Task 1: Getting Started

Question 1

What is the default block size on HDFS? **Answer:** 128 MB

What is the default replication factor of HDFS on Dataproc? **Answer:** 3

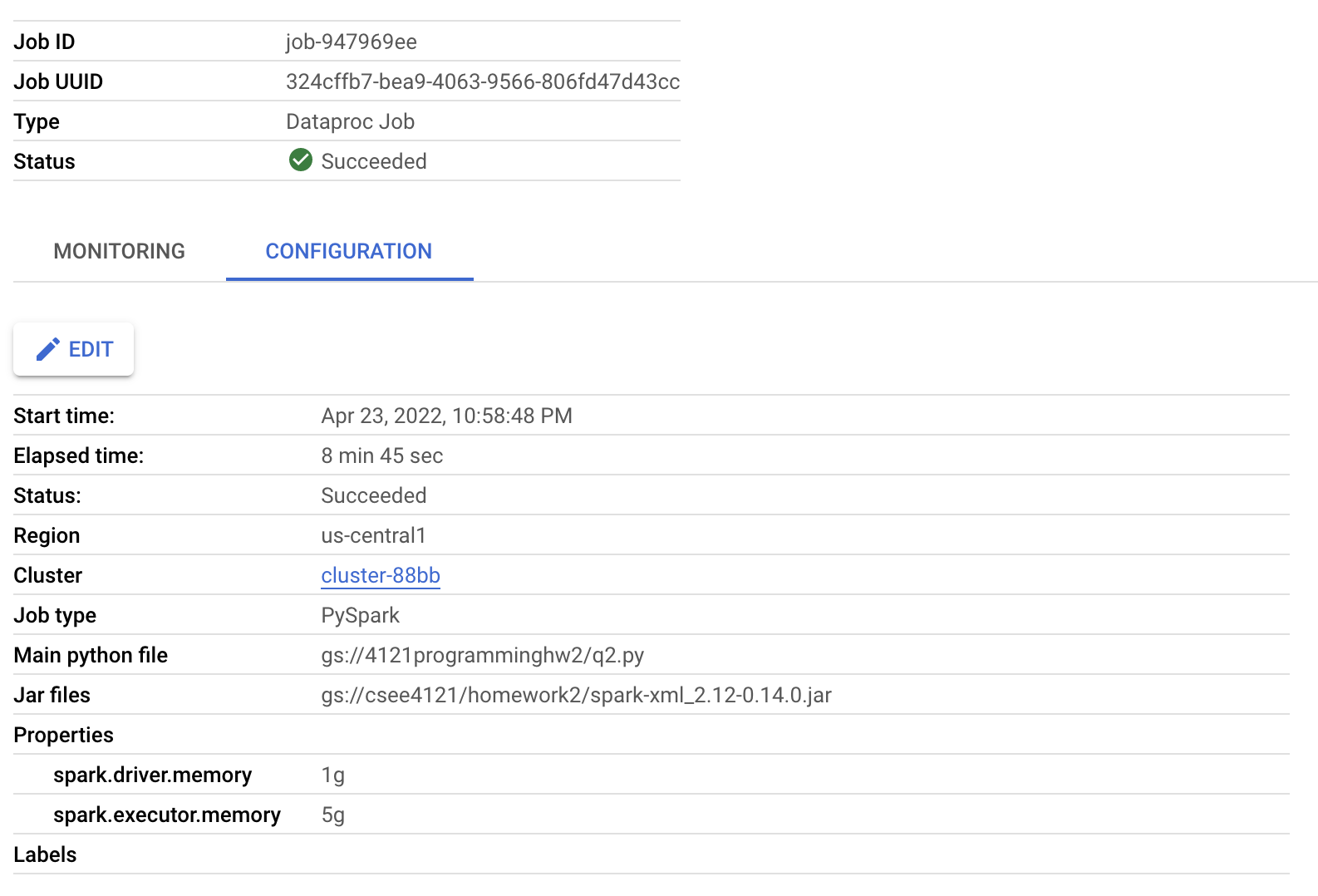
Task 2: Webgraph on Internal Links

Question 2

(Set the **Spark driver memory** to 1GB and the **Spark executor memory** to 5GB, Single Node cluster)

Use enwiki\_test.xml as input and run the program locally on a Single Node cluster using 4 cores. Include your screenshot of the dataproc job. What is the completion time of the task?

**Answer:** 8min 45s, see below:

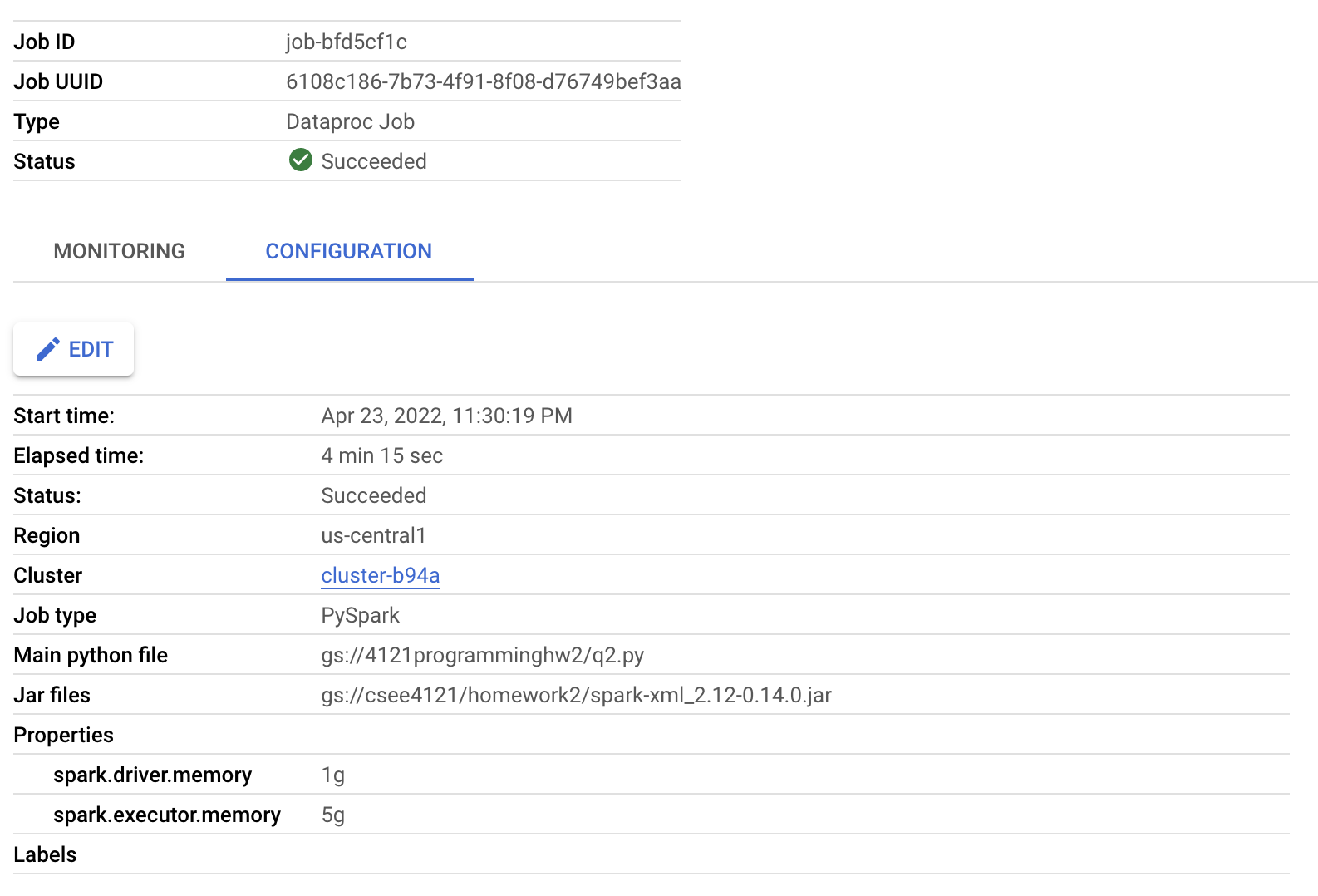


Question 3

(Set the **Spark driver memory** to 1GB and the **Spark executor memory** to 5GB, 3 node cluster)

Use enwiki\_test.xml as input and run the program under HDFS inside a 3 node cluster (2 worker nodes). Include your screenshot of the dataproc job. Is the performance getting better or worse in terms of completion time? Briefly explain.

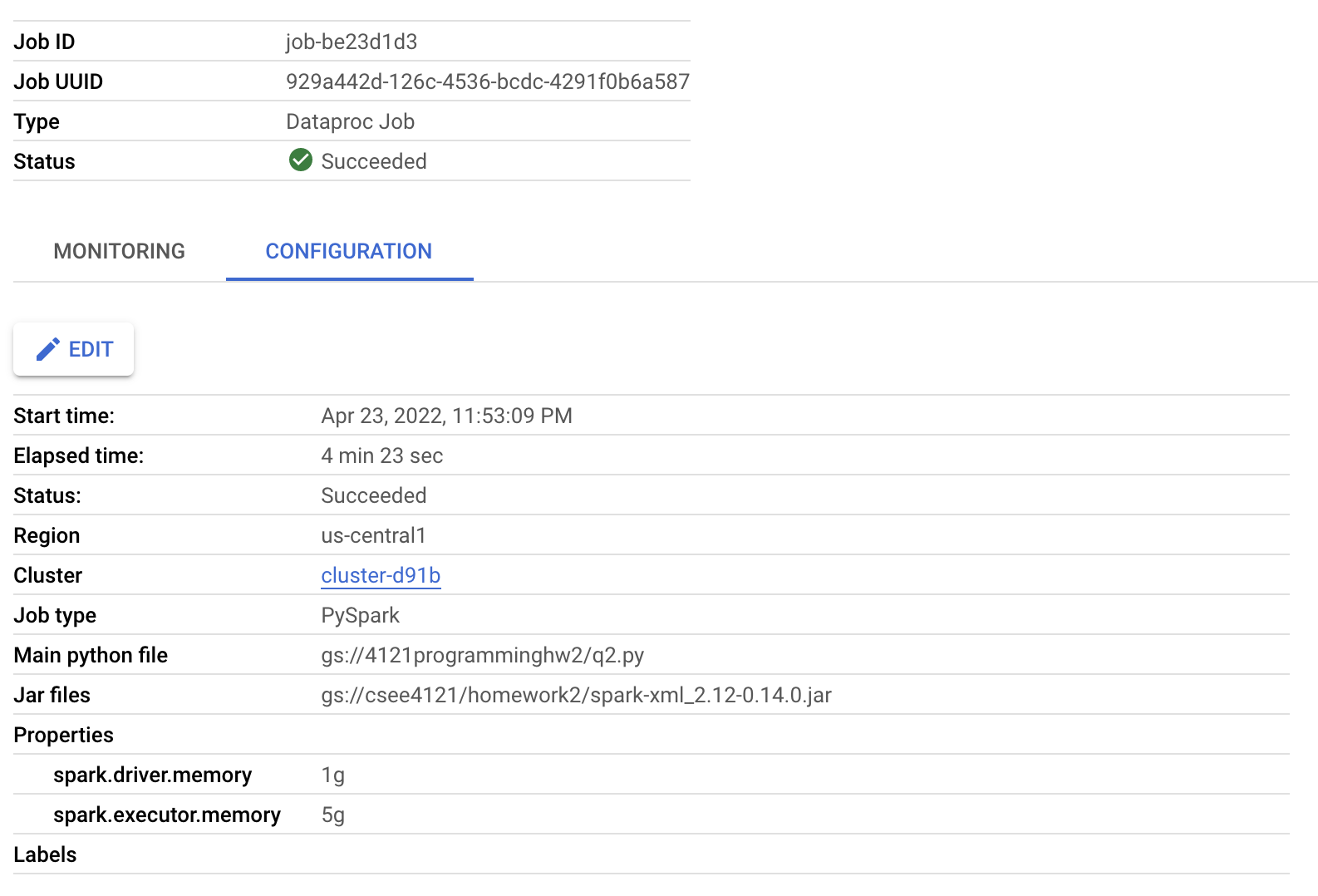
**Answer:** 4 min 15 seconds. As expected, the performance improves when we increase the number of nodes. It is roughly twice as fast compared to the previous question.



Question 4

For this question, change the default block size in HDFS to be 64MB and repeat Question 3. Include your screenshot of the dataproc job. Record run time, is the performance getting better or worse in terms of completion time? Briefly explain.

**Answer:** 4 min 23 s. The performance is a little bit worse since there are more blocks to manipulate (the HDFS file is split into 64MB sized blocks rather than 128MB sized blocks) which results in a higher completion time.



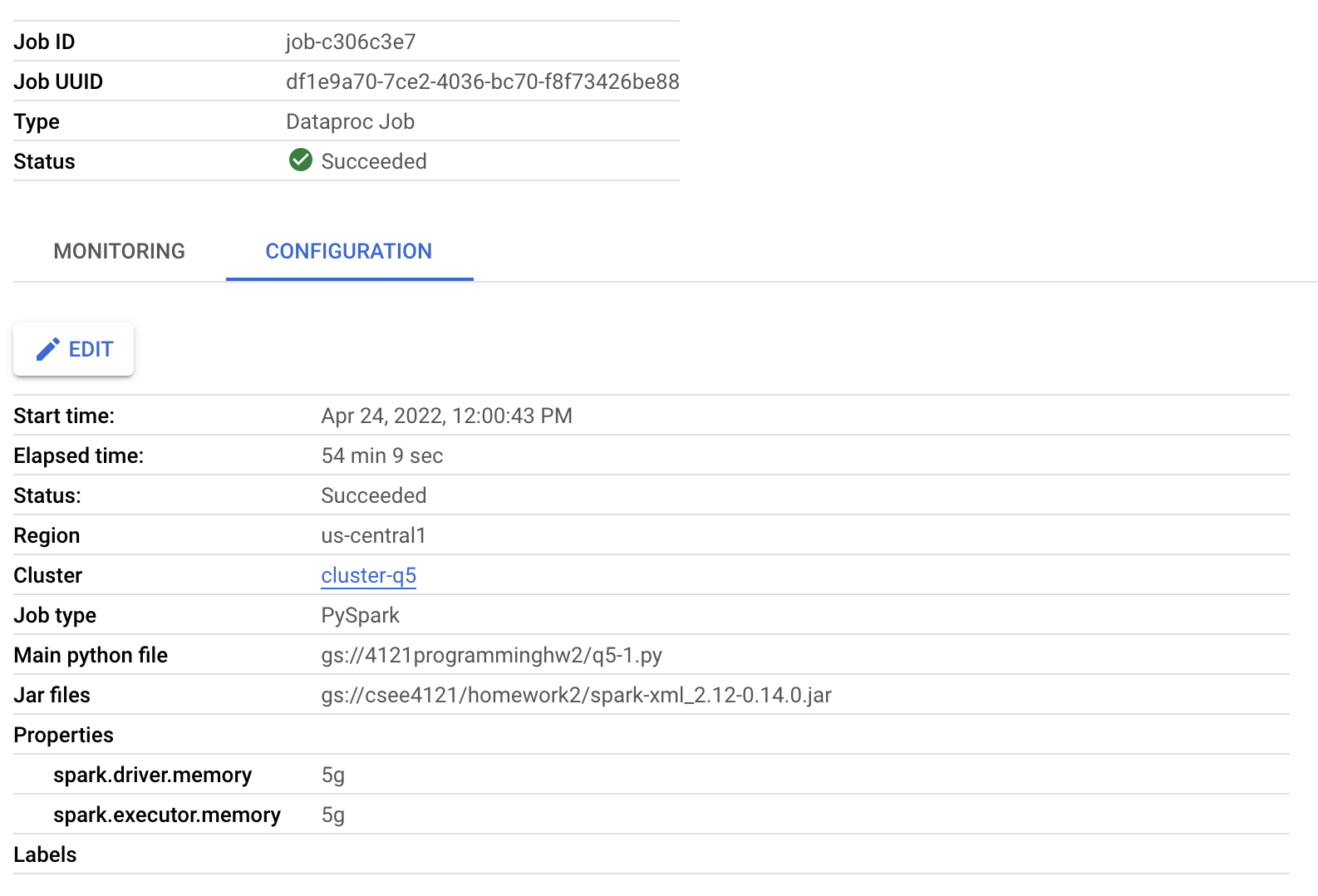
Question 5

(Set the **Spark driver memory** to 5GB and the **Spark executor memory** to 5GB to answer Question 5-7)

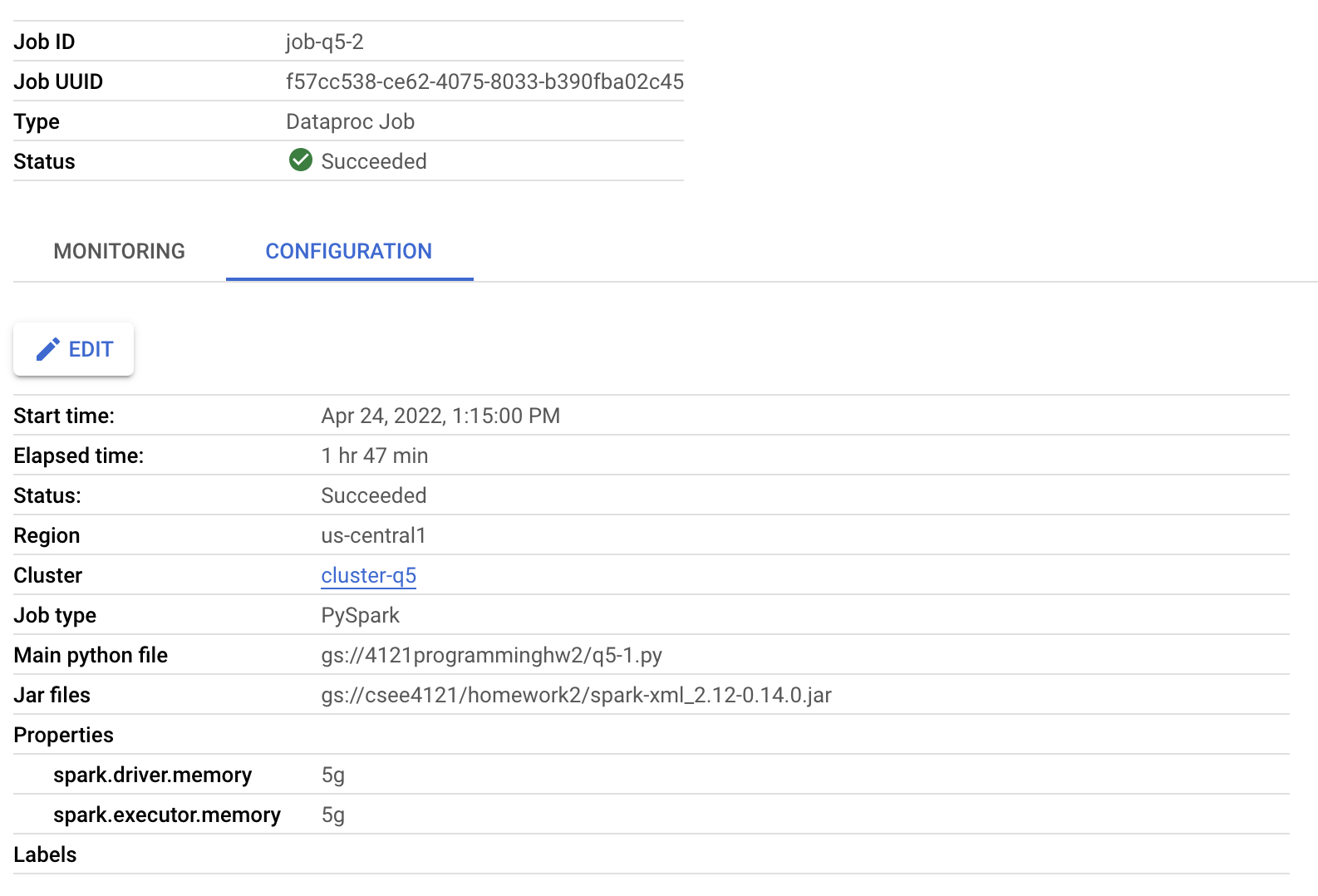
Use enwiki\_whole.xml as input and run the program under HDFS inside the Spark cluster you deployed. Record the completion time. Now, kill one of the worker nodes immediately. You could kill one of the worker nodes by go to the **VM Instances** tab on the Cluster details page and click on the name of one of the workers. Then click on the STOP button. Record the completion time. Does the job still finish? Do you observe any difference in the completion time? Briefly explain your observations. Include your screenshot of the dataproc jobs.

**Answer:** Q5-1: 54min 9s (baseline)

Even though the driver memory was increased from 1GB to 5GB, it still takes a very long time to complete the job with enwiki\_whole.xml due to the size of the data.



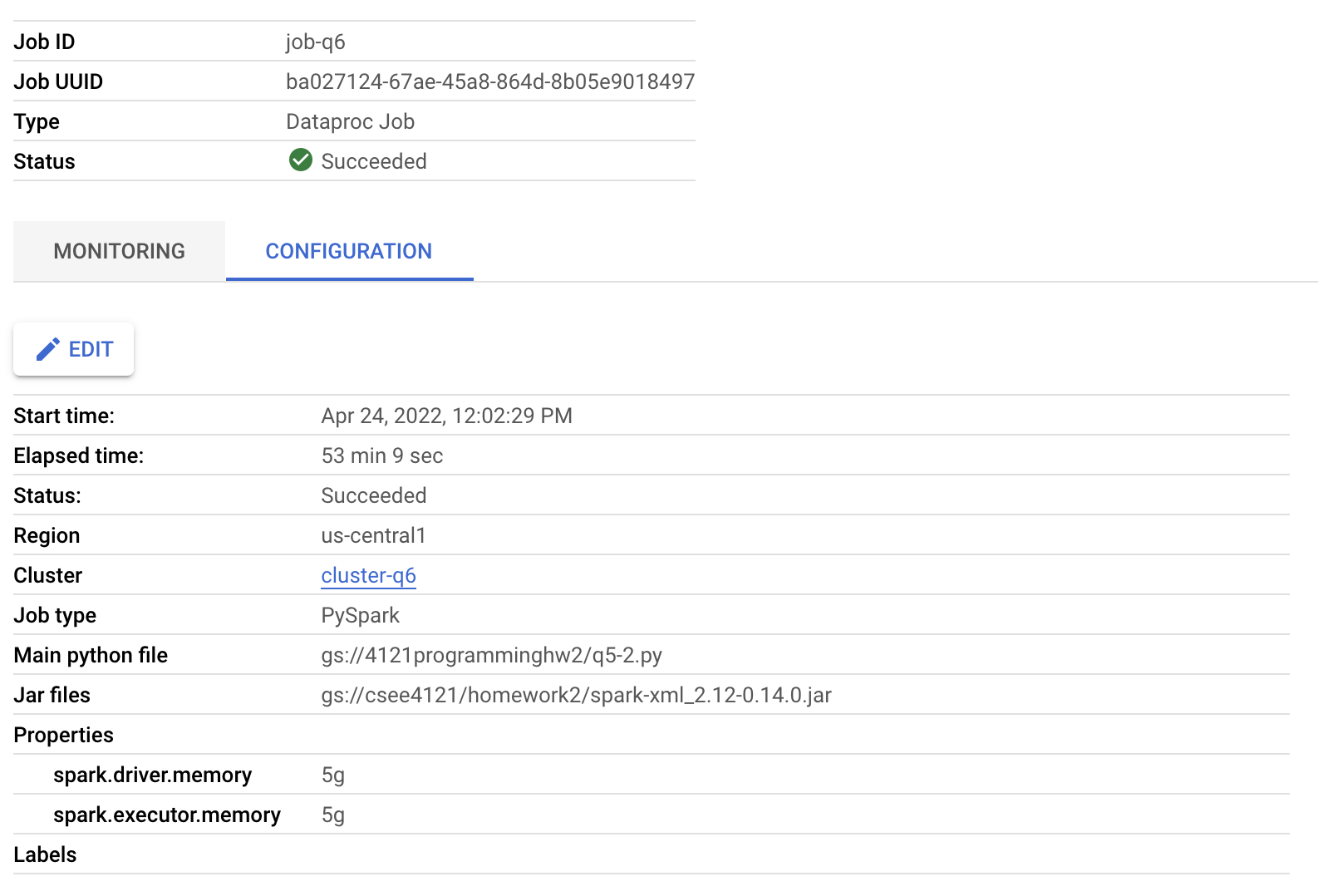
Q5-2: The job still runs after killing one of the worker nodes but it takes 1h 47min to complete, which is twice as long compared to Q5-1. This makes sense because there is one worker completing the job (as opposed to two workers from before).



Question 6

Only for this question, change the replication factor of enwiki\_whole.xml to 1 and repeat Question 5 without killing one of the worker nodes. Include your screenshot of the dataproc job. Do you observe any difference in the completion time? Briefly explain.

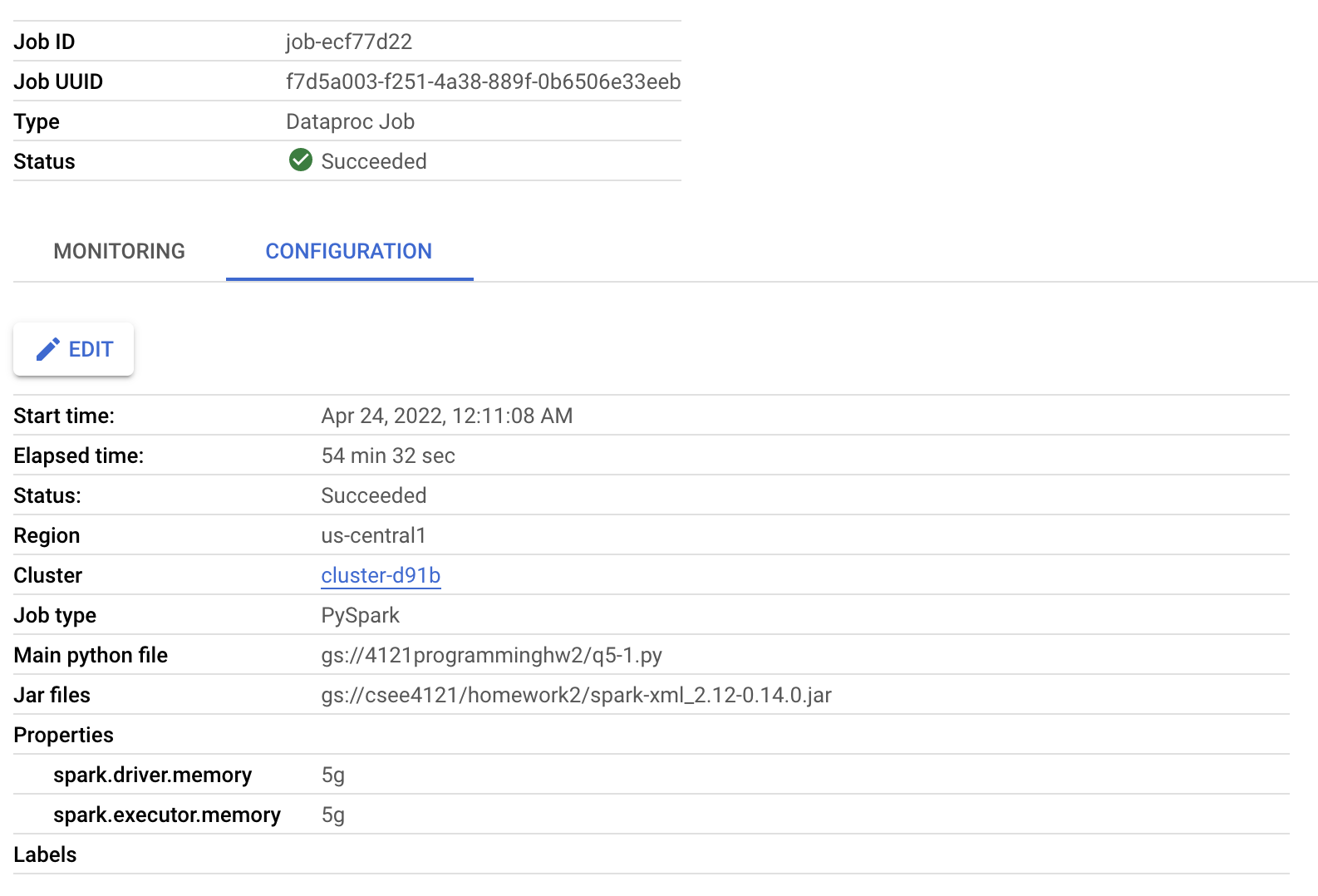
**Answer:** 53min 9s. With a replication factor of 1, only one copy of the block is kept in the cluster. This means fewer data blocks that the system needs to manage. As expected, we see slightly better performance compared to the baseline in Question 5. Note the downside of this is that the data may be more prone to getting corrupted if there is a database failure.



Question 7

Only for this question, change the default block size in HDFS to be 64MB and repeat Question 5 without killing one of the worker nodes. Record run time, include your screenshot of the dataproc job. Is the performance getting better or worse in terms of completion time? Briefly explain.

**Answer:** 54 min 32 s. Just like in Question 4, the performance is a little bit worse since there are more blocks to manipulate which results in a higher completion time.



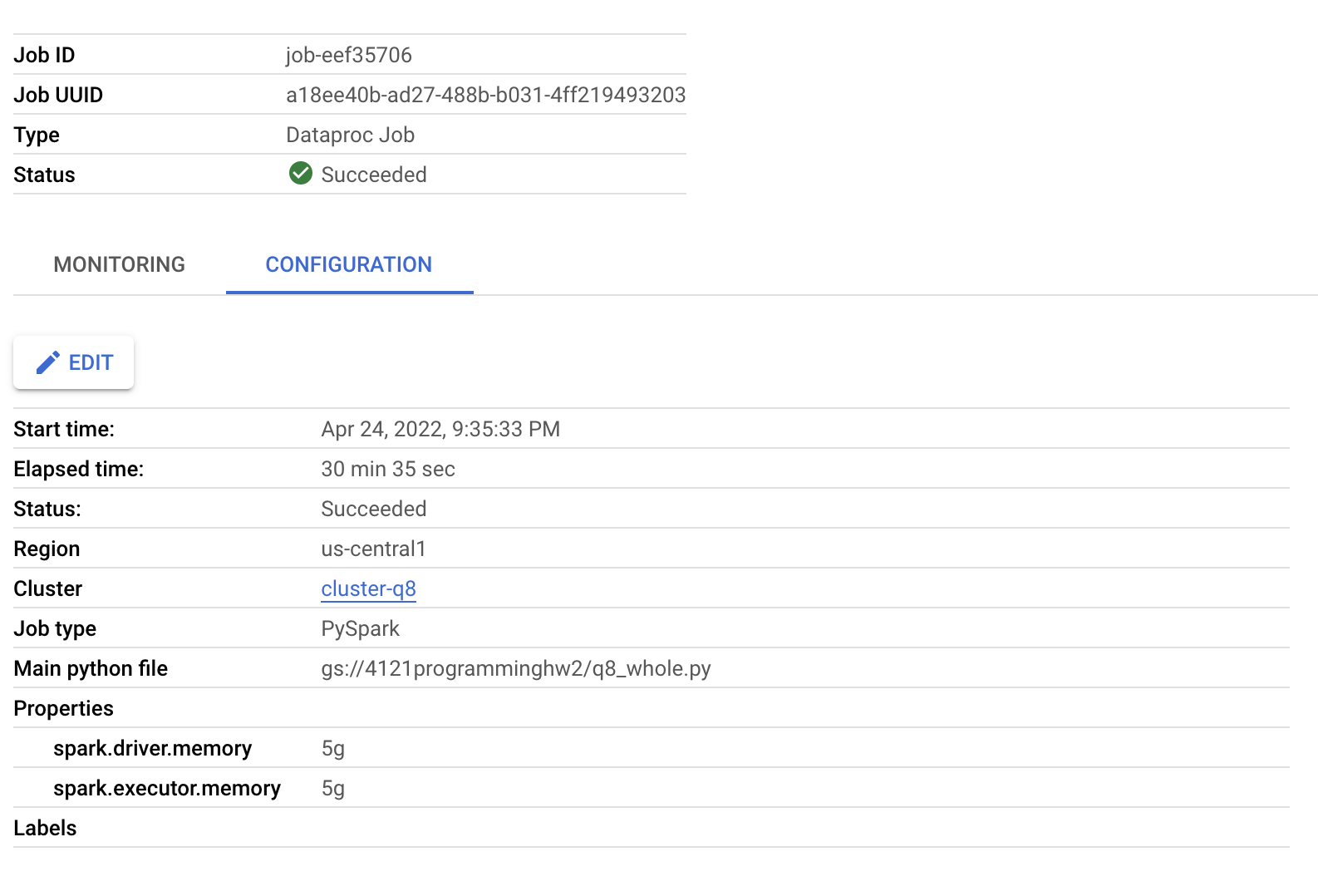
Task 3: Spark PageRank

Question 8

Set the Spark driver memory to 5GB and the Spark executor memory to 5GB whenever you run your PageRank program. Write a script to first run Task 2, and then run Task 3 using the csv output generated by Task 2, and answer the following questions. Always use 10 iterations for the PageRank program. When running Task 2, use enwiki\_whole.xml as input.

Use your output from Task 2 with enwiki\_whole.xml as input, run Task 3 using a 3 node cluster (2 worker nodes). Include your screenshot of the dataproc job. What is the completion time of the task?

**Answer:** 30 min 35 s



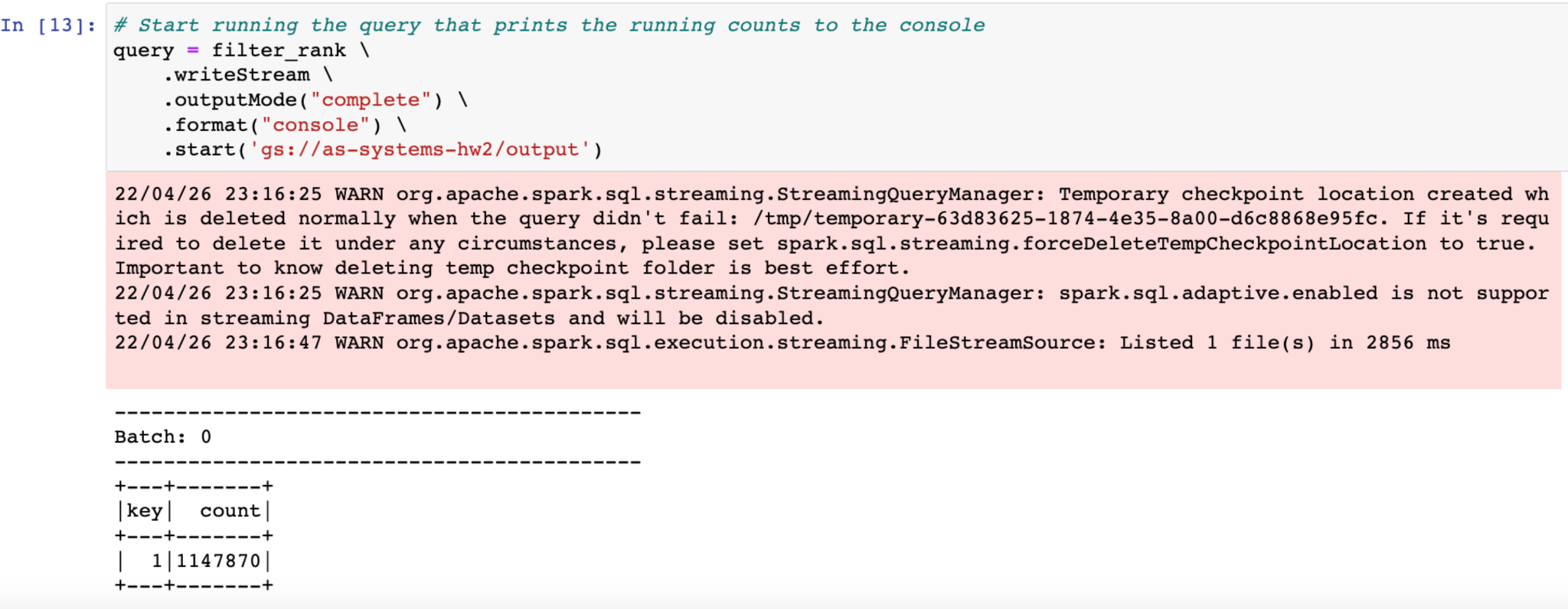
Part 2: Spark Streaming (Extra Credit)

Task 1: Stream Receiver

Question 9

Start a PageRank program you wrote in Part 1 Task 3 whose input is the link graph generated from “enwiki\_whole.xml” and store the output to a directory inside HDFS. Set your stream receiver to read the files generated by the PageRank program. Kill the receiver when the PageRank task is finished. How many articles in the database has a rank greater than **0.5**? You can SSH into the master node and start with ‘hdfs dfs -mv’ or ‘mv’ to help you write your program.

**Answer:** There are 1147870 articles with pagerank greater than 0.5.



Task 2: Stream Emitter

Question 10

Spark Streaming can also be used to send data via TCP sockets. The Emitter in this case will wait on a socket connection request from the receiver, and upon accepting the connection request it will start sending data. Do you think such data server design is feasible and efficient? Briefly explain.

**Answer:** The design is feasible and efficient. That’s because if A transmit the data to B before B manages to connect to the server, partial data that are transferred will be missing. Another possible reason is it is possible to send the data to wrong receivers if there is no TCP socket.

Question 11

How many hours did you spend in this assignment?

**Answer:** 32 hours