

# 異なる Elasticsearch クラスタへのノード参加検証

祖父江匠真

## 1 概要

今回は, CO<sub>2</sub> データなどが保存されている 3 ノードで構築されたクラスタに対して, リサイクル館の太陽光パネルの計測データを保存している Elasticsearch ノードが新たなノードとしてクラスタに参加できるか, Docker を用いて検証した.

## 2 手順

### 2.1 単一ノードで稼働するクラスタ A の構築とシャットダウン

まず, docker-compose を用いて単一ノード (コンテナ名は es04) でクラスタ (以後このクラスタをクラスタ A と呼ぶ) を構築する.

Listing 1 にクラスタ A の構築の際に使用した docker-compose.yml を示す.

Listing 1: クラスタ A の構築の際に使用した docker-compose.yml

```
1 version: '2.2'
2 services:
3   es04:
4     image: docker.elastic.co/elasticsearch/elasticsearch:7.17.9
5     container_name: es04
6     environment:
7       - node.name=es04
8       - bootstrap.memory_lock=true
9       - "ES_JAVA_OPTS=-Xms2048m-Xmx2048m"
10      - discovery.type=single-node
11     volumes:
12       - data04:/usr/share/elasticsearch/data
13     ports:
14       - 9200:9200
15     networks:
16       - esnet
17
18 volumes:
19   data04:
20     driver: local
21
22 networks:
23   esnet:
```

docker-compose を用いてノードを起動した後、クラスタの情報について問い合わせた結果を図 1 に示す。



```
sofue@sofue-DAIV-DGX750:~/apps/clustering-different-es-ver$ curl -XGET http://localhost:9200/
{
  "name" : "es04",
  "cluster_name" : "docker-cluster",
  "cluster_uuid" : "VbMebhfLQ0yQSlEx2nPFhg",
  "version" : {
    "number" : "7.17.9",
    "build_flavor" : "default",
    "build_type" : "docker",
    "build_hash" : "ef4822227ee6b9e70e502f0f0daa52435ee634d",
    "build_date" : "2023-01-31T05:34:43.305517834Z",
    "build_snapshot" : false,
    "lucene_version" : "8.11.1",
    "minimum_wire_compatibility_version" : "6.8.0",
    "minimum_index_compatibility_version" : "6.0.0-beta1"
  },
  "tagline" : "You Know, for Search"
}
```

図 1: クラスタの情報について問い合わせた結果

最後にノードをシャットダウンした。

## 2.2 クラスタ B の構築とシャットダウン

次に、クラスタ A に使用したノードとは別の 3 ノード (コンテナ名はそれぞれ es01, es02, es03) でクラスタ (以後このクラスタをクラスタ B と呼ぶ) を構築する。

Listing 2 にクラスタ B の構築の際に使用した docker-compose.yml を示す。

Listing 2: クラスタ B の構築の際に使用した docker-compose.yml

```
1 version: '2.2'
2 services:
3   es01:
4     image: docker.elastic.co/elasticsearch/elasticsearch:7.17.9
5     container_name: es01
6     environment:
7       - node.name=es01
8       - discovery.seed_hosts=es01,es02,es03
9       - cluster.initial_master_nodes=es01,es02,es03
10      - bootstrap.memory_lock=true
```

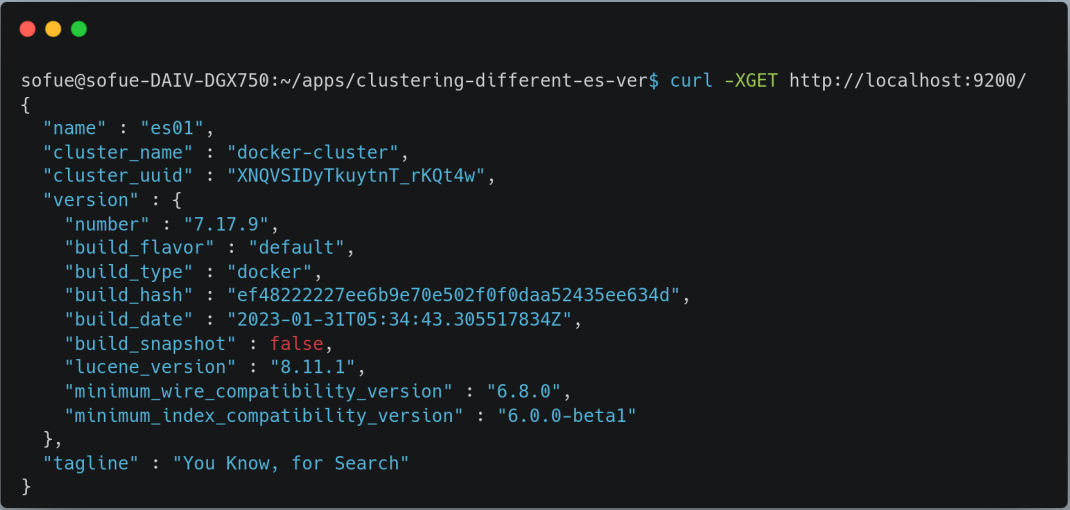
```

11     - "ES_JAVA_OPTS=-Xms2048m_-Xmx2048m"
12 volumes:
13     - data01:/usr/share/elasticsearch/data
14 ports:
15     - 9200:9200
16 networks:
17     - elastic
18 es02:
19     image: docker.elastic.co/elasticsearch/elasticsearch:7.17.9
20     container_name: es02
21     environment:
22         - node.name=es02
23         - discovery.seed_hosts=es01,es02,es03
24         - cluster.initial_master_nodes=es01,es02,es03
25         - bootstrap.memory_lock=true
26         - "ES_JAVA_OPTS=-Xms2048m_-Xmx2048m"
27     volumes:
28         - data02:/usr/share/elasticsearch/data
29     networks:
30         - elastic
31 es03:
32     image: docker.elastic.co/elasticsearch/elasticsearch:7.17.9
33     container_name: es03
34     environment:
35         - node.name=es03
36         - discovery.seed_hosts=es01,es02,es03
37         - cluster.initial_master_nodes=es01,es02,es03
38         - bootstrap.memory_lock=true
39         - "ES_JAVA_OPTS=-Xms2048m_-Xmx2048m"
40     volumes:
41         - data03:/usr/share/elasticsearch/data
42     networks:
43         - elastic
44
45 volumes:
46     data01:
47         driver: local
48     data02:
49         driver: local
50     data03:
51         driver: local
52
53 networks:
54     elastic:
55         driver: bridge

```

docker-compose を用いて 3 つのノードを起動した後、クラスタの情報について問

い合わせた結果を図 2に示す.



```
sofue@sofue-DAIV-DGX750:~/apps/clustering-different-es-ver$ curl -XGET http://localhost:9200/
{
  "name" : "es01",
  "cluster_name" : "docker-cluster",
  "cluster_uuid" : "XNQVSiDyTkuytnT_rKQt4w",
  "version" : {
    "number" : "7.17.9",
    "build_flavor" : "default",
    "build_type" : "docker",
    "build_hash" : "ef4822227ee6b9e70e502f0f0daa52435ee634d",
    "build_date" : "2023-01-31T05:34:43.305517834Z",
    "build_snapshot" : false,
    "lucene_version" : "8.11.1",
    "minimum_wire_compatibility_version" : "6.8.0",
    "minimum_index_compatibility_version" : "6.0.0-beta1"
  },
  "tagline" : "You Know, for Search"
}
```

図 2: クラスタの情報について問い合わせた結果

最後にノードを全てシャットダウンした.

## 2.3 クラスタ B への参加試行

次に, Listing 2の docker-compose.yml に, クラスタ A のノード (es04) を追加し, 合計 4 ノードでクラスタ B を起動する.

Listing 3に, 合計 4 ノードでクラスタ B を起動する際に使用した docker-compose.yml を示す.

```
version: '2.2'
services:
  es01:
    image: docker.elastic.co/elasticsearch/elasticsearch:7.17.9
    container_name: es01
    environment:
      - node.name=es01
      - discovery.seed_hosts=es01,es02,es03,es04
      - cluster.initial_master_nodes=es01,es02,es03,es04
      - bootstrap.memory_lock=true
      - "ES_JAVA_OPTS=-Xms2048m -Xmx2048m"
    ulimits:
      memlock:
        soft: -1
        hard: -1
    volumes:
      - data01:/usr/share/elasticsearch/data
    ports:
      - 9200:9200
    networks:
      - elastic
  es02:
    image: docker.elastic.co/elasticsearch/elasticsearch:7.17.9
    container_name: es02
    environment:
      - node.name=es02
      - discovery.seed_hosts=es01,es02,es03,es04
      - cluster.initial_master_nodes=es01,es02,es03,es04
      - bootstrap.memory_lock=true
      - "ES_JAVA_OPTS=-Xms2048m -Xmx2048m"
    ulimits:
      memlock:
        soft: -1
        hard: -1
    volumes:
      - data02:/usr/share/elasticsearch/data
    networks:
      - elastic
  es03:
    image: docker.elastic.co/elasticsearch/elasticsearch:7.17.9
    container_name: es03
    environment:
      - node.name=es03
      - discovery.seed_hosts=es01,es02,es03,es04
      - cluster.initial_master_nodes=es01,es02,es03,es04
      - bootstrap.memory_lock=true
      - "ES_JAVA_OPTS=-Xms2048m -Xmx2048m"
    ulimits:
      memlock:
        soft: -1
        hard: -1
    volumes:
      - data03:/usr/share/elasticsearch/data
    networks:
      - elastic
  es04:
    image: docker.elastic.co/elasticsearch/elasticsearch:7.17.9
    container_name: es04
    environment:
      - node.name=es04
      - discovery.seed_hosts=es01,es02,es03,es04
```

Listing 3: 合計4ノードでクラスターBを起動する際に使用した docker-compose.yml

```
1 version: '2.2'
2 services:
3   es01:
4     image: docker.elastic.co/elasticsearch/elasticsearch:7.17.9
5     container_name: es01
6     environment:
7       - node.name=es01
8       - discovery.seed_hosts=es01,es02,es03,es04
9       - cluster.initial_master_nodes=es01,es02,es03,es04
10      - bootstrap.memory_lock=true
11      - "ES_JAVA_OPTS=-Xms2048m_-Xmx2048m"
12     volumes:
13       - data01:/usr/share/elasticsearch/data
14     ports:
15       - 9200:9200
16     networks:
17       - elastic
18   es02:
19     image: docker.elastic.co/elasticsearch/elasticsearch:7.17.9
20     container_name: es02
21     environment:
22       - node.name=es02
23       - discovery.seed_hosts=es01,es02,es03,es04
24       - cluster.initial_master_nodes=es01,es02,es03,es04
25       - bootstrap.memory_lock=true
26       - "ES_JAVA_OPTS=-Xms2048m_-Xmx2048m"
27     volumes:
28       - data02:/usr/share/elasticsearch/data
29     networks:
30       - elastic
31   es03:
32     image: docker.elastic.co/elasticsearch/elasticsearch:7.17.9
33     container_name: es03
34     environment:
35       - node.name=es03
36       - discovery.seed_hosts=es01,es02,es03,es04
37       - cluster.initial_master_nodes=es01,es02,es03,es04
38       - bootstrap.memory_lock=true
39       - "ES_JAVA_OPTS=-Xms2048m_-Xmx2048m"
40     volumes:
41       - data03:/usr/share/elasticsearch/data
42     networks:
43       - elastic
44   es04:
45     image: docker.elastic.co/elasticsearch/elasticsearch:7.17.9
```

```

46     container_name: es04
47     environment:
48         - node.name=es04
49         - discovery.seed_hosts=es01,es02,es03,es04
50         - cluster.initial_master_nodes=es01,es02,es03,es04
51         - bootstrap.memory_lock=true
52         - "ES_JAVA_OPTS=-Xms2048m_-Xmx2048m"
53     volumes:
54         - data04:/usr/share/elasticsearch/data
55     networks:
56         - elastic
57
58 volumes:
59     data01:
60         driver: local
61     data02:
62         driver: local
63     data03:
64         driver: local
65     data04:
66         driver: local
67
68 networks:
69     elastic:
70         driver: bridge

```

クラスタの起動後、クラスタに参加しているノードの一覧を取得した結果を図4に示す。

```

sofue@sofue-DAIV-DGX750:~/apps/clustering-different-es-ver$ curl -X GET "localhost:9200/_cat/nodes?v&pretty"
ip          heap.percent ram.percent cpu load_1m load_5m load_15m node.role master name
172.22.0.4   18           21      5    0.75   1.24   1.02 cdfhilmrstw -   es02
172.22.0.2   16           21      5    0.75   1.24   1.02 cdfhilmrstw -   es01
172.22.0.5   33           21      5    0.75   1.24   1.02 cdfhilmrstw *   es03

```

図 4: 合計 4 ノードでクラスタを起動した後、クラスタに参加しているノードの一覧を取得した結果

図 4より、クラスタ A のノードがクラスタ B に参加できていないことが分かる。es04 コンテナ（クラスタ A のノード）で出力されたログの一部を図 5に示す。

```

es04 | "Caused by: org.elasticsearch.cluster.coordination.CoordinationStateRejectedException: This node previously joined a cluster
with UUID [VbMebhFLQyQSlEx2nPFhg] and is now trying to join a different cluster with UUID [XNQVSIIDyTkuytnT_rK0t4w]. This is forbidden
and usually indicates an incorrect discovery or cluster bootstrapping configuration. Note that the cluster UUID persists across
restarts and can only be changed by deleting the contents of the node's data paths [] which will also remove any data held by this
node.",
es04 | "at org.elasticsearch.cluster.coordination.JoinHelper.lambda$new$8(JoinHelper.java:213) ~[elasticsearch-7.17.9.jar:7.17.9]",
es04 | "at
org.elasticsearch.xpack.security.transport.SecurityServerTransportInterceptor$ProfileSecuredRequestHandler$1.doRun(SecurityServerTransp
ortInterceptor.java:341) ~[?:?]",
es04 | "at org.elasticsearch.common.util.concurrent.AbstractRunnable.run(AbstractRunnable.java:26) ~[elasticsearch-
7.17.9.jar:7.17.9]",
es04 | "at
org.elasticsearch.xpack.security.transport.SecurityServerTransportInterceptor$ProfileSecuredRequestHandler.messageReceived(SecurityServ
erTransportInterceptor.java:417) ~[?:?]",
es04 | "at org.elasticsearch.transport.RequestHandlerRegistry.processMessageReceived(RequestHandlerRegistry.java:67) ~[elasticsearch-
7.17.9.jar:7.17.9]",
es04 | "at org.elasticsearch.transport.InboundHandler$1.doRun(InboundHandler.java:272) ~[elasticsearch-7.17.9.jar:7.17.9]",
es04 | "at org.elasticsearch.common.util.concurrent.ThreadContext$ContextPreservingAbstractRunnable.doRun(ThreadContext.java:777) ~
[elasticsearch-7.17.9.jar:7.17.9]",
es04 | "at org.elasticsearch.common.util.concurrent.AbstractRunnable.run(AbstractRunnable.java:26) ~[elasticsearch-
7.17.9.jar:7.17.9]",
es04 | "at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1144) ~[?:?]",
es04 | "at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:642) ~[?:?]",
es04 | "at java.lang.Thread.run(Thread.java:1589) ~[?:?]" }

```

図 5: es04 コンテナのログ

図 5には、異なるクラスタ ID を持つクラスタにノードが参加することは禁止されており、これを行うためにはインデックスやドキュメント情報などが格納されているデータパス配下のフォルダ、ファイルを削除する必要があると書かれている。

以上の検証結果から、既に稼働しているノードを別のクラスタに新しいノードとして参加させることは出来ないことが分かった。

### 3 まとめ

今回は、CO<sub>2</sub> データなどが保存されている 3 ノードで構築されたクラスタに対し、リサイクル館の太陽光パネルの計測データを保存している Elasticsearch ノードが新たなノードとしてクラスタに参加できるか、Docker を用いて検証した。

検証の結果、Elasticsearch のノードは異なるクラスタ ID を持つクラスタに参加することはできないことが分かった。ノードが別のクラスタに参加するためには、インデックスやドキュメント情報などを格納しているデータパスのフォルダやファイルを削除する必要があるが、これはそのノードのデータを失うことを意味する。

したがって、リサイクル館の太陽光パネルの計測データが保存された Elasticsearch ノードをクラスタに参加させるには以下の 2 通りの方法が考えられる。

1. リサイクル館の太陽光パネルの計測データが保存された Elasticsearch ノードのバックアップを取り、ノードに保存されたインデックスやドキュメントのデータを削除した上で、CO<sub>2</sub> データなどが保存されたクラスタに新しいノードとして参加させる
2. CO<sub>2</sub> データなどが保存されたクラスタとは別で、サーバーゾーンに新たにクラスタを構築する。



## 参考文献

- [1] Elasticsearch B.V., "Data node's cluster uuid different from master node's cluster uuid ", <https://discuss.elastic.co/t/data-node-s-cluster-uuid-different-from-master-nodes-cluster-uuid/196737>, 参照 Dec 18,2023.