Projet Big Data Anas Ben Hassine 2BD2

Bonjour,

Partie avec Des Images Hadoop preinstalle

Pour Le projet Big Data, j'ai installé docker sur windows,

Puis j'ai accede au terminal et crée un repertoire de Hadoop et met un fichier

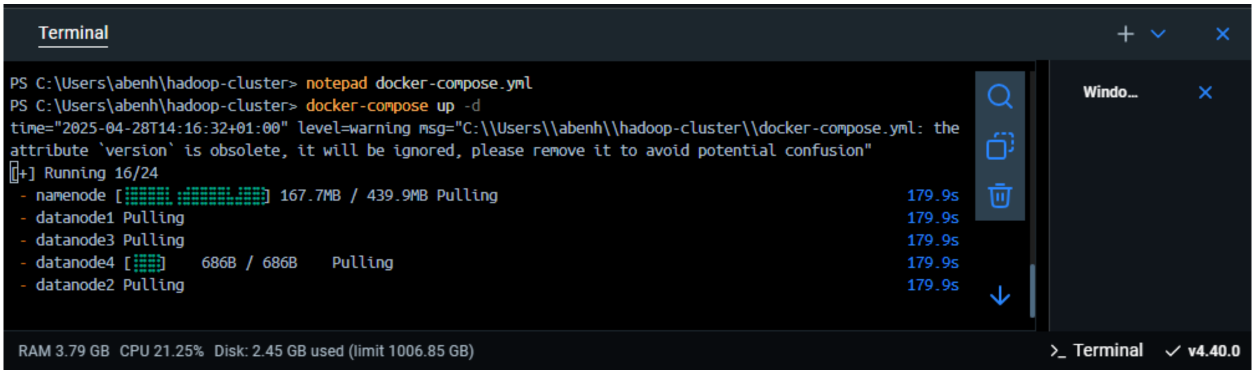
Docker-compose.yaml qui va creer 3 data nodes et une seul image name node voici son contenu:

```
version: '3'


networks:
  hadoop:
    driver: bridge


services:
  namenode:
    image: bde2020/hadoop-namenode:2.0.0-hadoop3.2.1-java8
    container_name: namenode
    environment:
      - CLUSTER_NAME=test
      - CORE_CONF_fs_defaultFS=hdfs://namenode:9000
    ports:
      - "9870:9870"
      - "9000:9000"
    volumes:
      - hadoop_namenode:/hadoop/dfs/name
    networks:
```

```
    - hadoop

  datanode1:
    image: bde2020/hadoop-datanode:2.0.0-hadoop3.2.1-java8
    container_name: datanode1
    environment:
      - CLUSTER_NAME=test
      - CORE_CONF_fs_defaultFS=hdfs://namenode:9000
    depends_on:
      - namenode
    ports:
      - "9864:9864"
    volumes:
      - hadoop_datanode1:/hadoop/dfs/data
    networks:
      - hadoop

  datanode2:
    image: bde2020/hadoop-datanode:2.0.0-hadoop3.2.1-java8
    container_name: datanode2
    environment:
      - CLUSTER_NAME=test
      - CORE_CONF_fs_defaultFS=hdfs://namenode:9000
    depends_on:
      - namenode
    ports:
```

```
    - "9865:9864"
  volumes:
    - hadoop_datanode2:/hadoop/dfs/data
  networks:
    - hadoop

  datanode3:
  image: bde2020/hadoop-datanode:2.0.0-hadoop3.2.1-java8
  container_name: datanode3
  environment:
    - CLUSTER_NAME=test
    - CORE_CONF_fs_defaultFS=hdfs://namenode:9000
  depends_on:
    - namenode
  ports:
    - "9866:9864"
  volumes:
    - hadoop_datanode3:/hadoop/dfs/data
  networks:
    - hadoop

volumes:
  hadoop_namenode:
  hadoop_datanode1:
  hadoop_datanode2:
  hadoop_datanode3:
```

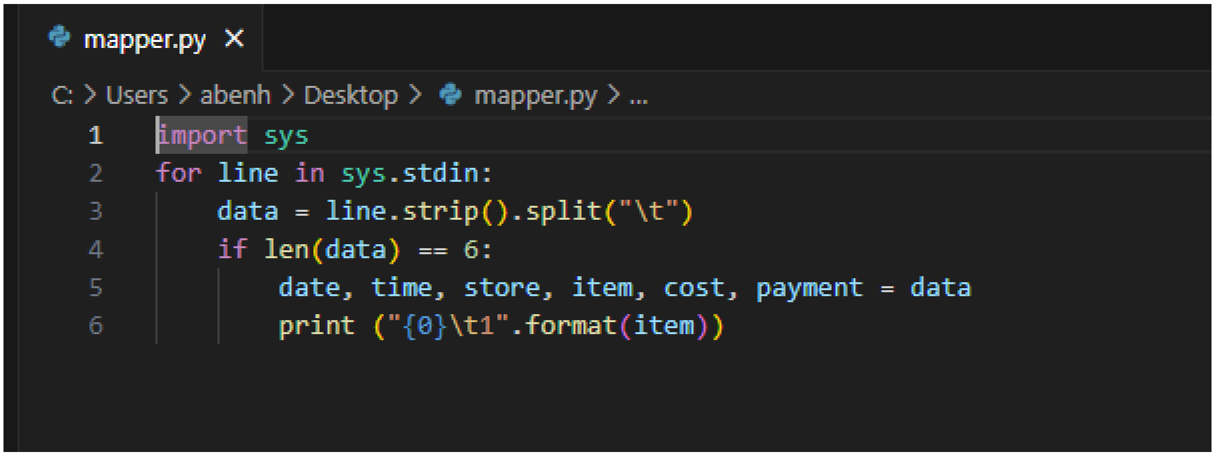
et puis fait le commande docker-compose up -d pour l’executer



```
PS C:\Users\abenh\hadoop-cluster> notepad docker-compose.yml
PS C:\Users\abenh\hadoop-cluster> docker-compose up -d
time="2025-04-28T14:16:32+01:00" level=warning msg="C:\\Users\\abenh\\hadoop-cluster\\docker-compose.yml: the
attribute `version` is obsolete, it will be ignored, please remove it to avoid potential confusion"
[+] Running 16/24
 - namenode [⣿⣿⣿⣿⣿⡀⣿⣿⣿⣿⣿⣿⣀⣿⣿⣿] 167.7MB / 439.9MB Pulling                                   179.9s
 - datanode1 Pulling                                                                  179.9s
 - datanode3 Pulling                                                                  179.9s
 - datanode4 [⣿⣿⣿]    686B / 686B    Pulling                                         179.9s
 - datanode2 Pulling                                                                  179.9s
```

Puis, j’ai implementé un mapper en python

J’ai ajouté apres la ligne #!/usr/bin/env python3 au premier ligne pour le mapper et reducer



```python
import sys
for line in sys.stdin:
    data = line.strip().split("\t")
    if len(data) == 6:
        date, time, store, item, cost, payment = data
        print ("{0}\t1".format(item))
```

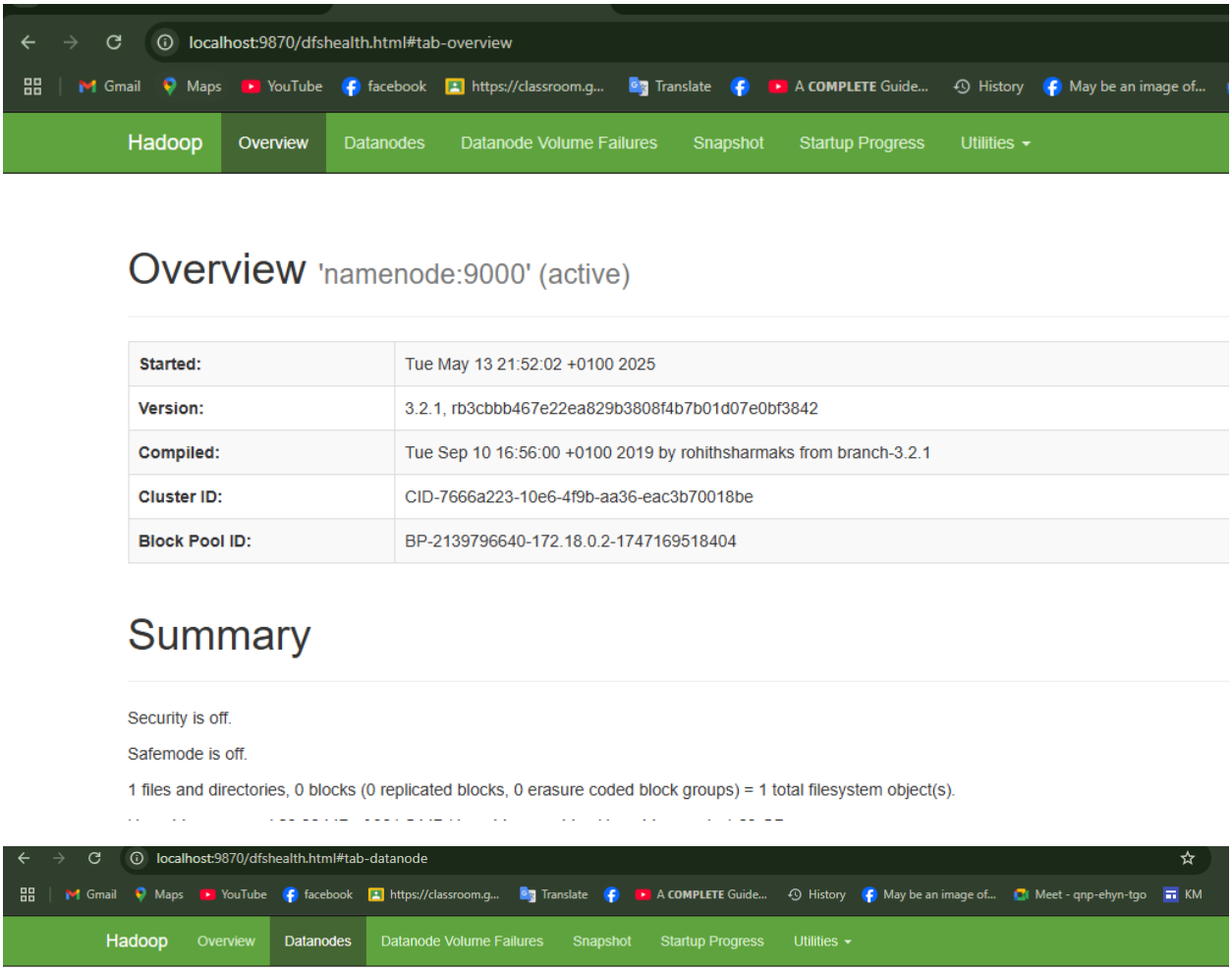
Et reducer.py

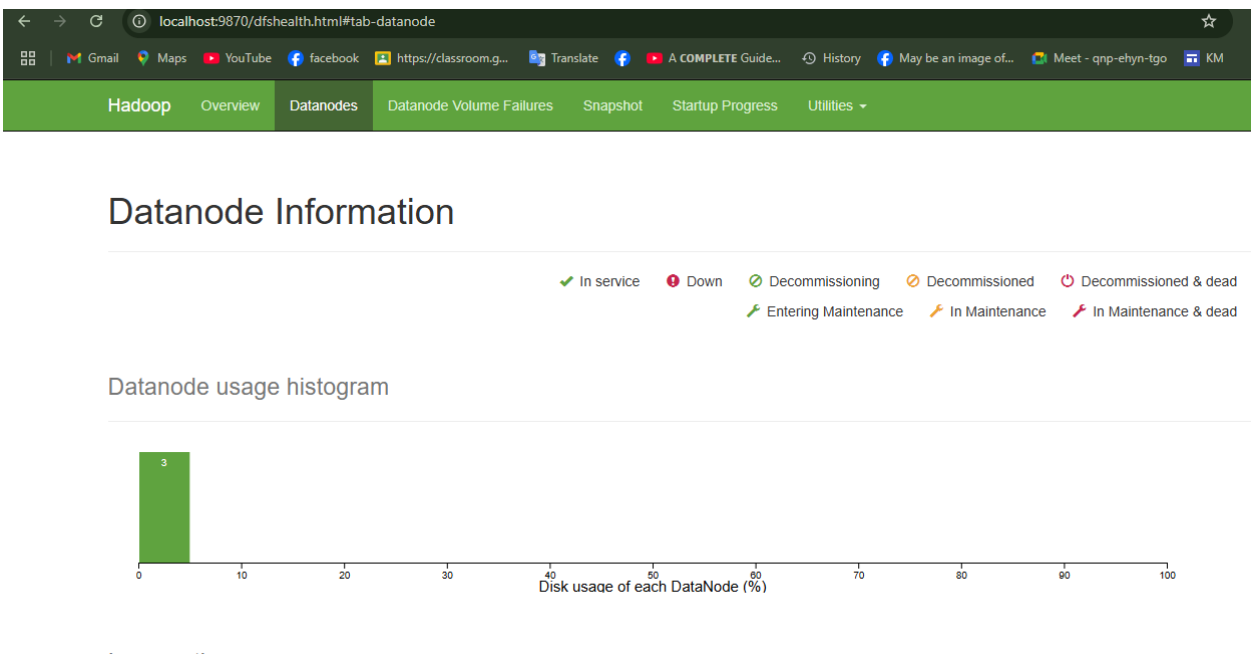
mapper.py reducer.py

C: > Users > abenh > Desktop > reducer.py > ...

```python
import sys
salesTotal = 0
oldKey = None

for line in sys.stdin:
    data_mapped = line.strip().split("\t")
    if len(data_mapped) != 2:
        continue

    thisKey, thisSale = data_mapped
    if oldKey and oldKey != thisKey:
        print (oldKey, "\t", salesTotal)
        oldKey = thisKey;
        salesTotal = 0
    oldKey = thisKey
    salesTotal += float(thisSale)
if oldKey != None:
    print (oldKey, "\t", salesTotal)
```

J’ai accede au WEB UI de Namenode





⇨ Toute les datanodes fonctionnent et sont connectés a namenode donc notre cluster est ready pour le Job map reduce

Puis, j'ai copié le fichier purchases.txt.gz dans le cluster

```
docker cp "C:\Users\abenh\Downloads\purchases.txt.gz" namenode:/purchases.txt.gz
```

```
docker exec -it namenode bash
hdfs dfs -mkdir -p /input
hdfs dfs -put /purchases.txt.gz /input/
```

et met le mapper.py et reducer.py dans le docker aussi

```
exit
docker cp "C:\Users\abenh\Downloads\mapper.py" namenode:/mapper.py
docker cp "C:\Users\abenh\Downloads\reducer.py" namenode:/reducer.py
docker exec -it namenode bash
```

et executer le Hadoop streaming pur que python soit interpreté

```
hadoop jar $HADOOP_HOME/share/hadoop/tools/lib/hadoop-streaming-*.jar \
-input /input/purchases.txt.gz \
-output /output \
-mapper mapper.py \
-reducer reducer.py
```

```
        Map-Reduce Framework
                Map input records=4138476
                Map output records=4138476
                Map output bytes=52880666
                Map output materialized bytes=61157630
                Input split bytes=184
                Combine input records=0
                Combine output records=0
                Reduce input groups=18
                Reduce shuffle bytes=61157630
                Reduce input records=4138476
```

root@25e7ca210b92:~# hadoop fs -cat /output-purchases/part-00000

2025-05-13 14:15:46,007 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false

Baby 230293.0

Books 229787.0

CDs 230039.0

Cameras 229320.0

Children's Clothing 230469.0

Computers 229059.0

Consumer Electronics 229761.0

Crafts 229749.0

DVDs 230274.0

Garden 230073.0

Health and Beauty 229667.0

Men's Clothing 230430.0

Music 230150.0

Pet Supplies 229222.0

Sporting Goods 229932.0

Toys 229964.0

Video Games 230237.0

Women's Clothing 230050.0

Partie avec sans Images Hadoop preinstallés

J’ai cree un autre repertoire cluster avec une autre

mkdir hadoop-cluster

cd hadoop-cluster

mkdir base

mkdir base/config

dans le repertoire BASE, j’ai crée un Fichier DockerFile sans extension

```
FROM ubuntu:20.04

# Avoid interaction during package installs
ENV DEBIAN_FRONTEND=noninteractive

# Install packages
RUN apt-get update && \
  apt-get install -y openjdk-8-jdk wget curl ssh rsync python3 python3-pip nano

# Hadoop & Spark versions
ENV HADOOP_VERSION=3.3.5
```

```
ENV SPARK_VERSION=3.3.3

ENV HADOOP_HOME=/opt/hadoop

ENV SPARK_HOME=/opt/spark

ENV JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64

ENV
PATH=$PATH:$HADOOP_HOME/bin:$HADOOP_HOME/sbin:$SPARK_HOME/bin:$SPARK_
HOME/sbin


# Install Hadoop

RUN wget https://dlcdn.apache.org/hadoop/common/hadoop-
$HADOOP_VERSION/hadoop-$HADOOP_VERSION.tar.gz && \

  tar -xvzf hadoop-$HADOOP_VERSION.tar.gz && \

  mv hadoop-$HADOOP_VERSION /opt/hadoop && \

  rm hadoop-$HADOOP_VERSION.tar.gz


# Install Spark

RUN wget https://dlcdn.apache.org/spark/spark-$SPARK_VERSION/spark-
$SPARK_VERSION-bin-hadoop3.tgz && \

  tar -xvzf spark-$SPARK_VERSION-bin-hadoop3.tgz && \

  mv spark-$SPARK_VERSION-bin-hadoop3 /opt/spark && \

  rm spark-$SPARK_VERSION-bin-hadoop3.tgz


# SSH without password

RUN ssh-keygen -A && \

  mkdir -p /root/.ssh && \

  ssh-keygen -t rsa -P "" -f /root/.ssh/id_rsa && \

  cat /root/.ssh/id_rsa.pub >> /root/.ssh/authorized_keys && \
```

```
chmod 0600 /root/.ssh/authorized_keys

# Copy config files later
COPY config/ /opt/hadoop/etc/hadoop/
COPY hadoop_env.sh /opt/hadoop/etc/hadoop/hadoop-env.sh

# Format HDFS
RUN /opt/hadoop/bin/hdfs namenode -format

CMD ["/bin/bash"]
```

Puis dans le repertoire base, j’ai cree Hadoop_env.sh avec le contenu:

```
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
```

puis, dans le repertoire base/config j’ai cree les fichier xml de configuration Hadoop

core-site.xml

config > core-site.xml

```xml
<configuration>
  <property>
    <name>fs.defaultFS</name>
    <value>hdfs://namenode:9000</value>
  </property>
  <property>
    <name>hadoop.tmp.dir</name>
    <value>/opt/hadoop_tmp</value>
  </property>
  <property>
    <name>io.file.buffer.size</name>
    <value>131072</value>
  </property>
</configuration>
```

hdfs-site

config > hdfs-site.xml

```xml
<configuration>
  <property>
    <name>dfs.replication</name>
    <value>3</value>
  </property>
  <property>
    <name>dfs.namenode.name.dir</name>
    <value>file:///opt/hadoop_tmp/hdfs/namenode</value>
  </property>
  <property>
    <name>dfs.datanode.data.dir</name>
    <value>file:///opt/hadoop_tmp/hdfs/datanode</value>
  </property>
</configuration>
```

mapred-site

config > mapred-site.xml

```xml
<configuration>
  <property>
    <name>mapreduce.framework.name</name>
    <value>yarn</value>
  </property>
</configuration>
```

yarn-site.xml

config > yarn-site.xml

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

Puis, apres les fichiers de configurations etaient faites, j'ai build le cluster

docker build -t hadoop-base:3.3.5-user base/

```
[+] Building 92.0s (6/13)                                                    docker:desktop-linux
 => [internal] load .dockerignore                                                            0.0s
 => => transferring context: 2B                                                              0.0s
 => [1/8] FROM docker.io/library/ubuntu:20.04@sha256:8feb4d8ca5354def3d8fce243717141ce31e2c428701f6682  22.3s
 => => resolve docker.io/library/ubuntu:20.04@sha256:8feb4d8ca5354def3d8fce243717141ce31e2c428701f6682b  0.0s
 => => sha256:13b7e930469f6d3575a320709035c6acf6f5485a76abcf03d1b92a64c09c2476 27.51MB / 27.51MB  20.8s
 => => extracting sha256:13b7e930469f6d3575a320709035c6acf6f5485a76abcf03d1b92a64c09c2476          1.4s
 => [internal] load build context                                                            0.1s
 => => transferring context: 1.11kB                                                          0.0s
 => [2/8] RUN apt-get update &&     apt-get install -y openjdk-8-jdk wget curl ssh rsync python3 pytho  66.8s
```

```
[+] Building 1972.5s (13/14)                                                 docker:desktop-linux
 => [8/8] RUN /opt/hadoop/bin/hdfs namenode -format                                          2.1s
 => exporting to image                                                                      78.1s
 => => exporting layers                                                                     63.2s
 => => exporting manifest sha256:08cd3a01ce24c6dab9af7316faf342eccb0107b5ca94fb8ccd46d895d73ad1ea  0.0s
 => => exporting config sha256:bb94d7b96afc722d48de272407c1d02398f2b33209104362bc18c43f4b3824ee  0.0s
 => => exporting attestation manifest sha256:5d40db21f3a22c504d6e267f77eee2b1189bea7194d8a96a660c0bf71c  0.0s
 => => exporting manifest list sha256:4e5ba6eb9ce5ee6d2d11468e9ff16f9c012b00fc5abf3f1f9aff8e101eba0e98  0.0s
 => => naming to docker.io/library/hadoop-base:3.3.5-user                                    0.0s
 => => unpacking to docker.io/library/hadoop-base:3.3.5-user                                14.8s
```

Apres le docker a ete built , j’ai fait un fichier docker-compose.yml mais cette fois avec spark

```
version: '3'


services:


  namenode:
    image: hadoop-base:3.3.5-User
    container_name: namenode
    ports:
      - "9870:9870"   # HDFS web UI
      - "9000:9000"   # HDFS communication
    volumes:
      - namenode_data:/opt/hadoop_tmp/hdfs/namenode
    environment:
      - NODE_TYPE=namenode
    command: bash -c "/opt/hadoop/bin/hdfs namenode -format && /opt/hadoop/sbin/start-dfs.sh && tail -f /dev/null"


  datanode1:
    image: hadoop-base:3.3.5-User
    container_name: datanode1
    volumes:
      - datanode1_data:/opt/hadoop_tmp/hdfs/datanode
    environment:
      - NODE_TYPE=datanode
```

```
    command: bash -c "/opt/hadoop/sbin/hadoop-daemon.sh start datanode && tail -f /dev/null"

  datanode2:
    image: hadoop-base:3.3.5-User
    container_name: datanode2
    volumes:
      - datanode2_data:/opt/hadoop_tmp/hdfs/datanode
    environment:
      - NODE_TYPE=datanode
    command: bash -c "/opt/hadoop/sbin/hadoop-daemon.sh start datanode && tail -f /dev/null"

  datanode3:
    image: hadoop-base:3.3.5-User
    container_name: datanode3
    volumes:
      - datanode3_data:/opt/hadoop_tmp/hdfs/datanode
    environment:
      - NODE_TYPE=datanode
    command: bash -c "/opt/hadoop/sbin/hadoop-daemon.sh start datanode && tail -f /dev/null"

  resourcemanager:
    image: hadoop-base:3.3.5-User
    container_name: resourcemanager
    ports:
```

```
    - "8088:8088"   # YARN web UI
  command: bash -c "/opt/hadoop/sbin/start-yarn.sh && tail -f /dev/null"

  historyserver:
  image: hadoop-base:3.3.5-User
  container_name: historyserver
  command: bash -c "/opt/hadoop/bin/mapred --daemon start historyserver && tail -f /dev/null"

  spark-master:
  image: hadoop-base:3.3.5-User
  container_name: spark-master
  ports:
    - "8080:8080"   # Spark web UI
    - "7077:7077"   # Spark master
  command: bash -c "$SPARK_HOME/sbin/start-master.sh && tail -f /dev/null"

  spark-worker1:
  image: hadoop-base:3.3.5-User
  container_name: spark-worker1
  command: bash -c "$SPARK_HOME/sbin/start-worker.sh spark://spark-master:7077 && tail -f /dev/null"

  spark-worker2:
  image: hadoop-base:3.3.5-User
  container_name: spark-worker2
```

command: bash -c "$SPARK_HOME/sbin/start-worker.sh spark://spark-master:7077 && tail -f /dev/null"

volumes:

namenode_data:

datanode1_data:

datanode2_data:

datanode3_data:

docker compose up -d

```
+] Running 9/9
✓ Container datanode03       Started
✓ Container datanode01       Started
✓ Container datanode02       Started
✓ Container namenode0        Started
✓ Container spark-master     Started
✓ Container spark-worker2    Started
✓ Container historyserver    Started
✓ Container resourcemanager  Started
✓ Container spark-worker1    Started
S C:\Users\abenh\cluster>
```

PS C:\Users\abenh\cluster> docker ps

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

0925905728a8 hadoop-base:3.3.5-user "bash -c '/opt/hadoo..." 5 minutes ago Up 5 minutes 0.0.0.0:9000->9000/tcp, 0.0.0.0:9870->9870/tcp namenode0

65cf71491021 hadoop-base:3.3.5-user "bash -c '/opt/hadoo..." 5 minutes ago Up 5 minutes

datanode01

472501f4c636 hadoop-base:3.3.5-user "bash -c '/opt/hadoo..." 5 minutes ago Up 5 minutes

datanode03

17aed1bb0f13 hadoop-base:3.3.5-user "bash -c '/opt/hadoo..." 5 minutes ago Up 5 minutes

datanode02

f15977056b02 hadoop-base:3.3.5-user "bash -c '/opt/hadoo..." 8 minutes ago Up 5 minutes

historyserver

toute les datanodes et le namenode + history server marchent

Puis, execute mapper et reducer

mapper.py

C: > Users > abenh > Desktop > mapper.py > ...

```python
import sys
for line in sys.stdin:
    data = line.strip().split("\t")
    if len(data) == 6:
        date, time, store, item, cost, payment = data
        print ("{0}\t1".format(item))
```

Mapper.py

mapper.py reducer.py

C: > Users > abenh > Desktop > reducer.py > ...

```python
import sys
salesTotal = 0
oldKey = None

for line in sys.stdin:
    data_mapped = line.strip().split("\t")
    if len(data_mapped) != 2:
        continue

    thisKey, thisSale = data_mapped
    if oldKey and oldKey != thisKey:
        print (oldKey, "\t", salesTotal)
        oldKey = thisKey;
        salesTotal = 0
    oldKey = thisKey
    salesTotal += float(thisSale)
if oldKey != None:
    print (oldKey, "\t", salesTotal)
```

Reducer.py

Puis, j'ai copié le fichier purchases.txt.gz dans le cluster

docker cp "C:\Users\abenh\Downloads\purchases.txt.gz" namenode:/purchases.txt.gzdocker cp

docker exec -it namenode bash

hdfs dfs -mkdir -p /input

hdfs dfs -put /purchases.txt.gz /input/

et met le mapper.py et reducer.py dans le docker aussi

exit

docker cp "C:\Users\abenh\Downloads\mapper.py" namenode:/mapper.py

docker cp "C:\Users\abenh\Downloads\reducer.py" namenode:/reducer.py

docker exec -it namenode bash

et executer le Hadoop streaming pur que python soit interpreté

hadoop jar $HADOOP_HOME/share/hadoop/tools/lib/hadoop-streaming-*.jar \

-input /input/purchases.txt.gz \

-output /output \

-mapper mapper.py \

-reducer reducer.py

```
        Map-Reduce Framework
                Map input records=4138476
                Map output records=4138476
                Map output bytes=52880666
                Map output materialized bytes=61157630
                Input split bytes=184
                Combine input records=0
                Combine output records=0
                Reduce input groups=18
                Reduce shuffle bytes=61157630
                Reduce input records=4138476
```

root@25e7ca210b92:~# hadoop fs -cat /output-purchases/part-00000

2025-05-13 14:15:46,007 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false

Baby 230293.0

Books 229787.0

CDs 230039.0

Cameras 229320.0

Children's Clothing 230469.0

Computers 229059.0

Consumer Electronics 229761.0

Crafts 229749.0

DVDs 230274.0

Garden 230073.0

Health and Beauty 229667.0

Men's Clothing 230430.0

Music 230150.0

Pet Supplies 229222.0

Sporting Goods 229932.0

Toys 229964.0

Video Games 230237.0

Women's Clothing 230050.0

- ⇨ Meme resultat par les deux methodes de map reduce mais il y’a gaspillage de stockage sans images preinstalles(jusqu’a 100GO sans telecharger des fichiers)

Partie II: SPARK

Pour la partie spark, j'ai decide de travailler avec le cluster predefine de la premiere partie puisque c'est plus simple ,

J'ai ajouté cela au dessous de docker-compose.yaml

2 worker nodes et 1 master node

```
  - hadoop


 spark-worker:
  image: bde2020/spark-worker:2.4.5-hadoop2.7
  container_name: spark-worker
  environment:
   - SPARK_MASTER=spark://spark-master:7077
  depends_on:
   - spark-master
  ports:
   - "8081:8081" # Spark worker UI


  networks:
   - hadoop
spark-master:
  image: bde2020/spark-master:2.4.5-hadoop2.7
```

```
  container_name: spark-master
  environment:
    - INIT_DAEMON_STEP=setup_spark
  ports:
    - "8080:8080" # Spark Web UI
    - "7077:7077" # Spark cluster port
  networks:
    - hadoop

spark-worker1:
  image: bde2020/spark-worker:2.4.5-hadoop2.7
  container_name: spark-worker1
  environment:
    - SPARK_MASTER=spark://spark-master:7077
  depends_on:
    - spark-master
  ports:
    - "8081:8081" # UI for worker1
  networks:
    - hadoop

spark-worker2:
  image: bde2020/spark-worker:2.4.5-hadoop2.7
  container_name: spark-worker2
  environment:
    - SPARK_MASTER=spark://spark-master:7077
```

depends_on:

- spark-master

ports:

- "8082:8081" # UI for worker2 (local port 8082)

networks:

- hadoop

Docker compose up -d

```
time="2025-05-14T14:41:12+01:00" level=warning msg="C:\\Users\\abenh\\hadoop-cluster\\docker-compose.yml: the attribute `ver
sion` is obsolete, it will be ignored, please remove it to avoid potential confusion"
[+] Running 0/7
 - spark-worker1 Pulling                                                                                                  4.4s
 - datanode1 Pulling                                                                                                      4.4s
 - datanode2 Pulling                                                                                                      4.4s
 - spark-master Pulling                                                                                                   4.4s
 - namenode Pulling                                                                                                       4.4s
 - spark-worker2 Pulling                                                                                                  4.4s
 - datanode3 Pulling                                                                                                      4.4s
```

Docker ps

```
PS C:\Users\abenh\hadoop-cluster> docker ps
CONTAINER ID   IMAGE                                          COMMAND                  CREATED          STATUS
                PORTS                                                   NAMES
0edf637855c4   bde2020/hadoop-datanode:2.0.0-hadoop3.2.1-java8   "/entrypoint.sh /run…"   9 seconds ago    Up 5 seconds (hea
lth: starting)   0.0.0.0:9866->9864/tcp                                datanode3
a58920e97044   bde2020/hadoop-datanode:2.0.0-hadoop3.2.1-java8   "/entrypoint.sh /run…"   9 seconds ago    Up 5 seconds (hea
lth: starting)   0.0.0.0:9865->9864/tcp                                datanode2
7df4998fa59c   bde2020/hadoop-datanode:2.0.0-hadoop3.2.1-java8   "/entrypoint.sh /run…"   9 seconds ago    Up 5 seconds (hea
lth: starting)   0.0.0.0:9864->9864/tcp                                datanode1
1900719bebcb   bde2020/spark-worker:2.4.5-hadoop2.7              "/bin/bash /worker.sh"   9 seconds ago    Up 5 seconds
                0.0.0.0:8081->8081/tcp                                  spark-worker1
```

Dans le spark master,

Docker exec -it spark-master bash

Installer pyspark dans le spark-master

Pip3 install pyspark (340 Mega)

```
ash-5.0# pip3 install pyspark
ollecting pyspark
 Downloading https://files.pythonhosted.org/packages/46/6c/48fdd43909f6f0425b068cb676c7510694a56a323a83a696f429760f598c/pys
ark-3.4.4.tar.gz (311.4MB)
    |                                | 287.4MB 925kB/s eta 0:00:26
```

Et executer le pyspark

pyspark --master spark://spark-master:7077 --conf spark.hadoop.fs.defaultFS=hdfs://namenode:9000

```
 warnings.warn("Python 3.7 support is deprecated in Spark 3.4.", FutureWarning)
elcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 3.4.4
      /_/

sing Python version 3.7.5 (default, Oct 17 2019 12:25:15)
park context Web UI available at http://d86dafe83300:4040
park context available as 'sc' (master = spark://spark-master:7077, app id = spark-application-1747231408491).
```

Puis, j'ai telecharge le Purchases.txt de hdfs et le met dans un RDD spark (resilient distributed dataset) apres verifier que purchases existe dans notre hdfs

rdd = sc.textFile("hdfs://namenode:9000/input/purchases.txt.gz")

rdd.take(5)

```
"2012-01-01\t09:00\tSan Jose\tMen's Clothing\t214.05\tAmex", "2012-01-01\t09:00\tFort Worth\tWomen's Clothing\t153.57\tVisa
, '2012-01-01\t09:00\tSan Diego\tMusic\t66.08\tCash', '2012-01-01\t09:00\tPittsburgh\tPet Supplies\t493.51\tDiscover', "201
-01-01\t09:00\tOmaha\tChildren's Clothing\t235.63\tMasterCard", "2012-01-01\t09:00\tStockton\tMen's Clothing\t247.18\tMaste
Card", '2012-01-01\t09:00\tAustin\tCameras\t379.6\tVisa', '2012-01-01\t09:00\tNew York\tConsumer Electronics\t296.8\tCash',
'2012-01-01\t09:00\tCorpus Christi\tToys\t25.38\tDiscover', '2012-01-01\t09:00\tFort Worth\tToys\t213.88\tVisa']
>>
```

Il a affiche les 10 premier lignes de mon rdd

Analyse de Notre RDD :

Affichage de nombre total de records

total_records = rdd.count()

print(f”total records:{total_records)”)

```
>>> total_records = rdd.count()
>>> print(f"Total records: {total_records}")
Total records: 4138476
>>>
```

Total records: 4138476 donc on a 4138476 lignes

Nettoyage des données (Séparation en colonnes)

Les enregistrements sont séparés par des tabulations.je sépare chaque ligne en colonnes pour faciliter l'analyse.

split_rdd = rdd.map(lambda line: line.split('\t'))

Afficher les premières lignes après séparation

split_rdd.take(5)

```
[['2012-01-01', '09:00', 'San Jose', "Men's Clothing", '214.05', 'Amex'], ['2012-01-01', '09:00', 'Fort Worth', "Women's Clo
thing", '153.57', 'Visa'], ['2012-01-01', '09:00', 'San Diego', 'Music', '66.08', 'Cash'], ['2012-01-01', '09:00', 'Pittsbur
gh', 'Pet Supplies', '493.51', 'Discover'], ['2012-01-01', '09:00', 'Omaha', "Children's Clothing", '235.63', 'MasterCard']]
>>>
```

[['2012-01-01', '09:00', 'San Jose', "Men's Clothing", '214.05', 'Amex'], ['2012-01-01', '09:00', 'Fort Worth', "Women's Clothing", '153.57', 'Visa'], ['2012-01-01', '09:00', 'San Diego', 'Music', '66.08', 'Cash'], ['2012-01-01', '09:00', 'Pittsburgh', 'Pet Supplies', '493.51', 'Discover'], ['2012-01-01', '09:00', 'Omaha', "Children's Clothing", '235.63', 'MasterCard']]

Maintenant faire la meme operation que le map reduce job du premiere partie mais en spark en utulisant notre split_rdd

```
item_counts = split_rdd.filter(lambda cols: len(cols) == 6).map(lambda cols: (cols[3], 1))

result = item_counts.reduceByKey(lambda a, b: a + b)

for item, count in result.collect():

   print(f"{item}\t{count}")
```

```
...
[Stage 3:>                                                          (0 + 1) / 1]
Men's Clothing  230430
Women's Clothing        230050
Music   230150
Pet Supplies    229222
Children's Clothing     230469
Cameras 229320
Consumer Electronics    229761
Toys    229964
Video Games     230237
```

[Stage 3:> (0 + 1) / 1]

Men's Clothing 230430

Women's Clothing 230050

Music 230150

Pet Supplies 229222

Children's Clothing 230469

Cameras 229320

Consumer Electronics 229761

Toys 229964

Video Games 230237

DVDs 230274

Garden 230073

Baby 230293

Books 229787

Crafts 229749

Sporting Goods 229932

CDs 230039

Computers 229059

Health and Beauty 229667

3. **Calcul du total des ventes par catégorie**

```
category_rdd = split_rdd.filter(lambda x: x[3] == "Men's Clothing")

# Convertir les ventes en float et calculer le total des ventes
total_sales = category_rdd.map(lambda x: float(x[4])).sum()

print(f"Total des ventes pour 'Men's Clothing': {total_sales}")
```

- ⇨ Total des ventes pour 'Men's Clothing': 57621279.04000138

4. **Trouver les achats par carte de credit**

```
# Filtrer les achats par carte de crédit
visa_rdd = split_rdd.filter(lambda x: x[5] == "Visa")

# Afficher les achats par Visa
visa_rdd.take(5)
```

- ⇨ [['2012-01-01', '09:00', 'Fort Worth', "Women's Clothing", '153.57', 'Visa'], ['2012-01-01', '09:00', 'Austin', 'Cameras', '379.6', 'Visa'], ['2012-01-01', '09:00', 'Fort Worth', 'Toys', '213.88', 'Visa'], ['2012-01-01', '09:00', 'Las Vegas', 'Video Games', '53.26', 'Visa'], ['2012-01-01', '09:00', 'Lincoln', 'Garden', '136.9', 'Visa']]

5. **Comptage des achats par ville**

Compter les achats par ville

city_counts = split_rdd.map(lambda x: (x[2], 1)).reduceByKey(lambda x, y: x + y)

Afficher les 5 premières villes avec le plus d'achats

city_counts.takeOrdered(5, key=lambda x: -x[1])

[('Philadelphia', 40748), ('Newark', 40577), ('Sacramento', 40561), ('Charlotte', 40509), ('Washington', 40503)]

6. Analyser les ventes par période

Extraire la date et l'heure pour une analyse par période

time_rdd = split_rdd.map(lambda x: (x[0], float(x[4])))

Calculer les ventes totales par date

daily_sales = time_rdd.reduceByKey(lambda x, y: x + y)

Afficher les ventes totales pour chaque date

daily_sales.take(5)

```
>>> # Afficher les ventes totales pour chaque date
... daily_sales.take(5)
[Stage 9:>                                                          (0 + 1) / 1]
[('2012-01-01', 2838071.4400000027), ('2012-01-02', 2808888.2900000005), ('2012-01-03', 2877953.3099999856), ('2012-01-04',
2829499.0200000056), ('2012-01-05', 2838836.420000009)]
>>>
```

=> [('2012-01-01', 2838071.4400000027), ('2012-01-02', 2808888.2900000005), ('2012-01-03', 2877953.3099999856), ('2012-01-04', 2829499.0200000056), ('2012-01-05', 2838836.420000009)]

7. Calcul des moyennes des achats par catégorie

Calculer la moyenne des achats pour chaque catégorie

category_sales = split_rdd.map(lambda x: (x[3], (float(x[4]), 1))) # (catégorie, (ventes, comptage))

category_avg_sales = category_sales.reduceByKey(lambda x, y: (x[0] + y[0], x[1] + y[1])) \

.mapValues(lambda x: x[0] / x[1])

Afficher la moyenne des ventes par catégorie

category_avg_sales.take(5)

```
... category_avg_sales.take(5)
[Stage 11:>                                                         (0 + 1) / 1]
[("Men's Clothing", 250.0597970750396), ("Women's Clothing", 249.6607214518531), ('Music', 249.81746556593728), ('Pet Suppli
es', 249.52775143747144), ("Children's Clothing", 250.03285014471038)]
>>>
>>>
```

- ⇨ [("Men's Clothing", 250.0597970750396), ("Women's Clothing", 249.6607214518531), ('Music', 249.81746556593728), ('Pet Supplies', 249.52775143747144), ("Children's Clothing", 250.03285014471038)]

8. Trouver les achats les plus élevés

Trouver les achats les plus élevés

max_purchase = split_rdd.map(lambda x: float(x[4])).max()

print(f"Le montant d'achat le plus élevé est: {max_purchase}")

- ⇨ Le montant d'achat le plus élevé est: 499.99

9. Écrire les résultats dans un fichier

category_avg_sales.saveAsTextFile("hdfs://namenode:9000/output/category_avg_sales.txt")