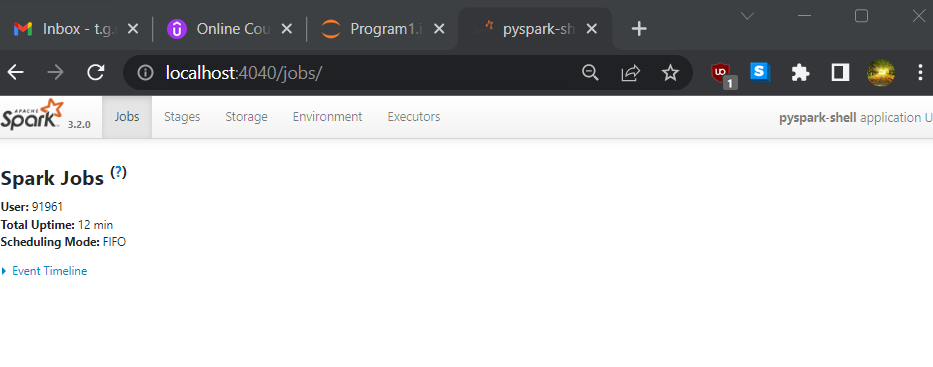
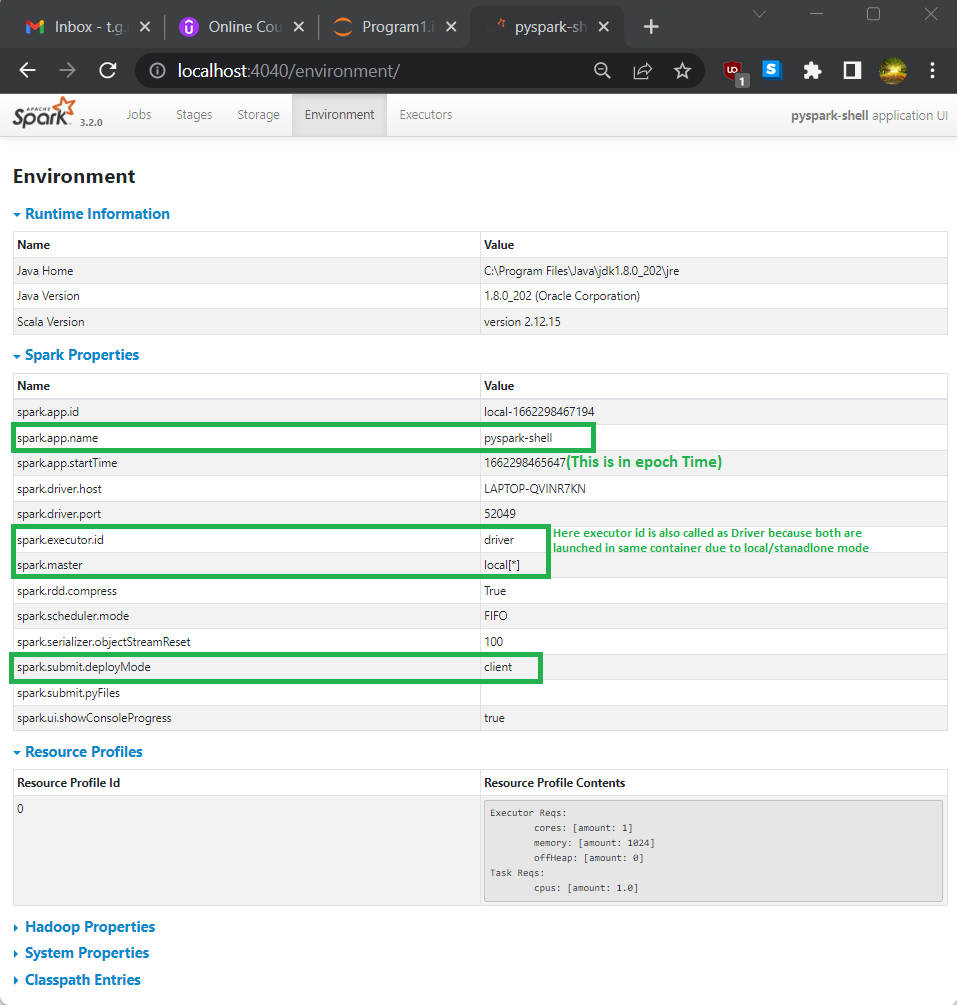
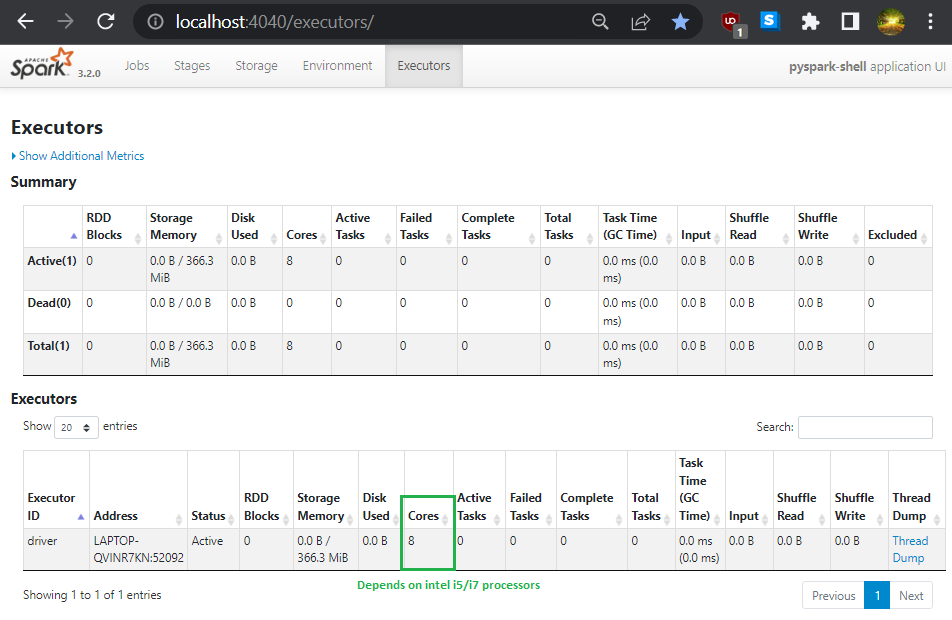
Post creation of Spark Context, we can view the below UI:

<http://localhost:4040/>

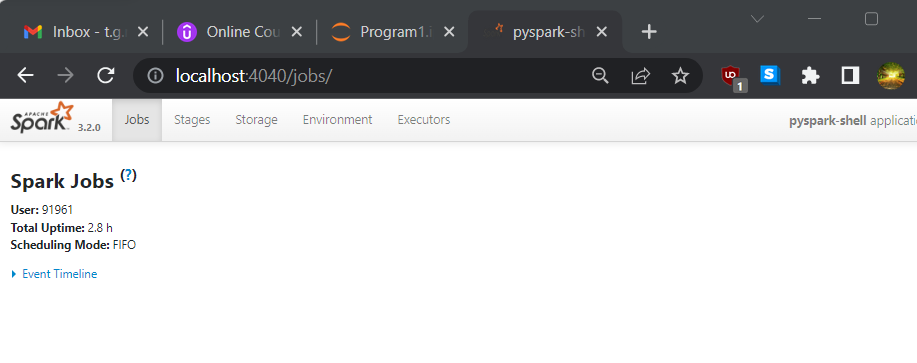




Executor and Driver is launched in same container, even the executor name is also displayed as driver.

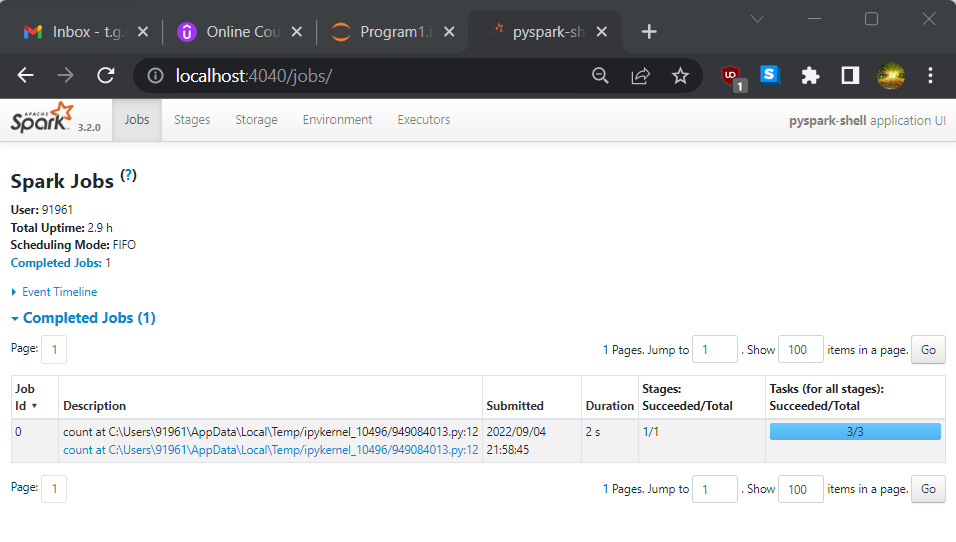


No job is created even after RDD creation:

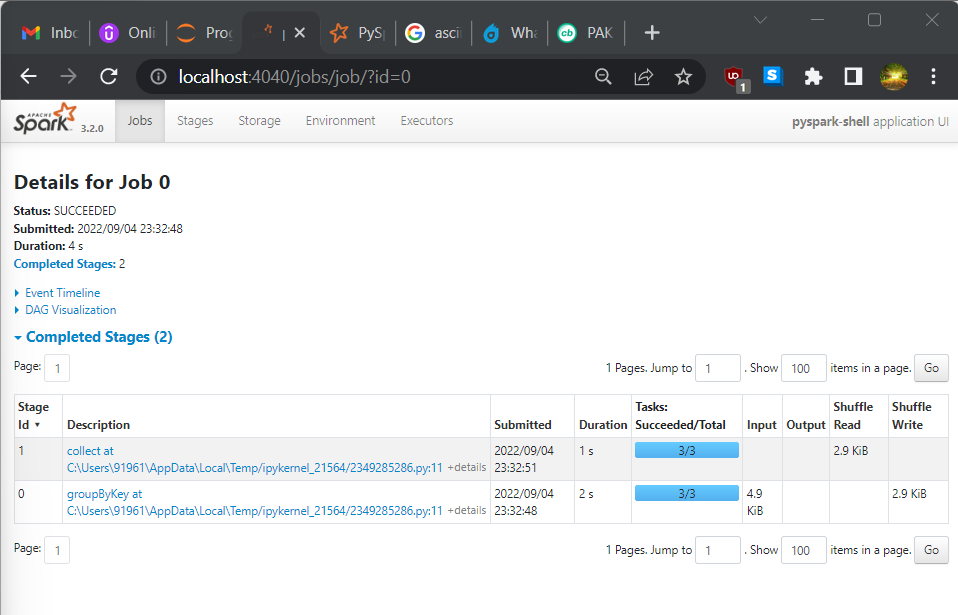


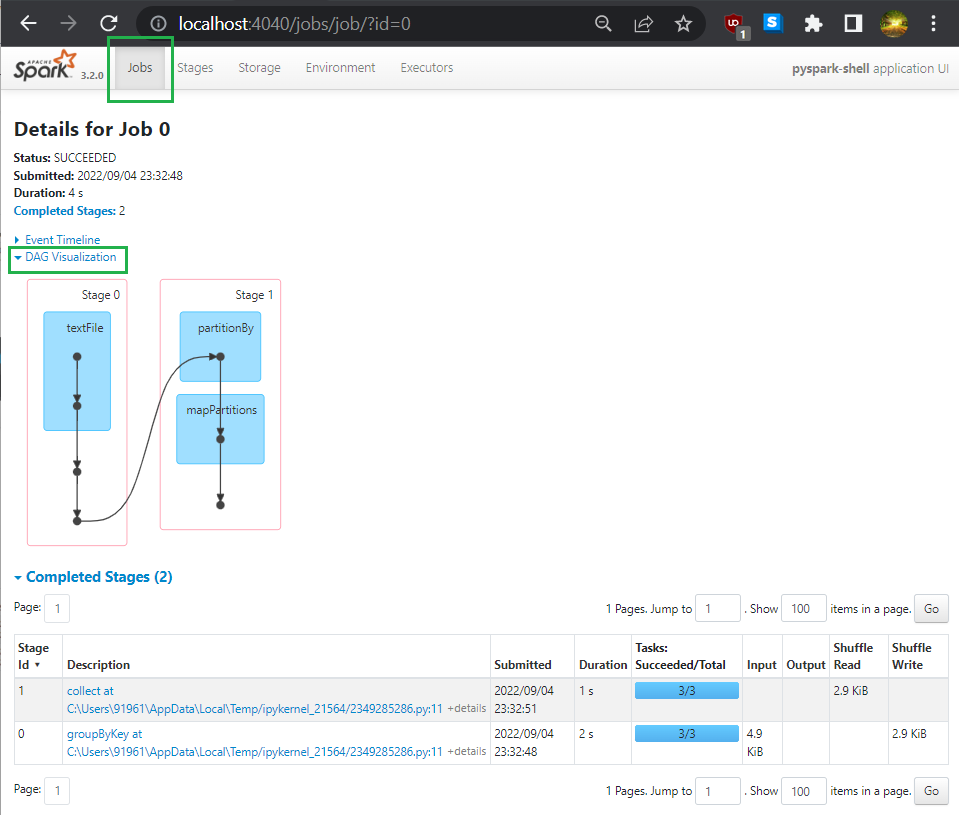
I performed an action, immediately job is created as seen below:

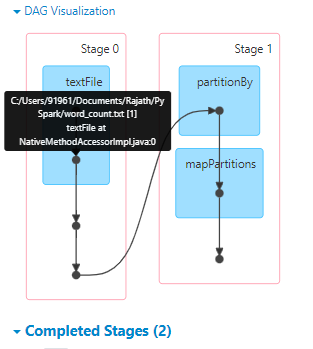
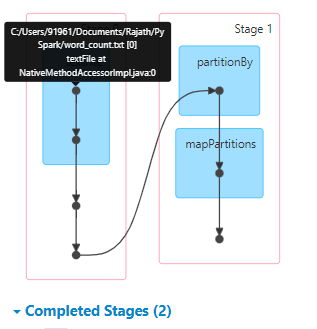
(prior to action multiple narrow (map, flatmap) and one wide(bykey) transformations were applied)

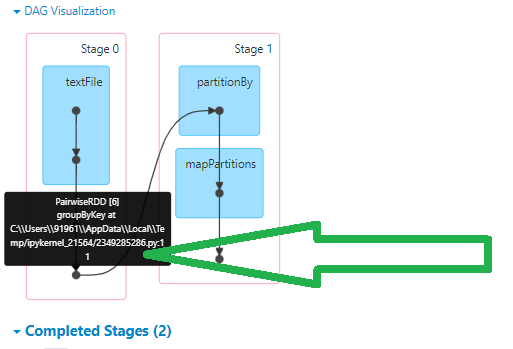
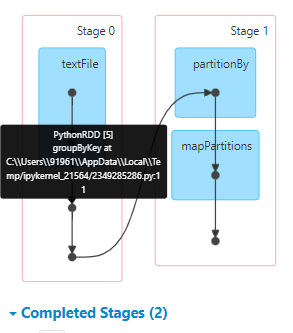


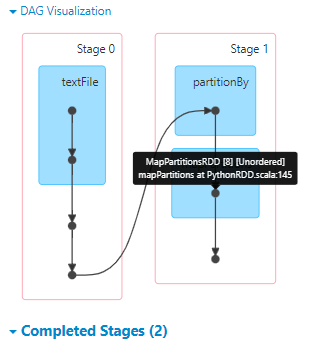
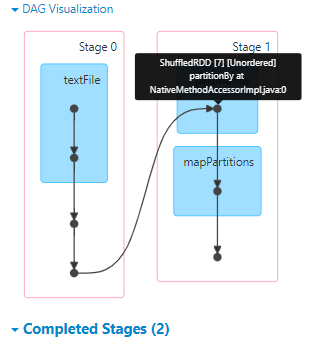
Click on hyperlink under the Description > “count at c:\….” and later select the Dag Visualization

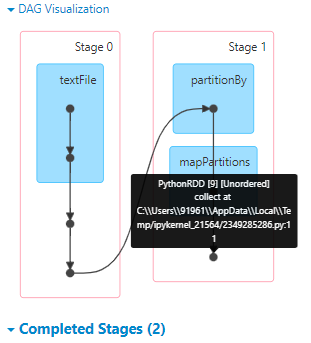




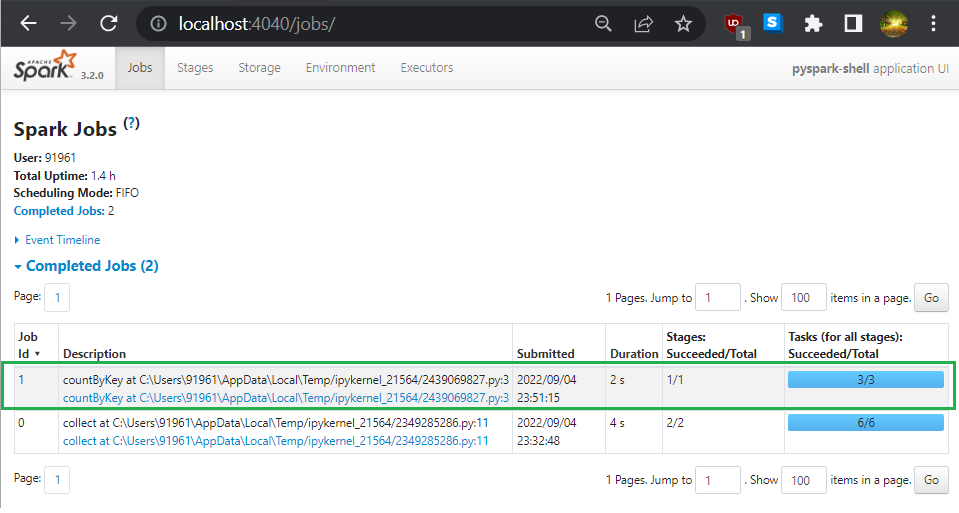


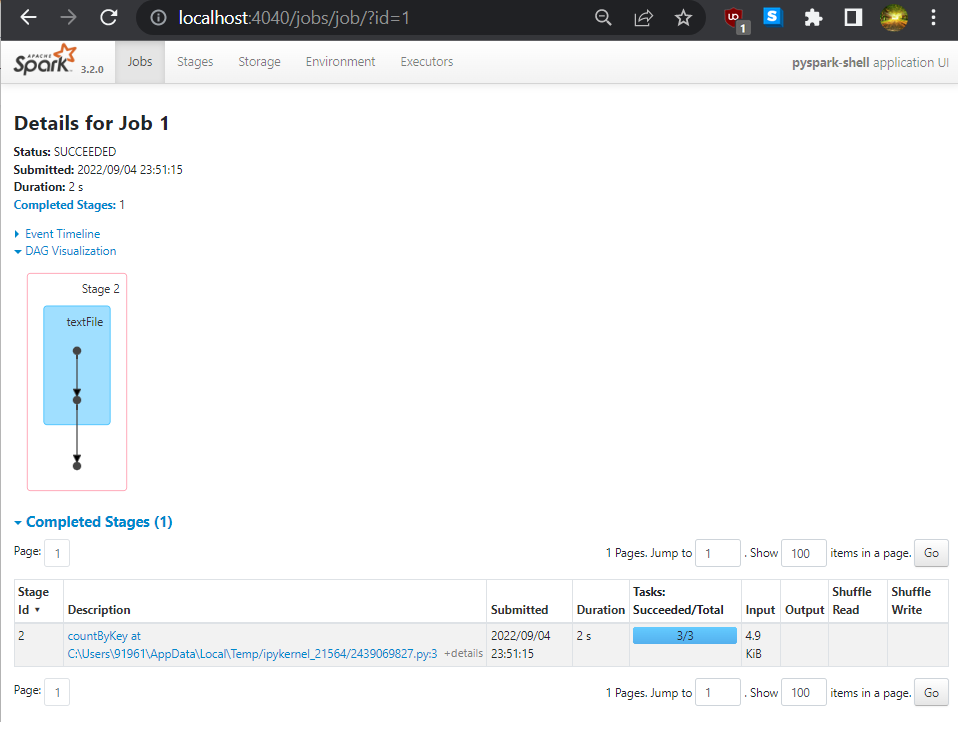






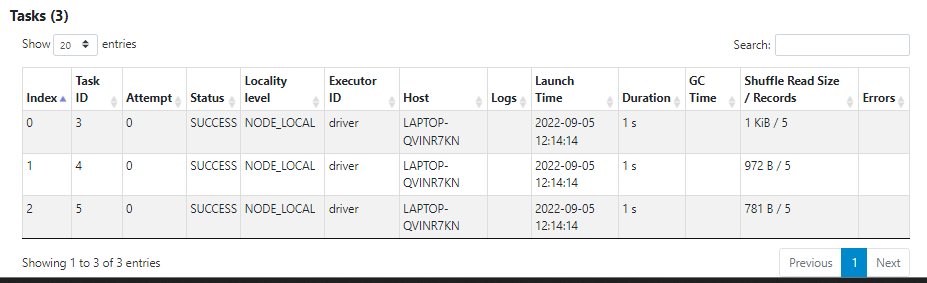
I performed one more action, this led to creation of one more job and that job has single stage only.





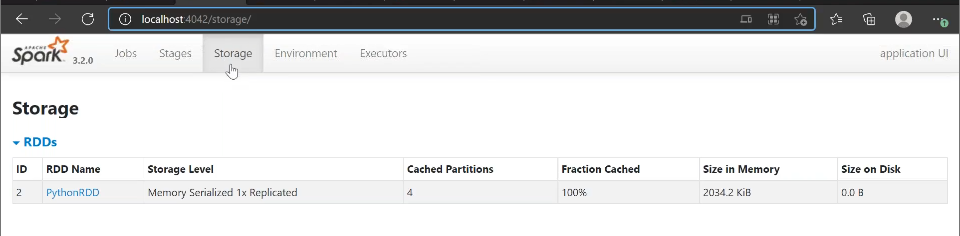
The number of tasks depends on the number of partitions: We can view by clicking again on the hyperlink : “C:\Users\91961…”

Found at the bottom of the page.

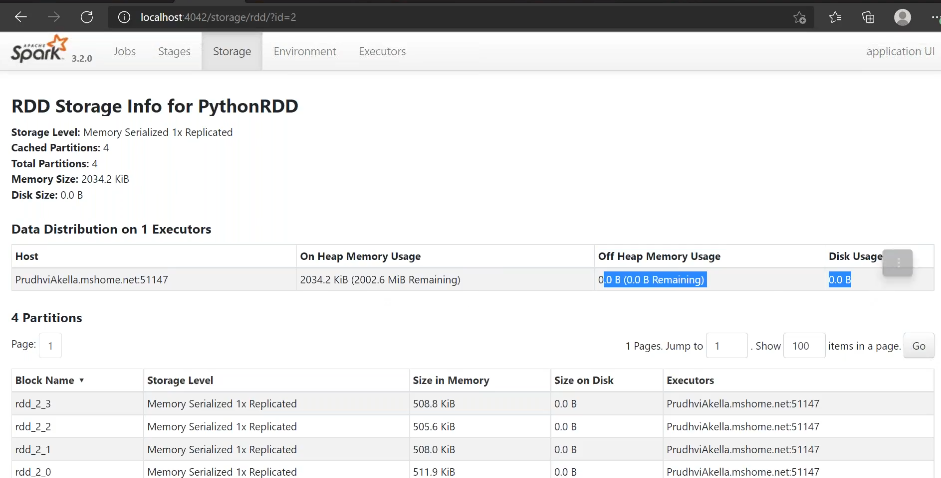


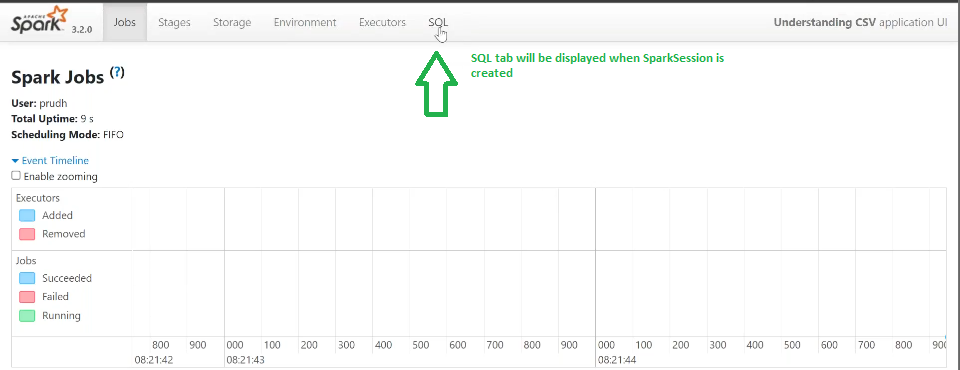
Note: During the wide transformation we can change the number of partitions

The executor memory is the nothing but the storage memory mentioned in the storage tab of the Spark UI:

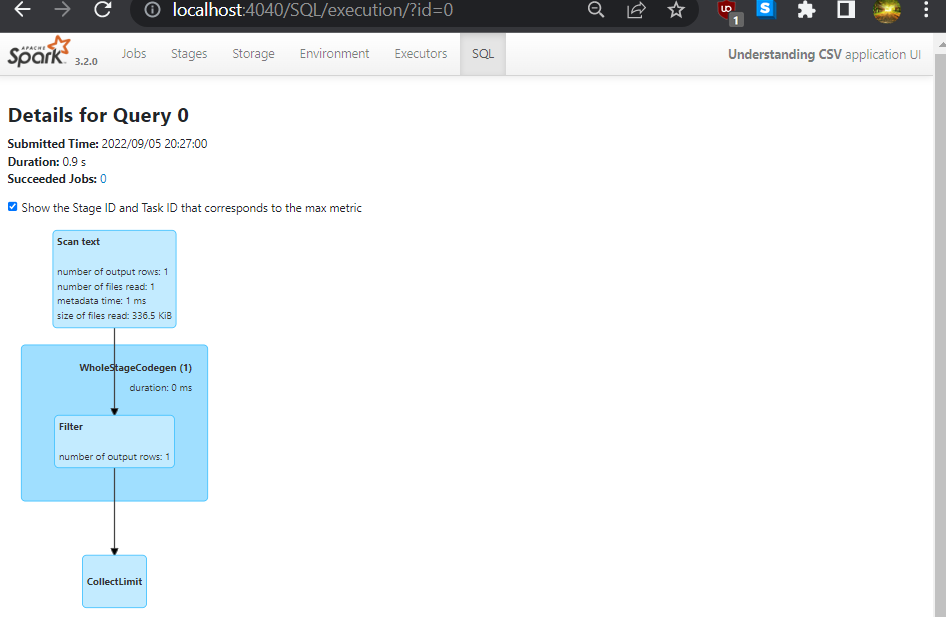


Caching and RDD always gives priority to Onheap > off heap > finally if both on&off are full then it will spill to disk.



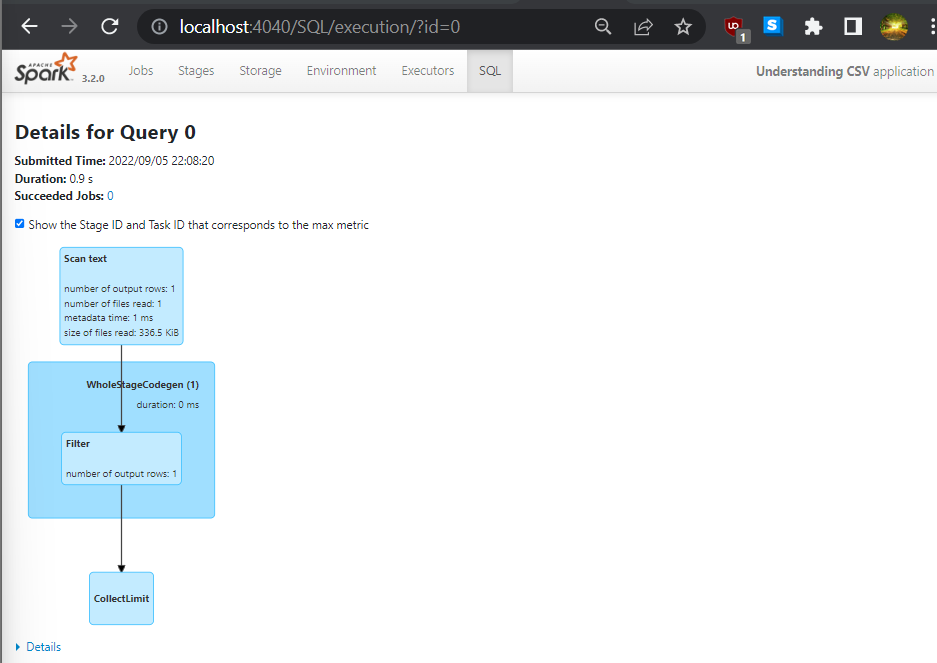


When header and inferschema is true, below is the output:



When header and inferschema is false, below is the output:

\*\*\*\*\*\*Even when the .header,true and .inferschema,true is deleted even then in the UI it tells 1 output row is read, because it will read the first line and understand how many columns this csv files has.



== Physical Plan ==

CollectLimit (3)

+- \* Filter (2)

+- Scan text (1)

(1) Scan text

Output [1]: [value#0]

Batched: false

Location: InMemoryFileIndex [file:/C:/Users/91961/Documents/Rajath/PySpark/Datasets/Datasets/covid/archive/country\_vaccinations.csv]

ReadSchema: struct<value:string>

(2) Filter [codegen id : 1]

Input [1]: [value#0]

Condition : (length(trim(value#0, None)) > 0)

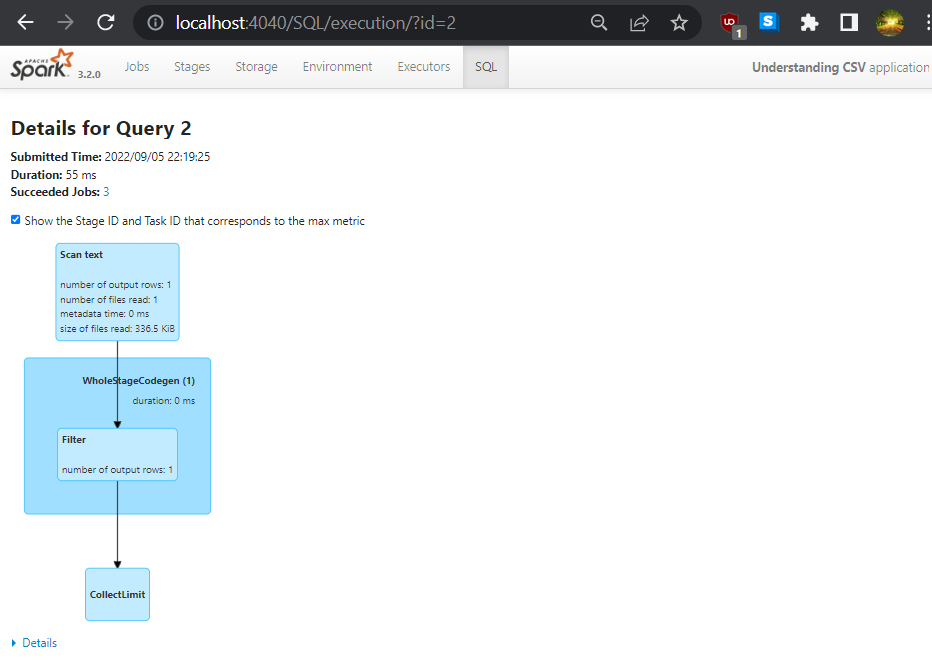
(3) CollectLimit

Input [1]: [value#0]

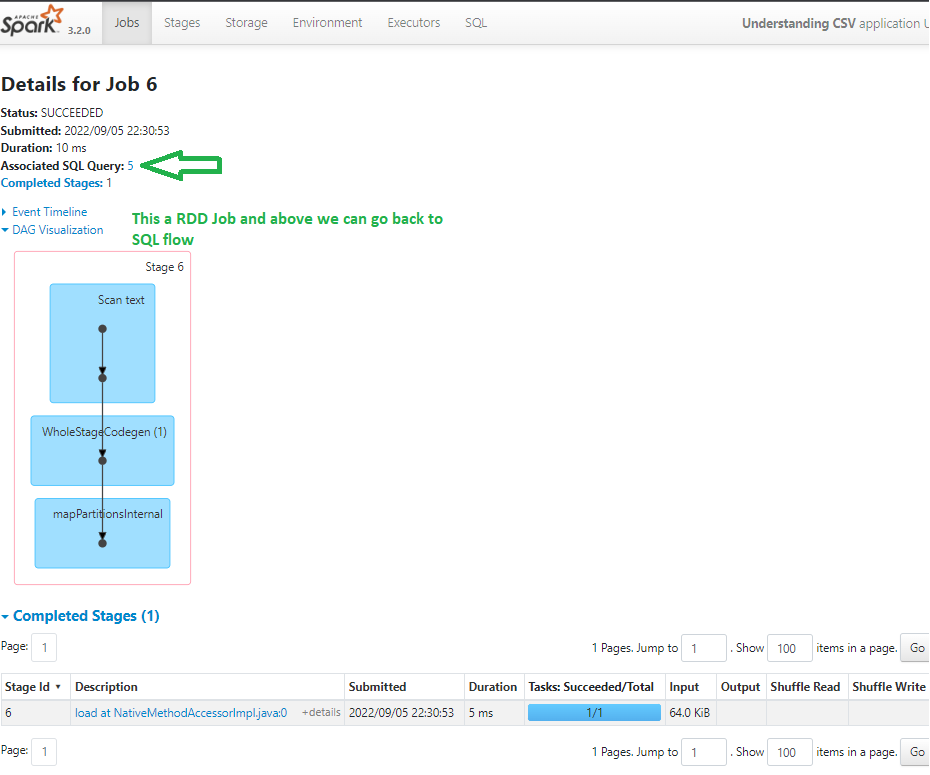
Arguments: 1

Even when the .inferschema,true is only deleted and run the step, the metadata time will be 0ms and output row will be 1 because .header,true

When inferschema is set to false, it takes less time to fetch meta data, becuase everything will be string, it makes it's work easy



Respective RDD job for PySparkSQL:



spark = SparkSession \

.builder \

.master('local[\*]') \

.config("spark.executor.memory", "5g") \

.config("spark.driver.memory", "5g") \

.config("spark.memory.offHeap.enabled",True) \

.config("spark.memory.offHeap.size","2g") \

.config("spark.sql.repl.eagerEval.enabled", True)\

.getOrCreate()

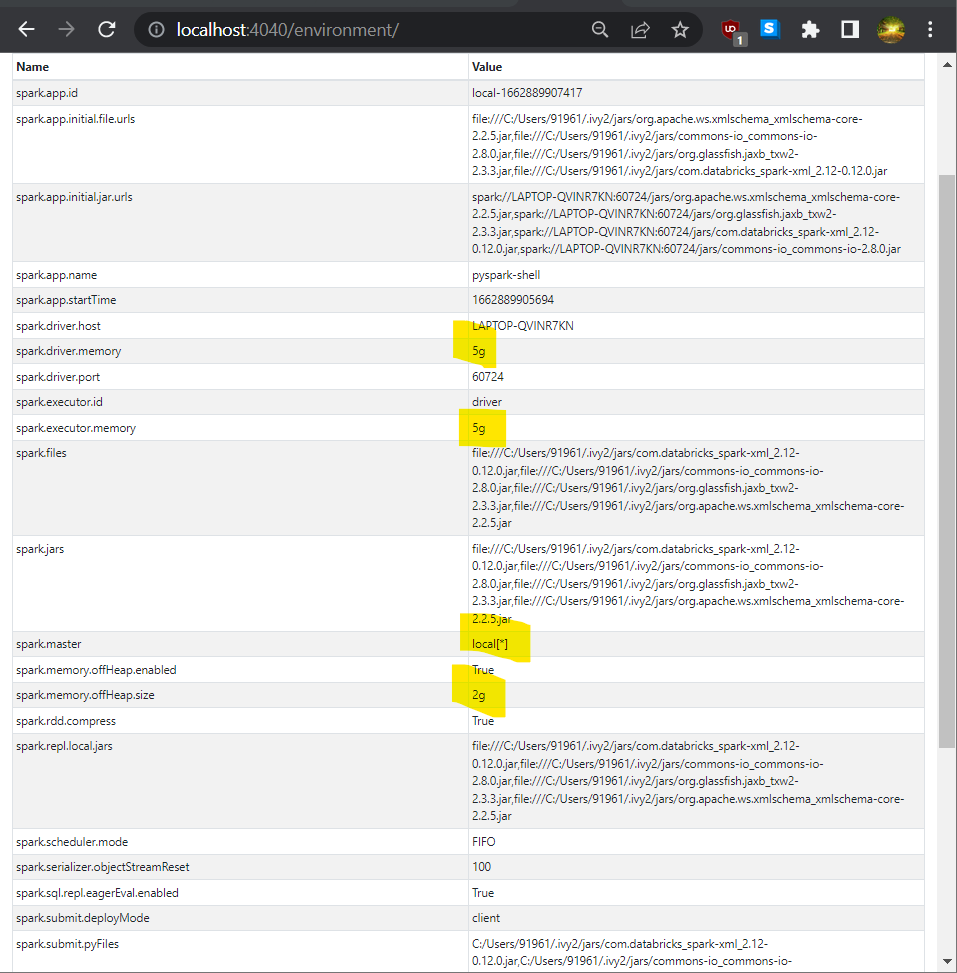
On heap memory is divided into 4 parts: There fore because of below splits, complete 5GB is not shown on the UI, instead it will show only 4.5GB on **Executors Tab**

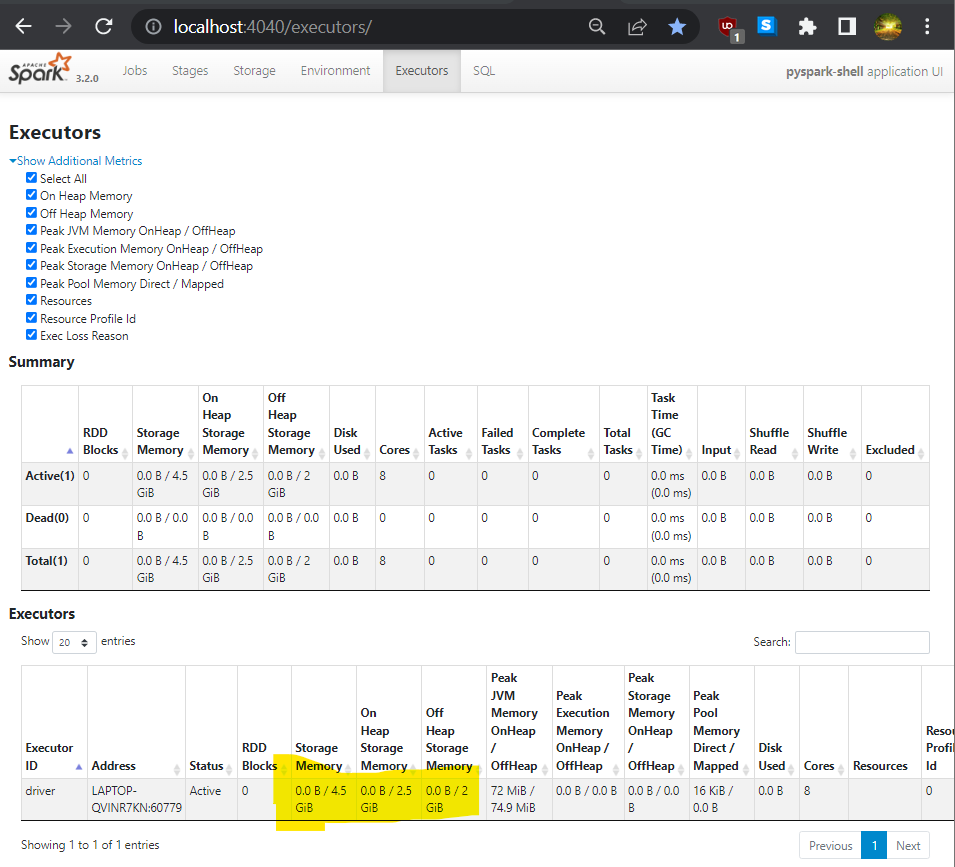
Storage Memory

Execution Memory

User Memory

Reserved Memory





How does Json Explode works?????

