Example Kafka Producer

May 15, 2022

[1]:

```
import json
import uuid
import os
import sys
import json
from pathlib import Path
from kafka import KafkaProducer, KafkaAdminClient
from kafka.admin.new_topic import NewTopic
from kafka.errors import TopicAlreadyExistsError
import heapq
import pandas as pd
import threading
```

[2]:

```
current_dir=Path('/home/jovyan/dsc650/dsc650-master/data/processed/bdd/
 ↪locations')
file_path=Path('/home/jovyan/dsc650/dsc650-master/data/processed/bdd/locations/
 ↪t=000.0/19b9aa10588646b3bf22c9b4865a7995.parquet')
```

[3]:

```
x=file_path
test=pd.read_parquet(x)
```

[4]:

```
dir_list=os.listdir(current_dir)

clean_dir_list=[]
offset_values=[]
for file in dir_list:
    file = float(file[2::])
    clean_dir_list.append(file)


clean_dir_list.sort()
```

[ ]:

```

```

0.0.1 Configuration Parameters

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

[5]:

```
config = dict(
    bootstrap_servers=['kafka.kafka.svc.cluster.local:9092'],
    first_name='Harsimar',
    last_name='Mangat'
)

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

config
```

[5]:

```
{'bootstrap_servers': ['kafka.kafka.svc.cluster.local:9092'],
 'first_name': 'Harsimar',
 'last_name': 'Mangat',
 'client_id': 'MangatHarsimar',
 'topic_prefix': 'MangatHarsimar'}
```

0.0.2 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.

[6]:

```
def create_kafka_topic(topic_name, config=config, num_partitions=1,␣
 ↪replication_factor=1):
    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))

create_kafka_topic('locations')
```

```
Topic "MangatHarsimar-locations" already exists
```

0.0.3 Kafka Producer

The following code creates a `KafkaProducer` object which you can use to send Python objects that are serialized as JSON.

Note: This producer serializes Python objects as JSON. This means that object must be JSON serializable. As an example, Python `DateTime` values are not JSON serializable and must be converted to a string (e.g. ISO 8601) or a numeric value (e.g. a Unix timestamp) before being sent.

[7]:
```
producer = KafkaProducer(
  bootstrap_servers=config['bootstrap_servers'],
  value_serializer=lambda x: json.dumps(x).encode('utf-8')
)
```

0.0.4 Send Data Function

The `send_data` function sends a Python object to a Kafka topic. This function adds the `topic_prefix` to the topic so `send_data('locations', data)` sends a JSON serialized message to `DoeJohn-locations`. The function also registers callbacks to let you know if the message has been sent or if an error has occured.

[8]:
```
def on_send_success(record_metadata):
    print('Message sent:\n    Topic: "{}"\n    Partition: {}\n    Offset: {}'.
 ↪format(
        record_metadata.topic,
        record_metadata.partition,
        record_metadata.offset
    ))

def on_send_error(excp):
    print('I am an errback', exc_info=excp)
    # handle exception

def send_data(topic, data, config=config, producer=producer, msg_key=None):
    topic_prefix = config['topic_prefix']
    topic_name = '{}-{}'.format(topic_prefix, topic)
    if msg_key is not None:
        key = msg_key
    else:
        key = uuid.uuid4().hex

    producer.send(
        topic_name,
        value=data,
        key=key.encode('utf-8')
    ).add_callback(on_send_success).add_errback(on_send_error)
```

[9]:

```
def print_test(item):
    location_dir = current_dir.joinpath('t=' + str(item).zfill(5))
    for file in os.listdir(location_dir):
        file_path=location_dir.joinpath(file)
        f1=pd.read_parquet(file_path)
        for index,row in f1.iterrows():
            #print(row.to_dict())
            row=row.to_dict()
            row['timestamp']=row['timestamp'].isoformat()
            send_data('locations',row)


for item in clean_dir_list:
    timer=threading.Timer(item,print_test,[item])
    b=threading.Barrier(1)
    timer.start()
    b.wait()
```

```
Message sent:
    Topic: "MangatHarsimar-locations"
    Partition: 0
    Offset: 478
Message sent:
    Topic: "MangatHarsimar-locations"
    Partition: 0
    Offset: 479
Message sent:
    Topic: "MangatHarsimar-locations"
    Partition: 0
    Offset: 480
Message sent:
    Topic: "MangatHarsimar-locations"
    Partition: 0
    Offset: 481
Message sent:
    Topic: "MangatHarsimar-locations"
    Partition: 0
    Offset: 482
Message sent:
    Topic: "MangatHarsimar-locations"
    Partition: 0
    Offset: 483
Message sent:
    Topic: "MangatHarsimar-locations"
    Partition: 0
    Offset: 484
Message sent:
    Topic: "MangatHarsimar-locations"
    Partition: 0
    Offset: 485
Message sent:
    Topic: "MangatHarsimar-locations"
    Partition: 0
    Offset: 486
Message sent:
    Topic: "MangatHarsimar-locations"
    Partition: 0
    Offset: 487
Message sent:
    Topic: "MangatHarsimar-locations"
    Partition: 0
    Offset: 488
Message sent:
    Topic: "MangatHarsimar-locations"
    Partition: 0
    Offset: 489
Message sent:
    Topic: "MangatHarsimar-locations"
    Partition: 0
    Offset: 490
Message sent:
    Topic: "MangatHarsimar-locations"
    Partition: 0
    Offset: 491
Message sent:
    Topic: "MangatHarsimar-locations"
    Partition: 0
    Offset: 492
Message sent:
    Topic: "MangatHarsimar-locations"
    Partition: 0
    Offset: 493
Message sent:
    Topic: "MangatHarsimar-locations"
    Partition: 0
    Offset: 494
Message sent:
    Topic: "MangatHarsimar-locations"
    Partition: 0
    Offset: 495
```

example_data = dict(key1=‘value1’, key2=‘value2’)

send_data(‘locations’, example_data)

[ ]:

[ ]:

[ ]:

[ ]:

[ ]: