CS443
Cloud Computing
Homework 3 Report

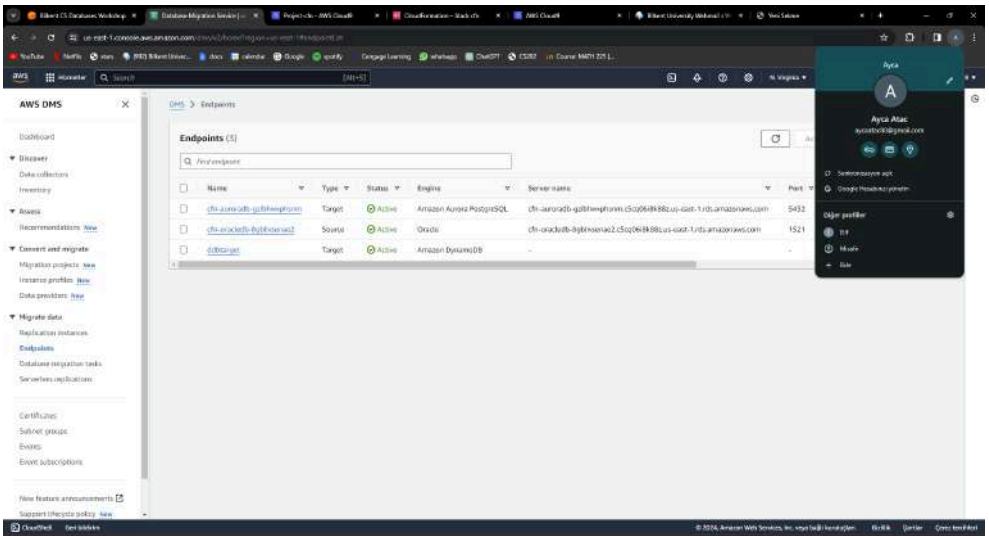
Onurcan Ataç
22002194

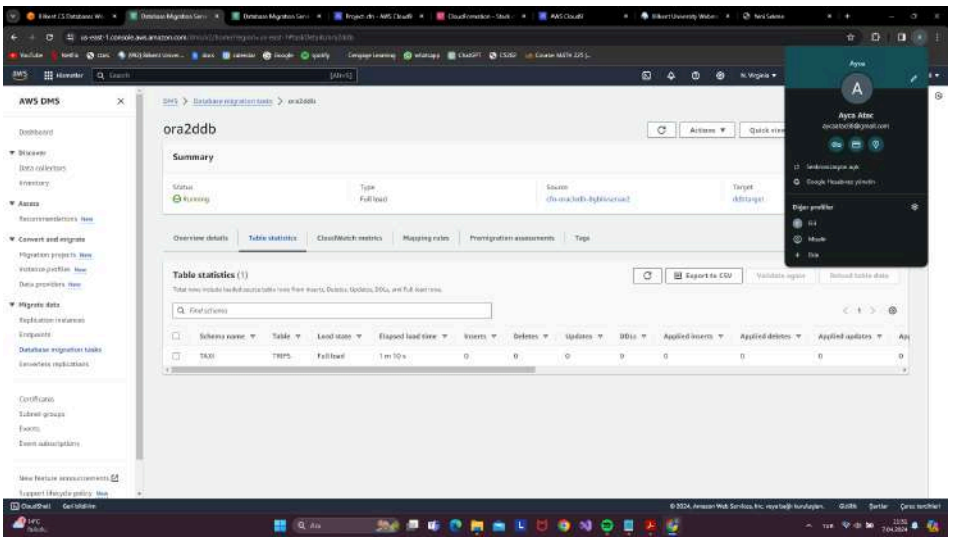
Date: 07.04.2024

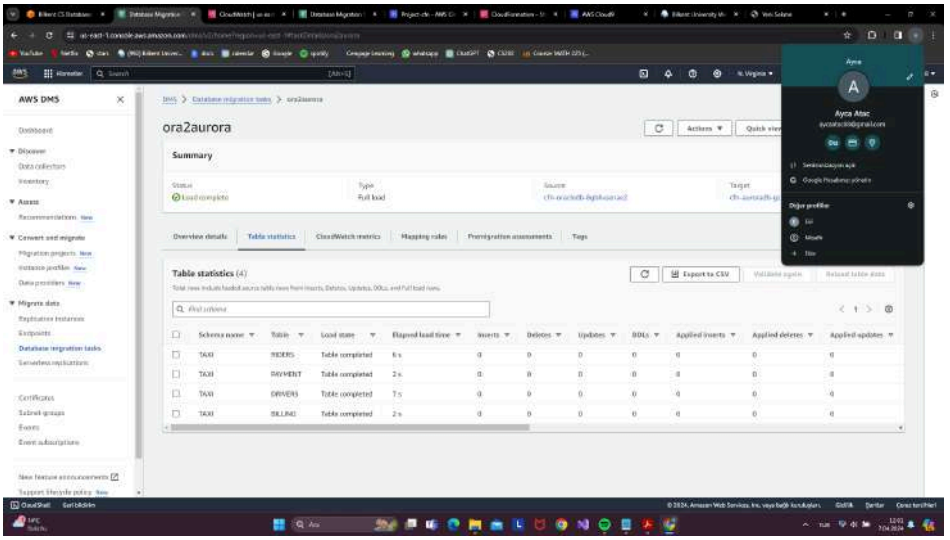
I have explained the work I have done for each part of the lab separately in this report. I also added related screenshots to the parts. In the account with my Bilkent mail, there was a bug about the Oracle source endpoint, I was getting a *"Test Endpoint failed: Application-Status: 1020912, Application-Message: Invalid username or password."* error even though I was entering the same password that is successfully accepted by the Oracle SQLPLUS login. Therefore I have switched computers and accounts to complete the homework. I used the email and computer of my sister in order to get rid of the bug. When I applied the same steps on the new account, the bug did not emerge. **Therefore, the name on my screenshots is *Ayça Ataç*, the name of my sister since the account and computer were under her name. I still completed all parts of the homework myself.** In the AWS console, my username was automatically “WSParticipantRole/Participant”, therefore I used her Google account popup to prove I was the one doing the work in the screenshots.

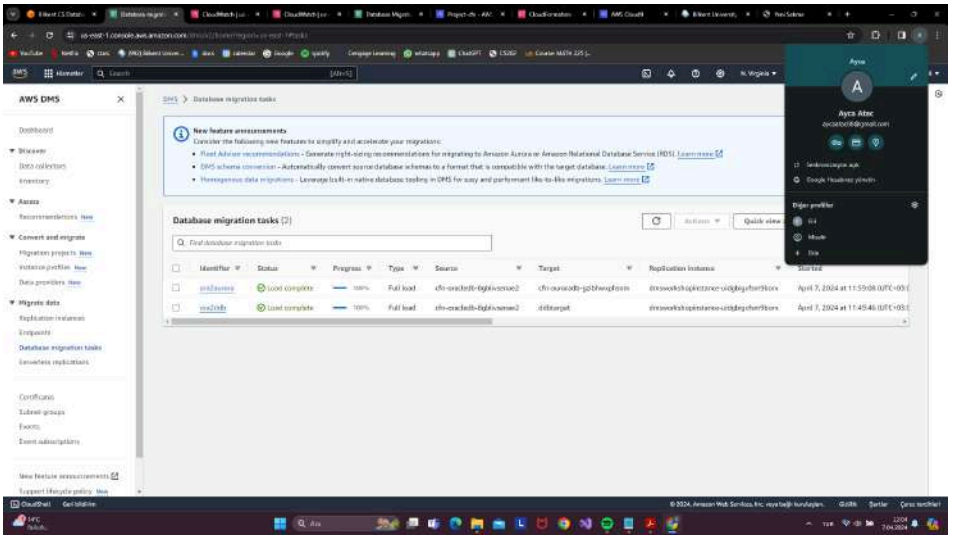
1. Lab1 - Data Migration using AWS DMS

In the first part of the assignment, I performed a migration using AWS DMS from RDS Oracle to Amazon DynamoDB and Amazon Aurora PostgreSQL databases. I used AWS Cloud 9 IDE to connect to Oracle SQLPLUS and PostgreSQL, and other coding purposes throughout the labs. I then created a taxi schema and verified the objects in the schema in Oracle. I also created the schema in PostgreSQL. I then created the source and target endpoints and tested their connections. After getting the successful results from the tests, I created replication tasks and monitored the migrations. I validated the results to finish up this part.



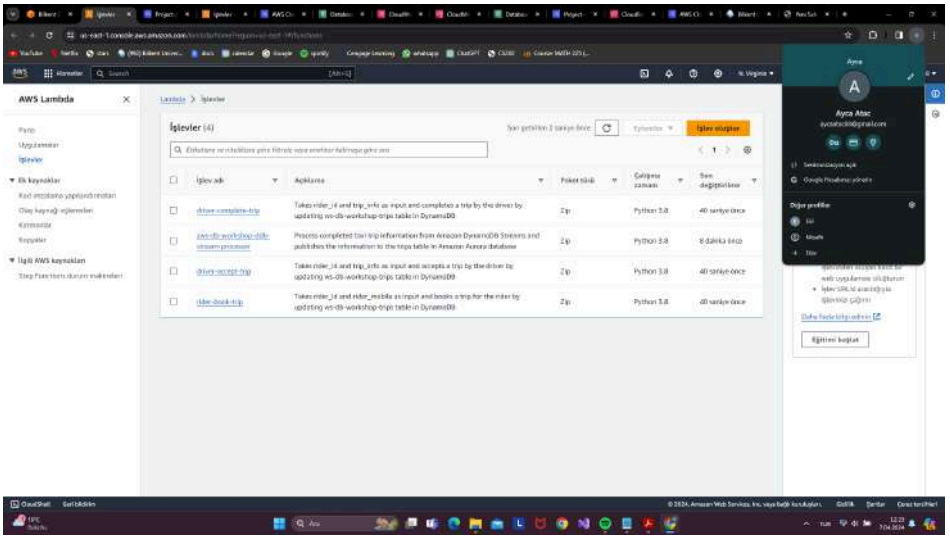

AWS DMS
ora2ddb
Summary
Table statistics (1)

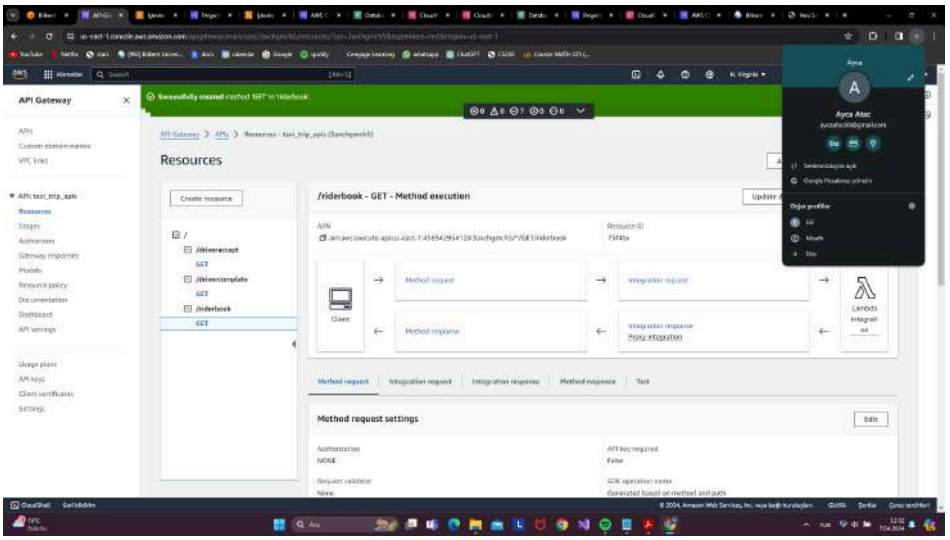

AWS DMS
ora2aurora
Summary
Table statistics (4)

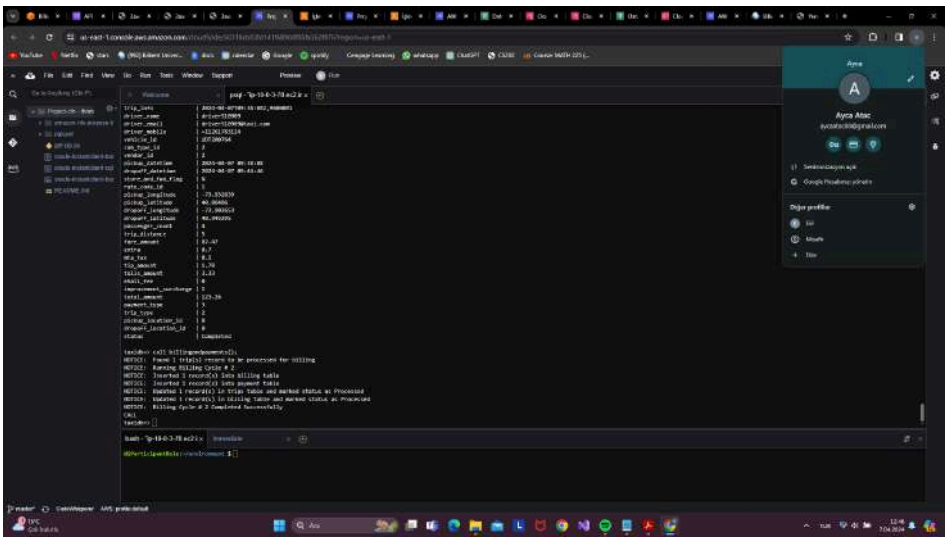

AWS DMS
Database migration tasks (2)

2. Lab2 - Data processing using Amazon DynamoDB and Amazon Aurora

I continued with the second part in which I created and used an API to add functionality to the taxi data and pass data from Amazon DynamoDB to the Aurora database using Lambda functions. I started by enabling streams for the DynamoDB tables. I continued with creating and deploying AWS Lambda functions. I then created a REST API by using Amazon API Gateway, by connecting the resources with corresponding Lambda functions. I then called the API functions via specific HTTP requests and therefore simulated the taxi ride workflow.

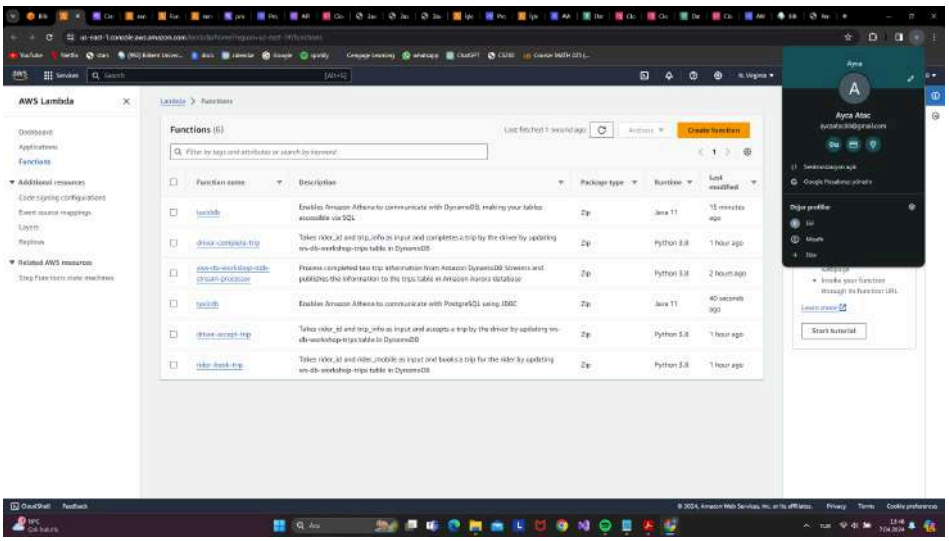


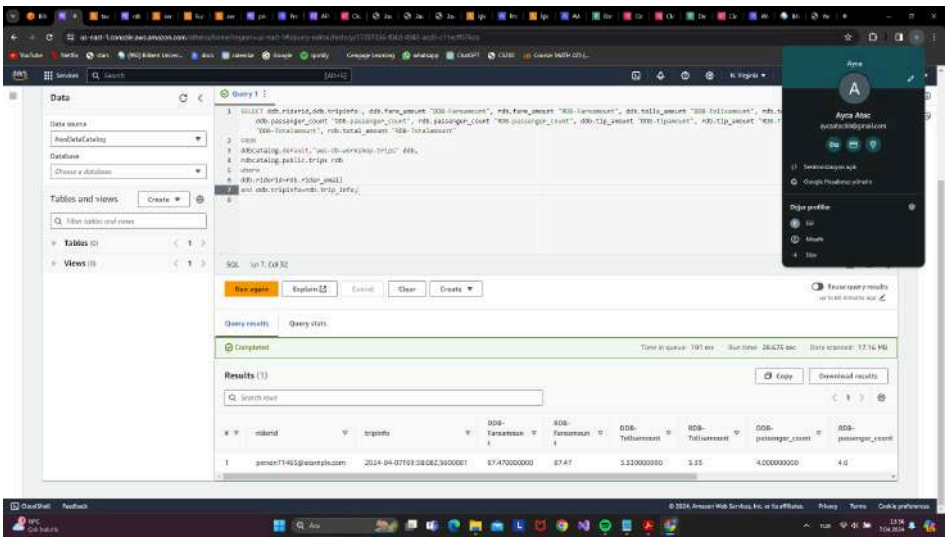




3. Lab3 - Query multiple data sources using Amazon Athena federated query

In the third part, I made use of the federated query service of Amazon Athena in order to execute queries for multiple different types of databases, in this case, DynamoDB and Aurora, also using S3 buckets. I created connectors that use Lambda functions which Athena will use to run queries. I then tested my implementations by entering queries in the Athena query editor and validated my results with the last query provided to finish the assignment.




Data
Tables and views
Create
Run again
Explain
Clear
Create
Query results
Query stats
Results (1)
Copy
Download results
rideId
tripInfo
DDB-fareamount
RDB-fareamount
DDB-TollAmount
RDB-TollAmount
DDB-passenger_count
RDB-passenger_count
1
passer71491@example.com
2024-04-07T01:38:08Z,9000001
87.47000000
87.47
5.320000000
5.35
4.000000000
4.0