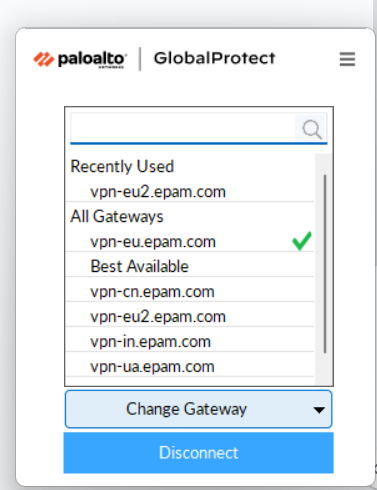
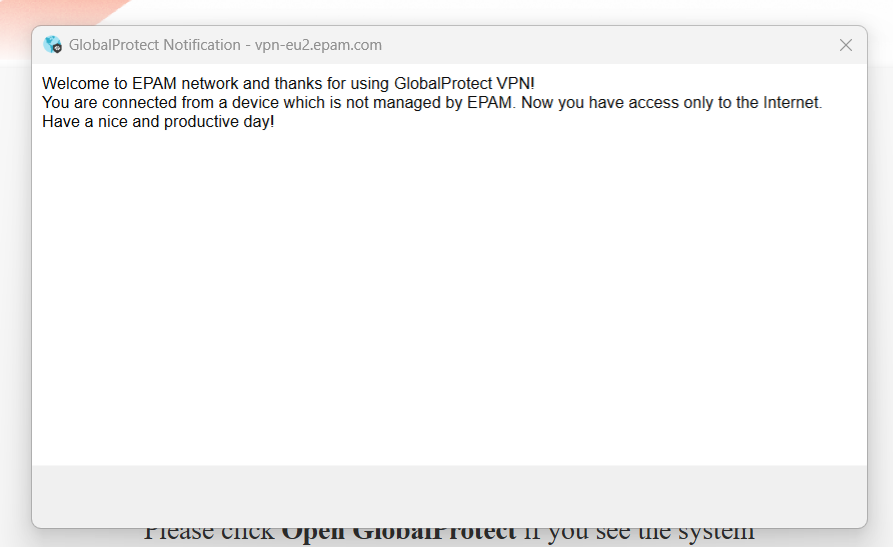
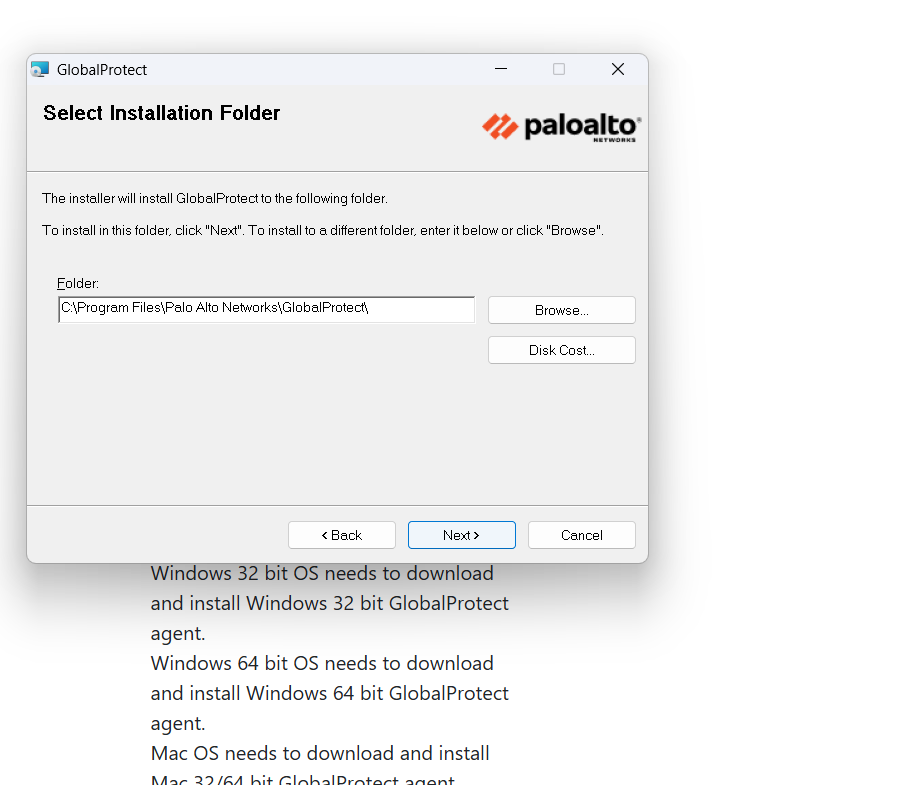


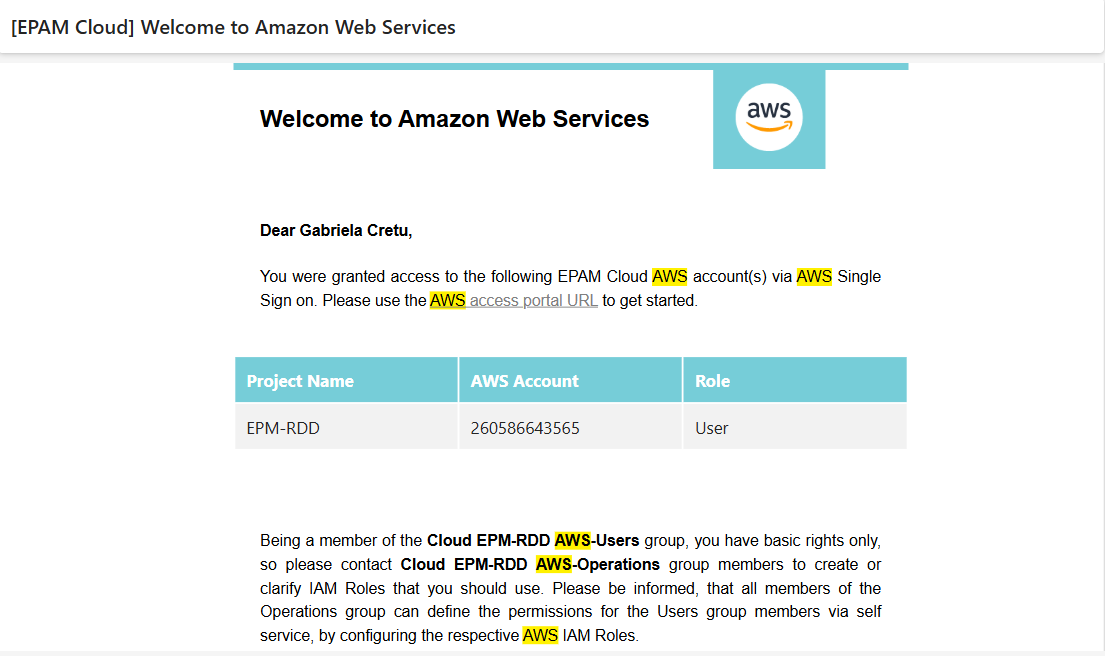
| Business Template  **IAM and S3 services** |
| --- |
|  |

\*First, I installed the VPN and configured it to connect to vpn-eu.epam.com.

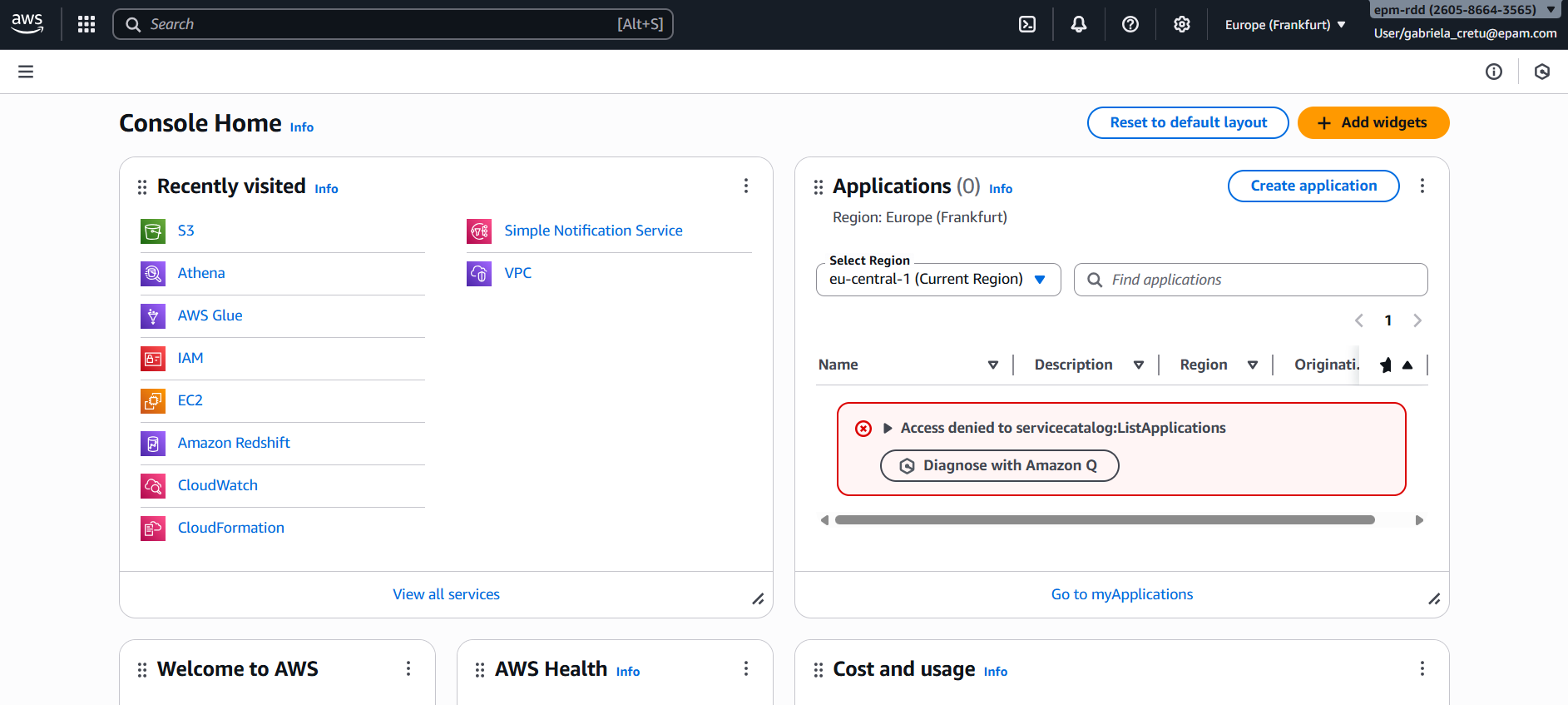
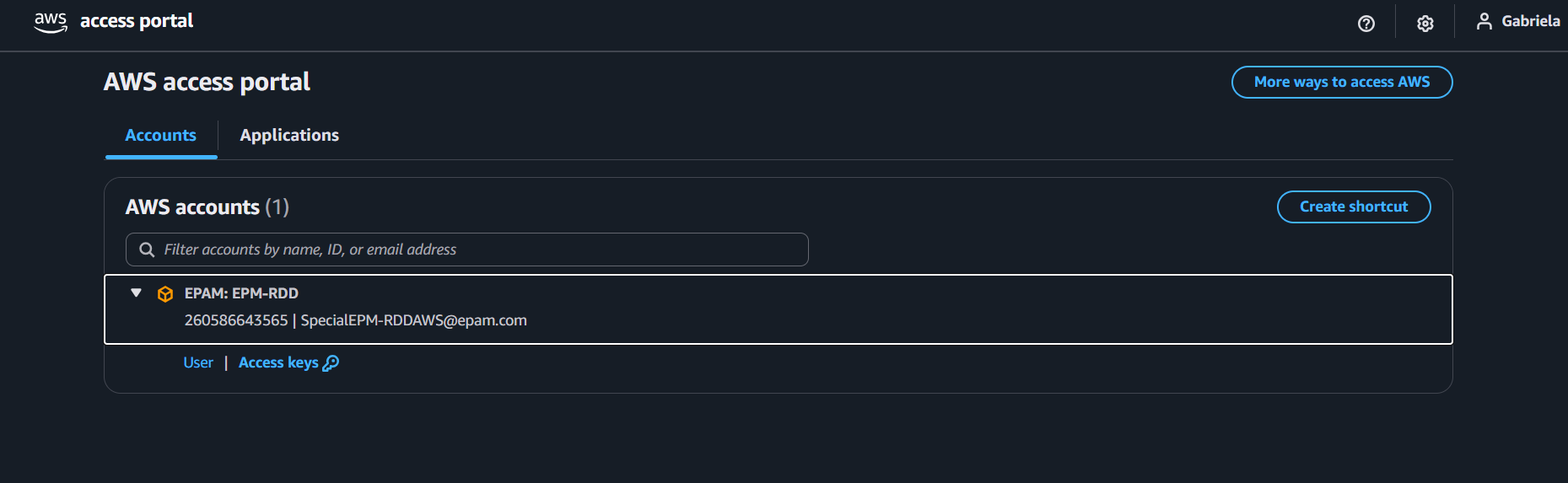


# 1. Establish connection to AWS account:

a. Login **AWS Console** with SSO using link provided in the email from Auto EPM-CSUP Notification



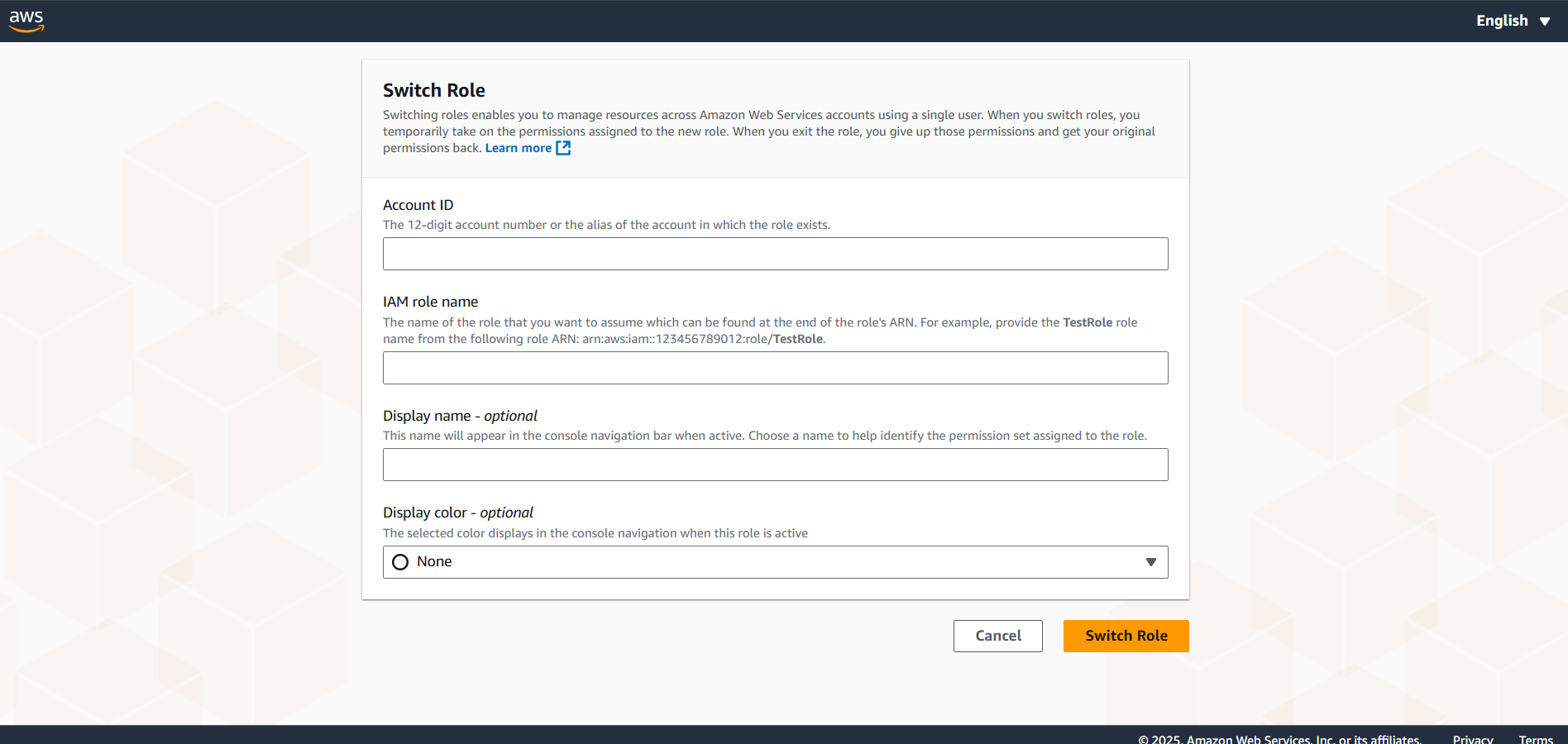
b. Choose “User” to login AWS Console



c. After you logged in the console in the right upper corner click drop down button

\

d. Click “Switch role” button

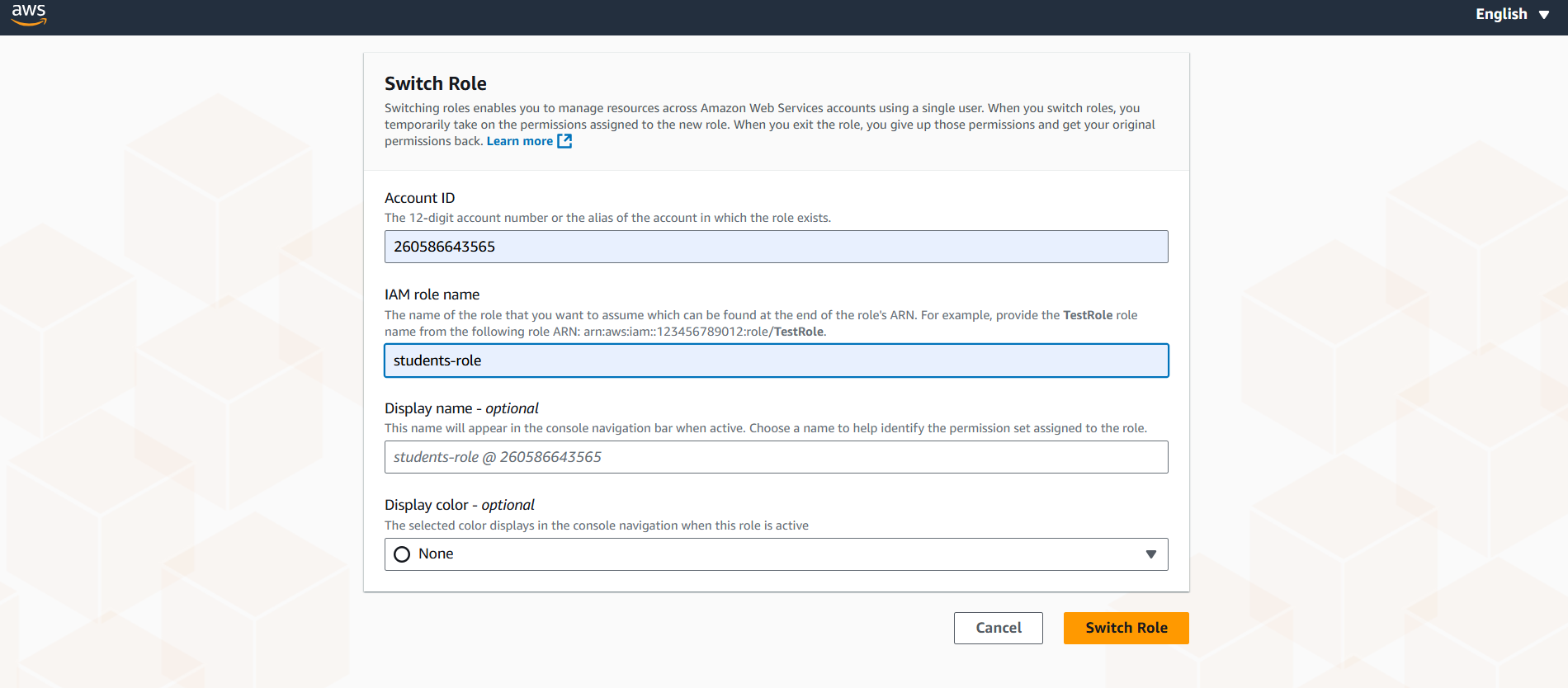


e. Account name – 260586643565

Role – students-role

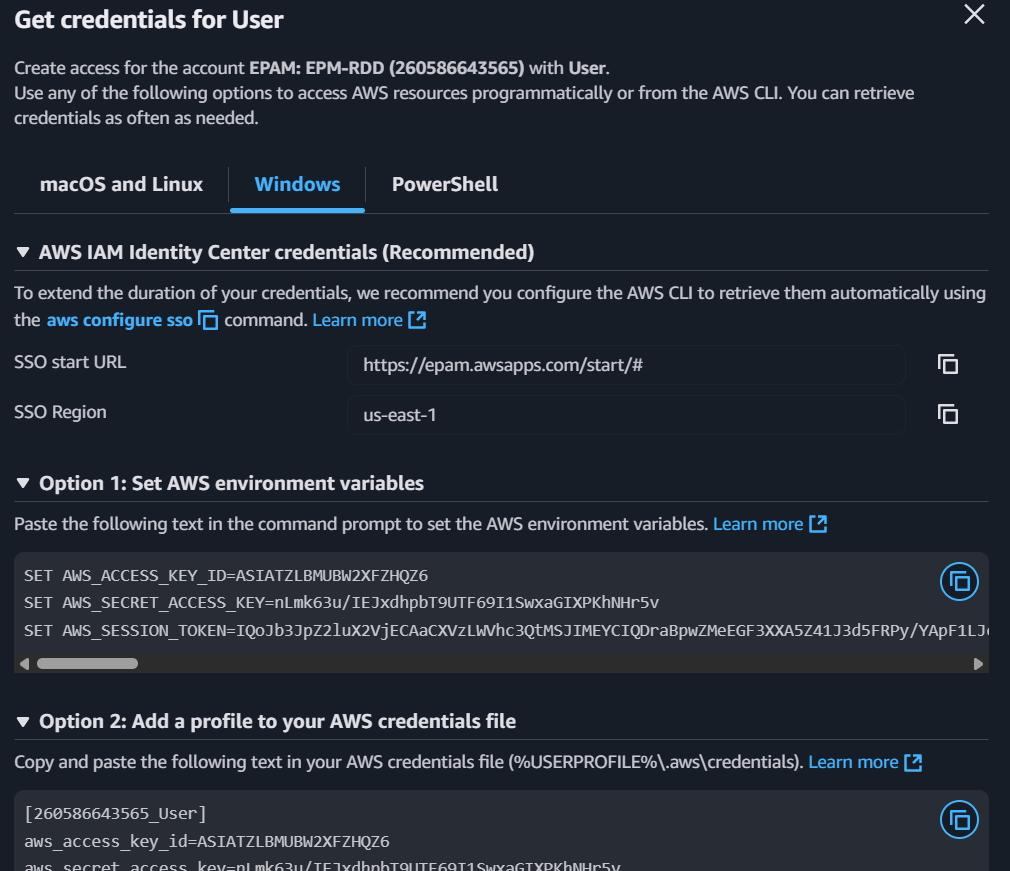
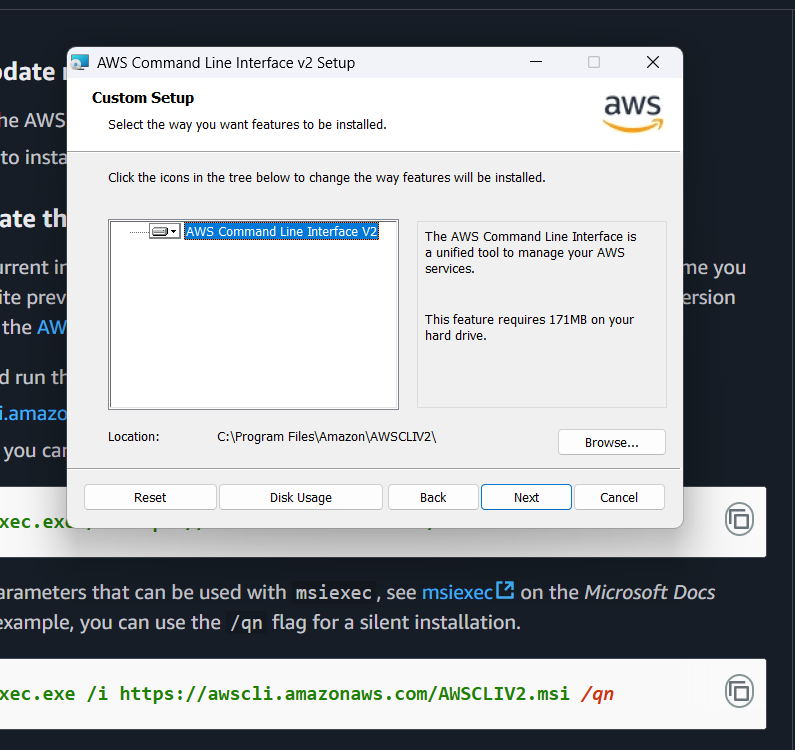
Display Name (optional) – Fill in the name you like.

Color (optional) – Chose the one you like.

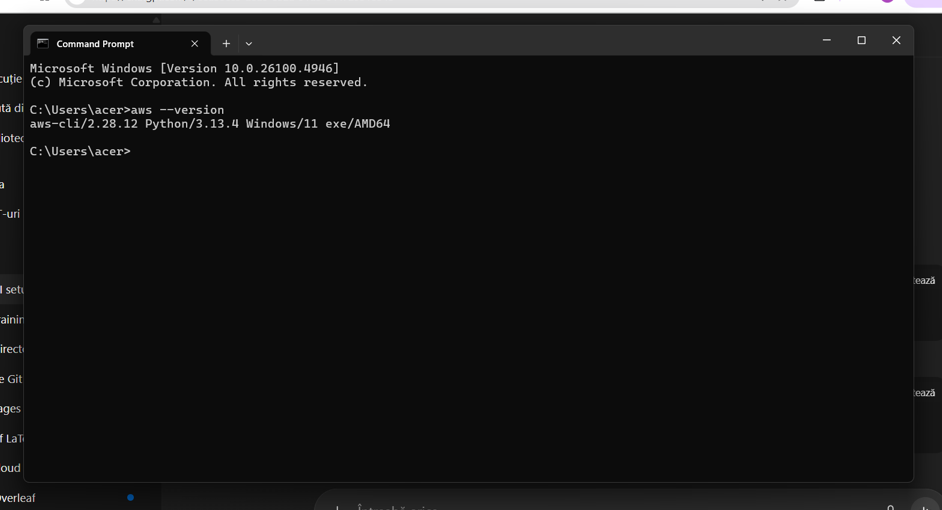


f. To use **AWS CLI** please firstly install the client (NB: PLEASE INSTALL VERSION 2 ONLY) https://docs.aws.amazon.com/cli/latest/userguide/getting-started-install.html and after that open SSO link from point b and choose “Access keys”

Then, I installed and configured the AWS Command Line Interface (CLI) on my laptop.



Next, I checked the installed AWS CLI version.



g. Run *aws configure sso* command<https://docs.aws.amazon.com/cli/latest/userguide/sso-configure-profile-token.html>:

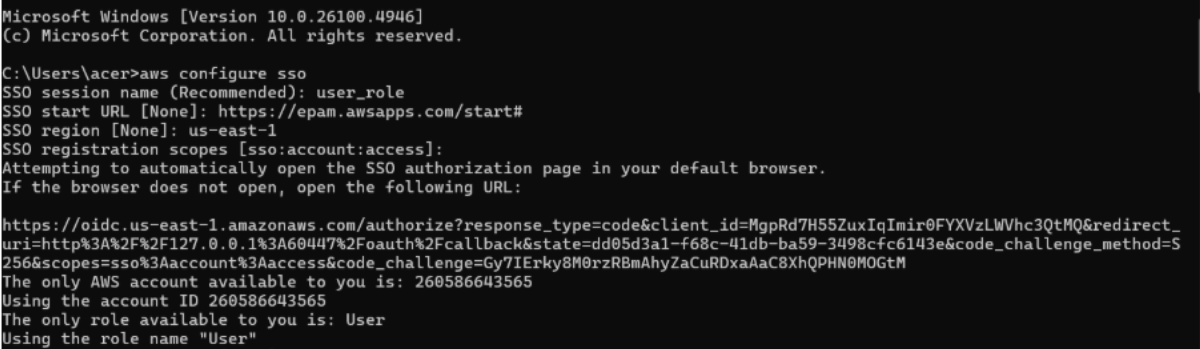
*SSO session name = user-role*

*SSO start URL = https://epam.awsapps.com/start#*

*SSO region = us-east-1*

*SSO registration scopes – keep blank and press enter*

I set up the AWS account using the command prompt.

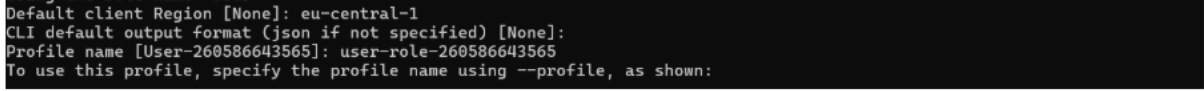


A window in your browser will be opened, enter or verify the code and press continue:

*CLI default client Region = eu-central-1*

*CLI default output format - keep blank and press enter*

*CLI profile name – user-role-260586643565*

**h. Check your aws config file. The config file is located at ~/.aws/config on Linux or macOS, or at C:\Users\USERNAME\.aws\config on Windows.<https://docs.aws.amazon.com/cli/latest/userguide/cli-chap-configure.html>

You should have something like:

*[profile user-role-260586643565]*

*sso\_session = user-role*

*sso\_account\_id = 260586643565*

*sso\_role\_name = User*

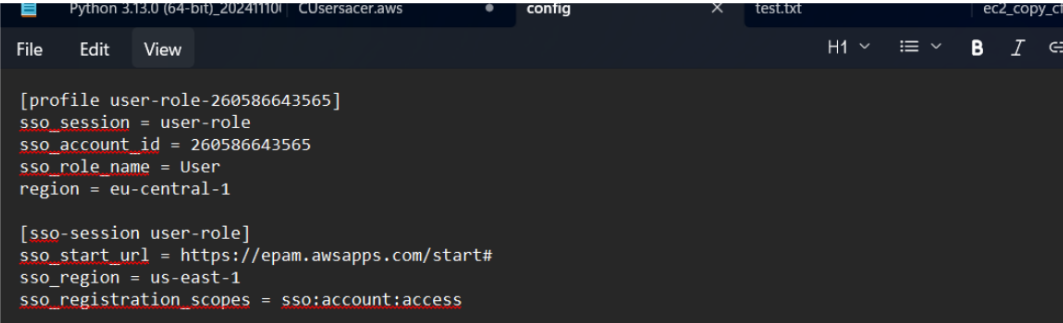
*region = eu-central-1*

*[sso-session user-role]*

*sso\_start\_url = https://epam.awsapps.com/start#/*

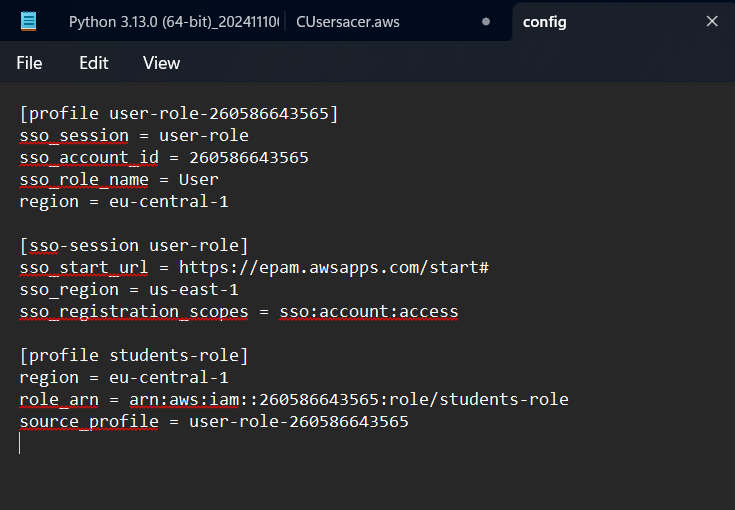
*sso\_region = us-east-1*

*sso\_registration\_scopes = sso:account:access*

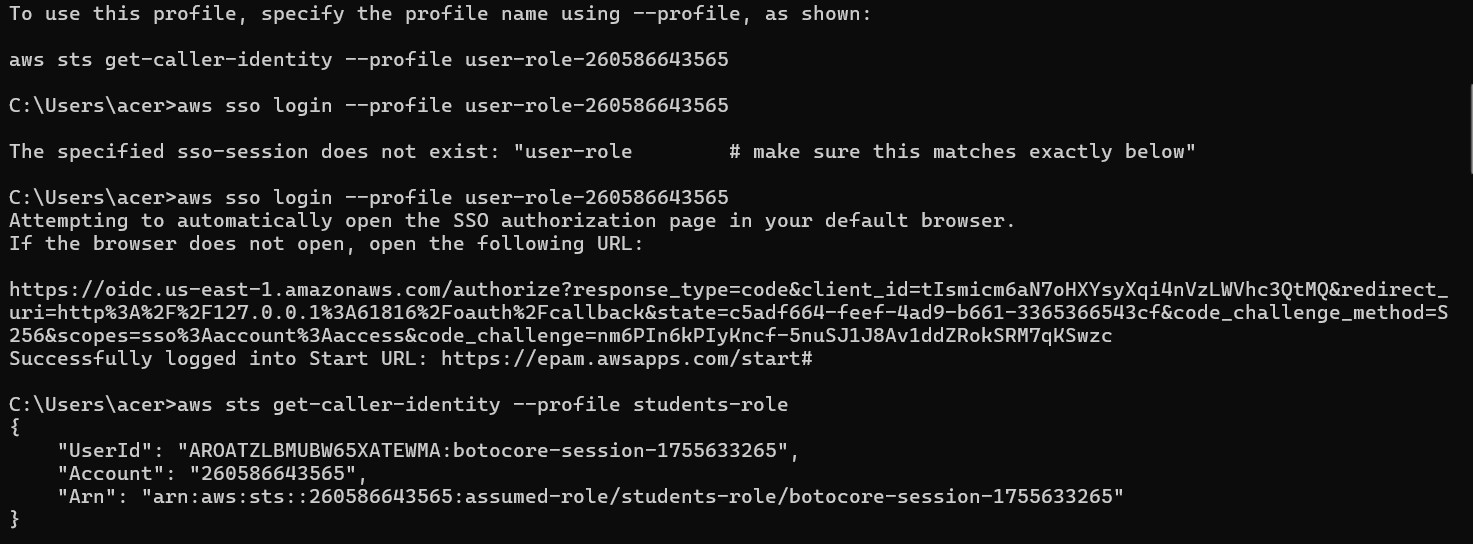
**

*The config file is located at this address "C:\Users\acer\.aws\config"*

i. Now assume *students-role* from *user-role-260586643565*, as **only** *students-role* has necessary permissions. To do this add new profile for *students-role* into config file. Here is documentation how you could do this<https://docs.aws.amazon.com/cli/latest/userguide/cli-configure-files.html> [IAM Role section].  
 Profile name should be *students-role,* region = *eu-central-1*

**

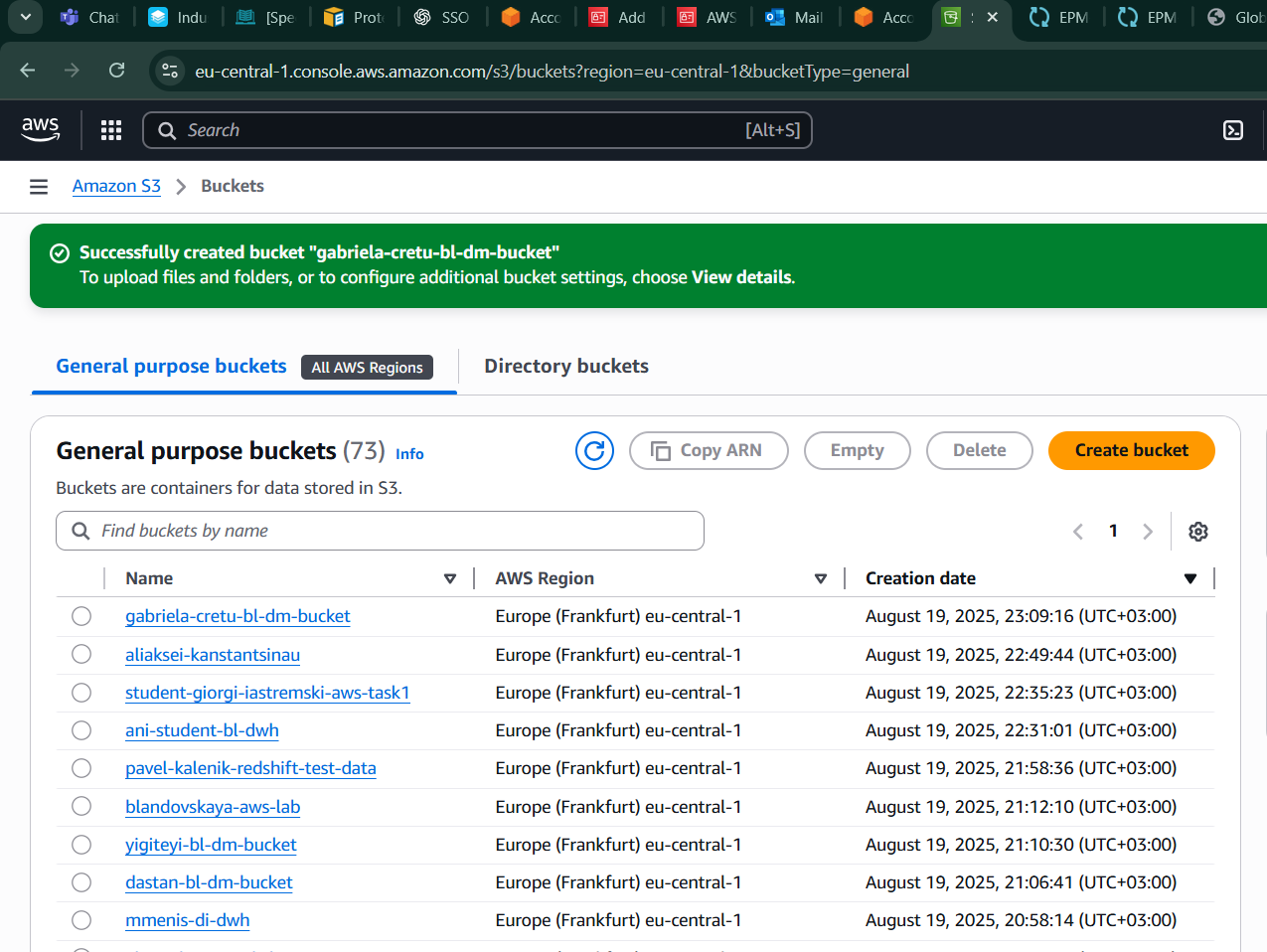
j. Now run *aws sts get-caller-identity --profile students-role* in case you did all right you should see something like



# 2 Working with AWS services:

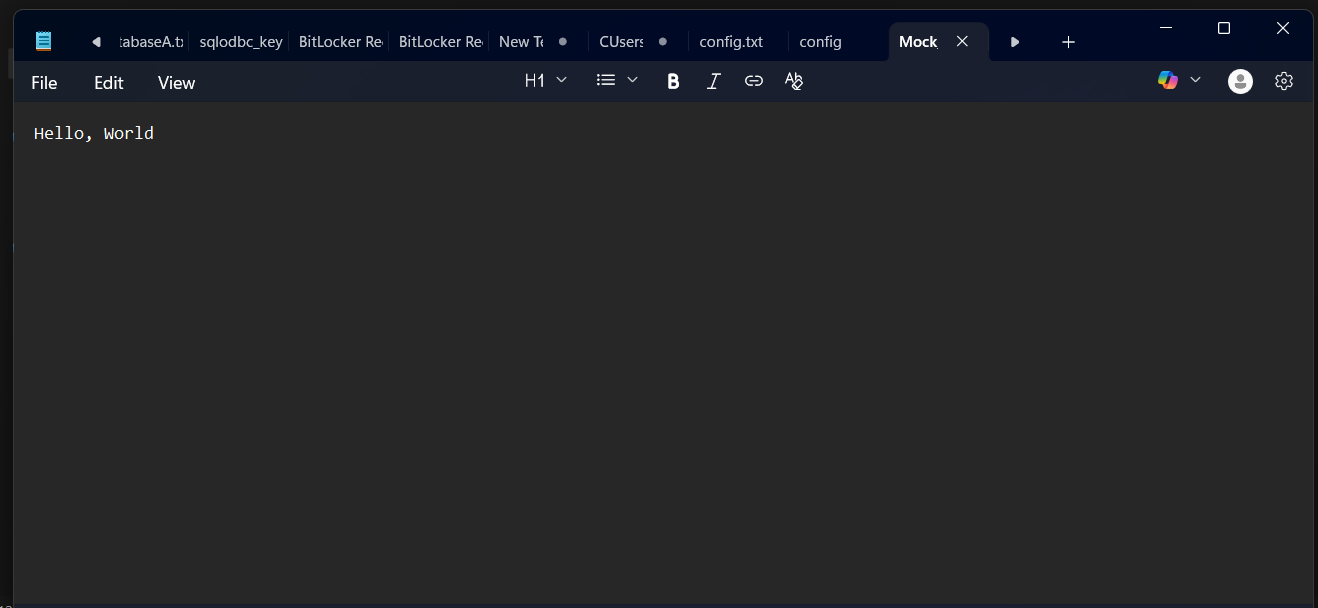
a. Create your own bucket with recognizable naming (to see who the owner from the name).

In the next step, I created a new S3 bucket.

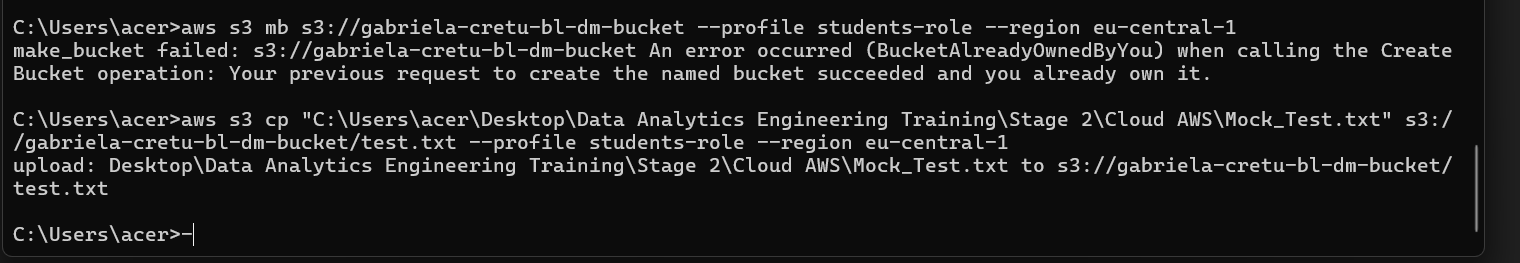


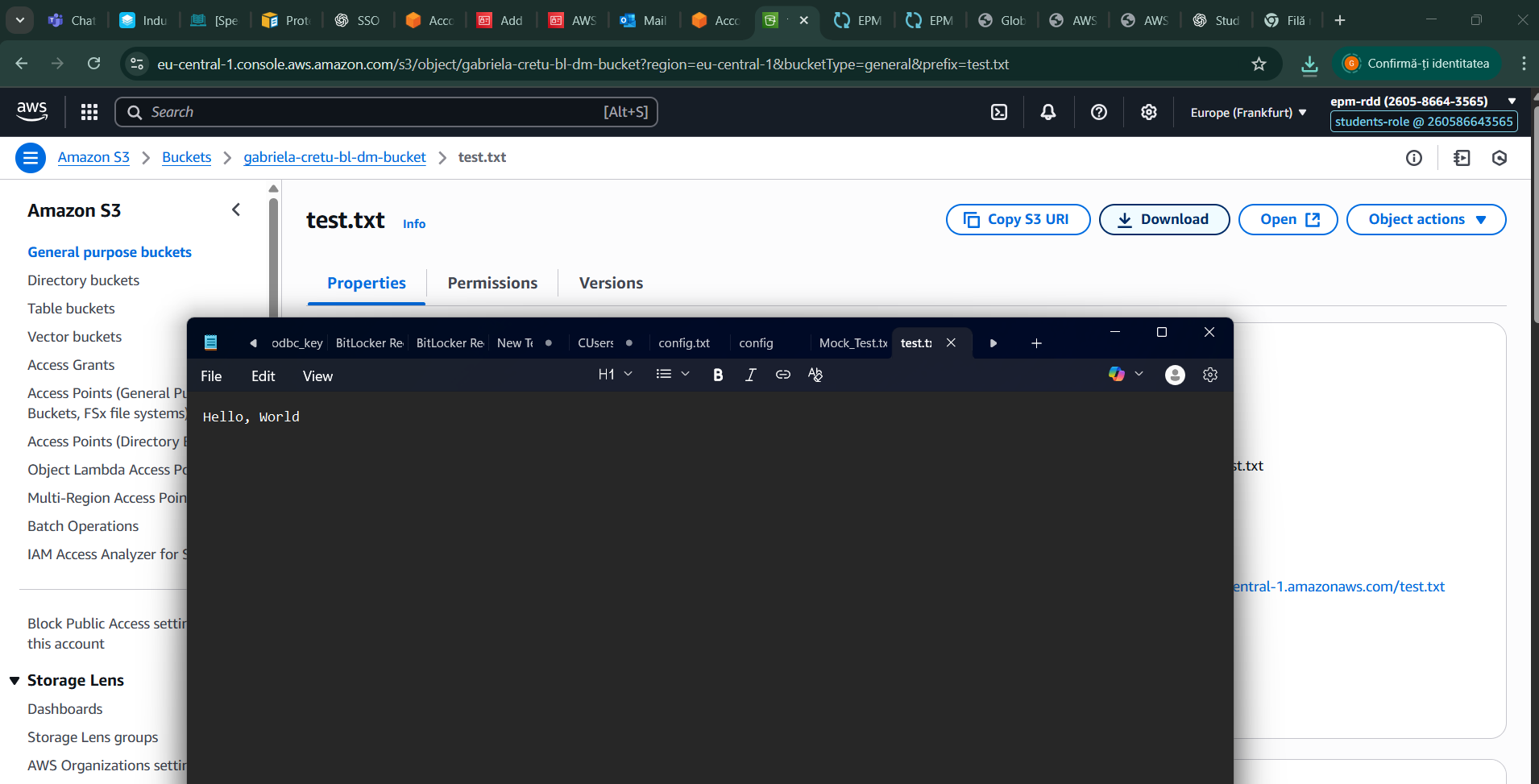
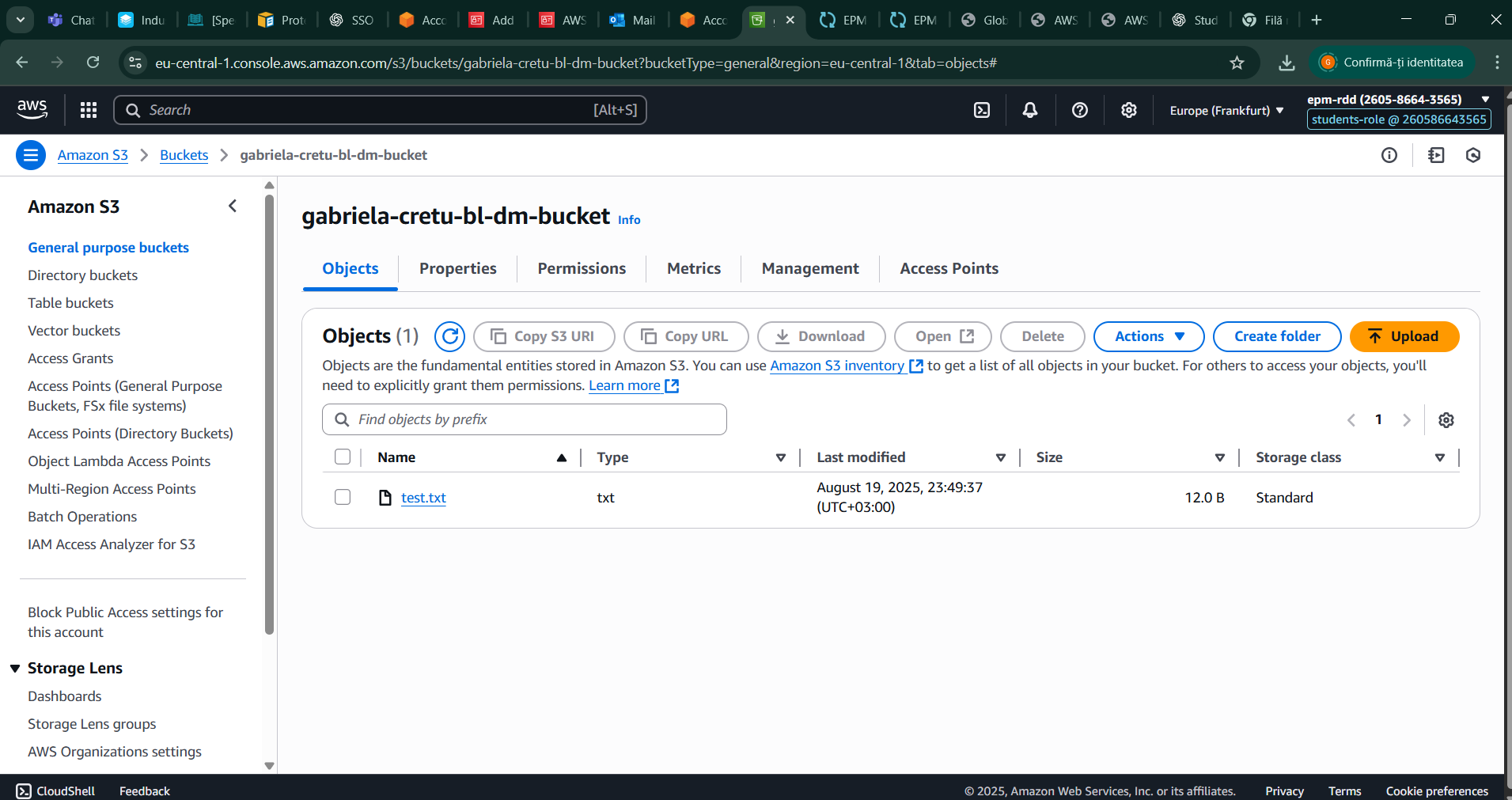
b. Use students-profile to run any aws cli command for s3: at least to put files using api call from local PC to the Cloud.

Next, I created a mock test file to verify that I could upload it to my S3 bucket.



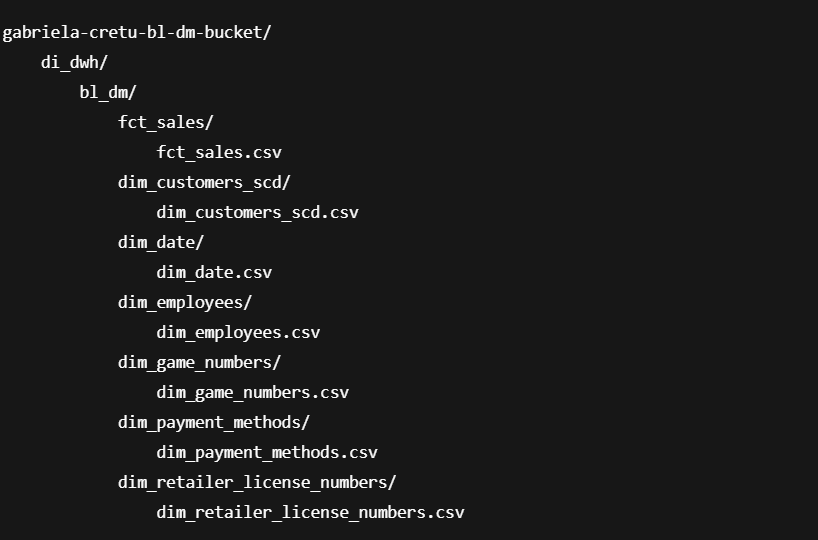
As we can see, the upload was successful, confirming that our connection is working properly.



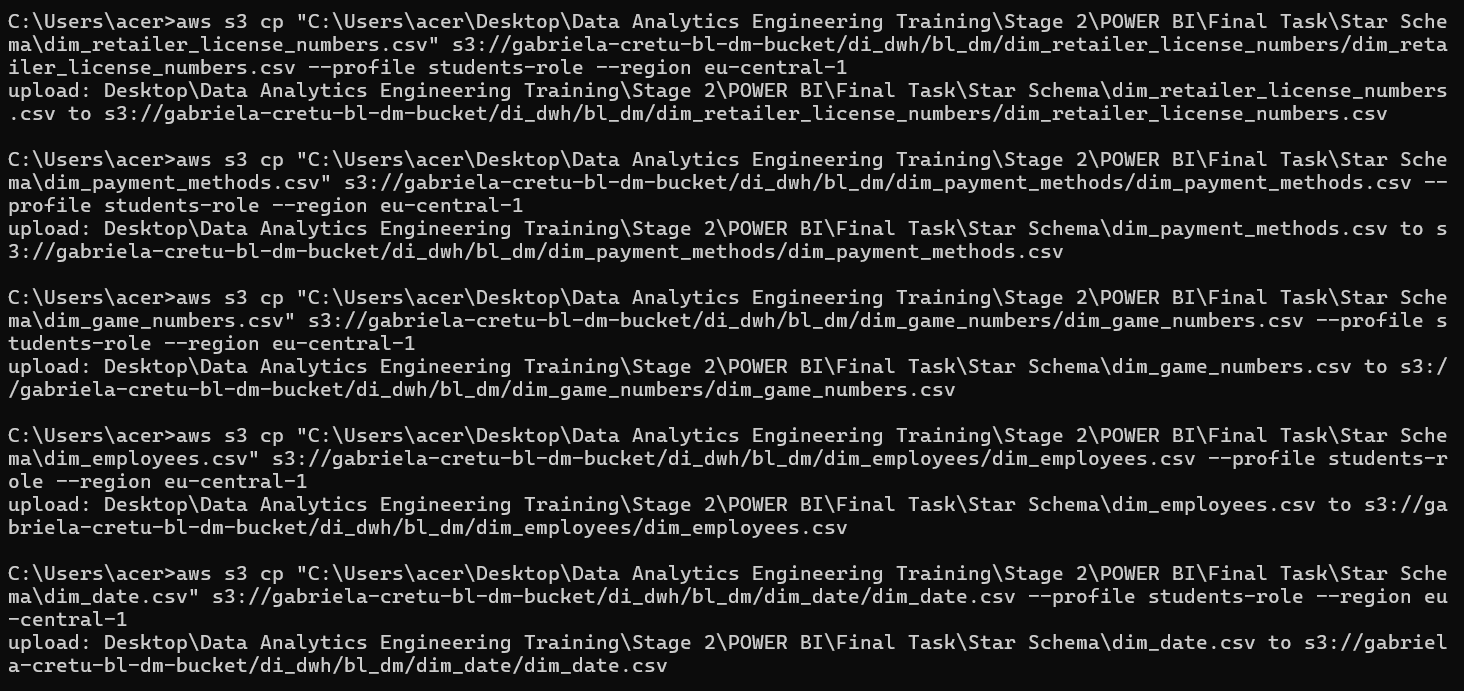


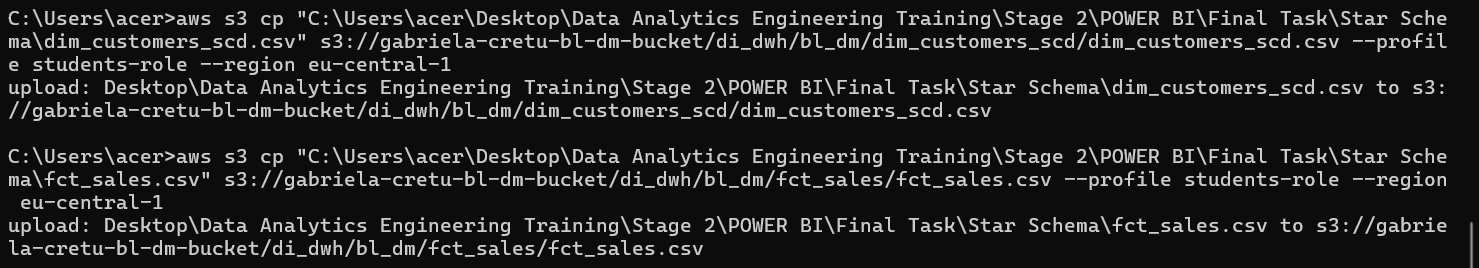
c. Upload you tables from you previously created oracle BL\_DM schema to S3 bucket with appropriate with next folders hierarchy: <your\_**bucket**>/<folder\_di\_dwh\_**database**>/<folder\_**schema**\_bl\_dm>/table1/….files…  
 ( <names> should be substituted by or own )

Next, I configured a folder hierarchy to correctly define my data warehouse in the cloud, following established conventions to ensure it functions properly.



Next, based on the defined hierarchy, I uploaded my tables.



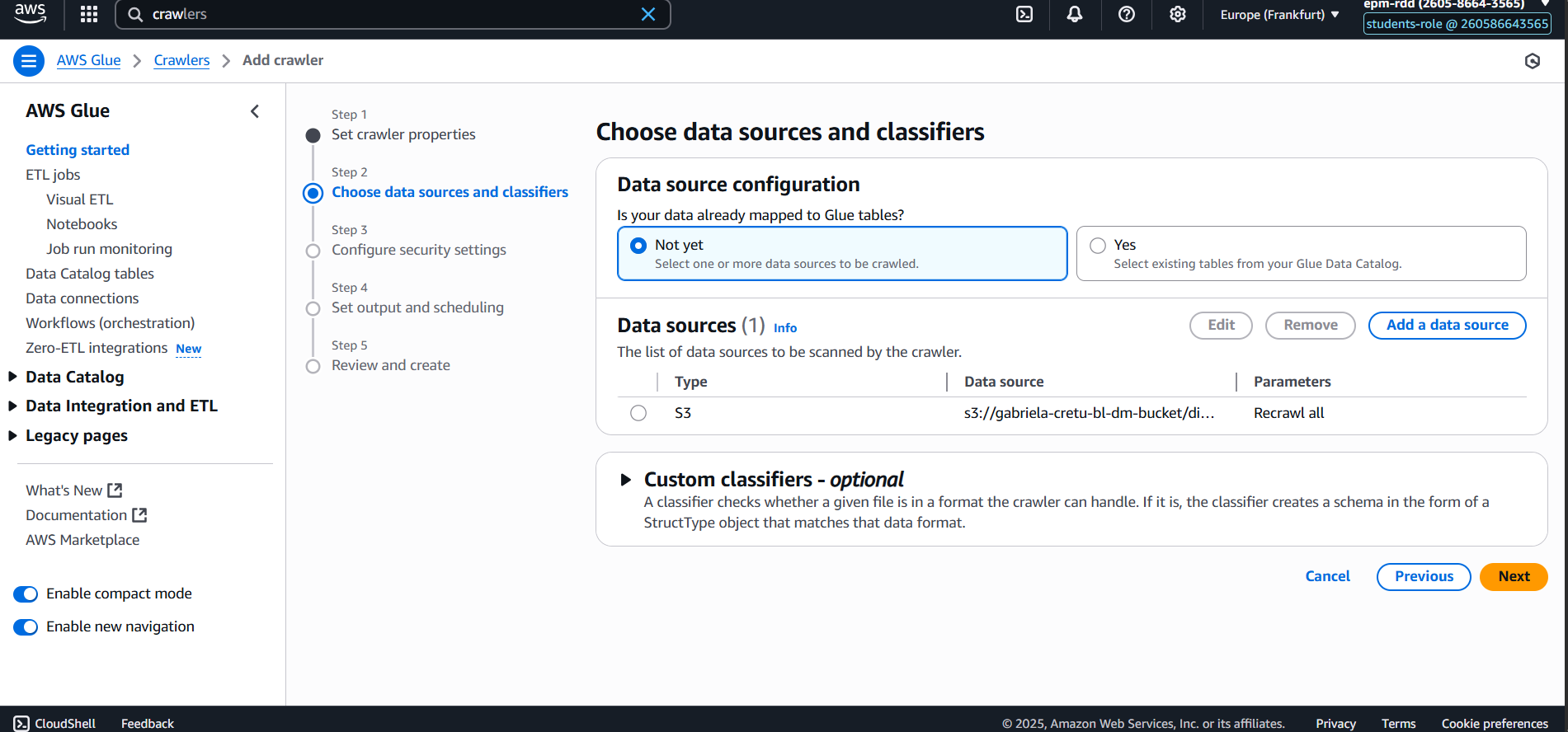
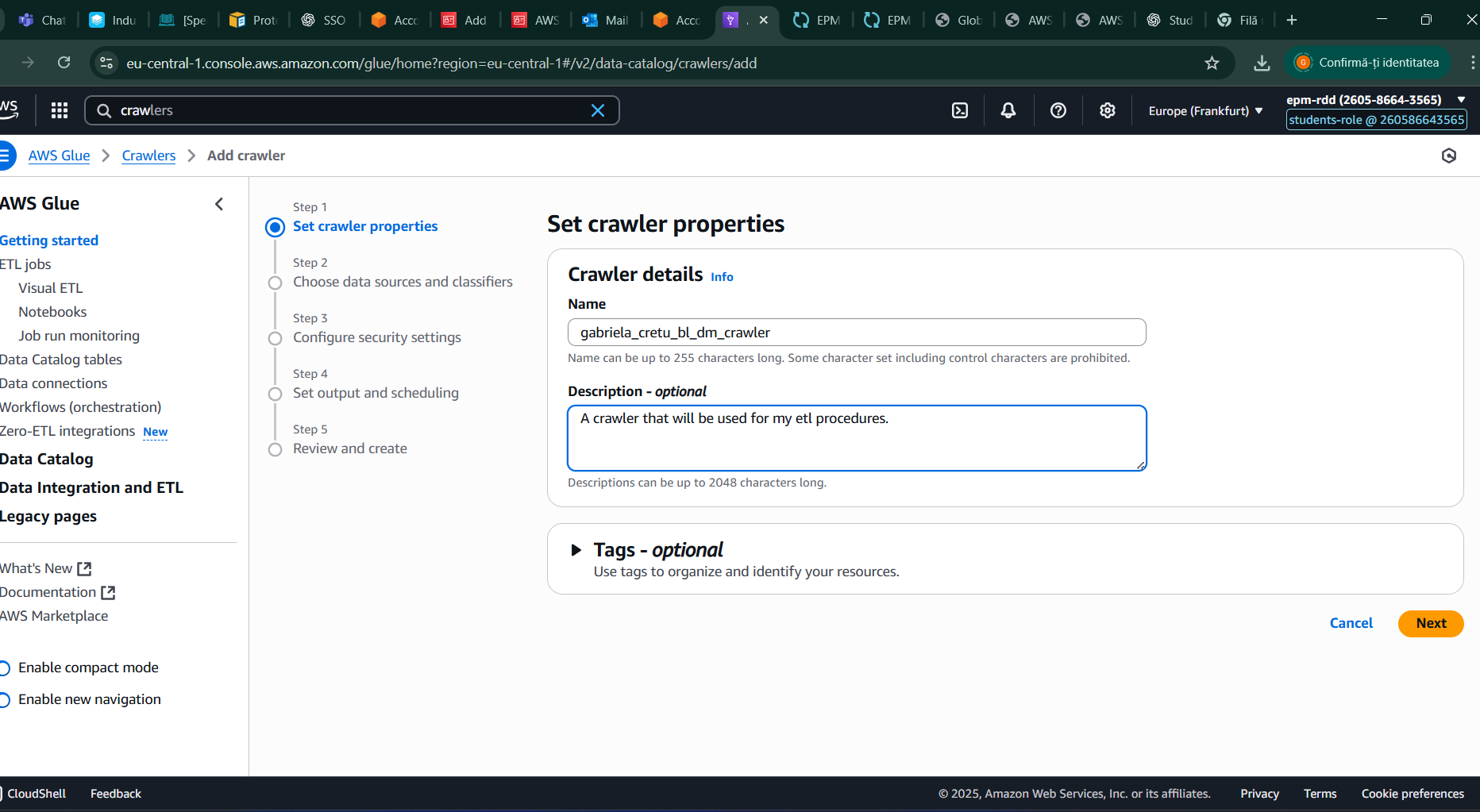


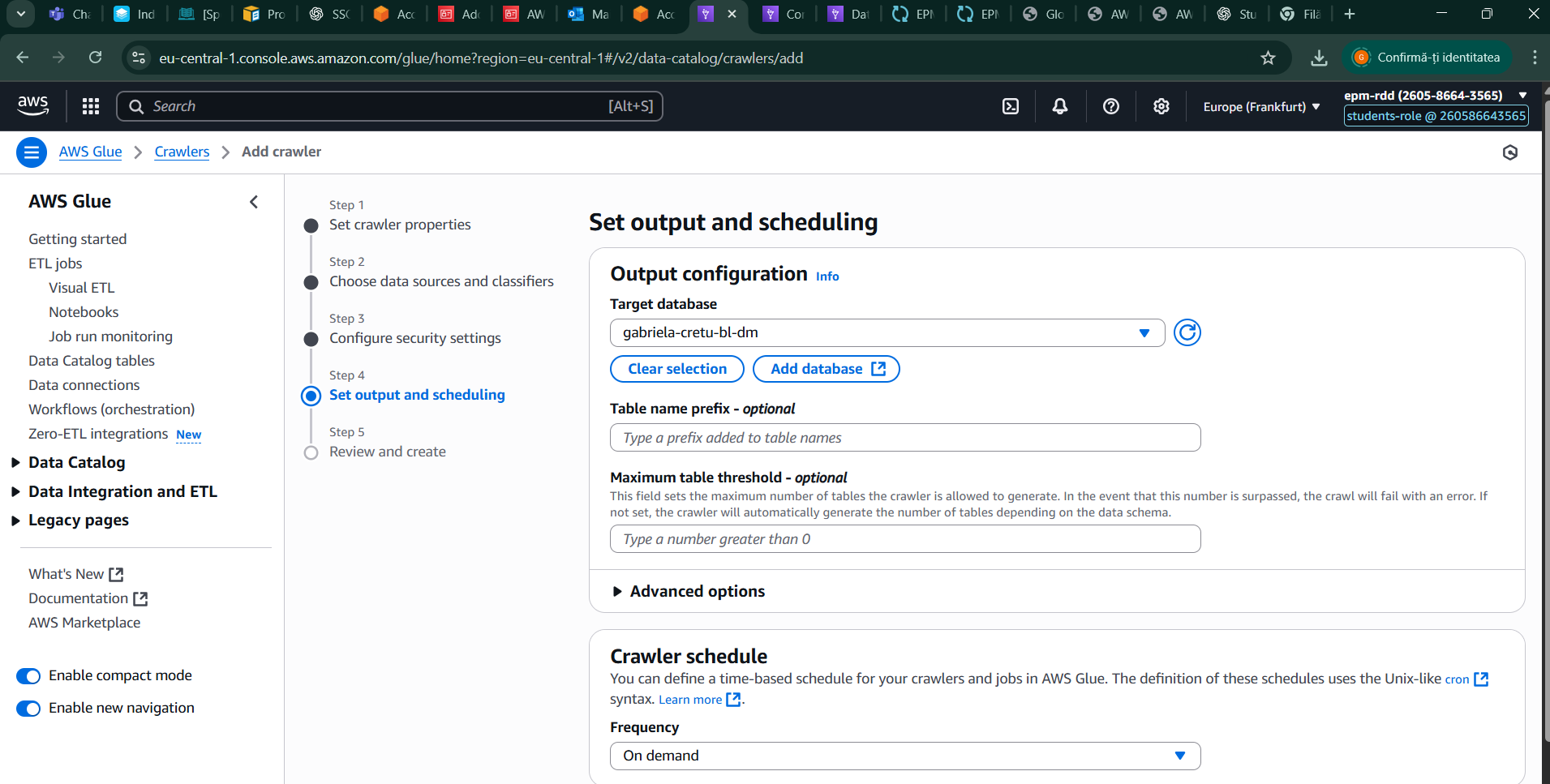
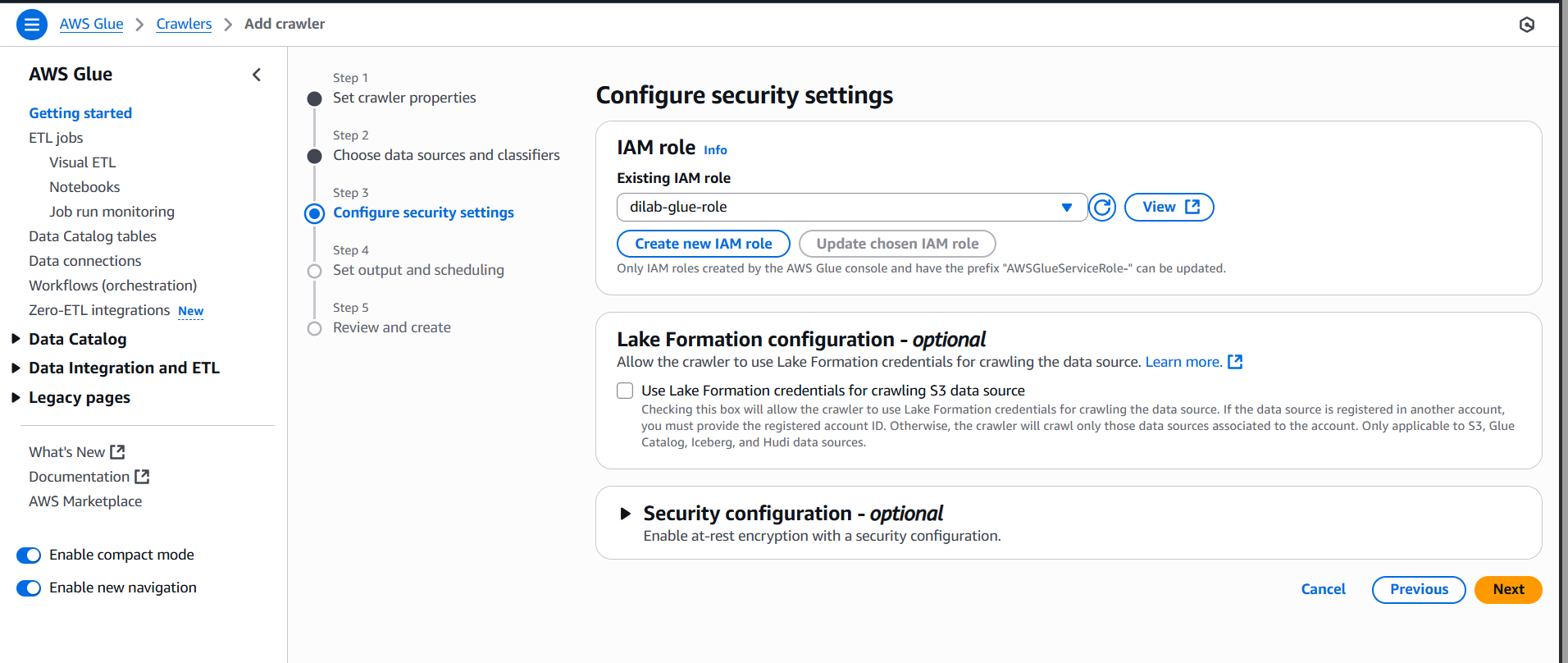
**NOTE:** Please, read AWS documentation before the next step:<https://docs.aws.amazon.com/athena/latest/ug/glue-best-practices.html>

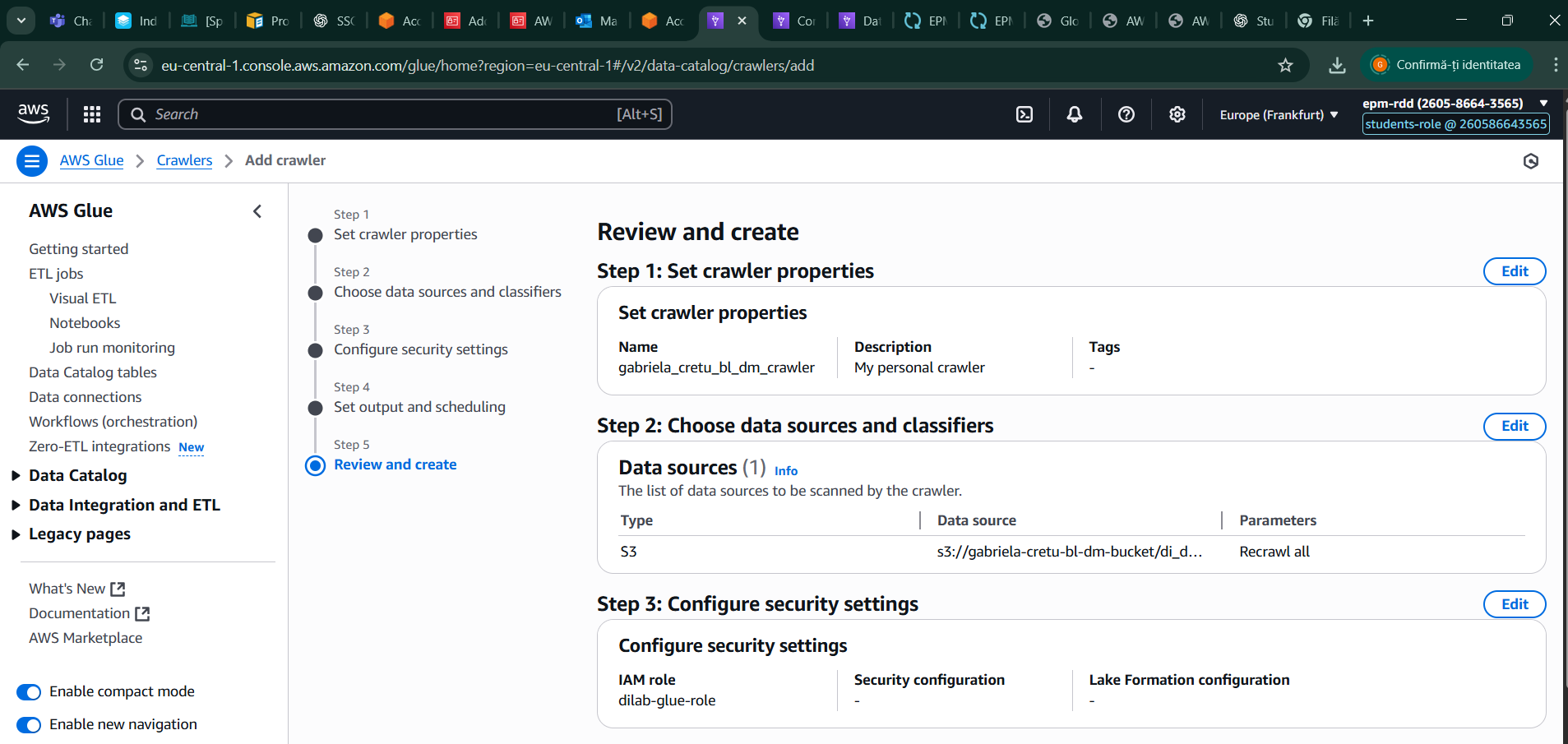
Pay attention at the crawlers naming convention (especially dashes and underscores usage).

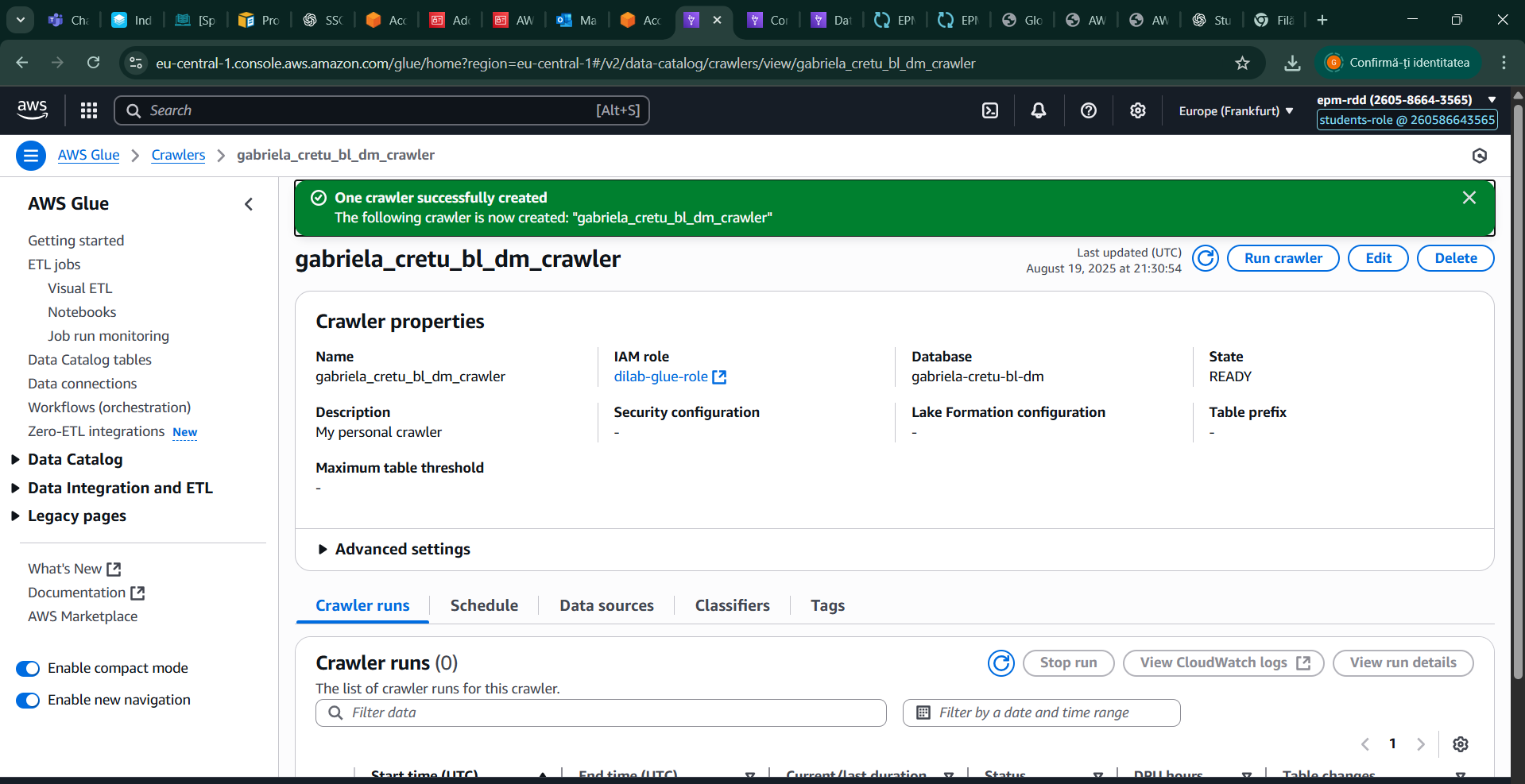
d. Add path to your tables ( <your\_**bucket**>/<folder\_di\_dwh\_**database**>/<folder\_**schema**\_bl\_dm>/… ) into existing Glue Crawler or create your own Crawler to find your tables in buckets and put them into the Glue DataCatalog Database. Run not very simple select on the tables from your s3 path, i.e. execute query in Athena. Don't forget about limits of 10000 ... 50000 rows. Show us properties of the tables, query results.

Next, I created my crawler.

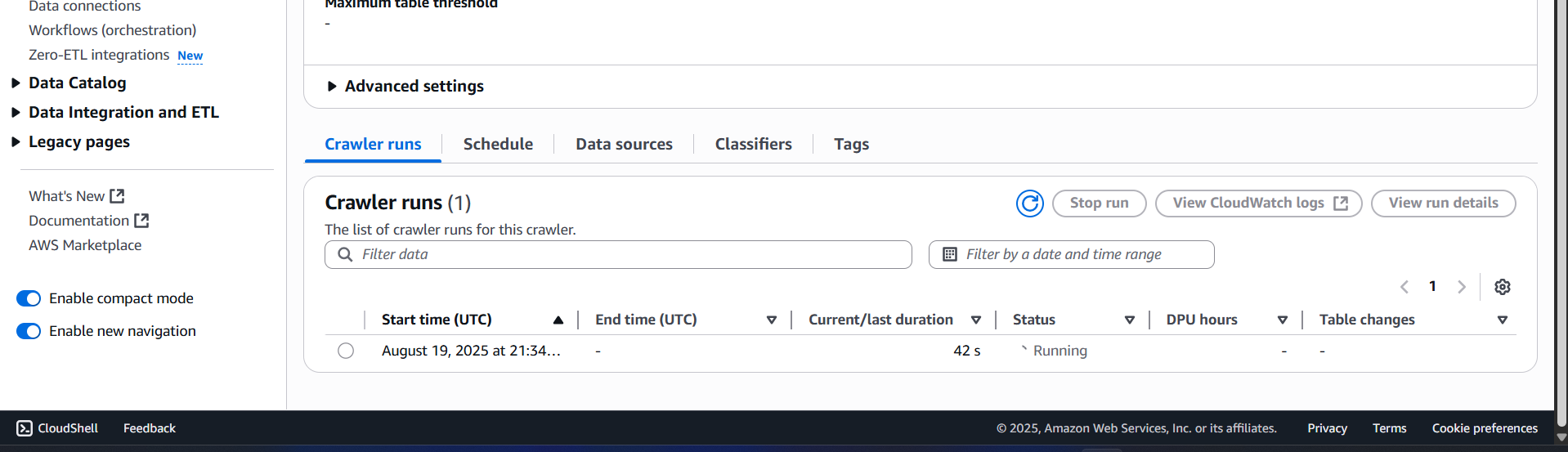
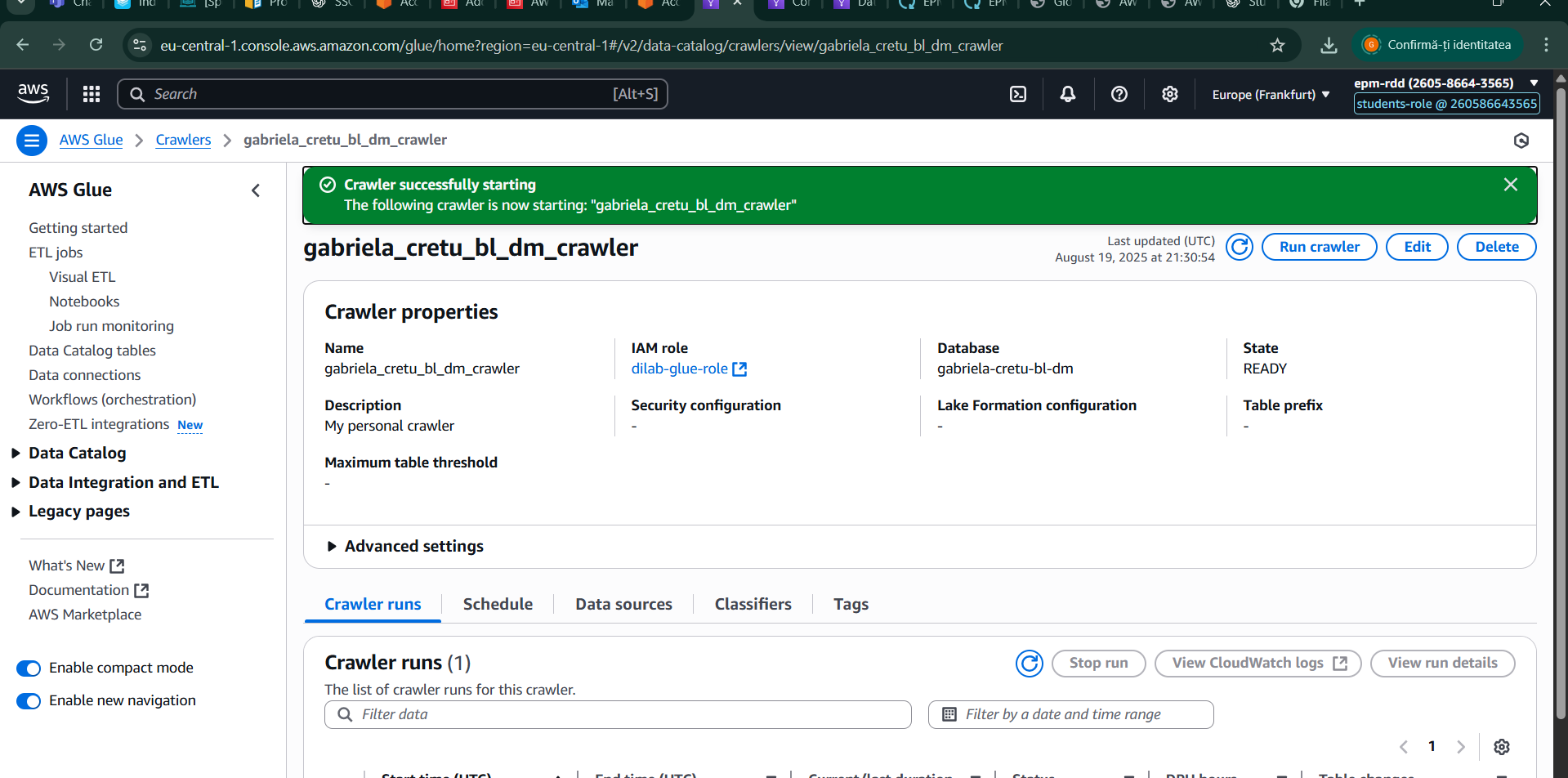




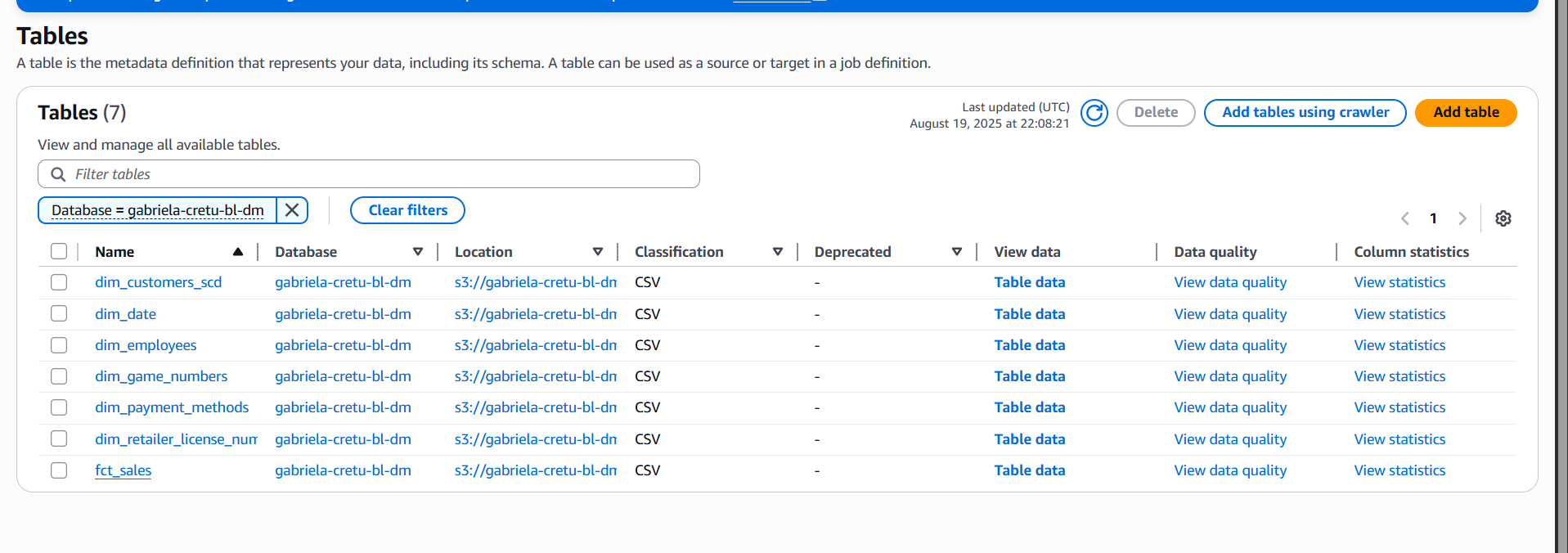


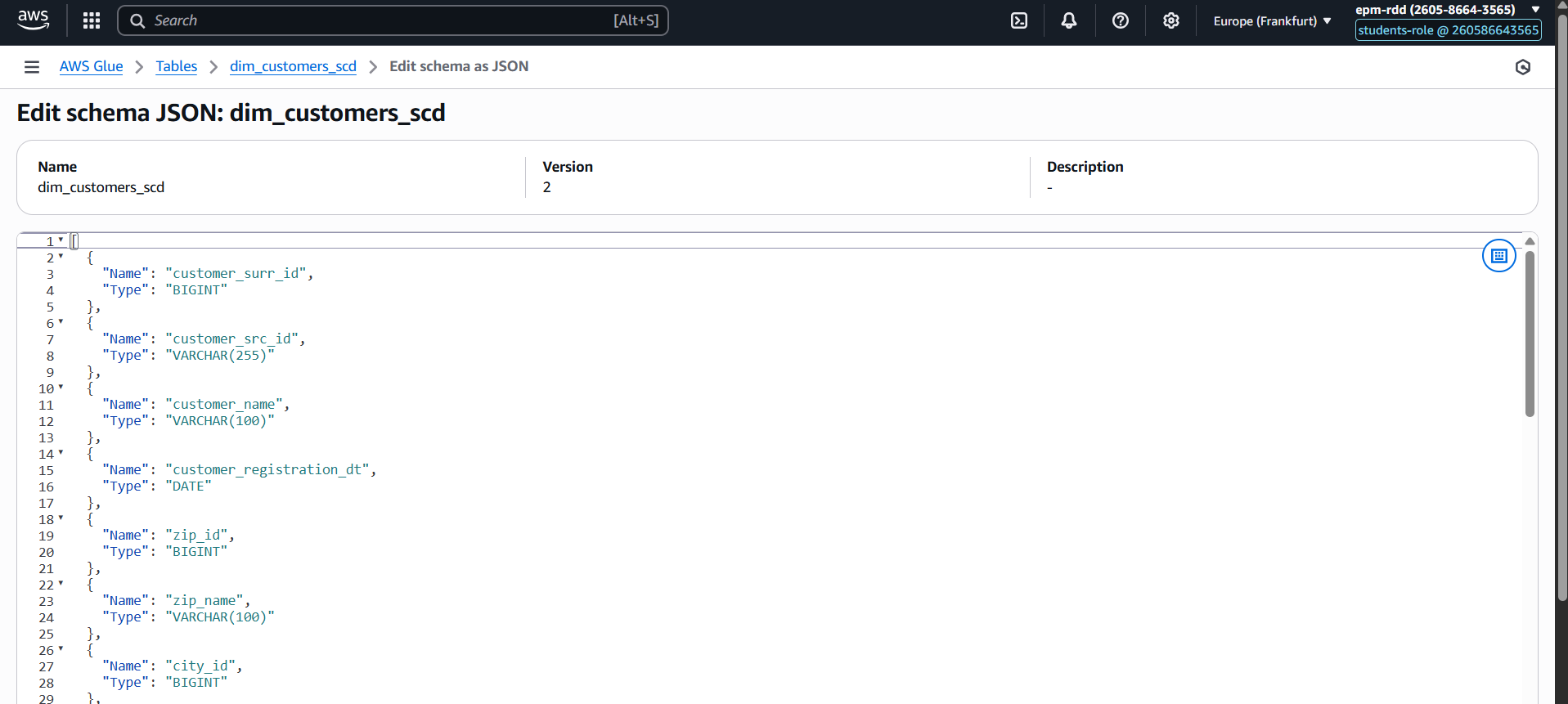


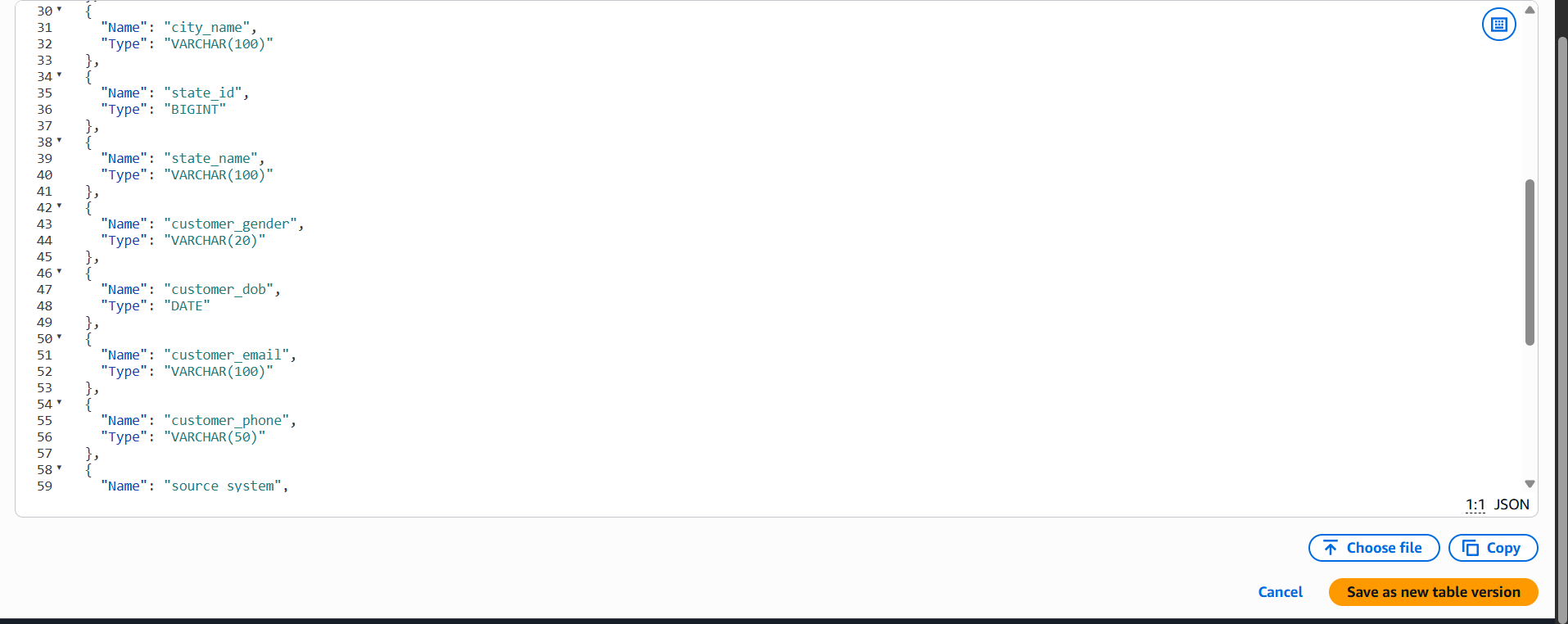
Next, I started the crawler to load my database tables into the cloud

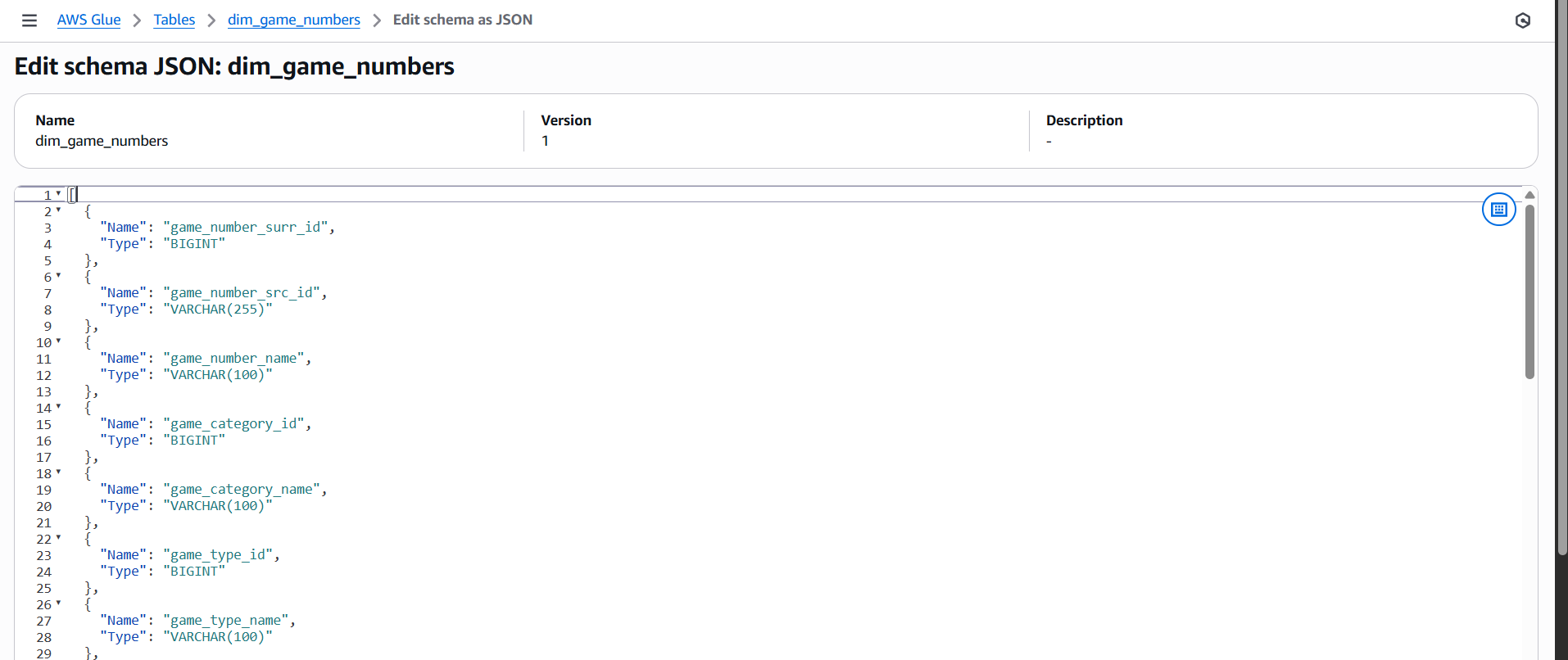


Before querying my tables, I first verified that the data types matched the ones I had defined for each column in the dimension and fact tables in DBeaver. Since the CSV files were not yet parsed, most columns were interpreted as strings, so I had to convert them back to their correct data types to ensure proper functionality during queries. I accomplished this by accessing the tables in AWS Glue, as shown below. Additionally, I updated the JSON code to reflect these correct data types and added partition indexes to the dimension tables based on the primary keys defined in DBeaver, providing more flexibility for querying.

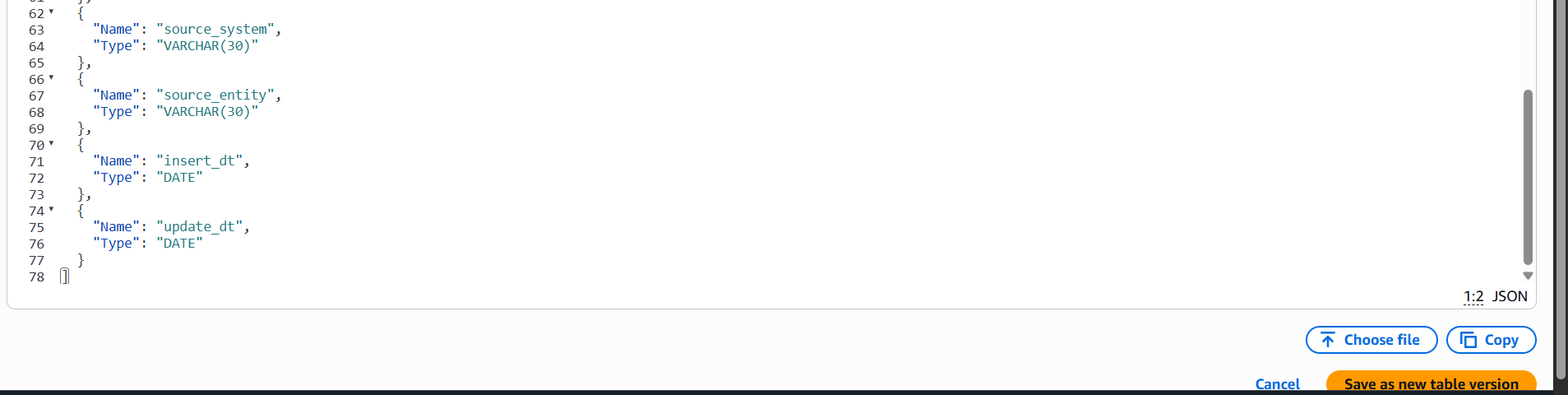


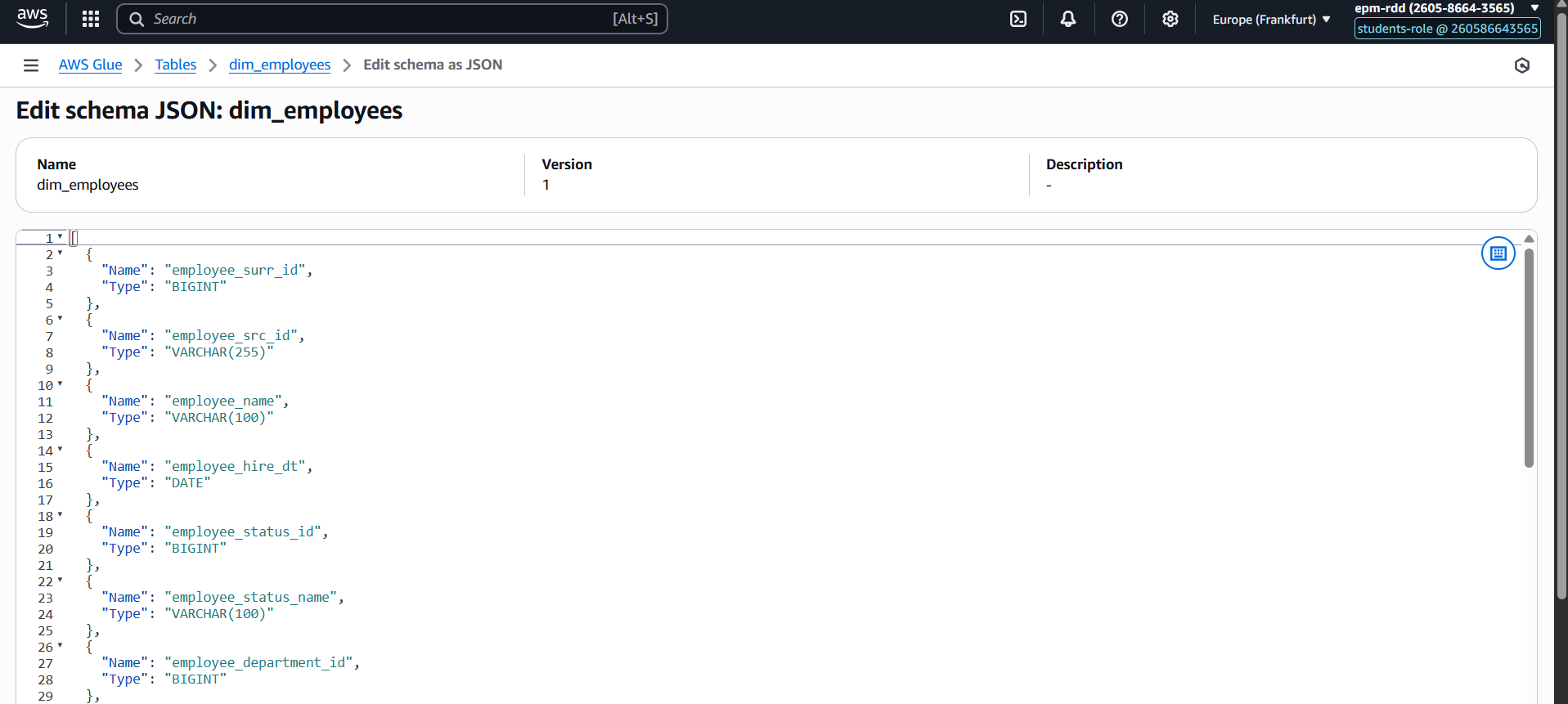


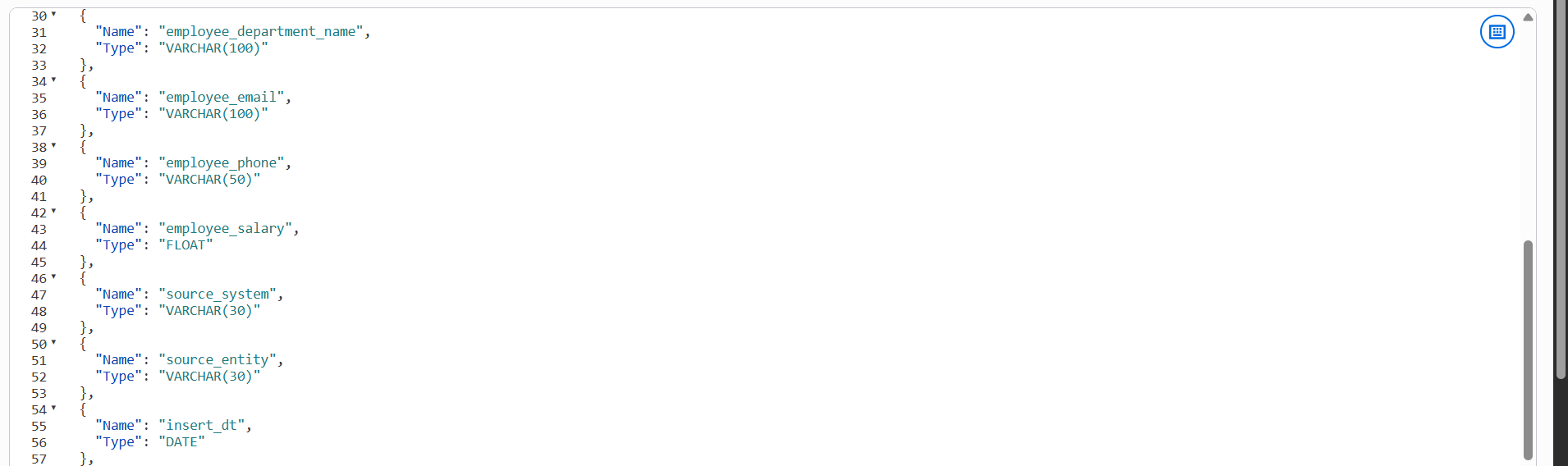


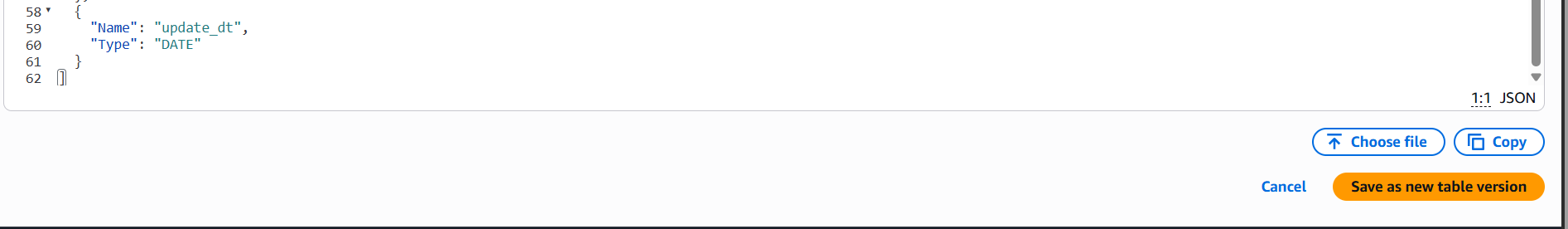


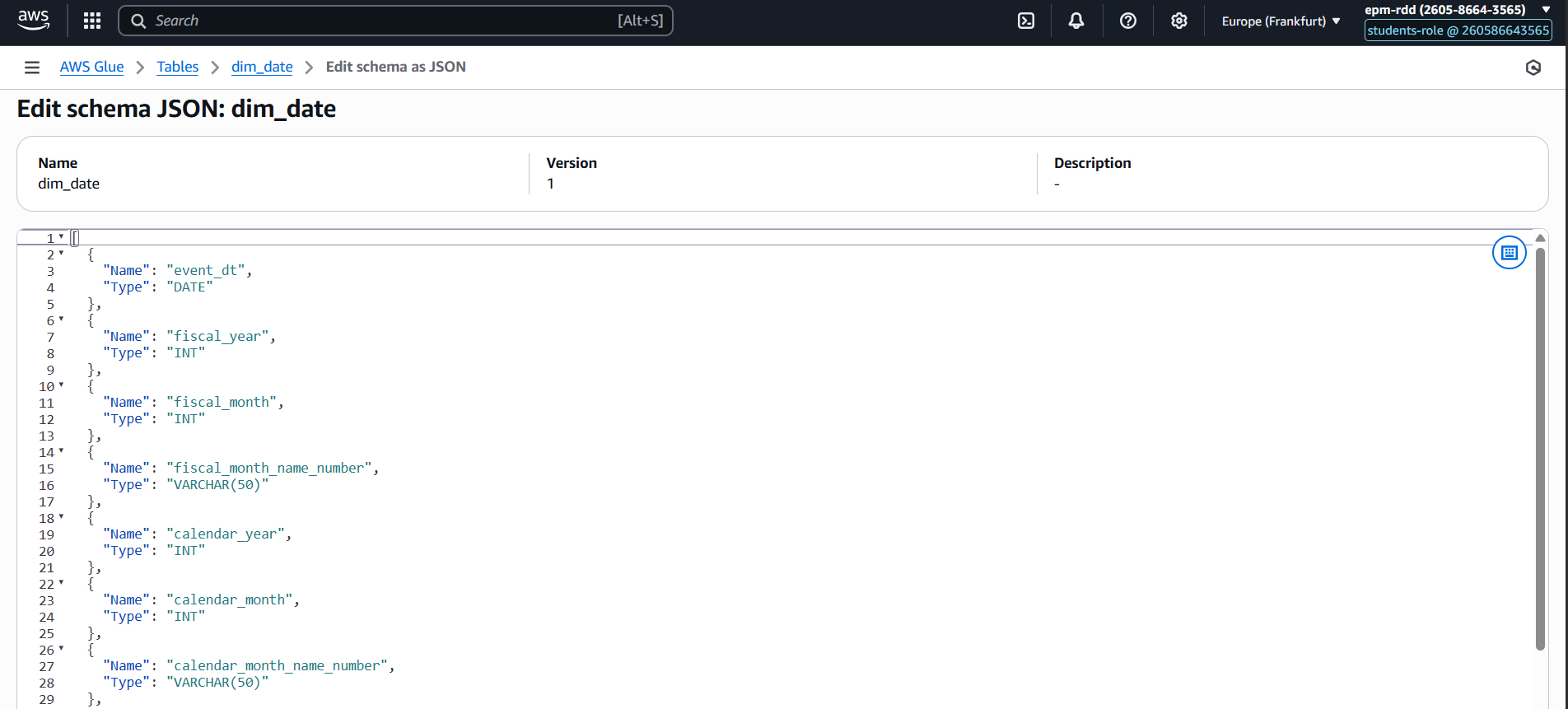


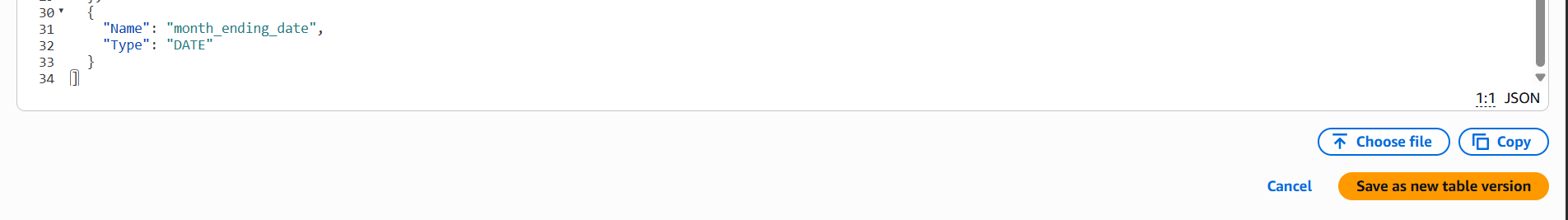






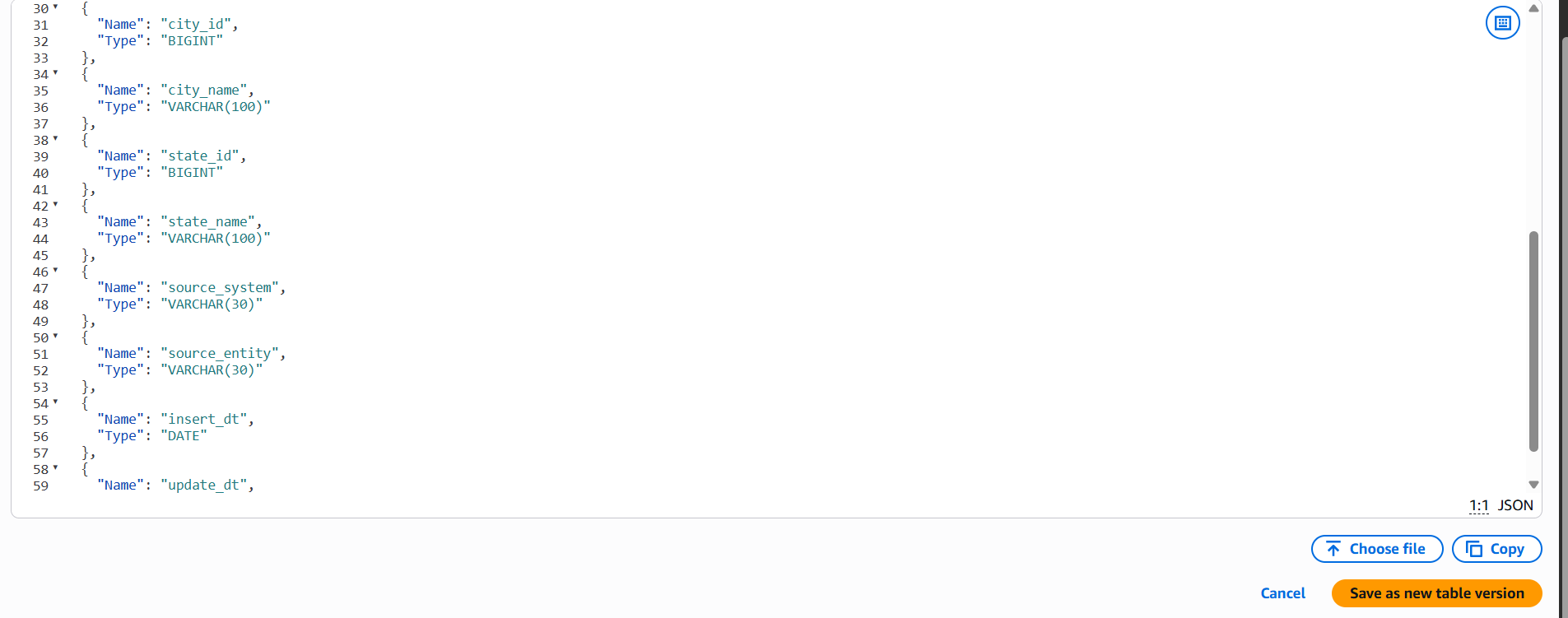


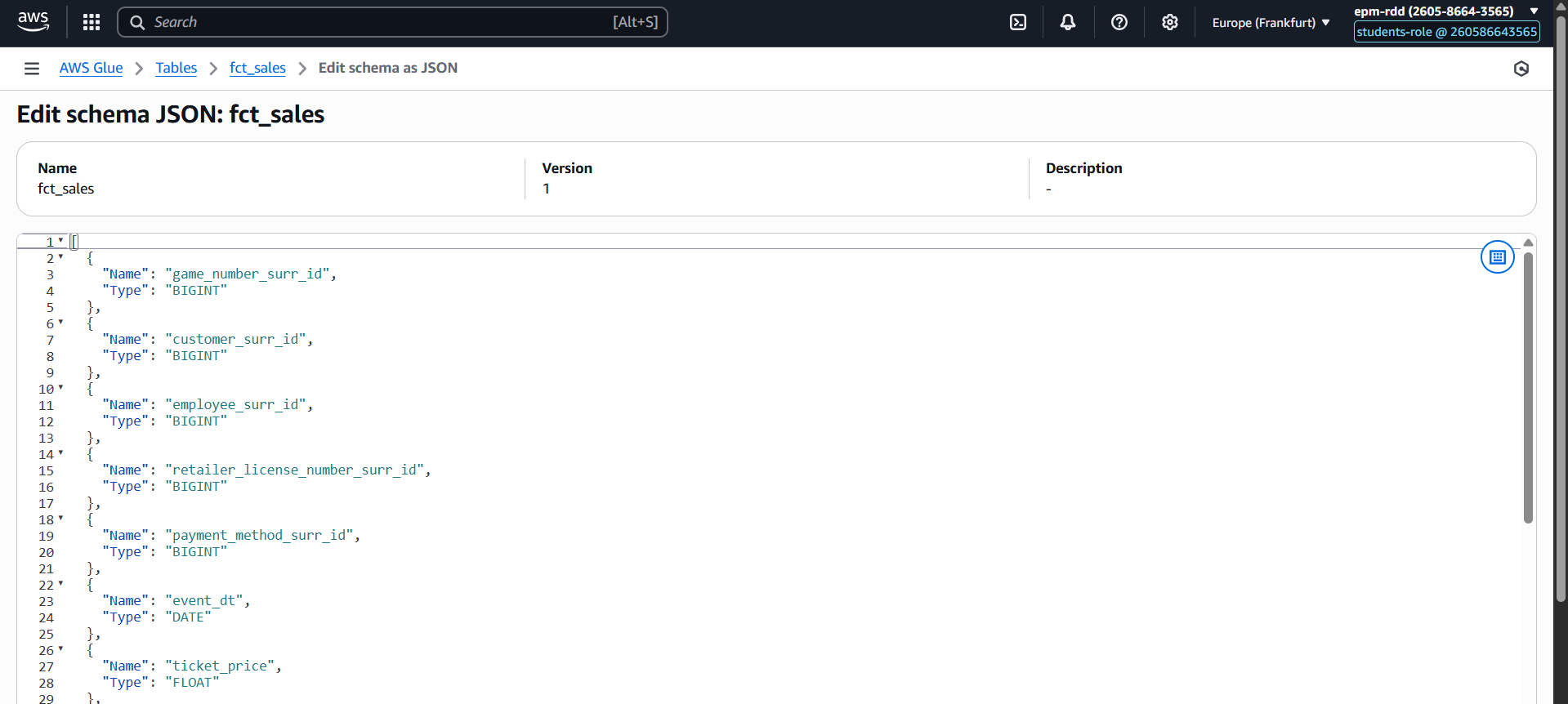


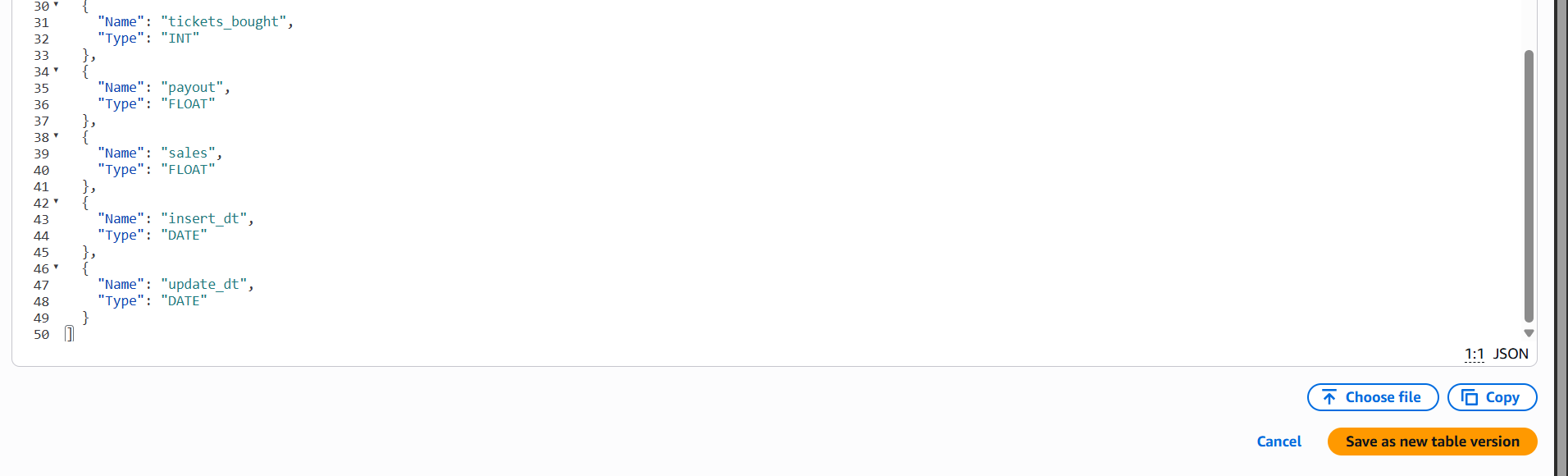












After making sure that the datatypes are as expected I ran some queries as seen here based on joins some select.

