Week 5 Homework: Batch Processing

In this homework we'll put what we learned about Spark in practice.

For this homework we will be using the FHVHV 2021-06 data found here. FHVHV Data

Code can be found by clicking here

Question 1:

Install Spark and PySpark

- Install Spark
- Run PySpark
- Create a local spark session
- **Execute spark.version.**

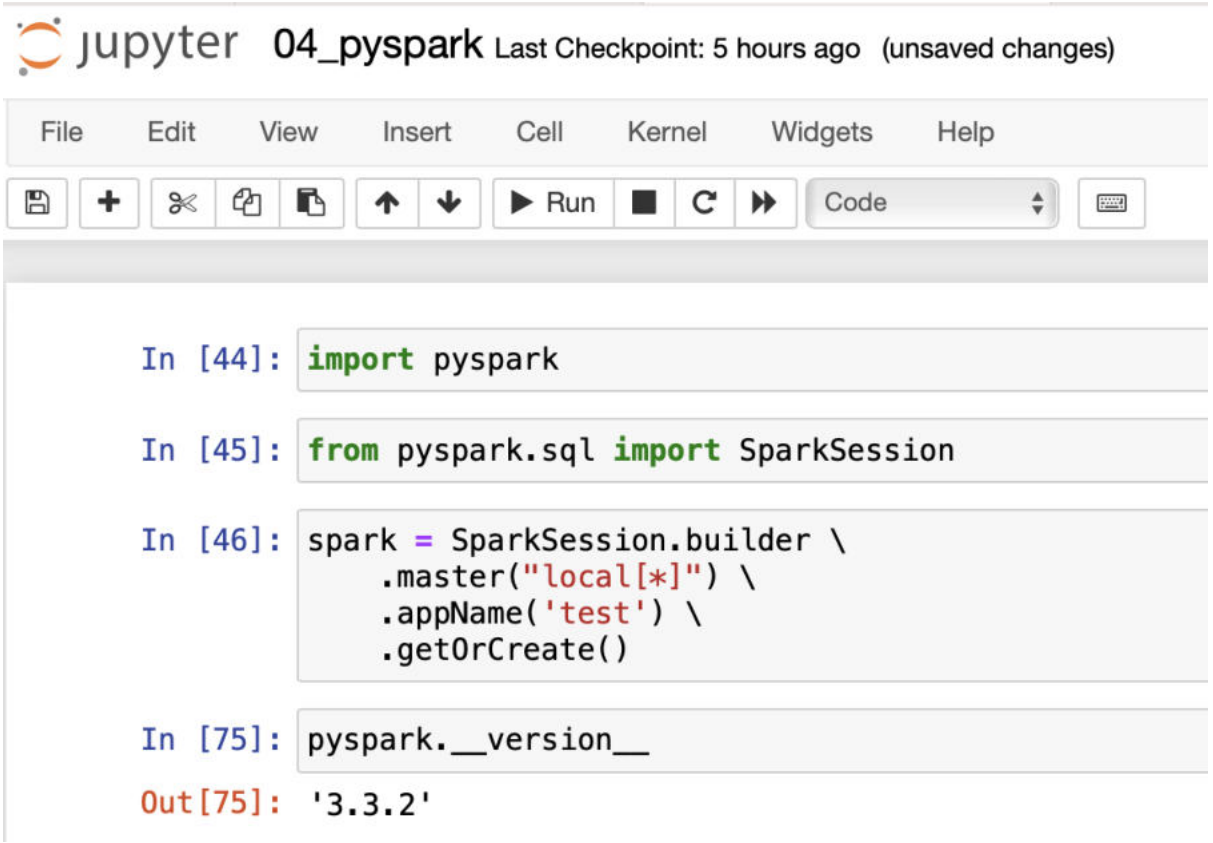
What's the output?

- **3.3.2**
- 2.1.4
- 1.2.3
- 5.4

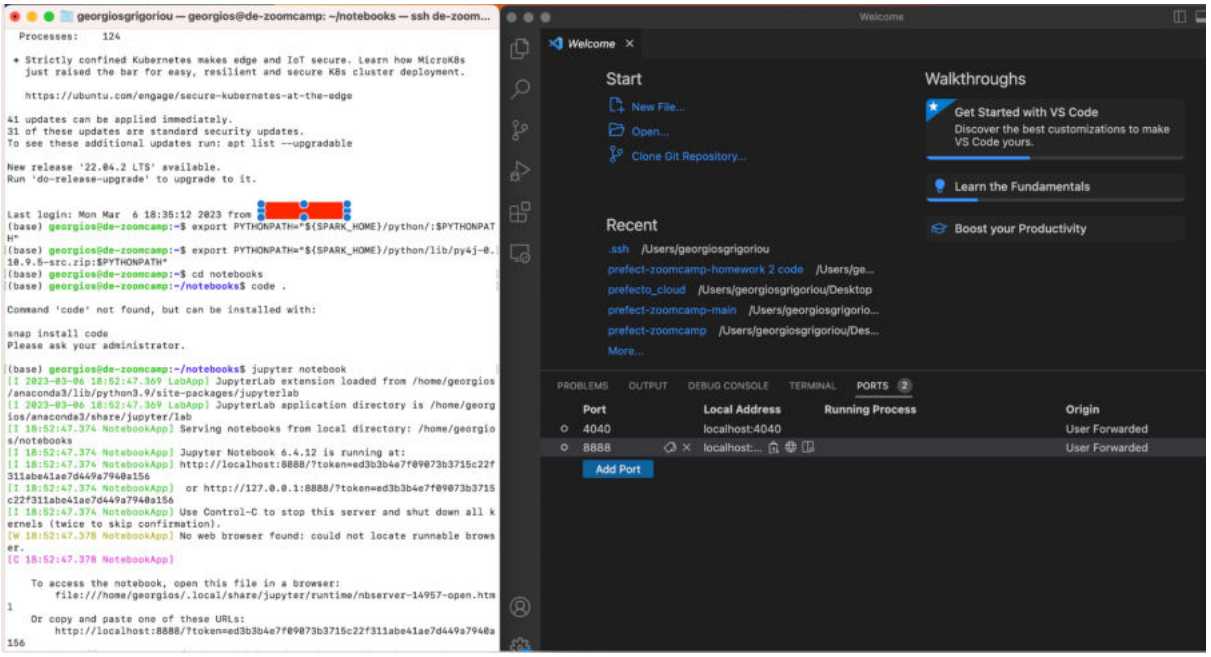
Solution

The correct answer is 3.3.2 . Running the command spark-shell in the terminal after I logged in to my VM of GCP, we could see that the answer correct answer is 3.3.2

And the PySpark version is 3.3.2 as well of course



Open Jupyter Notebook and run PySpark



Question 2:

HVFHW June 2021

Read it with Spark using the same schema as we did in the lessons.

We will use this dataset for all the remaining questions.

Repartition it to 12 partitions and save it to parquet.

What is the average size of the Parquet (ending with .parquet extension) Files that were created (in MB)? Select the answer which most closely matches.

- 2MB
- **24MB**
- 100MB
- 250MB

Solution

The correct answer is 24 MB . Below is the snippet from the code that I ran

Initialize a SparkContext and read binary files from a specified directory using binaryfiles method. The count method is then called om the resulting RDSD to get the number of files in the directory

```
In [12]: sc = SparkContext.getOrCreate()
         parquet_dir = "data/pq/fhvhv/2021/06/"
         rdd = sc.binaryFiles(parquet_dir)
         num_files = rdd.count()

In [13]: num_files
Out[13]: 12
```

Question 2:

What is the average size of the Parquet (ending with .parquet extension) Files that were created (in MB)? Select the answer which most closely matches?

```
In [14]: avg_size = rdd.map(lambda x: len(x[1])).reduce(lambda x, y: x + y) / (num_files * 1024 * 1024)
         print("The average size is equal to {} MB".format(int(avg_size)))
         [Stage 4:==================>                                      (1 + 2) / 3]
         The average size is equal to 22 MB

In [ ]:
```

Question 3:

Count records

How many taxi trips were there on June 15?

Consider only trips that started on June 15.

- 308,164
- 12,856
- **452,470**
- 50,982

Solution

The correct answer is **452,470**. Below is the snippet from the code that I ran

Question 3:

Count records

How many taxi trips were there on June 15?

Consider only trips that started on June 15.

Import Functions from pyspark.sql

```
In [15]: from pyspark.sql import functions as F

In [16]: df \
             .withColumn('pickup_date', F.to_date(df.pickup_datetime)) \
             .filter("pickup_date = '2021-06-15'") \
             .count()

Out[16]: 452470
```

Question 4:

Longest trip for each day

Now calculate the duration for each trip.

How long was the longest trip in Hours?

- 66.87 Hours
- 243.44 Hours
- 7.68 Hours
- 3.32 Hours

Solution

The correct answer is **66.88**. Below is the snippet from the code that I ran

Import Functions from pyspark.sql

```
In [17]: from pyspark.sql.functions import col, max, round, to_date
```

```
In [18]: df \
    .withColumn('duration', ((col('dropoff_datetime').cast('long') - col('pickup_datetime').cast('long')) / 60)/60) \
    .withColumn('pickup_date', to_date(col('pickup_datetime'))) \
    .groupBy('pickup_date') \
        .max('duration') \
    .withColumn('max_duration_rounded', round(col('max(duration)'), 2)) \
    .orderBy('max_duration_rounded', ascending=False) \
    .limit(5) \
    .show()
```

```
[Stage 13:=====================================>                  (8 + 4) / 12]
```

```
+-----------+------------------+--------------------+
|pickup_date|     max(duration)|max_duration_rounded|
+-----------+------------------+--------------------+
| 2021-06-25| 66.87888888888888|               66.88|
| 2021-06-22|25.549722222222222|               25.55|
| 2021-06-27|19.980833333333333|               19.98|
| 2021-06-26| 18.19722222222222|                18.2|
| 2021-06-23|16.466944444444444|               16.47|
+-----------+------------------+--------------------+
```

Question 5:

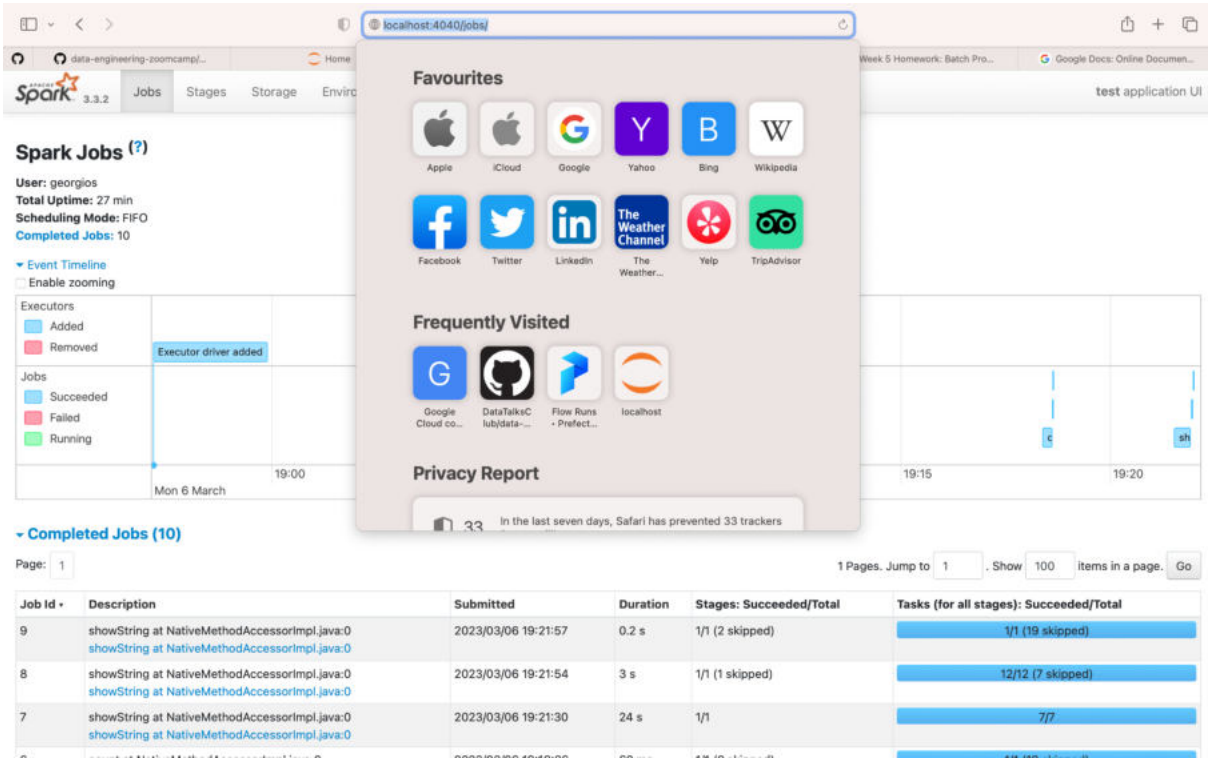
User Interface

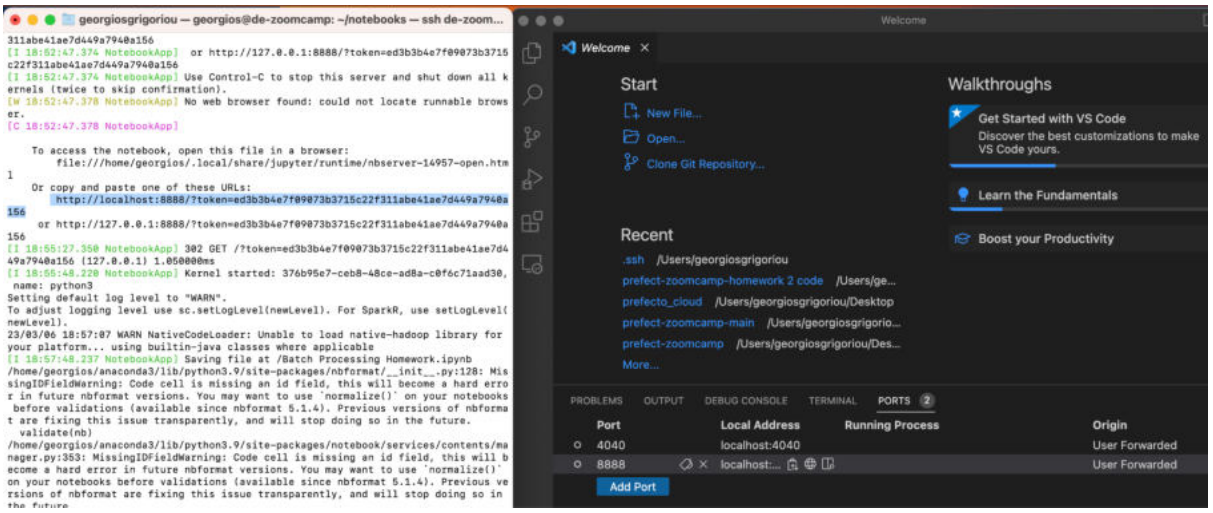
Spark's User Interface which shows application's dashboard runs on which local port?

- 80
- 443
- **4040**
- 8080

Solution

The correct answer is **4040.** That's why we forwarded the port as well in VSC





Question 6:

Most frequent pickup location zone

Load the zone lookup data into a temp view in Spark

Zone Data

Using the zone lookup data and the fhvhv June 2021 data, what is the name of the most frequent pickup location zone?

- East Chelsea

- Astoria
- Union Sq
- **Crown Heights North**

Solution

The correct answer is **Crown Heights North.** Below is the snippet from the code that I ran

```
In [30]: df.registerTempTable('fhvhv_2021_06')

In [28]: df_zones.registerTempTable('zones')

/home/georgios/spark/spark-3.3.2-bin-hadoop3/python/pyspark/sql/dataframe.py:229: FutureWarning: Deprecated in 2.0,
use createOrReplaceTempView instead.
  warnings.warn("Deprecated in 2.0, use createOrReplaceTempView instead.", FutureWarning)

In [32]: spark.sql("""
SELECT
    CONCAT(pul.Zone) AS pu_loc,
    COUNT(1)
FROM
    fhvhv_2021_06 fhv INNER JOIN zones pul ON fhv.PULocationID = pul.LocationID

GROUP BY
    1
ORDER BY
    2 DESC
LIMIT 1;
""").show()

[Stage 34:=====================================>                  (8 + 4) / 12]

+-----------------+--------+
|           pu_loc|count(1)|
+-----------------+--------+
|Crown Heights North|  231279|
+-----------------+--------+
```