

## Kafka Assignment

1. *Spin up dockerized containers for Kafka and Zookeeper using Kafka on Docker*
  1. Run the provided Docker Compose based Kafka cluster (kafka/docker-compose.yml):



2. *Install kafkacat*

1. Install Kafkacat via “`apt install kafkacat`” and check version:

```
(base) psalire@DESKTOP-33AVD7M:/mnt/c/users/Philip/Desktop$ kafkacat -V
kafkacat - Apache Kafka producer and consumer tool
https://github.com/edenhill/kafkacat
Copyright (c) 2014-2019, Magnus Edenhill
Version 1.5.0 (JSON, librdkafka 1.2.1 builtin.features=gzip,snappy,ssl,sasl,regex,plugins,sasl_oauthbearer)
```

2. Verify that it can connect to our Kafka cluster:

```
(base) psalire@DESKTOP-33AVD7M:/mnt/c/users/Philip/Desktop$ kafkacat -b 127.0.0.1:9092 -L
Metadata for all topics (from broker -1: 127.0.0.1:9092/bootstrap):
3 brokers:
  broker 2 at localhost:9093 (controller)
  broker 3 at localhost:9094
  broker 1 at localhost:9092
1 topics:
  topic "syslog" with 1 partitions:
    partition 0, leader 2, replicas: 2, isrs: 2
```

3. *Create a topic in the Kafka cluster using kafkacat*

1. Create a new topic:

```
(base) psalire@DESKTOP-33AVD7M:/mnt/c/users/Philip/Desktop$ kafkacat -b 127.0.0.1:9092 -t my-new-topic -P
hello new topic
```

2. Verify that the topic was created by consuming the topic:

```
(base) psalire@DESKTOP-33AVD7M:/mnt/c/users/Philip/Desktop$ kafkacat -b 127.0.0.1:9092 -t my-new-topic -C
hello new topic
% Reached end of topic my-new-topic [0] at offset 1
```

#### 4. Produce to and consume from the topic using *kafkacat*

##### 1. Run a Producer and Consumer simultaneously with *Kafkacat*:

```
(base) psalire@DESKTOP-33AVD7M:/mnt/c/users/Philip/Desktop$ kafkacat -b 127.0.0.1:9092 -t my-new-topic -P
my
new
topic topic
new new
|

2. Install kafkacat
* Install kafkacat via apt: sudo apt-get install kafkacat
* Verify that it can connect to the Kafka cluster:
  kafkacat -b 127.0.0.1:9092 -t my-new-topic -P

3. Create a topic in the Kafka cluster using kafkacat
* Create the topic:
  kafkacat -b 127.0.0.1:9092 -t my-new-topic -P

(base) psalire@DESKTOP-33AVD7M:/mnt/c/users/Philip/Desktop$ kafkacat -b 127.0.0.1:9092 -t my-new-topic -C
hello new topic
% Reached end of topic my-new-topic [0] at offset 1
my
% Reached end of topic my-new-topic [0] at offset 2
new
% Reached end of topic my-new-topic [0] at offset 3
topic topic
% Reached end of topic my-new-topic [0] at offset 4
new new
% Reached end of topic my-new-topic [0] at offset 5
^C(base) psalire@DESKTOP-33AVD7M:/mnt/c/users/Philip/Desktop$ kafkacat -b 127.0.0.1:9092 -t my-new-topic -C
% Auto-selecting Consumer mode (use -P or -C to override)
hello new topic
my
new
topic topic
new new
% Reached end of topic my-new-topic [0] at offset 5
```

## 5. Write a Java application to produce and consume from the Kafka topic using the kafka-clients directory in the repo

Often, Kafka Consumers are used to push to a database. As such, we've developed some code by modifying the kafka-clients code, which produces by reading from a CSV file and consumes to push to a SQLite database:

1. Build the source with mvn package
2. Run the Consumer with `java -cp target\kafka-client-java.jar assignment2.Consumer`
3. Run the Producer with `java -cp target\kafka-client-java.jar assignment2.Producer example.csv`

```
[main] INFO org.apache.kafka.common.utils.AppInfoParser - Kafka version: 2.5.0
[main] INFO org.apache.kafka.common.utils.AppInfoParser - Kafka commitId: 66563e712b0b9f84
[main] INFO org.apache.kafka.common.utils.AppInfoParser - Kafka startTimeMs: 1635992983525
[kafka-producer-network-thread | producer-1] INFO org.apache.kafka.clients.Metadata - [Producer clientId=producer-1] Cluster ID: VVCXK90cQ6LXKdgC4uEBA
[main] INFO org.apache.kafka.clients.producer.KafkaProducer - [Producer clientId=producer-1] Closing the Kafka producer with timeoutMillis = 922372036854775807 ms.
[kafka-producer-network-thread | producer-1] INFO assignment2.Producer - Received new metadata
Topic: csv_topic
Partition: 0
Offset: 10
Timestamp: 1635992983713
[kafka-producer-network-thread | producer-1] INFO assignment2.Producer - Received new metadata
Topic: csv_topic
Partition: 0
Offset: 11
Timestamp: 1635992983721
[kafka-producer-network-thread | producer-1] INFO assignment2.Producer - Received new metadata
Topic: csv_topic
Partition: 0
Offset: 12
Timestamp: 1635992983721
[kafka-producer-network-thread | producer-1] INFO assignment2.Producer - Received new metadata
Topic: csv_topic
Partition: 0
Offset: 13
Timestamp: 1635992983721
[kafka-producer-network-thread | producer-1] INFO assignment2.Producer - Received new metadata
Topic: csv_topic
Partition: 0
Offset: 14
Timestamp: 1635992983721
D:\Documents\OWE 272\Team-Project-31 (kafka-assignment -> origin)
^

ssl.truststore.password = null
ssl.truststore.type = JKS
value.deserializer = class org.apache.kafka.common.serialization.StringDeserializer

[main] INFO org.apache.kafka.common.utils.AppInfoParser - Kafka version: 2.5.0
[main] INFO org.apache.kafka.common.utils.AppInfoParser - Kafka commitId: 66563e712b0b9f84
[main] INFO org.apache.kafka.common.utils.AppInfoParser - Kafka startTimeMs: 1635992973461
[main] INFO org.apache.kafka.clients.consumer.KafkaConsumer - [Consumer clientId=consumer-my_app-1, groupId=my_app] Subscribed to topic(s): csv_topic
[main] INFO org.apache.kafka.clients.Metadata - [Consumer clientId=consumer-my_app-1, groupId=my_app] Cluster ID: VVCXK90cQ6LXKdgC4uEBA
[main] INFO org.apache.kafka.clients.consumer.internals.AbstractCoordinator - [Consumer clientId=consumer-my_app-1, groupId=my_app] Discovered group coordinator localhost:9092 (id: 2147483646 rack: null)
[main] INFO org.apache.kafka.clients.consumer.internals.AbstractCoordinator - [Consumer clientId=consumer-my_app-1, groupId=my_app] Join group failed with org.apache.kafka.common.errors.MemberIdRequiredException: The group member needs to have a valid member id before actually entering a consumer group
[main] INFO org.apache.kafka.clients.consumer.internals.AbstractCoordinator - [Consumer clientId=consumer-my_app-1, groupId=my_app] (Re-)joining group
[main] INFO org.apache.kafka.clients.consumer.internals.AbstractCoordinator - [Consumer clientId=consumer-my_app-1, groupId=my_app] Finished assignment for group at generation 24: (consumer-my_app-1-b985b361-112d-4d03-90c6-a075d8b712b-f-assignment(partitions=[csv_topic-0]))
[main] INFO org.apache.kafka.clients.consumer.internals.AbstractCoordinator - [Consumer clientId=consumer-my_app-1, groupId=my_app] Successfully joined group with generation 24
[main] INFO org.apache.kafka.clients.consumer.internals.ConsumerCoordinator - [Consumer clientId=consumer-my_app-1, groupId=my_app] Adding newly assigned partitions: csv_topic-0
[main] INFO org.apache.kafka.clients.consumer.internals.ConsumerCoordinator - [Consumer clientId=consumer-my_app-1, groupId=my_app] Setting offset for partition csv_topic-0 to the committed offset FetchPosition(offset=10, offsetEpoch=Optional[0], currentLeader=LeaderAndEpoch(leader=Optional[localhost:9094 (id: 3 rack: null)], epoch=0))
[main] INFO assignment2.Consumer - Inserting to SQLite db: Key: key | Value: value
[main] INFO assignment2.Consumer - Inserting to SQLite db: Key: hello | Value: world
[main] INFO assignment2.Consumer - Inserting to SQLite db: Key: this | Value: is
[main] INFO assignment2.Consumer - Inserting to SQLite db: Key: an | Value: example
[main] INFO assignment2.Consumer - Inserting to SQLite db: Key: csv | Value: file
```

(Producer on the left, Consumer on the right)

4. Query `assignment2.sqlite` that was created and see that the CSV values were inserted

The screenshot shows the SQLiteOnline.com web interface. The URL bar displays `https://sqliteonline.com`. The interface includes a navigation bar with buttons for File, Owner DB, Run, Export, and Import. Below this, a sidebar on the left lists several databases, including `assignment2.sqlite`, which is selected. The main area shows a SQL query: `SELECT * FROM assignment2`. Below the query, a table of results is displayed with two columns: `key` and `value`. The results are as follows:

key	value
key	value
hello	world
this	is
an	example
csv	file