



TẬP ĐOÀN CÔNG NGHIỆP - VIỄN THÔNG QUÂN ĐỘI

BÁO CÁO ASSIGNMENT

TRẦN NGỌC BẢO

tnbao120603@gmail.com.

Chương trình Viettel Digital Talent 2024

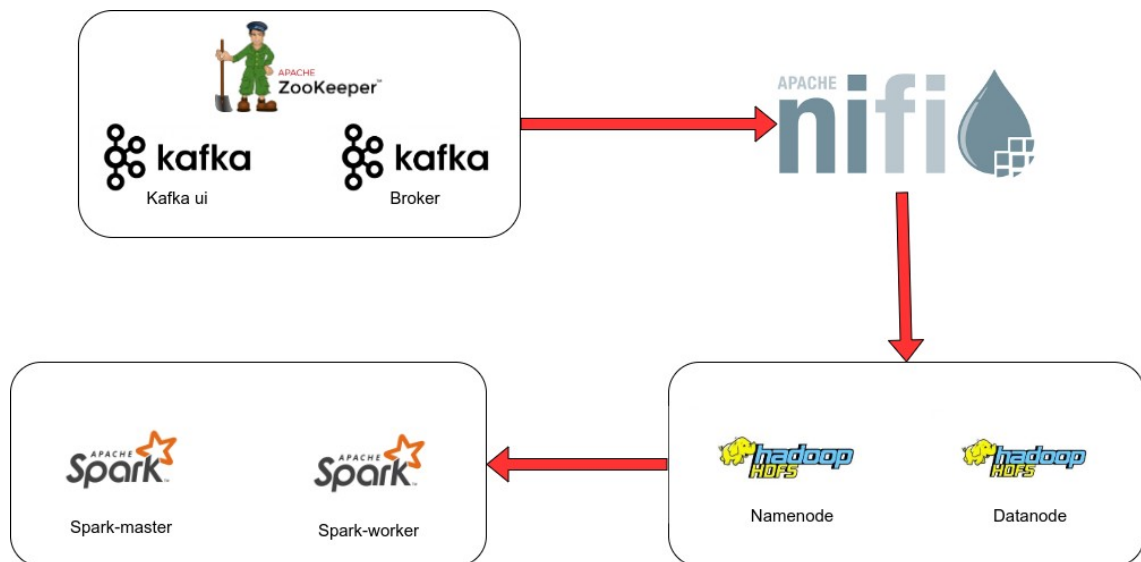
Lĩnh vực: Data Engineering

HÀ NỘI, 06/2024

CHƯƠNG 1. Mô tả cách thức triển khai

Triển khai trên môi trường Docker, cụ thể từng phần như sau:

- Data Source: Triển khai một cụm Kafka gồm có 1 Zookeeper để quản lý cụm Kafka, 1 Broker là thành phần chính của cụm để lưu trữ dữ liệu, 1 Kafka-ui để dễ dàng tương tác với cụm trên nền tảng website.
- Data Ingestion: Triển khai một node Nifi để chuyển tiếp dữ liệu từ Kafka tới HDFS của Hadoop.
- Data Storage: Triển khai một cụm Hadoop gồm có 1 Namenode và 1 DataNode để lưu trữ dữ liệu.
- Data Processing: Triển khai một cụm Spark gồm có 1 Spark-master và 1 Spark-worker để xử lý dữ liệu



Kết quả sau khi khởi động hệ thống bằng Docker compose:

```
baob@bao-Latitude-3520: ~/GitHub/IntroductionToDataEngineering$ docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS
18727b7f6c8d8  provectuslabs/kafka-ut             "/bin/sh -c 'java -..." 46 seconds ago Up 22 seconds (healthy) 0.0.0.0:8282->8080/tcp, :::8282->8080/tcp
b7ac5d5eb274   confluentinc/cp-kafka               "/etc/confluent/dock..." 46 seconds ago Up 33 seconds (healthy) 0.0.0.0:9092->9092/tcp, :::9092->9092/tcp
f790f5cffa39   confluentinc/cp-zookeeper           "/etc/confluent/dock..." 46 seconds ago Up 45 seconds (healthy) 2888/tcp, 0.0.0.0:2181->2181/tcp, :::2181->2181/tcp, 3888/tcp
9268bbdchf1c   zookeeper                           ".../scripts/start.sh"    46 seconds ago Up 45 seconds (healthy) 8080/tcp, 8080/tcp, 8443/tcp, 10000/tcp, 0.0.0.0:8181->8181/tcp, :::8181->8181/tcp
5630ea528a6a   apache/nifi                         "/entrypoint.sh /run..." 46 seconds ago Up 45 seconds (healthy) 0.0.0.0:9000->9000/tcp, :::9000->9000/tcp, 0.0.0.0:9870->9870/tcp, :::9870->9870/tcp
b1906e1c74d1   bde2020/hadoop-namenode:2.0.0-hadoop3.2.1-java8 "/entrypoint.sh /run..." 46 seconds ago Up 45 seconds (healthy) 9864/tcp
f3ffe3e83aff   bde2020/hadoop-datanode:2.0.0-hadoop3.2.1-java8 "/entrypoint.sh /run..." 46 seconds ago Up 45 seconds (healthy) 9864/tcp
9134029669fe   bitnami/spark-master                "/opt/bitnami/script..." 46 seconds ago Up 45 seconds (healthy) 0.0.0.0:8080->8080/tcp, :::8080->8080/tcp
9134029669fe   bitnami/spark-worker                "/opt/bitnami/script..." 46 seconds ago Up 45 seconds (healthy) 0.0.0.0:8080->8080/tcp, :::8080->8080/tcp
baob@bao-Latitude-3520: ~/GitHub/IntroductionToDataEngineering$
```

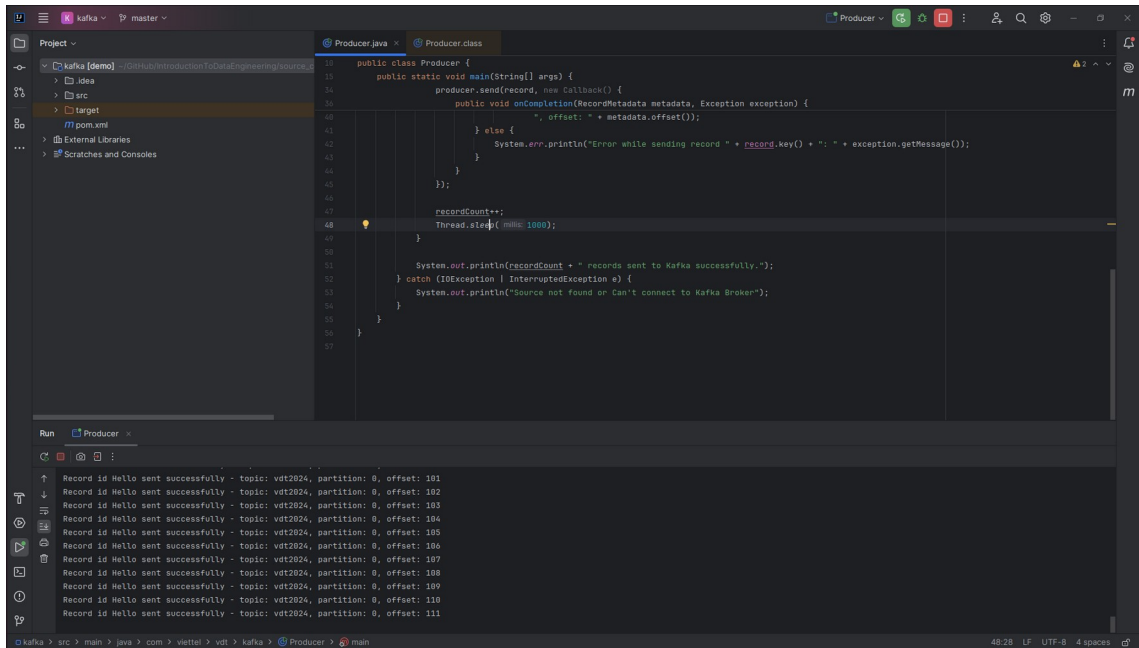
Link github:

<https://github.com/Tran-Ngoc-Bao/IntroductionToDataEngineering>

CHƯƠNG 2. THỰC HIỆN CÁC BƯỚC

2.1 Đẩy dữ liệu lên Kafka Topic

Viết một chương trình Java đọc từng dòng từ file log_action.csv và đẩy dữ liệu từng dòng theo định dạng string với dữ liệu mỗi trường cách nhau bởi dấu phẩy. Sau đó đẩy dữ liệu lên Kafka Topic vdt2024 theo từng giây.



Kết quả thu được hiển thị trên Kafka-ui:

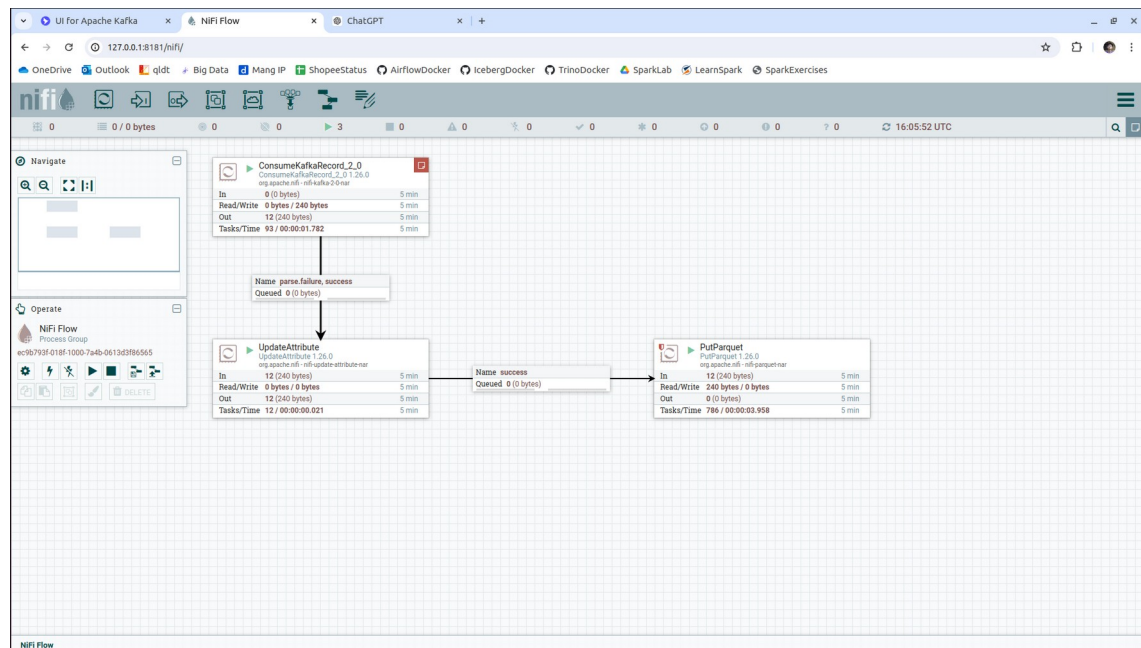
The screenshot shows the Kafka UI for the 'vdt2024' topic. The table displays the following data:

Partition	Offset	Timestamp	Message	Source
80	0	13:58:31 7/6/2024	Hello	12,execute,10,6/11/2024
81	0	13:58:32 7/6/2024	Hello	5,execute,8,6/13/2024
82	0	13:58:33 7/6/2024	Hello	33,write,9,6/11/2024
83	0	13:58:34 7/6/2024	Hello	17,write,4,6/13/2024
84	0	13:58:35 7/6/2024	Hello	1,read,8,6/10/2024
85	0	13:58:36 7/6/2024	Hello	25,write,5,6/10/2024
86	0	13:58:37 7/6/2024	Hello	34,execute,4,6/13/2024
87	0	13:58:38 7/6/2024	Hello	9,write,8,6/12/2024
88	0	13:58:39 7/6/2024	Hello	5,write,8,6/13/2024
89	0	13:58:40 7/6/2024	Hello	33,write,3,6/10/2024
90	0	13:58:41 7/6/2024	Hello	16,read,6,6/13/2024
91	0	13:58:42 7/6/2024	Hello	25,write,4,6/12/2024
92	0	13:58:43 7/6/2024	Hello	28,write,9,6/10/2024
93	0	13:58:44 7/6/2024	Hello	8,write,5,6/14/2024
94	0	13:58:45 7/6/2024	Hello	28,execute,1,6/10/2024
95	0	13:58:46 7/6/2024	Hello	30,read,8,6/12/2024
96	0	13:58:47 7/6/2024	Hello	8,read,3,6/14/2024
97	0	13:58:48 7/6/2024	Hello	20,write,8,6/14/2024
98	0	13:58:49 7/6/2024	Hello	26,execute,6,6/13/2024
99	0	13:58:50 7/6/2024	Hello	17,read,9,6/12/2024

2.2 Triển khai Nifi, kéo dữ liệu từ Kafka Topic vdt2024, xử lý và lưu dữ liệu dưới dạng parquet xuống HDFS

Xây dựng một luồng xử lý dữ liệu đơn giản bằng Nifi, gồm có các Processor sau:

- ConsumeKafkaRecord_2_0: Lấy dữ liệu real time từ Kafka và chuyển tiếp đi.
- UpdateAttribute: Giúp chuyển định dạng dữ liệu thô từ Kafka sang định dạng file parquet, rồi sau đó chuyển tiếp đi.
- PutParquet: Lưu trữ file parquet vào HDFS được cấu hình trong processor.

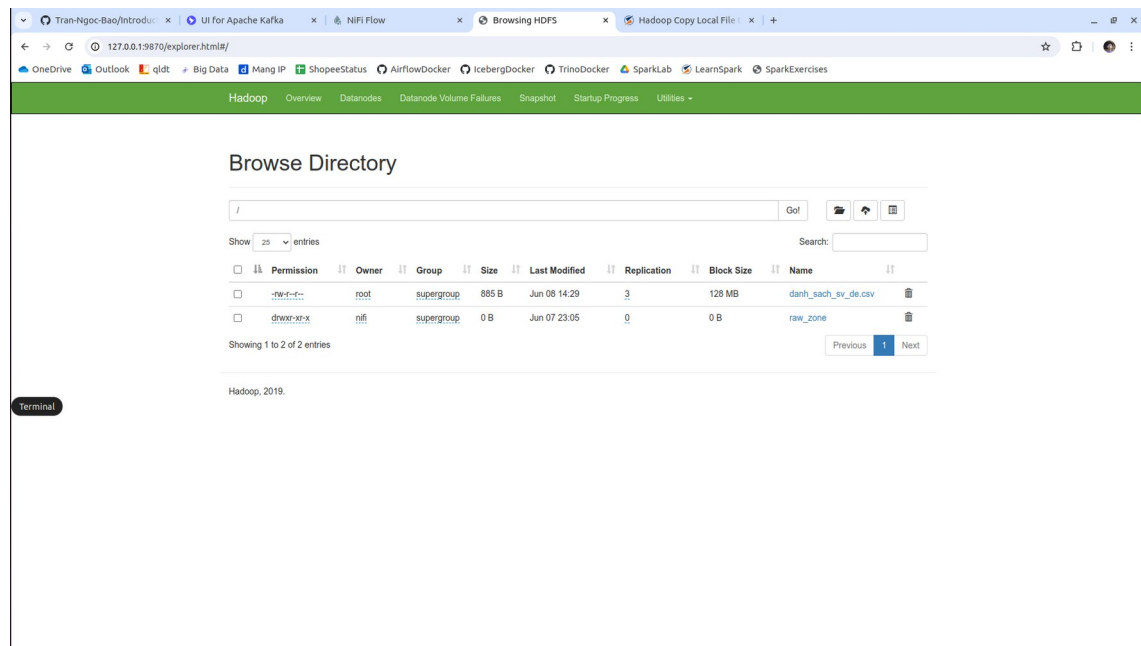


Kết quả thu được trên Hadoop:

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
-rwxr-xr-x	nifi	supergroup	1.07 KB	Jun 07 23:07	3	128 MB	f1d85b1f-6895-4cb3-887c-96aa81b719c2
-rwxr-xr-x	nifi	supergroup	1.1 KB	Jun 07 23:05	3	128 MB	118cc9ee-e441-4d26-a920-4a35ae0c41f2
-rwxr-xr-x	nifi	supergroup	1.07 KB	Jun 07 23:05	3	128 MB	187de224-e63d-4502-8738-b2960dca874b
-rwxr-xr-x	nifi	supergroup	1.1 KB	Jun 07 23:06	3	128 MB	1fb4d6dc-c90e-4994-8bb1-55e6d7fac65b
-rwxr-xr-x	nifi	supergroup	1.1 KB	Jun 07 23:06	3	128 MB	22c4a037-256e-43e3-a762-91c7ba8d6b18
-rwxr-xr-x	nifi	supergroup	1.1 KB	Jun 07 23:06	3	128 MB	29238c99-86bf-477e-9688-44ea668c0e0
-rwxr-xr-x	nifi	supergroup	1.08 KB	Jun 07 23:06	3	128 MB	2aa19378-2c9f-485e-98e6-6d47f5ada300
-rwxr-xr-x	nifi	supergroup	1.08 KB	Jun 07 23:07	3	128 MB	2ba8888e-0e57-46ed-b7dd-793775740a12
-rwxr-xr-x	nifi	supergroup	1.08 KB	Jun 07 23:05	3	128 MB	2cc06b41-df4e-42d5-996e-3aa165e622ba
-rwxr-xr-x	nifi	supergroup	1.1 KB	Jun 07 23:06	3	128 MB	3766313d-082f-4126-aa65-6d0c2ae479dc
-rwxr-xr-x	nifi	supergroup	1.08 KB	Jun 07 23:06	3	128 MB	3a2d1140-997e-4139-b3dc-f9db35786775
-rwxr-xr-x	nifi	supergroup	1.1 KB	Jun 07 23:07	3	128 MB	3f373b12-675c-4ddb-8455-8780410387e5
-rwxr-xr-x	nifi	supergroup	1.1 KB	Jun 07 23:06	3	128 MB	3f982b7c-7834-4515-9653-ca88a7630789
-rwxr-xr-x	nifi	supergroup	1.07 KB	Jun 07 23:05	3	128 MB	42e61962-75ea-479b-824c-9741c61363b
-rwxr-xr-x	nifi	supergroup	1.1 KB	Jun 07 23:06	3	128 MB	45cf9319-800c-4f9a-acaf-297753d8d640
-rwxr-xr-x	nifi	supergroup	1.08 KB	Jun 07 23:06	3	128 MB	474d6f9f-fa6d-4a6a-ba7b-415d17bdc095

2.3 Lưu trữ file danh_sach_sv_de.csv xuống HDFS

Đẩy trực tiếp file danh_sach_sc_de.csv từ local vào HDFS bằng câu lệnh trong terminal. Kết quả thu được:



2.4 Sử dụng Apache Spark xử lý dữ liệu lưu trữ dưới HDFS

Viết một chương trình Pyspark để xử lý dữ liệu với chi tiết mã nguồn như sau:

- Đọc file danh_sach_sv_de.csv
- Lọc thông tin cá nhân theo mã số sinh viên và đổi lại header cho dễ nhớ
- Đọc tất cả các file parquet được lưu ở đường dẫn /raw_zone/fact/activity trong HDFS
- Lọc log theo mã số sinh viên và tính tổng số file thực hiện theo từng phân loại
- Join 2 dataframe lại với nhau với khóa mã số sinh viên
- Xử lý dữ liệu thời gian theo đúng định dạng
- Lưu file dưới định dạng csv với các yêu cầu đề ra

```
process.py - IntroductionToDataEngineering - Visual Studio Code
File Edit Selection View Go Run Terminal Help
source_code > spark > process.py > ...
1 from pyspark.sql.functions import *
2 from pyspark.sql.session import SparkSession
3 from pyspark.context import SparkContext
4
5 if __name__ == "__main__":
6     sc = SparkContext("spark://spark-master:7077", "VDT2024")
7
8     spark = SparkSession(sc)
9
10    df_ds = spark.read.csv("hdfs://namenode:9000/danh_sach_sv_de.csv")
11
12    df_me = df_ds.filter(col("_c0") == "5").select(col("_c0").cast("Integer").alias("student_code"), col("_c1").alias("student_name"))
13
14    df_log_action_ds = spark.read.parquet("hdfs://namenode:9000/raw_zone/fact/activity/**")
15
16    df_log_action_me = df_log_action_ds.filter(col("student_code") == 5).groupBy("student_code", "timestamp", "activity").agg(sum("numberOfFile").alias("totalFile"))
17
18    df_result = df_me.join(df_log_action_me, ["student_code"], "inner")
19
20    df_result_tmp = df_result.withColumn("timestamp", when(substring(col("timestamp"), 2, 1) == "/", concat(lit("0"), col("timestamp"))).otherwise(col("timestamp")))
21
22    df_result_tmp2 = df_result_tmp.withColumn("timestamp", when(substring(col("timestamp"), 5, 1) == "/", concat(substring(col("timestamp"), 1, 3), lit("0"), substring(col("timestamp"), 4, 6))).otherwise(col("timestamp")))
23
24    df_result_final = df_result_tmp2.withColumn("timestamp", concat(substring(col("timestamp"), 7, 4), substring(col("timestamp"), 1, 2), substring(col("timestamp"), 4, 2))).orderBy(col("timestamp"), col("student_code"))
25
26    df_result_final.write.option("header", "true").option("delimiter", ";").mode("overwrite").csv("hdfs://namenode:9000/Tran_Ngoc_Bao")
27
```

Kết quả khi submit file Pyspark lên Spark-master:

AnalyzeGameD... UI for Apache K... NIFI Flow... Browsing HDFS... Spark Master at... DataFrame - Da... python 3.x - ca... mwillarrealty/do... ChatGPT

127.0.0.1:8080

OneDrive Outlook qldt Big Data Mang IP ShopeeStatus AirflowDocker IcebergDocker TrinoDocker SparkLab SparkExercises LearnSpark

Spark Master at spark://fed8b22359f9:7077

URL: spark://fed8b22359f9:7077

Alive Workers: 1

Cores in use: 1 Total: 0 Used

Memory in use: 1024.0 MB Total: 0.0 B Used

Resources in use:

Applications: 0 Running, 2 Completed

Drivers: 0 Running, 0 Completed

Status: ALIVE

Workers (1)

Worker Id	Address	State	Cores	Memory	Resources
worker-20240608072241-172.18.0.3:39139	172.18.0.3:39139	ALIVE	1 (0 Used)	1024.0 MB (0.0 B Used)	

Running Applications (0)

Application ID	Name	Cores	Memory per Executor	Resources Per Executor	Submitted Time	User	State	Duration
----------------	------	-------	---------------------	------------------------	----------------	------	-------	----------

Completed Applications (2)

Application ID	Name	Cores	Memory per Executor	Resources Per Executor	Submitted Time	User	State	Duration
app-20240608102151-0001	VDT2024	1	1024.0 MB		2024/06/08 10:21:51	spark	FINISHED	15 s
app-20240608101701-0000	VDT2024	1	1024.0 MB		2024/06/08 10:17:01	spark	FINISHED	10 s

Kết quả thu được trên HDFS:

