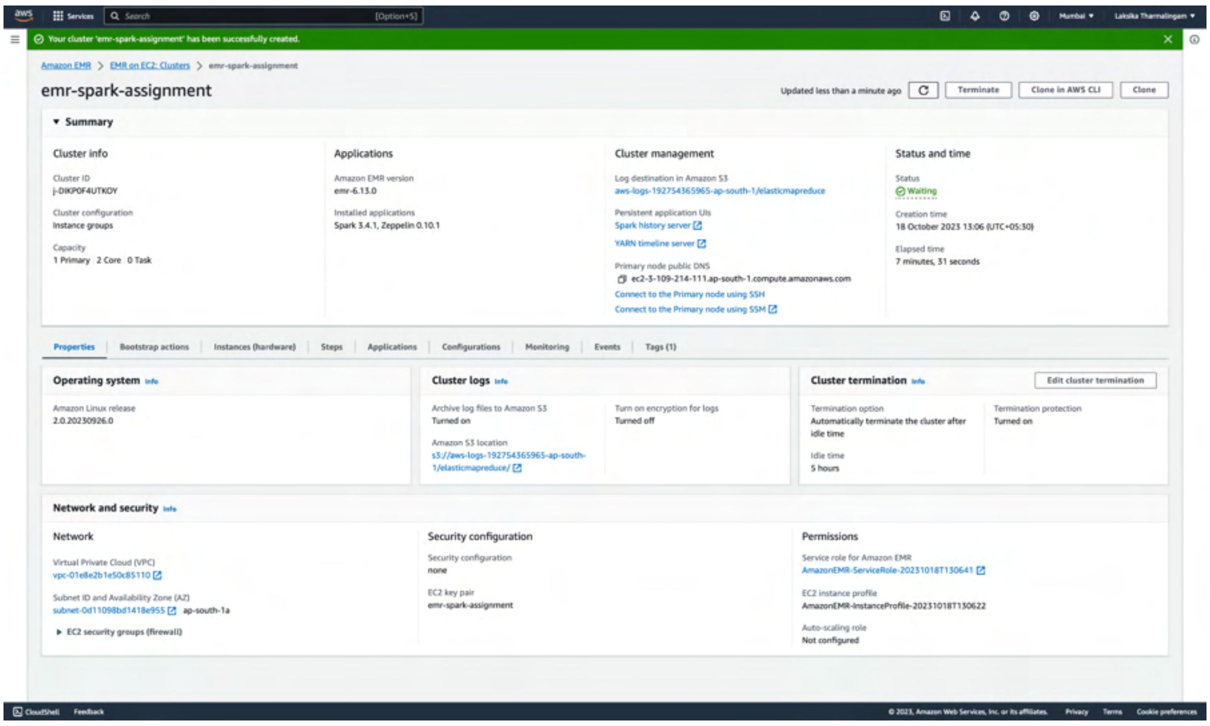
Assignment 01: Big Data Analytics using Hadoop and Spark on Amazon EMR

1. Task 1: Amazon EMR Setup


Your cluster 'emr-spark-assignment' has been successfully created.

Amazon EMR > EMR on EC2: Clusters > emr-spark-assignment

emr-spark-assignment

Updated less than a minute ago

Summary

Cluster info
Cluster ID: j-DIKPOF4UTKOY
Cluster configuration: Instance groups
Capacity: 1 Primary | 2 Core | 0 Task

Applications
Amazon EMR version: emr-6.13.0
Installed applications: Spark 3.4.1, Zeppelin 0.10.1

Cluster management
Log destination in Amazon S3: aws-logs-192754365965-ap-south-1/elasticmapreduce
Persistent application UIs: Spark history server, YARN timeline server
Primary node public DNS: ec2-3-109-214-111.ap-south-1.compute.amazonaws.com
Connect to the Primary node using SSH
Connect to the Primary node using SSM

Status and time
Status: Waiting
Creation time: 18 October 2023 13:06 (UTC+05:30)
Elapsed time: 7 minutes, 31 seconds

Operating system
Amazon Linux release: 2.0.20230926.0

Cluster logs
Archive log files to Amazon S3: Turned on
Turn on encryption for logs: Turned off
Amazon S3 location: s3://aws-logs-192754365965-ap-south-1/elasticmapreduce/

Cluster termination
Termination option: Automatically terminate the cluster after idle time
Termination protection: Turned on
Idle time: 5 hours

Network and security

Network
Virtual Private Cloud (VPC): vpc-01e8e2b1e50c85110
Subnet ID and Availability Zone (AZ): subnet-0d11098bd1418e955 ap-south-1a
EC2 security groups (firewall)

Security configuration
Security configuration: none
EC2 key pair: emr-spark-assignment

Permissions
Service role for Amazon EMR: AmazonEMR-ServiceRole-20231018T130641
EC2 instance profile: AmazonEMR-InstanceProfile-20231018T130622
Auto-scaling role: Not configured

2. Task 2: Importing and Managing Data in Amazon S3

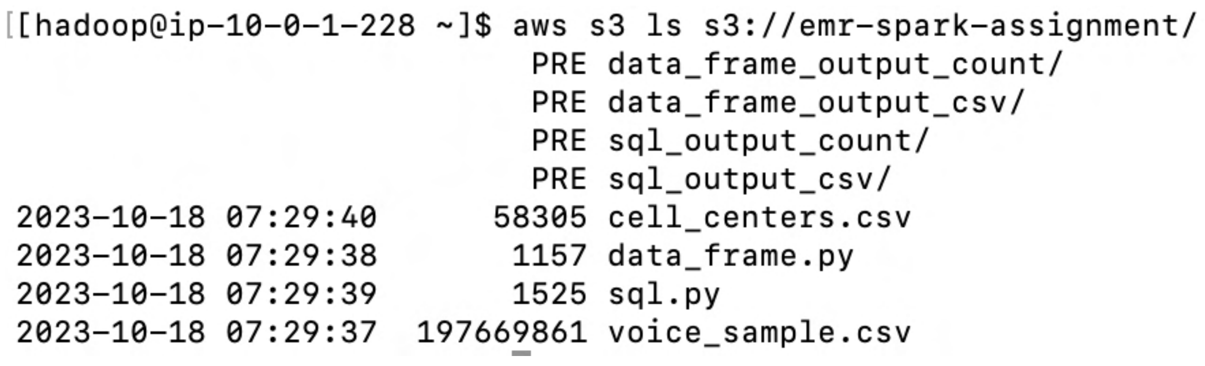
Commands used to upload :

```
aws s3 cp /Users/laksika/Desktop/cell_centers.csv s3://emr-spark-assignment
aws s3 cp /Users/laksika/Desktop/voice_sample.csv s3://emr-spark-assignment
aws s3 cp /Users/laksika/Desktop/sql.py s3://emr-spark-assignment
aws s3 cp /Users/laksika/Desktop/data_frame.py s3://emr-spark-assignment
```

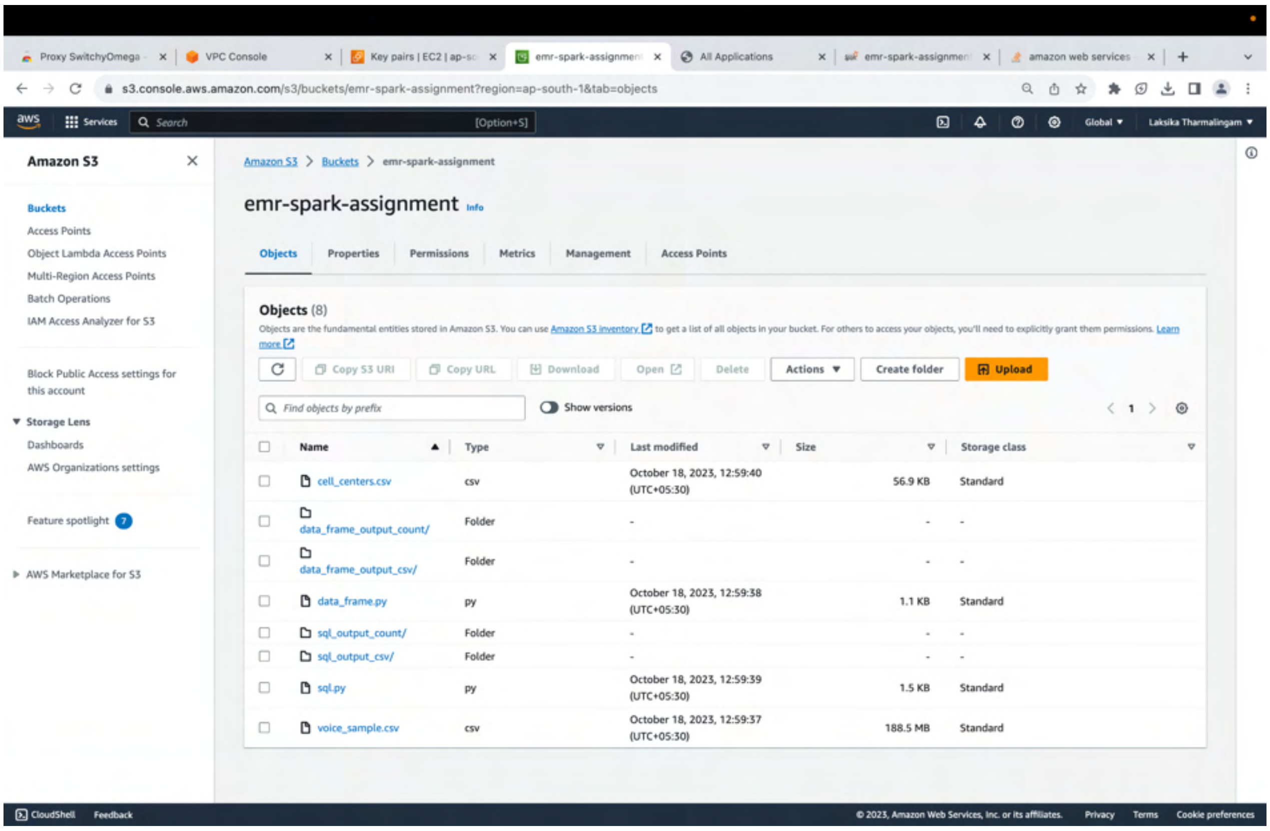
```
(base) laksika@Laksikas-MBP Desktop % aws s3 cp /Users/laksika/Desktop/cell_centers.csv s3://emr-spark-assignment
upload: ./cell_centers.csv to s3://emr-spark-assignment/cell_centers.csv
```

Commands used to list :

```
aws s3 ls s3://emr-spark-assignment
aws s3 ls s3://emr-spark-assignment --recursive
```



```
[[hadoop@ip-10-0-1-228 ~]$ aws s3 ls s3://emr-spark-assignment/
                           PRE data_frame_output_count/
                           PRE data_frame_output_csv/
                           PRE sql_output_count/
                           PRE sql_output_csv/
2023-10-18 07:29:40      58305 cell_centers.csv
2023-10-18 07:29:38       1157 data_frame.py
2023-10-18 07:29:39       1525 sql.py
2023-10-18 07:29:37  197669861 voice_sample.csv
```



s3.console.aws.amazon.com/s3/buckets/emr-spark-assignment?region=ap-south-1&tab=objects

Amazon S3 > Buckets > emr-spark-assignment

emr-spark-assignment

Objects (8)

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

Name	Type	Last modified	Size	Storage class
cell_centers.csv	csv	October 18, 2023, 12:59:40 (UTC+05:30)	56.9 KB	Standard
data_frame_output_count/	Folder	-	-	-
data_frame_output_csv/	Folder	-	-	-
data_frame.py	py	October 18, 2023, 12:59:38 (UTC+05:30)	1.1 KB	Standard
sql_output_count/	Folder	-	-	-
sql_output_csv/	Folder	-	-	-
sql.py	py	October 18, 2023, 12:59:39 (UTC+05:30)	1.5 KB	Standard
voice_sample.csv	csv	October 18, 2023, 12:59:37 (UTC+05:30)	188.5 MB	Standard

© 2023, Amazon Web Services, Inc. or its affiliates.

3. Task 3: Data Processing with Apache Spark via Amazon EMR

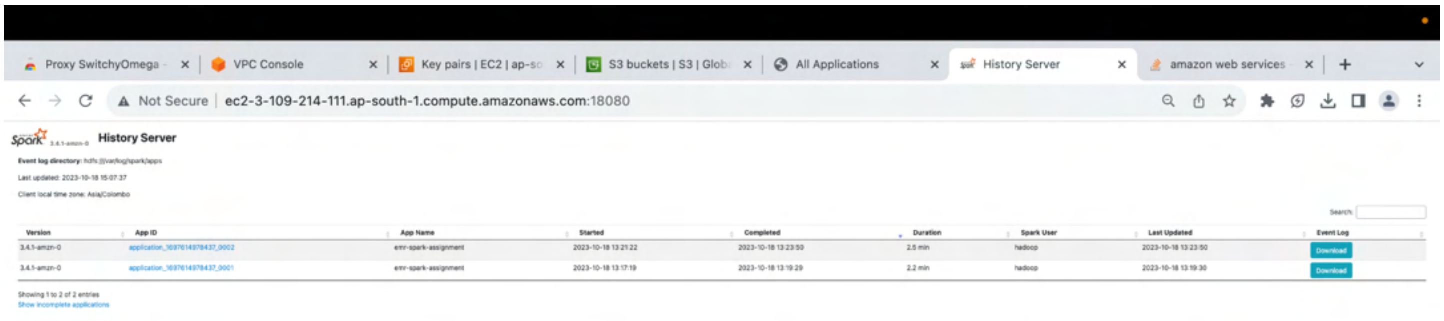
3.1. Spark DataFrame Approach

```
(
  df1
  .join(df2, ['LOCATION_ID'])
  .filter(df2['PROVINCE_NAME'] == 'Western')
  .groupBy('CALLER_ID').agg(countDistinct('CALL_DATE').alias('COUNT_OF_DISTINCT_CALL_DATE'))
  .filter(col('COUNT_OF_DISTINCT_CALL_DATE') == df1.select('CALL_DATE').distinct().count())
  .select('CALLER_ID')
)
```

3.2. Spark SQL Approach

```
spark.sql(
  '''
  SELECT CALLER_ID
  FROM (
     SELECT CALLER_ID, COUNT(DISTINCT CALL_DATE) AS COUNT_OF_DISTINCT_CALL_DATE
     FROM (
        SELECT CALLER_ID, CALL_DATE
        FROM view_of_voice_sample
        WHERE LOCATION_ID IN (SELECT LOCATION_ID FROM view_of_cell_centers WHERE PROVINCE_NAME = 'Western')
     )
     GROUP BY CALLER_ID
  )
  WHERE COUNT_OF_DISTINCT_CALL_DATE = (SELECT COUNT(DISTINCT CALL_DATE) FROM view_of_voice_sample)
  '''
)
```

3.3. Analysis and Comparison



Not Secure | ec2-3-109-214-111.ap-south-1.compute.amazonaws.com:18080

History Server

Event log directory: hdfs:///var/log/spark/apps

Last updated: 2023-10-18 15:07:37

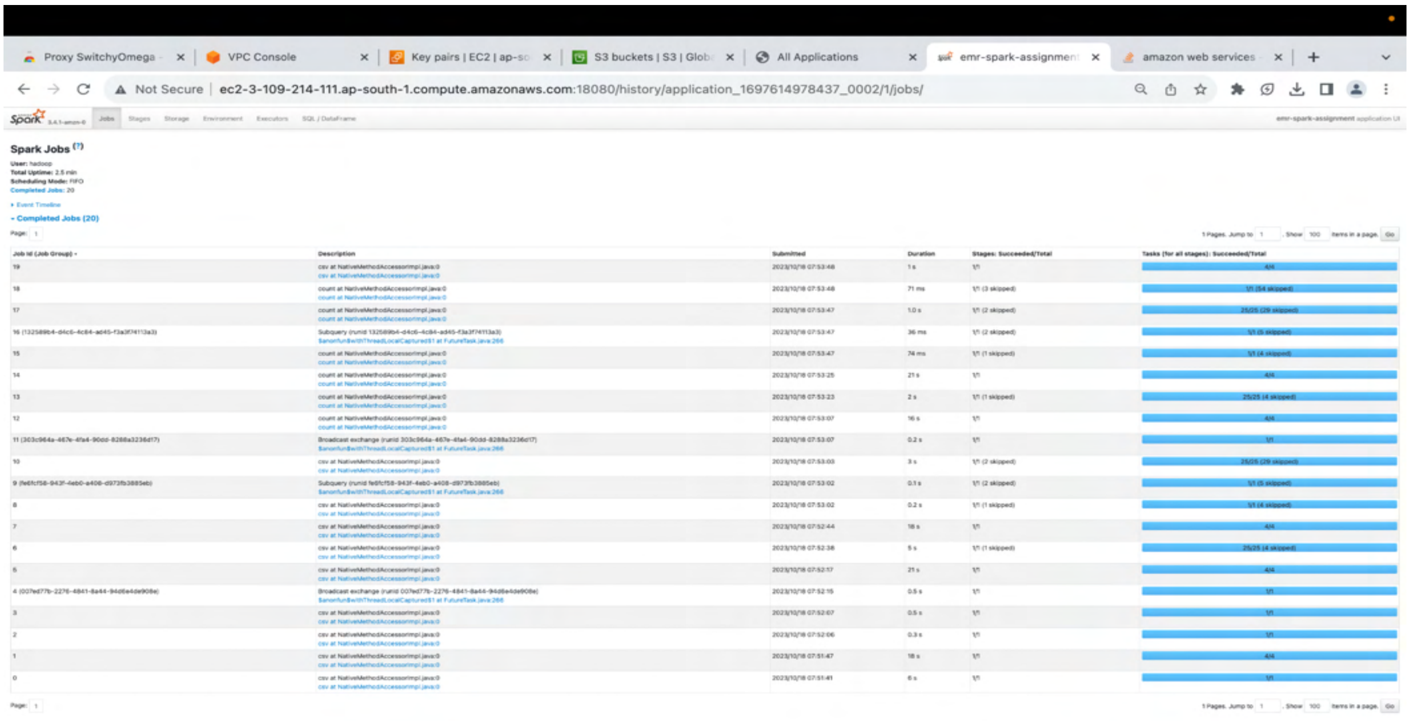
Client local time zone: Asia/Colombo

Version	App ID	App Name	Started	Completed	Duration	Spark User	Last Updated	Event Log
3.4.1-amzn-0	application_1697614978437_0002	emr-spark-assignment	2023-10-18 13:21:22	2023-10-18 13:23:50	2.5 min	hadoop	2023-10-18 13:23:50	Download
3.4.1-amzn-0	application_1697614978437_0001	emr-spark-assignment	2023-10-18 13:17:19	2023-10-18 13:19:29	2.2 min	hadoop	2023-10-18 13:19:30	Download

Showing 1 to 2 of 2 entries

Show incomplete applications

3.3.1. Spark SQL



Not Secure | ec2-3-109-214-111.ap-south-1.compute.amazonaws.com:18080/history/application_1697614978437_0002/1/jobs/

Spark Jobs (?)

User: hadoop
Total Uptime: 2.5 min
Scheduling Mode: FIFO
Completed Jobs: 20

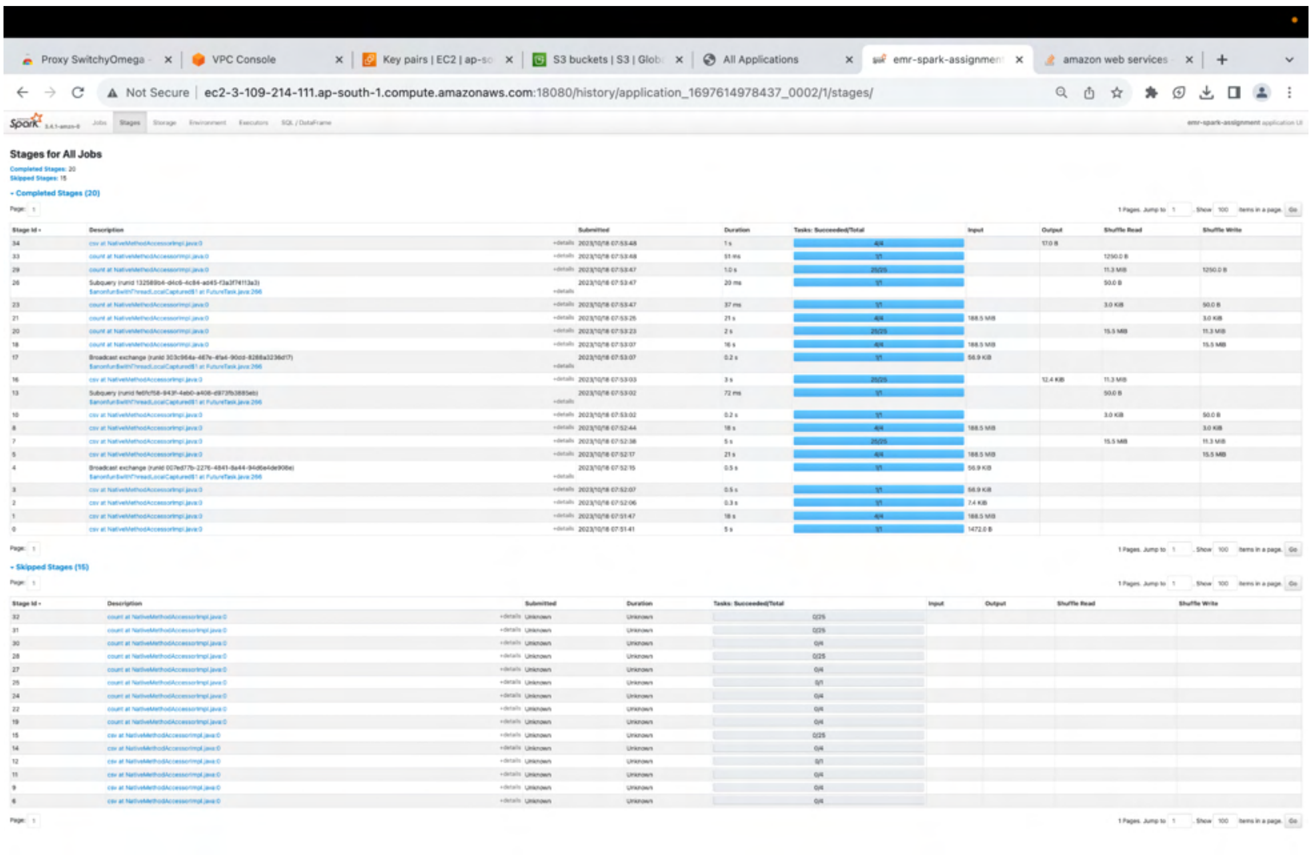
▸ Event Timeline

▾ Completed Jobs (20)

Page: 1 — 1 Pages. Jump to 1 . Show 100 items in a page. Go

Job Id (Job Group) ▾	Description	Submitted	Duration	Stages: Succeeded/Total	Tasks (for all stages): Succeeded/Total
19	csv at NativeMethodAccessorImpl.java:0 csv at NativeMethodAccessorImpl.java:0	2023/10/18 07:53:48	1 s	1/1	4/4
18	count at NativeMethodAccessorImpl.java:0 count at NativeMethodAccessorImpl.java:0	2023/10/18 07:53:48	71 ms	1/1 (3 skipped)	1/1 (54 skipped)
17	count at NativeMethodAccessorImpl.java:0 count at NativeMethodAccessorImpl.java:0	2023/10/18 07:53:47	1.0 s	1/1 (2 skipped)	25/25 (29 skipped)
16 (132589b4-d4c6-4c84-ad45-f3a3f74113a3)	Subquery (runId 132589b4-d4c6-4c84-ad45-f3a3f74113a3) $anonfun$withThreadLocalCaptured$1 at FutureTask.java:266	2023/10/18 07:53:47	36 ms	1/1 (2 skipped)	1/1 (5 skipped)
15	count at NativeMethodAccessorImpl.java:0 count at NativeMethodAccessorImpl.java:0	2023/10/18 07:53:47	74 ms	1/1 (1 skipped)	1/1 (4 skipped)
14	count at NativeMethodAccessorImpl.java:0 count at NativeMethodAccessorImpl.java:0	2023/10/18 07:53:25	21 s	1/1	4/4
13	count at NativeMethodAccessorImpl.java:0 count at NativeMethodAccessorImpl.java:0	2023/10/18 07:53:23	2 s	1/1 (1 skipped)	25/25 (4 skipped)
12	count at NativeMethodAccessorImpl.java:0 count at NativeMethodAccessorImpl.java:0	2023/10/18 07:53:07	16 s	1/1	4/4
11 (303c964a-467e-4fa4-90dd-8288a3236d17)	Broadcast exchange (runId 303c964a-467e-4fa4-90dd-8288a3236d17) $anonfun$withThreadLocalCaptured$1 at FutureTask.java:266	2023/10/18 07:53:07	0.2 s	1/1	1/1
10	csv at NativeMethodAccessorImpl.java:0 csv at NativeMethodAccessorImpl.java:0	2023/10/18 07:53:03	3 s	1/1 (2 skipped)	25/25 (29 skipped)
9 (fe6fcf58-943f-4eb0-a408-d973fb3885eb)	Subquery (runId fe6fcf58-943f-4eb0-a408-d973fb3885eb) $anonfun$withThreadLocalCaptured$1 at FutureTask.java:266	2023/10/18 07:53:02	0.1 s	1/1 (2 skipped)	1/1 (5 skipped)
8	csv at NativeMethodAccessorImpl.java:0 csv at NativeMethodAccessorImpl.java:0	2023/10/18 07:53:02	0.2 s	1/1 (1 skipped)	1/1 (4 skipped)
7	csv at NativeMethodAccessorImpl.java:0 csv at NativeMethodAccessorImpl.java:0	2023/10/18 07:52:44	18 s	1/1	4/4
6	csv at NativeMethodAccessorImpl.java:0 csv at NativeMethodAccessorImpl.java:0	2023/10/18 07:52:38	5 s	1/1 (1 skipped)	25/25 (4 skipped)
5	csv at NativeMethodAccessorImpl.java:0 csv at NativeMethodAccessorImpl.java:0	2023/10/18 07:52:17	21 s	1/1	4/4
4 (007ed77b-2276-4841-8a44-94d6e4de908e)	Broadcast exchange (runId 007ed77b-2276-4841-8a44-94d6e4de908e) $anonfun$withThreadLocalCaptured$1 at FutureTask.java:266	2023/10/18 07:52:15	0.5 s	1/1	1/1
3	csv at NativeMethodAccessorImpl.java:0 csv at NativeMethodAccessorImpl.java:0	2023/10/18 07:52:07	0.5 s	1/1	1/1
2	csv at NativeMethodAccessorImpl.java:0 csv at NativeMethodAccessorImpl.java:0	2023/10/18 07:52:06	0.3 s	1/1	1/1
1	csv at NativeMethodAccessorImpl.java:0 csv at NativeMethodAccessorImpl.java:0	2023/10/18 07:51:47	18 s	1/1	4/4
0	csv at NativeMethodAccessorImpl.java:0 csv at NativeMethodAccessorImpl.java:0	2023/10/18 07:51:41	6 s	1/1	1/1

Page: 1 — 1 Pages. Jump to 1 . Show 100 items in a page. Go


Proxy SwitchyOmega | VPC Console | Key pairs | EC2 | ap-so | S3 buckets | S3 | Glob | All Applications | emr-spark-assignment | amazon web services

Not Secure | ec2-3-109-214-111.ap-south-1.compute.amazonaws.com:18080/history/application_1697614978437_0002/1/stages/

Spark 3.4.1-amzn-0 Jobs Stages Storage Environment Executors SQL / DataFrame — emr-spark-assignment application UI

Stages for All Jobs

Completed Stages: 20
Skipped Stages: 15

- Completed Stages (20)

Page: 1 — 1 Pages. Jump to 1 . Show 100 items in a page. Go

Stage Id	Description	Submitted	Duration	Tasks: Succeeded/Total	Input	Output	Shuffle Read	Shuffle Write
34	csv at NativeMethodAccessorImpl.java:0	2023/10/18 07:53:48	1 s	4/4		17.0 B		
33	count at NativeMethodAccessorImpl.java:0	2023/10/18 07:53:48	51 ms	1/1			1250.0 B	
29	count at NativeMethodAccessorImpl.java:0	2023/10/18 07:53:47	1.0 s	25/25			11.3 MiB	1250.0 B
26	Subquery (runId 132589b4-d4c6-4c84-ad45-f3a3f74113a3) $anonfun$withThreadLocalCaptured$1 at FutureTask.java:266	2023/10/18 07:53:47	20 ms	1/1			50.0 B	
23	count at NativeMethodAccessorImpl.java:0	2023/10/18 07:53:47	37 ms	1/1			3.0 KiB	50.0 B
21	count at NativeMethodAccessorImpl.java:0	2023/10/18 07:53:26	21 s	4/4	188.5 MiB			3.0 KiB
20	count at NativeMethodAccessorImpl.java:0	2023/10/18 07:53:23	2 s	25/25			15.5 MiB	11.3 MiB
18	count at NativeMethodAccessorImpl.java:0	2023/10/18 07:53:07	16 s	4/4	188.5 MiB			15.5 MiB
17	Broadcast exchange (runId 303c964a-467e-4fa4-90dd-8288a3236d17) $anonfun$withThreadLocalCaptured$1 at FutureTask.java:266	2023/10/18 07:53:07	0.2 s	1/1	56.9 KiB			
16	csv at NativeMethodAccessorImpl.java:0	2023/10/18 07:53:03	3 s	25/25		12.4 KiB	11.3 MiB	
13	Subquery (runId fe6fcf58-943f-4eb0-a408-d973fb3885eb) $anonfun$withThreadLocalCaptured$1 at FutureTask.java:266	2023/10/18 07:53:02	72 ms	1/1			50.0 B	
10	csv at NativeMethodAccessorImpl.java:0	2023/10/18 07:53:02	0.2 s	1/1			3.0 KiB	50.0 B
8	csv at NativeMethodAccessorImpl.java:0	2023/10/18 07:52:44	18 s	4/4	188.5 MiB			3.0 KiB
7	csv at NativeMethodAccessorImpl.java:0	2023/10/18 07:52:38	5 s	25/25			15.5 MiB	11.3 MiB
5	csv at NativeMethodAccessorImpl.java:0	2023/10/18 07:52:17	21 s	4/4	188.5 MiB			15.5 MiB
4	Broadcast exchange (runId 007ed77b-2276-4841-8a44-94d6e4de908e) $anonfun$withThreadLocalCaptured$1 at FutureTask.java:266	2023/10/18 07:52:15	0.5 s	1/1	56.9 KiB			
3	csv at NativeMethodAccessorImpl.java:0	2023/10/18 07:52:07	0.5 s	1/1	56.9 KiB			
2	csv at NativeMethodAccessorImpl.java:0	2023/10/18 07:52:06	0.3 s	1/1	7.4 KiB			
1	csv at NativeMethodAccessorImpl.java:0	2023/10/18 07:51:47	18 s	4/4	188.5 MiB			
0	csv at NativeMethodAccessorImpl.java:0	2023/10/18 07:51:41	5 s	1/1	1472.0 B			

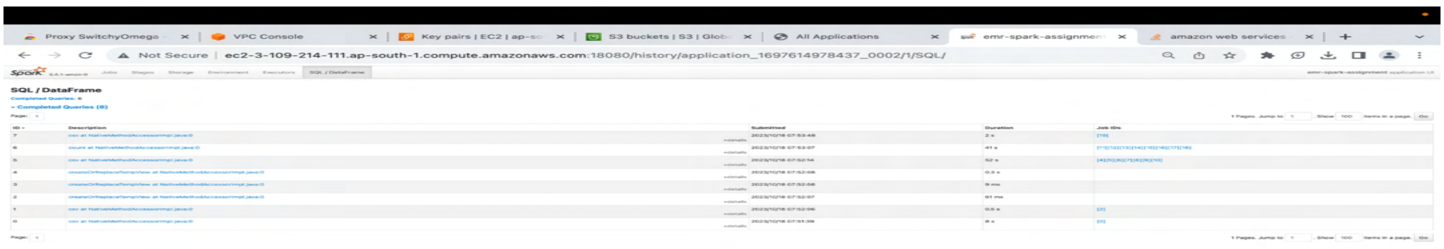
Page: 1 — 1 Pages. Jump to 1 . Show 100 items in a page. Go

- Skipped Stages (15)

Page: 1 — 1 Pages. Jump to 1 . Show 100 items in a page. Go

Stage Id	Description	Submitted	Duration	Tasks: Succeeded/Total	Input	Output	Shuffle Read	Shuffle Write
32	count at NativeMethodAccessorImpl.java:0	Unknown	Unknown	0/25				
31	count at NativeMethodAccessorImpl.java:0	Unknown	Unknown	0/25				
30	count at NativeMethodAccessorImpl.java:0	Unknown	Unknown	0/4				
28	count at NativeMethodAccessorImpl.java:0	Unknown	Unknown	0/25				
27	count at NativeMethodAccessorImpl.java:0	Unknown	Unknown	0/4				
25	count at NativeMethodAccessorImpl.java:0	Unknown	Unknown	0/1				
24	count at NativeMethodAccessorImpl.java:0	Unknown	Unknown	0/4				
22	count at NativeMethodAccessorImpl.java:0	Unknown	Unknown	0/4				
19	count at NativeMethodAccessorImpl.java:0	Unknown	Unknown	0/4				
15	csv at NativeMethodAccessorImpl.java:0	Unknown	Unknown	0/25				
14	csv at NativeMethodAccessorImpl.java:0	Unknown	Unknown	0/4				
12	csv at NativeMethodAccessorImpl.java:0	Unknown	Unknown	0/1				
11	csv at NativeMethodAccessorImpl.java:0	Unknown	Unknown	0/4				
9	csv at NativeMethodAccessorImpl.java:0	Unknown	Unknown	0/4				
6	csv at NativeMethodAccessorImpl.java:0	Unknown	Unknown	0/4				

Page: 1 — 1 Pages. Jump to 1 . Show 100 items in a page. Go


Not Secure | ec2-3-109-214-111.ap-south-1.compute.amazonaws.com:18080/history/application_1697614978437_0002/1/SQL/

Spark 3.4.1-amzn-0 Jobs Stages Storage Environment Executors SQL / DataFrame — emr-spark-assignment application UI

SQL / DataFrame

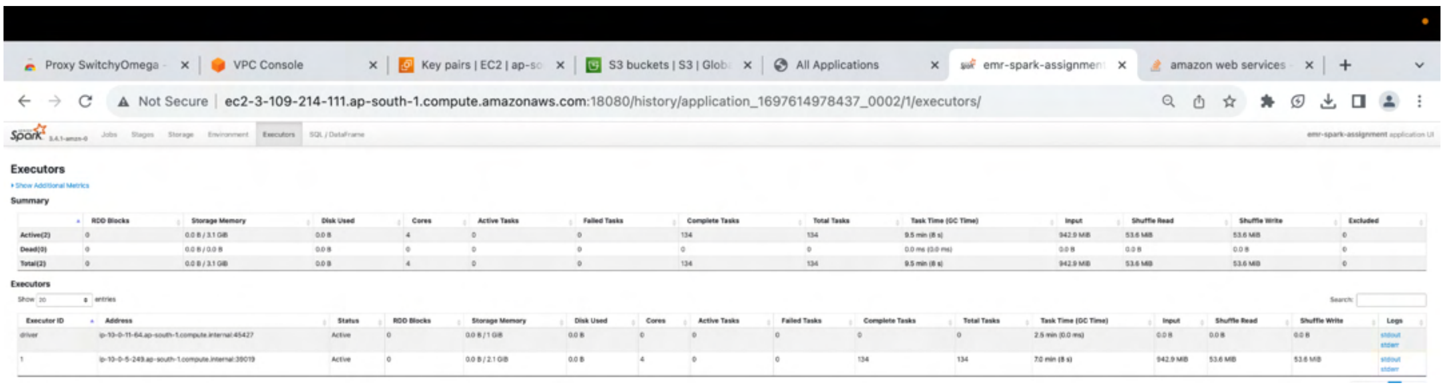
Completed Queries: 8

- Completed Queries (8)

Page: 1 — 1 Pages. Jump to 1 . Show 100 items in a page. Go

ID	Description	Submitted	Duration	Job IDs
7	csv at NativeMethodAccessorImpl.java:0	2023/10/18 07:53:48	2 s	[19]
6	count at NativeMethodAccessorImpl.java:0	2023/10/18 07:53:07	41 s	[11][12][13][14][15][16][17][18]
5	csv at NativeMethodAccessorImpl.java:0	2023/10/18 07:52:14	52 s	[4][5][6][7][8][9][10]
4	createOrReplaceTempView at NativeMethodAccessorImpl.java:0	2023/10/18 07:52:08	0.3 s	
3	createOrReplaceTempView at NativeMethodAccessorImpl.java:0	2023/10/18 07:52:08	9 ms	
2	createOrReplaceTempView at NativeMethodAccessorImpl.java:0	2023/10/18 07:52:07	61 ms	
1	csv at NativeMethodAccessorImpl.java:0	2023/10/18 07:52:06	0.5 s	[2]
0	csv at NativeMethodAccessorImpl.java:0	2023/10/18 07:51:39	8 s	[0]

Page: 1 — 1 Pages. Jump to 1 . Show 100 items in a page. Go


Not Secure | ec2-3-109-214-111.ap-south-1.compute.amazonaws.com:18080/history/application_1697614978437_0002/1/executors/

Spark 3.4.1-amzn-0 Jobs Stages Storage Environment Executors SQL / DataFrame — emr-spark-assignment application UI

Executors

▸ Show Additional Metrics

Summary

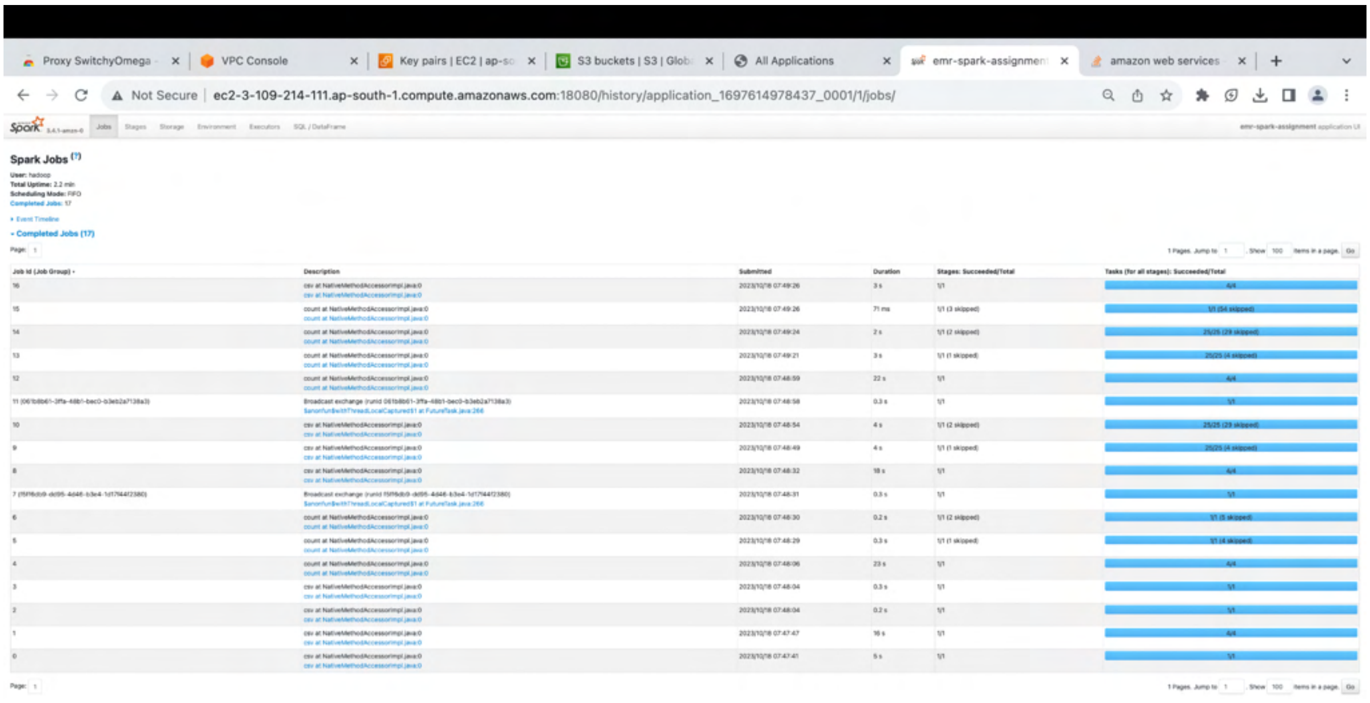
	RDD Blocks	Storage Memory	Disk Used	Cores	Active Tasks	Failed Tasks	Complete Tasks	Total Tasks	Task Time (GC Time)	Input	Shuffle Read	Shuffle Write	Excluded
Active(2)	0	0.0 B / 3.1 GiB	0.0 B	4	0	0	134	134	9.5 min (8 s)	942.9 MiB	53.6 MiB	53.6 MiB	0
Dead(0)	0	0.0 B / 0.0 B	0.0 B	0	0	0	0	0	0.0 ms (0.0 ms)	0.0 B	0.0 B	0.0 B	0
Total(2)	0	0.0 B / 3.1 GiB	0.0 B	4	0	0	134	134	9.5 min (8 s)	942.9 MiB	53.6 MiB	53.6 MiB	0

Executors

Show 20 entries — Search:

Executor ID	Address	Status	RDD Blocks	Storage Memory	Disk Used	Cores	Active Tasks	Failed Tasks	Complete Tasks	Total Tasks	Task Time (GC Time)	Input	Shuffle Read	Shuffle Write	Logs
driver	ip-10-0-11-64.ap-south-1.compute.internal:45427	Active	0	0.0 B / 1 GiB	0.0 B	0	0	0	0	0	2.5 min (0.0 ms)	0.0 B	0.0 B	0.0 B	stdout stderr
1	ip-10-0-5-249.ap-south-1.compute.internal:39019	Active	0	0.0 B / 2.1 GiB	0.0 B	4	0	0	134	134	7.0 min (8 s)	942.9 MiB	53.6 MiB	53.6 MiB	stdout stderr

3.3.2. DataFrame API

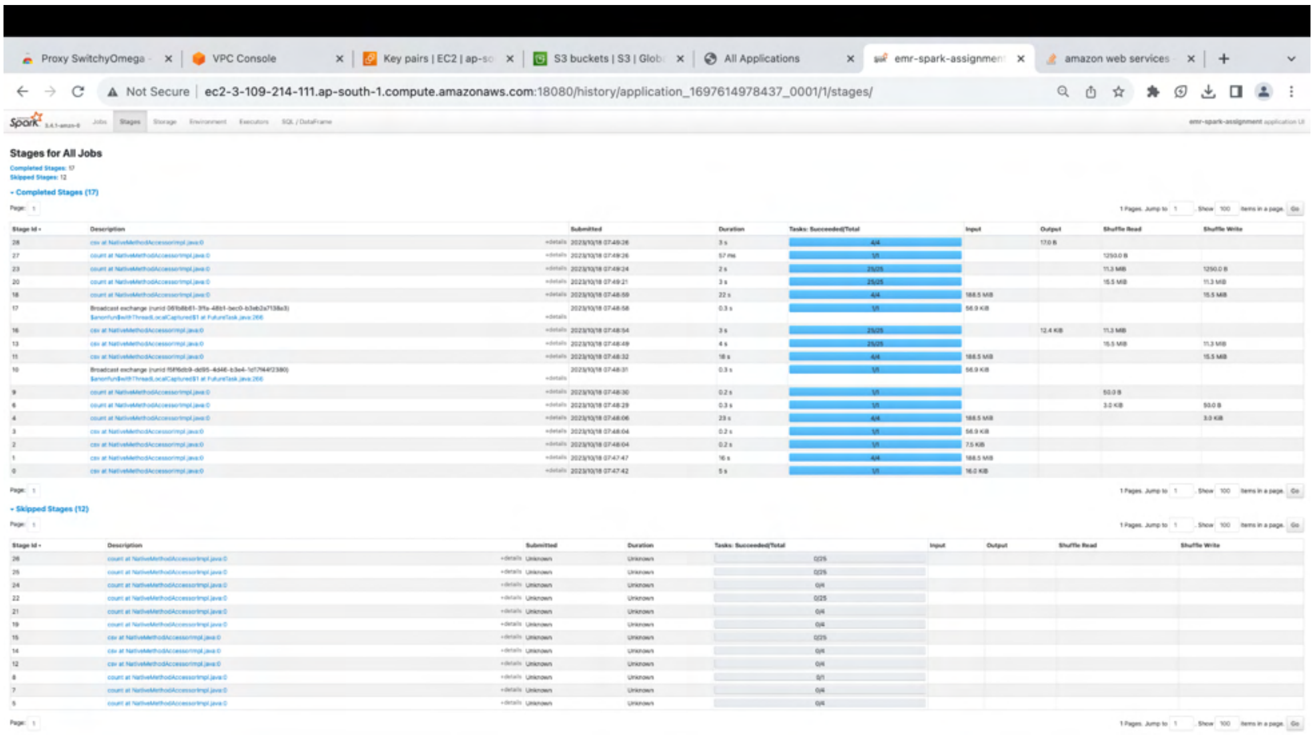

ec2-3-109-214-111.ap-south-1.compute.amazonaws.com:18080/history/application_1697614978437_0001/1/jobs/

Spark Jobs (?)

User: hadoop
Total Uptime: 2.2 min
Scheduling Mode: FIFO
Completed Jobs: 17

Completed Jobs (17)

Job Id (Job Group)	Description	Submitted	Duration	Stages: Succeeded/Total	Tasks (for all stages): Succeeded/Total
16	csv at NativeMethodAccessorImpl.java:0 csv at NativeMethodAccessorImpl.java:0	2023/10/18 07:49:26	3 s	1/1	4/4
15	count at NativeMethodAccessorImpl.java:0 count at NativeMethodAccessorImpl.java:0	2023/10/18 07:49:26	71 ms	1/1 (3 skipped)	1/1 (54 skipped)
14	count at NativeMethodAccessorImpl.java:0 count at NativeMethodAccessorImpl.java:0	2023/10/18 07:49:24	2 s	1/1 (2 skipped)	25/25 (29 skipped)
13	count at NativeMethodAccessorImpl.java:0 count at NativeMethodAccessorImpl.java:0	2023/10/18 07:49:21	3 s	1/1 (1 skipped)	25/25 (4 skipped)
12	count at NativeMethodAccessorImpl.java:0 count at NativeMethodAccessorImpl.java:0	2023/10/18 07:48:59	22 s	1/1	4/4
11 (061b8b61-3ffa-48b1-bec0-b3eb2a7138a3)	Broadcast exchange (runId 061b8b61-3ffa-48b1-bec0-b3eb2a7138a3) $anonfun$withThreadLocalCaptured$1 at FutureTask.java:266	2023/10/18 07:48:58	0.3 s	1/1	1/1
10	csv at NativeMethodAccessorImpl.java:0 csv at NativeMethodAccessorImpl.java:0	2023/10/18 07:48:54	4 s	1/1 (2 skipped)	25/25 (29 skipped)
9	csv at NativeMethodAccessorImpl.java:0 csv at NativeMethodAccessorImpl.java:0	2023/10/18 07:48:49	4 s	1/1 (1 skipped)	25/25 (4 skipped)
8	csv at NativeMethodAccessorImpl.java:0 csv at NativeMethodAccessorImpl.java:0	2023/10/18 07:48:32	18 s	1/1	4/4
7 (f5ff6db9-dd95-4d46-b3e4-1d17f44f2380)	Broadcast exchange (runId f5ff6db9-dd95-4d46-b3e4-1d17f44f2380) $anonfun$withThreadLocalCaptured$1 at FutureTask.java:266	2023/10/18 07:48:31	0.3 s	1/1	1/1
6	count at NativeMethodAccessorImpl.java:0 count at NativeMethodAccessorImpl.java:0	2023/10/18 07:48:30	0.2 s	1/1 (2 skipped)	1/1 (5 skipped)
5	count at NativeMethodAccessorImpl.java:0 count at NativeMethodAccessorImpl.java:0	2023/10/18 07:48:29	0.3 s	1/1 (1 skipped)	1/1 (4 skipped)
4	count at NativeMethodAccessorImpl.java:0 count at NativeMethodAccessorImpl.java:0	2023/10/18 07:48:06	23 s	1/1	4/4
3	csv at NativeMethodAccessorImpl.java:0 csv at NativeMethodAccessorImpl.java:0	2023/10/18 07:48:04	0.3 s	1/1	1/1
2	csv at NativeMethodAccessorImpl.java:0 csv at NativeMethodAccessorImpl.java:0	2023/10/18 07:48:04	0.2 s	1/1	1/1
1	csv at NativeMethodAccessorImpl.java:0 csv at NativeMethodAccessorImpl.java:0	2023/10/18 07:47:47	16 s	1/1	4/4
0	csv at NativeMethodAccessorImpl.java:0 csv at NativeMethodAccessorImpl.java:0	2023/10/18 07:47:41	5 s	1/1	1/1


ec2-3-109-214-111.ap-south-1.compute.amazonaws.com:18080/history/application_1697614978437_0001/1/stages/

Stages for All Jobs

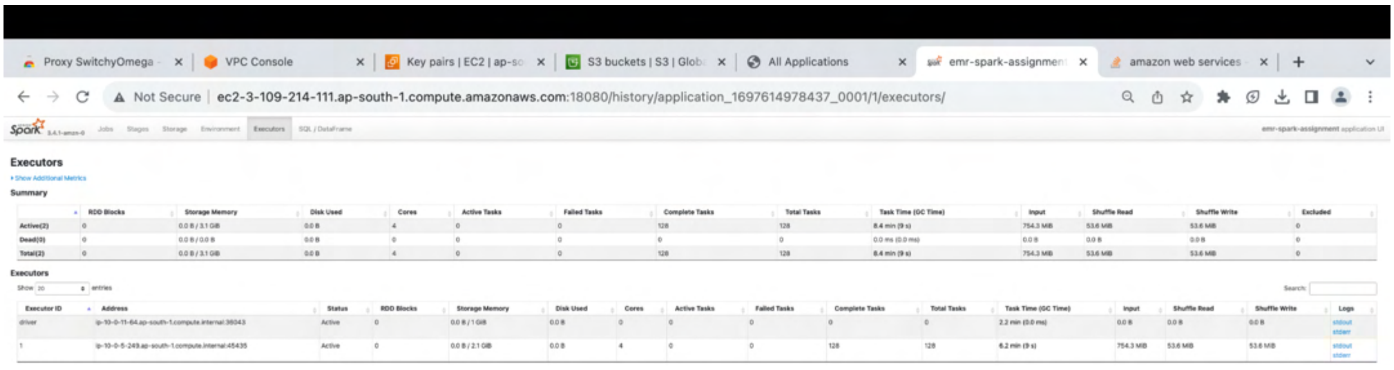
Completed Stages: 17
Skipped Stages: 12

Completed Stages (17)

Stage Id	Description	Submitted	Duration	Tasks: Succeeded/Total	Input	Output	Shuffle Read	Shuffle Write
28	csv at NativeMethodAccessorImpl.java:0	2023/10/18 07:49:26	3 s	4/4		17.0 B		
27	count at NativeMethodAccessorImpl.java:0	2023/10/18 07:49:26	57 ms	1/1			1250.0 B	
23	count at NativeMethodAccessorImpl.java:0	2023/10/18 07:49:24	2 s	25/25			11.3 MiB	1250.0 B
20	count at NativeMethodAccessorImpl.java:0	2023/10/18 07:49:21	3 s	25/25			15.5 MiB	11.3 MiB
18	count at NativeMethodAccessorImpl.java:0	2023/10/18 07:48:59	22 s	4/4	188.5 MiB			15.5 MiB
17	Broadcast exchange (runId 061b8b61-3ffa-48b1-bec0-b3eb2a7138a3) $anonfun$withThreadLocalCaptured$1 at FutureTask.java:266	2023/10/18 07:48:58	0.3 s	1/1	56.9 KiB			
16	csv at NativeMethodAccessorImpl.java:0	2023/10/18 07:48:54	3 s	25/25		12.4 KiB	11.3 MiB	
13	csv at NativeMethodAccessorImpl.java:0	2023/10/18 07:48:49	4 s	25/25			15.5 MiB	11.3 MiB
11	csv at NativeMethodAccessorImpl.java:0	2023/10/18 07:48:32	18 s	4/4	188.5 MiB			15.5 MiB
10	Broadcast exchange (runId f5ff6db9-dd95-4d46-b3e4-1d17f44f2380) $anonfun$withThreadLocalCaptured$1 at FutureTask.java:266	2023/10/18 07:48:31	0.3 s	1/1	56.9 KiB			
9	count at NativeMethodAccessorImpl.java:0	2023/10/18 07:48:30	0.2 s	1/1			50.0 B	
6	count at NativeMethodAccessorImpl.java:0	2023/10/18 07:48:29	0.3 s	1/1			3.0 KiB	50.0 B
4	count at NativeMethodAccessorImpl.java:0	2023/10/18 07:48:06	23 s	4/4	188.5 MiB			3.0 KiB
3	csv at NativeMethodAccessorImpl.java:0	2023/10/18 07:48:04	0.2 s	1/1	56.9 KiB			
2	csv at NativeMethodAccessorImpl.java:0	2023/10/18 07:48:04	0.2 s	1/1	7.5 KiB			
1	csv at NativeMethodAccessorImpl.java:0	2023/10/18 07:47:47	16 s	4/4	188.5 MiB			
0	csv at NativeMethodAccessorImpl.java:0	2023/10/18 07:47:42	5 s	1/1	16.0 KiB			

Skipped Stages (12)

Stage Id	Description	Submitted	Duration	Tasks: Succeeded/Total	Input	Output	Shuffle Read	Shuffle Write
26	count at NativeMethodAccessorImpl.java:0	Unknown	Unknown	0/25				
25	count at NativeMethodAccessorImpl.java:0	Unknown	Unknown	0/25				
24	count at NativeMethodAccessorImpl.java:0	Unknown	Unknown	0/4				
22	count at NativeMethodAccessorImpl.java:0	Unknown	Unknown	0/25				
21	count at NativeMethodAccessorImpl.java:0	Unknown	Unknown	0/4				
19	count at NativeMethodAccessorImpl.java:0	Unknown	Unknown	0/4				
15	csv at NativeMethodAccessorImpl.java:0	Unknown	Unknown	0/25				
14	csv at NativeMethodAccessorImpl.java:0	Unknown	Unknown	0/4				
12	csv at NativeMethodAccessorImpl.java:0	Unknown	Unknown	0/4				
8	count at NativeMethodAccessorImpl.java:0	Unknown	Unknown	0/1				
7	count at NativeMethodAccessorImpl.java:0	Unknown	Unknown	0/4				
5	count at NativeMethodAccessorImpl.java:0	Unknown	Unknown	0/4				


ec2-3-109-214-111.ap-south-1.compute.amazonaws.com:18080/history/application_1697614978437_0001/1/executors/

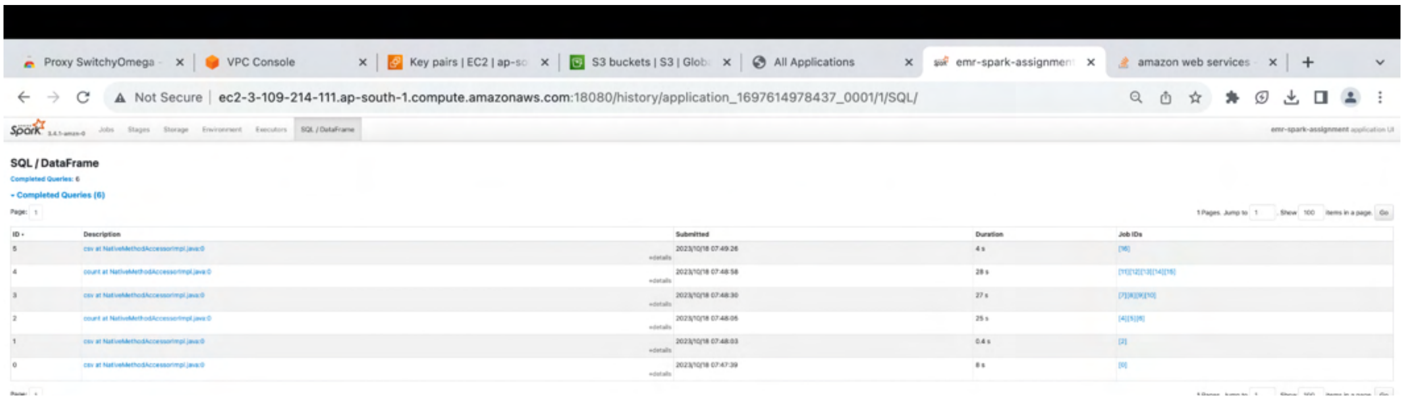
Executors

Summary

	RDD Blocks	Storage Memory	Disk Used	Cores	Active Tasks	Failed Tasks	Complete Tasks	Total Tasks	Task Time (GC Time)	Input	Shuffle Read	Shuffle Write	Excluded
Active(2)	0	0.0 B / 3.1 GiB	0.0 B	4	0	0	128	128	8.4 min (9 s)	754.3 MiB	53.6 MiB	53.6 MiB	0
Dead(0)	0	0.0 B / 0.0 B	0.0 B	0	0	0	0	0	0.0 ms (0.0 ms)	0.0 B	0.0 B	0.0 B	0
Total(2)	0	0.0 B / 3.1 GiB	0.0 B	4	0	0	128	128	8.4 min (9 s)	754.3 MiB	53.6 MiB	53.6 MiB	0

Executors

Executor ID	Address	Status	RDD Blocks	Storage Memory	Disk Used	Cores	Active Tasks	Failed Tasks	Complete Tasks	Total Tasks	Task Time (GC Time)	Input	Shuffle Read	Shuffle Write	Logs
driver	ip-10-0-11-64.ap-south-1.compute.internal:36043	Active	0	0.0 B / 1 GiB	0.0 B	0	0	0	0	0	2.2 min (0.0 ms)	0.0 B	0.0 B	0.0 B	stdout stderr
1	ip-10-0-5-249.ap-south-1.compute.internal:45435	Active	0	0.0 B / 2.1 GiB	0.0 B	4	0	0	128	128	6.2 min (9 s)	754.3 MiB	53.6 MiB	53.6 MiB	stdout stderr



ec2-3-109-214-111.ap-south-1.compute.amazonaws.com:18080/history/application_1697614978437_0001/1/SQL/

emr-spark-assignment application UI

SQL / DataFrame

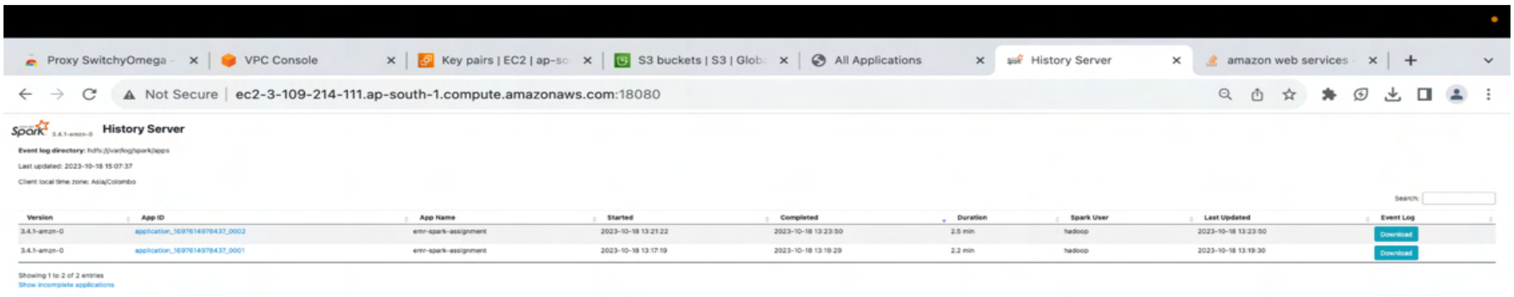
Completed Queries: 6

Completed Queries (6)

ID	Description	Submitted	Duration	Job IDs
5	csv at NativeMethodAccessorImpl.java:0	2023/10/18 07:49:26	4 s	[16]
4	count at NativeMethodAccessorImpl.java:0	2023/10/18 07:48:58	28 s	[11][12][13][14][15]
3	csv at NativeMethodAccessorImpl.java:0	2023/10/18 07:48:30	27 s	[7][8][9][10]
2	count at NativeMethodAccessorImpl.java:0	2023/10/18 07:48:05	25 s	[4][5][6]
1	csv at NativeMethodAccessorImpl.java:0	2023/10/18 07:48:03	0.4 s	[2]
0	csv at NativeMethodAccessorImpl.java:0	2023/10/18 07:47:39	8 s	[0]

3.3.3. Comparison

Execution Time



ec2-3-109-214-111.ap-south-1.compute.amazonaws.com:18080

History Server

Event log directory: hdfs:///var/log/spark/apps

Last updated: 2023-10-18 15:07:37

Client local time zone: Asia/Colombo

Version	App ID	App Name	Started	Completed	Duration	Spark User	Last Updated	Event Log
3.4.1-amzn-0	application_1697614978437_0002	emr-spark-assignment	2023-10-18 13:21:22	2023-10-18 13:23:50	2.5 min	hadoop	2023-10-18 13:23:50	Download
3.4.1-amzn-0	application_1697614978437_0001	emr-spark-assignment	2023-10-18 13:17:19	2023-10-18 13:19:29	2.2 min	hadoop	2023-10-18 13:19:30	Download

Showing 1 to 2 of 2 entries

Show incomplete applications

Spark DataFrame Approach

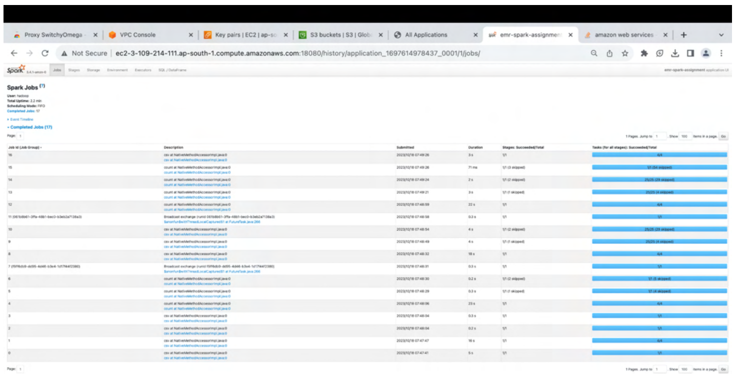
The Spark DataFrame approach completed the task in a shorter execution time (2.2 min). It processed data more swiftly, resulting in quicker results.

Spark SQL Approach

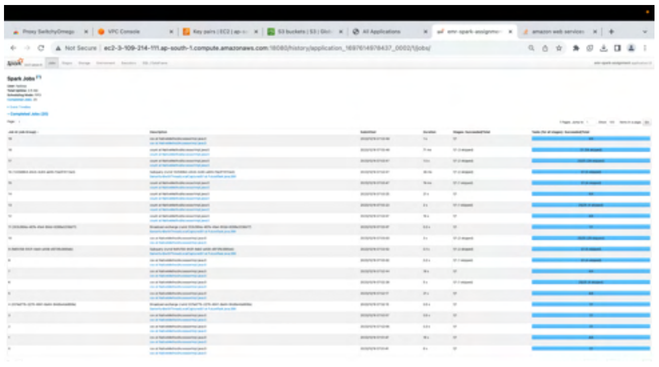
The Spark SQL approach took slightly longer (2.5 min) to complete the task due to the complexity of some job stages with skipped steps. This resulted in a minor delay in data processing.

In this performance comparison, the Spark DataFrame approach stands out for its swifter and more efficient execution. It completed the task in a shorter time frame, ensuring quicker results. In contrast, the Spark SQL approach, while effective, resulting in a slightly longer execution time.

Completed and Skipped Stages



Dataframe



SQL

Completed Stages:

Spark DataFrame: 17

Spark SQL: 20

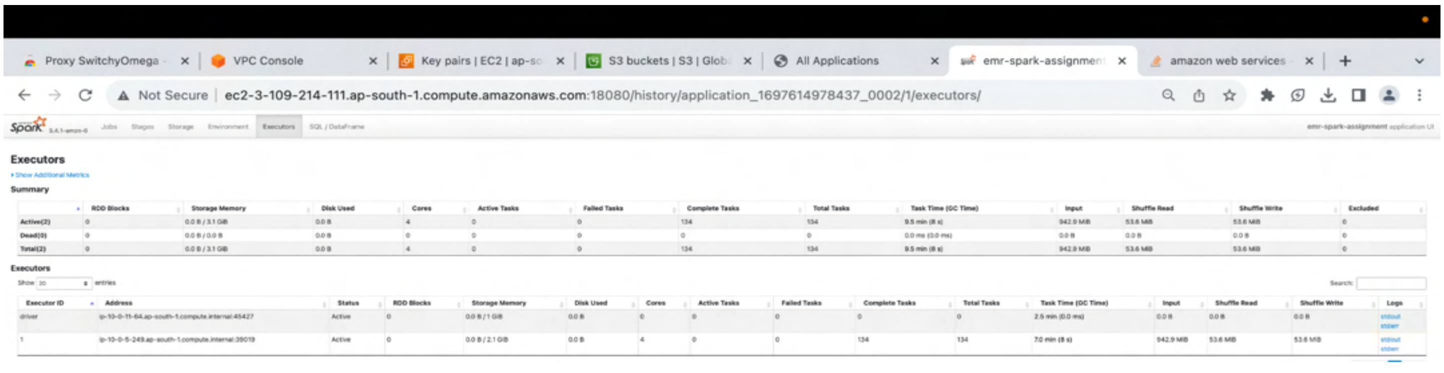
Skipped Stages:

Spark DataFrame: 12

Spark SQL: 15

While Spark SQL completed more stages, it also encountered more skipped stages compared to Spark DataFrame. This suggests that Spark SQL may involve more complex execution plans or face optimization challenges, which can impact the overall efficiency and execution time of jobs. Spark DataFrame, on the other hand, completed fewer stages but encountered fewer skips, indicating a more streamlined and efficient execution. In this case, the slightly lower number of completed stages in Spark DataFrame and the lower count of skipped stages may indicate a more efficient and straightforward execution, which can be beneficial for specific workloads.

Executors



ec2-3-109-214-111.ap-south-1.compute.amazonaws.com:18080/history/application_1697614978437_0002/1/executors/

Executors

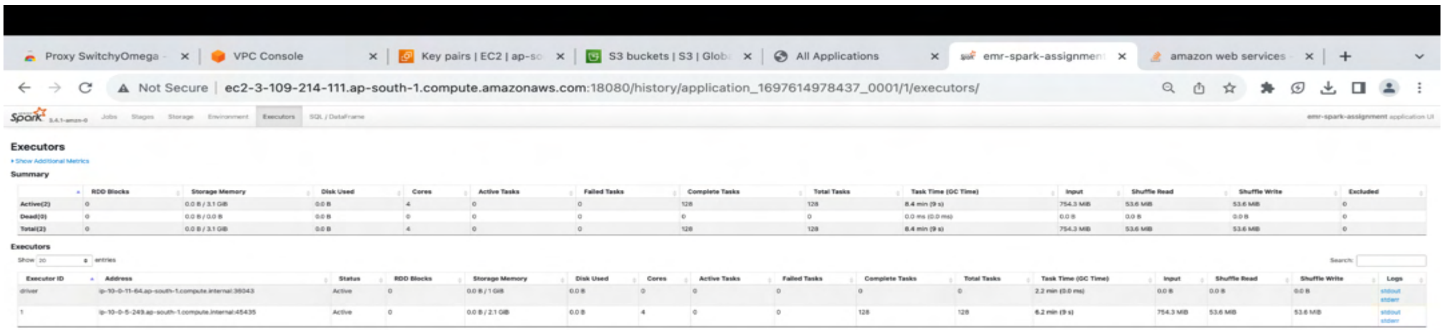
Summary

	RDD Blocks	Storage Memory	Disk Used	Cores	Active Tasks	Failed Tasks	Complete Tasks	Total Tasks	Task Time (GC Time)	Input	Shuffle Read	Shuffle Write	Excluded
Active(2)	0	0.0 B / 3.1 GiB	0.0 B	4	0	0	134	134	9.5 min (8 s)	942.9 MiB	53.6 MiB	53.6 MiB	0
Dead(0)	0	0.0 B / 0.0 B	0.0 B	0	0	0	0	0	0.0 ms (0.0 ms)	0.0 B	0.0 B	0.0 B	0
Total(2)	0	0.0 B / 3.1 GiB	0.0 B	4	0	0	134	134	9.5 min (8 s)	942.9 MiB	53.6 MiB	53.6 MiB	0

Executors

Executor ID	Address	Status	RDD Blocks	Storage Memory	Disk Used	Cores	Active Tasks	Failed Tasks	Complete Tasks	Total Tasks	Task Time (GC Time)	Input	Shuffle Read	Shuffle Write	Logs
driver	ip-10-0-11-64.ap-south-1.compute.internal:45427	Active	0	0.0 B / 1 GiB	0.0 B	0	0	0	0	0	2.5 min (0.0 ms)	0.0 B	0.0 B	0.0 B	stdout stderr
1	ip-10-0-5-249.ap-south-1.compute.internal:39019	Active	0	0.0 B / 2.1 GiB	0.0 B	4	0	0	134	134	7.0 min (8 s)	942.9 MiB	53.6 MiB	53.6 MiB	stdout stderr

Sql



ec2-3-109-214-111.ap-south-1.compute.amazonaws.com:18080/history/application_1697614978437_0001/1/executors/

Executors

Summary

	RDD Blocks	Storage Memory	Disk Used	Cores	Active Tasks	Failed Tasks	Complete Tasks	Total Tasks	Task Time (GC Time)	Input	Shuffle Read	Shuffle Write	Excluded
Active(2)	0	0.0 B / 3.1 GiB	0.0 B	4	0	0	128	128	8.4 min (9 s)	754.3 MiB	53.6 MiB	53.6 MiB	0
Dead(0)	0	0.0 B / 0.0 B	0.0 B	0	0	0	0	0	0.0 ms (0.0 ms)	0.0 B	0.0 B	0.0 B	0
Total(2)	0	0.0 B / 3.1 GiB	0.0 B	4	0	0	128	128	8.4 min (9 s)	754.3 MiB	53.6 MiB	53.6 MiB	0

Executors

Executor ID	Address	Status	RDD Blocks	Storage Memory	Disk Used	Cores	Active Tasks	Failed Tasks	Complete Tasks	Total Tasks	Task Time (GC Time)	Input	Shuffle Read	Shuffle Write	Logs
driver	ip-10-0-11-64.ap-south-1.compute.internal:36043	Active	0	0.0 B / 1 GiB	0.0 B	0	0	0	0	0	2.2 min (0.0 ms)	0.0 B	0.0 B	0.0 B	stdout stderr
1	ip-10-0-5-249.ap-south-1.compute.internal:45435	Active	0	0.0 B / 2.1 GiB	0.0 B	4	0	0	128	128	6.2 min (9 s)	754.3 MiB	53.6 MiB	53.6 MiB	stdout stderr

Dataframe

Spark DataFrame (Executor 1):

Task Shuffle Read Size: 754.3 MiB

Task Shuffle Write Size: 53.6 MiB

Execution Time: 6.2 min

Spark SQL (Executor 1):

Task Shuffle Read Size: 942.9 MiB

Task Shuffle Write Size: 53.6 MiB

Execution Time: 7.0 min

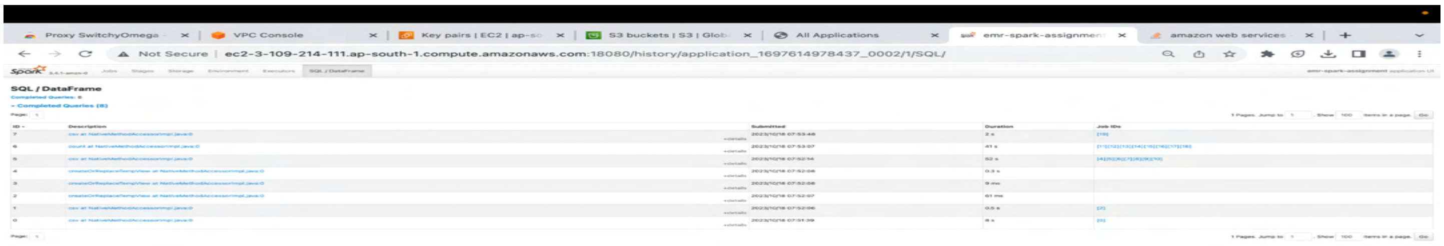
Task Shuffle Read Size: The Spark SQL approach had a larger task shuffle read size, indicating a higher volume of data read during shuffling. This might be due to more complex SQL operations or data transformations that require more data exchange between tasks.

Task Shuffle Write Size: Both approaches had similar task shuffle write sizes (53.6 MiB), indicating that they wrote a comparable amount of data during shuffling operations.

Resource Usage and Efficiency: The larger task shuffle read size in Spark SQL suggests a potential inefficiency in data shuffling, which can impact resource usage and overall job performance.

The analysis of task shuffle read and write sizes reveals that the Spark SQL approach had a larger data read volume during shuffling, which may be due to more complex SQL operations. This highlights the importance of optimizing data shuffling processes to ensure efficient resource usage and better overall job performance.

<u>Completed Queries</u>



ec2-3-109-214-111.ap-south-1.compute.amazonaws.com:18080/history/application_1697614978437_0002/1/SQL/

SQL / DataFrame

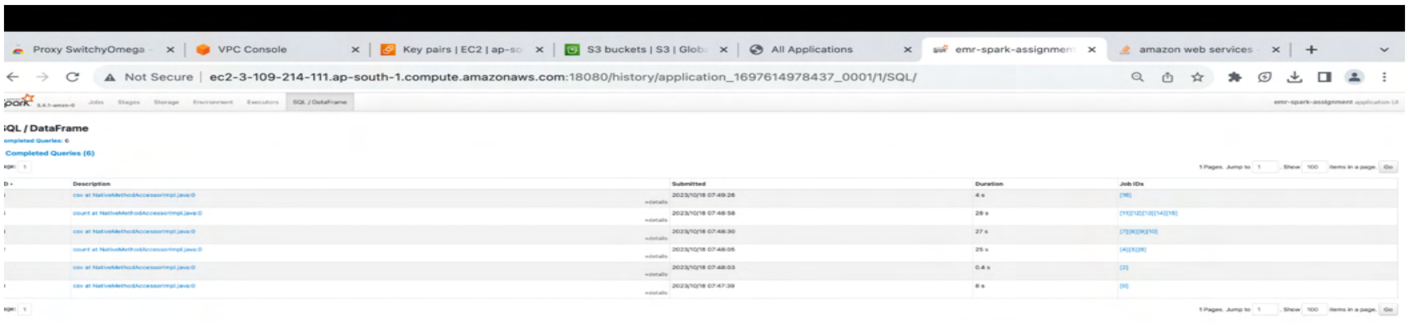
Completed Queries: 8

Completed Queries (8)

Page: 1 — 1 Pages. Jump to 1 . Show 100 items in a page. Go

ID	Description	Submitted	Duration	Job IDs
7	csv at NativeMethodAccessorImpl.java:0 +details	2023/10/18 07:53:46	2 s	[19]
6	count at NativeMethodAccessorImpl.java:0 +details	2023/10/18 07:53:07	41 s	[11][12][13][14][15][16][17][18]
5	csv at NativeMethodAccessorImpl.java:0 +details	2023/10/18 07:52:14	52 s	[4][5][6][7][8][9][10]
4	createOrReplaceTempView at NativeMethodAccessorImpl.java:0 +details	2023/10/18 07:52:08	0.3 s	
3	createOrReplaceTempView at NativeMethodAccessorImpl.java:0 +details	2023/10/18 07:52:08	9 ms	
2	createOrReplaceTempView at NativeMethodAccessorImpl.java:0 +details	2023/10/18 07:52:07	61 ms	
1	csv at NativeMethodAccessorImpl.java:0 +details	2023/10/18 07:52:06	0.5 s	[2]
0	csv at NativeMethodAccessorImpl.java:0 +details	2023/10/18 07:51:39	8 s	[0]

Sql



ec2-3-109-214-111.ap-south-1.compute.amazonaws.com:18080/history/application_1697614978437_0001/1/SQL/

SQL / DataFrame

Completed Queries: 6

Completed Queries (6)

Page: 1 — 1 Pages. Jump to 1 . Show 100 items in a page. Go

ID	Description	Submitted	Duration	Job IDs
5	csv at NativeMethodAccessorImpl.java:0 +details	2023/10/18 07:49:26	4 s	[16]
4	count at NativeMethodAccessorImpl.java:0 +details	2023/10/18 07:48:58	28 s	[11][12][13][14][15]
3	csv at NativeMethodAccessorImpl.java:0 +details	2023/10/18 07:48:30	27 s	[7][8][9][10]
2	count at NativeMethodAccessorImpl.java:0 +details	2023/10/18 07:48:05	25 s	[4][5][6]
1	csv at NativeMethodAccessorImpl.java:0 +details	2023/10/18 07:48:03	0.4 s	[2]
0	csv at NativeMethodAccessorImpl.java:0 +details	2023/10/18 07:47:39	8 s	[0]

Dataframe

Spark DataFrame:	*Spark SQL:*
Completed Queries: 6	Completed Queries: 8
Duration of Longest Query: 28 seconds	Duration of Longest Query: 52 seconds

Spark DataFrame executed fewer queries (6) and had a shorter longest query run time (28 seconds), which indicates efficient processing. In contrast, Spark SQL executed more queries (8) and had a longer longest query run time (52 seconds), suggesting potential resource-intensive or complex queries. Hence, for the given task and based on the available data, the Spark DataFrame method appears to be more efficient in terms of query execution time and possibly resource utilisation.

3.4. Documentation

Step 1: Amazon VPC Setup
To establish a secure network environment for our big data analytics project, I initiated by accessing the Amazon VPC console. Within the console, I created a new Virtual Private Cloud (VPC) thereby ensuring the network configuration met our specifications.

Step 2: Key Pair Generation
To facilitate secure access to our Amazon EMR cluster instances, I generated an SSH key pair. This key pair, created through the EC2 service, is of critical importance for maintaining the integrity and confidentiality of our cluster.

Step 3: Amazon S3 Configuration
Data management is pivotal in big data analytics. Therefore, I set up an Amazon S3 bucket, leveraging its scalability for data storage and retrieval. As an integral part of this step, I uploaded the required data files to our designated S3 bucket, ensuring data availability for our EMR cluster.

Step 4: Amazon EMR Cluster Creation
The heart of our big data analytics operations, the Amazon EMR cluster, was configured through the Amazon EMR console. In this step, I determined cluster details, including its name, EC2 instance types(m4.large), number of instances, S3 access and the designated key pair for SSH access.

Step 5: Python Script Development
The foundation for data processing lay in the development of Python scripts for both the DataFrame and SQL methods. These scripts were crafted to execute data processing tasks with the Spark framework. The finalised Python scripts were uploaded to a directory within the Amazon S3 bucket, making them accessible to the EMR cluster for execution.

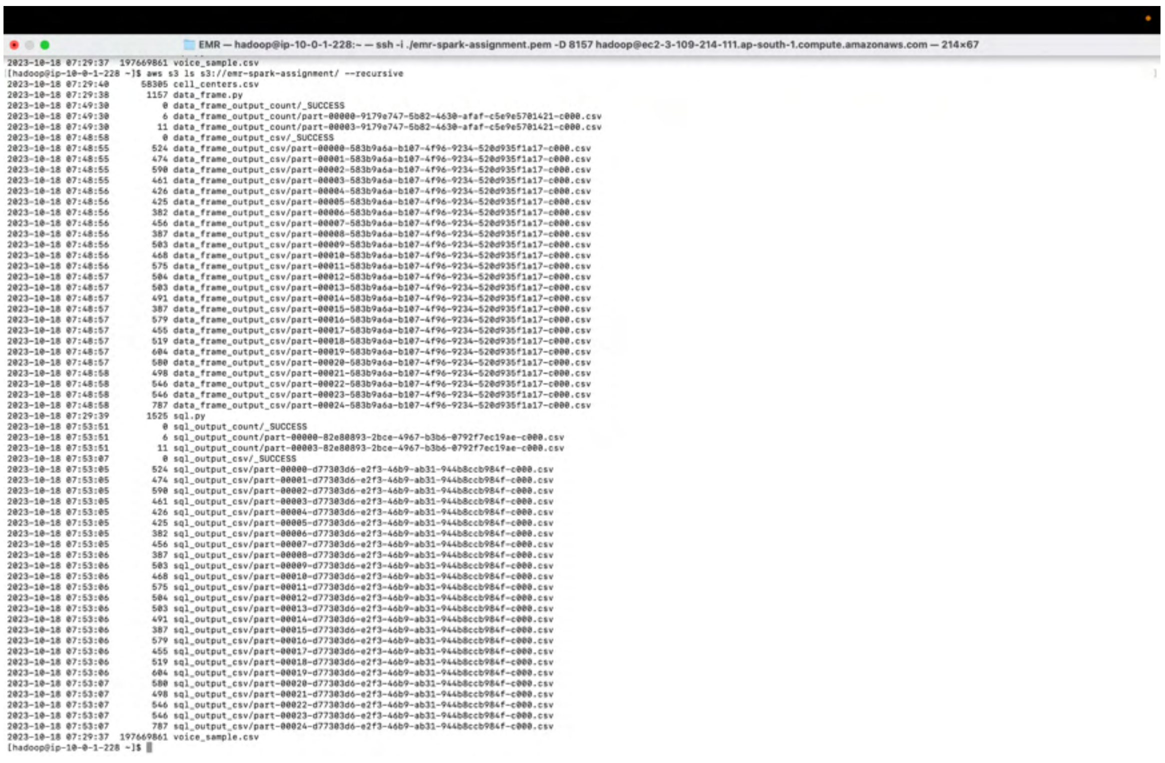
Step 6: EC2 Security Configuration

Security enhancements were integrated by modifying the security group settings for the Amazon EMR cluster instances. In particular, SSH access was explicitly granted to the cluster's EC2 instances, aligning security requirements with operational needs.

Step 7: Execution of Python Scripts

Our Python scripts, designed for data processing, were executed on the Amazon EMR cluster. I ran them as Spark applications by selecting the scripts directly from Amazon S3 through the EMR cluster's execution steps

Output:


EMR — hadoop@ip-10-0-1-228:~ — ssh -i ./emr-spark-assignment.pem -D 8157 hadoop@ec2-3-109-214-111.ap-south-1.compute.amazonaws.com — 214×67

```
2023-10-18 07:29:37  197669861 voice_sample.csv
[hadoop@ip-10-0-1-228 ~]$ aws s3 ls s3://emr-spark-assignment/ --recursive
2023-10-18 07:29:40      58305 cell_centers.csv
2023-10-18 07:29:38       1157 data_frame.py
2023-10-18 07:49:30          0 data_frame_output_count/_SUCCESS
2023-10-18 07:49:30          6 data_frame_output_count/part-00000-9179e747-5b82-4630-afaf-c5e9e5701421-c000.csv
2023-10-18 07:49:30         11 data_frame_output_count/part-00003-9179e747-5b82-4630-afaf-c5e9e5701421-c000.csv
2023-10-18 07:48:58          0 data_frame_output_csv/_SUCCESS
2023-10-18 07:48:55        524 data_frame_output_csv/part-00000-583b9a6a-b107-4f96-9234-520d935f1a17-c000.csv
2023-10-18 07:48:55        474 data_frame_output_csv/part-00001-583b9a6a-b107-4f96-9234-520d935f1a17-c000.csv
2023-10-18 07:48:55        590 data_frame_output_csv/part-00002-583b9a6a-b107-4f96-9234-520d935f1a17-c000.csv
2023-10-18 07:48:55        461 data_frame_output_csv/part-00003-583b9a6a-b107-4f96-9234-520d935f1a17-c000.csv
2023-10-18 07:48:56        426 data_frame_output_csv/part-00004-583b9a6a-b107-4f96-9234-520d935f1a17-c000.csv
2023-10-18 07:48:56        425 data_frame_output_csv/part-00005-583b9a6a-b107-4f96-9234-520d935f1a17-c000.csv
2023-10-18 07:48:56        382 data_frame_output_csv/part-00006-583b9a6a-b107-4f96-9234-520d935f1a17-c000.csv
2023-10-18 07:48:56        456 data_frame_output_csv/part-00007-583b9a6a-b107-4f96-9234-520d935f1a17-c000.csv
2023-10-18 07:48:56        387 data_frame_output_csv/part-00008-583b9a6a-b107-4f96-9234-520d935f1a17-c000.csv
2023-10-18 07:48:56        503 data_frame_output_csv/part-00009-583b9a6a-b107-4f96-9234-520d935f1a17-c000.csv
2023-10-18 07:48:56        468 data_frame_output_csv/part-00010-583b9a6a-b107-4f96-9234-520d935f1a17-c000.csv
2023-10-18 07:48:56        575 data_frame_output_csv/part-00011-583b9a6a-b107-4f96-9234-520d935f1a17-c000.csv
2023-10-18 07:48:57        504 data_frame_output_csv/part-00012-583b9a6a-b107-4f96-9234-520d935f1a17-c000.csv
2023-10-18 07:48:57        503 data_frame_output_csv/part-00013-583b9a6a-b107-4f96-9234-520d935f1a17-c000.csv
2023-10-18 07:48:57        491 data_frame_output_csv/part-00014-583b9a6a-b107-4f96-9234-520d935f1a17-c000.csv
2023-10-18 07:48:57        387 data_frame_output_csv/part-00015-583b9a6a-b107-4f96-9234-520d935f1a17-c000.csv
2023-10-18 07:48:57        579 data_frame_output_csv/part-00016-583b9a6a-b107-4f96-9234-520d935f1a17-c000.csv
2023-10-18 07:48:57        455 data_frame_output_csv/part-00017-583b9a6a-b107-4f96-9234-520d935f1a17-c000.csv
2023-10-18 07:48:57        519 data_frame_output_csv/part-00018-583b9a6a-b107-4f96-9234-520d935f1a17-c000.csv
2023-10-18 07:48:57        604 data_frame_output_csv/part-00019-583b9a6a-b107-4f96-9234-520d935f1a17-c000.csv
2023-10-18 07:48:57        580 data_frame_output_csv/part-00020-583b9a6a-b107-4f96-9234-520d935f1a17-c000.csv
2023-10-18 07:48:58        498 data_frame_output_csv/part-00021-583b9a6a-b107-4f96-9234-520d935f1a17-c000.csv
2023-10-18 07:48:58        546 data_frame_output_csv/part-00022-583b9a6a-b107-4f96-9234-520d935f1a17-c000.csv
2023-10-18 07:48:58        546 data_frame_output_csv/part-00023-583b9a6a-b107-4f96-9234-520d935f1a17-c000.csv
2023-10-18 07:48:58        787 data_frame_output_csv/part-00024-583b9a6a-b107-4f96-9234-520d935f1a17-c000.csv
2023-10-18 07:29:39       1525 sql.py
2023-10-18 07:53:51          0 sql_output_count/_SUCCESS
2023-10-18 07:53:51          6 sql_output_count/part-00000-82e80893-2bce-4967-b3b6-0792f7ec19ae-c000.csv
2023-10-18 07:53:51         11 sql_output_count/part-00003-82e80893-2bce-4967-b3b6-0792f7ec19ae-c000.csv
2023-10-18 07:53:07          0 sql_output_csv/_SUCCESS
2023-10-18 07:53:05        524 sql_output_csv/part-00000-d77303d6-e2f3-46b9-ab31-944b8ccb984f-c000.csv
2023-10-18 07:53:05        474 sql_output_csv/part-00001-d77303d6-e2f3-46b9-ab31-944b8ccb984f-c000.csv
2023-10-18 07:53:05        590 sql_output_csv/part-00002-d77303d6-e2f3-46b9-ab31-944b8ccb984f-c000.csv
2023-10-18 07:53:05        461 sql_output_csv/part-00003-d77303d6-e2f3-46b9-ab31-944b8ccb984f-c000.csv
2023-10-18 07:53:05        426 sql_output_csv/part-00004-d77303d6-e2f3-46b9-ab31-944b8ccb984f-c000.csv
2023-10-18 07:53:05        425 sql_output_csv/part-00005-d77303d6-e2f3-46b9-ab31-944b8ccb984f-c000.csv
2023-10-18 07:53:05        382 sql_output_csv/part-00006-d77303d6-e2f3-46b9-ab31-944b8ccb984f-c000.csv
2023-10-18 07:53:05        456 sql_output_csv/part-00007-d77303d6-e2f3-46b9-ab31-944b8ccb984f-c000.csv
2023-10-18 07:53:06        387 sql_output_csv/part-00008-d77303d6-e2f3-46b9-ab31-944b8ccb984f-c000.csv
2023-10-18 07:53:06        503 sql_output_csv/part-00009-d77303d6-e2f3-46b9-ab31-944b8ccb984f-c000.csv
2023-10-18 07:53:06        468 sql_output_csv/part-00010-d77303d6-e2f3-46b9-ab31-944b8ccb984f-c000.csv
2023-10-18 07:53:06        575 sql_output_csv/part-00011-d77303d6-e2f3-46b9-ab31-944b8ccb984f-c000.csv
2023-10-18 07:53:06        504 sql_output_csv/part-00012-d77303d6-e2f3-46b9-ab31-944b8ccb984f-c000.csv
2023-10-18 07:53:06        503 sql_output_csv/part-00013-d77303d6-e2f3-46b9-ab31-944b8ccb984f-c000.csv
2023-10-18 07:53:06        491 sql_output_csv/part-00014-d77303d6-e2f3-46b9-ab31-944b8ccb984f-c000.csv
2023-10-18 07:53:06        387 sql_output_csv/part-00015-d77303d6-e2f3-46b9-ab31-944b8ccb984f-c000.csv
2023-10-18 07:53:06        579 sql_output_csv/part-00016-d77303d6-e2f3-46b9-ab31-944b8ccb984f-c000.csv
2023-10-18 07:53:06        455 sql_output_csv/part-00017-d77303d6-e2f3-46b9-ab31-944b8ccb984f-c000.csv
2023-10-18 07:53:06        519 sql_output_csv/part-00018-d77303d6-e2f3-46b9-ab31-944b8ccb984f-c000.csv
2023-10-18 07:53:06        604 sql_output_csv/part-00019-d77303d6-e2f3-46b9-ab31-944b8ccb984f-c000.csv
2023-10-18 07:53:07        580 sql_output_csv/part-00020-d77303d6-e2f3-46b9-ab31-944b8ccb984f-c000.csv
2023-10-18 07:53:07        498 sql_output_csv/part-00021-d77303d6-e2f3-46b9-ab31-944b8ccb984f-c000.csv
2023-10-18 07:53:07        546 sql_output_csv/part-00022-d77303d6-e2f3-46b9-ab31-944b8ccb984f-c000.csv
2023-10-18 07:53:07        546 sql_output_csv/part-00023-d77303d6-e2f3-46b9-ab31-944b8ccb984f-c000.csv
2023-10-18 07:53:07        787 sql_output_csv/part-00024-d77303d6-e2f3-46b9-ab31-944b8ccb984f-c000.csv
2023-10-18 07:29:37  197669861 voice_sample.csv
[hadoop@ip-10-0-1-228 ~]$ 
```