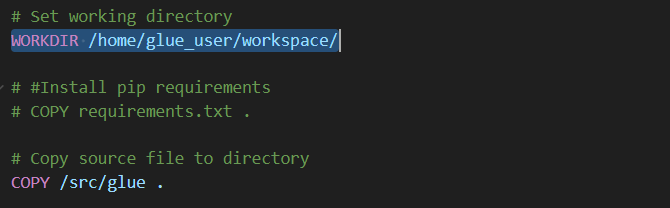
**TASK 4 Documentation**

**: Create Glue Job**

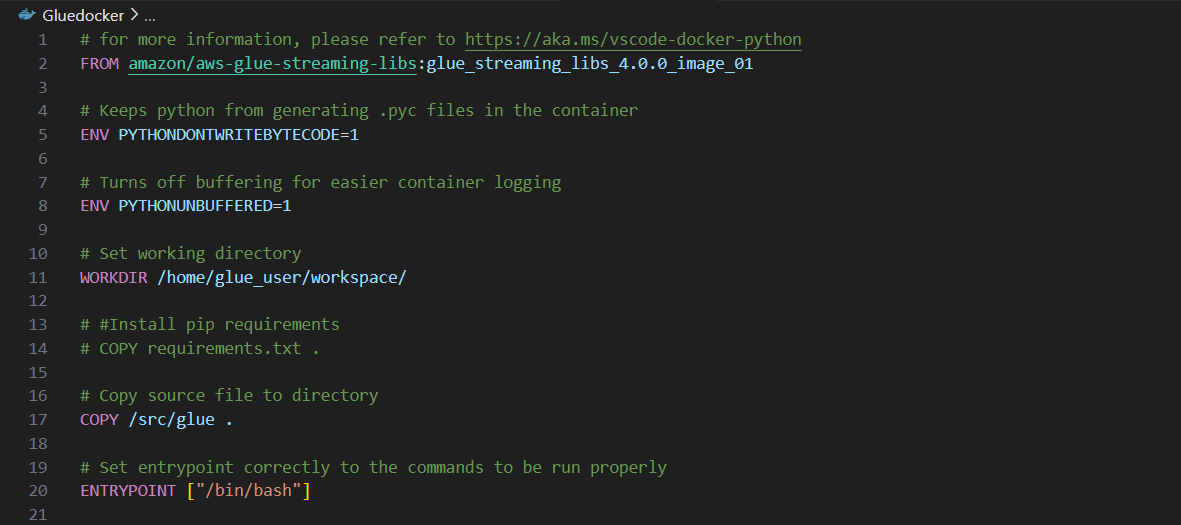
* **Create Docker Glue job to read from Kafka topic (use amazon/aws-glue-libs image)**
* **use local storage to store the data**

**step 1:**

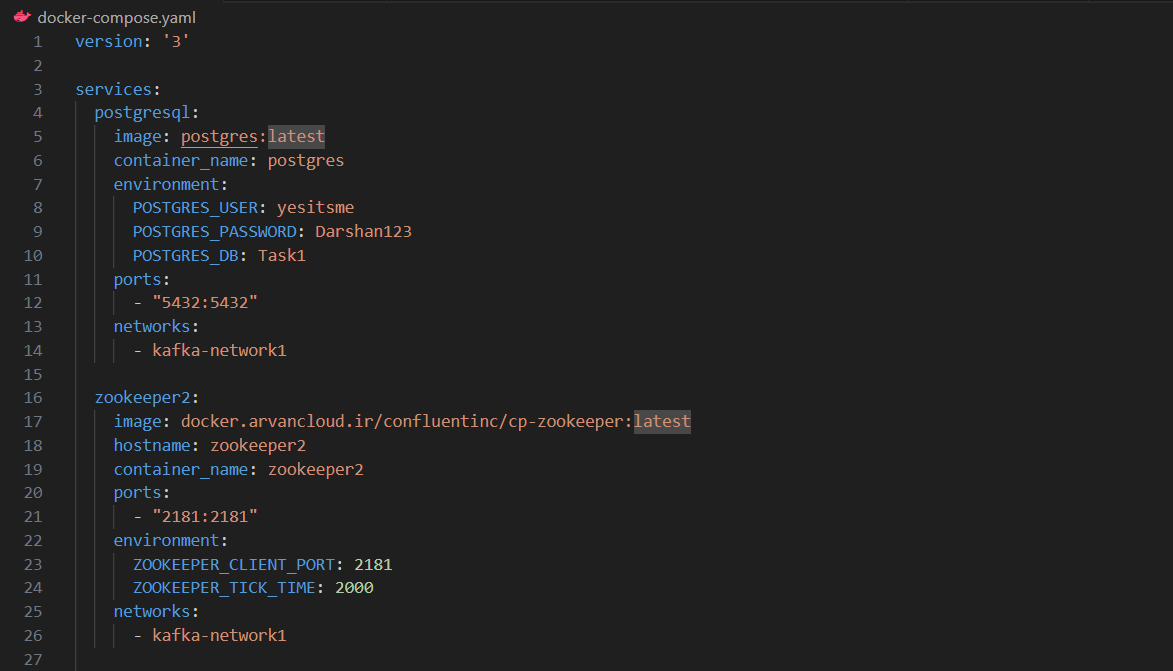
* **Our task is to make docker compose including configurations of aws-glue alog with kafka broker, zookeeper, connect and PostgreSQL.**
* **For the purpose made a separate Gludocker file as well where other configurations of glue such as image, working directory and gave path for csv.**

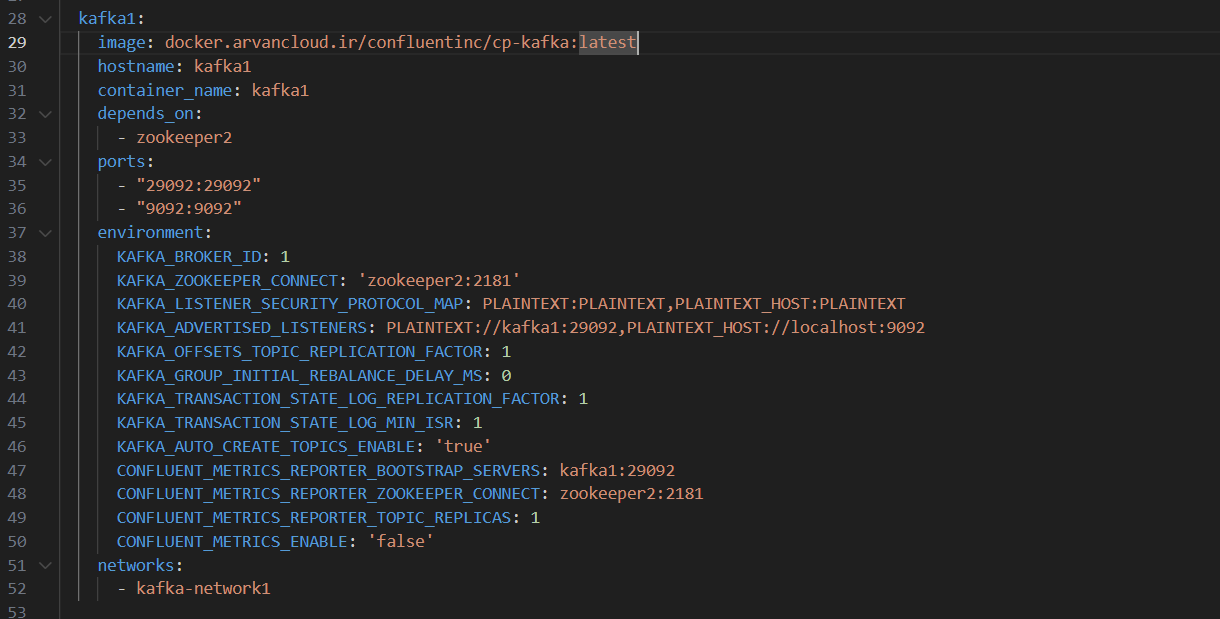
****

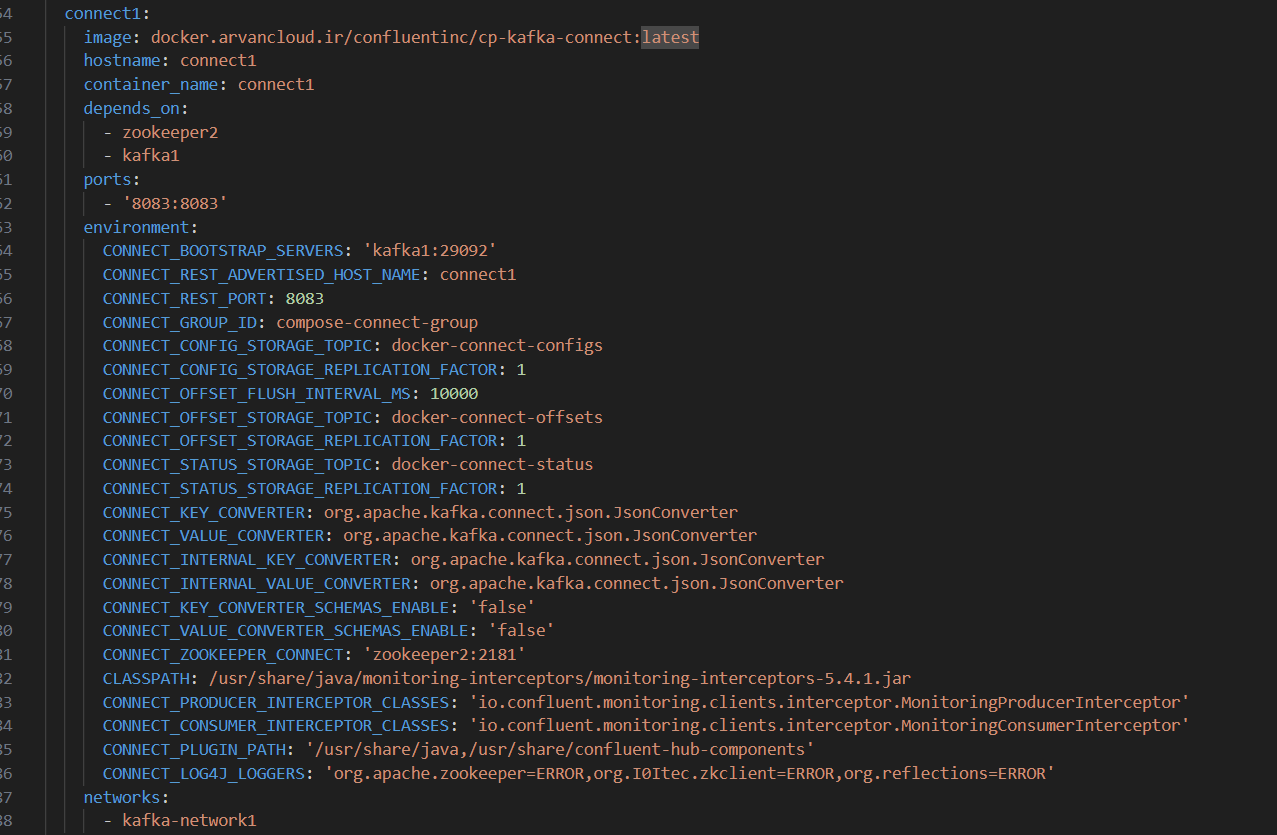
* **Made changes in docker-compose file as by adding configurations of aws-glue; by maintaining the gluedocker file alongside.**

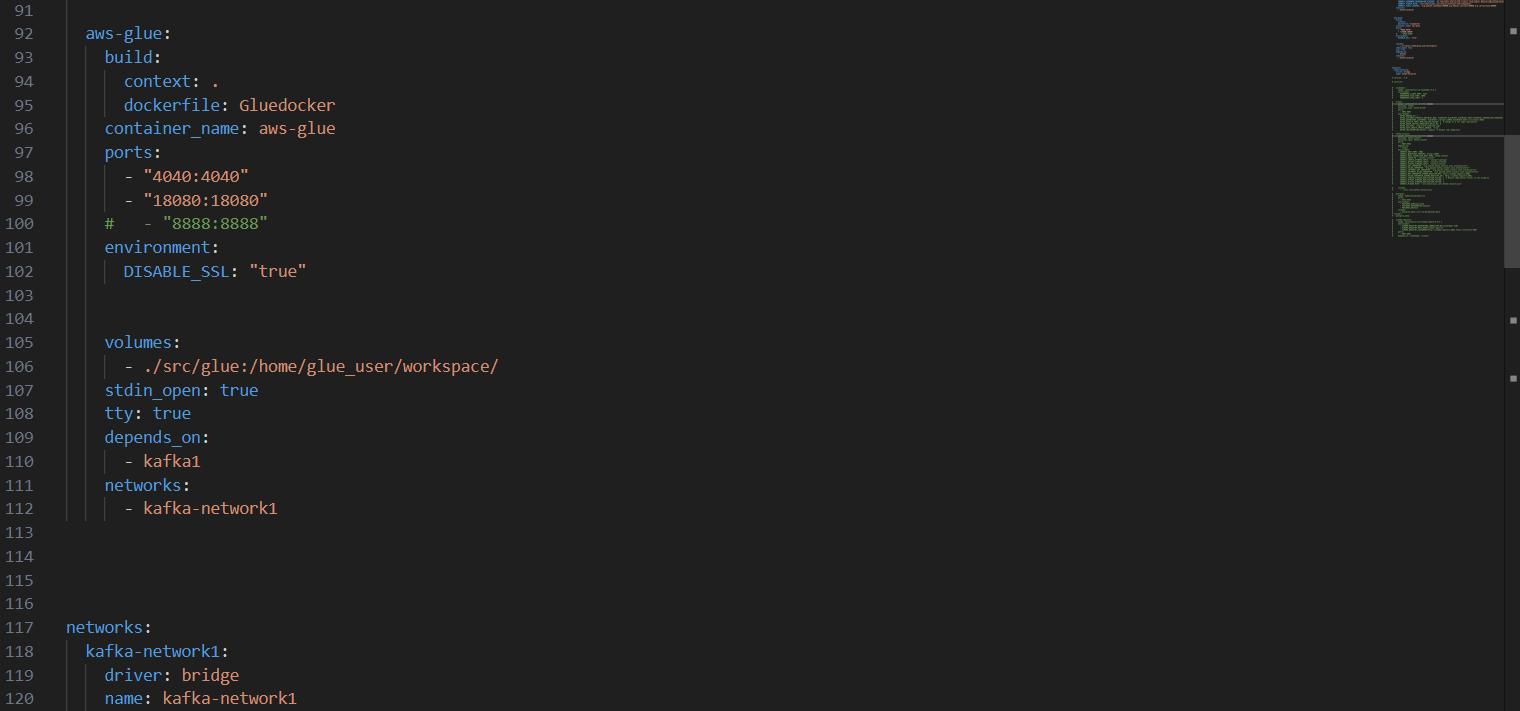
****

* **Now attached those configuration in dokcer compose within the same directory.**

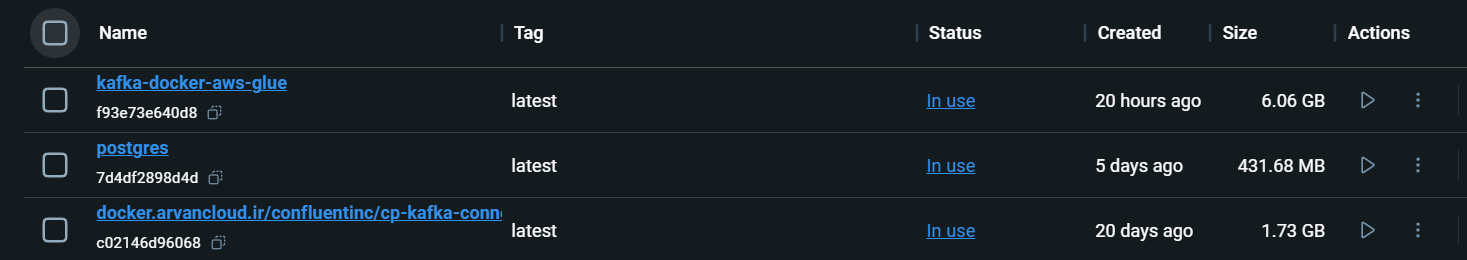
****

****

****

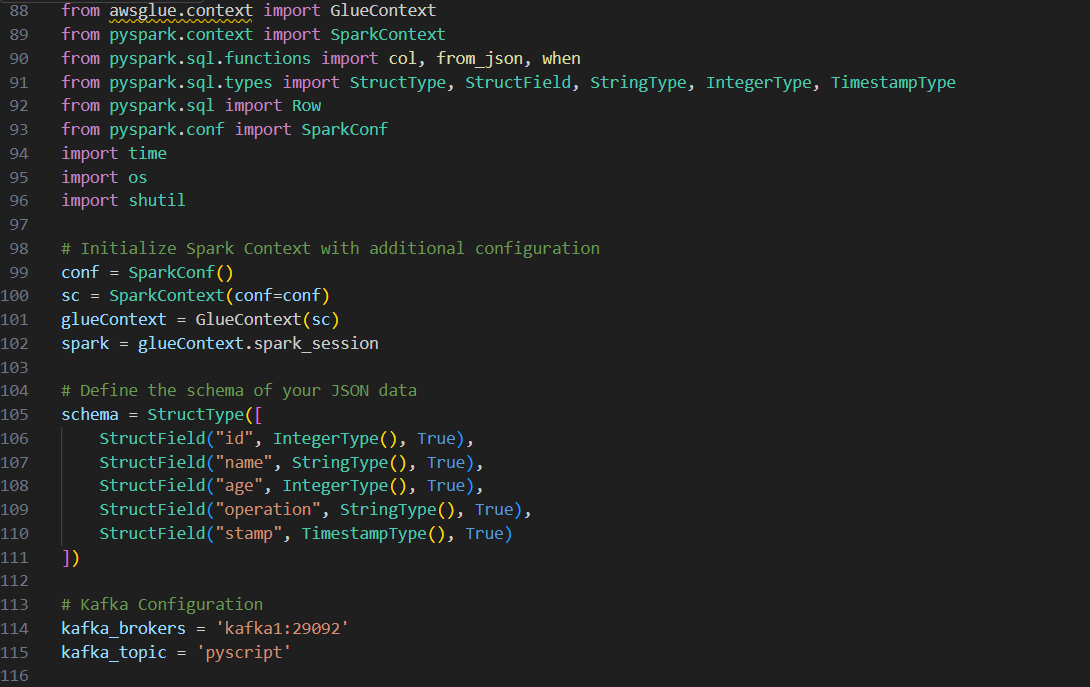
****

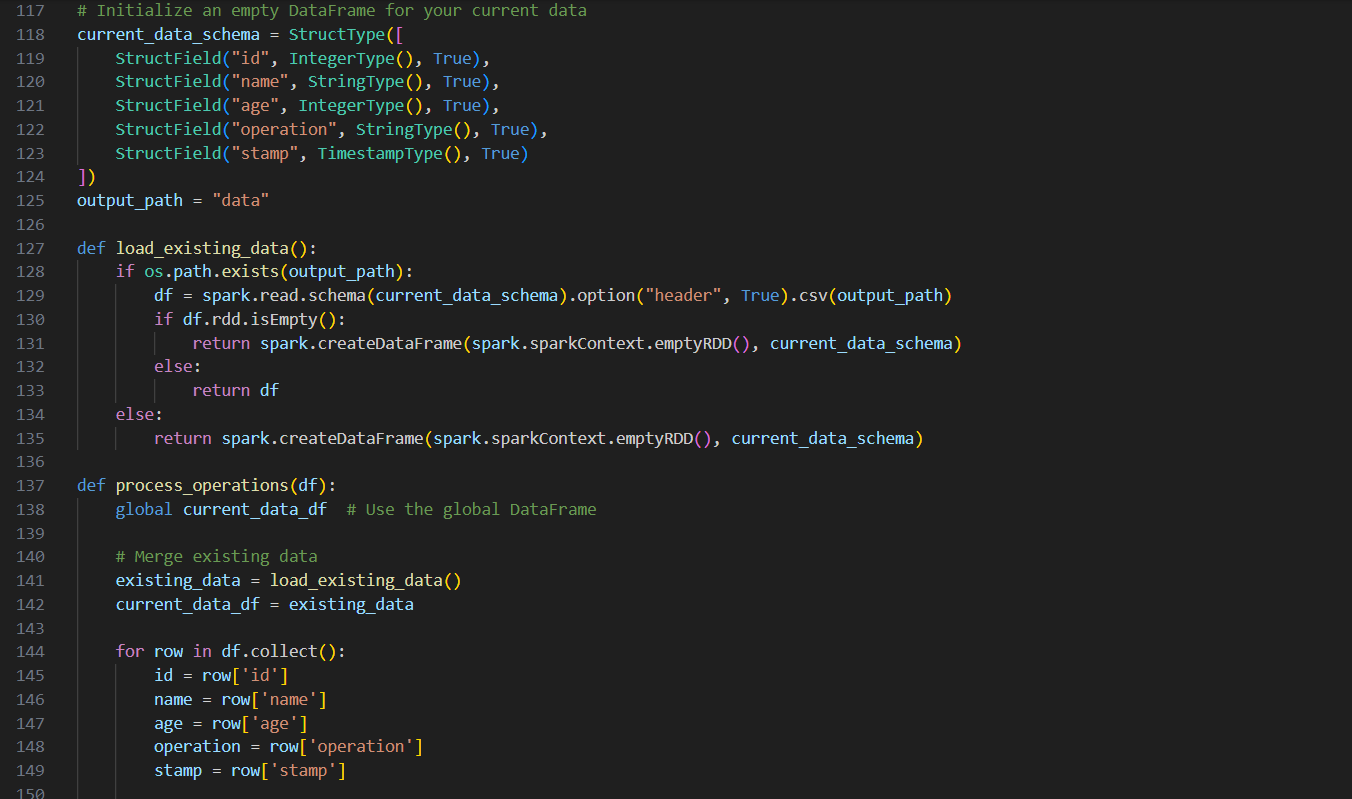
**Glue image also formed in docker:**

****

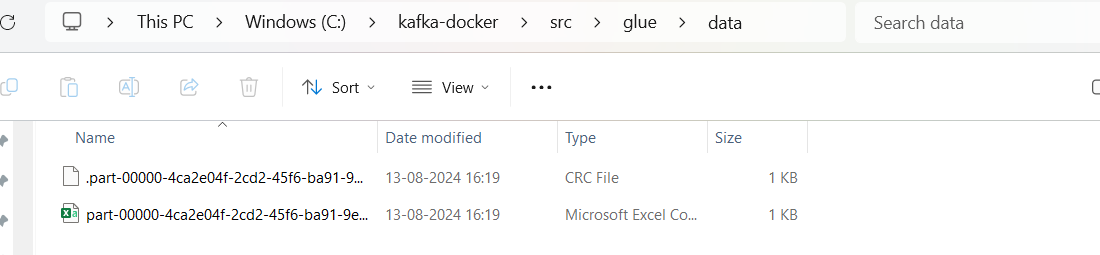
**Step 2:**

* **Made a python script for reading data from topic and put it into csv with the use of following libraries.**
* **This is a PySpark script that reads JSON data from a Kafka topic, processes the data based on the operation specified in the JSON message, and writes the resulting data to a CSV file.**
* **Glue\_file:**

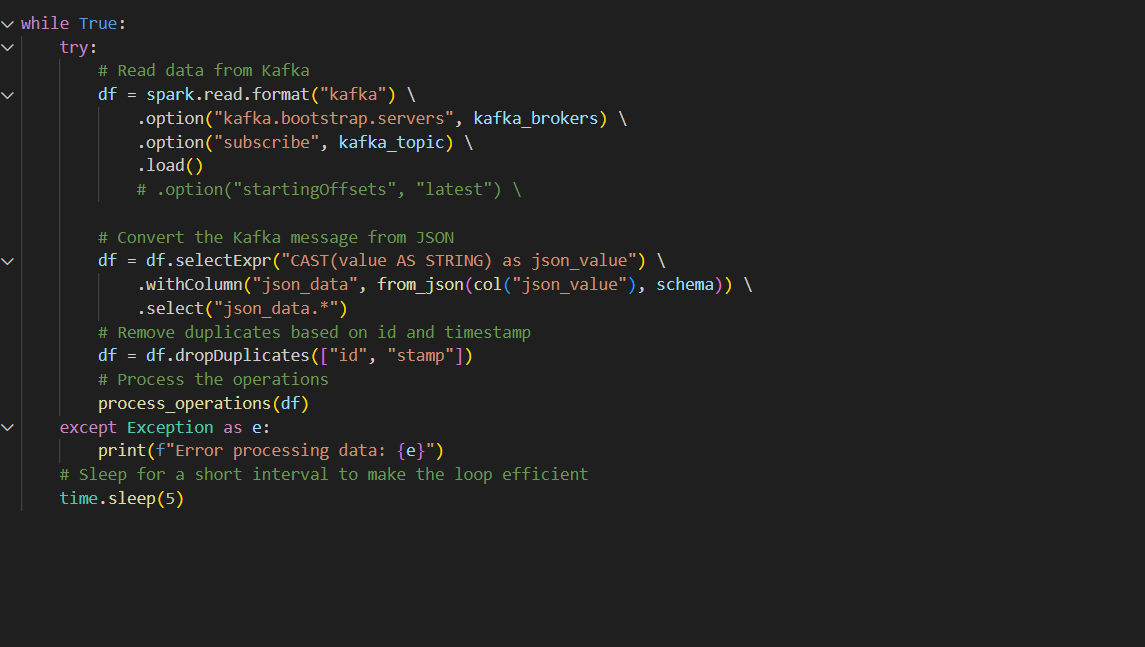
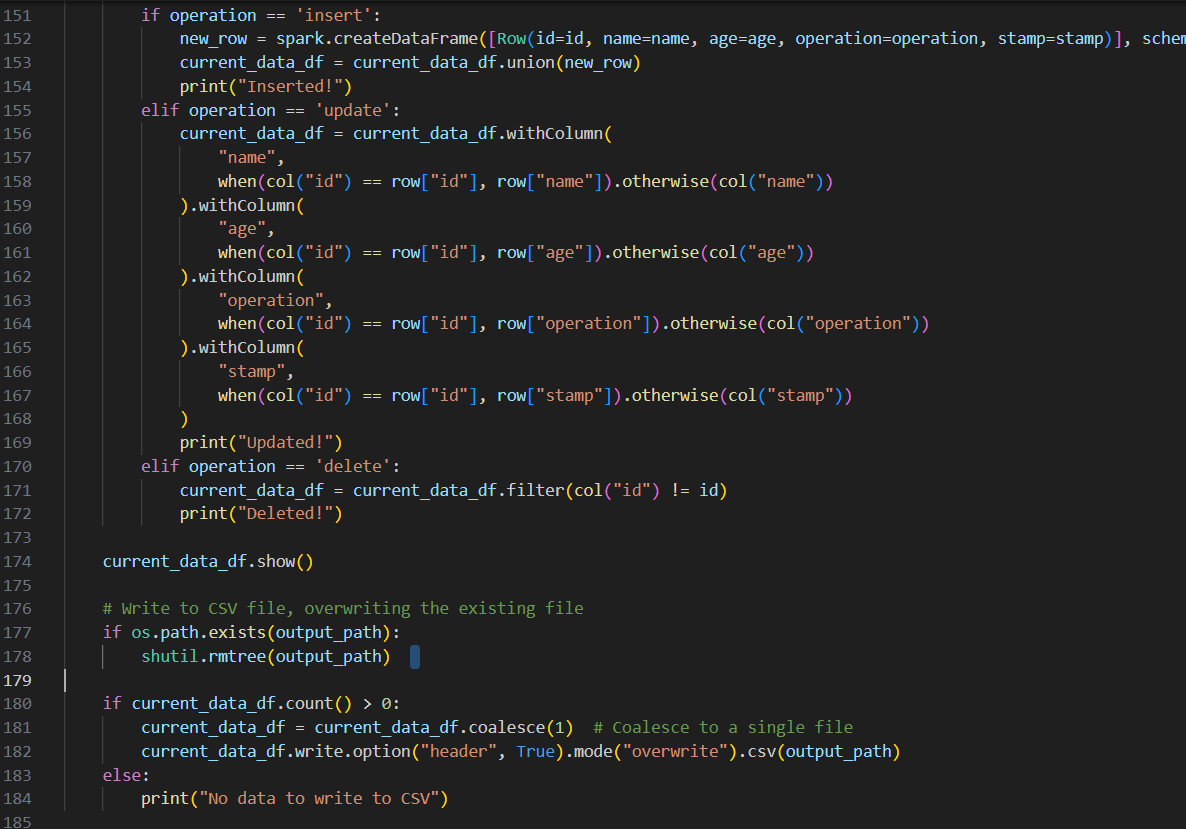
****

****

* **For csv made a separate path in “src/glue/data”:**

****

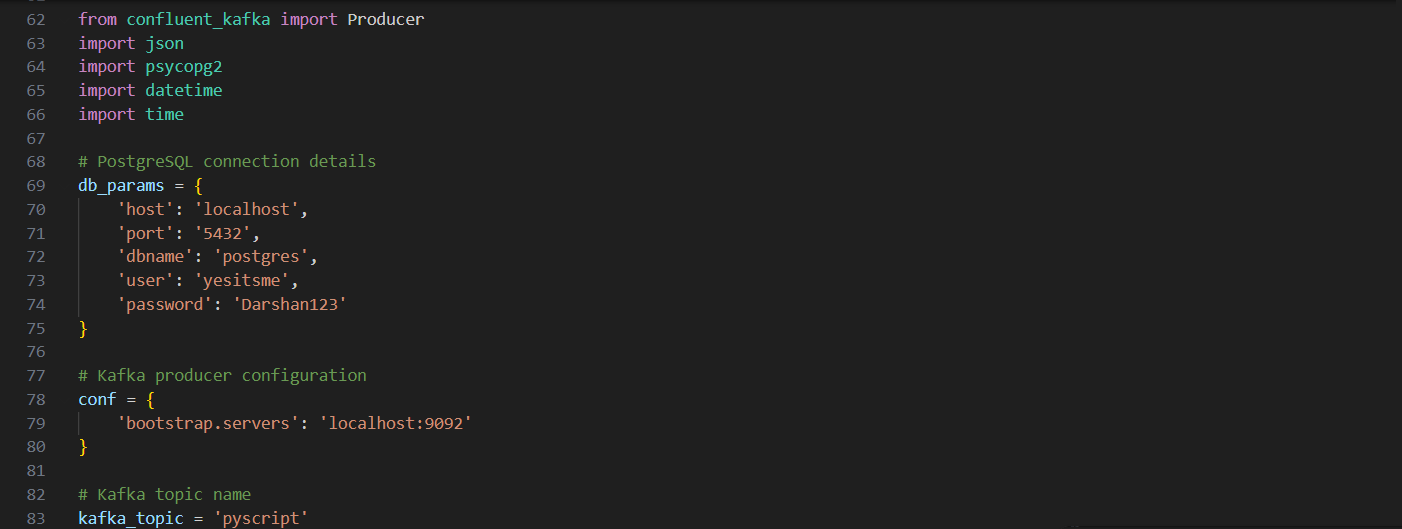
****

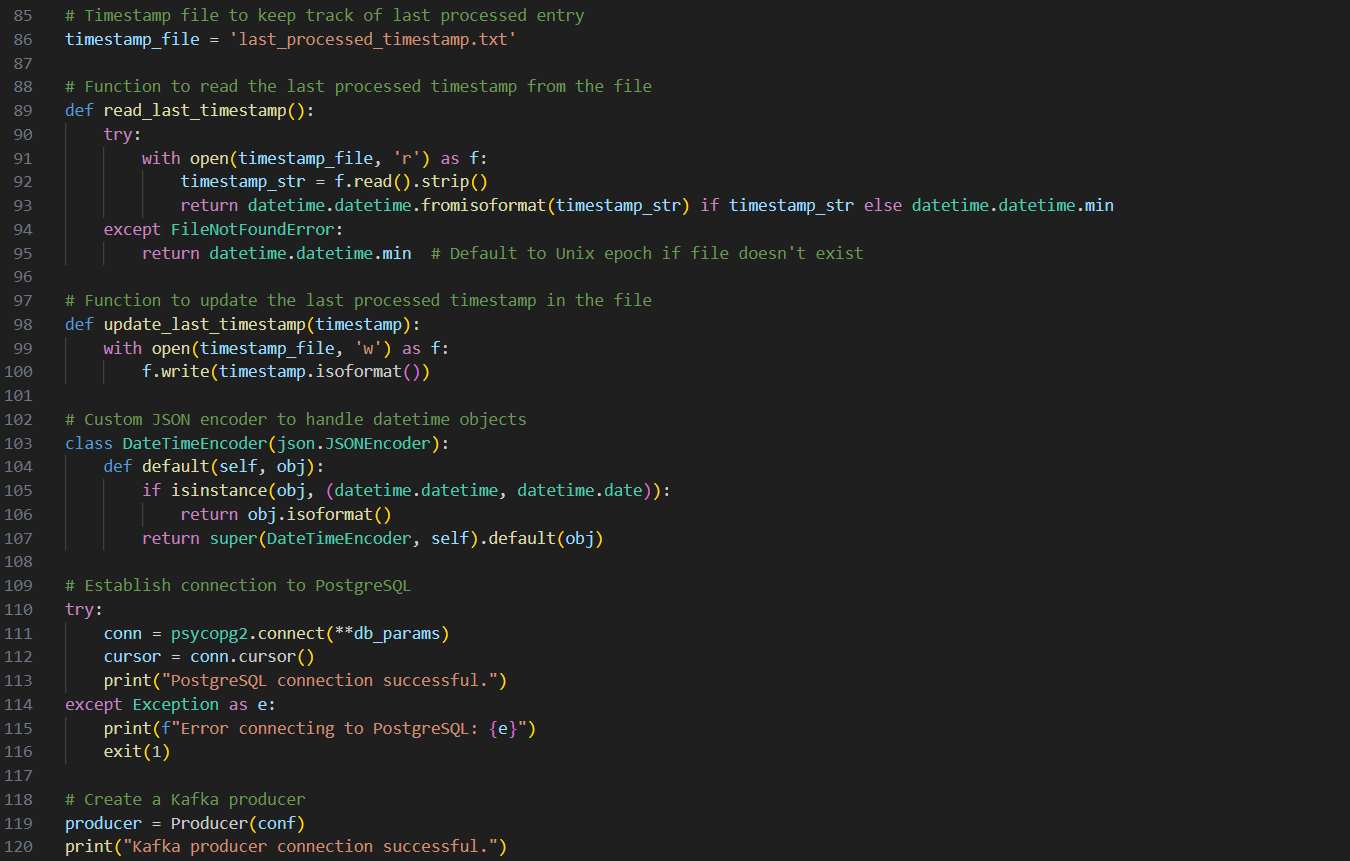
****

* **Used while loop as it will execute until the updated data goes into csv and then if you again updates data in database then it will again execute kafka\_to\_local file automatically.**

**Step 3:**

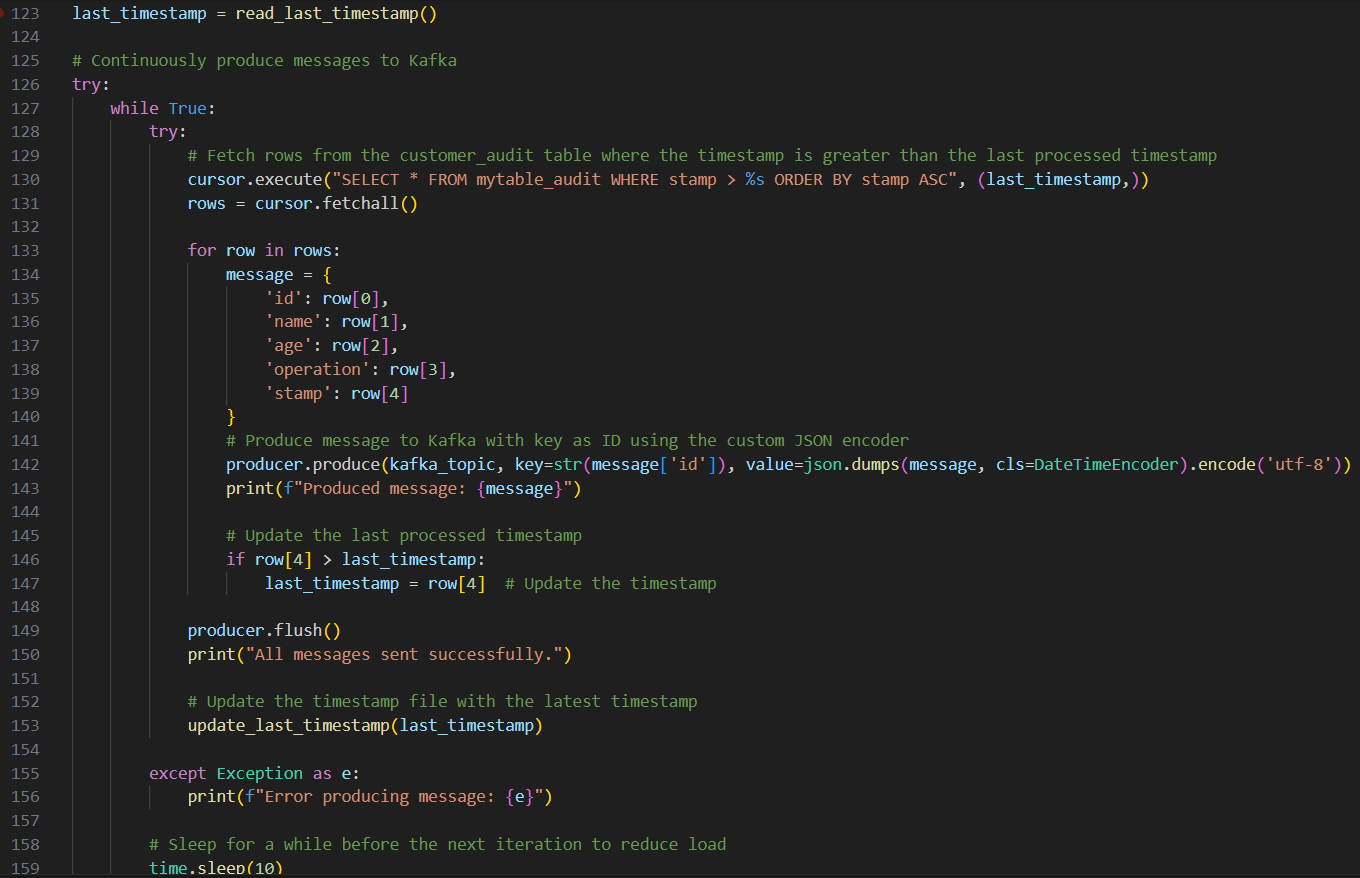
* **Updated procedure file with use of ‘audit’ table just for this task.**

****

****

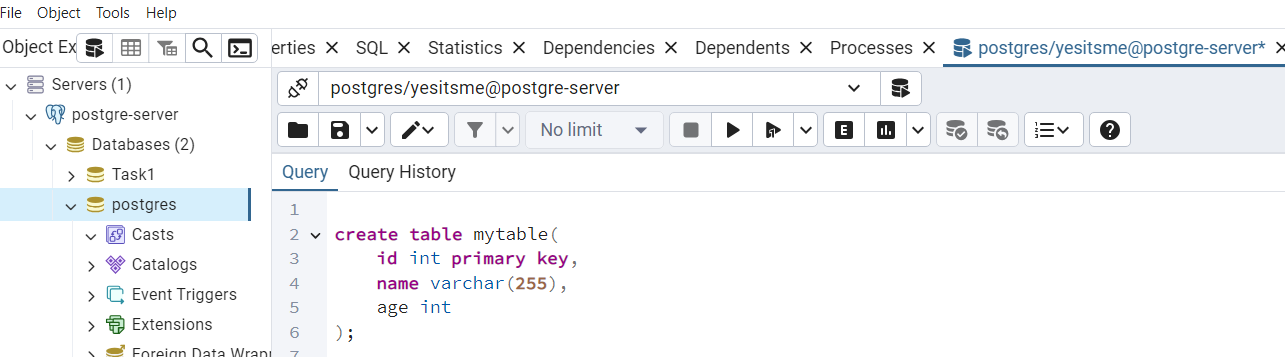
**Will make another file as to showcase the last updated operations time.**

****

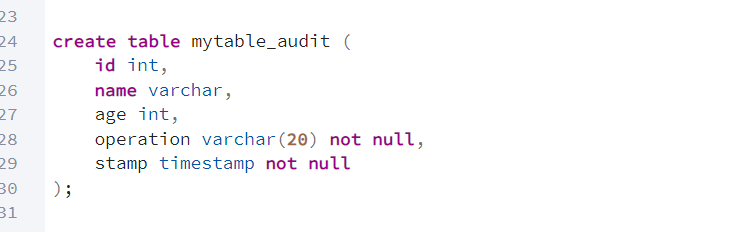
****

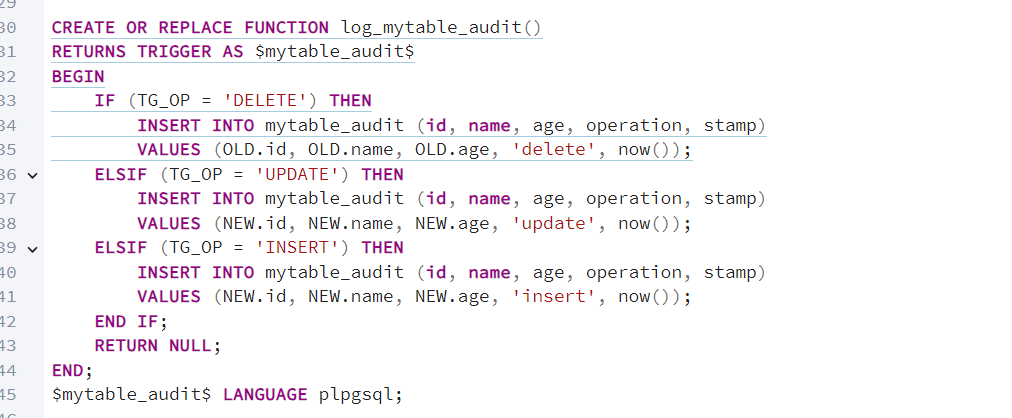
**Step 4:**

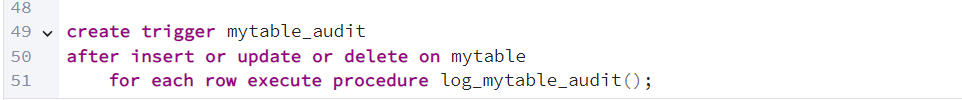
* **As I have used prostgre-server (pgadmin) database**
* **In producer region, I have made a table ‘mytable’ and performed operations on it to check the updated data with timestamp and operation name.**

****

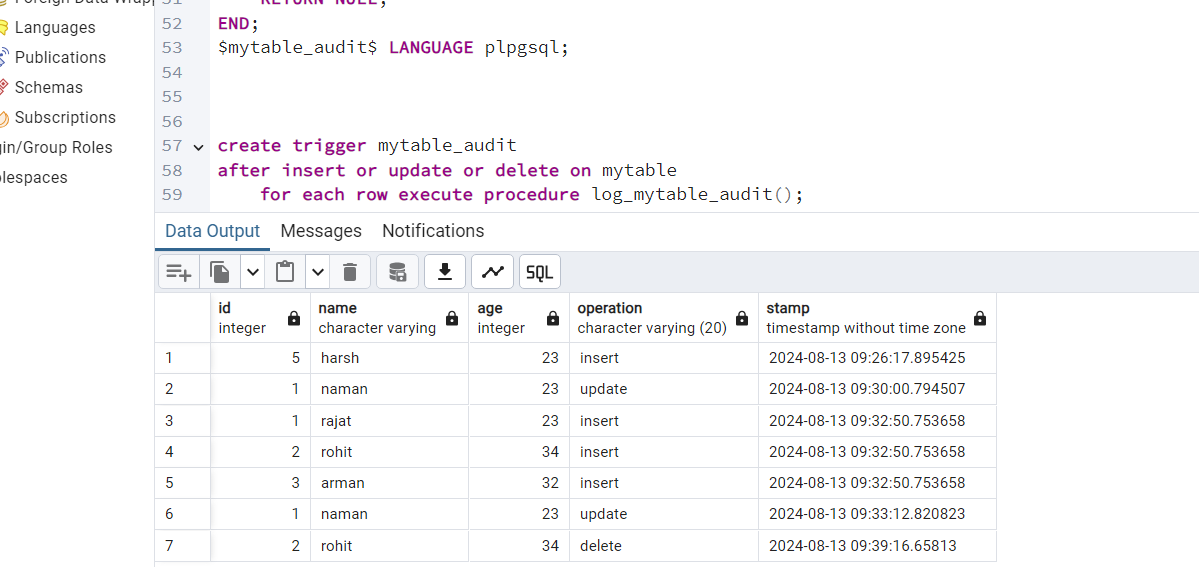
* **For that purpose I have made audit table with same existing schema**

**.**

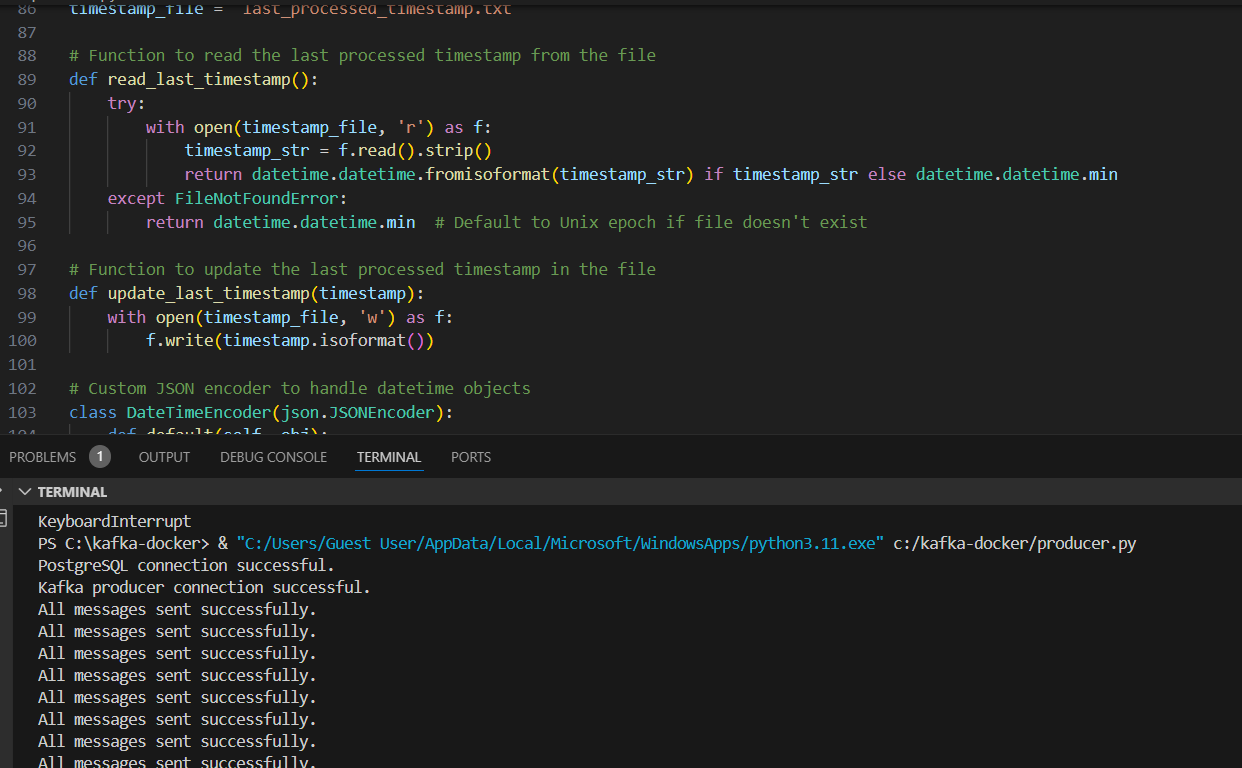
****

****

* **Then made a function for particular operation (insert, update, delete), which captures that and writes down into update audit table to showcase it.**

****

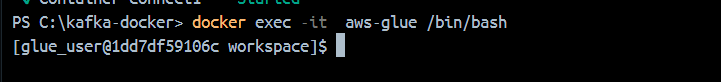
* **For that before and after changes (I have implemented a trigger which takes data of ‘mytable’ and performs function for each row.**
* **Now data into topic ‘pyscript’ went successfully will see the data into updated csv after running the python script for it.**

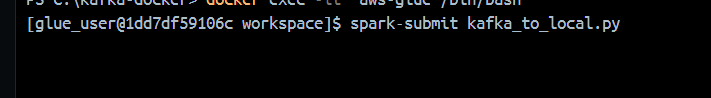
****

* **After using while loop in script we can see automated data operation after changes made from database, it will show live operations in terminal that which are happening live stream.**

**Step 4:**

* **Now go to the shell of aws-glue container image and then run ‘kafka\_to \_local’ file as we created to dump data into csv.**

****

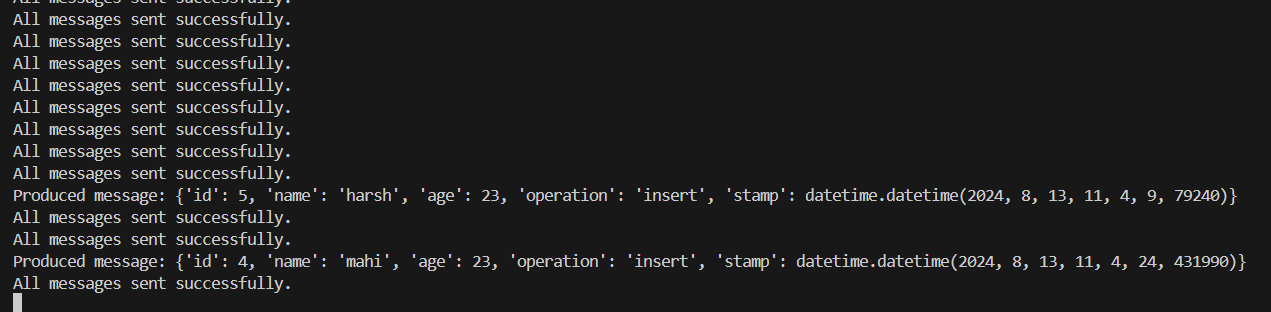
****

**Step 5:**

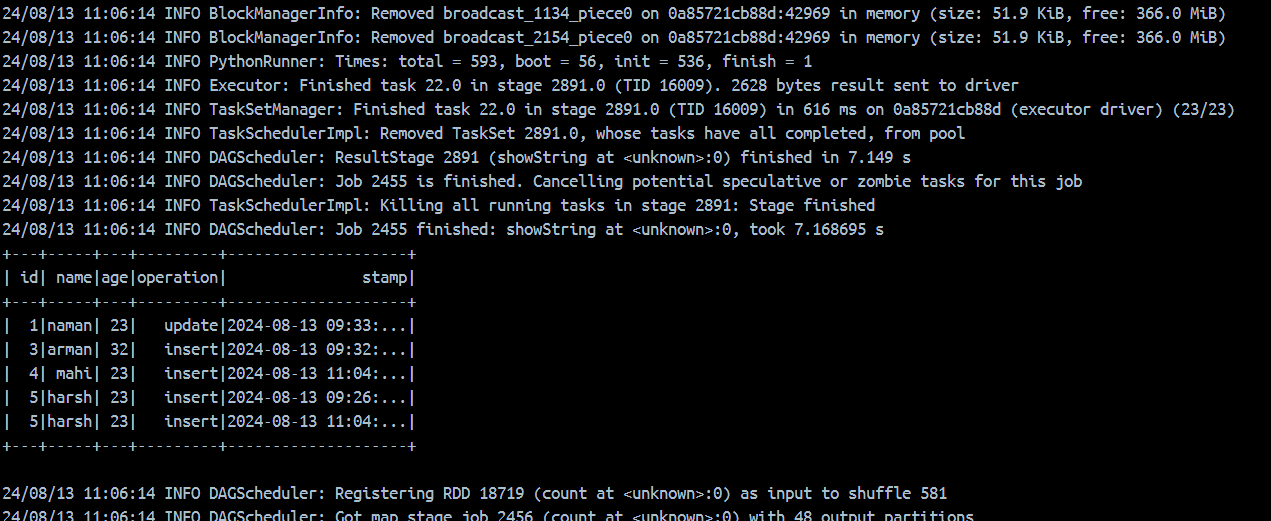
* **Before adding data into csv make sure that your producer file is on status and then execute glue script.**
* **After that every change that made by you will be reflect in csv automatically and it wont stop until you forced it to stop.**

**Lets see:**

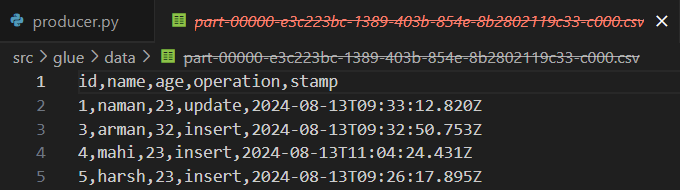
* **Here I made changes in database and producer which is already in running status is showing recent operations.**



* **Now will see docker terminal where glue script is running, it is also showing data updated in dataframe.**

****

* **And the changes made in databases is now reflected into csv after producer and glue terminal.**

****

**Now finally I can see the changes of dsatabase done after particular time into csv.**

**Issues Faced:**

**1.Broker not found (replication factor error as well).**

**Debug: As broker bootstrap services are there**

****

**Where, kafka1:29092 is for inside and localhost:9092 is for client port outside.**

**Just figured out this and solved particulary as,**

**Producer file is your client side to database so localhost:9092 is used there and kafka1:29092 at kafka\_to\_local ( glue file) which is your internal source where you are fetching changes.**

**2. As before error the previous topic was made into another port and broker so made another topic into kafka1:29092 and made changes alternatively into compose file and other files.**

**3. In glue file while making getting dataframe I wasn’t able to get topic data earlier so made changes there as well.**

**4.Continuous updation of changes into dataframe wasn’t showing on terminal so used while loop for that purpose which executes until your target gets fetch.**

**5.In csv there, I was getting all the data rather than the updated recently, so add timestamp to monitor it.**