

# 異なるバージョンの Elasticsearch ノードのクラスタリング

祖父江匠真

## 1 概要

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 システム (133.71.201.197) はバージョン 7.17.6 であり, CO<sub>2</sub> データを保存している Elasticsearch システム (133.71.106.141, 133.71.106.170, 133.71.106.136) はバージョン 7.17.9 である.

そこで今回は, Docker Compose を使用して, 異なるバージョンである 7.17.6 と 7.17.9 の Elasticsearch ノードをクラスタリングすることが可能かどうかを確認するために実施した検証結果について報告する.

## 2 Docker とは

Docker は, コンテナと呼ばれる軽量で独立したコンテナ型仮想環境用のプラットフォームである.

従来の仮想化では, VMWare などの仮想化ソフトウェアを用いて, ホスト OS 上にゲスト OS を構築する形式だった. しかし, Docker はホスト OS 上にゲスト OS なしで独立したコンテナ型の仮想環境として構築される. Docker コンテナを利用する場合は, Docker Engine をインストールすることでコンテナの立ち上げ, 停止, 削除といった操作を行うことができる.

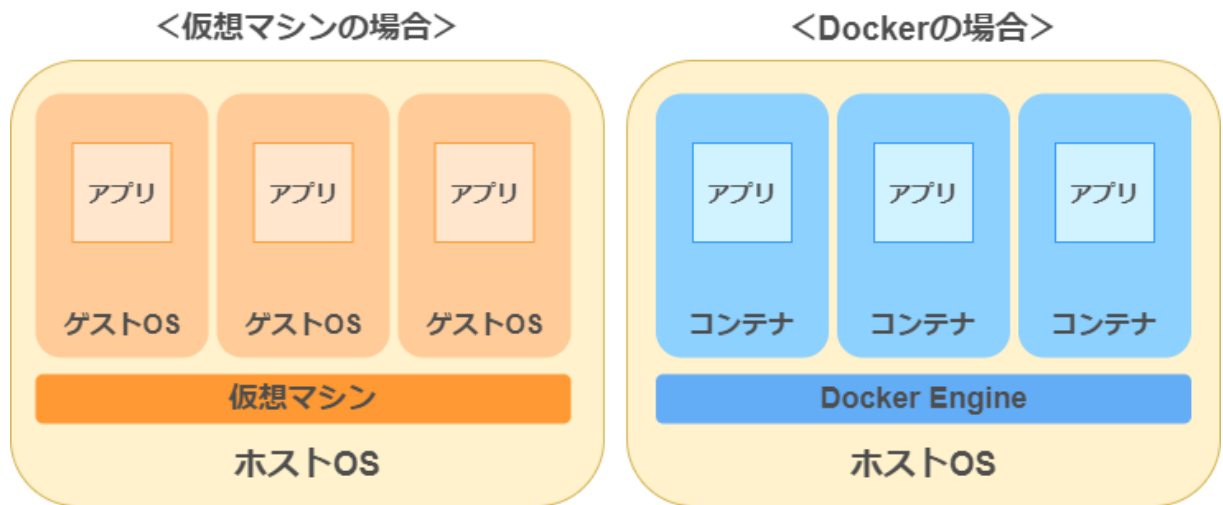


図 1: 仮想マシンと Docker の違い [2]

## 2.1 コンテナとは

コンテナは、アプリケーションとそのすべての依存関係（ライブラリ、実行環境など）をカプセル化した軽量な実行単位である。Docker の場合、コンテナの作成には Docker イメージが必要となる。

## 2.2 Docker イメージとは

Docker イメージとは、Docker コンテナを作成するためのテンプレートであり、アプリケーションの実行に必要なコード、ライブラリ、環境変数、設定ファイルなどを含んでいる。

Docker イメージからコンテナを起動すると、そのイメージのインスタンスが動作し、アプリケーションが実行される。イメージはコンテナの基盤となる静的な部分であり、コンテナはそのイメージを実行時に動的に展開したものと考えることができる。

Docker イメージは Docker Hub やその他のレジストリで共有されており、これらのサービスから取得することが可能である。

今回は Elasticsearch の開発元である Elastic 社が提供している Elasticsearch の Docker イメージを使って検証を行う。

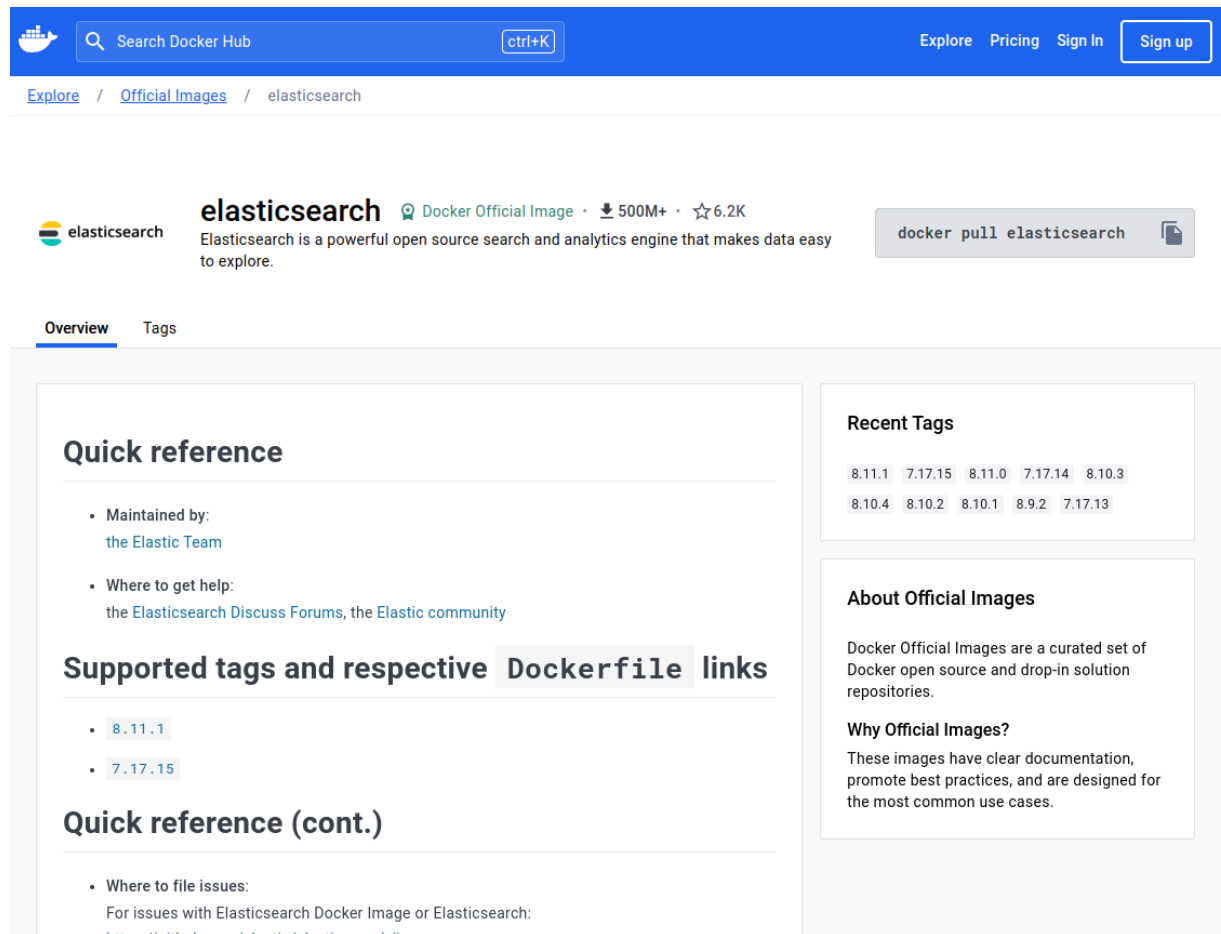


図 2: ElasticSearch の Docker イメージ

### 3 Docker Compose とは

Docker Compose は、複数のコンテナを定義し、実行するためのツールである。これは YAML ファイルを使用して設定され、複数のコンテナで協調して動作するアプリケーションの開発を単純化する。

### 4 検証環境のセットアップ

#### 4.1 全て同じバージョンの ElasticSearch を使用したクラスタ構成 (全ノード バージョン 7.17.9)

Listing 1 にクラスタを構成するのに使用した docker-compose.yml ファイルの内容を記載する。

Listing 1: 全て同じバージョンの ElasticSearch を使用したクラスタを構成する 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       - cluster.name=es-docker-cluster
9       - discovery.seed_hosts=es02,es03
10      - cluster.initial_master_nodes=es01,es02,es03
11      - bootstrap.memory_lock=true
12      - "ES_JAVA_OPTS=-Xms2048m-Xmx2048m"
13     volumes:
14       - data01:/usr/share/elasticsearch/data
15     ports:
16       - 9200:9200
17     networks:
18       - elastic
19   es02:
20     image: docker.elastic.co/elasticsearch/elasticsearch:7.17.9
21     container_name: es02
22     environment:
23       - node.name=es02
24       - cluster.name=es-docker-cluster
25       - discovery.seed_hosts=es01,es03
26       - cluster.initial_master_nodes=es01,es02,es03
27       - bootstrap.memory_lock=true
28       - "ES_JAVA_OPTS=-Xms2048m-Xmx2048m"
29     volumes:
30       - data02:/usr/share/elasticsearch/data
31     networks:
32       - elastic
33   es03:
34     image: docker.elastic.co/elasticsearch/elasticsearch:7.17.9
35     container_name: es03
36     environment:
37       - node.name=es03
38       - cluster.name=es-docker-cluster
39       - discovery.seed_hosts=es01,es02
40       - cluster.initial_master_nodes=es01,es02,es03
41       - bootstrap.memory_lock=true
42       - "ES_JAVA_OPTS=-Xms2048m-Xmx2048m"
43     volumes:
44       - data03:/usr/share/elasticsearch/data
```

```
45     networks:
46       - elastic
47
48 volumes:
49   data01:
50     driver: local
51   data02:
52     driver: local
53   data03:
54     driver: local
55
56 networks:
57   elastic:
58     driver: bridge
```

Listing 1の docker-compose.yml ファイルで記述している内容について説明する。

## サービスの定義

- **es01, es02, es03:** これらは Elasticsearch のノード（サーバー）である。各ノードは異なるコンテナとして定義されている。‘es01‘, ‘es02‘, ‘es03‘はそれぞれ異なるコンテナ名で, Elasticsearch の異なるインスタンスを実行する。

## 各ノードの設定

- **image:** 使用する Docker イメージ。ここでは Elasticsearch の 7.17.9 バージョンを使用している。
- **container\_name:** コンテナに割り当てられる名前。
- **environment:** 環境変数の設定。Elasticsearch のクラスタ設定, メモリ設定などを含む。
- **volumes:** データ永続化のためのボリュームマウント。データをコンテナの外に保存する。
- **ports:** ホストマシンとコンテナ間のポートマッピング。例えば, ‘9200:9200‘はホストの 9200 ポートをコンテナの 9200 ポートにマッピングする。
- **networks:** コンテナ間通信のためのネットワーク設定。ここでは ‘elastic‘ネットワークが使用されている。

## ボリュームとネットワークの設定

- **volumes:** ‘data01’, ‘data02’, ‘data03’はデータの永続化に使用されるボリュームである。これによりコンテナが削除されてもデータが保持される。
- **networks:** デフォルトのドライバである bridge ドライバを使用する ‘elastic’ ネットワークを定義している。これにより、異なるコンテナが相互に通信できるようになる。

この設定により、Elasticsearch の 3 ノードを含むクラスタが Docker 上で動作するようセットアップされる。

クラスタの起動には、docker compose up -d コマンドを使用する。

docker compose up -d コマンドを実行した後、curl コマンドを使用してクラスタに参加しているノードを一覧表示した結果を図 3 に示す。

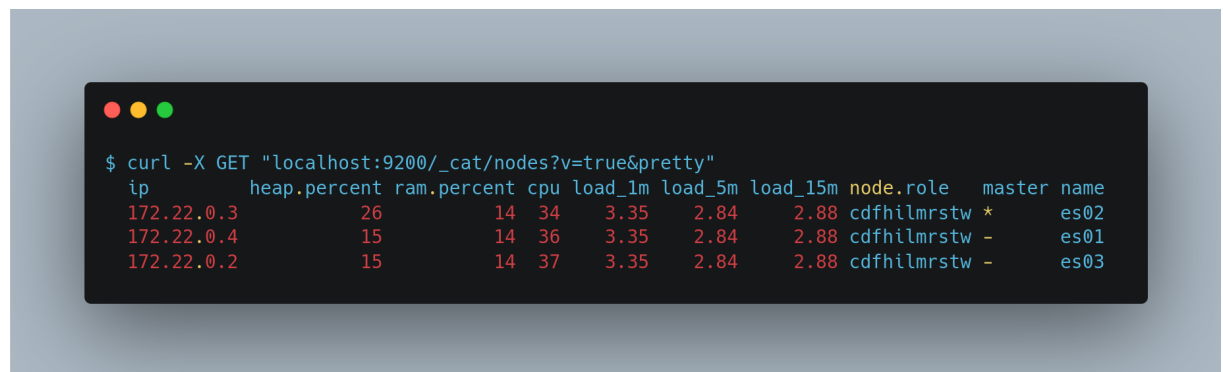


図 3: クラスタに参加しているノードを一覧表示した結果

図 3 より、3 つのノード（‘es01’、‘es02’、‘es03’）すべてが正常にクラスタに参加できていることが確認できる。

## 4.2 異なるバージョンの Elasticsearch を使用したクラスタ構成 (2 ノード バージョン 7.17.9, 1 ノード バージョン 7.17.6)

Listing 1 の docker-compose.yml を Listing 2 のように変更することで、es03 のノードで使用する Elasticsearch のバージョンを 7.17.9 から 7.17.6 に変更する。

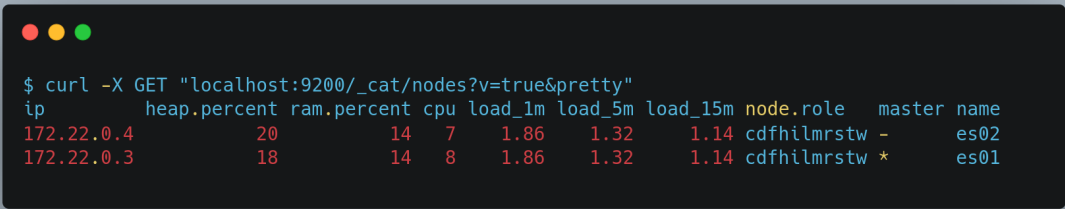
Listing 2: Listing 1 の docker-compose.yml から変更を加えた箇所

```
1 version: '2.2'
2 services:
3   ...
4   es03:
```

```
5 image: docker.elastic.co/elasticsearch/elasticsearch:7.17.6
6 ...
7
8 ...
```

変更後, docker compose up -d コマンドを実行してクラスタを起動する.

クラスタの起動後, curl コマンドを使用してクラスタに参加しているノードを一覧表示した結果を図 4に示す.



```
$ curl -X GET "localhost:9200/_cat/nodes?v=true&pretty"
ip          heap.percent ram.percent cpu load_1m load_5m load_15m node.role master name
172.22.0.4    20          14      7    1.86   1.32   1.14 cdfhilmrstw -   es02
172.22.0.3    18          14      8    1.86   1.32   1.14 cdfhilmrstw *   es01
```

図 4: クラスタに参加しているノードを一覧表示した結果

図 4より, バージョンが 7.17.9 である 2 つのノード ( 'es01', 'es02' ) のみが正常にクラスタに参加できていることが確認できる.

また, Elasticsearch 起動時に出力されたログを確認したところ, 図 5に示すように, クラスタに参加できなかった es03 のコンテナで Elasticsearch がエラーログを出力して終了していることが分かった.

```

es03 | {"type": "server", "timestamp": "2023-11-20T05:23:44,794Z", "level": "INFO", "component": "o.e.p.PluginsService",
"cluster.name": "es-docker-cluster", "node.name": "es03", "message": "no plugins loaded" }
es03 | {"type": "server", "timestamp": "2023-11-20T05:23:44,843Z", "level": "INFO", "component": "o.e.e.NodeEnvironment",
"cluster.name": "es-docker-cluster", "node.name": "es03", "message": "using [1] data paths, mounts
[[/usr/share/elasticsearch/data (/dev/sda6)]], net usable_space [153.4gb], net total_space [206.6gb], types [ext4]" }
es03 | {"type": "server", "timestamp": "2023-11-20T05:23:44,843Z", "level": "INFO", "component": "o.e.e.NodeEnvironment",
"cluster.name": "es-docker-cluster", "node.name": "es03", "message": "heap size [2gb], compressed ordinary object pointers
[true]" }
es01 | {"type": "server", "timestamp": "2023-11-20T05:23:44,872Z", "level": "INFO", "component": "o.e.n.Node",
"cluster.name": "es-docker-cluster", "node.name": "es01", "message": "node name [es01], node ID [xeg-kFwGRMmh8M4u8A-ZEA],
cluster name [es-docker-cluster], roles [transform, data_frozen, master, remote_cluster_client, data, ml, data_content,
data_hot, data_warm, data_cold, ingest]" }
es03 | {"type": "server", "timestamp": "2023-11-20T05:23:44,902Z", "level": "ERROR", "component":
"o.e.b.ElasticsearchUncaughtExceptionHandler", "cluster.name": "es-docker-cluster", "node.name": "es03", "message":
"uncaught exception in thread [main]",
es03 | "stacktrace": ["org.elasticsearch.bootstrap.StartupException: java.lang.IllegalStateException: cannot downgrade a
node from version [7.17.9] to version [7.17.6]",
es03 | "at org.elasticsearch.bootstrap.Elasticsearch.init(Elasticsearch.java:173) ~[elasticsearch-7.17.6.jar:7.17.6]",
es03 | "at org.elasticsearch.bootstrap.Elasticsearch.execute(Elasticsearch.java:160) ~[elasticsearch-7.17.6.jar:7.17.6]",
es03 | "at org.elasticsearch.cli.EnvironmentAwareCommand.execute(EnvironmentAwareCommand.java:77) ~[elasticsearch-
7.17.6.jar:7.17.6]",
es03 | "at org.elasticsearch.cli.Command.mainWithoutErrorHandling(Command.java:112) ~[elasticsearch-cli-
7.17.6.jar:7.17.6]",
es03 | "at org.elasticsearch.cli.Command.main(Command.java:77) ~[elasticsearch-cli-7.17.6.jar:7.17.6]",
es03 | "at org.elasticsearch.bootstrap.Elasticsearch.main(Elasticsearch.java:125) ~[elasticsearch-7.17.6.jar:7.17.6]",
es03 | "at org.elasticsearch.bootstrap.Elasticsearch.main(Elasticsearch.java:80) ~[elasticsearch-7.17.6.jar:7.17.6]",
es03 | "Caused by: java.lang.IllegalStateException: cannot downgrade a node from version [7.17.9] to version [7.17.6]",
es03 | "at org.elasticsearch.env.NodeMetadata.upgradeToCurrentVersion(NodeMetadata.java:95) ~[elasticsearch-
7.17.6.jar:7.17.6]",
es03 | "at org.elasticsearch.env.NodeEnvironment.loadNodeMetadata(NodeEnvironment.java:484) ~[elasticsearch-
7.17.6.jar:7.17.6]",
es03 | "at org.elasticsearch.env.NodeEnvironment.<init>(NodeEnvironment.java:356) ~[elasticsearch-7.17.6.jar:7.17.6]",
es03 | "at org.elasticsearch.node.Node.<init>(Node.java:429) ~[elasticsearch-7.17.6.jar:7.17.6]",
es03 | "at org.elasticsearch.node.Node.<init>(Node.java:309) ~[elasticsearch-7.17.6.jar:7.17.6]",
es03 | "at org.elasticsearch.bootstrap.Bootstrap$5.<init>(Bootstrap.java:234) ~[elasticsearch-7.17.6.jar:7.17.6]",
es03 | "at org.elasticsearch.bootstrap.Bootstrap.setup(Bootstrap.java:234) ~[elasticsearch-7.17.6.jar:7.17.6]",
es03 | "at org.elasticsearch.bootstrap.Bootstrap.init(Bootstrap.java:434) ~[elasticsearch-7.17.6.jar:7.17.6]",
es03 | "at org.elasticsearch.bootstrap.Elasticsearch.init(Elasticsearch.java:169) ~[elasticsearch-7.17.6.jar:7.17.6]",
es03 | "... 6 more"] }
es03 | uncaught exception in thread [main]
es03 | java.lang.IllegalStateException: cannot downgrade a node from version [7.17.9] to version [7.17.6]
es03 |     at org.elasticsearch.env.NodeMetadata.upgradeToCurrentVersion(NodeMetadata.java:95)
es03 |     at org.elasticsearch.env.NodeEnvironment.loadNodeMetadata(NodeEnvironment.java:484)
es03 |     at org.elasticsearch.env.NodeEnvironment.<init>(NodeEnvironment.java:356)
es03 |     at org.elasticsearch.node.Node.<init>(Node.java:429)
es03 |     at org.elasticsearch.node.Node.<init>(Node.java:309)
es03 |     at org.elasticsearch.bootstrap.Bootstrap$5.<init>(Bootstrap.java:234)
es03 |     at org.elasticsearch.bootstrap.Bootstrap.setup(Bootstrap.java:234)
es03 |     at org.elasticsearch.bootstrap.Bootstrap.init(Bootstrap.java:434)
es03 |     at org.elasticsearch.bootstrap.Elasticsearch.init(Elasticsearch.java:169)
es03 |     at org.elasticsearch.bootstrap.Elasticsearch.execute(Elasticsearch.java:160)
es03 |     at org.elasticsearch.cli.EnvironmentAwareCommand.execute(EnvironmentAwareCommand.java:77)
es03 |     at org.elasticsearch.cli.Command.mainWithoutErrorHandling(Command.java:112)
es03 |     at org.elasticsearch.cli.Command.main(Command.java:77)
es03 |     at org.elasticsearch.bootstrap.Elasticsearch.main(Elasticsearch.java:125)
es03 |     at org.elasticsearch.bootstrap.Elasticsearch.main(Elasticsearch.java:80)
es03 | For complete error details, refer to the log at /usr/share/elasticsearch/logs/es-docker-cluster.log
es03 exited with code 1

```

図 5: es03 のログ

## 5 まとめ

今回は、Docker Compose を使用して、異なるバージョンである 7.17.6 と 7.17.9 の Elasticsearch ノードをクラスタリングすることが可能かどうかを確認した。

今回の検証により、全てのノードが同じ 7.17.9 バージョンであればクラスタリン



グを行えることが確認できた。しかし、異なるマイナーバージョン（今回は 7.17.6 と 7.17.9）のノードでクラスタを構築しようとする、古いバージョンのノードはクラスタに参加できなかった。

そのため、リサイクル館の測定データを保存している ElasticSearch をバージョンアップする必要がある、次回は ElasticSearch のバージョンアップ方法について調査した結果を報告する。

## 参考文献

- [1] Elasticsearch B.V.,  
"Install Elasticsearch with Docker — Elasticsearch Guide [7.17] — Elastic ", <https://www.elastic.co/guide/en/elasticsearch/reference/7.17/docker.html>, 参照 Nov 20,2023.
- [2] RAKUS Developers Blog, "Docker とは一体何なんだ？【初心者向け】 - RAKUS Developers Blog — ラクス エンジニアブログ ", <https://tech-blog.rakus.co.jp/entry/20221007/docker>, 参照 Nov 20,2023.