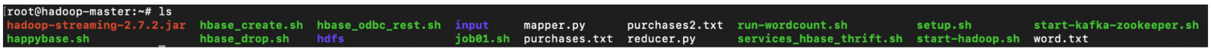
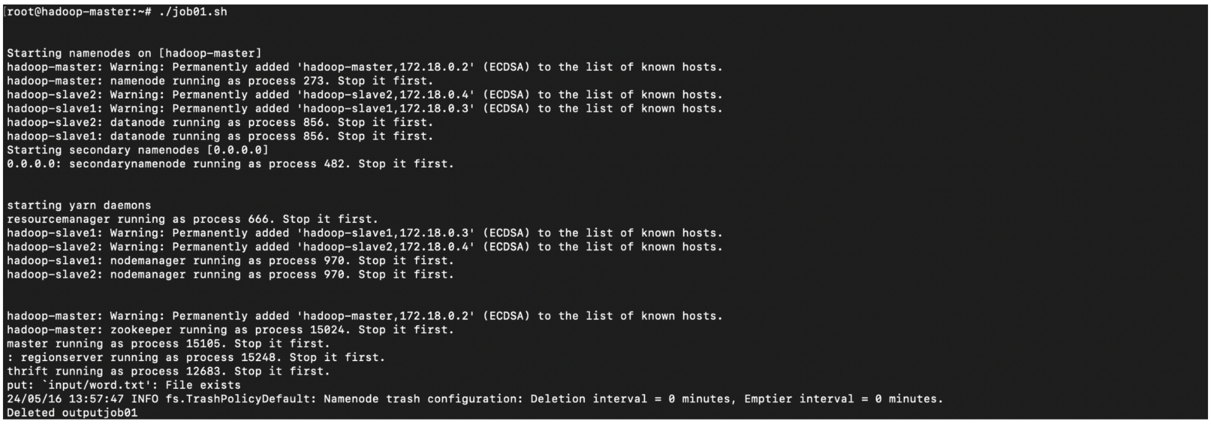
Dossier à rendre

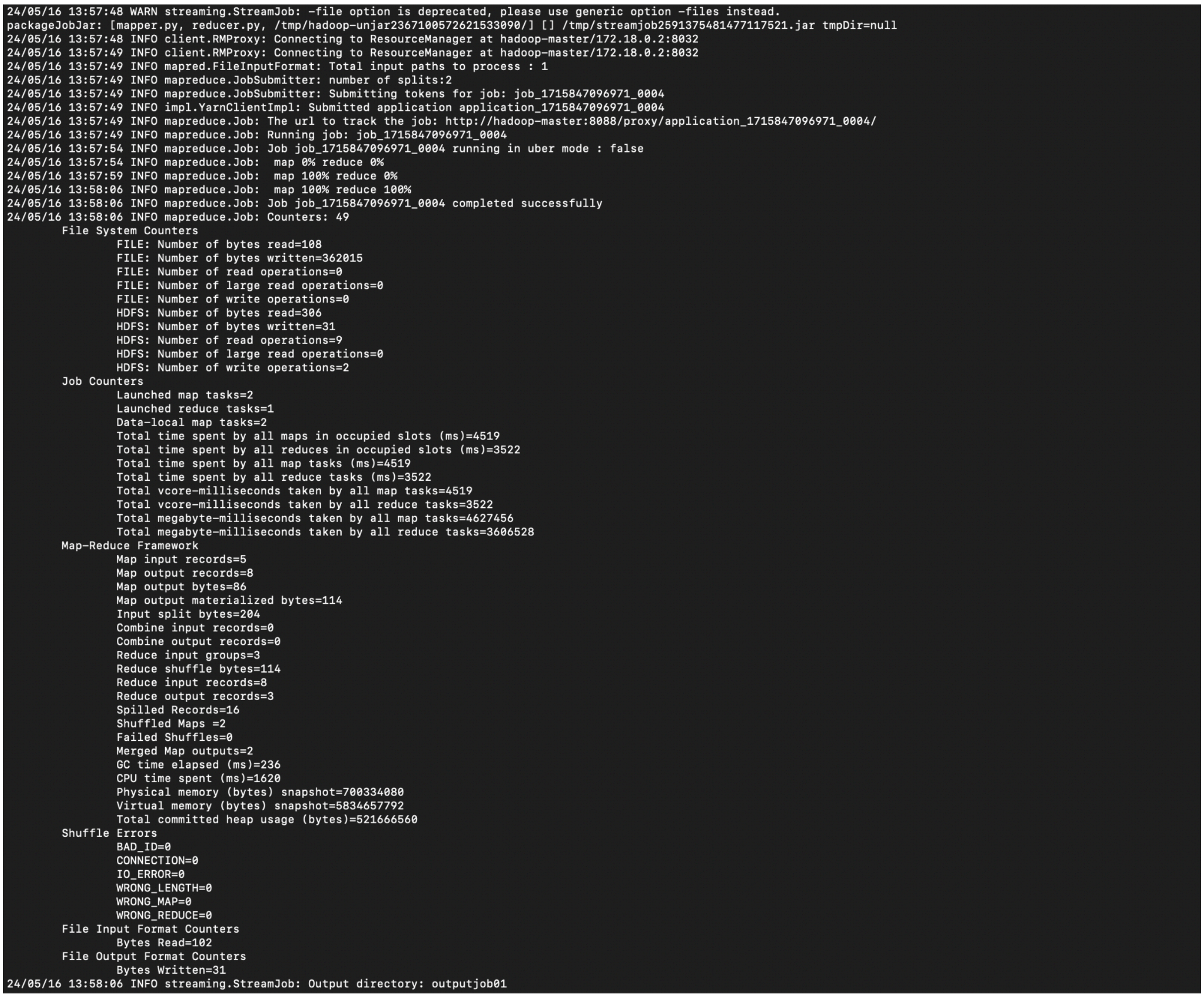
1. TD :



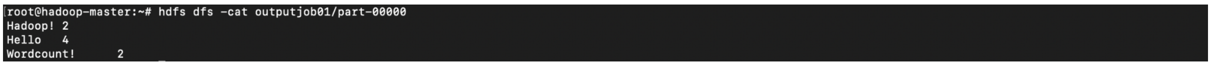
```
root@hadoop-master:~# ls
hadoop-streaming-2.7.2.jar  hbase_create.sh  hbase_odbc_rest.sh  input     mapper.py      purchases2.txt  run-wordcount.sh           setup.sh          start-kafka-zookeeper.sh
happybase.sh                hbase_drop.sh    hdfs                job01.sh  purchases.txt  reducer.py      services_hbase_thrift.sh   start-hadoop.sh   word.txt
```



```
root@hadoop-master:~# ./job01.sh


Starting namenodes on [hadoop-master]
hadoop-master: Warning: Permanently added 'hadoop-master,172.18.0.2' (ECDSA) to the list of known hosts.
hadoop-master: namenode running as process 273. Stop it first.
hadoop-slave2: Warning: Permanently added 'hadoop-slave2,172.18.0.4' (ECDSA) to the list of known hosts.
hadoop-slave1: Warning: Permanently added 'hadoop-slave1,172.18.0.3' (ECDSA) to the list of known hosts.
hadoop-slave2: datanode running as process 856. Stop it first.
hadoop-slave1: datanode running as process 856. Stop it first.
Starting secondary namenodes [0.0.0.0]
0.0.0.0: secondarynamenode running as process 482. Stop it first.


starting yarn daemons
resourcemanager running as process 666. Stop it first.
hadoop-slave1: Warning: Permanently added 'hadoop-slave1,172.18.0.3' (ECDSA) to the list of known hosts.
hadoop-slave2: Warning: Permanently added 'hadoop-slave2,172.18.0.4' (ECDSA) to the list of known hosts.
hadoop-slave1: nodemanager running as process 970. Stop it first.
hadoop-slave2: nodemanager running as process 970. Stop it first.


hadoop-master: Warning: Permanently added 'hadoop-master,172.18.0.2' (ECDSA) to the list of known hosts.
hadoop-master: zookeeper running as process 15024. Stop it first.
master running as process 15105. Stop it first.
: regionserver running as process 15248. Stop it first.
thrift running as process 12683. Stop it first.
put: `input/word.txt': File exists
24/05/16 13:57:47 INFO fs.TrashPolicyDefault: Namenode trash configuration: Deletion interval = 0 minutes, Emptier interval = 0 minutes.
Deleted outputjob01
```



```
24/05/16 13:57:48 WARN streaming.StreamJob: -file option is deprecated, please use generic option -files instead.
packageJobJar: [mapper.py, reducer.py, /tmp/hadoop-unjar2367100572621533090/] [] /tmp/streamjob2591375481477117521.jar tmpDir=null
24/05/16 13:57:48 INFO client.RMProxy: Connecting to ResourceManager at hadoop-master/172.18.0.2:8032
24/05/16 13:57:49 INFO client.RMProxy: Connecting to ResourceManager at hadoop-master/172.18.0.2:8032
24/05/16 13:57:49 INFO mapred.FileInputFormat: Total input paths to process : 1
24/05/16 13:57:49 INFO mapreduce.JobSubmitter: number of splits:2
24/05/16 13:57:49 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1715847096971_0004
24/05/16 13:57:49 INFO impl.YarnClientImpl: Submitted application application_1715847096971_0004
24/05/16 13:57:49 INFO mapreduce.Job: The url to track the job: http://hadoop-master:8088/proxy/application_1715847096971_0004/
24/05/16 13:57:49 INFO mapreduce.Job: Running job: job_1715847096971_0004
24/05/16 13:57:54 INFO mapreduce.Job: Job job_1715847096971_0004 running in uber mode : false
24/05/16 13:57:54 INFO mapreduce.Job:  map 0% reduce 0%
24/05/16 13:57:59 INFO mapreduce.Job:  map 100% reduce 0%
24/05/16 13:58:06 INFO mapreduce.Job:  map 100% reduce 100%
24/05/16 13:58:06 INFO mapreduce.Job: Job job_1715847096971_0004 completed successfully
24/05/16 13:58:06 INFO mapreduce.Job: Counters: 49
        File System Counters
                FILE: Number of bytes read=108
                FILE: Number of bytes written=362015
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=306
                HDFS: Number of bytes written=31
                HDFS: Number of read operations=9
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
        Job Counters
                Launched map tasks=2
                Launched reduce tasks=1
                Data-local map tasks=2
                Total time spent by all maps in occupied slots (ms)=4519
                Total time spent by all reduces in occupied slots (ms)=3522
                Total time spent by all map tasks (ms)=4519
                Total time spent by all reduce tasks (ms)=3522
                Total vcore-milliseconds taken by all map tasks=4519
                Total vcore-milliseconds taken by all reduce tasks=3522
                Total megabyte-milliseconds taken by all map tasks=4627456
                Total megabyte-milliseconds taken by all reduce tasks=3606528
        Map-Reduce Framework
                Map input records=5
                Map output records=8
                Map output bytes=86
                Map output materialized bytes=114
                Input split bytes=204
                Combine input records=0
                Combine output records=0
                Reduce input groups=3
                Reduce shuffle bytes=114
                Reduce input records=8
                Reduce output records=3
                Spilled Records=16
                Shuffled Maps =2
                Failed Shuffles=0
                Merged Map outputs=2
                GC time elapsed (ms)=236
                CPU time spent (ms)=1620
                Physical memory (bytes) snapshot=700334080
                Virtual memory (bytes) snapshot=5834657792
                Total committed heap usage (bytes)=521666560
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=102
        File Output Format Counters
                Bytes Written=31
24/05/16 13:58:06 INFO streaming.StreamJob: Output directory: outputjob01
```



```
root@hadoop-master:~# hdfs dfs -cat outputjob01/part-00000
Hadoop! 2
Hello   4
Wordcount!      2
```

2. L’analyse de la compréhension des outils du cloud et de son utilisation

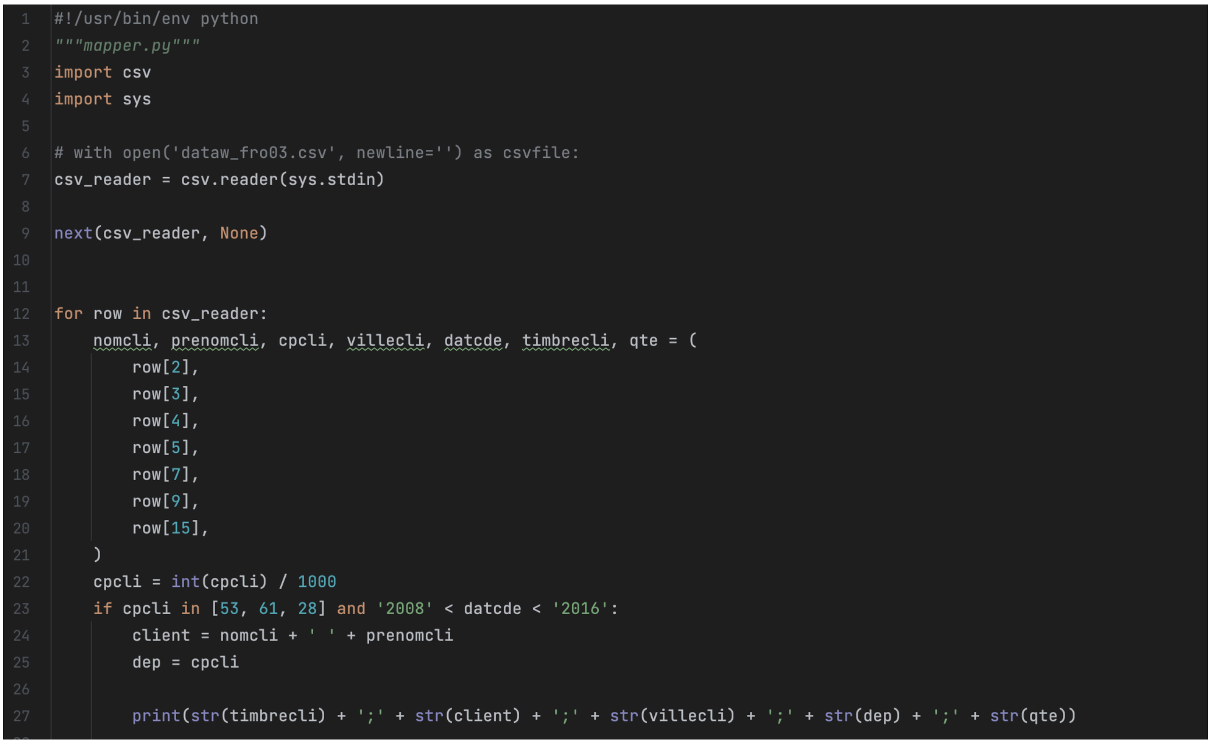
Pour l’accès au cloud hidora, j’utilise filezilla pour le transfert de fichier et ensuite grâce à un accès SSH je peux me connecter directement à la machine virtuel.

J’utilise l’endpoint (port 22) pour la connexion.

3. Expliquer la sécurité votre VM (par le CLOUD)

Les différents endpoints avec des ip privés ainsi qu’une identification (login/password) permettent une sécurité du cloud.

4. Les sources du TP avec des commentaires

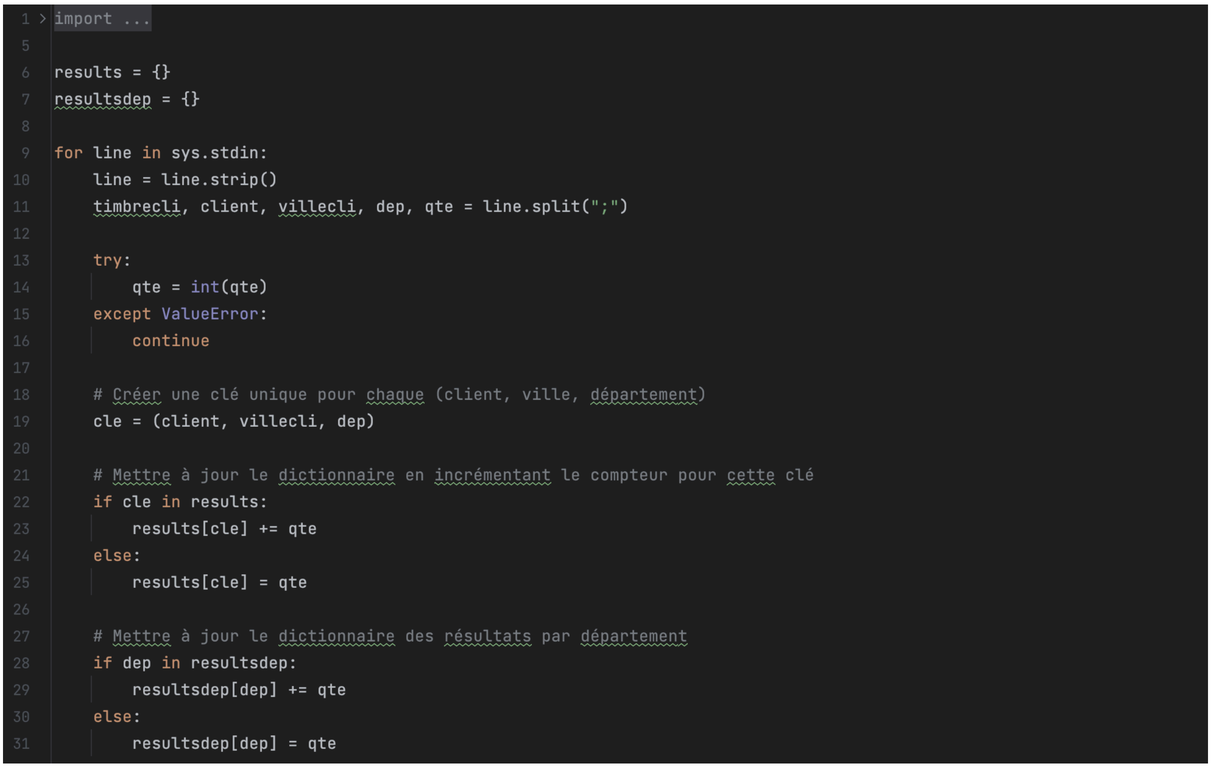


```python
#!/usr/bin/env python
"""mapper.py"""
import csv
import sys

# with open('dataw_fro03.csv', newline='') as csvfile:
csv_reader = csv.reader(sys.stdin)

next(csv_reader, None)


for row in csv_reader:
    nomcli, prenomcli, cpcli, villecli, datcde, timbrecli, qte = (
        row[2],
        row[3],
        row[4],
        row[5],
        row[7],
        row[9],
        row[15],
    )
    cpcli = int(cpcli) / 1000
    if cpcli in [53, 61, 28] and '2008' < datcde < '2016':
        client = nomcli + ' ' + prenomcli
        dep = cpcli

        print(str(timbrecli) + ';' + str(client) + ';' + str(villecli) + ';' + str(dep) + ';' + str(qte))
```

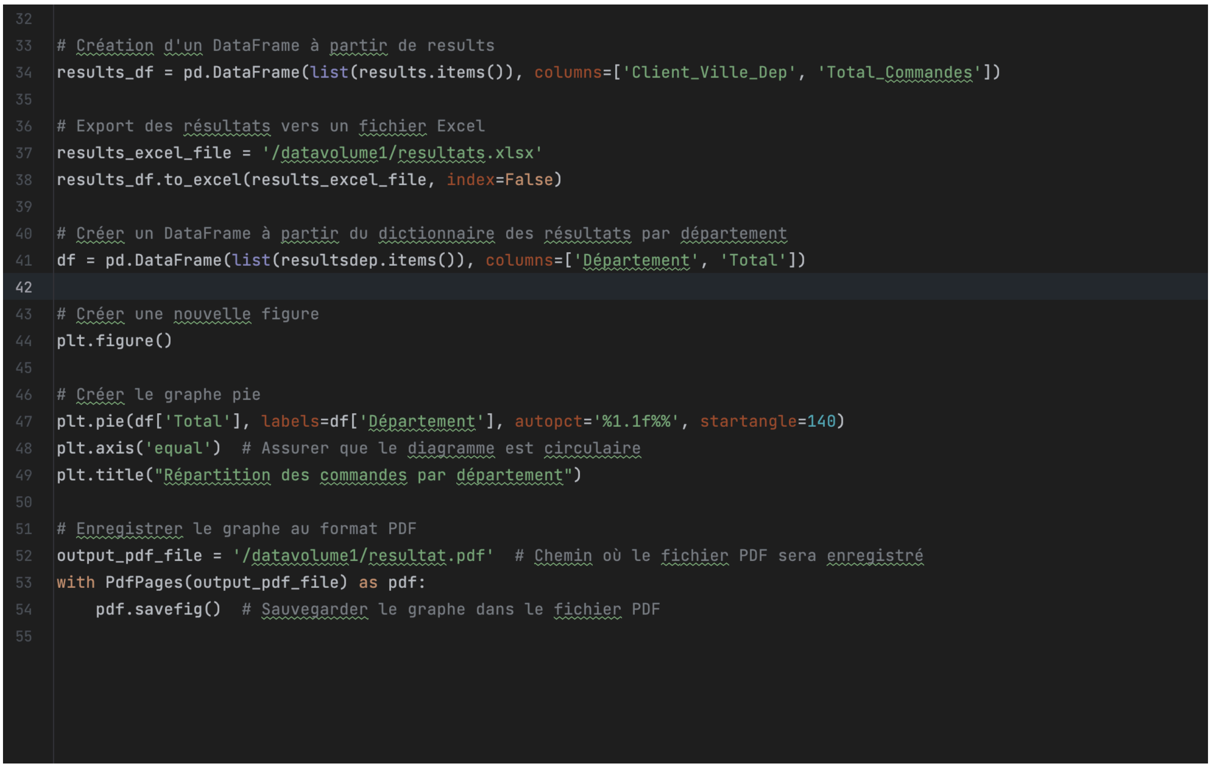
On est sur un mapper assez simple utilisant la librairie csv pour lire le fichier csv ouvert par hadoop.

Ensuite on identifie les différentes colonnes qui nous intéresse.

On filtre les données qui ne nous interesse pas et on les envoies pour le traitement par le reducer.



```
import ...

results = {}
resultsdep = {}

for line in sys.stdin:
    line = line.strip()
    timbrecli, client, villecli, dep, qte = line.split(";")

    try:
        qte = int(qte)
    except ValueError:
        continue

    # Créer une clé unique pour chaque (client, ville, département)
    cle = (client, villecli, dep)

    # Mettre à jour le dictionnaire en incrémentant le compteur pour cette clé
    if cle in results:
        results[cle] += qte
    else:
        results[cle] = qte

    # Mettre à jour le dictionnaire des résultats par département
    if dep in resultsdep:
        resultsdep[dep] += qte
    else:
        resultsdep[dep] = qte
```



```

# Création d'un DataFrame à partir de results
results_df = pd.DataFrame(list(results.items()), columns=['Client_Ville_Dep', 'Total_Commandes'])

# Export des résultats vers un fichier Excel
results_excel_file = '/datavolume1/resultats.xlsx'
results_df.to_excel(results_excel_file, index=False)

# Créer un DataFrame à partir du dictionnaire des résultats par département
df = pd.DataFrame(list(resultsdep.items()), columns=['Département', 'Total'])

# Créer une nouvelle figure
plt.figure()

# Créer le graphe pie
plt.pie(df['Total'], labels=df['Département'], autopct='%1.1f%%', startangle=140)
plt.axis('equal')  # Assurer que le diagramme est circulaire
plt.title("Répartition des commandes par département")

# Enregistrer le graphe au format PDF
output_pdf_file = '/datavolume1/resultat.pdf'  # Chemin où le fichier PDF sera enregistré
with PdfPages(output_pdf_file) as pdf:
    pdf.savefig()  # Sauvegarder le graphe dans le fichier PDF

```

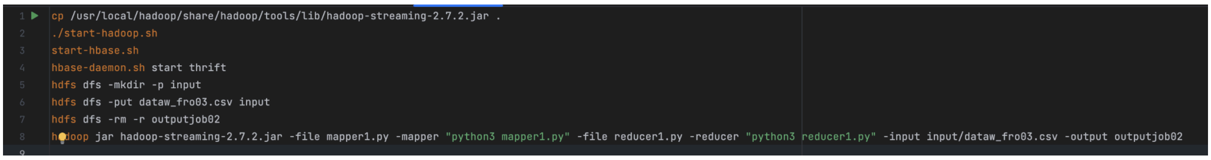
Le reducer traite les différentes données en deux objet transformer ensuite en dataframe.

Le premier permet d’afficher dans un excel les clients et la quantités de commandes qu’ils ont effectués.

Le deuxièmes les commandes par départements affiché dans une pie faites avec matplotlib.

5. Des procédures d’import des données (voir les fichier sh fournis sur la plateforme et explicité dans les fichiers pdf)

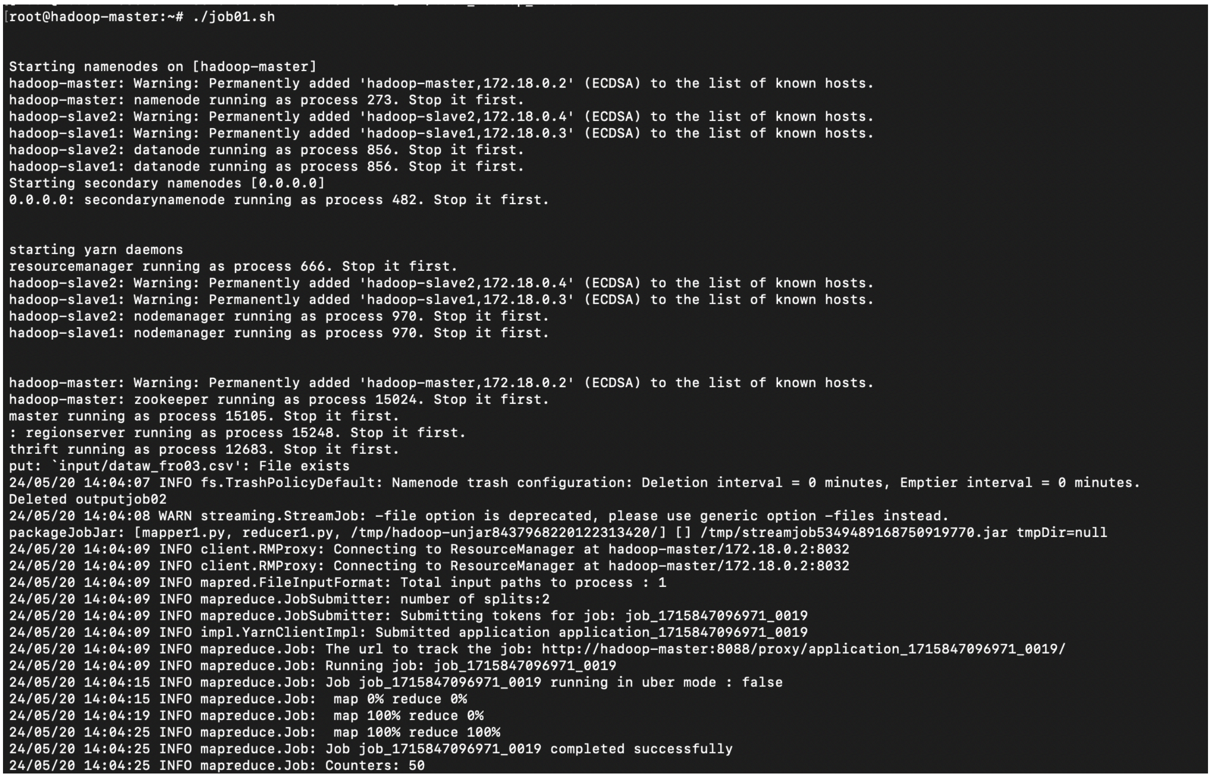
Pour pouvoir ensuite lancer mon mapper et reducer, j’adapte le fichier job01.sh.



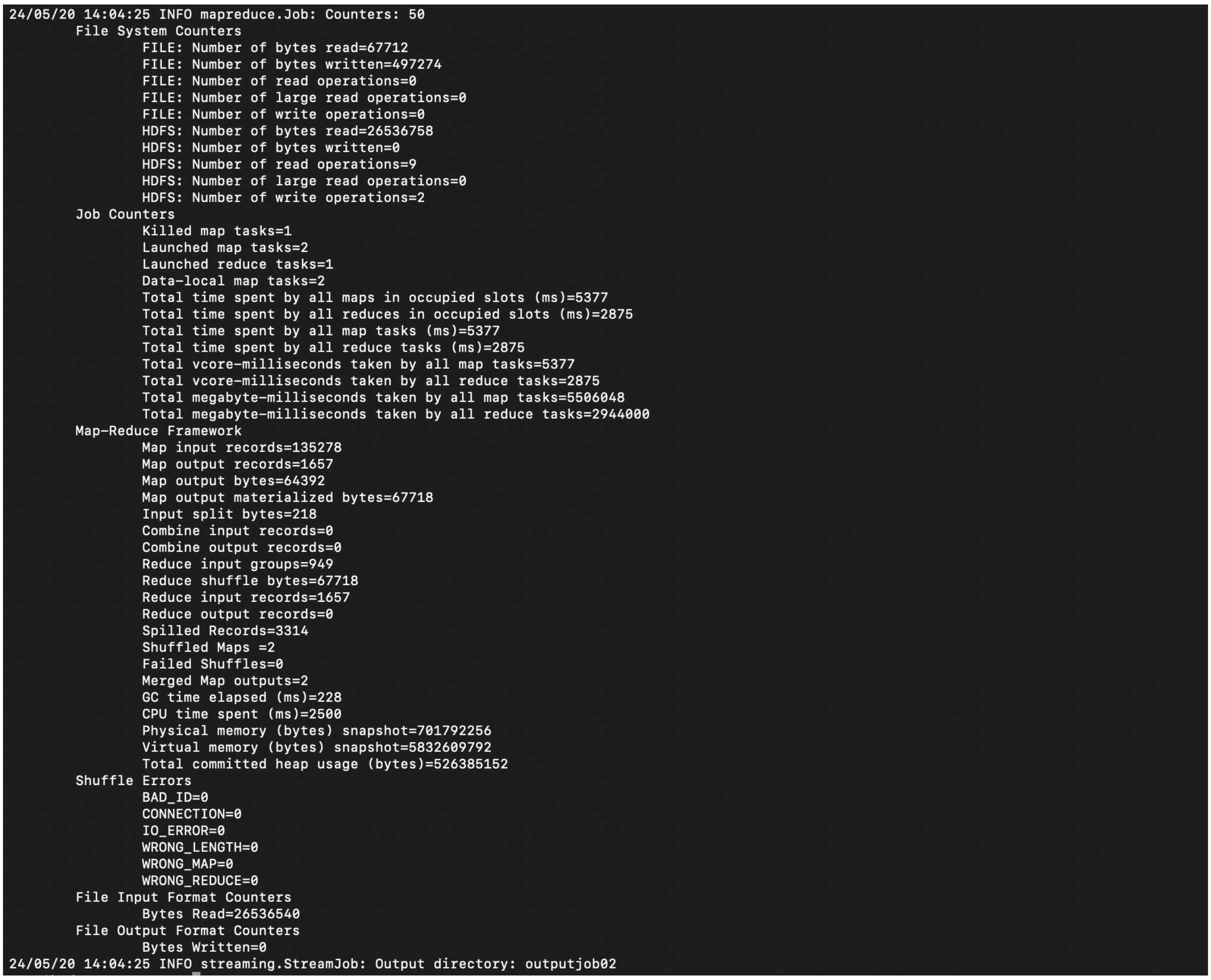
```
cp /usr/local/hadoop/share/hadoop/tools/lib/hadoop-streaming-2.7.2.jar .
./start-hadoop.sh
start-hbase.sh
hbase-daemon.sh start thrift
hdfs dfs -mkdir -p input
hdfs dfs -put dataw_fro03.csv input
hdfs dfs -rm -r outputjob02
hadoop jar hadoop-streaming-2.7.2.jar -file mapper1.py -mapper "python3 mapper1.py" -file reducer1.py -reducer "python3 reducer1.py" -input input/dataw_fro03.csv -output outputjob02
```

Ensuite grâce à Filezilla, j’envoie tous les documents sur le cloud, je m’y connecte en SSH.

Je copie tout sur le docker, et j’execute le job01.sh.



```
[root@hadoop-master:~# ./job01.sh


Starting namenodes on [hadoop-master]
hadoop-master: Warning: Permanently added 'hadoop-master,172.18.0.2' (ECDSA) to the list of known hosts.
hadoop-master: namenode running as process 273. Stop it first.
hadoop-slave2: Warning: Permanently added 'hadoop-slave2,172.18.0.4' (ECDSA) to the list of known hosts.
hadoop-slave1: Warning: Permanently added 'hadoop-slave1,172.18.0.3' (ECDSA) to the list of known hosts.
hadoop-slave2: datanode running as process 856. Stop it first.
hadoop-slave1: datanode running as process 856. Stop it first.
Starting secondary namenodes [0.0.0.0]
0.0.0.0: secondarynamenode running as process 482. Stop it first.


starting yarn daemons
resourcemanager running as process 666. Stop it first.
hadoop-slave2: Warning: Permanently added 'hadoop-slave2,172.18.0.4' (ECDSA) to the list of known hosts.
hadoop-slave1: Warning: Permanently added 'hadoop-slave1,172.18.0.3' (ECDSA) to the list of known hosts.
hadoop-slave2: nodemanager running as process 970. Stop it first.
hadoop-slave1: nodemanager running as process 970. Stop it first.


hadoop-master: Warning: Permanently added 'hadoop-master,172.18.0.2' (ECDSA) to the list of known hosts.
hadoop-master: zookeeper running as process 15024. Stop it first.
master running as process 15105. Stop it first.
: regionserver running as process 15248. Stop it first.
thrift running as process 12683. Stop it first.
put: `input/dataw_fro03.csv': File exists
24/05/20 14:04:07 INFO fs.TrashPolicyDefault: Namenode trash configuration: Deletion interval = 0 minutes, Emptier interval = 0 minutes.
Deleted outputjob02
24/05/20 14:04:08 WARN streaming.StreamJob: -file option is deprecated, please use generic option -files instead.
packageJobJar: [mapper1.py, reducer1.py, /tmp/hadoop-unjar8437968220122313420/] [] /tmp/streamjob5349489168750919770.jar tmpDir=null
24/05/20 14:04:09 INFO client.RMProxy: Connecting to ResourceManager at hadoop-master/172.18.0.2:8032
24/05/20 14:04:09 INFO client.RMProxy: Connecting to ResourceManager at hadoop-master/172.18.0.2:8032
24/05/20 14:04:09 INFO mapred.FileInputFormat: Total input paths to process : 1
24/05/20 14:04:09 INFO mapreduce.JobSubmitter: number of splits:2
24/05/20 14:04:09 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1715847096971_0019
24/05/20 14:04:09 INFO impl.YarnClientImpl: Submitted application application_1715847096971_0019
24/05/20 14:04:09 INFO mapreduce.Job: The url to track the job: http://hadoop-master:8088/proxy/application_1715847096971_0019/
24/05/20 14:04:09 INFO mapreduce.Job: Running job: job_1715847096971_0019
24/05/20 14:04:15 INFO mapreduce.Job: Job job_1715847096971_0019 running in uber mode : false
24/05/20 14:04:15 INFO mapreduce.Job:  map 0% reduce 0%
24/05/20 14:04:19 INFO mapreduce.Job:  map 100% reduce 0%
24/05/20 14:04:25 INFO mapreduce.Job:  map 100% reduce 100%
24/05/20 14:04:25 INFO mapreduce.Job: Job job_1715847096971_0019 completed successfully
24/05/20 14:04:25 INFO mapreduce.Job: Counters: 50
```



```
24/05/20 14:04:25 INFO mapreduce.Job: Counters: 50
        File System Counters
                FILE: Number of bytes read=67712
                FILE: Number of bytes written=497274
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=26536758
                HDFS: Number of bytes written=0
                HDFS: Number of read operations=9
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
        Job Counters
                Killed map tasks=1
                Launched map tasks=2
                Launched reduce tasks=1
                Data-local map tasks=2
                Total time spent by all maps in occupied slots (ms)=5377
                Total time spent by all reduces in occupied slots (ms)=2875
                Total time spent by all map tasks (ms)=5377
                Total time spent by all reduce tasks (ms)=2875
                Total vcore-milliseconds taken by all map tasks=5377
                Total vcore-milliseconds taken by all reduce tasks=2875
                Total megabyte-milliseconds taken by all map tasks=5506048
                Total megabyte-milliseconds taken by all reduce tasks=2944000
        Map-Reduce Framework
                Map input records=135278
                Map output records=1657
                Map output bytes=64392
                Map output materialized bytes=67718
                Input split bytes=218
                Combine input records=0
                Combine output records=0
                Reduce input groups=949
                Reduce shuffle bytes=67718
                Reduce input records=1657
                Reduce output records=0
                Spilled Records=3314
                Shuffled Maps =2
                Failed Shuffles=0
                Merged Map outputs=2
                GC time elapsed (ms)=228
                CPU time spent (ms)=2500
                Physical memory (bytes) snapshot=701792256
                Virtual memory (bytes) snapshot=5832609792
                Total committed heap usage (bytes)=526385152
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=26536540
        File Output Format Counters
                Bytes Written=0
24/05/20 14:04:25 INFO streaming.StreamJob: Output directory: outputjob02
```

Tout est alors envoyé dans le datavolume1 du docker.

Je les récupères grâce à Filezilla et les mets sur ma machine locale.

6. Vos recommandations par rapport à l’utilisation de ce cloud privé

Je n’avais encore jamais utilisé de cloud pour le travail de cette manière et j’ai trouvé cela très pratique. Mon mac (avec une puce M1) ne me permettant pas de pouvoir installé Hadoop sur ma machine, cela permet donc de mettre en place un espace de travail totalement géré et indépendant de la machine de chaque utilisateur.