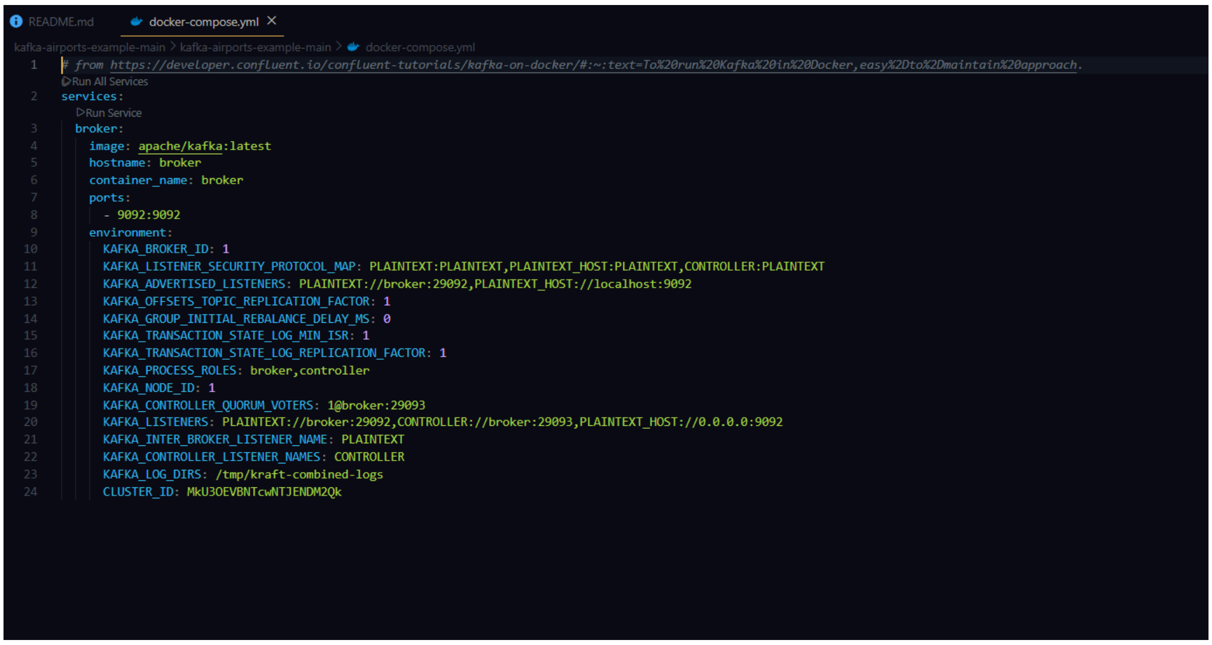
With docker:

links:

- https://www.baeldung.com/ops/kafka-docker-setup
- https://medium.com/@keshavmanglore/article-a-beginners-guide-to-kafka-with-python-real-time-data-processing-and-applications-5db39b320f3e

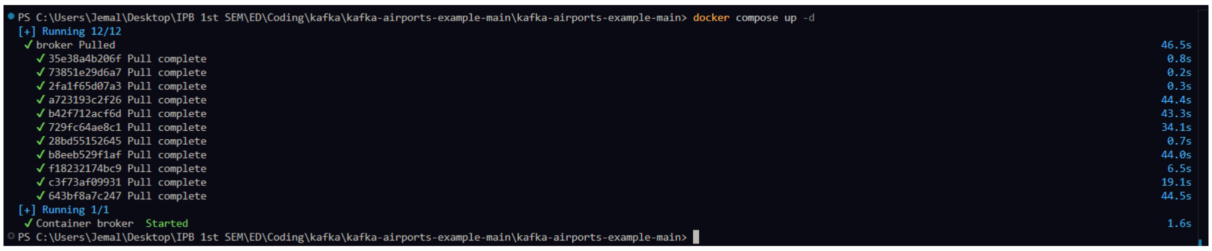
1. Create the docker-compose file and start the container



README.md
docker-compose.yml
kafka-airports-example-main > kafka-airports-example-main > docker-compose.yml
Run All Services
Run Service

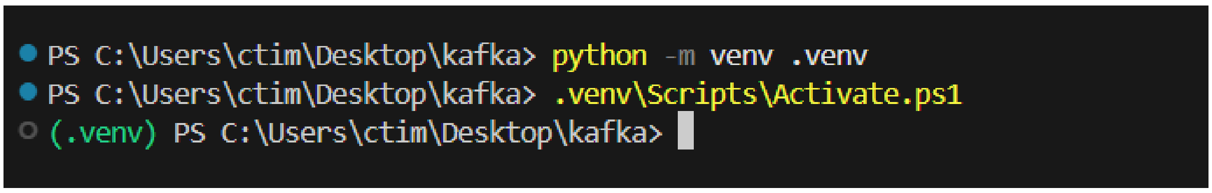
```yaml
# from https://developer.confluent.io/confluent-tutorials/kafka-on-docker/#:~:text=To%20run%20Kafka%20in%20Docker,easy%2Dto%2Dmaintain%20approach.
services:
  broker:
    image: apache/kafka:latest
    hostname: broker
    container_name: broker
    ports:
      - 9092:9092
    environment:
      KAFKA_BROKER_ID: 1
      KAFKA_LISTENER_SECURITY_PROTOCOL_MAP: PLAINTEXT:PLAINTEXT,PLAINTEXT_HOST:PLAINTEXT,CONTROLLER:PLAINTEXT
      KAFKA_ADVERTISED_LISTENERS: PLAINTEXT://broker:29092,PLAINTEXT_HOST://localhost:9092
      KAFKA_OFFSETS_TOPIC_REPLICATION_FACTOR: 1
      KAFKA_GROUP_INITIAL_REBALANCE_DELAY_MS: 0
      KAFKA_TRANSACTION_STATE_LOG_MIN_ISR: 1
      KAFKA_TRANSACTION_STATE_LOG_REPLICATION_FACTOR: 1
      KAFKA_PROCESS_ROLES: broker,controller
      KAFKA_NODE_ID: 1
      KAFKA_CONTROLLER_QUORUM_VOTERS: 1@broker:29093
      KAFKA_LISTENERS: PLAINTEXT://broker:29092,CONTROLLER://broker:29093,PLAINTEXT_HOST://0.0.0.0:9092
      KAFKA_INTER_BROKER_LISTENER_NAME: PLAINTEXT
      KAFKA_CONTROLLER_LISTENER_NAMES: CONTROLLER
      KAFKA_LOG_DIRS: /tmp/kraft-combined-logs
      CLUSTER_ID: MkU3OEVBNTcwNTJENDM2Qk
```

docker-compose up -d

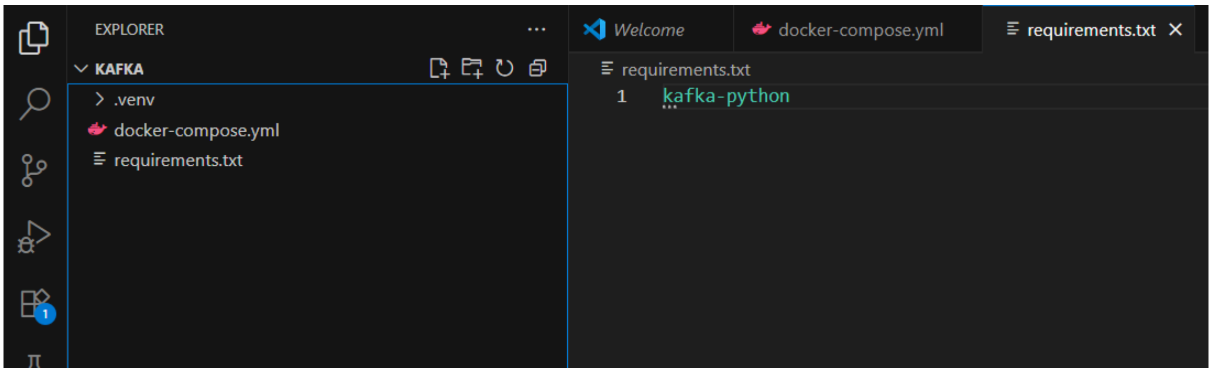


```
PS C:\Users\Jemal\Desktop\IPB 1st SEM\ED\Coding\kafka\kafka-airports-example-main\kafka-airports-example-main> docker compose up -d
 [+] Running 12/12
  ✓ broker Pulled                                                                                                              46.5s
    ✓ 35e38a4b206f Pull complete                                                                                                0.8s
    ✓ 73851e29d6a7 Pull complete                                                                                                0.2s
    ✓ 2fa1f65d07a3 Pull complete                                                                                                0.3s
    ✓ a723193c2f26 Pull complete                                                                                               44.4s
    ✓ b42f712acf6d Pull complete                                                                                               43.3s
    ✓ 729fc64ae8c1 Pull complete                                                                                               34.1s
    ✓ 28bd55152645 Pull complete                                                                                                0.7s
    ✓ b8eeb529f1af Pull complete                                                                                               44.0s
    ✓ f18232174bc9 Pull complete                                                                                                6.5s
    ✓ c3f73af09931 Pull complete                                                                                               19.1s
    ✓ 643bf8a7c247 Pull complete                                                                                               44.5s
 [+] Running 1/1
  ✓ Container broker  Started                                                                                                   1.6s
PS C:\Users\Jemal\Desktop\IPB 1st SEM\ED\Coding\kafka\kafka-airports-example-main\kafka-airports-example-main>
```

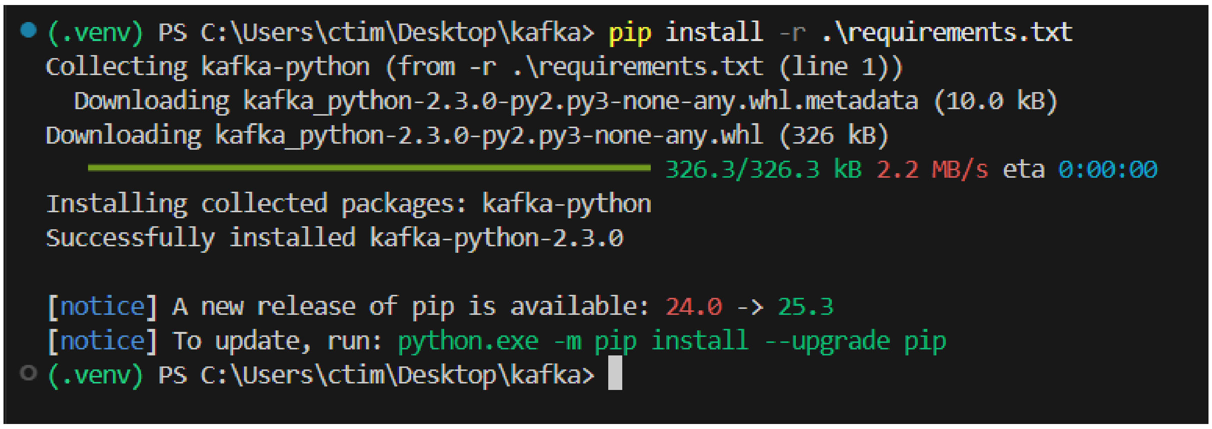
2. Create, activate, and install the requirements in a virtual environment



```
PS C:\Users\ctim\Desktop\kafka> python -m venv .venv
PS C:\Users\ctim\Desktop\kafka> .venv\Scripts\Activate.ps1
(.venv) PS C:\Users\ctim\Desktop\kafka>
```

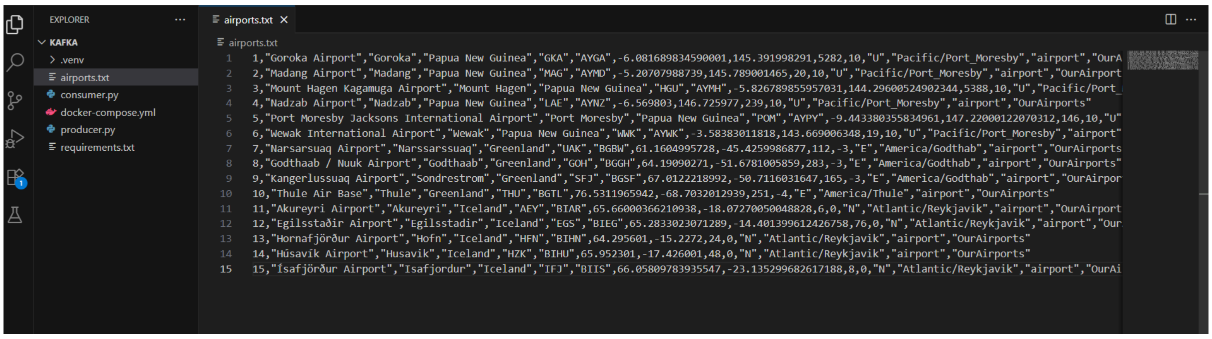


```
kafka-python
```



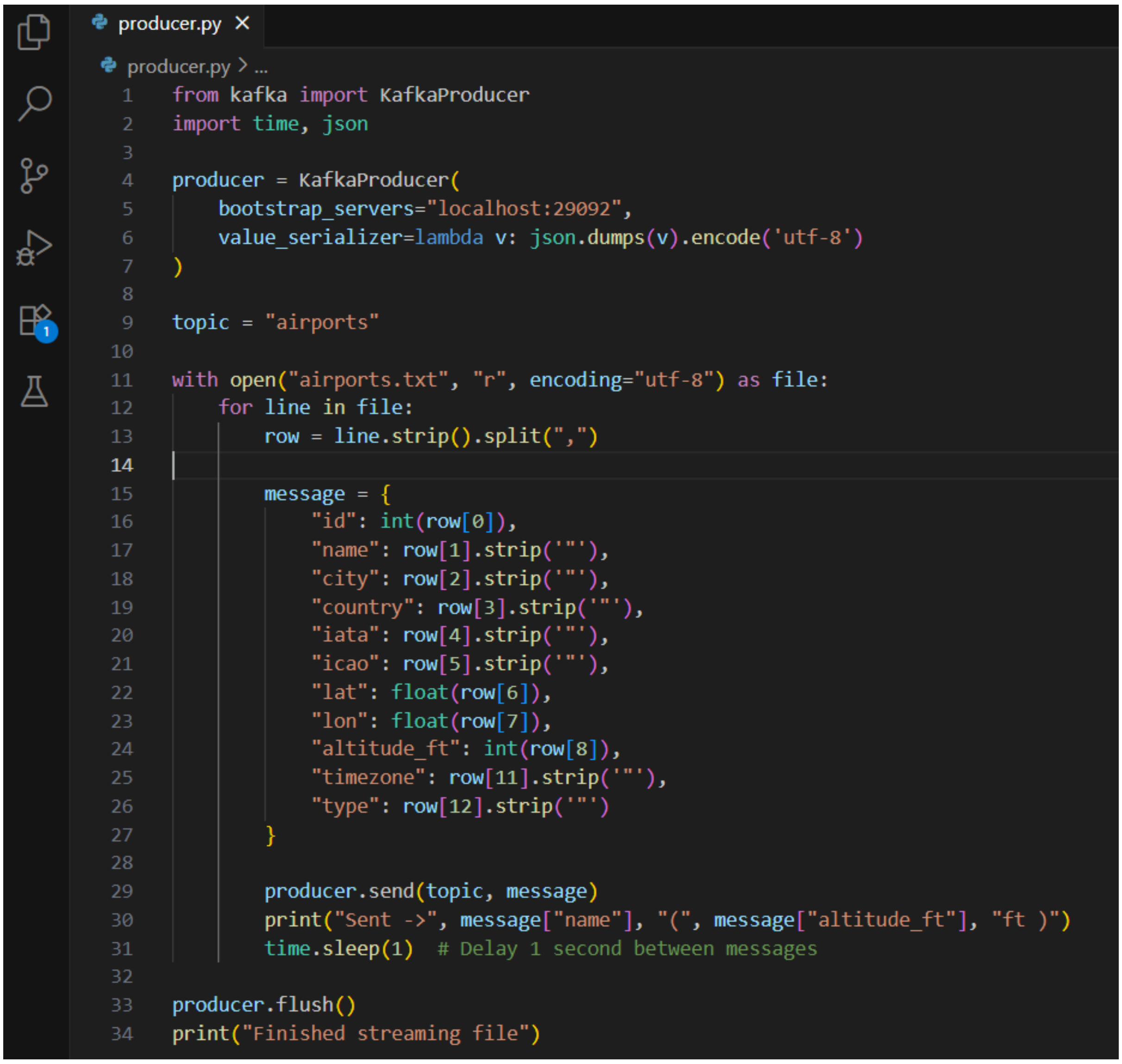
```
(.venv) PS C:\Users\ctim\Desktop\kafka> pip install -r .\requirements.txt
Collecting kafka-python (from -r .\requirements.txt (line 1))
  Downloading kafka_python-2.3.0-py2.py3-none-any.whl.metadata (10.0 kB)
Downloading kafka_python-2.3.0-py2.py3-none-any.whl (326 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 326.3/326.3 kB 2.2 MB/s eta 0:00:00
Installing collected packages: kafka-python
Successfully installed kafka-python-2.3.0

[notice] A new release of pip is available: 24.0 -> 25.3
[notice] To update, run: python.exe -m pip install --upgrade pip
(.venv) PS C:\Users\ctim\Desktop\kafka> 
```

3. Add or create a file with data to the directory:

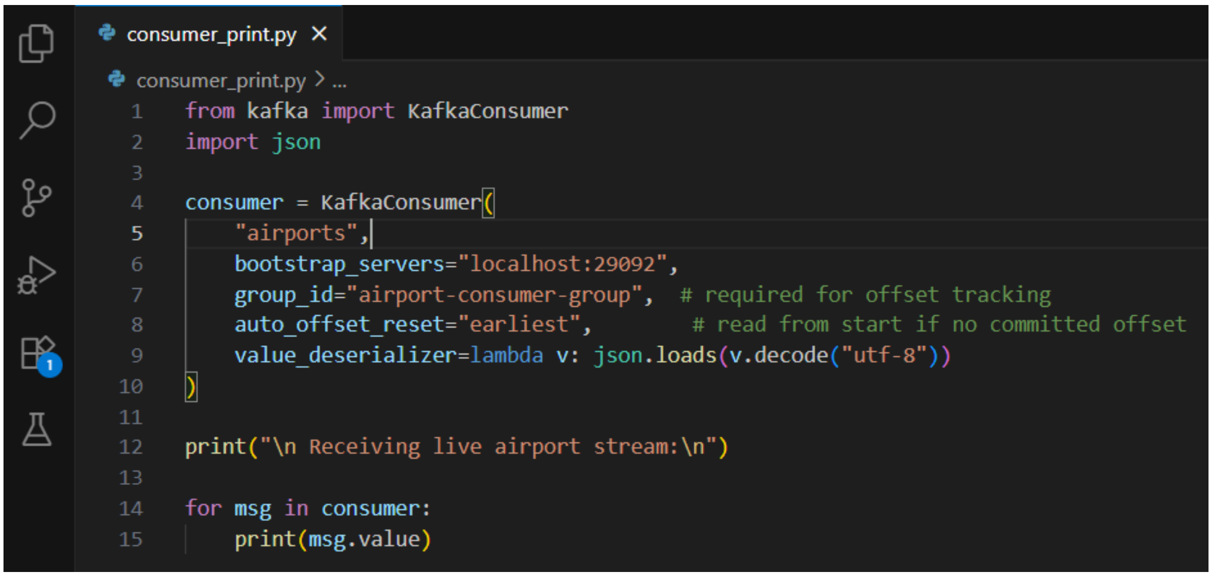


```
1,"Goroka Airport","Goroka","Papua New Guinea","GKA","AYGA",-6.081689834590001,145.391998291,5282,10,"U","Pacific/Port_Moresby","airport","OurA
2,"Madang Airport","Madang","Papua New Guinea","MAG","AYMD",-5.20707988739,145.789001465,20,10,"U","Pacific/Port_Moresby","airport","OurAirport
3,"Mount Hagen Kagamuga Airport","Mount Hagen","Papua New Guinea","HGU","AYMH",-5.826789855957031,144.29600524902344,5388,10,"U","Pacific/Port_
4,"Nadzab Airport","Nadzab","Papua New Guinea","LAE","AYNZ",-6.569803,146.725977,239,10,"U","Pacific/Port_Moresby","airport","OurAirports"
5,"Port Moresby Jacksons International Airport","Port Moresby","Papua New Guinea","POM","AYPY",-9.443380355834961,147.22000122070312,146,10,"U"
6,"Wewak International Airport","Wewak","Papua New Guinea","WWK","AYWK",-3.58383011818,143.669006348,19,10,"U","Pacific/Port_Moresby","airport"
7,"Narsarsuaq Airport","Narssarssuaq","Greenland","UAK","BGBW",61.1604995728,-45.4259986877,112,-3,"E","America/Godthab","airport","OurAirports
8,"Godthaab / Nuuk Airport","Godthaab","Greenland","GOH","BGGH",64.19090271,-51.6781005859,283,-3,"E","America/Godthab","airport","OurAirports"
9,"Kangerlussuaq Airport","Sondrestrom","Greenland","SFJ","BGSF",67.0122218992,-50.7116031647,165,-3,"E","America/Godthab","airport","OurAirpor
10,"Thule Air Base","Thule","Greenland","THU","BGTL",76.5311965942,-68.7032012939,251,-4,"E","America/Thule","airport","OurAirports"
11,"Akureyri Airport","Akureyri","Iceland","AEY","BIAR",65.66000366210938,-18.07270050048828,6,0,"N","Atlantic/Reykjavik","airport","OurAirport
12,"Egilsstaðir Airport","Egilsstadir","Iceland","EGS","BIEG",65.2833023071289,-14.401399612426758,76,0,"N","Atlantic/Reykjavik","airport","Our
13,"Hornafjörður Airport","Hofn","Iceland","HFN","BIHN",64.295601,-15.2272,24,0,"N","Atlantic/Reykjavik","airport","OurAirports"
14,"Húsavík Airport","Husavik","Iceland","HZK","BIHU",65.952301,-17.426001,48,0,"N","Atlantic/Reykjavik","airport","OurAirports"
15,"Ísafjörður Airport","Isafjordur","Iceland","IFJ","BIIS",66.05809783935547,-23.135299682617188,8,0,"N","Atlantic/Reykjavik","airport","OurAi
```

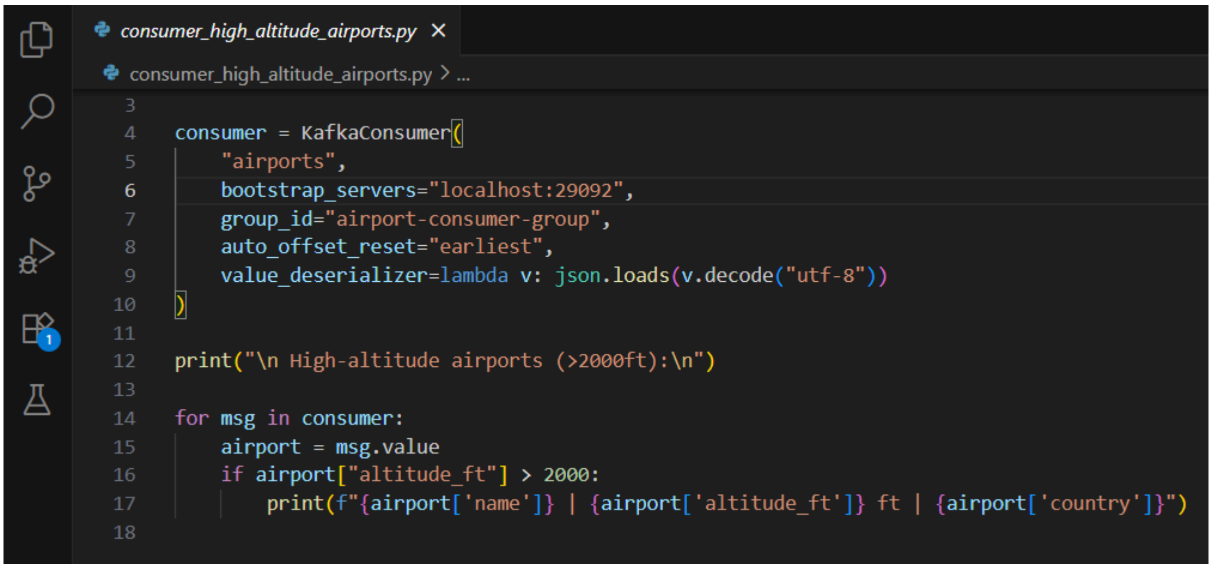
4. Create a producer.py to send the messages and a consumer.py to receive the messages



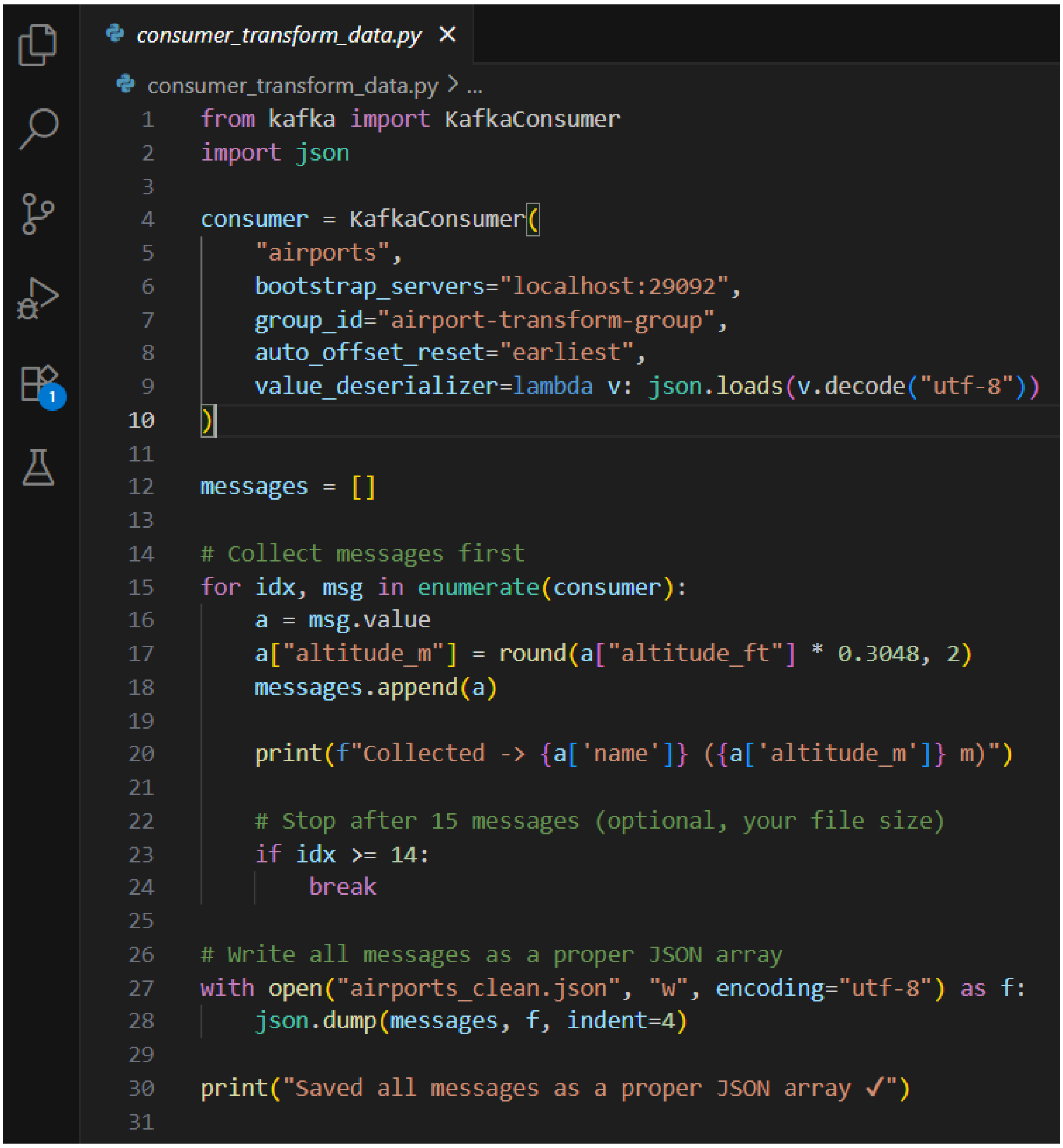
```python
from kafka import KafkaProducer
import time, json

producer = KafkaProducer(
    bootstrap_servers="localhost:29092",
    value_serializer=lambda v: json.dumps(v).encode('utf-8')
)

topic = "airports"

with open("airports.txt", "r", encoding="utf-8") as file:
    for line in file:
        row = line.strip().split(",")

        message = {
            "id": int(row[0]),
            "name": row[1].strip('"'),
            "city": row[2].strip('"'),
            "country": row[3].strip('"'),
            "iata": row[4].strip('"'),
            "icao": row[5].strip('"'),
            "lat": float(row[6]),
            "lon": float(row[7]),
            "altitude_ft": int(row[8]),
            "timezone": row[11].strip('"'),
            "type": row[12].strip('"')
        }

        producer.send(topic, message)
        print("Sent ->", message["name"], "(", message["altitude_ft"], "ft )")
        time.sleep(1)  # Delay 1 second between messages

producer.flush()
print("Finished streaming file")
```



```python
from kafka import KafkaConsumer
import json

consumer = KafkaConsumer(
    "airports",
    bootstrap_servers="localhost:29092",
    group_id="airport-consumer-group",  # required for offset tracking
    auto_offset_reset="earliest",        # read from start if no committed offset
    value_deserializer=lambda v: json.loads(v.decode("utf-8"))
)

print("\n Receiving live airport stream:\n")

for msg in consumer:
    print(msg.value)
```

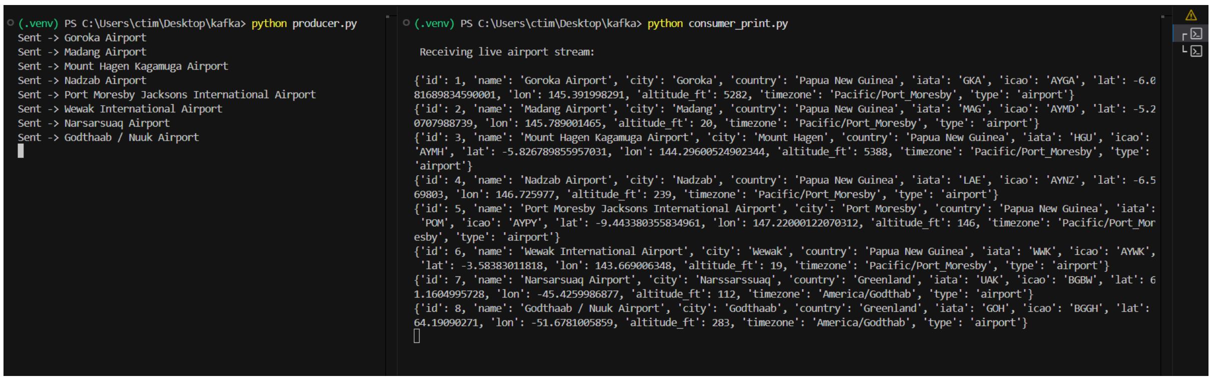

consumer_high_altitude_airports.py
consumer_high_altitude_airports.py > ...

```python
consumer = KafkaConsumer(
    "airports",
    bootstrap_servers="localhost:29092",
    group_id="airport-consumer-group",
    auto_offset_reset="earliest",
    value_deserializer=lambda v: json.loads(v.decode("utf-8"))
)

print("\n High-altitude airports (>2000ft):\n")

for msg in consumer:
    airport = msg.value
    if airport["altitude_ft"] > 2000:
        print(f"{airport['name']} | {airport['altitude_ft']} ft | {airport['country']}")

```



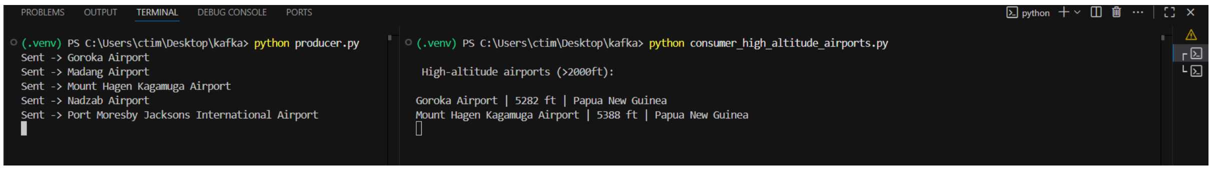
```python
from kafka import KafkaConsumer
import json

consumer = KafkaConsumer(
    "airports",
    bootstrap_servers="localhost:29092",
    group_id="airport-transform-group",
    auto_offset_reset="earliest",
    value_deserializer=lambda v: json.loads(v.decode("utf-8"))
)

messages = []

# Collect messages first
for idx, msg in enumerate(consumer):
    a = msg.value
    a["altitude_m"] = round(a["altitude_ft"] * 0.3048, 2)
    messages.append(a)

    print(f"Collected -> {a['name']} ({a['altitude_m']} m)")

    # Stop after 15 messages (optional, your file size)
    if idx >= 14:
        break

# Write all messages as a proper JSON array
with open("airports_clean.json", "w", encoding="utf-8") as f:
    json.dump(messages, f, indent=4)

print("Saved all messages as a proper JSON array ✓")

```

4. Initialize producer.py and consumer.py



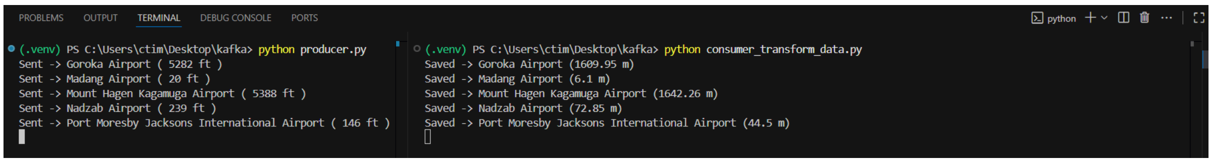
```
(.venv) PS C:\Users\ctim\Desktop\kafka> python producer.py
Sent -> Goroka Airport
Sent -> Madang Airport
Sent -> Mount Hagen Kagamuga Airport
Sent -> Nadzab Airport
Sent -> Port Moresby Jacksons International Airport
Sent -> Wewak International Airport
Sent -> Narsarsuaq Airport
Sent -> Godthaab / Nuuk Airport
```

```
(.venv) PS C:\Users\ctim\Desktop\kafka> python consumer_print.py

 Receiving live airport stream:

{'id': 1, 'name': 'Goroka Airport', 'city': 'Goroka', 'country': 'Papua New Guinea', 'iata': 'GKA', 'icao': 'AYGA', 'lat': -6.0
81689834590001, 'lon': 145.391998291, 'altitude_ft': 5282, 'timezone': 'Pacific/Port_Moresby', 'type': 'airport'}
{'id': 2, 'name': 'Madang Airport', 'city': 'Madang', 'country': 'Papua New Guinea', 'iata': 'MAG', 'icao': 'AYMD', 'lat': -5.2
0707988739, 'lon': 145.789001465, 'altitude_ft': 20, 'timezone': 'Pacific/Port_Moresby', 'type': 'airport'}
{'id': 3, 'name': 'Mount Hagen Kagamuga Airport', 'city': 'Mount Hagen', 'country': 'Papua New Guinea', 'iata': 'HGU', 'icao':
'AYMH', 'lat': -5.826789855957031, 'lon': 144.29600524902344, 'altitude_ft': 5388, 'timezone': 'Pacific/Port_Moresby', 'type':
'airport'}
{'id': 4, 'name': 'Nadzab Airport', 'city': 'Nadzab', 'country': 'Papua New Guinea', 'iata': 'LAE', 'icao': 'AYNZ', 'lat': -6.5
69803, 'lon': 146.725977, 'altitude_ft': 239, 'timezone': 'Pacific/Port_Moresby', 'type': 'airport'}
{'id': 5, 'name': 'Port Moresby Jacksons International Airport', 'city': 'Port Moresby', 'country': 'Papua New Guinea', 'iata':
 'POM', 'icao': 'AYPY', 'lat': -9.443380355834961, 'lon': 147.22000122070312, 'altitude_ft': 146, 'timezone': 'Pacific/Port_Mor
esby', 'type': 'airport'}
{'id': 6, 'name': 'Wewak International Airport', 'city': 'Wewak', 'country': 'Papua New Guinea', 'iata': 'WWK', 'icao': 'AYWK',
 'lat': -3.58383011818, 'lon': 143.669006348, 'altitude_ft': 19, 'timezone': 'Pacific/Port_Moresby', 'type': 'airport'}
{'id': 7, 'name': 'Narsarsuaq Airport', 'city': 'Narssarssuaq', 'country': 'Greenland', 'iata': 'UAK', 'icao': 'BGBW', 'lat': 6
1.1604995728, 'lon': -45.4259986877, 'altitude_ft': 112, 'timezone': 'America/Godthab', 'type': 'airport'}
{'id': 8, 'name': 'Godthaab / Nuuk Airport', 'city': 'Godthaab', 'country': 'Greenland', 'iata': 'GOH', 'icao': 'BGGH', 'lat':
64.19090271, 'lon': -51.6781005859, 'altitude_ft': 283, 'timezone': 'America/Godthab', 'type': 'airport'}
```



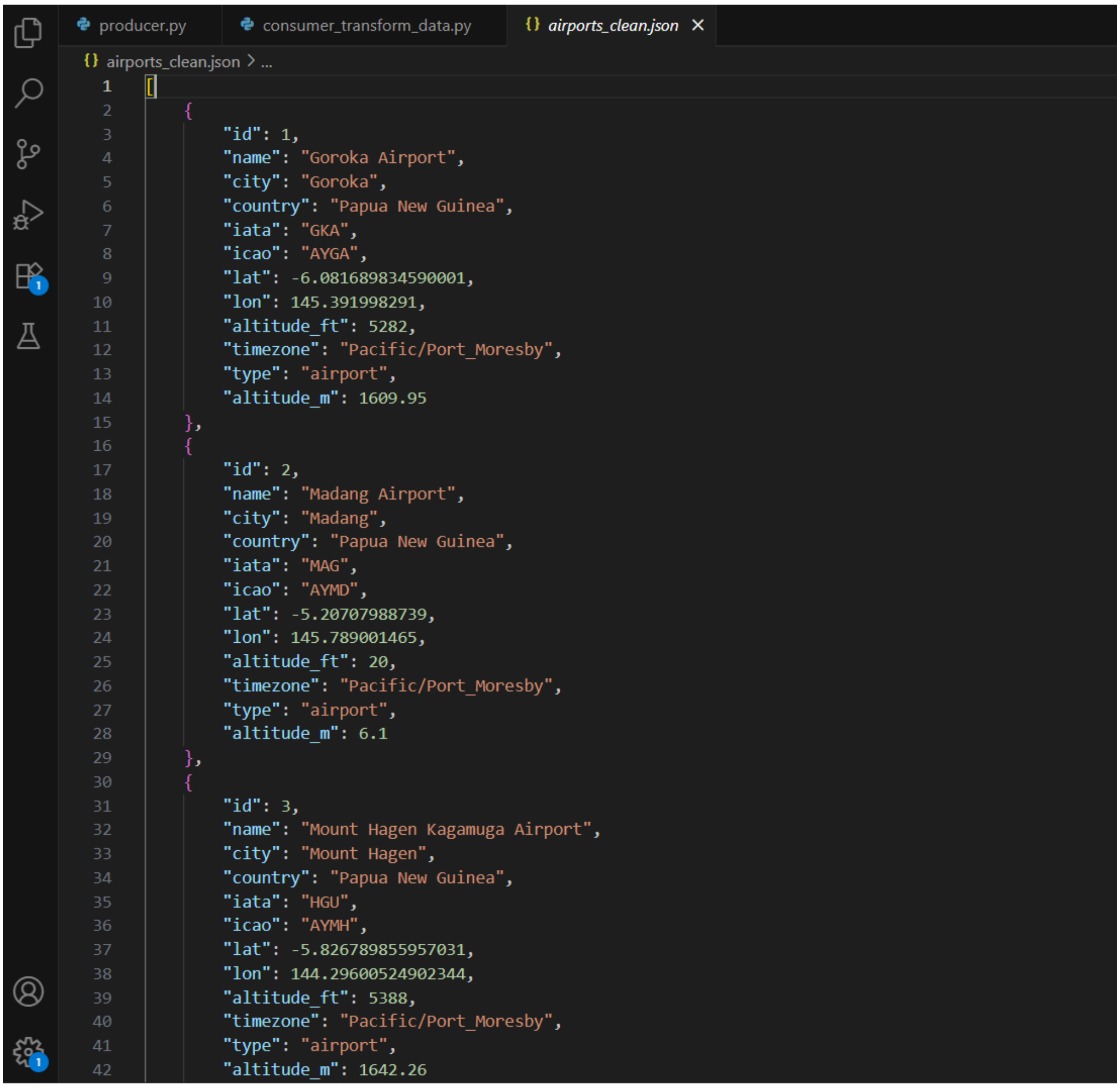
```
(.venv) PS C:\Users\ctim\Desktop\kafka> python producer.py
Sent -> Goroka Airport
Sent -> Madang Airport
Sent -> Mount Hagen Kagamuga Airport
Sent -> Nadzab Airport
Sent -> Port Moresby Jacksons International Airport
```

```
(.venv) PS C:\Users\ctim\Desktop\kafka> python consumer_high_altitude_airports.py

 High-altitude airports (>2000ft):

Goroka Airport | 5282 ft | Papua New Guinea
Mount Hagen Kagamuga Airport | 5388 ft | Papua New Guinea
```



```
(.venv) PS C:\Users\ctim\Desktop\kafka> python producer.py
Sent -> Goroka Airport ( 5282 ft )
Sent -> Madang Airport ( 20 ft )
Sent -> Mount Hagen Kagamuga Airport ( 5388 ft )
Sent -> Nadzab Airport ( 239 ft )
Sent -> Port Moresby Jacksons International Airport ( 146 ft )
```

```
(.venv) PS C:\Users\ctim\Desktop\kafka> python consumer_transform_data.py
Saved -> Goroka Airport (1609.95 m)
Saved -> Madang Airport (6.1 m)
Saved -> Mount Hagen Kagamuga Airport (1642.26 m)
Saved -> Nadzab Airport (72.85 m)
Saved -> Port Moresby Jacksons International Airport (44.5 m)
```



```json
[
    {
        "id": 1,
        "name": "Goroka Airport",
        "city": "Goroka",
        "country": "Papua New Guinea",
        "iata": "GKA",
        "icao": "AYGA",
        "lat": -6.081689834590001,
        "lon": 145.391998291,
        "altitude_ft": 5282,
        "timezone": "Pacific/Port_Moresby",
        "type": "airport",
        "altitude_m": 1609.95
    },
    {
        "id": 2,
        "name": "Madang Airport",
        "city": "Madang",
        "country": "Papua New Guinea",
        "iata": "MAG",
        "icao": "AYMD",
        "lat": -5.20707988739,
        "lon": 145.789001465,
        "altitude_ft": 20,
        "timezone": "Pacific/Port_Moresby",
        "type": "airport",
        "altitude_m": 6.1
    },
    {
        "id": 3,
        "name": "Mount Hagen Kagamuga Airport",
        "city": "Mount Hagen",
        "country": "Papua New Guinea",
        "iata": "HGU",
        "icao": "AYMH",
        "lat": -5.826789855957031,
        "lon": 144.29600524902344,
        "altitude_ft": 5388,
        "timezone": "Pacific/Port_Moresby",
        "type": "airport",
        "altitude_m": 1642.26
```