ${SPARK_HOME}/l$
# - SPARK_PID_DIR       Where the pid file is stored. (Default: /tmp)
# - SPARK_IDENT_STRING  A string representing this instance of spark. (Default$
# - SPARK_NICENESS      The scheduling priority for daemons. (Default: 0)
# - SPARK_NO_DAEMONIZE  Run the proposed command in the foreground. It will no$
# Options for native BLAS, like Intel MKL, OpenBLAS, and so on.
# You might get better performance to enable these options if using native BLA$
# - MKL_NUM_THREADS=1        Disable multi-threading of Intel MKL
# - OPENBLAS_NUM_THREADS=1   Disable multi-threading of OpenBLAS
# 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
```

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

Une fois le nœud hadoop est bien configuré et spark bien connecté à hadoop. Nous pouvons déposer des données

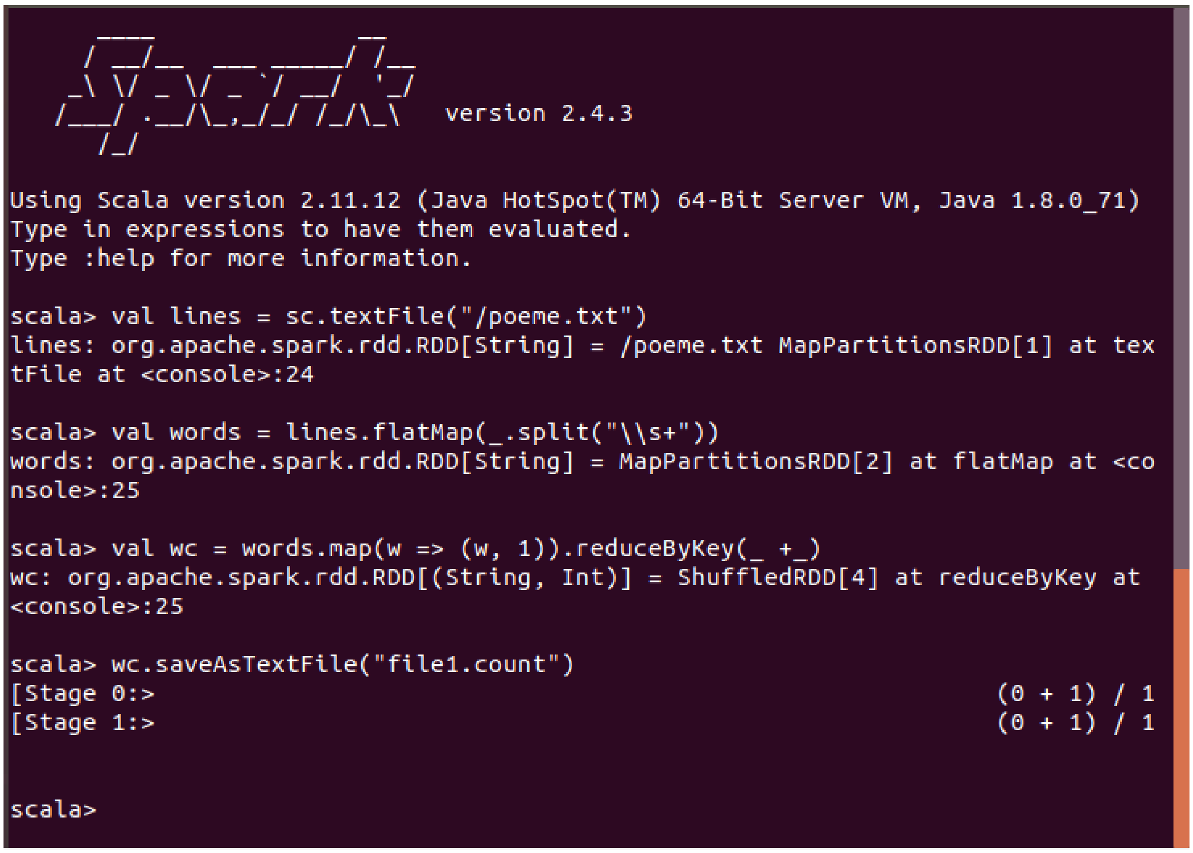
dans HDFS et les utiliser par spark.

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

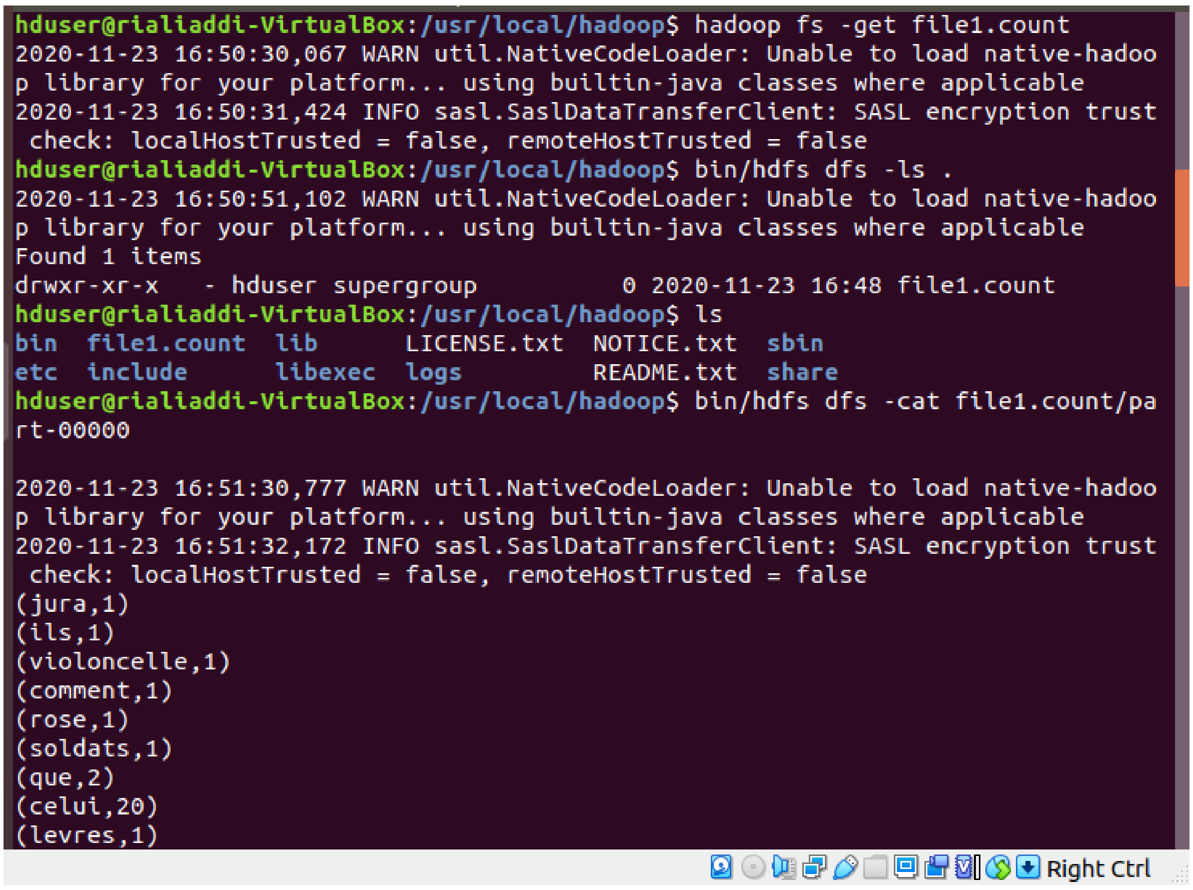


```
hduser@rialiaddi-VirtualBox:/usr/local/hadoop$ bin/hdfs dfs -put /home/hduser/D
ocuments/code/poeme.txt /
2020-11-23 16:44:45,278 WARN util.NativeCodeLoader: Unable to load native-hadoo
p library for your platform... using builtin-java classes where applicable
2020-11-23 16:44:47,170 INFO sasl.SaslDataTransferClient: SASL encryption trust
 check: localHostTrusted = false, remoteHostTrusted = false
hduser@rialiaddi-VirtualBox:/usr/local/hadoop$ bin/hdfs dfs -ls /
2020-11-23 16:45:17,139 WARN util.NativeCodeLoader: Unable to load native-hadoo
p library for your platform... using builtin-java classes where applicable
Found 1 items
-rw-r--r--   1 hduser supergroup       1668 2020-11-23 16:44 /poeme.txt
```

On va exécuter le traitement du « Word Count » en utilisant le terminal spark-shell :

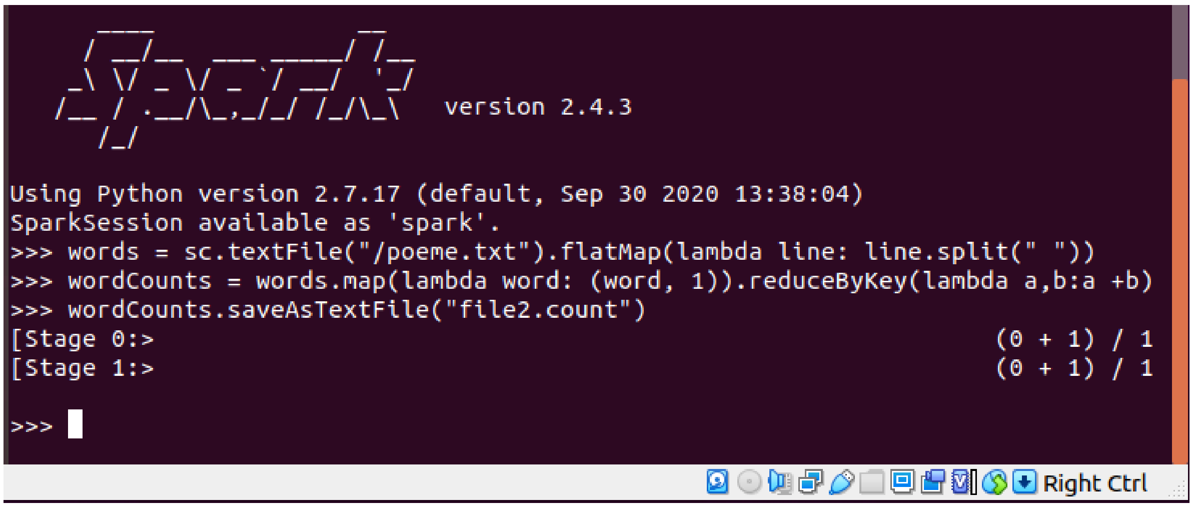


```
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 2.4.3
      /_/

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 tex
tFile at <console>:24

scala> val words = lines.flatMap(_.split("\\s+"))
words: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[2] at flatMap at <co
nsole>: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")
[Stage 0:>                                                          (0 + 1) / 1
[Stage 1:>                                                          (0 + 1) / 1


scala>
```



```
hduser@rialiaddi-VirtualBox:/usr/local/hadoop$ hadoop fs -get file1.count
2020-11-23 16:50:30,067 WARN util.NativeCodeLoader: Unable to load native-hadoo
p library for your platform... using builtin-java classes where applicable
2020-11-23 16:50:31,424 INFO sasl.SaslDataTransferClient: SASL encryption trust
 check: localHostTrusted = false, remoteHostTrusted = false
hduser@rialiaddi-VirtualBox:/usr/local/hadoop$ bin/hdfs dfs -ls .
2020-11-23 16:50:51,102 WARN util.NativeCodeLoader: Unable to load native-hadoo
p library for your platform... using builtin-java classes where applicable
Found 1 items
drwxr-xr-x   - hduser supergroup          0 2020-11-23 16:48 file1.count
hduser@rialiaddi-VirtualBox:/usr/local/hadoop$ ls
bin  file1.count  lib      LICENSE.txt  NOTICE.txt  sbin
etc  include      libexec  logs         README.txt  share
hduser@rialiaddi-VirtualBox:/usr/local/hadoop$ bin/hdfs dfs -cat file1.count/pa
rt-00000

2020-11-23 16:51:30,777 WARN util.NativeCodeLoader: Unable to load native-hadoo
p library for your platform... using builtin-java classes where applicable
2020-11-23 16:51:32,172 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)
```

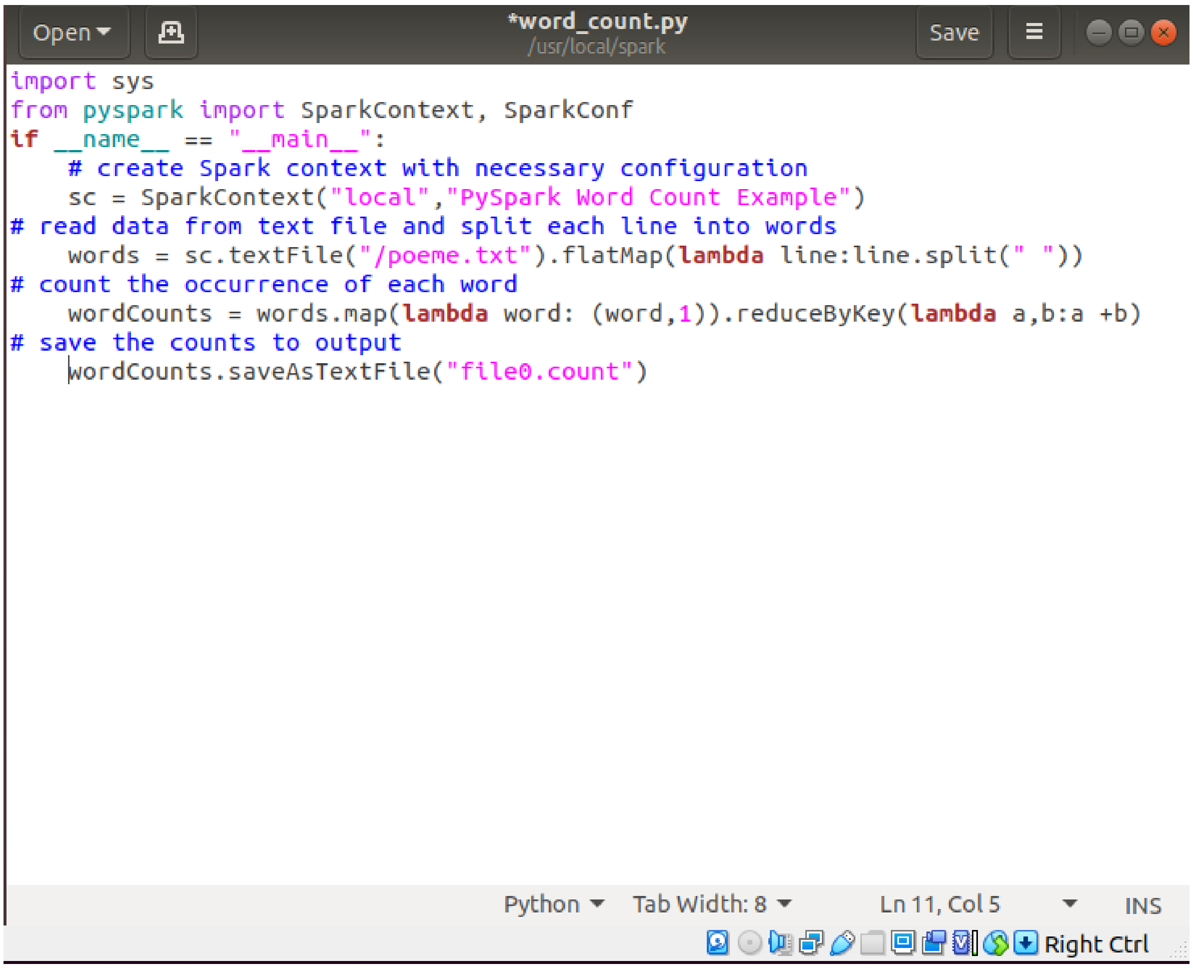
Maintenant on va exécuter le traitement du « Word Count » en utilisant le terminal **Pyspark :**



```
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   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")
[Stage 0:>                                                          (0 + 1) / 1
[Stage 1:>                                                          (0 + 1) / 1

>>> 
```

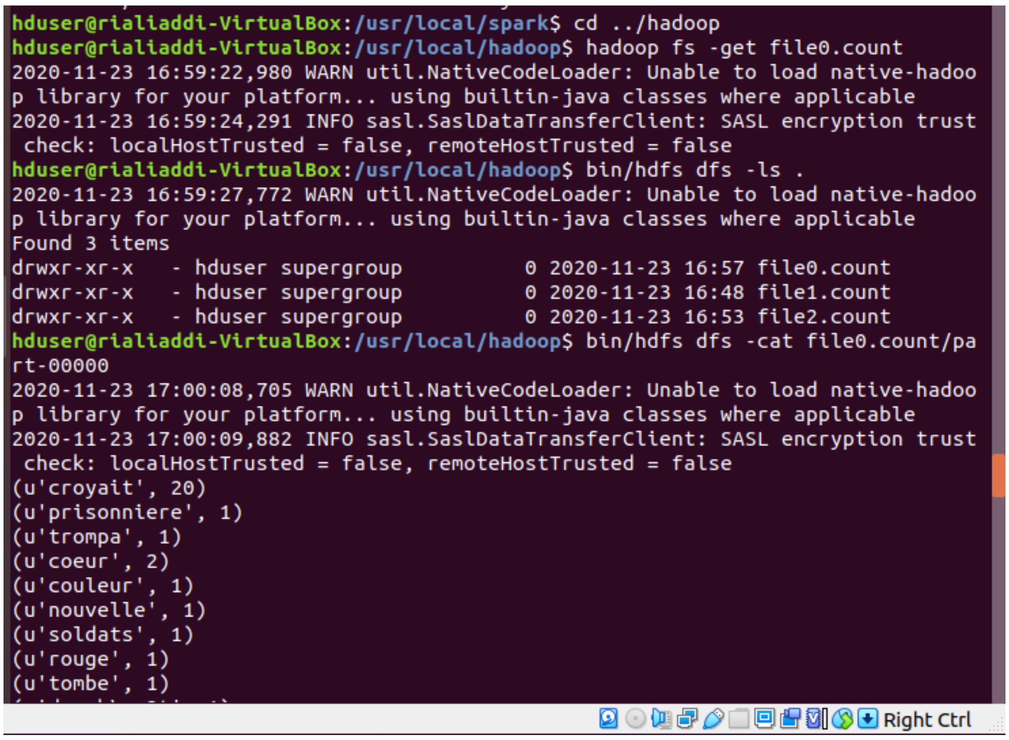
VI. Exécution du « Word Count » en utilisant un script python :

On va créer un fichier « **word_count.py** »



```
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")
```

Ensuite:



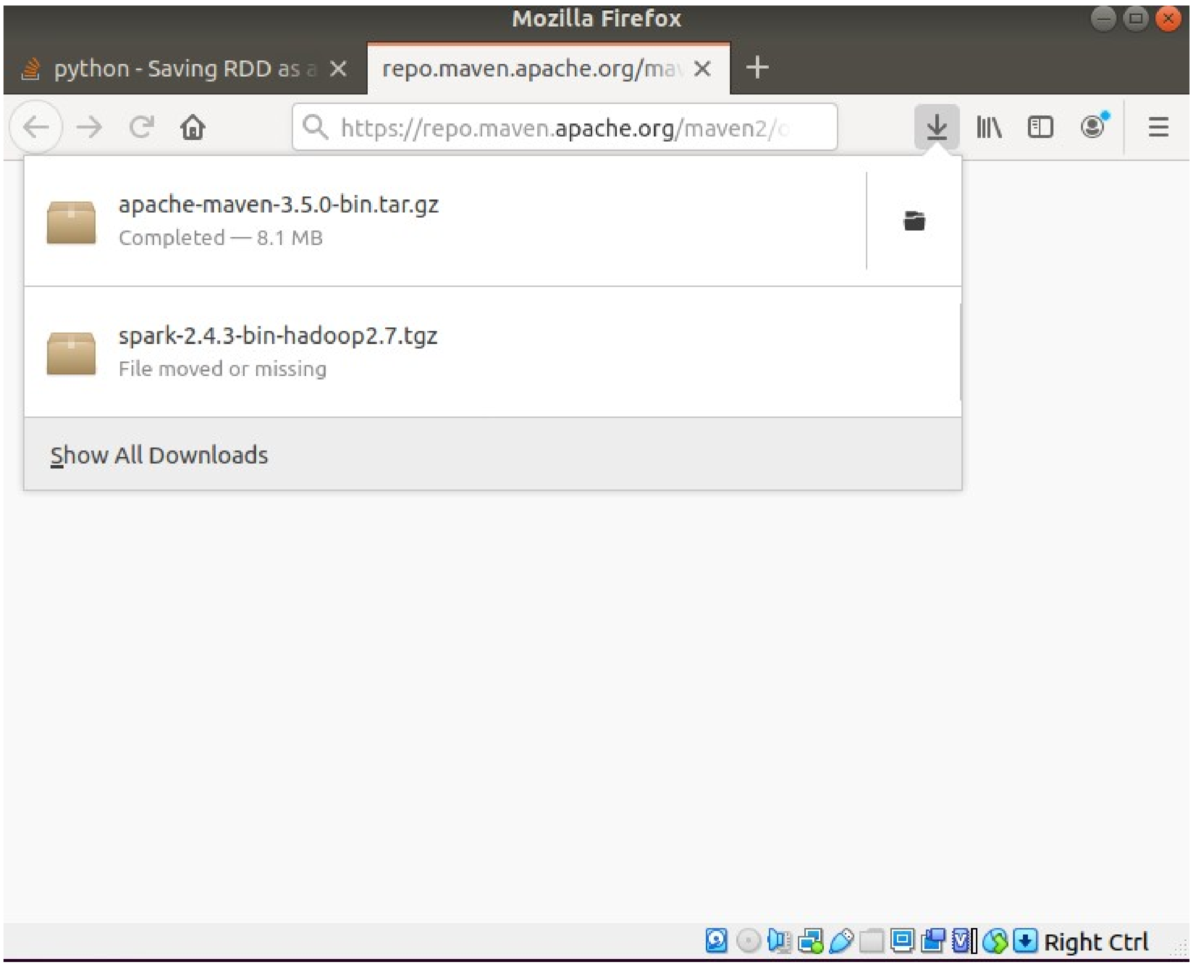
```
hduser@rialiaddi-VirtualBox:/usr/local/spark$ cd ../hadoop
hduser@rialiaddi-VirtualBox:/usr/local/hadoop$ hadoop fs -get file0.count
2020-11-23 16:59:22,980 WARN util.NativeCodeLoader: Unable to load native-hadoo
p library for your platform... using builtin-java classes where applicable
2020-11-23 16:59:24,291 INFO sasl.SaslDataTransferClient: SASL encryption trust
 check: localHostTrusted = false, remoteHostTrusted = false
hduser@rialiaddi-VirtualBox:/usr/local/hadoop$ bin/hdfs dfs -ls .
2020-11-23 16:59:27,772 WARN util.NativeCodeLoader: Unable to load native-hadoo
p library for your platform... using builtin-java classes where applicable
Found 3 items
drwxr-xr-x   - hduser supergroup          0 2020-11-23 16:57 file0.count
drwxr-xr-x   - hduser supergroup          0 2020-11-23 16:48 file1.count
drwxr-xr-x   - hduser supergroup          0 2020-11-23 16:53 file2.count
hduser@rialiaddi-VirtualBox:/usr/local/hadoop$ bin/hdfs dfs -cat file0.count/pa
rt-00000
2020-11-23 17:00:08,705 WARN util.NativeCodeLoader: Unable to load native-hadoo
p library for your platform... using builtin-java classes where applicable
2020-11-23 17:00:09,882 INFO sasl.SaslDataTransferClient: SASL encryption trust
 check: localHostTrusted = false, remoteHostTrusted = false
(u'croyait', 20)
(u'prisonniere', 1)
(u'trompa', 1)
(u'coeur', 2)
(u'couleur', 1)
(u'nouvelle', 1)
(u'soldats', 1)
(u'rouge', 1)
(u'tombe', 1)
```

VII. 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.

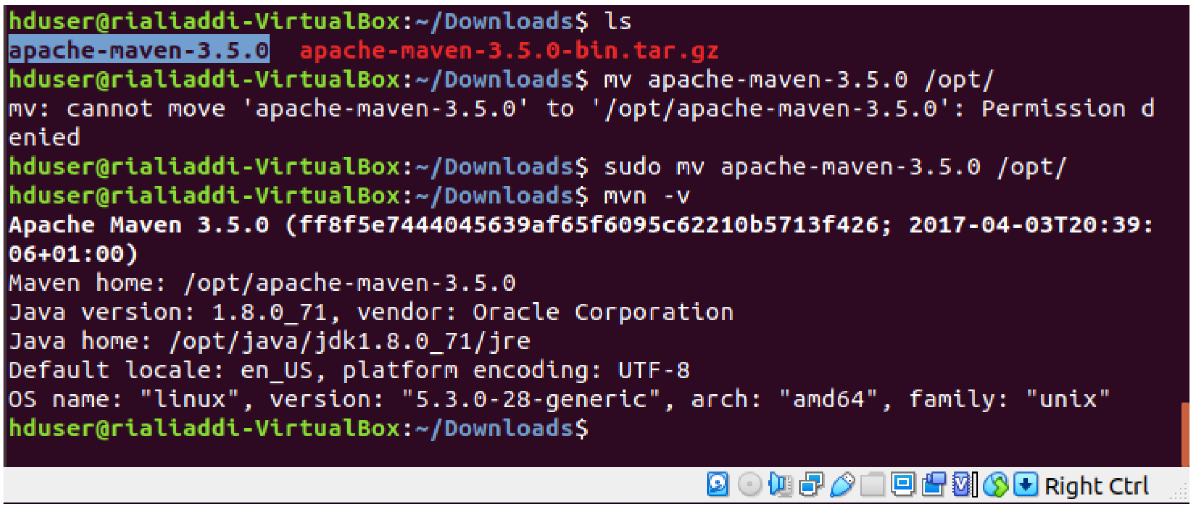
Installation d’Apache Maven:

Maven™



apache-maven-3.5.0-bin.tar.gz
Completed — 8.1 MB

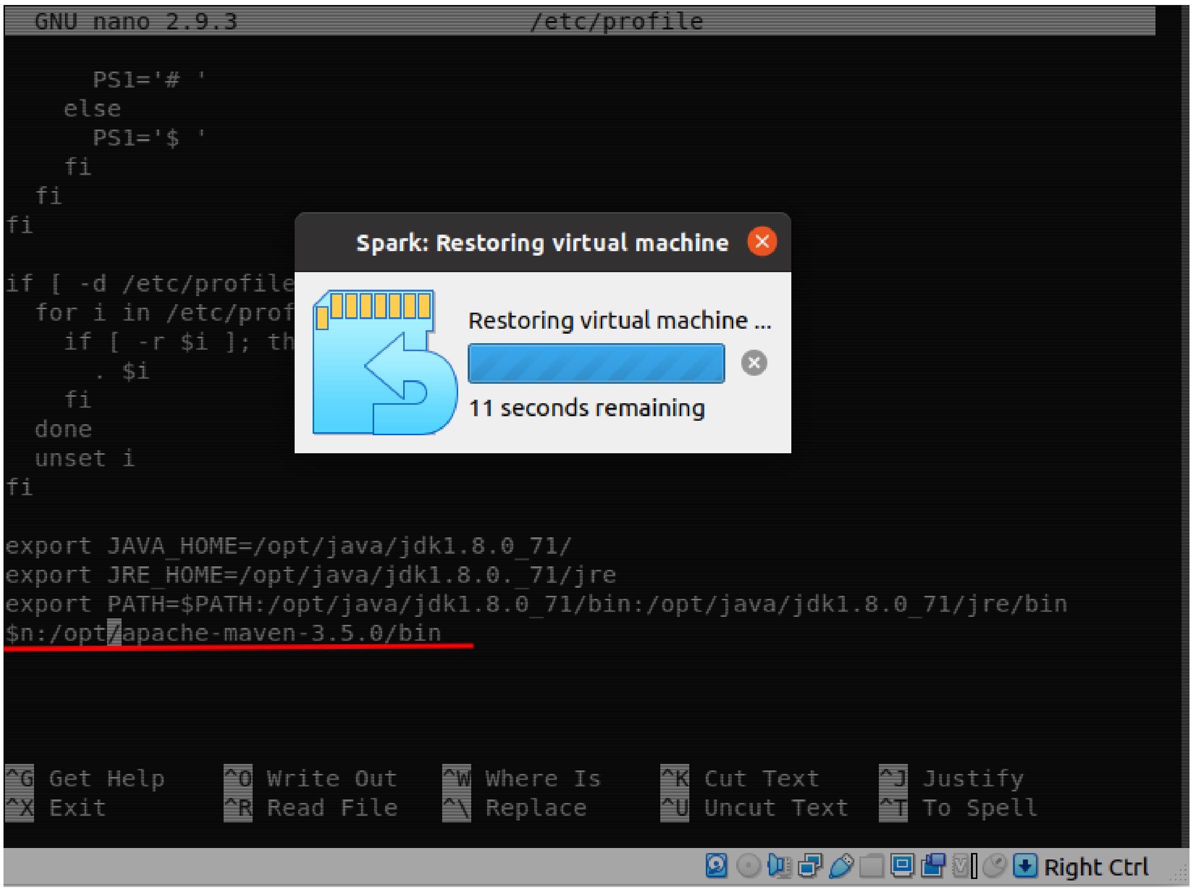
spark-2.4.3-bin-hadoop2.7.tgz
File moved or missing



```
hduser@rialiaddi-VirtualBox:~/Downloads$ ls
apache-maven-3.5.0  apache-maven-3.5.0-bin.tar.gz
hduser@rialiaddi-VirtualBox:~/Downloads$ mv apache-maven-3.5.0 /opt/
mv: cannot move 'apache-maven-3.5.0' to '/opt/apache-maven-3.5.0': Permission d
enied
hduser@rialiaddi-VirtualBox:~/Downloads$ sudo mv apache-maven-3.5.0 /opt/
hduser@rialiaddi-VirtualBox:~/Downloads$ 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@rialiaddi-VirtualBox:~/Downloads$
```

Pour mettre en place de manière permanente la variable d'environnement PATH pour tous les utilisateurs :

il faut ouvrir le fichier /etc/profile et modifiez le PATH en ajoutant le chemin où se trouve le bin de maven dans export PATH



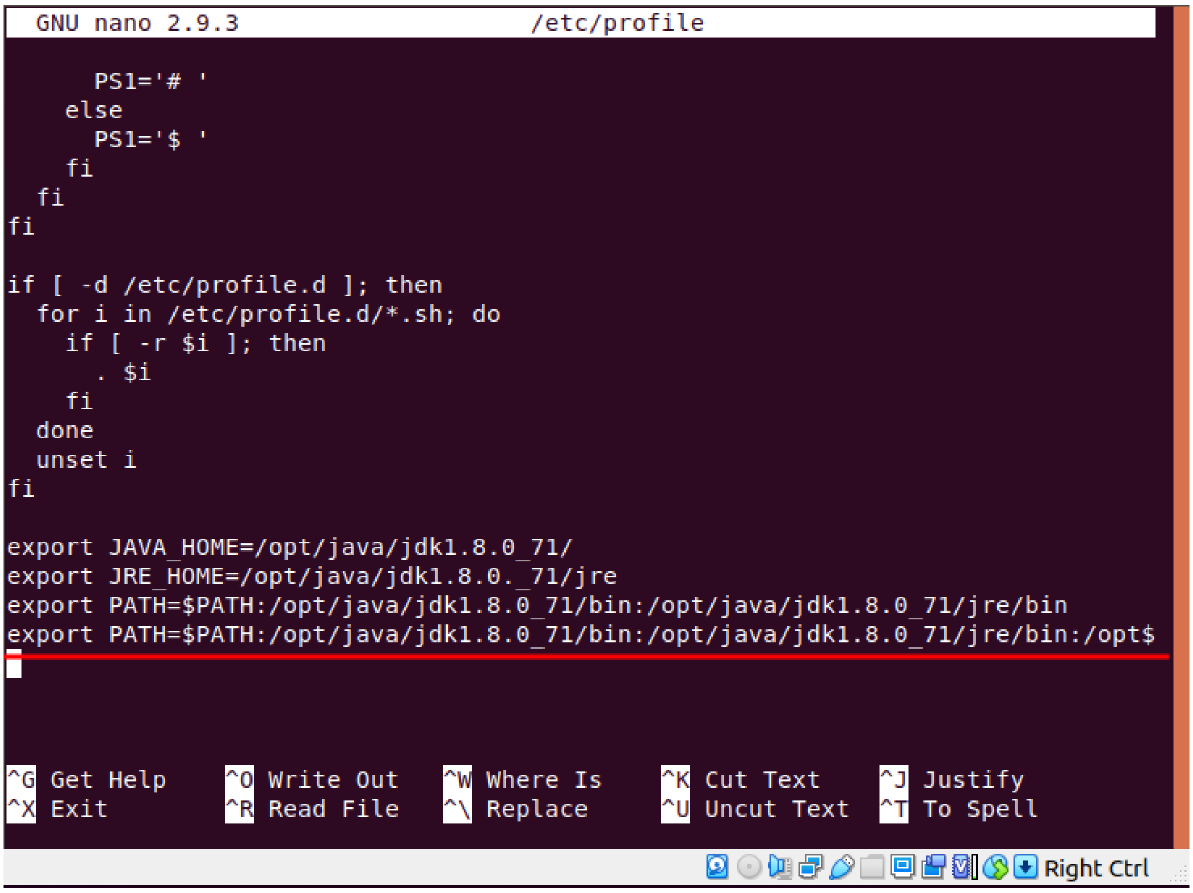
```
  GNU nano 2.9.3                      /etc/profile

      PS1='# '
    else
      PS1='$ '
    fi
  fi
fi

if [ -d /etc/profile
  for i in /etc/prof
    if [ -r $i ]; th
      . $i
    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
$n:/opt/apache-maven-3.5.0/bin



^G Get Help    ^O Write Out   ^W Where Is    ^K Cut Text    ^J Justify
^X Exit        ^R Read File   ^\ Replace     ^U Uncut Text  ^T To Spell
```

Spark: Restoring virtual machine

Restoring virtual machine ...

11 seconds remaining

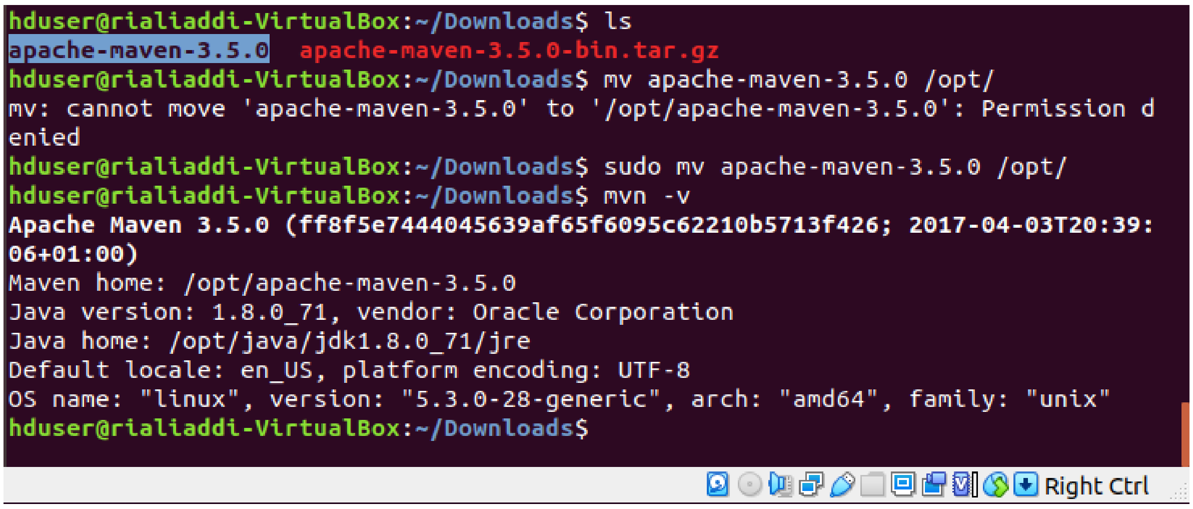
Right Ctrl



```
  GNU nano 2.9.3                      /etc/profile

      PS1='# '
    else
      PS1='$ '
    fi
  fi
fi

if [ -d /etc/profile.d ]; then
  for i in /etc/profile.d/*.sh; do
    if [ -r $i ]; then
      . $i
    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$



^G Get Help    ^O Write Out   ^W Where Is    ^K Cut Text    ^J Justify
^X Exit        ^R Read File   ^\ Replace     ^U Uncut Text  ^T To Spell
```

Right Ctrl

Après avoir enregistré le fichier profile, on va exécuter la commande source pour recharger le fichier dans la session hduser, par la suite on testera la configuration de maven en tapant:

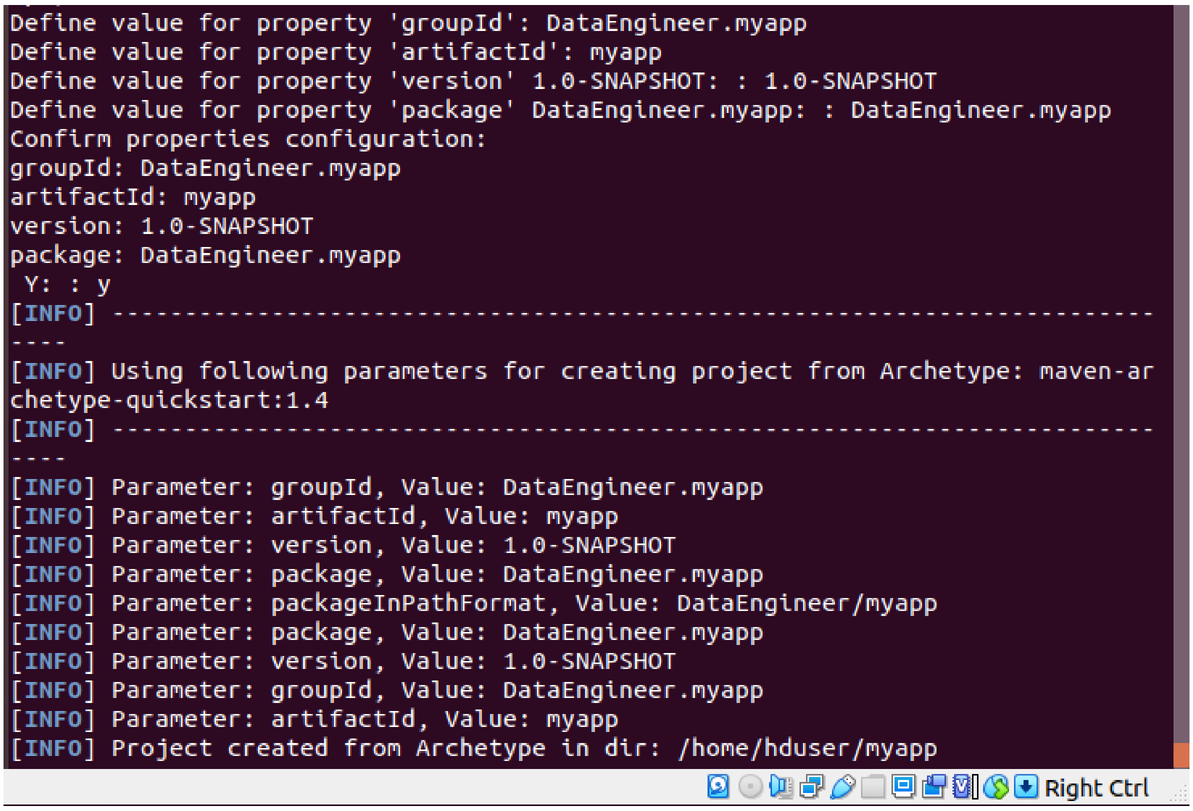


```
hduser@rialiaddi-VirtualBox:~/Downloads$ ls
apache-maven-3.5.0  apache-maven-3.5.0-bin.tar.gz
hduser@rialiaddi-VirtualBox:~/Downloads$ mv apache-maven-3.5.0 /opt/
mv: cannot move 'apache-maven-3.5.0' to '/opt/apache-maven-3.5.0': Permission d
enied
hduser@rialiaddi-VirtualBox:~/Downloads$ sudo mv apache-maven-3.5.0 /opt/
hduser@rialiaddi-VirtualBox:~/Downloads$ 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@rialiaddi-VirtualBox:~/Downloads$
```

Dans cette étape, on va créer un projet **Maven** :

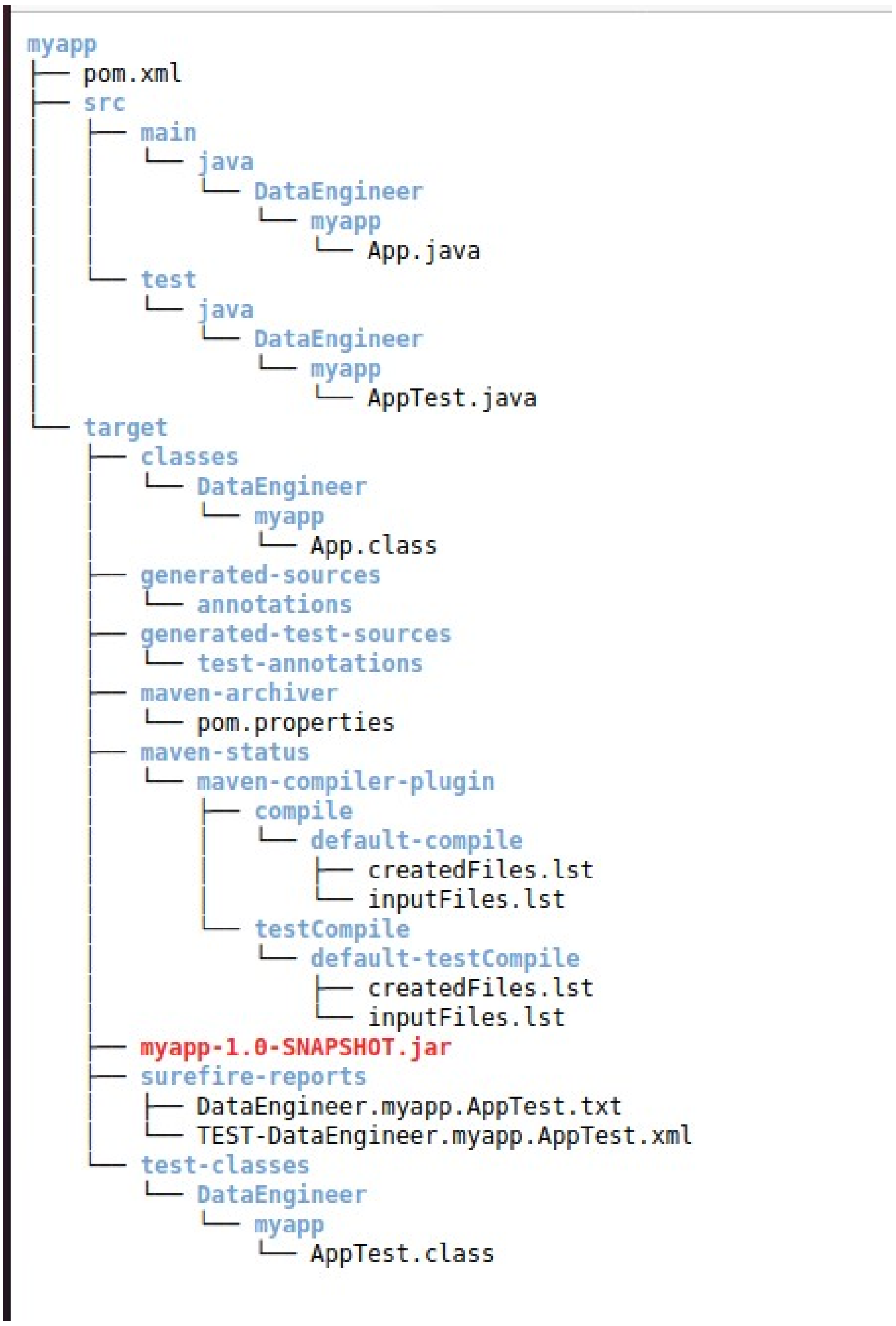
mvn archetype:generate -DarchetypeArtifactId=maven-archetype-quickstart – DarchetypeVersion=1.1

et on va choisir les paramètres suivants:



```
Define value for property 'groupId': DataEngineer.myapp
Define value for property 'artifactId': myapp
Define value for property 'version' 1.0-SNAPSHOT: : 1.0-SNAPSHOT
Define value for property 'package' DataEngineer.myapp: : DataEngineer.myapp
Confirm properties configuration:
groupId: DataEngineer.myapp
artifactId: myapp
version: 1.0-SNAPSHOT
package: DataEngineer.myapp
 Y: : y
[INFO] ----------------------------------------------------------------------
----
[INFO] Using following parameters for creating project from Archetype: maven-ar
chetype-quickstart:1.4
[INFO] ----------------------------------------------------------------------
----
[INFO] Parameter: groupId, Value: DataEngineer.myapp
[INFO] Parameter: artifactId, Value: myapp
[INFO] Parameter: version, Value: 1.0-SNAPSHOT
[INFO] Parameter: package, Value: DataEngineer.myapp
[INFO] Parameter: packageInPathFormat, Value: DataEngineer/myapp
[INFO] Parameter: package, Value: DataEngineer.myapp
[INFO] Parameter: version, Value: 1.0-SNAPSHOT
[INFO] Parameter: groupId, Value: DataEngineer.myapp
[INFO] Parameter: artifactId, Value: myapp
[INFO] Project created from Archetype in dir: /home/hduser/myapp
```

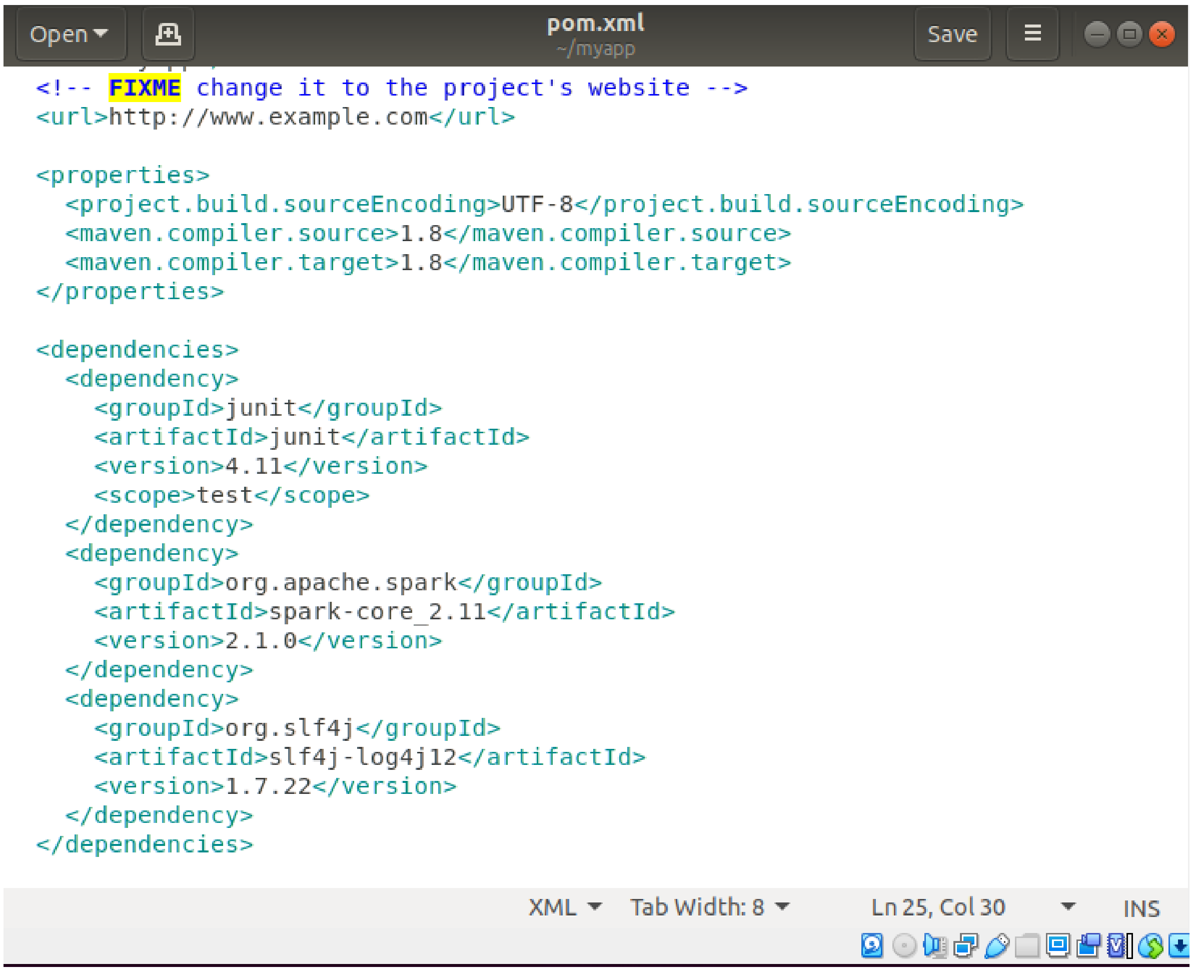
tree myapp/:



```
myapp
├── pom.xml
├── src
│   ├── main
│   │   └── java
│   │       └── DataEngineer
│   │           └── myapp
│   │               └── App.java
│   └── test
│       └── java
│           └── DataEngineer
│               └── myapp
│                   └── AppTest.java
└── target
    ├── classes
    │   └── DataEngineer
    │       └── myapp
    │           └── App.class
    ├── generated-sources
    │   └── annotations
    ├── generated-test-sources
    │   └── test-annotations
    ├── 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
```

Reconfiguration du projet Maven :

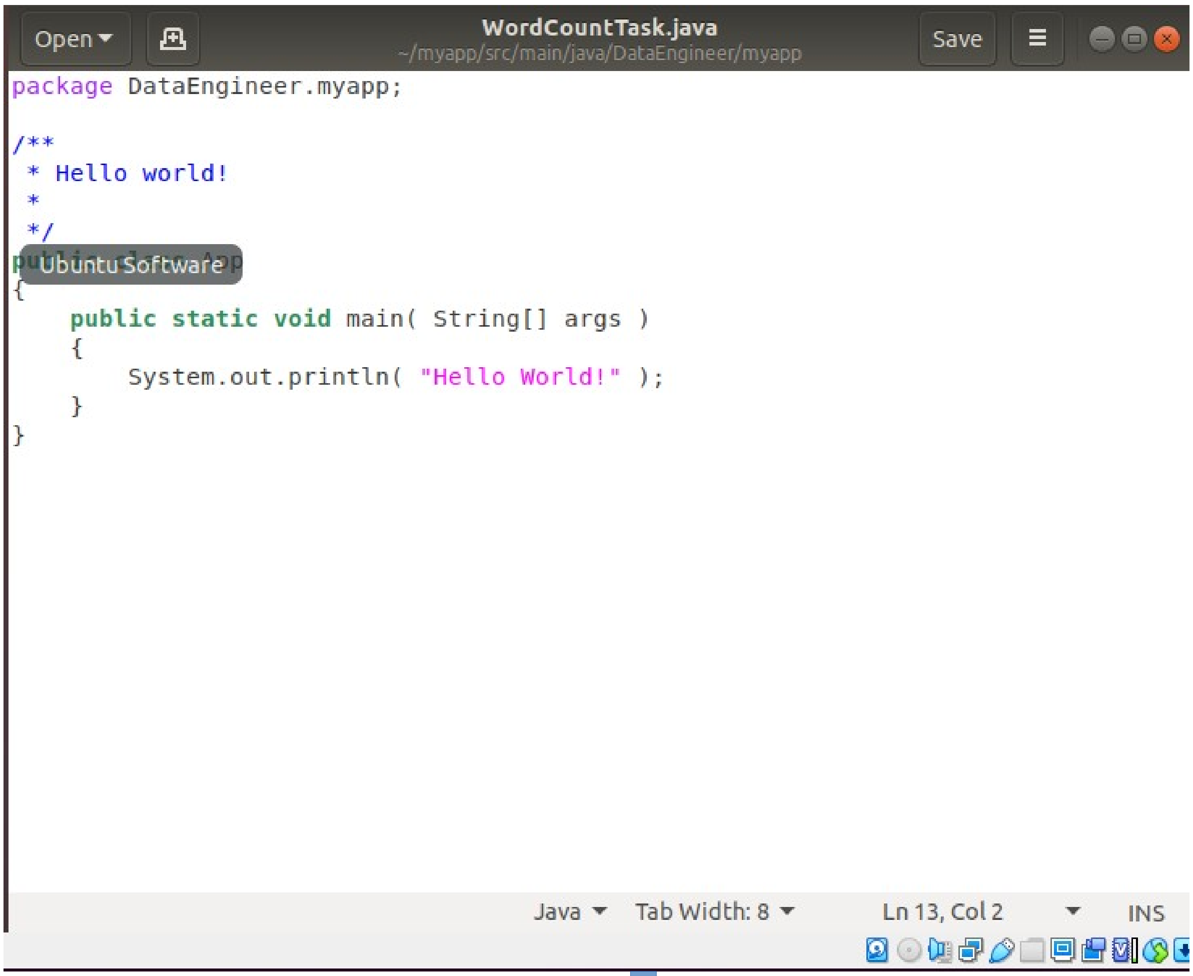
Rajouter dans le fichier **pom.xml** les dépendances nécessaires, pour cela:



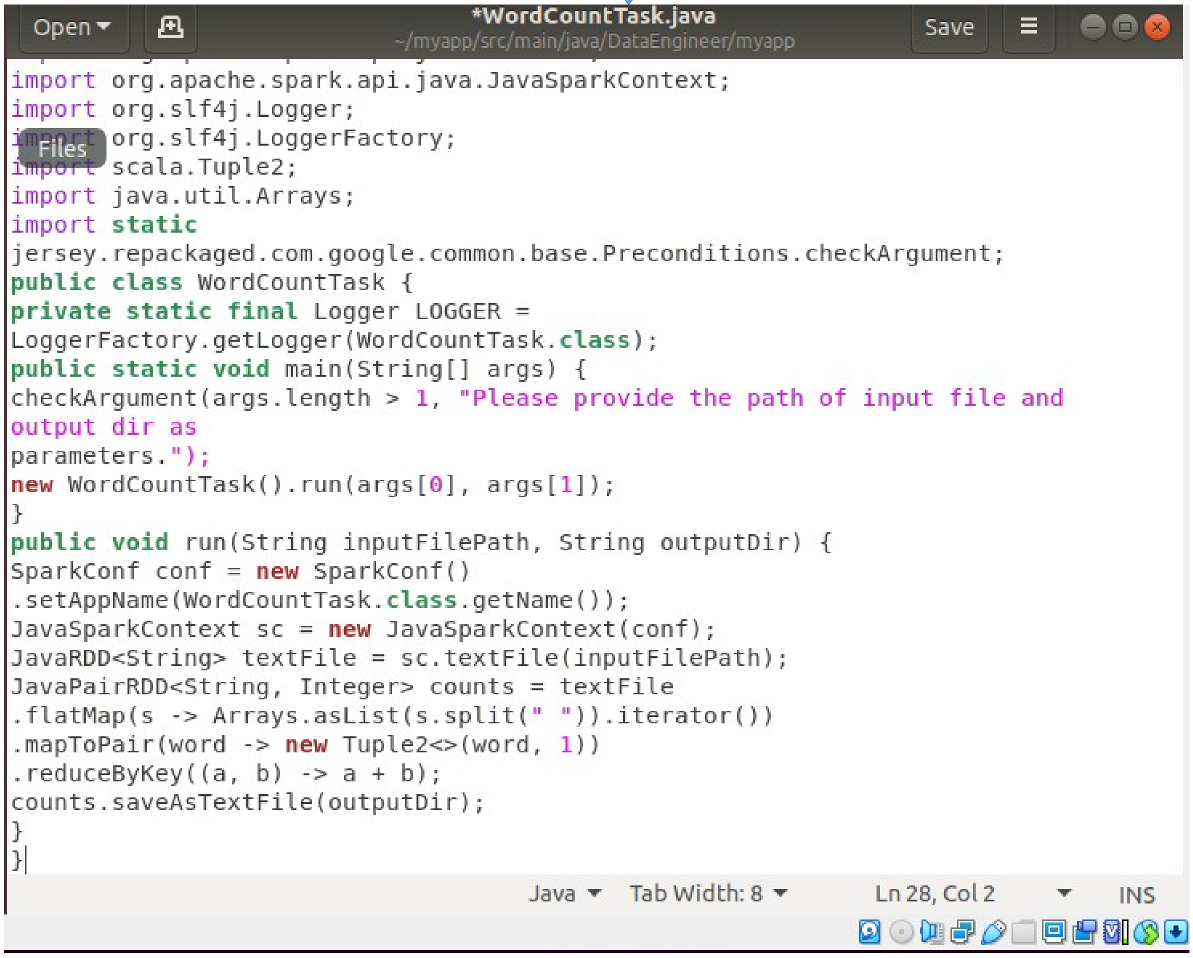
```
<!-- FIXME change it to the project's website -->
<url>http://www.example.com</url>

<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>4.11</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>
```

Dans le répertoire: /home/hduser/myapp/src/main/java/DataEngineer/myapp :

On renomme **App.java** en **WordCountTask.java** en changeant son contenu tel qu'on arrivera a ce résultat:

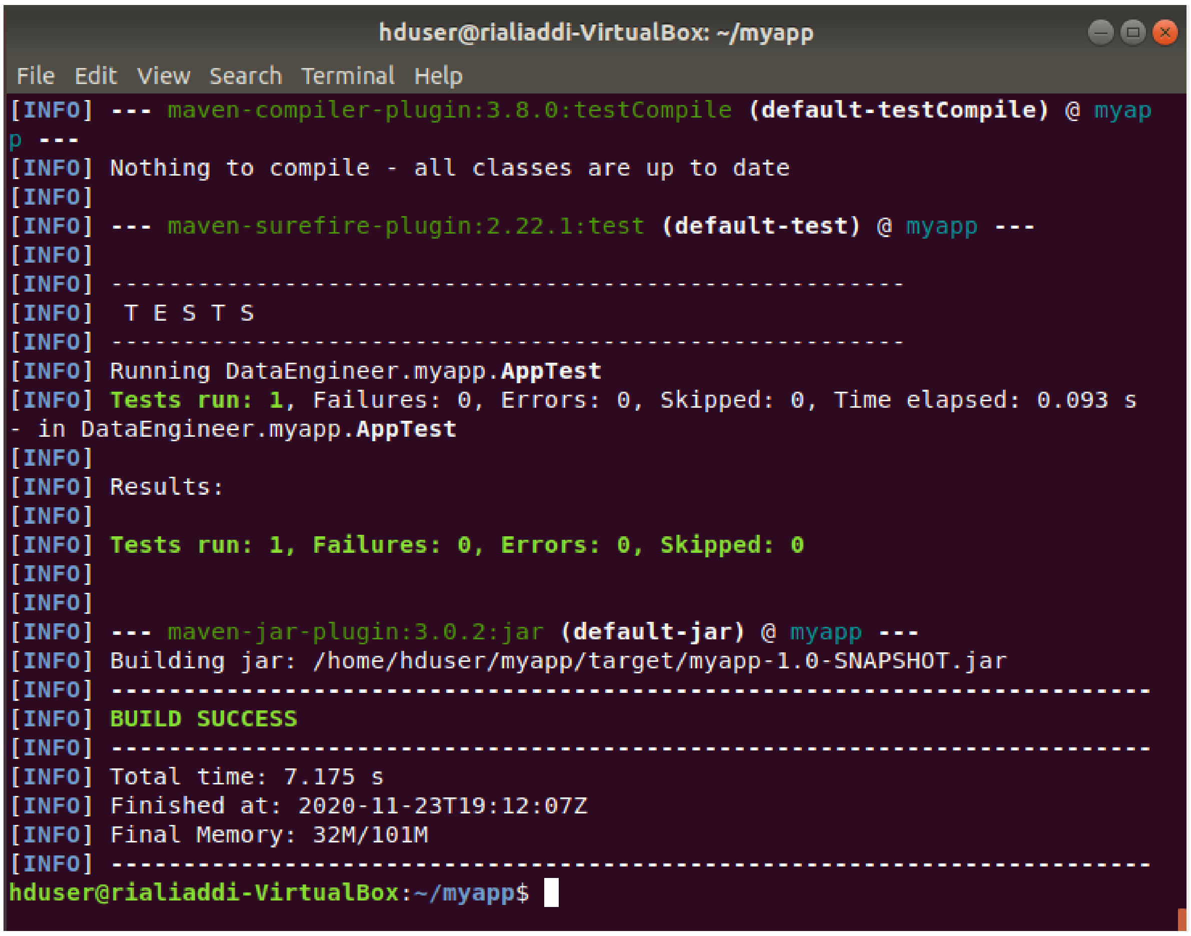


```java
package DataEngineer.myapp;

/**
 * Hello world!
 *
 */
public class App
{
    public static void main( String[] args )
    {
        System.out.println( "Hello World!" );
    }
}
```



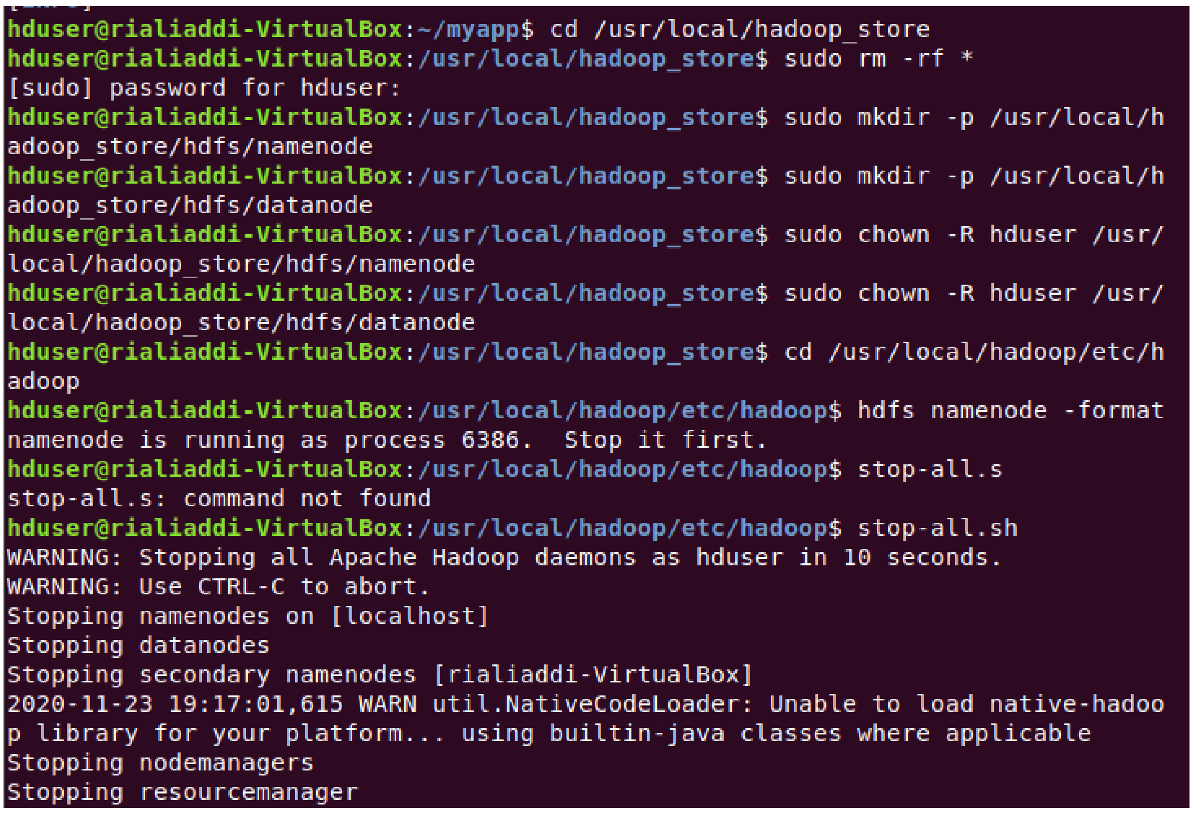
```java
import org.apache.spark.api.java.JavaSparkContext;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import scala.Tuple2;
import java.util.Arrays;
import static
jersey.repackaged.com.google.common.base.Preconditions.checkArgument;
public class WordCountTask {
private static final Logger 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);
}
}
```

Enfin, Enregistrons WordCountTask.java, et lançons la commande : **mvn package**

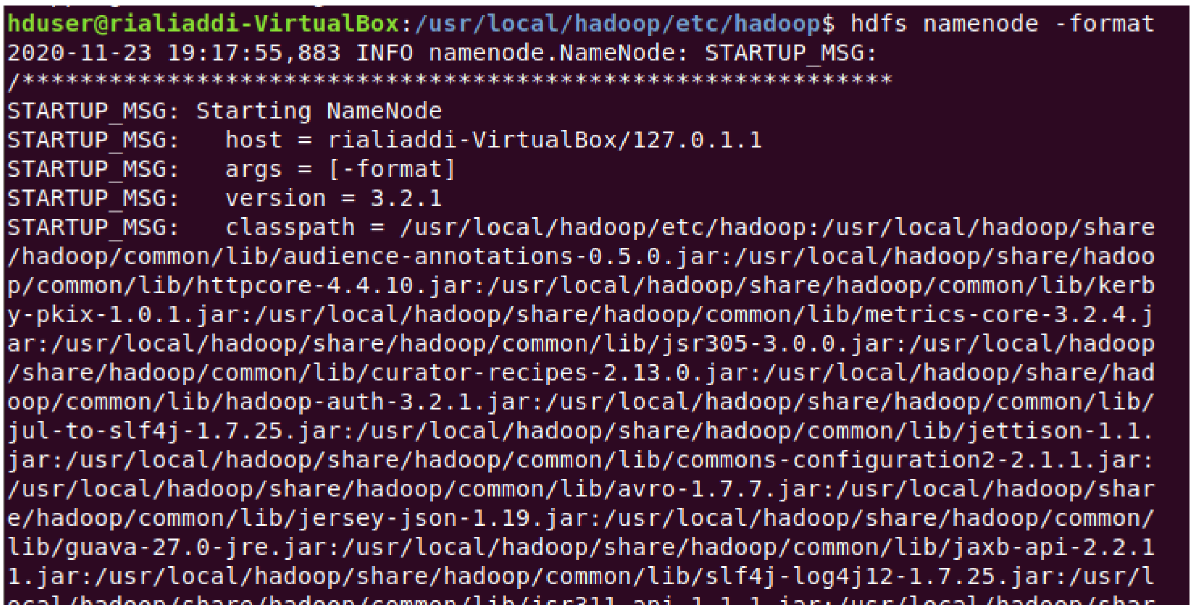

hduser@rialiaddi-VirtualBox: ~/myapp
File Edit View Search Terminal Help

```
[INFO] --- maven-compiler-plugin:3.8.0:testCompile (default-testCompile) @ myap
p ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- maven-surefire-plugin:2.22.1:test (default-test) @ myapp ---
[INFO]
[INFO] -------------------------------------------------------
[INFO]  T E S T S
[INFO] -------------------------------------------------------
[INFO] Running DataEngineer.myapp.AppTest
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.093 s
- in DataEngineer.myapp.AppTest
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO]
[INFO] --- maven-jar-plugin:3.0.2:jar (default-jar) @ myapp ---
[INFO] Building jar: /home/hduser/myapp/target/myapp-1.0-SNAPSHOT.jar
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 7.175 s
[INFO] Finished at: 2020-11-23T19:12:07Z
[INFO] Final Memory: 32M/101M
[INFO] ------------------------------------------------------------------------
hduser@rialiaddi-VirtualBox:~/myapp$
```

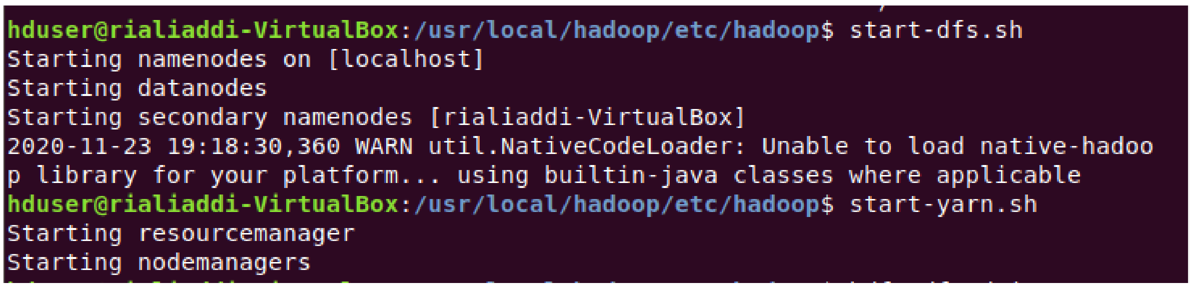
Nettoyage et Formatage du nœud hadoop :



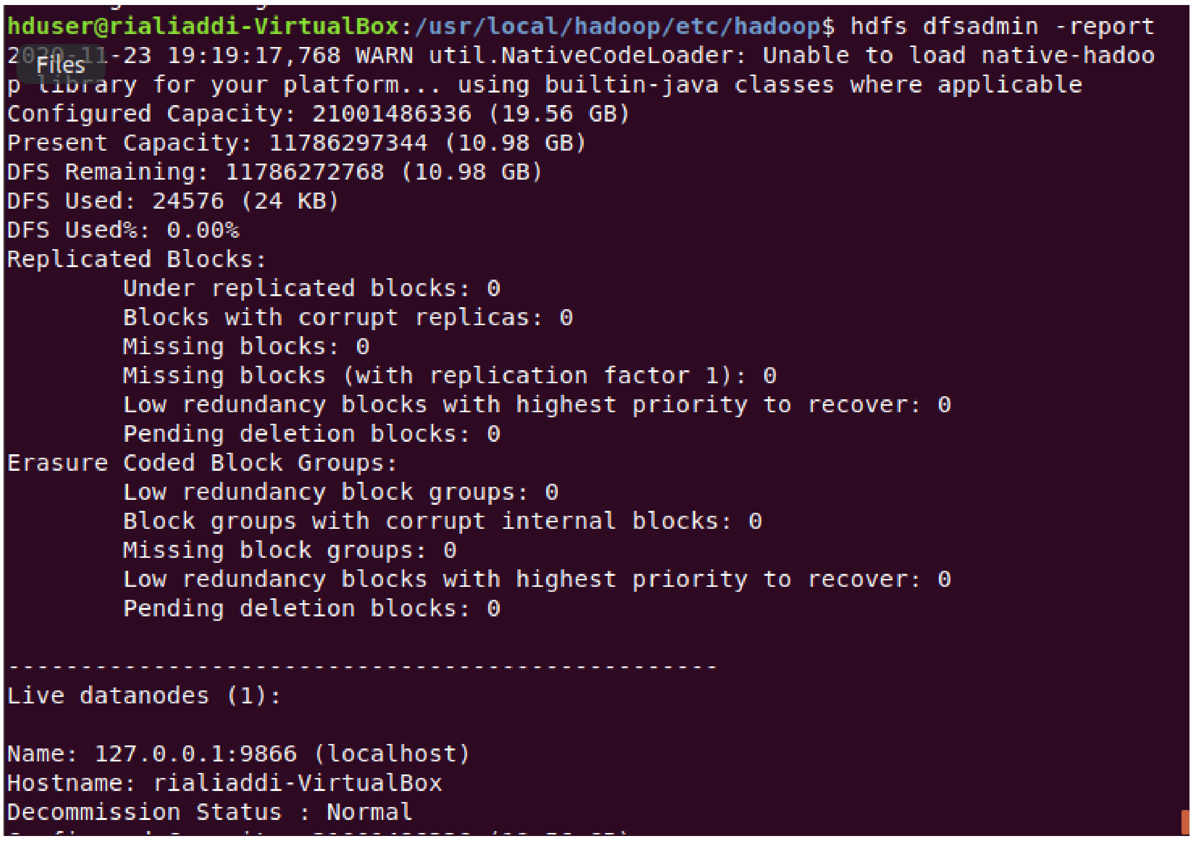
```
hduser@rialiaddi-VirtualBox:~/myapp$ cd /usr/local/hadoop_store
hduser@rialiaddi-VirtualBox:/usr/local/hadoop_store$ sudo rm -rf *
[sudo] password for hduser:
hduser@rialiaddi-VirtualBox:/usr/local/hadoop_store$ sudo mkdir -p /usr/local/h
adoop_store/hdfs/namenode
hduser@rialiaddi-VirtualBox:/usr/local/hadoop_store$ sudo mkdir -p /usr/local/h
adoop_store/hdfs/datanode
hduser@rialiaddi-VirtualBox:/usr/local/hadoop_store$ sudo chown -R hduser /usr/
local/hadoop_store/hdfs/namenode
hduser@rialiaddi-VirtualBox:/usr/local/hadoop_store$ sudo chown -R hduser /usr/
local/hadoop_store/hdfs/datanode
hduser@rialiaddi-VirtualBox:/usr/local/hadoop_store$ cd /usr/local/hadoop/etc/h
adoop
hduser@rialiaddi-VirtualBox:/usr/local/hadoop/etc/hadoop$ hdfs namenode -format
namenode is running as process 6386.  Stop it first.
hduser@rialiaddi-VirtualBox:/usr/local/hadoop/etc/hadoop$ stop-all.s
stop-all.s: command not found
hduser@rialiaddi-VirtualBox:/usr/local/hadoop/etc/hadoop$ stop-all.sh
WARNING: Stopping all Apache Hadoop daemons as hduser in 10 seconds.
WARNING: Use CTRL-C to abort.
Stopping namenodes on [localhost]
Stopping datanodes
Stopping secondary namenodes [rialiaddi-VirtualBox]
2020-11-23 19:17:01,615 WARN util.NativeCodeLoader: Unable to load native-hadoo
p library for your platform... using builtin-java classes where applicable
Stopping nodemanagers
Stopping resourcemanager
```



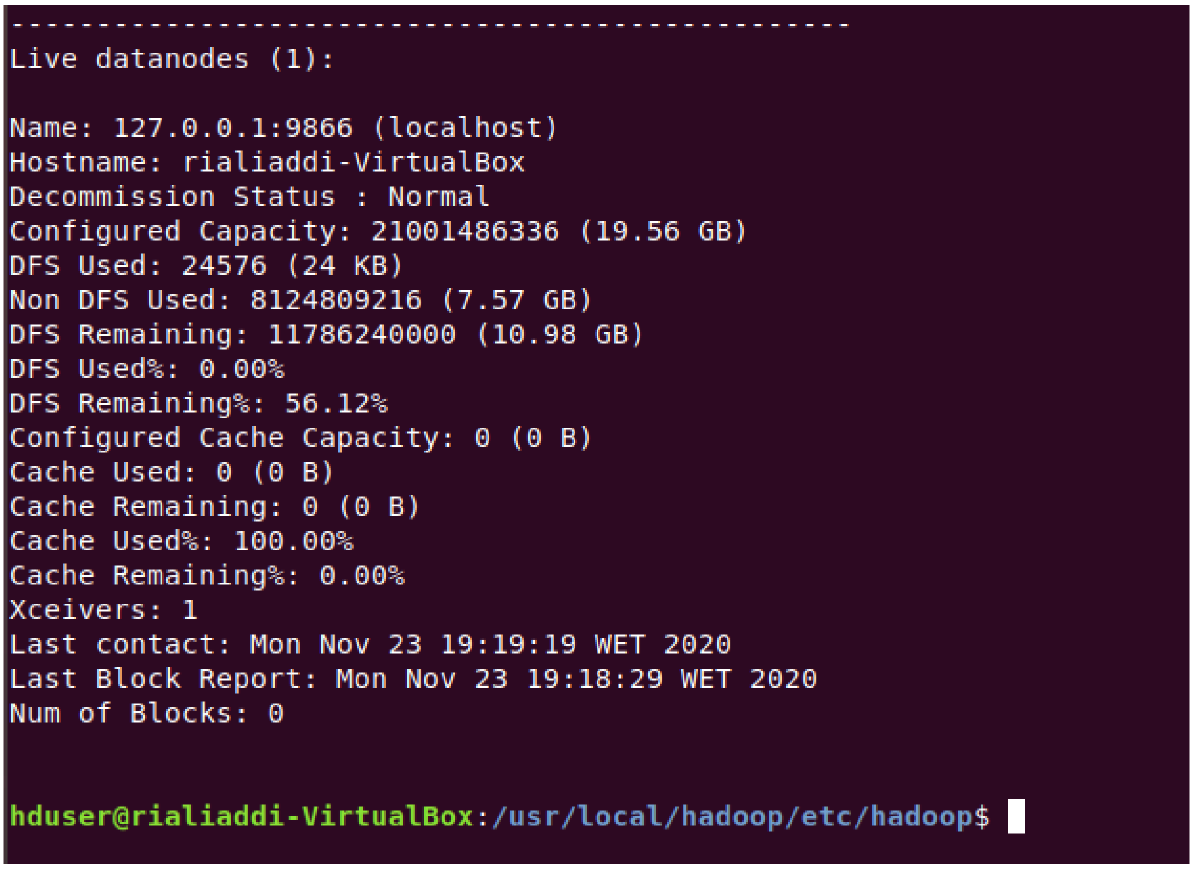
```
hduser@rialiaddi-VirtualBox:/usr/local/hadoop/etc/hadoop$ hdfs namenode -format
2020-11-23 19:17:55,883 INFO namenode.NameNode: STARTUP_MSG:
/************************************************************
STARTUP_MSG: Starting NameNode
STARTUP_MSG:   host = rialiaddi-VirtualBox/127.0.1.1
STARTUP_MSG:   args = [-format]
STARTUP_MSG:   version = 3.2.1
STARTUP_MSG:   classpath = /usr/local/hadoop/etc/hadoop:/usr/local/hadoop/share
/hadoop/common/lib/audience-annotations-0.5.0.jar:/usr/local/hadoop/share/hadoo
p/common/lib/httpcore-4.4.10.jar:/usr/local/hadoop/share/hadoop/common/lib/kerb
y-pkix-1.0.1.jar:/usr/local/hadoop/share/hadoop/common/lib/metrics-core-3.2.4.j
ar:/usr/local/hadoop/share/hadoop/common/lib/jsr305-3.0.0.jar:/usr/local/hadoop
/share/hadoop/common/lib/curator-recipes-2.13.0.jar:/usr/local/hadoop/share/had
oop/common/lib/hadoop-auth-3.2.1.jar:/usr/local/hadoop/share/hadoop/common/lib/
jul-to-slf4j-1.7.25.jar:/usr/local/hadoop/share/hadoop/common/lib/jettison-1.1.
jar:/usr/local/hadoop/share/hadoop/common/lib/commons-configuration2-2.1.1.jar:
/usr/local/hadoop/share/hadoop/common/lib/avro-1.7.7.jar:/usr/local/hadoop/shar
e/hadoop/common/lib/jersey-json-1.19.jar:/usr/local/hadoop/share/hadoop/common/
lib/guava-27.0-jre.jar:/usr/local/hadoop/share/hadoop/common/lib/jaxb-api-2.2.1
1.jar:/usr/local/hadoop/share/hadoop/common/lib/slf4j-log4j12-1.7.25.jar:/usr/l
```



```
hduser@rialiaddi-VirtualBox:/usr/local/hadoop/etc/hadoop$ start-dfs.sh
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [rialiaddi-VirtualBox]
2020-11-23 19:18:30,360 WARN util.NativeCodeLoader: Unable to load native-hadoo
p library for your platform... using builtin-java classes where applicable
hduser@rialiaddi-VirtualBox:/usr/local/hadoop/etc/hadoop$ start-yarn.sh
Starting resourcemanager
Starting nodemanagers
```



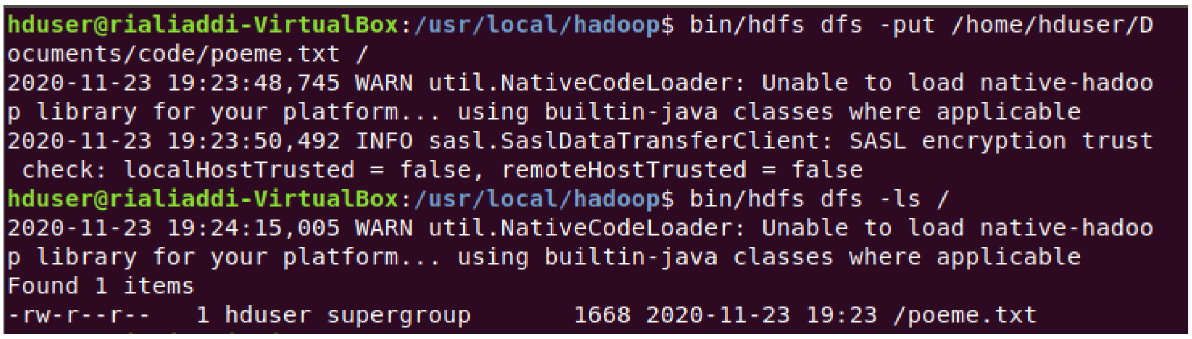
```
hduser@rialiaddi-VirtualBox:/usr/local/hadoop/etc/hadoop$ hdfs dfsadmin -report
2020-11-23 19:19:17,768 WARN util.NativeCodeLoader: Unable to load native-hadoo
p library for your platform... using builtin-java classes where applicable
Configured Capacity: 21001486336 (19.56 GB)
Present Capacity: 11786297344 (10.98 GB)
DFS Remaining: 11786272768 (10.98 GB)
DFS Used: 24576 (24 KB)
DFS Used%: 0.00%
Replicated Blocks:
        Under replicated blocks: 0
        Blocks with corrupt replicas: 0
        Missing blocks: 0
        Missing blocks (with replication factor 1): 0
        Low redundancy blocks with highest priority to recover: 0
        Pending deletion blocks: 0
Erasure Coded Block Groups:
        Low redundancy block groups: 0
        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: rialiaddi-VirtualBox
Decommission Status : Normal
```



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

Name: 127.0.0.1:9866 (localhost)
Hostname: rialiaddi-VirtualBox
Decommission Status : Normal
Configured Capacity: 21001486336 (19.56 GB)
DFS Used: 24576 (24 KB)
Non DFS Used: 8124809216 (7.57 GB)
DFS Remaining: 11786240000 (10.98 GB)
DFS Used%: 0.00%
DFS Remaining%: 56.12%
Configured Cache Capacity: 0 (0 B)
Cache Used: 0 (0 B)
Cache Remaining: 0 (0 B)
Cache Used%: 100.00%
Cache Remaining%: 0.00%
Xceivers: 1
Last contact: Mon Nov 23 19:19:19 WET 2020
Last Block Report: Mon Nov 23 19:18:29 WET 2020
Num of Blocks: 0


hduser@rialiaddi-VirtualBox:/usr/local/hadoop/etc/hadoop$
```

Dépôt du poeme.txt dans HDFS :

Nous allons déposer le poeme.txt dans le HDFS comme précédemment :

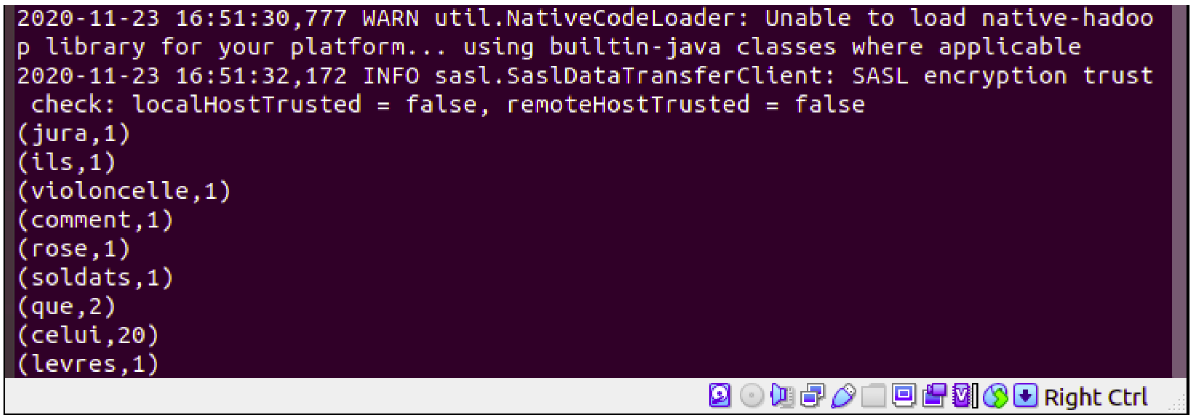


```
hduser@rialiaddi-VirtualBox:/usr/local/hadoop$ bin/hdfs dfs -put /home/hduser/D
ocuments/code/poeme.txt /
2020-11-23 19:23:48,745 WARN util.NativeCodeLoader: Unable to load native-hadoo
p library for your platform... using builtin-java classes where applicable
2020-11-23 19:23:50,492 INFO sasl.SaslDataTransferClient: SASL encryption trust
 check: localHostTrusted = false, remoteHostTrusted = false
hduser@rialiaddi-VirtualBox:/usr/local/hadoop$ bin/hdfs dfs -ls /
2020-11-23 19:24:15,005 WARN util.NativeCodeLoader: Unable to load native-hadoo
p library for your platform... using builtin-java classes where applicable
Found 1 items
-rw-r--r--   1 hduser supergroup       1668 2020-11-23 19:23 /poeme.txt
```

Après exécuter les commandes suivantes :

- **spark-submit --class DataEngineer.myapp.WordCountTask /home/hduser/myapp/target/myapp-1.0- SNAPSHOT.jar /poeme.txt /results**
- **bin/hdfs dfs -cat /results1/part-0000000000**

On arrive à ce résultat :



```
2020-11-23 16:51:30,777 WARN util.NativeCodeLoader: Unable to load native-hadoo
p library for your platform... using builtin-java classes where applicable
2020-11-23 16:51:32,172 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)
```

FIN