

## 5. Spark Streaming. Stateful streams

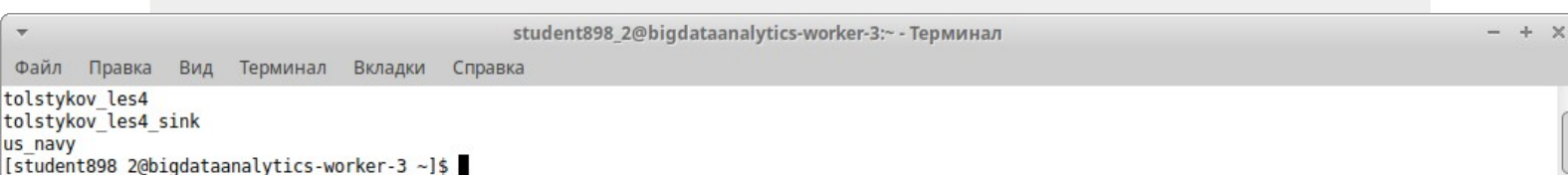
Загрузить в топик kafka свои данные, прочитать их в потоке, применить watermark и window. Повторить шаги выполненные на занятии. Дополнительно, объединить статичный и динамичный потоки. Задание на повышенный бал: Написать скрипт на python для конвертации файла csv в json.

Подключаемся к серверу

```
ssh -i ~/.ssh/id_rsa_student898_2 student898\_2@37.139.41.176
```

смотрим лист топиков

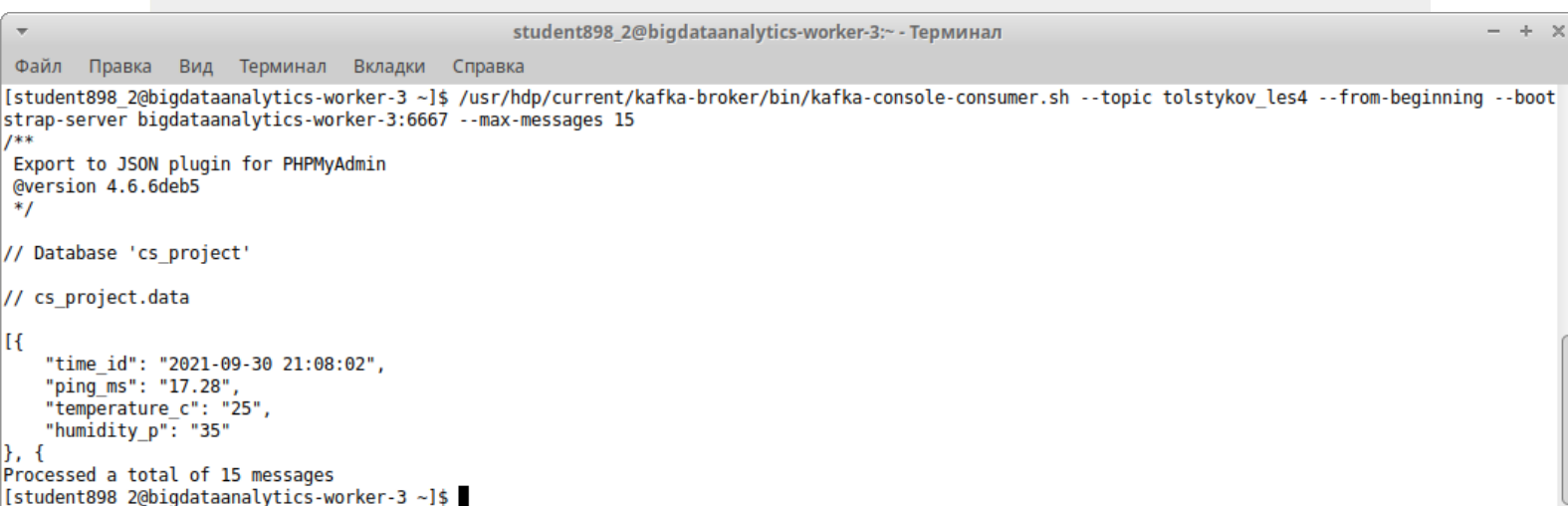
```
/usr/hdp/current/kafka-broker/bin/kafka-topics.sh --zookeeper bigdataanalytics-worker-3:2181 --list
```



```
student898_2@bigdataanalytics-worker-3:~ - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка
tolstykov_les4
tolstykov_les4_sink
us_navy
[student898_2@bigdataanalytics-worker-3 ~]$
```

Прочитать топик tolstykov\_les4

```
/usr/hdp/current/kafka-broker/bin/kafka-console-consumer.sh --topic tolstykov_les4 --from-beginning --bootstrap-server bigdataanalytics-worker-3:6667 --max-messages 15
```



```
student898_2@bigdataanalytics-worker-3:~ - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка
[student898_2@bigdataanalytics-worker-3 ~]$ /usr/hdp/current/kafka-broker/bin/kafka-console-consumer.sh --topic tolstykov_les4 --from-beginning --bootstrap-server bigdataanalytics-worker-3:6667 --max-messages 15
/**
Export to JSON plugin for PHPMysqlAdmin
@version 4.6.6deb5
*/
// Database 'cs_project'
// cs_project.data
[{"time_id": "2021-09-30 21:08:02", "ping_ms": "17.28", "temperature_c": "25", "humidity_p": "35"}, {"time_id": "2021-09-30 21:08:03", "ping_ms": "17.28", "temperature_c": "25", "humidity_p": "35"}]
Processed a total of 15 messages
[student898_2@bigdataanalytics-worker-3 ~]$
```

Запускаем `pyspark`

```
export SPARK_KAFKA_VERSION=0.10
```

```
/opt/spark-2.4.8/bin/pyspark --packages org.apache.spark:spark-sql-kafka-0-10_2.11:2.4.5 --driver-memory 512m --master local[1]
```



```
def console_output(df, freq):

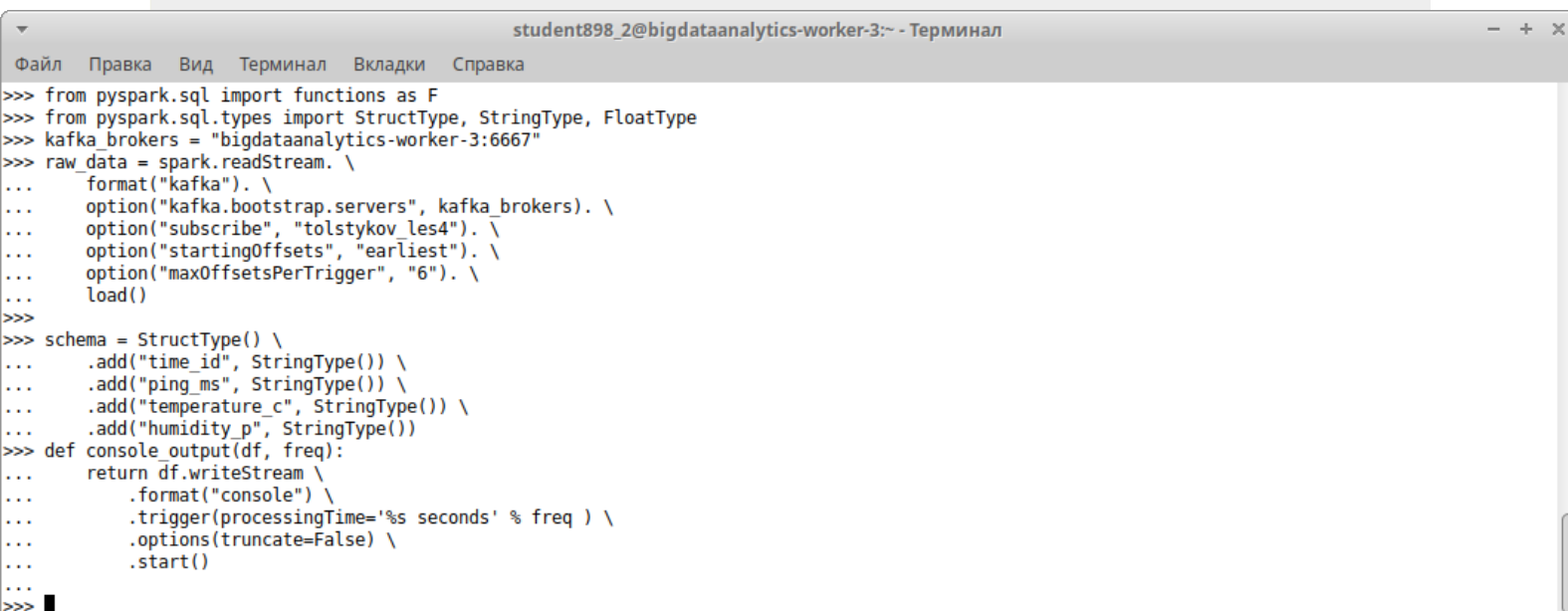
    return df.writeStream \

        .format("console") \

        .trigger(processingTime='%s seconds' % freq ) \

        .options(truncate=False) \

        .start()
```



```
student898_2@bigdataanalytics-worker-3:~ - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка
>>> from pyspark.sql import functions as F
>>> from pyspark.sql.types import StructType, StringType, FloatType
>>> kafka_brokers = "bigdataanalytics-worker-3:6667"
>>> raw_data = spark.readStream. \
...     format("kafka"). \
...     option("kafka.bootstrap.servers", kafka_brokers). \
...     option("subscribe", "tolstikov_les4"). \
...     option("startingOffsets", "earliest"). \
...     option("maxOffsetsPerTrigger", "6"). \
...     load()
>>> schema = StructType() \
...     .add("time_id", StringType()) \
...     .add("ping_ms", StringType()) \
...     .add("temperature_c", StringType()) \
...     .add("humidity_p", StringType())
>>> def console_output(df, freq):
...     return df.writeStream \
...         .format("console") \
...         .trigger(processingTime='%s seconds' % freq ) \
...         .options(truncate=False) \
...         .start()
>>>
```

Сделаем преобразование в плоскую структуру

```
parsed_data = raw_data \

    .select(F.from_json(F.col("value").cast("String"), schema).alias("value"), "offset") \

    .select("value.*", "offset")

out = console_output(raw_data, 10)

out.stop()
```

```
student898_2@bigdataanalytics-worker-3:~ - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка
+-----+
[null][20 20 20 20 22 68 75 6D 69 64 69 74 79 5F 70 22 3A 20 22 34 30 22] |tolstikov_les4|0 |18
|2022-01-17 19:08:37.245|0 |
[null][7D 2C 20 7B] |tolstikov_les4|0 |19
|2022-01-17 19:08:37.245|0 |
[null][20 20 20 20 22 74 69 6D 65 5F 69 64 22 3A 20 22 32 30 32 31 2D 30 39 2D 33 30 20 32 31 3A 31 30 3A 30 31 22 2C]|tolstikov_les4|0 |20
|2022-01-17 19:08:37.245|0 |
[null][20 20 20 20 22 70 69 6E 67 5F 6D 73 22 3A 20 22 31 38 2E 35 39 22 2C] |tolstikov_les4|0 |21
|2022-01-17 19:08:37.245|0 |
[null][20 20 20 20 22 74 65 6D 70 65 72 61 74 75 72 65 5F 63 22 3A 20 22 32 32 22 2C] |tolstikov_les4|0 |22
|2022-01-17 19:08:37.245|0 |
[null][20 20 20 20 22 68 75 6D 69 64 69 74 79 5F 70 22 3A 20 22 34 31 22] |tolstikov_les4|0 |23
|2022-01-17 19:08:37.245|0 |
+-----+
>>> -----
Batch: 4
+-----+
+-----+
+-----+
|key|value|topic|partition|offset|
|timestamp|timestampType|
+-----+
[null][7D 2C 20 7B] |tolstikov_les4|0 |24
|2022-01-17 19:08:37.245|0 |
[null][20 20 20 20 22 74 69 6D 65 5F 69 64 22 3A 20 22 32 30 32 31 2D 30 39 2D 33 30 20 32 31 3A 31 32 3A 30 32 22 2C]|tolstikov_les4|0 |25
|2022-01-17 19:08:37.245|0 |
[null][20 20 20 20 22 70 69 6E 67 5F 6D 73 22 3A 20 22 31 36 2E 37 33 22 2C] |tolstikov_les4|0 |26
|2022-01-17 19:08:37.245|0 |
[null][20 20 20 20 22 74 65 6D 70 65 72 61 74 75 72 65 5F 63 22 3A 20 22 32 32 22 2C] |tolstikov_les4|0 |27
|2022-01-17 19:08:37.245|0 |
[null][20 20 20 20 22 68 75 6D 69 64 69 74 79 5F 70 22 3A 20 22 34 32 22] |tolstikov_les4|0 |28
|2022-01-17 19:08:37.245|0 |
[null][7D 2C 20 7B] |tolstikov_les4|0 |29
|2022-01-17 19:08:37.245|0 |
+-----+
out.stop()
>>> out.stop()
>>>
```

```
extended_data = raw_data \

    .select(F.from_json(F.col("value").cast("String"), schema).alias("value"),
"offset") \

    .select("value.*", "offset") \

    .withColumn("receive_time", F.current_timestamp())

extended_data.printSchema()
```

```
student898_2@bigdataanalytics-worker-3:~ - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка
>>> extended_data = raw_data \
...     .select(F.from_json(F.col("value").cast("String"), schema).alias("value"), "offset") \
...     .select("value.*", "offset") \
...     .withColumn("receive_time", F.current_timestamp())
>>> extended_data.printSchema()
root
|-- time_id: string (nullable = true)
|-- ping_ms: string (nullable = true)
|-- temperature_c: string (nullable = true)
|-- humidity_p: string (nullable = true)
|-- offset: long (nullable = true)
|-- receive_time: timestamp (nullable = false)
>>>
```

```
def console_output(df, freq):

    return df.writeStream \

        .format("console") \

        .trigger(processingTime='%s seconds' % freq ) \

        .option("checkpointLocation", "checkpoints/duplicates_console_chk") \

        .options(truncate=False) \

        .start()
```

```
student898_2@bigdataanalytics-worker-3:~ - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка

>>> def console_output(df, freq):
...     return df.writeStream \
...         .format("console") \
...         .trigger(processingTime='%s seconds' % freq ) \
...         .option("checkpointLocation", "checkpoints/duplicates_console_chk") \
...         .options(truncate=False) \
...         .start()
... 
```

Подключаемся в другом окне, удаляем **checkpoints**

```
hdfs dfs -rm -f -r checkpoints
```

```
hdfs dfs -ls
```

```
student898_2@bigdataanalytics-worker-3:~ - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка

igore@igor-MS-7808:~$ ssh -i ~/.ssh/id_rsa_student898_2 student898_2@37.139.41.176
Last login: Fri Jan 21 20:01:40 2022 from 109-252-19-10.nat.spd-mgts.ru
[student898_2@bigdataanalytics-worker-3 ~]$ hdfs dfs -rm -f -r checkpoints
22/01/21 20:30:45 INFO fs.TrashPolicyDefault: Moved: 'hdfs://bigdataanalytics-head-0.mcs.local:8020/user/student898_2/checkpoints' to trash at: hdfs://
/bigdataanalytics-head-0.mcs.local:8020/user/student898_2/.Trash/Current/user/student898_2/checkpoints1642797045160
[student898_2@bigdataanalytics-worker-3 ~]$ hdfs dfs -ls
Found 7 items
drwx----- - student898_2 student898_2      0 2022-01-21 18:12 .Trash
drwxr-xr-x - student898_2 student898_2      0 2022-01-20 19:25 .sparkStaging
drwxr-xr-x - student898_2 student898_2      0 2021-12-15 22:13 for_stream
drwxr-xr-x - student898_2 student898_2      0 2022-01-17 13:39 input_csv_for_stream
drwxr-xr-x - student898_2 student898_2      0 2022-01-17 20:24 my_parquet_sink
drwxr-xr-x - student898_2 student898_2      0 2022-01-17 20:20 tolstykov_les4_file_checkpoint
drwxr-xr-x - student898_2 student898_2      0 2022-01-17 20:49 tolstykov_les4_kafka_checkpoint
[student898_2@bigdataanalytics-worker-3 ~]$ 
```

В первом окне запускаем

```
stream = console_output(extended_data , 5)
```

```
stream.stop()
```

```
student898_2@bigdataanalytics-worker-3:~ - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка

Batch: 13
-----
+-----+-----+-----+-----+-----+
|time_id|ping_ms|temperature_c|humidity_p|offset|receive_time|
+-----+-----+-----+-----+-----+
|null|null|null|null|78|2022-01-21 20:34:20.003|
|null|null|null|null|79|2022-01-21 20:34:20.003|
|null|null|null|null|80|2022-01-21 20:34:20.003|
|null|null|null|null|81|2022-01-21 20:34:20.003|
|null|null|null|null|82|2022-01-21 20:34:20.003|
|null|null|null|null|83|2022-01-21 20:34:20.003|
+-----+-----+-----+-----+-----+

Batch: 14
-----
+-----+-----+-----+-----+-----+
|time_id|ping_ms|temperature_c|humidity_p|offset|receive_time|
+-----+-----+-----+-----+-----+
|null|null|null|null|84|2022-01-21 20:34:25.004|
|null|null|null|null|85|2022-01-21 20:34:25.004|
|null|null|null|null|86|2022-01-21 20:34:25.004|
|null|null|null|null|87|2022-01-21 20:34:25.004|
|null|null|null|null|88|2022-01-21 20:34:25.004|
|null|null|null|null|89|2022-01-21 20:34:25.004|
+-----+-----+-----+-----+-----+

Batch: 15
-----
+-----+-----+-----+-----+-----+
|time_id|ping_ms|temperature_c|humidity_p|offset|receive_time|
+-----+-----+-----+-----+-----+
|null|null|null|null|90|2022-01-21 20:34:30.003|
|null|null|null|null|91|2022-01-21 20:34:30.003|
|null|null|null|null|92|2022-01-21 20:34:30.003|
|null|null|null|null|93|2022-01-21 20:34:30.003|
|null|null|null|null|94|2022-01-21 20:34:30.003|
|null|null|null|null|95|2022-01-21 20:34:30.003|
+-----+-----+-----+-----+-----+

stream.stop()
>>> stream.stop()
>>>
```

Во втором окне наблюдаем

```
hdfs dfs -du -h checkpoints/duplicates_console_chk
```

```
student898_2@bigdataanalytics-worker-3:~ - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка

174  348  checkpoints/duplicates_console_chk/commits
45   90   checkpoints/duplicates_console_chk/metadata
2.5 K  5.1 K checkpoints/duplicates_console_chk/offsets
30    60   checkpoints/duplicates_console_chk/sources
[student898_2@bigdataanalytics-worker-3 ~]$ hdfs dfs -du -h checkpoints/duplicates_console_chk
232  464  checkpoints/duplicates_console_chk/commits
45   90   checkpoints/duplicates_console_chk/metadata
3.4 K  6.8 K checkpoints/duplicates_console_chk/offsets
30    60   checkpoints/duplicates_console_chk/sources
[student898_2@bigdataanalytics-worker-3 ~]$ hdfs dfs -du -h checkpoints/duplicates_console_chk
348  696  checkpoints/duplicates_console_chk/commits
45   90   checkpoints/duplicates_console_chk/metadata
5.1 K  10.2 K checkpoints/duplicates_console_chk/offsets
30    60   checkpoints/duplicates_console_chk/sources
[student898_2@bigdataanalytics-worker-3 ~]$
```

Данные не читаются

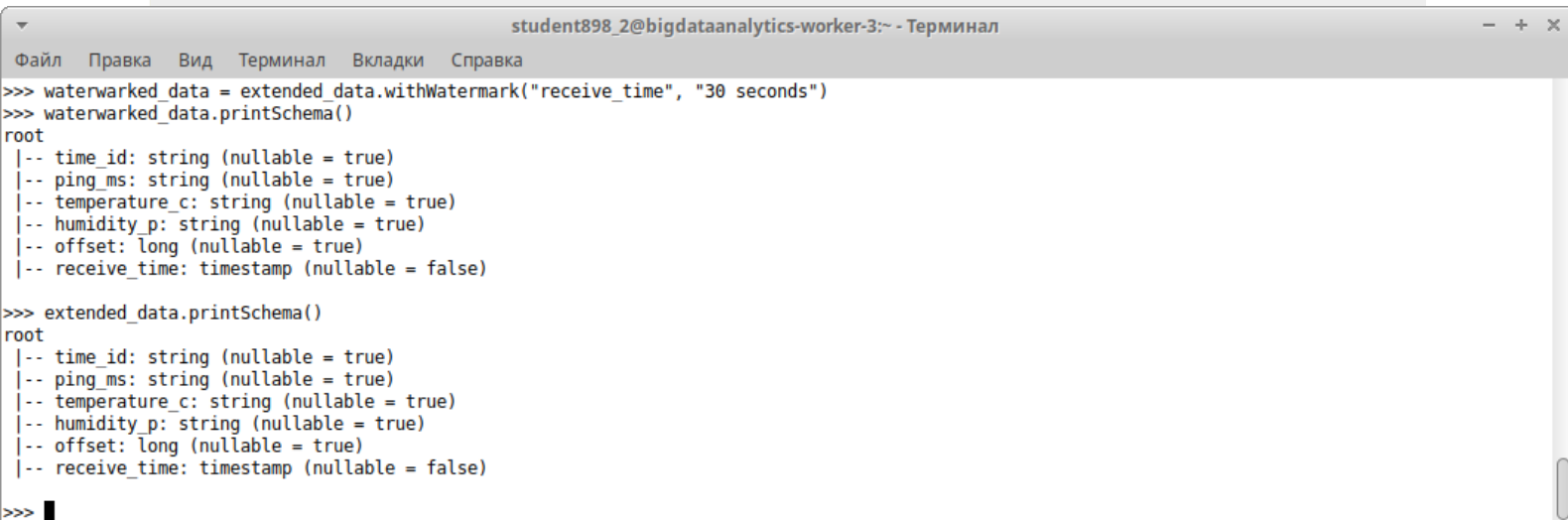
Задаём воте́рмарку, которая должна очищать чекпоинт. Первый параметр - название колонки, на которую смотрит воте́рмарка, второй параметр - гарантированное время

жизни информации о сообщении в чекпойнте. Именно для этого мы добавляли столбец `receive\_time`.

```
waterwarked_data = extended_data.withWatermark("receive_time", "30 seconds")
```

```
waterwarked_data.printSchema()
```

```
extended_data.printSchema()
```



```
student898_2@bigdataanalytics-worker-3:~ - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка
>>> waterwarked_data = extended_data.withWatermark("receive_time", "30 seconds")
>>> waterwarked_data.printSchema()
root
|-- time_id: string (nullable = true)
|-- ping_ms: string (nullable = true)
|-- temperature_c: string (nullable = true)
|-- humidity_p: string (nullable = true)
|-- offset: long (nullable = true)
|-- receive_time: timestamp (nullable = false)
>>> extended_data.printSchema()
root
|-- time_id: string (nullable = true)
|-- ping_ms: string (nullable = true)
|-- temperature_c: string (nullable = true)
|-- humidity_p: string (nullable = true)
|-- offset: long (nullable = true)
|-- receive_time: timestamp (nullable = false)
>>> █
```

Схема не поменялась. Вотермарка только следит за чекпойнтом, но никак не влияет на наши данные.

Теперь данные можно проверить на наличие дубликатов. Дубли проверяем по двум колонкам: `species` и `receive\_time`. Таким образом будут отсеиваться дубли по полю `species` внутри одного микробатча, так как столбец `receive\_time` для всех записей внутри этого микробатча одинаковый. Для этого пишем новый датасет

Написать скрипт на python для конвертации файла csv в json.

```
import csv
```

```
import json
```

```
with open('test.csv') as f:
```

```
    reader = csv.DictReader(f)
```

```
    rows = list(reader)
```

```
with open('test.json', 'w') as f:
```

```
json.dump(rows, f)
```