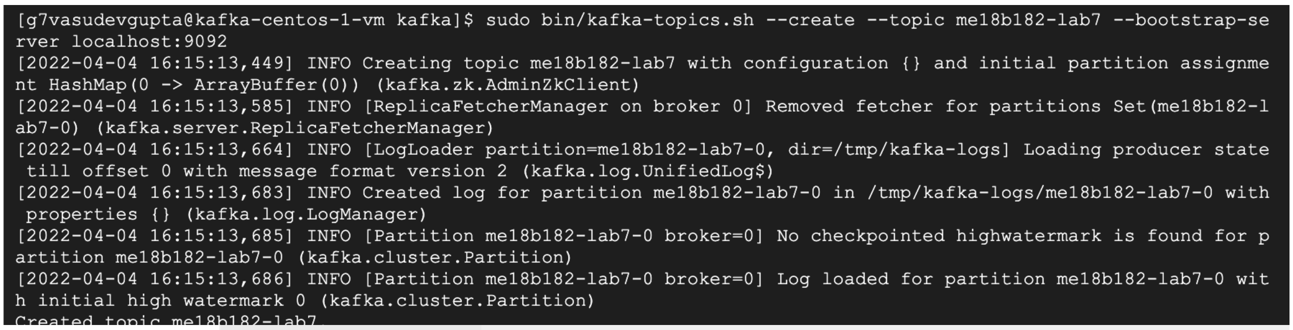
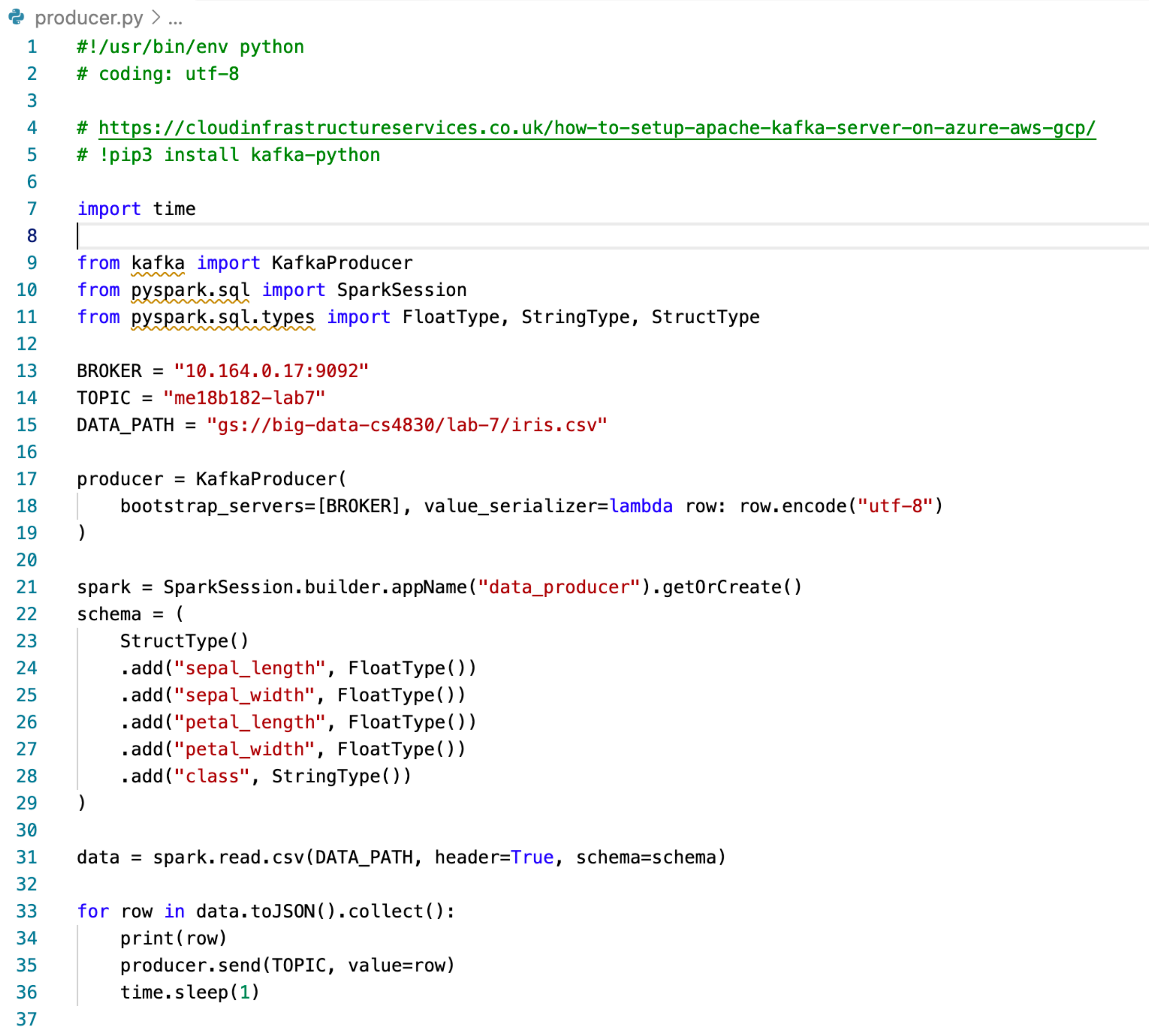
CS4830 : Big Data Lab-7 Assignment

Vasudev Gupta (ME18B182)

1) File producer.py reads every line from iris.csv file line by line and converts it into json. Then it encodes every row and publishes it to topic- ***me18b182-lab7***. This is decoded back when subscriber.py is run and converted into data frame following the same schema used to train the model.



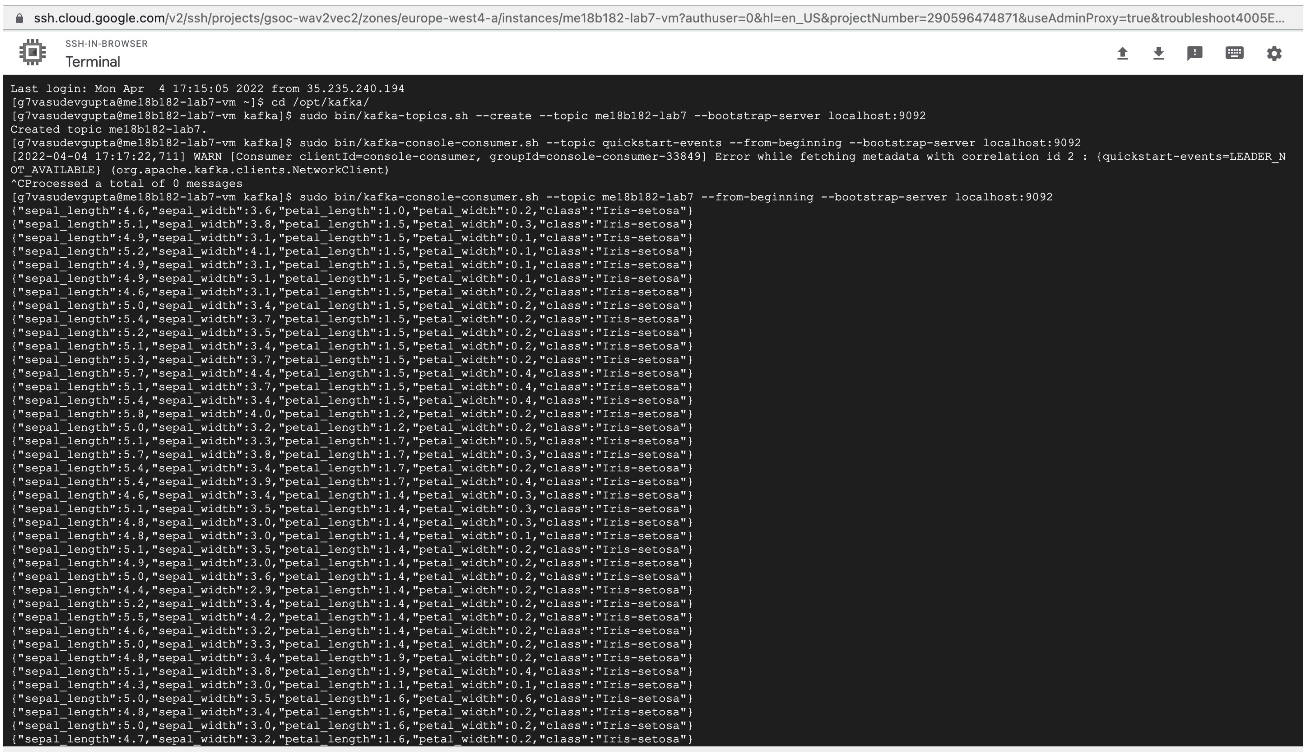
```
[g7vasudevgupta@kafka-centos-1-vm kafka]$ sudo bin/kafka-topics.sh --create --topic me18b182-lab7 --bootstrap-se
rver localhost:9092
[2022-04-04 16:15:13,449] INFO Creating topic me18b182-lab7 with configuration {} and initial partition assignme
nt HashMap(0 -> ArrayBuffer(0)) (kafka.zk.AdminZkClient)
[2022-04-04 16:15:13,585] INFO [ReplicaFetcherManager on broker 0] Removed fetcher for partitions Set(me18b182-l
ab7-0) (kafka.server.ReplicaFetcherManager)
[2022-04-04 16:15:13,664] INFO [LogLoader partition=me18b182-lab7-0, dir=/tmp/kafka-logs] Loading producer state
 till offset 0 with message format version 2 (kafka.log.UnifiedLog$)
[2022-04-04 16:15:13,683] INFO Created log for partition me18b182-lab7-0 in /tmp/kafka-logs/me18b182-lab7-0 with
 properties {} (kafka.log.LogManager)
[2022-04-04 16:15:13,685] INFO [Partition me18b182-lab7-0 broker=0] No checkpointed highwatermark is found for p
artition me18b182-lab7-0 (kafka.cluster.Partition)
[2022-04-04 16:15:13,686] INFO [Partition me18b182-lab7-0 broker=0] Log loaded for partition me18b182-lab7-0 wit
h initial high watermark 0 (kafka.cluster.Partition)
Created topic me18b182-lab7
```



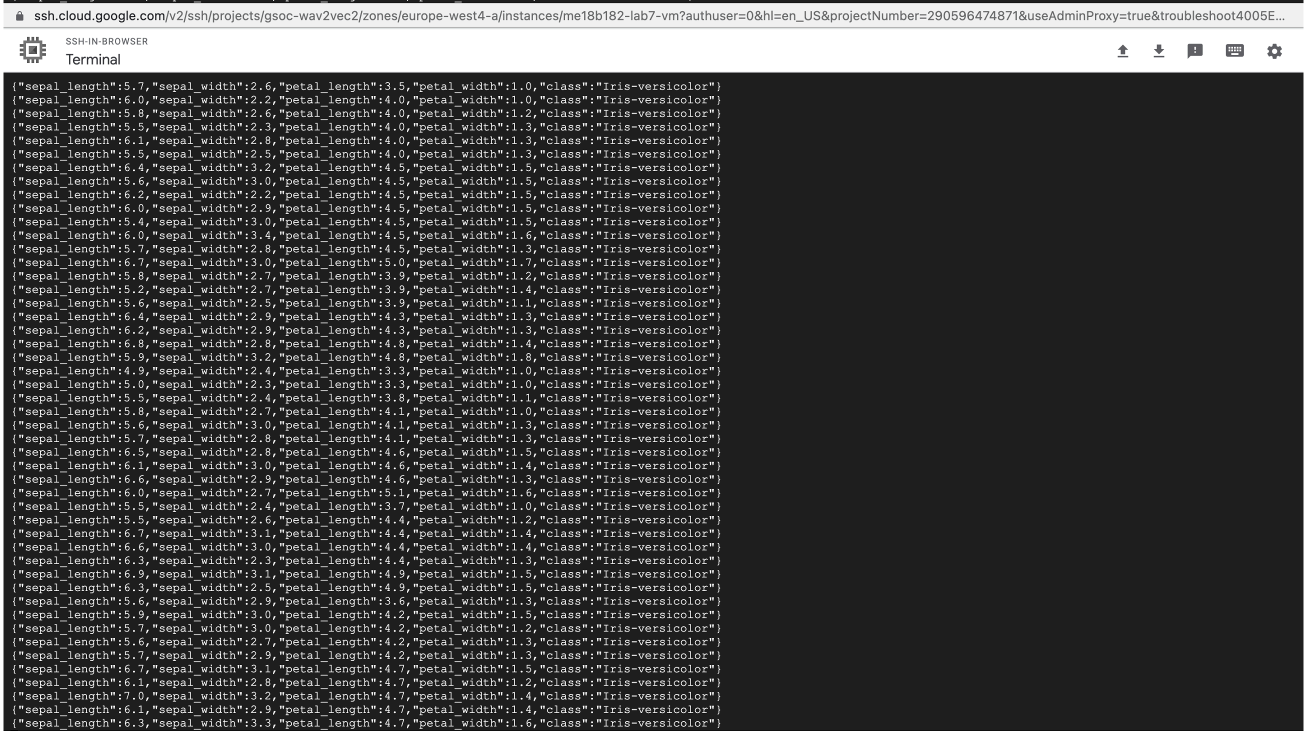
producer.py > ...

```python
#!/usr/bin/env python
# coding: utf-8

# https://cloudinfrastructureservices.co.uk/how-to-setup-apache-kafka-server-on-azure-aws-gcp/
# !pip3 install kafka-python

import time

from kafka import KafkaProducer
from pyspark.sql import SparkSession
from pyspark.sql.types import FloatType, StringType, StructType

BROKER = "10.164.0.17:9092"
TOPIC = "me18b182-lab7"
DATA_PATH = "gs://big-data-cs4830/lab-7/iris.csv"

producer = KafkaProducer(
    bootstrap_servers=[BROKER], value_serializer=lambda row: row.encode("utf-8")
)

spark = SparkSession.builder.appName("data_producer").getOrCreate()
schema = (
    StructType()
    .add("sepal_length", FloatType())
    .add("sepal_width", FloatType())
    .add("petal_length", FloatType())
    .add("petal_width", FloatType())
    .add("class", StringType())
)

data = spark.read.csv(DATA_PATH, header=True, schema=schema)

for row in data.toJSON().collect():
    print(row)
    producer.send(TOPIC, value=row)
    time.sleep(1)

```

The below picture shows the console output in real-time as the producer.py is run.


ssh.cloud.google.com/v2/ssh/projects/gsoc-wav2vec2/zones/europe-west4-a/instances/me18b182-lab7-vm?authuser=0&hl=en_US&projectNumber=290596474871&useAdminProxy=true&troubleshoot4005E...
SSH-IN-BROWSER
Terminal

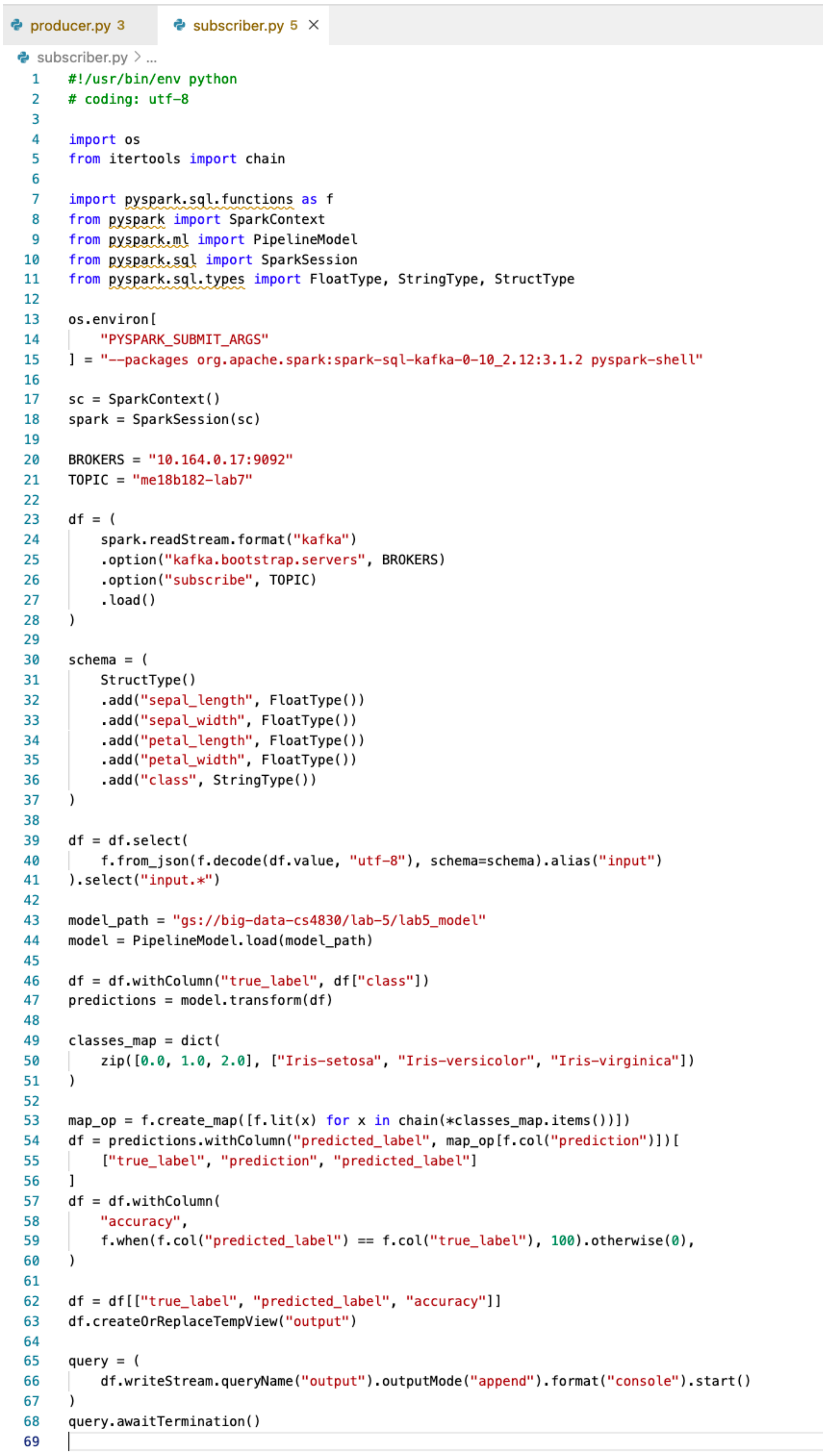
```
Last login: Mon Apr  4 17:15:05 2022 from 35.235.240.194
[g7vasudevgupta@me18b182-lab7-vm ~]$ cd /opt/kafka/
[g7vasudevgupta@me18b182-lab7-vm kafka]$ sudo bin/kafka-topics.sh --create --topic me18b182-lab7 --bootstrap-server localhost:9092
Created topic me18b182-lab7.
[g7vasudevgupta@me18b182-lab7-vm kafka]$ sudo bin/kafka-console-consumer.sh --topic quickstart-events --from-beginning --bootstrap-server localhost:9092
[2022-04-04 17:17:22,711] WARN [Consumer clientId=console-consumer, groupId=console-consumer-33849] Error while fetching metadata with correlation id 2 : {quickstart-events=LEADER_N
OT_AVAILABLE} (org.apache.kafka.clients.NetworkClient)
^CProcessed a total of 0 messages
[g7vasudevgupta@me18b182-lab7-vm kafka]$ sudo bin/kafka-console-consumer.sh --topic me18b182-lab7 --from-beginning --bootstrap-server localhost:9092
{"sepal_length":4.6,"sepal_width":3.6,"petal_length":1.0,"petal_width":0.2,"class":"Iris-setosa"}
{"sepal_length":5.1,"sepal_width":3.8,"petal_length":1.5,"petal_width":0.3,"class":"Iris-setosa"}
{"sepal_length":4.9,"sepal_width":3.1,"petal_length":1.5,"petal_width":0.1,"class":"Iris-setosa"}
{"sepal_length":5.2,"sepal_width":4.1,"petal_length":1.5,"petal_width":0.1,"class":"Iris-setosa"}
{"sepal_length":4.9,"sepal_width":3.1,"petal_length":1.5,"petal_width":0.1,"class":"Iris-setosa"}
{"sepal_length":4.9,"sepal_width":3.1,"petal_length":1.5,"petal_width":0.1,"class":"Iris-setosa"}
{"sepal_length":4.6,"sepal_width":3.1,"petal_length":1.5,"petal_width":0.2,"class":"Iris-setosa"}
{"sepal_length":5.0,"sepal_width":3.4,"petal_length":1.5,"petal_width":0.2,"class":"Iris-setosa"}
{"sepal_length":5.4,"sepal_width":3.7,"petal_length":1.5,"petal_width":0.2,"class":"Iris-setosa"}
{"sepal_length":5.2,"sepal_width":3.5,"petal_length":1.5,"petal_width":0.2,"class":"Iris-setosa"}
{"sepal_length":5.1,"sepal_width":3.4,"petal_length":1.5,"petal_width":0.2,"class":"Iris-setosa"}
{"sepal_length":5.3,"sepal_width":3.7,"petal_length":1.5,"petal_width":0.2,"class":"Iris-setosa"}
{"sepal_length":5.7,"sepal_width":4.4,"petal_length":1.5,"petal_width":0.4,"class":"Iris-setosa"}
{"sepal_length":5.1,"sepal_width":3.7,"petal_length":1.5,"petal_width":0.4,"class":"Iris-setosa"}
{"sepal_length":5.4,"sepal_width":3.4,"petal_length":1.5,"petal_width":0.4,"class":"Iris-setosa"}
{"sepal_length":5.8,"sepal_width":4.0,"petal_length":1.2,"petal_width":0.2,"class":"Iris-setosa"}
{"sepal_length":5.0,"sepal_width":3.2,"petal_length":1.2,"petal_width":0.2,"class":"Iris-setosa"}
{"sepal_length":5.1,"sepal_width":3.3,"petal_length":1.7,"petal_width":0.5,"class":"Iris-setosa"}
{"sepal_length":5.7,"sepal_width":3.8,"petal_length":1.7,"petal_width":0.3,"class":"Iris-setosa"}
{"sepal_length":5.4,"sepal_width":3.4,"petal_length":1.7,"petal_width":0.2,"class":"Iris-setosa"}
{"sepal_length":5.4,"sepal_width":3.9,"petal_length":1.7,"petal_width":0.4,"class":"Iris-setosa"}
{"sepal_length":4.6,"sepal_width":3.4,"petal_length":1.4,"petal_width":0.3,"class":"Iris-setosa"}
{"sepal_length":5.1,"sepal_width":3.5,"petal_length":1.4,"petal_width":0.3,"class":"Iris-setosa"}
{"sepal_length":4.8,"sepal_width":3.0,"petal_length":1.4,"petal_width":0.3,"class":"Iris-setosa"}
{"sepal_length":4.8,"sepal_width":3.0,"petal_length":1.4,"petal_width":0.1,"class":"Iris-setosa"}
{"sepal_length":5.1,"sepal_width":3.5,"petal_length":1.4,"petal_width":0.2,"class":"Iris-setosa"}
{"sepal_length":4.9,"sepal_width":3.0,"petal_length":1.4,"petal_width":0.2,"class":"Iris-setosa"}
{"sepal_length":5.0,"sepal_width":3.6,"petal_length":1.4,"petal_width":0.2,"class":"Iris-setosa"}
{"sepal_length":4.4,"sepal_width":2.9,"petal_length":1.4,"petal_width":0.2,"class":"Iris-setosa"}
{"sepal_length":5.2,"sepal_width":3.4,"petal_length":1.4,"petal_width":0.2,"class":"Iris-setosa"}
{"sepal_length":5.5,"sepal_width":4.2,"petal_length":1.4,"petal_width":0.2,"class":"Iris-setosa"}
{"sepal_length":4.6,"sepal_width":3.2,"petal_length":1.4,"petal_width":0.2,"class":"Iris-setosa"}
{"sepal_length":5.0,"sepal_width":3.3,"petal_length":1.4,"petal_width":0.2,"class":"Iris-setosa"}
{"sepal_length":4.8,"sepal_width":3.4,"petal_length":1.9,"petal_width":0.2,"class":"Iris-setosa"}
{"sepal_length":5.1,"sepal_width":3.8,"petal_length":1.9,"petal_width":0.4,"class":"Iris-setosa"}
{"sepal_length":4.3,"sepal_width":3.0,"petal_length":1.1,"petal_width":0.1,"class":"Iris-setosa"}
{"sepal_length":5.0,"sepal_width":3.5,"petal_length":1.6,"petal_width":0.6,"class":"Iris-setosa"}
{"sepal_length":4.8,"sepal_width":3.4,"petal_length":1.6,"petal_width":0.2,"class":"Iris-setosa"}
{"sepal_length":5.0,"sepal_width":3.0,"petal_length":1.6,"petal_width":0.2,"class":"Iris-setosa"}
{"sepal_length":4.7,"sepal_width":3.2,"petal_length":1.6,"petal_width":0.2,"class":"Iris-setosa"}
```


ssh.cloud.google.com/v2/ssh/projects/gsoc-wav2vec2/zones/europe-west4-a/instances/me18b182-lab7-vm?authuser=0&hl=en_US&projectNumber=290596474871&useAdminProxy=true&troubleshoot4005E...
SSH-IN-BROWSER
Terminal

```
{"sepal_length":5.7,"sepal_width":2.6,"petal_length":3.5,"petal_width":1.0,"class":"Iris-versicolor"}
{"sepal_length":6.0,"sepal_width":2.2,"petal_length":4.0,"petal_width":1.0,"class":"Iris-versicolor"}
{"sepal_length":5.8,"sepal_width":2.6,"petal_length":4.0,"petal_width":1.2,"class":"Iris-versicolor"}
{"sepal_length":5.5,"sepal_width":2.3,"petal_length":4.0,"petal_width":1.3,"class":"Iris-versicolor"}
{"sepal_length":6.1,"sepal_width":2.8,"petal_length":4.0,"petal_width":1.3,"class":"Iris-versicolor"}
{"sepal_length":5.5,"sepal_width":2.5,"petal_length":4.0,"petal_width":1.3,"class":"Iris-versicolor"}
{"sepal_length":6.4,"sepal_width":3.2,"petal_length":4.5,"petal_width":1.5,"class":"Iris-versicolor"}
{"sepal_length":5.6,"sepal_width":3.0,"petal_length":4.5,"petal_width":1.5,"class":"Iris-versicolor"}
{"sepal_length":6.2,"sepal_width":2.2,"petal_length":4.5,"petal_width":1.5,"class":"Iris-versicolor"}
{"sepal_length":6.0,"sepal_width":2.9,"petal_length":4.5,"petal_width":1.5,"class":"Iris-versicolor"}
{"sepal_length":5.4,"sepal_width":3.0,"petal_length":4.5,"petal_width":1.5,"class":"Iris-versicolor"}
{"sepal_length":6.0,"sepal_width":3.4,"petal_length":4.5,"petal_width":1.6,"class":"Iris-versicolor"}
{"sepal_length":5.7,"sepal_width":2.8,"petal_length":4.5,"petal_width":1.3,"class":"Iris-versicolor"}
{"sepal_length":6.7,"sepal_width":3.0,"petal_length":5.0,"petal_width":1.7,"class":"Iris-versicolor"}
{"sepal_length":5.8,"sepal_width":2.7,"petal_length":3.9,"petal_width":1.2,"class":"Iris-versicolor"}
{"sepal_length":5.2,"sepal_width":2.7,"petal_length":3.9,"petal_width":1.4,"class":"Iris-versicolor"}
{"sepal_length":5.6,"sepal_width":2.5,"petal_length":3.9,"petal_width":1.1,"class":"Iris-versicolor"}
{"sepal_length":6.4,"sepal_width":2.9,"petal_length":4.3,"petal_width":1.3,"class":"Iris-versicolor"}
{"sepal_length":6.2,"sepal_width":2.9,"petal_length":4.3,"petal_width":1.3,"class":"Iris-versicolor"}
{"sepal_length":6.8,"sepal_width":2.8,"petal_length":4.8,"petal_width":1.4,"class":"Iris-versicolor"}
{"sepal_length":5.9,"sepal_width":3.2,"petal_length":4.8,"petal_width":1.8,"class":"Iris-versicolor"}
{"sepal_length":4.9,"sepal_width":2.4,"petal_length":3.3,"petal_width":1.0,"class":"Iris-versicolor"}
{"sepal_length":5.0,"sepal_width":2.3,"petal_length":3.3,"petal_width":1.0,"class":"Iris-versicolor"}
{"sepal_length":5.5,"sepal_width":2.4,"petal_length":3.8,"petal_width":1.1,"class":"Iris-versicolor"}
{"sepal_length":5.8,"sepal_width":2.7,"petal_length":4.1,"petal_width":1.0,"class":"Iris-versicolor"}
{"sepal_length":5.6,"sepal_width":3.0,"petal_length":4.1,"petal_width":1.3,"class":"Iris-versicolor"}
{"sepal_length":5.7,"sepal_width":2.8,"petal_length":4.1,"petal_width":1.3,"class":"Iris-versicolor"}
{"sepal_length":6.5,"sepal_width":2.8,"petal_length":4.6,"petal_width":1.5,"class":"Iris-versicolor"}
{"sepal_length":6.1,"sepal_width":3.0,"petal_length":4.6,"petal_width":1.4,"class":"Iris-versicolor"}
{"sepal_length":6.6,"sepal_width":2.9,"petal_length":4.6,"petal_width":1.3,"class":"Iris-versicolor"}
{"sepal_length":6.0,"sepal_width":2.7,"petal_length":5.1,"petal_width":1.6,"class":"Iris-versicolor"}
{"sepal_length":5.5,"sepal_width":2.4,"petal_length":3.7,"petal_width":1.0,"class":"Iris-versicolor"}
{"sepal_length":5.5,"sepal_width":2.6,"petal_length":4.4,"petal_width":1.2,"class":"Iris-versicolor"}
{"sepal_length":6.7,"sepal_width":3.1,"petal_length":4.4,"petal_width":1.4,"class":"Iris-versicolor"}
{"sepal_length":6.6,"sepal_width":3.0,"petal_length":4.4,"petal_width":1.4,"class":"Iris-versicolor"}
{"sepal_length":6.3,"sepal_width":2.3,"petal_length":4.4,"petal_width":1.3,"class":"Iris-versicolor"}
{"sepal_length":6.9,"sepal_width":3.1,"petal_length":4.9,"petal_width":1.5,"class":"Iris-versicolor"}
{"sepal_length":6.3,"sepal_width":2.5,"petal_length":4.9,"petal_width":1.5,"class":"Iris-versicolor"}
{"sepal_length":5.6,"sepal_width":2.9,"petal_length":3.6,"petal_width":1.3,"class":"Iris-versicolor"}
{"sepal_length":5.9,"sepal_width":3.0,"petal_length":4.2,"petal_width":1.5,"class":"Iris-versicolor"}
{"sepal_length":5.7,"sepal_width":3.0,"petal_length":4.2,"petal_width":1.2,"class":"Iris-versicolor"}
{"sepal_length":5.6,"sepal_width":2.7,"petal_length":4.2,"petal_width":1.3,"class":"Iris-versicolor"}
{"sepal_length":5.7,"sepal_width":2.9,"petal_length":4.2,"petal_width":1.3,"class":"Iris-versicolor"}
{"sepal_length":6.7,"sepal_width":3.1,"petal_length":4.7,"petal_width":1.5,"class":"Iris-versicolor"}
{"sepal_length":6.1,"sepal_width":2.8,"petal_length":4.7,"petal_width":1.2,"class":"Iris-versicolor"}
{"sepal_length":7.0,"sepal_width":3.2,"petal_length":4.7,"petal_width":1.4,"class":"Iris-versicolor"}
{"sepal_length":6.1,"sepal_width":2.9,"petal_length":4.7,"petal_width":1.4,"class":"Iris-versicolor"}
{"sepal_length":6.3,"sepal_width":3.3,"petal_length":4.7,"petal_width":1.6,"class":"Iris-versicolor"}
```

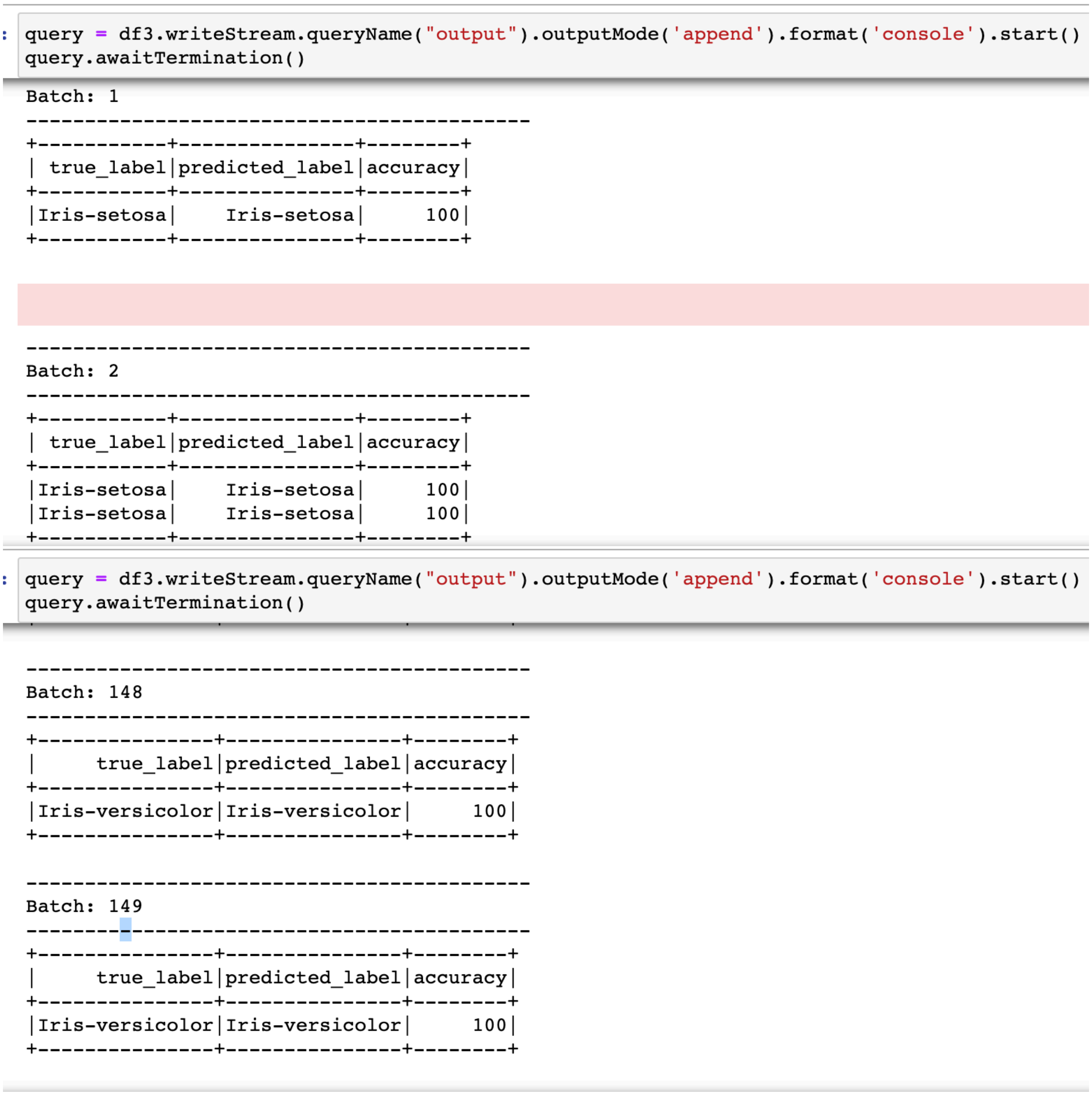
2) In order to make real-time predictions, we saved the pipeline in GCS bucket. The pipeline was saved using .save() method and loaded for real-time predictions using .load() method.

Following is the code for subscriber.py



```python
#!/usr/bin/env python
# coding: utf-8

import os
from itertools import chain

import pyspark.sql.functions as f
from pyspark import SparkContext
from pyspark.ml import PipelineModel
from pyspark.sql import SparkSession
from pyspark.sql.types import FloatType, StringType, StructType

os.environ[
    "PYSPARK_SUBMIT_ARGS"
] = "--packages org.apache.spark:spark-sql-kafka-0-10_2.12:3.1.2 pyspark-shell"

sc = SparkContext()
spark = SparkSession(sc)

BROKERS = "10.164.0.17:9092"
TOPIC = "me18b182-lab7"

df = (
    spark.readStream.format("kafka")
    .option("kafka.bootstrap.servers", BROKERS)
    .option("subscribe", TOPIC)
    .load()
)

schema = (
    StructType()
    .add("sepal_length", FloatType())
    .add("sepal_width", FloatType())
    .add("petal_length", FloatType())
    .add("petal_width", FloatType())
    .add("class", StringType())
)

df = df.select(
    f.from_json(f.decode(df.value, "utf-8"), schema=schema).alias("input")
).select("input.*")

model_path = "gs://big-data-cs4830/lab-5/lab5_model"
model = PipelineModel.load(model_path)

df = df.withColumn("true_label", df["class"])
predictions = model.transform(df)

classes_map = dict(
    zip([0.0, 1.0, 2.0], ["Iris-setosa", "Iris-versicolor", "Iris-virginica"])
)

map_op = f.create_map([f.lit(x) for x in chain(*classes_map.items())])
df = predictions.withColumn("predicted_label", map_op[f.col("prediction")])[
    ["true_label", "prediction", "predicted_label"]
]
df = df.withColumn(
    "accuracy",
    f.when(f.col("predicted_label") == f.col("true_label"), 100).otherwise(0),
)

df = df[["true_label", "predicted_label", "accuracy"]]
df.createOrReplaceTempView("output")

query = (
    df.writeStream.queryName("output").outputMode("append").format("console").start()
)
query.awaitTermination()

```

Following is the screenshot of the outputs when subscriber.py is run



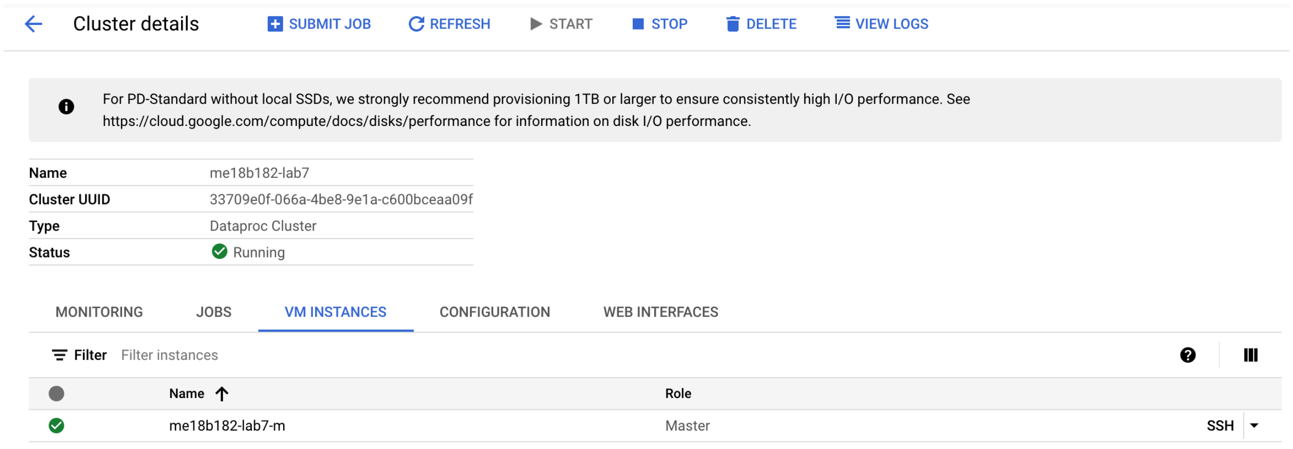
```
query = df3.writeStream.queryName("output").outputMode('append').format('console').start()
query.awaitTermination()
```

```
Batch: 1
-------------------------------------------
+-----------+---------------+--------+
| true_label|predicted_label|accuracy|
+-----------+---------------+--------+
|Iris-setosa|    Iris-setosa|     100|
+-----------+---------------+--------+


-------------------------------------------
Batch: 2
-------------------------------------------
+-----------+---------------+--------+
| true_label|predicted_label|accuracy|
+-----------+---------------+--------+
|Iris-setosa|    Iris-setosa|     100|
|Iris-setosa|    Iris-setosa|     100|
+-----------+---------------+--------+
```

```
query = df3.writeStream.queryName("output").outputMode('append').format('console').start()
query.awaitTermination()
```

```
-------------------------------------------
Batch: 148
-------------------------------------------
+---------------+---------------+--------+
|     true_label|predicted_label|accuracy|
+---------------+---------------+--------+
|Iris-versicolor|Iris-versicolor|     100|
+---------------+---------------+--------+

-------------------------------------------
Batch: 149
-------------------------------------------
+---------------+---------------+--------+
|     true_label|predicted_label|accuracy|
+---------------+---------------+--------+
|Iris-versicolor|Iris-versicolor|     100|
+---------------+---------------+--------+
```

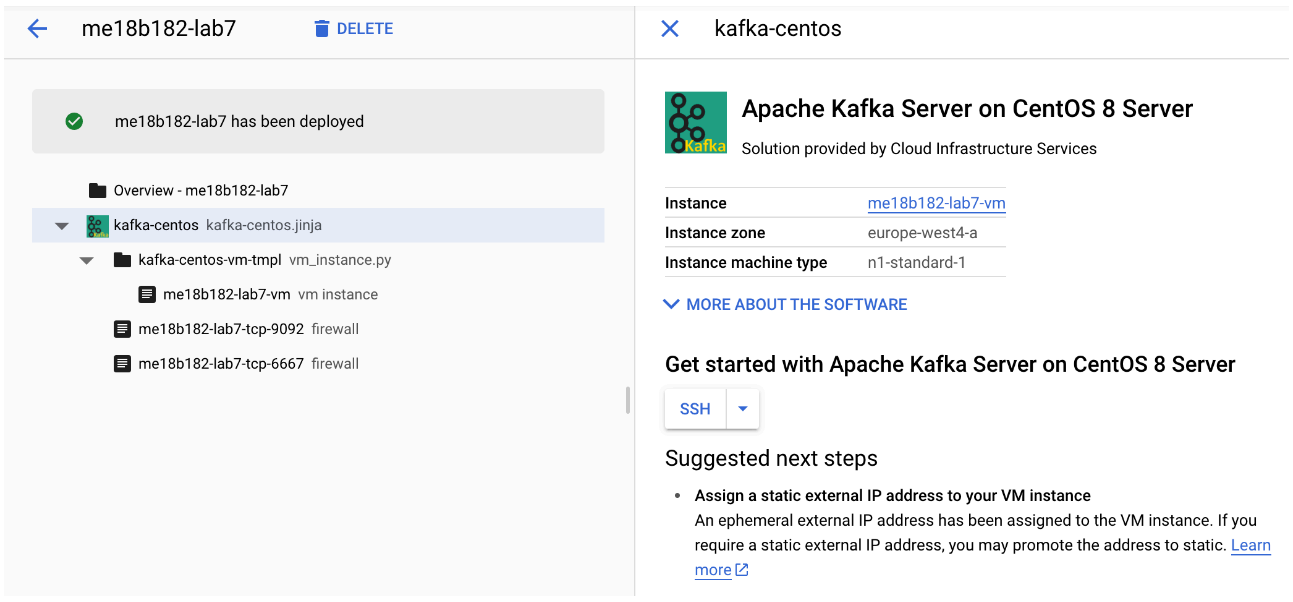
Screenshot of kafka and dataproc can also be found below:



For PD-Standard without local SSDs, we strongly recommend provisioning 1TB or larger to ensure consistently high I/O performance. See https://cloud.google.com/compute/docs/disks/performance for information on disk I/O performance.

Name	me18b182-lab7
Cluster UUID	33709e0f-066a-4be8-9e1a-c600bceaa09f
Type	Dataproc Cluster
Status	Running

	Name	Role	
✓	me18b182-lab7-m	Master	SSH



me18b182-lab7 | DELETE

me18b182-lab7 has been deployed

- Overview - me18b182-lab7
- kafka-centos kafka-centos.jinja
  - kafka-centos-vm-tmpl vm_instance.py
    - me18b182-lab7-vm vm instance
  - me18b182-lab7-tcp-9092 firewall
  - me18b182-lab7-tcp-6667 firewall

kafka-centos

Apache Kafka Server on CentOS 8 Server

Solution provided by Cloud Infrastructure Services

Instance	me18b182-lab7-vm
Instance zone	europe-west4-a
Instance machine type	n1-standard-1

MORE ABOUT THE SOFTWARE

Get started with Apache Kafka Server on CentOS 8 Server

SSH

Suggested next steps

- **Assign a static external IP address to your VM instance**
  An ephemeral external IP address has been assigned to the VM instance. If you require a static external IP address, you may promote the address to static. Learn more