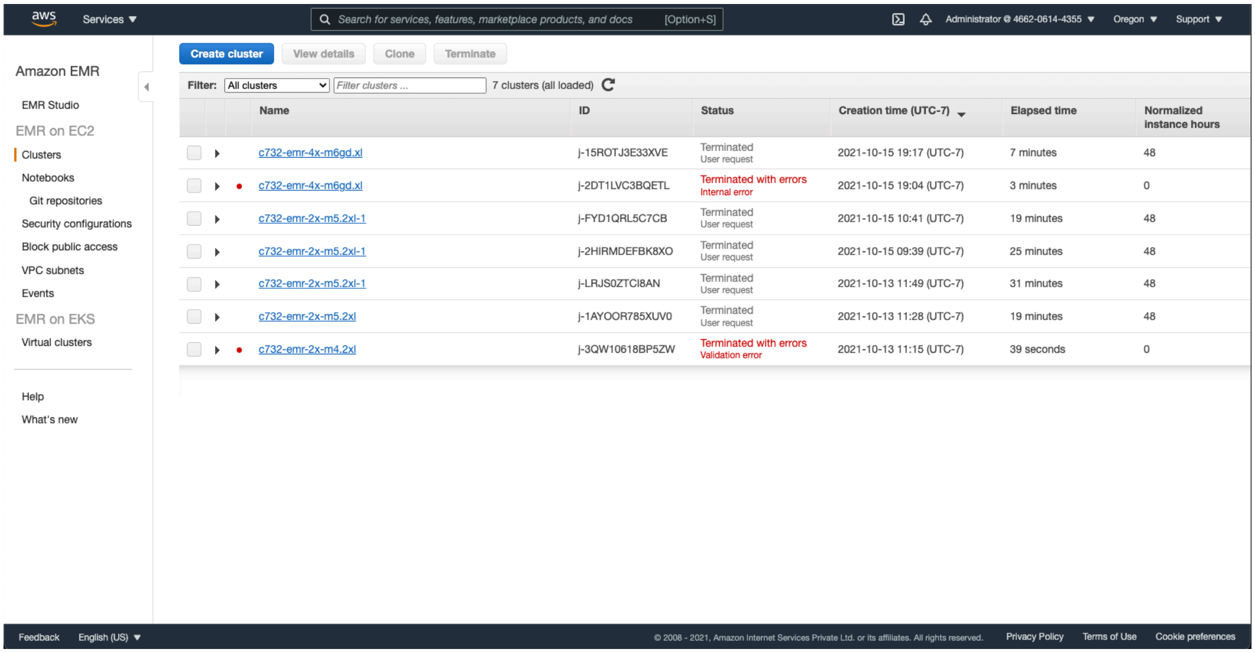
1.



Filter: All clusters | Filter clusters ... | 7 clusters (all loaded)

		Name	ID	Status	Creation time (UTC-7)	Elapsed time	Normalized instance hours
□		c732-emr-4x-m6gd.xl	j-15ROTJ3E33XVE	Terminated User request	2021-10-15 19:17 (UTC-7)	7 minutes	48
□	●	c732-emr-4x-m6gd.xl	j-2DT1LVC3BQETL	Terminated with errors Internal error	2021-10-15 19:04 (UTC-7)	3 minutes	0
□		c732-emr-2x-m5.2xl-1	j-FYD1QRL5C7CB	Terminated User request	2021-10-15 10:41 (UTC-7)	19 minutes	48
□		c732-emr-2x-m5.2xl-1	j-2HIRMDEFBK8XO	Terminated User request	2021-10-15 09:39 (UTC-7)	25 minutes	48
□		c732-emr-2x-m5.2xl-1	j-LRJS0ZTCI8AN	Terminated User request	2021-10-13 11:49 (UTC-7)	31 minutes	48
□		c732-emr-2x-m5.2xl	j-1AYOOR785XUV0	Terminated User request	2021-10-13 11:28 (UTC-7)	19 minutes	48
□	●	c732-emr-2x-m4.2xl	j-3QW10618BP5ZW	Terminated with errors Validation error	2021-10-13 11:15 (UTC-7)	39 seconds	0

2.
a. Almost 96.15% of the input data is prefiltered by S3(only 3.84% is the input data is sent to spark when compared to input data unfiltered by S3)

b. Comparing the filtered with unfiltered ones, almost all the filtering operations (filter by qflag, station, observation) were performed by S3Select and only other operations like divide by 10(to convert to Celsius), selecting only required columns and writing the output as a directory of JSON files in GZIP compressed format.

3.
a. Operations like actions (collect, saveAsTextFile) and shuffle operations (like sortByKey) took most of the time. I think that the application is IO bound as most of the time spent is on IO operations to move the data around the partitions among nodes.

b. The hourly costs for “m6gd.xlarge” instance types is $0.1808. The time consumed for processing reddit-5 dataset is about 3.9 mins. So, the time to process a dataset about 10X the size of reddit-5 dataset will be around 39 mins (because I think using the same config to process larger data will proportionately take larger time), which would cost approximately $0.47008($0.11752 per instance * 4 instances) using the same 4 instances each with 4 cores.

For 4 instances with 4 cores, input data was organized as 16 files, so that each can work with their own data, so in case of 16 instances, there will be 16*4 cores, so the input data has to be organized(partitioned) into 64 files to achieve maximum parallelism and complete the processing much quicker.