

Approach to the Lab Assignment – 3

Step-1: Installation of required packages

- Initialized a conda environment and installed 'python' and activated it.

```
$ conda create --name cs594 python=3.10  
$ conda activate cs594
```

- Installed 'kafka-python' and 'kcat'

```
$ python -m pip install kafka-python  
$ brew install kcat
```

Step-2: Established a secure SSH tunnel to the Kafka server

- Used SSH to create a tunnel to the Kafka server and ran it in background using -NTf and local port forwarding -L

```
$ ssh -o ServerAliveInterval=60 -L 9092:localhost:9092 vmeka@cs594.cs.uic.edu -NTf
```

- entered the password for my NetID after issuing the command and did the demo for lab.

Step-3: Implemented the Producer Mode

```
Producer Mode -> Writes Data to Broker  
  
# Create a producer to write data to kafka  
# Ref: https://kafka-python.readthedocs.io/en/master/apidoc/KafkaProducer.html  
  
# [TODO]: Replace '...' with the address of your Kafka bootstrap server  
producer = KafkaProducer(bootstrap_servers=["localhost:9092"],  
                          value_serializer=lambda x: dumps(x).encode('utf-8'))  
  
# [TODO]: Add cities of your choice  
cities = ["Mumbai", "Delhi", "Hyderabad", "Kolkata", "Bangalore", "Chennai", "Vizag"]  
  
# Write data via the producer  
print("Writing to Kafka Broker")  
for i in range(10):  
    data = f'{datetime.now().strftime("%Y-%m-%d %H:%M:%S")},{cities[randint(0, len(cities)-1)]},{randint(18, 32)}°C'  
    print(f"Writing: {data}")  
    producer.send(topic=topic, value=data)  
    sleep(1)  
[2]  
Python  
... Writing to Kafka Broker  
Writing: 2025-02-06 23:27:39,Kolkata,31°C  
Writing: 2025-02-06 23:27:40,Delhi,28°C  
Writing: 2025-02-06 23:27:41,Bangalore,27°C  
Writing: 2025-02-06 23:27:42,Hyderabad,26°C  
Writing: 2025-02-06 23:27:43,Vizag,32°C  
Writing: 2025-02-06 23:27:44,Vizag,26°C  
Writing: 2025-02-06 23:27:45,Hyderabad,26°C  
Writing: 2025-02-06 23:27:46,Hyderabad,27°C  
Writing: 2025-02-06 23:27:47,Bangalore,21°C  
Writing: 2025-02-06 23:27:48,Bangalore,23°C
```

Step-4: Implemented the Consumer Mode

```
Consumer Mode -> Reads Data from Broker

# Create a consumer to read data from kafka
# Ref: https://kafka-python.readthedocs.io/en/master/apidoc/KafkaConsumer.html
topic="recitation-x"
# [TODO]: Complete the missing ... parameters/arguments using the Kafka documentation
consumer = KafkaConsumer(
    topic,
    bootstrap_servers=["localhost:9092"],
    auto_offset_reset="earliest", #Experiment with different values
    # Commit that an offset has been read
    enable_auto_commit=True,
    # How often to tell Kafka, an offset has been read
    auto_commit_interval_ms=1000,
    consumer_timeout_ms=5000
)

print('Reading Kafka Broker')
for message in consumer:
    message = message.value.decode()
    # Default message.value type is bytes!
    print.loads(message)
    os.system(f"echo {message} >> kafka_log.csv")

# Closing consumer and producer connections
consumer.close()
producer.close()
```

[5] Python

```
... Reading Kafka Broker
{'timestamp': '2025-02-03 16:02:00', 'city': 'Denver', 'temperature': '259C'}
{'timestamp': '2025-02-03 16:02:02', 'city': 'New York', 'temperature': '280C'}
{'timestamp': '2025-02-03 16:02:03', 'city': 'Chicago', 'temperature': '209C'}
{'timestamp': '2025-02-03 16:02:04', 'city': 'New Jersey', 'temperature': '230C'}
```

Step-5 Verified Kafka_log.csv

```
Preview lab03.md | kafka_log.csv | KafkaDemo.ipynb M | movies_data.csv U

mlip-kafka-lab > kafka_log.csv
355 {timestamp: 2025-02-06 22:58:39, city: Los Angeles, temperature: 29\u00baC}
356 {timestamp: 2025-02-06 22:58:40, city: Los Angeles, temperature: 23\u00baC}
357 {timestamp: 2025-02-06 22:58:41, city: Los Angeles, temperature: 27\u00baC}
358 {timestamp: 2025-02-06 22:58:42, city: Denver, temperature: 23\u00baC}
359 {timestamp: 2025-02-06 22:58:43, city: Los Angeles, temperature: 28\u00baC}
360 {timestamp: 2025-02-06 22:58:44, city: Denver, temperature: 27\u00baC}
361 2025-02-06 23:15:21,Banglore,24\u00baC
362 2025-02-06 23:15:22,Chennai,26\u00baC
363 2025-02-06 23:15:23,Hyderabad,26\u00baC
364 2025-02-06 23:15:24,Banglore,27\u00baC
365 2025-02-06 23:15:25,Vizag,28\u00baC
366 2025-02-06 23:15:26,Chennai,23\u00baC
367 2025-02-06 23:15:27,Delhi,27\u00baC
368 2025-02-06 23:15:28,Chennai,23\u00baC
369 2025-02-06 23:15:29,Kolkata,30\u00baC
370 2025-02-06 23:15:30,Delhi,28\u00baC
371 2025-02-06 23:23:32,Chennai,19\u00baC
372 2025-02-06 23:23:33,Delhi,31\u00baC
373 2025-02-06 23:23:34,Kolkata,24\u00baC
374 2025-02-06 23:23:35,Vizag,32\u00baC
375 2025-02-06 23:23:36,Kolkata,20\u00baC
376 2025-02-06 23:23:37,Mumbai,25\u00baC
377 2025-02-06 23:23:38,Hyderabad,27\u00baC
378 2025-02-06 23:23:39,Delhi,27\u00baC
379 2025-02-06 23:23:40,Chennai,18\u00baC
380 2025-02-06 23:23:41,Chennai,18\u00baC
381 2025-02-06 23:27:39,Kolkata,31\u00baC
382 2025-02-06 23:27:40,Delhi,28\u00baC
383 2025-02-06 23:27:41,Banglore,27\u00baC
384 2025-02-06 23:27:42,Hyderabad,26\u00baC
385 2025-02-06 23:27:43,Vizag,32\u00baC
386 2025-02-06 23:27:44,Vizag,26\u00baC
387 2025-02-06 23:27:45,Hyderabad,26\u00baC
388 2025-02-06 23:27:46,Hyderabad,27\u00baC
389 2025-02-06 23:27:47,Banglore,21\u00baC
390 2025-02-06 23:27:48,Banglore,23\u00baC
391
```

Step-6: Used Kafka's CLI tool kcat to manage and monitor Kafka topics and messages

```
$ kcat -b localhost:9092 -t recitation-x -o beginning -C -f '\nKey (%K bytes): %k\t\nValue (%S bytes): %s\nTimestamp: %T\tPartition: %p\tOffset: %o\n--\n' -e
```

```
#kcat command: connect to local Kafka broker, specify a topic, and consume messages from the earliest offset
kcat_command = f"kcat -b localhost:9092 -t {topic} -o beginning -C -f '\nKey (%K bytes): %k\t\nValue (%S bytes): %s\nTimestamp: %T\tPartition: %p\tOffset: %o\n--\n' -e"
os.system(kcat_command)

...

Key (-1 bytes):
Value (82 bytes): {"timestamp": "2025-02-03 16:02:00", "city": "Denver", "temperature": "25\u00baC"}
Timestamp: 1738620121358      Partition: 0      Offset: 0
--

Key (-1 bytes):
Value (84 bytes): {"timestamp": "2025-02-03 16:02:02", "city": "New York", "temperature": "28\u00baC"}
Timestamp: 1738620122359      Partition: 0      Offset: 1
--

Key (-1 bytes):
Value (83 bytes): {"timestamp": "2025-02-03 16:02:03", "city": "Chicago", "temperature": "20\u00baC"}
Timestamp: 1738620123360      Partition: 0      Offset: 2
--

Key (-1 bytes):
Value (86 bytes): {"timestamp": "2025-02-03 16:02:04", "city": "New Jersey", "temperature": "23\u00baC"}
Timestamp: 1738620124364      Partition: 0      Offset: 3
--

Key (-1 bytes):
Value (84 bytes): {"timestamp": "2025-02-03 16:02:05", "city": "New York", "temperature": "25\u00baC"}
Timestamp: 1738620125367      Partition: 0      Offset: 4
--

...

Key (-1 bytes):
Value (40 bytes): "2025-02-06 23:27:48,Banglore,23\u00baC"
Timestamp: 1738906068529      Partition: 0      Offset: 389
--

Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...
% Reached end of topic recitation-x [0] at offset 390: exiting
```

```
$ kcat -b localhost:9092 -L
```

```
os.system("kcat -b localhost:9092 -L")

[9]

...

Metadata for all topics (from broker 1: localhost:9092/1):
1 brokers:
  broker 1 at localhost:9092 (controller)
34 topics:
  topic "recitation-x" with 1 partitions:
    partition 0, leader 1, replicas: 1, isrs: 1
  topic "movielog3" with 1 partitions:
    partition 0, leader 1, replicas: 1, isrs: 1
  topic "movielog2" with 1 partitions:
    partition 0, leader 1, replicas: 1, isrs: 1
  topic "movielog1" with 1 partitions:
    partition 0, leader 1, replicas: 1, isrs: 1
  topic "recitation-abcd" with 1 partitions:
    partition 0, leader 1, replicas: 1, isrs: 1
  topic "recitation-rohan" with 1 partitions:
    partition 0, leader 1, replicas: 1, isrs: 1
  topic "movielog10" with 1 partitions:
    partition 0, leader 1, replicas: 1, isrs: 1
  topic "topic-apoorv-2" with 1 partitions:
    partition 0, leader 1, replicas: 1, isrs: 1
  topic "nutrition_facts" with 1 partitions:
    partition 0, leader 1, replicas: 1, isrs: 1
  topic "topic-apoorv-3" with 1 partitions:
    partition 0, leader 1, replicas: 1, isrs: 1
  topic "city_topic" with 1 partitions:
    partition 0, leader 1, replicas: 1, isrs: 1
  ...
  partition 48, leader 1, replicas: 1, isrs: 1
  partition 49, leader 1, replicas: 1, isrs: 1
  topic "ride_request" with 1 partitions:
    partition 0, leader 1, replicas: 1, isrs: 1

Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...

... 0
```

Step-7: (Optional) Reading movielog streams

- found the list of all topics and then read some movielog streams to get an idea of what the data looks like and saved them into a csv file

```
$ kcat -b localhost:9092 -t movielog2 -C -e > movies_data.csv
```

```
os.system("kcat -b localhost:9092 -t movielog2 -C -e > movies_data.csv")

[ ]

... % Reached end of topic movielog2 [0] at offset 3968: exiting

... 0
```

```
mlip-kafka-lab > movies_data.csv
3937 2025-02-01T17:35:38,147936,GET /data/m/Likainen+pommi+2011/4.mpg
3938 2025-02-01T17:35:38,148140,GET /data/m/nostalgia+1983/4.mpg
3939 2025-02-01T17:35:38,30385,GET /data/m/different+for+girls+1996/11.mpg
3940 2025-02-01T17:35:38,97902,GET /data/m/house+of+fools+2002/2.mpg
3941 2025-02-01T17:35:38,6884,GET /data/m/mickey+donald+goofy+the+three+musketeers+2004/1.mpg
3942 2025-02-01T17:35:38,4416,GET /data/m/interstellar+2014/10.mpg
3943 2025-02-01T17:35:38,134335,GET /data/m/yankee+doodle+dandy+1942/6.mpg
3944 2025-02-01T17:35:38,8922,GET /data/m/river+queen+2005/10.mpg
3945 2025-02-01T17:35:38,137237,GET /data/m/harry+potter+and+the+deathly+hallows+part+2+2011/8.mpg
3946 2025-02-01T17:35:38,139932,GET /data/m/wishful+thinking+1996/5.mpg
3947 2025-02-01T17:35:38,69300,GET /data/m/the+home+of+dark+butterflies+2008/0.mpg
3948 2025-02-01T17:35:39,49549,GET /data/m/confessions+of+a+superhero+2007/4.mpg
3949 2025-02-01T17:35:39,17284,GET /data/m/whiplash+2014/6.mpg
3950 2025-02-01T17:35:39,133133,GET /data/m/pandorum+2009/7.mpg
3951 2025-02-01T17:35:39,32553,GET /data/m/the+gruffalo+2009/2.mpg
3952 2025-02-01T17:35:39,38904,GET /data/m/the+war+within+2005/1.mpg
3953 2025-02-01T17:35:39,133708,GET /data/m/pandoras+promise+2013/9.mpg
3954 2025-02-01T17:35:39,50286,GET /data/m/pulse+2001/8.mpg
3955 2025-02-01T17:35:39,109976,GET /data/m/king+of+hearts+1966/0.mpg
3956 2025-02-01T17:35:40,21152,GET /data/m/a+pain+in+the+ass+2008/4.mpg
3957 2025-02-01T17:35:40,79355,GET /data/m/young+thugs+nostalgia+1998/3.mpg
3958 2025-02-01T17:35:40,108445,GET /data/m/the+princess+and+the+frog+2009/4.mpg
3959 2025-02-01T17:35:40,6992,GET /data/m/blade+runner+1982/3.mpg
3960 2025-02-01T17:35:40,93006,GET /data/m/exists+2014/3.mpg
3961 2025-02-01T17:35:40,3428,GET /data/m/p.s.+2004/5.mpg
3962 2025-02-01T17:35:40,12070,GET /data/m/leave+her+to+heaven+1945/7.mpg
3963 2025-02-01T17:35:40,101504,GET /data/m/burnt+by+the+sun+1994/1.mpg
3964 2025-02-01T17:35:40,57140,GET /data/m/the+war+within+2005/0.mpg
3965 2025-02-01T17:35:40,66929,GET /data/m/inglourious+basterds+2009/0.mpg
3966 2025-02-01T17:35:40,89798,GET /data/m/harry+potter+and+the+prisoner+of+azkaban+2004/0.mpg
3967 2025-02-01T17:35:40,102017,GET /data/m/the+tale+of+zatichi+1962/0.mpg
3968 2025-02-01T17:35:40,36337,GET /data/m/ishq+1997/0.mpg
3969
```

Step-8: Closed the SSH tunnel to the Kafka server

- Killed the connection using the kill command after looking up the process id with the port and passing it as arguments.

```
$ lsof -ti:9092 | xargs kill -9
```
