Step 3- Date wise total booking transformation for ingested batch data

Prerequisite steps before performing date aggregation for the batch data from RDS

Setup in Amazon

I AM setup

- Login in to the AWS acoount
- Navigate to I AM dashboard
- Create a user and download the key pair

EMR Setup

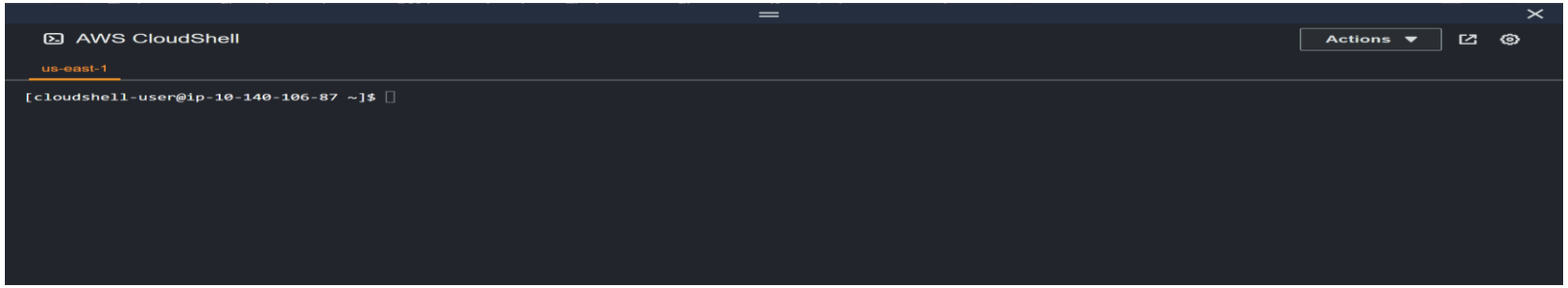
- Navigate to EMR page
- Create a new cluster
- Application bundle for EMR should include the following (Hue,Spark,Hadoop,Scoop,Hive)
- Cluster config, keep the default options or increase it based on requirement
- Cluster scaling and provisioning keep it as default or change it based on requirement
- Networking keep it as default or change it based on requirement
- Cluster termination set the idle time to automatically terminate the cluster
- Security configuration select the key pair created
- Select the IAM service role if created and make sure this role has permission to perform actions in the EMR so attach the necessary policy for this role such as (EMRFullAccess or Administrator Access) can modify latter based on requirement
- Select the EC2 instance profile for EMR (Will be equal to the user created in I AM setup)
- Click on Create cluster

Ingest Batch data with scoop from AWS RDS

- After setting up EMR cluster SSH into the cluster
- Download the JDBC driver and move it the scoop lib folder
- Switch as Hadoop admin and perform the scoop import command to import batch data from AWS RDS
- Validate the batch data is imported in the hdfs folder

Date Aggregation on Batch Data

- In the aws cloud shell upload the datewise_bookings_aggregates_spark.py python file by clicking on the actions dropdown



- The user key pair should be uploaded to the cloud shell as well

*Important replace the below mentioned parameters in all commands as per your configurations

1) i dev.pem: This option specifies the private key file (eg dev.pem)

2) ec2-user@ec2-54-162-87-251.compute-1.amazonaws.com: This specifies the user and address of the remote machine

3)From Directory (if it’s a copy)

4) To Directory (if it’s a copy)

- Copy the uploaded datewise_bookings_aggregates_spark.py to the EMR local with the below command secure copy protocol (scp)

```
scp -i dev.pem datewise_bookings_aggregates_spark.py ec2-user@ec2-54-162-87-251.compute-1.amazonaws.com:/home/ec2-user
```

- Login to the EMR with below mentioned command

```
ssh -i dev.pem ec2-user@ec2-54-162-87-251.compute-1.amazonaws.com
```

Date Aggregation on Batch Data

- Move the datewise_bookings_aggregates_spark.py to a hdfs directory with the below command

```
hdfs dfs -put datewise_bookings_aggregates_spark.py /user
```

- Run as Hadoop admin with below mentioned command

```
sudo su - hdfs
```

- Get the datewise_bookings_aggregates_spark.py from hdfs directory to Hadoop admin local to start the spark job with below command

```
hdfs dfs -get /user/datewise_bookings_aggregates_spark.py /var/lib/hadoop-hdfs
```

- Submit the spark job running as Hadoop admin so there is no write permission issues in hdfs

```
spark-submit --packages org.apache.spark:spark-sql-kafka-0-10_2.12:3.2.1
datewise_bookings_aggregates_spark.py
```

- Validate the transformed data in hdfs directory by navigating to the directory
- To navigate to a directory

```
hdfs dfs  -ls /directory/path
```

- To view the data use the below command

```
hdfs dfs -cat  /path/to/directory
```

Validate the transformed data in hdfs directory by navigating to the directory with -ls

us-east-1.console.aws.amazon.com/cloudshell/home?region=us-east-1#5bf7746c-8b8c-4830-b897-d49b20c8aee6

Bookmarks | Gmail | YouTube | Maps | keyboard shortcuts... | New Tab | Program Calendar... | 5. Data Structures... | GitHub - awesome... | All Bookmarks

aws | Services | Search [Alt+S] | N. Virginia | upgradfelixphilipk @ 6909-8915-0119

AWS CloudShell | Actions

us-east-1

```
23/11/29 06:16:48 INFO ServerInfo: Adding filter to /: org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter
23/11/29 06:16:48 INFO ServerInfo: Adding filter to /api: org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter
23/11/29 06:16:48 INFO ServerInfo: Adding filter to /jobs/job/kill: org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter
23/11/29 06:16:48 INFO ServerInfo: Adding filter to /stages/stage/kill: org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter
23/11/29 06:16:48 INFO ServerInfo: Adding filter to /metrics/json: org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter
23/11/29 06:16:48 INFO YarnClientSchedulerBackend: SchedulerBackend is ready for scheduling beginning after reached minRegisteredResourcesRatio: 0.0
-bash-4.2$ exit
logout
[ec2-user@ip-172-31-38-241 ~]$ hdfs dfs -ls /user/Data
ls: `/user/Data': No such file or directory
[ec2-user@ip-172-31-38-241 ~]$ hdfs dfs -ls /user/ec2-user
Found 1 items
drwxr-xr-x   - hdfs hdfsadmingroup          0 2023-11-29 06:17 /user/ec2-user/Data
[ec2-user@ip-172-31-38-241 ~]$ hdfs dfs -ls /user/ec2-user/Data
Found 2 items
drwxr-xr-x   - hdfs hdfsadmingroup          0 2023-11-29 04:54 /user/ec2-user/Data/Booking_Batch_Data
drwxr-xr-x   - hdfs hdfsadmingroup          0 2023-11-29 06:17 /user/ec2-user/Data/Datewise_Total_Bookings
[ec2-user@ip-172-31-38-241 ~]$ hdfs dfs -ls /user/ec2-user/Data/Datewise_Total_Bookings
Found 2 items
-rw-r--r--   1 hdfs hdfsadmingroup          0 2023-11-29 06:17 /user/ec2-user/Data/Datewise_Total_Bookings/_SUCCESS
-rw-r--r--   1 hdfs hdfsadmingroup       3769 2023-11-29 06:17 /user/ec2-user/Data/Datewise_Total_Bookings/part-00000-4a5a55d6-f0bd-42d5-b135-9a3a33b39eba-c000.
csv
[ec2-user@ip-172-31-38-241 ~]$
```