1. **Cấu hình Kafka trên Docker**

zookeeper:

    image: confluentinc/cp-zookeeper:latest

    environment:

      ZOOKEEPER\_CLIENT\_PORT: 2181

      ZOOKEEPER\_TICK\_TIME: 2000

      ZOOKEEPER\_INIT\_LIMIT: 5

      ZOOKEEPER\_SYNC\_LIMIT: 2

    ports:

      - “2181:2181”

    container\_name: zookeeper\_1

    healthcheck:

      test: nc -z zookeeper 2181 || exit -1

      start\_period: 15s

      interval: 5s

      timeout: 10s

      retries: 10

  broker\_01:

    image: confluentinc/cp-kafka:latest

    depends\_on:

      zookeeper:

        condition: service\_healthy

    container\_name: broker\_01

    environment:

      KAFKA\_BROKER\_ID: 1

      KAFKA\_ZOOKEEPER\_CONNECT: zookeeper:2181

      KAFKA\_LISTENERS: PLAINTEXT\_HOST://broker\_01:9092,PLAINTEXT://broker\_01:29092

      KAFKA\_ADVERTISED\_LISTENERS: PLAINTEXT\_HOST://localhost:9092,PLAINTEXT://broker\_01:29092

      KAFKA\_LISTENER\_SECURITY\_PROTOCOL\_MAP: PLAINTEXT:PLAINTEXT,PLAINTEXT\_HOST:PLAINTEXT

      KAFKA\_INTER\_BROKER\_LISTENER\_NAME: PLAINTEXT

      KAFKA\_JMX\_PORT: 9090

      KAFKA\_LOG\_DIRS: /var/lib/kafka/data

      KAFKA\_NUM\_PARTITIONS: 2

      KAFKA\_AUTO\_CREATE\_TOPICS\_ENABLE: “true”

      KAFKA\_OFFSETS\_TOPIC\_REPLICATION\_FACTOR: 1

      KAFKA\_TRANSACTION\_STATE\_LOG\_MIN\_ISR: 1

      KAFKA\_TRANSACTION\_STATE\_LOG\_REPLICATION\_FACTOR: 1

      KAFKA\_GROUP\_INITIAL\_REBALANCE\_DELAY\_MS: 100

      CONFLUENT\_METRICS\_ENABLE: ‘false’

    ports:

      - 9092:9092

    healthcheck:

      test: nc -z broker\_01 9092 || exit -1

      start\_period: 15s

      interval: 5s

      timeout: 10s

      retries: 10

  broker\_02:

      image: confluentinc/cp-kafka:latest

      depends\_on:

        zookeeper:

          condition: service\_healthy

      container\_name: broker\_02

      environment:

        KAFKA\_BROKER\_ID: 2

        KAFKA\_ZOOKEEPER\_CONNECT: zookeeper:2181

        KAFKA\_LISTENERS: PLAINTEXT\_HOST://broker\_02:9094,PLAINTEXT://broker\_02:29092

        KAFKA\_ADVERTISED\_LISTENERS: PLAINTEXT\_HOST://localhost:9094,PLAINTEXT://broker\_02:29092

        KAFKA\_LISTENER\_SECURITY\_PROTOCOL\_MAP: PLAINTEXT:PLAINTEXT,PLAINTEXT\_HOST:PLAINTEXT

        KAFKA\_INTER\_BROKER\_LISTENER\_NAME: PLAINTEXT

        KAFKA\_JMX\_PORT: 9090

        KAFKA\_LOG\_DIRS: /var/lib/kafka/data

        KAFKA\_NUM\_PARTITIONS: 2

        KAFKA\_AUTO\_CREATE\_TOPICS\_ENABLE: “true”

        KAFKA\_OFFSETS\_TOPIC\_REPLICATION\_FACTOR: 1

        KAFKA\_TRANSACTION\_STATE\_LOG\_MIN\_ISR: 1

        KAFKA\_TRANSACTION\_STATE\_LOG\_REPLICATION\_FACTOR: 1

        KAFKA\_GROUP\_INITIAL\_REBALANCE\_DELAY\_MS: 100

        CONFLUENT\_METRICS\_ENABLE: ‘false’

      ports:

        - 9094:9094

      healthcheck:

        test: nc -z broker\_02 9094 || exit -1

        start\_period: 15s

        interval: 5s

        timeout: 10s

        retries: 10

  kafka-ui:

    container\_name: kafka-ui\_1

    image: provectuslabs/kafka-ui:latest

    ports:

      - 8080:8080

    depends\_on:

      broker\_01:

        condition: service\_healthy

      broker\_02:

        condition: service\_healthy

    environment:

      KAFKA\_CLUSTERS\_0\_NAME: vdt-kafka-cluster

      KAFKA\_CLUSTERS\_0\_BOOTSTRAPSERVERS: broker\_01:29092,broker\_02:29092

      KAFKA\_CLUSTERS\_0\_METRICS\_PORT: 9090

    healthcheck:

      test: nc -z kafka-ui 8080 || exit -1

      start\_period: 15s

      interval: 5s

      timeout: 10s

      retries: 10

1. Thực thi   
   \_ Chạy lệnh docker-compose up -d để tiến hành chạy các container cần thiết.
2. Kiểm tra kết quả  
   \_B1: Kiểm tra kafka, vào **http://localhost:8080** nếu như thu được kết quả như hình là chạy thành công A screenshot of a computer

   Description automatically generated

KAFKA CONSUMER

* + - Tạo Kafka consumer trên NIFI

\_ Giả sử như chúng ta đã có messenge sẵn trên kafka và cần consume theo từ csv và chuyến hoá thành parquet chúng ta cấu hình như sau:

Component KafkaConsume\_2\_6  
\_ B1: Tạo component kafkaconsume\_2\_6(vì chúng ta đang sử dụng bản kafka có version > 3) sau đó thiết lập các thông số dữ liệu gồm kafka\_brokers, topic name, group id như hình sau:  
A screenshot of a computer

Description automatically generated  
\_ B2: Kiểm tra cấu hình đã đúng hay chưa bằng cách nhấp vào dấu tích ở phía trên tay phải của cấu hình, nếu như cấu hình đúng, kết quả sẽ trả ra như sau:  
A close-up of a white background

Description automatically generated

* Tạo Kafka consumer trên java project

\_ B1: Khởi tạo Maven Project và thêm các dependency liên quan, chúng ta sẽ dùng IDEA intelliJ để tạo một maven project, sau dó vào file “pom.xml” để thêm các dependency cần thiết như sau:

A computer screen with white text

Description automatically generated

\_B2: Thiết lập các tham số, chúng ta sẽ khai báo các tham số thiết yếu sau, đảm bảo các giá trị của chúng đúng với giá trị đã được tạo trên kafka.

    private static final String TOPIC\_NAME = "orders";

    private static final String BOOTSTRAP\_SERVERS = "localhost:9092,localhost:9094";

    private static final String GROUP\_ID = "vdt-orders";

\_B3: Khai báo các thuộc tính và logic xử lí dữ liệu

    public static void main(String[] args) {

        Properties props = new Properties();

        props.put(ConsumerConfig.BOOTSTRAP\_SERVERS\_CONFIG, BOOTSTRAP\_SERVERS);

        props.put(ConsumerConfig.GROUP\_ID\_CONFIG, GROUP\_ID);

        props.put(ConsumerConfig.KEY\_DESERIALIZER\_CLASS\_CONFIG, StringDeserializer.class.getName());

        props.put(ConsumerConfig.VALUE\_DESERIALIZER\_CLASS\_CONFIG, StringDeserializer.class.getName());

        props.put(ConsumerConfig.AUTO\_OFFSET\_RESET\_CONFIG, "earliest");

        try (Consumer<String, String> consumer = new KafkaConsumer<>(props)) {

            consumer.subscribe(Collections.singletonList(TOPIC\_NAME));

            boolean stopFlags = true;

            while (stopFlags) {

                ConsumerRecords<String, String> records = consumer.poll(Duration.ofMillis(100));

                // Process each record

                records.forEach(record -> {

                    String csvData = record.value();

                    try (CSVReader csvReader = new CSVReader(new StringReader(csvData))) {

                        String[] fields = csvReader.readNext();

                        // Process the fields or perform operations on the CSV data

                        for (String field : fields) {

                            System.out.print(field + " ");

                        }

                        System.out.println();

                    } catch (Exception e) {

                        e.printStackTrace();

                    }

                });

            }

        } catch (Exception e) {

            e.printStackTrace();

        }

\_B4: Khởi chạy và kết quả: Nếu như chúng ta cấu hình đúng thì chúng ta sẽ thu dược kết quả như trên console sau:

A black screen with white text

Description automatically generated