

# Elasticsearch の基礎

2019-04-23

# トピックス

1. Elasticsearch とは
2. Elasticsearch の基本概念
3. master node と data node
4. mapping
5. (near) リアルタイムとは

# Elasticsearch とは

特徴 (「データ分析基盤構築入門」より、ただし太字は独自)

- OSS (Apache Lisence v2)
- **ドキュメント指向**
- **分散システム**
- マルチテナント
- **RESTful API**
- **(near) リアルタイム**

# ドキュメント指向

- ドキュメント (document)
  - Elasticsearch で扱うデータの最小単位
  - JSON
  - id を持つ
- (スキーマレスに使えるので) 柔軟なデータの登録が可能

# 分散システム

- データを複数 node 間で分散して保持、検索する
- スケールアウトを想定した設計

# RESTful API

- あらゆる操作が REST API で提供されている
  - document の追加、更新、削除
  - document の検索
  - 設定の確認、変更
  - メトリクスの取得

# RESTful API

- 例. document の追加

```
readonly ES_INDEX=index1
readonly ES_TYPE=_doc

request=$(cat <<EOF
{
  "name": "Alice",
  "age": 21,
  "registered_at": "2019-04-23T03:00:00Z"
}
EOF
)

curl -XPOST -H "Content-Type: application/json" \
  http://$ES_HOST:$ES_PORT/$ES_INDEX/$ES_TYPE \
  -d "$request"
```

# (near) リアルタイム

- 登録したデータはほぼリアルタイムに検索結果に反映
- デフォルト設定だと 1 秒後には反映されるっぽい



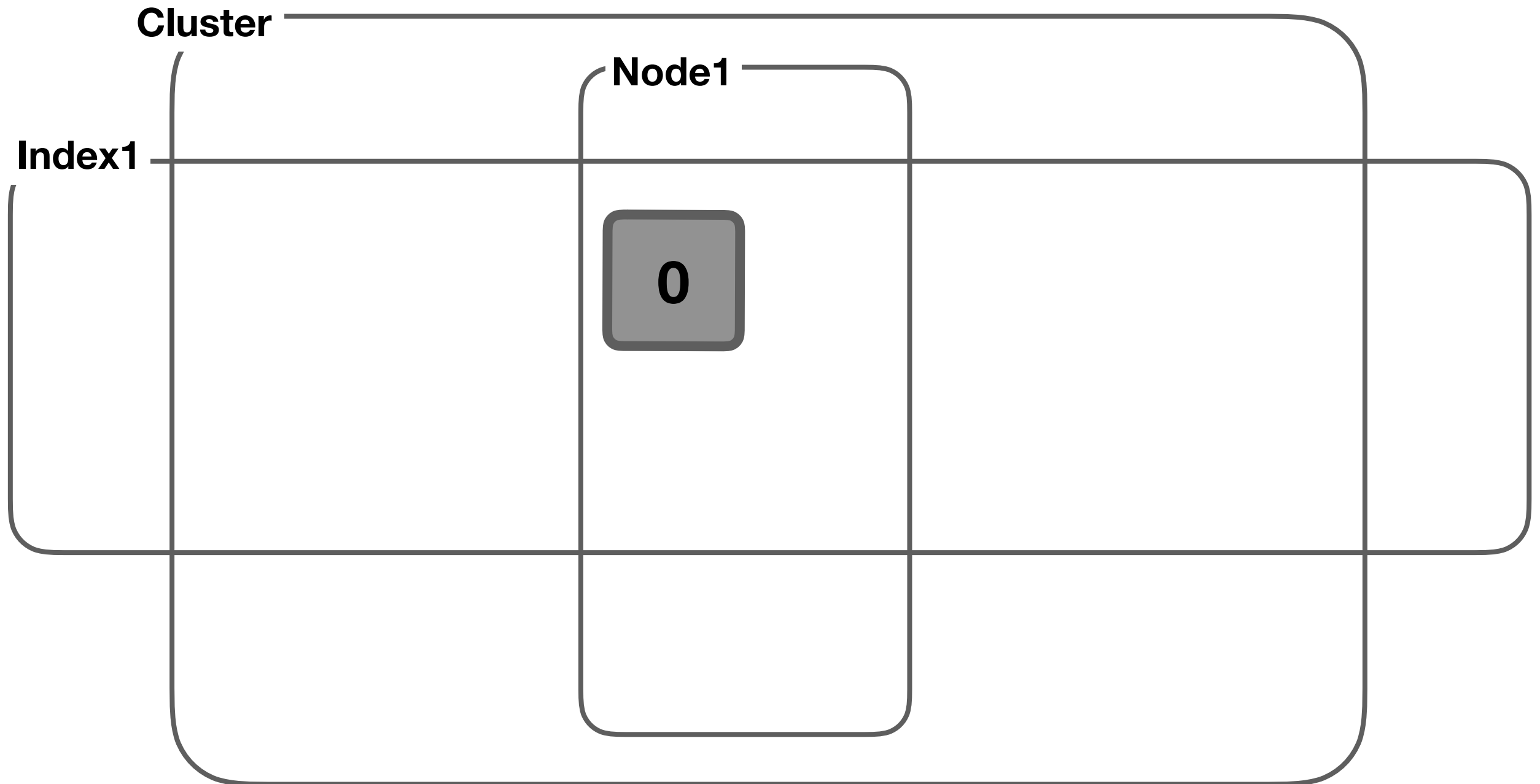
# トピックス

1. Elasticsearch とは
2. Elasticsearch の基本概念
3. master node と data node
4. mapping
5. (near) リアルタイムとは

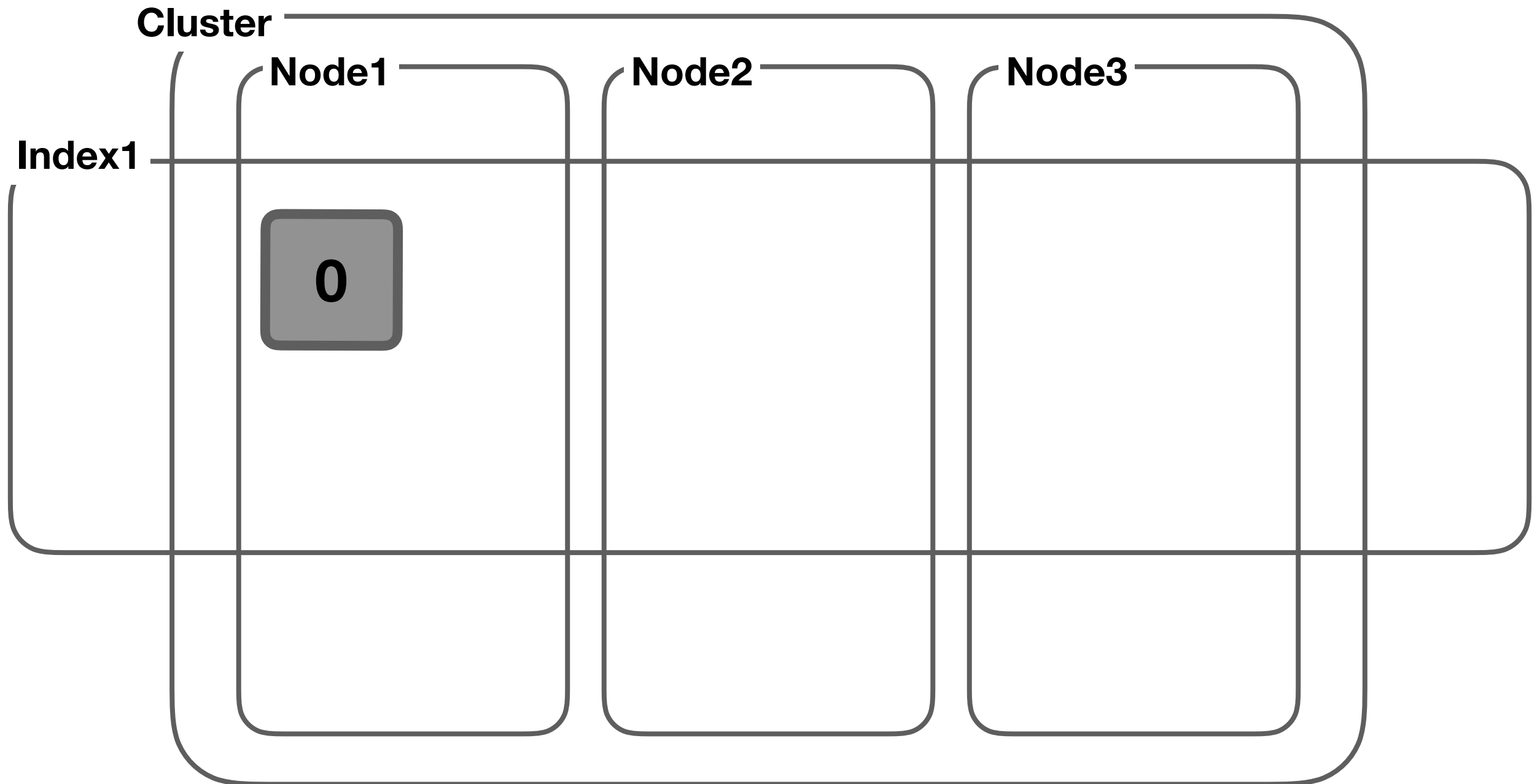
# Elasticsearch の基本概念

- Elasticsearch は 1 台以上の **node** で **cluster** を構成する
- cluster には 0 以上の **index** が存在する
- index には 1 つ以上の **shard** が存在する
- document はいずれかの shard に保存される
  - client は index を指定して保存する
  - 保存先の shard は id (正確には routing) によって決まる

# 1 node, 1 index, 1 shard

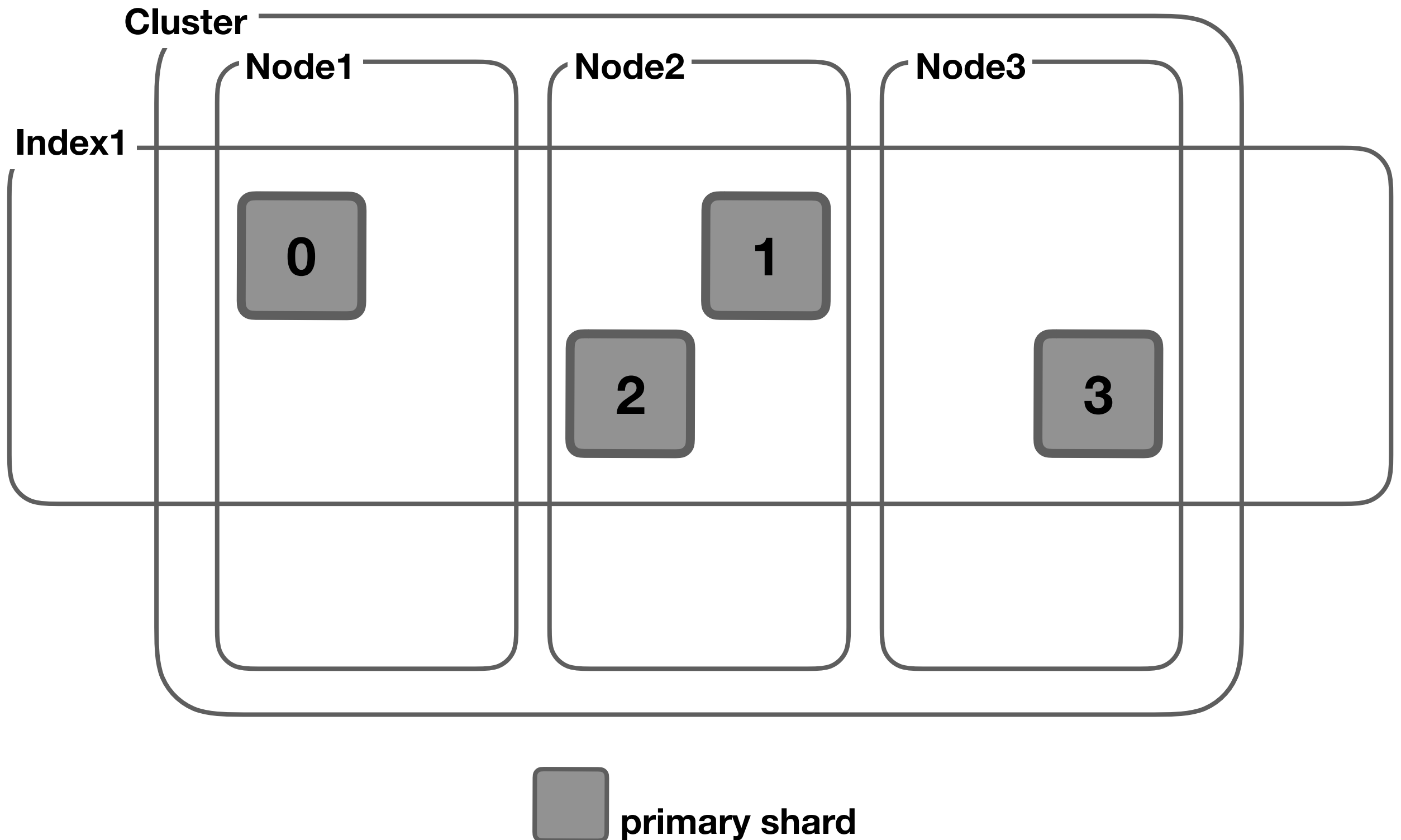


# 3 node 構成に

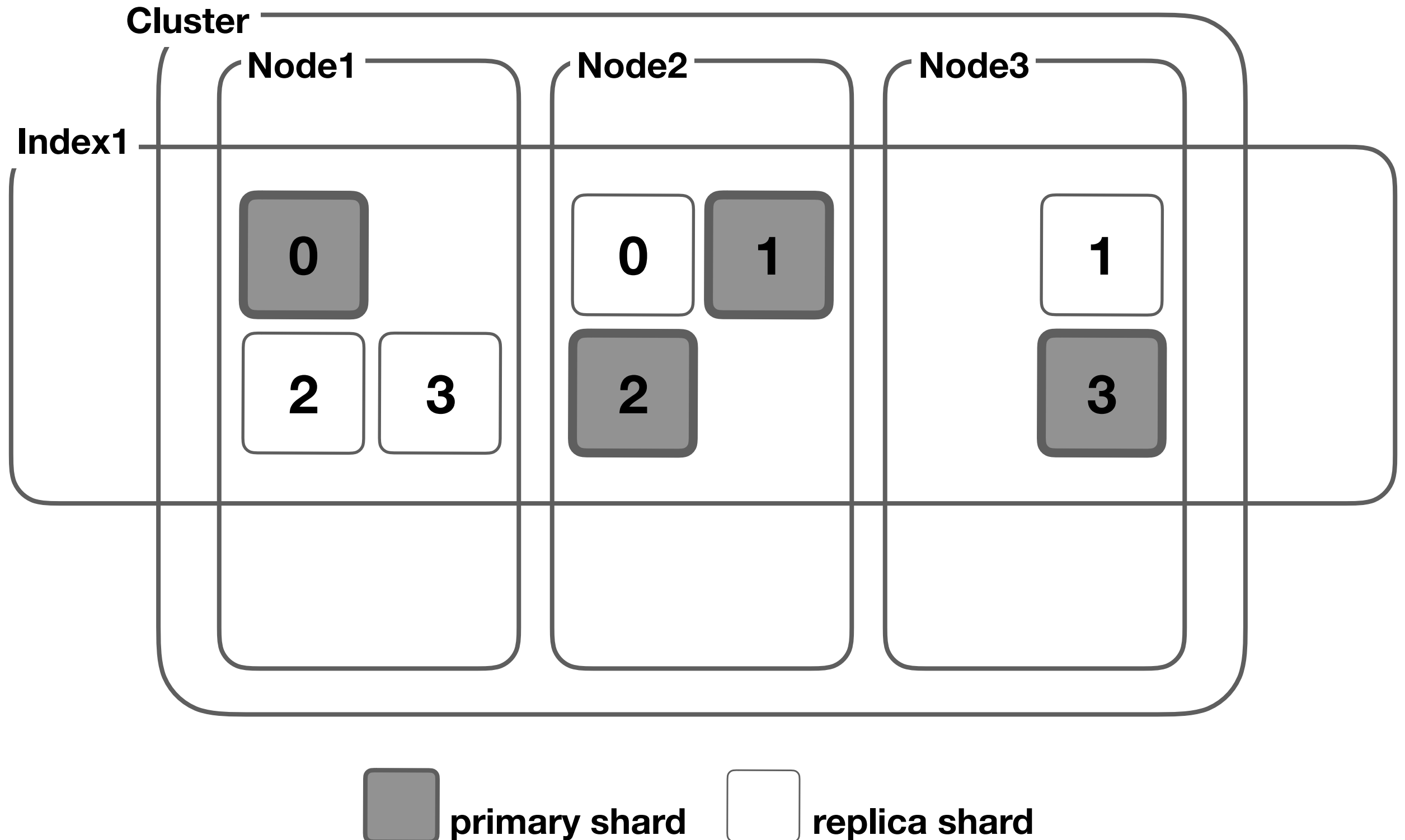


 **primary shard**

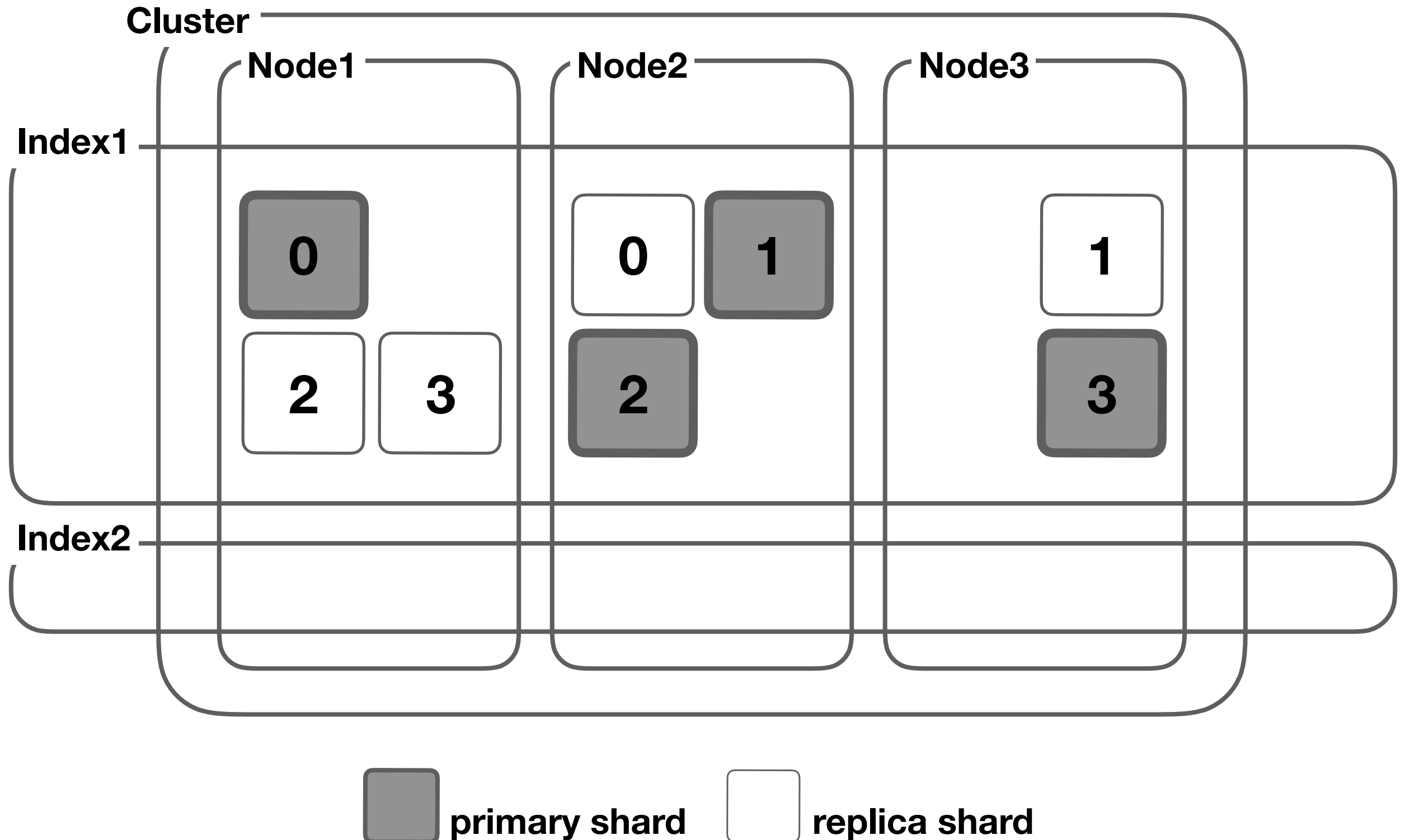
# shard 数を 1 -> 4 に



# 各 shard に 1 replica 追加



# index 追加



# 稼働中のシステムでは？

- 9 node で cluster を構成
- 1 index のみ, 128 primary shards
  - 適切な primary shard 数は扱うデータ等によって異なる
  - 動的に primary shard 数を変更することはできない
- replica 数は 1 (各 primary shard に 1 replica 存在)
- id, routing は独自に設定



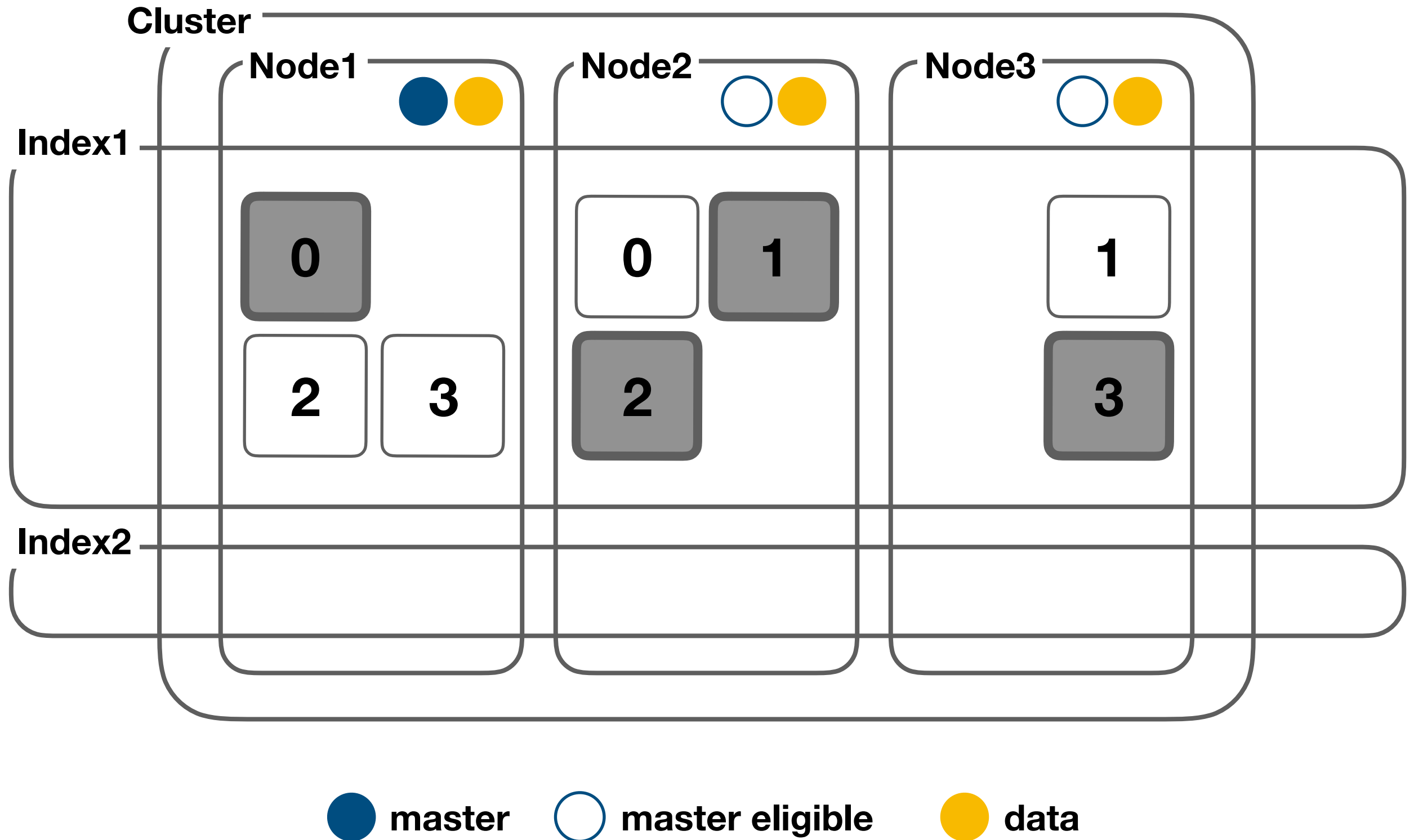
# トピックス

1. Elasticsearch とは
2. Elasticsearch の基本概念
3. master node と data node
4. mapping
5. (near) リアルタイムとは

# master node と data node

- 各 node は起動時に役割を設定される
- 役割1: master eligible (master 候補)
  - この nodeの中から master が一台選ばれる
- 役割2: data (document の保存先)
  - 割り当てられた shard への document の保存、検索を担当
- master eligible, data は兼任できる

# master node と data node



# 稼働中のシステムでは？

- master eligible node が 3 台
- data node が 6 台
- master eligible と data を兼任した node はいない
  - 本番環境では分けることが推奨されている
  - master に余計な負荷をかけないように

# トピックス

1. Elasticsearch とは
2. Elasticsearch の基本概念
3. master node と data node
4. mapping
5. (near) リアルタイムとは

# mapping

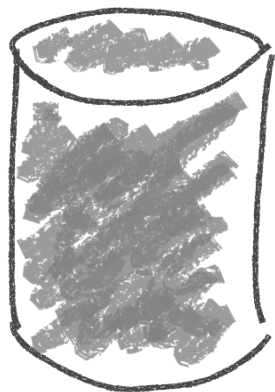
- index 設定の一部
- document をどのように保存するか、index するか
  - フィールドの型 (例. text, keyword, long, date)
  - analyzer の設定
- document に応じて自動で生成される
- 事前に定義しておくこともできる

# mapping

- 自動で生成される mapping の例 (Elasticsearch v7.0)

```
{  
  "name": "Alice",  
  "age": 21,  
  "registered_at": "2019-04-23T03:00:00Z"  
}
```

**POST /index1/doc**



**Elasticsearch**

**GET /index1/mappings**

```
{  
  "index1": {  
    "mappings": {  
      "properties": {  
        "age": {  
          "type": "long"  
        },  
        "name": {  
          "type": "text",  
          "fields": {  
            "keyword": {  
              "type": "keyword",  
              "ignore_above": 256  
            }  
          }  
        },  
        "registered_at": {  
          "type": "date"  
        }  
      }  
    }  
  }  
}
```

# mapping

- 自動で生成される mapping の例 (Elasticsearch v7.0)

age の型は long

name の型は text (全文検索用)

name.keyword で keyword  
としても扱える

```
{
  "index1": {
    "mappings": {
      "properties": {
        "age": {
          "type": "long"
        },
        "name": {
          "type": "text",
          "fields": {
            "keyword": {
              "type": "keyword",
              "ignore_above": 256
            }
          }
        },
        "registered_at": {
          "type": "date"
        }
      }
    }
  }
}
```



# 稼働中のシステムでは？

- mapping は事前に定義している
- 文字列はすべて string 型で not\_analyzed として保存
  - 検索時には保存した文字列の完全一致、部分一致検索
  - 全文検索のような用途をするフィールドはない
  - (v7.0 では keyword 型にあたる設定)

# 稼働中のシステムでは？

- タグについては dynamic\_templates を使用

```
{
  "index1": {
    "mappings": {
      "dynamic_templates": [
        {
          "string-double-tags": {
            "path_match": "tag.*",
            "mapping": {
              "fields": {
                "double": {
                  "type": "double"
                }
              }
            },
            "type": "keyword"
          }
        ]
      }
    }
  }
}
```

# トピックス

1. Elasticsearch とは
2. Elasticsearch の基本概念
3. master node と data node
4. mapping
5. (near) リアルタイムとは

# (near) リアルタイムとは

- ユーザが document の保存を依頼すると...
  1. in-memory buffer に書き込み (ここでレスポンス返却)
  2. **refresh** (ここで検索に反映。CRUD はもっと早い?)
  3. flush (ここで永続化される。translog をクリア)
  4. merge (過去の segment をまとめてディスク削減)

注意: v2.x での説明を参考にしている

# refresh\_interval

- index 設定の refresh\_interval で refresh 間隔を設定可能

```
# GET /index1/_settings
{
  "index1": {
    "settings": {
      "index": {
        "refresh_interval": "30s",
        "number_of_shards": "3",
        "provided_name": "index1",
        "creation_date": "1555723945152",
        "number_of_replicas": "0",
        "uuid": "3lJEo2YZQWmNzSKjZKDNFw",
        "version": {
          "created": "7000099"
        }
      }
    }
  }
}
```

# 資料まとめ

- <https://github.com/tiqwab/slides/tree/master/elasticsearch>