

133.71.201.197 の ElasticSearch サーバーのデータの移行 について

祖父江匠真

1 概要

今回は, 133.71.201.197 の ElasticSearch サーバーにある pcs_recyclekan という名前のインデックス以外のインデックスについて調査を行い, 133.71.106.141 の ElasticSearch サーバーにデータ移行を行ったことについて報告する.

2 133.71.201.197 の ElasticSearch サーバーにあるインデックスについて

図 1 に 133.71.201.197 の ElasticSearch サーバーにあるインデックスの一覧を示す.

インデックス管理

[インデックス](#)[データストリーム](#)[インデックステンプレート](#)[コンポーネントテン](#)

Elasticsearch インデックスを個々に、または一斉に更新します。[詳細情報](#)

<input type="checkbox"/> 名前	ヘルス	ステータス
<input type="checkbox"/> pcs_test	● yellow	open
<input type="checkbox"/> pcs_recyclekan	● yellow	open
<input type="checkbox"/> leaf_load	● yellow	open
<input type="checkbox"/> pcs_log	● yellow	open
<input type="checkbox"/> pcs_test4	● yellow	open
<input type="checkbox"/> leaf	● yellow	open
<input type="checkbox"/> leaf_grid	● yellow	open

ページごとの行数: 10

図 1: 133.71.201.197 の Elasticsearch サーバーにあるインデックスの一覧

これらのインデックスが保存しているデータについて説明する。

- pcs_test
 - 恵村君がプログラムの検証目的で使用しているインデックス
- pcs_recyclekan
 - リサイクル館の太陽光発電に関するデータを保存しているインデックス

- leaf_load
 - leaf のデータが保存されているインデックス
- pcs_log
 - リサイクル館の太陽光発電に関するデータを Elasticsearch にインサートする Python プログラムのログ情報を保存しているインデックス
- pcs_test4
 - 恵村君がプログラムの検証目的で使用しているインデックス
- leaf
 - leaf のデータが保存されているインデックス
- leaf_grid
 - leaf のデータが保存されているインデックス

なお, leaf, leaf_grid, leaf_load インデックスについて, Kibana で各インデックスのマッピング情報を確認したところ, 3 つともすべて同じマッピング情報を保持しており, 同じフィールドを持つドキュメントをそれぞれのインデックスで保存していることが分かった.

leaf, leaf_grid, leaf_load インデックスのマッピング情報をリスト 1 に示す.

Listing 1: leaf と名の付くインデックスが持つマッピング情報

```

1  {
2    "mappings": {
3      "_doc": {
4        "properties": {
5          "@timestamp": {
6            "type": "date"
7          },
8          "A1": {
9            "type": "float"
10         },
11         "A2": {
12           "type": "float"
13         },
14         "F": {
15           "type": "float"
16         },
17         "JPtime": {
18           "type": "date"

```

```

19     },
20     "P": {
21         "type": "float"
22     },
23     "P1": {
24         "type": "float"
25     },
26     "P2": {
27         "type": "float"
28     },
29     "PF": {
30         "type": "float"
31     },
32     "PF1": {
33         "type": "float"
34     },
35     "PF2": {
36         "type": "float"
37     },
38     "Q": {
39         "type": "float"
40     },
41     "Q1": {
42         "type": "float"
43     },
44     "Q2": {
45         "type": "float"
46     },
47     "S": {
48         "type": "float"
49     },
50     "S1": {
51         "type": "float"
52     },
53     "S2": {
54         "type": "float"
55     },
56     "V1": {
57         "type": "float"
58     },
59     "V2": {
60         "type": "float"
61     },
62     "data_time": {
63         "type": "date"
64     },
65     "type": {

```

```

66         "type": "text",
67         "fields": {
68             "keyword": {
69                 "type": "keyword",
70                 "ignore_above": 256
71             }
72         }
73     }
74 }
75 }
76 }
77 }

```

pcs_test インデックスと pcs_test4 インデックスに関しては、恵村君が検証用途で
 使用しているものであるため、今回の移行対象からは除外し、pcs_log, leaf, leaf_load,
 leaf_grid インデックスのみを移行対象とした。

3 データ移行手順について

データ移行手順について、まず移行元の Elasticsearch サーバーのデータをローカ
 ルマシンに JSON 形式でエクスポートして、作成した Python プログラムを実行し
 て移行先の Elasticsearch サーバーにデータをインサートした。

3.1 データのエクスポート

移行元の Elasticsearch サーバーのデータのローカルマシンへのエクスポートに
 は、elasticsearch-dump [1] ライブラリを使用して JSON 形式でエクスポートした。その際、
 pcs_log, leaf, leaf_load, leaf_grid という名前のインデックスのデータをエクスポート
 した。

3.2 データのインポート

エクスポートした JSON ファイルを、作成した Python プログラムから読み込ん
 で、Python の elasticsearch ライブラリを用いて移行先の Elasticsearch サーバーにイ
 ンサートした。

移行先の Elasticsearch サーバーにおけるインデックス名については、133.71.201.197
 の Elasticsearch サーバーと同名のインデックスに保存した。

4 データ移行が正常に行えたか確認

図 2に移行元の Elasticsearch サーバーの leaf という文字列を含むインデックスのドキュメント数をカウントしたものを, 図 3に移行先の Elasticsearch サーバーの leaf という文字列を含むインデックスのドキュメント数をカウントしたものを示す.

図 2と図 3より, ドキュメント数が一致していることからデータ移行が正常に行えたと判断できる.

<input type="checkbox"/> 名前	ヘルス	ステータス	プライマリ	レプリカ	ドキュメント数
<input type="checkbox"/> leaf_load	● yellow	open	1	1	3123
<input type="checkbox"/> leaf	● yellow	open	1	1	5070
<input type="checkbox"/> leaf_grid	● yellow	open	1	1	1948

図 2: 133.71.201.197 の Elasticsearch サーバーの leaf という文字列を含むインデックスのドキュメントのカウント結果

<input type="checkbox"/> 名前	ヘルス	ステータス	プライマリ	レプリカ	ドキュメント数
<input type="checkbox"/> leaf_load	● green	open	1	1	3123
<input type="checkbox"/> leaf	● green	open	1	1	5070
<input type="checkbox"/> leaf_grid	● green	open	1	1	1948

図 3: 133.71.106.141 の Elasticsearch サーバーの leaf という文字列を含むインデックスのドキュメントのカウント結果

次に, 図 4に移行元の Elasticsearch サーバーの pcs_log インデックスのドキュメント数をカウントしたものを, 図 5に移行先の Elasticsearch サーバーの pcs_log インデックスのドキュメント数をカウントしたものを示す.

図 4と図 5より, ドキュメント数が一致していることからデータ移行が正常に行えたと判断できる.

<input type="checkbox"/> 名前	ヘルス	ステータス	プライマリ	レプリカ	ドキュメント数
<input type="checkbox"/> pcs_log	● yellow	open	1	1	1535

図 4: 133.71.201.197 の Elasticsearch サーバーの pcs_log インデックスのドキュメントのカウント結果

<input type="checkbox"/> 名前	ヘルス	ステータス	プライマリ	レプリカ	ドキュメント数
<input type="checkbox"/> pcs_log	● green	open	1	1	1535

図 5: 133.71.106.141 の Elasticsearch サーバーの pcs_log インデックスのドキュメントのカウント結果

5 まとめ

今回は, 133.71.201.197 の Elasticsearch サーバーにある pcs_log, leaf, leaf_load, leaf_grid インデックスを 133.71.106.141 の Elasticsearch サーバーにデータ移行して, 各インデックスのドキュメント数を移行元サーバーと移行先サーバーとの間で比較することで, データ移行が正常に行えたことを報告した.

次回は, 以前 CO₂ データを移行した際に Elasticsearch の Bulk API を使用してデータ移行を行ったことが原因で, ラズベリーパイ上で実行している CO₂ データのインサート用のプログラムから移行先の Elasticsearch サーバーのインデックスにインサートが出来なくなった問題について, 原因の特定と解決方法の調査を行い報告する.

参考文献

- [1] Ferron H, "ElasticDump ", <https://github.com/elasticsearch-dump/elasticsearch-dump>, 参照 June 19,2023.