# Hướng dẫn cấu hình hệ thống

## Tổng quan: build 1 project có chức năng sau:

Có 1 api để insert nhiều dữ liệu vào trong kafka, từ kafka import vào trong logstash và cuối cùng là đẩy lên Elastic search.

Có các api để search trên elasticsearch: Thêm thư viên search để bọc ngoài thư viện:

<**dependency**>

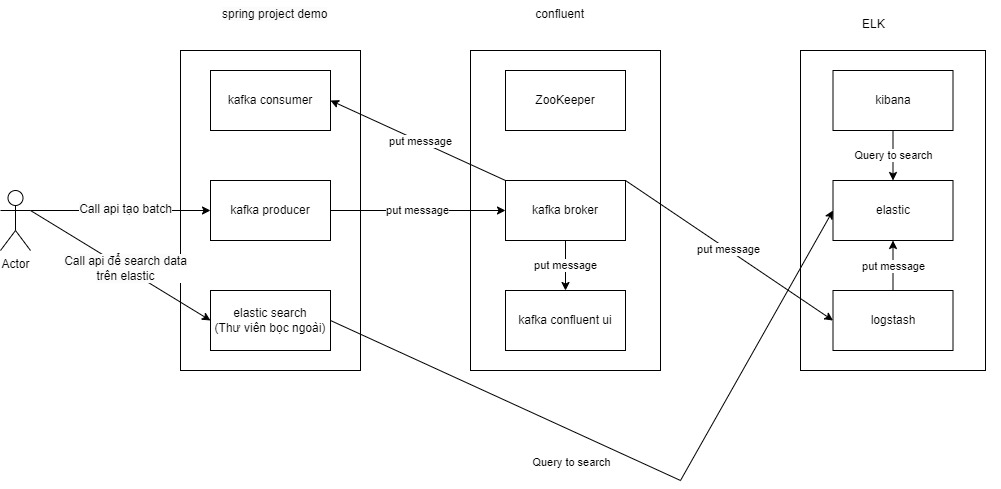
<**groupId**>org.springframework.data</**groupId**>

<**artifactId**>spring-data-elasticsearch</**artifactId**> <**version**>4.0.0.RELEASE</**version**>

</**dependency**>

(phần này optional, nếu có thời gian thì làm)

Sơ đồ hệ thống:



## Chi tiết

Cài đặt kafka:

* Có thể cài bình thường (download về) rồi chạy 🡪 cần cài thêm kafka Ui cho trực quan, ưu điểm là các file cấu hình ở sẵn đấy rồi, dễ config, nhược điểm, phải cài zuukeeper và phải dùng cli
* Cài bằng docker-compose: k dễ config, phải hiểu rõ các khái niệm trong docker: volumes…, viết file docker-compose cũng khó
* Cách dễ nhất là cài bằng confluent: https://www.confluent.io/blog/set-up-and-run-kafka-on-windows-and-wsl-2/

Config thì dùng Ui của confluent ngay trên browser là dc (đang chỉ cần config cho độ lớn bản tin trong kafka to lên là dc)

Cài như này là cài luôn tất cả những cái liên quan: zookeeper, ksql server…, có giao diện trực quan

Cài đặt ELK:

elastic search + kibana https://www.elastic.co/guide/en/elasticsearch/reference/current/docker.html

1. Cài elastic search:

* docker network create elastic
* docker pull docker.elastic.co/elasticsearch/elasticsearch:8.14.1
* docker run --name es01 --net elastic -p 9200:9200 -it -m 10GB docker.elastic.co/elasticsearch/elasticsearch:8.14.1

chú ý cái 10GB kia, mình cần bản tin kafka có dữ liệu lớn nên phải cho là 10GB

Copy the generated elastic password and enrollment token để sau đó còn login vào được kibana

Có thể gen lại:

* docker exec -it es01 /usr/share/elasticsearch/bin/elasticsearch-reset-password -u elastic
* docker exec -it es01 /usr/share/elasticsearch/bin/elasticsearch-create-enrollment-token -s kibana

1. Cài kibana:

* docker pull docker.elastic.co/kibana/kibana:8.14.1
* docker run --name kib01 --net elastic -p 5601:5601 docker.elastic.co/kibana/kibana:8.14.1

truy cập theo đường link ở console để có được link và mã code của kibana, sau đó login vào kibana và điền elastic password and enrollment token của elasticsearch đã gen trước đó

1. Cài Logstash:

Em cái bằng docker nó cứ lỗi nên e cài bình thường:

* download về bằng lệnh curl: curl -L https://artifacts.elastic.co/downloads/logstash/logstash-8.14.1-linux-x86\_64.tar.gz > logstash-8.14.1-linux-x86\_64.tar.gz
* Giải nén: tar -xvzf community\_images.tar.gz

Config và chạy thử

App Có 1 rest api:

curl --location 'http://localhost:9991/sendBathDataToKafka/ numberBatches' \

--header 'Authorization: Basic ZWxhc3RpYzpWVlBOSGFZOFpEbE1qbitXcXF6Tw=='

cho phép insert 1 request vào trong kaffka với topic là my-first-topic. Message gửi vào kafka có dạng sau:

public class BatchMessage {  
 String message;  
 private List<Object> value;  
}

Trong đó value là 1 list có numberBatches object.

Link project: https://github.com/hongdatchy/spring\_conflent\_ELK

Lưu ý thay link connect với kafka, elasticsearch, chạy project thì nó tự tạo topic: my-first-topic

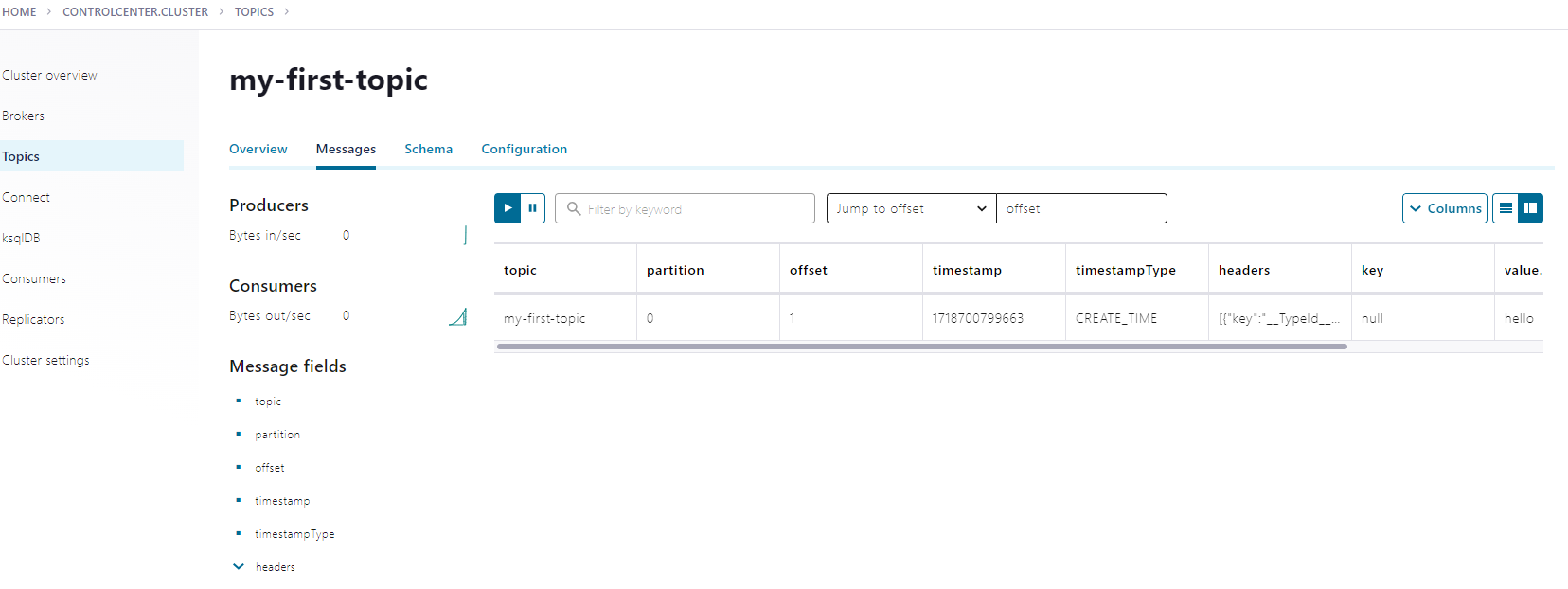
Truy cập vào kafka: <http://localhost:9021/>, kiểm tra topic: my-first-topic xem đã được tạo bởi code spring chưa.

Config logstash để hứng bản tin từ kafka: logstash-filter.conf

input {  
 kafka {  
 bootstrap\_servers => "localhost:9092"  
 topics => ["my-first-topic"]  
 codec => "json"  
 }  
}  
  
filter {  
 # Split each element in the array into a separate event  
 split {  
 field => "[value]"  
 }  
  
 # Rename fields  
 mutate {  
 rename => { "[value][name]" => "name" }  
 rename => { "[value][age]" => "age" }  
 rename => { "[value][major]" => "major" }  
 rename => { "[value][birthday]" => "birthday" }  
 }  
  
 # Convert birthday from Unix timestamp to date format, if it's present  
 date {  
 match => [ "birthday", "UNIX\_MS" ]  
 target => "birthday"  
 # Handle missing birthday field gracefully  
 tag\_on\_failure => []  
 }  
  
 # Remove unwanted fields  
 mutate {  
 remove\_field => ["@version", "host", "message", "[value]"]  
 }  
}  
  
output {  
 elasticsearch {  
 index => "my-index"  
 hosts => ["https://localhost:9200"]  
 user => "elastic"  
 password => "VVPNHaY8ZDlMjn+WqqzO"  
 ssl => true  
 ssl\_certificate\_verification => false  
 }  
 stdout {  
 codec => rubydebug  
 }  
}

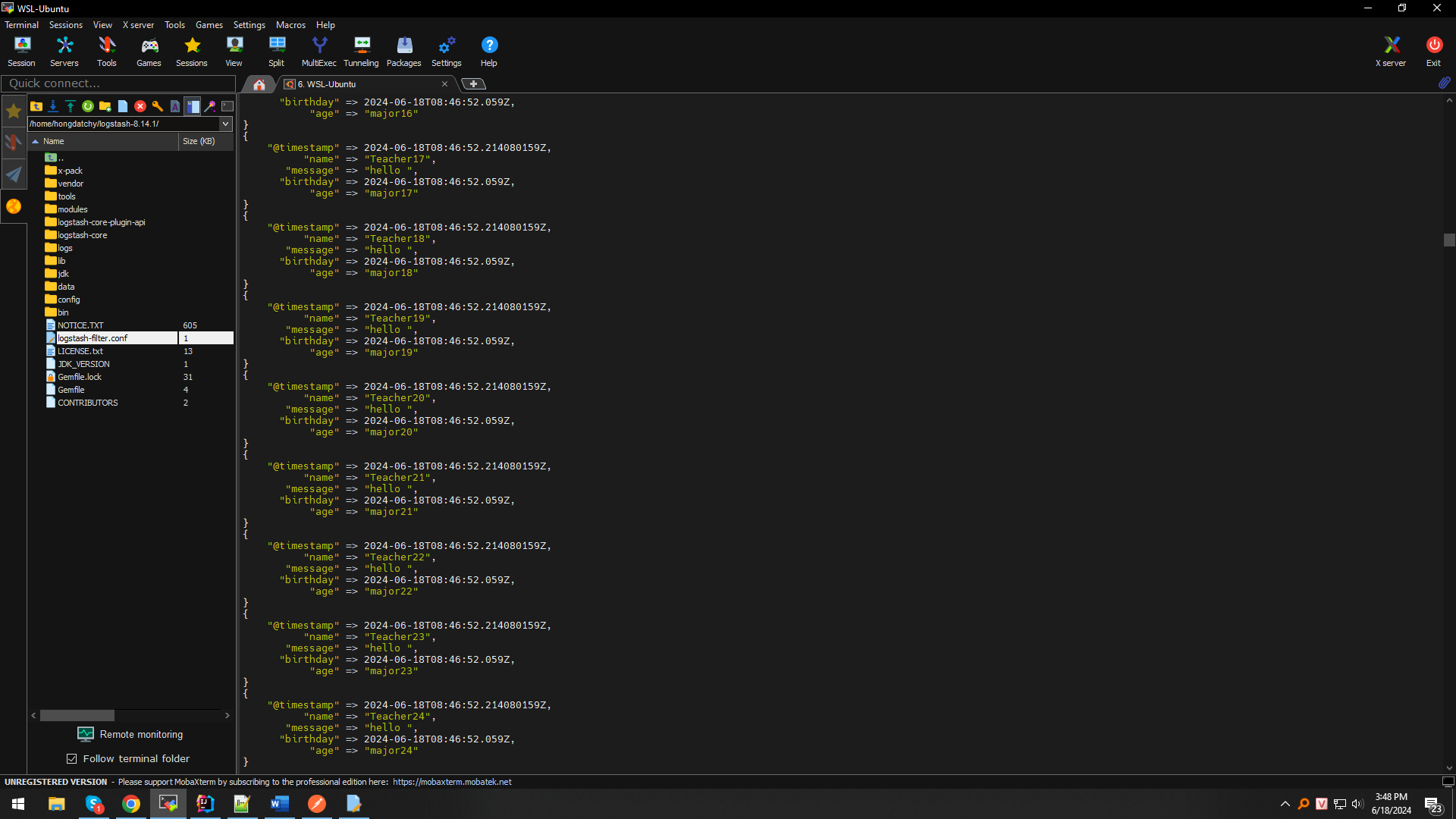
Message đến đầu vào trong logstash ở đầu input, tách dữ liệu từ 1 list (biến value) thành numberBatches object ở đầu filter và đưa vào trong elasticsearch ở output

call api <http://localhost:9991/sendBathDataToKafka/1000>

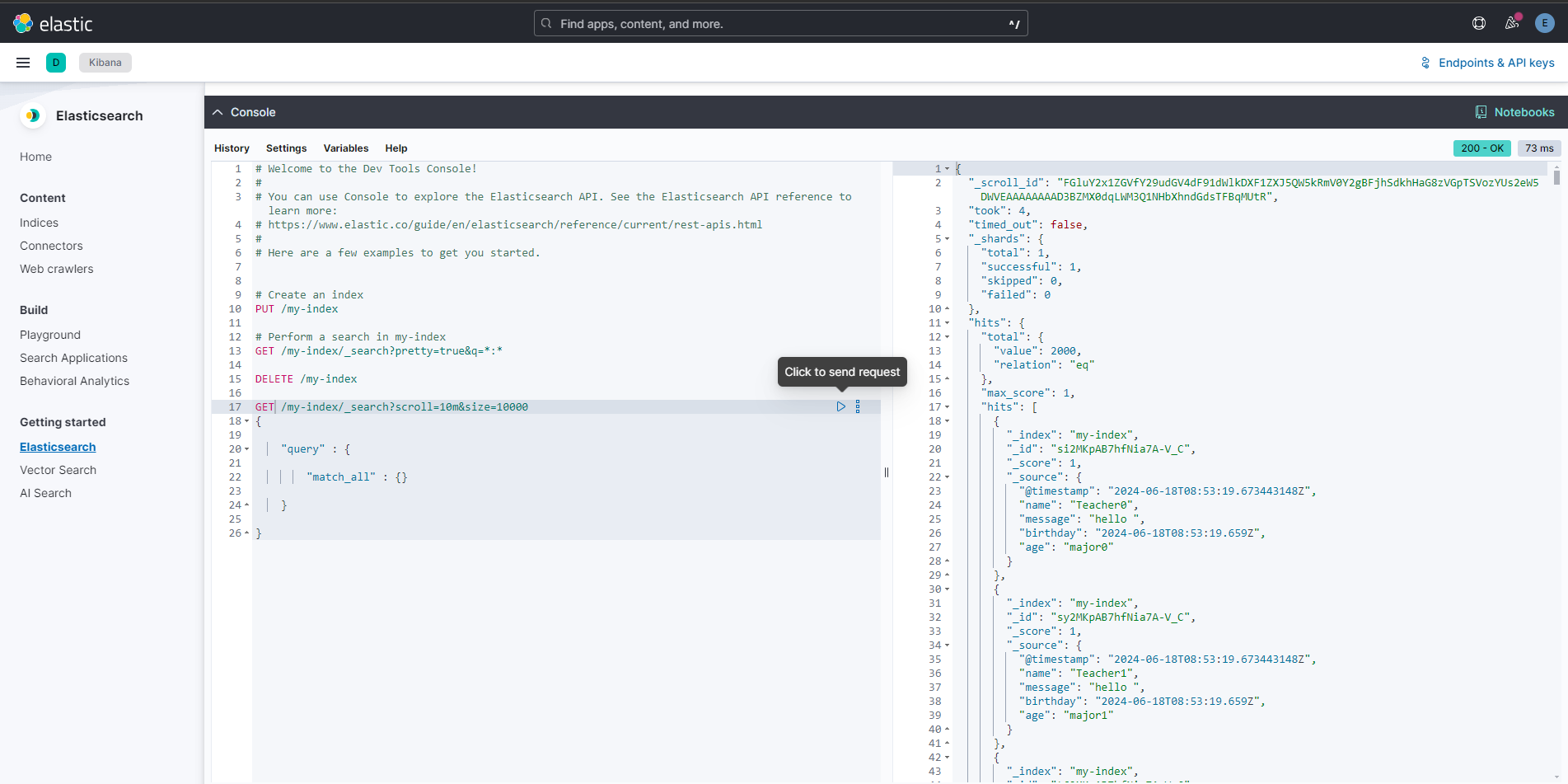
kiểm tra kafka: 

Hoặc nhìn trong code spring (cũng có consumer)

Còn kia là log của logstash



Kiểm tra trên elastic search:



Dữ liệu đã được insert vào và tách ra thành nhiều docment