**Restore mapping data**

curl -X GET "http://localhost:9200/kibana\_sample\_data\_flights/\_mapping" \

-H "Content-Type: application/json" \

-H "Authorization: Basic ZWxhc3RpYzoxMjM0NTY=" \

> /mnt/backups/kibana\_sample\_data\_flights\_mapping.json

**Run cái này để tạo file 1 => sau đó run mà không có cmd (từ khúc > ) để lấy scroll\_id pass vào câu lệnh tiếp theo**

curl -X POST "http://localhost:9200/kibana\_sample\_data\_flights/\_search?scroll=1m" \

-H "Content-Type: application/json" \

-H "Authorization: Basic ZWxhc3RpYzoxMjM0NTY=" \

-d '{

"size": 10000,

"query": {

"match\_all": {}

}

}' > /mnt/backups/part1.ndjson

**Pass scroll\_id vào và chạy en khi không còn**

curl -X POST "http://localhost:9200/\_search/scroll" \

-H "Content-Type: application/json" \

-H "Authorization: Basic ZWxhc3RpYzoxMjM0NTY=" \

-d '{

"scroll": "1m",

"scroll\_id": " FGluY2x1ZGVfY29udGV4dF91dWlkDXF1ZXJ5QW5kRmV0Y2gBFm53amNNcFFLUmtpWmR6QzItdTNrVVEAAAAAAAADahZuSHpQS1JOclJmMmxLaXJxYWFUNWZn"

}' > /mnt/backups/part2.ndjson

Kiểm tra nội dung file .ndjson:

Sau khi backup, bạn nên kiểm tra nhanh nội dung file để đảm bảo không bị thiếu hoặc lỗi:

* **head -n 5 /mnt/backups/kibana\_sample\_data\_flights\_data.ndjson**
* **wc -l /mnt/backups/clean\_part1.ndjson (Nếu trả về 0 hoặc 1 thì lỗi)**

Nếu bạn thấy file **có dữ liệu** nhưng lệnh:

wc -l /mnt/backups/part1.ndjson

trả về 0, thì có khả năng nội dung file nằm trên **một dòng duy nhất** thay vì mỗi document là **một dòng riêng biệt**! 🧠

Khi backup từ Elasticsearch, dữ liệu trả về là một **JSON object lớn**, không phải từng document theo chuẩn **NDJSON (newline-delimited JSON)**. Để kiểm tra, bạn có thể dùng lệnh:

head -n 10 /mnt/backups/part1.ndjson

Nếu nội dung trông giống thế này (tất cả trong **một dòng**):

{"took": 3, "timed\_out": false, "hits": {"total": 10000, "hits": [{"\_index": "kibana\_sample\_data\_flights", "\_id": "1", "\_source": {...}}, {"\_index": "kibana\_sample\_data\_flights", "\_id": "2", "\_source": {...}}]}}

👉 Đó là lý do **wc -l** trả về 0!

**🛠️ Cách giải quyết: Trích xuất từng document thành NDJSON**

Bạn cần **trích xuất từng document** từ phần "hits.hits" để đúng chuẩn **.ndjson**.

Dùng **jq** để xử lý file nhanh chóng:

cat /mnt/backups/part1.ndjson | jq -c '.hits.hits[].\_source' > /mnt/backups/clean\_part1.ndjson

🔸 **Giải thích:**

* jq: Bộ lọc JSON mạnh mẽ
* -c: In ra mỗi object trên **một dòng**
* .hits.hits[].\_source: Lấy **dữ liệu chính** của mỗi document

Sau đó, kiểm tra lại:

wc -l /mnt/backups/clean\_part1.ndjson

Nếu thấy **số dòng > 0** → 🎉 **Thành công!** Bạn đã chuẩn hóa thành **NDJSON** rồi!

**✅ Nếu chưa có jq, bạn có thể cài (trong Docker):**

apt-get update

apt-get install jq -y

**🚀 Giờ bạn đã có file sạch để restore lại Elasticsearch:**

curl -X POST "http://localhost:9200/kibana\_sample\_data\_flights/\_bulk" \

-H "Content-Type: application/json" \

-H "Authorization: Basic ZWxhc3RpYzoxMjM0NTY=" \

--data-binary "@/mnt/backups/clean\_part1.ndjson"

Mọi thứ sẽ chạy mượt mà thôi! ✌️ Nếu còn vấn đề gì, cứ thoải mái nhắn nhé — mình sẽ giúp bạn tới cùng! 🔥

À, mình hiểu rồi! 🚀  
Bạn đã **backup mapping** bằng lệnh này:

curl -X GET "http://localhost:9200/kibana\_sample\_data\_flights/\_mapping" \

-H "Content-Type: application/json" \

-H "Authorization: Basic ZWxhc3RpYzoxMjM0NTY=" \

> /mnt/backups/kibana\_sample\_data\_flights\_mapping.json

Lệnh này chỉ tải về **mapping** (cấu trúc của index) chứ không phải toàn bộ dữ liệu.  
Bây giờ nếu bạn muốn khôi phục lại index từ file này, mình sẽ chỉ cách làm! 👇

## 🛠 **1. Tạo lại index với mapping đã backup:**

Nếu index đã bị xóa, bạn có thể tạo lại bằng cách:

curl -X PUT "http://localhost:9200/kibana\_sample\_data\_flights" \

-H "Content-Type: application/json" \

-H "Authorization: Basic ZWxhc3RpYzoxMjM0NTY=" \

--data-binary "@/mnt/backups/kibana\_sample\_data\_flights\_mapping.json"

🔧 **Lưu ý:** File backup của bạn có thể chứa thêm **tên index** ở lớp ngoài, nên bạn cần kiểm tra hoặc chỉnh sửa file lại một chút.

Ví dụ, nếu file của bạn thế này:

{

"kibana\_sample\_data\_flights": {

"mappings": {

"properties": {

"FlightNum": { "type": "keyword" },

"Origin": { "type": "keyword" },

"flightTimeHour": { "type": "float" }

}

}

}

}

👉 Bạn cần **bỏ lớp tên index** và giữ nguyên phần **mappings** thôi:

{

"mappings": {

"properties": {

"FlightNum": { "type": "keyword" },

"Origin": { "type": "keyword" },

"flightTimeHour": { "type": "float" }

}

}

}

## ✅ **2. Kiểm tra lại index:**

Sau khi tạo index thành công, kiểm tra lại mapping đã được áp dụng chưa:

curl -X GET "http://localhost:9200/kibana\_sample\_data\_flights/\_mapping?pretty"

Nếu mapping hiện ra đúng như bạn mong muốn là thành công rồi! 🎯

## 🛢️ **3. Khôi phục dữ liệu (nếu cần):**

Mapping chỉ là cấu trúc dữ liệu. Nếu bạn muốn backup và restore luôn cả **dữ liệu**, lần sau nên dùng **Bulk API** hoặc **Snapshot API**. Ví dụ để backup dữ liệu:

curl -X GET "http://localhost:9200/kibana\_sample\_data\_flights/\_search?scroll=1m&size=1000" \

-H "Content-Type: application/json" \

-H "Authorization: Basic ZWxhc3RpYzoxMjM0NTY=" \

> /mnt/backups/kibana\_sample\_data\_flights\_data.json

Rồi khôi phục lại bằng **Bulk API**:

jq -c ". | { index: { \_index: \"kibana\_sample\_data\_flights\" } }, ." part1.ndjson > clean\_part1.ndjson

curl -X POST "http://localhost:9200/kibana\_sample\_data\_flights/\_bulk" \

-H "Content-Type: application/json" \

-H "Authorization: Basic ZWxhc3RpYzoxMjM0NTY=" \

--data-binary "@/mnt/backups/kibana\_sample\_data\_flights\_data.json"

## 🚀 **Tóm lại:**

* **File bạn backup là mapping** → Chỉnh sửa và tạo lại index bằng mapping đó.
* **Nếu muốn khôi phục cả dữ liệu**, bạn nên backup bằng **Bulk API** hoặc **Snapshot API**.

Bạn thử làm theo cách mình hướng dẫn xem chạy ổn chưa nhé? Nếu còn vướng chỗ nào, mình sẽ giúp bạn ngay! ✌️✨

## 🏗️ \*\*2. Tạo lại index với mapping\*\*

Để \*\*restore đúng cấu trúc\*\*, bạn cần tạo lại index bằng file mapping đã backup:

```bash

curl -X PUT "http://localhost:9200/kibana\_sample\_data\_flights" \

-H "Content-Type: application/json" \

-H "Authorization: Basic ZWxhc3RpYzoxMjM0NTY=" \

--data-binary "@/mnt/backups/my\_backup\_repo/kibana\_sample\_data\_flights\_mapping.json"

```

Nếu thành công, bạn sẽ thấy:

```json

{

"acknowledged": true,

"shards\_acknowledged": true,

"index": "kibana\_sample\_data\_flights"

}

```

---

## 📥 \*\*3. Import lại dữ liệu\*\*

Sau khi tạo index xong, bạn \*\*import lại dữ liệu\*\* bằng API `\_bulk`:

```bash

curl -X POST "http://localhost:9200/kibana\_sample\_data\_flights/\_bulk" \

-H "Content-Type: application/json" \

-H "Authorization: Basic ZWxhc3RpYzoxMjM0NTY=" \

--data-binary "@/mnt/backups/kibana\_sample\_data\_flights\_data.ndjson"

```

---

## 🔍 \*\*4. Kiểm tra kết quả\*\*

Sau khi import xong, bạn có thể kiểm tra bằng cách query thử:

```bash

curl -X GET "http://localhost:9200/kibana\_sample\_data\_flights/\_search?pretty=true"

```

Nếu dữ liệu trả về đúng, là bạn đã \*\*restore thành công\*\* rồi! 🎉

---

## 🚩 \*\*5. Một số lưu ý quan trọng\*\*

🔸 \*\*Nếu index đã tồn tại:\*\* Bạn cần xóa index cũ trước khi tạo lại:

```bash

curl -X DELETE "http://localhost:9200/kibana\_sample\_data\_flights"

```

🔸 \*\*Kiểm tra lỗi khi import:\*\*

Nếu gặp lỗi trong `\_bulk`, bạn có thể thêm tham số `errors`:

```bash

curl -X POST "http://localhost:9200/kibana\_sample\_data\_flights/\_bulk" \

-H "Content-Type: application/json" \

-H "Authorization: Basic ZWxhc3RpYzoxMjM0NTY=" \

--data-binary "@/mnt/backups/kibana\_sample\_data\_flights\_data.ndjson" \

| jq '.errors'

```

Nếu `false`, thì không có lỗi!

🔸 \*\*Đảm bảo dữ liệu đúng format:\*\*

- Mỗi dòng trong file `.ndjson` là một JSON object hợp lệ.

- Dữ liệu phải đúng \*\*mapping\*\* đã tạo, đặc biệt với \*\*keyword\*\* và \*\*geo\_point\*\*.

---

## 🏁 \*\*6. Kết luận\*\*

Với quy trình này, bạn có thể \*\*backup & restore toàn bộ dữ liệu và cấu trúc\*\* trong Elasticsearch cực kỳ dễ dàng! 🚀

Mình đã giúp bạn đi qua từng bước — từ \*\*mount thư mục\*\*, \*\*tạo lại index\*\*, đến \*\*import dữ liệu\*\* bằng API. Nếu còn bất kỳ vấn đề nào, cứ thoải mái nói mình biết nhé! ✌️