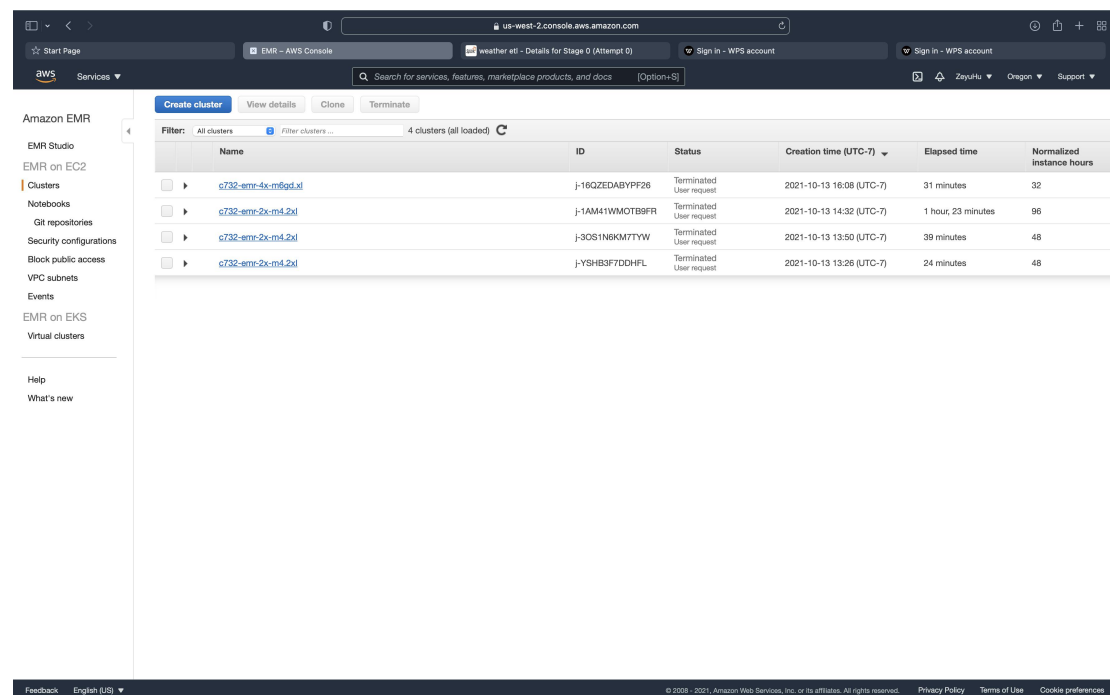


1. Take a screen shot of your list of EMR clusters (if more than one page, only the page with the most recent), showing that all have Terminated status.



The screenshot shows the AWS Management Console for the 'us-west-2' region. The left sidebar shows the 'Amazon EMR' section with 'Clusters' selected. The main area displays a table of EMR clusters. There are 4 clusters listed, all with a 'Terminated' status. The table columns are: Name, ID, Status, Creation time (UTC-7), Elapsed time, and Normalized instance hours.

Name	ID	Status	Creation time (UTC-7)	Elapsed time	Normalized instance hours
c732-eme-4x-m6gd.xl	j-18QZEDABYPF26	Terminated User request	2021-10-13 16:08 (UTC-7)	31 minutes	32
c732-eme-2x-m4.2x1	j-1AM41WMCTB9FR	Terminated User request	2021-10-13 14:32 (UTC-7)	1 hour, 23 minutes	96
c732-eme-2x-m4.2x1	j-3CS1N6KM7TYW	Terminated User request	2021-10-13 13:50 (UTC-7)	39 minutes	48
c732-eme-2x-m4.2x1	j-YSHB3F7DDHFL	Terminated User request	2021-10-13 13:26 (UTC-7)	24 minutes	48

For Section 2:

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

$$1 - 97.7 \text{ KiB} / 2.6 \text{ MiB} = 1 - 3.67\% = 96.33\%$$

So the fraction of the input file prefiltered by S3 was 96.33%.

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?

S3 performed the filter part and the selection part, whereas spark performed "1/10" the calculation part.

For Section 3:

a. Reviewing the job times in the Spark history, which operations took the most time? Is the application IO-bound or compute-bound?

In school cluster, it took 24min to run reddit the most time cost in spark is reduceBy. While in AWS, it took 6.7 min and the most time cost step on AWS is collect. The right and read took over 2 mins in total. I think this application IO bound as Write, read, reduceBy and sortBy all evolve with IO, and those part took a lot of time. So it is IO bound.

b. 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?

It took 6.7 min to calculate, the money cost: $4 \times 6.7 \times 10 \times \$0.1808 / 60 = \$0.8$

If it is 16 instances and each has 4 core, then the total would be $16 \times 4 = 64$, then we should repartition the input into $N \times 64$ (where $N = 2$ or 3) evenly so that each core would be taken full advantage.