Zajęcia zintegrowane 1

2025L

(20.03.2025)

Celem zajęć zintegrowanych jest zapoznanie się z działaniem brokerów wiadomości na przykładzie rozwiązania Kafka.

1. Uruchom przygotowaną maszyną wirtualną w VirtualBox (psd_ubuntu3_2025) i zaloguj się na konto: psd i skopuj do katalogu ~kafka/Downloads pliki tgz i jar, które są dostępne w ~psd/Downloads
2. Uruchom serwis kafka
   a. *$ su -l kafka*
   b. *$ mkdir ~/kafka*
   c. cd ~/kafka
   d. *$ tar -xvzf ~/Downloads/kafka.tgz --strip 1*
   e. *$ vi ~/kafka/config/server.properties* I dodaj dwie linie na końcu pliku

      delete.topic.enable = true
      log.dirs=/home/kafka/logs

   f. Utwórz plik /etc/systemd/system/kafka.service o zawartości:

      [Unit]
      Requires=zookeeper.service
      After=zookeeper.service

      [Service]
      Type=simple
      User=kafka
      ExecStart=/bin/sh -c '/home/kafka/kafka/bin/kafka-server-start.sh /home/kafka/kafka/config/server.properties > /home/kafka/kafka/kafka.log 2>&1'
      ExecStop=/home/kafka/kafka/bin/kafka-server-stop.sh
      Restart=on-abnormal

      [Install]
      WantedBy=multi-user.target

   g. Utwórz plik /etc/systemd/system/zookeeper.service o zawartości:

      [Unit]
      Requires=network.target remote-fs.target
      After=network.target remote-fs.target

      [Service]
      Type=simple
      User=kafka
      ExecStart=/home/kafka/kafka/bin/zookeeper-server-start.sh /home/kafka/kafka/config/zookeeper.properties
      ExecStop=/home/kafka/kafka/bin/zookeeper-server-stop.sh
      Restart=on-abnormal

      [Install]

WantedBy=multi-user.target

   h. *$ sudo systemctl start kafka*

3. Weryfikacja działania kafki
   a. *$ sudo systemctl status kafka*
   b. *~kafka/kafka/bin/kafka-topics.sh --create --bootstrap-server localhost:9092 --replication-factor 1 --partitions 1 --topic PSDTopic*
   c. *echo "Hello, World" | ~kafka/kafka/bin/kafka-console-producer.sh --broker-list localhost:9092 --topic PSDTopic > /dev/null*
   d. *~kafka/kafka/bin/kafka-console-consumer.sh --bootstrap-server localhost:9092 --topic PSDTopic --from-beginning*
4. Uruchom i przeanalizuj wykorzystanie brokera kafki za pomocą kafdrop.
   java --add-opens=java.base/sun.nio.ch=ALL-UNNAMED -jar kafdrop-3.29.0.jar --kafka.brokerConnect=localhost:9092
5. Napisz przykładowego producenta i konsumenta. Można wzorować się na załączonym materiale, ale wskazane są własne rozszerzenia.
6. Napisz krótkie sprawozdanie (1-2 strony)

Materiały pomocnicze:

https://kafka.apache.org/

https://github.com/obsidiandynamics/kafdrop

PRODUCENT

```
import time
import json
import random
from datetime import datetime
from data_generator1 import generate_message1
from kafka import KafkaProducer

# Messages will be serialized as JSON
def serializer(message):
    return json.dumps(message).encode('utf-8')

# Kafka Producer
producer = KafkaProducer(
    bootstrap_servers=['localhost:9092'],
    value_serializer=serializer
)

if __name__ == '__main__':
    # Infinite loop - runs until you kill the program
    while True:
        # Generate a message
        dummy_message = generate_message1()

        # Send it to our 'messages' topic
        print(f'Producing message @ {datetime.now()} | Message =
{str(dummy_message)}')
        producer.send('messages', dummy_message)

        # Sleep for a random number of seconds
        time_to_sleep = random.randint(1, 11)
        time.sleep(time_to_sleep)
```

GENERATOR DANYCH

```
import random
import string

user_ids = list(range(1, 101))

def generate_message1() -> dict:
    random_user_id = random.choice(user_ids)
    # Generate a random message
    temp=str(random.randint(-15,30))
    message = ''.join(temp)
    return {
        'user_id': random_user_id,
        'message': message
    }
```

KONSUMENT

```
import json
from kafka import KafkaConsumer

if __name__ == '__main__':
        # Kafka Consumer
        consumer = KafkaConsumer(
        'messages',
        bootstrap_servers='localhost:9092',
        auto_offset_reset='earliest'
        )
        for message in consumer:
                print(json.loads(message.value))
```