Assignment 9.1

By Kurt Stoneburner

https://sparkbyexamples.com/pyspark-tutorial/ (https://sparkbyexamples.com/pyspark-tutorial/)

watermarks are discussed here: https://spark.apache.org/docs/latest/structured-streaming-programming-guide.html (https://spark.apache.org/docs/latest/structured-streaming-programming-guide.html)

https://sparkbyexamples.com/pyspark/pyspark-exception-java-gateway-process-exited-before-sending-the-driver-its-port-number/ (https://sparkbyexamples.com/pyspark/pyspark-exception-java-gateway-process-exited-before-sending-the-driver-its-port-number/)

https://quabr.com/58723314/pyspark-failed-to-find-data-source-kafka (https://quabr.com/58723314/pyspark-failed-to-find-data-source-kafka)

https://search.maven.org/search?q=g:org.apache.spark%20AND%20a:spark-streaming-kafka-0-8-assembly_2.11 (https://search.maven.org/search?q=g:org.apache.spark%20AND%20a:spark-streaming-kafka-0-8-assembly_2.11)

https://www.rittmanmead.com/blog/2017/01/getting-started-with-spark-streaming-with-python-and-kafka/ (https://www.rittmanmead.com/blog/2017/01/getting-started-with-spark-streaming-with-python-and-kafka/)

https://www.analyticsvidhya.com/blog/2021/06/setting-up-real-time-structured-streaming-with-spark-and-kafka-on-windows-os/ (https://www.analyticsvidhya.com/blog/2021/06/setting-up-real-time-structured-streaming-with-spark-and-kafka-on-windows-os/)

https://spark.apache.org/docs/latest//api/python/reference/api/pyspark.SparkConf.html (https://spark.apache.org/docs/latest//api/python/reference/api/pyspark.SparkConf.html)

https://stackoverflow.com/questions/54227744/pyspark-2-x-programmatically-adding-maven-jar-coordinates-to-spark?__cpo=aHR0cHM6Ly9zdGFja292ZXJmbG93LmNvbQ (https://stackoverflow.com/questions/54227744/pyspark-2-x-programmatically-adding-maven-jar-coordinates-to-spark?__cpo=aHR0cHM6Ly9zdGFja292ZXJmbG93LmNvbQ)

https://duckduckgo.com/?t=ffab&q=SparkSession+readstream+Failed+to+find+data+source%3A+kafka&ia=web (https://duckduckgo.com/?t=ffab&q=SparkSession+readstream+Failed+to+find+data+source%3A+kafka&ia=web)

reintsall pyspark: https://sparkbyexamples.com/pyspark-tutorial/#pyspark-installation (https://sparkbyexamples.com/pyspark-tutorial/#pyspark-installation)

conda install -c conda-forge pyspark

https://kontext.tech/column/spark/298/get-the-current-spark-context-settingsconfigurations

(https://kontext.tech/column/spark/298/get-the-current-spark-context-settingsconfigurations)

Find Spark add_packages resolved the local Spark Issue. https://github.com/minrk/findspark/pull/11 (https://github.com/minrk/findspark/pull/11)

In [ ]:
```
```

In [1]:
```
import os
import time
import shutil
import json
from pathlib import Path

import pandas as pd

from kafka import KafkaProducer, KafkaAdminClient
from kafka.admin.new_topic import NewTopic
from kafka.errors import TopicAlreadyExistsError

from pyspark.sql import SparkSession
from pyspark.streaming import StreamingContext
from pyspark import SparkConf
from pyspark.sql.functions import window, from_json, col
from pyspark.sql.types import StringType, TimestampType, DoubleType,
from pyspark.sql.functions import udf
from IPython.display import clear_output
import findspark
findspark.init()


current_dir = Path(os.getcwd()).absolute()
checkpoint_dir = current_dir.joinpath('checkpoints')
locations_checkpoint_dir = checkpoint_dir.joinpath('locations')
accelerations_checkpoint_dir = checkpoint_dir.joinpath('accelerations

if locations_checkpoint_dir.exists():
    shutil.rmtree(locations_checkpoint_dir)

if accelerations_checkpoint_dir.exists():
    shutil.rmtree(accelerations_checkpoint_dir)

locations_checkpoint_dir.mkdir(parents=True, exist_ok=True)
```

Configuration Parameters

TODO: Change the configuration prameters to the appropriate values for your setup.

In [2]:
```
config = dict(
    bootstrap_servers=['kafka.kafka.svc.cluster.local:9092'],
    first_name='Kurt',
```

```
    last_name='Stoneburner'
)

config['client_id'] = '{}{}'.format(
    config['last_name'],
    config['first_name']
)
config['topic_prefix'] = '{}{}'.format(
    config['last_name'],
    config['first_name']
)

config['locations_topic'] = '{}-locations'.format(config['topic_prefi
config['accelerations_topic'] = '{}-accelerations'.format(config['top
config['simple_topic'] = '{}-simple'.format(config['topic_prefix'])

```

Out[2]:
```
{'bootstrap_servers': ['kafka.kafka.svc.cluster.local:9092'],
 'first_name': 'Kurt',
 'last_name': 'Stoneburner',
 'client_id': 'StoneburnerKurt',
 'topic_prefix': 'StoneburnerKurt',
 'locations_topic': 'StoneburnerKurt-locations',
 'accelerations_topic': 'StoneburnerKurt-accelerations',
 'simple_topic': 'StoneburnerKurt-simple'}
```

Create Topic Utility Function

The `create_kafka_topic` helps create a Kafka topic based on your configuration settings. For instance, if your first name is *John* and your last name is *Doe*, `create_kafka_topic('locations')` will create a topic with the name `DoeJohn-locations`. The function will not create the topic if it already exists.

In [3]:
```
def create_kafka_topic(topic_name, config=config, num_partitions=1, r
    bootstrap_servers = config['bootstrap_servers']
    client_id = config['client_id']
    topic_prefix = config['topic_prefix']
    name = '{}-{}'.format(topic_prefix, topic_name)

    admin_client = KafkaAdminClient(
        bootstrap_servers=bootstrap_servers,
        client_id=client_id
    )

    topic = NewTopic(
        name=name,
        num_partitions=num_partitions,
        replication_factor=replication_factor
    )

    topic_list = [topic]
    try:
        admin_client.create_topics(new_topics=topic_list)
        print('Created topic "{}"'.format(name))
```

```
    except TopicAlreadyExistsError as e:
        print('Topic "{}" already exists'.format(name))
for topic in ['locations','accelerations']:
```

```
Topic "StoneburnerKurt-locations" already exists
Topic "StoneburnerKurt-accelerations" already exists
```

In [4]:

```

#//*** Close Spark if already running. Guarantees Spark is loaded wit
#//*** Prevents some Ipython/Notebook related issues.

spark = SparkSession.builder\
        .appName("Assignment09")\
        .getOrCreate()

df_locations = spark\
    .readStream.format("kafka")\
    .option("kafka.bootstrap.servers", config['bootstrap_servers'][0]
    .option("subscribe", config['locations_topic'])\
    .load()

```

TODO: Create a data frame called `df_accelerations` that reads from the accelerations topic you published to in assignment 8. In order to read data from this topic, make sure that you are running the notebook you created in assignment 8 that publishes acceleration and location data to the `LastnameFirstname-simple` topic.

In [5]:

```
df_accelerations = spark\
    .readStream.format("kafka")\
    .option("kafka.bootstrap.servers", config['bootstrap_servers'][0]
    .option("subscribe", config['accelerations_topic'])\
    .load()
```

TODO: Create two streaming queries, `ds_locations` and `ds_accelerations` that publish to the `LastnameFirstname-simple` topic. See http://spark.apache.org/docs/latest/structured-streaming-programming-guide.html#starting-streaming-queries (http://spark.apache.org/docs/latest/structured-streaming-programming-guide.html#starting-streaming-queries) and http://spark.apache.org/docs/latest/structured-streaming-kafka-integration.html (http://spark.apache.org/docs/latest/structured-streaming-kafka-integration.html) for more information.

In [ ]:

```
ds_locations = df_locations.writeStream \
.format("kafka") \
.option("checkpointLocation", locations_checkpoint_dir) \
.option("kafka.bootstrap.servers",  config['bootstrap_servers'][0]) \
.option("topic", config['simple_topic']) \
.start()


ds_accelerations = df_locations.writeStream \
.format("kafka") \
.option("checkpointLocation", locations_checkpoint_dir) \
```

```
12 .option("kafka.bootstrap.servers",  config['bootstrap_servers'][0]) \
13 .option("topic", config['simple_topic']) \
14 .start()
15 try:
16     ds_locations.awaitTermination()
17     ds_accelerations.awaitTermination()
18 except KeyboardInterrupt:
```

In [ ]: