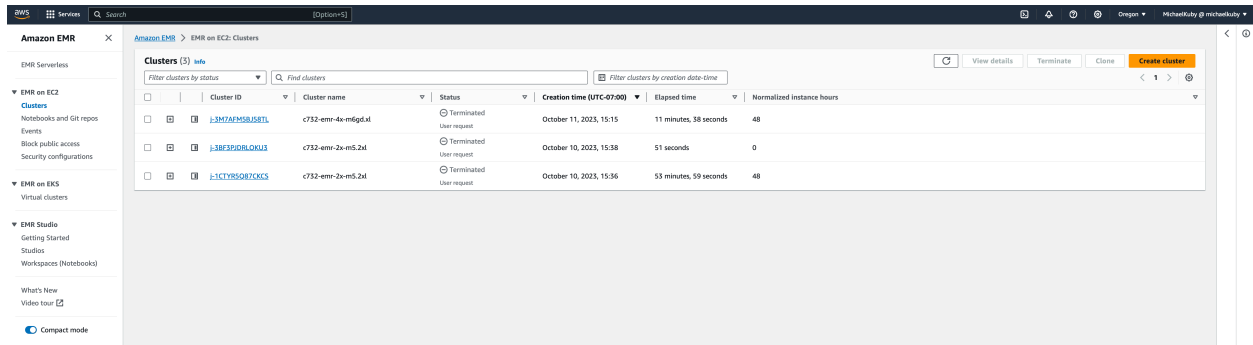


1. Take a screen shot of your list of EMR clusters (if more than one page, only the page with the most recent), showing that all have Terminated status.



The screenshot shows the Amazon EMR console with a list of three clusters, all in a 'Terminated' status. The table includes columns for Cluster ID, Cluster name, Status, Creation time (UTC-07:00), Elapsed time, and Normalized instance hours.

Cluster ID	Cluster name	Status	Creation time (UTC-07:00)	Elapsed time	Normalized instance hours
j-3H7AFH5BGBTL	c732-emr-4a-mfgpl.xl	Terminated User request	October 11, 2023, 15:15	11 minutes, 38 seconds	48
j-3BF3PGBLQJL3	c732-emr-2a-m5.2xl	Terminated User request	October 10, 2023, 15:38	51 seconds	0
j-1CTY8S0B7CXS	c732-emr-2a-m5.2xl	Terminated User request	October 10, 2023, 15:36	53 minutes, 59 seconds	48

2. For Section 2:

- a. What fraction of the input file was prefiltered by S3 before it was sent to Spark?

The original run showed the following data:

Input Size / Records: 2.6 MiB / 3245

Output Size / Records: 27.2 KiB / 3245

The run with S3 filtering showed the following data:

Input Size / Records: 97.7 KiB / 3245

Output Size / Records: 27.2 KiB / 3245

If we assume that $2.6 * 1000 = 2600$ KiB is the original size of the input file, and with S3 filtering, the input is 97.7 KiB, then $1 - (97.7 \text{ KiB} / 2600 \text{ KiB}) = 0.9624$, or approximately 96% of the data was prefiltered before it was sent to Spark. That's pretty impressive.

- b. Comparing the different input numbers for the regular version versus the prefiltered one, what operations were performed by S3 and which ones performed in Spark?

Since Spark creates the plan for computation, instead of procuring ALL of the data from S3 simply to filter out most of it, it can request S3 to do the filtering across the partitions before sending over the data that actually needs to be computed. Reading from <https://docs.aws.amazon.com/emr/latest/ReleaseGuide/emr-spark-s3select.html>, we can see that S3 filtering can be done on the majority of filters except for those that cast, aggregate, or work on columns with complex attributes on objects.

With respect to `weather_etl_s3_select.py`, these filtering operations should be, and do appear to be, being done by S3: Filter out observations where the `qflat` is set to something other than

Null; Keep only records from stations that start with 'CA' (Canadian data); Keep only maximum temperature observations.

We then divide the temperature by 10 to put it into C. Since this requires computation, this will not be done by S3; nor what follows in our program, since it utilizes the results of this computation.

- 3. For Section 3: Look up the hourly costs of the m6gd.xlarge instance on the EC2 On-Demand Pricing page. Estimate the cost of processing a dataset ten times as large as reddit-5 using just those 4 instances. If you wanted instead to process this larger dataset making full use of 16 instances, how would it have to be organized?**

Cost analysis:

For the cost analysis, I will ignore start-up times and focus on the actual run-time of jobs.

The cost per instance-hour is \$0.1808. We used four instances, and my Total Uptime was 4.9 minutes. Billing is done per minute.

Considering a dataset ten times larger than reddit-5, we can estimate the Total Uptime as $4.9 * 10 = 49$ minutes. So, the portion of an hour elapsed is given by $49/60 = 0.817\%$ of an hour.

The cost per instance would be $\$0.1808 * 0.817 = \0.1477138 .

And the total cost for all four instances would be $\$0.1477138 * 4 = \0.5908544 , or \$0.60.

So, we could estimate the cost of processing a dataset ten times as large as reddit-5 using just those four instances at 60 cents.

Full use of 16 instances analysis:

Suppose we want to use 16 instances, each with four cores. In that case, we need to recognize that the degree of parallelism available to us is $16 \text{ instances} * 4 \text{ cores per instance} = 64$ parallelizable cores. We want the number of partitions to be between 1-2 orders of magnitude greater than the number of available cores. Hence, it would be appropriate for the input data to be split between 640 – 6400 separate files.