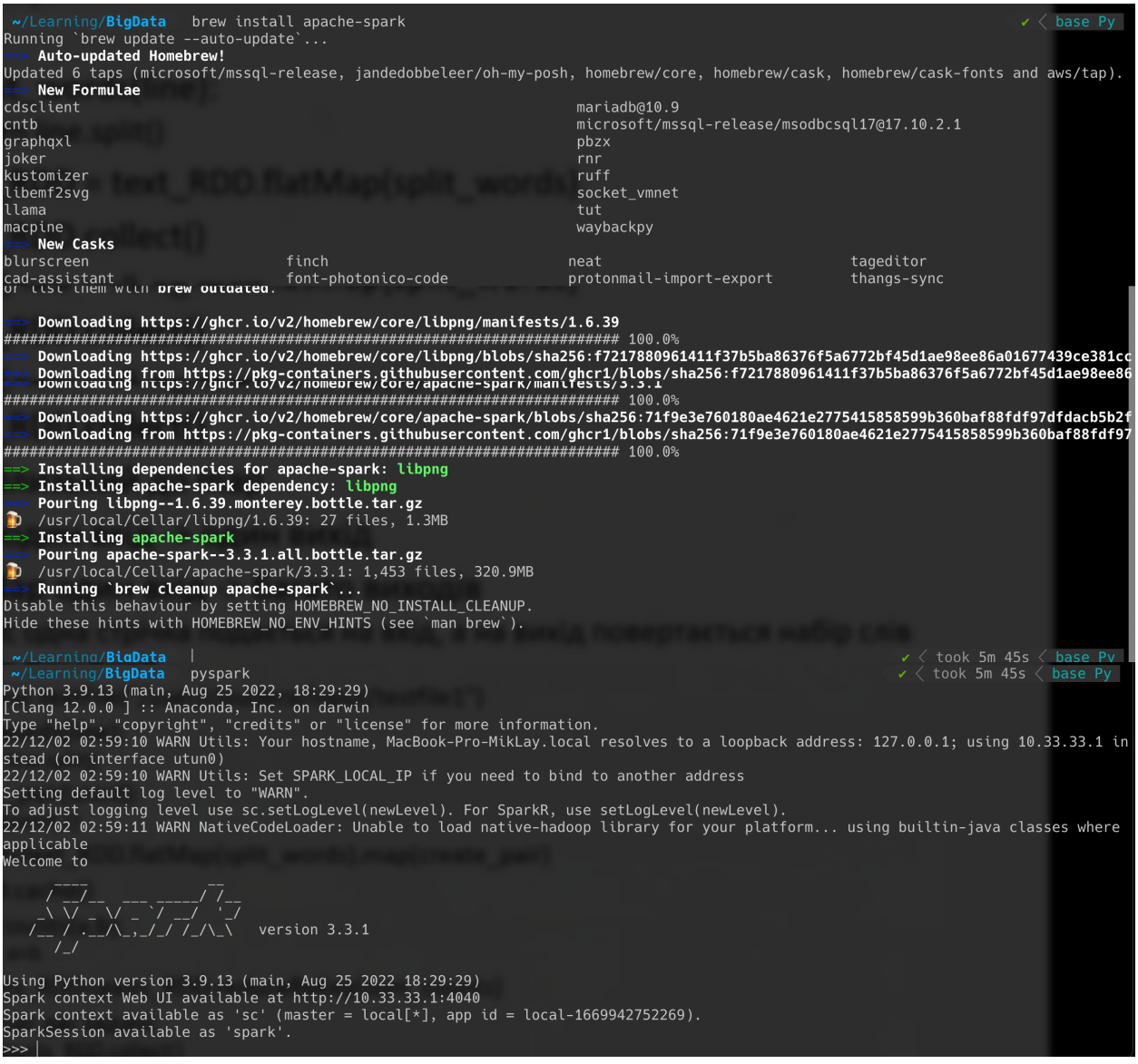
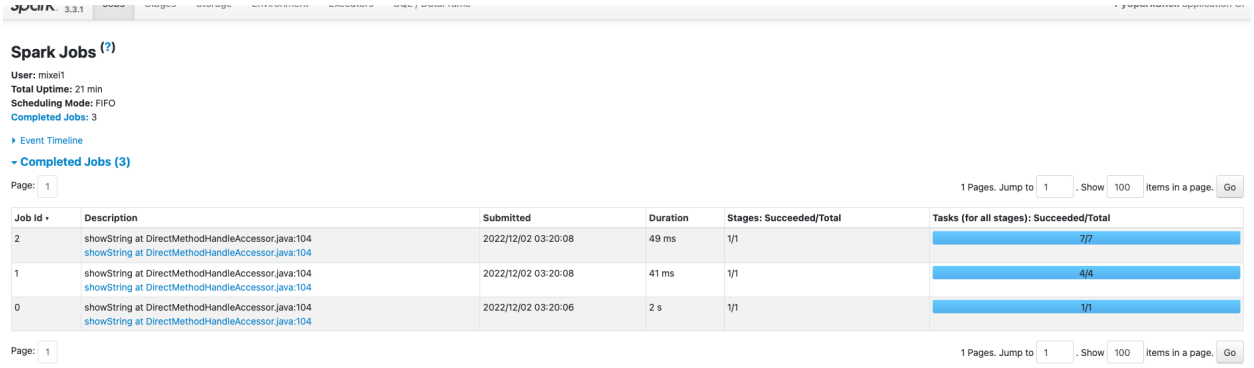
Звіт №5

Загалом враження після роботи зі PySpark хороші, я б сказав навіть чудові. Використання Spark неймовірно полегшело написання коду та й загалом імплементацію кроків map/reduce. Цікаво буде реалізувати множення матриць на Spark й порівняти.

Етап встановлення PySpark через brew:



Spark Web UI



Spark history server

Спочатку треба було створити файл конфігурації в середині Cellars

```
/usr/lo/Cellar/apache-spark/3.3.1/libexec/conf   ls                                                    ✔ 〈 base Py
fairscheduler.xml.template   metrics.properties.template  spark-env.sh.template
log4j2.properties.template   spark-defaults.conf.template workers.template

/usr/lo/Cellar/apache-spark/3.3.1/libexec/conf   webstorm spark-defaults.conf.template               ✔ 〈 base Py

/usr/lo/Cellar/apache-spark/3.3.1/libexec/conf   sudo cp spark-defaults.conf.template spark-defaults.conf   ✔ 〈 base Py
Password:

/usr/lo/Ce/ap/3.3.1/libexec/conf   webstorm spark-defaults.conf                          ✔ 〈 took 4s 〈 base Py

/usr/lo/Cellar/apache-spark/3.3.1/libexec/conf   |                                                    ✔ 〈 base Py


#
# Licensed to the Apache Software Foundation (ASF) under one or more
# contributor license agreements.  See the NOTICE file distributed with
# this work for additional information regarding copyright ownership.
# The ASF licenses this file to You under the Apache License, Version 2.0
# (the "License"); you may not use this file except in compliance with
# the License.  You may obtain a copy of the License at
#
#    http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.
#

# Default system properties included when running spark-submit.
# This is useful for setting default environmental settings.

# Example:
# spark.master                     spark://master:7077
spark.eventLog.enabled           true
spark.history.fs.logDirectory file:///Users/mixei1/Learning/BigData/logs/
# spark.eventLog.dir               hdfs://namenode:8021/directory
# spark.serializer                 org.apache.spark.serializer.KryoSerializer
# spark.driver.memory              5g
# spark.executor.extraJavaOptions  -XX:+PrintGCDetails -Dkey=value -Dnumbers="one two three"
File Name to write : ../conf/spark-defaults.conf
^G Get Help           ^T  To Files
^C Cancel             TAB Complete
```

```
⌀ /  ls                                                                                    ✔ 〈 base Py
Applications System       Volumes      cores        etc          opt          sbin         usr
Library      Users        bin          dev          home         private      tmp          var

⌀ /  cd tmp                                                                                ✔ 〈 base Py

/tmp  ls                                                                                   ✔ 〈 base Py
CrashUpload-izwnd                                  mysqlx.sock
com.apple.launchd.VOrNwjA6QP                      mysqlx.sock.lock
dumps                                              powerlog
mysql.sock                                         spark-mixei1-org.apache.spark.deploy.history.HistoryServer-1.pid
mysql.sock.lock                                    steam.pipe

/tmp  mkdir spark-events                                                                   ✔ 〈 base Py

/tmp                                                                                       ✔ 〈 base Py
```

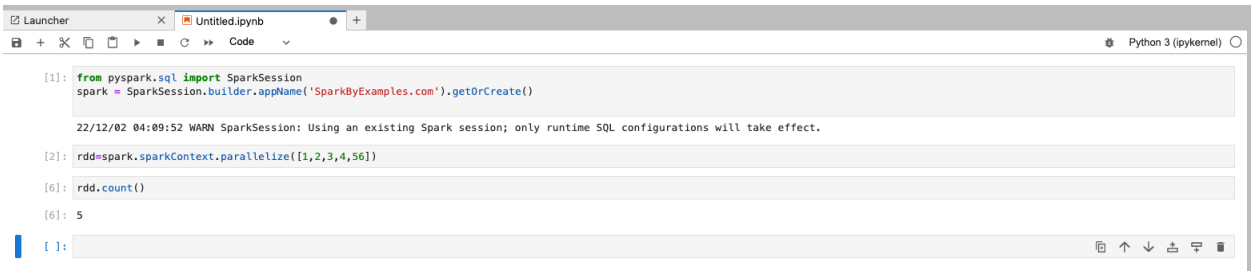
Далі заранив Spark history server
Спочатку пробував через конфіг налаштувати, а далі щоб не витрачати часу просто створив папку до котрої він звертається по замовчуванню

```
/usr/lo/Ce/ap/3.3.1/libexec/conf  cd ../sbin/                                  ✔ 〈 took 2m 28s 〈 base Py

/usr/lo/Cellar/apache-spark/3.3.1/libexec/sbin  ls                                         ✔ 〈 base Py
decommission-slave.sh          start-history-server.sh        start-worker.sh              stop-slave.sh
decommission-worker.sh         start-master.sh                start-workers.sh             stop-slaves.sh
slaves.sh                      start-mesos-dispatcher.sh      stop-all.sh                  stop-thriftserver.sh
spark-config.sh                start-mesos-shuffle-service.sh stop-history-server.sh       stop-worker.sh
spark-daemon.sh                start-slave.sh                 stop-master.sh               stop-workers.sh
spark-daemons.sh               start-slaves.sh                stop-mesos-dispatcher.sh     workers.sh
start-all.sh                   start-thriftserver.sh          stop-mesos-shuffle-service.sh

/usr/lo/Cellar/apache-spark/3.3.1/libexec/sbin  start-history-server.sh                    ✔ 〈 base Py
zsh: command not found: start-history-server.sh

/usr/lo/Ce/apache-spark/3.3.1/libexec/sbin  bash start-history-server.sh                127 ✘ 〈 base Py
starting org.apache.spark.deploy.history.HistoryServer, logging to /usr/local/Cellar/apache-spark/3.3.1/libexec/logs/spark-mixei1-or
g.apache.spark.deploy.history.HistoryServer-1-MacBook-Pro-MikLay.local.out
```

Spyder IDE & Jupyter Notebook

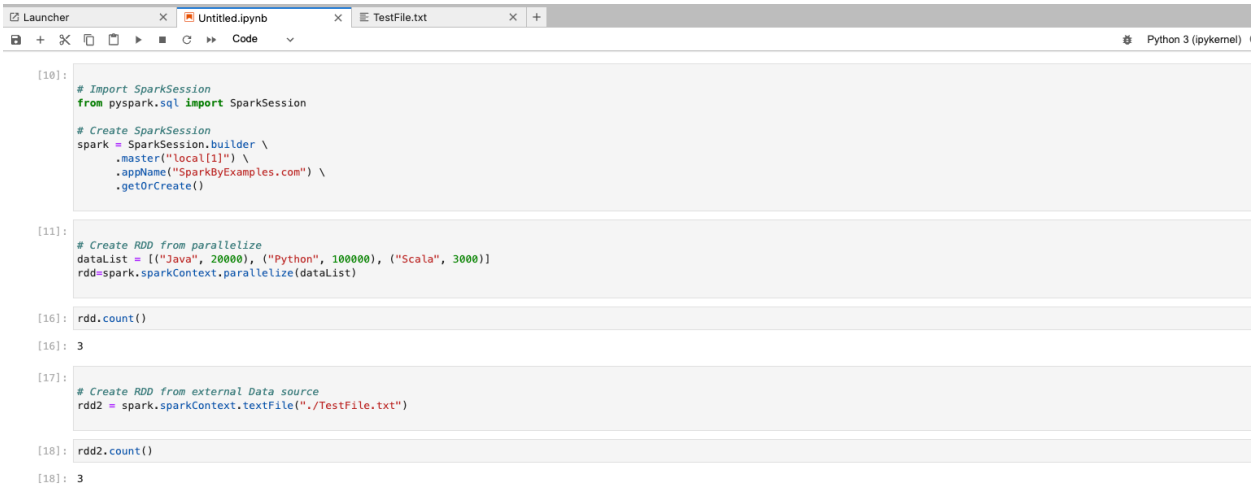
Довелося погратись з налаштуваннями

```
~  export PYSPARK_DRIVER_PYTHON=jupyter                                    ✔ 〈 took 2m 21s 〈 base Py

~  export PYSPARK_DRIVER_PYTHON_OPTS='lab'                                 ✔ 〈 base Py

~  pyspark                                                                 ✔ 〈 base Py
Note: NumExpr detected 12 cores but "NUMEXPR_MAX_THREADS" not set, so enforcing safe limit of 8.
NumExpr defaulting to 8 threads.
[I 2022-12-02 03:51:23.482 ServerApp] jupyterlab | extension was successfully linked.
[I 2022-12-02 03:51:23.496 ServerApp] Writing Jupyter server cookie secret to /Users/mixei1/Library/Jupyter/runtime/jupyter_cookie_s
ecret
[I 2022-12-02 03:51:23.685 ServerApp] nbclassic | extension was successfully linked.
[I 2022-12-02 03:51:23.685 ServerApp] panel.io.jupyter_server_extension | extension was successfully linked.
[I 2022-12-02 03:51:23.763 ServerApp] nbclassic | extension was successfully loaded.
[I 2022-12-02 03:51:23.764 LabApp] JupyterLab extension loaded from /Users/mixei1/opt/anaconda3/lib/python3.9/site-packages/jupyterl
ab
[I 2022-12-02 03:51:23.764 LabApp] JupyterLab application directory is /Users/mixei1/opt/anaconda3/share/jupyter/lab
[I 2022-12-02 03:51:23.769 ServerApp] jupyterlab | extension was successfully loaded.
[I 2022-12-02 03:51:23.770 ServerApp] panel.io.jupyter_server_extension | extension was successfully loaded.
[I 2022-12-02 03:51:23.771 ServerApp] Serving notebooks from local directory: /Users/mixei1
[I 2022-12-02 03:51:23.772 ServerApp] Jupyter Server 1.18.1 is running at:
[I 2022-12-02 03:51:23.772 ServerApp] http://localhost:8888/lab?token=c52ea4a04452db05192d89d4d6248f933a5977cd33b63717
[I 2022-12-02 03:51:23.772 ServerApp]  or http://127.0.0.1:8888/lab?token=c52ea4a04452db05192d89d4d6248f933a5977cd33b63717
[I 2022-12-02 03:51:23.772 ServerApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 2022-12-02 03:51:23.780 ServerApp]

    To access the server, open this file in a browser:
        file:///Users/mixei1/Library/Jupyter/runtime/jpserver-51112-open.html
    Or copy and paste one of these URLs:
        http://localhost:8888/lab?token=c52ea4a04452db05192d89d4d6248f933a5977cd33b63717
     or http://127.0.0.1:8888/lab?token=c52ea4a04452db05192d89d4d6248f933a5977cd33b63717
[I 2022-12-02 03:51:27.940 LabApp] Build is up to date
```

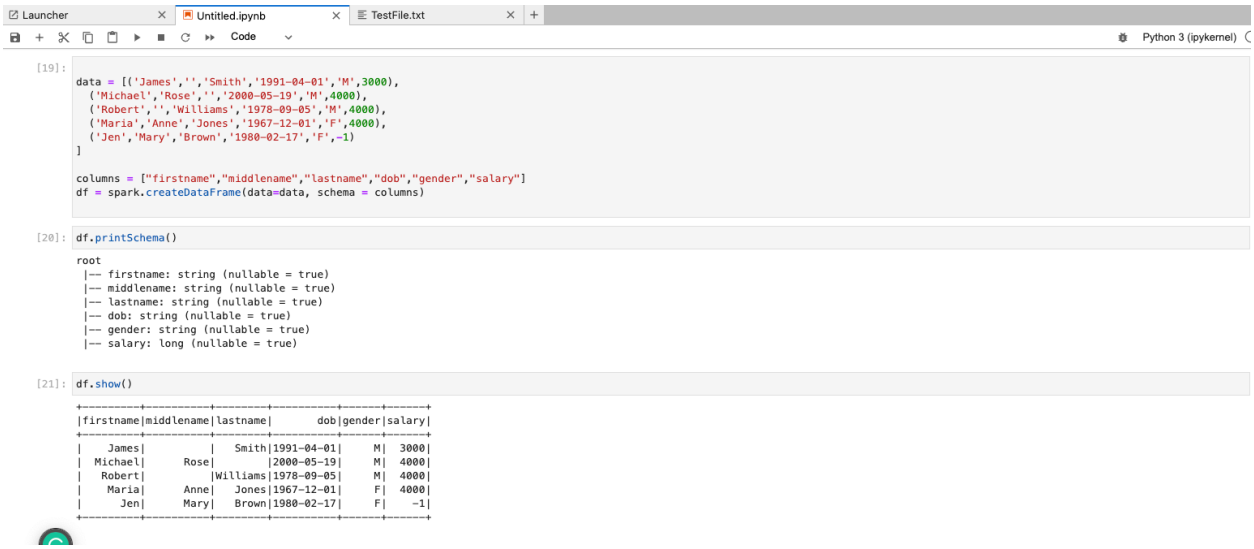


PySpark RDD – Resilient Distributed Dataset

RDD Creation



PySpark DataFrame



```
[28]: df = spark.read.csv("./TestFile.csv")
      df.printSchema()

root
 |-- _c0: string (nullable = true)
 |-- _c1: string (nullable = true)
```

```
[29]: df.show()

+---------+-------------+
|      _c0|          _c1|
+---------+-------------+
|     name|      surname|
| TestName|  TestSurname|
|TestName1| TestSurname1|
+---------+-------------+
```

PySpark SQL

```
[37]: df.createOrReplaceTempView("PERSON_DATA")
      df2 = spark.sql("SELECT * from PERSON_DATA")
      df2.printSchema()
      df2.show()

root
 |-- firstname: string (nullable = true)
 |-- middlename: string (nullable = true)
 |-- lastname: string (nullable = true)
 |-- dob: string (nullable = true)
 |-- gender: string (nullable = true)
 |-- salary: long (nullable = true)

+---------+----------+--------+----------+------+------+
|firstname|middlename|lastname|       dob|gender|salary|
+---------+----------+--------+----------+------+------+
|    James|          |   Smith|1991-04-01|     M|  3000|
|  Michael|      Rose|        |2000-05-19|     M|  4000|
|   Robert|          |Williams|1978-09-05|     M|  4000|
|    Maria|      Anne|   Jones|1967-12-01|     F|  4000|
|      Jen|      Mary|   Brown|1980-02-17|     F|    -1|
+---------+----------+--------+----------+------+------+
```

```
[38]: groupDF = spark.sql("SELECT gender, count(*) from PERSON_DATA group by gender")
      groupDF.show()

+------+--------+
|gender|count(1)|
+------+--------+
|     M|       3|
|     F|       2|
+------+--------+
```

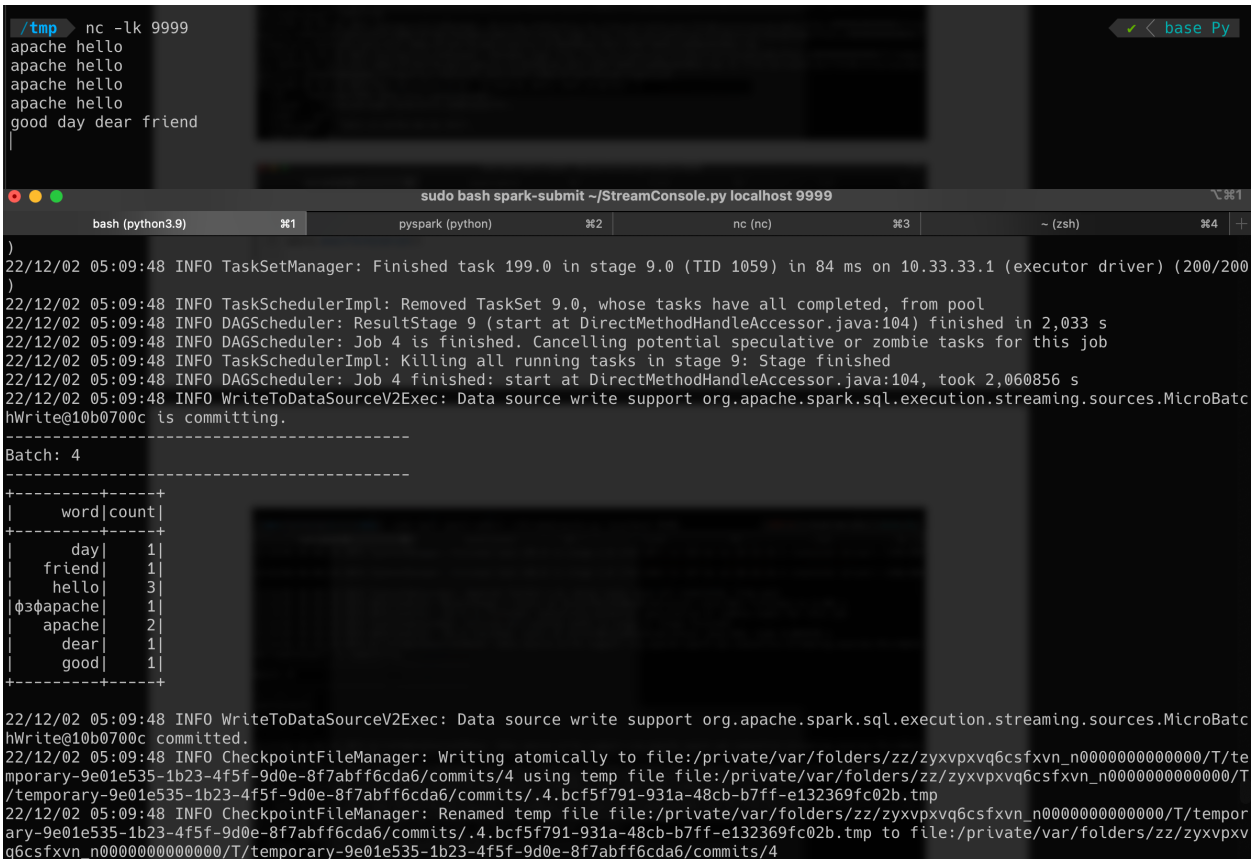
PySpark Streaming

Я знайшов хороший та більш зрозумілий приклад використання: **https://spark.apache.org/docs/2.2.0/structured-streaming-programming-guide.html**
Налаштував під свої потреби:

Untitled.ipynb | StreamConsole.py | TestFile.csv

```python
from pyspark.sql import SparkSession
from pyspark.sql.functions import explode
from pyspark.sql.functions import split

spark = SparkSession \
    .builder \
    .appName("StructuredNetworkWordCount") \
    .getOrCreate()

# Create DataFrame representing the stream of input lines from connection to localhost:9999
lines = spark \
    .readStream \
    .format("socket") \
    .option("host", "localhost") \
    .option("port", 9999) \
    .load()

# Split the lines into words
words = lines.select(
   explode(
       split(lines.value, " ")
   ).alias("word")
)

# Generate running word count
wordCounts = words.groupBy("word").count()

# Start running the query that prints the running counts to the console
query = wordCounts \
    .writeStream \
    .outputMode("complete") \
    .format("console") \
    .start()

query.awaitTermination()
```

```
/usr/lo/Ce/ap/3.3.1/bin  sudo bash spark-submit ~/StreamConsole.py localhost 9999   INT ✘  took 1m 12s  base Py
bash (python3.9) ⌘1 | pyspark (python) ⌘2 | nc (nc) ⌘3 | ~ (zsh) ⌘4
22/12/02 05:08:36 INFO TaskSetManager: Finished task 195.0 in stage 1.0 (TID 207) in 155 ms on 10.33.33.1 (executor driver) (199/200
)
22/12/02 05:08:36 INFO TaskSetManager: Finished task 198.0 in stage 1.0 (TID 210) in 127 ms on 10.33.33.1 (executor driver) (200/200
)
22/12/02 05:08:36 INFO TaskSchedulerImpl: Removed TaskSet 1.0, whose tasks have all completed, from pool
22/12/02 05:08:36 INFO DAGScheduler: ResultStage 1 (start at DirectMethodHandleAccessor.java:104) finished in 3,401 s
22/12/02 05:08:36 INFO DAGScheduler: Job 0 is finished. Cancelling potential speculative or zombie tasks for this job
22/12/02 05:08:36 INFO TaskSchedulerImpl: Killing all running tasks in stage 1: Stage finished
22/12/02 05:08:36 INFO DAGScheduler: Job 0 finished: start at DirectMethodHandleAccessor.java:104, took 3,881578 s
22/12/02 05:08:36 INFO WriteToDataSourceV2Exec: Data source write support org.apache.spark.sql.execution.streaming.sources.MicroBatc
hWrite@7e31aaff is committing.
-------------------------------------------
Batch: 0
-------------------------------------------
+----+-----+
|word|count|
+----+-----+
+----+-----+

22/12/02 05:08:36 INFO WriteToDataSourceV2Exec: Data source write support org.apache.spark.sql.execution.streaming.sources.MicroBatc
hWrite@7e31aaff committed.
22/12/02 05:08:36 INFO CheckpointFileManager: Writing atomically to file:/private/var/folders/zz/zyxvpxvq6csfxvn_n0000000000000/T/te
mporary-9e01e535-1b23-4f5f-9d0e-8f7abff6cda6/commits/0 using temp file file:/private/var/folders/zz/zyxvpxvq6csfxvn_n0000000000000/T
/temporary-9e01e535-1b23-4f5f-9d0e-8f7abff6cda6/commits/.0.99290cde-2027-438f-8020-2a906eb44d6d.tmp
22/12/02 05:08:36 INFO CheckpointFileManager: Renamed temp file file:/private/var/folders/zz/zyxvpxvq6csfxvn_n0000000000000/T/tempor
ary-9e01e535-1b23-4f5f-9d0e-8f7abff6cda6/commits/.0.99290cde-2027-438f-8020-2a906eb44d6d.tmp to file:/private/var/folders/zz/zyxvpxv
q6csfxvn_n0000000000000/T/temporary-9e01e535-1b23-4f5f-9d0e-8f7abff6cda6/commits/0
22/12/02 05:08:36 INFO MicroBatchExecution: Streaming query made progress: {
  "id" : "3add1baa-599c-4649-9711-5d1bf436c602",
  "runId" : "f53a11e0-8a0b-4570-8773-c09053a56f75",
  "name" : null,
  "timestamp" : "2022-12-02T03:08:30.707Z",
  "batchId" : 0,
```


/tmp nc -lk 9999
apache hello
apache hello
apache hello
apache hello
good day dear friend
base Py
sudo bash spark-submit ~/StreamConsole.py localhost 9999
bash (python3.9) ⌘1
pyspark (python) ⌘2
nc (nc) ⌘3
~ (zsh) ⌘4
)
22/12/02 05:09:48 INFO TaskSetManager: Finished task 199.0 in stage 9.0 (TID 1059) in 84 ms on 10.33.33.1 (executor driver) (200/200
)
22/12/02 05:09:48 INFO TaskSchedulerImpl: Removed TaskSet 9.0, whose tasks have all completed, from pool
22/12/02 05:09:48 INFO DAGScheduler: ResultStage 9 (start at DirectMethodHandleAccessor.java:104) finished in 2,033 s
22/12/02 05:09:48 INFO DAGScheduler: Job 4 is finished. Cancelling potential speculative or zombie tasks for this job
22/12/02 05:09:48 INFO TaskSchedulerImpl: Killing all running tasks in stage 9: Stage finished
22/12/02 05:09:48 INFO DAGScheduler: Job 4 finished: start at DirectMethodHandleAccessor.java:104, took 2,060856 s
22/12/02 05:09:48 INFO WriteToDataSourceV2Exec: Data source write support org.apache.spark.sql.execution.streaming.sources.MicroBatc
hWrite@10b0700c is committing.

Batch: 4

+---------+-----+
| word|count|
+---------+-----+
day	1
friend	1
hello	3
ϕɜϕapache	1
apache	2
dear	1
good	1
+---------+-----+
22/12/02 05:09:48 INFO WriteToDataSourceV2Exec: Data source write support org.apache.spark.sql.execution.streaming.sources.MicroBatc
hWrite@10b0700c committed.
22/12/02 05:09:48 INFO CheckpointFileManager: Writing atomically to file:/private/var/folders/zz/zyxvpxvq6csfxvn_n0000000000000/T/te
mporary-9e01e535-1b23-4f5f-9d0e-8f7abff6cda6/commits/4 using temp file file:/private/var/folders/zz/zyxvpxvq6csfxvn_n0000000000000/T
/temporary-9e01e535-1b23-4f5f-9d0e-8f7abff6cda6/commits/.4.bcf5f791-931a-48cb-b7ff-e132369fc02b.tmp
22/12/02 05:09:48 INFO CheckpointFileManager: Renamed temp file file:/private/var/folders/zz/zyxvpxvq6csfxvn_n0000000000000/T/tempor
ary-9e01e535-1b23-4f5f-9d0e-8f7abff6cda6/commits/.4.bcf5f791-931a-48cb-b7ff-e132369fc02b.tmp to file:/private/var/folders/zz/zyxvpxv
q6csfxvn_n0000000000000/T/temporary-9e01e535-1b23-4f5f-9d0e-8f7abff6cda6/commits/4