Projekt 2 - sprawozdanie
Zestaw danych 5 - Chicago Bicycle Sharing
Kafka Streams

Projekt zawiera następujące pliki:

- **event-data-kafka-streams.jar**
  Program Kafka Streams do przetwarzania strumieni danych
- **TestProducer.jar**
  Program zasilający źródłowe tematy Kafki
- **reset-kafka.sh**
  Skrypt tworzący źródłowe tematy Kafki i resetujący środowisko
- **data-to-topic.sh**
  Skrypt uruchamiający program zasilający źródłowe tematy Kafki
- **run-program.sh**
  Skrypt uruchamiający program przetwarzania strumieni danych
- **get-result.sh**
  Skrypt odczytujący wyniki z miejsca docelowego
- **BicycleSharing.zip**
  Projekt przetwarzający strumienie danych

Uruchom projekt według poniższej instrukcji:

1. Otwórz Google Cloud Shell i uruchom klaster za pomocą polecenia:

```
gcloud dataproc clusters create ${CLUSTER_NAME} \
--enable-component-gateway --region ${REGION} --subnet default \
--master-machine-type n1-standard-2 --master-boot-disk-size 50 \
--num-workers 2 --worker-machine-type n1-standard-2 --worker-boot-disk-size 50 \
--image-version 2.1-debian11 --optional-components ZEPPELIN,ZOOKEEPER \
--project ${PROJECT_ID} --max-age=3h \
--metadata "run-on-master=true" \
--initialization-actions \
gs://goog-dataproc-initialization-actions-${REGION}/kafka/kafka.sh
```

2. Otwórz terminal SSH do węzła master. To będzie terminal nadawczy.

3. Prześlij skrypt reset-kafka.sh do terminala nadawczego.

3. Nadaj mu prawo do wykonywania: `chmod +x reset-kafka.sh`

4. Wykonaj skrypt: `./reset-kafka.sh`

5. Prześlij archiwum `bicycle_result.zip` do swojego bucketa na platformie Google Cloud.

6. Pobierz zip do terminala nadawczego:
   `gsutil cp gs://{bucket}/bicycle_result.zip bicycle_result.zip`

7. Rozpakuj dane wejściowe: unzip bicycle_result.zip

8. Prześlij program TestProducer.jar do terminala nadawczego.

9. Prześlij skrypt data-to-topic.sh do terminala nadawczego.

10. Nadaj mu prawo do wykonywania: chmod +x data-to-topic.sh

11. Wykonaj skrypt: ./data-to-topic.sh

12. Otwórz nowy terminal SSH do węzła master.

13. Prześlij do niego plik z danymi Divvy_Bicycle_Stations.csv.

14. Pobierz bibliotekę OpenCSV:
    wget https://repo1.maven.org/maven2/com/opencsv/opencsv/5.8/opencsv-5.8.jar

15. Prześlij program event-data-kafka-streams.jar.

16. Prześlij skrypt run-program.sh.

17. Nadaj mu prawo do wykonywania: chmod +x run-program.sh

18. Wykonaj skrypt i podaj parametry: delay, D, P np. ./run-program.sh A 10 7

19. Otwórz nowy terminal. To będzie odbiorca stanu ETL.

20. Prześlij do niego skrypt get-result.sh.

21. Nadaj mu prawo do wykonywania: chmod +x get-result.sh

22. Odczytaj wynik z tematu output-topic: ./get-result.sh output-topic

23. Otwórz nowy terminal, odpowiedzialny za wykrywanie anomalii.

24. Wykonaj skrypt get-result.sh.sh w nowym terminalu z parametrem alert-topic.
    ./get-result.sh alert-topic