WORK PROGRESS

By

NANDHINI R

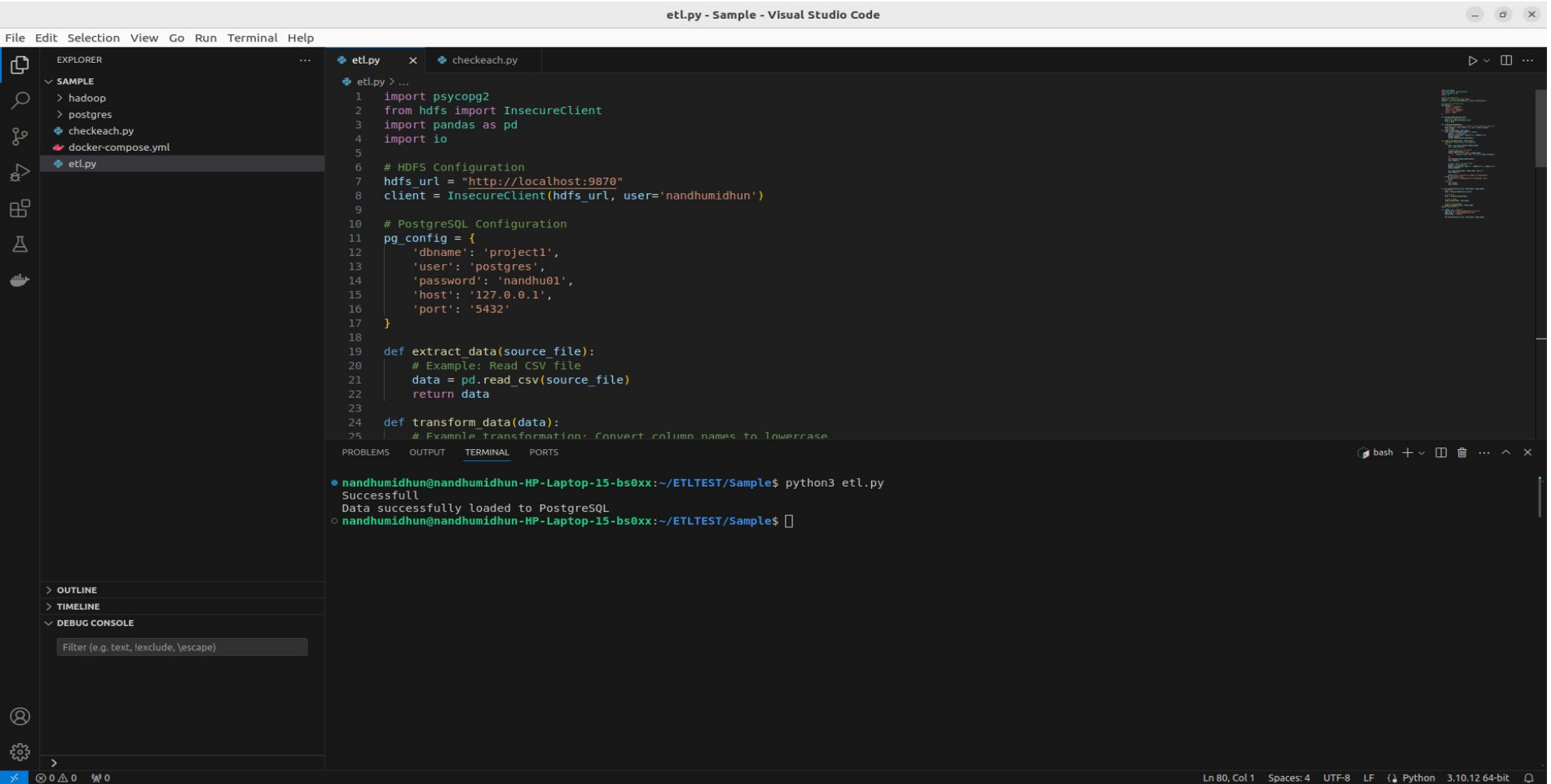
ETL PROCESS

STEP 1: Extract the data from the CSV file

STEP 2: Transform the data [Ex: Convert the Column name into lower Case]

STEP 3: Load the data into Postgresql Database and Hadoop [HDFS]

ETL


etl.py - Sample - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
SAMPLE
hadoop
postgres
checkeach.py
docker-compose.yml
etl.py
etl.py
checkeach.py
etl.py > ...
1 import psycopg2
2 from hdfs import InsecureClient
3 import pandas as pd
4 import io
5
6 # HDFS Configuration
7 hdfs_url = "http://localhost:9870"
8 client = InsecureClient(hdfs_url, user='nandhumidhun')
9
10 # PostgreSQL Configuration
11 pg_config = {
12 'dbname': 'project1',
13 'user': 'postgres',
14 'password': 'nandhu01',
15 'host': '127.0.0.1',
16 'port': '5432'
17 }
18
19 def extract_data(source_file):
20 # Example: Read CSV file
21 data = pd.read_csv(source_file)
22 return data
23
24 def transform_data(data):
25 # Example transformation: Convert column names to lowercase
PROBLEMS OUTPUT TERMINAL PORTS
bash
nandhumidhun@nandhumidhun-HP-Laptop-15-bs0xx:~/ETLTEST/Sample$ python3 etl.py
Successfull
Data successfully loaded to PostgreSQL
nandhumidhun@nandhumidhun-HP-Laptop-15-bs0xx:~/ETLTEST/Sample$
OUTLINE
TIMELINE
DEBUG CONSOLE
Filter (e.g. text, !exclude, \escape)
Ln 80, Col 1 Spaces: 4 UTF-8 LF Python 3.10.12 64-bit

nandhumidhun@nandhumidhun-HP-Laptop-15-bs0xx: ~

nandhumidhun@nandhumidhun-HP-Laptop-15-bs0xx: ~

```
nandhumidhun@nandhumidhun-HP-Laptop-15-bs0xx:~$ hdfs dfs -ls /dockerhadoop
nandhumidhun@nandhumidhun-HP-Laptop-15-bs0xx:~$ psql -h 127.0.0.1 -U postgres -d project1 -p 5432
Password for user postgres:
psql (16.3 (Ubuntu 16.3-1.pgdg22.04+1))
SSL connection (protocol: TLSv1.3, cipher: TLS_AES_256_GCM_SHA384, compression: off)
Type "help" for help.

project1=# SELECT * FROM sample;
 id | name | email
----+------+-------
(0 rows)

project1=# SELECT * FROM sample;
 id |  name  |      email
----+--------+-----------------
  1 | Nandhu | nandhu@gmail.com
  2 | Midhun | midhun@gmail.com
  3 | xxxx   | xx@gmail.com
(3 rows)

project1=# exit;
nandhumidhun@nandhumidhun-HP-Laptop-15-bs0xx:~$ hdfs dfs -ls /dockerhadoop
Found 1 items
-rw-r--r--   1 nandhumidhun supergroup         86 2024-06-20 15:05 /dockerhadoop/file.csv
nandhumidhun@nandhumidhun-HP-Laptop-15-bs0xx:~$ hdfs dfs -cat /dockerhadoop/file.csv
id,name,email
1,Nandhu,nandhu@gmail.com
2,Midhun,midhun@gmail.com
3,xxxx,xx@gmail.com
nandhumidhun@nandhumidhun-HP-Laptop-15-bs0xx:~$
```

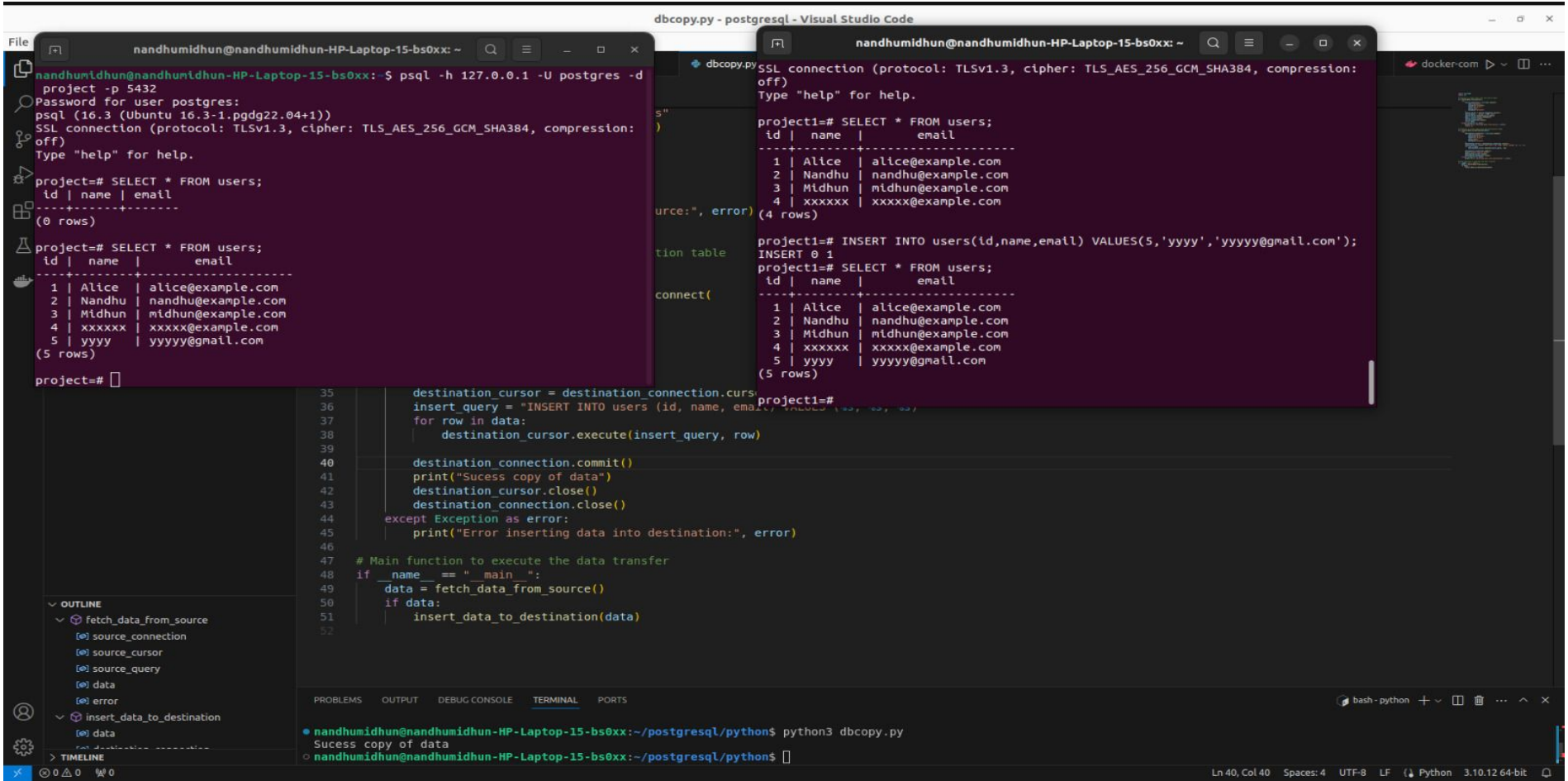
DATA PIPELINE PROCESS

STEP 1: Insert the data from one database into another database

STEP 2: Fetch the data from API URL and insert into database

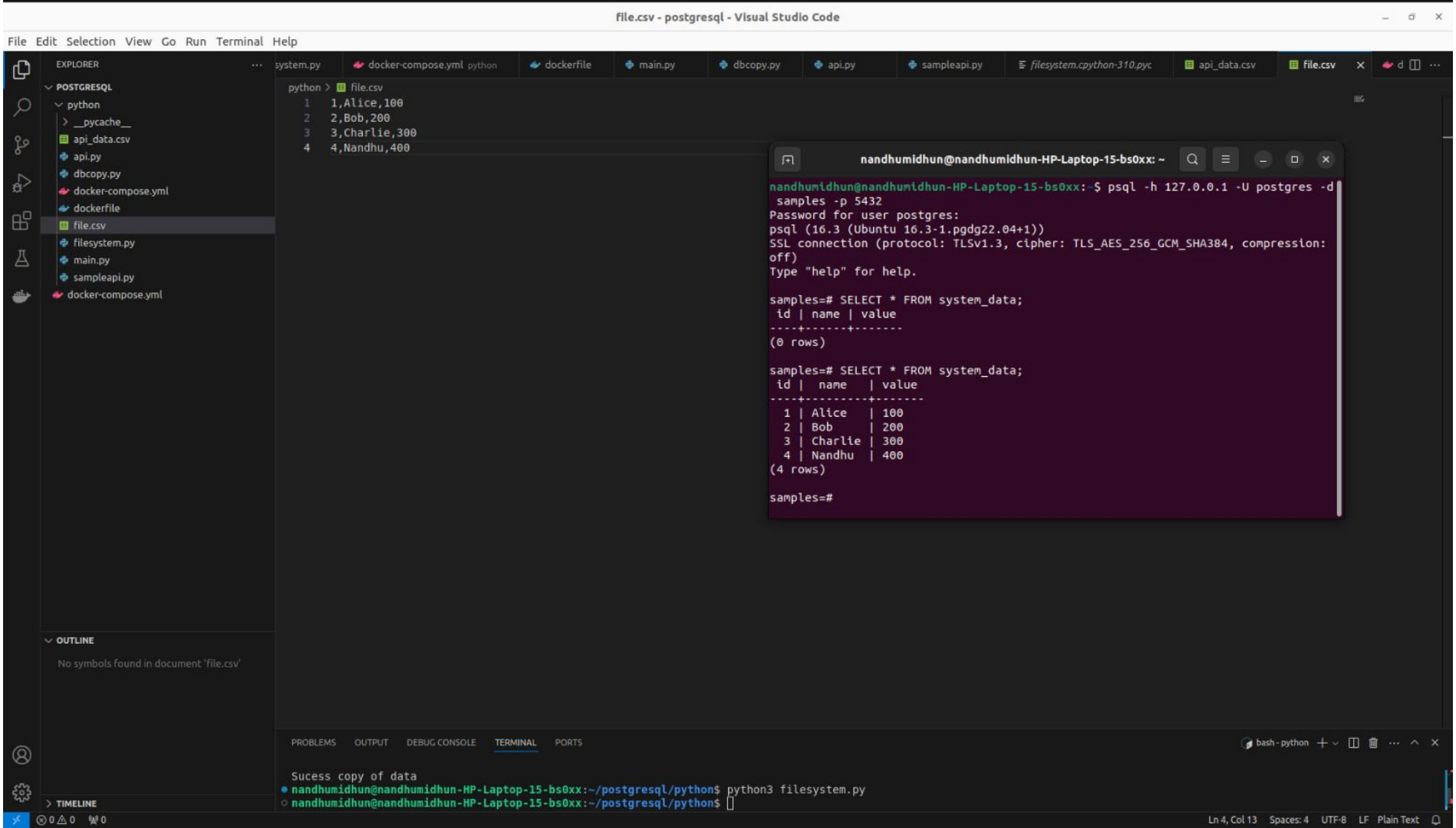
STEP 3: Read the CSV file and insert into the database

DATA PIPELINE



userid	id	title	completed
1	1	delectus aut autem	false
1	2	quis ut nam facilis et officia qui	false
1	3	fugiat veniam minus	false
1	4	et porro tempora	true
1	5	laboriosam mollitia et enim quasi adipisci quia provident illum	false
1	6	qui ullam ratione quibusdam voluptatem quia omnis	false
1	7	illo expedita consequatur quia in	false
1	8	quo adipisci enim quam ut ab	true
1	9	molestiae perspiciatis ipsa	false
1	10	illo est ratione doloremque quia maiores aut	true
1	11	vero rerum temporibus dolor	true
1	12	ipsa repellendus fugit nisi	true
1	13	et doloremque nulla	false
1	14	repellendus sunt dolores architecto voluptatum	true
1	15	ab voluptatum amet voluptas	true
1	16	accusamus eos facilis sint et aut voluptatem	true
1	17	quo laboriosam deleniti aut qui	true
1	18	dolorum est consequatur ea mollitia in culpa	false
1	19	molestiae ipsa aut voluptatibus pariatur dolor nihil	true
1	20	ullam nobis libero sapiente ad optio sint	true
2	21	suscipit repellat esse quibusdam voluptatem incidunt	false
2	22	distinctio vitae autem nihil ut molestias quo	true
2	23	et itaque necessitatibus maxime molestiae qui quas velit	false
2	24	adipisci non ad dicta qui amet quaerat doloribus ea	false
2	25	voluptas quo tenetur perspiciatis explicabo natus	true
2	26	aliquam aut quasi	true
2	27	veritatis pariatur delectus	true
2	28	nesciunt totam sit blanditiis sit	false
2	29	laborum aut in quam	false
2	30	nemo perspiciatis repellat ut dolor libero commodi blanditiis omnis	true
2	31	repudiandae totam in est sint facere fuga	false
2	32	earum doloribus ea doloremque quis	false
2	33	sint sit aut vero	false
2	34	porro aut necessitatibus eaque distinctio	false
2	35	repellendus veritatis molestias dicta incidunt	true
2	36	excepturi deleniti adipisci voluptatem et neque optio illum ad	true
2	37	sunt cum tempora	false
2	38	totam quia non	false
2	39	doloremque quibusdam asperiores libero corrupti illum qui omnis	false
2	40	totam atque quo nesciunt	true
3	41	aliquid amet impedit consequatur aspernatur placeat eaque fugiat suscipit	false
3	42	rerum perferendis error quia ut eveniet	false
3	43	tempore ut sint quis recusandae	true
3	44	cum debitis quis accusamus doloremque ipsa natus sapiente omnis	true
3	45	velit soluta adipisci molestias reiciendis harum	false
3	46	vel voluptatem repellat nihil placeat corporis	false
3	47	nam qui rerum fugiat accusamus	false
3	48	sit reprehenderit omnis quia	false
3	49	ut necessitatibus aut maiores debitis officia blanditiis velit et	false
3	50	cupiditate necessitatibus ullam aut quis dolor voluptate	true

:


file.csv - postgresql - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
POSTGRESQL
python
__pycache__
api_data.csv
api.py
dbcopy.py
docker-compose.yml
dockerfile
file.csv
filesystem.py
main.py
sampleapi.py
docker-compose.yml
OUTLINE
No symbols found in document 'file.csv'
TIMELINE
system.py
docker-compose.yml python
dockerfile
main.py
dbcopy.py
api.py
sampleapi.py
filesystem.cpython-310.pyc
api_data.csv
file.csv
python > file.csv
1,Alice,100
2,Bob,200
3,Charlie,300
4,Nandhu,400
nandhumidhun@nandhumidhun-HP-Laptop-15-bs0xx: ~
nandhumidhun@nandhumidhun-HP-Laptop-15-bs0xx:~$ psql -h 127.0.0.1 -U postgres -d samples -p 5432
Password for user postgres:
psql (16.3 (Ubuntu 16.3-1.pgdg22.04+1))
SSL connection (protocol: TLSv1.3, cipher: TLS_AES_256_GCM_SHA384, compression: off)
Type "help" for help.
samples=# SELECT * FROM system_data;
id | name | value
(0 rows)
samples=# SELECT * FROM system_data;
id | name | value
1 | Alice | 100
2 | Bob | 200
3 | Charlie | 300
4 | Nandhu | 400
(4 rows)
samples=#
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
bash - python
Sucess copy of data
nandhumidhun@nandhumidhun-HP-Laptop-15-bs0xx:~/postgresql/python$ python3 filesystem.py
nandhumidhun@nandhumidhun-HP-Laptop-15-bs0xx:~/postgresql/python$
Ln 4, Col 13 Spaces: 4 UTF-8 LF Plain Text

SALES DATA ANALYSIS WITH HADOOP AND HIVE PROCESS

STEP 1: Load the sales data into HDFS

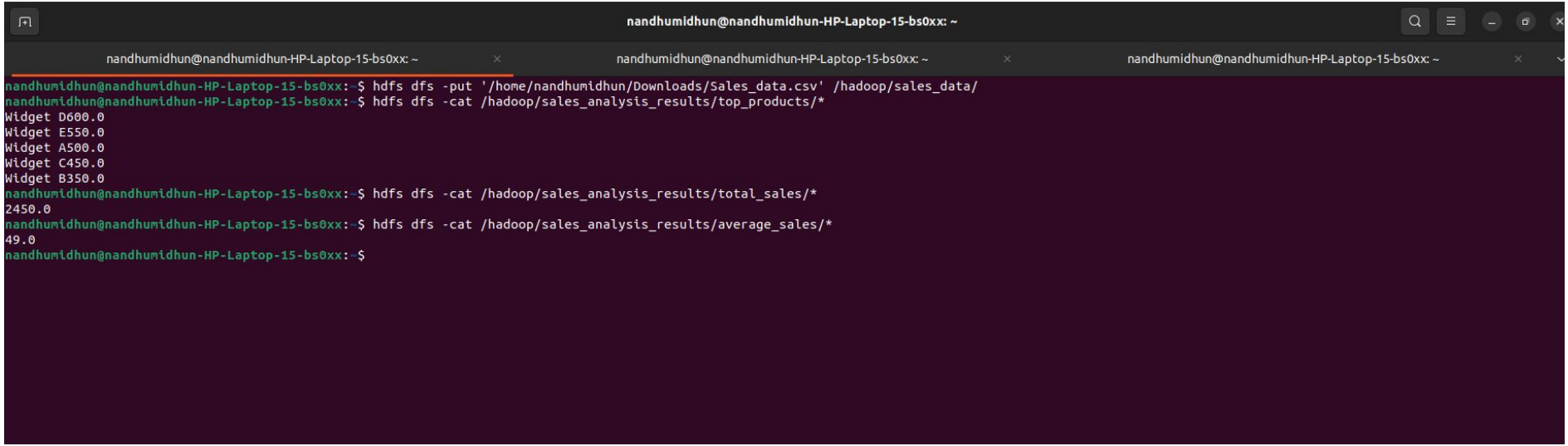
STEP 2: Create an external Hive table to map the CSV file

STEP 3: Load the data into the Hive table.

STEP 4: Save the query results to HDFS for further processing or reporting.

STEP 5: Optionally, clean up the Hive table and HDFS files if no longer needed.

SALES DATA ANALYSIS WITH HADOOP AND HIVE



```
No rows affected (0.062 seconds)
0: jdbc:hive2://localhost:10000> SELECT * FROM sales_data;
INFO  : Compiling command(queryId=nandhumidhun_20240628165628_960c068b-55ba-4bf1-ba41-35fa3d530891): SELECT * FROM sales_data
INFO  : No Stats for hadoop@sales_data, Columns: sale_id, sale_date, product_id, customer_id, product_name, sale_amount
INFO  : Semantic Analysis Completed (retrial = false)
INFO  : Created Hive schema: Schema(fieldSchemas:[FieldSchema(name:sales_data.sale_id, type:int, comment:null), FieldSchema(name:sales_data.product_id, type:int, comment:null), FieldSchema(name:sales_data
.product_name, type:string, comment:null), FieldSchema(name:sales_data.sale_date, type:string, comment:null), FieldSchema(name:sales_data.sale_amount, type:double, comment:null), FieldSchema(name:sales_da
ta.customer_id, type:int, comment:null)], properties:null)
INFO  : Completed compiling command(queryId=nandhumidhun_20240628165628_960c068b-55ba-4bf1-ba41-35fa3d530891); Time taken: 0.169 seconds
INFO  : Concurrency mode is disabled, not creating a lock manager
INFO  : Executing command(queryId=nandhumidhun_20240628165628_960c068b-55ba-4bf1-ba41-35fa3d530891): SELECT * FROM sales_data
INFO  : Completed executing command(queryId=nandhumidhun_20240628165628_960c068b-55ba-4bf1-ba41-35fa3d530891); Time taken: 0.0 seconds
```

sales_data.sale_id	sales_data.product_id	sales_data.product_name	sales_data.sale_date	sales_data.sale_amount	sales_data.customer_id
1	101	Widget A	2024-01-01	50.0	1001
2	102	Widget B	2024-01-02	35.0	1002
3	103	Widget C	2024-01-03	45.0	1003
4	104	Widget D	2024-01-04	60.0	1004
5	105	Widget E	2024-01-05	55.0	1005
6	101	Widget A	2024-01-06	50.0	1006
7	102	Widget B	2024-01-07	35.0	1007
8	103	Widget C	2024-01-08	45.0	1008
9	104	Widget D	2024-01-09	60.0	1009
10	105	Widget E	2024-01-10	55.0	1010
11	101	Widget A	2024-01-11	50.0	1011
12	102	Widget B	2024-01-12	35.0	1012
13	103	Widget C	2024-01-13	45.0	1013
14	104	Widget D	2024-01-14	60.0	1014
15	105	Widget E	2024-01-15	55.0	1015
16	101	Widget A	2024-01-16	50.0	1016
17	102	Widget B	2024-01-17	35.0	1017
18	103	Widget C	2024-01-18	45.0	1018
19	104	Widget D	2024-01-19	60.0	1019
20	105	Widget E	2024-01-20	55.0	1020
21	101	Widget A	2024-01-21	50.0	1021
22	102	Widget B	2024-01-22	35.0	1022
23	103	Widget C	2024-01-23	45.0	1023
24	104	Widget D	2024-01-24	60.0	1024
25	105	Widget E	2024-01-25	55.0	1025
26	101	Widget A	2024-01-26	50.0	1026
27	102	Widget B	2024-01-27	35.0	1027
28	103	Widget C	2024-01-28	45.0	1028
29	104	Widget D	2024-01-29	60.0	1029
30	105	Widget E	2024-01-30	55.0	1030
31	101	Widget A	2024-01-31	50.0	1031
32	102	Widget B	2024-02-01	35.0	1032
33	103	Widget C	2024-02-02	45.0	1033
34	104	Widget D	2024-02-03	60.0	1034
35	105	Widget E	2024-02-04	55.0	1035
36	101	Widget A	2024-02-05	50.0	1036
37	102	Widget B	2024-02-06	35.0	1037
38	103	Widget C	2024-02-07	45.0	1038
39	104	Widget D	2024-02-08	60.0	1039

```
INFO  : Job running in-process (local Hadoop)
INFO  : 2024-06-28 16:57:26,211 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 0.21 sec
INFO  : MapReduce Total cumulative CPU time: 210 msec
INFO  : Ended Job = job_local822845722_0010
INFO  : MapReduce Jobs Launched:
INFO  : Stage-Stage-1:  Cumulative CPU: 0.21 sec   HDFS Read: 46724 HDFS Write: 162 HDFS EC Read: 0 SUCCESS
INFO  : Total MapReduce CPU Time Spent: 210 msec
INFO  : Completed executing command(queryId=nandhumidhun_20240628165724_62740a7a-072e-45d9-91c5-932865291d2e); Time taken: 1.403 seconds
+--------------+
| total_sales  |
+--------------+
| 2450.0       |
+--------------+
1 row selected (1.601 seconds)
0: jdbc:hive2://localhost:10000> SELECT AVG(sale_amount) AS average_sales FROM sales_data;
INFO  : Compiling command(queryId=nandhumidhun_20240628165745_67b656ee-f617-4099-b59a-edeb0997b4ce): SELECT AVG(sale_amount) AS average_sales FROM sales_data
INFO  : No Stats for hadoop@sales_data, Columns: sale_amount
INFO  : Semantic Analysis Completed (retrial = false)
INFO  : Created Hive schema: Schema(fieldSchemas:[FieldSchema(name:average_sales, type:double, comment:null)], properties:null)
INFO  : Completed compiling command(queryId=nandhumidhun_20240628165745_67b656ee-f617-4099-b59a-edeb0997b4ce); Time taken: 0.194 seconds
INFO  : Concurrency mode is disabled, not creating a lock manager
INFO  : Executing command(queryId=nandhumidhun_20240628165745_67b656ee-f617-4099-b59a-edeb0997b4ce): SELECT AVG(sale_amount) AS average_sales FROM sales_data
WARN  : Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. tez) or using Hive 1.X releases.
INFO  : Query ID = nandhumidhun_20240628165745_67b656ee-f617-4099-b59a-edeb0997b4ce
INFO  : Total jobs = 1
INFO  : Launching Job 1 out of 1
INFO  : Starting task [Stage-1:MAPRED] in serial mode
INFO  : Number of reduce tasks determined at compile time: 1
INFO  : In order to change the average load for a reducer (in bytes):
INFO  :   set hive.exec.reducers.bytes.per.reducer=<number>
INFO  : In order to limit the maximum number of reducers:
INFO  :   set hive.exec.reducers.max=<number>
INFO  : In order to set a constant number of reducers:
INFO  :   set mapreduce.job.reduces=<number>
INFO  : number of splits:1
INFO  : Submitting tokens for job: job_local1785308139_0011
INFO  : Executing with tokens: []
INFO  : The url to track the job: http://localhost:8080/
INFO  : Job running in-process (local Hadoop)
INFO  : 2024-06-28 16:57:47,173 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 0.19 sec
INFO  : MapReduce Total cumulative CPU time: 190 msec
INFO  : Ended Job = job_local1785308139_0011
INFO  : MapReduce Jobs Launched:
INFO  : Stage-Stage-1:  Cumulative CPU: 0.19 sec   HDFS Read: 50506 HDFS Write: 162 HDFS EC Read: 0 SUCCESS
INFO  : Total MapReduce CPU Time Spent: 190 msec
INFO  : Completed executing command(queryId=nandhumidhun_20240628165745_67b656ee-f617-4099-b59a-edeb0997b4ce); Time taken: 1.421 seconds
+----------------+
| average_sales  |
+----------------+
| 49.0           |
+----------------+
1 row selected (1.653 seconds)
0: jdbc:hive2://localhost:10000>
```

```
WARN  : Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. tez) or using Hive 1.X releases.
INFO  : Query ID = nandhumidhun_20240628165858_d9e8389c-ae92-4da7-a911-4104db755d21
INFO  : Total jobs = 2
INFO  : Launching Job 1 out of 2
INFO  : Starting task [Stage-1:MAPRED] in serial mode
INFO  : Number of reduce tasks not specified. Estimated from input data size: 1
INFO  : In order to change the average load for a reducer (in bytes):
INFO  :   set hive.exec.reducers.bytes.per.reducer=<number>
INFO  : In order to limit the maximum number of reducers:
INFO  :   set hive.exec.reducers.max=<number>
INFO  : In order to set a constant number of reducers:
INFO  :   set mapreduce.job.reduces=<number>
INFO  : number of splits:1
INFO  : Submitting tokens for job: job_local1275533349_0012
INFO  : Executing with tokens: []
INFO  : The url to track the job: http://localhost:8080/
INFO  : Job running in-process (local Hadoop)
INFO  : 2024-06-28 16:58:59,747 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 0.21 sec
INFO  : MapReduce Total cumulative CPU time: 210 msec
INFO  : Ended Job = job_local1275533349_0012
INFO  : Launching Job 2 out of 2
INFO  : Starting task [Stage-2:MAPRED] in serial mode
INFO  : Number of reduce tasks determined at compile time: 1
INFO  : In order to change the average load for a reducer (in bytes):
INFO  :   set hive.exec.reducers.bytes.per.reducer=<number>
INFO  : In order to limit the maximum number of reducers:
INFO  :   set hive.exec.reducers.max=<number>
INFO  : In order to set a constant number of reducers:
INFO  :   set mapreduce.job.reduces=<number>
INFO  : number of splits:1
INFO  : Submitting tokens for job: job_local1032948224_0013
INFO  : Executing with tokens: []
INFO  : The url to track the job: http://localhost:8080/
INFO  : Job running in-process (local Hadoop)
INFO  : 2024-06-28 16:59:01,193 Stage-2 map = 100%,  reduce = 100%, Cumulative CPU 0.12 sec
INFO  : MapReduce Total cumulative CPU time: 120 msec
INFO  : Ended Job = job_local1032948224_0013
INFO  : MapReduce Jobs Launched:
INFO  : Stage-Stage-1:  Cumulative CPU: 0.21 sec   HDFS Read: 54288 HDFS Write: 162 HDFS EC Read: 0 SUCCESS
INFO  : Stage-Stage-2:  Cumulative CPU: 0.12 sec   HDFS Read: 54288 HDFS Write: 162 HDFS EC Read: 0 SUCCESS
INFO  : Total MapReduce CPU Time Spent: 330 msec
INFO  : Completed executing command(queryId=nandhumidhun_20240628165858_d9e8389c-ae92-4da7-a911-4104db755d21); Time taken: 2.839 seconds
+--------------+--------------+
| product_name  | total_sales  |
+--------------+--------------+
| Widget D      | 600.0        |
| Widget E      | 550.0        |
| Widget A      | 500.0        |
| Widget C      | 450.0        |
| Widget B      | 350.0        |
+--------------+--------------+
5 rows selected (3.052 seconds)
0: jdbc:hive2://localhost:10000>
```

DATA ANALYTICS PLATFORM PROCESS

STEP 1: Upload Data to HDFS

STEP 2: Create Hive Tables

STEP 3: Load Data into Hive Tables

STEP 4: Join Tables to Create a Unified View

STEP 5: Sample Queries for Analytics

STEP 6: Data Visualization

DATA ANALYTICS PLATFORM

```
nandhumidhun@nandhumidhun-HP-Laptop-15-bs0xx:~$ hdfs dfs -cat /output/*
Category,avg(Discount)
Furniture,0.1
Office Supplies,0.05
Technology,0.09999999999999999
nandhumidhun@nandhumidhun-HP-Laptop-15-bs0xx:~$ hdfs dfs -cat /output1/*
ProductID,Category,Discount,Amount,Profit,total_amount
P001,Office Supplies,0.05,100.5,20.25,2035.125
P002,Technology,0.1,1500.0,300.0,450000.0
P003,Furniture,0.08,250.75,50.15,12575.1125
P004,Office Supplies,0.02,20.25,4.5,91.125
P005,Technology,0.15,1200.0,180.0,216000.0
P006,Office Supplies,0.05,50.8,10.2,518.16
P007,Office Supplies,0.03,15.5,2.1,32.550000000000004
P008,Furniture,0.12,900.0,120.0,108000.0
P009,Technology,0.05,35.0,5.25,183.75
P010,Office Supplies,0.1,200.0,30.0,6000.0
```

```
INFO  : Compiling command(queryId=nandhumidhun_20240629145743_5835f7f2-ee31-4720-95d2-ee2771c4ad3b): SELECT * FROM users
INFO  : No Stats for hadoop@users, Columns: user_id, name, location, age
INFO  : Semantic Analysis Completed (retrial = false)
INFO  : Created Hive schema: Schema(fieldSchemas:[FieldSchema(name:users.user_id, type:int, comment:null), FieldSchema(name:users.name, type:string, comment:null), FieldSchema(name:users.age, type:int, comment:null), FieldSchema(name:users.location, type:string, comment:null)], properties:null)
INFO  : Completed compiling command(queryId=nandhumidhun_20240629145743_5835f7f2-ee31-4720-95d2-ee2771c4ad3b); Time taken: 0.112 seconds
INFO  : Concurrency mode is disabled, not creating a lock manager
INFO  : Executing command(queryId=nandhumidhun_20240629145743_5835f7f2-ee31-4720-95d2-ee2771c4ad3b): SELECT * FROM users
INFO  : Completed executing command(queryId=nandhumidhun_20240629145743_5835f7f2-ee31-4720-95d2-ee2771c4ad3b); Time taken: 0.0 seconds
+----------------+-------------+------------+-----------------+
| users.user_id  | users.name  | users.age  | users.location  |
+----------------+-------------+------------+-----------------+
| 1              | User A      | 40         | California      |
| 2              | User B      | 55         | Washington      |
| 3              | User C      | 29         | Washington      |
| 4              | User D      | 33         | California      |
| 5              | User E      | 38         | New Jersey      |
| 6              | User F      | 22         | Texas           |
| 7              | User G      | 65         | California      |
| 8              | User H      | 43         | California      |
| 9              | User I      | 44         | Florida         |
| 10             | User J      | 36         | Florida         |
| 11             | User K      | 19         | Washington      |
| 12             | User L      | 19         | New Jersey      |
| 13             | User M      | 40         | Washington      |
| 14             | User N      | 52         | California      |
| 15             | User O      | 24         | California      |
| 16             | User P      | 63         | New York        |
| 17             | User Q      | 46         | California      |
| 18             | User R      | 31         | California      |
| 19             | User S      | 59         | New Jersey      |
| 20             | User T      | 52         | California      |
| 21             | User U      | 33         | Florida         |
| 22             | User V      | 44         | Florida         |
| 23             | User W      | 35         | Washington      |
| 24             | User X      | 42         | Texas           |
| 25             | User Y      | 33         | Texas           |
| 26             | User Z      | 64         | New Jersey      |
| 27             | User [      | 42         | New York        |
| 28             | User \      | 55         | Florida         |
| 29             | User ]      | 66         | Texas           |
| 30             | User ^      | 40         | Washington      |
| 31             | User _      | 52         | New Jersey      |
| 32             | User `      | 60         | Florida         |
| 33             | User a      | 29         | California      |
| 34             | User b      | 52         | Florida         |
| 35             | User c      | 24         | New York        |
| 36             | User d      | 40         | Texas           |
| 37             | User e      | 23         | Washington      |
| 38             | User f      | 50         | California      |
| 39             | User g      | 62         | New Jersey      |
| 40             | User h      | 58         | New Jersey      |
| 41             | User i      | 68         | Florida         |
| 42             | User j      | 64         | Florida         |
```

```
nandhumidhun@nandhumidhun-HP-Laptop-15-bs0xx: $ psql -h 127.0.0.1 -p 5432 -U postgres -d project
Password for user postgres:
psql (16.3 (Ubuntu 16.3-1.pgdg22.04+1))
SSL connection (protocol: TLSv1.3, cipher: TLS_AES_256_GCM_SHA384, compression: off)
Type "help" for help.

project=# SELECT * FROM transactional_data;
 transaction_id | product_id | user_id | quantity | price |      timestamp
----------------+------------+---------+----------+-------+---------------------
              1 |        144 |      48 |        2 | 47.03 | 2023-06-01 11:00:00
              2 |        115 |      43 |        5 | 40.47 | 2023-06-01 12:00:00
              3 |        140 |       5 |        5 | 41.64 | 2023-06-01 13:00:00
              4 |        101 |      28 |        3 | 46.53 | 2023-06-01 14:00:00
              5 |        130 |       3 |        1 | 28.55 | 2023-06-01 15:00:00
              6 |        130 |      41 |        2 | 24.74 | 2023-06-01 16:00:00
              7 |        145 |      40 |        5 | 38.94 | 2023-06-01 17:00:00
              8 |        150 |      50 |        5 | 25.32 | 2023-06-01 18:00:00
              9 |        102 |      48 |        5 | 25.34 | 2023-06-01 19:00:00
             10 |        106 |      43 |        4 | 29.71 | 2023-06-01 20:00:00
             11 |        150 |      14 |        4 | 48.42 | 2023-06-01 21:00:00
             12 |        144 |      37 |        2 | 40.56 | 2023-06-01 22:00:00
             13 |        115 |      25 |        4 | 42.06 | 2023-06-01 23:00:00
             14 |        138 |      47 |        1 | 34.13 | 2023-06-02 00:00:00
             15 |        135 |      34 |        4 | 26.19 | 2023-06-02 01:00:00
             16 |        149 |      29 |        5 | 14.69 | 2023-06-02 02:00:00
             17 |        125 |      50 |        1 | 11.06 | 2023-06-02 03:00:00
             18 |        103 |       1 |        2 | 30.46 | 2023-06-02 04:00:00
             19 |        137 |       6 |        1 | 18.44 | 2023-06-02 05:00:00
             20 |        115 |      46 |        5 | 14.46 | 2023-06-02 06:00:00
             21 |        125 |      24 |        3 | 14.53 | 2023-06-02 07:00:00
             22 |        123 |       2 |        3 | 27.28 | 2023-06-02 08:00:00
             23 |        124 |      20 |        3 | 35.73 | 2023-06-02 09:00:00
             24 |        132 |      34 |        5 | 32.12 | 2023-06-02 10:00:00
             25 |        135 |      31 |        2 | 11.98 | 2023-06-02 11:00:00
             26 |        124 |      29 |        3 | 35.93 | 2023-06-02 12:00:00
             27 |        141 |      19 |        5 | 45.49 | 2023-06-02 13:00:00
             28 |        111 |       5 |        2 | 31.96 | 2023-06-02 14:00:00
             29 |        102 |      17 |        5 | 11.88 | 2023-06-02 15:00:00
             30 |        119 |      47 |        3 | 30.39 | 2023-06-02 16:00:00
             31 |        120 |      32 |        3 |  14.6 | 2023-06-02 17:00:00
             32 |        106 |      29 |        5 |  19.5 | 2023-06-02 18:00:00
             33 |        101 |      20 |        1 | 21.37 | 2023-06-02 19:00:00
             34 |        108 |      20 |        2 | 27.57 | 2023-06-02 20:00:00
             35 |        106 |       8 |        4 | 31.43 | 2023-06-02 21:00:00
             36 |        134 |      39 |        2 | 28.49 | 2023-06-02 22:00:00
             37 |        123 |      37 |        2 | 44.61 | 2023-06-02 23:00:00
             38 |        120 |      35 |        5 | 39.35 | 2023-06-03 00:00:00
             39 |        142 |      50 |        1 | 14.03 | 2023-06-03 01:00:00
             40 |        110 |      42 |        5 |  18.9 | 2023-06-03 02:00:00
             41 |        121 |      50 |        5 | 19.85 | 2023-06-03 03:00:00
             42 |        105 |      33 |        4 | 19.71 | 2023-06-03 04:00:00
             43 |        130 |      21 |        1 | 44.68 | 2023-06-03 05:00:00
```

```
ORDER BY total_sales DESC
WARN  : Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. tez) or using Hive 1.X releases.
INFO  : Query ID = nandhumidhun_20240701161819_967810f3-103b-4d21-bf39-355be7f42a89
INFO  : Total jobs = 2
INFO  : Launching Job 1 out of 2
INFO  : Starting task [Stage-1:MAPRED] in serial mode
INFO  : Number of reduce tasks not specified. Estimated from input data size: 1
INFO  : In order to change the average load for a reducer (in bytes):
INFO  :   set hive.exec.reducers.bytes.per.reducer=<number>
INFO  : In order to limit the maximum number of reducers:
INFO  :   set hive.exec.reducers.max=<number>
INFO  : In order to set a constant number of reducers:
INFO  :   set mapreduce.job.reduces=<number>
INFO  : number of splits:1
INFO  : Submitting tokens for job: job_local1577149079_0002
INFO  : Executing with tokens: []
INFO  : The url to track the job: http://localhost:8080/
INFO  : Job running in-process (local Hadoop)
INFO  : 2024-07-01 16:18:20,908 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 0.49 sec
INFO  : MapReduce Total cumulative CPU time: 490 msec
INFO  : Ended Job = job_local1577149079_0002
INFO  : Launching Job 2 out of 2
INFO  : Starting task [Stage-2:MAPRED] in serial mode
INFO  : Number of reduce tasks determined at compile time: 1
INFO  : In order to change the average load for a reducer (in bytes):
INFO  :   set hive.exec.reducers.bytes.per.reducer=<number>
INFO  : In order to limit the maximum number of reducers:
INFO  :   set hive.exec.reducers.max=<number>
INFO  : In order to set a constant number of reducers:
INFO  :   set mapreduce.job.reduces=<number>
INFO  : number of splits:1
INFO  : Submitting tokens for job: job_local1793540612_0003
INFO  : Executing with tokens: []
INFO  : The url to track the job: http://localhost:8080/
INFO  : Job running in-process (local Hadoop)
INFO  : 2024-07-01 16:18:22,336 Stage-2 map = 100%,  reduce = 100%, Cumulative CPU 0.31 sec
INFO  : MapReduce Total cumulative CPU time: 310 msec
INFO  : Ended Job = job_local1793540612_0003
INFO  : MapReduce Jobs Launched:
INFO  : Stage-Stage-1:  Cumulative CPU: 0.49 sec   HDFS Read: 15284 HDFS Write: 0 HDFS EC Read: 0 SUCCESS
INFO  : Stage-Stage-2:  Cumulative CPU: 0.31 sec   HDFS Read: 15284 HDFS Write: 0 HDFS EC Read: 0 SUCCESS
INFO  : Total MapReduce CPU Time Spent: 800 msec
INFO  : Completed executing command(queryId=nandhumidhun_20240701161819_967810f3-103b-4d21-bf39-355be7f42a89); Time taken: 2.961 seconds
+-------------+-------------+
|   category  | total_sales |
+-------------+-------------+
| Hardware    | 1534.0      |
| Apparel     | 1435.0      |
| Electronics | 958.0       |
| Home        | 856.0       |
+-------------+-------------+
4 rows selected (3.33 seconds)
0: jdbc:hive2://localhost:10000>
```

```
INFO  : Number of reduce tasks not specified. Estimated from input data size: 1
INFO  : In order to change the average load for a reducer (in bytes):
INFO  :   set hive.exec.reducers.bytes.per.reducer=<number>
INFO  : In order to limit the maximum number of reducers:
INFO  :   set hive.exec.reducers.max=<number>
INFO  : In order to set a constant number of reducers:
INFO  :   set mapreduce.job.reduces=<number>
INFO  : number of splits:1
INFO  : Submitting tokens for job: job_local988817042_0004
INFO  : Executing with tokens: []
INFO  : The url to track the job: http://localhost:8080/
INFO  : Job running in-process (local Hadoop)
INFO  : 2024-07-01 16:18:44,321 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 0.46 sec
INFO  : MapReduce Total cumulative CPU time: 460 msec
INFO  : Ended Job = job_local988817042_0004
INFO  : Launching Job 2 out of 2
INFO  : Starting task [Stage-2:MAPRED] in serial mode
INFO  : Number of reduce tasks determined at compile time: 1
INFO  : In order to change the average load for a reducer (in bytes):
INFO  :   set hive.exec.reducers.bytes.per.reducer=<number>
INFO  : In order to limit the maximum number of reducers:
INFO  :   set hive.exec.reducers.max=<number>
INFO  : In order to set a constant number of reducers:
INFO  :   set mapreduce.job.reduces=<number>
INFO  : number of splits:1
INFO  : Submitting tokens for job: job_local931053625_0005
INFO  : Executing with tokens: []
INFO  : The url to track the job: http://localhost:8080/
INFO  : Job running in-process (local Hadoop)
INFO  : 2024-07-01 16:18:45,760 Stage-2 map = 100%,  reduce = 100%, Cumulative CPU 0.21 sec
INFO  : MapReduce Total cumulative CPU time: 210 msec
INFO  : Ended Job = job_local931053625_0005
INFO  : MapReduce Jobs Launched:
INFO  : Stage-Stage-1:  Cumulative CPU: 0.46 sec   HDFS Read: 22926 HDFS Write: 0 HDFS EC Read: 0 SUCCESS
INFO  : Stage-Stage-2:  Cumulative CPU: 0.21 sec   HDFS Read: 22926 HDFS Write: 0 HDFS EC Read: 0 SUCCESS
INFO  : Total MapReduce CPU Time Spent: 670 msec
INFO  : Completed executing command(queryId=nandhumidhun_20240701161842_f39fb62a-f0d5-4174-a904-bd54ddda5e5b); Time taken: 2.913 seconds
+--------------+--------------+
| product_name  | total_sales  |
+--------------+--------------+
| "Product i"   | 569.0        |
| "Product O"   | 438.0        |
| "Product F"   | 344.0        |
| "Product r"   | 317.0        |
| "Product B"   | 273.0        |
| "Product T"   | 240.0        |
| "Product X"   | 216.0        |
| "Product h"   | 210.0        |
| "Product m"   | 195.0        |
| "Product l"   | 176.0        |
+--------------+--------------+
10 rows selected (3.25 seconds)
0: jdbc:hive2://localhost:10000>
```

```
INFO  : Ended Job = job_local334021857_0006
INFO  : Launching Job 2 out of 2
INFO  : Starting task [Stage-2:MAPRED] in serial mode
INFO  : Number of reduce tasks determined at compile time: 1
INFO  : In order to change the average load for a reducer (in bytes):
INFO  :   set hive.exec.reducers.bytes.per.reducer=<number>
INFO  : In order to limit the maximum number of reducers:
INFO  :   set hive.exec.reducers.max=<number>
INFO  : In order to set a constant number of reducers:
INFO  :   set mapreduce.job.reduces=<number>
INFO  : number of splits:1
INFO  : Submitting tokens for job: job_local1849434121_0007
INFO  : Executing with tokens: []
INFO  : The url to track the job: http://localhost:8080/
INFO  : Job running in-process (local Hadoop)
INFO  : 2024-07-01 16:21:00,142 Stage-2 map = 100%,  reduce = 100%, Cumulative CPU 0.32 sec
INFO  : MapReduce Total cumulative CPU time: 320 msec
INFO  : Ended Job = job_local1849434121_0007
INFO  : MapReduce Jobs Launched:
INFO  : Stage-Stage-1:  Cumulative CPU: 0.33 sec   HDFS Read: 30568 HDFS Write: 0 HDFS EC Read: 0 SUCCESS
INFO  : Stage-Stage-2:  Cumulative CPU: 0.32 sec   HDFS Read: 30568 HDFS Write: 0 HDFS EC Read: 0 SUCCESS
INFO  : Total MapReduce CPU Time Spent: 650 msec
INFO  : Completed executing command(queryId=nandhumidhun_20240701162056_15ee8806-5d17-496b-acf2-50eab4cb97a4); Time taken: 2.904 seconds
+------+-------------+
| age  | user_count  |
+------+-------------+
| 22   | 1           |
| 23   | 1           |
| 24   | 1           |
| 29   | 2           |
| 30   | 1           |
| 33   | 4           |
| 38   | 1           |
| 40   | 1           |
| 42   | 1           |
| 43   | 1           |
| 44   | 1           |
| 46   | 1           |
| 50   | 1           |
| 52   | 4           |
| 54   | 1           |
| 55   | 2           |
| 58   | 2           |
| 59   | 1           |
| 60   | 1           |
| 62   | 1           |
| 64   | 1           |
| 65   | 1           |
| 66   | 1           |
| 68   | 1           |
+------+-------------+
24 rows selected (3.219 seconds)
0: jdbc:hive2://localhost:10000>
```

userId Analysis Dashboard

A complete analysis on userId

Filters

completed: All

userId Analysis Dashboard

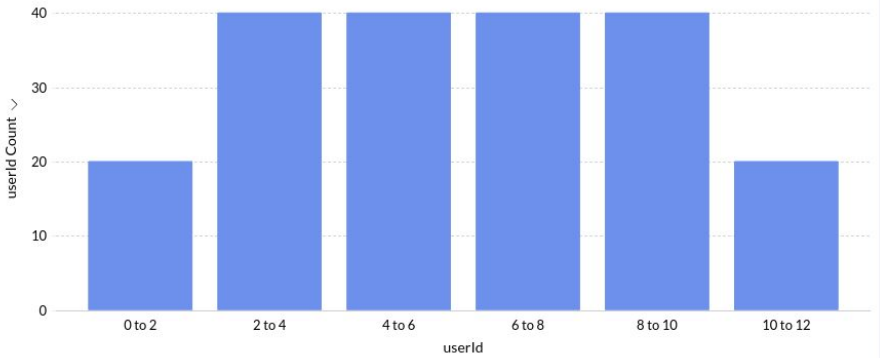
completed wise average userId
0.55K

completed with MIN userId(true)
0.51K

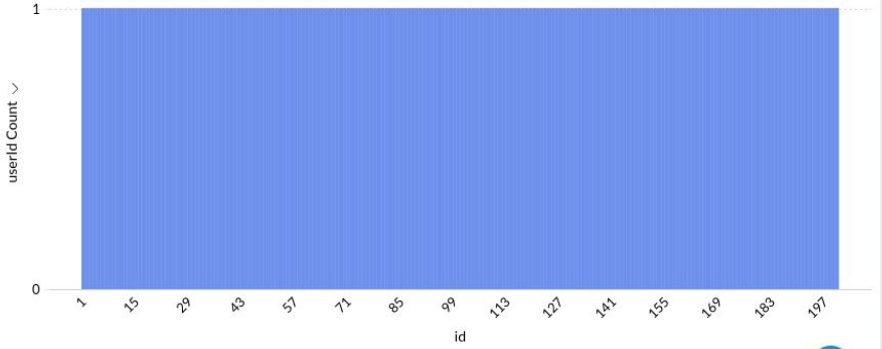
completed with MAX userId(false)
0.59K

Average of userId
5.50

Distribution of userId



id-wise userId



completed-wise userId

Top 10 title by userId

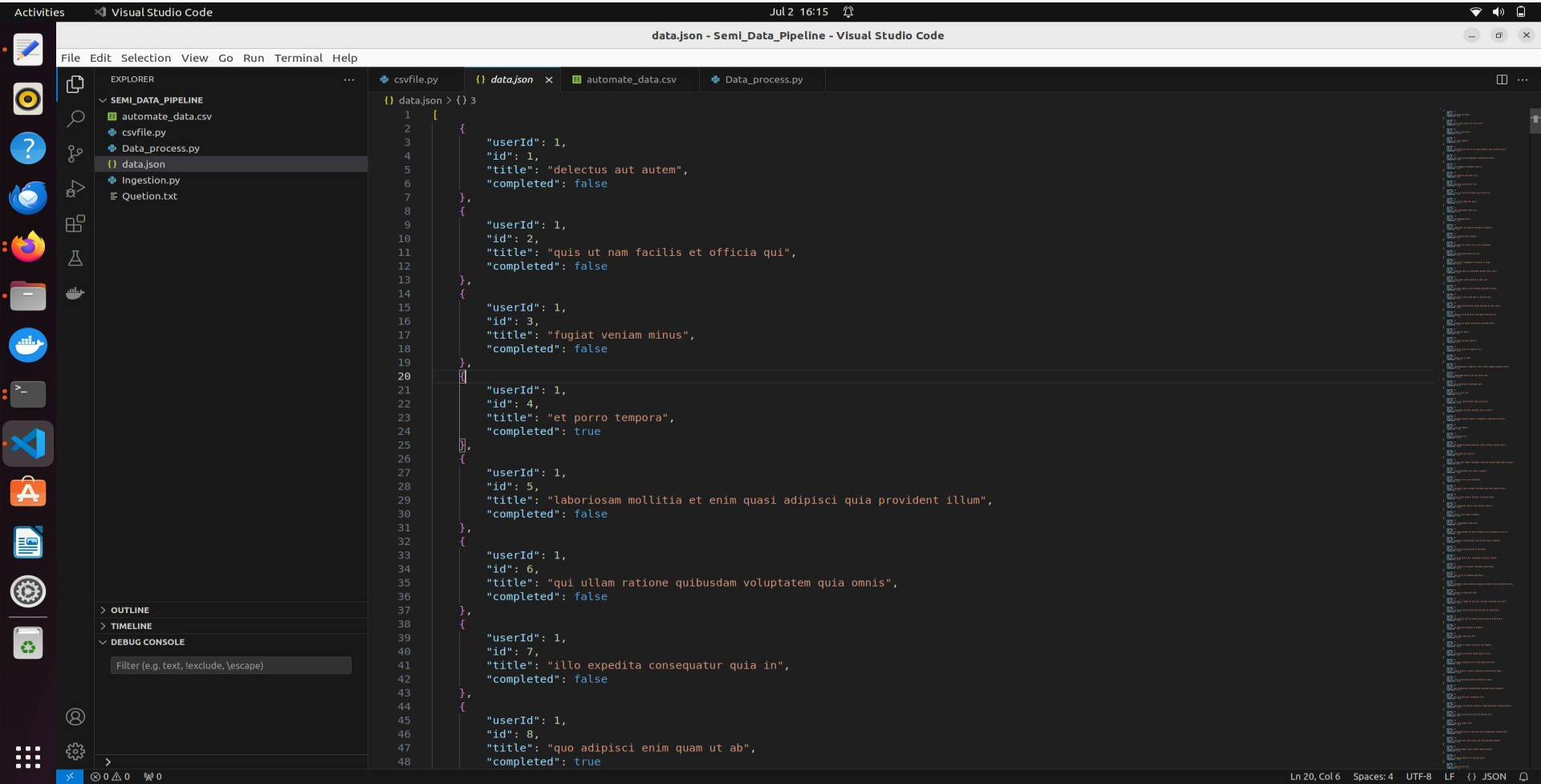
SEMI-STRUCTURED DATA PIPELINE PROCESS

STEP 1: Create a pipeline that ingests semi-structured data

STEP 2: Transform and normalize the data into a unified format.

STEP 3: Load the processed data into a HDFS

SEMI-STRUCTURED DATA PIPELINE




```
[
    {
        "userId": 1,
        "id": 1,
        "title": "delectus aut autem",
        "completed": false
    },
    {
        "userId": 1,
        "id": 2,
        "title": "quis ut nam facilis et officia qui",
        "completed": false
    },
    {
        "userId": 1,
        "id": 3,
        "title": "fugiat veniam minus",
        "completed": false
    },
    {
        "userId": 1,
        "id": 4,
        "title": "et porro tempora",
        "completed": true
    },
    {
        "userId": 1,
        "id": 5,
        "title": "laboriosam mollitia et enim quasi adipisci quia provident illum",
        "completed": false
    },
    {
        "userId": 1,
        "id": 6,
        "title": "qui ullam ratione quibusdam voluptatem quia omnis",
        "completed": false
    },
    {
        "userId": 1,
        "id": 7,
        "title": "illo expedita consequatur quia in",
        "completed": false
    },
    {
        "userId": 1,
        "id": 8,
        "title": "quo adipisci enim quam ut ab",
        "completed": true
```

Browse Directory

/user/hive/warehouse Go!

Show 25 entries Search:

	Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
☐	-rw-r--r--	nandhumidhun	supergroup	27.65 KB	Jul 02 12:00	1	128 MB	data.json
☐	drwxrwxr-x	anonymous	supergroup	0 B	Jul 01 11:36	0	0 B	hadoop.db
☐	drwxrwxr-x	anonymous	supergroup	0 B	Jun 27 13:46	0	0 B	hive.db
☐	drwxrwxr-x	anonymous	supergroup	0 B	Jun 30 13:21	0	0 B	temp_products
☐	drwxrwxr-x	anonymous	supergroup	0 B	Jun 30 13:21	0	0 B	temp_transactions
☐	drwxrwxr-x	anonymous	supergroup	0 B	Jun 30 19:31	0	0 B	transaction_details_output

Showing 1 to 6 of 6 entries Previous 1 Next

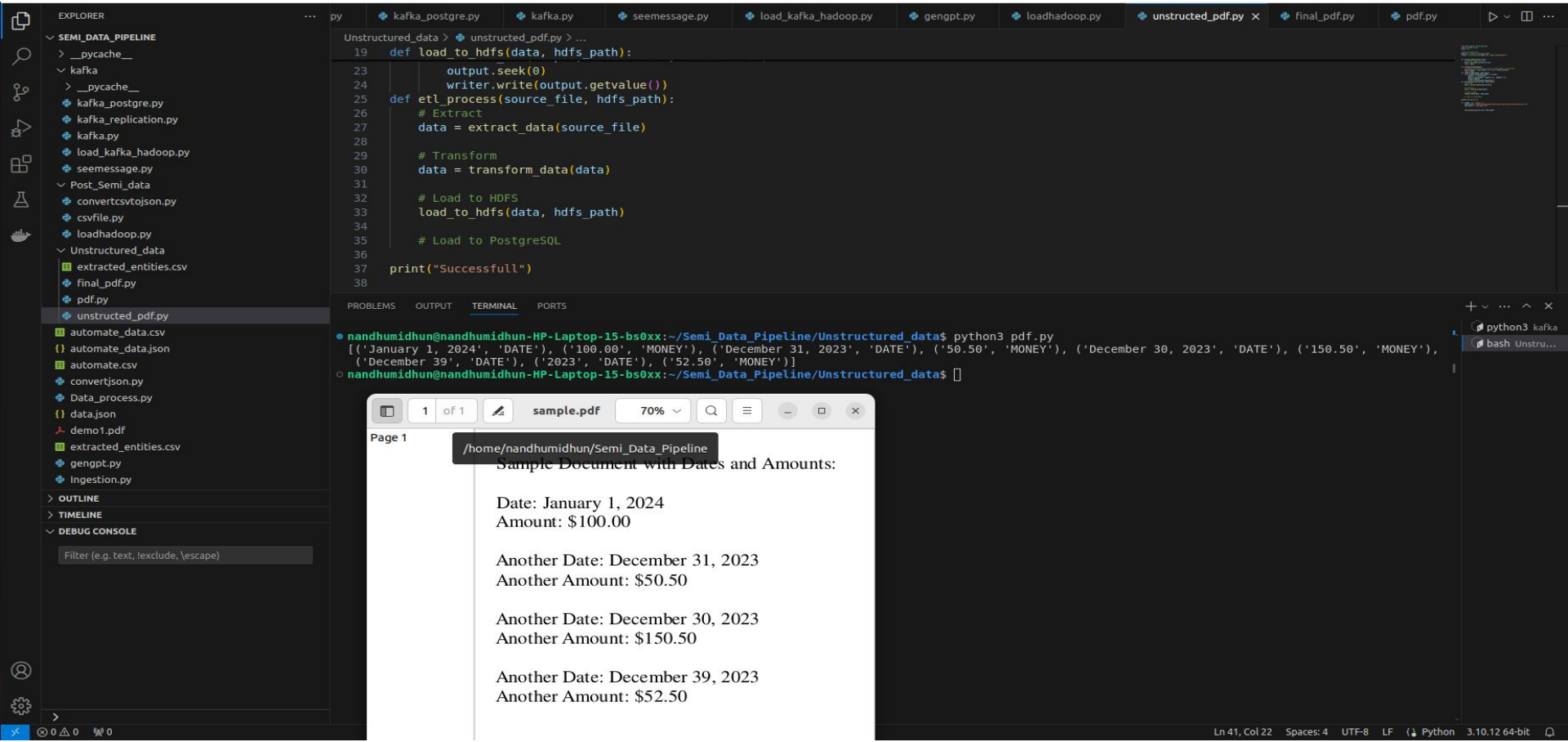
AUTOMATED DOCUMENT PROCESSING PROCESS

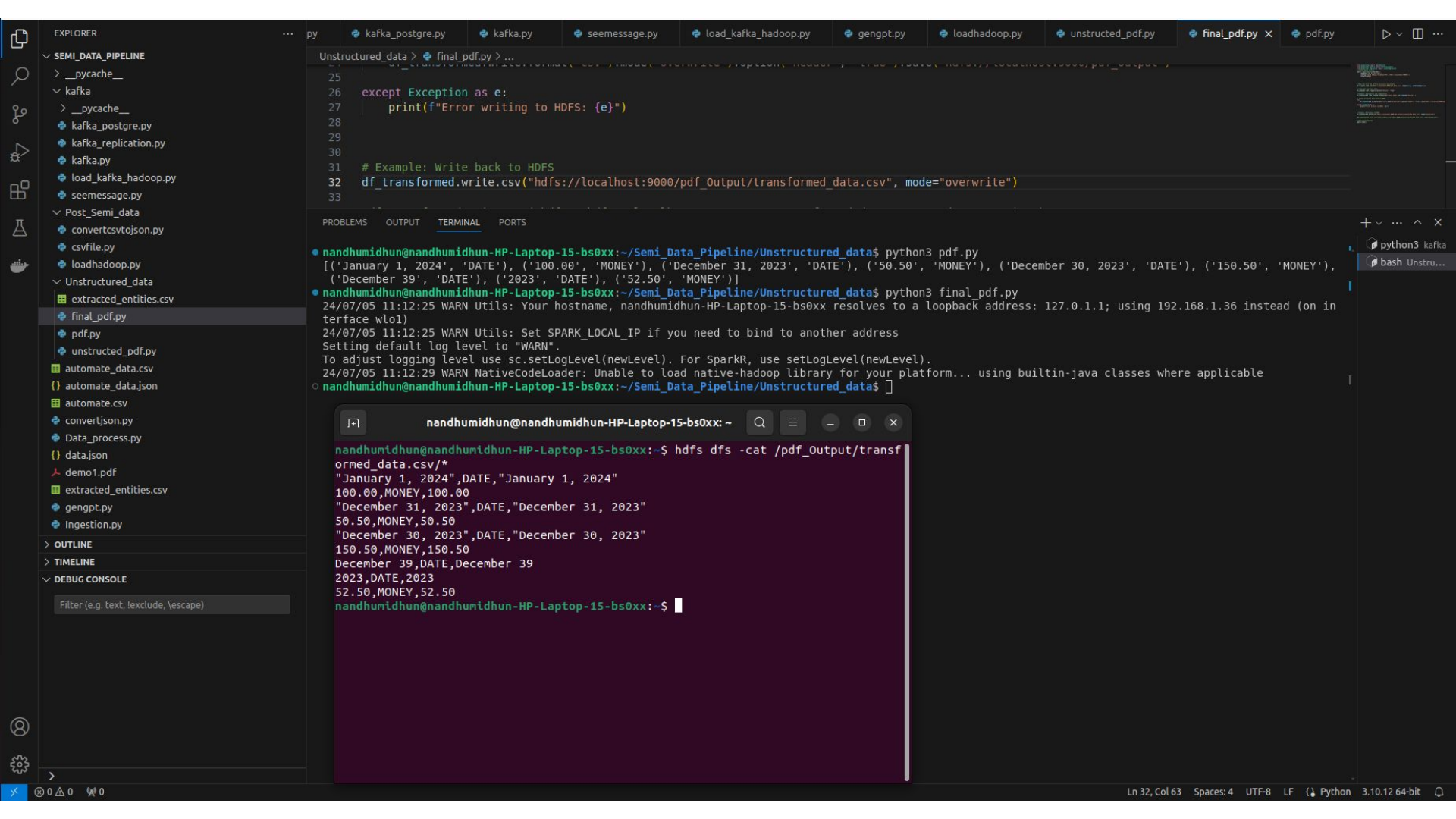
STEP 1: Build an AI-driven solution that extracts relevant information from unstructured documents

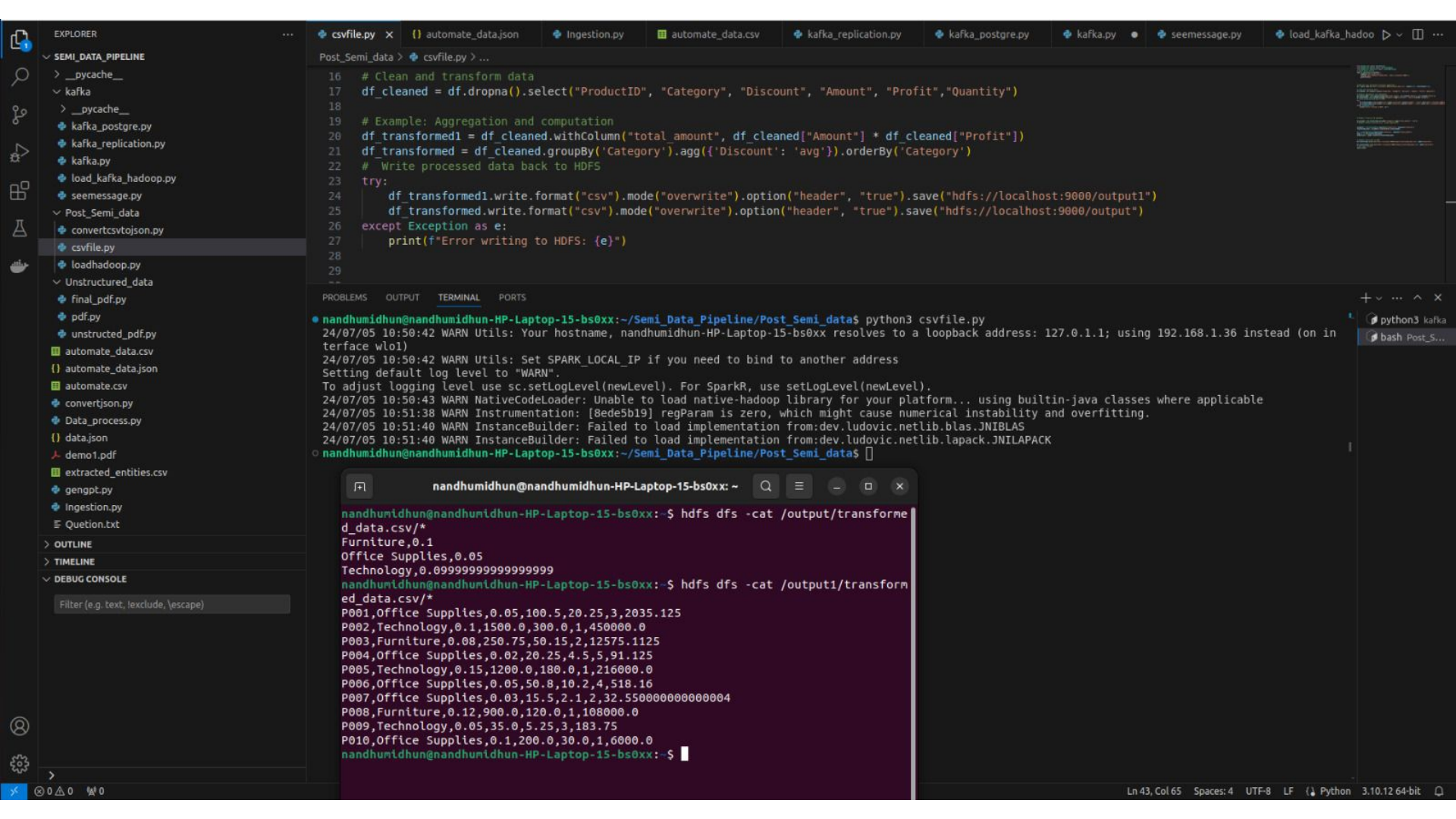
STEP 2: Use natural language processing (NLP) techniques to identify key entities

STEP 3: Showcase how this automation reduces manual effort, improves accuracy, and accelerates decision-making

AUTOMATED DOCUMENT PROCESSING




EXPLORER
SEMI_DATA_PIPELINE
__pycache__
kafka
__pycache__
kafka_postgre.py
kafka_replication.py
kafka.py
load_kafka_hadoop.py
seemessage.py
Post_Semi_data
convertcsvtojson.py
csvfile.py
loadhadoop.py
Unstructured_data
extracted_entities.csv
final_pdf.py
pdf.py
unstructured_pdf.py
automate_data.csv
automate_data.json
automate.csv
convertjson.py
Data_process.py
data.json
demo1.pdf
extracted_entities.csv
gengpt.py
Ingestion.py
OUTLINE
TIMELINE
DEBUG CONSOLE
Filter (e.g. text, !exclude, \escape)
kafka_postgre.py
kafka.py
seemessage.py
load_kafka_hadoop.py
gengpt.py
loadhadoop.py
unstructured_pdf.py
final_pdf.py
pdf.py
Unstructured_data > final_pdf.py > ...
except Exception as e:
print(f"Error writing to HDFS: {e}")
# Example: Write back to HDFS
df_transformed.write.csv("hdfs://localhost:9000/pdf_Output/transformed_data.csv", mode="overwrite")
PROBLEMS
OUTPUT
TERMINAL
PORTS
nandhumidhun@nandhumidhun-HP-Laptop-15-bs0xx:~/Semi_Data_Pipeline/Unstructured_data$ python3 pdf.py
[('January 1, 2024', 'DATE'), ('100.00', 'MONEY'), ('December 31, 2023', 'DATE'), ('50.50', 'MONEY'), ('December 30, 2023', 'DATE'), ('150.50', 'MONEY'), ('December 39', 'DATE'), ('2023', 'DATE'), ('52.50', 'MONEY')]
nandhumidhun@nandhumidhun-HP-Laptop-15-bs0xx:~/Semi_Data_Pipeline/Unstructured_data$ python3 final_pdf.py
24/07/05 11:12:25 WARN Utils: Your hostname, nandhumidhun-HP-Laptop-15-bs0xx resolves to a loopback address: 127.0.1.1; using 192.168.1.36 instead (on interface wlo1)
24/07/05 11:12:25 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
24/07/05 11:12:29 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
nandhumidhun@nandhumidhun-HP-Laptop-15-bs0xx:~/Semi_Data_Pipeline/Unstructured_data$
python3 kafka
bash Unstru...
nandhumidhun@nandhumidhun-HP-Laptop-15-bs0xx: ~
nandhumidhun@nandhumidhun-HP-Laptop-15-bs0xx:~$ hdfs dfs -cat /pdf_Output/transformed_data.csv/*
"January 1, 2024",DATE,"January 1, 2024"
100.00,MONEY,100.00
"December 31, 2023",DATE,"December 31, 2023"
50.50,MONEY,50.50
"December 30, 2023",DATE,"December 30, 2023"
150.50,MONEY,150.50
December 39,DATE,December 39
2023,DATE,2023
52.50,MONEY,52.50
nandhumidhun@nandhumidhun-HP-Laptop-15-bs0xx:~$
Ln 32, Col 63
Spaces: 4
UTF-8
LF
Python
3.10.12 64-bit

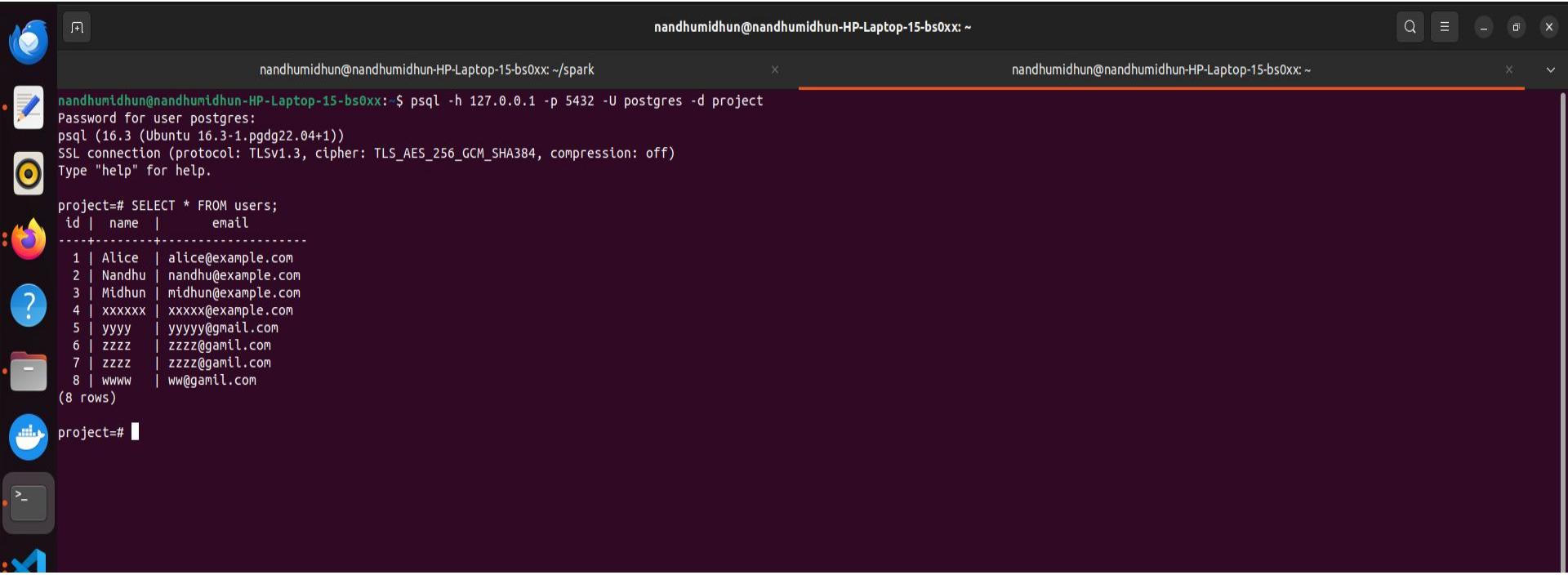

EXPLORER
SEMI_DATA_PIPELINE
__pycache__
kafka
__pycache__
kafka_postgre.py
kafka_replication.py
kafka.py
load_kafka_hadoop.py
seemessage.py
Post_Semi_data
convertcsvtojson.py
csvfile.py
loadhadoop.py
Unstructured_data
final_pdf.py
pdf.py
unstructed_pdf.py
automate_data.csv
automate_data.json
automate.csv
convertjson.py
Data_process.py
data.json
demo1.pdf
extracted_entities.csv
gengpt.py
Ingestion.py
Quetion.txt
OUTLINE
TIMELINE
DEBUG CONSOLE
Filter (e.g. text, !exclude, \escape)
csvfile.py
automate_data.json
Ingestion.py
automate_data.csv
kafka_replication.py
kafka_postgre.py
kafka.py
seemessage.py
load_kafka_hadoo
Post_Semi_data > csvfile.py > ...
16 # Clean and transform data
17 df_cleaned = df.dropna().select("ProductID", "Category", "Discount", "Amount", "Profit","Quantity")
18
19 # Example: Aggregation and computation
20 df_transformed1 = df_cleaned.withColumn("total_amount", df_cleaned["Amount"] * df_cleaned["Profit"])
21 df_transformed = df_cleaned.groupBy('Category').agg({'Discount': 'avg'}).orderBy('Category')
22 # Write processed data back to HDFS
23 try:
24 df_transformed1.write.format("csv").mode("overwrite").option("header", "true").save("hdfs://localhost:9000/output1")
25 df_transformed.write.format("csv").mode("overwrite").option("header", "true").save("hdfs://localhost:9000/output")
26 except Exception as e:
27 print(f"Error writing to HDFS: {e}")
28
29
PROBLEMS
OUTPUT
TERMINAL
PORTS
nandhumidhun@nandhumidhun-HP-Laptop-15-bs0xx:~/Semi_Data_Pipeline/Post_Semi_data$ python3 csvfile.py
24/07/05 10:50:42 WARN Utils: Your hostname, nandhumidhun-HP-Laptop-15-bs0xx resolves to a loopback address: 127.0.1.1; using 192.168.1.36 instead (on interface wlo1)
24/07/05 10:50:42 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
24/07/05 10:50:43 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
24/07/05 10:51:38 WARN Instrumentation: [8ede5b19] regParam is zero, which might cause numerical instability and overfitting.
24/07/05 10:51:40 WARN InstanceBuilder: Failed to load implementation from:dev.ludovic.netlib.blas.JNIBLAS
24/07/05 10:51:40 WARN InstanceBuilder: Failed to load implementation from:dev.ludovic.netlib.lapack.JNILAPACK
nandhumidhun@nandhumidhun-HP-Laptop-15-bs0xx:~/Semi_Data_Pipeline/Post_Semi_data$
python3 kafka
bash Post_S...
nandhumidhun@nandhumidhun-HP-Laptop-15-bs0xx: ~
nandhumidhun@nandhumidhun-HP-Laptop-15-bs0xx:~$ hdfs dfs -cat /output/transformed_data.csv/*
Furniture,0.1
Office Supplies,0.05
Technology,0.09999999999999999
nandhumidhun@nandhumidhun-HP-Laptop-15-bs0xx:~$ hdfs dfs -cat /output1/transformed_data.csv/*
P001,Office Supplies,0.05,100.5,20.25,3,2035.125
P002,Technology,0.1,1500.0,300.0,1,450000.0
P003,Furniture,0.08,250.75,50.15,2,12575.1125
P004,Office Supplies,0.02,20.25,4.5,5,91.125
P005,Technology,0.15,1200.0,180.0,1,216000.0
P006,Office Supplies,0.05,50.8,10.2,4,518.16
P007,Office Supplies,0.03,15.5,2.1,2,32.550000000000004
P008,Furniture,0.12,900.0,120.0,1,108000.0
P009,Technology,0.05,35.0,5.25,3,183.75
P010,Office Supplies,0.1,200.0,30.0,1,6000.0
nandhumidhun@nandhumidhun-HP-Laptop-15-bs0xx:~$
Ln 43, Col 65
Spaces: 4
UTF-8
LF
Python
3.10.12 64-bit

RELATIONAL DATA PIPELINE PROCESS

STEP 1: Set up a pipeline to replicate data between relational databases

STEP 2: Highlight the benefits of real-time data replication for reporting, analytics, and business continuity

RELATIONAL DATA PIPELINE



File Edit Selection View Go Run Terminal Help

EXPLORER

SEMI_DATA_PIPELINE
- __pycache__
- kafka
  - __pycache__
  - kafka_postgre.py
  - kafka_replication.py
  - kafka.py
  - load_kafka_hadoop.py
  - seemessage.py
- Post_Semi_data
  - convertcsvtojson.py
  - csvfile.py
  - loadhadoop.py
- Unstructured_data
  - final_pdf.py
  - pdf.py
  - unstructed_pdf.py
- automate_data.csv
- automate_data.json
- automate.csv
- convertjson.py
- Data_process.py
- data.json
- demo1.pdf
- extracted_entities.csv
- gengpt.py
- Ingestion.py
- Quetion.txt

OUTLINE

TIMELINE

DEBUG CONSOLE

Filter (e.g. text, !exclude, \escape)

automate_data.csv | kafka_replication.py | kafka_postgre.py | kafka.py | seemessage.py | load_kafka_hadoop.py | gengpt.py | loadhadoop.py | unstructed_p

kafka > seemessage.py > ...

```
18  consumer.subscribe([kafka_topic])
19
20  # Consume messages
21  try:
22      i=0
23      while i<16:
24          msg = consumer.poll(timeout=1.0)
25          if msg is None:
26              continue
27          if msg.error():
28              if msg.error().code() == KafkaError._PARTITION_EOF:
29                  continue
30              else:
31                  print(msg.error())
32          print(f"Received message: {msg.value().decode('utf-8')}")
```

PROBLEMS OUTPUT TERMINAL PORTS

```
nandhumidhun@nandhumidhun-HP-Laptop-15-bs0xx:~/Semi_Data_Pipeline/kafka$ python3 kafka_postgre.py
nandhumidhun@nandhumidhun-HP-Laptop-15-bs0xx:~/Semi_Data_Pipeline/kafka$ python3 seemessage.py
Received message: [2, "Nandhu", "nandhu@example.com"]
Received message: [3, "Midhun", "midhun@example.com"]
Received message: [4, "xxxxxx", "xxxxx@example.com"]
Received message: [5, "yyyy", "yyyyy@gmail.com"]
Received message: [6, "zzzz", "zzzz@gamil.com"]
Received message: [7, "zzzz", "zzzz@gamil.com"]
Received message: [8, "wwww", "ww@gamil.com"]
Received message: [1, "Alice", "alice@example.com"]
Received message: [3, "Midhun", "midhun@example.com"]
Received message: [4, "xxxxxx", "xxxxx@example.com"]
Received message: [5, "yyyy", "yyyyy@gmail.com"]
Received message: [2, "Nandhu", "nandhu@example.com"]
Received message: [6, "zzzz", "zzzz@gamil.com"]
Received message: [7, "zzzz", "zzzz@gamil.com"]
Received message: [8, "wwww", "ww@gamil.com"]
```

python3 kafka

bash Post_S...

REAL-TIME EVENT MONITORING AND ANALYTICS SYSTEM PROCESS

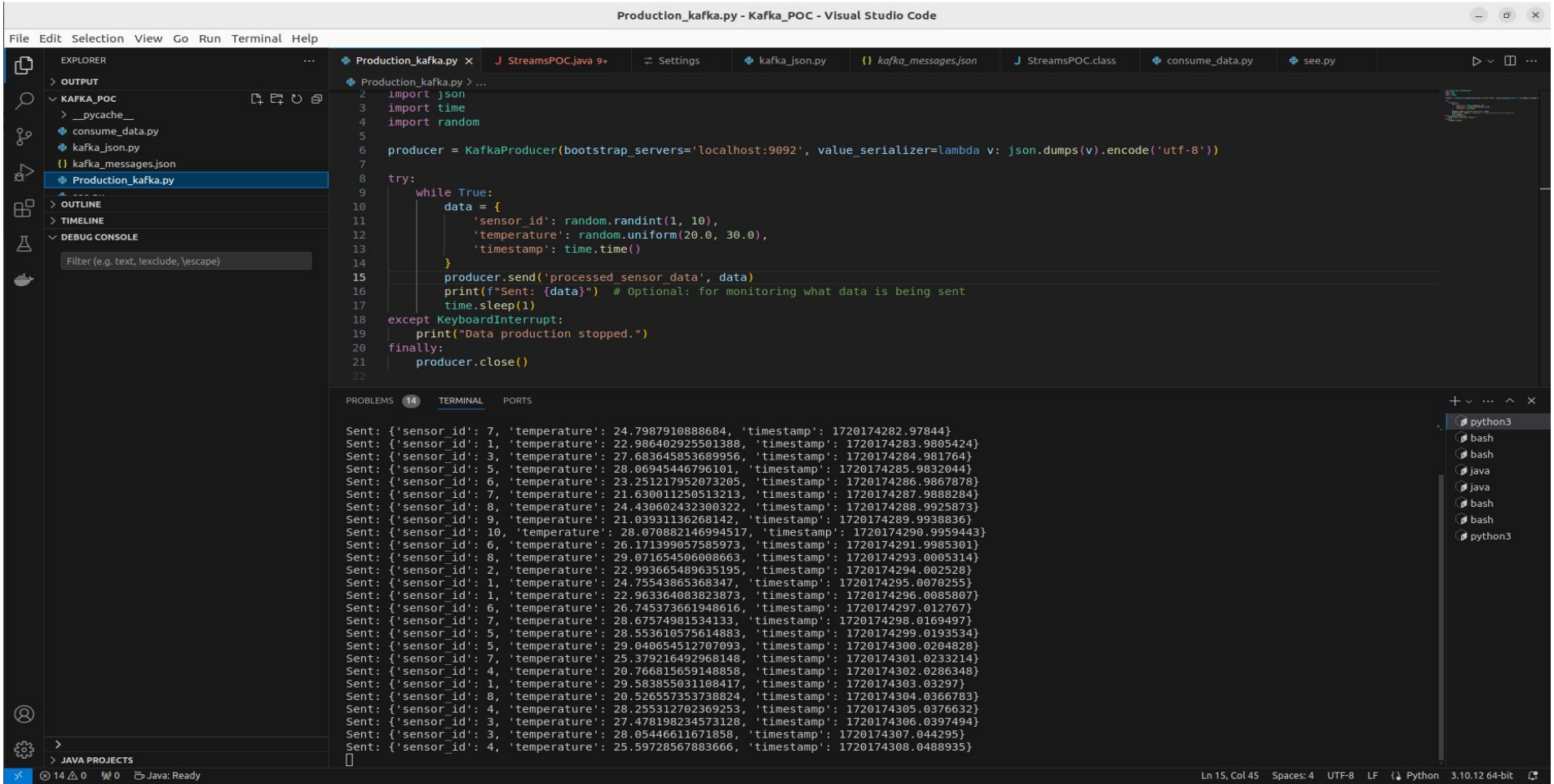
STEP 1: Create Data Producers

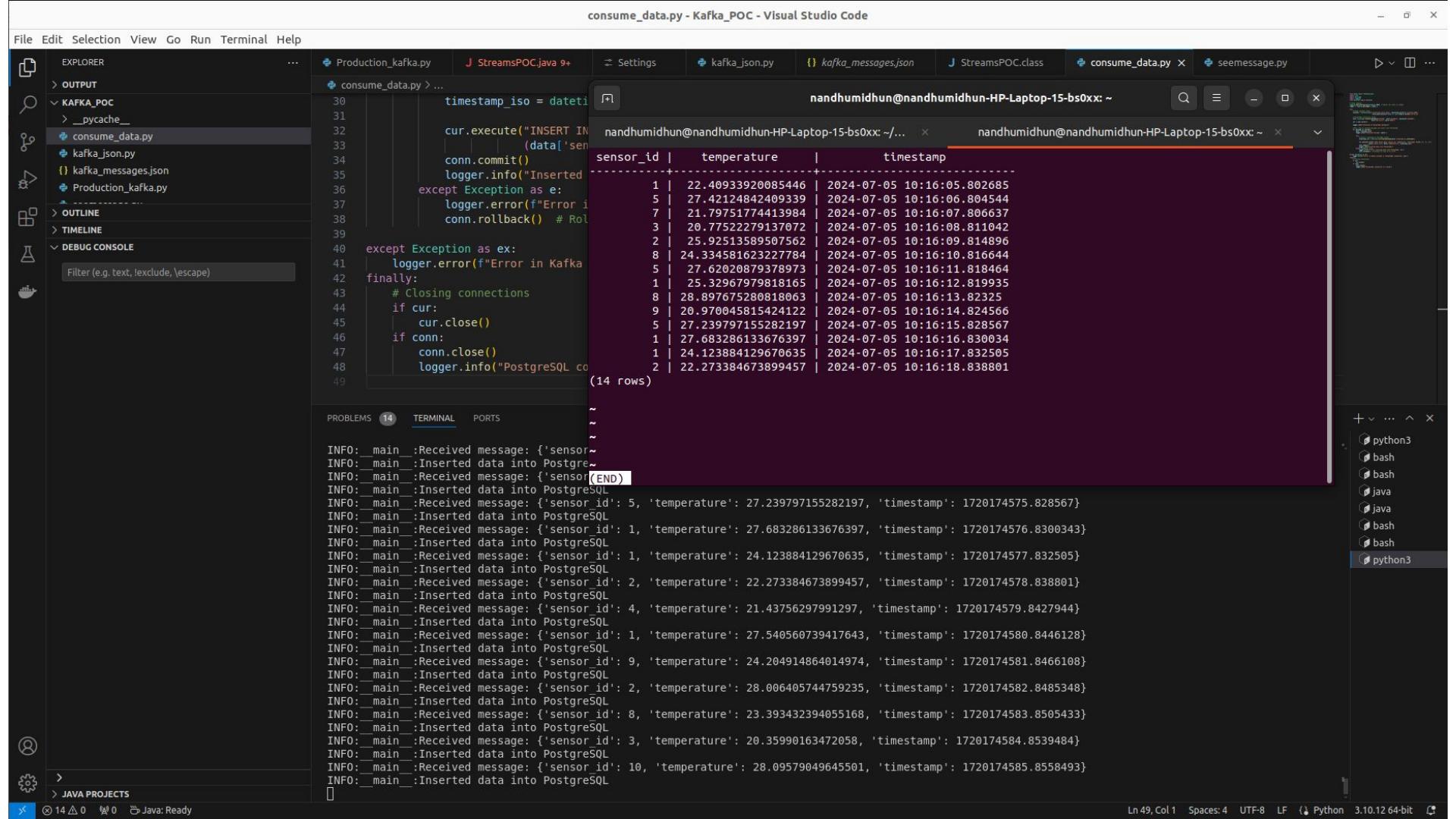
STEP 2: Data Processing with Kafka Streams

STEP 3: Store Processed Data

STEP 4: Visualization

REAL-TIME EVENT MONITORING AND ANALYTICS SYSTEM




consume_data.py - Kafka_POC - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
OUTPUT
KAFKA_POC
__pycache__
consume_data.py
kafka_json.py
kafka_messages.json
Production_kafka.py
OUTLINE
TIMELINE
DEBUG CONSOLE
Filter (e.g. text, !exclude, \escape)
Production_kafka.py
StreamsPOC.java 9+
Settings
kafka_json.py
kafka_messages.json
StreamsPOC.class
consume_data.py
seemessage.py
consume_data.py > ...
timestamp_iso = dateti
cur.execute("INSERT IN
(data['sen
conn.commit()
logger.info("Inserted
except Exception as e:
logger.error(f"Error i
conn.rollback() # Rol
except Exception as ex:
logger.error(f"Error in Kafka
finally:
# Closing connections
if cur:
cur.close()
if conn:
conn.close()
logger.info("PostgreSQL co
nandhumidhun@nandhumidhun-HP-Laptop-15-bs0xx: ~
nandhumidhun@nandhumidhun-HP-Laptop-15-bs0xx: ~/...
nandhumidhun@nandhumidhun-HP-Laptop-15-bs0xx: ~
sensor_id | temperature | timestamp
1 | 22.40933920085446 | 2024-07-05 10:16:05.802685
5 | 27.42124842409339 | 2024-07-05 10:16:06.804544
7 | 21.79751774413984 | 2024-07-05 10:16:07.806637
3 | 20.77522279137072 | 2024-07-05 10:16:08.811042
2 | 25.92513589507562 | 2024-07-05 10:16:09.814896
8 | 24.334581623227784 | 2024-07-05 10:16:10.816644
5 | 27.62020879378973 | 2024-07-05 10:16:11.818464
1 | 25.32967979818165 | 2024-07-05 10:16:12.819935
8 | 28.897675280818063 | 2024-07-05 10:16:13.82325
9 | 20.970045815424122 | 2024-07-05 10:16:14.824566
5 | 27.239797155282197 | 2024-07-05 10:16:15.828567
1 | 27.683286133676397 | 2024-07-05 10:16:16.830034
1 | 24.123884129670635 | 2024-07-05 10:16:17.832505
2 | 22.273384673899457 | 2024-07-05 10:16:18.838801
(14 rows)
(END)
PROBLEMS 14
TERMINAL
PORTS
INFO:__main__:Received message: {'sensor
INFO:__main__:Inserted data into Postgre
INFO:__main__:Received message: {'sensor
INFO:__main__:Inserted data into PostgreSQL
INFO:__main__:Received message: {'sensor_id': 5, 'temperature': 27.239797155282197, 'timestamp': 1720174575.828567}
INFO:__main__:Inserted data into PostgreSQL
INFO:__main__:Received message: {'sensor_id': 1, 'temperature': 27.683286133676397, 'timestamp': 1720174576.8300343}
INFO:__main__:Inserted data into PostgreSQL
INFO:__main__:Received message: {'sensor_id': 1, 'temperature': 24.123884129670635, 'timestamp': 1720174577.832505}
INFO:__main__:Inserted data into PostgreSQL
INFO:__main__:Received message: {'sensor_id': 2, 'temperature': 22.273384673899457, 'timestamp': 1720174578.838801}
INFO:__main__:Inserted data into PostgreSQL
INFO:__main__:Received message: {'sensor_id': 4, 'temperature': 21.43756297991297, 'timestamp': 1720174579.8427944}
INFO:__main__:Inserted data into PostgreSQL
INFO:__main__:Received message: {'sensor_id': 1, 'temperature': 27.540560739417643, 'timestamp': 1720174580.8446128}
INFO:__main__:Inserted data into PostgreSQL
INFO:__main__:Received message: {'sensor_id': 9, 'temperature': 24.204914864014974, 'timestamp': 1720174581.8466108}
INFO:__main__:Inserted data into PostgreSQL
INFO:__main__:Received message: {'sensor_id': 2, 'temperature': 28.006405744759235, 'timestamp': 1720174582.8485348}
INFO:__main__:Inserted data into PostgreSQL
INFO:__main__:Received message: {'sensor_id': 8, 'temperature': 23.393432394055168, 'timestamp': 1720174583.8505433}
INFO:__main__:Inserted data into PostgreSQL
INFO:__main__:Received message: {'sensor_id': 3, 'temperature': 20.35990163472058, 'timestamp': 1720174584.8539484}
INFO:__main__:Inserted data into PostgreSQL
INFO:__main__:Received message: {'sensor_id': 10, 'temperature': 28.09579049645501, 'timestamp': 1720174585.8558493}
INFO:__main__:Inserted data into PostgreSQL
python3
bash
bash
java
java
bash
bash
python3
JAVA PROJECTS
Java: Ready
Ln 49, Col 1
Spaces: 4
UTF-8
LF
Python
3.10.12 64-bit

File Edit Selection View Go Run Terminal Help

EXPLORER

OUTPUT

KAFKA_POC

__pycache__

consumer_data.py

Production_kafka.py

StreamsPOC.class

StreamsPOC.java 9+

OUTLINE

TIMELINE

DEBUG CONSOLE

Filter (e.g. text, !exclude, \escape)

JAVA PROJECTS

Production_kafka.py | StreamsPOC.java 9+ | Settings | StreamsPOC.class | consumer_data.py

StreamsPOC.java > StreamsPOC > main(String[])

```
7   public class StreamsPOC {
9       public static void main(String[] args) {
14          props.put(StreamsConfig.DEFAULT_VALUE_SERDE_CLASS_CONFIG, Serdes.String().getClass().getName());
15
16          StreamsBuilder builder = new StreamsBuilder();
17          KStream<String, String> sourceStream = builder.stream("processed_sensor_data");
18
19          // Process your stream or define transformations here
20          sourceStream.foreach((key, value) -> {
21              System.out.println("Key: " + key + ", Value: " + value);
22          });
23
24          KafkaStreams streams = new KafkaStreams(builder.build(), props);
25          streams.start();
26
```

PROBLEMS 14 | TERMINAL | PORTS

```
[streams-poc-2-105d75c2-6050-4bc8-afb8-bdbc61ae8e55-StreamThread-1] INFO org.apache.kafka.clients.consumer.internals.ConsumerCoordinator - [Consumer clientId=streams-poc-2-105d75c2-6050-4bc8-afb8-bdbc61ae8e55-StreamThread-1-consumer, groupId=streams-poc-2] Found no committed offset for partition processed_sensor_data-0
[streams-poc-2-105d75c2-6050-4bc8-afb8-bdbc61ae8e55-StreamThread-1] INFO org.apache.kafka.clients.consumer.internals.SubscriptionState - [Consumer clientId=streams-poc-2-105d75c2-6050-4bc8-afb8-bdbc61ae8e55-StreamThread-1-consumer, groupId=streams-poc-2] Resetting offset for partition processed_sensor_data-0 to position FetchPosition{offset=0, offsetEpoch=Optional.empty, currentLeader=LeaderAndEpoch{leader=Optional[nandhumidhun-HP-Laptop-15-bs0xx:9092 (id: 0 rack: null)], epoch=0}}.
Key: null, Value: {"sensor_id": 4, "temperature": 23.19852952294821, "timestamp": 1720241569.3474495}
Key: null, Value: {"sensor_id": 7, "temperature": 22.60240145854672, "timestamp": 1720241570.3576665}
Key: null, Value: {"sensor_id": 6, "temperature": 26.285091007789205, "timestamp": 1720241571.359596}
Key: null, Value: {"sensor_id": 7, "temperature": 29.06402170076111, "timestamp": 1720241572.3628755}
Key: null, Value: {"sensor_id": 8, "temperature": 27.698797717029294, "timestamp": 1720241573.366271}
Key: null, Value: {"sensor_id": 4, "temperature": 21.27913630499162, "timestamp": 1720241574.3673408}
Key: null, Value: {"sensor_id": 9, "temperature": 28.144526804550857, "timestamp": 1720241575.3713806}
Key: null, Value: {"sensor_id": 8, "temperature": 26.943628802194596, "timestamp": 1720241576.3753252}
Key: null, Value: {"sensor_id": 2, "temperature": 28.535616731767313, "timestamp": 1720241577.3774042}
Key: null, Value: {"sensor_id": 4, "temperature": 29.16208078654908, "timestamp": 1720241578.3793068}
Key: null, Value: {"sensor_id": 9, "temperature": 21.631640244999563, "timestamp": 1720241579.3808105}
Key: null, Value: {"sensor_id": 5, "temperature": 21.896282465271987, "timestamp": 1720241580.38255}
Key: null, Value: {"sensor_id": 4, "temperature": 25.55688583862497, "timestamp": 1720241581.3847117}
Key: null, Value: {"sensor_id": 4, "temperature": 29.37446271879176, "timestamp": 1720241582.386458}
Key: null, Value: {"sensor_id": 8, "temperature": 25.857415568051238, "timestamp": 1720241583.3874972}
Key: null, Value: {"sensor_id": 6, "temperature": 29.263712100359893, "timestamp": 1720241584.3893502}
Key: null, Value: {"sensor_id": 6, "temperature": 25.859430550798844, "timestamp": 1720241585.3911266}
Key: null, Value: {"sensor_id": 4, "temperature": 24.716697737499338, "timestamp": 1720241586.3935878}
Key: null, Value: {"sensor_id": 2, "temperature": 23.2595969873049, "timestamp": 1720241587.394147}
Key: null, Value: {"sensor_id": 8, "temperature": 25.137590893318645, "timestamp": 1720241588.3954144}
Key: null, Value: {"sensor_id": 1, "temperature": 20.881868362280766, "timestamp": 1720241589.3970525}
Key: null, Value: {"sensor_id": 6, "temperature": 24.844951993004745, "timestamp": 1720241590.398852}
Key: null, Value: {"sensor_id": 8, "temperature": 21.95386135713793, "timestamp": 1720241591.4009237}
Key: null, Value: {"sensor_id": 4, "temperature": 29.81146163668466, "timestamp": 1720241592.403235}
Key: null, Value: {"sensor_id": 10, "temperature": 25.21964599641193, "timestamp": 1720241593.4085279}
Key: null, Value: {"sensor_id": 8, "temperature": 27.2423285302754, "timestamp": 1720241594.4150567}
Key: null, Value: {"sensor_id": 7, "temperature": 27.983232449215933, "timestamp": 1720241595.4168944}
```

python3

python3

java

Ln 17, Col 74 Spaces: 4 UTF-8 LF { } Java

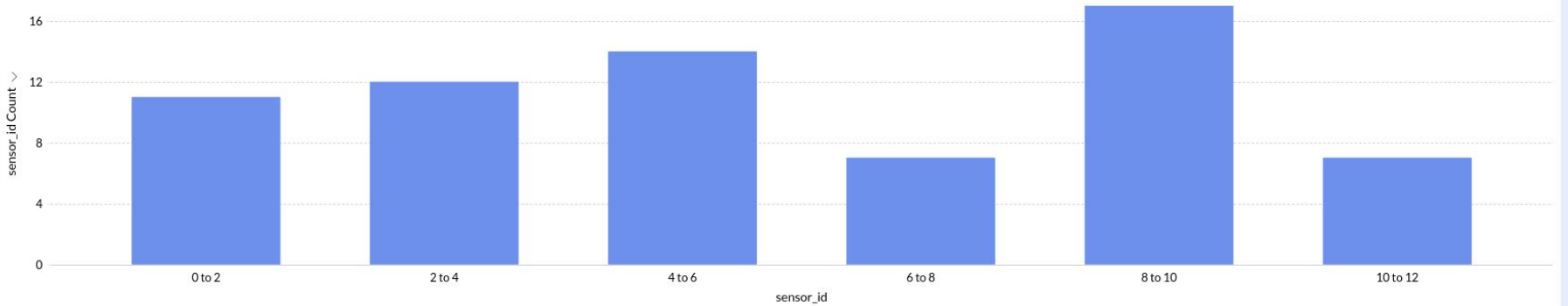
sensor_id Analysis Dashboard

A complete analysis on sensor_id

Edit Design | Ask Zia | Share | Themes

sensor_id Analysis Dashboard

Distribution of sensor_id



Top 10 timestamp by sensor_id

REAL-TIME STOCK MARKET ANALYTICS USING APACHE KAFKA AND HADOOP PROCESS

STEP 1: Fetch Data from NSE API
[https://www.nseindia.com/api/equity-stockIndices?index=NIFTY%2050]

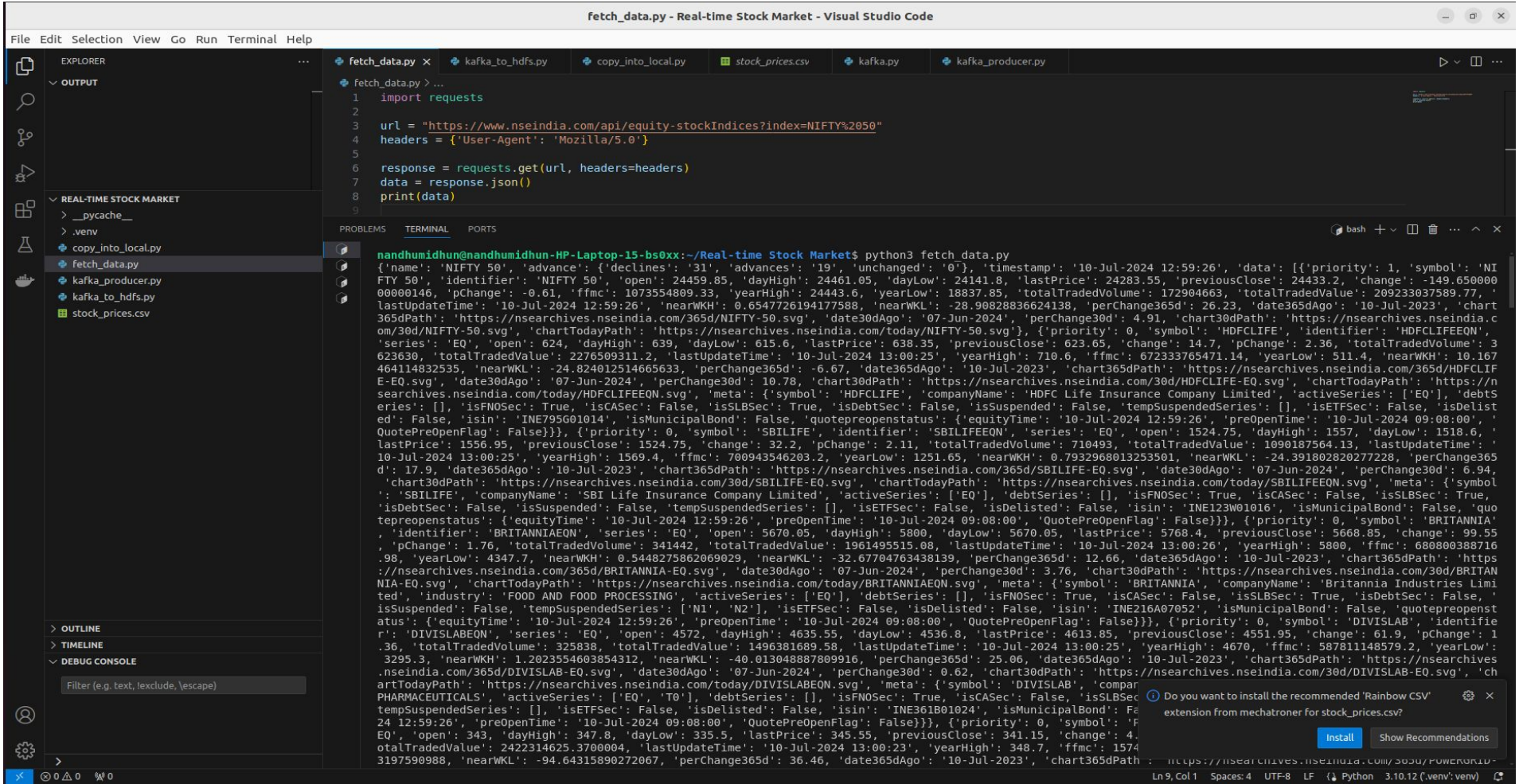
STEP 2: Produce Data to Kafka

STEP 3: Consume Data from Kafka and Store in HDFS

STEP 4: Analyze Data using Spark

STEP 5: Visualize Results

REAL-TIME STOCK MARKET ANALYTICS USING APACHE KAFKA AND HADOOP



```
nandhumidhun@nandhumidhun-HP-Laptop-15-bs0xx:~/kafka$ kafka-console-consumer.sh --bootstrap-server localhost:9092 --topic nse_stock_data --from-beginning
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/nandhumidhun/kafka/libs/slf4j-reload4j-1.7.36.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/nandhumidhun/kafka/libs/slf4j-simple-1.7.36.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Reload4jLoggerFactory]
{"name": "NIFTY 50", "advance": {"declines": "30", "advances": "20", "unchanged": "0"}, "timestamp": "10-Jul-2024 13:04:26", "data": [{"priority": 1, "symbol": "NIFTY 50", "identifier": "NIFTY 50", "open"
: 24459.85, "dayHigh": 24461.05, "dayLow": 24141.8, "lastPrice": 24283.65, "previousClose": 24433.2, "change": -149.54999999999927, "pChange": -0.61, "ffmc": 1073629150.29, "yearHigh": 24443.6, "yearLow":
 18837.85, "totalTradedVolume": 174845350, "totalTradedValue": 211622411574.23, "lastUpdateTime": "10-Jul-2024 13:04:26", "nearWKH": 0.6543635143759393, "nearWKL": -28.908819212383595, "perChange365d": 26
.23, "date365dAgo": "10-Jul-2023", "chart365dPath": "https://nsearchives.nseindia.com/365d/NIFTY-50.svg", "date30dAgo": "07-Jun-2024", "perChange30d": 4.91, "chart30dPath": "https://nsearchives.nseindia.c
om/30d/NIFTY-50.svg", "chartTodayPath": "https://nsearchives.nseindia.com/today/NIFTY-50.svg"}, {"priority": 0, "symbol": "HDFCLIFE", "identifier": "HDFCLIFEEQN", "series": "EQ", "open": 624, "dayHigh": 6
40.75, "dayLow": 615.6, "lastPrice": 639.2, "previousClose": 623.65, "change": 15.55, "pChange": 2.49, "totalTradedVolume": 4025188, "totalTradedValue": 2533453327.2, "lastUpdateTime": "10-Jul-2024 13:04:
56", "yearHigh": 710.6, "ffmc": 6745471232219.2, "yearLow": 511.4, "nearWKH": 10.047846889952151, "nearWKL": -24.990222917481444, "perChange365d": -6.67, "date365dAgo": "10-Jul-2023", "chart365dPath": "http
s://nsearchives.nseindia.com/365d/HDFCLIFE-EQ.svg", "date30dAgo": "07-Jun-2024", "perChange30d": 10.78, "chart30dPath": "https://nsearchives.nseindia.com/30d/HDFCLIFE-EQ.svg", "chartTodayPath": "https://n
searchives.nseindia.com/today/HDFCLIFEEQN.svg", "meta": {"symbol": "HDFCLIFE", "companyName": "HDFC Life Insurance Company Limited", "activeSeries": ["EQ"], "debtSeries": [], "isFNOSec": true, "isCASec":
 false, "isSLBSec": true, "isDebtSec": false, "isSuspended": false, "tempSuspendedSeries": [], "isETFSec": false, "isDelisted": false, "isin": "INE795G01014", "isMunicipalBond": false, "quotepreopenstatus"
: {"equityTime": "10-Jul-2024 13:04:26", "preOpenTime": "10-Jul-2024 09:08:00", "QuotePreOpenFlag": false}}}, {"priority": 0, "symbol": "SBILIFE", "identifier": "SBILIFEEQN", "series": "EQ", "open": 1524.
75, "dayHigh": 1559.9, "dayLow": 1518.6, "lastPrice": 1557.5, "previousClose": 1524.75, "change": 32.75, "pChange": 2.15, "totalTradedVolume": 763503, "totalTradedValue": 1172786418.18, "lastUpdateTime":
"10-Jul-2024 13:04:56", "yearHigh": 1569.4, "ffmc": 701890036331.18, "yearLow": 1251.65, "nearWKH": 0.758251561106161, "nearWKL": -24.435744816841762, "perChange365d": 17.9, "date365dAgo": "10-Jul-2023",
"chart365dPath": "https://nsearchives.nseindia.com/365d/SBILIFE-EQ.svg", "date30dAgo": "07-Jun-2024", "perChange30d": 6.94, "chart30dPath": "https://nsearchives.nseindia.com/30d/SBILIFE-EQ.svg", "chartTod
ayPath": "https://nsearchives.nseindia.com/today/SBILIFEEQN.svg", "meta": {"symbol": "SBILIFE", "companyName": "SBI Life Insurance Company Limited", "activeSeries": ["EQ"], "debtSeries": [], "isFNOSec": t
rue, "isCASec": false, "isSLBSec": true, "isDebtSec": false, "isSuspended": false, "tempSuspendedSeries": [], "isETFSec": false, "isDelisted": false, "isin": "INE123W01016", "isMunicipalBond": false, "quo
tepreopenstatus": {"equityTime": "10-Jul-2024 13:04:26", "preOpenTime": "10-Jul-2024 09:08:00", "QuotePreOpenFlag": false}}}, {"priority": 0, "symbol": "BRITANNIA", "identifier": "BRITANNIAEQN", "series":
 "EQ", "open": 5670.05, "dayHigh": 5800, "dayLow": 5670.05, "lastPrice": 5760, "previousClose": 5668.85, "change": 91.15, "pChange": 1.61, "totalTradedVolume": 348441, "totalTradedValue": 2001779607.36, "
lastUpdateTime": "10-Jul-2024 13:04:56", "yearHigh": 5800, "ffmc": 678646423980, "yearLow": 4347.7, "nearWKH": 0.6896551724137931, "nearWKL": -32.48384203141892, "perChange365d": 12.66, "date365dAgo": "10
-Jul-2023", "chart365dPath": "https://nsearchives.nseindia.com/365d/BRITANNIA-EQ.svg", "date30dAgo": "07-Jun-2024", "perChange30d": 3.76, "chart30dPath": "https://nsearchives.nseindia.com/30d/BRITANNIA-EQ
.svg", "chartTodayPath": "https://nsearchives.nseindia.com/today/BRITANNIAEQN.svg", "meta": {"symbol": "BRITANNIA", "companyName": "Britannia Industries Limited", "industry": "FOOD AND FOOD PROCESSING", "
activeSeries": ["EQ"], "debtSeries": [], "isFNOSec": true, "isCASec": false, "isSLBSec": true, "isDebtSec": false, "isSuspended": false, "tempSuspendedSeries": ["N1", "N2"], "isETFSec": false, "isDelisted
": false, "isin": "INE216A07052", "isMunicipalBond": false, "quotepreopenstatus": {"equityTime": "10-Jul-2024 13:04:26", "preOpenTime": "10-Jul-2024 09:08:00", "QuotePreOpenFlag": false}}}, {"priority": 0
, "symbol": "POWERGRID", "identifier": "POWERGRIDEQN", "series": "EQ", "open": 343, "dayHigh": 347.8, "dayLow": 335.5, "lastPrice": 345.75, "previousClose": 341.15, "change": 4.6, "pChange": 1.35, "totalT
radedVolume": 7133957, "totalTradedValue": 2443380272.5, "lastUpdateTime": "10-Jul-2024 13:04:55", "yearHigh": 348.7, "ffmc": 1575457182711.87, "yearLow": 177.53, "nearWKH": 0.8459994264410636, "nearWKL":
 -94.75581591843633, "perChange365d": 36.46, "date365dAgo": "10-Jul-2023", "chart365dPath": "https://nsearchives.nseindia.com/365d/POWERGRID-EQ.svg", "date30dAgo": "07-Jun-2024", "perChange30d": 10.28, "c
hart30dPath": "https://nsearchives.nseindia.com/30d/POWERGRID-EQ.svg", "chartTodayPath": "https://nsearchives.nseindia.com/today/POWERGRIDEQN.svg", "meta": {"symbol": "POWERGRID", "companyName": "Power Gr
id Corporation of India Limited", "industry": "POWER", "activeSeries": ["EQ"], "debtSeries": [], "isFNOSec": true, "isCASec": false, "isSLBSec": true, "isDebtSec": true, "isSuspended": false, "tempSuspend
edSeries": ["IL"], "isETFSec": false, "isDelisted": false, "isin": "INE752E01010", "isMunicipalBond": false, "quotepreopenstatus": {"equityTime": "10-Jul-2024 13:04:26", "preOpenTime": "10-Jul-2024 09:08:
00", "QuotePreOpenFlag": false}}}, {"priority": 0, "symbol": "GRASIM", "identifier": "GRASIMEQN", "series": "EQ", "open": 2781, "dayHigh": 2812.25, "dayLow": 2741, "lastPrice": 2794.7, "previousClose": 27
61.8, "change": 32.9, "pChange": 1.19, "totalTradedVolume": 774465, "totalTradedValue": 2156002134.9, "lastUpdateTime": "10-Jul-2024 13:04:56", "yearHigh": 2812.25, "ffmc": 1031020432901.76, "yearLow": 17
33.25, "nearWKH": 0.6240554715974818, "nearWKL": -61.240444252127496, "perChange365d": 58.45, "date365dAgo": "10-Jul-2023", "chart365dPath": "https://nsearchives.nseindia.com/365d/GRASIM-EQ.svg", "date30d
Ago": "07-Jun-2024", "perChange30d": 16.05, "chart30dPath": "https://nsearchives.nseindia.com/30d/GRASIM-EQ.svg", "chartTodayPath": "https://nsearchives.nseindia.com/today/GRASIMEQN.svg", "meta": {"symbol
": "GRASIM", "companyName": "Grasim Industries Limited", "industry": "CEMENT AND CEMENT PRODUCTS", "activeSeries": ["EQ"], "debtSeries": [], "isFNOSec": true, "isCASec": false, "isSLBSec": true, "isDebtSe
c": false, "isSuspended": false, "tempSuspendedSeries": ["IL"], "isETFSec": false, "isDelisted": false, "isin": "INE047A01021", "isMunicipalBond": false, "quotepreopenstatus": {"equityTime": "10-Jul-2024
 13:04:26", "preOpenTime": "10-Jul-2024 09:08:00", "QuotePreOpenFlag": false}}}, {"priority": 0, "symbol": "DIVISLAB", "identifier": "DIVISLABEQN", "series": "EQ", "open": 4572, "dayHigh": 4635.55, "dayLow
": 4536.8, "lastPrice": 4605.3, "previousClose": 4551.95, "change": 53.35, "pChange": 1.17, "totalTradedVolume": 328400, "totalTradedValue": 1508193420, "lastUpdateTime": "10-Jul-2024 13:04:55", "yearHigh
": 4670, "ffmc": 587683723660.8, "yearLow": 3295.3, "nearWKH": 1.385438972162737, "nearWKL": -39.753588444147724, "perChange365d": 25.06, "date365dAgo": "10-Jul-2023", "chart365dPath": "https://nsearchive
s.nseindia.com/365d/DIVISLAB-EQ.svg", "date30dAgo": "07-Jun-2024", "perChange30d": 0.62, "chart30dPath": "https://nsearchives.nseindia.com/30d/DIVISLAB-EQ.svg", "chartTodayPath": "https://nsearchives.nsei
ndia.com/today/DIVISLABEQN.svg", "meta": {"symbol": "DIVISLAB", "companyName": "Divi's Laboratories Limited", "industry": "PHARMACEUTICALS", "activeSeries": ["EQ", "T0"], "debtSeries": [], "isFNOSec": tru
e, "isCASec": false, "isSLBSec": true, "isDebtSec": false, "isSuspended": false, "tempSuspendedSeries": [], "isETFSec": false, "isDelisted": false, "isin": "INE361B01024", "isMunicipalBond": false, "quote
preopenstatus": {"equityTime": "10-Jul-2024 13:04:26", "preOpenTime": "10-Jul-2024 09:08:00", "QuotePreOpenFlag": false}}}, {"priority": 0, "symbol": "MARUTI", "identifier": "MARUTIEQN", "series": "EQ", "
open": 12950.05, "dayHigh": 13300, "dayLow": 12828.05, "lastPrice": 12962.5, "previousClose": 12827.7, "change": 134.8, "pChange": 1.05, "totalTradedVolume": 1604580, "totalTradedValue": 20986911560.39999
8, "lastUpdateTime": "10-Jul-2024 13:04:55", "yearHigh": 13300, "ffmc": 1711494742331.93, "yearLow": 9254.15, "nearWKH": 2.537593984962406, "nearWKL": -40.07229189066528, "perChange365d": 31.75, "date365d
Ago": "10-Jul-2023", "chart365dPath": "https://nsearchives.nseindia.com/365d/MARUTI-EQ.svg", "date30dAgo": "07-Jun-2024", "perChange30d": 0.13, "chart30dPath": "https://nsearchives.nseindia.com/30d/MARUTI
-EQ.svg", "chartTodayPath": "https://nsearchives.nseindia.com/today/MARUTIEQN.svg", "meta": {"symbol": "MARUTI", "companyName": "Maruti Suzuki India Limited", "industry": "AUTOMOBILES - 4 WHEELERS", "acti
veSeries": ["EQ"], "debtSeries": [], "isFNOSec": true, "isCASec": false, "isSLBSec": true, "isDebtSec": false, "isSuspended": false, "tempSuspendedSeries": ["IL"], "isETFSec": false, "isDelisted": false,
"isin": "INE585B01010", "isMunicipalBond": false, "quotepreopenstatus": {"equityTime": "10-Jul-2024 13:04:26", "preOpenTime": "10-Jul-2024 09:08:00", "QuotePreOpenFlag": false}}}, {"priority": 0, "symbol"
: "APOLLOHOSP", "identifier": "APOLLOHOSPEQN", "series": "EQ", "open": 6320.3, "dayHigh": 6417, "dayLow": 6301.15, "lastPrice": 6379.85, "previousClose": 6320.3, "change": 59.55, "pChange": 0.94, "totalTr
adedVolume": 210819, "totalTradedValue": 1343570568.9, "lastUpdateTime": "10-Jul-2024 13:04:53", "yearHigh": 6874.45, "ffmc": 642655589387.49, "yearLow": 4726, "nearWKH": 7.194757398773712, "nearWKL": -34
```

kafka_to_hdfs.py - Real-time Stock Market - Visual Studio Code

File Edit Selection View Go Run Terminal Help

EXPLORER

OUTPUT

REAL-TIME STOCK MARKET

__pycache__

.venv

copy_into_local.py

fetch_data.py

kafka_producer.py

kafka_to_hdfs.py

stock_prices.csv

OUTLINE

TIMELINE

DEBUG CONSOLE

Filter (e.g. text, !exclude, \escape)

fetch_data.py | kafka_to_hdfs.py | copy_into_local.py | stock_prices.csv | kafka.py | kafka_producer.py

kafka_to_hdfs.py > ...

```
39          exploded_df.createOrReplaceTempView("stocks")
40
41          # Run the SQL query to calculate average of 'lastPrice'
42          result = spark.sql("SELECT data.symbol, AVG(CAST(data.lastPrice AS FLOAT)) AS avg_price FROM stocks GROUP BY data.symbol")
43          result.show()
44          consumer.close()
45          spark.stop()
46
47  except KeyboardInterrupt:
48      print("Interrupted by user")
49  except Exception as e:
50      print(f"Error: {e}")
51  finally:
52      consumer.close()
53      spark.stop()  # Stop Spark session gracefully
54
```

PROBLEMS TERMINAL PORTS

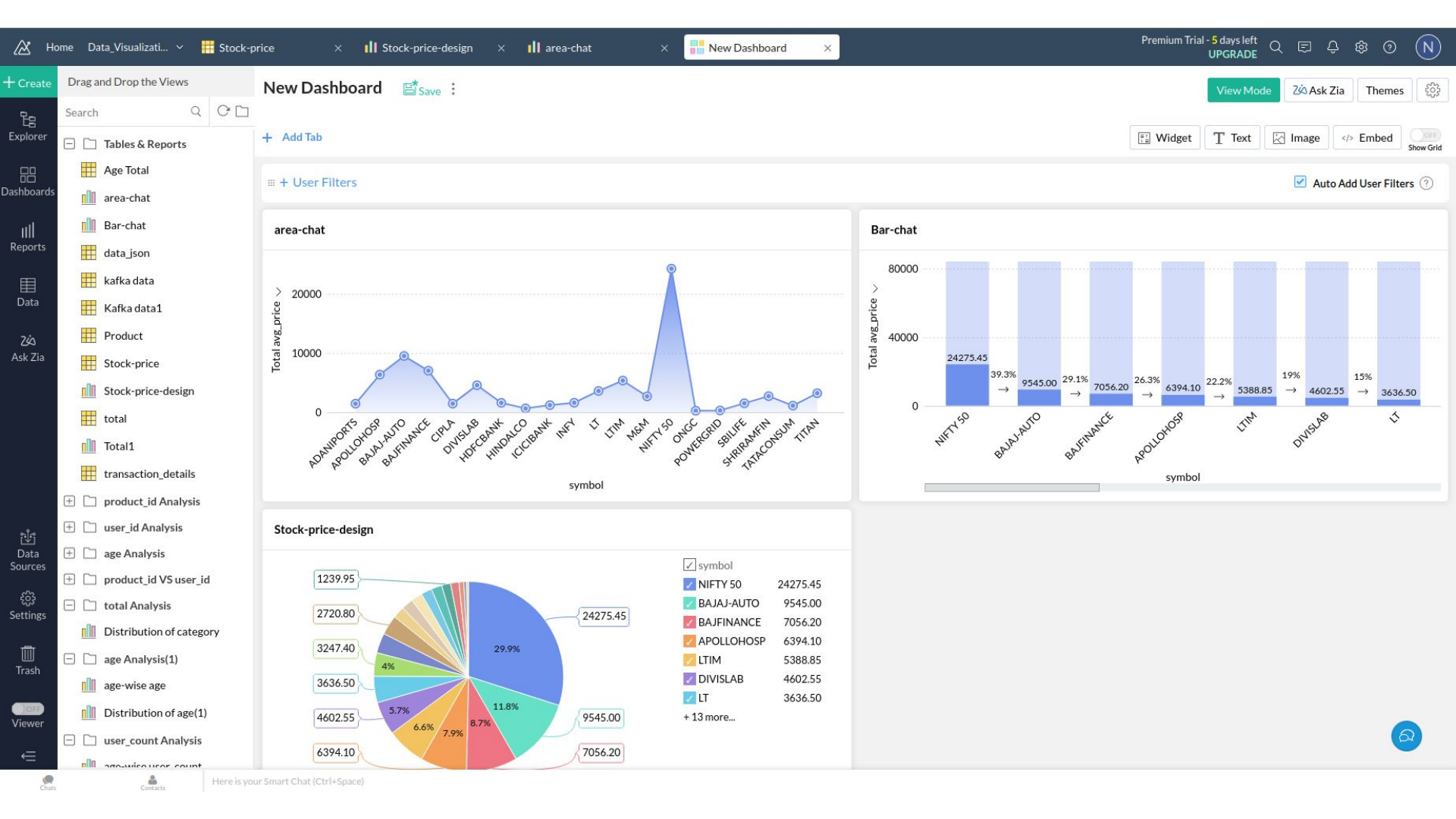
bash

```
24/07/10 12:47:36 ERROR Executor: Exception in task 2.0 in stage 73.0 (TID 168): Connection reset
(.venv) nandhumidhun@nandhumidhun-HP-Laptop-15-bs0xx:~/Real-time Stock Market$ python3 kafka_to_hdfs.py
24/07/10 12:49:33 WARN Utils: Your hostname, nandhumidhun-HP-Laptop-15-bs0xx resolves to a loopback address: 127.0.1.1; using 192.168.1.38 instead (on interface wlo1)
24/07/10 12:49:33 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
24/07/10 12:49:35 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Starting to consume messages...
24/07/10 12:49:46 WARN SparkStringUtils: Truncated the string representation of a plan since it was too large. This behavior can be adjusted by setting 'spark.sql.debug.maxToStringFields'.
+----------+-----------------+
|    symbol|        avg_price|
+----------+-----------------+
|        LT| 3638.550048828125|
| ICICIBANK|          1238.75|
|     TITAN| 3234.300048828125|
|SHRIRAMFIN|  2757.89990234375|
|  NIFTY 50|   24275.900390625|
|  DIVISLAB|  4622.89990234375|
|  HDFCBANK|1626.8499755859375|
|     CIPLA|1504.8499755859375|
|   SBILIFE|           1543.5|
|  HINDALCO|            690.5|
|BAJAJ-AUTO|   9515.7998046875|
|      INFY|          1646.75|
|       M&M| 2728.449951171875|
| POWERGRID|345.95001220703125|
|      ONGC|            294.0|
```

Do you want to install the recommended 'Rainbow CSV' extension from mechatroner for stock_prices.csv?

Install Show Recommendations

Ln 45, Col 9 Spaces: 4 UTF-8 LF Python 3.10.12 ('.venv': venv)


Home
Data_Visualizati...
Stock-price
Stock-price-design
area-chat
New Dashboard
Premium Trial - 5 days left
UPGRADE
Create
Drag and Drop the Views
Search
Tables & Reports
Age Total
area-chat
Bar-chat
data_json
kafka data
Kafka data1
Product
Stock-price
Stock-price-design
total
Total1
transaction_details
product_id Analysis
user_id Analysis
age Analysis
product_id VS user_id
total Analysis
Distribution of category
age Analysis(1)
age-wise age
Distribution of age(1)
user_count Analysis
Explorer
Dashboards
Reports
Data
Ask Zia
Data Sources
Settings
Trash
Viewer
New Dashboard
Save
View Mode
Ask Zia
Themes
Add Tab
Widget
Text
Image
Embed
Show Grid
+ User Filters
Auto Add User Filters
area-chat
Total avg_price
20000
10000
0
ADANIPORTS
APOLLOHOSP
BAJAJ-AUTO
BAJFINANCE
CIPLA
DIVISLAB
HDFCBANK
HINDALCO
ICICIBANK
INFY
LT
LTIM
M&M
NIFTY 50
ONGC
POWERGRID
SBILIFE
SHRIRAMFIN
TATACONSUM
TITAN
symbol
Bar-chat
Total avg_price
80000
40000
0
24275.45
39.3%
9545.00
29.1%
7056.20
26.3%
6394.10
22.2%
5388.85
19%
4602.55
15%
3636.50
NIFTY 50
BAJAJ-AUTO
BAJFINANCE
APOLLOHOSP
LTIM
DIVISLAB
LT
symbol
Stock-price-design
1239.95
2720.80
3247.40
3636.50
4602.55
6394.10
24275.45
9545.00
7056.20
29.9%
11.8%
8.7%
7.9%
6.6%
5.7%
4%
symbol
NIFTY 50 24275.45
BAJAJ-AUTO 9545.00
BAJFINANCE 7056.20
APOLLOHOSP 6394.10
LTIM 5388.85
DIVISLAB 4602.55
LT 3636.50
+ 13 more...
Chats
Contacts
Here is your Smart Chat (Ctrl+Space)

PREDICTIVE MAINTENANCE FOR MANUFACTURING EQUIPMENT PROCESS

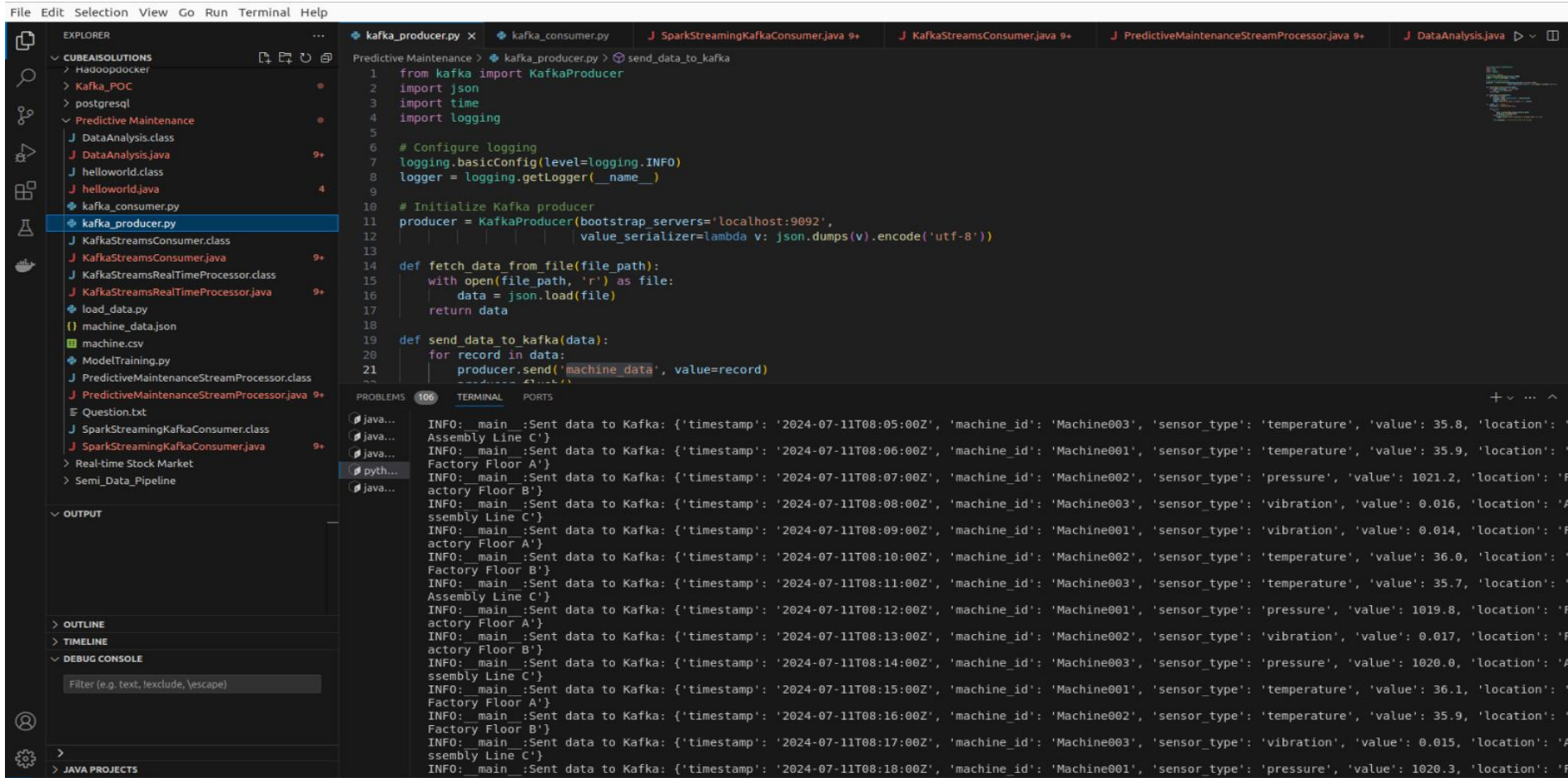
STEP 1: Data Ingestion and ETL with Apache NiFi

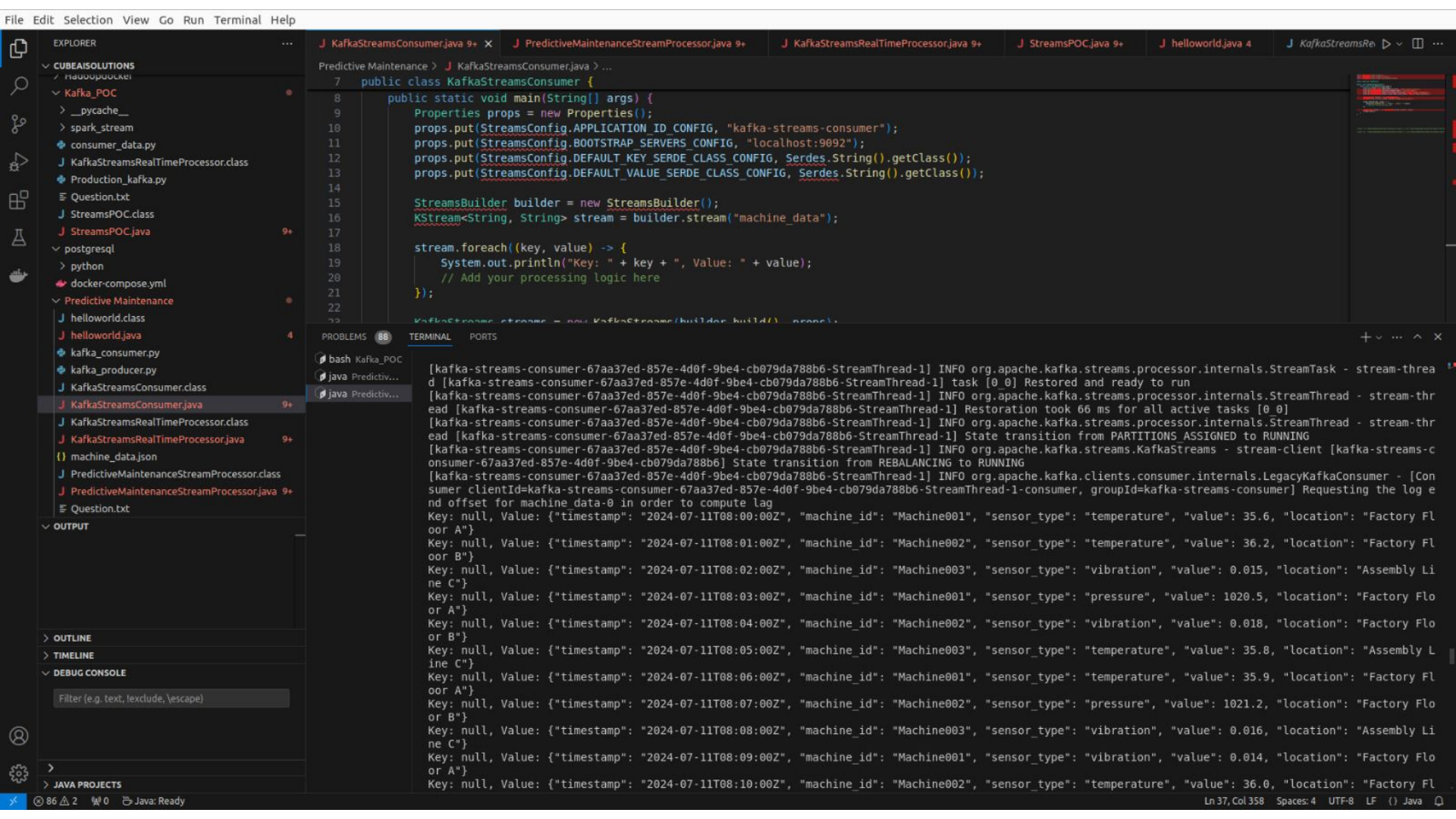
STEP 2: Data Integration with Apache Kafka

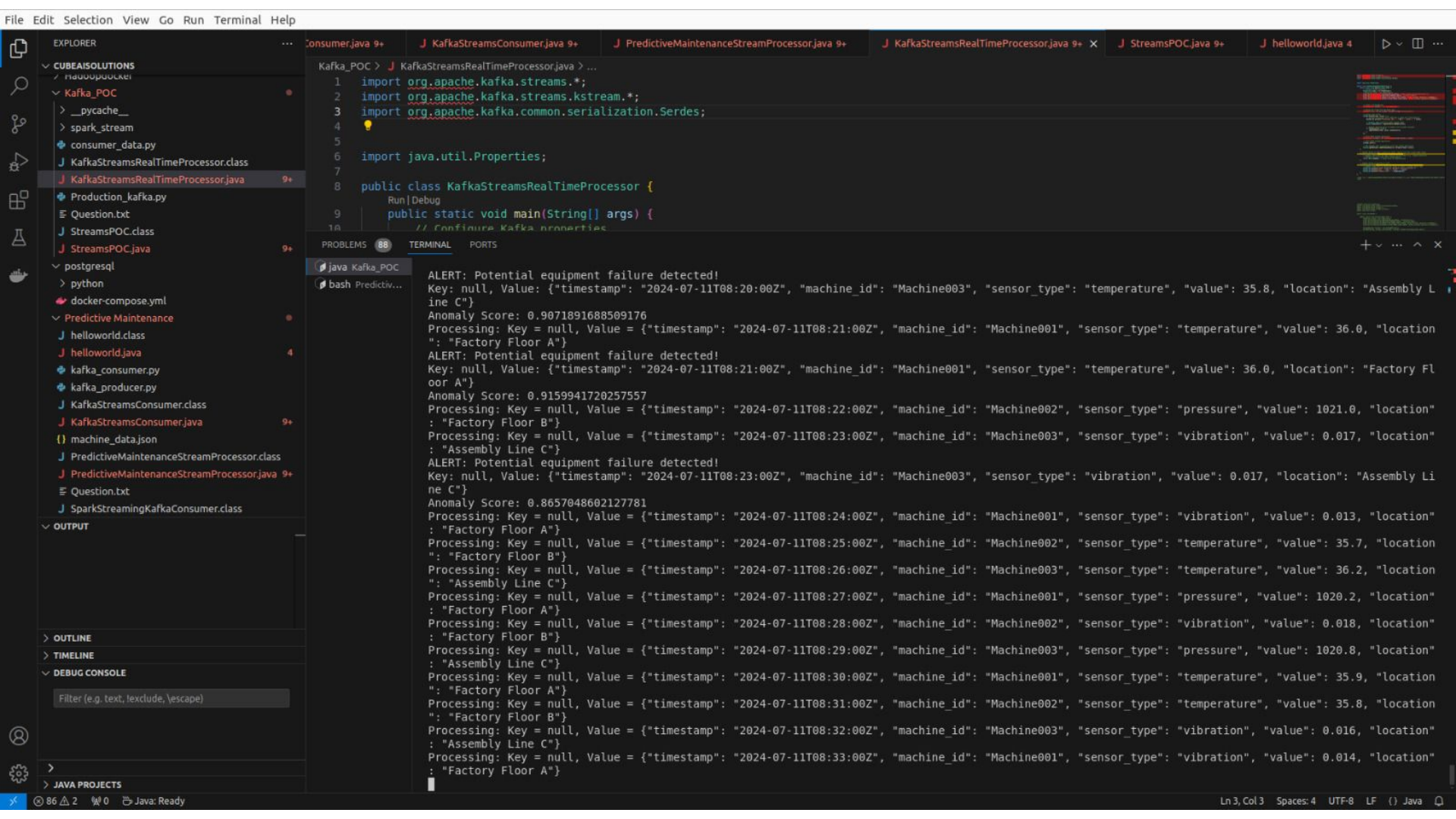
STEP 3: Batch Processing with Apache Spark

STEP 4: Stream Processing with Apache Kafka Streams

PREDICTIVE MAINTENANCE FOR MANUFACTURING EQUIPMENT




File Edit Selection View Go Run Terminal Help
EXPLORER
CUBEAISOLUTIONS
Kafka_POC
__pycache__
spark_stream
consumer_data.py
KafkaStreamsRealTimeProcessor.class
Production_kafka.py
Question.txt
StreamsPOC.class
StreamsPOC.java 9+
postgresql
python
docker-compose.yml
Predictive Maintenance
helloworld.class
helloworld.java 4
kafka_consumer.py
kafka_producer.py
KafkaStreamsConsumer.class
KafkaStreamsConsumer.java 9+
KafkaStreamsRealTimeProcessor.class
KafkaStreamsRealTimeProcessor.java 9+
machine_data.json
PredictiveMaintenanceStreamProcessor.class
PredictiveMaintenanceStreamProcessor.java 9+
Question.txt
OUTPUT
OUTLINE
TIMELINE
DEBUG CONSOLE
Filter (e.g. text, !exclude, \escape)
JAVA PROJECTS
KafkaStreamsConsumer.java 9+
PredictiveMaintenanceStreamProcessor.java 9+
KafkaStreamsRealTimeProcessor.java 9+
StreamsPOC.java 9+
helloworld.java 4
Predictive Maintenance > KafkaStreamsConsumer.java > ...
7 public class KafkaStreamsConsumer {
8 public static void main(String[] args) {
9 Properties props = new Properties();
10 props.put(StreamsConfig.APPLICATION_ID_CONFIG, "kafka-streams-consumer");
11 props.put(StreamsConfig.BOOTSTRAP_SERVERS_CONFIG, "localhost:9092");
12 props.put(StreamsConfig.DEFAULT_KEY_SERDE_CLASS_CONFIG, Serdes.String().getClass());
13 props.put(StreamsConfig.DEFAULT_VALUE_SERDE_CLASS_CONFIG, Serdes.String().getClass());
14
15 StreamsBuilder builder = new StreamsBuilder();
16 KStream<String, String> stream = builder.stream("machine_data");
17
18 stream.foreach((key, value) -> {
19 System.out.println("Key: " + key + ", Value: " + value);
20 // Add your processing logic here
21 });
22
PROBLEMS 88
TERMINAL
PORTS
bash Kafka_POC
java Predictiv...
java Predictiv...
[kafka-streams-consumer-67aa37ed-857e-4d0f-9be4-cb079da788b6-StreamThread-1] INFO org.apache.kafka.streams.processor.internals.StreamTask - stream-thread [kafka-streams-consumer-67aa37ed-857e-4d0f-9be4-cb079da788b6-StreamThread-1] task [0_0] Restored and ready to run
[kafka-streams-consumer-67aa37ed-857e-4d0f-9be4-cb079da788b6-StreamThread-1] INFO org.apache.kafka.streams.processor.internals.StreamThread - stream-thread [kafka-streams-consumer-67aa37ed-857e-4d0f-9be4-cb079da788b6-StreamThread-1] Restoration took 66 ms for all active tasks [0_0]
[kafka-streams-consumer-67aa37ed-857e-4d0f-9be4-cb079da788b6-StreamThread-1] INFO org.apache.kafka.streams.processor.internals.StreamThread - stream-thread [kafka-streams-consumer-67aa37ed-857e-4d0f-9be4-cb079da788b6-StreamThread-1] State transition from PARTITIONS_ASSIGNED to RUNNING
[kafka-streams-consumer-67aa37ed-857e-4d0f-9be4-cb079da788b6-StreamThread-1] INFO org.apache.kafka.streams.KafkaStreams - stream-client [kafka-streams-consumer-67aa37ed-857e-4d0f-9be4-cb079da788b6] State transition from REBALANCING to RUNNING
[kafka-streams-consumer-67aa37ed-857e-4d0f-9be4-cb079da788b6-StreamThread-1] INFO org.apache.kafka.clients.consumer.internals.LegacyKafkaConsumer - [Consumer clientId=kafka-streams-consumer-67aa37ed-857e-4d0f-9be4-cb079da788b6-StreamThread-1-consumer, groupId=kafka-streams-consumer] Requesting the log end offset for machine_data-0 in order to compute lag
Key: null, Value: {"timestamp": "2024-07-11T08:00:00Z", "machine_id": "Machine001", "sensor_type": "temperature", "value": 35.6, "location": "Factory Floor A"}
Key: null, Value: {"timestamp": "2024-07-11T08:01:00Z", "machine_id": "Machine002", "sensor_type": "temperature", "value": 36.2, "location": "Factory Floor B"}
Key: null, Value: {"timestamp": "2024-07-11T08:02:00Z", "machine_id": "Machine003", "sensor_type": "vibration", "value": 0.015, "location": "Assembly Line C"}
Key: null, Value: {"timestamp": "2024-07-11T08:03:00Z", "machine_id": "Machine001", "sensor_type": "pressure", "value": 1020.5, "location": "Factory Floor A"}
Key: null, Value: {"timestamp": "2024-07-11T08:04:00Z", "machine_id": "Machine002", "sensor_type": "vibration", "value": 0.018, "location": "Factory Floor B"}
Key: null, Value: {"timestamp": "2024-07-11T08:05:00Z", "machine_id": "Machine003", "sensor_type": "temperature", "value": 35.8, "location": "Assembly Line C"}
Key: null, Value: {"timestamp": "2024-07-11T08:06:00Z", "machine_id": "Machine001", "sensor_type": "temperature", "value": 35.9, "location": "Factory Floor A"}
Key: null, Value: {"timestamp": "2024-07-11T08:07:00Z", "machine_id": "Machine002", "sensor_type": "pressure", "value": 1021.2, "location": "Factory Floor B"}
Key: null, Value: {"timestamp": "2024-07-11T08:08:00Z", "machine_id": "Machine003", "sensor_type": "vibration", "value": 0.016, "location": "Assembly Line C"}
Key: null, Value: {"timestamp": "2024-07-11T08:09:00Z", "machine_id": "Machine001", "sensor_type": "vibration", "value": 0.014, "location": "Factory Floor A"}
Key: null, Value: {"timestamp": "2024-07-11T08:10:00Z", "machine_id": "Machine002", "sensor_type": "temperature", "value": 36.0, "location": "Factory Fl
86 2 0 Java: Ready
Ln 37, Col 358 Spaces: 4 UTF-8 LF Java


File Edit Selection View Go Run Terminal Help
EXPLORER
CUBEAISOLUTIONS
Kafka_POC
__pycache__
spark_stream
consumer_data.py
KafkaStreamsRealTimeProcessor.class
KafkaStreamsRealTimeProcessor.java 9+
Production_kafka.py
Question.txt
StreamsPOC.class
StreamsPOC.java 9+
postgresql
python
docker-compose.yml
Predictive Maintenance
helloworld.class
helloworld.java 4
kafka_consumer.py
kafka_producer.py
KafkaStreamsConsumer.class
KafkaStreamsConsumer.java 9+
machine_data.json
PredictiveMaintenanceStreamProcessor.class
PredictiveMaintenanceStreamProcessor.java 9+
Question.txt
SparkStreamingKafkaConsumer.class
OUTPUT
OUTLINE
TIMELINE
DEBUG CONSOLE
Filter (e.g. text, !exclude, \escape)
JAVA PROJECTS
KafkaStreamsConsumer.java 9+
PredictiveMaintenanceStreamProcessor.java 9+
KafkaStreamsRealTimeProcessor.java 9+
StreamsPOC.java 9+
helloworld.java 4
Kafka_POC > KafkaStreamsRealTimeProcessor.java > ...
1 import org.apache.kafka.streams.*;
2 import org.apache.kafka.streams.kstream.*;
3 import org.apache.kafka.common.serialization.Serdes;
4
5
6 import java.util.Properties;
7
8 public class KafkaStreamsRealTimeProcessor {
Run | Debug
9 public static void main(String[] args) {
// Configure Kafka properties
PROBLEMS 88
TERMINAL
PORTS
java Kafka_POC
bash Predictiv...
ALERT: Potential equipment failure detected!
Key: null, Value: {"timestamp": "2024-07-11T08:20:00Z", "machine_id": "Machine003", "sensor_type": "temperature", "value": 35.8, "location": "Assembly Line C"}
Anomaly Score: 0.9071891688509176
Processing: Key = null, Value = {"timestamp": "2024-07-11T08:21:00Z", "machine_id": "Machine001", "sensor_type": "temperature", "value": 36.0, "location": "Factory Floor A"}
ALERT: Potential equipment failure detected!
Key: null, Value: {"timestamp": "2024-07-11T08:21:00Z", "machine_id": "Machine001", "sensor_type": "temperature", "value": 36.0, "location": "Factory Floor A"}
Anomaly Score: 0.9159941720257557
Processing: Key = null, Value = {"timestamp": "2024-07-11T08:22:00Z", "machine_id": "Machine002", "sensor_type": "pressure", "value": 1021.0, "location": "Factory Floor B"}
Processing: Key = null, Value = {"timestamp": "2024-07-11T08:23:00Z", "machine_id": "Machine003", "sensor_type": "vibration", "value": 0.017, "location": "Assembly Line C"}
ALERT: Potential equipment failure detected!
Key: null, Value: {"timestamp": "2024-07-11T08:23:00Z", "machine_id": "Machine003", "sensor_type": "vibration", "value": 0.017, "location": "Assembly Line C"}
Anomaly Score: 0.8657048602127781
Processing: Key = null, Value = {"timestamp": "2024-07-11T08:24:00Z", "machine_id": "Machine001", "sensor_type": "vibration", "value": 0.013, "location": "Factory Floor A"}
Processing: Key = null, Value = {"timestamp": "2024-07-11T08:25:00Z", "machine_id": "Machine002", "sensor_type": "temperature", "value": 35.7, "location": "Factory Floor B"}
Processing: Key = null, Value = {"timestamp": "2024-07-11T08:26:00Z", "machine_id": "Machine003", "sensor_type": "temperature", "value": 36.2, "location": "Assembly Line C"}
Processing: Key = null, Value = {"timestamp": "2024-07-11T08:27:00Z", "machine_id": "Machine001", "sensor_type": "pressure", "value": 1020.2, "location": "Factory Floor A"}
Processing: Key = null, Value = {"timestamp": "2024-07-11T08:28:00Z", "machine_id": "Machine002", "sensor_type": "vibration", "value": 0.018, "location": "Factory Floor B"}
Processing: Key = null, Value = {"timestamp": "2024-07-11T08:29:00Z", "machine_id": "Machine003", "sensor_type": "pressure", "value": 1020.8, "location": "Assembly Line C"}
Processing: Key = null, Value = {"timestamp": "2024-07-11T08:30:00Z", "machine_id": "Machine001", "sensor_type": "temperature", "value": 35.9, "location": "Factory Floor A"}
Processing: Key = null, Value = {"timestamp": "2024-07-11T08:31:00Z", "machine_id": "Machine002", "sensor_type": "temperature", "value": 35.8, "location": "Factory Floor B"}
Processing: Key = null, Value = {"timestamp": "2024-07-11T08:32:00Z", "machine_id": "Machine003", "sensor_type": "vibration", "value": 0.016, "location": "Assembly Line C"}
Processing: Key = null, Value = {"timestamp": "2024-07-11T08:33:00Z", "machine_id": "Machine001", "sensor_type": "vibration", "value": 0.014, "location": "Factory Floor A"}
Ln 3, Col 3
Spaces: 4
UTF-8
LF
Java

PredictiveMaintenanceStreamProcessor.java - CUBEAISOLUTIONS - Visual Studio Code

File Edit Selection View Go Run Terminal Help

EXPLORER

CUBEAISOLUTIONS

Predictive Maintenance

- DataAnalysis.java 9+
- helloworld.class
- helloworld.java 4
- kafka_producer.py
- KafkaStreamsConsumer.class
- KafkaStreamsConsumer.java 9+
- KafkaStreamsRealTimeProcessor.class
- KafkaStreamsRealTimeProcessor.java 9+
- load_data.py
- machine_data.json
- machine.csv
- ModelTraining.py
- PredictiveMaintenanceStreamProcessor.class
- PredictiveMaintenanceStreamProcessor.java 9+
- PredictiveMaintenanceStreamProcessor$Tempe...
- PredictiveMaintenanceStreamProcessor$Tempe...
- PredictiveMaintenanceStreamProcessor$Tempe...
- PredictiveMaintenanceStreamProcessor$Tempe...
- Question.txt
- SparkStreamingKafkaConsumer.class

OUTPUT

OUTLINE

TIMELINE

DEBUG CONSOLE

Filter (e.g. text, !exclude, \escape)

JAVA PROJECTS

PredictiveMaintenanceStreamProcessor.java 9+ | SparkStreamingKafkaConsumer.class | DataAnalysis.java 9+ | load_data.py | ModelTraining.py | KafkaStreamsRealTimeProcessor.java 9

Predictive Maintenance > PredictiveMaintenanceStreamProcessor.java > PredictiveMaintenanceStreamProcessor > main(String[])

```
15  public class PredictiveMaintenanceStreamProcessor {
16      public static void main(String[] args) {
36              .windowedBy(TimeWindows.ofSizeWithNoGrace(Duration.ofMinutes(5)))
37              .aggregate(
38                  TemperatureAggregator::new, // Initializer
39                  (key, newValue, aggregator) -> aggregator.add(newValue), // Adder
40                  Materialized.<String, TemperatureAggregator, WindowStore<Bytes, byte[]>>as("temp-avg-store")
41                      .withValueSerde(new TemperatureAggregatorSerde()) // Serde for TemperatureAggregator
42              )
43              .mapValues(TemperatureAggregator::average); // Calculate the average
44
45
46          //System.out.println(value);
47
48          KafkaStreams streams = new KafkaStreams(builder.build(), props);
49          streams.start();
50
51          Runtime.getRuntime().addShutdownHook(new Thread(streams::close));
52      }
53
54      // Custom class to handle temperature aggregation
55      static class TemperatureAggregator {
56          private double sum = 0.0;
57          private int count = 0;
58
```

PROBLEMS 123 | TERMINAL | PORTS

- java Predictive Main...
- python3 Predictive ...
- java Predictive M...
- bash Predictive M...
- java Predictive Main...
- bash Predictive Mai...

aver — No results

```
Received record: key=null, value={"timestamp": "2024-07-11T08:27:00Z", "machine_id": "Machine001",
tion": "Factory Floor A"}
Received record: key=null, value={"timestamp": "2024-07-11T08:28:00Z", "machine_id": "Machine002", "sensor_type": "vibration", "value": 0.018, "loca
tion": "Factory Floor B"}
Received record: key=null, value={"timestamp": "2024-07-11T08:29:00Z", "machine_id": "Machine003", "sensor_type": "pressure", "value": 1020.8, "loca
tion": "Assembly Line C"}
Received record: key=null, value={"timestamp": "2024-07-11T08:30:00Z", "machine_id": "Machine001", "sensor_type": "temperature", "value": 35.9, "loc
ation": "Factory Floor A"}
Received record: key=null, value={"timestamp": "2024-07-11T08:31:00Z", "machine_id": "Machine002", "sensor_type": "temperature", "value": 35.8, "loc
ation": "Factory Floor B"}
Received record: key=null, value={"timestamp": "2024-07-11T08:32:00Z", "machine_id": "Machine003", "sensor_type": "vibration", "value": 0.016, "loca
tion": "Assembly Line C"}
Received record: key=null, value={"timestamp": "2024-07-11T08:33:00Z", "machine_id": "Machine001", "sensor_type": "vibration", "value": 0.014, "loca
tion": "Factory Floor A"}
Received record: key=null, value={"timestamp": "2024-07-11T08:35:00Z", "machine_id": "Machine001", "sensor_type": "vibration", "value": 0.014, "loca
tion": "Factory Floor A"}
Received record: key=null, value={"timestamp": "2024-07-11T08:32:00Z", "machine_id": "Machine003", "sensor_type": "vibration", "value": 0.016, "loca
tion": "Assembly Line C"}
Received record: key=null, value={"timestamp": "2024-07-11T08:33:00Z", "machine_id": "Machine001", "sensor_type": "vibration", "value": 0.014, "loca
tion": "Factory Floor A"}
Received record: key=null, value={"timestamp": "2024-07-11T08:35:00Z", "machine_id": "Machine001", "sensor_type": "vibration", "value": 0.014, "loca
tion": "Factory Floor A"}
```

121 ⚠ 2 | 0 | Java: Ready | Ln 45, Col 9 | Spaces: 4 | UTF-8 | LF | Java

File Edit Selection View Go Run Terminal Help

EXPLORER

CUBEAISOLUTIONS

- E-Commerce Transaction
- ETL
- Hadoopdocker
- Kafka_POC
- postgresql
- Predictive Maintenance
  - DataAnalysis.class
  - DataAnalysis.java 9+
  - helloworld.class
  - helloworld.java 4
  - kafka_consumer.py
  - kafka_producer.py
  - KafkaStreamsConsumer.class
  - KafkaStreamsConsumer.java 9+
  - KafkaStreamsRealTimeProcessor.class
  - KafkaStreamsRealTimeProcessor.java 9+
  - load_data.py
  - machine_data.json
  - machine.csv
  - ModelTraining.java 9+
  - PredictiveMaintenanceStreamProcessor.class
  - PredictiveMaintenanceStreamProcessor.java 9+
  - Question.txt
  - SparkStreamingKafkaConsumer.class
  - SparkStreamingKafkaConsumer.java 9+
- Real-time Stock Market

OUTPUT

OUTLINE

TIMELINE

DEBUG CONSOLE

Filter (e.g. text, !exclude, \escape)

JAVA PROJECTS

aStreamsConsumer.java 9+ | PredictiveMaintenanceStreamProcessor.java 9+ | DataAnalysis.java 9+ | load_data.py | machine.csv | ModelTraining.java 9+ | KafkaStreamsRealTimePro

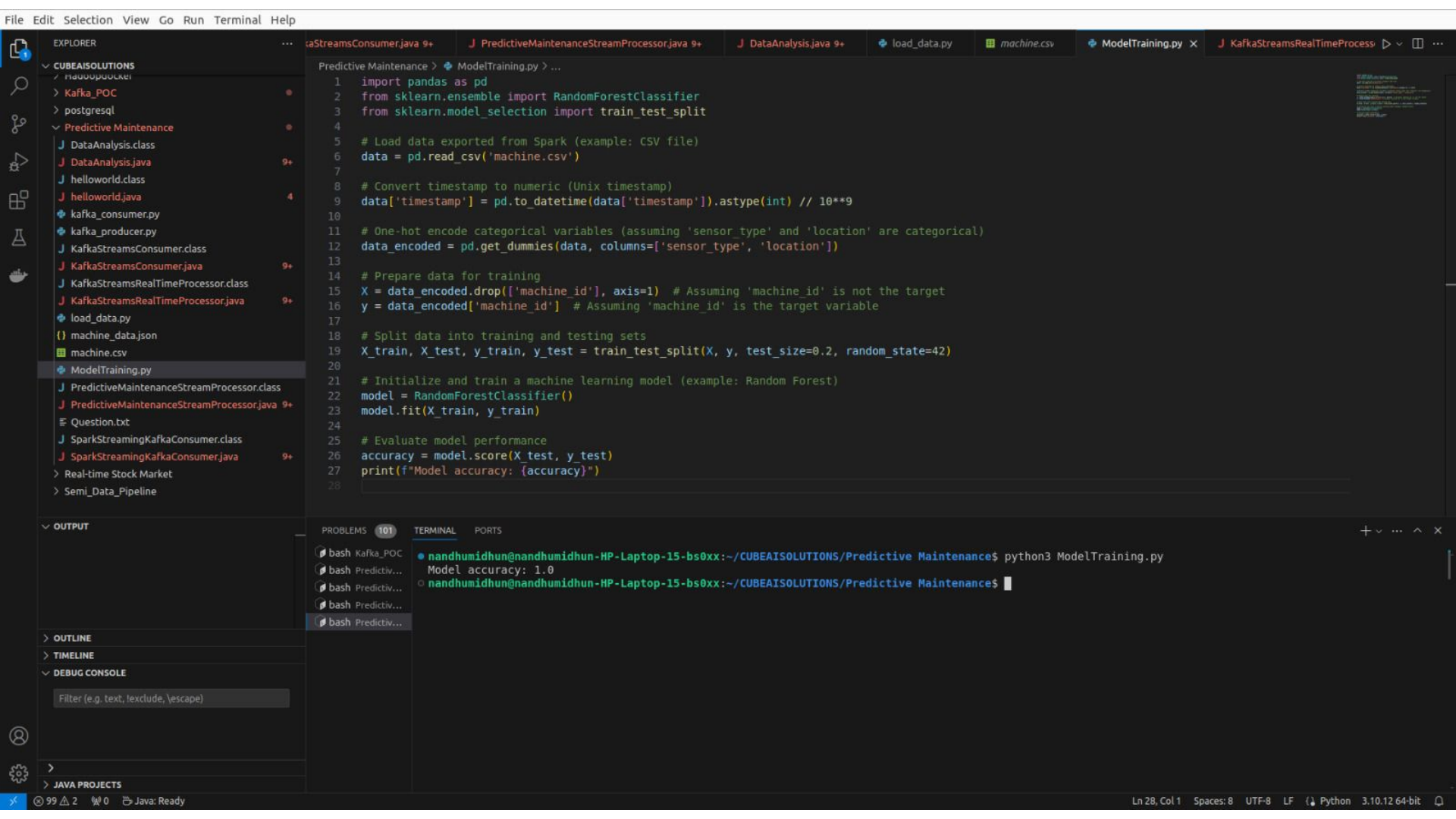
Predictive Maintenance > DataAnalysis.java > ...

```
6  public class DataAnalysis {
7      public static void main(String[] args) {
23
24         // Show the aggregated results
25         aggregatedData.show();
26
27         spark.stop();
28     }
29 }
30
31
32
33
34 javac -cp "/home/nandhumidhun/spark/jars/*" DataAnalysis.java
35 //java --add-exports java.base/sun.nio.ch=ALL-UNNAMED -cp "/home/nandhumidhun/spark/jars/*:." DataAnalysis
```

PROBLEMS 131 | TERMINAL | PORTS

bash Kafka_POC
bash Predictiv...
bash Predictiv...
bash Predictiv...
bash Predictiv...

```
24/07/11 17:31:08 INFO TaskSchedulerImpl: Removed TaskSet 3.0, whose tasks have all completed, from pool
24/07/11 17:31:08 INFO DAGScheduler: ResultStage 3 (show at DataAnalysis.java:25) finished in 0.187 s
24/07/11 17:31:08 INFO DAGScheduler: Job 2 is finished. Cancelling potential speculative or zombie tasks for this job
24/07/11 17:31:08 INFO TaskSchedulerImpl: Killing all running tasks in stage 3: Stage finished
24/07/11 17:31:08 INFO DAGScheduler: Job 2 finished: show at DataAnalysis.java:25, took 0.222068 s
24/07/11 17:31:08 INFO CodeGenerator: Code generated in 18.278092 ms
+----------+------------------+-------------+
|machine_id|   avg_temperature|avg_vibration|
+----------+------------------+-------------+
|Machine003|198.57990909090913|       0.0158|
|Machine001|327.71961538461534|      0.01375|
|Machine002|201.98809090909097|      0.01725|
+----------+------------------+-------------+

24/07/11 17:31:08 INFO SparkContext: SparkContext is stopping with exitCode 0.
24/07/11 17:31:08 INFO SparkUI: Stopped Spark web UI at http://192.168.1.38:4040
24/07/11 17:31:09 INFO MapOutputTrackerMasterEndpoint: MapOutputTrackerMasterEndpoint stopped!
24/07/11 17:31:09 INFO MemoryStore: MemoryStore cleared
```


File Edit Selection View Go Run Terminal Help
EXPLORER
CUBEAISOLUTIONS
Kafka_POC
postgresql
Predictive Maintenance
DataAnalysis.class
DataAnalysis.java
helloworld.class
helloworld.java
kafka_consumer.py
kafka_producer.py
KafkaStreamsConsumer.class
KafkaStreamsConsumer.java
KafkaStreamsRealTimeProcessor.class
KafkaStreamsRealTimeProcessor.java
load_data.py
machine_data.json
machine.csv
ModelTraining.py
PredictiveMaintenanceStreamProcessor.class
PredictiveMaintenanceStreamProcessor.java
Question.txt
SparkStreamingKafkaConsumer.class
SparkStreamingKafkaConsumer.java
Real-time Stock Market
Semi_Data_Pipeline
aStreamsConsumer.java 9+
PredictiveMaintenanceStreamProcessor.java 9+
DataAnalysis.java 9+
load_data.py
machine.csv
ModelTraining.py
KafkaStreamsRealTimeProcess
Predictive Maintenance > ModelTraining.py > ...
1 import pandas as pd
2 from sklearn.ensemble import RandomForestClassifier
3 from sklearn.model_selection import train_test_split
4
5 # Load data exported from Spark (example: CSV file)
6 data = pd.read_csv('machine.csv')
7
8 # Convert timestamp to numeric (Unix timestamp)
9 data['timestamp'] = pd.to_datetime(data['timestamp']).astype(int) // 10**9
10
11 # One-hot encode categorical variables (assuming 'sensor_type' and 'location' are categorical)
12 data_encoded = pd.get_dummies(data, columns=['sensor_type', 'location'])
13
14 # Prepare data for training
15 X = data_encoded.drop(['machine_id'], axis=1) # Assuming 'machine_id' is not the target
16 y = data_encoded['machine_id'] # Assuming 'machine_id' is the target variable
17
18 # Split data into training and testing sets
19 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
20
21 # Initialize and train a machine learning model (example: Random Forest)
22 model = RandomForestClassifier()
23 model.fit(X_train, y_train)
24
25 # Evaluate model performance
26 accuracy = model.score(X_test, y_test)
27 print(f"Model accuracy: {accuracy}")
OUTPUT
PROBLEMS 101
TERMINAL
PORTS
bash Kafka_POC
bash Predictiv...
bash Predictiv...
bash Predictiv...
bash Predictiv...
nandhumidhun@nandhumidhun-HP-Laptop-15-bs0xx:~/CUBEAISOLUTIONS/Predictive Maintenance$ python3 ModelTraining.py
Model accuracy: 1.0
nandhumidhun@nandhumidhun-HP-Laptop-15-bs0xx:~/CUBEAISOLUTIONS/Predictive Maintenance$
OUTLINE
TIMELINE
DEBUG CONSOLE
Filter (e.g. text, !exclude, \escape)
JAVA PROJECTS
99 2
0
Java: Ready
Ln 28, Col 1
Spaces: 8
UTF-8
LF
Python
3.10.12 64-bit

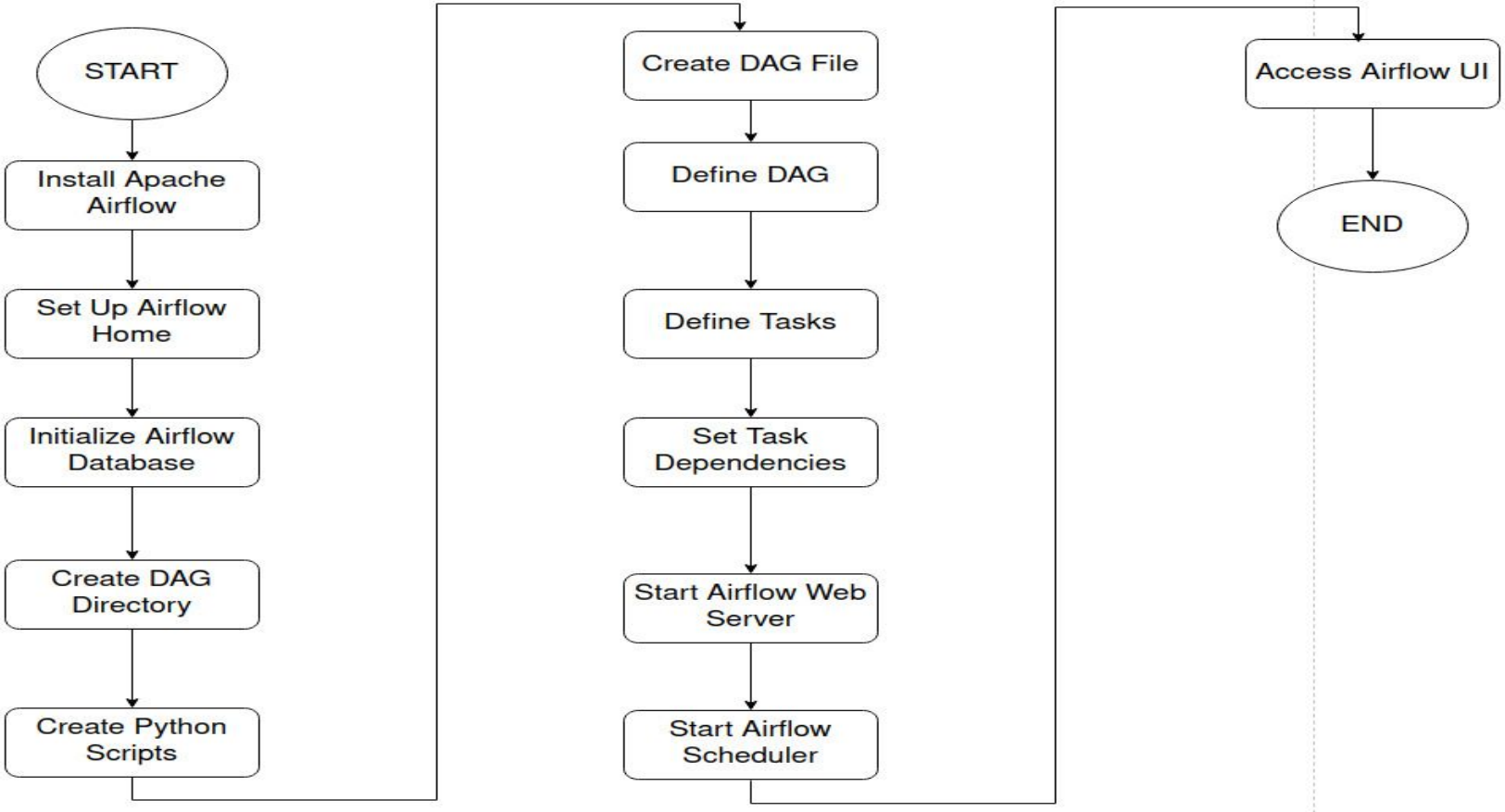
EXECUTE DAILY PYTHON SCRIPTS PROCESS

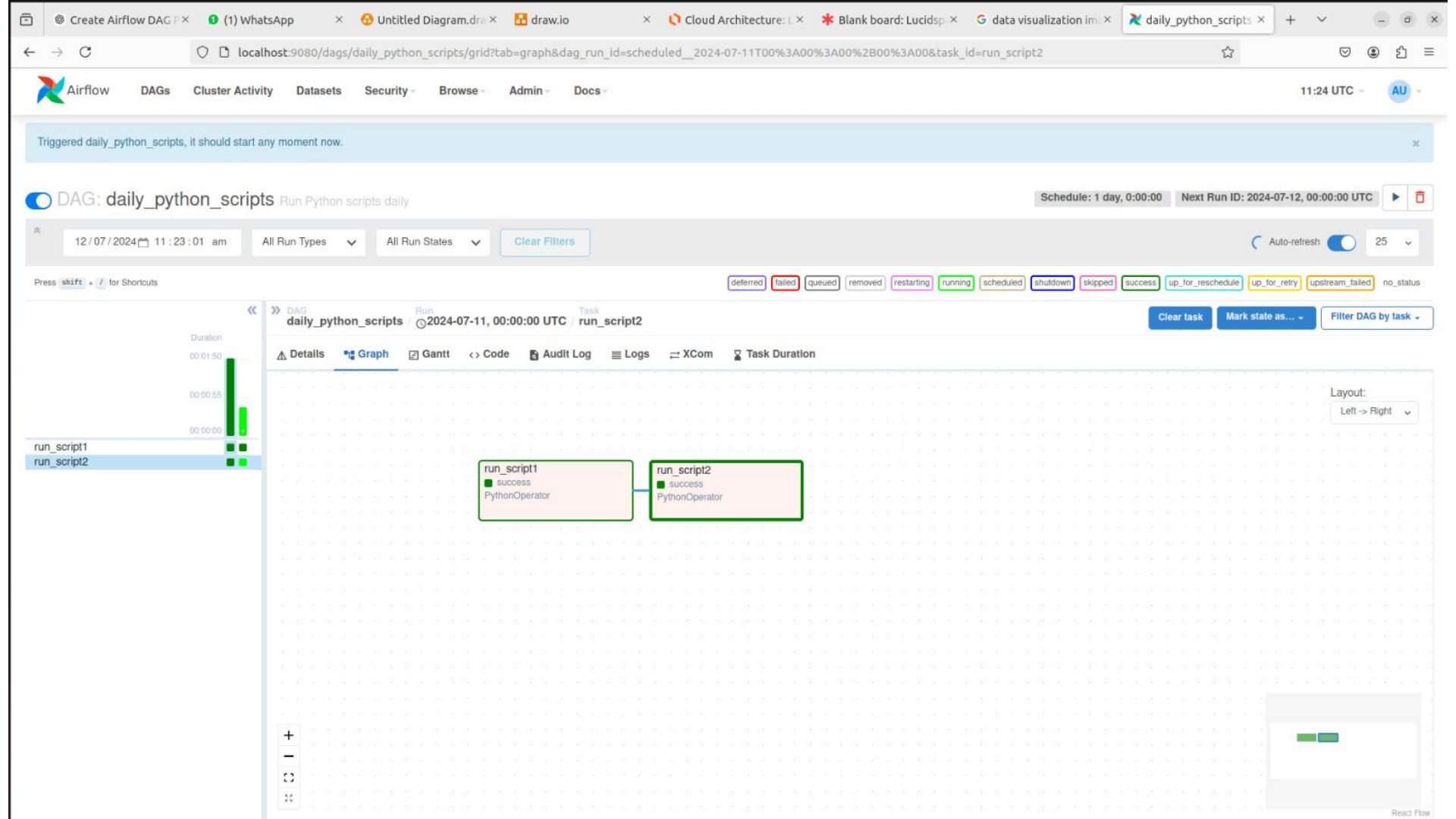
STEP 1: Create the Python Scripts

STEP 2: Create the DAG File

STEP 3: Start the Airflow Web Server

EXECUTE DAILY PYTHON SCRIPTS




Create Airflow DAG
(1) WhatsApp
Untitled Diagram.dra
draw.io
Cloud Architecture:
Blank board: Lucidsp
data visualization im
daily_python_scripts
localhost:9080/dags/daily_python_scripts/grid?tab=graph&dag_run_id=scheduled__2024-07-11T00%3A00%3A00%2B00%3A00&task_id=run_script2
Airflow
DAGs
Cluster Activity
Datasets
Security
Browse
Admin
Docs
11:24 UTC
AU
Triggered daily_python_scripts, it should start any moment now.
DAG: daily_python_scripts
Run Python scripts daily
Schedule: 1 day, 0:00:00
Next Run ID: 2024-07-12, 00:00:00 UTC
12 / 07 / 2024 11 : 23 : 01 am
All Run Types
All Run States
Clear Filters
Auto-refresh
25
Press shift + / for Shortcuts
deferred
failed
queued
removed
restarting
running
scheduled
shutdown
skipped
success
up_for_reschedule
up_for_retry
upstream_failed
no_status
Duration
00:01:50
00:00:55
00:00:00
run_script1
run_script2
DAG
daily_python_scripts
Run
2024-07-11, 00:00:00 UTC
Task
run_script2
Clear task
Mark state as...
Filter DAG by task
Details
Graph
Gantt
Code
Audit Log
Logs
XCom
Task Duration
Layout:
Left -> Right
run_script1
success
PythonOperator
run_script2
success
PythonOperator
React Flow

MEDALLION TASK

Bronze Layer

- **Purpose:** Raw data ingestion.
- **Characteristics:** This layer contains raw data from various sources, often stored in its original format without any transformations or cleaning. It acts as the landing zone for all incoming data.
- **Examples:** Logs, sensor data, transactional data from databases.

Silver Layer

- **Purpose:** Data cleansing and enrichment.
- **Characteristics:** This layer processes and cleans the data from the bronze layer. It involves removing duplicates, handling missing values, and transforming the data into a more usable format. This layer often includes normalized and enriched data.
- **Examples:** Processed sensor readings, cleaned and standardized transactional data.

Gold Layer

- **Purpose:** High-quality, aggregated data for analytics and reporting.
- **Characteristics:** This layer contains highly refined data ready for consumption by business intelligence tools, machine learning models, and other analytical applications. It often involves aggregations, calculations, and business logic applied to the silver

nandhumidhun@nandhumidhun-HP-Laptop-15-bs0xx: ~/kafka

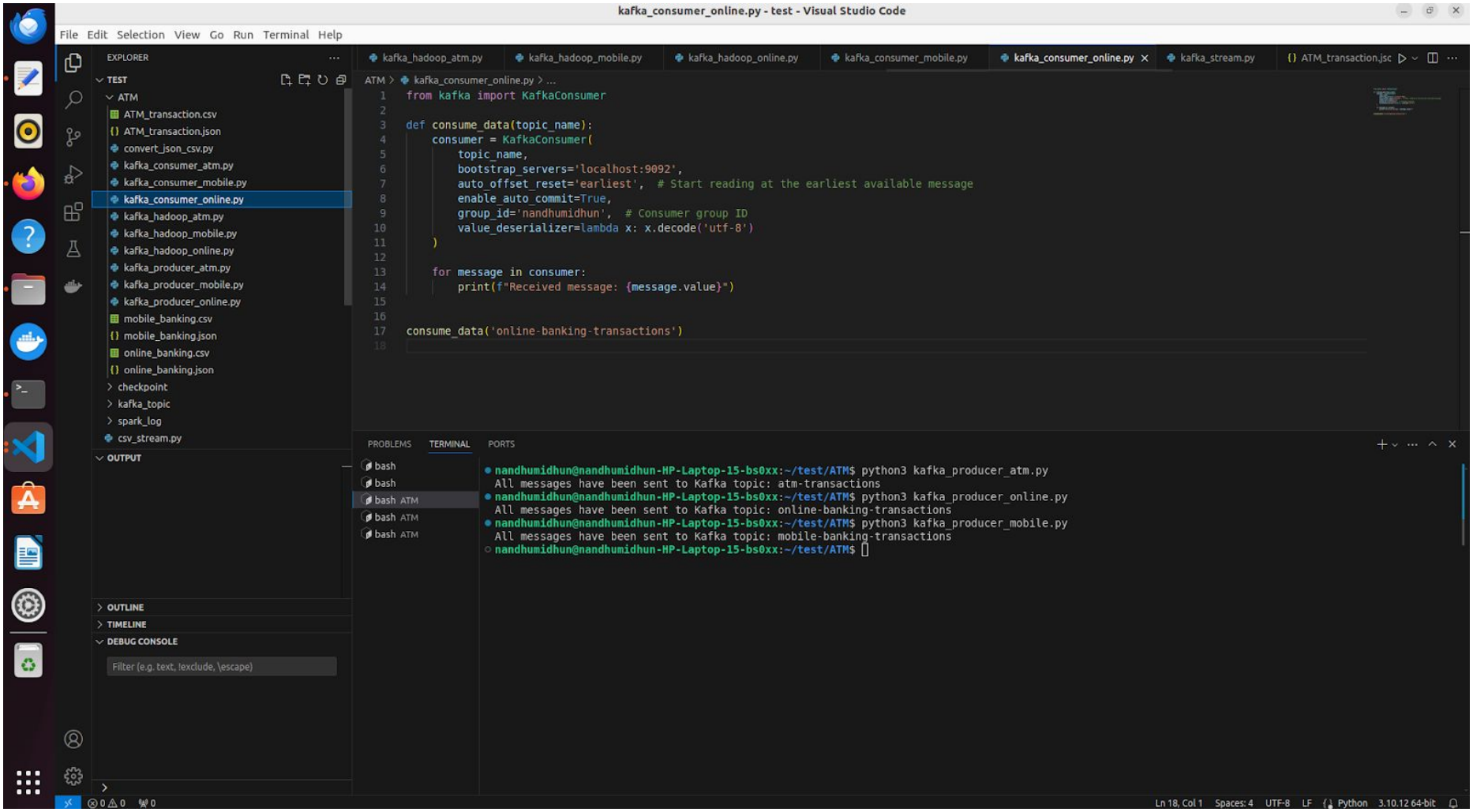
nandhumidhun@nandhumidhun-HP-Laptop-15-bs0xx: ~

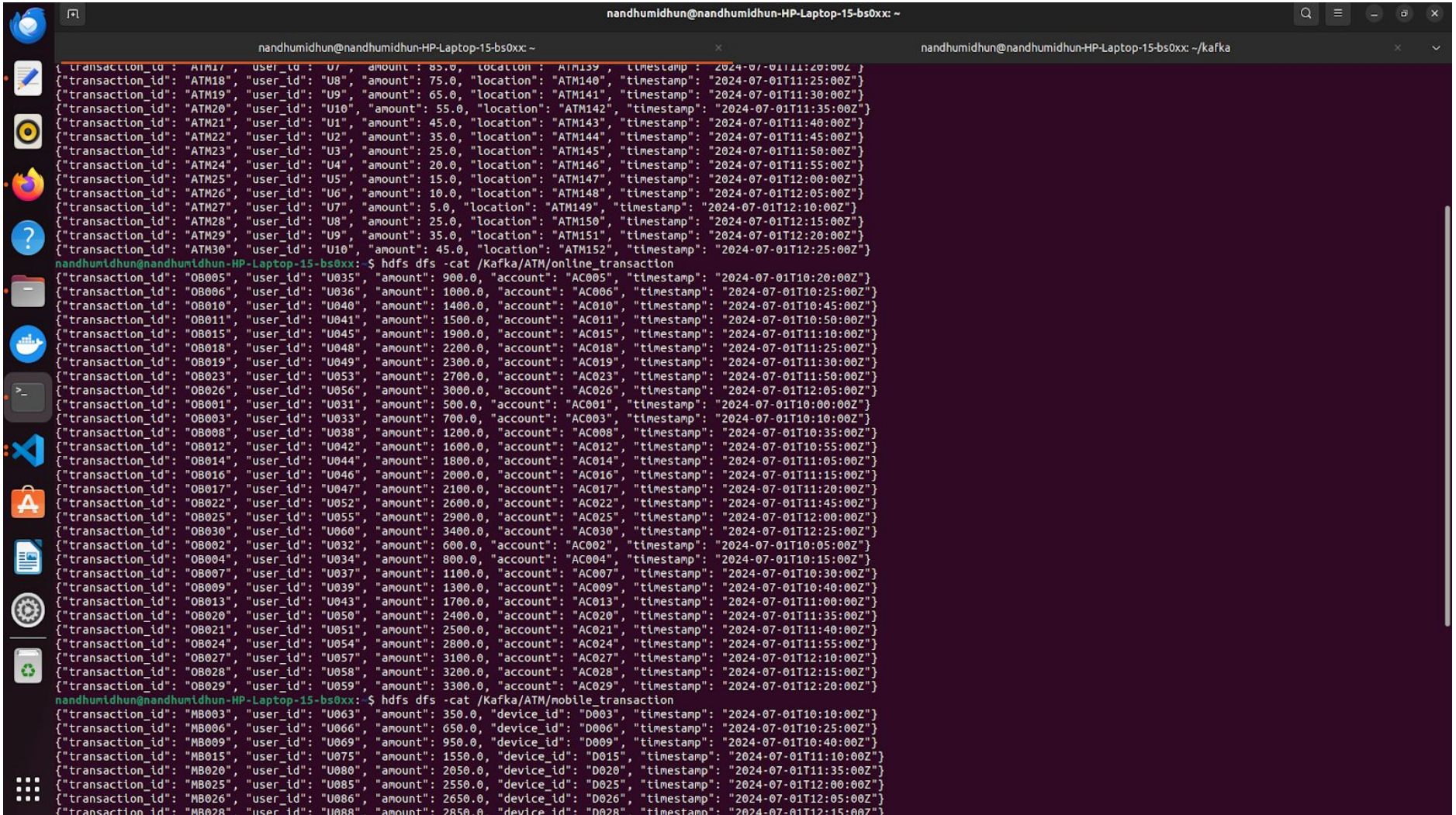
```
nandhumidhun@nandhumidhun-HP-Laptop-15-bs0xx:~/kafka$ cd
nandhumidhun@nandhumidhun-HP-Laptop-15-bs0xx:~$ hdfs dfs -cat /Kafka/kafka_etl/bronze/*
userId,id,title,completed
1,1,delectus aut autem,False
1,2,quis ut nam facilis et officia qui,False
1,3,fugiat veniam minus,False
1,4,et porro tempora,True
1,5,laboriosam mollitia et enim quasi adipisci quia provident illum,False
1,6,qui ullam ratione quibusdam voluptatem quia omnis,False
1,7,illo expedita consequatur quia in,False
1,8,quo adipisci enim quam ut ab,True
1,9,molestiae perspiciatis ipsa,False
1,10,illo est ratione doloremque quia maiores aut,True
1,11,vero rerum temporibus dolor,True
1,12,ipsa repellendus fugit nisi,True
1,13,et doloremque nulla,False
1,14,repellendus sunt dolores architecto voluptatum,True
1,15,ab voluptatum amet voluptas,True
1,16,accusamus eos facilis sint et aut voluptatem,True
1,17,quo laboriosam deleniti aut qui,True
1,18,dolorum est consequatur ea mollitia in culpa,False
1,19,molestiae ipsa aut voluptatibus pariatur dolor nihil,True
1,20,ullam nobis libero sapiente ad optio sint,True
2,21,suscipit repellat esse quibusdam voluptatem incidunt,False
2,22,distinctio vitae autem nihil ut molestias quo,True
2,23,et itaque necessitatibus maxime molestiae qui quas velit,False
2,24,adipisci non ad dicta qui amet quaerat doloribus ea,False
2,25,voluptas quo tenetur perspiciatis explicabo natus,True
2,26,aliquam aut quasi,True
2,27,veritatis pariatur delectus,True
2,28,nesciunt totam sit blanditiis sit,False
2,29,laborum aut in quam,False
2,30,nemo perspiciatis repellat ut dolor libero commodi blanditiis omnis,True
2,31,repudiandae totam in est sint facere fuga,False
2,32,earum doloribus ea doloremque quis,False
2,33,sint sit aut vero,False
2,34,porro aut necessitatibus eaque distinctio,False
2,35,repellendus veritatis molestias dicta incidunt,True
2,36,excepturi deleniti adipisci voluptatem et neque optio illum ad,True
2,37,sunt cum tempora,False
2,38,totam quia non,False
2,39,doloremque quibusdam asperiores libero corrupti illum qui omnis,False
2,40,totam atque quo nesciunt,True
3,41,aliquid amet impedit consequatur aspernatur placeat eaque fugiat suscipit,False
3,42,rerum perferendis error quia ut eveniet,False
3,43,tempore ut sint quis recusandae,True
3,44,cum debitis quis accusamus doloremque ipsa natus sapiente omnis,True
3,45,velit soluta adipisci molestias reiciendis harum,False
3,46,vel voluptatem repellat nihil placeat corporis,False
3,47,nam qui rerum fugiat accusamus,False
3,48,sit reprehenderit omnis quia,False
3,49,ut necessitatibus aut maiores debitis officia blanditiis velit et,False
3,50,cupiditate necessitatibus ullam aut quis dolor voluptate,True
```

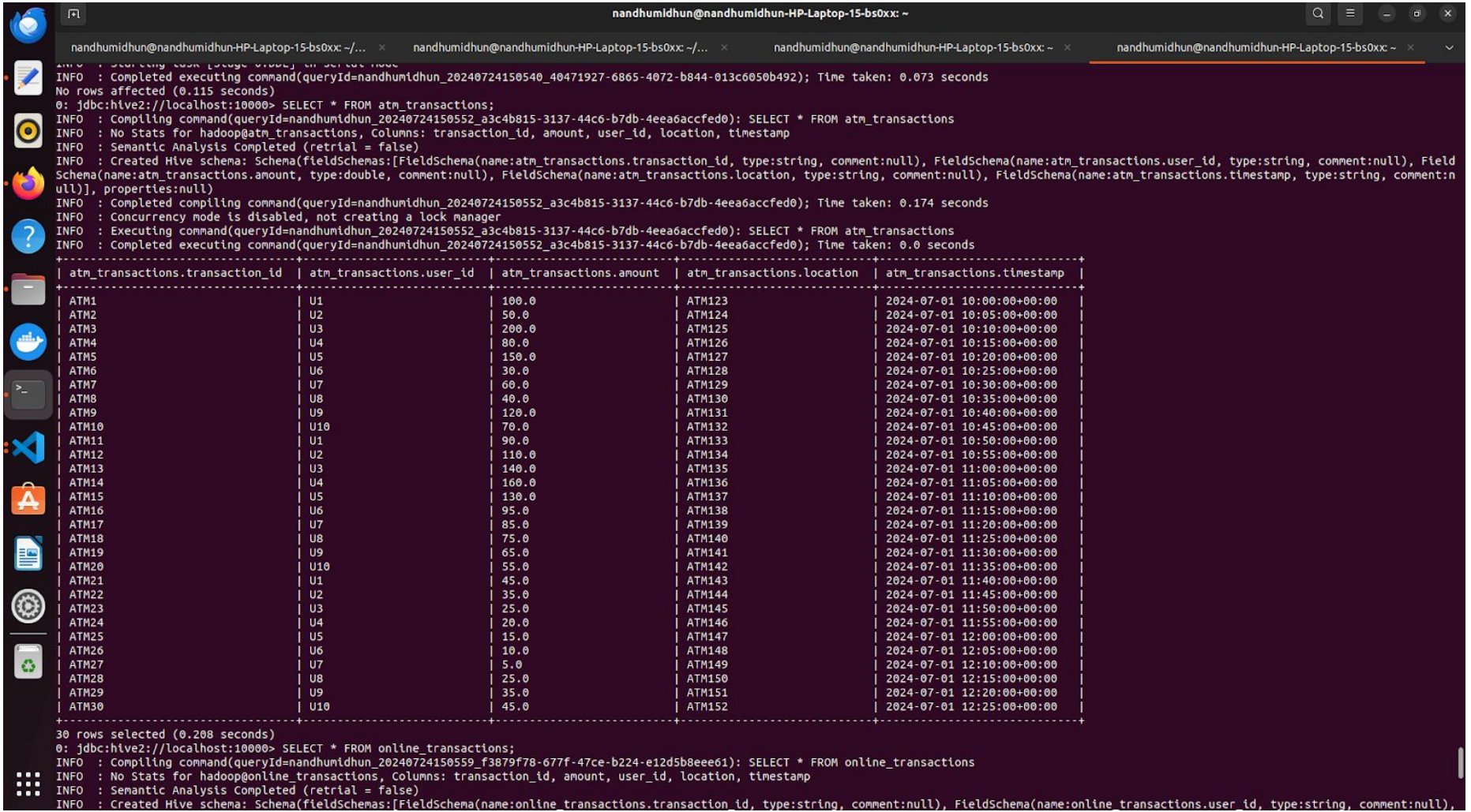
```
9,161,ex hic consequuntur earum omnis alias ut occaecati culpa,True
9,162,omnis laboriosam molestias animi sunt dolore,True
9,163,natus corrupti maxime laudantium et voluptatem laboriosam odit,False
9,164,reprehenderit quos aut aut consequatur est sed,False
9,165,fugiat perferendis sed aut quidem,False
9,166,quos quo possimus suscipit minima ut,False
9,167,et quis minus quo a asperiores molestiae,False
9,168,recusandae quia qui sunt libero,False
9,169,ea odio perferendis officiis,True
9,170,quisquam aliquam quia doloribus aut,False
9,171,fugiat aut voluptatibus corrupti deleniti velit iste odio,True
9,172,et provident amet rerum consectetur et voluptatum,False
9,173,harum ad aperiam quis,False
9,174,similique aut quo,False
9,175,laudantium eius officia perferendis provident perspiciatis asperiores,True
9,176,magni soluta corrupti ut maiores rem quidem,False
9,177,et placeat temporibus voluptas est tempora quos quibusdam,False
9,178,nesciunt itaque commodi tempore,True
9,179,omnis consequuntur cupiditate impedit itaque ipsam quo,True
9,180,debitis nisi et dolorem repellat et,True
10,181,ut cupiditate sequi aliquam fuga maiores,False
10,182,inventore saepe cumque et aut illum enim,True
10,183,omnis nulla eum aliquam distinctio,True
10,184,molestias modi perferendis perspiciatis,False
10,185,voluptates dignissimos sed doloribus animi quaerat aut,False
10,186,explicabo odio est et,False
10,187,consequuntur animi possimus,False
10,188,vel non beatae est,True
10,189,culpa eius et voluptatem et,True
10,190,accusamus sint iusto et voluptatem exercitationem,True
10,191,temporibus atque distinctio omnis eius impedit tempore molestias pariatur,True
10,192,ut quas possimus exercitationem sint voluptates,False
10,193,rerum debitis voluptatem qui eveniet tempora distinctio a,True
10,194,sed ut vero sit molestiae,False
10,195,rerum ex veniam mollitia voluptatibus pariatur,True
10,196,consequuntur aut ut fugit similique,True
10,197,dignissimos quo nobis earum saepe,True
10,198,quis eius est sint explicabo,True
10,199,numquam repellendus a magnam,True
10,200,ipsam aperiam voluptates qui,False
nandhumidhun@nandhumidhun-HP-Laptop-15-bs0xx:~$ hdfs dfs -cat /Kafka/kafka_etl/gold/*
userId,id
1,20
2,20
3,20
4,20
5,20
6,20
7,20
8,20
9,20
10,20
nandhumidhun@nandhumidhun-HP-Laptop-15-bs0xx:~$
```

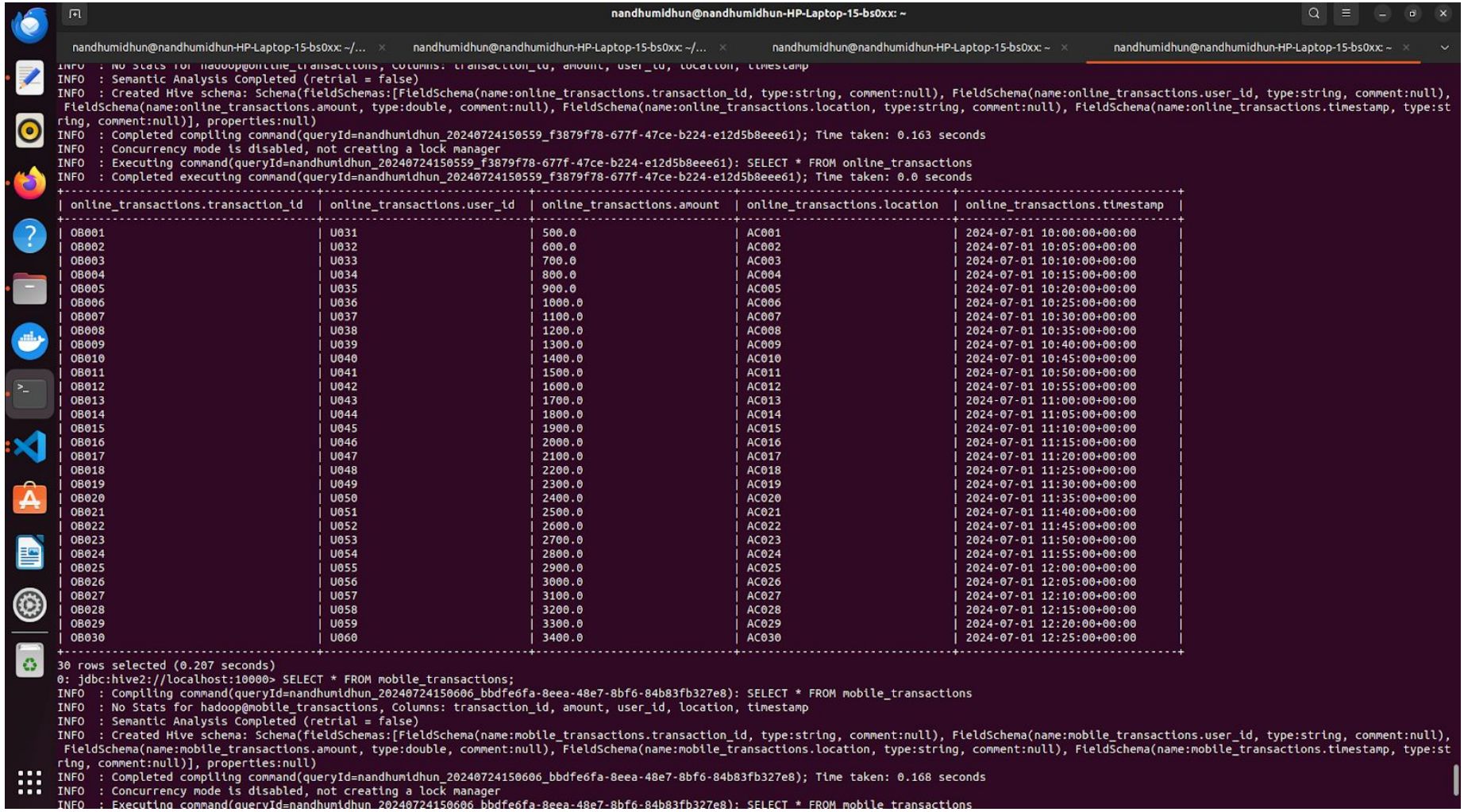
Banking Applications

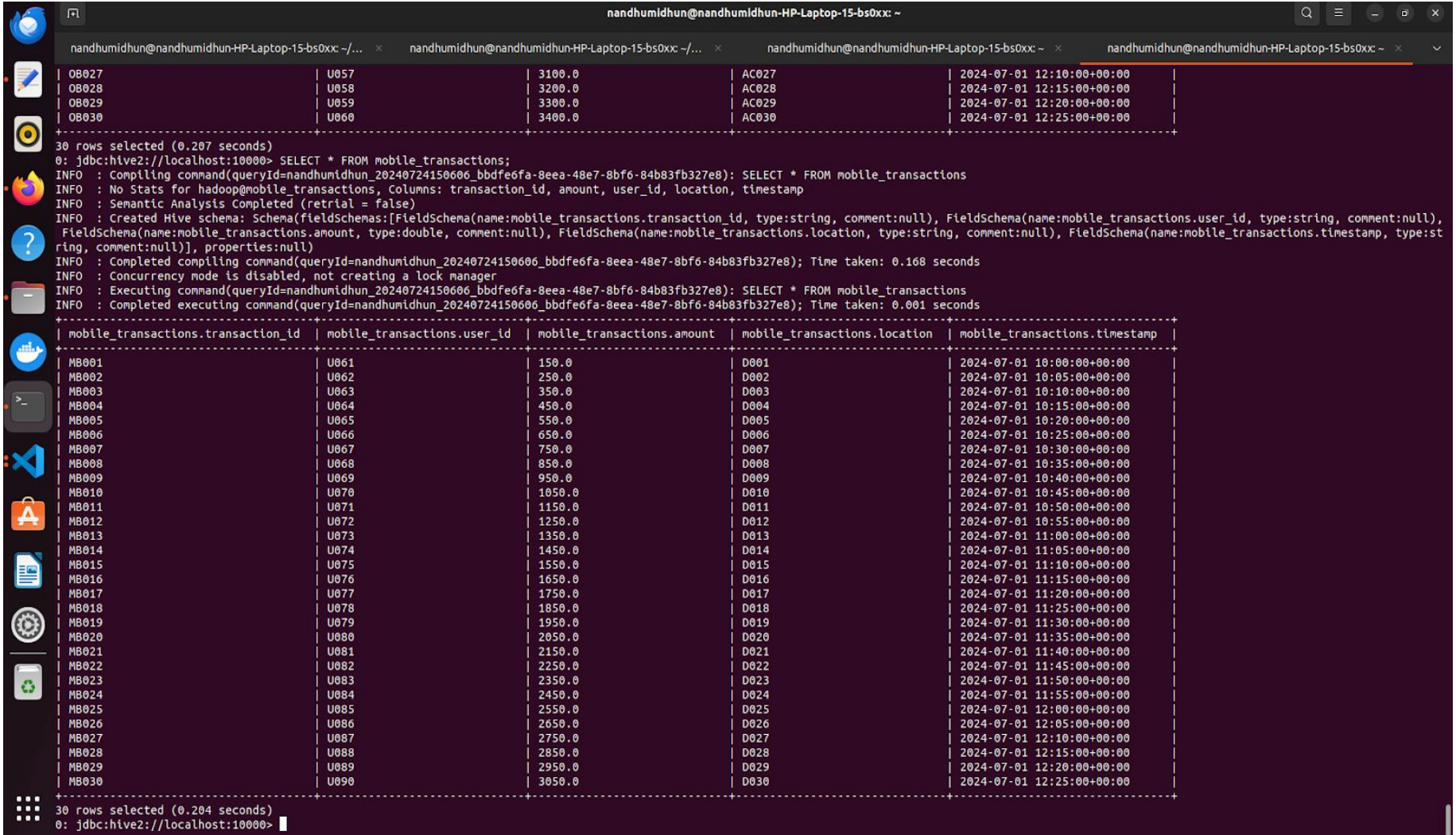
STEP 1: Data Ingestion
STEP 2: Data Storage
STEP 3: Data Processing
STEP 4: Data Transformation
STEP 5: Fraud Detection Analytics
STEP 6: Real-time Alerts
STEP 7: Data Visualization and Reporting

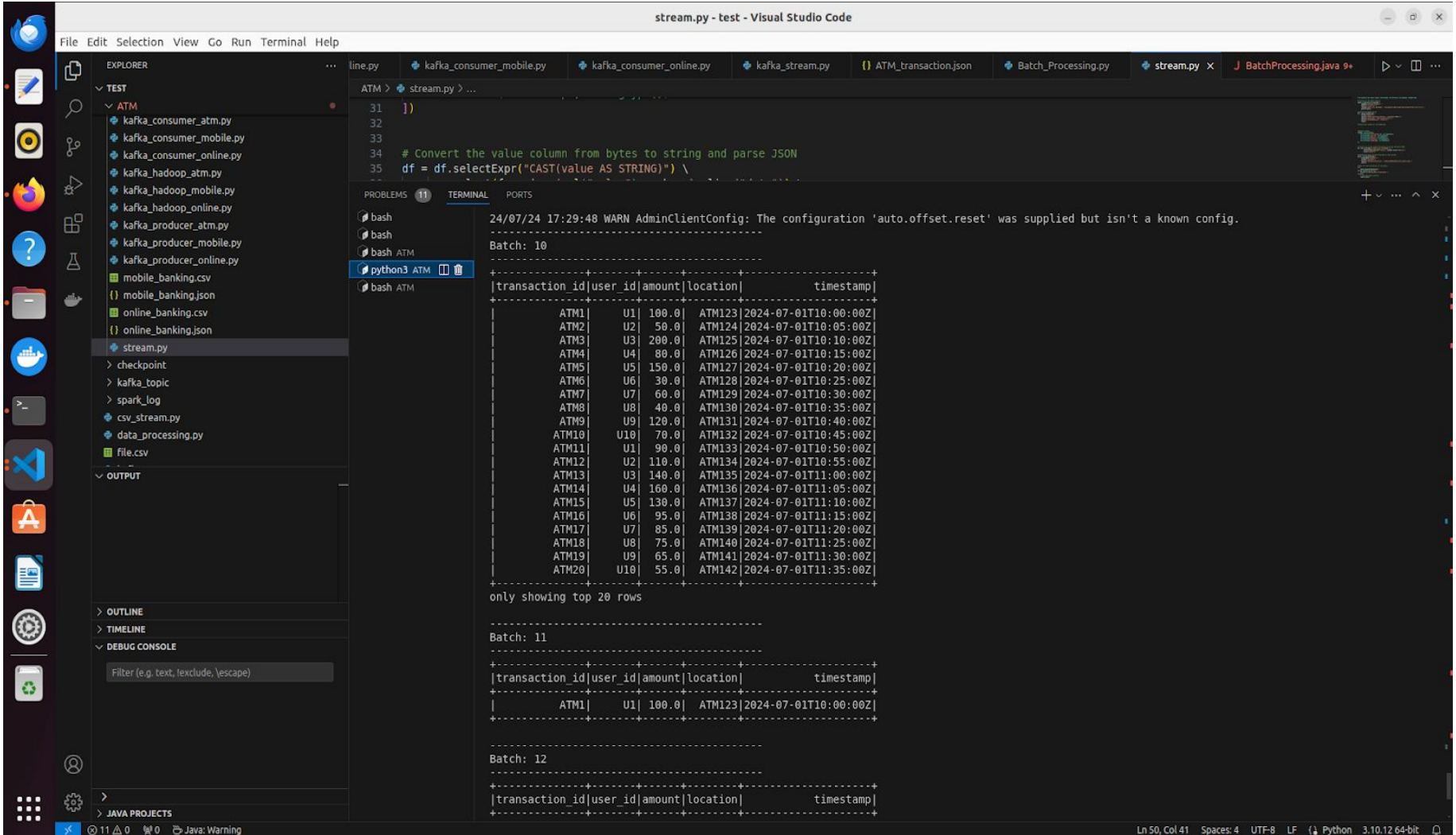

kafka_consumer_online.py - test - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
TEST
ATM
ATM_transaction.csv
ATM_transaction.json
convert_json_csv.py
kafka_consumer_atm.py
kafka_consumer_mobile.py
kafka_consumer_online.py
kafka_hadoop_atm.py
kafka_hadoop_mobile.py
kafka_hadoop_online.py
kafka_producer_atm.py
kafka_producer_mobile.py
kafka_producer_online.py
mobile_banking.csv
mobile_banking.json
online_banking.csv
online_banking.json
checkpoint
kafka_topic
spark_log
csv_stream.py
OUTPUT
OUTLINE
TIMELINE
DEBUG CONSOLE
Filter (e.g. text, !exclude, \escape)
kafka_hadoop_atm.py
kafka_hadoop_mobile.py
kafka_hadoop_online.py
kafka_consumer_mobile.py
kafka_consumer_online.py
kafka_stream.py
ATM_transaction.jsc
ATM > kafka_consumer_online.py > ...
1 from kafka import KafkaConsumer
2
3 def consume_data(topic_name):
4 consumer = KafkaConsumer(
5 topic_name,
6 bootstrap_servers='localhost:9092',
7 auto_offset_reset='earliest', # Start reading at the earliest available message
8 enable_auto_commit=True,
9 group_id='nandhumidhun', # Consumer group ID
10 value_deserializer=lambda x: x.decode('utf-8')
11)
12
13 for message in consumer:
14 print(f"Received message: {message.value}")
15
16
17 consume_data('online-banking-transactions')
18
PROBLEMS TERMINAL PORTS
bash
bash
bash ATM
bash ATM
bash ATM
nandhumidhun@nandhumidhun-HP-Laptop-15-bs0xx:~/test/ATM$ python3 kafka_producer_atm.py
All messages have been sent to Kafka topic: atm-transactions
nandhumidhun@nandhumidhun-HP-Laptop-15-bs0xx:~/test/ATM$ python3 kafka_producer_online.py
All messages have been sent to Kafka topic: online-banking-transactions
nandhumidhun@nandhumidhun-HP-Laptop-15-bs0xx:~/test/ATM$ python3 kafka_producer_mobile.py
All messages have been sent to Kafka topic: mobile-banking-transactions
nandhumidhun@nandhumidhun-HP-Laptop-15-bs0xx:~/test/ATM$
Ln 18, Col 1 Spaces: 4 UTF-8 LF Python 3.10.12 64-bit

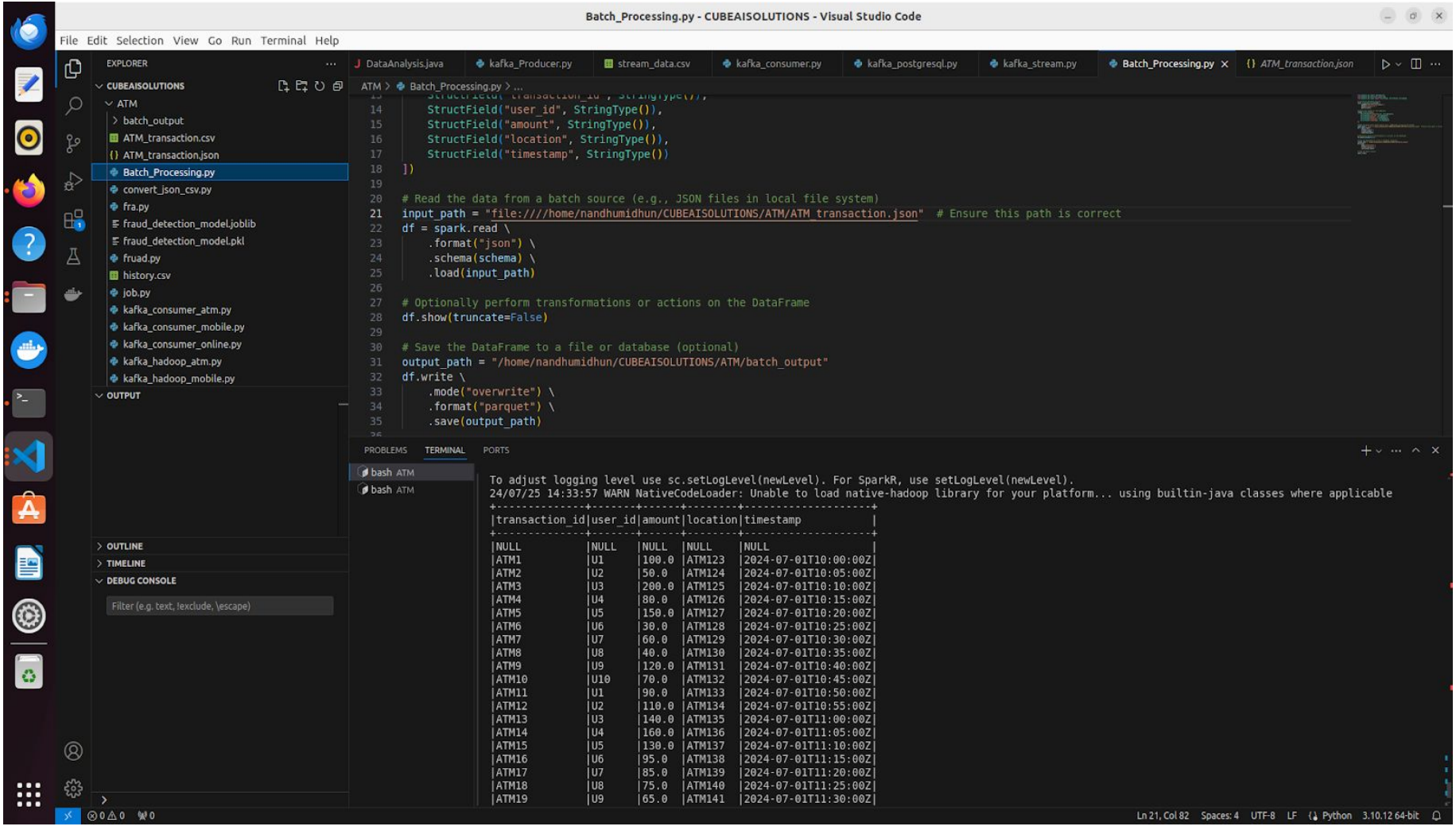

nandhumidhun@nandhumidhun-HP-Laptop-15-bs0xx: ~
nandhumidhun@nandhumidhun-HP-Laptop-15-bs0xx: ~
nandhumidhun@nandhumidhun-HP-Laptop-15-bs0xx: ~/kafka
{"transaction_id": "ATM17", "user_id": "U7", "amount": 85.0, "location": "ATM139", "timestamp": "2024-07-01T11:20:00Z"}
{"transaction_id": "ATM18", "user_id": "U8", "amount": 75.0, "location": "ATM140", "timestamp": "2024-07-01T11:25:00Z"}
{"transaction_id": "ATM19", "user_id": "U9", "amount": 65.0, "location": "ATM141", "timestamp": "2024-07-01T11:30:00Z"}
{"transaction_id": "ATM20", "user_id": "U10", "amount": 55.0, "location": "ATM142", "timestamp": "2024-07-01T11:35:00Z"}
{"transaction_id": "ATM21", "user_id": "U1", "amount": 45.0, "location": "ATM143", "timestamp": "2024-07-01T11:40:00Z"}
{"transaction_id": "ATM22", "user_id": "U2", "amount": 35.0, "location": "ATM144", "timestamp": "2024-07-01T11:45:00Z"}
{"transaction_id": "ATM23", "user_id": "U3", "amount": 25.0, "location": "ATM145", "timestamp": "2024-07-01T11:50:00Z"}
{"transaction_id": "ATM24", "user_id": "U4", "amount": 20.0, "location": "ATM146", "timestamp": "2024-07-01T11:55:00Z"}
{"transaction_id": "ATM25", "user_id": "U5", "amount": 15.0, "location": "ATM147", "timestamp": "2024-07-01T12:00:00Z"}
{"transaction_id": "ATM26", "user_id": "U6", "amount": 10.0, "location": "ATM148", "timestamp": "2024-07-01T12:05:00Z"}
{"transaction_id": "ATM27", "user_id": "U7", "amount": 5.0, "location": "ATM149", "timestamp": "2024-07-01T12:10:00Z"}
{"transaction_id": "ATM28", "user_id": "U8", "amount": 25.0, "location": "ATM150", "timestamp": "2024-07-01T12:15:00Z"}
{"transaction_id": "ATM29", "user_id": "U9", "amount": 35.0, "location": "ATM151", "timestamp": "2024-07-01T12:20:00Z"}
{"transaction_id": "ATM30", "user_id": "U10", "amount": 45.0, "location": "ATM152", "timestamp": "2024-07-01T12:25:00Z"}
nandhumidhun@nandhumidhun-HP-Laptop-15-bs0xx:~$ hdfs dfs -cat /Kafka/ATM/online_transaction
{"transaction_id": "OB005", "user_id": "U035", "amount": 900.0, "account": "AC005", "timestamp": "2024-07-01T10:20:00Z"}
{"transaction_id": "OB006", "user_id": "U036", "amount": 1000.0, "account": "AC006", "timestamp": "2024-07-01T10:25:00Z"}
{"transaction_id": "OB010", "user_id": "U040", "amount": 1400.0, "account": "AC010", "timestamp": "2024-07-01T10:45:00Z"}
{"transaction_id": "OB011", "user_id": "U041", "amount": 1500.0, "account": "AC011", "timestamp": "2024-07-01T10:50:00Z"}
{"transaction_id": "OB015", "user_id": "U045", "amount": 1900.0, "account": "AC015", "timestamp": "2024-07-01T11:10:00Z"}
{"transaction_id": "OB018", "user_id": "U048", "amount": 2200.0, "account": "AC018", "timestamp": "2024-07-01T11:25:00Z"}
{"transaction_id": "OB019", "user_id": "U049", "amount": 2300.0, "account": "AC019", "timestamp": "2024-07-01T11:30:00Z"}
{"transaction_id": "OB023", "user_id": "U053", "amount": 2700.0, "account": "AC023", "timestamp": "2024-07-01T11:50:00Z"}
{"transaction_id": "OB026", "user_id": "U056", "amount": 3000.0, "account": "AC026", "timestamp": "2024-07-01T12:05:00Z"}
{"transaction_id": "OB001", "user_id": "U031", "amount": 500.0, "account": "AC001", "timestamp": "2024-07-01T10:00:00Z"}
{"transaction_id": "OB003", "user_id": "U033", "amount": 700.0, "account": "AC003", "timestamp": "2024-07-01T10:10:00Z"}
{"transaction_id": "OB008", "user_id": "U038", "amount": 1200.0, "account": "AC008", "timestamp": "2024-07-01T10:35:00Z"}
{"transaction_id": "OB012", "user_id": "U042", "amount": 1600.0, "account": "AC012", "timestamp": "2024-07-01T10:55:00Z"}
{"transaction_id": "OB014", "user_id": "U044", "amount": 1800.0, "account": "AC014", "timestamp": "2024-07-01T11:05:00Z"}
{"transaction_id": "OB016", "user_id": "U046", "amount": 2000.0, "account": "AC016", "timestamp": "2024-07-01T11:15:00Z"}
{"transaction_id": "OB017", "user_id": "U047", "amount": 2100.0, "account": "AC017", "timestamp": "2024-07-01T11:20:00Z"}
{"transaction_id": "OB022", "user_id": "U052", "amount": 2600.0, "account": "AC022", "timestamp": "2024-07-01T11:45:00Z"}
{"transaction_id": "OB025", "user_id": "U055", "amount": 2900.0, "account": "AC025", "timestamp": "2024-07-01T12:00:00Z"}
{"transaction_id": "OB030", "user_id": "U060", "amount": 3400.0, "account": "AC030", "timestamp": "2024-07-01T12:25:00Z"}
{"transaction_id": "OB002", "user_id": "U032", "amount": 600.0, "account": "AC002", "timestamp": "2024-07-01T10:05:00Z"}
{"transaction_id": "OB004", "user_id": "U034", "amount": 800.0, "account": "AC004", "timestamp": "2024-07-01T10:15:00Z"}
{"transaction_id": "OB007", "user_id": "U037", "amount": 1100.0, "account": "AC007", "timestamp": "2024-07-01T10:30:00Z"}
{"transaction_id": "OB009", "user_id": "U039", "amount": 1300.0, "account": "AC009", "timestamp": "2024-07-01T10:40:00Z"}
{"transaction_id": "OB013", "user_id": "U043", "amount": 1700.0, "account": "AC013", "timestamp": "2024-07-01T11:00:00Z"}
{"transaction_id": "OB020", "user_id": "U050", "amount": 2400.0, "account": "AC020", "timestamp": "2024-07-01T11:35:00Z"}
{"transaction_id": "OB021", "user_id": "U051", "amount": 2500.0, "account": "AC021", "timestamp": "2024-07-01T11:40:00Z"}
{"transaction_id": "OB024", "user_id": "U054", "amount": 2800.0, "account": "AC024", "timestamp": "2024-07-01T11:55:00Z"}
{"transaction_id": "OB027", "user_id": "U057", "amount": 3100.0, "account": "AC027", "timestamp": "2024-07-01T12:10:00Z"}
{"transaction_id": "OB028", "user_id": "U058", "amount": 3200.0, "account": "AC028", "timestamp": "2024-07-01T12:15:00Z"}
{"transaction_id": "OB029", "user_id": "U059", "amount": 3300.0, "account": "AC029", "timestamp": "2024-07-01T12:20:00Z"}
nandhumidhun@nandhumidhun-HP-Laptop-15-bs0xx:~$ hdfs dfs -cat /Kafka/ATM/mobile_transaction
{"transaction_id": "MB003", "user_id": "U063", "amount": 350.0, "device_id": "D003", "timestamp": "2024-07-01T10:10:00Z"}
{"transaction_id": "MB006", "user_id": "U066", "amount": 650.0, "device_id": "D006", "timestamp": "2024-07-01T10:25:00Z"}
{"transaction_id": "MB009", "user_id": "U069", "amount": 950.0, "device_id": "D009", "timestamp": "2024-07-01T10:40:00Z"}
{"transaction_id": "MB015", "user_id": "U075", "amount": 1550.0, "device_id": "D015", "timestamp": "2024-07-01T11:10:00Z"}
{"transaction_id": "MB020", "user_id": "U080", "amount": 2050.0, "device_id": "D020", "timestamp": "2024-07-01T11:35:00Z"}
{"transaction_id": "MB025", "user_id": "U085", "amount": 2550.0, "device_id": "D025", "timestamp": "2024-07-01T12:00:00Z"}
{"transaction_id": "MB026", "user_id": "U086", "amount": 2650.0, "device_id": "D026", "timestamp": "2024-07-01T12:05:00Z"}


nandhumidhun@nandhumidhun-HP-Laptop-15-bs0xx: ~
nandhumidhun@nandhumidhun-HP-Laptop-15-bs0xx: ~/...
nandhumidhun@nandhumidhun-HP-Laptop-15-bs0xx: ~/...
nandhumidhun@nandhumidhun-HP-Laptop-15-bs0xx: ~
nandhumidhun@nandhumidhun-HP-Laptop-15-bs0xx: ~
INFO : Completed executing command(queryId=nandhumidhun_20240724150540_40471927-6865-4072-b844-013c6050b492); Time taken: 0.073 seconds
No rows affected (0.115 seconds)
0: jdbc:hive2://localhost:10000> SELECT * FROM atm_transactions;
INFO : Compiling command(queryId=nandhumidhun_20240724150552_a3c4b815-3137-44c6-b7db-4eea6accfed0): SELECT * FROM atm_transactions
INFO : No Stats for hadoop@atm_transactions, Columns: transaction_id, amount, user_id, location, timestamp
INFO : Semantic Analysis Completed (retrial = false)
INFO : Created Hive schema: Schema(fieldSchemas:[FieldSchema(name:atm_transactions.transaction_id, type:string, comment:null), FieldSchema(name:atm_transactions.user_id, type:string, comment:null), FieldSchema(name:atm_transactions.amount, type:double, comment:null), FieldSchema(name:atm_transactions.location, type:string, comment:null), FieldSchema(name:atm_transactions.timestamp, type:string, comment:null)], properties:null)
INFO : Completed compiling command(queryId=nandhumidhun_20240724150552_a3c4b815-3137-44c6-b7db-4eea6accfed0); Time taken: 0.174 seconds
INFO : Concurrency mode is disabled, not creating a lock manager
INFO : Executing command(queryId=nandhumidhun_20240724150552_a3c4b815-3137-44c6-b7db-4eea6accfed0): SELECT * FROM atm_transactions
INFO : Completed executing command(queryId=nandhumidhun_20240724150552_a3c4b815-3137-44c6-b7db-4eea6accfed0); Time taken: 0.0 seconds
atm_transactions.transaction_id	atm_transactions.user_id	atm_transactions.amount	atm_transactions.location	atm_transactions.timestamp
ATM1	U1	100.0	ATM123	2024-07-01 10:00:00+00:00
ATM2	U2	50.0	ATM124	2024-07-01 10:05:00+00:00
ATM3	U3	200.0	ATM125	2024-07-01 10:10:00+00:00
ATM4	U4	80.0	ATM126	2024-07-01 10:15:00+00:00
ATM5	U5	150.0	ATM127	2024-07-01 10:20:00+00:00
ATM6	U6	30.0	ATM128	2024-07-01 10:25:00+00:00
ATM7	U7	60.0	ATM129	2024-07-01 10:30:00+00:00
ATM8	U8	40.0	ATM130	2024-07-01 10:35:00+00:00
ATM9	U9	120.0	ATM131	2024-07-01 10:40:00+00:00
ATM10	U10	70.0	ATM132	2024-07-01 10:45:00+00:00
ATM11	U1	90.0	ATM133	2024-07-01 10:50:00+00:00
ATM12	U2	110.0	ATM134	2024-07-01 10:55:00+00:00
ATM13	U3	140.0	ATM135	2024-07-01 11:00:00+00:00
ATM14	U4	160.0	ATM136	2024-07-01 11:05:00+00:00
ATM15	U5	130.0	ATM137	2024-07-01 11:10:00+00:00
ATM16	U6	95.0	ATM138	2024-07-01 11:15:00+00:00
ATM17	U7	85.0	ATM139	2024-07-01 11:20:00+00:00
ATM18	U8	75.0	ATM140	2024-07-01 11:25:00+00:00
ATM19	U9	65.0	ATM141	2024-07-01 11:30:00+00:00
ATM20	U10	55.0	ATM142	2024-07-01 11:35:00+00:00
ATM21	U1	45.0	ATM143	2024-07-01 11:40:00+00:00
ATM22	U2	35.0	ATM144	2024-07-01 11:45:00+00:00
ATM23	U3	25.0	ATM145	2024-07-01 11:50:00+00:00
ATM24	U4	20.0	ATM146	2024-07-01 11:55:00+00:00
ATM25	U5	15.0	ATM147	2024-07-01 12:00:00+00:00
ATM26	U6	10.0	ATM148	2024-07-01 12:05:00+00:00
ATM27	U7	5.0	ATM149	2024-07-01 12:10:00+00:00
ATM28	U8	25.0	ATM150	2024-07-01 12:15:00+00:00
ATM29	U9	35.0	ATM151	2024-07-01 12:20:00+00:00
ATM30	U10	45.0	ATM152	2024-07-01 12:25:00+00:00
30 rows selected (0.208 seconds)
0: jdbc:hive2://localhost:10000> SELECT * FROM online_transactions;
INFO : Compiling command(queryId=nandhumidhun_20240724150559_f3879f78-677f-47ce-b224-e12d5b8eee61): SELECT * FROM online_transactions
INFO : No Stats for hadoop@online_transactions, Columns: transaction_id, amount, user_id, location, timestamp
INFO : Semantic Analysis Completed (retrial = false)
INFO : Created Hive schema: Schema(fieldSchemas:[FieldSchema(name:online_transactions.transaction_id, type:string, comment:null), FieldSchema(name:online_transactions.user_id, type:string, comment:null),


nandhumidhun@nandhumidhun-HP-Laptop-15-bs0xx: ~
nandhumidhun@nandhumidhun-HP-Laptop-15-bs0xx: ~/...
nandhumidhun@nandhumidhun-HP-Laptop-15-bs0xx: ~/...
nandhumidhun@nandhumidhun-HP-Laptop-15-bs0xx: ~
nandhumidhun@nandhumidhun-HP-Laptop-15-bs0xx: ~
INFO : No stats for hadoop@online_transactions, Columns: transaction_id, amount, user_id, location, timestamp
INFO : Semantic Analysis Completed (retrial = false)
INFO : Created Hive schema: Schema(fieldSchemas:[FieldSchema(name:online_transactions.transaction_id, type:string, comment:null), FieldSchema(name:online_transactions.user_id, type:string, comment:null), FieldSchema(name:online_transactions.amount, type:double, comment:null), FieldSchema(name:online_transactions.location, type:string, comment:null), FieldSchema(name:online_transactions.timestamp, type:string, comment:null)], properties:null)
INFO : Completed compiling command(queryId=nandhumidhun_20240724150559_f3879f78-677f-47ce-b224-e12d5b8eee61); Time taken: 0.163 seconds
INFO : Concurrency mode is disabled, not creating a lock manager
INFO : Executing command(queryId=nandhumidhun_20240724150559_f3879f78-677f-47ce-b224-e12d5b8eee61): SELECT * FROM online_transactions
INFO : Completed executing command(queryId=nandhumidhun_20240724150559_f3879f78-677f-47ce-b224-e12d5b8eee61); Time taken: 0.0 seconds
online_transactions.transaction_id	online_transactions.user_id	online_transactions.amount	online_transactions.location	online_transactions.timestamp
OB001	U031	500.0	AC001	2024-07-01 10:00:00+00:00
OB002	U032	600.0	AC002	2024-07-01 10:05:00+00:00
OB003	U033	700.0	AC003	2024-07-01 10:10:00+00:00
OB004	U034	800.0	AC004	2024-07-01 10:15:00+00:00
OB005	U035	900.0	AC005	2024-07-01 10:20:00+00:00
OB006	U036	1000.0	AC006	2024-07-01 10:25:00+00:00
OB007	U037	1100.0	AC007	2024-07-01 10:30:00+00:00
OB008	U038	1200.0	AC008	2024-07-01 10:35:00+00:00
OB009	U039	1300.0	AC009	2024-07-01 10:40:00+00:00
OB010	U040	1400.0	AC010	2024-07-01 10:45:00+00:00
OB011	U041	1500.0	AC011	2024-07-01 10:50:00+00:00
OB012	U042	1600.0	AC012	2024-07-01 10:55:00+00:00
OB013	U043	1700.0	AC013	2024-07-01 11:00:00+00:00
OB014	U044	1800.0	AC014	2024-07-01 11:05:00+00:00
OB015	U045	1900.0	AC015	2024-07-01 11:10:00+00:00
OB016	U046	2000.0	AC016	2024-07-01 11:15:00+00:00
OB017	U047	2100.0	AC017	2024-07-01 11:20:00+00:00
OB018	U048	2200.0	AC018	2024-07-01 11:25:00+00:00
OB019	U049	2300.0	AC019	2024-07-01 11:30:00+00:00
OB020	U050	2400.0	AC020	2024-07-01 11:35:00+00:00
OB021	U051	2500.0	AC021	2024-07-01 11:40:00+00:00
OB022	U052	2600.0	AC022	2024-07-01 11:45:00+00:00
OB023	U053	2700.0	AC023	2024-07-01 11:50:00+00:00
OB024	U054	2800.0	AC024	2024-07-01 11:55:00+00:00
OB025	U055	2900.0	AC025	2024-07-01 12:00:00+00:00
OB026	U056	3000.0	AC026	2024-07-01 12:05:00+00:00
OB027	U057	3100.0	AC027	2024-07-01 12:10:00+00:00
OB028	U058	3200.0	AC028	2024-07-01 12:15:00+00:00
OB029	U059	3300.0	AC029	2024-07-01 12:20:00+00:00
OB030	U060	3400.0	AC030	2024-07-01 12:25:00+00:00
30 rows selected (0.207 seconds)
0: jdbc:hive2://localhost:10000> SELECT * FROM mobile_transactions;
INFO : Compiling command(queryId=nandhumidhun_20240724150606_bbdfe6fa-8eea-48e7-8bf6-84b83fb327e8): SELECT * FROM mobile_transactions
INFO : No Stats for hadoop@mobile_transactions, Columns: transaction_id, amount, user_id, location, timestamp
INFO : Semantic Analysis Completed (retrial = false)
INFO : Created Hive schema: Schema(fieldSchemas:[FieldSchema(name:mobile_transactions.transaction_id, type:string, comment:null), FieldSchema(name:mobile_transactions.user_id, type:string, comment:null), FieldSchema(name:mobile_transactions.amount, type:double, comment:null), FieldSchema(name:mobile_transactions.location, type:string, comment:null), FieldSchema(name:mobile_transactions.timestamp, type:string, comment:null)], properties:null)
INFO : Completed compiling command(queryId=nandhumidhun_20240724150606_bbdfe6fa-8eea-48e7-8bf6-84b83fb327e8); Time taken: 0.168 seconds
INFO : Concurrency mode is disabled, not creating a lock manager
INFO : Executing command(queryId=nandhumidhun_20240724150606_bbdfe6fa-8eea-48e7-8bf6-84b83fb327e8): SELECT * FROM mobile_transactions


nandhumidhun@nandhumidhun-HP-Laptop-15-bs0xx: ~
nandhumidhun@nandhumidhun-HP-Laptop-15-bs0xx: ~/...
nandhumidhun@nandhumidhun-HP-Laptop-15-bs0xx: ~/...
nandhumidhun@nandhumidhun-HP-Laptop-15-bs0xx: ~
nandhumidhun@nandhumidhun-HP-Laptop-15-bs0xx: ~
OB027	U057	3100.0	AC027	2024-07-01 12:10:00+00:00
OB028	U058	3200.0	AC028	2024-07-01 12:15:00+00:00
OB029	U059	3300.0	AC029	2024-07-01 12:20:00+00:00
OB030	U060	3400.0	AC030	2024-07-01 12:25:00+00:00
30 rows selected (0.207 seconds)
0: jdbc:hive2://localhost:10000> SELECT * FROM mobile_transactions;
INFO : Compiling command(queryId=nandhumidhun_20240724150606_bbdfe6fa-8eea-48e7-8bf6-84b83fb327e8): SELECT * FROM mobile_transactions
INFO : No Stats for hadoop@mobile_transactions, Columns: transaction_id, amount, user_id, location, timestamp
INFO : Semantic Analysis Completed (retrial = false)
INFO : Created Hive schema: Schema(fieldSchemas:[FieldSchema(name:mobile_transactions.transaction_id, type:string, comment:null), FieldSchema(name:mobile_transactions.user_id, type:string, comment:null), FieldSchema(name:mobile_transactions.amount, type:double, comment:null), FieldSchema(name:mobile_transactions.location, type:string, comment:null), FieldSchema(name:mobile_transactions.timestamp, type:string, comment:null)], properties:null)
INFO : Completed compiling command(queryId=nandhumidhun_20240724150606_bbdfe6fa-8eea-48e7-8bf6-84b83fb327e8); Time taken: 0.168 seconds
INFO : Concurrency mode is disabled, not creating a lock manager
INFO : Executing command(queryId=nandhumidhun_20240724150606_bbdfe6fa-8eea-48e7-8bf6-84b83fb327e8): SELECT * FROM mobile_transactions
INFO : Completed executing command(queryId=nandhumidhun_20240724150606_bbdfe6fa-8eea-48e7-8bf6-84b83fb327e8); Time taken: 0.001 seconds
mobile_transactions.transaction_id	mobile_transactions.user_id	mobile_transactions.amount	mobile_transactions.location	mobile_transactions.timestamp
MB001	U061	150.0	D001	2024-07-01 10:00:00+00:00
MB002	U062	250.0	D002	2024-07-01 10:05:00+00:00
MB003	U063	350.0	D003	2024-07-01 10:10:00+00:00
MB004	U064	450.0	D004	2024-07-01 10:15:00+00:00
MB005	U065	550.0	D005	2024-07-01 10:20:00+00:00
MB006	U066	650.0	D006	2024-07-01 10:25:00+00:00
MB007	U067	750.0	D007	2024-07-01 10:30:00+00:00
MB008	U068	850.0	D008	2024-07-01 10:35:00+00:00
MB009	U069	950.0	D009	2024-07-01 10:40:00+00:00
MB010	U070	1050.0	D010	2024-07-01 10:45:00+00:00
MB011	U071	1150.0	D011	2024-07-01 10:50:00+00:00
MB012	U072	1250.0	D012	2024-07-01 10:55:00+00:00
MB013	U073	1350.0	D013	2024-07-01 11:00:00+00:00
MB014	U074	1450.0	D014	2024-07-01 11:05:00+00:00
MB015	U075	1550.0	D015	2024-07-01 11:10:00+00:00
MB016	U076	1650.0	D016	2024-07-01 11:15:00+00:00
MB017	U077	1750.0	D017	2024-07-01 11:20:00+00:00
MB018	U078	1850.0	D018	2024-07-01 11:25:00+00:00
MB019	U079	1950.0	D019	2024-07-01 11:30:00+00:00
MB020	U080	2050.0	D020	2024-07-01 11:35:00+00:00
MB021	U081	2150.0	D021	2024-07-01 11:40:00+00:00
MB022	U082	2250.0	D022	2024-07-01 11:45:00+00:00
MB023	U083	2350.0	D023	2024-07-01 11:50:00+00:00
MB024	U084	2450.0	D024	2024-07-01 11:55:00+00:00
MB025	U085	2550.0	D025	2024-07-01 12:00:00+00:00
MB026	U086	2650.0	D026	2024-07-01 12:05:00+00:00
MB027	U087	2750.0	D027	2024-07-01 12:10:00+00:00
MB028	U088	2850.0	D028	2024-07-01 12:15:00+00:00
MB029	U089	2950.0	D029	2024-07-01 12:20:00+00:00
MB030	U090	3050.0	D030	2024-07-01 12:25:00+00:00
30 rows selected (0.204 seconds)
0: jdbc:hive2://localhost:10000>


stream.py - test - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
TEST
ATM
kafka_consumer_atm.py
kafka_consumer_mobile.py
kafka_consumer_online.py
kafka_hadoop_atm.py
kafka_hadoop_mobile.py
kafka_hadoop_online.py
kafka_producer_atm.py
kafka_producer_mobile.py
kafka_producer_online.py
mobile_banking.csv
mobile_banking.json
online_banking.csv
online_banking.json
stream.py
checkpoint
kafka_topic
spark_log
csv_stream.py
data_processing.py
file.csv
OUTPUT
OUTLINE
TIMELINE
DEBUG CONSOLE
Filter (e.g. text, !exclude, \escape)
JAVA PROJECTS
line.py
kafka_consumer_mobile.py
kafka_consumer_online.py
kafka_stream.py
ATM_transaction.json
Batch_Processing.py
stream.py
BatchProcessing.java 9+
ATM > stream.py > ...
31])
32
33
34 # Convert the value column from bytes to string and parse JSON
35 df = df.selectExpr("CAST(value AS STRING)") \
PROBLEMS 11
TERMINAL
PORTS
bash
bash
bash ATM
python3 ATM
bash ATM
24/07/24 17:29:48 WARN AdminClientConfig: The configuration 'auto.offset.reset' was supplied but isn't a known config.
Batch: 10
+--------------+-------+------+--------+--------------------+
|transaction_id|user_id|amount|location| timestamp|
+--------------+-------+------+--------+--------------------+
ATM1	U1	100.0	ATM123	2024-07-01T10:00:00Z
ATM2	U2	50.0	ATM124	2024-07-01T10:05:00Z
ATM3	U3	200.0	ATM125	2024-07-01T10:10:00Z
ATM4	U4	80.0	ATM126	2024-07-01T10:15:00Z
ATM5	U5	150.0	ATM127	2024-07-01T10:20:00Z
ATM6	U6	30.0	ATM128	2024-07-01T10:25:00Z
ATM7	U7	60.0	ATM129	2024-07-01T10:30:00Z
ATM8	U8	40.0	ATM130	2024-07-01T10:35:00Z
ATM9	U9	120.0	ATM131	2024-07-01T10:40:00Z
ATM10	U10	70.0	ATM132	2024-07-01T10:45:00Z
ATM11	U1	90.0	ATM133	2024-07-01T10:50:00Z
ATM12	U2	110.0	ATM134	2024-07-01T10:55:00Z
ATM13	U3	140.0	ATM135	2024-07-01T11:00:00Z
ATM14	U4	160.0	ATM136	2024-07-01T11:05:00Z
ATM15	U5	130.0	ATM137	2024-07-01T11:10:00Z
ATM16	U6	95.0	ATM138	2024-07-01T11:15:00Z
ATM17	U7	85.0	ATM139	2024-07-01T11:20:00Z
ATM18	U8	75.0	ATM140	2024-07-01T11:25:00Z
ATM19	U9	65.0	ATM141	2024-07-01T11:30:00Z
ATM20	U10	55.0	ATM142	2024-07-01T11:35:00Z
+--------------+-------+------+--------+--------------------+
only showing top 20 rows
Batch: 11
+--------------+-------+------+--------+--------------------+
|transaction_id|user_id|amount|location| timestamp|
+--------------+-------+------+--------+--------------------+
| ATM1| U1| 100.0| ATM123|2024-07-01T10:00:00Z|
+--------------+-------+------+--------+--------------------+
Batch: 12
+--------------+-------+------+--------+--------------------+
|transaction_id|user_id|amount|location| timestamp|
+--------------+-------+------+--------+--------------------+
Ln 50, Col 41
Spaces: 4
UTF-8
LF
Python
3.10.12 64-bit
Java: Warning


Batch_Processing.py - CUBEAISOLUTIONS - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
CUBEAISOLUTIONS
ATM
batch_output
ATM_transaction.csv
ATM_transaction.json
Batch_Processing.py
convert_json_csv.py
fra.py
fraud_detection_model.joblib
fraud_detection_model.pkl
fruad.py
history.csv
job.py
kafka_consumer_atm.py
kafka_consumer_mobile.py
kafka_consumer_online.py
kafka_hadoop_atm.py
kafka_hadoop_mobile.py
OUTPUT
OUTLINE
TIMELINE
DEBUG CONSOLE
Filter (e.g. text, !exclude, \escape)
DataAnalysis.java
kafka_Producer.py
stream_data.csv
kafka_consumer.py
kafka_postgresql.py
kafka_stream.py
Batch_Processing.py
ATM_transaction.json
ATM > Batch_Processing.py > ...
StructField("user_id", StringType()),
StructField("amount", StringType()),
StructField("location", StringType()),
StructField("timestamp", StringType())
])
# Read the data from a batch source (e.g., JSON files in local file system)
input_path = "file:////home/nandhumidhun/CUBEAISOLUTIONS/ATM/ATM_transaction.json" # Ensure this path is correct
df = spark.read \
.format("json") \
.schema(schema) \
.load(input_path)
# Optionally perform transformations or actions on the DataFrame
df.show(truncate=False)
# Save the DataFrame to a file or database (optional)
output_path = "/home/nandhumidhun/CUBEAISOLUTIONS/ATM/batch_output"
df.write \
.mode("overwrite") \
.format("parquet") \
.save(output_path)
PROBLEMS
TERMINAL
PORTS
bash ATM
bash ATM
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
24/07/25 14:33:57 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
+--------------+-------+------+--------+--------------------+
|transaction_id|user_id|amount|location|timestamp |
+--------------+-------+------+--------+--------------------+
NULL	NULL	NULL	NULL	NULL
ATM1	U1	100.0	ATM123	2024-07-01T10:00:00Z
ATM2	U2	50.0	ATM124	2024-07-01T10:05:00Z
ATM3	U3	200.0	ATM125	2024-07-01T10:10:00Z
ATM4	U4	80.0	ATM126	2024-07-01T10:15:00Z
ATM5	U5	150.0	ATM127	2024-07-01T10:20:00Z
ATM6	U6	30.0	ATM128	2024-07-01T10:25:00Z
ATM7	U7	60.0	ATM129	2024-07-01T10:30:00Z
ATM8	U8	40.0	ATM130	2024-07-01T10:35:00Z
ATM9	U9	120.0	ATM131	2024-07-01T10:40:00Z
ATM10	U10	70.0	ATM132	2024-07-01T10:45:00Z
ATM11	U1	90.0	ATM133	2024-07-01T10:50:00Z
ATM12	U2	110.0	ATM134	2024-07-01T10:55:00Z
ATM13	U3	140.0	ATM135	2024-07-01T11:00:00Z
ATM14	U4	160.0	ATM136	2024-07-01T11:05:00Z
ATM15	U5	130.0	ATM137	2024-07-01T11:10:00Z
ATM16	U6	95.0	ATM138	2024-07-01T11:15:00Z
ATM17	U7	85.0	ATM139	2024-07-01T11:20:00Z
ATM18	U8	75.0	ATM140	2024-07-01T11:25:00Z
ATM19	U9	65.0	ATM141	2024-07-01T11:30:00Z
Ln 21, Col 82
Spaces: 4
UTF-8
LF
Python
3.10.12 64-bit

```
Real-time Stock Market
OUTPUT
OUTLINE
TIMELINE
DEBUG CONSOLE
Filter (e.g. text, !exclude, \escape)
```

```
Not Fraudulent
nandhumidhun@nandhumidhun-MP-Laptop-15-bs0xx:~/CUBEAISOLUTIONS/ATM$ python3 job_amount.py
Fraudulent
Alert: Fraud detected for transaction: {'amount': 3420}
nandhumidhun@nandhumidhun-MP-Laptop-15-bs0xx:~/CUBEAISOLUTIONS/ATM$ python3 job_amount.py
Not Fraudulent
nandhumidhun@nandhumidhun-MP-Laptop-15-bs0xx:~/CUBEAISOLUTIONS/ATM$ python3 user_amount.py
Not Fraudulent: {'amount': 100, 'user_id': 'U1'}
Not Fraudulent: {'amount': 50, 'user_id': 'U2'}
Not Fraudulent: {'amount': 200, 'user_id': 'U3'}
Not Fraudulent: {'amount': 80, 'user_id': 'U4'}
Not Fraudulent: {'amount': 150, 'user_id': 'U5'}
Not Fraudulent: {'amount': 30, 'user_id': 'U6'}
Not Fraudulent: {'amount': 60, 'user_id': 'U7'}
Not Fraudulent: {'amount': 40, 'user_id': 'U8'}
Not Fraudulent: {'amount': 120, 'user_id': 'U9'}
Not Fraudulent: {'amount': 70, 'user_id': 'U10'}
Not Fraudulent: {'amount': 90, 'user_id': 'U1'}
Not Fraudulent: {'amount': 110, 'user_id': 'U2'}
Not Fraudulent: {'amount': 140, 'user_id': 'U3'}
Not Fraudulent: {'amount': 160, 'user_id': 'U4'}
```

Ln 19, Col 1 Spaces: 4 UTF-8 LF Python 3.10.12 64-bit

ChatGPT | (4) WhatsApp | Untitled-1 | jinja2.exceptions.Templa

https://analytics.zoho.in/workspace/365624000000002019/view/365624000000024002

Home | Data_Visualizati... | fraud_amount | fraud | Untitled-1

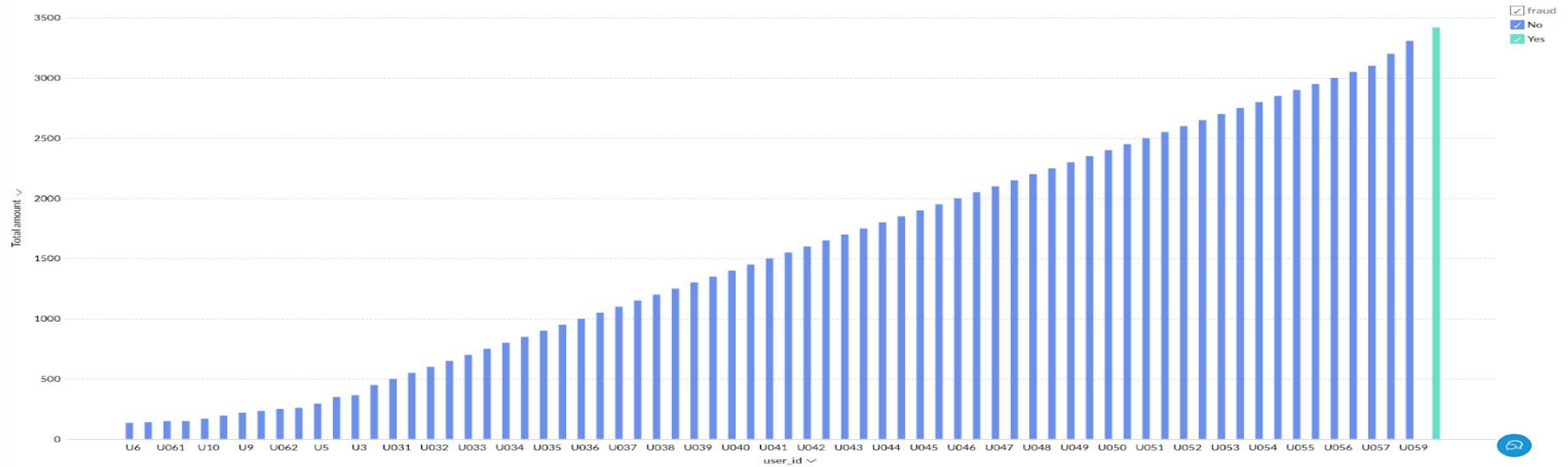
Premium Trial - 8 days left UPGRADE

Create | Explorer | Dashboards | Reports | Data | Ask Zia | Data Sources | Settings | Trash | Viewer

Untitled-1 Save

Edit Design | Insights

Sort | Underlying Data



Here is your Smart Chat (Ctrl+Space)

Credit card fraud Detection

STEP 1: Create kafka topic
STEP 2: Verify the data are correctly send to kafka topic
STEP 3: Real-time Streaming Process
STEP 4: Detect the fraud by using the email_id
STEP 5: load the data into kinesis data stream by using lambda

kafka_producer.py - test - Visual Studio Code

File Edit Selection View Go Run Terminal Help

EXPLORER

TEST

Credit_Card

kafka_producer_csv.py
kafka_producer1.py
KafkaStreamProcessing.class
KafkaStreamProcessing.java 9+
LeaderElection.class
LeaderElection.java 9+
path_to_threads.py
reassignment.json
stream.py
Stream.py
Transaction.py

credit_card1

credit-card.csv
credit-card.json
credit-card.zip
csv_json.py
kafka_consumer.py
kafka_consumer1.py
kafka_producer.py
kafka_streaming.py
kafkaproducer.py
KafkaTransaction.class
KafkaTransaction.java 9+
test.py

E-Commerce Analysis
kafka_topic

TIMELINE
DEBUG CONSOLE

Filter (e.g. text, !exclude, \escape)

OUTLINE
OUTPUT
JAVA PROJECTS

jit-card.json | kafkaproducer.py | kafka_consumer1.py | credit-card.csv | kafka_producer.py | kafka_consumer.py credit_card1 | test.py | KafkaTransaction.java 9+

credit_card1 > kafka_producer.py > ...

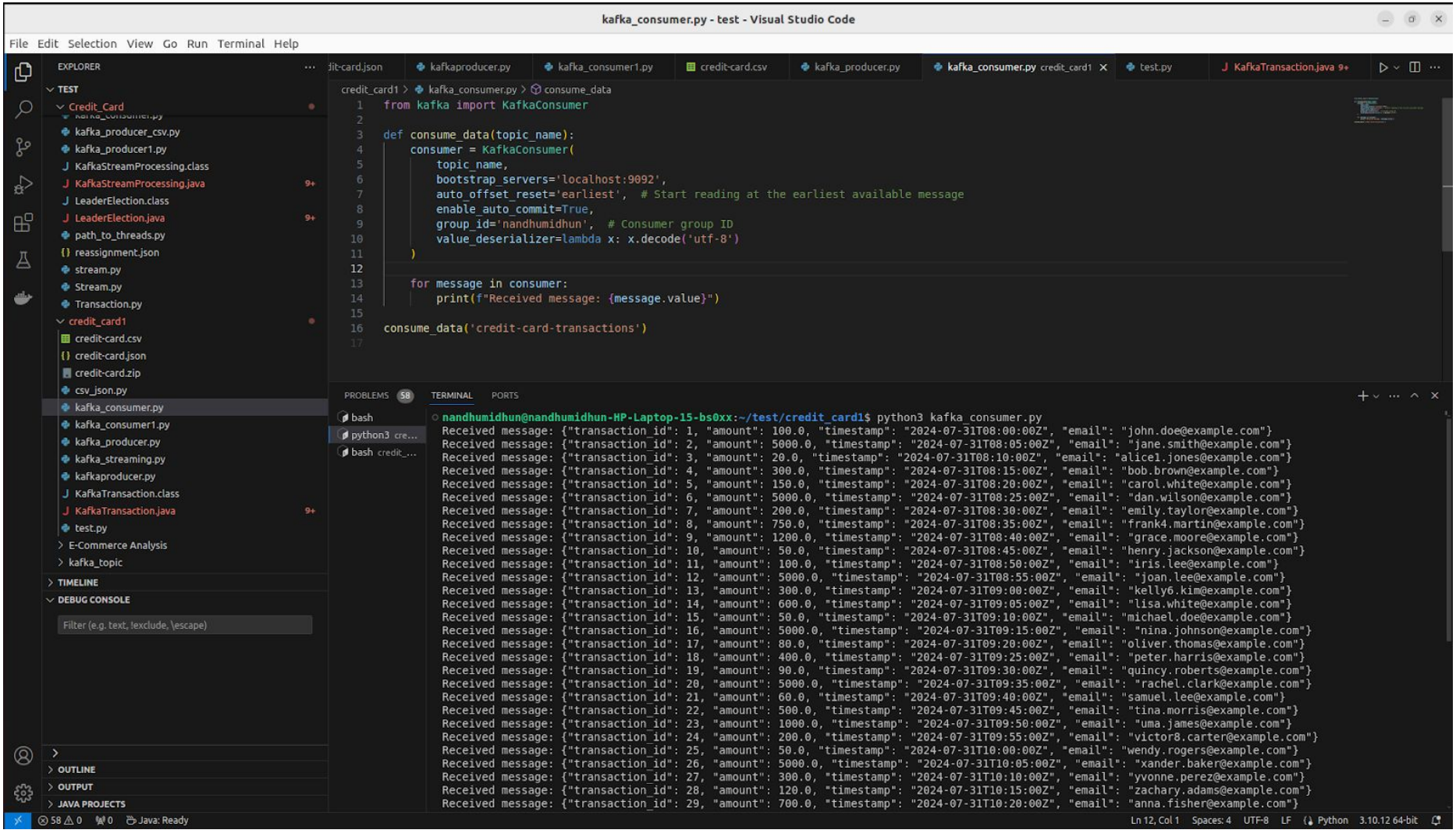
```python
# Read the CSV file
csv_file_path = 'credit-card.csv'
df = pd.read_csv(csv_file_path)

# Create a Kafka producer
producer = KafkaProducer(
    bootstrap_servers='localhost:9092',
    value_serializer=lambda v: json.dumps(v).encode('utf-8')
)

# Send each row in the CSV file to the Kafka topic
topic_name = 'credit-card-transactions'
for _, row in df.iterrows():
    message = row.to_dict()
    producer.send(topic_name, message)

# Ensure all messages are sent before exiting
producer.flush()

print("All messages sent to Kafka topic successfully.")
```

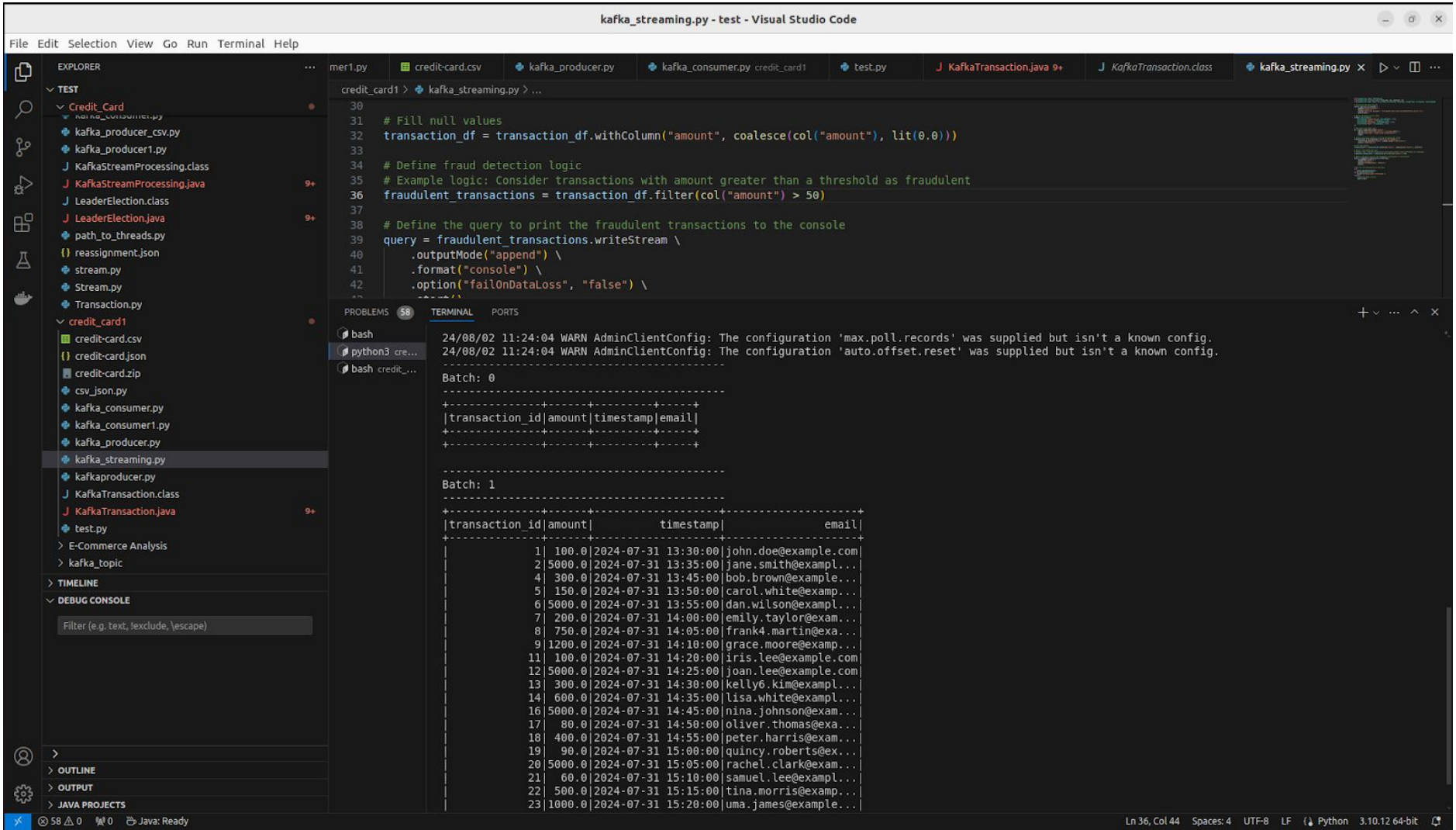
PROBLEMS 58 | TERMINAL | PORTS

bash
bash credit_...

```
nandhumidhun@nandhumidhun-HP-Laptop-15-bs0xx:~/test/credit_card1$ python3 kafka_producer.py
All messages sent to Kafka topic successfully.
nandhumidhun@nandhumidhun-HP-Laptop-15-bs0xx:~/test/credit_card1$ python3 kafka_producer.py
All messages sent to Kafka topic successfully.
nandhumidhun@nandhumidhun-HP-Laptop-15-bs0xx:~/test/credit_card1$
```

⊗ 58 ⚠ 0 | 0 | Java: Ready | Ln 24, Col 1 | Spaces: 4 | UTF-8 | LF | Python | 3.10.12 64-bit


kafka_consumer.py - test - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
TEST
Credit_Card
kafka_producer_csv.py
kafka_producer1.py
KafkaStreamProcessing.class
KafkaStreamProcessing.java
LeaderElection.class
LeaderElection.java
path_to_threads.py
reassignment.json
stream.py
Stream.py
Transaction.py
credit_card1
credit-card.csv
credit-card.json
credit-card.zip
csv_json.py
kafka_consumer.py
kafka_consumer1.py
kafka_producer.py
kafka_streaming.py
kafkaproducer.py
KafkaTransaction.class
KafkaTransaction.java
test.py
E-Commerce Analysis
kafka_topic
TIMELINE
DEBUG CONSOLE
Filter (e.g. text, !exclude, \escape)
OUTLINE
OUTPUT
JAVA PROJECTS
jit-card.json
kafkaproducer.py
kafka_consumer1.py
credit-card.csv
kafka_producer.py
kafka_consumer.py credit_card1
test.py
KafkaTransaction.java
credit_card1 > kafka_consumer.py > consume_data
1 from kafka import KafkaConsumer
2
3 def consume_data(topic_name):
4 consumer = KafkaConsumer(
5 topic_name,
6 bootstrap_servers='localhost:9092',
7 auto_offset_reset='earliest', # Start reading at the earliest available message
8 enable_auto_commit=True,
9 group_id='nandhumidhun', # Consumer group ID
10 value_deserializer=lambda x: x.decode('utf-8')
11)
12
13 for message in consumer:
14 print(f"Received message: {message.value}")
15
16 consume_data('credit-card-transactions')
17
PROBLEMS 58
TERMINAL
PORTS
bash
python3 cre...
bash credit_...
nandhumidhun@nandhumidhun-HP-Laptop-15-bs0xx:~/test/credit_card1$ python3 kafka_consumer.py
Received message: {"transaction_id": 1, "amount": 100.0, "timestamp": "2024-07-31T08:00:00Z", "email": "john.doe@example.com"}
Received message: {"transaction_id": 2, "amount": 5000.0, "timestamp": "2024-07-31T08:05:00Z", "email": "jane.smith@example.com"}
Received message: {"transaction_id": 3, "amount": 20.0, "timestamp": "2024-07-31T08:10:00Z", "email": "alice1.jones@example.com"}
Received message: {"transaction_id": 4, "amount": 300.0, "timestamp": "2024-07-31T08:15:00Z", "email": "bob.brown@example.com"}
Received message: {"transaction_id": 5, "amount": 150.0, "timestamp": "2024-07-31T08:20:00Z", "email": "carol.white@example.com"}
Received message: {"transaction_id": 6, "amount": 5000.0, "timestamp": "2024-07-31T08:25:00Z", "email": "dan.wilson@example.com"}
Received message: {"transaction_id": 7, "amount": 200.0, "timestamp": "2024-07-31T08:30:00Z", "email": "emily.taylor@example.com"}
Received message: {"transaction_id": 8, "amount": 750.0, "timestamp": "2024-07-31T08:35:00Z", "email": "frank4.martin@example.com"}
Received message: {"transaction_id": 9, "amount": 1200.0, "timestamp": "2024-07-31T08:40:00Z", "email": "grace.moore@example.com"}
Received message: {"transaction_id": 10, "amount": 50.0, "timestamp": "2024-07-31T08:45:00Z", "email": "henry.jackson@example.com"}
Received message: {"transaction_id": 11, "amount": 100.0, "timestamp": "2024-07-31T08:50:00Z", "email": "iris.lee@example.com"}
Received message: {"transaction_id": 12, "amount": 5000.0, "timestamp": "2024-07-31T08:55:00Z", "email": "joan.lee@example.com"}
Received message: {"transaction_id": 13, "amount": 300.0, "timestamp": "2024-07-31T09:00:00Z", "email": "kelly6.kim@example.com"}
Received message: {"transaction_id": 14, "amount": 600.0, "timestamp": "2024-07-31T09:05:00Z", "email": "lisa.white@example.com"}
Received message: {"transaction_id": 15, "amount": 50.0, "timestamp": "2024-07-31T09:10:00Z", "email": "michael.doe@example.com"}
Received message: {"transaction_id": 16, "amount": 5000.0, "timestamp": "2024-07-31T09:15:00Z", "email": "nina.johnson@example.com"}
Received message: {"transaction_id": 17, "amount": 80.0, "timestamp": "2024-07-31T09:20:00Z", "email": "oliver.thomas@example.com"}
Received message: {"transaction_id": 18, "amount": 400.0, "timestamp": "2024-07-31T09:25:00Z", "email": "peter.harris@example.com"}
Received message: {"transaction_id": 19, "amount": 90.0, "timestamp": "2024-07-31T09:30:00Z", "email": "quincy.roberts@example.com"}
Received message: {"transaction_id": 20, "amount": 5000.0, "timestamp": "2024-07-31T09:35:00Z", "email": "rachel.clark@example.com"}
Received message: {"transaction_id": 21, "amount": 60.0, "timestamp": "2024-07-31T09:40:00Z", "email": "samuel.lee@example.com"}
Received message: {"transaction_id": 22, "amount": 500.0, "timestamp": "2024-07-31T09:45:00Z", "email": "tina.morris@example.com"}
Received message: {"transaction_id": 23, "amount": 1000.0, "timestamp": "2024-07-31T09:50:00Z", "email": "uma.james@example.com"}
Received message: {"transaction_id": 24, "amount": 200.0, "timestamp": "2024-07-31T09:55:00Z", "email": "victor8.carter@example.com"}
Received message: {"transaction_id": 25, "amount": 50.0, "timestamp": "2024-07-31T10:00:00Z", "email": "wendy.rogers@example.com"}
Received message: {"transaction_id": 26, "amount": 5000.0, "timestamp": "2024-07-31T10:05:00Z", "email": "xander.baker@example.com"}
Received message: {"transaction_id": 27, "amount": 300.0, "timestamp": "2024-07-31T10:10:00Z", "email": "yvonne.perez@example.com"}
Received message: {"transaction_id": 28, "amount": 120.0, "timestamp": "2024-07-31T10:15:00Z", "email": "zachary.adams@example.com"}
Received message: {"transaction_id": 29, "amount": 700.0, "timestamp": "2024-07-31T10:20:00Z", "email": "anna.fisher@example.com"}
Java: Ready
Ln 12, Col 1
Spaces: 4
UTF-8
LF
Python
3.10.12 64-bit


kafka_streaming.py - test - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
TEST
Credit_Card
kafka_producer_csv.py
kafka_producer1.py
KafkaStreamProcessing.class
KafkaStreamProcessing.java
LeaderElection.class
LeaderElection.java
path_to_threads.py
reassignment.json
stream.py
Stream.py
Transaction.py
credit_card1
credit-card.csv
credit-card.json
credit-card.zip
csv_json.py
kafka_consumer.py
kafka_consumer1.py
kafka_producer.py
kafka_streaming.py
kafkaproducer.py
KafkaTransaction.class
KafkaTransaction.java
test.py
E-Commerce Analysis
kafka_topic
TIMELINE
DEBUG CONSOLE
Filter (e.g. text, !exclude, \escape)
OUTLINE
OUTPUT
JAVA PROJECTS
credit_card1 > kafka_streaming.py > ...
30
31 # Fill null values
32 transaction_df = transaction_df.withColumn("amount", coalesce(col("amount"), lit(0.0)))
33
34 # Define fraud detection logic
35 # Example logic: Consider transactions with amount greater than a threshold as fraudulent
36 fraudulent_transactions = transaction_df.filter(col("amount") > 50)
37
38 # Define the query to print the fraudulent transactions to the console
39 query = fraudulent_transactions.writeStream \
40 .outputMode("append") \
41 .format("console") \
42 .option("failOnDataLoss", "false") \
PROBLEMS 58
TERMINAL
PORTS
bash
python3 cre...
bash credit_...
24/08/02 11:24:04 WARN AdminClientConfig: The configuration 'max.poll.records' was supplied but isn't a known config.
24/08/02 11:24:04 WARN AdminClientConfig: The configuration 'auto.offset.reset' was supplied but isn't a known config.
Batch: 0
+-------------+------+---------+-----+
|transaction_id|amount|timestamp|email|
+-------------+------+---------+-----+
+-------------+------+---------+-----+
Batch: 1
+-------------+------+-------------------+--------------------+
|transaction_id|amount| timestamp| email|
+-------------+------+-------------------+--------------------+
1	100.0	2024-07-31 13:30:00	john.doe@example.com
2	5000.0	2024-07-31 13:35:00	jane.smith@exampl...
4	300.0	2024-07-31 13:45:00	bob.brown@example...
5	150.0	2024-07-31 13:50:00	carol.white@examp...
6	5000.0	2024-07-31 13:55:00	dan.wilson@exampl...
7	200.0	2024-07-31 14:00:00	emily.taylor@exam...
8	750.0	2024-07-31 14:05:00	frank4.martin@exa...
9	1200.0	2024-07-31 14:10:00	grace.moore@examp...
11	100.0	2024-07-31 14:20:00	iris.lee@example.com
12	5000.0	2024-07-31 14:25:00	joan.lee@example.com
13	300.0	2024-07-31 14:30:00	kelly6.kim@exampl...
14	600.0	2024-07-31 14:35:00	lisa.white@exampl...
16	5000.0	2024-07-31 14:45:00	nina.johnson@exam...
17	80.0	2024-07-31 14:50:00	oliver.thomas@exa...
18	400.0	2024-07-31 14:55:00	peter.harris@exam...
19	90.0	2024-07-31 15:00:00	quincy.roberts@ex...
20	5000.0	2024-07-31 15:05:00	rachel.clark@exam...
21	60.0	2024-07-31 15:10:00	samuel.lee@exampl...
22	500.0	2024-07-31 15:15:00	tina.morris@examp...
23	1000.0	2024-07-31 15:20:00	uma.james@example...
Ln 36, Col 44
Spaces: 4
UTF-8
LF
Python
3.10.12 64-bit
Java: Ready

KafkaTransaction.java - test - Visual Studio Code

File Edit Selection View Go Run Terminal Help

EXPLORER

TEST

Credit_Card

kafka_producer_csv.py
kafka_producer1.py
KafkaStreamProcessing.class
KafkaStreamProcessing.java 9+
LeaderElection.class
LeaderElection.java 9+
path_to_threads.py
reassignment.json
stream.py
Stream.py
Transaction.py

credit_card1

credit-card.csv
credit-card.json
credit-card.zip
csv_json.py
kafka_consumer.py
kafka_consumer1.py
kafka_producer.py
kafka_streaming.py
kafkaproducer.py
KafkaTransaction.class
KafkaTransaction.java 9+
test.py

E-Commerce Analysis
kafka_topic

TIMELINE
DEBUG CONSOLE

Filter (e.g. text, !exclude, \escape)

OUTLINE
OUTPUT
JAVA PROJECTS

mer1.py | credit-card.csv | kafka_producer.py | kafka_consumer.py credit_card1 | test.py | KafkaTransaction.java 9+ | KafkaTransaction.class | kafka_streaming.py

credit_card1 > KafkaTransaction.java > ...

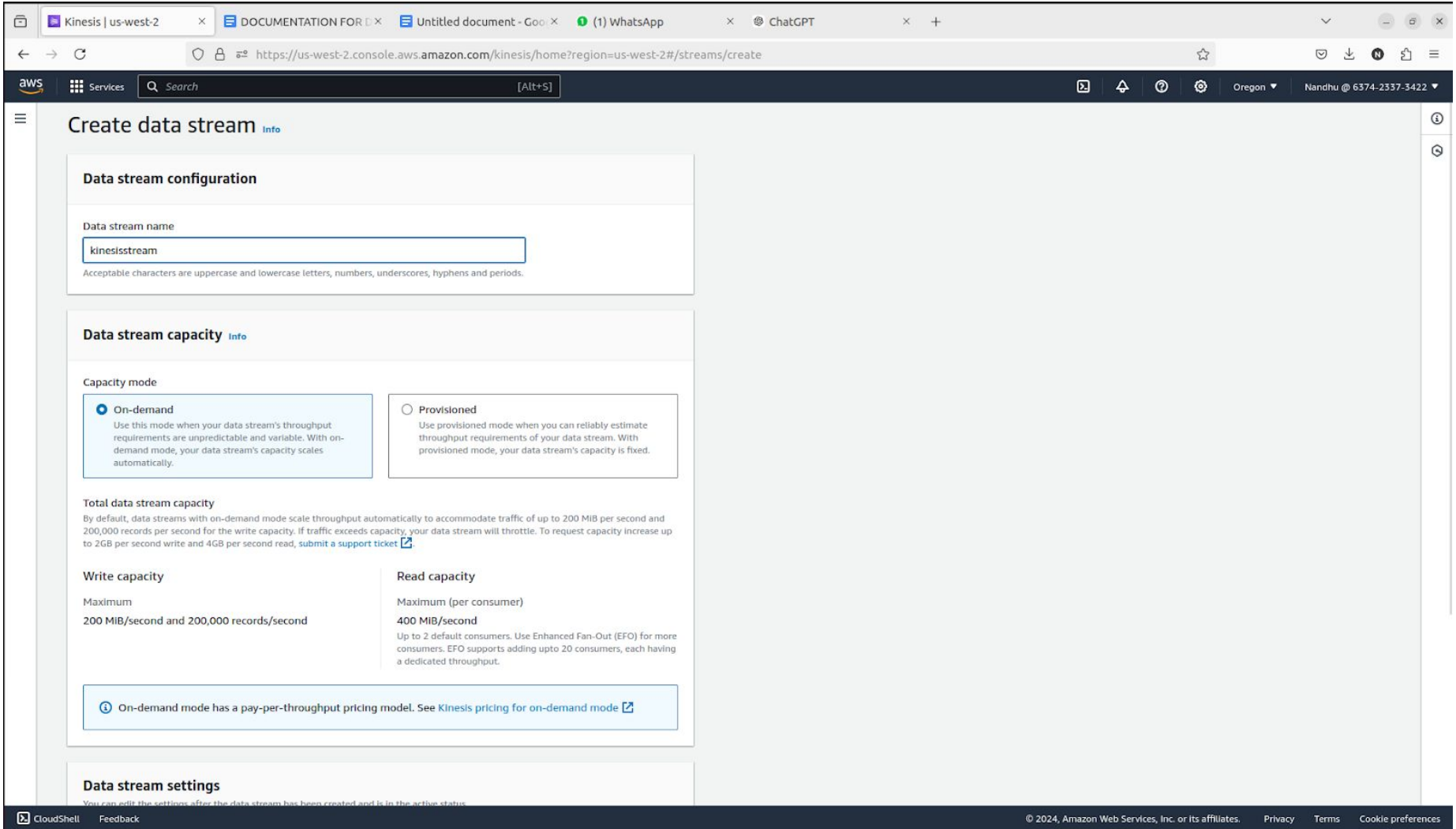
```
11  public class KafkaTransaction {
13      public static void main(String[] args) {
31          // Consume and check messages
32          while (true) {
33              consumer.poll(java.time.Duration.ofMillis(millis:100)).forEach(record -> {
34                  try {
35                      JsonNode jsonNode = mapper.readTree(record.value());
36                      String email = jsonNode.path("email").asText(null);
37                      int transactionId = jsonNode.path("transaction_id").asInt(-1);
38
39                      // Check if the email matches the pattern and if transaction_id meets criteria
40                      if (email != null && emailPattern.matcher(email).matches()) {
41                          System.out.println("Fraudulent Transaction: Email: " + email + ", Transaction ID: " + transactionId);
42                      } else {
```

PROBLEMS 58 | TERMINAL | PORTS

bash
java credit_c...
bash credit_...

```
ction-consumer-group] Found no committed offset for partition credit-card-transactions-0
[main] INFO org.apache.kafka.clients.consumer.internals.SubscriptionState - [Consumer clientId=consumer-transaction-consumer-group-1, groupId=transaction-consumer-group] Resetting offset for partition credit-card-transactions-0 to position FetchPosition{offset=0, offsetEpoch=Optional.empty, currentLeader=LeaderAndEpoch{leader=Optional[nandhumidhun-HP-Laptop-15-bs0xx:9092 (id: 0 rack: null)], epoch=0}}.
Normal Transaction: Email: john.doe@example.com, Transaction ID: 1
Normal Transaction: Email: jane.smith@example.com, Transaction ID: 2
Fraudulent Transaction: Email: alice1.jones@example.com, Transaction ID: 3
Normal Transaction: Email: bob.brown@example.com, Transaction ID: 4
Normal Transaction: Email: carol.white@example.com, Transaction ID: 5
Normal Transaction: Email: dan.wilson@example.com, Transaction ID: 6
Normal Transaction: Email: emily.taylor@example.com, Transaction ID: 7
Fraudulent Transaction: Email: frank4.martin@example.com, Transaction ID: 8
Normal Transaction: Email: grace.moore@example.com, Transaction ID: 9
Normal Transaction: Email: henry.jackson@example.com, Transaction ID: 10
Normal Transaction: Email: iris.lee@example.com, Transaction ID: 11
Normal Transaction: Email: joan.lee@example.com, Transaction ID: 12
Fraudulent Transaction: Email: kelly6.kim@example.com, Transaction ID: 13
Normal Transaction: Email: lisa.white@example.com, Transaction ID: 14
Normal Transaction: Email: michael.doe@example.com, Transaction ID: 15
Normal Transaction: Email: nina.johnson@example.com, Transaction ID: 16
Normal Transaction: Email: oliver.thomas@example.com, Transaction ID: 17
Normal Transaction: Email: peter.harris@example.com, Transaction ID: 18
Normal Transaction: Email: quincy.roberts@example.com, Transaction ID: 19
Normal Transaction: Email: rachel.clark@example.com, Transaction ID: 20
Normal Transaction: Email: samuel.lee@example.com, Transaction ID: 21
Normal Transaction: Email: tina.morris@example.com, Transaction ID: 22
Normal Transaction: Email: uma.james@example.com, Transaction ID: 23
Fraudulent Transaction: Email: victor8.carter@example.com, Transaction ID: 24
Normal Transaction: Email: wendy.rogers@example.com, Transaction ID: 25
Normal Transaction: Email: xander.baker@example.com, Transaction ID: 26
Normal Transaction: Email: yvonne.perez@example.com, Transaction ID: 27
Normal Transaction: Email: zachary.adams@example.com, Transaction ID: 28
Normal Transaction: Email: anna.fisher@example.com, Transaction ID: 29
Normal Transaction: Email: benjamin.green@example.com, Transaction ID: 30
Normal Transaction: Email: claire.nelson@example.com, Transaction ID: 31
Normal Transaction: Email: david.morris@example.com, Transaction ID: 32
```

Ln 56, Col 57 Spaces: 4 UTF-8 LF Java


Kinesis | us-west-2
DOCUMENTATION FOR D
Untitled document - Goo
(1) WhatsApp
ChatGPT
https://us-west-2.console.aws.amazon.com/kinesis/home?region=us-west-2#/streams/create
aws
Services
Search
[Alt+S]
Oregon
Nandhu @ 6374-2337-3422
Create data stream Info
Data stream configuration
Data stream name
kinesisstream
Acceptable characters are uppercase and lowercase letters, numbers, underscores, hyphens and periods.
Data stream capacity Info
Capacity mode
On-demand
Use this mode when your data stream's throughput requirements are unpredictable and variable. With on-demand mode, your data stream's capacity scales automatically.
Provisioned
Use provisioned mode when you can reliably estimate throughput requirements of your data stream. With provisioned mode, your data stream's capacity is fixed.
Total data stream capacity
By default, data streams with on-demand mode scale throughput automatically to accommodate traffic of up to 200 MiB per second and 200,000 records per second for the write capacity. If traffic exceeds capacity, your data stream will throttle. To request capacity increase up to 2GB per second write and 4GB per second read, submit a support ticket
Write capacity
Maximum
200 MiB/second and 200,000 records/second
Read capacity
Maximum (per consumer)
400 MiB/second
Up to 2 default consumers. Use Enhanced Fan-Out (EFO) for more consumers. EFO supports adding upto 20 consumers, each having a dedicated throughput.
On-demand mode has a pay-per-throughput pricing model. See Kinesis pricing for on-demand mode
Data stream settings
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

kinesis.py - AWS - Visual Studio Code

File Edit Selection View Go Run Terminal Help

EXPLORER

AWS
- __pycache__
- aws_lambda.py
- check_dynamo.py
- check_lambda.py
- check.py
- dynamo.py
- dynamo1.py
- kafka_kinesis.py
- kafka_to_s3.py
- kinesis.py
- knesis.py
- lambda.py
- load_file.py
- response.json

TIMELINE

DEBUG CONSOLE

Filter (e.g. text, !exclude, \escape)

OUTLINE

OUTPUT

dynamo.py | dynamo1.py | kinesis.py | kafka_kinesis.py | check.py | check_dynamo.py | lambda.py | knesis.py | check_lambda.py | kafka_tx

kinesis.py > ...

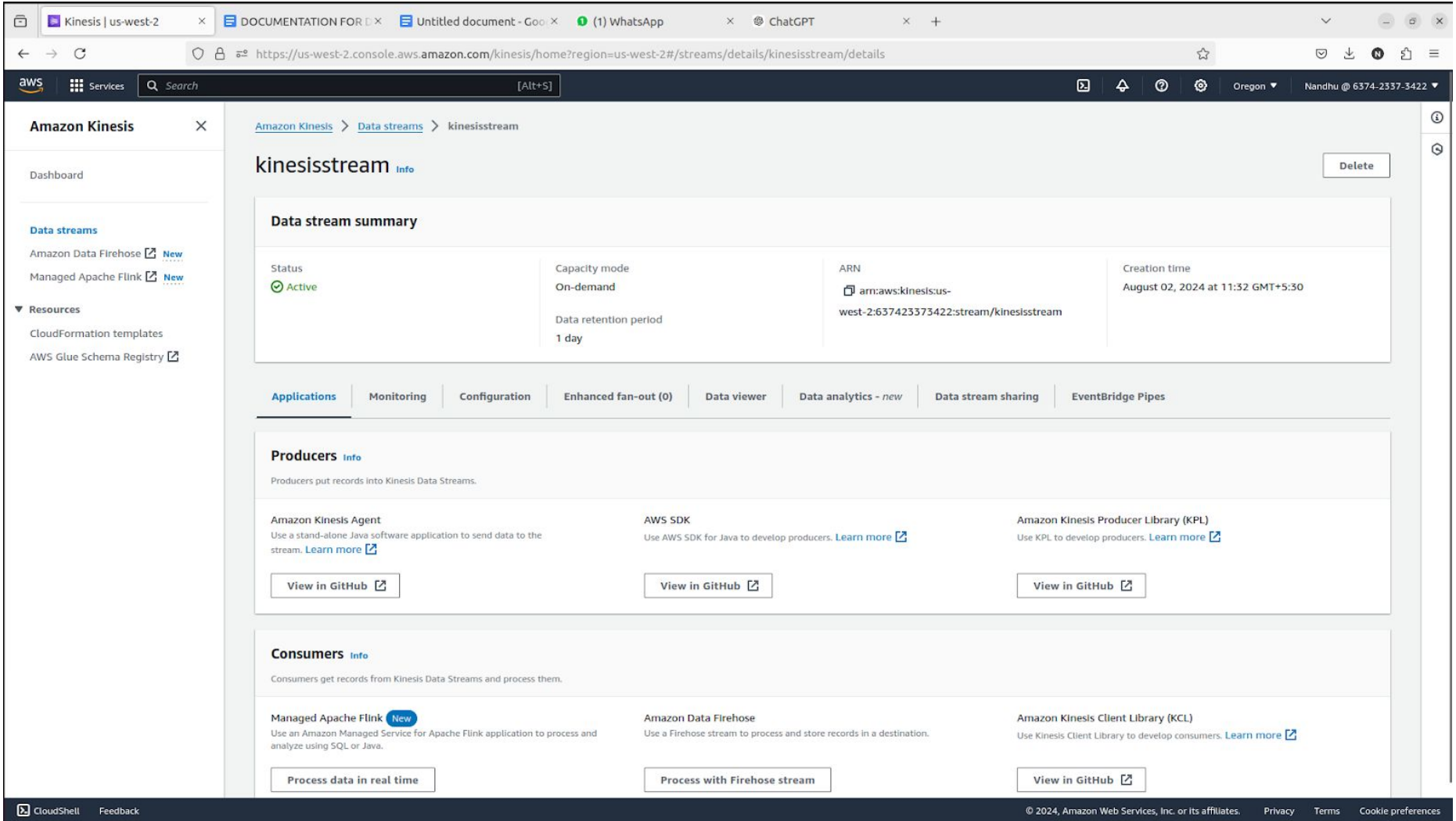
```python
import boto3
import csv
import json
from time import sleep

# Define Kinesis stream details
my_stream_name = 'kinesisstream'

# Initialize Kinesis client without specifying credentials directly
kinesis_client = boto3.client('kinesis', region_name='us-west-2')

# Path to the CSV file
csv_file_path = '/home/nandhumidhun/test/credit_card1/credit-card.csv'

# Read data from CSV file and send to Kinesis
with open(csv_file_path, mode='r') as csv_file:
    csv_reader = csv.DictReader(csv_file)
    for row in csv_reader:
        json_data = {
            "transaction_id": row["transaction_id"],
            "amount": row["amount"],
            "timestamp": row["timestamp"],
            "email": row["email"]
        }
        print(json_data)  # Print the JSON data
        kinesis_client.put_record(
```

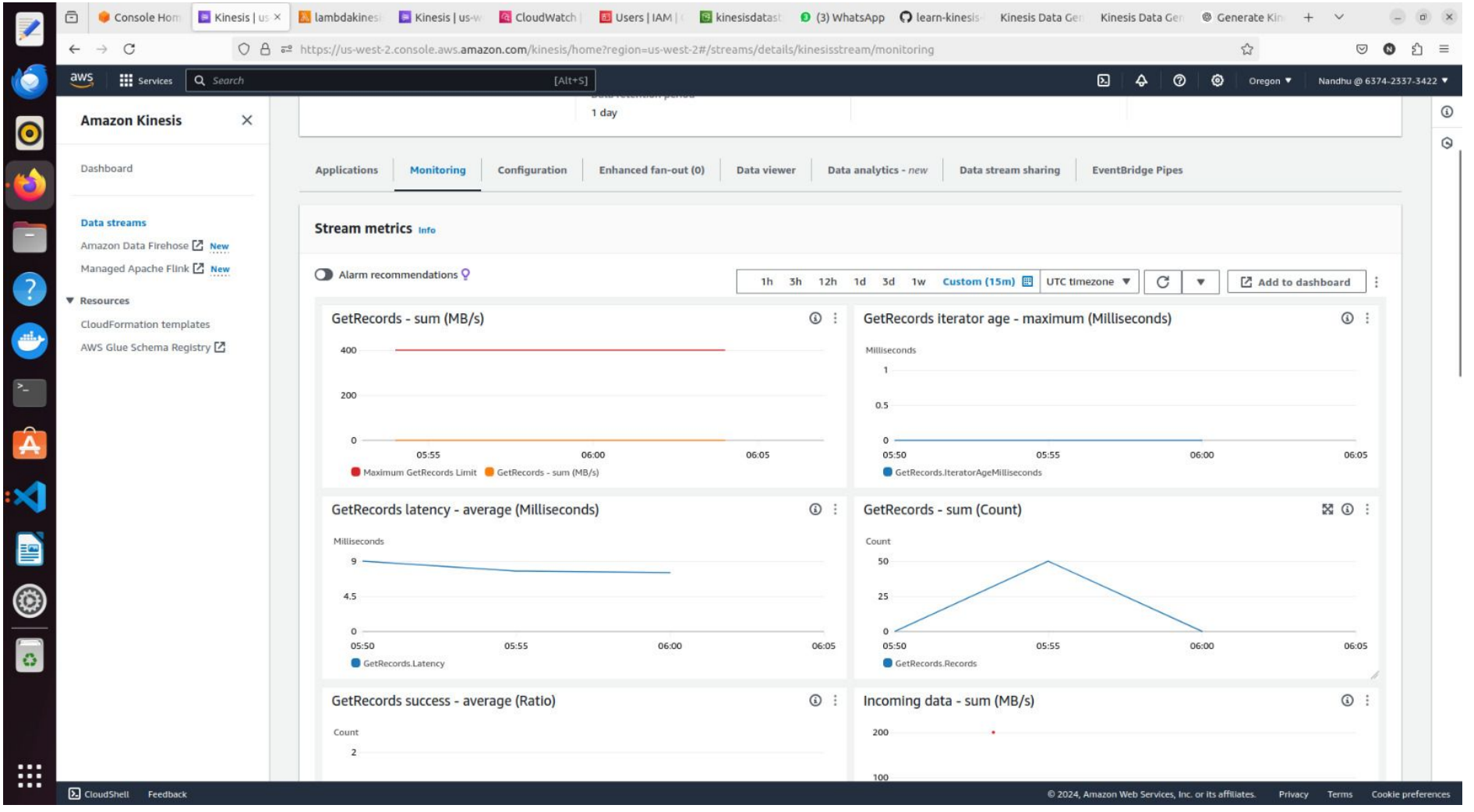
PROBLEMS TERMINAL PORTS

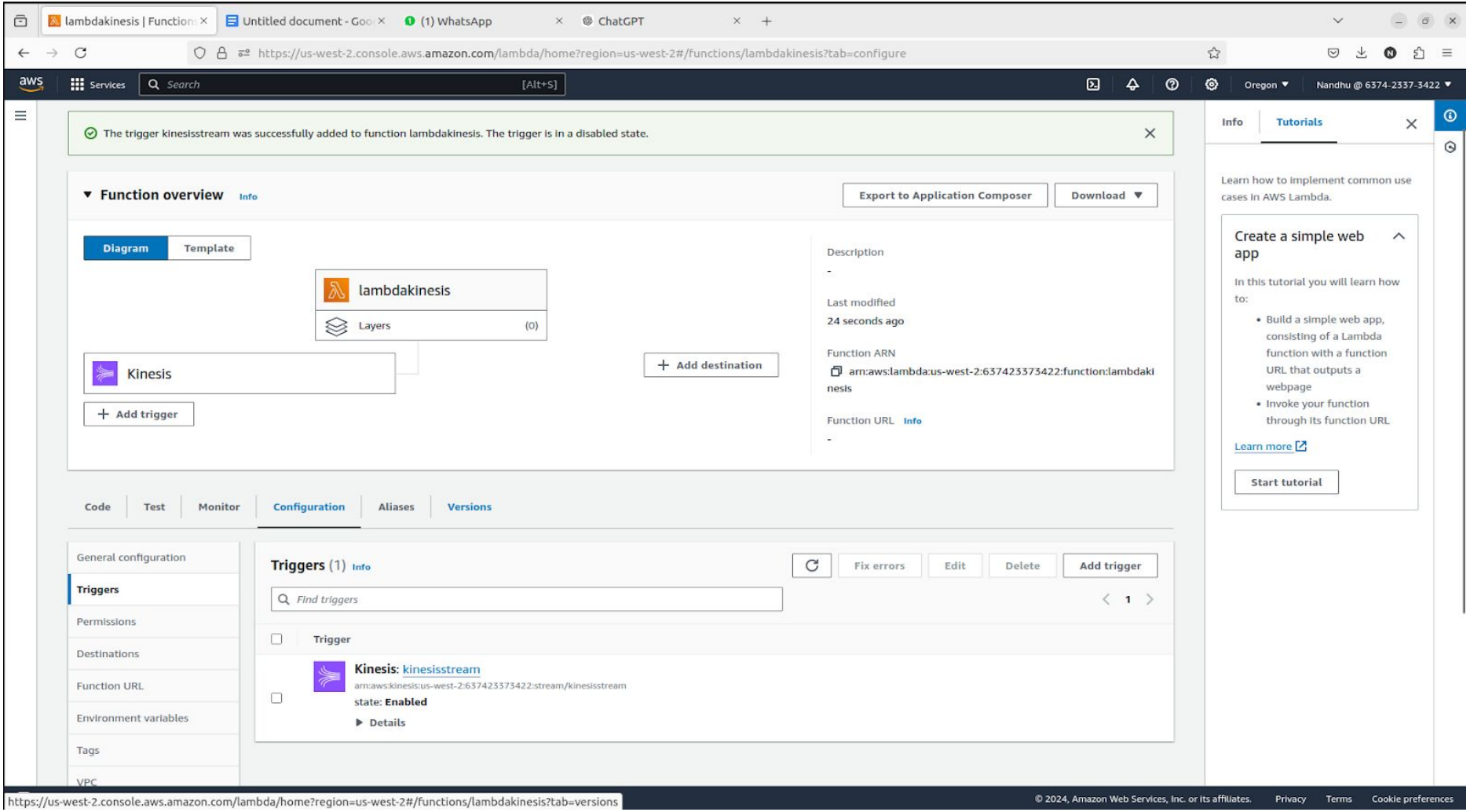
bash

```
nandhumidhun@nandhumidhun-HP-Laptop-15-bs0xx:~/AWS$ python3 kinesis.py
{'transaction_id': '1', 'amount': '100.00', 'timestamp': '2024-07-31T08:00:00Z', 'email': 'john.doe@example.com'}
{'transaction_id': '2', 'amount': '5000.00', 'timestamp': '2024-07-31T08:05:00Z', 'email': 'jane.smith@example.com'}
{'transaction_id': '3', 'amount': '20.00', 'timestamp': '2024-07-31T08:10:00Z', 'email': 'alice1.jones@example.com'}
{'transaction_id': '4', 'amount': '300.00', 'timestamp': '2024-07-31T08:15:00Z', 'email': 'bob.brown@example.com'}
{'transaction_id': '5', 'amount': '150.00', 'timestamp': '2024-07-31T08:20:00Z', 'email': 'carol.white@example.com'}
{'transaction_id': '6', 'amount': '5000.00', 'timestamp': '2024-07-31T08:25:00Z', 'email': 'dan.wilson@example.com'}
{'transaction_id': '7', 'amount': '200.00', 'timestamp': '2024-07-31T08:30:00Z', 'email': 'emily.taylor@example.com'}
{'transaction_id': '8', 'amount': '750.00', 'timestamp': '2024-07-31T08:35:00Z', 'email': 'frank4.martin@example.com'}
{'transaction_id': '9', 'amount': '1200.00', 'timestamp': '2024-07-31T08:40:00Z', 'email': 'grace.moore@example.com'}
{'transaction_id': '10', 'amount': '50.00', 'timestamp': '2024-07-31T08:45:00Z', 'email': 'henry.jackson@example.com'}
{'transaction_id': '11', 'amount': '100.00', 'timestamp': '2024-07-31T08:50:00Z', 'email': 'iris.lee@example.com'}
{'transaction_id': '12', 'amount': '5000.00', 'timestamp': '2024-07-31T08:55:00Z', 'email': 'joan.lee@example.com'}
{'transaction_id': '13', 'amount': '300.00', 'timestamp': '2024-07-31T09:00:00Z', 'email': 'kelly6.kim@example.com'}
{'transaction_id': '14', 'amount': '600.00', 'timestamp': '2024-07-31T09:05:00Z', 'email': 'lisa.white@example.com'}
{'transaction_id': '15', 'amount': '50.00', 'timestamp': '2024-07-31T09:10:00Z', 'email': 'michael.doe@example.com'}
{'transaction_id': '16', 'amount': '5000.00', 'timestamp': '2024-07-31T09:15:00Z', 'email': 'nina.johnson@example.com'}
{'transaction_id': '17', 'amount': '80.00', 'timestamp': '2024-07-31T09:20:00Z', 'email': 'oliver.thomas@example.com'}
{'transaction_id': '18', 'amount': '400.00', 'timestamp': '2024-07-31T09:25:00Z', 'email': 'peter.harris@example.com'}
{'transaction_id': '19', 'amount': '90.00', 'timestamp': '2024-07-31T09:30:00Z', 'email': 'quincy.roberts@example.com'}
{'transaction_id': '20', 'amount': '5000.00', 'timestamp': '2024-07-31T09:35:00Z', 'email': 'rachel.clark@example.com'}
{'transaction_id': '21', 'amount': '60.00', 'timestamp': '2024-07-31T09:40:00Z', 'email': 'samuel.lee@example.com'}
```

Ln 25, Col 48 Spaces: 4 UTF-8 LF Python 3.10.12 64-bit


Kinesis | us-west-2
DOCUMENTATION FOR
Untitled document - Goo
(1) WhatsApp
ChatGPT
https://us-west-2.console.aws.amazon.com/kinesis/home?region=us-west-2#/streams/details/kinesisstream/details
aws
Services
Search
[Alt+S]
Oregon
Nandhu @ 6374-2337-3422
Amazon Kinesis
Dashboard
Data streams
Amazon Data Firehose New
Managed Apache Flink New
Resources
CloudFormation templates
AWS Glue Schema Registry
Amazon Kinesis > Data streams > kinesisstream
kinesisstream Info
Delete
Data stream summary
Status
Active
Capacity mode
On-demand
Data retention period
1 day
ARN
arn:aws:kinesis:us-west-2:637423373422:stream/kinesisstream
Creation time
August 02, 2024 at 11:32 GMT+5:30
Applications
Monitoring
Configuration
Enhanced fan-out (0)
Data viewer
Data analytics - new
Data stream sharing
EventBridge Pipes
Producers Info
Producers put records into Kinesis Data Streams.
Amazon Kinesis Agent
Use a stand-alone Java software application to send data to the stream. Learn more
View in GitHub
AWS SDK
Use AWS SDK for Java to develop producers. Learn more
View in GitHub
Amazon Kinesis Producer Library (KPL)
Use KPL to develop producers. Learn more
View in GitHub
Consumers Info
Consumers get records from Kinesis Data Streams and process them.
Managed Apache Flink New
Use an Amazon Managed Service for Apache Flink application to process and analyze using SQL or Java.
Process data in real time
Amazon Data Firehose
Use a Firehose stream to process and store records in a destination.
Process with Firehose stream
Amazon Kinesis Client Library (KCL)
Use Kinesis Client Library to develop consumers. Learn more
View in GitHub
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


Console Hom
Kinesis | us
lambdakines
Kinesis | us-w
CloudWatch
Users | IAM |
kinesisdatast
(3) WhatsApp
learn-kinesis
Kinesis Data Ge
Kinesis Data Ge
Generate Kin
https://us-west-2.console.aws.amazon.com/kinesis/home?region=us-west-2#/streams/details/kinesisstream/monitoring
aws
Services
Search
[Alt+S]
Oregon
Nandhu @ 6374-2337-3422
Amazon Kinesis
Dashboard
Data streams
Amazon Data Firehose New
Managed Apache Flink New
Resources
CloudFormation templates
AWS Glue Schema Registry
1 day
Applications
Monitoring
Configuration
Enhanced fan-out (0)
Data viewer
Data analytics - new
Data stream sharing
EventBridge Pipes
Stream metrics Info
Alarm recommendations
1h 3h 12h 1d 3d 1w Custom (15m)
UTC timezone
Add to dashboard
GetRecords - sum (MB/s)
400
200
0
05:55
06:00
06:05
Maximum GetRecords Limit
GetRecords - sum (MB/s)
GetRecords iterator age - maximum (Milliseconds)
Milliseconds
1
0.5
0
05:50
05:55
06:00
06:05
GetRecords.IteratorAgeMilliseconds
GetRecords latency - average (Milliseconds)
Milliseconds
9
4.5
0
05:50
05:55
06:00
06:05
GetRecords.Latency
GetRecords - sum (Count)
Count
50
25
0
05:50
05:55
06:00
06:05
GetRecords.Records
GetRecords success - average (Ratio)
Count
2
Incoming data - sum (MB/s)
200
100
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


lambdakinesis | Function
Untitled document - Goo
(1) WhatsApp
ChatGPT
https://us-west-2.console.aws.amazon.com/lambda/home?region=us-west-2#/functions/lambdakinesis?tab=configure
Services
Search
[Alt+S]
Oregon
Nandhu @ 6374-2337-3422
The trigger kinesisstream was successfully added to function lambdakinesis. The trigger is in a disabled state.
Function overview
Info
Export to Application Composer
Download
Diagram
Template
lambdakinesis
Layers
(0)
Kinesis
Add trigger
Add destination
Description
-
Last modified
24 seconds ago
Function ARN
arn:aws:lambda:us-west-2:637423373422:function:lambdakinesis
Function URL
Info
-
Code
Test
Monitor
Configuration
Aliases
Versions
General configuration
Triggers
Permissions
Destinations
Function URL
Environment variables
Tags
VPC
Triggers (1)
Info
Fix errors
Edit
Delete
Add trigger
Find triggers
1
Trigger
Kinesis: kinesisstream
arn:aws:kinesis:us-west-2:637423373422:stream/kinesisstream
state: Enabled
Details
Info
Tutorials
Learn how to implement common use cases in AWS Lambda.
Create a simple web app
In this tutorial you will learn how to:
Build a simple web app, consisting of a Lambda function with a function URL that outputs a webpage
Invoke your function through its function URL
Learn more
Start tutorial
https://us-west-2.console.aws.amazon.com/lambda/home?region=us-west-2#/functions/lambdakinesis?tab=versions
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

ETL PROCESS IN AWS

STEP 1: Create an S3 Bucket
STEP 2: Set Up AWS Glue
STEP 3: Amazon Redshift
STEP 4: Data Loading
STEP 5: Schedule the ETL Job
STEP 6: Testing and Validation

Create bucket Info

Buckets are containers for data stored in S3.

General configuration

AWS Region

US West (Oregon) us-west-2

Bucket type Info

- (●) **General purpose**
  Recommended for most use cases and access patterns. General purpose buckets are the original S3 bucket type. They allow a mix of storage classes that redundantly store objects across multiple Availability Zones.
- ( ) **Directory - *New***
  Recommended for low-latency use cases. These buckets use only the S3 Express One Zone storage class, which provides faster processing of data within a single Availability Zone.

Bucket name Info

localdatainput

Bucket name must be unique within the global namespace and follow the bucket naming rules. See rules for bucket naming

Copy settings from existing bucket - *optional*

Only the bucket settings in the following configuration are copied.

Choose bucket

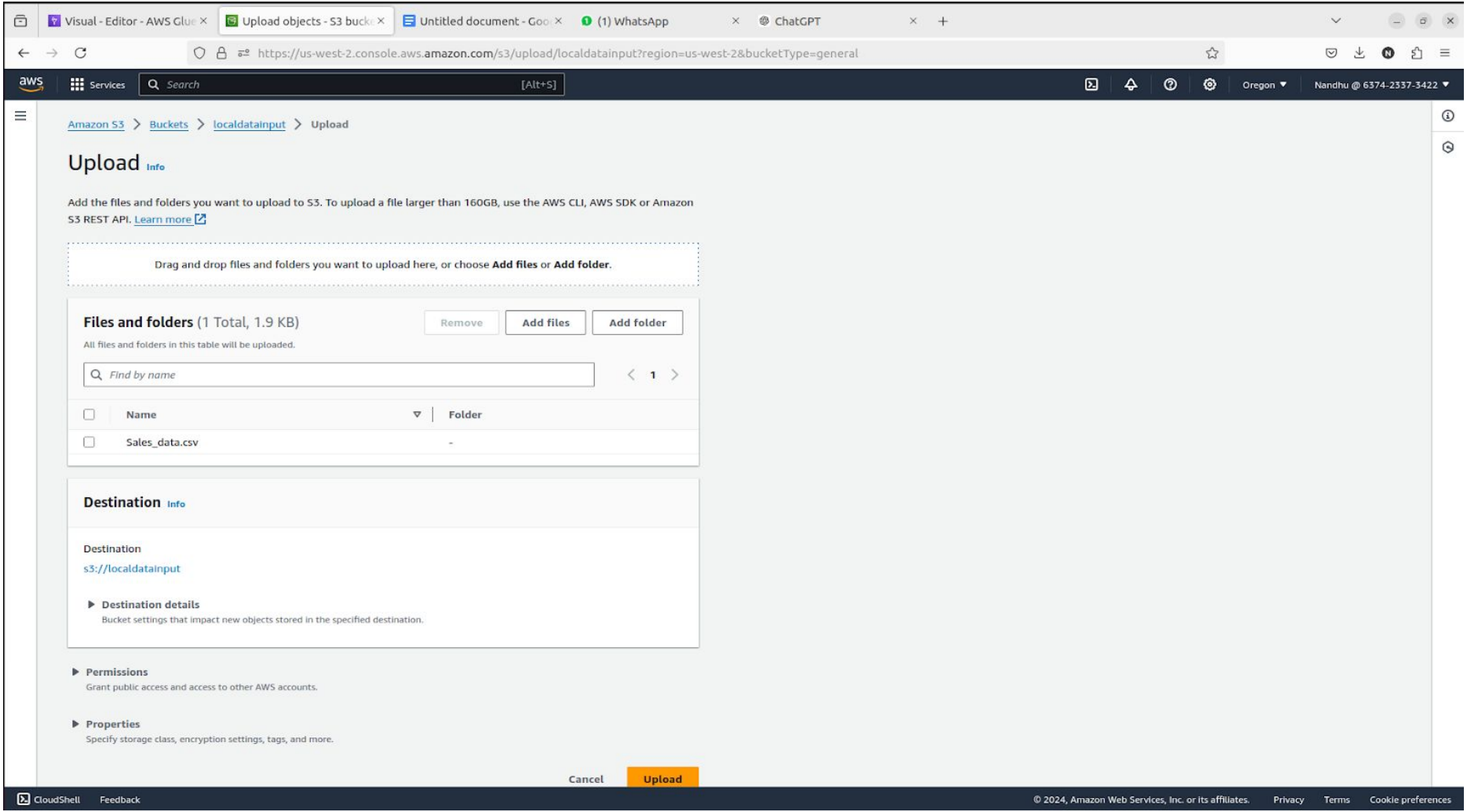
Format: s3://bucket/prefix

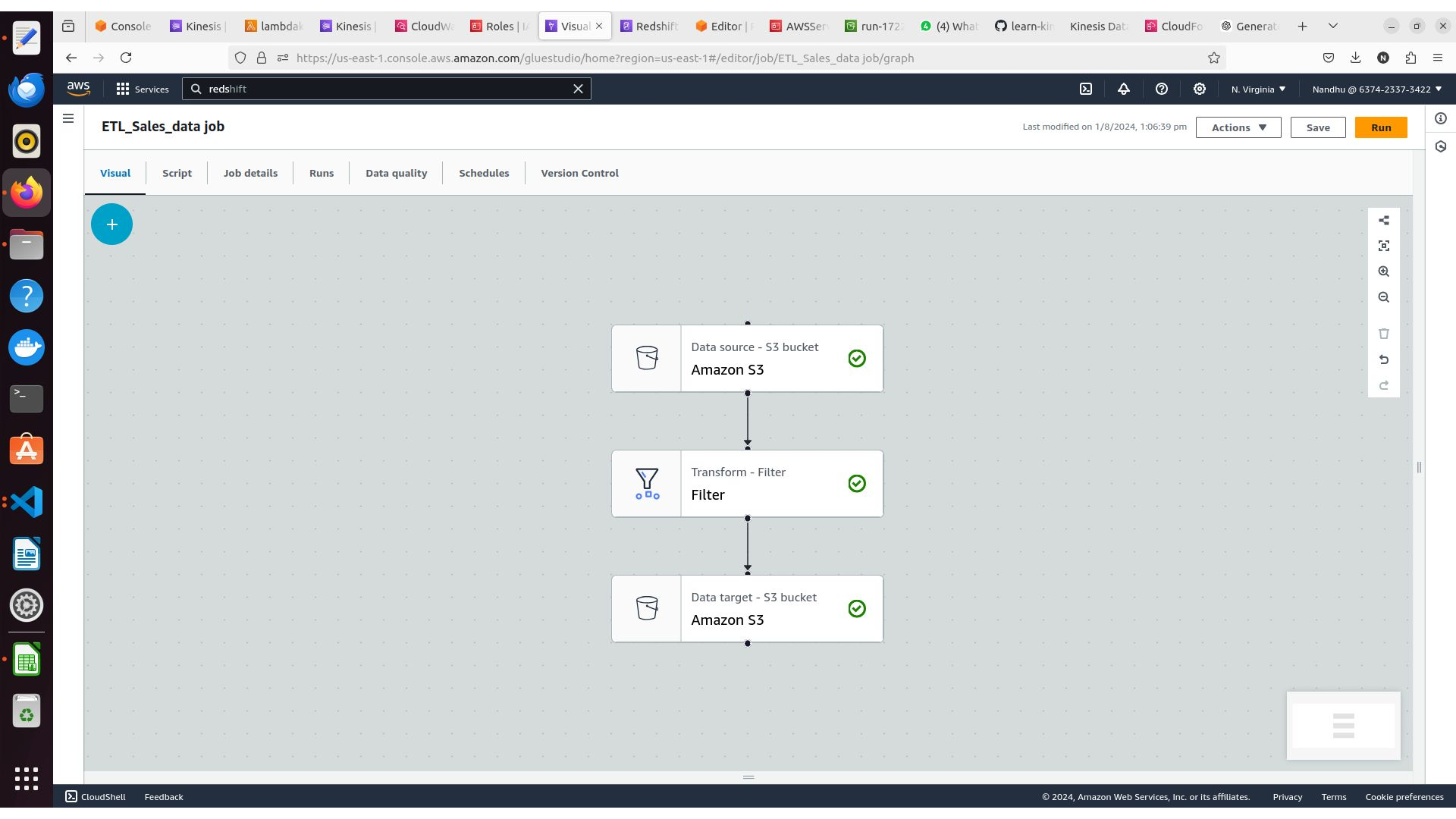
Object Ownership Info

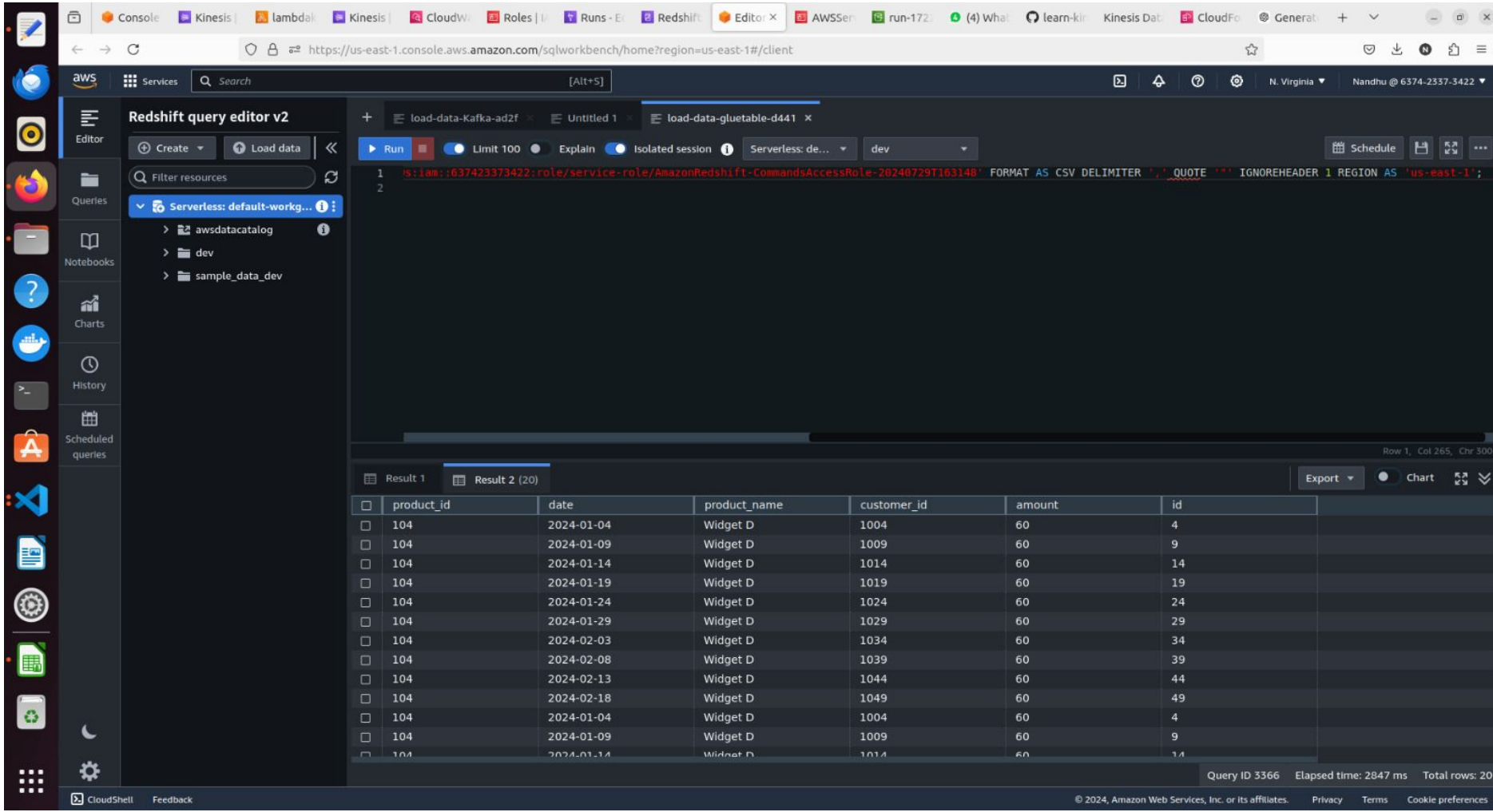
Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.

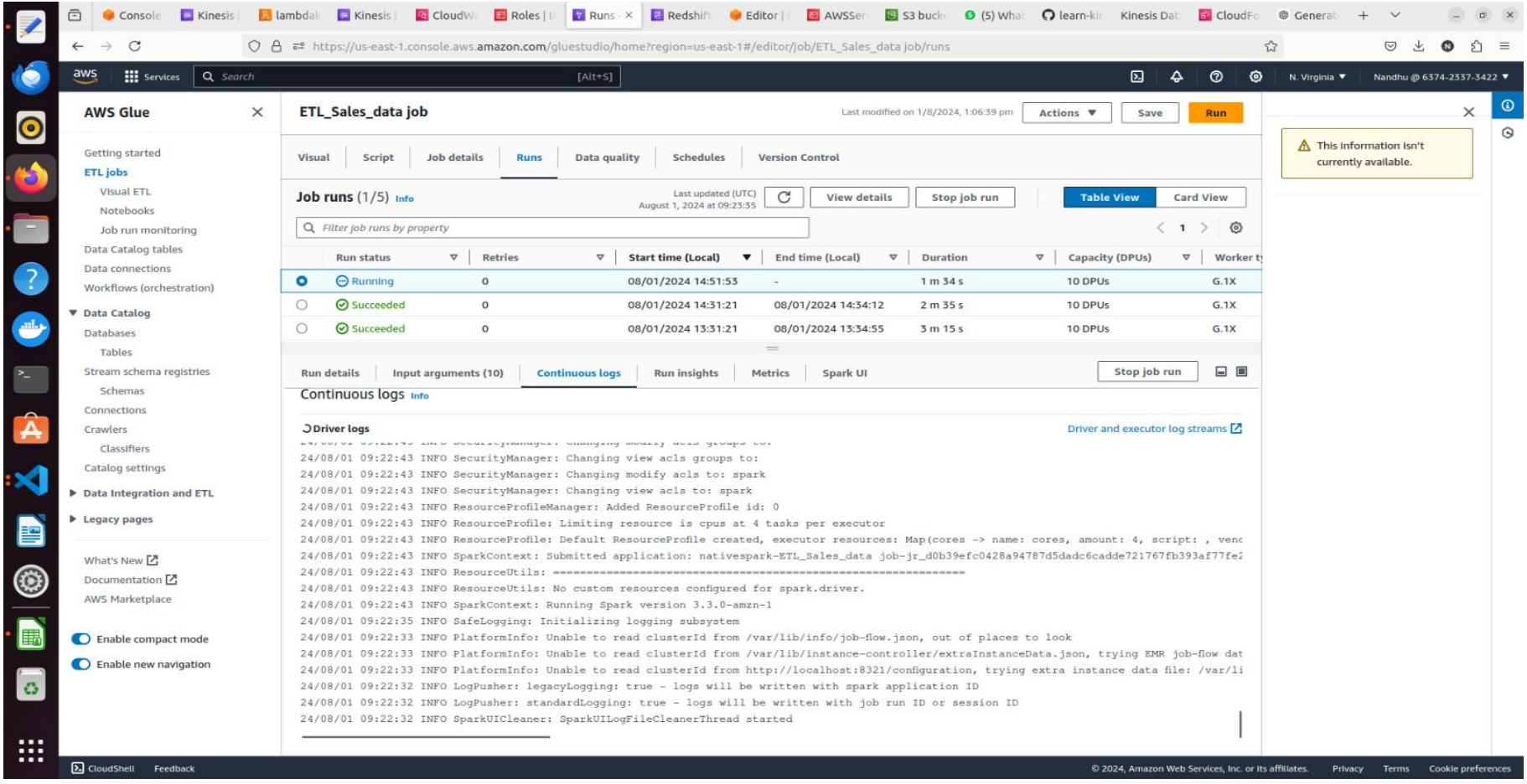
- (●) **ACLs disabled (recommended)**
  All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.
- ( ) **ACLs enabled**
  Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.

Object Ownership


Visual - Editor - AWS Glue
Upload objects - S3 buck
Untitled document - Goo
(1) WhatsApp
ChatGPT
https://us-west-2.console.aws.amazon.com/s3/upload/localdatainput?region=us-west-2&bucketType=general
aws
Services
Search
[Alt+S]
Oregon
Nandhu @ 6374-2337-3422
Amazon S3 > Buckets > localdatainput > Upload
Upload Info
Add the files and folders you want to upload to S3. To upload a file larger than 160GB, use the AWS CLI, AWS SDK or Amazon S3 REST API. Learn more
Drag and drop files and folders you want to upload here, or choose Add files or Add folder.
Files and folders (1 Total, 1.9 KB)
Remove
Add files
Add folder
All files and folders in this table will be uploaded.
Find by name
1
Name
Folder
Sales_data.csv
-
Destination Info
Destination
s3://localdatainput
Destination details
Bucket settings that impact new objects stored in the specified destination.
Permissions
Grant public access and access to other AWS accounts.
Properties
Specify storage class, encryption settings, tags, and more.
Cancel
Upload
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

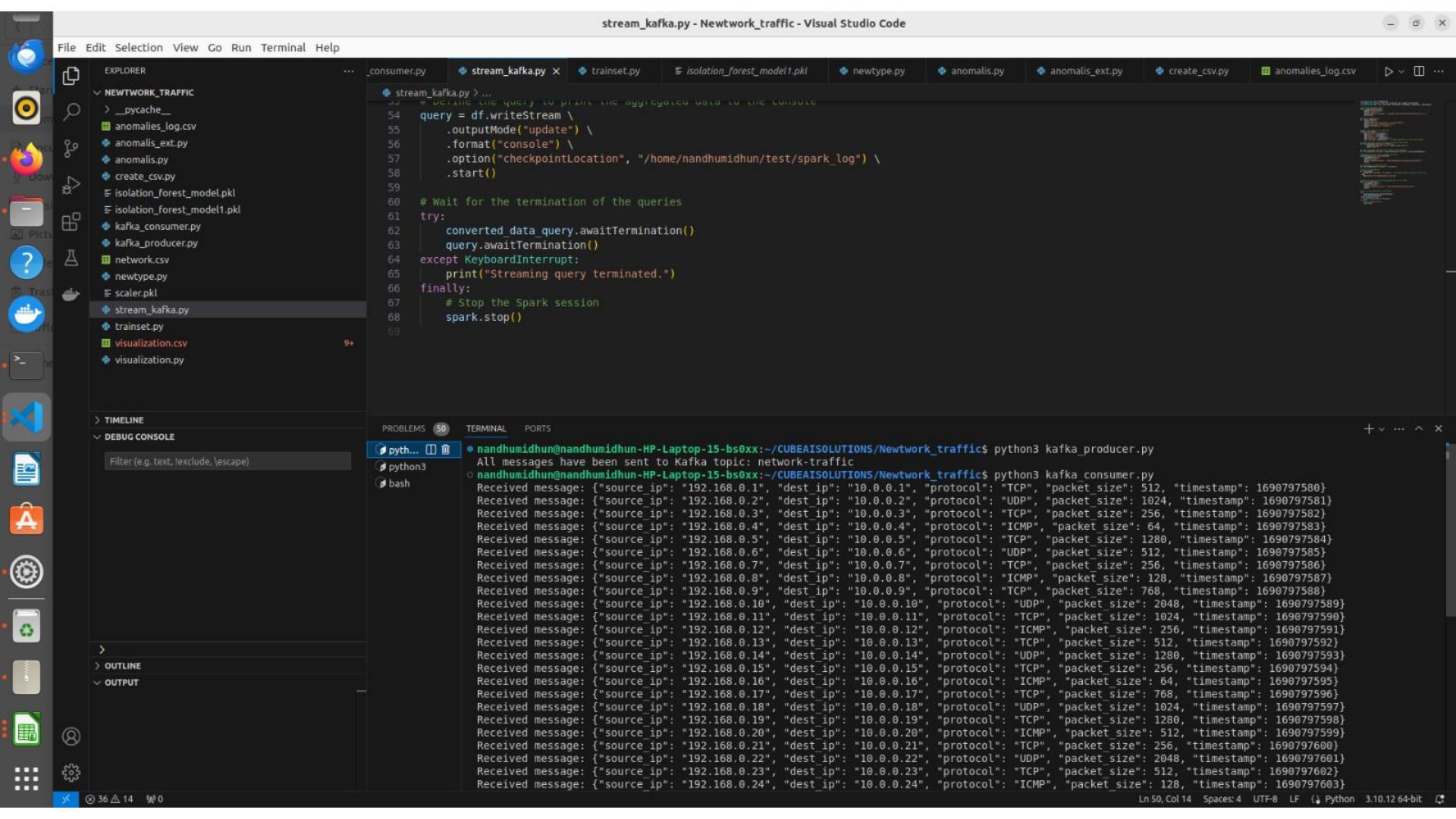

Console
Kinesis
lambdak
Kinesis
CloudW
Roles
Visual
Redshift
Editor
AWSSer
run-172
(4) What
learn-kin
Kinesis Data
CloudFo
Generat
https://us-east-1.console.aws.amazon.com/gluestudio/home?region=us-east-1#/editor/job/ETL_Sales_data job/graph
aws
Services
redshift
N. Virginia
Nandhu @ 6374-2337-3422
ETL_Sales_data job
Last modified on 1/8/2024, 1:06:39 pm
Actions
Save
Run
Visual
Script
Job details
Runs
Data quality
Schedules
Version Control
Data source - S3 bucket
Amazon S3
Transform - Filter
Filter
Data target - S3 bucket
Amazon S3
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


Redshift query editor v2
Editor
Queries
Notebooks
Charts
History
Scheduled queries
Create
Load data
Filter resources
Serverless: default-workg...
awsdatacatalog
dev
sample_data_dev
load-data-Kafka-ad2f
Untitled 1
load-data-gluetable-d441
Run
Limit 100
Explain
Isolated session
Serverless: de...
dev
Schedule
1 's:iam::637423373422:role/service-role/AmazonRedshift-CommandsAccessRole-20240729T163148' FORMAT AS CSV DELIMITER ',' QUOTE '"' IGNOREHEADER 1 REGION AS 'us-east-1';
2
Row 1, Col 265, Chr 300
Result 1
Result 2 (20)
Export
Chart
product_id | date | product_name | customer_id | amount | id
104 | 2024-01-04 | Widget D | 1004 | 60 | 4
104 | 2024-01-09 | Widget D | 1009 | 60 | 9
104 | 2024-01-14 | Widget D | 1014 | 60 | 14
104 | 2024-01-19 | Widget D | 1019 | 60 | 19
104 | 2024-01-24 | Widget D | 1024 | 60 | 24
104 | 2024-01-29 | Widget D | 1029 | 60 | 29
104 | 2024-02-03 | Widget D | 1034 | 60 | 34
104 | 2024-02-08 | Widget D | 1039 | 60 | 39
104 | 2024-02-13 | Widget D | 1044 | 60 | 44
104 | 2024-02-18 | Widget D | 1049 | 60 | 49
104 | 2024-01-04 | Widget D | 1004 | 60 | 4
104 | 2024-01-09 | Widget D | 1009 | 60 | 9
Query ID 3366
Elapsed time: 2847 ms
Total rows: 20
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

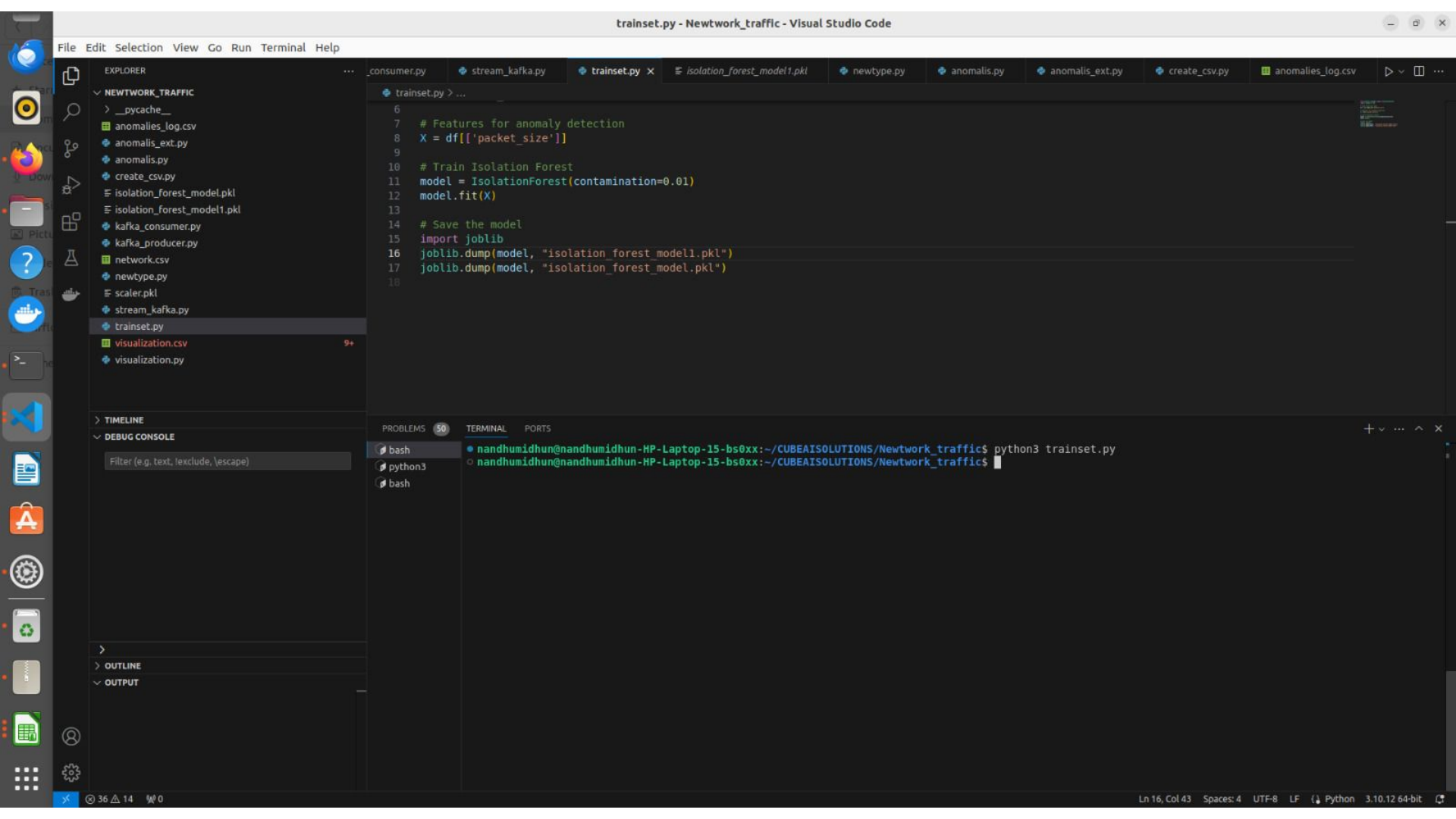

AWS Glue
Getting started
ETL jobs
Visual ETL
Notebooks
Job run monitoring
Data Catalog tables
Data connections
Workflows (orchestration)
Data Catalog
Databases
Tables
Stream schema registries
Schemas
Connections
Crawlers
Classifiers
Catalog settings
Data Integration and ETL
Legacy pages
What's New
Documentation
AWS Marketplace
Enable compact mode
Enable new navigation
ETL_Sales_data job
Last modified on 1/8/2024, 1:06:39 pm
Actions
Save
Run
Visual
Script
Job details
Runs
Data quality
Schedules
Version Control
Job runs (1/5) Info
Last updated (UTC)
August 1, 2024 at 09:23:35
View details
Stop job run
Table View
Card View
Filter job runs by property
Run status
Retries
Start time (Local)
End time (Local)
Duration
Capacity (DPUs)
Worker t
Running 0 08/01/2024 14:51:53 - 1 m 34 s 10 DPUs G.1X
Succeeded 0 08/01/2024 14:31:21 08/01/2024 14:34:12 2 m 35 s 10 DPUs G.1X
Succeeded 0 08/01/2024 13:31:21 08/01/2024 13:34:55 3 m 15 s 10 DPUs G.1X
Run details
Input arguments (10)
Continuous logs
Run insights
Metrics
Spark UI
Stop job run
Continuous logs Info
Driver logs
Driver and executor log streams
24/08/01 09:22:43 INFO SecurityManager: Changing view acls groups to:
24/08/01 09:22:43 INFO SecurityManager: Changing modify acls to: spark
24/08/01 09:22:43 INFO SecurityManager: Changing view acls to: spark
24/08/01 09:22:43 INFO ResourceProfileManager: Added ResourceProfile id: 0
24/08/01 09:22:43 INFO ResourceProfile: Limiting resource is cpus at 4 tasks per executor
24/08/01 09:22:43 INFO ResourceProfile: Default ResourceProfile created, executor resources: Map(cores -> name: cores, amount: 4, script: , venc
24/08/01 09:22:43 INFO SparkContext: Submitted application: nativespark-ETL_Sales_data job-jr_d0b39efc0428a94787d5dadc6cadde721767fb393af77fe2
24/08/01 09:22:43 INFO ResourceUtils: ==
24/08/01 09:22:43 INFO ResourceUtils: No custom resources configured for spark.driver.
24/08/01 09:22:43 INFO SparkContext: Running Spark version 3.3.0-amzn-1
24/08/01 09:22:35 INFO SafeLogging: Initializing logging subsystem
24/08/01 09:22:33 INFO PlatformInfo: Unable to read clusterId from /var/lib/info/job-flow.json, out of places to look
24/08/01 09:22:33 INFO PlatformInfo: Unable to read clusterId from /var/lib/instance-controller/extraInstanceData.json, trying EMR job-flow dat
24/08/01 09:22:33 INFO PlatformInfo: Unable to read clusterId from http://localhost:8321/configuration, trying extra instance data file: /var/li
24/08/01 09:22:32 INFO LogPusher: legacyLogging: true - logs will be written with spark application ID
24/08/01 09:22:32 INFO LogPusher: standardLogging: true - logs will be written with job run ID or session ID
24/08/01 09:22:32 INFO SparkUICleaner: SparkUILogFileCleanerThread started
This information isn't currently available.
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

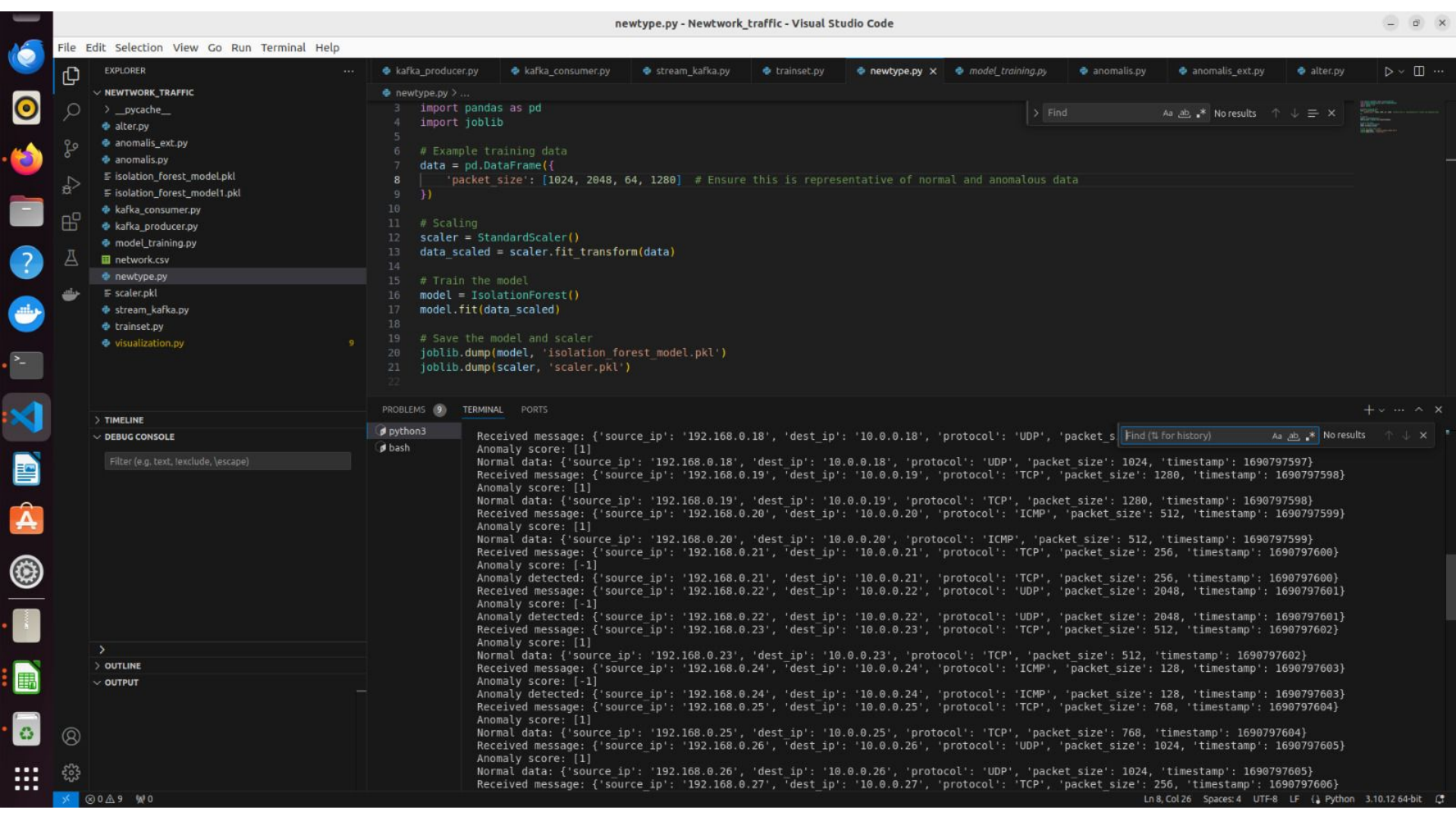
REAL-TIME ANOMALY DETECTION IN NETWORK TRAFFIC

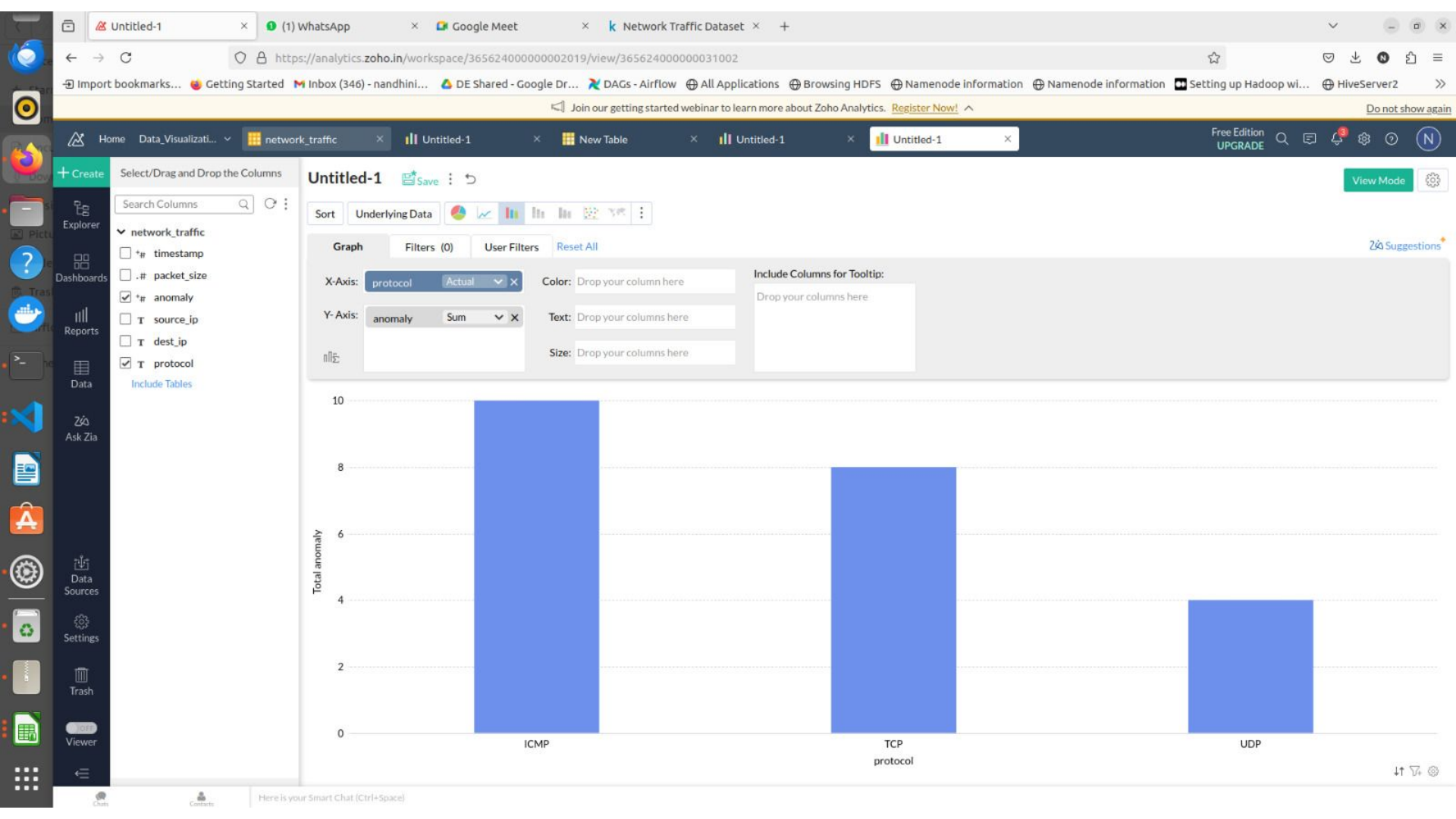
STEP 1: Data Ingestion Using kafka
STEP 2: Data Processing Spark Structured Streaming
STEP 3: Train an Anomaly Detection Model
STEP 4: Real-Time Anomaly Scoring
STEP 5: Detect the Anomaly
STEP 6: Visualize Data


stream_kafka.py - Newtwork_traffic - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
NEWTWORK_TRAFFIC
__pycache__
anomalies_log.csv
anomalis_ext.py
anomalis.py
create_csv.py
isolation_forest_model.pkl
isolation_forest_model1.pkl
kafka_consumer.py
kafka_producer.py
network.csv
newtype.py
scaler.pkl
stream_kafka.py
trainset.py
visualization.csv
visualization.py
TIMELINE
DEBUG CONSOLE
Filter (e.g. text, !exclude, \escape)
OUTLINE
OUTPUT
stream_kafka.py
trainset.py
isolation_forest_model1.pkl
newtype.py
anomalis.py
anomalis_ext.py
create_csv.py
anomalies_log.csv
stream_kafka.py > ...
query = df.writeStream \
.outputMode("update") \
.format("console") \
.option("checkpointLocation", "/home/nandhumidhun/test/spark_log") \
.start()
# Wait for the termination of the queries
try:
converted_data_query.awaitTermination()
query.awaitTermination()
except KeyboardInterrupt:
print("Streaming query terminated.")
finally:
# Stop the Spark session
spark.stop()
PROBLEMS 50
TERMINAL
PORTS
pyth...
python3
bash
nandhumidhun@nandhumidhun-HP-Laptop-15-bs0xx:~/CUBEAISOLUTIONS/Newtwork_traffic$ python3 kafka_producer.py
All messages have been sent to Kafka topic: network-traffic
nandhumidhun@nandhumidhun-HP-Laptop-15-bs0xx:~/CUBEAISOLUTIONS/Newtwork_traffic$ python3 kafka_consumer.py
Received message: {"source_ip": "192.168.0.1", "dest_ip": "10.0.0.1", "protocol": "TCP", "packet_size": 512, "timestamp": 1690797580}
Received message: {"source_ip": "192.168.0.2", "dest_ip": "10.0.0.2", "protocol": "UDP", "packet_size": 1024, "timestamp": 1690797581}
Received message: {"source_ip": "192.168.0.3", "dest_ip": "10.0.0.3", "protocol": "TCP", "packet_size": 256, "timestamp": 1690797582}
Received message: {"source_ip": "192.168.0.4", "dest_ip": "10.0.0.4", "protocol": "ICMP", "packet_size": 64, "timestamp": 1690797583}
Received message: {"source_ip": "192.168.0.5", "dest_ip": "10.0.0.5", "protocol": "TCP", "packet_size": 1280, "timestamp": 1690797584}
Received message: {"source_ip": "192.168.0.6", "dest_ip": "10.0.0.6", "protocol": "UDP", "packet_size": 512, "timestamp": 1690797585}
Received message: {"source_ip": "192.168.0.7", "dest_ip": "10.0.0.7", "protocol": "TCP", "packet_size": 256, "timestamp": 1690797586}
Received message: {"source_ip": "192.168.0.8", "dest_ip": "10.0.0.8", "protocol": "ICMP", "packet_size": 128, "timestamp": 1690797587}
Received message: {"source_ip": "192.168.0.9", "dest_ip": "10.0.0.9", "protocol": "TCP", "packet_size": 768, "timestamp": 1690797588}
Received message: {"source_ip": "192.168.0.10", "dest_ip": "10.0.0.10", "protocol": "UDP", "packet_size": 2048, "timestamp": 1690797589}
Received message: {"source_ip": "192.168.0.11", "dest_ip": "10.0.0.11", "protocol": "TCP", "packet_size": 1024, "timestamp": 1690797590}
Received message: {"source_ip": "192.168.0.12", "dest_ip": "10.0.0.12", "protocol": "ICMP", "packet_size": 256, "timestamp": 1690797591}
Received message: {"source_ip": "192.168.0.13", "dest_ip": "10.0.0.13", "protocol": "TCP", "packet_size": 512, "timestamp": 1690797592}
Received message: {"source_ip": "192.168.0.14", "dest_ip": "10.0.0.14", "protocol": "UDP", "packet_size": 1280, "timestamp": 1690797593}
Received message: {"source_ip": "192.168.0.15", "dest_ip": "10.0.0.15", "protocol": "TCP", "packet_size": 256, "timestamp": 1690797594}
Received message: {"source_ip": "192.168.0.16", "dest_ip": "10.0.0.16", "protocol": "ICMP", "packet_size": 64, "timestamp": 1690797595}
Received message: {"source_ip": "192.168.0.17", "dest_ip": "10.0.0.17", "protocol": "TCP", "packet_size": 768, "timestamp": 1690797596}
Received message: {"source_ip": "192.168.0.18", "dest_ip": "10.0.0.18", "protocol": "UDP", "packet_size": 1024, "timestamp": 1690797597}
Received message: {"source_ip": "192.168.0.19", "dest_ip": "10.0.0.19", "protocol": "TCP", "packet_size": 1280, "timestamp": 1690797598}
Received message: {"source_ip": "192.168.0.20", "dest_ip": "10.0.0.20", "protocol": "ICMP", "packet_size": 512, "timestamp": 1690797599}
Received message: {"source_ip": "192.168.0.21", "dest_ip": "10.0.0.21", "protocol": "TCP", "packet_size": 256, "timestamp": 1690797600}
Received message: {"source_ip": "192.168.0.22", "dest_ip": "10.0.0.22", "protocol": "UDP", "packet_size": 2048, "timestamp": 1690797601}
Received message: {"source_ip": "192.168.0.23", "dest_ip": "10.0.0.23", "protocol": "TCP", "packet_size": 512, "timestamp": 1690797602}
Received message: {"source_ip": "192.168.0.24", "dest_ip": "10.0.0.24", "protocol": "ICMP", "packet_size": 128, "timestamp": 1690797603}
Ln 50, Col 14
Spaces: 4
UTF-8
LF
Python
3.10.12 64-bit

stream_kafka.py - Newtwork_traffic - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
NEWTWORK_TRAFFIC
__pycache__
anomalies_log.csv
anomalis_ext.py
anomalis.py
create_csv.py
isolation_forest_model.pkl
isolation_forest_model1.pkl
kafka_consumer.py
kafka_producer.py
network.csv
newtype.py
scaler.pkl
stream_kafka.py
trainset.py
visualization.csv 9+
visualization.py
TIMELINE
DEBUG CONSOLE
Filter (e.g. text, !exclude, \escape)
OUTLINE
OUTPUT
_consumer.py
stream_kafka.py
trainset.py
isolation_forest_model1.pkl
newtype.py
anomalis.py
anomalis_ext.py
create_csv.py
anomalies_log.csv
stream_kafka.py > ...
54 query = df.writeStream \
55 .outputMode("update") \
56 .format("console") \
57 .option("checkpointLocation", "/home/nandhumidhun/test/spark_log") \
58 .start()
60 # Wait for the termination of the queries
61 try:
62 converted_data_query.awaitTermination()
63 query.awaitTermination()
64 except KeyboardInterrupt:
65 print("Streaming query terminated.")
66 finally:
67 # Stop the Spark session
68 spark.stop()
PROBLEMS 50
TERMINAL
PORTS
bash
python3
bash
192.168.0.8	10.0.0.8	ICMP	128	2023-07-31 15:29:47
192.168.0.9	10.0.0.9	TCP	768	2023-07-31 15:29:48
192.168.0.10	10.0.0.10	UDP	2048	2023-07-31 15:29:49
192.168.0.11	10.0.0.11	TCP	1024	2023-07-31 15:29:50
192.168.0.12	10.0.0.12	ICMP	256	2023-07-31 15:29:51
192.168.0.13	10.0.0.13	TCP	512	2023-07-31 15:29:52
192.168.0.14	10.0.0.14	UDP	1280	2023-07-31 15:29:53
192.168.0.15	10.0.0.15	TCP	256	2023-07-31 15:29:54
192.168.0.16	10.0.0.16	ICMP	64	2023-07-31 15:29:55
192.168.0.17	10.0.0.17	TCP	768	2023-07-31 15:29:56
192.168.0.18	10.0.0.18	UDP	1024	2023-07-31 15:29:57
192.168.0.19	10.0.0.19	TCP	1280	2023-07-31 15:29:58
192.168.0.20	10.0.0.20	ICMP	512	2023-07-31 15:29:59
192.168.0.21	10.0.0.21	TCP	256	2023-07-31 15:30:00
192.168.0.22	10.0.0.22	UDP	2048	2023-07-31 15:30:01
192.168.0.23	10.0.0.23	TCP	512	2023-07-31 15:30:02
only showing top 20 rows
Batch: 11
window	packet_size_avg
{2023-07-31 15:30...	806.4
{2023-07-31 15:25...	691.2
Ln 50, Col 14 Spaces: 4 UTF-8 LF Python 3.10.12 64-bit

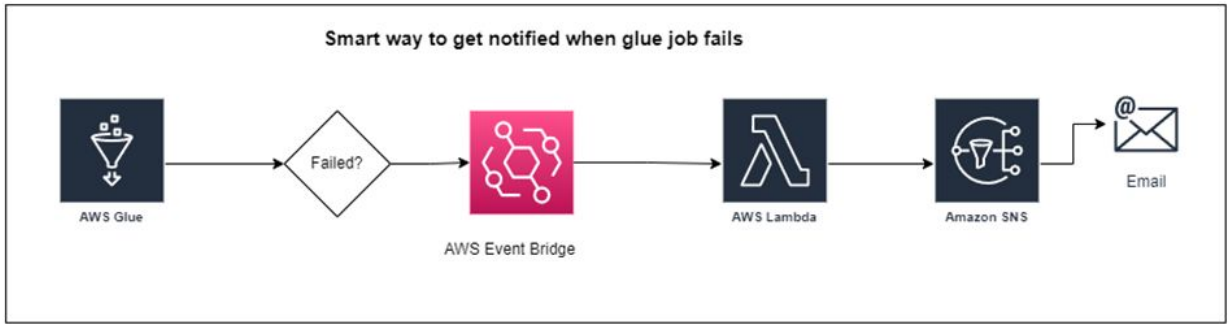

trainset.py - Newtwork_traffic - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
NEWTWORK_TRAFFIC
__pycache__
anomalies_log.csv
anomalis_ext.py
anomalis.py
create_csv.py
isolation_forest_model.pkl
isolation_forest_model1.pkl
kafka_consumer.py
kafka_producer.py
network.csv
newtype.py
scaler.pkl
stream_kafka.py
trainset.py
visualization.csv 9+
visualization.py
TIMELINE
DEBUG CONSOLE
Filter (e.g. text, !exclude, \escape)
OUTLINE
OUTPUT
_consumer.py
stream_kafka.py
trainset.py
isolation_forest_model1.pkl
newtype.py
anomalis.py
anomalis_ext.py
create_csv.py
anomalies_log.csv
trainset.py > ...
6
7 # Features for anomaly detection
8 X = df[['packet_size']]
9
10 # Train Isolation Forest
11 model = IsolationForest(contamination=0.01)
12 model.fit(X)
13
14 # Save the model
15 import joblib
16 joblib.dump(model, "isolation_forest_model1.pkl")
17 joblib.dump(model, "isolation_forest_model.pkl")
18
PROBLEMS 50
TERMINAL
PORTS
bash
python3
bash
nandhumidhun@nandhumidhun-HP-Laptop-15-bs0xx:~/CUBEAISOLUTIONS/Newtwork_traffic$ python3 trainset.py
nandhumidhun@nandhumidhun-HP-Laptop-15-bs0xx:~/CUBEAISOLUTIONS/Newtwork_traffic$
36 14 0
Ln 16, Col 43 Spaces: 4 UTF-8 LF Python 3.10.12 64-bit

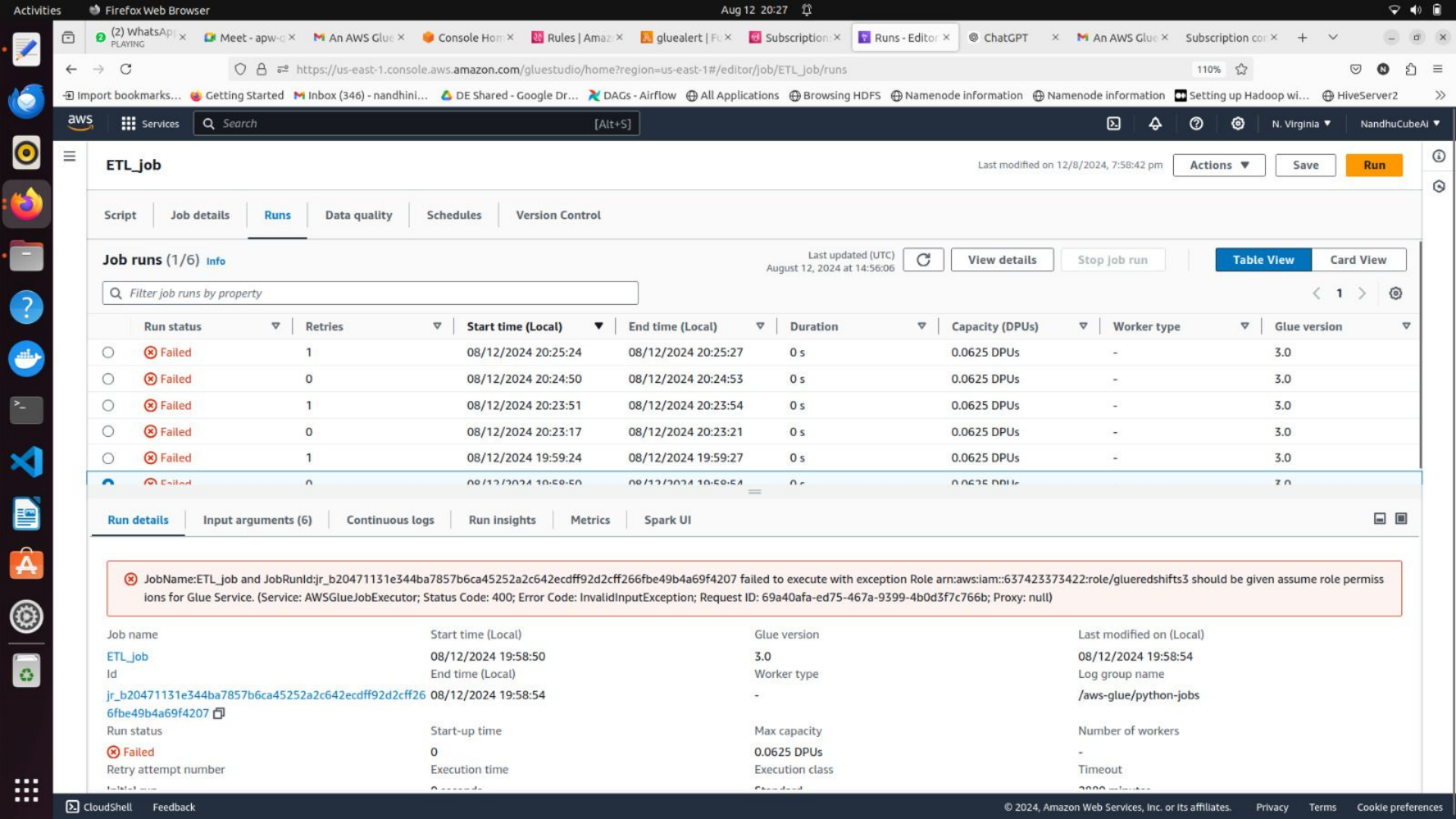

newtype.py - Newtwork_traffic - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
NEWTWORK_TRAFFIC
__pycache__
alter.py
anomalis_ext.py
anomalis.py
isolation_forest_model.pkl
isolation_forest_model1.pkl
kafka_consumer.py
kafka_producer.py
model_training.py
network.csv
newtype.py
scaler.pkl
stream_kafka.py
trainset.py
visualization.py
kafka_producer.py
kafka_consumer.py
stream_kafka.py
trainset.py
newtype.py
model_training.py
anomalis.py
anomalis_ext.py
alter.py
import pandas as pd
import joblib
# Example training data
data = pd.DataFrame({
'packet_size': [1024, 2048, 64, 1280] # Ensure this is representative of normal and anomalous data
})
# Scaling
scaler = StandardScaler()
data_scaled = scaler.fit_transform(data)
# Train the model
model = IsolationForest()
model.fit(data_scaled)
# Save the model and scaler
joblib.dump(model, 'isolation_forest_model.pkl')
joblib.dump(scaler, 'scaler.pkl')
PROBLEMS 9
TERMINAL
PORTS
python3
bash
Received message: {'source_ip': '192.168.0.18', 'dest_ip': '10.0.0.18', 'protocol': 'UDP', 'packet_size': 1024, 'timestamp': 1690797597}
Anomaly score: [1]
Normal data: {'source_ip': '192.168.0.18', 'dest_ip': '10.0.0.18', 'protocol': 'UDP', 'packet_size': 1024, 'timestamp': 1690797597}
Received message: {'source_ip': '192.168.0.19', 'dest_ip': '10.0.0.19', 'protocol': 'TCP', 'packet_size': 1280, 'timestamp': 1690797598}
Anomaly score: [1]
Normal data: {'source_ip': '192.168.0.19', 'dest_ip': '10.0.0.19', 'protocol': 'TCP', 'packet_size': 1280, 'timestamp': 1690797598}
Received message: {'source_ip': '192.168.0.20', 'dest_ip': '10.0.0.20', 'protocol': 'ICMP', 'packet_size': 512, 'timestamp': 1690797599}
Anomaly score: [1]
Normal data: {'source_ip': '192.168.0.20', 'dest_ip': '10.0.0.20', 'protocol': 'ICMP', 'packet_size': 512, 'timestamp': 1690797599}
Received message: {'source_ip': '192.168.0.21', 'dest_ip': '10.0.0.21', 'protocol': 'TCP', 'packet_size': 256, 'timestamp': 1690797600}
Anomaly score: [-1]
Anomaly detected: {'source_ip': '192.168.0.21', 'dest_ip': '10.0.0.21', 'protocol': 'TCP', 'packet_size': 256, 'timestamp': 1690797600}
Received message: {'source_ip': '192.168.0.22', 'dest_ip': '10.0.0.22', 'protocol': 'UDP', 'packet_size': 2048, 'timestamp': 1690797601}
Anomaly score: [-1]
Anomaly detected: {'source_ip': '192.168.0.22', 'dest_ip': '10.0.0.22', 'protocol': 'UDP', 'packet_size': 2048, 'timestamp': 1690797601}
Received message: {'source_ip': '192.168.0.23', 'dest_ip': '10.0.0.23', 'protocol': 'TCP', 'packet_size': 512, 'timestamp': 1690797602}
Anomaly score: [1]
Normal data: {'source_ip': '192.168.0.23', 'dest_ip': '10.0.0.23', 'protocol': 'TCP', 'packet_size': 512, 'timestamp': 1690797602}
Received message: {'source_ip': '192.168.0.24', 'dest_ip': '10.0.0.24', 'protocol': 'ICMP', 'packet_size': 128, 'timestamp': 1690797603}
Anomaly score: [-1]
Anomaly detected: {'source_ip': '192.168.0.24', 'dest_ip': '10.0.0.24', 'protocol': 'ICMP', 'packet_size': 128, 'timestamp': 1690797603}
Received message: {'source_ip': '192.168.0.25', 'dest_ip': '10.0.0.25', 'protocol': 'TCP', 'packet_size': 768, 'timestamp': 1690797604}
Anomaly score: [1]
Normal data: {'source_ip': '192.168.0.25', 'dest_ip': '10.0.0.25', 'protocol': 'TCP', 'packet_size': 768, 'timestamp': 1690797604}
Received message: {'source_ip': '192.168.0.26', 'dest_ip': '10.0.0.26', 'protocol': 'UDP', 'packet_size': 1024, 'timestamp': 1690797605}
Anomaly score: [1]
Normal data: {'source_ip': '192.168.0.26', 'dest_ip': '10.0.0.26', 'protocol': 'UDP', 'packet_size': 1024, 'timestamp': 1690797605}
Received message: {'source_ip': '192.168.0.27', 'dest_ip': '10.0.0.27', 'protocol': 'TCP', 'packet_size': 256, 'timestamp': 1690797606}
TIMELINE
DEBUG CONSOLE
OUTLINE
OUTPUT
Ln 8, Col 26
Spaces: 4
UTF-8
LF
Python
3.10.12 64-bit

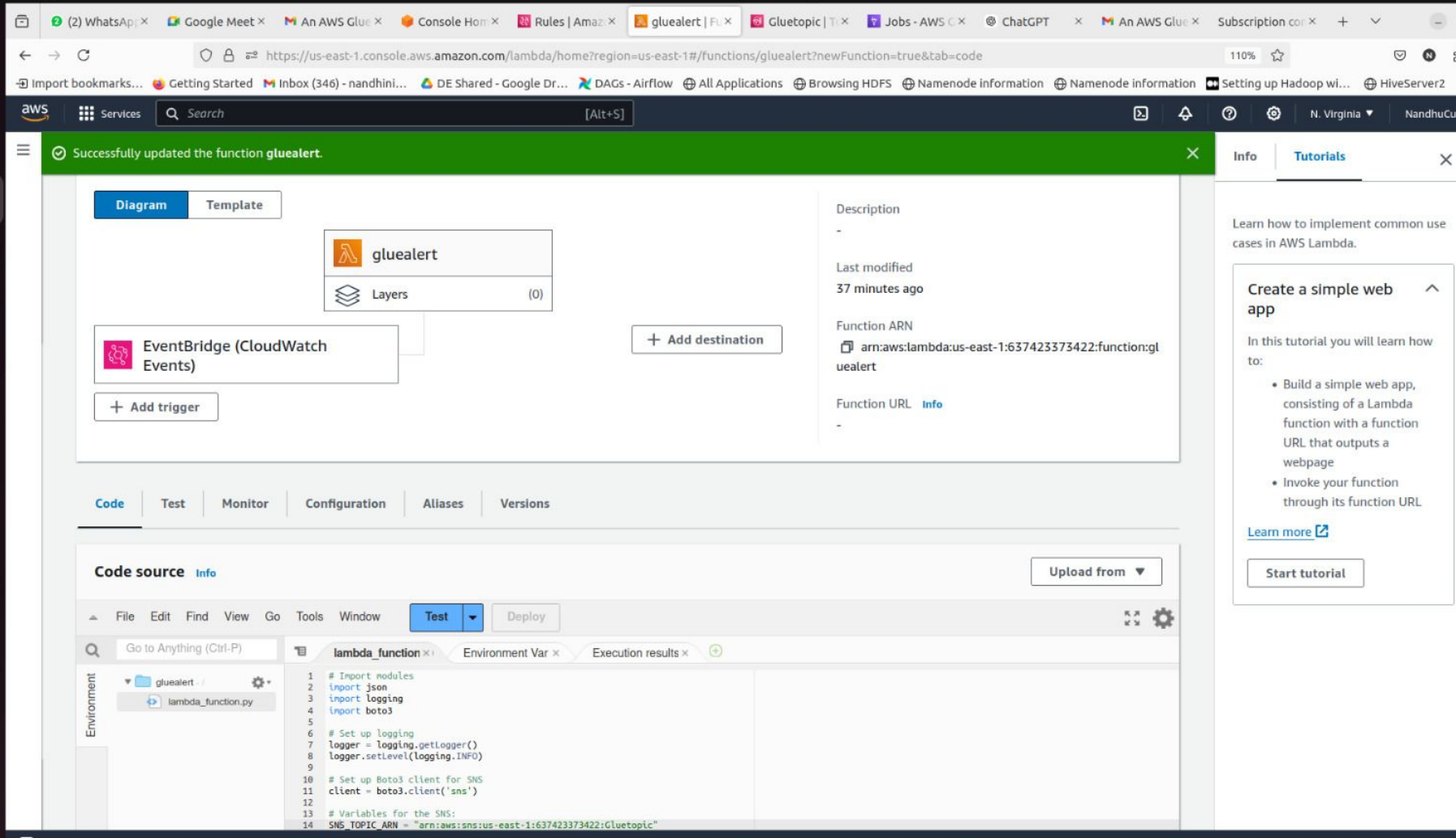

Untitled-1
(1) WhatsApp
Google Meet
Network Traffic Dataset
https://analytics.zoho.in/workspace/365624000000002019/view/365624000000031002
Import bookmarks...
Getting Started
Inbox (346) - nandhini...
DE Shared - Google Dr...
DAGs - Airflow
All Applications
Browsing HDFS
Namenode information
Namenode information
Setting up Hadoop wi...
HiveServer2
Join our getting started webinar to learn more about Zoho Analytics. Register Now!
Do not show again
Home
Data_Visualizati...
network_traffic
Untitled-1
New Table
Untitled-1
Untitled-1
Free Edition
UPGRADE
Create
Explorer
Dashboards
Reports
Data
Ask Zia
Data Sources
Settings
Trash
Viewer
Select/Drag and Drop the Columns
Search Columns
network_traffic
timestamp
packet_size
anomaly
source_ip
dest_ip
protocol
Include Tables
Untitled-1
Save
View Mode
Sort
Underlying Data
Graph
Filters (0)
User Filters
Reset All
Suggestions
X-Axis:
protocol
Actual
Y- Axis:
anomaly
Sum
Color:
Drop your column here
Text:
Drop your columns here
Size:
Drop your columns here
Include Columns for Tooltip:
Drop your columns here
10
8
6
4
2
0
Total anomaly
ICMP
TCP
UDP
protocol
Here is your Smart Chat (Ctrl+Space)

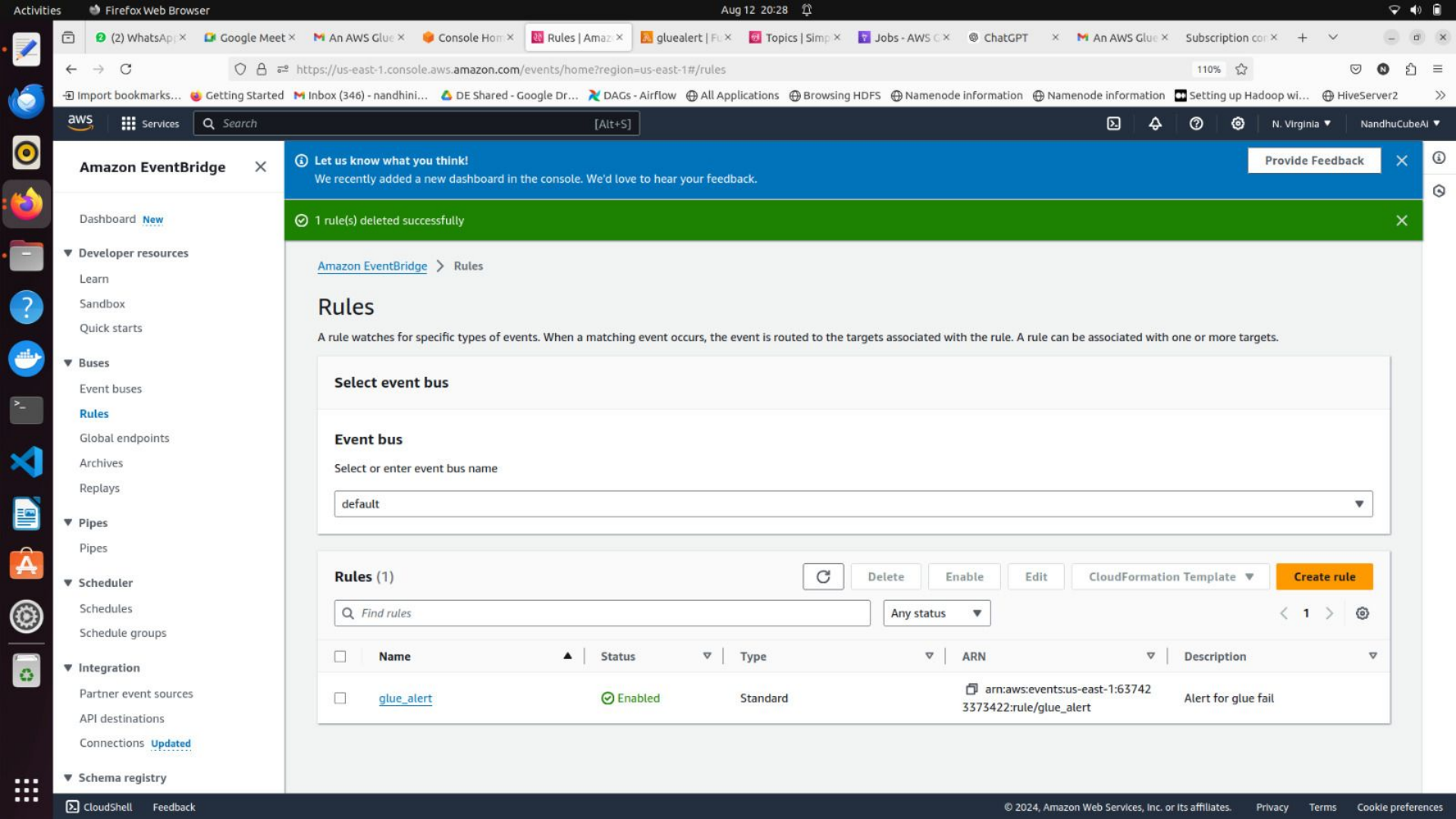
AWS GLUE JOB FAILURE NOTIFICATION WITH EVENTBRIDGE, LAMBDA, AND SNS

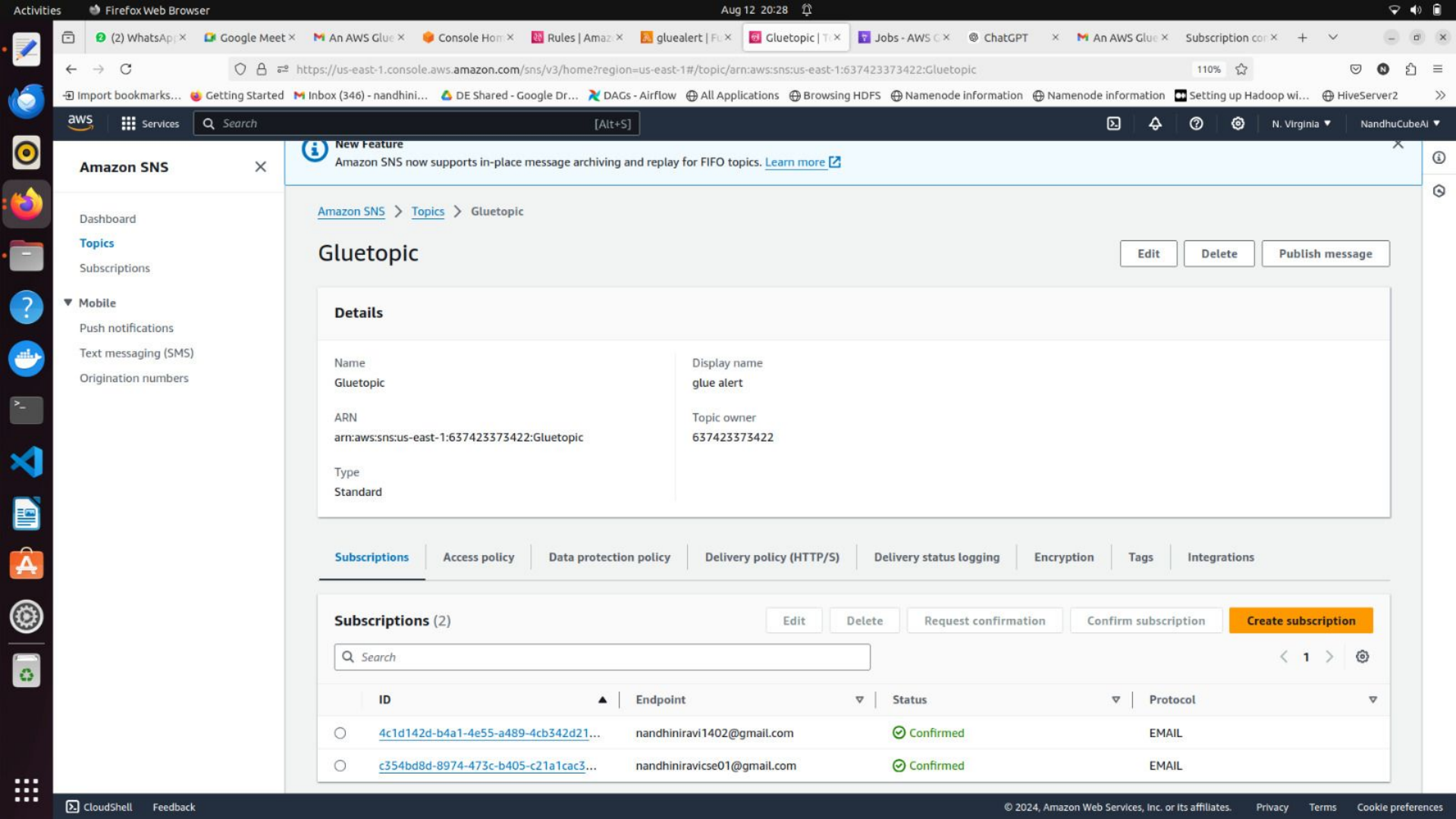
STEP 1: Open AWS Glue and Create the failed script
STEP 2: Create Event Bridge
STEP 3: Create lambda function and connect it to Event Bridge and SNS
STEP 4: Create SNS and create subscription

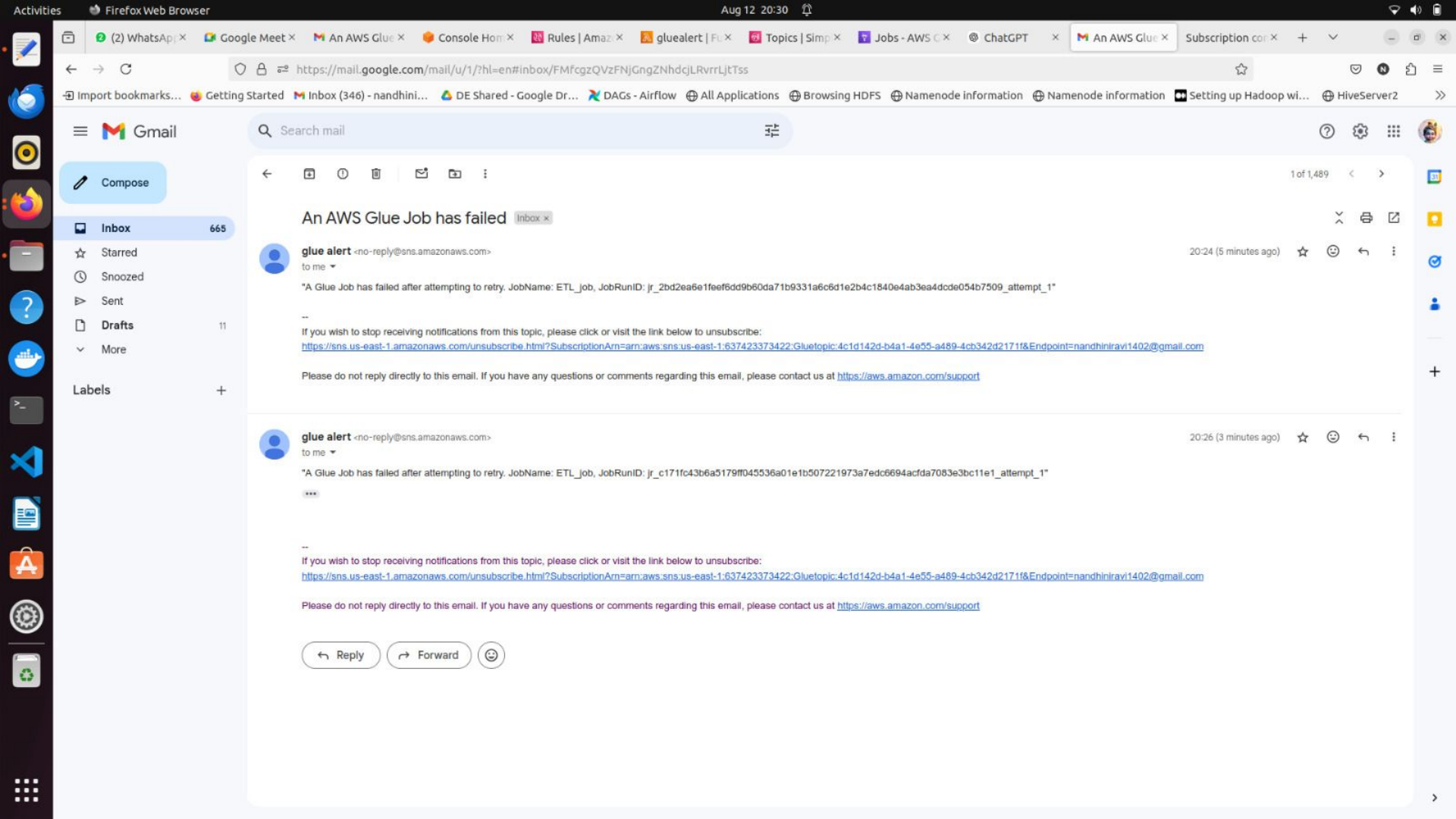



Activities
Firefox Web Browser
Aug 12 20:27
https://us-east-1.console.aws.amazon.com/gluestudio/home?region=us-east-1#/editor/job/ETL_job/runs
aws
Services
Search
[Alt+S]
N. Virginia
NandhuCubeAi
ETL_job
Last modified on 12/8/2024, 7:58:42 pm
Actions
Save
Run
Script
Job details
Runs
Data quality
Schedules
Version Control
Job runs (1/6) Info
Last updated (UTC)
August 12, 2024 at 14:56:06
View details
Stop job run
Table View
Card View
Filter job runs by property
Run status
Retries
Start time (Local)
End time (Local)
Duration
Capacity (DPUs)
Worker type
Glue version
Failed 1 08/12/2024 20:25:24 08/12/2024 20:25:27 0 s 0.0625 DPUs - 3.0
Failed 0 08/12/2024 20:24:50 08/12/2024 20:24:53 0 s 0.0625 DPUs - 3.0
Failed 1 08/12/2024 20:23:51 08/12/2024 20:23:54 0 s 0.0625 DPUs - 3.0
Failed 0 08/12/2024 20:23:17 08/12/2024 20:23:21 0 s 0.0625 DPUs - 3.0
Failed 1 08/12/2024 19:59:24 08/12/2024 19:59:27 0 s 0.0625 DPUs - 3.0
Run details
Input arguments (6)
Continuous logs
Run insights
Metrics
Spark UI
JobName:ETL_job and JobRunId:jr_b20471131e344ba7857b6ca45252a2c642ecdff92d2cff266fbe49b4a69f4207 failed to execute with exception Role arn:aws:iam::637423373422:role/glueredshifts3 should be given assume role permissions for Glue Service. (Service: AWSGlueJobExecutor; Status Code: 400; Error Code: InvalidInputException; Request ID: 69a40afa-ed75-467a-9399-4b0d3f7c766b; Proxy: null)
Job name
ETL_job
Start time (Local)
08/12/2024 19:58:50
Glue version
3.0
Last modified on (Local)
08/12/2024 19:58:54
Id
jr_b20471131e344ba7857b6ca45252a2c642ecdff92d2cff266fbe49b4a69f4207
End time (Local)
08/12/2024 19:58:54
Worker type
-
Log group name
/aws-glue/python-jobs
Run status
Failed
Start-up time
0
Max capacity
0.0625 DPUs
Number of workers
-
Retry attempt number
Execution time
Execution class
Timeout
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


Activities
Firefox Web Browser
Aug 12 20:28
https://us-east-1.console.aws.amazon.com/lambda/home?region=us-east-1#/functions/gluealert?newFunction=true&tab=code
Import bookmarks...
Getting Started
Inbox (346) - nandhini...
DE Shared - Google Dr...
DAGs - Airflow
All Applications
Browsing HDFS
Namenode information
Namenode information
Setting up Hadoop wi...
HiveServer2
aws
Services
Search
[Alt+S]
N. Virginia
NandhuCubeAi
Successfully updated the function gluealert.
Diagram
Template
gluealert
Layers
(0)
EventBridge (CloudWatch Events)
Add trigger
Add destination
Description
-
Last modified
37 minutes ago
Function ARN
arn:aws:lambda:us-east-1:637423373422:function:gluealert
Function URL
Info
-
Code
Test
Monitor
Configuration
Aliases
Versions
Code source
Info
Upload from
File
Edit
Find
View
Go
Tools
Window
Test
Deploy
Go to Anything (Ctrl-P)
lambda_function
Environment Var
Execution results
Environment
gluealert
lambda_function.py
# Import modules
import json
import logging
import boto3
# Set up logging
logger = logging.getLogger()
logger.setLevel(logging.INFO)
# Set up Boto3 client for SNS
client = boto3.client('sns')
# Variables for the SNS:
SNS_TOPIC_ARN = "arn:aws:sns:us-east-1:637423373422:Gluetopic"
Info
Tutorials
Learn how to implement common use cases in AWS Lambda.
Create a simple web app
In this tutorial you will learn how to:
Build a simple web app, consisting of a Lambda function with a function URL that outputs a webpage
Invoke your function through its function URL
Learn more
Start tutorial
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


Activities
Firefox Web Browser
Aug 12 20:28
https://us-east-1.console.aws.amazon.com/events/home?region=us-east-1#/rules
Import bookmarks...
Getting Started
Inbox (346) - nandhini...
DE Shared - Google Dr...
DAGs - Airflow
All Applications
Browsing HDFS
Namenode information
Namenode information
Setting up Hadoop wi...
HiveServer2
aws
Services
Search
[Alt+S]
N. Virginia
NandhuCubeAi
Amazon EventBridge
Let us know what you think!
We recently added a new dashboard in the console. We'd love to hear your feedback.
Provide Feedback
1 rule(s) deleted successfully
Dashboard New
Developer resources
Learn
Sandbox
Quick starts
Buses
Event buses
Rules
Global endpoints
Archives
Replays
Pipes
Pipes
Scheduler
Schedules
Schedule groups
Integration
Partner event sources
API destinations
Connections Updated
Schema registry
Amazon EventBridge > Rules
Rules
A rule watches for specific types of events. When a matching event occurs, the event is routed to the targets associated with the rule. A rule can be associated with one or more targets.
Select event bus
Event bus
Select or enter event bus name
default
Rules (1)
Delete
Enable
Edit
CloudFormation Template
Create rule
Find rules
Any status
1
Name
Status
Type
ARN
Description
glue_alert
Enabled
Standard
arn:aws:events:us-east-1:637423373422:rule/glue_alert
Alert for glue fail
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


Activities
Firefox Web Browser
Aug 12 20:28
https://us-east-1.console.aws.amazon.com/sns/v3/home?region=us-east-1#/topic/arn:aws:sns:us-east-1:637423373422:Gluetopic
aws
Services
Search
[Alt+S]
N. Virginia
NandhuCubeAi
Amazon SNS
Dashboard
Topics
Subscriptions
Mobile
Push notifications
Text messaging (SMS)
Origination numbers
New Feature
Amazon SNS now supports in-place message archiving and replay for FIFO topics. Learn more
Amazon SNS > Topics > Gluetopic
Gluetopic
Edit
Delete
Publish message
Details
Name
Gluetopic
Display name
glue alert
ARN
arn:aws:sns:us-east-1:637423373422:Gluetopic
Topic owner
637423373422
Type
Standard
Subscriptions
Access policy
Data protection policy
Delivery policy (HTTP/S)
Delivery status logging
Encryption
Tags
Integrations
Subscriptions (2)
Edit
Delete
Request confirmation
Confirm subscription
Create subscription
Search
1
ID
Endpoint
Status
Protocol
4c1d142d-b4a1-4e55-a489-4cb342d21...
nandhiniravi1402@gmail.com
Confirmed
EMAIL
c354bd8d-8974-473c-b405-c21a1cac3...
nandhiniravicse01@gmail.com
Confirmed
EMAIL
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


Activities
Firefox Web Browser
Aug 12 20:30
(2) WhatsAp
Google Meet
An AWS Glue
Console Hon
Rules | Amaz
gluealert | Fu
Topics | Simp
Jobs - AWS C
ChatGPT
An AWS Glue
Subscription cor
https://mail.google.com/mail/u/1/?hl=en#inbox/FMfcgzQVzFNjGngZNhdcjLRvrrLjtTss
Import bookmarks...
Getting Started
Inbox (346) - nandhini...
DE Shared - Google Dr...
DAGs - Airflow
All Applications
Browsing HDFS
Namenode information
Namenode information
Setting up Hadoop wi...
HiveServer2
Gmail
Search mail
Compose
Inbox 665
Starred
Snoozed
Sent
Drafts 11
More
Labels
1 of 1,489
An AWS Glue Job has failed
Inbox ×
glue alert <no-reply@sns.amazonaws.com>
to me
20:24 (5 minutes ago)
"A Glue Job has failed after attempting to retry. JobName: ETL_job, JobRunID: jr_2bd2ea6e1feef6dd9b60da71b9331a6c6d1e2b4c1840e4ab3ea4dcde054b7509_attempt_1"
--
If you wish to stop receiving notifications from this topic, please click or visit the link below to unsubscribe:
https://sns.us-east-1.amazonaws.com/unsubscribe.html?SubscriptionArn=arn:aws:sns:us-east-1:637423373422:Gluetopic:4c1d142d-b4a1-4e55-a489-4cb342d2171f&Endpoint=nandhiniravi1402@gmail.com
Please do not reply directly to this email. If you have any questions or comments regarding this email, please contact us at https://aws.amazon.com/support
glue alert <no-reply@sns.amazonaws.com>
to me
20:26 (3 minutes ago)
"A Glue Job has failed after attempting to retry. JobName: ETL_job, JobRunID: jr_c171fc43b6a5179ff045536a01e1b507221973a7edc6694acfda7083e3bc11e1_attempt_1"
--
If you wish to stop receiving notifications from this topic, please click or visit the link below to unsubscribe:
https://sns.us-east-1.amazonaws.com/unsubscribe.html?SubscriptionArn=arn:aws:sns:us-east-1:637423373422:Gluetopic:4c1d142d-b4a1-4e55-a489-4cb342d2171f&Endpoint=nandhiniravi1402@gmail.com
Please do not reply directly to this email. If you have any questions or comments regarding this email, please contact us at https://aws.amazon.com/support
Reply
Forward

THANK YOU

ANY QUERY