Quang Phung – Assignment Report

1. Download and Install:

1.1. Apache Kafka:

- Visit https://kafka.apache.org/downloads
- Download the latest binary file (kafka_2.12-3.7.0.tgz)

1.2. Apache Nifi:

- Visit https://nifi.apache.org/download/
- Download the latest binary file (NiFi Standard 2.0.0-M3)

1.3. Apache Hadoop:

- Visit https://hadoop.apache.org/releases.html
- Download the latest binary file (Version 3.4.0 for aarch64)

1.4. Apache Spark:

- Visit https://spark.apache.org/downloads.html
- Download the latest binary file (spark-3.5.1-bin-hadoop3.tgz)

1.5. Java:

- Visit https://www.oracle.com/java/technologies/downloads/
- Download the latest binary file (ARM64 Compressed Archive)

1.6. Python:

- Visit https://www.python.org/downloads/
- Download the latest binary file (Version 3.12.4)

2. Set-Up and Run:

2.1. Apache Kafka:

- **Step 1: GET KAFKA**

  $ tar -xzf kafka_2.13-3.7.0.tgz

  $ cd kafka_2.13-3.7.0
- **Step 2:**

NOTE: Your local environment must have Java 8+ installed.

Kafka with ZooKeeper

Run the following commands to start all services in the correct order:

Start the ZooKeeper service

$ bin/zookeeper-server-start.sh config/zookeeper.properties

Open another terminal session and run:

Start the Kafka broker service

$ bin/kafka-server-start.sh config/server.properties

Once all services have successfully launched, you will have a basic Kafka environment running and ready to use.

- **Step 3: CREATE A TOPIC TO STORE YOUR EVENTS**

  Kafka is a distributed event streaming platform that lets you read, write, store, and process events (also called records or messages in the documentation) across many machines.

  Example events are payment transactions, geolocation updates from mobile phones, shipping orders, sensor measurements from IoT devices or medical equipment, and much more. These events are organized and stored in topics. Very simplified, a topic is similar to a folder in a filesystem, and the events are the files in that folder.

  So before you can write your first events, you must create a topic. Open another terminal session and run:

$ bin/kafka-topics.sh –create –topic quickstart-events –bootstrap-server localhost:9092

- **Step 4: Run the Producer.py file on github to push the message on Kafka**

Download the independency:

$ pip install kafka

And Run the file **producer.py**

2.2. Apache Nifi:

- **Turn on Nifi:**

  $ cd path/to/nifi

  $ ./nifi.sh start

- **Get UserName and Password:**

  $ ./bin/nifi.sh set-single-user-credentials <username> <password>

- **Open: https://127.0.0.1:8080/nifi/**

- **Nifi Flow Configuration:**
  - **Kafka Consumer Processor:** To consume data from Kafka topic vdt2024
  - **Convert Record Processor:** To convert JSON data to Avro format
  - **PutHDFS Processor:** To store the data in HDFS at /raw_zone/fact/activity in Parquet format

- **Create Flow:**
  - Drag and drop a ConsumeKafkaRecord_2_0 processor.
  - Drag and drop a ConvertRecord processor to convert JSON to Parquet.
  - Drag and drop a PutHDFS processor to write to HDFS.

- **Configure Processors:**

- **ConsumeKafkaRecord_2_0:**
  - **Kafka Brokers:** localhost:9092
  - **Topic Name:** vdt2024
  - **Value Deserializer:** org.apache.kafka.common.serialization.StringDeserializer

Configure Processor | ConsumeKafkaRecord_2_0 1.26.0

Stopped

SETTINGS | SCHEDULING | PROPERTIES | RELATIONSHIPS | COMMENTS

Required field

Property	Value
Kafka Brokers	**localhost:9092**
Topic Name(s)	**vdt2024**
Topic Name Format	**names**
Record Reader	**JsonTreeReader**
Record Writer	**ParquetRecordSetWriter**
Honor Transactions	**true**
Security Protocol	**PLAINTEXT**
SASL Mechanism	**GSSAPI**
Kerberos Credentials Service	No value set
Kerberos Service Name	No value set
Kerberos Principal	No value set
Kerberos Keytab	No value set

Configure Processor | ConsumeKafkaRecord_2_0 1.26.0

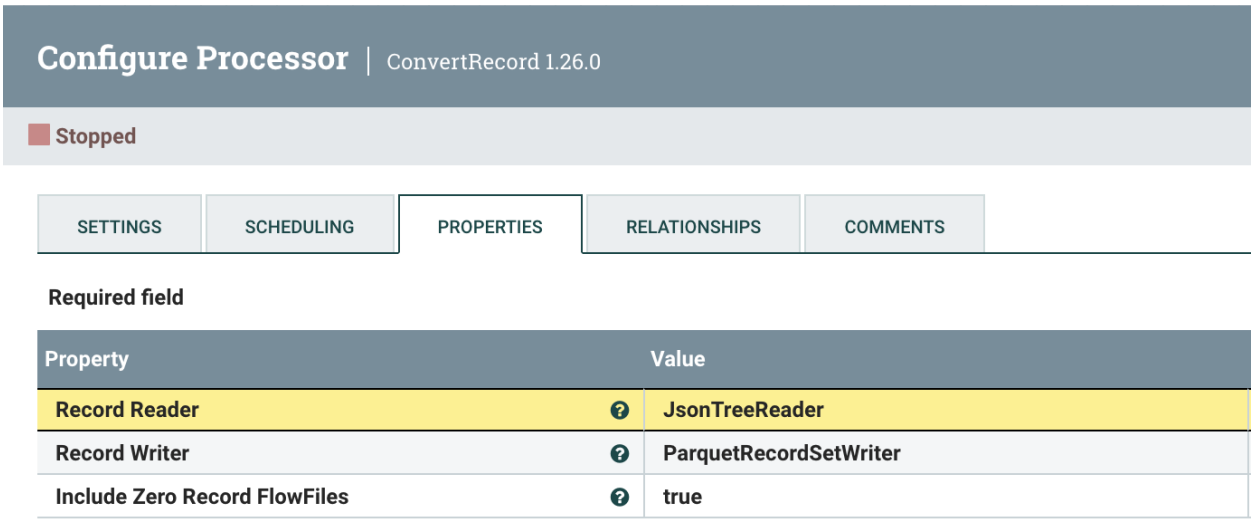
Stopped

SETTINGS | SCHEDULING | PROPERTIES | RELATIONSHIPS | COMMENTS

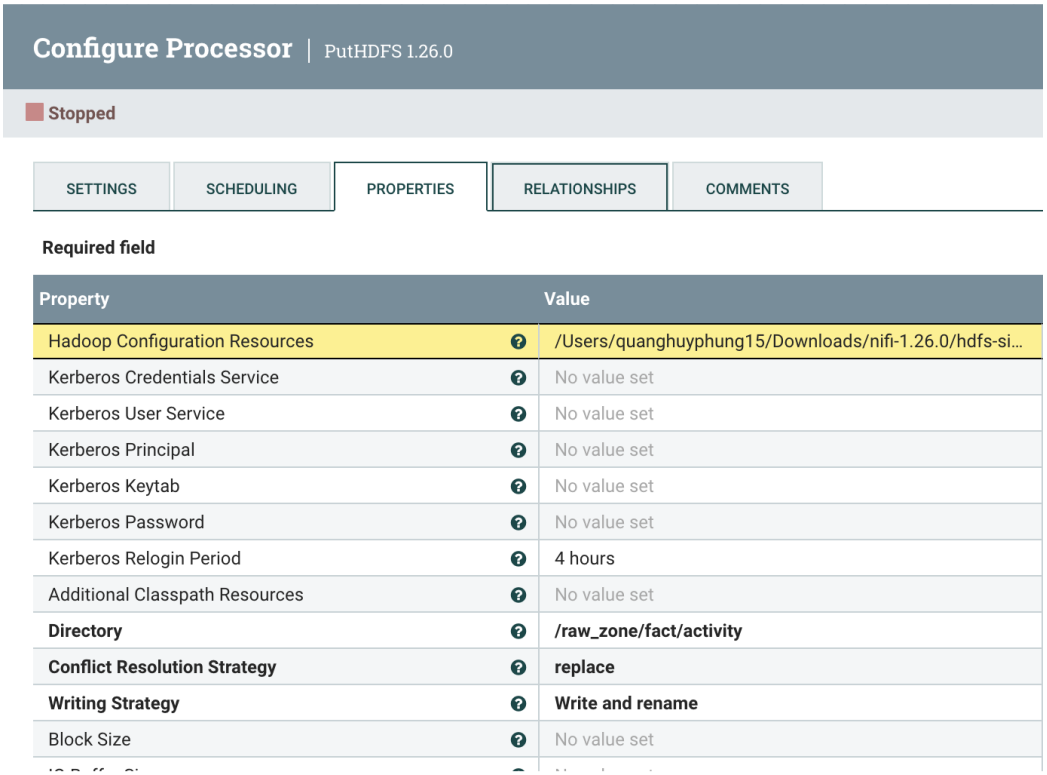
Required field

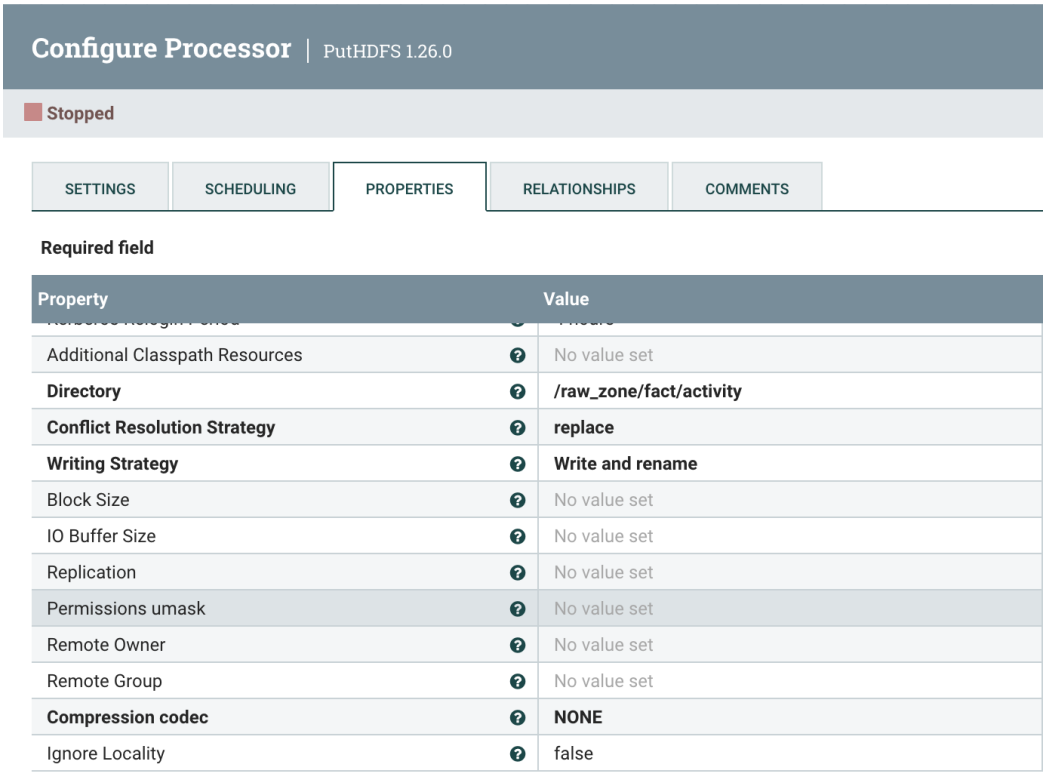
Property	Value
Kerberos Principal	No value set
Kerberos Keytab	No value set
SSL Context Service	No value set
Group ID	**nifi-consumer-group**
Separate By Key	false
Key Attribute Encoding	**UTF-8 Encoded**
Offset Reset	**latest**
Message Header Encoding	UTF-8
Headers to Add as Attributes (Regex)	No value set
Max Poll Records	10000
Max Uncommitted Time	1 secs
Communications Timeout	**60 secs**

- **ConvertRecord:**
    - **Record Reader:** JsonTreeReader
    - **Record Writer:** ParquetRecordSetWriter

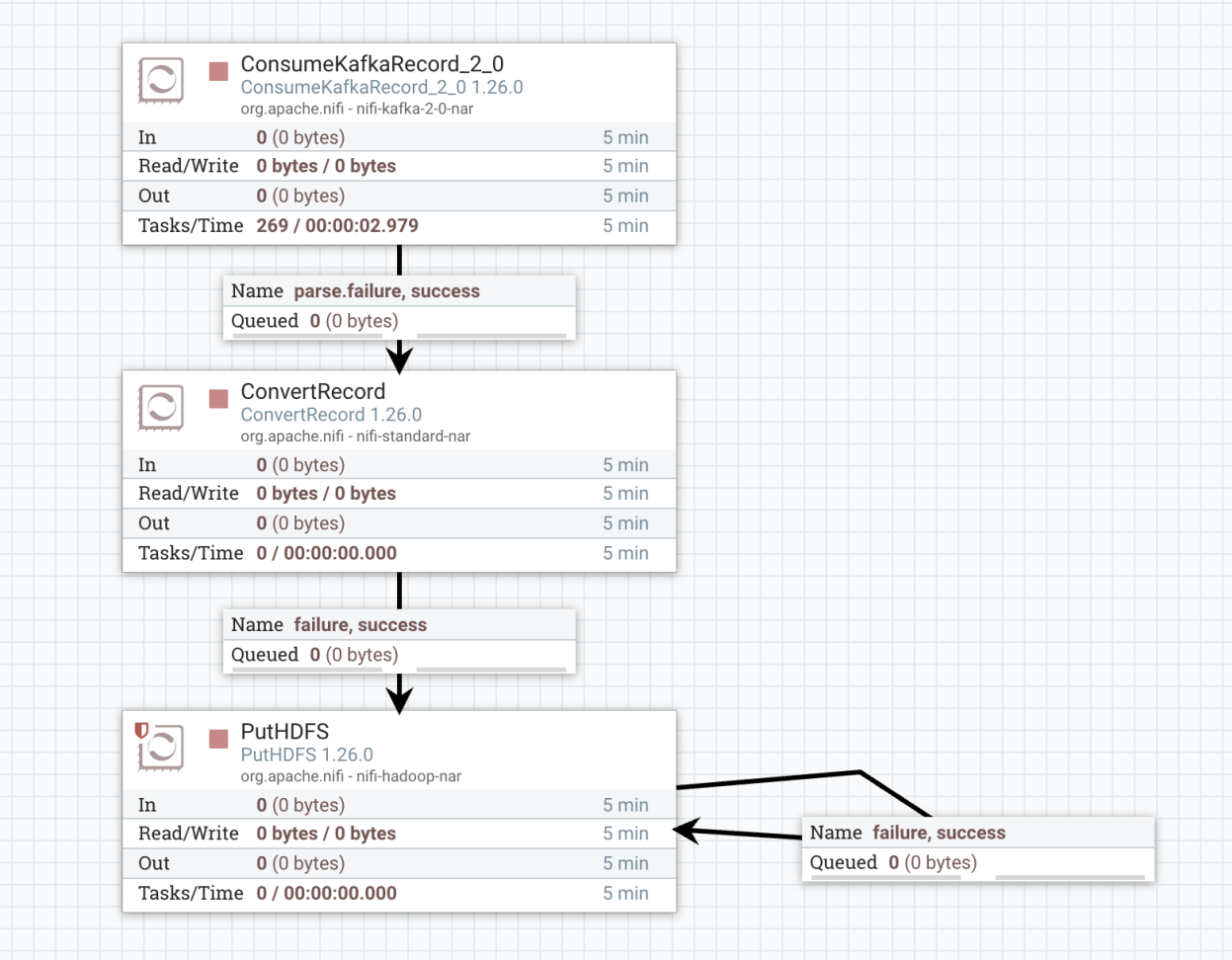


- **PutHDFS:**
    - **Directory:** /raw_zone/fact/activity
    - **File Type:** parquet


Configure Processor | PutHDFS 1.26.0
Stopped
SETTINGS
SCHEDULING
PROPERTIES
RELATIONSHIPS
COMMENTS
Required field
Property
Value
Hadoop Configuration Resources
/Users/quanghuyphung15/Downloads/nifi-1.26.0/hdfs-si...
Kerberos Credentials Service
No value set
Kerberos User Service
No value set
Kerberos Principal
No value set
Kerberos Keytab
No value set
Kerberos Password
No value set
Kerberos Relogin Period
4 hours
Additional Classpath Resources
No value set
Directory
/raw_zone/fact/activity
Conflict Resolution Strategy
replace
Writing Strategy
Write and rename
Block Size
No value set


Configure Processor | PutHDFS 1.26.0
Stopped
SETTINGS
SCHEDULING
PROPERTIES
RELATIONSHIPS
COMMENTS
Required field
Property
Value
Additional Classpath Resources
No value set
Directory
/raw_zone/fact/activity
Conflict Resolution Strategy
replace
Writing Strategy
Write and rename
Block Size
No value set
IO Buffer Size
No value set
Replication
No value set
Permissions umask
No value set
Remote Owner
No value set
Remote Group
No value set
Compression codec
NONE
Ignore Locality
false

- **Overall:**



2.3. Apache Hadoop:

- **Clone Hadoop docker from this repo:**

  https://github.com/big-data-europe/docker-hadoop
- **Run using:**

  $ docker-compose up
- **Visit: http://localhost:9870/**
- **Upload file: "danh_sach_sv_de.csv" to "/raw_zone/fact/activity"**

2.4. Apache Spark:

- **Go to the downloaded Spark file**

  $ cd path/to/spark
- **Create python environment:**

  $ python -m venv .pyspark-env

$ source .pyspark-env/bin/activate

- **Install pyspark and jupyterlab:**

  $ pip install pyspark

  $ pip install findspark

  $ pip install jupyterlab

- **Launch Jupyterlab**

  $ jupyter-lab

- **Run the data-processing.py code**