Task-2

Aim: Implement the spark word count program on databricks and visualise how the RDD is partitioned between the executors and explore till the last thread in cluster.

- **Algorithm:**

- Start
- Create rdd using parallelize method (we can various other methods also like textFile, wholeTextFile etc.)
- Convert the rdd into keyvalue pair using map (we can use various other method like flatMap, wholeTextFile etc.)
- Aggregate all the same key value using reduceByKey.
- Print the output using collect() method
- End.

- **Program of word count problem:**

```
data = [1,2,3,4,5]

rdd = sc.parallelize(data,5)

rdd.collect()

word_count = rdd.map(lambda x: (x,1))\

        .reduceByKey(lambda x,y:(x+y))

word_count.collect()
```

- **Approach / How to do this on databricks.**

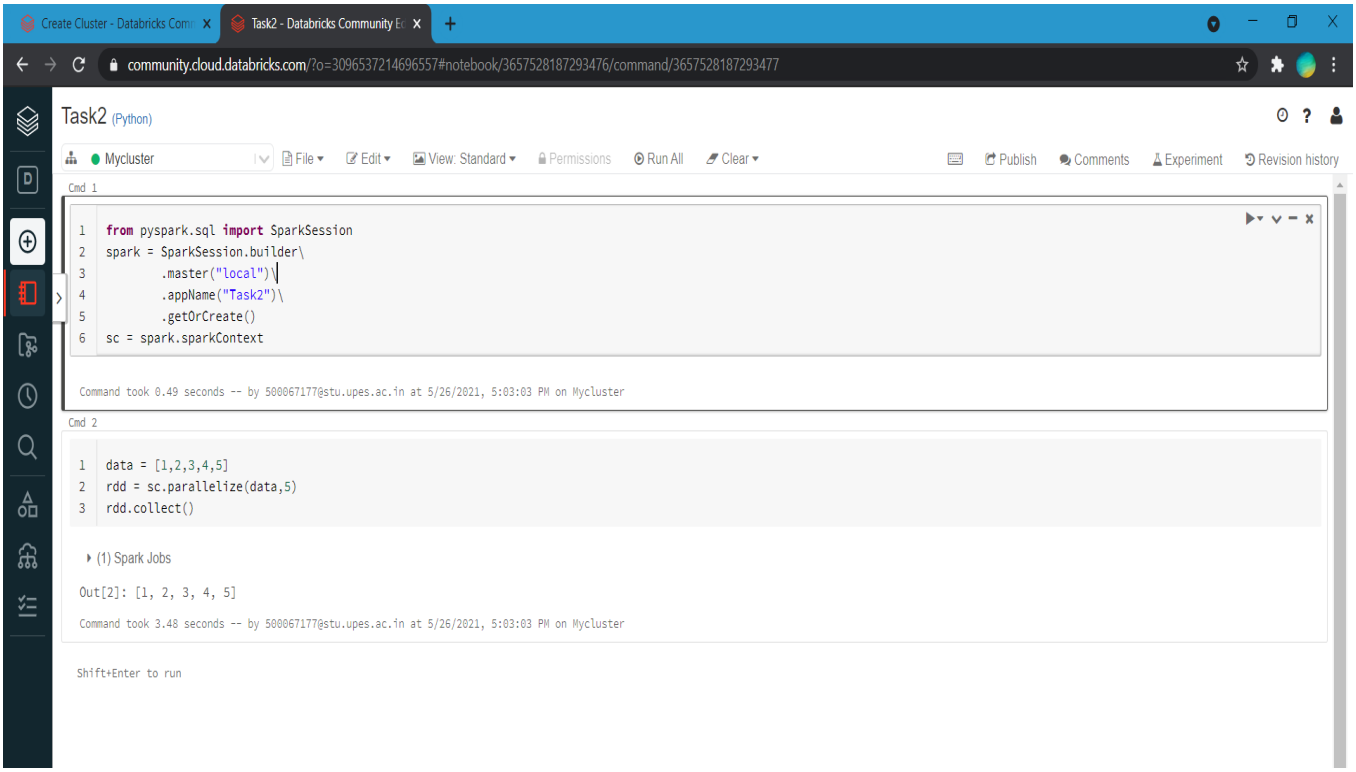
Steps are as follow:

Step 1: Open the Databricks account

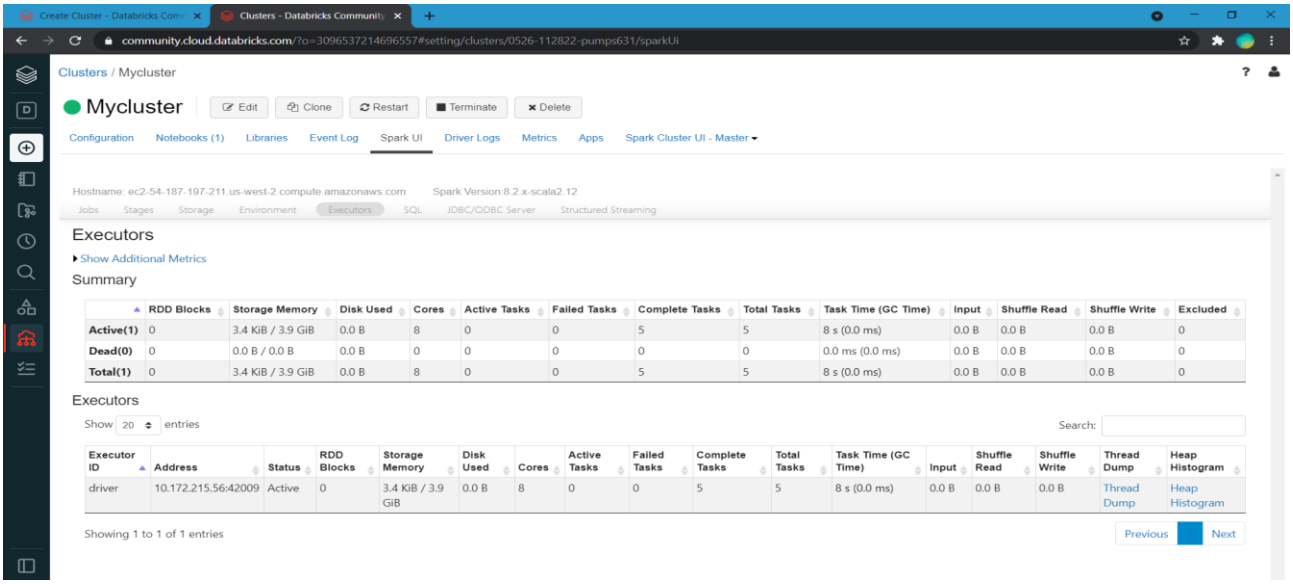
Step 2: Create a cluster and attach it to the notebook.

Step 3: Write the code of word count.

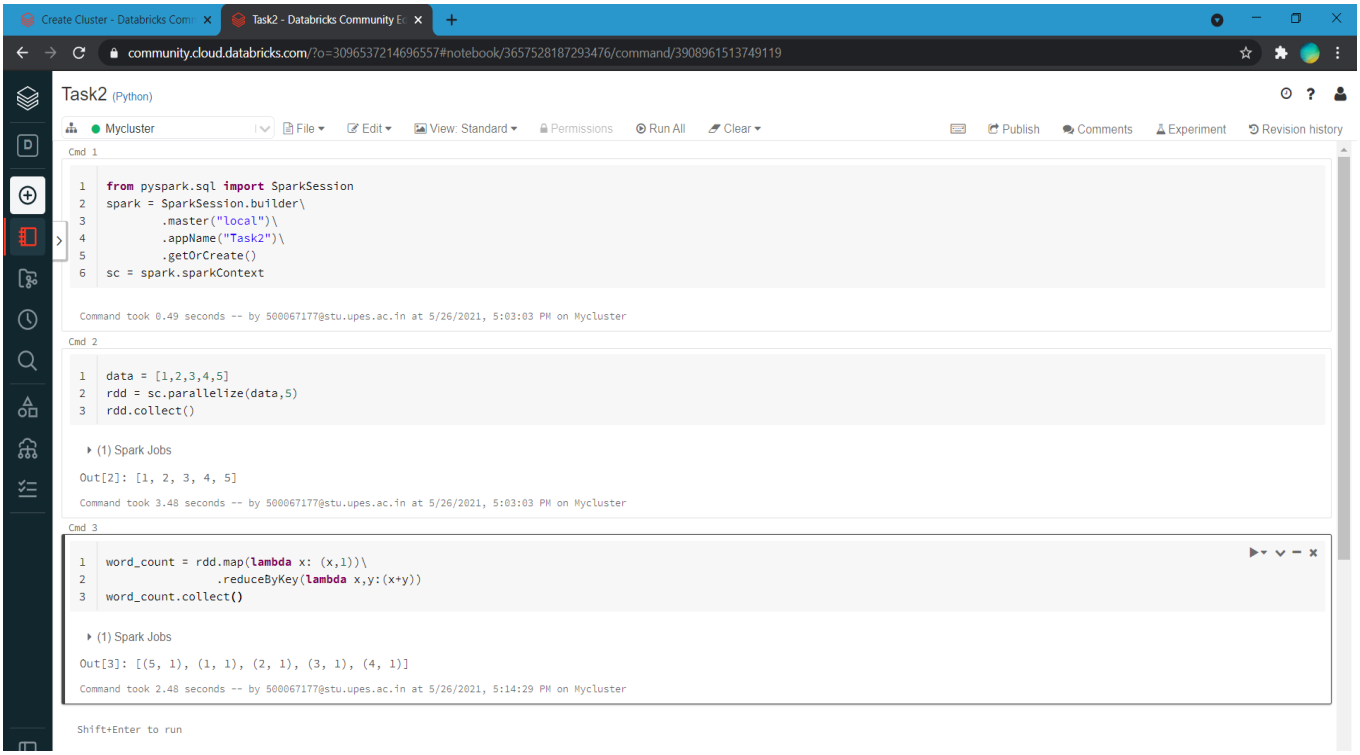
a) Initially I create a rdd(resilient distributed dataset) and set the number of partitions is 5.



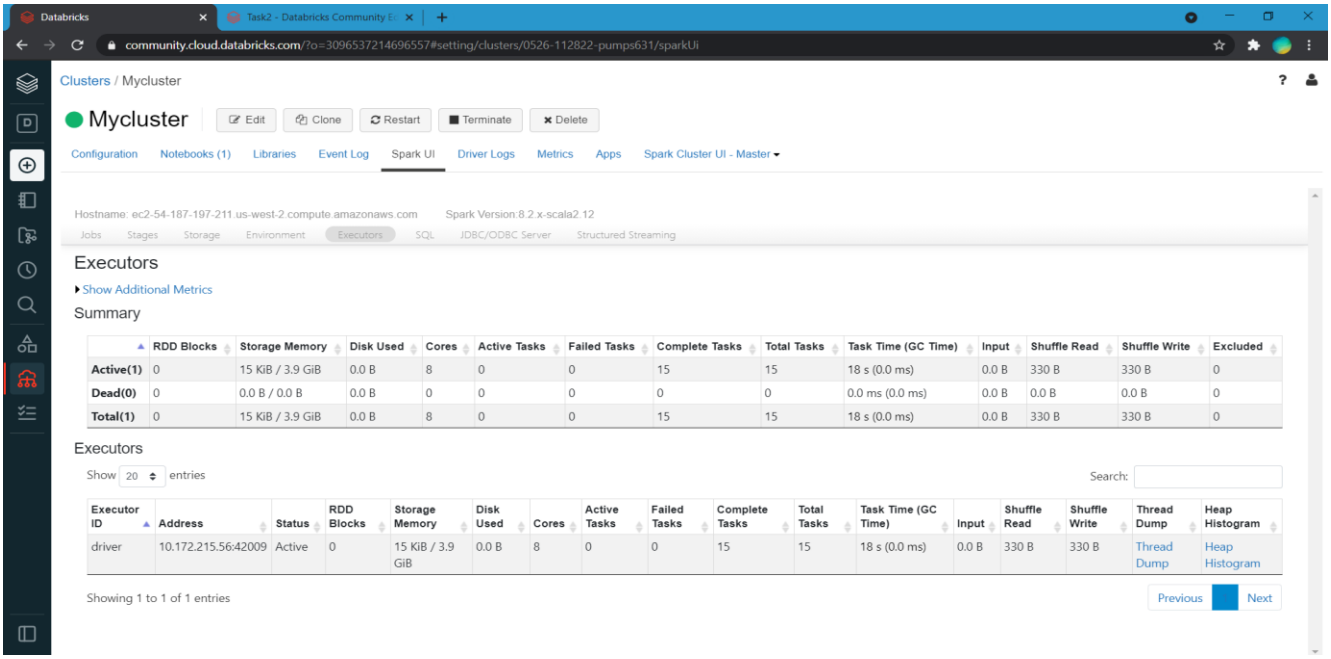
Because I set the partition value manually to 5 So it create 5 tasks in the backend and it will be handled by 5 executor As we know number of partitions = number of executors.



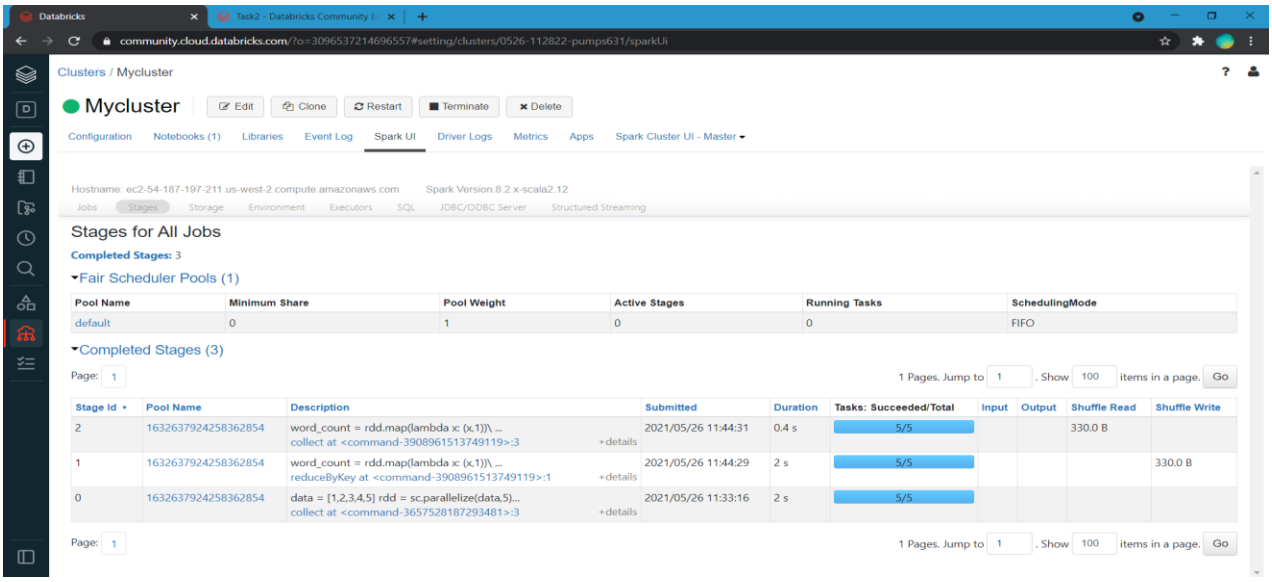
b) With the help of **Map** I convert the rdds into key value pair rdd and after that using **reduceByKey** I aggregate all the values of same key.



Where, key represents the word and value represents the frequency of that word.



After executing the map and reduceByKey the number of task increases from 5 to 15. The reason behind this is because each executor executes the map and reduceByKey for each partitioned value so it become 10 and added into the previously calculated value 5.



By following these process we easily get our active executors.