1. Использовал docker-образы Bitnami для установки обеих систем: Kafka и Spark, т.к. установить непосредственно по инструкции, показанной на лекции, не получилось.

ПО	Версия ПО	Ссылка на Git	Команда docker
Kafka	3.8.0	https://github.com/bitnami/containers/tree/main/bitnami/kafka	docker pull bitnami/kafka
Spark	3.5.2	https://github.com/bitnami/containers/tree/main/bitnami/spark	docker pull bitnami/spark

2. Создал **docker-compose.yml** файл для запуска контейнеров в единой сети:

```
services:
zookeeper:
image: bitnami/zookeeper
ports:
- "2181:2181"
environment:
ALLOW_ANONYMOUS_LOGIN: "yes"
networks:
- kafka-network
kafka:
image: bitnami/kafka
environment:
KAFKA_CFG_ZOOKEEPER_CONNECT: zookeeper:2181
KAFKA_LISTENERS:
INSIDE://PLAINTEXT://0.0.0.0:29092,OUTSIDE://PLAINTEXT://0.0.0.0:9092
KAFKA_ADVERTISED_LISTENERS: INSIDE://kafka:29092,OUTSIDE://172.21.182.15:9092
KAFKA_LISTENER_SECURITY_PROTOCOL_MAP:
INSIDE:PLAINTEXT,OUTSIDE:PLAINTEXT
KAFKA_INTER_BROKER_LISTENER_NAME: INSIDE
ports:
- "9092:9092"
depends_on:
- zookeeper
networks:
- kafka-network
spark-master:
image: bitnami/spark
ports:
- "8080:8080"
- "7077:7077"
volumes:
- /mnt/d/Курсы и тренинги/Data Engineer Нетология/Spark Structured
Streaming:/opt/spark/work-dir
environment:
- SPARK_MODE=master
networks:
- kafka-network
spark-worker:
image: bitnami/spark
volumes:
- /mnt/d/Курсы и тренинги/Data Engineer Нетология/Spark Structured
Streaming:/opt/spark/work-dir
```

```
environment:
- SPARK_MODE=worker
- SPARK_WORKER_MASTER=spark://spark-master:7077
depends_on:
- spark-master
networks:
- kafka-network
networks:
kafka-network:
driver: bridge
```

Здесь добавлена сеть **kafka-network** для того, чтобы оба контейнера видели друг друга и могли обмениваться данными.

Добавление Zookeeper было необязательным, просто «из коробки» Kafka запускалась с ошибками и, потратив много сил, нашел в сети решение, основанное на Zookeeper. Сейчас понимаю, что можно было запустить без Zookeper, в режиме KRaft.

Что важно: пришлось переписать настройки Listeners, т.к. исходные Listeners от Bitnami настроены так, что не допускают подключения внешних клиентов:

```
- KAFKA_CFG_LISTENERS=PLAINTEXT://:9092,CONTROLLER://:9093
- KAFKA_CFG_ADVERTISED_LISTENERS=PLAINTEXT://:9092
```

В **docker-compose.yml** я разделил Listeners на внутренних и внешних, через указание разных портов, на которых Kafka слушает внутренних и внешних клиентов.
В качестве IP для внешнего хоста использовал IP WSL своего ПК
('172.21.182.15' определил командой `ip addr` в Bash)

Также в контейнеры добавлена опция `volumes` с указанием пути до python и spark-скриптов на моем локальном ПК для монтирования томов и обеспечения возможности запуска скриптов непосредственно внутри контейнеров.

3. Выполнил в терминале `docker-compose up`, развернул контейнеры в единой сети
4. В своем Python-окружении установил необходимые библиотеки:
   - pip install kafka
   - pip install pyspark
5. Поскольку Kafka установлена в контейнере, пришлось внести правки в скрипт **producer.py**:
   - вместо строки `KafkaProducer(bootstrap_servers=['localhost:9092']`
     прописал `KafkaProducer(bootstrap_servers=['172.21.182.15:9092'],`
     где '172.21.182.15' – мой IP WSL (определил командой `ip addr` в Bash)
   - в строке `producer.send`
     добавил методы `.add_callback` и `.add_errback`, вызывающие соответствующие функции для формирования логов и получения обратной связи о том, было ли сообщение успешно отправлено:

```

# Функция для обратного вызова при успешной отправке сообщения
def on_send_success(record_metadata):
    print(f"Message sent to {record_metadata.topic} partition
{record_metadata.partition}with offset {record_metadata.offset}")

# Функция для обратного вызова при ошибке
def on_send_error(excp):
    print(f"Message delivery failed: {excp}")
```

6. Конечная версия скрипта **producer.py**, генерирующего входной поток для Kafka:

```
from time import sleep
from json import dumps
from kafka import KafkaProducer
from random import randrange, choices
import string

# Функция для обратного вызова при успешной отправке сообщения
def on_send_success(record_metadata):
    print(f"Message sent to {record_metadata.topic} partition {record_metadata.partition} with
offset {record_metadata.offset}")

# Функция для обратного вызова при ошибке
def on_send_error(excp):
    print(f"Message delivery failed: {excp}")

producer = KafkaProducer(bootstrap_servers=['172.21.182.15:9092'],
                         value_serializer=lambda x: dumps(x).encode('utf-8'),
                         retries=5)

def push():
    for e in range(50):
        text = ''.join(choices(string.ascii_uppercase + string.digits, k=20))
        user = {'id': randrange(5), 'action': text}
        # Отправляем сообщение с обработчиками
        producer.send('netology-spark',
value=user).add_callback(on_send_success).add_errback(on_send_error)
        sleep(randrange(3))
        print("Produced message ", e)

try:
    while True:
        push()
except KeyboardInterrupt:
    producer.close()
    print("Exit")
```

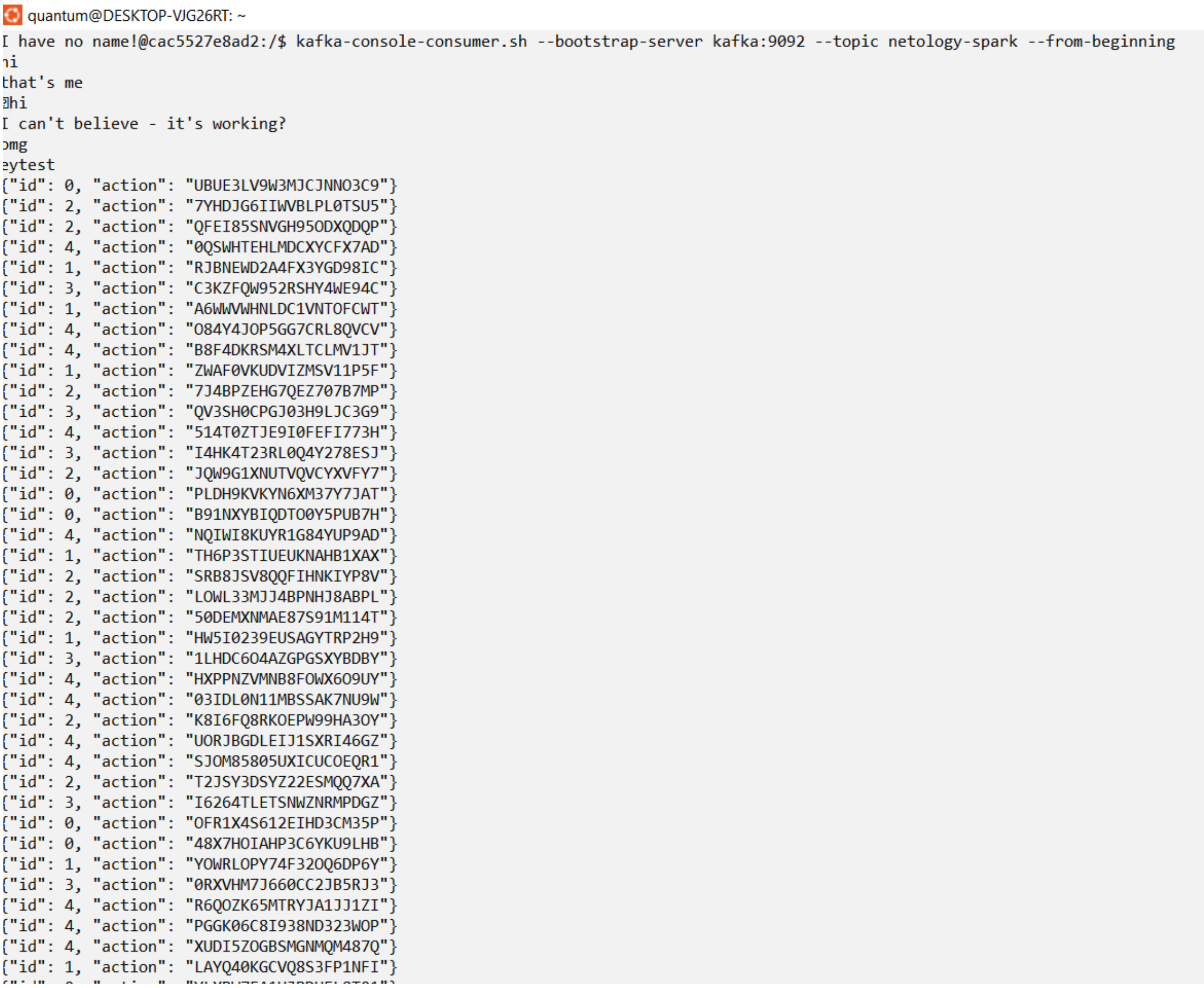
7. Запуск скрипта генерирует принты вида:

```
Message sent to netology-spark partition 0 with offset 255
Produced message  0
Produced message  1
Message sent to netology-spark partition 0 with offset 256
Message sent to netology-spark partition 0 with offset 257
Produced message  2
Message sent to netology-spark partition 0 with offset 258
Produced message  3
Message sent to netology-spark partition 0 with offset 259
Produced message  4
Produced message  5
Message sent to netology-spark partition 0 with offset 260
```

Message sent to netology-spark partition 0 with offset 261

которые свидетельствует об успешной отправке сообщений в Kafka.

8. Заходим в контейнер Kafka. Для этого:
    - командой docker ps смотрим id контейнера Kafka
    - вводим команду docker exec -it <id контейнера> /bin/bash
    - для чтения входящего потока внутри контейнера вводим kafka-console-consumer.sh --bootstrap-server kafka:9092 --topic netology-spark --from-beginning
    - видим поступающие от продюсера сообщения в формате бинарных данных:



```
I have no name!@cac5527e8ad2:/$ kafka-console-consumer.sh --bootstrap-server kafka:9092 --topic netology-spark --from-beginning
ıi
that's me
▯hi
I can't believe - it's working?
ɔmg
ɛytest
{"id": 0, "action": "UBUE3LV9W3MJCJNNO3C9"}
{"id": 2, "action": "7YHDJG6IIWVBLPL0TSU5"}
{"id": 2, "action": "QFEI85SNVGH95ODXQDQP"}
{"id": 4, "action": "0QSWHTEHLMDCXYCFX7AD"}
{"id": 1, "action": "RJBNEWD2A4FX3YGD98IC"}
{"id": 3, "action": "C3KZFQW952RSHY4WE94C"}
{"id": 1, "action": "A6WWVWHNLDC1VNTOFCWT"}
{"id": 4, "action": "O84Y4JOP5GG7CRL8QVCV"}
{"id": 4, "action": "B8F4DKRSM4XLTCLMV1JT"}
{"id": 1, "action": "ZWAF0VKUDVIZMSV11P5F"}
{"id": 2, "action": "7J4BPZEHG7QEZ707B7MP"}
{"id": 3, "action": "QV3SH0CPGJ03H9LJC3G9"}
{"id": 4, "action": "514T0ZTJE9I0FEFI773H"}
{"id": 3, "action": "I4HK4T23RL0Q4Y278ESJ"}
{"id": 2, "action": "JQW9G1XNUTVQVCYXVFY7"}
{"id": 0, "action": "PLDH9KVKYN6XM37Y7JAT"}
{"id": 0, "action": "B91NXYBIQDTO0Y5PUB7H"}
{"id": 4, "action": "NQIWI8KUYR1G84YUP9AD"}
{"id": 1, "action": "TH6P3STIUEUKNAHB1XAX"}
{"id": 2, "action": "SRB8JSV8QQFIHNKIYP8V"}
{"id": 2, "action": "LOWL33MJJ4BPNHJ8ABPL"}
{"id": 2, "action": "50DEMXNMAE87S91M114T"}
{"id": 1, "action": "HW5I0239EUSAGYTRP2H9"}
{"id": 3, "action": "1LHDC6O4AZGPGSXYBDBY"}
{"id": 4, "action": "HXPPNZVMNB8FOWX6O9UY"}
{"id": 4, "action": "03IDL0N11MBSSAK7NU9W"}
{"id": 2, "action": "K8I6FQ8RKOEPW99HA3OY"}
{"id": 4, "action": "UORJBGDLEIJ1SXRI46GZ"}
{"id": 4, "action": "SJOM85805UXICUCOEQR1"}
{"id": 2, "action": "T2JSY3DSYZ22ESMQQ7XA"}
{"id": 3, "action": "I6264TLETSNWZNRMPDGZ"}
{"id": 0, "action": "OFR1X4S612EIHD3CM35P"}
{"id": 0, "action": "48X7HOIAHP3C6YKU9LHB"}
{"id": 1, "action": "YOWRLOPY74F32OQ6DP6Y"}
{"id": 3, "action": "0RXVHM7J660CC2JB5RJ3"}
{"id": 4, "action": "R6QOZK65MTRYJA1JJ1ZI"}
{"id": 4, "action": "PGGK06C8I938ND323WOP"}
{"id": 4, "action": "XUDI5ZOGBSMGNMQM487Q"}
{"id": 1, "action": "LAYQ40KGCVQ8S3FP1NFI"}
```

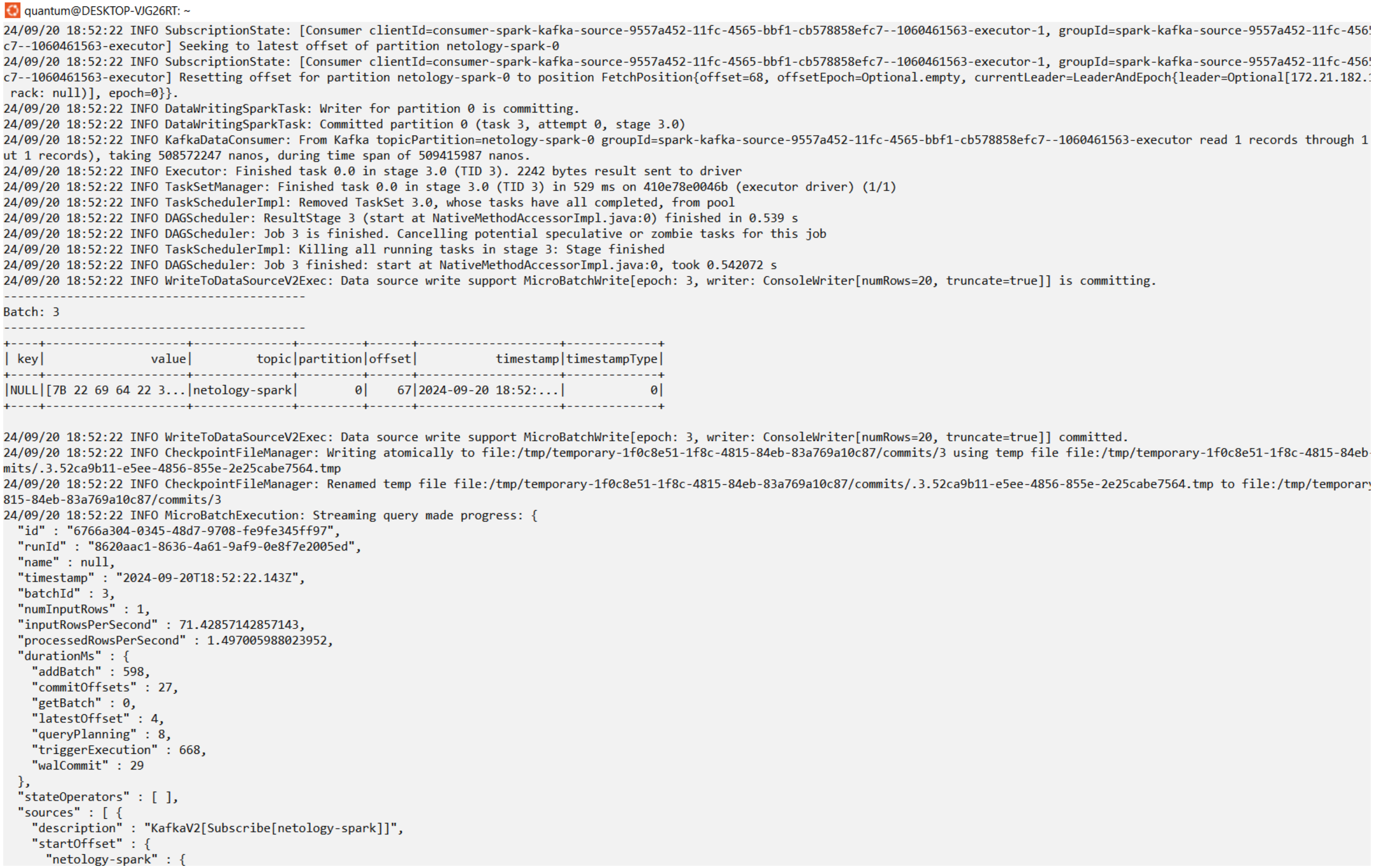
9. Дальше заходим в контейнер Spark с помощью команды docker exec - t <id контейнера Spark> /bin/bash

10. Для получения входного потока из Kafka внутри контейнера Spark выполняем /opt/bitnami/spark/conf$ spark-submit --packages org.apache.spark:spark-sql-kafka-0-10_2.12:3.5.2 /opt/spark/work-dir/structure_streaming_kafka_test.py

    Я создал бэкап версию Spark-скрипта (structure_streaming_kafka_test.py) и работал с ней. Данный скрипт хранится в директории, к которой был смонтирован том в разделе volumes docker-compose.yml

11. Дальнейшие манипуляции и распаковка данных из Kafka осуществлялись исключительно через редактирование файла **structure_streaming_kafka_test.py:**

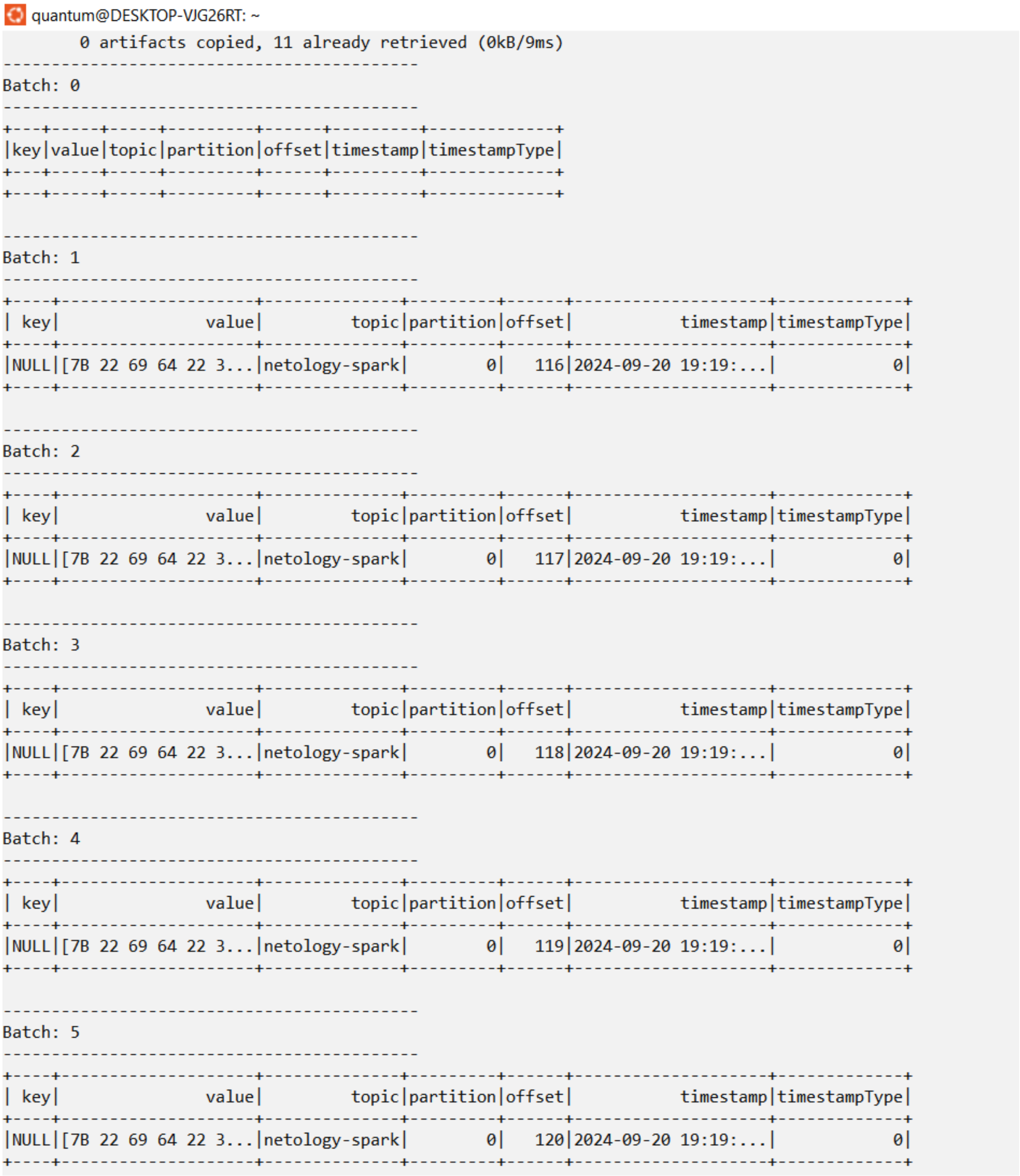
- Чтение бинарных данных с логированием на уровне **info**:



quantum@DESKTOP-VJG26RT: ~

```
24/09/20 18:52:22 INFO SubscriptionState: [Consumer clientId=consumer-spark-kafka-source-9557a452-11fc-4565-bbf1-cb578858efc7--1060461563-executor-1, groupId=spark-kafka-source-9557a452-11fc-456
c7--1060461563-executor] Seeking to latest offset of partition netology-spark-0
24/09/20 18:52:22 INFO SubscriptionState: [Consumer clientId=consumer-spark-kafka-source-9557a452-11fc-4565-bbf1-cb578858efc7--1060461563-executor-1, groupId=spark-kafka-source-9557a452-11fc-456
c7--1060461563-executor] Resetting offset for partition netology-spark-0 to position FetchPosition{offset=68, offsetEpoch=Optional.empty, currentLeader=LeaderAndEpoch{leader=Optional[172.21.182.
 rack: null)], epoch=0}}.
24/09/20 18:52:22 INFO DataWritingSparkTask: Writer for partition 0 is committing.
24/09/20 18:52:22 INFO DataWritingSparkTask: Committed partition 0 (task 3, attempt 0, stage 3.0)
24/09/20 18:52:22 INFO KafkaDataConsumer: From Kafka topicPartition=netology-spark-0 groupId=spark-kafka-source-9557a452-11fc-4565-bbf1-cb578858efc7--1060461563-executor read 1 records through 1
ut 1 records), taking 508572247 nanos, during time span of 509415987 nanos.
24/09/20 18:52:22 INFO Executor: Finished task 0.0 in stage 3.0 (TID 3). 2242 bytes result sent to driver
24/09/20 18:52:22 INFO TaskSetManager: Finished task 0.0 in stage 3.0 (TID 3) in 529 ms on 410e78e0046b (executor driver) (1/1)
24/09/20 18:52:22 INFO TaskSchedulerImpl: Removed TaskSet 3.0, whose tasks have all completed, from pool
24/09/20 18:52:22 INFO DAGScheduler: ResultStage 3 (start at NativeMethodAccessorImpl.java:0) finished in 0.539 s
24/09/20 18:52:22 INFO DAGScheduler: Job 3 is finished. Cancelling potential speculative or zombie tasks for this job
24/09/20 18:52:22 INFO TaskSchedulerImpl: Killing all running tasks in stage 3: Stage finished
24/09/20 18:52:22 INFO DAGScheduler: Job 3 finished: start at NativeMethodAccessorImpl.java:0, took 0.542072 s
24/09/20 18:52:22 INFO WriteToDataSourceV2Exec: Data source write support MicroBatchWrite[epoch: 3, writer: ConsoleWriter[numRows=20, truncate=true]] is committing.
-------------------------------------------
Batch: 3
-------------------------------------------
+----+--------------------+--------------+---------+------+--------------------+-------------+
| key|               value|         topic|partition|offset|           timestamp|timestampType|
+----+--------------------+--------------+---------+------+--------------------+-------------+
|NULL|[7B 22 69 64 22 3...|netology-spark|        0|    67|2024-09-20 18:52:...|            0|
+----+--------------------+--------------+---------+------+--------------------+-------------+

24/09/20 18:52:22 INFO WriteToDataSourceV2Exec: Data source write support MicroBatchWrite[epoch: 3, writer: ConsoleWriter[numRows=20, truncate=true]] committed.
24/09/20 18:52:22 INFO CheckpointFileManager: Writing atomically to file:/tmp/temporary-1f0c8e51-1f8c-4815-84eb-83a769a10c87/commits/3 using temp file file:/tmp/temporary-1f0c8e51-1f8c-4815-84eb-
mits/.3.52ca9b11-e5ee-4856-855e-2e25cabe7564.tmp
24/09/20 18:52:22 INFO CheckpointFileManager: Renamed temp file file:/tmp/temporary-1f0c8e51-1f8c-4815-84eb-83a769a10c87/commits/.3.52ca9b11-e5ee-4856-855e-2e25cabe7564.tmp to file:/tmp/temporary
815-84eb-83a769a10c87/commits/3
24/09/20 18:52:22 INFO MicroBatchExecution: Streaming query made progress: {
  "id" : "6766a304-0345-48d7-9708-fe9fe345ff97",
  "runId" : "8620aac1-8636-4a61-9af9-0e8f7e2005ed",
  "name" : null,
  "timestamp" : "2024-09-20T18:52:22.143Z",
  "batchId" : 3,
  "numInputRows" : 1,
  "inputRowsPerSecond" : 71.42857142857143,
  "processedRowsPerSecond" : 1.497005988023952,
  "durationMs" : {
    "addBatch" : 598,
    "commitOffsets" : 27,
    "getBatch" : 0,
    "latestOffset" : 4,
    "queryPlanning" : 8,
    "triggerExecution" : 668,
    "walCommit" : 29
  },
  "stateOperators" : [ ],
  "sources" : [ {
    "description" : "KafkaV2[Subscribe[netology-spark]]",
    "startOffset" : {
      "netology-spark" : {
```

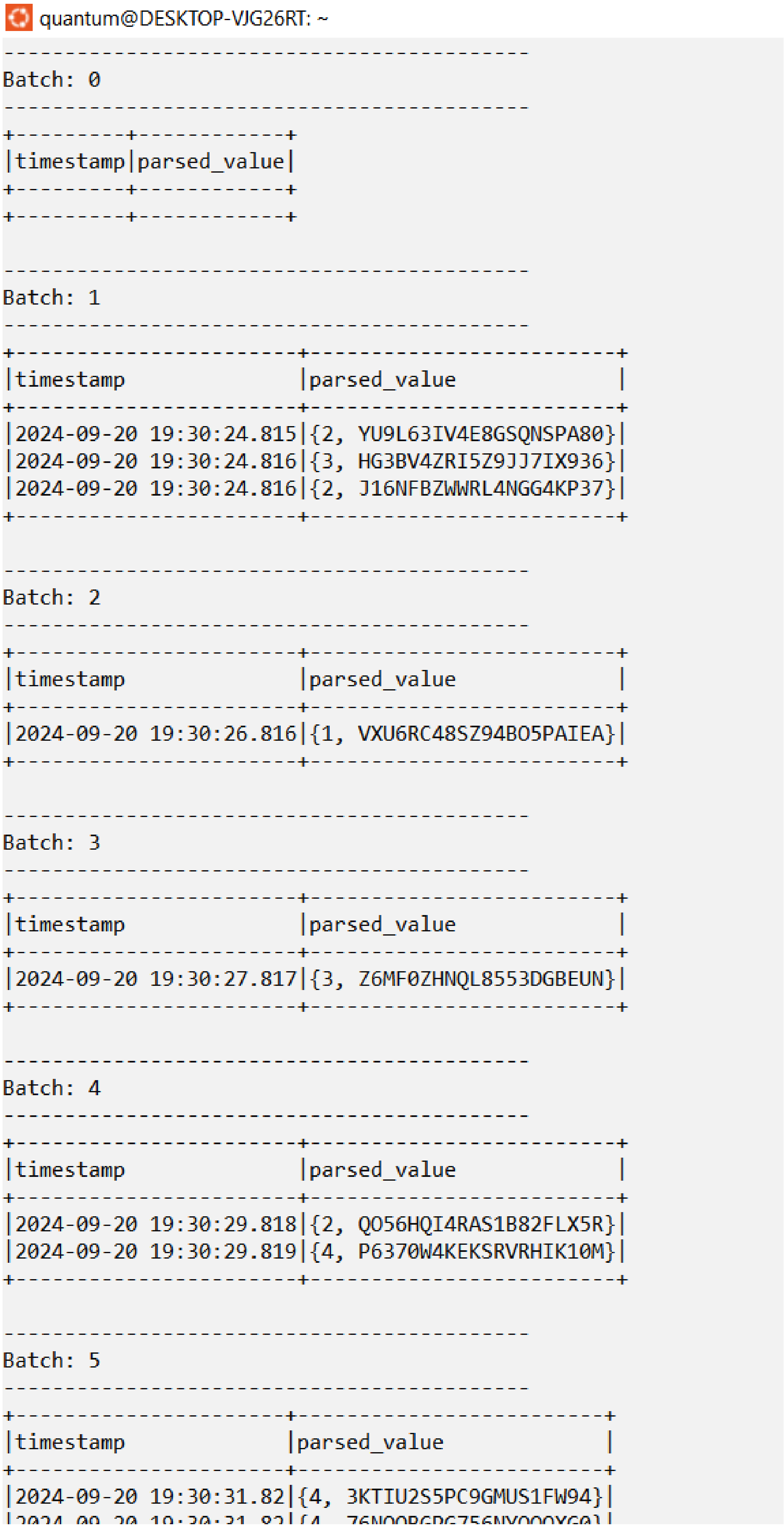
- Чтение бинарных данных с логированием на уровне **error**:



quantum@DESKTOP-VJG26RT: ~

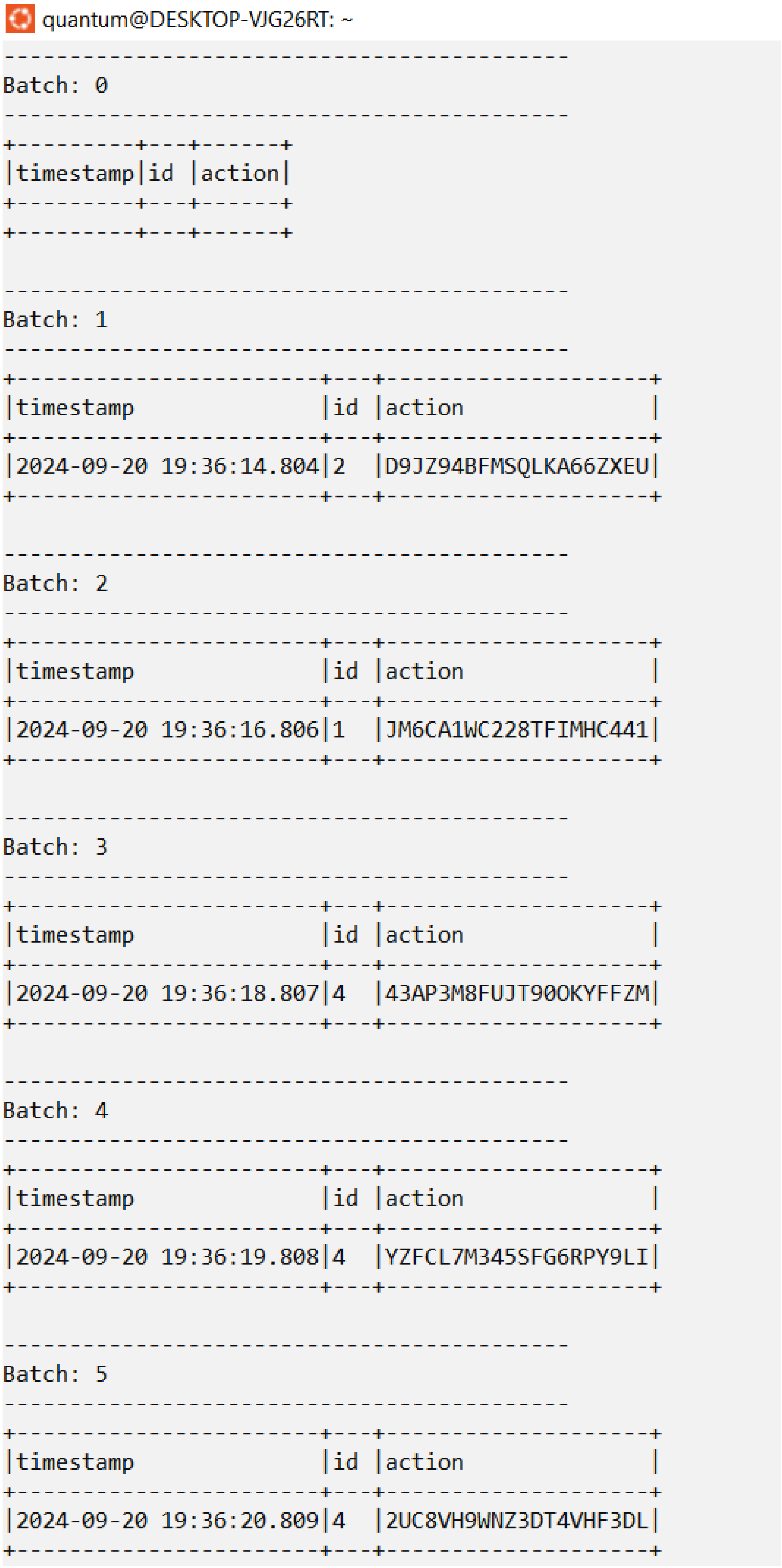
```
        0 artifacts copied, 11 already retrieved (0kB/9ms)
-------------------------------------------
Batch: 0
-------------------------------------------
+---+-----+-----+---------+------+---------+-------------+
|key|value|topic|partition|offset|timestamp|timestampType|
+---+-----+-----+---------+------+---------+-------------+
+---+-----+-----+---------+------+---------+-------------+


-------------------------------------------
Batch: 1
-------------------------------------------
+----+--------------------+--------------+---------+------+--------------------+-------------+
| key|               value|         topic|partition|offset|           timestamp|timestampType|
+----+--------------------+--------------+---------+------+--------------------+-------------+
|NULL|[7B 22 69 64 22 3...|netology-spark|        0|   116|2024-09-20 19:19:...|            0|
+----+--------------------+--------------+---------+------+--------------------+-------------+


-------------------------------------------
Batch: 2
-------------------------------------------
+----+--------------------+--------------+---------+------+--------------------+-------------+
| key|               value|         topic|partition|offset|           timestamp|timestampType|
+----+--------------------+--------------+---------+------+--------------------+-------------+
|NULL|[7B 22 69 64 22 3...|netology-spark|        0|   117|2024-09-20 19:19:...|            0|
+----+--------------------+--------------+---------+------+--------------------+-------------+


-------------------------------------------
Batch: 3
-------------------------------------------
+----+--------------------+--------------+---------+------+--------------------+-------------+
| key|               value|         topic|partition|offset|           timestamp|timestampType|
+----+--------------------+--------------+---------+------+--------------------+-------------+
|NULL|[7B 22 69 64 22 3...|netology-spark|        0|   118|2024-09-20 19:19:...|            0|
+----+--------------------+--------------+---------+------+--------------------+-------------+


-------------------------------------------
Batch: 4
-------------------------------------------
+----+--------------------+--------------+---------+------+--------------------+-------------+
| key|               value|         topic|partition|offset|           timestamp|timestampType|
+----+--------------------+--------------+---------+------+--------------------+-------------+
|NULL|[7B 22 69 64 22 3...|netology-spark|        0|   119|2024-09-20 19:19:...|            0|
+----+--------------------+--------------+---------+------+--------------------+-------------+


-------------------------------------------
Batch: 5
-------------------------------------------
+----+--------------------+--------------+---------+------+--------------------+-------------+
| key|               value|         topic|partition|offset|           timestamp|timestampType|
+----+--------------------+--------------+---------+------+--------------------+-------------+
|NULL|[7B 22 69 64 22 3...|netology-spark|        0|   120|2024-09-20 19:19:...|            0|
+----+--------------------+--------------+---------+------+--------------------+-------------+
```

- Распакованные данные после применения схемы и перевода в json:



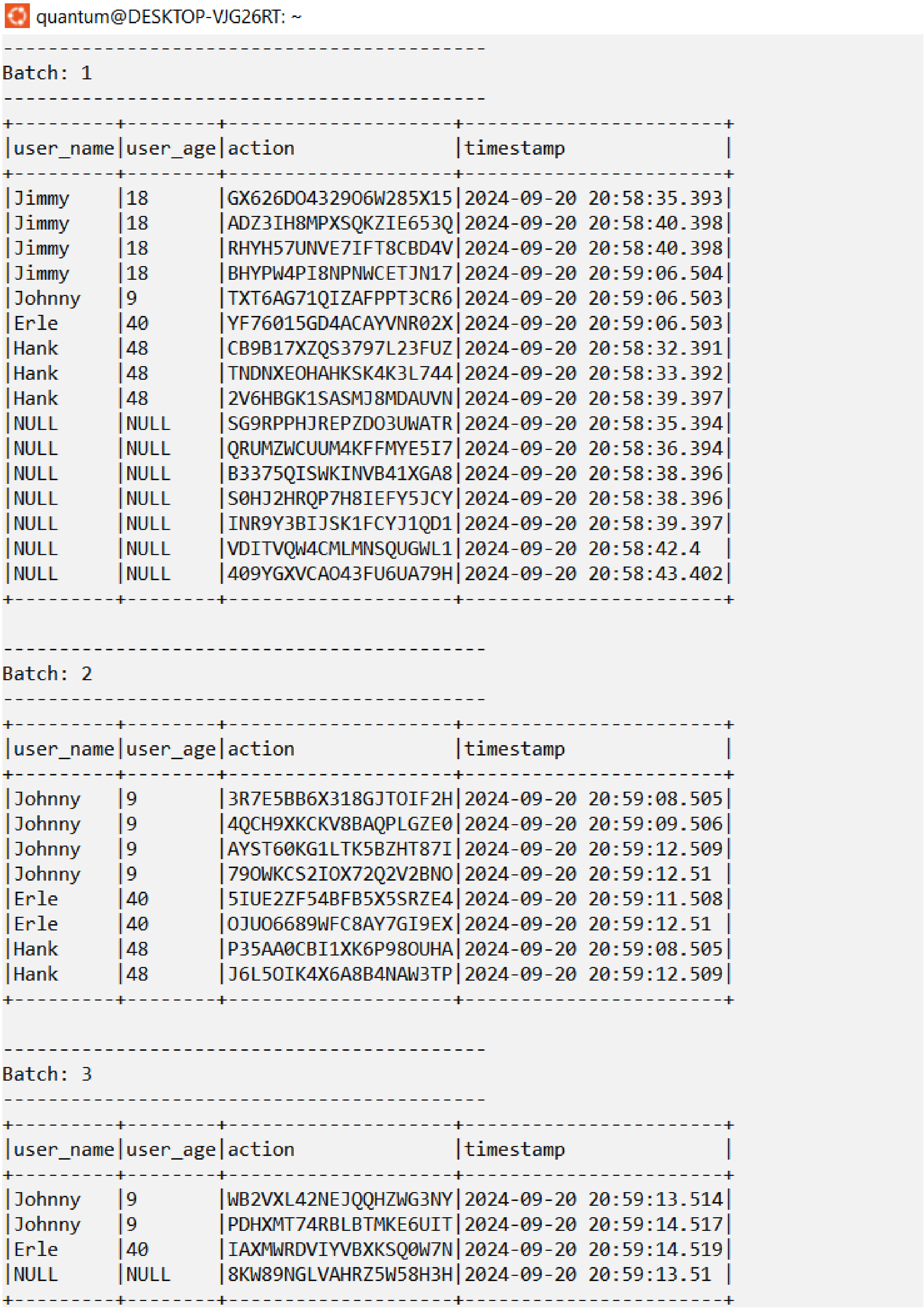
```
-------------------------------------------
Batch: 0
-------------------------------------------
+---------+------------+
|timestamp|parsed_value|
+---------+------------+
+---------+------------+


-------------------------------------------
Batch: 1
-------------------------------------------
+-----------------------+-------------------------+
|timestamp              |parsed_value             |
+-----------------------+-------------------------+
|2024-09-20 19:30:24.815|{2, YU9L63IV4E8GSQNSPA80}|
|2024-09-20 19:30:24.816|{3, HG3BV4ZRI5Z9JJ7IX936}|
|2024-09-20 19:30:24.816|{2, J16NFBZWWRL4NGG4KP37}|
+-----------------------+-------------------------+


-------------------------------------------
Batch: 2
-------------------------------------------
+-----------------------+-------------------------+
|timestamp              |parsed_value             |
+-----------------------+-------------------------+
|2024-09-20 19:30:26.816|{1, VXU6RC48SZ94BO5PAIEA}|
+-----------------------+-------------------------+


-------------------------------------------
Batch: 3
-------------------------------------------
+-----------------------+-------------------------+
|timestamp              |parsed_value             |
+-----------------------+-------------------------+
|2024-09-20 19:30:27.817|{3, Z6MF0ZHNQL8553DGBEUN}|
+-----------------------+-------------------------+


-------------------------------------------
Batch: 4
-------------------------------------------
+-----------------------+-------------------------+
|timestamp              |parsed_value             |
+-----------------------+-------------------------+
|2024-09-20 19:30:29.818|{2, QO56HQI4RAS1B82FLX5R}|
|2024-09-20 19:30:29.819|{4, P6370W4KEKSRVRHIK10M}|
+-----------------------+-------------------------+


-------------------------------------------
Batch: 5
-------------------------------------------
+----------------------+-------------------------+
|timestamp             |parsed_value             |
+----------------------+-------------------------+
|2024-09-20 19:30:31.82|{4, 3KTIU2S5PC9GMUS1FW94}|
|2024-09-20 19:30:31.82|{4, Z6NOOBGBG756NYOOOXG0}|
```

- Распакованные данные до уровня user_id и action:



```
-------------------------------------------
Batch: 0
-------------------------------------------
+---------+---+------+
|timestamp|id |action|
+---------+---+------+
+---------+---+------+

-------------------------------------------
Batch: 1
-------------------------------------------
+-----------------------+---+--------------------+
|timestamp              |id |action              |
+-----------------------+---+--------------------+
|2024-09-20 19:36:14.804|2  |D9JZ94BFMSQLKA66ZXEU|
+-----------------------+---+--------------------+

-------------------------------------------
Batch: 2
-------------------------------------------
+-----------------------+---+--------------------+
|timestamp              |id |action              |
+-----------------------+---+--------------------+
|2024-09-20 19:36:16.806|1  |JM6CA1WC228TFIMHC441|
+-----------------------+---+--------------------+

-------------------------------------------
Batch: 3
-------------------------------------------
+-----------------------+---+--------------------+
|timestamp              |id |action              |
+-----------------------+---+--------------------+
|2024-09-20 19:36:18.807|4  |43AP3M8FUJT90OKYFFZM|
+-----------------------+---+--------------------+

-------------------------------------------
Batch: 4
-------------------------------------------
+-----------------------+---+--------------------+
|timestamp              |id |action              |
+-----------------------+---+--------------------+
|2024-09-20 19:36:19.808|4  |YZFCL7M345SFG6RPY9LI|
+-----------------------+---+--------------------+

-------------------------------------------
Batch: 5
-------------------------------------------
+-----------------------+---+--------------------+
|timestamp              |id |action              |
+-----------------------+---+--------------------+
|2024-09-20 19:36:20.809|4  |2UC8VH9WNZ3DT4VHF3DL|
+-----------------------+---+--------------------+
```

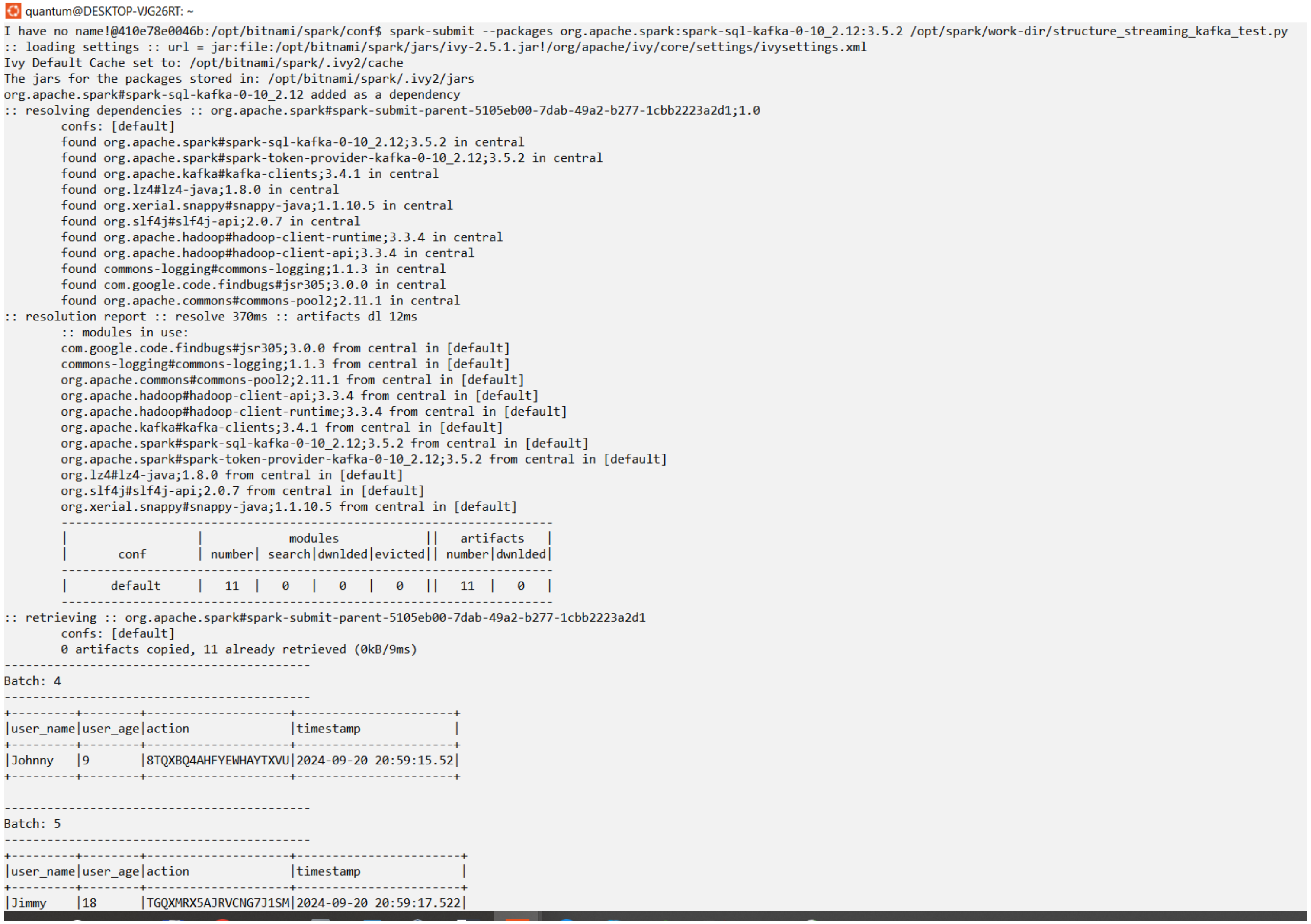
- Данные после джойна входящего и статического потоков:



quantum@DESKTOP-VJG26RT: ~

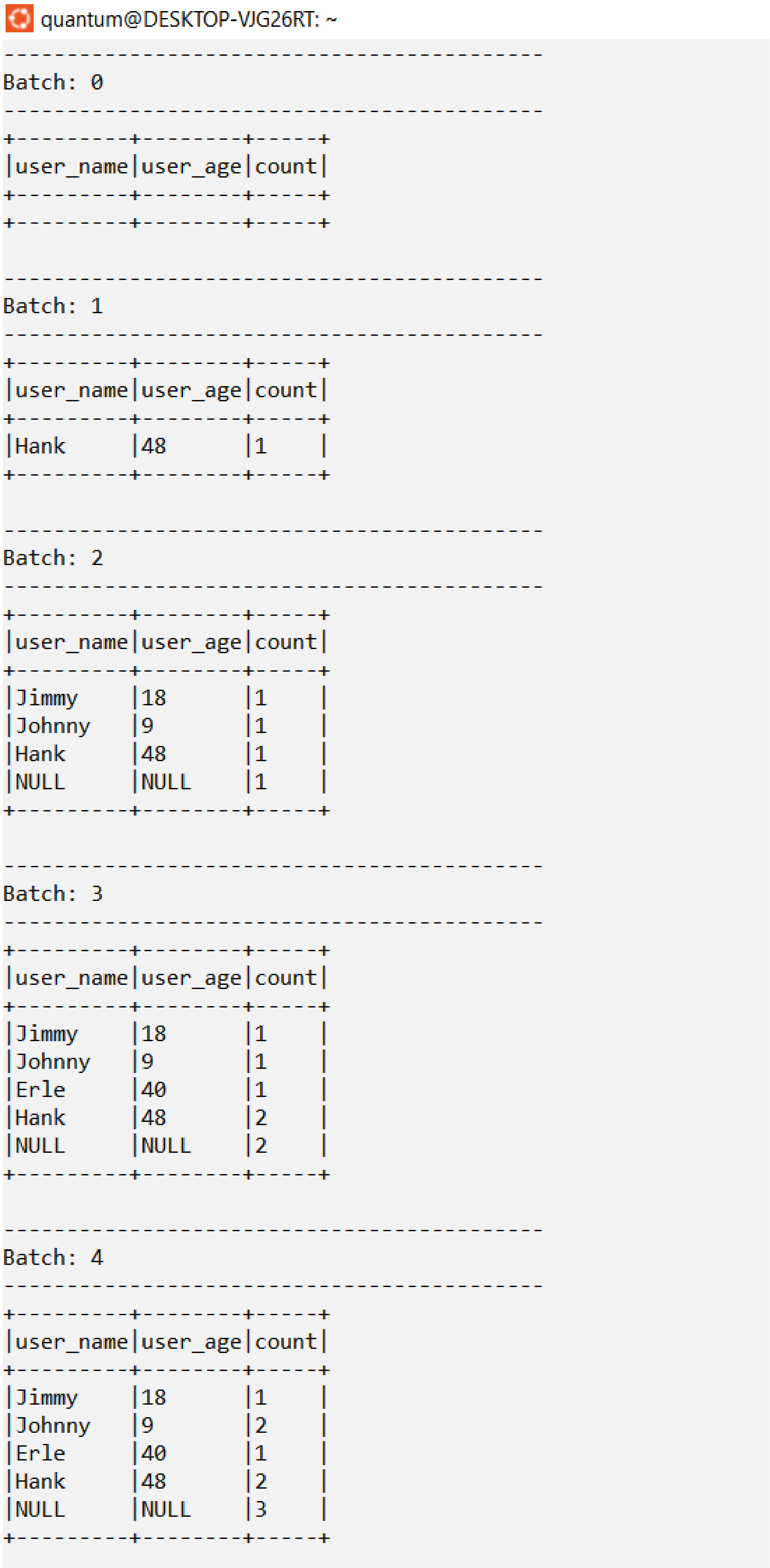
```
-------------------------------------------
Batch: 1
-------------------------------------------
+---------+--------+--------------------+-----------------------+
|user_name|user_age|action              |timestamp              |
+---------+--------+--------------------+-----------------------+
|Jimmy    |18      |GX626DO432906W285X15|2024-09-20 20:58:35.393|
|Jimmy    |18      |ADZ3IH8MPXSQKZIE653Q|2024-09-20 20:58:40.398|
|Jimmy    |18      |RHYH57UNVE7IFT8CBD4V|2024-09-20 20:58:40.398|
|Jimmy    |18      |BHYPW4PI8NPNWCETJN17|2024-09-20 20:59:06.504|
|Johnny   |9       |TXT6AG71QIZAFPPT3CR6|2024-09-20 20:59:06.503|
|Erle     |40      |YF76015GD4ACAYVNR02X|2024-09-20 20:59:06.503|
|Hank     |48      |CB9B17XZQS3797L23FUZ|2024-09-20 20:58:32.391|
|Hank     |48      |TNDNXEOHAHKSK4K3L744|2024-09-20 20:58:33.392|
|Hank     |48      |2V6HBGK1SASMJ8MDAUVN|2024-09-20 20:58:39.397|
|NULL     |NULL    |SG9RPPHJREPZDO3UWATR|2024-09-20 20:58:35.394|
|NULL     |NULL    |QRUMZWCUUM4KFFMYE5I7|2024-09-20 20:58:36.394|
|NULL     |NULL    |B3375QISWKINVB41XGA8|2024-09-20 20:58:38.396|
|NULL     |NULL    |S0HJ2HRQP7H8IEFY5JCY|2024-09-20 20:58:38.396|
|NULL     |NULL    |INR9Y3BIJSK1FCYJ1QD1|2024-09-20 20:58:39.397|
|NULL     |NULL    |VDITVQW4CMLMNSQUGWL1|2024-09-20 20:58:42.4  |
|NULL     |NULL    |409YGXVCAO43FU6UA79H|2024-09-20 20:58:43.402|
+---------+--------+--------------------+-----------------------+


-------------------------------------------
Batch: 2
-------------------------------------------
+---------+--------+--------------------+-----------------------+
|user_name|user_age|action              |timestamp              |
+---------+--------+--------------------+-----------------------+
|Johnny   |9       |3R7E5BB6X318GJTOIF2H|2024-09-20 20:59:08.505|
|Johnny   |9       |4QCH9XKCKV8BAQPLGZE0|2024-09-20 20:59:09.506|
|Johnny   |9       |AYST60KG1LTK5BZHT87I|2024-09-20 20:59:12.509|
|Johnny   |9       |79OWKCS2IOX72Q2V2BNO|2024-09-20 20:59:12.51 |
|Erle     |40      |5IUE2ZF54BFB5X5SRZE4|2024-09-20 20:59:11.508|
|Erle     |40      |OJUO6689WFC8AY7GI9EX|2024-09-20 20:59:12.51 |
|Hank     |48      |P35AA0CBI1XK6P98OUHA|2024-09-20 20:59:08.505|
|Hank     |48      |J6L5OIK4X6A8B4NAW3TP|2024-09-20 20:59:12.509|
+---------+--------+--------------------+-----------------------+


-------------------------------------------
Batch: 3
-------------------------------------------
+---------+--------+--------------------+-----------------------+
|user_name|user_age|action              |timestamp              |
+---------+--------+--------------------+-----------------------+
|Johnny   |9       |WB2VXL42NEJQQHZWG3NY|2024-09-20 20:59:13.514|
|Johnny   |9       |PDHXMT74RBLBTMKE6UIT|2024-09-20 20:59:14.517|
|Erle     |40      |IAXMWRDVIYVBXKSQ0W7N|2024-09-20 20:59:14.519|
|NULL     |NULL    |8KW89NGLVAHRZ5W58H3H|2024-09-20 20:59:13.51 |
+---------+--------+--------------------+-----------------------+
```

- После добавления чекпойнтов фиксируем, что чтение данных из входного потока возобновляется не с 0, а с последнего прочитанного батча:



```
I have no name!@410e78e0046b:/opt/bitnami/spark/conf$ spark-submit --packages org.apache.spark:spark-sql-kafka-0-10_2.12:3.5.2 /opt/spark/work-dir/structure_streaming_kafka_test.py
:: loading settings :: url = jar:file:/opt/bitnami/spark/jars/ivy-2.5.1.jar!/org/apache/ivy/core/settings/ivysettings.xml
Ivy Default Cache set to: /opt/bitnami/spark/.ivy2/cache
The jars for the packages stored in: /opt/bitnami/spark/.ivy2/jars
org.apache.spark#spark-sql-kafka-0-10_2.12 added as a dependency
:: resolving dependencies :: org.apache.spark#spark-submit-parent-5105eb00-7dab-49a2-b277-1cbb2223a2d1;1.0
        confs: [default]
        found org.apache.spark#spark-sql-kafka-0-10_2.12;3.5.2 in central
        found org.apache.spark#spark-token-provider-kafka-0-10_2.12;3.5.2 in central
        found org.apache.kafka#kafka-clients;3.4.1 in central
        found org.lz4#lz4-java;1.8.0 in central
        found org.xerial.snappy#snappy-java;1.1.10.5 in central
        found org.slf4j#slf4j-api;2.0.7 in central
        found org.apache.hadoop#hadoop-client-runtime;3.3.4 in central
        found org.apache.hadoop#hadoop-client-api;3.3.4 in central
        found commons-logging#commons-logging;1.1.3 in central
        found com.google.code.findbugs#jsr305;3.0.0 in central
        found org.apache.commons#commons-pool2;2.11.1 in central
:: resolution report :: resolve 370ms :: artifacts dl 12ms
        :: modules in use:
        com.google.code.findbugs#jsr305;3.0.0 from central in [default]
        commons-logging#commons-logging;1.1.3 from central in [default]
        org.apache.commons#commons-pool2;2.11.1 from central in [default]
        org.apache.hadoop#hadoop-client-api;3.3.4 from central in [default]
        org.apache.hadoop#hadoop-client-runtime;3.3.4 from central in [default]
        org.apache.kafka#kafka-clients;3.4.1 from central in [default]
        org.apache.spark#spark-sql-kafka-0-10_2.12;3.5.2 from central in [default]
        org.apache.spark#spark-token-provider-kafka-0-10_2.12;3.5.2 from central in [default]
        org.lz4#lz4-java;1.8.0 from central in [default]
        org.slf4j#slf4j-api;2.0.7 from central in [default]
        org.xerial.snappy#snappy-java;1.1.10.5 from central in [default]
        ---------------------------------------------------------------------
        |                  |            modules            ||   artifacts   |
        |       conf       | number| search|dwnlded|evicted|| number|dwnlded|
        ---------------------------------------------------------------------
        |      default     |   11  |   0   |   0   |   0   ||   11  |   0   |
        ---------------------------------------------------------------------
:: retrieving :: org.apache.spark#spark-submit-parent-5105eb00-7dab-49a2-b277-1cbb2223a2d1
        confs: [default]
        0 artifacts copied, 11 already retrieved (0kB/9ms)
-------------------------------------------
Batch: 4
-------------------------------------------
+---------+--------+--------------------+----------------------+
|user_name|user_age|action              |timestamp             |
+---------+--------+--------------------+----------------------+
|Johnny   |9       |8TQXBQ4AHFYEWHAYTXVU|2024-09-20 20:59:15.52|
+---------+--------+--------------------+----------------------+

-------------------------------------------
Batch: 5
-------------------------------------------
+---------+--------+--------------------+-----------------------+
|user_name|user_age|action              |timestamp              |
+---------+--------+--------------------+-----------------------+
|Jimmy    |18      |TGQXMRX5AJRVCNG7J1SM|2024-09-20 20:59:17.522|
```

- Чтение данных после агрегации:

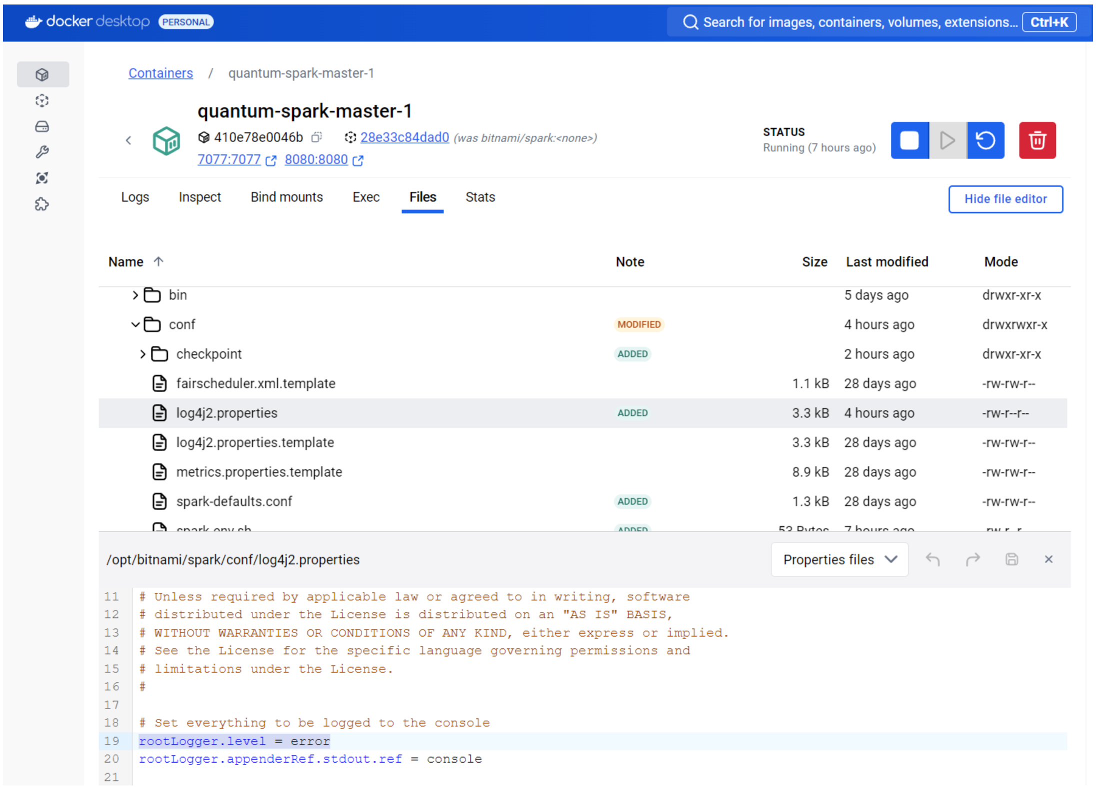


```
-------------------------------------------
Batch: 0
-------------------------------------------
+---------+--------+-----+
|user_name|user_age|count|
+---------+--------+-----+
+---------+--------+-----+

-------------------------------------------
Batch: 1
-------------------------------------------
+---------+--------+-----+
|user_name|user_age|count|
+---------+--------+-----+
|Hank     |48      |1    |
+---------+--------+-----+

-------------------------------------------
Batch: 2
-------------------------------------------
+---------+--------+-----+
|user_name|user_age|count|
+---------+--------+-----+
|Jimmy    |18      |1    |
|Johnny   |9       |1    |
|Hank     |48      |1    |
|NULL     |NULL    |1    |
+---------+--------+-----+

-------------------------------------------
Batch: 3
-------------------------------------------
+---------+--------+-----+
|user_name|user_age|count|
+---------+--------+-----+
|Jimmy    |18      |1    |
|Johnny   |9       |1    |
|Erle     |40      |1    |
|Hank     |48      |2    |
|NULL     |NULL    |2    |
+---------+--------+-----+

-------------------------------------------
Batch: 4
-------------------------------------------
+---------+--------+-----+
|user_name|user_age|count|
+---------+--------+-----+
|Jimmy    |18      |1    |
|Johnny   |9       |2    |
|Erle     |40      |1    |
|Hank     |48      |2    |
|NULL     |NULL    |3    |
+---------+--------+-----+
```

12. Итоговый Spark-скрипт:

```
from pyspark.sql import SparkSession
from time import sleep
from pyspark.sql.functions import col,from_json
from pyspark.sql.types import StructType, StringType, IntegerType

# явным образом задаем структуру json-контента
schema = StructType().add("id",IntegerType()).add("action", StringType())

users_schema = StructType().add("id",IntegerType()).add("user_name",
StringType()).add("user_age", IntegerType())

spark = SparkSession.builder.appName("SparkStreamingKafka").getOrCreate()

input_stream = spark \
  .readStream \
  .format("kafka") \
  .option("kafka.bootstrap.servers", "kafka:29092") \
  .option("subscribe", "netology-spark") \
  .option("failOnDataLoss", False) \
  .load()

#покажем входящий контент
#input_stream.writeStream.format("console").outputMode("append").start().awaitTermination()
#input_stream = input_stream.writeStream.format("console").outputMode("append").start()

json_stream = input_stream.select(col("timestamp").cast("string"),
from_json(col("value").cast("string"), schema).alias("parsed_value"))
#json_stream.writeStream.format("console").outputMode("append").option("truncate",
False).start().awaitTermination()

#выделим интересующие элементы
clean_data = json_stream.select(col("timestamp"), col("parsed_value.id").alias("id"),
col("parsed_value.action").alias("action"))
#clean_data.writeStream.format("console").outputMode("append").option("truncate",
False).start().awaitTermination()

#добавим join со статическим dataset - создаем данные
users_data = [(1,"Jimmy",18),(2,"Hank",48),(3,"Johnny",9),(4,"Erle",40)]
users = spark.createDataFrame(data=users_data,schema=users_schema)

#делаем join
#join_stream = clean_data.join(users, clean_data.id == users.id,
"left_outer").select(users.user_name, users.user_age, clean_data.action, clean_data.timestamp)
join_stream = clean_data.join(users, clean_data.id == users.id,
"left_outer").select(users.user_name, users.user_age, clean_data.action, clean_data.timestamp)
#join_stream.writeStream.format("console").outputMode("append").option("truncate",
False).start().awaitTermination()

#убираем terminate
#res_checkpoints= join_stream.writeStream.\
#format("console").\
#outputMode("append").\
#option("truncate", False).\
#option("checkpointLocation", "checkpoint/target").\
#start()

#sleep(10)
#res_checkpoints.stop()

#добавим агрегат - отображать число уникальных айдюков
stat_stream = clean_data.groupBy("id").count()
#stat_stream.writeStream.format("console").outputMode("complete").option("truncate",
False).start().awaitTermination()

join_stream_agg = stat_stream.join(users, stat_stream.id == users.id,
"left_outer").select(users.user_name, users.user_age, col('count'))

res_checkpoints_agg= join_stream_agg.writeStream.\
format("console").\
outputMode("complete").\
option("truncate", False).\
option("checkpointLocation", "checkpoint/target").\
start()\

sleep(10)
res_checkpoints_agg.stop()
```

13. Для сокращения количества логов внесена правка в конфигурационный файл **log4j2.properties** в контейнере Spark:
    - Оригинальный файл **/opt/bitnami/spark/conf/log4j2.properties.template** скопирован; в копии убрано окончание **.template**
    - В новом файле найден параметр rootLogger.level = info; ‘info’ заменен на ‘error’



Name	Note	Size	Last modified	Mode
bin			5 days ago	drwxr-xr-x
conf	MODIFIED		4 hours ago	drwxrwxr-x
checkpoint	ADDED		2 hours ago	drwxr-xr-x
fairscheduler.xml.template		1.1 kB	28 days ago	-rw-rw-r--
log4j2.properties	ADDED	3.3 kB	4 hours ago	-rw-r--r--
log4j2.properties.template		3.3 kB	28 days ago	-rw-rw-r--
metrics.properties.template		8.9 kB	28 days ago	-rw-rw-r--
spark-defaults.conf	ADDED	1.3 kB	28 days ago	-rw-rw-r--
spark-env.sh	ADDED	53 Bytes	7 hours ago	-rw-r--r--

/opt/bitnami/spark/conf/log4j2.properties

```
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.
#

# Set everything to be logged to the console
rootLogger.level = error
rootLogger.appenderRef.stdout.ref = console

```