1. Create the Dataproc cluster

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. Cluster is active and running

A screenshot of a computer

Description automatically generated

1. Now from the vm instances open the SSH in new browser

A screenshot of a computer

Description automatically generated

1. Check the pyspar and the python version by typing “pyspark” command in the terminal

A screenshot of a computer program

Description automatically generated

1. Setup Kafka
   1. Download Kafka:

$ wget https://downloads.apache.org/kafka/3.5.1/kafka\_2.12-3.5.1.tgz

$ tar -xvf kafka\_2.12-3.3.1.tgz

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

1. Setup Spark

$ pip3 install msgpack

$ pip3 install kafka-python

A screen shot of a computer

Description automatically generated

1. Start Zookeeper:

# Start Zookeeper

cd kafka\_2.12-3.5.1/

bin/zookeeper-server-start.sh config/zookeeper.properties

A screen shot of a computer

Description automatically generated

1. Now open another ssh terminal by again clicking the vm instances -> ssh this will be terminal 2.
2. Now in this terminal start kafka broker:

# Start Kafka broker in a new (second) terminal.

cd kafka\_2.12-3.5.1/

bin/kafka-server-start.sh config/server.properties

A screenshot of a computer program

Description automatically generated

1. Now again open a new terminal and in this we create the kafka topic:

cd kafka\_2.12-3.5.1/

# Create Kafka topic in a new (third) terminal.

bin/kafka-topics.sh --create --topic input\_event --bootstrap-server localhost:9092 --partitions 3 --replication-factor 1

A screenshot of a computer program

Description automatically generated

1. In this same terminal now we Create and Run the Kafka Consumer
   1. We create a Python script named consumer.py and implement the consumer logic using the kafka-python library and run it.

vi consumer.py

python3 consumer.py

Code in Consumer.py:

from kafka import KafkaConsumer

consumer = KafkaConsumer('input\_event', bootstrap\_servers=['localhost:9092'])

for msg in consumer:

print(msg)

A screenshot of a computer program

Description automatically generated

1. Open a new terminal terminal 4 now we Create and Run the Kafka Producer.
   1. We create a Python script named consumer.py and implement the consumer logic using the kafka-python library and run it.

vi producer.py

python3 producer.py

Code in Producer.py:

from kafka import KafkaProducer

producer = KafkaProducer(bootstrap\_servers='localhost:9092')

producer.send('input\_event', b'(1, Main Menu), (2, Phone), (3, Smart Phone), (4, iPhone)')

producer.close()

A screenshot of a computer

Description automatically generated

1. Now when the producer code in ran we see the output in the consumer terminal i.e., terminal 3

A screenshot of a computer screen

Description automatically generated

1. Here we can see that the message sent from the producer is correctly received by the consumer.