Exercise 2:

Aim:

To understand and implement real-time data communication using Apache Kafka by building a Python producer that sends messages and a Python consumer that receives them via a Kafka topic. This exercise demonstrates core messaging concepts such as publish-subscribe, message queuing, brokers, and topic-based communication.

Procedure:

Phase 1:

- Set Up Kafka Broker Using Docker Compose
- Create a folder named **Kafka_Lab** on your Desktop.
- Inside the folder, create a **docker-compose.yml** file defining Kafka and Zookeeper services.
- Run **docker-compose up -d** to start Kafka and Zookeeper containers.
- Verify the setup with **docker ps** to ensure both containers are running.

Phase 2:

- Create a Kafka Topic
- Enter the Kafka container using: **docker exec -it kafka /bin/bash**
- create a topic named user-events:
    - **kafka-topics --create --topic user-events --bootstrap-server localhost:9092 --partitions 1 --replication-factor 1**
- **Exit**

Phase 3:

- Set Up Python Environment
- Create and activate a virtual environment:
    - **python -m venv kafka_env**
    - **kafka_env\Scripts\activate** # Windows
- Install the Kafka Python client:
    - **pip install kafka-python**

Phase 4:

- Create producer.py to simulate and send JSON-encoded user data to Kafka.
- Create consumer.py to read and display the JSON messages from the Kafka topic.

24MCA010
GIRIDHAR N

Phase 5:

- Run and Test the System
- In one terminal, run the consumer:
  - **python consumer.py**
- In another terminal, run the producer:
  - **python producer.py**
- Observe how messages sent by the producer are consumed and printed in real-time by the consumer.

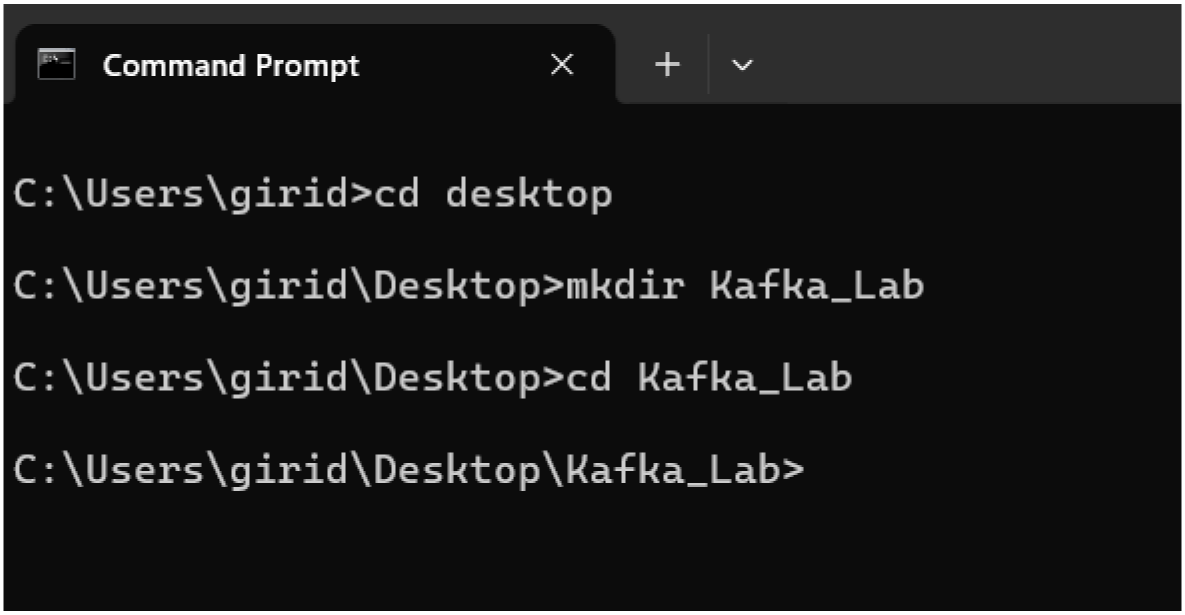
Phase 6:

- To stop and remove Kafka and Zookeeper containers, run:
  - **docker-compose down**

Output:

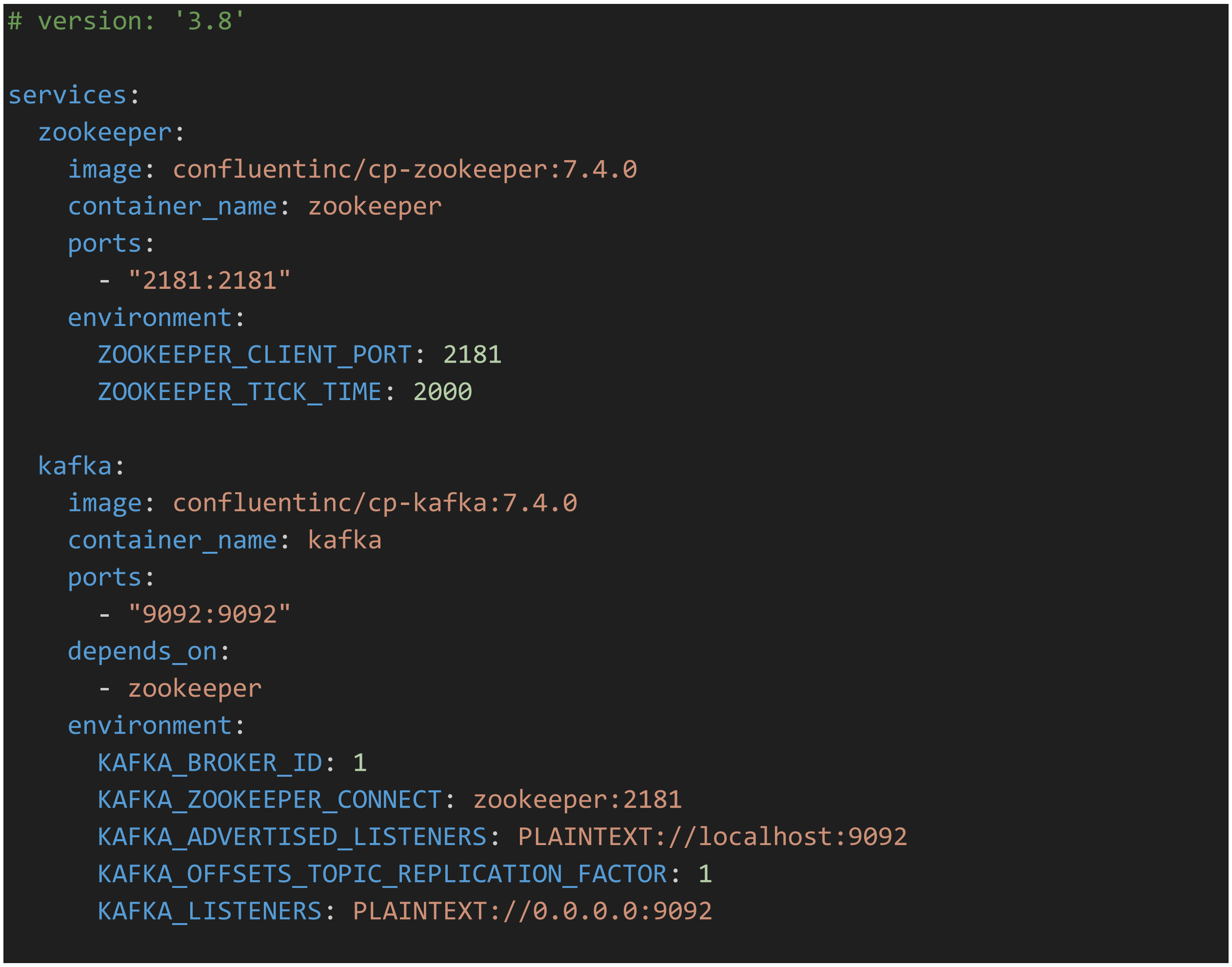
Phase 1:

- Creating folder.



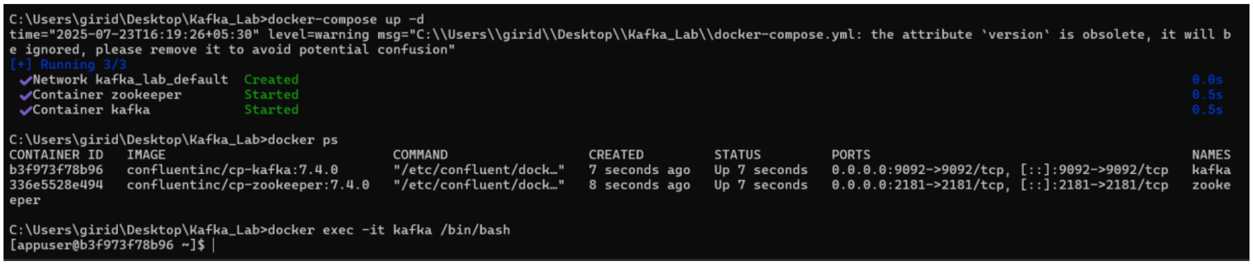
```
C:\Users\girid>cd desktop

C:\Users\girid\Desktop>mkdir Kafka_Lab

C:\Users\girid\Desktop>cd Kafka_Lab

C:\Users\girid\Desktop\Kafka_Lab>
```

- Creating docker-compose.yml file inside Kafka_Lab:



```yaml
# version: '3.8'

services:
  zookeeper:
    image: confluentinc/cp-zookeeper:7.4.0
    container_name: zookeeper
    ports:
      - "2181:2181"
    environment:
      ZOOKEEPER_CLIENT_PORT: 2181
      ZOOKEEPER_TICK_TIME: 2000

  kafka:
    image: confluentinc/cp-kafka:7.4.0
    container_name: kafka
    ports:
      - "9092:9092"
    depends_on:
      - zookeeper
    environment:
      KAFKA_BROKER_ID: 1
      KAFKA_ZOOKEEPER_CONNECT: zookeeper:2181
      KAFKA_ADVERTISED_LISTENERS: PLAINTEXT://localhost:9092
      KAFKA_OFFSETS_TOPIC_REPLICATION_FACTOR: 1
      KAFKA_LISTENERS: PLAINTEXT://0.0.0.0:9092
```

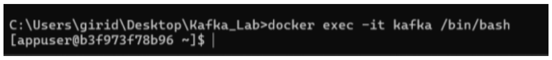
- compose up and docker ps



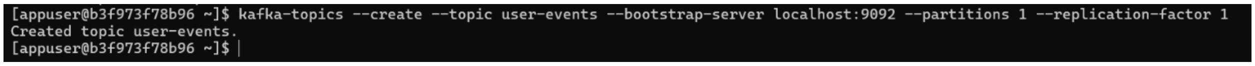
```
C:\Users\girid\Desktop\Kafka_Lab>docker-compose up -d
time="2025-07-23T16:19:26+05:30" level=warning msg="C:\\Users\\girid\\Desktop\\Kafka_Lab\\docker-compose.yml: the attribute `version` is obsolete, it will b
e ignored, please remove it to avoid potential confusion"
[+] Running 3/3
 ✔Network kafka_lab_default  Created                                                                                                                   0.0s
 ✔Container zookeeper        Started                                                                                                                   0.5s
 ✔Container kafka            Started                                                                                                                   0.5s

C:\Users\girid\Desktop\Kafka_Lab>docker ps
CONTAINER ID   IMAGE                             COMMAND                  CREATED         STATUS         PORTS                                        NAMES
b3f973f78b96   confluentinc/cp-kafka:7.4.0       "/etc/confluent/dock…"   7 seconds ago   Up 7 seconds   0.0.0.0:9092->9092/tcp, [::]:9092->9092/tcp   kafka
336e5528e494   confluentinc/cp-zookeeper:7.4.0   "/etc/confluent/dock…"   8 seconds ago   Up 7 seconds   0.0.0.0:2181->2181/tcp, [::]:2181->2181/tcp   zooke
eper

C:\Users\girid\Desktop\Kafka_Lab>docker exec -it kafka /bin/bash
[appuser@b3f973f78b96 ~]$ |
```

Phase 2:

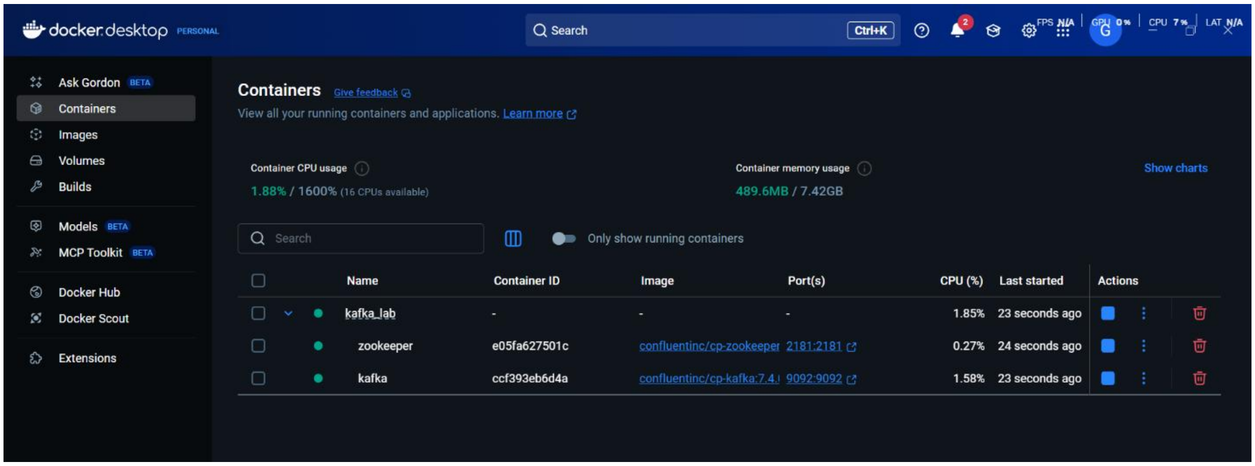
- Kafka container



```
C:\Users\girid\Desktop\Kafka_Lab>docker exec -it kafka /bin/bash
[appuser@b3f973f78b96 ~]$ |
```



```
[appuser@b3f973f78b96 ~]$ kafka-topics --create --topic user-events --bootstrap-server localhost:9092 --partitions 1 --replication-factor 1
Created topic user-events.
[appuser@b3f973f78b96 ~]$ |
```



Containers

View all your running containers and applications.

Container CPU usage

1.88% / 1600% (16 CPUs available)

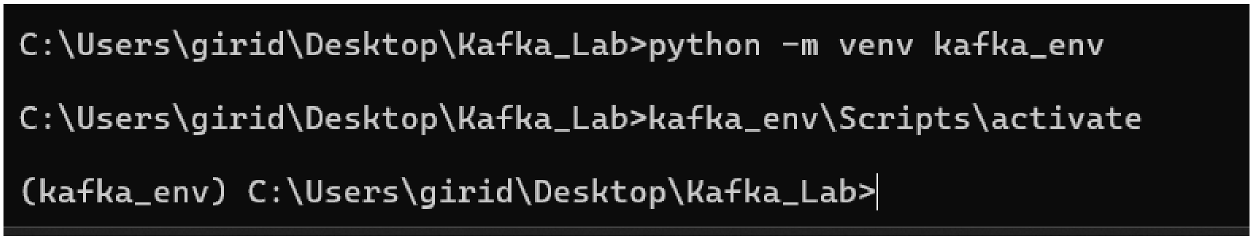
Container memory usage

489.6MB / 7.42GB

	Name	Container ID	Image	Port(s)	CPU (%)	Last started	Actions
□	kafka_lab	-	-	-	1.85%	23 seconds ago	
□	zookeeper	e05fa627501c	confluentinc/cp-zookeeper	2181:2181	0.27%	24 seconds ago	
□	kafka	ccf393eb6d4a	confluentinc/cp-kafka:7.4.(	9092:9092	1.58%	23 seconds ago	

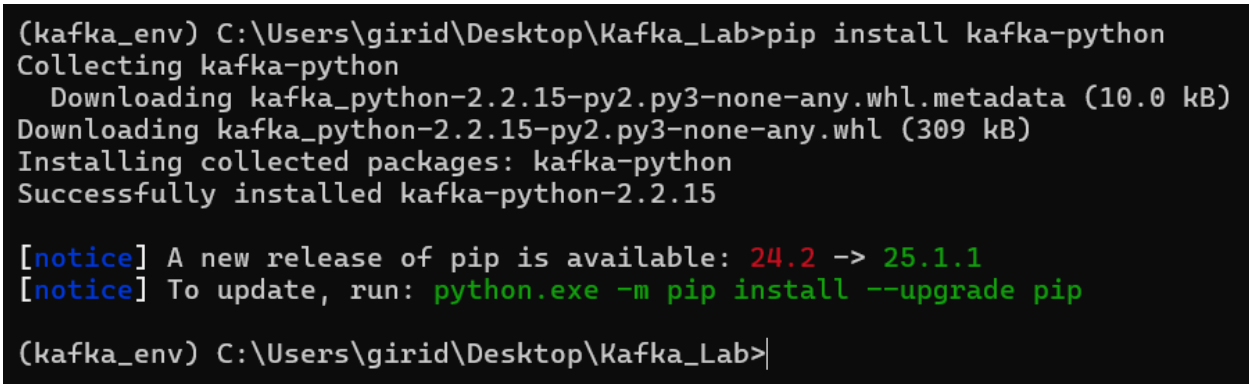
Phase 3:

- Creating python venv



```
C:\Users\girid\Desktop\Kafka_Lab>python -m venv kafka_env

C:\Users\girid\Desktop\Kafka_Lab>kafka_env\Scripts\activate

(kafka_env) C:\Users\girid\Desktop\Kafka_Lab>|
```

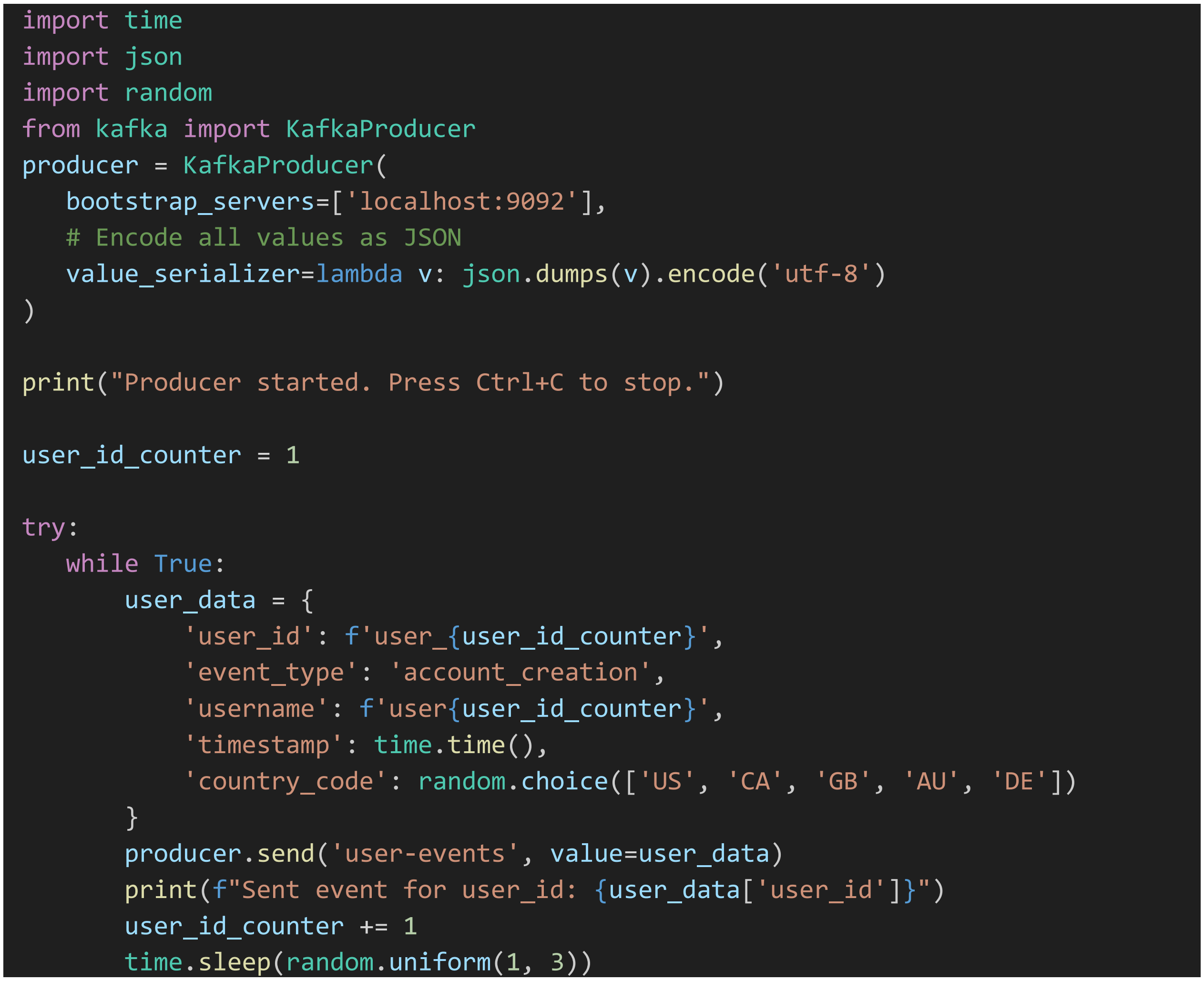
Installing kafka-python:



```
(kafka_env) C:\Users\girid\Desktop\Kafka_Lab>pip install kafka-python
Collecting kafka-python
  Downloading kafka_python-2.2.15-py2.py3-none-any.whl.metadata (10.0 kB)
Downloading kafka_python-2.2.15-py2.py3-none-any.whl (309 kB)
Installing collected packages: kafka-python
Successfully installed kafka-python-2.2.15

[notice] A new release of pip is available: 24.2 -> 25.1.1
[notice] To update, run: python.exe -m pip install --upgrade pip

(kafka_env) C:\Users\girid\Desktop\Kafka_Lab>
```

Phase 4:

producer.py:



```python
import time
import json
import random
from kafka import KafkaProducer
producer = KafkaProducer(
    bootstrap_servers=['localhost:9092'],
    # Encode all values as JSON
    value_serializer=lambda v: json.dumps(v).encode('utf-8')
)

print("Producer started. Press Ctrl+C to stop.")

user_id_counter = 1

try:
    while True:
        user_data = {
            'user_id': f'user_{user_id_counter}',
            'event_type': 'account_creation',
            'username': f'user{user_id_counter}',
            'timestamp': time.time(),
            'country_code': random.choice(['US', 'CA', 'GB', 'AU', 'DE'])
        }
        producer.send('user-events', value=user_data)
        print(f"Sent event for user_id: {user_data['user_id']}")
        user_id_counter += 1
        time.sleep(random.uniform(1, 3))
```

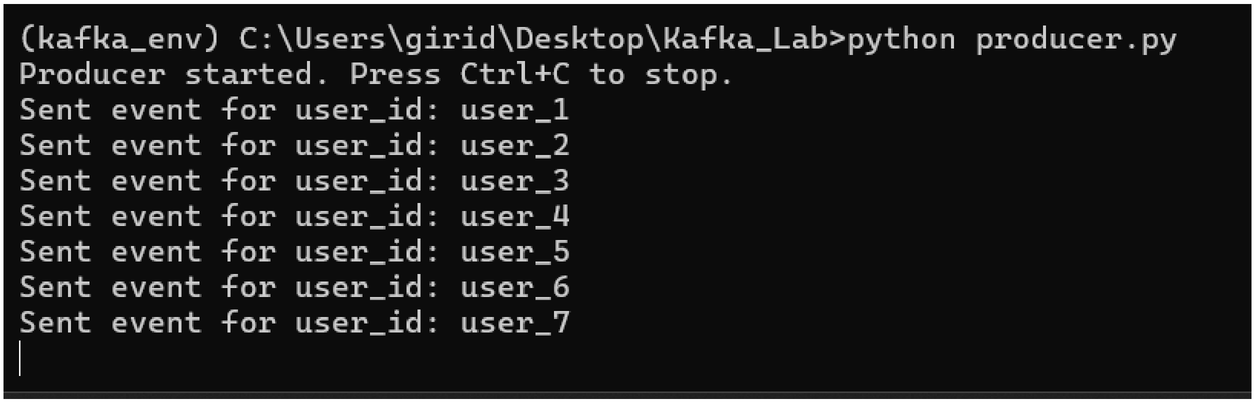
```
 except KeyboardInterrupt:
    print("Stopping producer.")
 finally:
    producer.flush()
    producer.close()
    print("Producer closed.")
```

consumer.py:

```
import json
from kafka import KafkaConsumer
consumer = KafkaConsumer(
    'user-events',  # The topic to subscribe to
    bootstrap_servers=['localhost:9092'],
    auto_offset_reset='earliest',  # Start from the earliest message
    value_deserializer=lambda m: json.loads(m.decode('utf-8'))  # Decode JSON
messages
)

print("Consumer started. Waiting for messages... (Press Ctrl+C to stop)")

try:
    for message in consumer:
        event_data = message.value  # Deserialized JSON data
        print("\n-------------------------------------------")
        print("Received new event:")
        print(f" Topic: {message.topic}")
        print(f" Partition: {message.partition}, Offset: {message.offset}")
        print(f" User ID: {event_data.get('user_id')}")
        print(f" Event Type: {event_data.get('event_type')}")
        print(f" Username: {event_data.get('username')}")
        print("-------------------------------------------")
except KeyboardInterrupt:
    print("\nStopping consumer.")
finally:
    consumer.close()
    print("Consumer closed.")
```

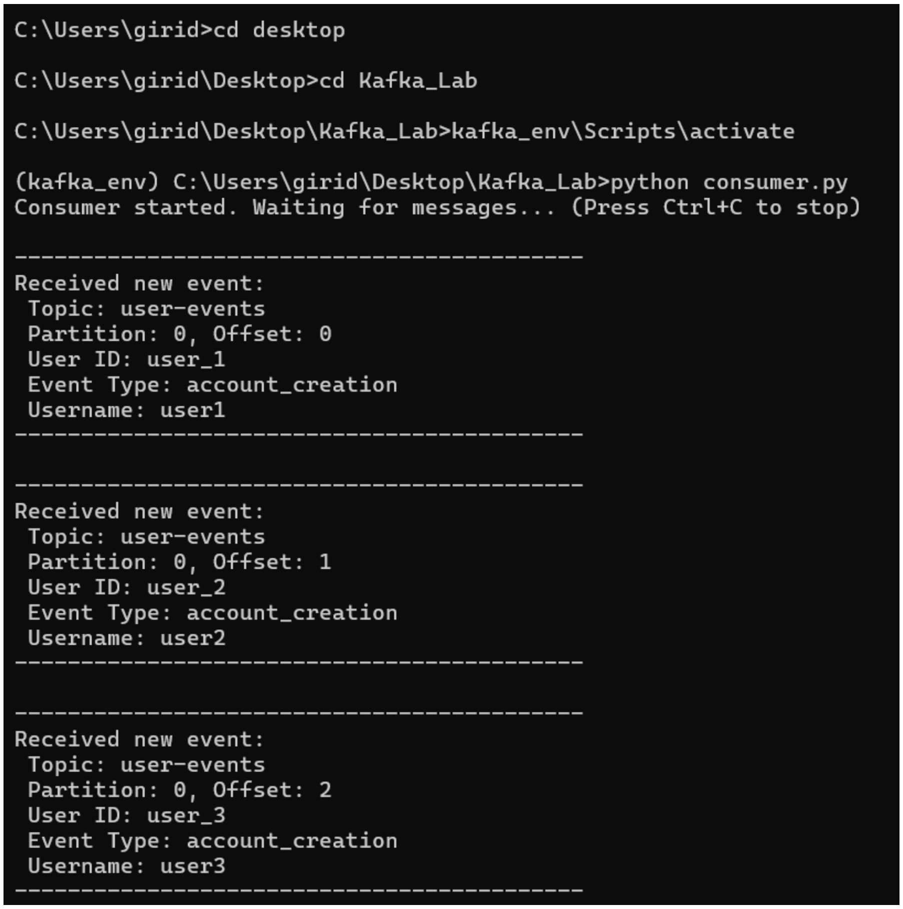
Phase 5:

- **producer.py:**



```
(kafka_env) C:\Users\girid\Desktop\Kafka_Lab>python producer.py
Producer started. Press Ctrl+C to stop.
Sent event for user_id: user_1
Sent event for user_id: user_2
Sent event for user_id: user_3
Sent event for user_id: user_4
Sent event for user_id: user_5
Sent event for user_id: user_6
Sent event for user_id: user_7
```

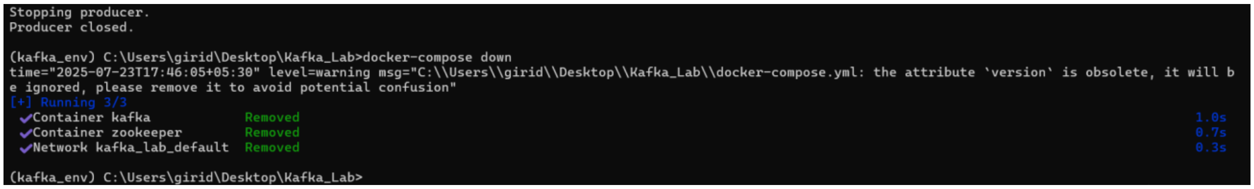
- **consumer.py**



```
C:\Users\girid>cd desktop

C:\Users\girid\Desktop>cd Kafka_Lab

C:\Users\girid\Desktop\Kafka_Lab>kafka_env\Scripts\activate

(kafka_env) C:\Users\girid\Desktop\Kafka_Lab>python consumer.py
Consumer started. Waiting for messages... (Press Ctrl+C to stop)

-------------------------------------------
Received new event:
 Topic: user-events
 Partition: 0, Offset: 0
 User ID: user_1
 Event Type: account_creation
 Username: user1
-------------------------------------------

-------------------------------------------
Received new event:
 Topic: user-events
 Partition: 0, Offset: 1
 User ID: user_2
 Event Type: account_creation
 Username: user2
-------------------------------------------

-------------------------------------------
Received new event:
 Topic: user-events
 Partition: 0, Offset: 2
 User ID: user_3
 Event Type: account_creation
 Username: user3
-------------------------------------------
```

Phase 6:

- Removing kafka containers



```
Stopping producer.
Producer closed.

(kafka_env) C:\Users\girid\Desktop\Kafka_Lab>docker-compose down
time="2025-07-23T17:46:05+05:30" level=warning msg="C:\\Users\\girid\\Desktop\\Kafka_Lab\\docker-compose.yml: the attribute `version` is obsolete, it will b
e ignored, please remove it to avoid potential confusion"
[+] Running 3/3
 ✓Container kafka           Removed                                                                                          1.0s
 ✓Container zookeeper       Removed                                                                                          0.7s
 ✓Network kafka_lab_default  Removed                                                                                          0.3s

(kafka_env) C:\Users\girid\Desktop\Kafka_Lab>
```