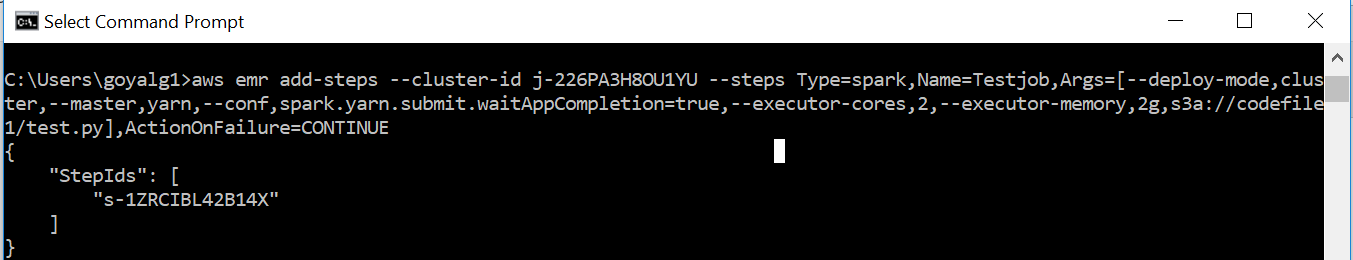
To create an Job at AWS EMR cluster, need to run the below command

----Job creation at EMR

aws emr add-steps --cluster-id j-226PA3H8OU1YU --steps Type=spark,Name=Testjob,Args=[--deploy-mode,cluster,--master,yarn,--conf,spark.yarn.submit.waitAppCompletion=true,--executor-cores,2,--executor-memory,2g,s3a://codefile1/test.py],ActionOnFailure=CONTINUE

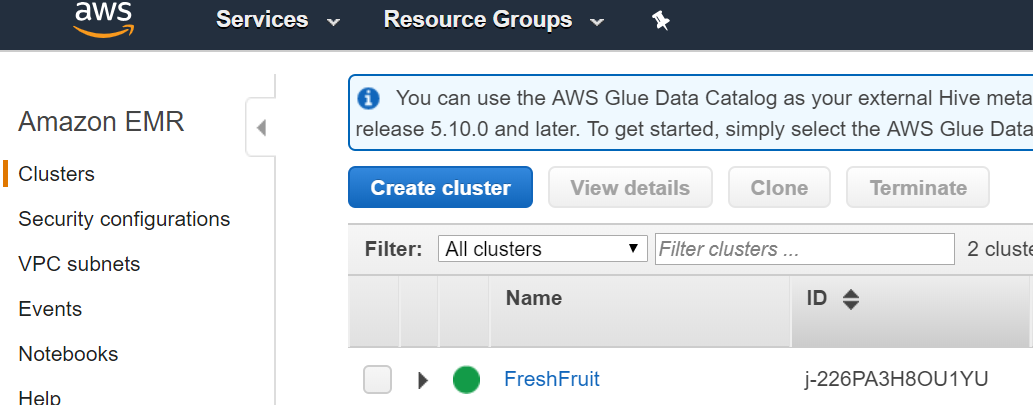
PFB the snapshot for the same:



Meaning of every parameter:

aws emr add-steps: This will be invoking point for creation of EMR job

--cluster-id j-226PA3H8OU1YU: Name of EMR cluster, clusterid can be extracted from AWS console, pfb the snapsot:



--steps Type=spark, Since this is Spark job

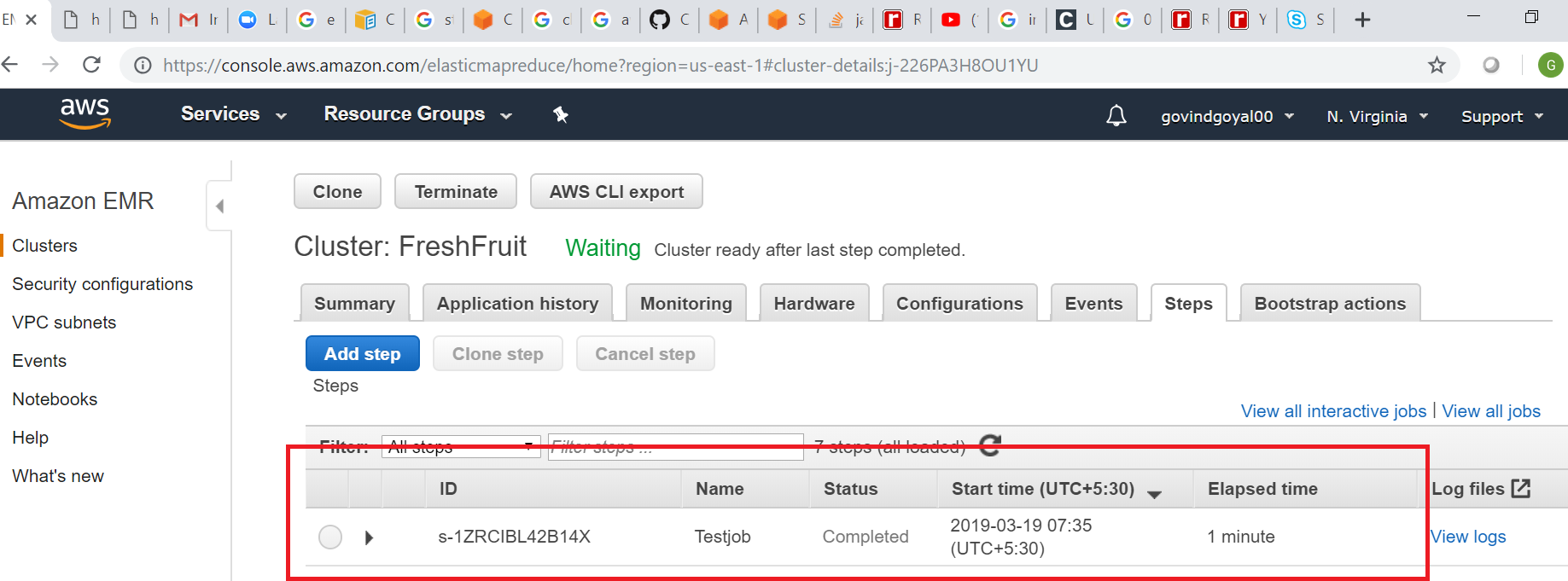
Name=Testjob, Name of Job

Args=[--deploy-mode,cluster,--master,yarn,--conf,spark.yarn.submit.waitAppCompletion=true,--executor-cores,2,--executor-memory,2g,: Arguments that will require for Job to proceed. Allocating 2Core processor and 2GB RAM from parameters: executor-cores,2,--executor-memory,2g, If the data is huge then use the higher parameters.

s3a://codefile1/test.py],: Location of code file

ActionOnFailure=CONTINUE

Once the Job is created, get back to AWS console and check the status:

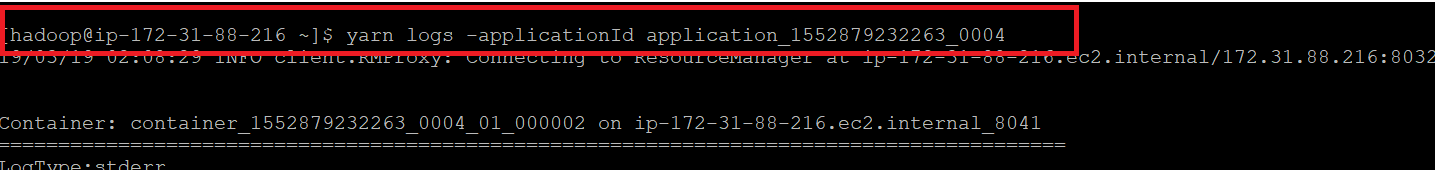


Get the logs for the application from console after clicking on VIEW LOGS. I am attaching the log files for the successful job.



To extract the output from the job, extract the application id(Job id) from any of the above attached files, in our case the name is “application\_1552879232263\_0004”. Connect to AWS EMR console and run the below command:

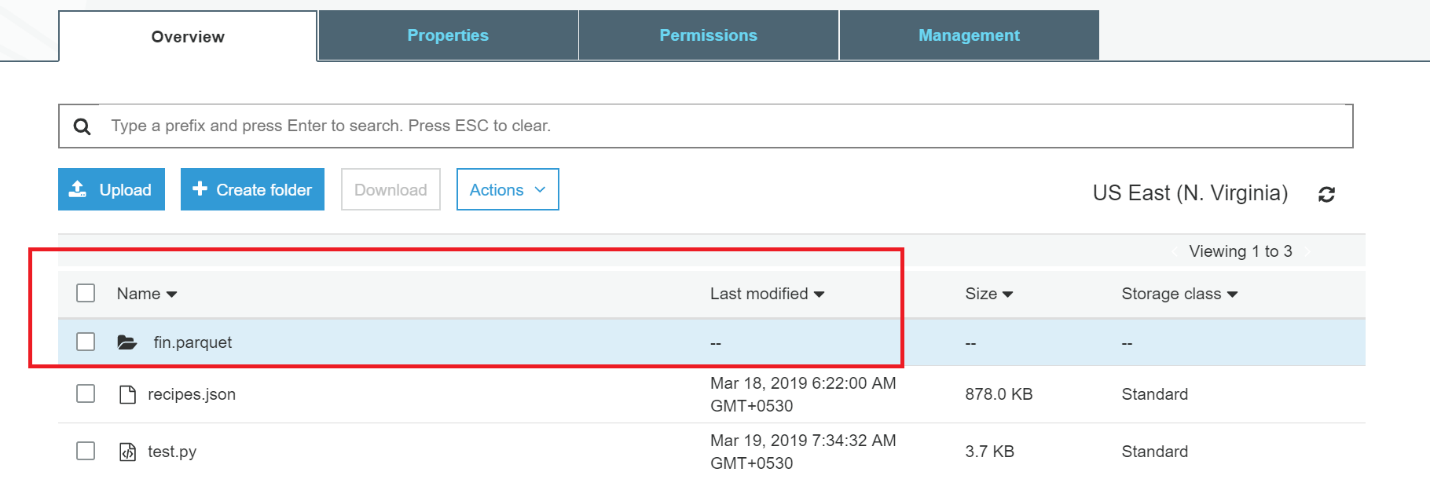
yarn logs -applicationId application\_1552879232263\_0004



Attaching the entire logs and output from Job:



Check at the S3 bucket if the Parquet files are created:



\*Appication ID can be found in the job history, if the job is running then add one to the previous executable job